It's time for your computer to grow up.

Meet PractiCalc.™ The world's most versatile spreadsheet at only $40.*

Games are fun when it's time to play. But at heart, your Commodore 64™ or VIC-20™ is a full-grown computer straining to gallop at the touch of your fingers.

But the problem is, most software's been designed for toys. Not for real computers. Until now.

Commodore 64 and VIC-20. Not just for games...

Now comes PractiCalc. And there's not much in the world of grown-up computing it won't do. It'll keep addresses (in alphabetical or numerical order), phone numbers (likewise), make budgets and menus, project profits, keep track of expenses, inventories, investments, what have you. For your home, your school, your business.

Budgets, inventories, projections, you name it...

Spreadsheet analysis...

PractiCalc gives you everything you'd expect from a spreadsheet, like adding and subtracting, multiplying and dividing, calculating square roots, logarithms, exponential numbers and even trig functions. And PractiCalc gives you a lot more.

Want to make a chart? Hit a key and the high or low resolution graphics of PractiCalc Plus or PractiCalc 64 will turn numbers into graphs. It'll even print out everything that shows on the screen.

High or low resolution graphics...

Want a lot of facts and figures at your fingertips? PractiCalc stores more than 2,000 cells in up to 100 columns and 250 rows. It'll sort them, search them, or shuffle them for you.

You can add entries, delete them, or move them around –numerically or alphabetically, and instantly, of course. And you'll find PractiCalc unusually friendly, flexible, and forgiving.

High-speed and wild card search...

All of which is a far cry from just playing with a joystick. Still, PractiCalc has one thing in common with a computer game. The price. Only $40.

Alpha or numeric sort...

PRACTICALC™ by COMPUTER SOFTWARE ASSOCIATES, INC.

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

*$40 is suggested retail price for tape version of PractiCalc-20 ($45 for disk version; PractiCalc Plus or PractiCalc 64 = $50 for tape version, $55 for disk).

How To Make Custom Characters On The 64

Gary Davis

Before reading this article, be sure to see "Introduction to Custom Characters For VIC And 64" in this issue, especially if you're unfamiliar with the concepts of redefined characters. The following article includes "Chred 64," a character-editing utility that makes the task of customizing characters easy and fun.

The Commodore 64 allows you to change any character in the character set to suit your own needs. In order to understand how this is done, it is first necessary to understand how the 64 (and most other computers) store the character set.

If you look closely at the letters the computer puts on the screen, you will notice that each character is made up of little dots in an 8×8 grid (see the figure).

Since there are 64 possible dots, or *pixels*, that can be either on or off, we need 64 "switches" for each character. This is done by using eight memory locations for each character. Since one memory location, or *byte*, is divided into eight *bits*, using eight bytes gives us the 64 switches we need for each character.

The bytes for each character are stored consecutively, with the first byte for each character representing the top row of dots in the character, the second byte the second row of dots, and so on. For a pixel to be on, the bit at its location must be *set*; for a pixel to be off, the bit must be *clear*. This is not as complicated as it sounds. The figure shows how the bit patterns of sets and clears are converted into the numbers that represent the character. When you make a series of bytes for every character and store them in a computer, you have what is known as a *character generator*.

Relocating The Character Set

The character generator in most computers, including the Commodore 64, is stored in Read Only Memory (ROM). This way the computer is ready to display characters on the screen as soon as it is turned on.

Unfortunately, when the character generator is in ROM, you can't change the characters to suit your needs. When you can't change the existing character set, the simplest way to customize a new character set is to move it to Random Access Memory (RAM), and then tell the computer to use your character set rather than the one it has in ROM.

Telling the Commodore 64 where the new character set is located is relatively simple to do. Within the video controller chip (sometimes known as the VIC-II chip) is a special memory location that allows you to set a new character pointer (the location of the first byte of your character set).

Now let's try an experiment. Type: POKE 53272,19 and press RETURN. Your screen will be

Pixel Pattern For Letter A

The character codes for A are stored in memory as: 24,60,102,126,102,102,102,0

For each row add the number at the top of each column in which a pixel is turned on. The resulting numbers are then stored consecutively from top to bottom.

NEW FOR COMMODORE

Davka Corporation is pleased to offer its popular games and educational programs for Commodore 64 computers—with one disk drive. Look for this software at your favorite computer store or order direct from us. Write for a free catalog. Gift certificates, from $25, are also available.

IN TIME FOR HOLIDAY GIFT-GIVING!

From Davka's Entertainment Division, two new releases:

I.Q. BASEBALL The only baseball facts answer game! With realistic animation, sounds and music, you'll swing away, getting hits if you know answers about baseball lore and legend. "Who hit the most homers in a 162-game season? If a ball hits the foul pole, it's a foul ball—True or False?" The perfect game for baseball buffs of all ages. **$24.95**

FUZZBALL You scoot around cleaning up fuzz balls while avoiding obstacles. Be careful or you'll slam into the walls! Clean up for a high score before your time runs out. A fast-action arcade-style game. **$24.95**

From Davka's Educational Division, some current releases:

BIBLE BASEBALL An exciting way to see how much you know about the Old Testament—while you have fun learning even more! When your computer asks a question, your answer could get you a single, double, triple, or a home run—with your players racing around the bases! **$24.95** (supplemental disks available with additional questions, as well as teachers disks to create your own questions)

GAME OF THE MACCABEES This program provides useful and interesting information about the Festival of Chanukah, combined with an exciting arcade type combat game. At any stage in the battle, the player may request new questions and win fresh troops for the "army" of Maccabees. Designed for high school students and adults. **$24.95**

ALL ABOUT CHANUKAH This program is designed to teach and explain the laws and customs of Chanukah. There are six complete sections—with everything from visual instructions on blessing the candles to an animated dreidel game. **$29.95**

MasterCard VISA

Credit card holders may order toll-free:

1-800-621-8227

In Illinois call 1-312-944-4070

(add $2 for shipping and handling)

Davka™

CORPORATION

845 North Michigan Avenue • Suite 843
Chicago, Illinois 60611

filled with strange characters, but don't worry. You have told the 64 to use a RAM character generator, but you haven't supplied one yet. To return your screen to normal type: POKE 53272,21 and press RETURN. You won't be able to read what you are typing until you press RETURN, but the computer understands. If this doesn't work, you can always restore the screen by pressing the RESTORE and RUN/STOP keys at the same time.

When you are designing a new character set, it is nice to have the normal one loaded into RAM to start with. Then you can make changes to it. Program 1 copies the 64's character set from ROM to RAM.

Before you type in this program, *you must enter:*

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

This saves a place in RAM memory for your new character set and protects it from being overwritten by a BASIC program.

Now, type in the program and RUN it. After about 45 seconds the computer will come back and say READY. Now type POKE 53272,19 and press RETURN. Nothing appears to happen, but the characters you are now seeing on your screen are coming from your RAM character generator, not from ROM as usual.

To test this, type POKE 2056,255. The top of all the letter A's on the screen should now be a solid line. Try POKEing different numbers into memory locations between 2048 and 6143 and watch the results on the characters.

Using A Character Editor

By sketching an 8×8 grid as seen in the figure, it is possible to map out the entire character set on graph paper and convert your new characters to numbers to POKE into memory.

This method, however, can be both time-consuming and frustrating. A far better way is to create your new characters on the screen and let the computer do all the calculations. With this thought in mind, I wrote a character editor called "Chred 64." With this utility (Program 2), you can redefine any of the text or graphics symbols and save them on tape or disk. This can then be loaded and used with any program.

In order to reserve memory for the alternate character set, it is necessary to set the start of BASIC pointer to 8192. This will leave you with 32K of RAM free for your BASIC program. To do this, *you must* type in the following:

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

"Chred 64," a utility program for creating custom characters on the Commodore 64.

Now the memory from 2048 to 8191 is free to hold your new character set. You may type in or LOAD Chred 64. After typing Chred 64 for the first time, be sure to SAVE it on tape or disk before you RUN it. If you have made a typing error, it is possible that the computer will "crash" and you'll have to type it all over again if you haven't SAVEd a copy.

When you run Chred 64, the program first copies the resident character set from ROM to RAM and resets the character base to point to the RAM character set. The program then expands the current character being edited to eight times its normal size.

To edit the current character being displayed, you may use the cursor control keys, the asterisk, and the space bar.

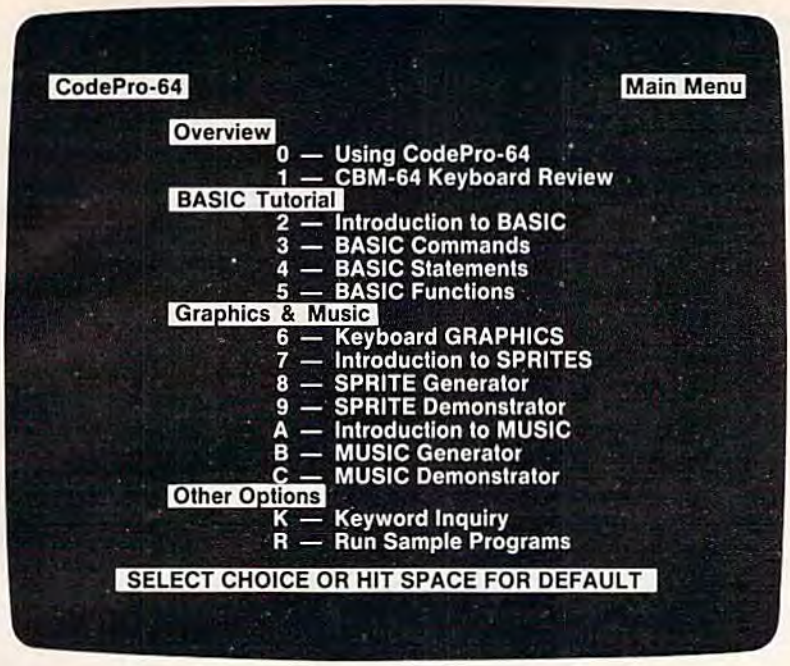
To turn on a pixel, position the cursor and press the asterisk. To turn off a pixel, press the space bar. To clear the entire character, press CLR.

To edit a different character, press F1. You will be asked to supply a row and column. This refers to the block of characters displayed on the lower-right corner of the screen. Just type a row number followed by the column number or letter. The character you selected will now be displayed, ready for you to edit.

More Editing Features

An interesting feature of the 64 is that, unlike the Commodore PET, the reverse-field (inverse video) characters are stored as part of the character set. This allows 256 redefined characters. To edit a character not being displayed, press F3. This will select and display the next block of 64 characters. Rest assured that you may mix characters from any of the blocks; only 64 characters are shown at a time for the purpose of editing.

Sometimes you may wish to edit more than



NEW! For the Commodore 64™ ANNOUNCING

CodePro-64™

A new concept in interactive visual learning . . .

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

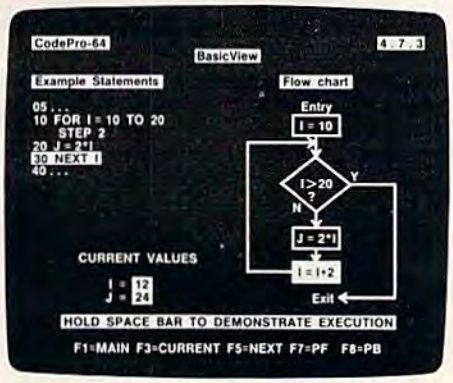
SEE PROGRAM EXECUTION

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

After entering an example you invoke our exclusive **BasicView™** which **shows you how the BASIC program example executes.**

You step through and actually see the execution of sample program statements by simply pressing the space bar. CodePro-64 does the rest.

You see statements with corresponding **flow chart graphics** and variable value displays. You learn by visual examples.



EXTENSIVE TUTORIAL

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

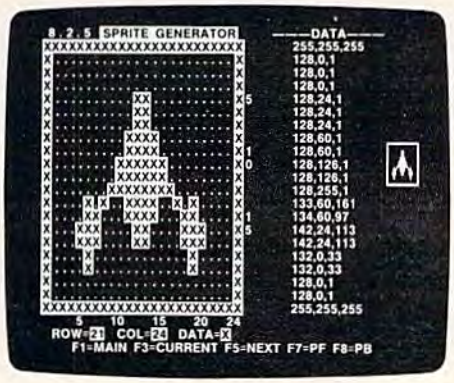
By seeing graphic displays of program segment execution you learn by visual example. You **learn faster and grasp programming concepts easier** with CodePro-64 because you immediately see the results of your input.

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

CodePro-64 **lets you follow your interests** and practice with interactive examples. But you can never get "lost". F1 will always return you to the main menu. Once you have practiced and mastered the BASIC language elements you move on to more advanced concepts. You learn about sprite and music programming.

SPRITE GENERATOR & DEMONSTRATOR

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



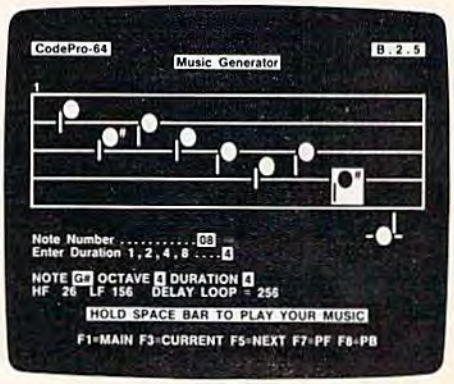
We also help you learn to program with sprites by giving you a **sprite demonstrator** so you can see the effect of changing register values. You can experiment by moving your sprite around in a screen segment, change its color or priority, and see the effects of your changes. You learn by visual examples.

MUSIC GENERATOR & DEMONSTRATOR

To teach you music programming CodePro-64 gives you an interactive music generator and demonstrator. First we help you set all your SID parameters (attack/decay, sustain/release, waveform, etc.). Then you enter notes to play and **we show your tune graphically as it plays,** note by note, on the scale. You learn by seeing and hearing the results of your input.

OUR GUARANTEE

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



Our music demonstrator **lets you experiment** with various combinations of music programming parameters and hear the results. You can **quickly modify any of the SID register values** to hear the effects of the change. For example, you could easily change waveform and attack/decay values while holding all other SID values constant. By seeing your input and hearing the result you quickly learn how to create new musical sounds and special sound effects.

AND MORE . . .

We don't have enough space to tell you everything CodePro-64 offers. You need to see for yourself. BASIC tutorials, graphics, sprites, music, keyboard review, sample programs—the main menu shown above gives you just a summary of the contents of this powerful educational product.

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

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

HOW TO ORDER

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

SYSTEMS MANAGEMENT ASSOCIATES
3700 Computer Drive, Dept. CA
Raleigh, N.C. 27609

Please **specify tape or diskette** version. MasterCard/VISA accepted. For faster service on credit card orders call (919)-787-7703.

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

Ad f[illegible] **Dealer inquiries invited.**

one character at a time to make a larger shape. This can be easily accomplished by pressing F5. Instead of a single character, you will be able to edit a block of four characters. To go back to single character mode, just press F5 again.

After you have redefined several characters, the text on the screen may become unreadable as your new characters replace the existing ones. To restore the character set to normal, *without* destroying your new character set, press F7. To return to your new character set, press F7 again.

When you are done working with a character set, you can restore the font to the normal character set by pressing R. You will be asked "Are you sure?" Now is your last chance to save your character set. If you are really done, press Y; otherwise, press N.

After you have gone to the effort of creating a new character set, you will probably want to save it on disk or tape for use in other programs. To save your character set, press S. Follow the directions given on the screen. After the character set is saved, you will be returned to the editor. (When typing Chred 64, omit line 225 for use with tape.)

Sometimes you may wish to alter a character set that you have already created and saved. To load another character set, press L and follow the directions given on the screen. Be careful — the new character set is loaded on top of the current character set, so be sure to save it if you want to use it later.

OK, you've developed your new character set. To use it with another program, you will have to type POKE 8192, 0:POKE 44, 32: NEW, just as you do when you load Chred 64. To load in the character set, place the cassette containing your new character set in the recorder, or the disk in the drive. For tape, type LOAD "filename",1,1 where "filename" is the name you gave when you saved the character set. For disk, type LOAD"filename",8,1. To use the new character set, POKE 53272, 19. To return to the normal character set, POKE 53272, 21.

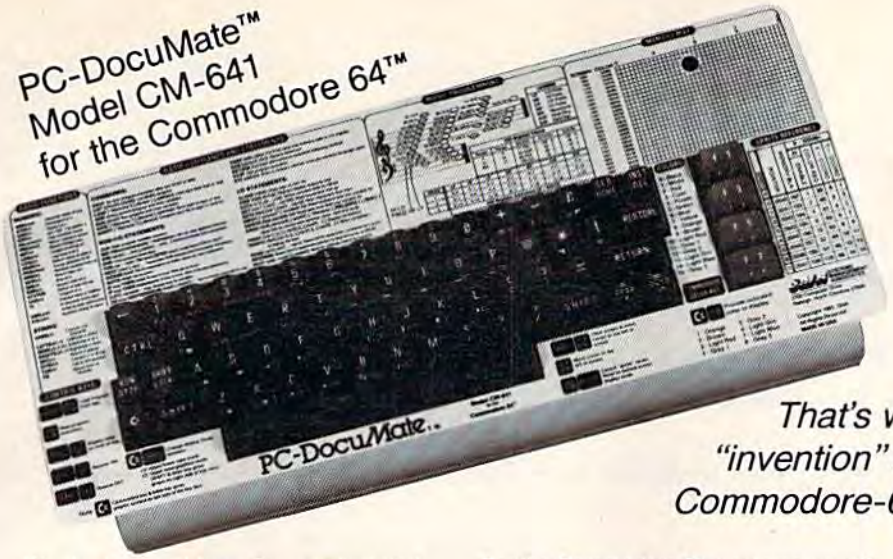
I hope you have as much fun using this program as I had writing it. I would like to express my thanks to Don Moses for the use of his 64 in developing this program, and to the Central Ohio PET Users Group for providing detailed maps of the video chip registers.

If you do not want to type in this program yourself, please send a blank disk or cassette tape along with a stamped, self-addressed mailer and $3 to the address below. I will send you a copy of Chred 64 and a character set developed with it.

Gary Davis
1147 Carbone Drive
Columbus, OH 43224

See program listings on page 196.

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



EUREKA!

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

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

INCONVENIENT MANUALS

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

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

SIMPLE SOLUTION

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

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

STILL CONFUSED

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

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

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

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

GETTING EASIER

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

But we have a confession to make.

WE CHEATED

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

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

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

SOME SPECIFICS

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

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

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

OUR GUARANTEE

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

SOLVE YOUR PROGRAMMING PROBLEMS WITH PC-DocuMate™

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

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

Check ___ Money order ___ MC/VISA ___

Name _____

Address _____

City _____ State _____ Zip _____

Card # _____ Exp. _____

Signature _____

Foreign orders (except Canada) add $5.00 US

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

Canadians: Please send $18.95 CDN for each template to:

Systems Management Associates
55A Westmore Dr., Dept. J-1
Rexdale, ONTARIO M9V3Y6

Dealer inquiries invited.

How To Make Custom Characters On The VIC

Gregg Keizer, Assistant Book Editor

Before reading this article, be sure to see "Introduction To Custom Characters For VIC And 64" in this issue, especially if you're unfamiliar with the concepts of redefined characters.

The standard characters provided with the VIC are certainly useful, afford plenty of variety, and can be combined to create new shapes and figures. Many games on the VIC, for instance, often use only the standard character set to display the screen and show objects or user-controlled figures.

But there will be times when you need to draw a new shape or figure that the standard character set just can't produce. You'll often find this true as you design your own games. Or perhaps you simply want to experiment, to see what you can do with the VIC.

Creating custom characters takes up more memory, which can be crucial when you're using the VIC, and it can take time to design and add them to a program. But when you're looking for just the right figure, and it shows on the screen during a game, you'll agree that it was worth the effort.

Fooling The VIC

You've already looked through the article "Introduction to Custom Characters," and perhaps read "The Inner World of Computers, Part 1: Binary Numbers" elsewhere in this issue, so you know what custom characters are and how to design them using graph paper. Now that you have the figures in mind, you can actually begin to place those custom characters in the VIC.

Remember that the character set of most computers, including the VIC, is located in Read Only Memory, or ROM, and is permanently stored there. The VIC's character memory begins at location 32768, which stores the number value of the top row of the @ character. The number value of the second row of that character is at location 32769, the third row at location 32770, and so on.

In order to change the character set and insert your own custom characters, you first need to change the place where the VIC looks to find its character set. You can do this by changing the *pointer*, which fortunately is in a Random Access Memory (RAM) location. By changing this memory location, you are in effect instructing the computer to look elsewhere for its character set.

The VIC looks to location 36869 for its pointer. Although the pointer's value is usually 240 or 242, it can be changed by POKEing a new value into that location. Entering POKE 36869,255, for example, fools the computer into looking to a new location in RAM, 7168, for character data, instead of the ROM location 32768. You can begin your custom character set in a RAM location ranging from 4096 to 7168, but the best place to start is at 7168.

Right now, however, there is nothing there for the VIC to look at. You need to copy the character set data to your new RAM location. As explained in "Introduction to Custom Characters," this isn't that hard.

What you need to do is free enough space from BASIC to fit in your recopied character set, as well as protect it from the BASIC's operations. Then you have to tell the VIC to read the numbers in the character set stored in ROM and copy those numbers starting at your new RAM location, 7168.

A short routine such as the one below does all the work for you in only three program lines.

```
10 PRINT"{CLR}":POKE 36869,255
20 POKE 52,28:POKE 56,28:CLR
30 FOR I=7168 TO 7679:POKE I,PEEK(25600+I
   ):NEXT
```

COMPUTE!'s First Book Of VIC

Authors: **COMPUTE!** Magazine contributors
Price: $12.95
On Sale: Now

Finally, it's VIC's turn!

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

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

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

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

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

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

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

Line	Function
10	Tells the VIC to go to location 7168 to get the data for its character set, rather than looking to ROM location 32768.
20	Frees 512 bytes of memory from BASIC by changing the pointers to the top of available RAM memory. A value of 28 takes 512 bytes, just enough for the 64 characters you would normally copy to RAM, and also protects this area from BASIC.
30	Copies the first 64 characters from their ROM locations to your new RAM location. This is done by PEEKing at the values from 32768 to 33280 (PEEK 25600+I), and then POKEing those values into the new locations running from 7168 to 7679. (We're moving only 64 characters out of a total character set of 256 in order to conserve memory.)

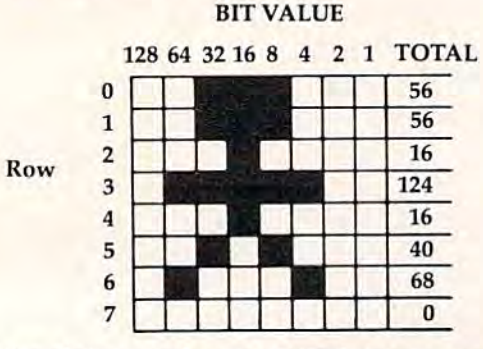
Now that you have part of the character set moved to RAM, the VIC told to look there from now on for its character data, and the area protected from BASIC, you can begin to place your own characters in this set.

Placing Your Custom Characters

Your custom characters have already been designed, either using graph paper or perhaps with the "VIC-20 Character Developer" utility program (see sidebar). You've added up the dot values and should have eight numbers for each new character. These represent the eight bytes of memory each character requires to be displayed on the screen.

What you now need to do is POKE these new numbers where the old values are, replacing one of the standard characters with one of your custom characters. For example, a custom character such as that in the figure would have the following values:

Custom Characters



The eight numbers to POKE into a memory location are 56,56,16,124,16,40,68, and 0 to create this character.

The most convenient way to replace old characters is with DATA and READ statements. By placing all the new numbers into DATA statements, and then having the VIC READ them, your programming task will be simplified. The computer always READs the DATA in the order it's listed, so be sure the numbers are in the right order, and that there are eight numbers for each character. The DATA statements can be anywhere in the program, as long as they are in the same order as the READ commands.

Glance through the Screen Code table in *Personal Computing on the VIC*, the guide that came with your computer. You've copied the first 64 characters, from the @ to the ?, into RAM, but you'll lose some of these when you replace them with your custom characters. Decide which standard characters you won't need, and look up the screen codes for those. If you are designing a game that uses some of the letters in a display, for example, make sure those won't be lost when you develop your custom characters.

Because you changed the pointer, your character set now begins at memory location 7168. Each character takes up eight bytes of memory, so by multiplying the screen code number by 8 and adding it to 7168, you can find the location of the top row of any character. For example, the letter A, with a screen code value of 1, begins at location 7176.

A format you can use to replace a standard character with your own custom character is:

```
FOR C(your new character)=X TO X+7:READ
D:POKE C,D:NEXT
```

where X is the memory location of a character you want to replace. To replace the A character with the custom character from the figure, for instance, you would write:

```
FOR C=7176 TO 7183:READ D:POKE
C,D:NEXT
```

and include somewhere in the program the DATA statement:

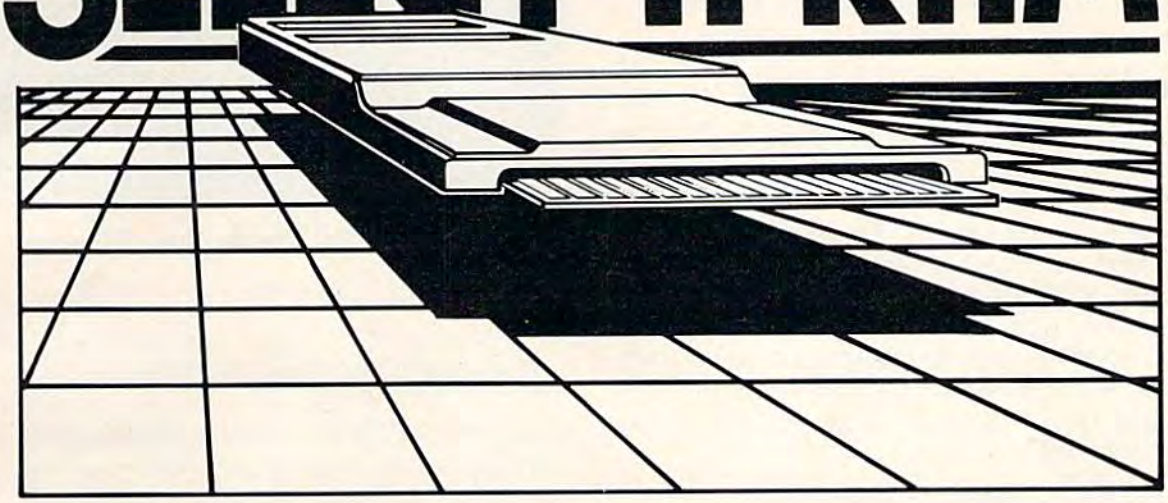
```
DATA 56,56,16,124,16,40,68,0
```

Add these two lines to the program used to copy characters into RAM, and you'll see the custom figure every time you press the A key. Notice, however, that you have lost the A character. There is now no way to print that on the screen. In other words, make sure that the characters you replace are ones you won't want to use.

If you have several new characters and they are replacing standard characters right after each other on the Screen Code table, you can place more than one in a READ statement, simplifying your programming. Replacing the first five standard characters, for example, would look like this:

64K for VIC 20™

SELECT·A·RAM™



SELECT·A·RAM

STANDARD FEATURES

- 8K BLOCKS SELECTABLE FROM THE KEYBOARD OR BY SOFTWARE COMMAND
- TWO EXPANSION SLOTS
- WRITE PROTECTION
- RESET SWITCH
- EXPANDABLE TO 192K WITH ADDITION OF 64K EXPANSION MODULES
- COMPATIBLE WITH ROM CARTRIDGES
- ONE YEAR WARRANTY ON PARTS AND LABOR
- 15 DAY MONEY BACK GUARANTEE

SELECT-A-RAM .....$169.

64K EXPANSION MODULE ....$149.

TRADE-INS ACCEPTED

3K $5 8K$10 16K-$20

OPTIONAL POWER SUPPLY FOR USE WITH ADDITIONAL 64K RAM MODULES OR MULTIPLE EXPANSION SLOTS .... **$25.**

VIC 20 IS A TRADEMARK OF COMMODORE ELECTRONICS LIMITED

512-441-3202 PO BOX 43006 Austin, Tx. 78745-0001

MasterCard

APS

ADVANCED·PROCESSOR·SYSTEMS

VISA

VIC-20 Character Developer

E. A. Cottrell

The utility program described here takes some of the tedium out of customizing characters for the VIC. It automatically converts the binary dot pattern numbers into decimal numbers for you.

Creating custom characters on the VIC-20 is not difficult. Eight bytes of information are in ROM for each character. To create custom characters, you must change the location to which the VIC looks for the characters to a location in RAM, and then POKE the desired information into the memory addresses normally used by the VIC. I won't go into detail here; see the related articles in this issue. You can also find more information in the *VIC-20 Programmer's Reference Guide.*

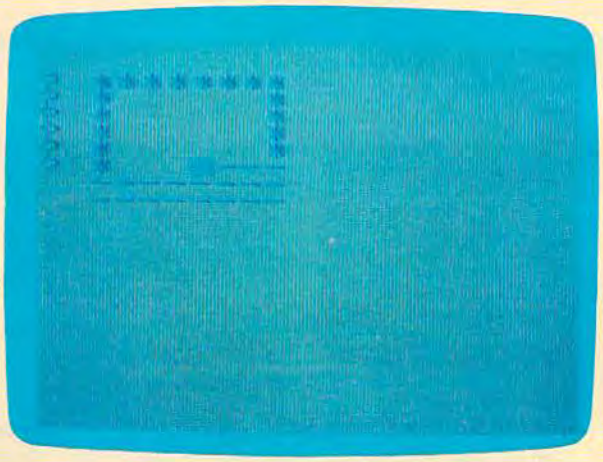
Relieving The Tedium

The most tedious part of this process for me is converting the binary representation of the characters into decimal numbers which can be POKEd into memory. The short utility program with this article eliminates that problem and speeds up the whole character development process. It also provides the hexadecimal equivalent for machine language programmers.

This program uses the full screen. Therefore, care must be taken when typing, especially with the PRINT statements. It is very important that all semicolons and commas be entered as listed.

When the program is first run, the screen will display eight rows of eight dashes in the top left corner, with a question mark in front of the first row. To design a custom character, enter asterisks (*) in place of the dashes (—), then press RETURN. The question mark then moves to the front of the second row. Repeat the process for all eight rows, creating your character with the asterisks as you go.

When RETURN is pressed after the eighth row is entered, the decimal and hexadecimal equivalents of each row are displayed. Below this, the custom character is shown, then the prompt "PRINT-OUT (Y/N)?". Pressing the Y key at this time will dump the screen to the printer. If you do not have a printer, lines 360-380 and lines 500-560 may be omitted.



Designing a hollow box with the "VIC-20 Character Developer."

Pressing the N key in response to the print-out prompt leads to another prompt, "ARE YOU DONE (Y/N)?". Pressing N here produces the prompt, "NEW CHARACTER (Y/N)?". A Y response clears the screen and gives you a clean grid with which to create your next character. Pressing N leaves the character you are working with in the grid and allows you to change only the parts desired. Remember that RETURN must be pressed for all eight rows whether all are changed or not.

The print-out can be very useful when designing characters which will face different directions. Simply turn the copy to the angle desired and enter the new character.

This program is more than a useful tool. Kids have great fun creating all sorts of things. If you answer Y to the DONE prompt, the @ key will display the new character. To restart the program, enter RUN 30 to keep from waiting for the transfer of the character set from ROM to RAM.

See program listing on page 195.

Sample Screen Dump To Printer

```
                    DEC     HEX

? --****--          60      3C
? -*----*-          66      42
? *-*--*-*          165     A5
? *------*          129     81
? *-*--*-*          165     A5
? *--**--*          153     99
? -*----*-          66      42
? --****--          60      3C

@

PRINT-OUT (Y/N)?
```

```
FOR C=7168 TO 7207:READ D:POKE
C,D:NEXT
```

You would then need five DATA statements, one for each new character created.

8K Expanded VIC

If you have an expanded VIC, with 8K or more of RAM, you'll need to enter additional commands before you run any program which copies a character set and creates custom characters.

"Creating Graphics on the Expanded VIC," by Ed Harris, in the February 1983 issue of COMPUTE! Magazine, includes a short machine language program that allows you to create custom characters on your expanded VIC.

If you don't want to use this utility, you can enter a few additional POKEs *before* you load and run any program creating custom characters. Enter each individual POKE, then press RETURN:

```
POKE 43,1:POKE 44,32
POKE 8192,0:NEW
POKE 36869,240:POKE 36866,150
POKE 648,30
```

The first line of POKEs sets the pointer to the start of the BASIC program, much like the POKE 52,28 did in the unexpanded VIC. The second POKE, the first memory location of BASIC, must be set to 0, or you won't be able to run your programs. The third line of POKEs relocates the screen, while the last POKE makes it possible for the operating system to see the screen.

As you enter the last two lines of POKE statements, the screen will change drastically. Don't worry — you haven't done anything wrong. You do, however, need to be careful as you enter these lines, for you can't really see what you've typed on the screen, due to the jumbled display.

Once these are entered, you can LOAD and RUN your program to copy characters and create custom figures. Line 20 in the program, used to copy characters to RAM, must be eliminated, however, if you use the expanded VIC. If you leave it in, the pointers in BASIC will change again, and you won't see the correct screen display.

Custom Hints

You now have the ability to design and enter your own custom characters on the VIC. Refer to the accompanying sidebar, "VIC-20 Character Developer," for a simple utility which will make it easier to design custom characters. This utility will also calculate the byte values of each row of a character for the DATA statements you'll be placing in your programs.

As you type in the lines for your custom characters, remember several things.

Each character needs eight numbers in the DATA statement to define it, even if some of the values are 0. If you don't have all eight numbers, the VIC will READ from the next DATA statement and your figure will not appear as you expected.

List the DATA statements in the same sequence as the READ commands used to replace the characters. The first READ statement looks to the first DATA statement, the second READ command looks to the second group of data numbers, and so on.

When you're finished, RUN the program to make sure the custom characters replace the right standard characters and appear as you wanted. Check the DATA statements, the byte values of the new characters, and the FOR C=X TO Y statements if you have problems. More than likely, any errors can be found in these places.

Users Groups

COMPUTE!'s Gazette is compiling a nationwide guide of VIC-20 and Commodore 64 users groups which will be published periodically. Please send us your group's name, address, the name of a contact person, and a phone number (optional). Also include any other relevant information: the group's membership, when it was founded, whether it operates a phone-in bulletin board system for modem users, etc. Address to:

Kathy Yakal, Editorial Assistant
COMPUTE!'s Gazette
P.O. Box 5406
Greensboro, NC 27403

VIC/64 Program Lifesaver

"UNNEW" Rescues Lost Programs

Vern Buis

If you have ever lost a BASIC program by accidentally typing NEW, then read on. This short machine language routine for both the Commodore 64 and VIC-20 (any memory size) provides an easy means of recovering BASIC programs that have been "erased" — and it loads and executes in only ten seconds.

Sooner or later — practically every programmer does it — thinking a program has been saved, you type NEW to clear out the memory, and a split-second after pressing RETURN, you wind up screaming.

But on the VIC-20 and Commodore 64, typing NEW does not really erase the program from memory. NEW just makes the computer (and the programmer) *think* the program is gone. As long as you don't start typing another program or switch off the machine, the program is still there. To get it back, all you have to do is fool the computer into remembering where in its memory the program begins and ends.

That's what "VIC/64 Program Lifesaver" does. By loading and running this short machine language utility immediately after committing the grievous error, you can save your lost program, save your hours of work, and even save your sanity.

Entering The Lifesaver

The Lifesaver is listed as a BASIC loader, a BASIC program that creates a machine language program. The same listing works for either the VIC-20 or Commodore 64. Be sure to read the following special instructions before typing the program. The procedure is somewhat different from most and requires that certain steps be followed exactly.

First, if you are using tape instead of disk, enter line 60 as follows:

```
60 CLR:SAVE"UNNEW",1,1
```

After typing the listing, *do not RUN it.* Instead, save it on disk or tape with a filename such as "LIFESAVER/BASIC" or "UNNEW/BASIC". Do not use the filename "UNNEW" This filename must be reserved.

Now enter RUN. The BASIC loader creates the machine language program and automatically saves it on disk or tape under the filename "UNNEW". This is what you'll actually use to rescue lost programs; the BASIC loader can be set aside as a backup in case you need to create another copy.

Using The Lifesaver

OK, let's say you've just typed NEW and wiped out hours of valuable labor. (To test the Lifesaver, you can load a BASIC program and erase it with NEW.) Recovering it is easy.

To load the Lifesaver from tape, enter:

```
LOAD"UNNEW",1,1
```

To load the Lifesaver from disk, enter:

```
LOAD"UNNEW",8,1
```

Either way, it loads pretty fast, because the program is short. Now, to activate the Lifesaver, enter:

```
SYS 525 [RETURN]
CLR     [RETURN]
```

QUICKSILVA IS HERE AMERICA

QUICKSILVA

Awesome software which extends your computer's potential to the maximum.
Enjoyed by millions in Europe, now Quicksilva comes to America.

QUICKSILVA

Britain's leading manufacturer and distributor of video games and other software. Programs available for the Timex-Sinclair 1000 and 2000™ VIC-20™ and Commodore 64™, BBC™ and Dragon™ Atari 400 and 800™.

Distributor

inquiries invited for USA and Canada: Please contact us for further information

QS™

THE GAME LORDS OF BRITAIN

Dealer

inquiries invited for USA and Canada: Please contact us for further information

QUICKSILVA INC.
426 West Nakoma
San Antonio, TX 78216
Tel: (512) 492-9054

Please send me a free color catalog
I currently own or plan to buy a ..... computer
I enclose a stamped, self-addressed envelope
Name .....
Street .....
City .....
State ..... Zip .....
G-2

(Incidentally, CLR means to type the keyword CLR, not to press the CLR/HOME key.)

That's all there is to it. When you enter LIST, the BASIC program you thought was forever lost at sea is back, safe and sound.

The Lifesaver itself also remains in memory, but probably not for long. It's tucked away in memory which is unprotected (locations used by the input buffer and BASIC interpreter), so you'll have to load it again each time you want to use it. But unless you're either very unlucky or (shall we say) prone to inadvertent actions, the Lifesaver isn't something you should be needing often.

Why It Works

Instead of erasing the program in memory when you type NEW, the VIC and 64 simply reset two key pointers in such a way that the operating system doesn't "see" that the program is still there. These pointers keep track of where in memory a BASIC program begins and ends. NEW moves the top-of-program pointer down to the bottom of BASIC memory, and the first two bytes of BASIC memory are set to zero. These first two bytes serve as a pointer to the address for the second line of BASIC code. When they are set to zero, the operating system believes that no program is in memory.

The Lifesaver works by skipping the first two bytes of BASIC memory (the address pointer) and the next two bytes (BASIC line number). It scans upward for a zero byte — the end-of-line indicator. Upon finding the zero byte, the routine POKEs its address, plus one, into the second-line-of-BASIC address pointer. One of the erased pointers is thereby restored.

Next, the Lifesaver scans byte-by-byte through the BASIC memory area until it finds three consecutive zero bytes. This is the end-of-program indicator. Once it locates these zeros, the routine POKEs the address of the third zero, plus one, into the top-of-BASIC/start-of-variables pointer at locations 45-46. This completely restores the erased program.

For those who might want to relocate the Lifesaver to a safer memory area — to preserve it for frequent use or to combine it with other utility routines — the machine language program is written to be fully relocatable. It uses no absolute JMP or JSR instructions. The area used here was chosen to make it load easily into a 64 or VIC with any memory configuration and to minimize the danger of it loading atop a BASIC program.

See program listing on page 203.

AARDVARK *Action Software*

PRESENTS. . .

ZEUS

WITHOUT QUESTION, ABSOLUTELY —

The Fastest

ARCADE STYLE GAME

AVAILABLE, THAT WE KNOW OF.

FAST AND FURIOUS. You become the wizard fighting off the thunderbolts of an angry Zeus. Multiple action. Are you good enough to keep up as the difficulty increases wave after wave. Hours of challenging fun. This is one game you may never completely master.

AVAILABLE ON: TRS-80C(16K), CMD 64, VIC 20(13K)
STOCK #1007 TAPE:$24.95 DISK: $29.95

PYRAMID... Our toughest and most intrigueing adventure. Average time though the pyramid is 50-70 hours. Clues everywhere and some very ingenious problems. This one is popular around the world.

AVAILABLE ON: TRS-80C (16K) CMD 64
VIC 20 (13K) TIMEX/SINCLAIR TI/99-4A
STOCK #5002 TAPE: $19.95 DISK $24.95

DUNGEONS OF DEATH. . . A serious D & D type game for you up to 6 players at a time. You get a choice of race and character, on screen mapping of the maze, a chance to grow from game to game, and a 15 page manual. Requires extended basic on TRS-80C.

AVAILABLE ON: TRS-80C(16K), CMD 64, VIC 20(13K), TRS-80C 32K DISC
STOCK #7001 TAPE: $19.95 DISK: $24.95

QUEST. . . A different kind of adventure. Played on a computer generated map of Alesia. You lead a small band of exployers on a mission to conquer the Citadel of Moorlock. Takes 2-5 hours to play and is completely different each time.

AVAILABLE ON: TRS-80C(16K) CMD 64
VIC 20(13K) TIMEX/SINCLAIR TI99/4A, EXTENDED BASIC REQUIRED ON TI
STOCK # 7007 TAPE: $19.95 DISK: $24.95

Quest is also available on TRS-80C as a 32K extra memory, extra fun adventure. It's a little more difficult, and will test the most experienced explorer of the kingdom.

STOCK #7006 TAPE: $24.95 DISK: $29.95

AARDVARK *Action Software*

IS AVAILABLE AT SOFTWARE RETAILERS EVERYWHERE. IF YOUR LOCAL COMPUTER OR SOFTWARE STORE DOES NOT CARRY IT — ASK THEM WHY NOT. SEND ONE DOLLAR FOR ILLUSTRATED CATALOG. DEALER INQUIRES INVITED.

HOURS OF FUN AND ENJOYMENT FROM THE "ADVENTURE PLACE"

TO ORDER: SEND AMOUNT INDICATED PLUS $2.00 SHIPPING, PER ORDER. INCLUDE STOCK NUMBER, QUANTITY DESIRED AND YOUR PREFERENCE OF TAPE OR DISK. BE SURE TO INDICATE TYPE OF SYSTEM, AMOUNT OF MEMORY, ETC. WHEN USING CHARGE CARD TO ORDER BY MAIL, BE SURE TO INCLUDE EXPIRATION DATE.

CHARGE CARDS WELCOME

1-800-624-4327

PHONE ORDERS ACCEPTED
8:00 A.M. TO 8:00 P.M. E.S.T. MON-FRI

AARDVARK *Action Software*

2352 S. COMMERCE RD. WALLED LAKE, MI 48088 313/669-2110

Understanding Sound On The 64

Part 2

Gregg Peele, Editorial Programmer

Ever wished you could create just that right sound for a game effect? Or that right tone for a song? The conclusion of this two-part article and the accompanying utility program may be just what you need to create interesting new sounds on your 64.

Last month we explored some of the basics of producing sound on the Commodore 64. We discussed ADSR (attack, decay, sustain, and release) and used these parameters along with volume, pitch, and waveform to produce various sounds. This month, we will look even further into the capabilities of the 64's built-in "synthesizer on a chip," the Sound Interface Device (SID). We'll discuss filters, ring modulation and synchronization, and present a utility, "Soundmaker 64," which will make it easier to use these techniques within your own programs.

Changed Your Filters Lately?

The Commodore 64 SID chip has three filters — but unlike the filters in your car, they should never need replacing. However, they do share some similarities with car filters. Just as an oil filter allows oil to pass while blocking out other unwanted particles, the SID chip filters let parts of sounds pass — selectively *filtering* out the remainder of the sound. Synthesizer filters provide an important means of manipulating sounds to produce various effects.

The three filters are called *high pass, low pass,* and *band pass.* The high-pass filter is designed to remove the lower frequencies, letting the higher frequencies pass. The low-pass filter has the opposite effect — it removes the high frequencies while allowing low frequencies to pass. The band pass filter allows a band or group of frequencies to pass through while frequencies above and below the band are suppressed.

The filter you choose is activated by turning on bits 4 (low pass), 5 (band pass), or 6 (high pass) in SID register 24 (read the accompanying short article, "Switching Bits," for details on turning bits on or off). These filters can be used in combinations for additional effects. For instance, adding the low and high pass filters together creates the inverse effect of the band pass filter; only the higher and lower frequencies pass, suppressing the middle frequencies.

The amount of sound that is removed by a filter is determined by the *cutoff frequency.* The filter cuts off the sound beginning at this frequency. The cutoff frequency for filtering is controlled by the lower three bits in SID register 21 and all eight bits in register 22. Some of the most interesting effects possible on the 64 are created by incrementing or decrementing these series of bits while a sound is being played. Want the sound of an alien ship as it lands? Use your normal alien ship sound, add a filter, and gradually increment or decrement these eight bits as your ship descends. A certain combination of waveforms and a changing filter can create just the right sound effect for a descending alien ship.

READ THE LABEL

THE FILING ASSISTANT

DATA BASE MANAGEMENT for the **COMMODORE 64**

This data management package allows you to create record formats, sort information, and produce reports. It includes the following features:

Design records up to 256 characters, up to 25 fields, (up to 35 characters per field)
Designate up to 10 two-level sorts
"Step through" records while displaying selected fields to screen
Add, copy, change, or delete records
"Find" a record by key field, or "locate" by record number
Set function keys to insert a frequently used word, phrase, zip code, etc.

No programming or code necessary. On-screen prompts lead to full operation in one to two hours.

THE FILING ASSISTANT contains a built-in mail label program that allows you to print a 1 to 6 line label. You can include more than one field per line and print up to four labels across the page. Labels to be printed can be selected on one or more fields.

The program also allows you to design your own report formats. Your reports may include:

Automatic record counting
Computation between fields
Totals and/or sub-totals per column
Record selection by one or more fields

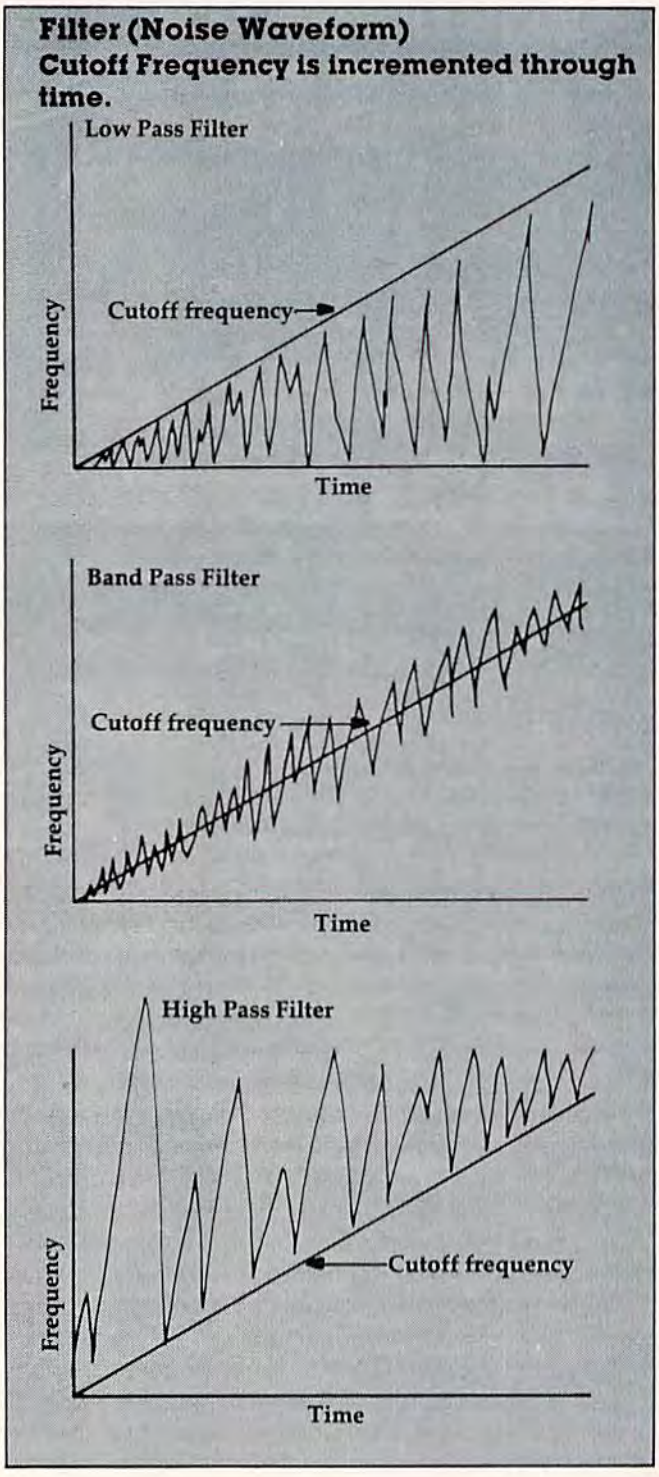
TECHNICAL NOTE

This program supports the serial or the user port —you can configure just about any printer. It can be used with more than one 1541 disk drive or with the "C-64 Link-IEEE Interface" which allows you to work with the Commodore 2031, 4040, and 8050 disk drives.

THE FILING ASSISTANT is part of a series. It will integrate with **THE WRITER'S ASSISTANT** (word processing) and **THE SPREADSHEET ASSISTANT** (forecasting and financial analysis). Even if you are a beginner **THE FILING ASSISTANT** will allow you to manage personal and business records. The **ASSISTANT SERIES** provides you with the most complete and practical information management package available for the Commodore 64.

RAINBOW Computer Corporation

490 Lancaster Pike
Frazer, Pennsylvania 19355



"Soundmaker 64" lets you experiment with all the parameters of the sound chip and add the results to other programs.

Additive And Subtractive Synthesis

Filtering is an example of *subtractive synthesis.* Subtractive synthesis is a method of manipulating sounds by subtracting parts of a single sound — pushing other parts which normally may not be heard into the forefront. *Additive synthesis,* however, brings two sounds together to form a totally new sound. Both *ring modulation* and *synchronization* are examples of additive sound synthesis.

Ring Modulation

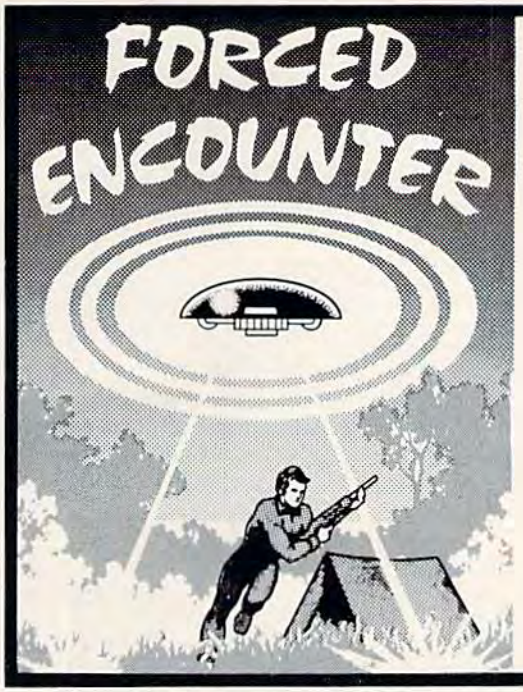
Ring modulation is a form of additive sound synthesis that dramatically changes the timbre or tone quality of two tones. Tones that have been fed through a ring modulator do not retain their original pitches or timbres. Instead, the sums and remainders of the two frequencies are retained. For instance, if the first sound is a tone that vibrates at 100 vibrations per second (vps), and the second tone vibrates at 200 vps, then the ring-modulated tone will be a combination of the sum (300 vps) and the difference (100 vps).

Usually the ring-modulated tone sounds very different than the two original tones. Since most tones are complex phenomena consisting of many less obvious inner frequencies (harmonics), the ring-modulated tone may be very complex in tonal character.

To achieve ring modulation on the 64, you have to set bit 2 of the waveform byte when using the triangle waveform (POKE register 4 with 21). Voice 3 must be set to some frequency. No other parameters of Voice 3 have any effect on ring modulation.

Synchronization on the 64 also adds two tones together to produce a new and different sound. If bit 1 of the waveform byte is set (POKE register 4 with 19), then setting Voice 3 to a definite pitch (POKE registers 14 and 15 for the pitch of Voice 3) and manipulating the pitch of Voice 1 (registers 0 and 1) cause the tone quality of the resulting pitch to change.

Synchronization happens when the two waveforms are linked to make the waveform of Voice 1 dependent on whether it is *in sync* with the frequency produced by Voice 3. Since the two waveforms are not usually in sync, the waveform is distorted, producing different and sometimes interesting waveforms. In sync mode, the pitch of



FORCED ENCOUNTER INSTRUCTIONS

Before play begins the saucer will fly to the top of the screen. Each time the saucer is destroyed a new saucer appears at a lower height on the screen. When the fifth saucer at the lowest height is destroyed a bonus of 200 points is awarded and a bonus man is given. Only one bonus man is given per game.

The man may be controlled by the joystick or the keyboard. The CRSR control keys located in the lower right corner of the keyboard move the man left and right and the shift key fires the gun.

The saucer is seeking biological specimens and uses gas bombs to stun its prey before taking them aboard the ship. The saucer also has a capture net made up of electrostatic charged particles. The net cannot be destroyed and if the man is caught his movements are restricted but he can still fire his weapon. Once the net hits the ground it is harmless and disappears.

A skill level can be selected when the title page is on the screen and the level will stay the same from one game to another until it is manually changed. Use the F1 function key to change the skill level, located at the bottom of the screen. Levels range from the easiest at 0 to the hardest at 9. When play begins you have three men, the number of men remaining appears in the lower left corner of the play field. A bonus man can be awarded giving a maximum of four men per game. The high game score is reflected on the lower right of the play field. The highest game score at Magic Carpet Software is 9,872. Good Luck!!

Commodore 64

$19.95 tape or disk

OTHER GAMES FOR THE 64 INCLUDE
WEATHER WAR II .....
MEDICINE MAN .....
AND FOR THE VIC-20 COMPUTER
FROG .....
PHANTOM FIGHTER .....

$19.95
TAPE

MAGIC CARPET
P.O. Box 35115 Phoenix, AZ 85069

GAMES FOR COMMODORE COMPUTERS

ADD 60 COMMANDS TO YOUR COMMODORE 64 WITH:

grafDOS

An enhanced disk operating system adds 40 *NEW* commands to both **BASIC** and **DOS**! Includes easy to use Disk Commands, HIRES Graphics, LORES Graphics, Text and Sprites.

MINIMON

A powerful machine code monitor which adds 20 *NEW* commands to examine memory, disassemble, text dump, miniassembler and *MORE!*

ONLY **$39.95** **MINIMON available separately for $19.95**

DUST COVERS
Attractive, brown vinyl covers.

COMPUTER or DRIVE	$7.95
OLD STYLE DATASETTE	$3.95
NEW STYLE DATASETTE	$3.95
1525 PRINTER	$7.95

Send for our *FREE catalog* of the lowest prices for:
CBM-64 GAMES • NEW PRODUCTS
VIC-20 GAMES • BOOKS
AND MORE!

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

Visa/MC/Check/Money Order — Add $2.00
CA residents add 6½% sales tax
DEALER INQUIRIES INVITED

Switching Bits

Registers are simply memory locations that have special functions. In the SID chip, there are 28 registers, most of which are dedicated to sound production. In Soundmaker 64, the registers we POKE do not begin at the sound chip (54272), but at 49152. The machine language routine then "mirrors" these registers by copying them to the sound chip registers.

To understand how registers can be manipulated, a brief discussion of bytes is necessary. Bytes are memory locations that can store values. On the Commodore 64 and most other small personal computers, bytes consist of eight smaller units called bits (binary digits). A bit can be either on or off, usually represented by 1 or 0. Thus, any byte's value can be represented by eight numbers, where the numbers are limited to 1 and 0. Example: **10110111**

Since only two possibilities exist for each bit (1 or 0), this numbering system is referred to as binary or base two.

Our usual number system is known as decimal or base ten. This system uses ten different symbols to designate numbers (0-9). The position of the number from the rightmost side determines the *power of 10* by which the number is to be multiplied. For instance, here's how the decimal number 2156 breaks down:

Decimal:	**2**	**1**	**5**	**6**
Position:	**3**	**2**	**1**	**0**

Power of 10: $2*10^3 + 1*10^2 + 5*10^1 + 6*10^0 = 2156$
Value: (2000) + (100) + (50) + (6) = 2156

The value of binary numbers is determined the same way, except the position of the digit (the bit) determines the *power of 2* by which the number is multiplied. Here's how the binary number 10110111 breaks down when translated to decimal:

Binary:	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**1**
Position:	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Power of 2: $1*2^7 + 0*2^6 + 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 1*2^0 = 183$
Value: (128) + (0) + (32) + (16) + (0) + (4) + (2) + (1) = 183

Now we can attack our original problem – changing bits within registers on the 64. Bits are labeled 0 to 7 *from right to left*. To set bit three of a register to 1 (in other words, turn it on), you can POKE the register with an 8 (2 to the third power). This POKE erases what was in that register and replaces it with an 8 (00001000).

You can change certain bits within a register without affecting the other bits with the keywords OR and AND. The OR operator is used to compare two bytes, bit for bit, to alter certain bits without altering the others. Here's an example:

Original byte — 00010001 = 17
OR with 8 — 00001000 = 8
Resulting byte — 00011001 = 25

The OR statement looks at each bit of the original byte and compares it to the corresponding bit of the OR byte. If one OR the other of the bits is a 1, then the resulting byte will contain a 1 in that particular bit. If neither of the bits is a 1, the resulting byte will contain a 0 in that bit. Thus, here's how you would set bit 3 to a 1 without changing the other bits, in BASIC:

```
POKE byte,(PEEK(byte)OR 8)
```

The AND keyword also looks at each bit of one byte and compares it to the corresponding bit of the second byte, but in a different way. Here's an example of AND:

Original byte — 00001111 = 15
AND with 254 — 11111110 = 254
Resulting byte — 00001110 = 14

The AND operation looks at the two bytes and sets a 1 in the corresponding bit of the resulting byte *only* if both the first bit AND the second bit are 1's. Thus, ANDing 15 with 254 clears the 0 bit. In BASIC this is written: **POKE(byte),(PEEK(byte)AND254)**

the tone you hear depends on the pitch of Voice 3, not Voice 1 as would normally be the case.

Paddling With The SID

The SID chip also contains two registers (25-26) connected to the two joystick ports. These registers will contain a number from 0 to 255, depending on the resistance of a potentiometer attached to the ports (255 at maximum resistance). Since game paddles are really potentiometers (variable resistors), these ports can be used to register paddle movement and can easily be used to change values in other registers within the chip while sounds are being produced.

This simple routine can be added to a sound program to control the pitch of Voice 1 with a paddle plugged into port one while a tone is being played:

```
10 POKE 54272+1,PEEK(54272+25):GOTO10
```

This line connects the paddle value to the high-byte frequency value of Voice 1. It's much easier to study the effects of changing sound values if you can hear the sound playing as you experiment. That is the basis of "Soundmaker 64."

Soundmaker 64

Soundmaker 64 allows you to create your own sounds and manipulate them by changing various parameters. Attack, decay, sustain, and release are included as well as pitch, filters, ring modulation, and synchronization. The pulse waveform may be manipulated to change the pulse width of the sound — altering the timbre of the resulting sound considerably.

To use Soundmaker 64, type in the program and save it on disk or tape. When you are sure you have a saved copy, run the program. After a brief delay while the program loads a small machine language routine into memory, the word "Attack" appears at the upper-right corner of your screen. Using the + and - keys, you can increase or decrease the attack value for your sound. The current value POKEd is represented by both a bar graph and a number. The number varies in units of sixteen or one depending on which parameter you are working with. These values are meant to serve as a reference point only, since they may differ from the actual value by one unit. The increments were selected to make the changes in parameters very easy to hear and the program easy to use.

Once you have decided on the attack value, simply hit RETURN and the next parameter appears. Keep in mind that "Sustain" and "Volume" must be a reasonably high number for the sound to be audible. When you have picked all the parameters ("Pulse wave low" is the last one on the screen), then you can play the sound with the function keys. F1 plays the sound with the sawtooth waveform, F3 with the triangle waveform, F5 with the noise waveform, and F7 with the pulse waveform.

Ring Modulation And Sync

The up-arrow key (beside the asterisk) plays your sound as it is ring modulated with Voice 3, and the left-arrow key (beside the 1) plays the synchronized sound resulting from the pitches of Voice 1 and Voice 3. (Ring modulation and synchronization are limited to Voice 1.)

Once you have heard Voice 1, simply hit the 2 key and you will again be prompted for the parameters. As with Voice 1, you play Voice 2 with the function keys. To hear Voices 1 and 2 simultaneously, hit the space bar. To select the parameters for Voice 3, press the 3 key. The space bar then plays all voices previously defined. If you have selected ring modulation or synchronization for Voice 1, you may not be able to use Voice 3 as a separate sound.

Changing Sounds

To alter any parameter at any time after entering it originally, merely press the key which is in reverse field on the parameter name and press the + or - key to raise or lower the value. When done, hit RETURN.

You can even change parameters as the sound is playing. To do this, hit one of the function keys or one of the arrow keys to start the note and, without releasing it, hit the reverse field character of the parameter you wish to change. Then change the sound with the + and - keys.

To use the filters as the sound is being played, you must first start the sound that you want, then, without releasing the key, hit either H (for high pass), B (for band pass), or L (for low pass). Next, hit F for filter, and use the + and - keys to increment or decrement the cutoff frequency. As before, hit RETURN to end the note.

To save the sound or sounds that you have created, press Q while the note is playing. The screen clears and a program appears on the screen. Type NEW and press RETURN over the lines as they are listed on the screen. Then you can play this sound, or save it on tape or disk and use it later as a routine in your own programs. To use it as a routine, you'll need a delay loop such as this to set the duration:

```
70 FOR T=1TO 2000:NEXT T
```

Then, to turn off the sound, use this line:

```
80 FORT=49152+4TO49152+18STEP7:POKET,
(PEEK(T)AND254):NEXT:SYS53017
```

To turn on the sound in your own program, you can either GOSUB the whole routine, or use this line (with your own line number):

```
FORT=49152+4TO49152+18STEP7:POKET,
(PEEK(T)OR1):NEXT:SYS53017
```

A Bit About The Program

Soundmaker 64 uses a tiny machine language (ML) routine which copies the contents of 24 bytes starting at 49152 to the sound registers beginning at 54272. The ML routine copies the registers in the order they should be POKEd to properly create a sound.

This is done because sound registers are *write only* registers. That is, when values are POKEd into the SID registers, they cannot be PEEKed later. Instead, you must store the values in variables or other memory locations. The ML routine stores these values in a safe area of memory and allows us to copy them at any time to the SID registers. The ability to "remember" the values which have been POKEd into the SID chip makes Soundmaker 64 possible.

See program listing on page 198.

DAN CARMICHAEL, ASSISTANT EDITOR

Animating With Custom Characters

In this month's column we'll show you how to spice up your programs with animation. The basic information also applies to the Commodore 64.

Elsewhere in this issue we've shown how to create custom characters for your programs. Now we'll learn how to animate those custom characters and add a little action to the screen.

Almost everyone remembers their first experience with animation. You might have drawn a stickman on successive pages of a scratch pad, and when the pages were rapidly flipped, the stickman appeared to run across the pages. This is the same principle used in computer animation. By creating various custom characters (like a stickman) in different positions, and then POKEing them to the screen, you can simulate animation.

Before proceeding, turn off your VIC, remove any memory expansion, then turn it on again. Enter, SAVE to tape or disk, and then RUN Program 1. This program creates a few custom characters (stored in the DATA statements). Line 10 reserves a special place in memory for them (512 bytes at the top of BASIC memory), then POKEs the characters into memory. The POKE (36869) in line 50 directs the operating system to look at our special character set in RAM memory instead of the standard character ROM.

Program 1

```
10 POKE52,28:POKE56,28:CLR          :rem 18
20 FORA=7313TO7679:POKEA,0:NEXT   :rem 148
30 FORA=7168TO7312:READB:POKEA,B:NEXT
                                    :rem 55
50 POKE36869,255:PRINT"{CLR}{5 DOWN} @ A
   {SPACE}B C D E F G H I{DOWN}{3 SPACES}
   J K L M N O P"                   :rem 56
60 GETA$:IFA$=""THEN60             :rem 239
7168 DATA 56,84,146,146,146,146,84,56
                                    :rem 62
7176 DATA 56,68,138,146,146,162,68,56
                                    :rem 64
7184 DATA 56,68,130,130,254,130,68,56
                                    :rem 43
7192 DATA 56,68,162,146,146,138,68,56
                                    :rem 62
7200 DATA 12,44,90,28,24,164,66,1  :rem 83
7208 DATA 12,12,24,60,24,88,40,8   :rem 35
7216 DATA 12,12,8,28,12,14,16,32   :rem 26
7224 DATA 12,12,8,30,44,10,20,32   :rem 14
7232 DATA 0,0,0,0,0,0,0,0          :rem 156
7240 DATA 0,0,0,0,0,0,0,0          :rem 155
7248 DATA 24,24,60,60,255,255,126,66
                                    :rem 248
7256 DATA 8,88,26,188,111,253,126,74
                                     :rem 4
7264 DATA 138,17,144,56,109,229,62,72
                                    :rem 49
7272 DATA 130,17,128,40,76,165,26,64
                                    :rem 246
7280 DATA 0,16,128,40,4,164,24,0    :rem 26
7288 DATA 0,0,0,40,0,36,0,0         :rem 20
7296 DATA 0,0,0,8,0,0,0,0           :rem 174
7304 DATA 159,255,77,111,192,254,16,59
                                    :rem 100
7312 DATA 0                          :rem 23
```

This program displays the custom characters with a PRINT statement. To create the animation effects, we could use the PRINT statement, but it would be easier and more efficient to use POKE.

The next few programs use the custom characters created in Program 1, so don't turn off the computer. Press RUN/STOP—RESTORE, then enter NEW. This will clear BASIC memory of the program that was there, but will leave our custom character set intact, still protected.

COMPUTE!'s First Book of VIC Games

Authors: COMPUTE! Magazine Contributors
Price: $12.95
On Sale: Now

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

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

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

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

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

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

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

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

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

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

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

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

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

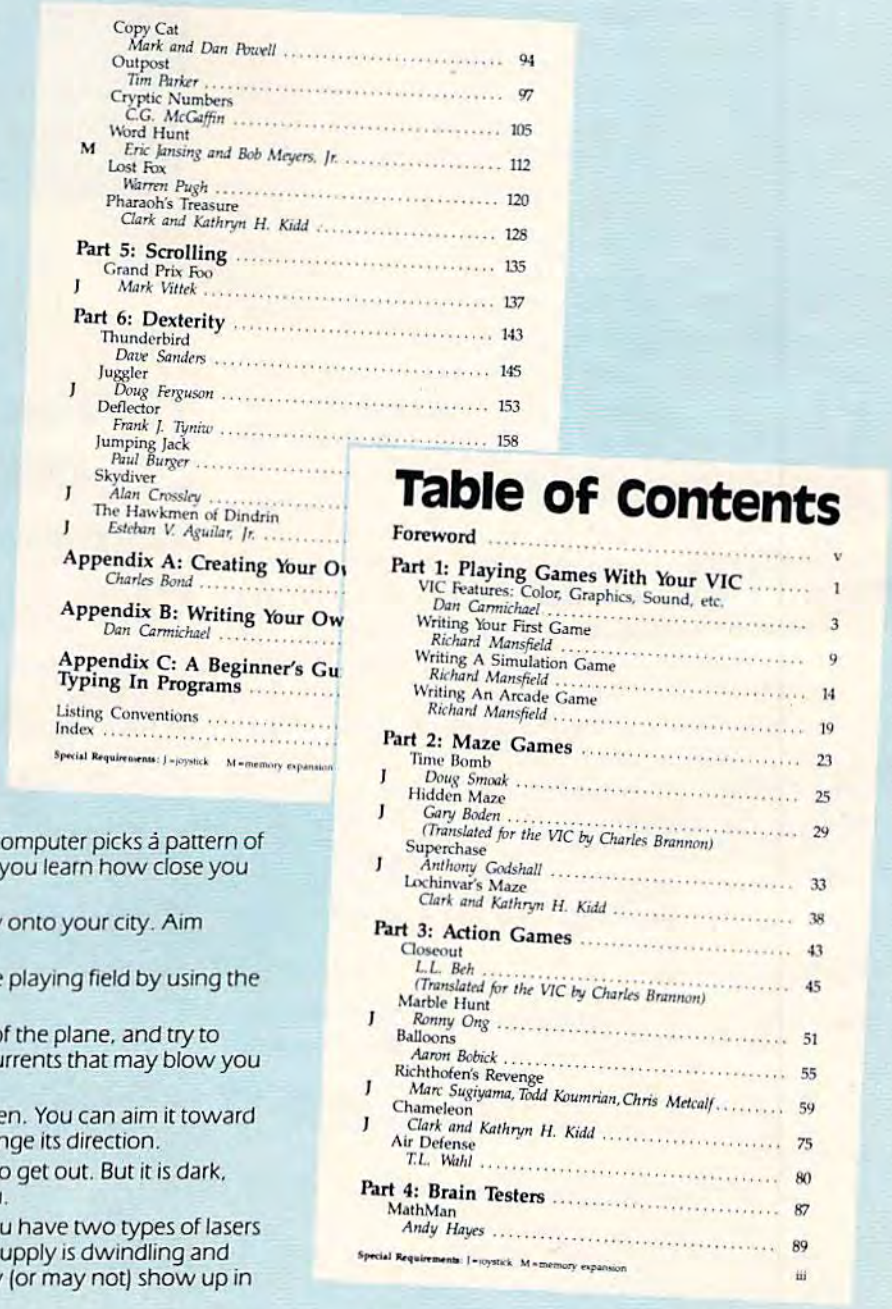


Table of Contents

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

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

ORDER NOW

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

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

Rolling And Spinning Wheels

If you turn the computer off then back on, you will have to rerun Program 1. Now enter, then RUN, Program 2.

Program 2

```
10 PRINT"{CLR}":C=PEEK(646):POKE38630,C:P
   OKE38632,C:POKE36869,255:SP=50
20 FORA=0TO3:POKE7910,A:POKE7912,3-A:FORT
   =1TOSP:NEXTT:NEXTA:GOTO20
```

As you can see, all this program does is POKE custom characters to the center of the screen. The effect of the spinning wheels is created by POKEing the screen with various wheels (characters) with the spoke in different positions. You can vary the speed of the spin by changing the variable SP in line 10. A lower value will increase the speed of the spin, a higher value will slow it.

Study line 20 of Program 2. It contains a programming technique that might be of interest to you. While the "FOR A . . . " loop is set to count forward (to animate the left wheel running clockwise), the second POKE (7912) in line 20 in effect counts backwards. This is how the right wheel is made to run counterclockwise. By subtracting the variable A from the constant 3, the FOR A loop seems to run backwards. It's like having two FOR/NEXT loops in one — one counting forwards, the other counting backwards.

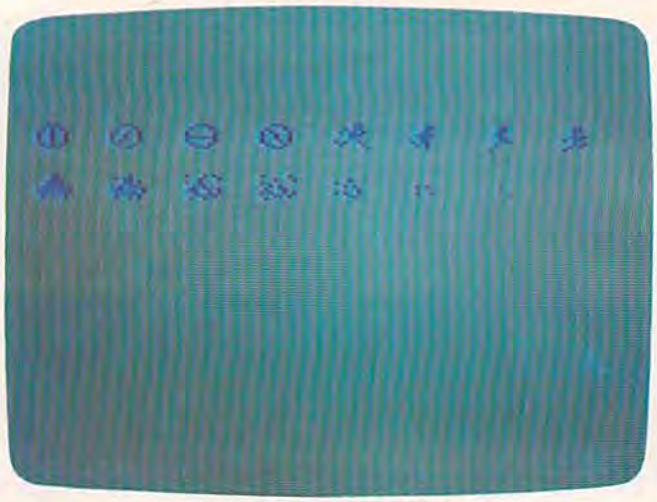
Again, press RUN/STOP—RESTORE, then enter NEW. Now enter Program 3 and RUN. This program uses the same wheel characters, but instead of spinning in one place, the various wheels are POKEd to successive positions, producing the effect of the wheel rolling across the screen. Again, the variable SP in line 10 controls the speed. If you change the speed and make the program run too fast, the rolling effect is almost lost.

Program 3

```
10 PRINT"{CLR}":POKE36869,255:C=PEEK(646)
   :FORA=38400TO38905:POKEA,C:NEXT:N=0:SP
   =90
20 FORA=7680TO8185:POKEA,N:POKEA-1,32:N=N
   +1:FORT=1TOSP:NEXTT:IFN=4THENN=0
30 NEXTA
```

Notice **POKE A-1,32** in line 20. This POKE puts a blank space (screen POKE character 32) on the screen just before the current position of the rolling wheel. This causes the wheel to erase the space just behind itself as it rolls across the screen. Remove this POKE from the program, RUN it, and see what happens.

Speed is important when animating custom characters. If they move too fast, the eye cannot see all the subtle changes in the characters, and the effect is lost. On the other hand, if the program runs too slow, the characters appear to jump, and the effect of smooth animation is lost.



A close-up of the custom characters used in the example programs.

The Running Stickman

Press RUN/STOP—RESTORE, then enter NEW. Now enter Program 4 and RUN.

Program 4

```
10 PRINT"{CLR}":POKE36879,27:FORA=38400TO
   38905:POKEA,0:NEXT:POKE36869,255:N=4:S
   P=100
20 FORA=4TO7:POKE7910,A:FORT=1TOSP:NEXTT:
   NEXTA
30 GETA$:IFA$=""THEN20
35 SP=150
40 POKE7910,32:FORA=7680TO8185:POKEA,N:PO
   KEA-1,32:FORT=1TOSP:NEXTT:N=N+1:IFN=8T
   HENN=4
50 NEXTA
```

As you can see, Program 4 also POKEs custom characters (various poses of a stickman) to the center of the screen. The animated running effect here is produced with four different custom character stickmen. If you watch the animated action closely, it even looks as if the stickman is alternating legs while running.

Now press any key on the keyboard and watch the stickman run across the screen. Again, speed is an important factor here. Change the variable SP in line 10 to 0, then RUN. As you can see, the speed is so fast that not only is the animation effect lost, but it's hard to even recognize the character as it moves.

Program 4 shows the main drawback of animating with POKEd custom characters: jumpy action. This effect is caused by the characters moving one character block, or eight pixels, at one time. For truly smooth animation, high-resolution graphics is better because it allows you to move objects one pixel at a time. However, we will not touch upon hi-res animation here because

it is an advanced programming technique that requires machine language.

Exploding UFOs And Other Effects

POKEd animation isn't limited to making objects move. You can also create interesting effects with stationary objects. Again, press RUN/STOP—RESTORE, then enter NEW. Now enter and RUN Program 5. This program uses custom characters to make a flying saucer explode and disintegrate. Change the variable SP in line 10 to 250, then RUN it again. Notice how tiny bits (pixels) of the saucer seem to fly outward, then disappear.

Program 5

```
10 PRINT"{CLR}":POKE36879,27:POKE36869,25
   5:POKE38630,0:SP=120
15 POKE7910,10:FORT=1TO500:NEXTT
20 FORA=11TO16:POKE7910,A:FORT=1TOSP:NEXT
   T:NEXTA:FORT=1TO500:NEXTT:GOTO15
```

Once you've conquered the art of animating with custom characters, you can add the final touch: sound. Adding the sounds of footsteps as the stickman runs across the screen, or an exploding sound as the flying saucer disintegrates, will provide that little extra touch that makes your graphics animations fun and enjoyable.

Merging Programs On The 64

John A. Winnie

For intermediate programmers, "Merger" allows you to build up large programs by working on smaller portions separately and then linking them together later. This approach is used by many professionals.

If you do much BASIC programming, sooner or later you'll need to merge two short programs to form a larger one. Or perhaps you'll need to append onto a program a series of DATA statements – DATA for sprites, redefined characters, sound and music, or whatever. Here is a quick and easy way to add those DATA statements – or any other BASIC statements, for that matter – onto the end of your programs.

Of course, various techniques for merging programs have been around for some time. When all that is needed is a simple append, however, the method presented here does the job nicely. The program below, "Merger," is designed to merge with any programs which are appended to it, and it allows you to keep on appending indefinitely.

Using Merger

After typing and saving Merger, load it in the usual way. Next, RUN Merger, and then load in your main program. Now, as Merger instructs, POKE locations 43 and 44 with 1 and 8, respectively. Your main program is now appended to Merger and ready for any DATA statements you may want to add later.

Remember, Merger allows you to append programs only, not to insert them. So to prepare for using Merger later, begin your programs with a line number greater than five. For the same reason, all DATA statements to be added should begin with a line number higher than those already present in the program. When you have finished, just erase Merger by deleting lines 1 through 5.

How Merger Works

First, clear out your Commodore 64 by typing NEW and pressing RETURN. Then enter the following simple program:

```
10 REM
```

Press RETURN, and the one-line program is now entered into memory beginning at address 2048 and running on upward. To see just how the program is stored, enter:

```
FOR I = 2048 TO 2056:PRINT PEEK (I) :NEXT I
```

If all this has been done correctly, you now should see a list of memory contents which looks like this:

```
0,7,8,10,0,143,0,0,0
```

The 0 in address 2048 is invariable: all BASIC programs begin with zero. They also always end with a zero; in fact, they always end with exactly three zeros – which is just what we see here in memory locations 2054 through 2056. From this point on in memory, BASIC will store any variables and other information that it may need to execute the program.

In general, when a BASIC line is stored, it will end with a single zero, not three zeros. When a new line is appended to the program, its code begins immediately after that single zero. So in the example above, if the line

```
20 REM
```

were now added to our sample program, the (link

Commodore 64 and VIC-20

Graphics

"ELF" from Picture Library

DRAW
with your joystick!

Now, you can create **high-resolution pictures** on-screen with your joystick as a "pen." Design critters, objects, pie-charts — whatever your imagination wishes! **SAVE your creations to tape or disk, and PRINT them on a VIC printer.** Educational and fun!

Draw narrow or wide lines, curvy or straight; set points; add captions; create background patterns; change picture, background, and border colors; reverse colors for a negative; even connect dots with straight lines automatically! **You control every dot on the screen.**

A large "Picture Library" is included to get you started, plus a 20-page instruction manual. Joystick required. VIC printer and disk drive optional. **Now on disk or cassette!**

"FLY" from Picture Library

For the Commodore 64:
'64 Panorama $29.95

For the VIC-20:
VIC-PICS $29.95
(Full features need 8K mem exp; reduced version included for unexpanded VIC.)

Available for other printers—Call!

PRINT (the unprintable) . . . with Smart ASCII Plus–$59.95

Now, print the unprintable **Commodore graphics** on your **dot-addressable parallel printer*** with Smart ASCII Plus. This **powerful, low-cost software interface** converts your **user port** into a fast, intelligent port for "Centronics" protocol printers, and we even supply the cable!

Six flexible print modes: GRAPHICS, TRANSLATE, DaisyTRANSLATE, CBM ASCII, True ASCII, PIPELINE. GRAPHICS mode creates actual VIC/64 keyboard graphics. TRANSLATE converts normally unprintable control-codes into text: (CLR), (RVS), (BLU), etc., with an extended mode for Daisywheel printers. Convenient set-up menu and simplified operation. Compatible with most application programs: WordPro 3+, EasyScript, Quick Brown Fox (for the VIC), Writer's Assistant, etc.

Complete with connecting cable for printer and instruction manual. On cassette. Copy to your disk for quick loading. (Upgrades available for original Smart ASCII owners.)

*Requires dot-addressable printer such as: Epson FX-80 or MX-80/100 with Graftrax; Okidata Microline 84; C. Itoh Prowriter 1 & 2; Star Micronics Gemini-10 or 15. **Other printers—Call!**

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

(816) 333-7200

Send for a free brochure.

MAIL ORDER: Add $1.50 shipping and handling ($3.50 for C.O.D.); VISA/Mastercard accepted (card# and exp. date). MO residents add 5.625% sales tax. Foreign orders payable U.S.$, U.S. Bank ONLY; add $5 shp/hndlg.

MIDWEST MICRO inc.

311 WEST 72nd ST. • KANSAS CITY • MO • 64114

There's a New Chick in Town

. . . with a wing load of Quality Software for the Commodore 64™ and VIC 20™

Once you've compared our programs, their features and prices, you'll agree there's no competition in sight. You'll also discover another important reason to go TOTL. . . Customer Assistance After You Buy. . . Something nearly unknown in the low cost software field.

some features

Menu driven, Easy to use ■ Available on tape or disk ■ Compatible With Most Column Expansion Hardware ■ Built-in ASCII Translation for Non Commodore Hardware ■ Color Variables for Easy Modification ■ Machine language speed for Word Processing and Mailing List & Label software.

and there are 5 unique programs to choose from

1. **WORD PROCESSING** has the speed and versatility to produce documents, forms and letters in a straightforward approach that is easily and quickly learned.
2. **MAILING LIST AND LABEL** lets you organize your mailing lists, collection catalogs, menus, recipes and anything that demands listing or sorting.
3. **TOTL TIME MANAGER** helps you plan schedules and analyze events and activities by persons, project catagory and date. Ideal for project planning in the home or business.
4. **RESEARCH ASSISTANT** turns your computer into an advanced, automated indexing and cross reference system. A must for the student, educator or the research professional.
5. **SMALL BUSINESS ACCOUNTING** is a set of straightforward accounting programs. Frees the salesman, entrepreneur or service professional from time consuming record keeping tasks.

TOTL SOFTWARE, INC. quality you can afford

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

1555 Third Avenue, Walnut Creek, CA 94596

PLEASE SEND ME MORE INFORMATION ON TOTL SOFTWARE

Name: _____

Address: _____

_____ Zip _____

of the) new line would now come in at address 2055 – the address of the middle zero in the triplet; a new triplet of zeros would appear later in memory, signaling the end of line 20 and the new end of the program. (Try this later to see for yourself.) So, to merge programs, we simply have to make sure that we load in the new section at the address of the middle zero (2055, in our example) within the three zeros which signal the end of our original program. What we need to do is raise the floor of BASIC to this new address, load the section to be merged, and then lower the floor to its original value (here, 2049).

Tinkering With BASIC

Raising the floor of BASIC is easy. The new address is simply POKEd into addresses 43 and 44 in low-byte, high-byte order. (HI = INT (ADDRESS#/256) :LO = ADDRESS#-256*HI.) Finding this new address is another matter, but fortunately, this turns out to be easy as well.

As I mentioned above, BASIC needs to know where it is safe to begin to store its variables. In other words, BASIC needs to know the first address to come after the three zeros which end the program. Hence, the computer stores this address in a pair of memory locations in the usual low-byte, high-byte form. In the 64, these locations are addresses 45 and 46.

To see this, enter PRINT PEEK(45),PEEK(46), and out should come the pair 9,8. Since the address 2057 is the first address to follow our sample program, and 2057 = 256*8 + 9, we have the expected result.

Now that we have the address of the first location after the end of the program, the rest is easy. The new program is simply loaded into memory two places before this location. In our example, we load at location 2055 (2057-2). And that's all there is to it.

The basic idea behind Merger should now be clear. Everything of interest is packed into line 4. First, for any program which begins with these lines, the new floor for BASIC is computed using the contents of locations 45 and 46, as described above. Next, the floor of BASIC is raised to the new location. As a result, any new program now loaded will start right at the tail end of the previous program – just where we want it.

```
1 PRINT "{CLR}{10 DOWN}";TAB(15)"MERGE?"
  :GETA$:IFA$=""THENFORI=0TO500:NEXT:GOT
  01
2 IFA$<>"Y"THEN5
3 PRINT"{CLR}{10 DOWN}";TAB(5)"LOAD YOUR
   ADDITION.{2 SPACES}THEN POKE 43,1 AND
   44,8."
4 E=256*PEEK(46)+PEEK(45)-2:H=INT(E/256)
  :L=E-256*H:POKE43,L:POKE44,H:END
5 REM
```

The Automatic Proofreader

• IMPORTANT: We added a POKE to the "Automatic Proofreader" (October) to protect it from being erased when you LOAD another program from tape. The POKE does protect the Proofreader, and the Proofreader itself is not affected. However, a quirk in the VIC-20's operating system means that programs typed in with the Proofreader and SAVEd to tape cannot be LOADed properly later. If you LOAD a program SAVEd while the Proofreader was in memory, you see ?LOAD ERROR. This applies only to VIC tape SAVEs (disk SAVEs work OK, and the quirk was fixed in the 64). The solution is this special LOAD procedure:

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

The program will LOAD just fine in the future. This month, the Proofreader has been updated to prevent this problem. It also has been improved in other ways. Please observe these new procedures:

1. After first entering the Proofreader, SAVE it before typing RUN. The new Proofreader erases its BASIC loader from memory when first activated.
2. The new Proofreader checks itself for typing errors in the DATA statements when first RUN.
3. The new version now works on both the VIC and 64. Since the POKE to protect the Proofreader has been removed for the VIC's sake, when using tape you must reLOAD the Proofreader and RUN it whenever you type in a new program.
4. SAVE to tape wipes out the Proofreader, so press RUN/STOP-RESTORE before SAVE.

We strongly recommend that you type in the new version of the Proofreader and discard the old one. We apologize for any inconvenience this may have caused you.

"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. Watch out for typing an l instead of a 1, or an O instead of a 0, extra commas, etc.
2. SAVE it on tape or disk at least twice *before running it for the first time.* If you mistype the Proofreader, it may cause a system crash when you first run it. By SAVEing a copy beforehand, you can reLOAD it and hunt for your error. Also, you'll want a backup copy of the Proofreader because you'll use it again and again – every time you enter a program from COMPUTE!'s Gazette.
3. RUN the Proofreader. It will be POKEd into a relatively safe area of memory, the cassette buffer.
4. Type RUN to activate the Proofreader. If you ever need to reactivate it, just enter 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 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.

When you're done with the Proofreader, disable it by pressing RUN/STOP-RESTORE (hold down the RUN/STOP key and press RESTORE). If you need it again, enter SYS 886. It will then be ready once again to act as your personal typing aid. However, sometimes the Proofreader can be wiped out of memory. In this case, you'll have to reLOAD the Proofreader from tape or disk.

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
```

C. REGENA

DATA, READ And RESTORE Statements

By now you have typed in programs from COMPUTE!'s Gazette and probably have them running for your own use. If you have had any trouble, the most likely place for errors (other than simple typing errors that are easily recognized) is in DATA statements. This month we'll discuss some data examples that will help you understand how the data is used and why typing the DATA statements exactly is so important. If you do get some errors involving data, you can follow the computer's RESTORE and READ process to pinpoint trouble spots.

DATA statements are used in combination with READ statements. Generally, fewer program statements can be used in a DATA-READ procedure than using "assignment" (LET or A=4 type) statements. A DATA statement starts with the line number, then the word DATA (which may be abbreviated D shift-A), then numbers or words separated by commas. A DATA statement may *not* be combined with any other statements for the line number. A DATA statement may be placed anywhere in the program. If the computer comes to a DATA statement, the statement is ignored until a READ statement is encountered.

When the computer comes to the first READ statement, it then looks for the first DATA statement and assigns the value in the DATA statement to the variable in the READ statement. Numbers or strings may be used, and all items are separated by commas. You may READ one or any number of items. When the computer meets another READ statement, the computer will read the very next data item which hasn't been used, whether it is in the same DATA statement previously used or not. The computer keeps track of where it is in a series of data items. Let's look at an example.

```
10 READ A,B,C,D,E,F
20 DATA 3,4,7,2,7,8
30 PRINT A+B,C-D,E*F
```

These three lines are equivalent to:

```
10 A=3
20 B=4
30 C=7
40 D=2
50 E=7
60 F=8
70 PRINT A+B,C-D,E*F
```

When the computer executes line 10, the instructions are to read and assign values to the variables whose names are A, B, C, D, E, and F. The computer finds the values in the DATA statement, line 20. The computer assigns the first data item to A, the second data item to B, and so forth according to the READ statement. You can try typing the DATA statement as line 5 or line 35 instead of line 20, and you'll see it doesn't matter where the DATA statement is located. If you have several READ statements and several DATA statements, however, then your DATA statements must be in the proper sequence matching the READ statements, but it doesn't matter where they are placed in the program.

It also doesn't matter if you type a few extra numbers in the DATA statement; the extra items will just be ignored until another READ statement needs them. However, you must have enough items to satisfy the READ statement. If you have too few items, you will get an OUT OF DATA error.

Matching Variables To Data

Another stipulation is that you need to match strings with string variable names. If you use a string variable in the READ statement, the item in the DATA statement will become the value for the string variable (and it may be a number or alphabetic characters or symbols). If you want to READ a numeric variable, you must have a cor-

COMPUTER MAIL ORDER

VIC-20 $99

commodore COMPUTER

COMMODORE 64. $219

HARDWARE

1525 80 Column Printer	$219.00
1600 VIC Modem	$59.00
1610 VIC Term 40	$49.00
1650 AD/AA Modem	$89.00
1701 14" Color Monitor	$249.00
1311 Joysticks (each)	$5.99
1312 Paddles	$11.99
1110 VIC 8K	$42.00
1111 VIC 16K	$69.00
1011 RS-232 Interface	$42.00
1211 Super Expander	$53.00

1541 SINGLE DISK DRIVE $249

1530 DATASETTE $69.00

1520 COLOR PRINTER/PLOTTER $169

PRINTERS

Epson (MX80 FT, MX100, RX80, FX80, FX100)	CALL
Okidata (82, 83, 84, 92, 93)	CALL
Star Gemini 10X	$299.00
Smith Corona TP-1	$469.00
C.Itoh Prowriter 8510P	$379.00
C.Itoh Gorilla	$209.00

SOFTWARE

commodore

1906 Super Alien	$14.00
1910 Radar Rat Race	$14.00
1917 VooDoo Castle	$19.00
1922 Cosmic Cruncher	$14.00
1923 Gorf	$17.00
1924 Omega Race	$17.00
110 VIC Reference Guide	$15.00
CBM 64 Reference Guide	$18.00

EASY BUSINESS SERIES
C-64 Disks

EasyCalc 64	$59.00
EasyFinance 64	$19.00
EasyMail 64	$19.00
EasyScript 64	$39.00
Word/Name Machine	$16.00

PROGRAMMER SERIES
C-64 Disks

Assembler	$16.00
Logo	$39.00
Pilot	$39.00
Pet Emulator	$16.00
Screen Editor	$16.00
Video/Music Support	$15.00

ART & MUSIC SERIES
C-64 Disks

Music Machine	$16.00
Music Composer	$16.00
Meta Music I	$39.00

FLOPPY DISKS

Maxell	$32.00
Verbatum	$26.00
Elephant	$18.95

CARDCO.

Light Pen	$32.00
Cassette Interface	$29.00
Parallel Printer Interface	$69.00
3-Slot VIC Expansion Interface	$32.00
6-Slot VIC Expansion Interface	$79.00

ARTWORX
C-64 Cassettes

Bridge 3.0	$14.95

C-64/VIC 20 Cassettes

Teacher's Pet	$11.95

BRODERBUND
VIC 20 Cassettes

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

COMMERCIAL DATA
VIC 20 Cassettes

Motor Mouse	$23.00
Centipode	$23.00
Road Toad	$23.00

C-64 Cassettes

Road Toad	$23.00

CREATIVE SOFTWARE
VIC 20 Cassettes

Home Finance	$16.00
Home Inventory	$16.00

VIC 20 Cartridges

Astro Blitz	$29.00
Black Hole	$29.00
Trashman	$29.00
Choplifter	$29.00

EPYX
VIC 20 Cassettes

Sword of Fargoal	$24.00
Rescue at Rigel	$24.00

QUICK BROWN FOX
VIC 20 Cartridges

Word Processor	$49.00

UMI
VIC 20 Cartridges

Amok	$30.00
Meteor Run	$40.00
Alien Blitz	$30.00

VIC 20 Cassettes

Viterm A	$16.00
The Alien	$16.00

HES
VIC 20 Cassettes

Torg	$14.00
Skier	$14.95
Tank Wars	$14.95
Pak Bomber	$12.95
Dam Bomber	$12.95
Maze of Mikor	$14.95
Laser Blitz	$14.95
Pinball	$12.95
Tank Trap	$14.95

VIC 20 Cartridges

VIC Forth	$47.95
HES Mon	$29.00
HES Writer	$29.00
Aggressor	$31.95
Synthesound	$47.95
Shamus	$29.00
Protector	$32.00
Turtle Graphics	$31.95

C-64 Cartridges

HES Mon	$29.00
HES Writer	$35.00

NUFEKOP
VIC 20 Cassettes

Alien Panic	$10.00
Race Fun-Drag Race	$16.00
The Catup	$10.00
Exterminator	$19.00

C-64 Cassettes

3-D Man	$16.00

ROMOX

Princess and Frog (20)	$24.00
Anteater (20/64)	$26.00
Typo (20/64)	$26.00
Whiz Kid (20/64)	$26.00

RAINBOW
C-64 Disks

Personal Finance	$48.00
Writer's Assistant	$95.00

MICROSPEC
VIC 20 Cassettes

Spelling Bee-Grades 2, 3, 4, 5, or 6	$8.00
Math Drill	$8.00
Portfolio Manager	$16.00
Data Manager	$16.00

VIC 20 Disks

General Ledger	$69.00
Mailing List Manager	$35.00
Inventory Package	$69.00
Payroll	$69.00
Data Base	$49.00

C-64 Cassettes

Black Box	$12.00
Color Sketch	$20.00
Match Maker	$16.00

C-64 Disks

Mailing List Manager	$45.00
Inventory Package	$79.00
General Ledger	$79.00
Data Base	$69.00
Black Box	$16.00
Color Sketch	$22.00
Match Maker	$20.00

TRONIX
VIC 20 Cassettes

Galactic Blitz	$19.95
Swarm	$22.95
Sidewinder	$22.95

VICTORY
C-64/VIC 20 Cassettes

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

PROFESSIONAL SOFTWARE

Word Pro 64	$65.00

WEST 1-800-648-3311 EAST 1-800-233-8950

VISA

In NV call (702)588-5654. Dept 1106a. Order Status # 588-5654
P.O. Box 6689. Stateline, NV 89449

In PA call (717)327-9575. Dept. 1106a. Order Status # 327-9576
477 E. Third St. Williamsport, PA 17701

MasterCard

No risk, no deposit on C.O.D. orders. Pre-paid orders receive free shipping within the UPS continental United States with no waiting period for certified checks or money orders. Add 3% (minimum $3.00) shipping and handling on all C.O.D. and credit card orders. Larger shipments may require additional charges. NV and PA residents add sales tax. All items subject to availability and price change. We stock manufacturer's and 3rd party software for most all computers on the market. Call today for our catalog.

responding *number* in the DATA statement. If the computer finds an alphabetic character or symbol, you'll get a BAD DATA error.

Here's an example using string and numeric expressions:

```
10 FOR I=1 TO 4
20 READ NAME$,AGE,C$
30 PRINT
40 PRINT NAME$;" IS ";AGE;"YEARS OLD."
50 PRINT "FAVORITE COLOR: ";C$
60 NEXT I
70 DATA RICH,12,BLACK,BOB,7,BLUE,RANDY,3
80 DATA GREEN,GRANT,3,RED
```

This example illustrates a process repeated several times. Three variables are read each time, NAME$, AGE, and C$. The first time through the program loop, the data items are read and assigned as follows. NAME$="RICH", AGE=12, and C$="BLACK". Each time through the loop three more items are read. Note that it doesn't matter if all three items are not in the same DATA statement. Each DATA statement may consist of one item or several items separated by commas, although it saves memory to put as many items as possible in each DATA statement. The computer accepts up to four VIC or two Commodore 64 screen lines for each program line. Be sure you do not put a comma at the end of the list of items.

A common use of DATA and READ statements is to read variables into an array (subscripted variables). An example is:

```
10 FOR I=0 TO 10
20 READ A(I)
30 NEXT I
40 DATA 7,4,3,5,6,2,7,3,8,5,3
```

The first time through the loop, I is zero and A(0) will be given the value of 7, the first data item. The second time through the loop I is incremented to 1, and A(1) is assigned 4. The process continues.

Saving Memory And Time

If you like to write music on your computer, you may find using DATA statements is a way to play many notes without writing too many program lines. In the following examples I have not made the DATA lines as long as they could be; you can put as many items in one line as there is room. In these examples, line 10 turns on the volume. Lines 20-40 set up different variables for creating sounds on the Commodore 64. Lines 50-110 are the procedure to READ the note or tone number and the length from the DATA statements, then play the note. The length is determined in line 90.

Commodore 64 Version

```
10 POKE 54296,15
20 POKE 54277,4
30 V1=54273
40 V2=54272
50 FOR C=1 TO 14
60 READ N1,N2,L
70 POKE V1,N1:POKE V2,N2
80 POKE 54276,17
90 FOR D=1 TO 100*L:NEXT D
100 POKE 54276,16
110 NEXT C
120 POKE 54296,0
130 DATA 34,75,1,34,75,1,51,97,1,51,97
140 DATA 1,57,172,1,57,172,1,51,97,2
150 DATA 45,198,1,45,198,1,43,52,1
160 DATA 43,52,1,38,126,1,38,126,1
170 DATA 34,75,4
180 END
```

VIC-20 Version

```
10 POKE 36878,15
50 FOR C=1 TO 14
60 READ N,L
70 POKE 36876,N
90 FOR D=1 TO 100*L:NEXT D
100 POKE 36876,0
110 NEXT C
130 DATA 195,1,195,1,215,1,215,1
140 DATA 219,1,219,1,215,2,209,1
150 DATA 209,1,207,1,207,1,201,1
160 DATA 201,1,195,4
180 END
```

You may also like to try graphics using DATA and READ statements. To POKE graphics onto the screen, refer to the user's guide that came with your computer. Look at the screen and color memory maps, page 144 for the VIC-20 and pages 138-39 for the Commodore 64. These maps give you the location numbers. I usually sketch a picture on graph paper corresponding to these maps. The character numbers that you POKE or place on the screen are called screen codes or screen display codes and are on pages 141-42 in the VIC-20 manual and pages 132-34 in the 64 manual. The following sample programs clear the screen, then display 16 graphic characters to draw a picture.

Line 2 in the Commodore 64 version changes the screen to white. Line 5 clears the screen. Lines 10-50 POKE the graphics on the screen, and lines 60-90 contain the data for the graphics. Line 20 READs S, the screen location, and C, the screen character code number, then line 30 places that graphic character on the screen. Line 40 assigns yellow to that character's location.

VIC Face

```
5 PRINT"{CLR}"
10 FOR I=1 TO 16
20 READ S,C
30 POKE S,C
40 POKE S+30720,7
50 NEXT I
60 DATA 7865,85,7866,67,7867,67,7868,73
70 DATA 7887,66,7888,81,7889,81,7890,93
80 DATA 7909,66,7910,74,7911,75,7912,93
90 DATA 7931,74,7932,64,7933,64,7934,75
100 END
```

Write For FREE Catalog

VISA

(602) 855-3357

Call your order in and get a FREE program

NEW VIC SOFTWARE CBM 64

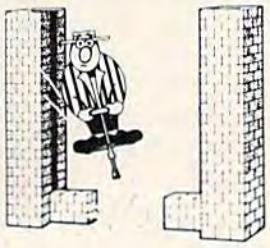
Call your order in and get a FREE program

Write For FREE Catalog

MasterCard

(602) 855-3357

CRICKET



POP TOP

AVAILABLE 11/1/83

SHOGUN

A New Hi-Res Graphic Adventure For The Vic-20. Different Every Time!

WORMS - CBM 64

PAK-20 — 20 Programs for the 64

GOLD QUEST

A Hi-Res Disk Adventure For The CBM-64.

TREASURE PAK

BUG BLAST

PARATROOPER

VARG

TARGET COMMAND

MOW

COSMIC CRUZER

SOFTWARE FROM COMPUTERMAT WILL TURN YOUR VIC OR 64 INTO A HOME ARCADE.

MUSIC MAKER

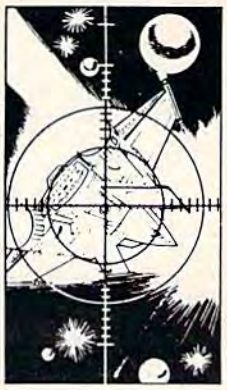
COMMODORE 64

(In Text)

Four Programs Include:

1. Music Maker
2. Music Player
3. Player Utility
4. Music Demo

WE HAVE MORE GAMES AND CHALLENGING PROGRAMS THAN YOU CAN SHAKE A JOYSTICK AT.



SPACE PAK

ALIEN INVASION

64 ARCADE PAK

STOMPERS - 64

SUPER PAK - 64

SNAKE OUT

HEAD ON

ComputerMat • Box 1664 C • Lake Havasu City, Az. 86403 *Add $1.00 For Shipping*

64 Face

```
2 POKE 53281,1
5 PRINT "{CLR}"
10 FOR I=1 TO 16
20 READ S,C
30 POKE S,C
40 POKE S+54272,7
50 NEXT I
60 DATA 1402,85,1403,67,1404,67,1405,73
70 DATA 1442,66,1443,81,1444,81,1445,93
80 DATA 1482,66,1483,74,1484,75,1485,93
90 DATA 1522,74,1523,64,1524,64,1525,75
100 END
```

The RESTORE Statement

The RESTORE statement tells the computer to RESTORE the data and start with the very first data item on the next READ statement. You may want to use the RESTORE statement if you want to do a procedure more than once but hate typing numbers in DATA statements. If you program a song, you can play the song more than one time by putting a RESTORE statement before the first statement that reads the first note, and put the whole procedure in a FOR-NEXT loop for the number of times you want the song to play, or a GOTO loop that plays continuously.

Let's look at an example. Suppose you have just drawn a face with one of the above programs. Now you want another face in another section of the screen. The screen characters used will be the same, and the relative positions of the characters will be the same. If we would like to use the same DATA statements we already have in the program, we can RESTORE the data for the next procedure. Try it by adding these lines to the "VIC Face" or "64 Face" program above.

```
92 RESTORE
93 FOR I=1 TO 16
94 READ S,C
95 POKE S+104,C
96 POKE S+104+30720,3
97 NEXT I
```

For the Commodore 64, change line 96 to:

```
96 POKE S+104+54272,3
```

With a little practice and practical experience you'll be able to see patterns in your work and learn to use DATA and READ statements along with RESTORE statements in the most efficient way.

Debugging

I mentioned that I think DATA statements are the most likely place to have an error when you are typing in a program from a published listing. Here are some suggestions to help you find the error.

When the program stops with an error message, you can PRINT the values of any variable and the computer will tell you the current value of that variable. As soon as you edit the program (make changes or add or delete a line), the variables will be zero, so PRINT the values first.

Let's say you are trying to RUN the VIC Face program above (the first version without the RESTORE lines). Now suppose the program prints part of the face, then stops with an error message. First, you can look to see how much of the face appeared before the error. If about half the face got printed, then the first half of the DATA statement items are OK. You may also type PRINT I and press RETURN, and the current value of I will be printed. You may also PRINT S,C to see what the values of S and C are. S should be a four-digit number starting with 7, and C should be a two-digit character number. If you know the value for I and the face looks all right as drawn so far, you can count the number of pairs of numbers that you know are OK to try to pinpoint the trouble area. You may also list certain line numbers. For example, type LIST 60-90, then press RETURN to see the DATA statements.

When you list the DATA statements, you can compare the screen listing to the published listing. Make sure you distinguish between the number zero and the letter O. Make sure you have the right number of commas in the right order.

If you get an OUT OF DATA error, the computer has read all the data items but needs more. List the DATA statements and make sure you have the right number of commas. You should also check any program statements, such as FOR statements, to make sure the numbers are typed correctly. The program may be trying to make too many loops. In the Face programs, you could LIST 10 to make sure line 10 ends with a 16 and not a 17.

If you get a BAD DATA error, it means the computer was trying to read a value for a numeric variable but the DATA statement contained a string. Again, check for commas in the right places. You may also try PRINTing some previous variables to see the latest correct value.

Bake A Cake

This program illustrates the use of DATA, READ, and RESTORE statements in a recipe file. First, you may go through a list of ingredients and press Y (yes) if you have the ingredient or N (no) if you do not have the ingredient. When the inventory list is complete, the computer program will tell you which cakes can be made with the ingredients you have. You may then choose a cake recipe, and that recipe will be printed on the screen. You then have the option to convert the recipe. If you would like to convert the recipe, enter a number or decimal fraction – such as 3 to triple the recipe or .5 to halve the recipe. The converted recipe will be printed.

Finally! An Affordable Full-Size, Full-Feature PRINTER

SUG. LIST $299

$229.95!!

For your **VIC-20®, C-64® ATARI®**
Centronics Parallel Types
And RS-232 Serial Types

BASIC PRINTER
(Requires one Option Below)

FEATURES:

- Full graphics capability.
- In the graphic mode, a column of graphic data can be repeated as many times as you want with a single command.
- Double width character output under software control (5 char. per inch).
- Print position addressable by character or dot (positioning control).
- Graphic character and double width character modes can be intermixed on a single line.
- Automatic printing. When the text exceeds the maximum line length no data is lost due to overflow.
- Self-test printing mode.
- Paper width is adjustable up to 10 inches. Standard plain paper.
- 50 cps print speed.
- 80 characters per line.
- 5 × 7 dot matrix.
- Full 2 yr. Warranty.
- Foreign character sets For U.S., U.K., Sweden, and Germany.

Any of these **Options** allow you to connect and print - cables included.

APROPRINT-2064™ (pictured) . . . .Add: $35.95
For Commodore **VIC-20 & C-64** - Cable included.

APROPRINT-4080™ . . . . . .Add: $45.95
For **all Atari Computers** - Cable included.

APROPRINT-1000™ . . . . . .Add: $29.95
RS-232-Serial - Name your computer

APROPRINT-8000™ . . . . . .Add: $29.95
Centronics type Parallel - Name your computer

ADD: $8.00 shipping (cont. USA), $25.00 (Canada, HI, AK)

(All other foreign orders Add $55.00 (shipped by Air)

The ONE VIC-20® Memory Expansion Board that DOES IT ALL!

Maximum Memory allows you to use more powerful programs for:
- EDUCATION • ENTERTAINMENT • MAIL LISTS
- BUSINESS APPLICATIONS • FINANCIAL RECORDS

APROPOS TECHNOLOGY

RAMAX

A perfect investment to give your family and yourself more enjoyment and use from your home computer! The ease of operation, the neat appearance, and the real POWER it adds to your VIC at this low price makes it a MUST for every VIC home!

SPECIAL LOW PRICE!

Only $124.95

Price includes shipping and handling within Continental USA. Foreign orders please add $15.00. Calif. Residents add 6% sales tax.

10 DAY MONEY-BACK GUARANTEE
If not satisfied, simply return in original condition for your money back.

RAMAX Jr.™

Already own an 8k Expander? Get the NEW **RAMAX Jr.™**! Identical to the RAMAX™ except with 19k instead of 27k. Our instructions will show you how to use your 8k as BLK 3 with Jr. to get the full complement of memory!

Special Only $109.95
Shipping included

To equal the total memory of RAMAX™ you would have to buy a 16k Memory Expansion, PLUS an 8k Expansion, PLUS 3k Expansion. THEN you would need a "mother board". With RAMAX™ you buy just ONE piece . . . at ABOUT HALF THE PRICE!

RAMAX™ Features and Specifications:

- Adds up to a full 27k bytes of additional RAM to the standard VIC-20's internal RAM of 5k.
- Built-in switch allows User selection of any combination of 5 areas or RAM memory*:
  BLK1 (8k: Adr. 8192-16383
  BLK 2 (8k: Adr. 16384-24575)
  BLK 3 (8k: Adr. 24576-32767)
  BLK 5 (Adr. 40960-49151, allows/disallows 8k ROM games)
  RAM (3k: Adr. 1024-4095)
  RESET (Resets computer without power off/on)
- Built-in electrical Fuse to protect equipment.
- Totally self-contained. No external power supply needed.
- Two (2) extension connectors allow ANY additional cartridges and/or devices designed for the VIC expansion port.
- Very low power consumption (.175 amp usual).
- High reliability gold-plated connectors are designed for long life.
- Complete Operating Manual.
- 6 month parts and labor warranty to original purchaser.
- Factory service.

*Many VIC-20 cartridges and programs require certain configurations of the memory (i.e certain games will only run on the unexpanded VIC while others require the upper portion of the expanded memory). With RAMAX™ you have switches that turn-on and turn-off portions of the memory to provide the right area of memory - all without plugging or unplugging. It's so easy!

TO ORDER:

Send Check or Money Order For the Total
Calif. residents add 6% tax.
Or Contact your Local Dealer
Phone orders Call **(805) 482-3604**

VISA All Prices U.S. Dollars MasterCard

DEALER INQUIRIES WELCOME

WE SERVICE WHAT WE SELL
VIC-20 & Commodore-64 are registered trademarks of Commodore International. Atari is a trademark of Atari Inc.

APROPOS TECHNOLOGY
1071-A Avenida Acaso
Camarillo, CA 93010

New Product!
APROSPAND-64™ Gives your Commodore 64 full expandability. This superbly designed expansion module plugs into the 64 & gives you 4 switchable (singly or in any combination) expansion connectors - plus fuse protection - plus a reset button! only **$54.95**
Shipping Included

APROPOS TECHNOLOGY

For convenience in programming (especially with the limited memory of the VIC-20), the amounts in the recipes are given in decimals. For example, 2/3 cup sugar is written as .67 c. sugar. In the Commodore 64 version you may want to change the decimals to fractions.

The DATA statements in lines 101-107 contain the measure, then the ingredient, for 25 ingredients. Line 3 READs A$, the measure, and B$(N,0), the name of the ingredient, then assigns I$(N) equal to the measure plus a space plus the name of the ingredient. By the way, that's a zero in the subscript for B$. B$ is used in the inventory list, and I$ is used in printing the recipe. Z is the number of ingredients minus one (because the subscripts start with the number zero).

As you are typing the DATA statements, you may notice two or more commas together with nothing between them (,,,). Be sure you get the right number of commas as you are typing. This indicates a null string, or a string variable equal to "". You may also use "" in your DATA statements (two double quote marks with no space between).

The recipes are in the DATA statements in lines 109-123. The first item is the name of the cake. The next items are the amounts of the ingredients in the following order: cups shortening, cups flour, cups sugar, cups brown sugar, tsp. baking powder, tsp. salt, tsp. soda, cups cherry juice, number of cherries, cups bananas (mashed), cups sauerkraut, cups milk, cups buttermilk, number of eggs, number of egg whites, tsp. red food coloring, ounces of chocolate, tbsp. cocoa, tsp. vanilla, tsp. cinnamon, tsp. nutmeg, tsp. vinegar, cups salad oil, cups water, and cups of oatmeal. The DATA will contain a null string if the cake does not contain that ingredient.

An example is the first part of line 109, the data for banana cake:

```
109 DATA BANANA,.67,2.5,1.67,,1,1,1,,,1,,
    ,.67,2,,,,,,,,,,,CHERRY ...
```

The name of the cake is BANANA. The recipe is .67 cup shortening, 2.5 cups flour, 1.67 cups sugar, 1 tsp. baking powder, 1 tsp. salt, 1 tsp. soda, 1 cup mashed bananas, .67 cup buttermilk, and 2 eggs.

Program Explanation

How does the program work? To print a recipe, the data is RESTOREd first, then the first 50 ingredients and measures are read and ignored (line 73). The key you pressed to choose a recipe is E$, and the ASCII code will be 65 for A, 66 for B, and so forth. Lines 75-77 figure out which recipe was chosen and read through the title and ingredients to get to the appropriate recipe. Line 79 prints the title of the cake. Line 81 reads the amount from the DATA statement. If the amount is a null or zero, then that ingredient is not printed. However, if there is a value, the value is printed, followed by the corresponding measure and name of ingredient from the I$ array.

Line 83 defines variables M(I) for measure and C$(I) for ingredient for only those ingredients in the recipe. These values are used in printing the converted recipe, lines 97-99.

For the inventory list, the computer keeps track of your Y or N answers in the B$(N,1) array, where N varies from 0 to 24 for the ingredients. Line 33 checks to see if an N is stored as a no answer for flour, sugar, or salt. If any one of these three items has a no, then no cakes can be baked. Y is a variable for the number of Y answers, and if there are not enough ingredients with a Y answer you cannot make a cake (checked in line 35).

Line 47 RESTOREs the data, then ignores the first 50 items. Lines 49-51 check through the recipe for each cake. If there is an amount listed for an ingredient, then the corresponding B$(N,1) value is checked. If it is N for no, then you are lacking one of the ingredients required for the cake. The rest of the ingredients are skipped over and the computer goes to the next recipe. If each of the required ingredients also has a B$ value of Y for yes, then the cake can be made and the name of the cake is printed.

These are real recipes that really work. No baking instructions are given because there isn't room in the VIC-20 version, but usually the person baking knows how to bake the cake and just needs to be reminded of the amount of each ingredient. In case you want to try these recipes, all are for cakes to be baked in two 9-inch layers. Mix the shortening with the sugars, then add the dry ingredients alternately with the liquid, then add the eggs and vanilla. Bake at 350 to 375 degrees.

In the cherry cake, cut up the cherries before adding to the mixture. For the oatmeal cake, prepare the oatmeal separately. While the oatmeal cools, mix everything else, then add the oatmeal to it. The wacky cake can be mixed all at once. Just dump everything into one big bowl and mix it up. For the red velvet cake combine the ingredients as usual, except for the vinegar and soda. Mix the vinegar and soda together and fold into the rest of the batter. The sauerkraut cake is worth trying – just don't tell anyone what it is until after they have eaten it. Rinse the sauerkraut well, then chop it into small pieces before combining it with the rest of the cake batter.

Typing In The Program

The VIC-20 version is very close to full memory. Be sure to leave out all the spaces and be sure to use the abbreviations for all the BASIC words, such as ? for PRINT and D shift-A for DATA. If you need to

SJB DISTRIBUTORS. THE MOST COMPETITIVE PRICES ON COMMODORE.

commodore

NEW COMMODORE PRODUCTS

The Executive 64	Call
CBM B128-80	$ 825
CBM B256-80	1095
CBM BX700	2990
B Series Software	Call
CBM 1520 Plotter	169
CBM 1526 Printer	349

SOFTWARE FOR CBM 64

BUSINESS

WordPro 3+/64 w/Spell Right Plus	$ 79
Spell Right Plus	55
Calc Result (Advanced)	125
Calc Result (Easy)	75
Busicalc II	95
Mirage Concepts (Powerful Data Base)	95
M File (merge with WordPro)	89
Home Utilities	49
64 Mailing List (Galactic)	28
The Manager	50
Home Accountant (continental)	75
Code Writer (Writes Basic Programs)	95
Stock (investment analysis)	80
Agricultural Management	Call
General Ledger, A/R, A/P, P/R, Inv	Call

RECREATION

Assembler Package (cassette or disk, compiled, includes editor, loader, disassembler)	39
Sprite Master (access)	30
Neutral Zone (access)	35
Space Belt	19
Pet Emulator	30
Coco II (build your own games)	40
Vic Tree (programmers utilities)	75
Micro-Term (save to printer,disk)	39
Hesmon	35
Synthesound	45
Gothmogs Lair	30
Road Toad	15
Commodore Games	Call

INTERFACES & ACCESSORIES

80 Column Expander	$159
VIC 1600 Modem	95
VIC 1650 (auto answer, auto dial)	150
VIC 1525 Graphic Printer	225
VIC 1530 Datasette Recorder	65
VIC 1541 Disk Drive	249
VIC Switch (connect 8 64's or Vics to printer, dd)	149
PET-IEEE cable	33
IEEE-IEEE cable (2m)	49
5 Slot Expander for 64	65
Parallel Interface (Epson, Okidata, IDS, NEC)	70
Programmers Reference Guide	18
Verbatim Diskettes (10 per box)	26
Hes Modem	75
ADA 1450	149
ADA 1800 (new)	129
Numeric Keypad	35

VIC PRODUCTS & ACCESSORIES

8K RAM Memory Expansion Cartridge	$ 40
16K RAM	70
24K RAM	105
VIC 3 Slot Expander	27
VIC 6 Slot Expander	70
Gorf (64 also)	30
Omega Race	30
Arcade Joystick - Heavy duty w/2 firing buttons! Great for the VIC or 64	25
Auto Clock	125

MONITORS - GREAT RESOLUTION (64 OR VIC)

CBM 1701 Color Monitor	$ 249
Amdek Color Plus	299
Panasonic TR-120 (w/speaker)	155
Panasonic CT-160	279
BMC (green screen)	95
Video/Audio Cable	15

PRINTERS - LETTER QUALITY

CBM 6400, 40 cps	$1450
Diablo 620, 25 cps	949
Transtar 140 (serial)	1395
Transtar 130, 16 cps (auto load, wp features!)	769
NEC 3500 Series	1600
NEC 7700 Series	2350

PRINTERS - DOT MATRIX

CBM 8023, 150 cps/graphics	$ 545
CBM 4023 Printer	395
Epson FX Printer, 160 cps	549
Epson MX-80 FT w/graftrax	Call
Epson FX-100	859
Okidata 82A, 120 cps (serial and parallel)	429
NEC 8023A (parallel)	429
Okidata 92	559
Star Gemini, 10X	329
Star Gemini, 15	499
Transtar 315 (hi-res, color)	575

COMMODORE BUSINESS SERIES

SuperPet (5 languages, 2 processors)	$1059
CBM 8032 Computer, 80 Column	625
CBM Memory Expansion, 64K	259
CBM 8050, 1 mg. Dual Drive	995
CBM 8250, 2 mg. Dual Drive	1295
CBM D9060, 5 mg. Hard Disk	1995
CBM D9090, 7.5 mg. Hard Disk	2250
CBM 2031, 170K Single Drive (New)	295
DC Hayes Smart Modem	220

BUSINESS SOFTWARE-8032

WordPro 4+ or 5+	$ 309
InfoPro	219
Administrator	489
VisiCalc (expanded)	199
BPI A/R, G/L, Job Cost, Inventory, Payroll	ea.325

MasterCard, Visa, Money Order, Bank Check

COD (add $5) accepted.
Add 3% surcharge for credit cards.
In stock items shipped within 48 hours.
F.O.B, Dallas, Texas (Texas Res. add 5% tax)
All products shipped with manufacturer's warranty.
Prices are subject to change without notice.

TO ORDER
CALL TOLL FREE
800-527-4893
800-442-1048
(Within Texas)
Business Hours
Mon.- Fri. 8 to 6, Sat. 10-2
Write for free catalog.

SOFTWARE OF THE MONTH

Mirage Concepts
Machine Language Data Base
2000 Characters PL Record
200 Fields Per Record
Multiple Files Per Disk

PRODUCT OF THE MONTH

INTERPOD (intelligent IEEE RS232, serial interface for VIC or C64)	$ 179

SJB DISTRIBUTORS INC.
10520 Plano Road, Suite 206
Dallas, Texas 75238
(214) 343-1328

edit a line, be sure the edited line also contains the abbreviations.

Be sure you copy the DATA statements exactly for the correct recipes. There are no spaces except in the names of the cakes. If there are commas together, do not put spaces between the commas.

For the Commodore 64 version, please add line 2 POKE 53281,1 to change to a white screen, and change the following lines:

```
17 PRINT"{CLR}{BLU}IN THE FOLLOWING LIST,
    PRESS":PRINT "{2 SPACES}'Y' IF YOU HA
   VE THE INGREDIENT"
19 PRINT "{2 SPACES}'N'IF YOU DO NOT
   {DOWN}":PRINT"{2 SPACES}'S' TO START O
   VER.{2 DOWN}"
63 PRINT "D DEVIL'S FOOD CAKE":PRINT "E G
   OLD LAYER CAKE":PRINT "F OATMEAL CAKE"
99 FOR N=Ø TO I-1:PRINT INT(F*M(N)*1ØØ)/1
   ØØ;TAB(6);C$(N):NEXT:GOTO87
```

These lines adjust the printing for the Commodore 64 screen.

If you prefer to save typing effort and time, you may receive a copy of "Bake A Cake" by sending $3, a blank cassette, and a stamped, self-addressed mailer to C. Regena, P.O. Box 1502, Cedar City, Utah 84720. Be sure to specify which computer version.

See program listing on page 202.

One-Touch Commands For The 64

David Martin

This utility program is an ideal application for the normally unused special function keys to the right of the Commodore 64 keyboard. An entire command can be typed on the screen with a single key press. Programmers in particular will appreciate the repetitive typing this technique can save.

Unlike people, computers excel at performing boring, repetitive tasks. What's more, time-consuming tasks which annoy us can be performed by an uncomplaining computer in a fraction of a second. So it only makes sense to let computers handle the simple little things they do best.

One of these tiny jobs is the routine typing of frequently used commands. During a session with your computer, how many times do you type RUN, LIST, SAVE, or LOAD? Probably many more times than you think. If you're a hunt-and-peck person new to typewriter-style keyboards, this can be a major annoyance. Even if you're a fast touch-typist, you probably stumble over such often-used commands as POKE 53281,1:PRINT {BLK} (which sets up an easier-to-read white screen background with contrasting black characters).

The utility presented here can liberate you from all that. It redefines the special function keys (F1 through F8 to the right of the keyboard) so that a single key press enters a whole command. The short while it takes to type in this program can pay for itself many times over.

One-Touch Commands

Be sure to type the program carefully. As always, save it twice on tape or disk before running it for the first time. The program is in the familiar form of a BASIC loader – a BASIC program which includes a machine language program encoded in DATA statements. A mistyped number can "crash" the computer when the program is first run, forcing you to switch off/on to clear the machine. Saving the program beforehand can keep you from losing all your work.

Actually, this BASIC loader contains two machine language programs. Neither program consumes any memory normally used by BASIC (see Programmer's Notes below). After activating the utility, it erases the BASIC loader from memory and allows you to load your own programs. The utility keeps working "in the background," so to speak, until you turn off the computer or reset it by pressing RUN/STOP-RESTORE.

The utility is very easy to use. First, enter and run the BASIC loader. You'll see a screen prompt which asks:

```
F1?
```

Now, type in whatever command you'd like to have available at a stroke of the F1 key. Then press RETURN. For instance, if you answer the prompt by typing LIST and pressing RETURN, then hitting F1 after the utility is activated will print the command LIST on the screen.

There's a way to save even more key strokes, too. If you answer the prompt by typing the command followed by a reverse arrow – using the reverse-arrow key in the upper-left corner of the keyboard – then the utility will even press RETURN for you, when activated. Otherwise, it will be up to you to press RETURN when using each command. In other words, answering the prompt like this:

```
F1? LIST [Press RETURN]
```

means that when the utility is working, it will type the command LIST on the screen for you, but you'll still have to press RETURN yourself to actually execute the command. But if you answer the prompt like this:

F1? LIST ← [*Press RETURN*]

it means the utility, when working, will type LIST *and* press RETURN for you when you hit the F1 key. The reverse-arrow makes the command *self-executing*. Pressing the function key will execute the command instantly. Depending on the command, this may or may not be desirable. For instance, you probably wouldn't want the command NEW to execute instantly because it would be too easy to accidentally wipe out a BASIC program. (In fact, you probably wouldn't want to program a function key with NEW at all.)

You can also answer the prompt with more than one command. An example might be:

F1? LOAD ← RUN ← [*Press RETURN*]

which means F1 will automatically LOAD and RUN the next program from tape.

After answering the F1? prompt, the utility asks for F2, F3, and so on through F8. After F8, the utility immediately activates itself and erases the BASIC loader from memory (you did SAVE it, didn't you?).

The function keys are now programmed. They will remain so until you shut off the computer or trigger a "warm start" by pressing RUN/STOP-RESTORE.

Programmer's Notes

The one-touch command utility consists of two machine language programs tucked away in different parts of the Commodore 64's memory. The first part is in the cassette buffer, starting at memory location 828 ($033C hexadecimal). This program asks for the key definitions. Each time RETURN is pressed, it stores the ASCII values of the characters into high memory.

After entering F8, control jumps to the second program, stored in high memory at location 49152 ($C000 hex). This is a 4K block of unused memory in the 64. The first two POKEs in the first line of the BASIC loader fool BASIC into thinking that memory ends at 53248. To restore normal vectors, you can enter POKE 56,160:POKE 55,0.

The first machine language program also sets up an interrupt. Every sixtieth of a second, the computer checks the second program to see if a function key has been pressed. If so, the key's definition is printed on the screen. If a reverse-arrow was defined after the command, the program forces a RETURN to execute the command also.

See program listing on page 202.

VIC/64 Disk Defaulter

Eric Brandon, Programming Assistant

This month's "Power BASIC" — a continuing series of useful utilities and routines — saves typing for people who regularly use a disk drive instead of a cassette recorder. The machine language routines are in the form of easy-to-use BASIC loaders.

When Commodore designed the operating system used in the VIC-20 and Commodore 64, the designers assumed that most people would be using a cassette recorder for storage instead of the more expensive disk drive. That's why, when you type LOAD or SAVE, the computer responds by prompting "Press Play On Tape" or "Press Record & Play On Tape." It *defaults* to the tape recorder.

If you're using a disk drive, you have to type the device number —,8— after each command (as in LOAD"filename",8). This can become bothersome after a while.

"Disk Defaulter" is a short utility, written in machine language, that modifies the computer's operating system to recognize the disk drive as the default device instead of the cassette recorder. As long as the utility is activated, you no longer have to append ,8 to the LOAD, SAVE, and VERIFY commands.

To use Disk Defaulter, enter Program 1 for the VIC-20 or Program 2 for the Commodore 64. When you type RUN, this BASIC loader will POKE the machine language into some free memory space and activate the utility. To turn it off (for instance, if you want to use cassette), press RUN/STOP-RESTORE. To turn it back on, type SYS 679.

To load machine language programs, you still must type LOAD"filename ",8,1. Also, pressing SHIFT-RUN/STOP will not access the disk drive because it results in a "Missing Filename Error." But otherwise, all LOAD, SAVE, and VERIFY commands will refer to disk.

The only program we have found that will interfere with Disk Defaulter is the PAL Assembler for the Commodore 64.

Program 1: VIC Version

```
10 I=679
20 READ A:IF A=256 THEN 1000
30 POKE I,A:I=I+1:GOTO 20
679 DATA 169,188,141,48,3,169,2
686 DATA 141,49,3,169,195,141,50
693 DATA 3,169,2,141,51,3,96
700 DATA 162,8,134,186,76,73,245
707 DATA 162,8,134,186,76,133,246,256
1000 PRINT"DISK DEFAULTER ACTIVATED
1010 PRINT"USE RUN/STOP RESTORE TO DEACTI
     VATE
1020 PRINT"TYPE SYS 679 TO REACTIVATE
1030 SYS 679
```

Program 2: 64 Version

```
10 I=679
20 READ A:IF A=256 THEN1000
30 POKE I,A:I=I+1:GOTO20
679 DATA 169,188,141,48,3,169,2
686 DATA 141,49,3,169,195,141,50
693 DATA 3,169,2,141,51,3,96
700 DATA 162,8,134,186,76,165,244
707 DATA 162,8,134,186,76,237,245,256
1000 PRINT"DISK DEFAULTER ACTIVATED
1010 PRINT"USE RUN/STOP RESTORE TO DEACTI
     VATE
1020 PRINT"TYPE SYS 679 TO REACTIVATE
1030 SYS 679
```

EPYX
TOP 10 GAME SALE

* BUY TWO — GET ONE FREE *

SPECIAL FREE GAME
Buy Two Epyx Games From Protecto And Get A Free Game

The Thinking Man's Paradise

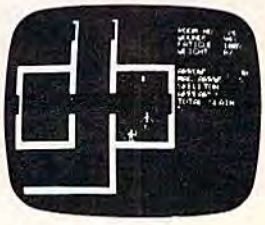
(Disk / Cassette)

- **Award Winning Games**
- **Fantastic Graphics**
- **Skill (not luck) Needed**
- **Already in top 10 charts**

COMMODORE - 64 / VIC-20

Buy Any Two Epyx Games From Protecto And Send The Proof Of Purchase Seals To Epyx And Epyx Will Send You A Free Game.

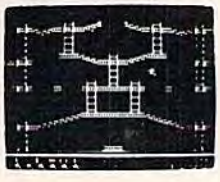
JUMPMAN
If you like Donkey Kong, you'll love Jumpman. Over 30 different screens with 8 speeds and 5 skill levels make this the fastest action game in the country. You must leap girders, climb ropes, and scale ladders to reach and diffuse bombs while avoiding robots, birds, bullets, explosives, crumbling walls, vanishing escape routes, and many other obstacles. (Truly a fantastic game!)
List $39.95 **Sale $27.95**



TEMPLE OF APSHAI (computer game of the year)
This is the standard by which other adventure games are judged. Full color graphics portray the temple and all its contents - magic, monsters, doomed cities and damsels in distress. Do battle in real time with over 20 monsters, expansion modules will keep your adventure alive in the future.
List $39.95 **Sale $29.95**

SWORD OF FARGOAL
Search for the wondrous sword in the depths of an ever changing dungeon. Make yourself invisible, teleport to a new location, drink a healing potion or use enchanted treasures, but watch out for traps and hideous creatures who will try to stop you. (Fantastic dungeon adventure)
List $29.95 **Sale $21.95**

CRUSH CRUMBLE & CHOMP (Computer game of the year nominee)
Choose one of six monsters or create your own, and use your monster to destroy one of four unsuspecting cities. The cities aren't totally defenseless, they call on police, national guard, and even a mad scientist, complete with helicopter, to save humanity from the relentless threat.
List $29.95 **Sale $21.95**



JUMPMAN JUNIOR
The devilish ALIENATORS are back! And they have overrun the Jupiter Command Substation. In this cartridge format sequel to the best-selling Jumpman, players must leap through 12 all new screens featuring electrocution traps, moving walls, hellstones, and dangers much too bizarre to be believed. How many screens can you master? Twelve different screens, 8 speeds.
List $39.95 **Sale $27.95**

We have all Epyx games

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

The Assembler

People often use the words *machine language* and *assembly language* interchangeably. However, *machine language* is becoming the more common term; it is more accurate — when you program in this language, you're speaking directly to your computer in its native tongue.

Unfortunately, the computer's internal language is almost impossible for humans to work with. These machines communicate only with numbers, and very odd numbers at that. They're binary, consisting of only 1's and 0's, grouped together in eight-digit clusters called *bytes:* 01100111, 11110001, and so on. Humans find it easier to work with words. That's where an *assembler* comes in.

The Primary Tool

This month we're going to build the basic tool for machine language (ML) programming. Type in Program 1 and you'll have your own working assembler. (Be careful with line 244; it's too long to type in normally. You'll need to abbreviate some of the BASIC words to make it all fit on one line. Use gO for GOTO, ? for PRINT, and pO for POKE. Remember to SHIFT the second letter of each abbreviation.)

The assembler works like this: you type in a wordlike, three-letter code, and the assembler looks up the correct number (in the computer's language) and POKEs it into RAM memory to start forming an ML program. In a minute we'll create a simple ML program to show you how ML programming is done. But let's clear up a few possible sources of confusion first.

These wordlike codes are called *mnemonics,* which means they've been designed to be simple to remember. It's easy enough to remember what USA stands for. Likewise, you can quickly pick up the essential ML words. There are 56 of these commands available to you, roughly as many words as there are in BASIC. But, like BASIC, there is a core group of about 20 important ones. They are the only ones you need to use to get almost anything accomplished. What's more, the ML words *are* easy to learn and remember. For example, BRK stands for Break (like BASIC's STOP), JSR is Jump to SubRoutine (GOSUB), and RTS is ReTurn from Subroutine (RETURN). The command which does the same thing as BASIC's GOTO is called JMP, for JuMP.

A Kind Of Swing

ML programming involves a kind of swing between Command and Target. First you give a command, then you give the specific target for that command. Then another command, another target. These paired-event phenomena are called by many names and appear in many disguises in programming as well as in real life. They're called Operator/Operand, Instruction/Argument, Mnemonic/Address, Analyst/Analysand, Shopper/Apples, Thief/Victim. Notice that the first half of the pair is the more general, the second more specific. At a given moment, the apple is the specific thing the shopper's involved with, but the shopper will be buying other things during this visit to the store. Similarly, a thief is always a thief, but a victim is a victim only that once (we hope). Also, the transaction which all these pairs have in common is that the first half of the pair is *doing something to* the second half. Together they form a complete action in the sense that Open/Envelope or Eat/Peach are paired (command/target) actions.

A Robot Thanksgiving

If you think about it, you can see this do-it-to-it rhythm throughout BASIC programming: PEEK (8), PRINT "HELLO", SAVE "PROGRAM", X=15, X=X+1, GOTO 1500, etc. The reason

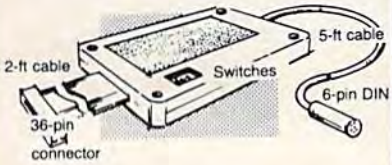
NEW!
Universal Input/Output Board for VIC-20/64

16-channel A/D converter & 1-channel D/A output; 16 high-current discrete outputs.

- University physics & electrical engineering labs and hospital applications.

VIC-20 uses MW-311V $205.00
CBM-64 uses MW-311C $225.00

MW-302: VIC-20/64 Parallel Printer Interface.



Works with all centronics type parallel matrix & letter printers and plotters—Epson, C.Itoh, Okidata, Nec, Gemini 10, TP-I Smith Corona, and most others. Hardware driven; works off the serial port. Quality construction: Steel DIN connectors & shielded cables. Has these switch selectable options: Device 4, 5, 6 or 7; ASCII or PET ASCII; 7-bit or 8-bit output; upper & lower case or upper only. Recommended by PROFESSIONAL SOFTWARE for WordPro 3 Plus for the 64, and by City Software for PaperClip.

MW-302 ..... $ 79.95

Dealer inquiries invited.

VISA MasterCard

Micro World Electronix, Inc.
3333 S. Wadsworth Blvd. #C105,
Lakewood, CO 80227
(303) 987-9532 or 987-2671

and The Alien said..."Let Commodore Speak... and Sing!"

Yes, the VOICE BOX™ from The **Alien** Group, the world's ONLY *singing* speech synthesizer, now grants the power of speech to the VIC 20™ and the Commodore 64™. A commented, all-BASIC demo program gets the VOICE BOX talking right away, and, since it can be "taught" to say anything, the VOICE BOX has an unlimited vocabulary! The voice speaks with natural speech inflection controlled either from the program or from the precise, built-in Pitch control. *No other speech synthesizer has this feature!*

Want to add speech to a new or existing BASIC program! The VOICE BOX has FOUR ways to do it on VIC 20's of any memory size and on any Commodore 64: entirely from BASIC, or using one of the three machine language programs readily added to other programs—English text-to-speech, the same with the lip-synch "Alien" face added, or use of the 64 basic phonemes as input. A challenging spelling quiz that accepts new words (expanded memory required with VIC 20) is provided on the cassette supplied.

The VOICE BOX plugs directly into the computers user port, comes with built-in speaker, Volume and Pitch controls and lots of instructions from The **Alien** Group, the people who got Atari® and Apple® to speak!

Available at leading computer stores everywhere, or order direct by sending $95.00 to: The **Alien** Group, 27 W. 23rd St., NY, NY 10010. Specify whether for VIC 20, Commodore 64 cassette or Commodore 64 disk. Programs for a high-res talking human face and a comprehensive music and singing system available on separate cassette for $25.00 (expanded RAM necessary when used on VIC 20). Extra main cassette for either computer available for $19.00.

Vic 20 and Commodore 64 are trademarks of Commodore Electronics Ltd. VOICE BOX is a trademark of The **Alien** Group

TRAIN DISPATCHER

Game for Commodore 64™ and VIC 20™ (Requires 16K memory)

REALISTIC SIMULATION OF A COMPUTER CONTROLLED CTC OFFICE.
HOW QUICKLY AND EFFICIENTLY CAN YOU MANEUVER TRAINS OVER A 20 STATION, 150 MILE TERRITORY?
BUILD YOUR OWN SKILLS FROM "CUB DISPATCHER" TO "TRAINMASTER".

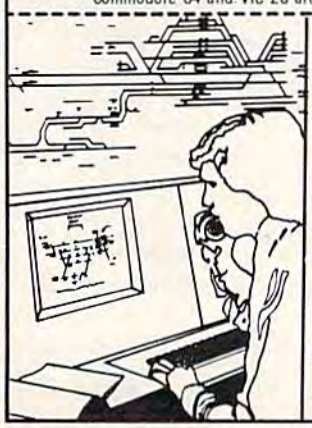
FEATURES

24 DISPLAYS including Train Sheets, Block Permits, Territory Overview, 20 Interlockings.
FULL CONTROL Throw switches, clear and cancel signals, route and maneuver up to 12 trains moving in both directions.
REAL TIME All displays continuously updated, including train locations, switch positions, signals, block permits, and crew time.
PERFORMANCE Updated and displayed continuously from cumulative speed of all trains.
VERSATILE Plays at 5 levels, from leisurely "Visitor" and "Cub Dispatcher" levels to frantic "Trainmaster".
AUTHENTIC Created by designers of CTC Systems for operating railroads.

TRAIN DISPATCHER was created as an action game, but it also provides a dramatic and realistic picture of true-to-life railroad operations.

As you develop and apply your own strategies for quickly and smoothly moving trains over the territory, you will increase your appreciation for actual railroad operations.

Commodore 64 and VIC 20 are registered Trademarks of Commodore Business Machines, Inc.



PURVIS SYSTEMS INCORPORATED
470 Streets Run Road Pittsburgh, PA 15236

Enclosed is $24.95 for Train Dispatcher + $2.50 for postage and handling. NY, RI, PA, WA, VA, and CA residents add state sales tax.

Charge To: ☐ MasterCard Card # _____
☐ VISA Exp. Date _____

CHECK ONE ☐ 64 ☐ VIC 20
CHECK ONE ☐ TAPE ☐ DISC

Signature _____
Name _____
Street _____
City _____ State, Zip _____

we're stressing this distinction, this rhythmic swing between actor and acted-upon, is because an ML program is constructed in precisely this way — you make a list of tiny, elementary actions for the computer to later carry out. It's like a robot Thanksgiving dinner: spear/turkey, raise/arm, insert/food, chew/morsel, lower/arm, spear . . . . But list enough of these mini-instructions and you can do amazing things.

One result of all this is that an ML program doesn't look like a BASIC program. BASIC tends to spread these pairs out along a line:

```
100 Y=3: X=X+1: POKE 63222,Y: Y=PEEK
    (1200)
```

ML lists each tiny action-pair on its own line:

```
100 LDY #3
110 INX
120 STY 63222
130 LDY 1200
```

These two programs are doing exactly the same thing, but in different ways. STY and LDY mean STore Y and LoaD Y (it's like a variable in BASIC). INX means INcrement X (raise it by one). The # sign means to think of the number as *literally* the number three, not address three. Without the #, the computer assumes you mean a memory location.

Take a look at the mnemonics here. They're all three-letter words. They are always the first thing on each line. And they usually have their target right next to them (the INX doesn't because the mnemonic itself already contains the specific information required). The other half of the pair, those numbers, are called *addressing modes* in ML. In general, that's because numbers are usually being sent to and from addresses in the computer's memory while an ML program is running. That, plus simple arithmetic, is the essence of what a computer does to accomplish any given task.

We'll get to the addressing modes (there are about ten) in a future column, but you can already recognize two of them: line 100's mode is called *immediate addressing* (the number is immediately after the instruction, not in some memory location elsewhere in the computer) and line 110's mode is called *implied addressing* (because the instruction contains its own target).

Putting The Assembler To Work

Enough theory, let's do something. Let's assemble a small program. If you've typed in Program 1, the first thing to do is to change line 10 so that the assembler will accept ordinary decimal numbers. It's designed to work with either decimal or hexadecimal, but we've not yet touched on hex so we'll stick with the familiar. Change the line to:

```
10 H=0
```

Then RUN the assembler and type in 830 when it asks you where you want to put the ML program. That's a safe place until you next load in a program from cassette. ML can be put into a variety of places in RAM. BASIC, of course, has a computer-determined starting location in memory, but *you* specify the start of an ML program. Now you'll see that address printed on screen. The addresses where the instructions are being stored will function as the "line numbers" for your reference when programming. Unlike BASIC, you can't go back up and change a line. If you make a mistake, start over. (There are easier ways to fix errors, but that, too, is for a future column.)

Now type LDY #0, hit RETURN, and you've written a line of ML which will put a zero into the Y register. (You'll see the numbers forming the ML version of your program appear to the right of the mnemonic/address you've typed.) Then the assembler will furnish you with the next available "line number" address in RAM, 832. The mnemonic/address pair LDY #0 uses up two bytes.

You are ready to type in your next pair: LDA #66. Hit RETURN on this line and you've put the code for the letter B into the A register. Then type in the rest of our ML program, one pair per line:

```
JSR 65490
DEY
BNE 834
RTS
```

That's it. To let the assembler know that you're through with your program, type END instead of a normal mnemonic and it will tell you the start and end addresses of your ML program. Then, having done its job, the assembler quits. The mnemonics and addresses were all POKEd into their proper places after being translated into the machine's language. To see what happens when this RUNs, you can type SYS 830 and see the effect of the small ML loop we wrote. You'll get 256 B's on screen in record time. Not something you've been anxious to do? More useful things are on their way.

In the coming months we'll go into detail about these mnemonics and instructions. We'll also make some modifications to the assembler itself. It's written in BASIC, after all, and there are some valuable things to learn about ML by simply adding to the functions of the assembler program. You'll likely find yourself using the assembler frequently as you get a better feel for ML programming.

If you have any questions or suggestions, write to me c/o COMPUTE!'s Gazette.

See program listing on page 200.

COMMODORE-64™ and VIC-20™ users!

Meet an...

IMP

DOT MATRIX PLAIN PAPER PRINTER

Impact Printer by Fidelity Electronics 4"×4.5"2"

IMPACT PRINTER COMES COMPLETE WITH ALL 20/64 INTERFACES CABLES

Short on Price and Size... Long on Quality!

Features: High quality print head- MCBF equal to 500,000 lines, 5 × 7, 30cps impact dot matrix, 144 dots per line, 2.25"w plain paper roll and cartridge ribbon included. Up to 40 column width, lower case descenders, upper/lower case characters, can reproduce the entire 20/64 graphics set, dot addressable graphics, supports custom character capability, reverse character mode, permits user selection of two character sets. Transformer included. 90-day warranty. N.Y. residents add applicable sales tax.

$129.95

Send Check or M.O. + $6.95 Shipping /Handling

All Major Credit Cards Accepted

DEALER INQUIRIES INVITED

"Call THE Printer Experts"

INSTITUTIONAL COMPUTER DEVELOPMENT CORP.

For Information and Orders

Toll-Free 1-800-645-4710

(in N.Y., outside cont. U.S. 516-221-3000)

2951 MERRICK RD. DEPT. G11 BELLMORE, NY 11710

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

Kyan's powerful, inexpensive software for powerful, inexpensive PC's.

It makes the most of your VIC-20 or Commodore 64.

Spreadsheet. Among the most advanced "What If" analyses for business or personal financial management: projections, budgets, estimates, etc.

☐ *Tabloid 20™* *$32.50*

Advanced spreadsheet cartridge for VIC-20. Includes manual. Store data and models on cassette and print on VIC printer. Supports add-on memory for very large models. No add-on memory or disk drive required. 254 rows by 63 columns.

☐ *Tabloid 64™* *$39.90*

Super spreadsheet on diskette for Commodore 64. Includes manual. Disk drive required, VIC printer recommended.

Pascal. Write programs in the modern structured language that beats BASIC.

☐ *Pascal 20* *$32.50*

Cartridge for VIC-20. Includes editor and manual. Add-on memory, cassette, and VIC printer supported, but not required.

☐ *Pascal 64* *$39.90*

Diskette for Commodore 64. Includes editor and manual. Disk drive required, VIC printer recommended.

Total amount enclosed. . . . . $_____

NYS residents add 7% sales tax. Dealer inquiries welcome.

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

Mail coupon with check or money order to:

KYAN SOFTWARE, Box 1157, Latham, NY 12110 (518) 371-5181

Kyan Software

SUPERCHARGE YOUR VIC-20* WITH COMPUSCOPE'S SUPERMOTHER

$89.95

- ★ **COPY CARTRIDGES** Includes instruction manual System software
- ★ **ADD UP TO 35K**
- ★ **PAUSE SWITCH** Freezes program
- **-8 SWITCH SELECTABLE SLOTS** Run single or multiple VIC-20 program cartridges; add memory; run utility program — all at the same time
- ★ **SYSTEM RESET BUTTON** Provides "non-destructive" system restart
- ★ **FULL BUFFERING** Assures accurate data transmission
- ★ **REPLACEABLE FUSE** With safety cap
- ★ **WRITE PROTECTION** Provides easy cartridge emulation

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

THE 64 Expander

Designed for reliable operation using an 80 column converter and word processor software. **$79.95**

FEATURES:

- ★ Four Expansion Slots - 3 vertical, 1 horizontal.
- ★ Pause Switch
- ★ Fully Buffered Address and Data Lines
- ★ Power Switching on 3 Vertical Slots
- ★ One Horizontal Slot - for Large Expansion Cards
- ★ System Reset Button
- ★ Fuse Protected
- ★ Allows Use of DMA Controllers
- ★ Will Support Parallel Disk Drive
- ★ Write Protection
- ★ Gold Connectors
- ★ External Power Jack (power adapter optional)

MasterCard, Visa, Money Order, Bank Check

Add $2.50 shipping and handling, COD (add $2.00). All orders shipped within 24 hours.

TO ORDER CALL TOLL FREE (800) 821-9211

Within Oregon (503) 842-4431

COMPUSCOPE, 6400 Signal Rd., Tillamook, OR 97141

HORIZONS: 64

CHARLES BRANNON
PROGRAM EDITOR

Last month, I promised we'd look at some of the latest hardware and software for the Commodore 64. This isn't meant as a review, but it will give you a first look at these products in less time than it takes to do a full-scale evaluation. You'll generally see the in-depth review in a later issue of either COMPUTE!'s Gazette or COMPUTE! Magazine.

Neutral Zone. This is a nicely done multicolor space game which finely scrolls left and right. There are some attempts at 3-D sprite animation, and the sound effects are OK. The game is written entirely in machine language, so joystick response is quick. Graphically, the game is quite beautiful, with multicolored space scenes, cruising starships, and a gigantic mothership.

One interesting feature is that *Neutral Zone* requires a "dongle" that plugs into joystick port one. Although you can copy the tape or disk, it will not run on a machine without the dongle. This is a good way to address the problem of software piracy. It lets you make backup copies, but protects the manufacturer from unethical copying. Let me know what you think of this, and other, approaches to software piracy. ($34.95 tape/disk, Access Software, 925 East 900 South, Salt Lake City, UT 84105.)

Screen Graphics 64. Tired of all those PEEKs and POKEs? *Screen Graphics 64* adds 24 commands to BASIC to make graphics programming a good bit easier. Most of the commands are obvious: HIRES, MULTI, TIC, DOT, DRAW, BOX, CIRCLE, CHAR, BLOCK, MODE, FILL, PIXEL. These control the high-resolution screen and draw axes, dots, lines, rectangles, circles, and solid rectangles; fill any shape, change dot modes (enabling you to reverse screen areas); add any text from any set to the screen; and return the dot stored at pixel position X,Y.

The coordinate drawing system is based with 0,0 at the lower-left corner of the screen — great for mathematicians, but it will make converting Apple and Atari programs a little more difficult.

There are also some commands to make sprite programming much easier. You can imbed sprite shapes in your programs, copy them, animate them, and remove sprites. Even multicolor sprites are easy to use.

There are many products on the market that add graphics commands to BASIC. Based on a similar VIC-20 product, *GRAPH-VICs,* this one is well implemented and easy to learn. ($24.95 tape/$27.95 disk, Abacus Software, P.O. Box 7211, Grand Rapids, MI 49510.)

Crossfire. This Apple translation by Sierra On-Line plays very well on the 64. The game is apparently done with high-resolution graphics. Even though most of the sprites go unused, the motion is smooth and fast. The sound and multipart music is especially well done. This game is so fast that if you aren't really sharp, your average game can be measured in seconds. The background music is almost hypnotic, but some may find it grating after a few hours of play.

In *Crossfire,* you move through the streets of a city, shooting aliens moving in all the horizontal and vertical corridors. The screen is filled with shooting, and it's hard not to blunder into an alien. The shapes of the creatures grow more complex as you play, and the difficulty skyrockets. Challenging and addicting. It's available at your local dealer for about $29.95 retail/disk (Sierra On-Line sells wholesale only).

The Zork Trilogy. If you can't enter sentences like this on your favorite word-based adventure game, it's time for *Zork:* "Open the brown bag, remove the bottle of water, open it and drink the water."

Zork puts the power of mainframe-based text adventures on your 64. It uses some of the most sophisticated techniques known. For example, to compact the verbose descriptions, words are stored using only five bits. This requires a special shift code to bank select different five-bit character sets, but it makes for very detailed prose. You control a "mind clone" through an underground kingdom on your way to fame and fortune. You battle trolls, cross gorges, find jewels, and solve puzzles. And watch out for the Grue!

Prices the Competition Can't Touch

Commodore 64

Avalon Hill Game Company

180-701 B-1 Nuclear Bomber (C)	$12
180-702 Midway Campaign (C)	$12
180-703 North Atlantic Convoy Raider (C)	$12
180-704 Nukewar (C)	$12
180-706 Planet Miners (C)	$12
180-712 Computer Stocks & Bonds (C)	$15
180-719 Andromeda Conquest (C)	$14
181-721 Computer Football Strategy (C)	$12
181-732 Telengard (C)	$16

Broderbund

David's Midnight Magic	$23
Choplifter (CT)	$34
Serpentine (CT)	$27
Sea Fox (CT)	$27

Datamost

Roundabout	$20
Bilestoad	$20
Mating Zone	$20

EPYX/Automated Simulations

14E-036 Jump Man (D)	$27

Human Engineered Software (HES)

HEE-307 6502 Professional Dev. System (C)	$23
HEE-400 Retro Ball (Crt)	$27
HEE-401 Hesmon (Crt)	$27
HEE-402 Turtle Graphics II (Crt)	$45
HEE-404 Heswriter 64 (Crt)	$35
HEE-412 Gridrunner (Crt)	$27

Infocom

63E-001 Zork (D)	$ 27
63E-002 Zork II (D)	$ 27
63E-003 Deadline (D)	$ 34
63E-004 Starcross (D)	$ 27
63E-005 Zork III (D)	$ 27

Sierra On-Line

54E-048 Frogger (D)	$23
Jaw Breaker	$20

Sirius Software

70E-036 Blade of Blackpoole (D)	$27
70E-037 Type Attack (Crt)	$27
70E-043 Repton (D)	$27
70E-046 Critical Mass (D)	$27
70E-424 Snake Byte (Crt)	$23
70E-445 Spider City (Crt)	$27
70E-447 Squish 'em (Crt)	$23
70E-448 Final Orbit (Crt)	$23

Spinnaker

SKE-001 Snooper Troops #1 (D)	$30
SKE-004 Facemaker (D)	$ 23
SKE-006 Kindercomp (D)	$ 20
SKE-008 Hey Diddle Diddle (D)	$ 20
SKE-009 In Search of the Most Amazing Thing	$ 27
Fraction Fever (CT)	$ 20
Alphabet Zoo (CT)	$ 20
Delta Drawing (CT)	$ 23

Snyapse Software

SSE-011 Ft. Apocalypse (D)	$ 23
SSE-016 Drelbs (D)	$ 23
SSE-019 Survivor (D)	$ 23
SSE-020 Pharoah's Curse (D)	$ 23
SSE-311 Ft. Apocalypse (C)	$ 23
SSE-316 Drelbs (C)	$ 23
SSE-319 Survivor (C)	$23
SSE-320 Pharoah's Curse (C)	$ 23

United Microwave Industries

92E-302 Renaissance (C)	$ 20
92E-331 Motor Mania (C)	$ 20

Commodore 64	*$219*
VIC 20	*$ 99*
1525 Printer	*$229*
1530 Datasette	*$ 64*
1541 Disk Drive	*$249*
1600 Modem	*$ 89*
1701 Commodore Monitor	*$289*
VIC 1311 Joystick	*$ 8*
VIC 1312 Game Paddles	*$ 16*

VIC 1210 3K Memory Expander	$34
VIC 1110 8K Memory Expander	$52
VIC 1111 16K Memory Expander	$89
VIC 1011 RS 232 Terminal Interface	$43
VIC 1211 Super Expander	$59
VIC 1213 Vicmon Machine Language Monitor	$45
VT 106A Recreation Pack	$45
VT 107A Home Calculation Pack	$45
VIC 1600 Vicmodem	$89
VM Programmer's Reference Guide	$14

Educational

Books

Kids and the Vic	$ 18
Programmer's Reference Guide (Vic)	$ 14
Programmer's Reference Guide (64)	$ 18

Language Arts

Super Hangman (C)	$ 14
Simon/Hess (C)	$ 13
Concentration (C)	$ 13
Home Babysitting	$ 23

Math

Sky Math (C)	$ 12
Space Division	$ 12
Bingo Speed Math (CT)	$ 23
Number Crunch (CT)	$ 23
Number Chaser	$ 17
Number Gulper	$ 17

Music

Vic Music Composer (CT)	$ 42
HES Synthesound (CT)	$ 49

Reading and Language Arts

Finger Spelling (D, C)	$ 12
My Spelling Easel (Ages 3-10)	$ 26

Programming Techniques

Intro to Basic Prog. I	$ 22
Intro to Basic Prog. II	$ 22
Programmer's Aid Cart	$ 22
Turtle Graphics/Hess (CT)	$ 29
Gorteck & the Microchip	$ 23

Social Studies/Science

Visible Solar System	$ 23
Reaganomics (CT)	$ 27

Commodore Software

Avenger	$ 23
Superslot	$ 23
Super Alien	$ 23
Jupiter Lander	$ 23
Draw Poker	$ 23
Midnight Drive	$ 23
Radar Rat Race	$ 23
Raid on Fort Knox	$ 23
Sargon II Chess	$ 29
Cosmic Cruncher	$ 23
Gorf	$ 29
Omega Race	$ 29
Sea Wolf	$ 23
Adventureland	$ 29
Pirate Cove	$ 29
Mission Impossible	$ 29
The Count	$ 29
Voodoo Castle	$ 29
The Sky is Falling	$ 23
Mole Attack	$ 23
Bingo Speed Math	$ 23
Home Babysitter	$ 23
Visible Solar System	$ 23
Personal Finance	$ 29
Quick Brown Fox	$ 65

Call Toll Free

1-800-634-6766

Order Line Only

COMPUTER OUTLET

1095 E. Twain (702) 796-0296
Las Vegas, Nevada 89109

Information & Order Inquiries (702) 369-5523

Mon. — Fri. 8 A.M. to 6 P.M. — Sat. 9 A.M. to 6 P.M.
Dealers' Inquiries Invited

Vic 20

Avalon Hill

Nuke War	$12

Automated Simulations

Rescue at Rigel (C)	$20
Ricochet (C)	$15
Monster Maze (CT)	$27
Sword of Fargoal	$27

Broderbund

Martian Raider	$15
Multisound Synthesizer	$15
Shark Trap	$15
Sky Blazer (CT)	$27
Sea Fox (CT)	$27
A.E. (CT)	$27

Creative Software

Black Hole (CT)	$36
Trashman (CT)	$36
Astroblitz (CT)	$36
City Bomber & Minefield (CT)	$20
Apple Panic (CT)	$36
Choplifter (CT)	$36
Serpentine (CT)	$36
Videomania (CT)	$36
Terraguard (CT)	$36

HES Software

HES Mon (CT)	$ 29
HES Writer (CT)	$ 29
Synthesound Music Synthesizer (CT)	$ 49
Turtle Graphics (CT)	$ 29
VIC Forth (CT)	$ 45
Victrek (C)	$ 15
Predator (CT)	$ 27

Sirius

Type Attack	$27
Snake Byte	$27

Thorn EMI

River Rescue (CT)	$ 29
Mutant Herd (CT)	$ 29

Tronix

Galactic Blitz (C)	$17
Swarm (C)	$20
Sidewinder (C)	$20
Gold Fever (CT)	$27
Deadly Skies (CT)	$27

United Microware

Spiders of Mars (CT)	$ 34
Meteor Run (CT)	$ 34
Amok	(C) $ 15 (CT) $ 27
Alien Blitz	(C) $ 17 (CT) $ 27
Skymath (C)	$ 12
Space Division (C)	$ 12
Super Hangman (C)	$ 14
The Alien (C)	$ 17
3D Maze (C)	$ 12
Kosmic Kamikaze (C)	$ 17
Sub Chase (C)	$ 15
Renaissance	$ 20
Cloud Burst (CT)	$ 27
Satellites & Meteorites (CT)	$ 34
Outworld (CT)	$ 34
Wordcraft	$ 65

We accept VISA and MasterCard

ORDERING INFORMATION AND TERMS: *For Fast Delivery send cashier checks, money orders or direct bank wire transfers. Personal and company checks allow 3 weeks to clear. C.O.D. orders ($3.00 minimum) and 1% of all orders over $300. School purchase orders welcomed. Prices reflect a cash discount only and are subject to change. Please enclose your phone number with any orders.* ***Shipping*** *— Software ($2.50 minimum). Shipping — Hardware (please call). Foreign orders, APO & FPO orders — $10 minimum and 15% of all orders over $100. Nevada residents add 5¾% sales tax. All goods are new and include factory warranty. Due to our low prices, all sales are final. All returns must have a return authorization number. Call 702-369-5523 to obtain one before returning goods for replacement.*

This is such a popular and well-done adventure that it has inspired two *Zork* sequels. There is a special *Zork* User Group, and paraphernalia such as cheat books, maps, bumper stickers, T-shirts, and buttons. *Zork* was originally written and sold by Infocom, Inc., a Cambridge, Massachusetts, company. Recently, Commodore itself has started marketing the *Zork* series. Infocom is earning a formidable reputation for its "interactive prose" programs. Other Infocom products for the 64 are *Starcross*, *Suspended* (a science-fantasy nightmare), *Deadline*, and *Witness* (both detective scenarios).

All of these are text-only adventures. Infocom says it "puts the graphics where the sun don't shine," in your brain, holding that the best graphics are in your imagination. This is arguable, but the fine detail of these games would be wasted on crude graphic displays. Some people like adventures with full-screen pseudo-realistic pictures, but the *Zork* trilogy and its kin really don't need them.

Telecommunicate!

How many of you out there own modems? These devices let your 64 send and receive data over the telephone. Boring? Hardly. With a modem, you can exchange programs with friends. You can dial up any of several Commodore-oriented bulletin boards. And then there's The Source and CompuServe. Having used them, I feel the under-$100 VICmodem is like adding the planet Earth to your computer. It significantly expands your personal sphere of communication. For the majority of people, one-way television and radio, plus an occasional phone call and letter, provide most of our long-distance needs. With a nationwide bulletin board, however, you're casually chatting with dozens of people from all over the country. On CompuServe Citizen's Band radio simulation, you can talk to people nationwide on such diverse subjects as computers, birth control, and peanut butter. I feel that this is substantially changing our "world concepts," when your next-door neighbor is 2000 miles away. (See the special telecomputing issue of COMPUTE!'s Gazette, September 1983.)

If there's sufficient interest, we'll talk about some of the latest developments in telecommunications, and show how you and your friends can make the most of your modems.

Rumors

Like rumors? There are some significant new products on the 64 horizon. For the memory-hungry, yes, there is memory expansion for the Commodore 64. One in particular is worth notice, the Monolith from Richvale Telecommunications. This not-yet-released cartridge is supposed to add 512K of RAM memory (1/2 megabyte, or 524,288 bytes)!

In addition, the Monolith has an on-board 68000 co-processor. This is the daddy microprocessor of the new high-speed, super-powerful 16-bit chips, and is used in Apple's Lisa. Expect big-system performance. The price? I hear it will be around $350.

The Unexploited SID

What are you doing with your 64's fantastic built-in sound synthesizer? Not many programmers seem to be exploiting its advanced features. There should be more experimentation with the filters, ring-modulation, and synchronization. These features are truly essential to really make the SID (Sound Interface Device) sing. Keep reading Gregg Peele's continuing series on 64 sound and music.

In any case, we're always interested in getting the utmost from your computer. If you've done something interesting, send it in!

Next month, we'll look at more products, including a software simulation of 80 columns, and various aids for BASIC and machine language programmers. Keep those cards and letters coming from over the Horizon .

MAILPRO
DATA ORGANIZER AND MAILING LIST PROGRAM FOR COMMODORE 64™

COMPARE THESE FEATURES:

- fast file definition
- easy updating
- rapid printing with total format and record selection control
- WORDPRO compatible
- up to 4000 records on 1541

MAILPRO 64.....$129.95

Also available for COMMODORE 8032 . . . $179.95

Payment in U.S. funds with order or major credit cards. customs brokerage extra

PRO-LINE SOFTWARE
PRO-LINE SOFTWARE LTD.
(416) 273-6350
755 THE QUEENSWAY EAST, UNIT 8
MISSISSAUGA, ONTARIO CANADA, L4Y 4C5

COMMODORE 64 $199.95 *

COMMODORE 64 only	$199.95 *
* with the purchase of	
one 1541 Disk Drive at	$299.95
one 1525E Printer at	$279.95
ALL FOR	$779.85

COMMODORE 64 only	$199.95 *
* with the purchase of	
one 1541 Disk Drive at	$299.95
one 14" color monitor 1701	$299.95
ALL FOR	$799.85

COMMODORE 64 only	$199.95*
* with the purchase of	
one 1541 Disk Drive at	$299.95
one 1526 Printer(FAST!) at	$339.95
(includes ifc/cable dir. connect)	
ALL FOR	$839.85

COMMODORE'S SX-100 PORTABLE!

$995.00

- ★ Full 64K
- ★ Color Monitor built-in
- ★ Disk drive built-in
- ★ FREE software package
- ★ Completely compatible with the Commodore 64
- ★ LOTS OF SOFTWARE AVAILABLE!

CALL US NOW – PRICES MAY BE EVEN LOWER!

	COMMODORE 64	$259.95
1541	Disk Drive	279.95
1525E	Dot Matrix printer; inc. direct connect int/cable	269.95
1526	Dot Matrix printer; inc. direct connect int/cable	339.95
1701	14" Color Monitor, good quality	289.95
1530	Datasette program recorder	69.95
1600	Modem (for VIC-20 / Com. 64)	79.95
1650	NEW! Auto dial/Ans. modem	109.95

VIC-20 in stock-call for best price!
Software avail. for VIC-20 & Com. 64-Call us!

Accessories from DATA 20

Z-80 Card	For Commodore 64	$269.95
80 Column Card	For Commodore 64	169.95
40 Column Card	For VIC-20	99.95
40 Column Card	w/8K mem built-in for VIC-20	139.95
Video Pack 64K	For VIC-20	329.95

FREE WITH PURCHASE of each Z-80 card, 80 column, 40 column or Video Pack 64, you receive the following:
Word Master word processor software
Mailing List software
Telecommunications software

Due to the tremendous demand of these products, goods are subject to availability. Place your orders now!

Some of the goods shown not available at all stores; call first! products subject to availability; we are not responsible for typographical errors; prices subject to change without notice; this ad supercedes all previous ads; minimum shipg & handling $5.95; all orders subject to verification.

Olympic Sales Company
SERVING YOU SINCE 1947

Telex: 67 34 77 Toll-Free Phone Orders:
Toll-free (in CA) 800-252-2153 800-421-8045 (out of CA)
Order Desks open 6 days a week! 7:00 AM to 6:00 PM Mon-Sat
P.O. Box 74545 216 So. Oxford Ave. Los Angeles, CA 90004
Phone: (213) 739-1130 Cable: "OLYRAV" LSA

COMPUTER CASSETTES
100% Error-Free • Fully Guaranteed

LENGTH	12 PACK	24 PACK
C-05	79¢	69¢
C-10	89¢	79¢
C-20	99¢	89¢
Boxes	26¢	21¢

UPS $3.00 Pkg. $18.00 Case

C-10's 39¢
(Min. 500 Case Lot)
w/labels ADD 4¢,
w/boxes ADD 13¢

FOR ORDERS ONLY
1-800-528-6050
Extension 3005

MICRO-80™ INC.
2665-C Busby Road
Oak Harbor, WA 98277

For Commodore 64 ™

"FAMILY" FINANCES

Complete personal bookkeeping system that makes your budgeting, forecasting & reconciliation a breeze! Gives you instant access to past & future payments and incomes in levels of detail from yearly & monthly summaries, to master accounts, to separate entries.

APRIL 03, 1983 10:23:15

MASTER MENU

1. PRINT REPORTS MENU
2. INCOME MENU
3. EXPENSE MENU
4. BALANCE CHECKBOOK MENU
5. BUDGET / FORECAST MENU
6. AUDIT ACCOUNTS MENU
7. AUDIT DETAILS MENU
8. END RUN

WHAT # DO YOU WANT ?_

$35.95 "Family Finances" Program
On high-quality 5 1/4" floppy diskette & 50 page easy-to-understand manual.

$2.95 Manual only, (credited to purchase).

Postage & handling included.
Mail check or money order to:

SOPHWARE
19123 Hamlin Street
Suite 4, Dept P6
Reseda, CA 91335

Commodore 64 is a trademark of Commodore Electronics Limited

CP/M®
FOR YOUR COMMODORE® C-64

- Run All CPM® Formated Software
- Supports 5¼" or 8" Disk Drives
- Z-80, Disk and DMA Controller on Interface Card
- Plugs into the Commodore 64® Expansion Port
- Uses Soft Sectored IBM 5¼" and 8" Format Disk Drives

Interface Card with 8" Disk Drive	599.00
Interface Card with 5¼" Disk Drive	499.00
Interface Card without Disk Drive	349.00

® Commodore 64 is a registered trademark of Commodore Business Machines, Inc.
® CP/M is a registered trademark of Digital Research, Inc.

estes ENGINEERING, INC.

To Order
Phone (913) 827-0629
825 N. 5th Street
Salina, Kansas 67401

Now for the "64"

STCP — 300/1200 Baud

Standard Terminal Communications Package

'PFO' IOD OOA CP<D1>D2 BELL 12 30 00 10 14 36

Don't settle for non-standard Communications Protocol! Access Micro Net, Source, Bulletin Boards, Local Main frame, etc

- Complete Package — includes RS232 Interface Board and software (does not include modem)
- Communicates in Industry Standard ASCII
- Upload/Download to/from Disk
- Automatic File Translation
- Can be controlled from keyboard or user supplied basic or machine language program

Specify 3.0 or 4.0 ROMS or 8032 Commodore Computer 4040 or 8050 or PEDISK II Disk or CBM64 on 1541

Price: $129.95

ATARI AND PET EPROM PROGRAMMER

Programs 2716 and 2532 EPROMs. Includes hardware and software. PET = $75.00 – ATARI (includes sophisticated machine language monitor) = $119.95

Prowriter Printer. Excellent dot matrix print. Parallel = $489.00 Serial = $600.00. IEEE = $589.00

NEW

VIC RABBIT CARTRIDGE AND CBM 64 RABBIT CARTRIDGE

NEW FEATURE! DATA FILES!

64 RABBIT

High Speed Cassette Load and Save!

$39.95

(includes cartridge and manual)

for VIC

Don't waste your Life away waiting to LOAD and SAVE programs on Cassette Deck.

Load or Save 8K in approximately 30 seconds! Try it: your Un-Rabbitized VIC or 64 takes almost 3 minutes. It's not only fast but VERY RELIABLE.

Almost as fast as 1541 Disk Drive! Don't be foolish. Why buy the disk when you can get the Rabbit for much, much less!

Allows one to APPEND Basic Programs!

Easy to install — just plugs in Expansion Connector on rear of the VIC. Rabbit Works with or without Expansion Memory. Works with VIC or 64 Cassette Deck. 12 Commands provide other neat features. Fast Data Files – two data file modes. Also Available for 2001, 4001, and 8032.

EHS's CBM 64 Monitor Cartridge

$25.00

Every Commodore 64 Owner should have this Machine Language Monitor to explore the interesting and exciting world of Machine Level Code. If you can't think of a need for it now, someday you'll wish it was in your Software Library. Commands include Display Registers, Memory, Disassemble, Assemble, Transfer, etc. Over 20 Commands. Cartridge and manual - $ 25.00

More than just an Assembler/Editor!

Now for the "64"

MAE

for PET APPLE ATARI

~~$169.95~~

New Price ~~$99.95~~

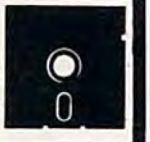
It's a Professionally Designed Software Development System

Blast off with the software used on the space shuttle project!

- Designed to improve Programmer Productivity
- Similar syntax and commands — No need to relearn peculiar syntaxes and commands when you go from PET to APPLE to ATARI
- Coresident Assembler/Editor — No need to load the Editor then the Assembler then the Editor, etc
- Also includes Word Processor, Relocating Loader, and much more
- Options: EPROM Programmer, unimplemented opcode circuitry
- STILL NOT CONVINCED. Send for free spec sheet!

Christmas special $59.95

5¼ INCH SOFT SECTORED DISKETTES



Highest quality. We use them on our PETs, APPLEs, ATARIs, and other computers. $22.50/10 or $44.50/20

EPROMS 2716 = $4.50 2532 = $7.50

Over 40 Commodore Programs by Baker (on 4040) = $25.00

Eastern House

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

VISA MasterCard

We are the only software and computer company you will ever need!

SOFTWARE GALLERY ® MAIL ORDER

P.O. BOX 3354, CHERRY HILL, N.J. 08034

TOLL FREE (800) 992-3300

FOR ORDER ONLY

FOR INFORMATION & N.J. CALL (609) 424-7106

Disk Drive 1541

For the Vic 20 and CMD64. Fast high capacity storage and retrieval of data.

$239

CMD64 Computer	$199
1525 Graphic Printer	$249
1530 Datassette	$59
1600 Modem	$59

1110	8K Memory Expander	$42
1111	16K Memory Expander	$68
1011	RS 232 Terminal Interface	$42
1211	Super Expander	$53
1212	Programmer's Aid Cartridge	$39
1213	Machine Language Monitor	$39
1311	Joystick	$7
1312	Game Paddles	$12

Color Monitor 1701

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

Lower Price **$249**

PRINTERS

EPSON

RX-80	CALL
MX-80 FT	CALL
FX-80 & FX-100	Call

SMITH CORONA

TP-1	$525

OKIDATA

82-A	$395	92	$495
83-A	$634	93	$809
84-P	$949		

STAR

GEMINI 10	$314
GEMINI 15	$449

C-ITOH

PROWRITER	$360
PROWRITER 2	$645
STARWRITER	$1199
PRINTMASTER	$1545
GORRILA	$209

CARDCO

Parallel Printer Interface	$57
16K Expander	$59
6 Slot Expander Interface	$72
3 Slot Expansion Interface	$31
Video Pak 80	$150

FLOPPY DISK

Elephant SS/DD(10)	$25
Elephant SS/SD(10	$18

Surge Protector

4 OUTLET	$37.95
6 OUTLET	$82.95

DATA 20

Z80 VIDEO & CPM	$249

MONITORS

AMDEK (300)	$154.95
COLOR I	$299.95
COLOR II	$589.95

WICO

The Boss	$15.95	Joystick	$22.50
Famous Red Ball			$25.95
Track Ball			$49.95

COMMODORE 64 TOP 20 SOFTWARE

Frogger	$25	Motor Mania	$22	Hes Mon	$28
Zork I	$27	Zork 3	$27	Heswriter	$34
Jumpman	$29	Grid Runner	$27	Facemaker	$26
Temple of Apshai	$29	Jawbreaker	$22	Wordpro 64	$77
Fort Apocaphypse	$26	Starcross	$28	Shamus	$26
Deadline	$35	Suspended	$35	Quick Brown Fox	$48
Zork 2	$27	Repton	$29		

VIC 20

Shamus	$26
Choplifter	$29
Gortek	$21
Grid Runner	$27
Gold Fever	$27
Spider City	$28
Side Winder	$22

COMPUTER COVERS $7.99 ea. Reg. $15.95

Features heavy duty canvas with vinyl interior, waterproof

CMD64	$7.99
Vic 20	$7.99
	$7.99
	$7.99
	$7.99

We Carry a Large Selection of Commodore Software, call for FREE catalog with order...or send $1 (refundable with order) UPS shipping 3% for hardware and $3 for software. N.J. resident add 6% sales tax. Note: Atari, Commodore, Epson, IBM, and Okidata are Registered trademarks

VISA MasterCard

PRICES SUBJECT TO CHANGE WITHOUT NOTICE.

NEWS & PRODUCTS

Editor/Assembler And World Processor for Commodore 64

Elcomp has released an editor/assembler and a word processor for the 64.

The editor/assembler, *Macrofire,* includes an editor with 24 commands and an assembler that can translate 10K of source code in about five seconds. The $89 program, which is available on tape or disk, has full macro capability and includes a function that allows you to assemble files larger than available memory.

Elcomp's word processor, *Blitztext,* is also available on tape or disk for $89. It includes horizontal and vertical scrolling, the ability to handle text files up to four disks long, left and right margin justification, and formatted output to any device. The program can be used with almost any printer.

Other products available from Elcomp include: *Tricks for VICs,* a book of ready-to-run programs and tutorials for the VIC-20, $9.95; *MORE On the 64,* a collection of machine language subroutines for the Commodore 64, $9.95; and *The Great Book of GAMES, Vol. 1,* an explanation of game programming on the Commodore 64, $9.95.

Elcomp Publishing, Inc.
53 Redrock Lane
Pomona, CA 91766
(714) 623-8314

Data Base, Terminal Program, And Adventure Game

Arfon Micro has released a handful of new software products for the VIC-20 and Commodore 64.

MicroBase is a data base and mailing label program for the VIC and 64. It allows up to 12 fields per record, 80 characters per field, and 176 characters per record. Sorting can be done by any field. The program is available for $29.95 on tape and $34.95 on disk.

Microterm 64 is a terminal communications program for the 64 that allows file transfer under ASCII and Commodore protocols. It includes the ability to download incoming files to disk, tape, or printer, and upload and transmit files from tape or disk. *Microterm 64* is available on tape or disk for $39.95 and $44.95, respectively.

Baldor's Castle is a realtime graphic adventure game for the VIC. The castle contains more than 70 rooms on three levels. Nine types of monsters patrol the castle. *Baldor's Castle* is available on cartridge for $49.95. A Commodore 64 version of the game is planned. It will feature more than 250 rooms on 10 levels and the option of designing your own castle.

Arfon Microelectronics, Inc.
111 Rena Drive, Suite C
Lafayette, LA 70503
(318) 988-2478

Computer Math For Kids

Integrity Software has introduced a line of programs designed to teach math concepts to young students.

Climb Time teaches addition and subtraction to preschool and kindergarten children through animated visual display. The program is available for the VIC-20 and Commodore 64 for $12.50.

Math Bash is an addition and subtraction drill game for first- and second-graders. The game, available for $12.50 for the unexpanded VIC, has four skill levels.

Math Bash II, designed for third- and fourth-graders, includes drills on multiplication and division. Two skill levels are available for each drill. The VIC program sells for $12.50.

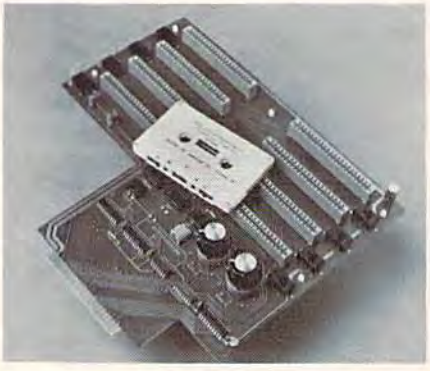
Chopper for the 64 includes addition, subtraction, multiplication, and division in one package. Correct answers to math questions free the choppers from behind a wall. *Chopper* includes 12 play levels, sound, and sprite graphics. The program, designed for first through fourth grades, sells for $22.50.

Integrity Software
Box 29
Bristol, VT 05443

8-Slot VIC Expander

Compuscope is producing Supermother, which adds 35K of memory to the VIC-20 and provides eight expansion slots.

Supermother includes a system RESET button, a pause



Compuscope's Supermother provides eight expansion slots for the VIC.

button that freezes a program or game in progress, and a replaceable fuse. The board includes a write-protection switch, which, when used with *Blocksave* software (included with the board), allows you to make backup copies of cartridge programs on tape or disk.

The board, which is compatible with most VIC expansion modules, sells for $129.95.

Compuscope
6400 Signal St.
Tillamook, OR 97141

VIC-20 Adventure

MicRo Information Systems has produced an adventure game called *Zorlok* for the VIC-20.

In the game, you become the great-great-grandson of Zorlok the Wizard. You must enter his castle, wipe out a plague of monsters, and regain his treasures.

The tape version of the program requires 8K expansion and sells for $39.95; the disk version requires 16K expansion and sells for $45.95.

MicRo Information Systems
P.O. Box 73
Wayne, NJ 07470
(201) 696-3296

Intelligent Multiple Interface

Interpod, an intelligent 6502-based multiple interface,

YORK 10™ CASSETTES DISKETTES

NEW!
LOWER PRICES
LONGER LENGTHS

DATA TRAC / C-06, C-12, C-24

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

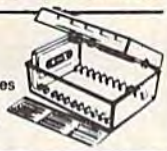
- Premium 5-screw shell with leader — BASF tape
- Internationally acclaimed. Thousands of repeat users.
- Error Free • Money back Guarantee

	C-06	C-12	C-24	HARD BOX
1 Dozen	7.00	7.50	9.00	2.50
2 Dozen	13.00	14.00	17.00	4.00

500 C-12's—38¢ each • w/ labels
Shipping $17.500 — add 4¢
500 Boxes 13¢ ea. • shipping $10.500

TRACTOR FEED • DIE-CUT BLANK CASSETTE LABELS
$3.00/100 $20.00/1000

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



BASF QUALIMETRIC FLEXI-DISC
5¼" SSDD, Soft Sect.
Lifetime warranty!
$26.95/10 $120.00/50 $215.00/100

MICRO CASSETTES
in convenient short lengths

	MC-10	MC-20	MC-30
1 Doz.	16.50	18.00	19.00
2 Doz.	32.50	34.50	36.00

Fits Epson and Sharp micro drives

Same superior tape in premium shell with leaders. Includes box.

SHIPPING/HANDLING $3.50
Any quantity (except 500 special)

NOTE: Outside 48 Contin. States shipping $3.50 PLUS $1 per caddy: per dozen cassettes: per dozen boxes: per 10 discs.

In Cont. U.S. shipments are by UPS unless Parcel Post requested.

California residents add Sales Tax

WRITE FOR FREE BROCHURE
ASK FOR QUANTITY DISCOUNTS
for IMMEDIATE SHIPMENT
on Credit Card Orders

VISA MasterCard

Call: 213/700-0330

YORK 10™ Computerware
9525 Vassar Ave. #G Chatsworth, CA 91311

HUNDREDS OF PROGRAMS AVAILABLE FOR THE COMMODORE 64 & VIC 20

Commodore 64

All Prices up to 30% OFF RETAIL

VIC 20

COMMODORE	
Commodore 64 Computer	249.95
1541 Disk Drive	269.95
1701 Color Monitor	269.00
1525 Printer	249.95
1530 Datasette	69.95
HOME	
Quick Brown Fox (CT)	47.50
Paper Clip (D)	96.25
WordPro 3 Plus (D)	67.95
CalcResult Advanced (D)	135.00
Hesmon (CT)	29.95
Home Accountant (D)	49.95
EDUCATIONAL	
Facemaker (D)	26.25
Kindercomp (D)	22.50
In Search of the Most Amazing Thing (D)	29.95
Early Games (D/CASS)	22.50
Electronic Party (VIC 20/CASS)	22.50
Square Pairs (VIC 20/CASS)	22.50
Turtle Trax (VIC 20/CASS)	22.50

GAMES	
Zork I, II, or III (D)	29.00
Deadline (D)	37.00
Starcross (D)	29.00
Jumpman (D/CASS)	29.95
Spritemaster (D/CASS)	26.95
Neutral Zone (D/CASS)	27.50
Temple of Apshai (D/CASS)	29.95
HARDWARE AND ACCESSORIES	
Cardco Parallel Printer Int.	63.95
The Connection Parallel Int.	89.95
Cardco 3 Slot Exp. Board	31.95
16K Memory Expander (VIC 20)	57.00
Data 20 8K Display Mgr (40/80 col. plus Word Proc.)	119.95
Data 20 Display Mgr (40/80)	79.95
Zenith 12" Green Monitor	109.00
Amdek Color I Monitor	319.95
C. Itoh Prowriter 8510AP	395.95
Control Data Diskettes (10)	23.00

Write or call for FREE CATALOG. TO ORDER: CALL 1-714-951-5596
8:00 A.M. - 8:00 P.M. PST Mon.-Sat. or send check or credit card number, signature and expiration date.

CENTURY MICRO PRODUCTS
P.O. Box 2520, Mission Viejo, CA 92690
Visa/Mastercard add 3%. Personal checks allow 2 weeks to clear. CA residents add sales tax. Shipping and handling and $3.00 (hardware extra). Prices subject to change.

CROWN COMPUTER and Software

Trainers Corner Shopping Center
Rts. 309 & 663, Quakertown, Pa. 18951

LARGE SELECTION — **FAST SERVICE**

In Pa. (215)538-1665
Out of Pa. (800)344-4499
SEND FOR FREE CATALOG

COMMODORE VIC-20		
Commodore		
Gorf c	39.95	31.95
Vic Avenger c	29.95	23.95
Creative		
Choplifter c	29.95	23.95
Serpentine c	29.95	23.95
Household Finance c	19.95	16.95
Car Cost c	19.95	13.95
Computer Software Associates		
PractiCalc 20 t	39.95	34.95
PractiCalc 20 d	44.95	36.95
PractiCalc Plus t	49.95	41.95
PractiCalc Plus d	54.95	46.95
Vic Pak #1 t	19.95	16.95
HES		
HES Writer c	39.95	31.95
Turtle Grapic c	39.95	31.95
Gridrunner c	29.95	24.95
6502 ProDevelopment c	29.95	23.96
Micro-Spec		
General Ledger d	89.95	69.95
Data Base Manager d	59.95	47.95
Quick Brown Fox		
Quick Brown Fox c	64.95	54.95

COMMODORE 64		
Commodore		
Disk Bonus Pack d	19.95	16.95
Computer Software Associates		
PractiCalc 64 t	49.95	41.95
PractiCalc 64 d	54.95	46.95
Inventory 64 d	24.95	21.95
Hes		
Retro Ball c	39.95	29.95
6502 ProDevelopment t	29.95	23.95
Coco c	49.95	39.95
Micro-Spec		
General ledger d	99.95	79.95
Data-Base Manager d	89.95	69.95
Professional Software		
Wordpro 3 Plus d	89.95	71.95
Spinnaker		
Face Maker d	34.95	26.95
Face Maker c	34.95	27.95
Snooper Troops I or II d	39.95	29.95
Synaspe		
Fort Apocalypse c	34.95	27.95
Fort Apocalypse d	34.95	27.95
Shamus c	34.95	27.95
Shamus d	34.95	27.95

TERMS: Prices subject to change. VISA and MASTERCARD add 3%. Personal Checks allow 2 weeks to clear. C.O.D. on Certified Check. PA. Residents add 6% Sales Tax.

The Interpod, an intelligent freestanding interface for the VIC and 64, uses a 6502 to communicate with all Commodore and RS-232 devices.

gives serial and parallel capabilities to the VIC and 64.

The interface lets the computer communicate with all Commodore serial IEEE and IEEE-488 devices, as well as RS-232 devices. It is a stand-alone unit that uses no computer memory.

Up to 30 IEEE devices and any RS-232 modem or printer can be accessed by the system when the Interpod is plugged into the serial port. Through the Interpod, the computer can communicate with voltmeters, plotters, and other peripherals.

The interface is available for $180.

Limbic Systems, Inc.
1056 Elwell Court
Palo Alto, CA 94303
(415) 964-8788

COMPUTE!'s Gazette *for Commodore 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.

Bug-Swatter:
Modifications And Corrections

• IMPORTANT: We added a POKE to the "Automatic Proofreader" (October 1983) to protect it from being destroyed when you LOAD another program from tape. The POKE does protect the Proofreader, and the Proofreader itself is not affected. However, a quirk in the VIC-20's operating system means that programs typed in with the Proofreader and SAVEd to tape cannot be LOADed properly later. If you LOAD a program SAVEd while you had the Proofreader in memory, you will get a ?LOAD ERROR message. This applies only to VIC-20 tape SAVEs (disk SAVEs work OK, and the quirk was fixed in the Commodore 64). The solution is to use this special LOAD procedure:

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

The program will LOAD just fine in the future. This month, the Proofreader has been updated to prevent this problem. The Proofreader also has been improved in other ways. We strongly recommend that you type in the new version of the Proofreader and discard the old one (the new version works on either the VIC or 64 and checks itself for typing errors). We apologize for any inconvenience this may have caused you.

• Many readers ran into recurring errors when attempting to run the "Single Drive File Copy" program that accompanied September's "64 Explorer" column. The errors involved numerous GOTOs and GOSUBs without corresponding target lines.

Here's what happened: just before the working version of the program was listed on our printer, 28 seemingly needless REM statements were deleted to save space. Usually, deleting a REM statement does not affect program operation. But in this case the REMs were being used to separate blocks of code to make the listing easier to read, and they were targets of the GOTOs and GOSUBs. The program will work if all GOTOs and GOSUBs without target lines are renumbered to branch to the next higher line number.

Since this solution would require lots of editing, and some references might be missed, we've decided to list all the deleted line numbers. The program can be fixed by adding these lines, and by changing the GOTO statement in line 1090 from GOTO 5000 to GOTO 4000.

(By the way, this is a good argument for adhering to the principle of never making a REM statement the target of a branch instruction. Many people, when entering listings, routinely omit REMs to save typing and memory.)

Add these lines: 10 REM, 30 REM, 100 REM, 120 REM, 200 REM, 220 REM, 300 REM, 320 REM, 400 REM, 420 REM, 500 REM, 520 REM, 600 REM, 620 REM, 700 REM, 720 REM, 800 REM, 820 REM, 900 REM, 920 REM, 1000 REM, 1020 REM, 2000 REM, 2020 REM, 3000 REM, 3020 REM, 4000 REM, 4020 REM.

• "TeleTerm 64" (September) works as published. But if you are having problems with uppercase letters appearing as lowercase or vice versa, try this modification: delete line 124 and reenter the statement as line 95.

• In the VIC-20 version of "The Viper" (August), omit the colon between THEN and PRINT in line 570.

• The author of "VIC/64 Mailing List" (August) suggests these changes to correct an infrequent bug in the Examine and Change options:

```
47 A$="":INPUT"{CLR}WHICH ITEM";A$:A=VAL(
   A$):IFA$=""ORA<1THEN19
48 READA$:IFA$="END"THEN19
49 IFA$<>"XX"THEN48
50 READA$:IFA<>VAL(A$)THEN48
```

If you are using "VIC/64 Mailing List" with adhesive mailing labels spaced one inch apart, the author suggests these changes for proper spacing:

```
95 N2$=LEFT$(B$,X):N1$=RIGHT$(B$,LEN(B$)-
   X)
101 READA:GOSUB92:GOSUB102:GOTO99
104 PRINT#1,CHR$(10)CHR$(10):RETURN
114 IFZ=4THEN120
116 PRINT#1,CHR$(10)CHR$(10)"ITEM";A:GOSU
   B92
120 GOSUB92:GOSUB102:CLOSE1:GOTO60
```

Also, in case the instructions for modifying the program for tape were unclear, here is the line to change:

```
55 PRINT"{CLR}":SAVE R$,1:END
```

We appreciate receiving both corrections and modifications from readers. Please address them to:

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

How To Type In *COMPUTE!'s Gazette* Programs

Many of the programs which are listed in *COMPUTE!'s Gazette* contain special control characters (cursor control, color keys, inverse video, etc.). To make it easy to know exactly what to type when entering one of these programs into your computer, we have established the following listing conventions.

Generally, any VIC-20 or Commodore 64 program listings will contain bracketed words which spell out any special characters: {DOWN} would mean to press the cursor down key. {5 SPACES} would mean to press the space bar five times.

To indicate that a key should be *shifted* (hold down the SHIFT key while pressing the other key), the key would be underlined in our listings. For example, S would mean to type the S key while holding the shift key. This would appear on your screen as a "heart" symbol. If you find an underlined key enclosed in braces (e.g., {10 N}), you should type the key as many times as indicated (in our example, you would enter ten shifted N's).

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

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

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

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

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

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

When You Read:	Press:	See:	When You Read:	Press:	See:	When You Read:	Press:	See:
{CLEAR}	SHIFT CLR/HOME		{CYN}	CTRL 4		{7}	C 7	
{HOME}	CLR/HOME		{PUR}	CTRL 5		{8}	C 8	
{UP}	SHIFT ↑ CRSR ↓		{GRN}	CTRL 6		{F1}	f1	
{DOWN}	↑ CRSR ↓		{BLU}	CTRL 7		{F2}	f2	
{LEFT}	SHIFT ← CRSR →		{YEL}	CTRL 8		{F3}	f3	
{RIGHT}	← CRSR →		{1}	C 1		{F4}	f4	
{RVS}	CTRL 9		{2}	C 2		{F5}	f5	
{OFF}	CTRL 0		{3}	C 3		{F6}	f6	
{BLK}	CTRL 1		{4}	C 4		{F7}	f7	
{WHT}	CTRL 2		{5}	C 5		{F8}	f8	
{RED}	CTRL 3		{6}	C 6				

A Beginner's Guide To Typing In Programs

What Is A Program?

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

BASIC Programs

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

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

Brackets And Special Characters

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

About DATA Statements

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

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

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

Get To Know Your Machine

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

A Quick Review

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

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

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

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

Chicken Little

(Article on page 64.)

BEFORE TYPING...

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

Program 1: VIC Version, Instructions And Character Set

```
0 IF PEEK(7344)=PEEK(7344+25600)THEN4
                                    :rem 23
1 POKE52,28:POKE56,28:CLR:FORT=7168TO7679
  :POKET,PEEK(T+25600):NEXT         :rem 62
4 PRINTCHR$(147)                    :rem 175
5 GOSUB10000:GOSUB6000              :rem 38
30 PRINT:PRINT"PLEASE WAIT FOR THE
   {3 SPACES}GAME TO LOAD."         :rem 189
40 POKE198,5:POKE631,78:POKE632,69:POKE63
   3,87:POKE634,13:POKE635,131:END
                                    :rem 226
6000 PRINTCHR$(147);SPC(4);CHR$(18);"CHIC
     KEN LITTLE";CHR$(146)          :rem 71
7000 PRINT:PRINT"YOU PLAY THE PART OF
     {2 SPACES}CHICKEN LITTLE."     :rem 48
7001 PRINT                          :rem 85
7010 PRINT"YOU CAN MOVE IN THE{3 SPACES}B
     OTTOM QUARTER OF THE SCREEN."
                                    :rem 116
7011 PRINT                          :rem 86
7020 PRINT"PIECES OF SKY (BLOWN{2 SPACES}
     RANDOMLY BY THE WIND){2 SPACES}FALL
     {SPACE}DOWN AT YOU."           :rem 221
7030 PRINT:PRINT"TO SCORE POINTS, TOSS RO
     CKS AT THE FALLING"            :rem 200
7031 PRINT"PIECES OF SKY WITH THE SPACEBA
     R."                            :rem 109
7040 PRINT:PRINT"YOU CAN ALSO HIT NON- FA
     LLING PIECES."                 :rem 66
7050 PRINT:PRINTCHR$(18);"HIT A KEY TO CO
     NTINUE";CHR$(146)              :rem 153
7055 GETA$:IFA$=""THEN7055          :rem 197
7056 PRINTCHR$(147)                 :rem 77
7060 PRINT"KEYS I,J,K AND M MOVE UP,LEFT,
     RIGHT AND DOWNRESPECTIVELY.":PRINT
                                    :rem 172
7080 PRINT"YOUR SCORE AND THE{4 SPACES}NU
     MBER OF REMAINING{3 SPACES}CHICKENS
     {SPACE}ARE SHOWN AT THE TOP.":rem 93
7090 PRINT:PRINT"YOU GET 50 POINTS FOR HI
     TTING FALLING SKY,"            :rem 165
7100 PRINT"AND 10 POINTS FOR{5 SPACES}OTH
     ERS."                          :rem 148
7690 PRINT:PRINTCHR$(18);"HIT A KEY TO CO
     NTINUE";CHR$(146)              :rem 163
7700 GETA$:IFA$=""THEN7700          :rem 191
7701 PRINTCHR$(147)                 :rem 74
7702 PRINT"EACH TIME YOU CLEAR{3 SPACES}T
     HE SCREEN AND ADVANCE TO THE NEXT LE
     VEL,"                          :rem 18
7703 PRINT"THE PIECES OF SKY FALL FASTER.
     "                              :rem 218
7704 PRINT:PRINTCHR$(18);           :rem 28
7705 PRINT"IF MORE THAN EIGHT{4 SPACES}PI
     ECES OF SKY HIT THE GROUND, THE GAME
      IS{3 SPACES}OVER."            :rem 9
7711 PRINT:PRINT:PRINT:PRINT:PRINT
                                    :rem 121
7720 FORT=1TO25:PRINTCHR$(145);SPC(4);CHR
     $(18);"GOOD LUCK!":FORP=1TO150:NEXTP
                                    :rem 171
7730 PRINTCHR$(145);SPC(4);"GOOD LUCK!":F
     ORP=1TO150:NEXTP:NEXTT         :rem 150
7740 PRINTCHR$(147)                 :rem 77
9180 RETURN                         :rem 178
10000 READA:IFA<0THENRETURN         :rem 211
10010 FORI=ATOA+7:READJ:POKEI,J:NEXT
                                    :rem 39
10020 GOTO 10000                    :rem 29
10031 DATA 7168,48,44,100,131,129,195,36,
      24                            :rem 73
10041 DATA 7176,12,18,233,33,241,14,9,16
                                    :rem 171
10050 DATA 7184,0,0,28,34,36,24,0,0
                                    :rem 164
10060 DATA 7200,36,227,1,0,0,195,65,82
                                    :rem 67
10070 DATA 7216,0,0,0,0,0,1,3,133   :rem 45
10080 DATA 7224,16,8,4,2,1,0,0,1    :rem 9
10090 DATA7248,121,1,1,2,252,68,130,1
                                    :rem 61
10100 DATA 7264,1,1,6,4,10,5,2,1    :rem 4
10110 DATA 7272,192,48,8,12,9,58,228,8
                                    :rem 92
10120 DATA 7296,41,42,60,72,164,135,68,56
                                    :rem 237
10130 DATA 7304,252,2,1,1,1,253,3,1
                                    :rem 153
19999 DATA -1                       :rem 141
```

Program 2: VIC Version, Main Program

```
0 QW=4:POKE7678,4:POKE7675,9        :rem 149
1 TY=3:POKE7677,0:DIMC(30)          :rem 249
2 POKE7679,3:TY$=":3"               :rem 150
3 POKE36869,255:PS=PEEK(7675):ZZ=30720
                                    :rem 202
4 PRINTCHR$(147)                    :rem 175
5 TY=PEEK(7679):TY$=CHR$(58)+CHR$(TY+48)
                                    :rem 58
6 QW=PEEK(7678)                     :rem 109
7 V=36878:S1=36874:S2=36875:S3=36876:S4=3
  6877                              :rem 2
11 POKE36879,8:X=11:Y=20            :rem 105
12 PRINTCHR$(147)                   :rem 222
15 DEFFNA(Z)=INT(RND(1)*Z)          :rem 24
16 FORT=1TO30:C(T)=FNA(352)+7701:POKEC(T)
   ,0:POKEC(T)+ZZ,6:NEXT            :rem 231
17 X=11:Y=20                        :rem 145
18 IFQW=1THENM=1:GOTO20             :rem 205
19 FORM=1TOQW                       :rem 87
20 A=((Y*22)+X)+7680                :rem 236
21 PRINTCHR$(159);CHR$(19);"SCORE:"SC
                                    :rem 163
22 PRINTCHR$(19);CHR$(156);SPC(12);"CHICK
   ENS:";CHR$(157);TY$              :rem 226
30 POKEA,1:POKEA+1,6:POKEA+22,7:POKEA+23,
   10:POKEA+ZZ,7:POKEA+ZZ+1,7:POKEA+ZZ+22
   ,7                               :rem 170
31 POKEA+ZZ+23,7                    :rem 166
40 IFPEEK(197)<>64THENGOSUB9000     :rem 102
50 IFF=1THENGOSUB9040               :rem 36
55 IFM<>1THEN65                     :rem 139
```

```
56 IFR1=1THENGOSUB2170                :rem 100
57 IFR=31THEN7000                     :rem 228
60 IFR1=0THENGOSUB2160                 :rem 93
65 IFQW=1THENGOTO18                   :rem 225
2000 NEXTM:GOTO18                      :rem 42
2160 R=R+1:IF R=31THEN7000            :rem 184
2162 R1=1:IFC(R)=0ORPEEK(C(R))=32THEN2160
                                      :rem 111
2165 Q=C(R):QQ=6                      :rem 145
2166 IFQ1<>0THENQ1=32:QQ=0             :rem 38
2170 POKEQ,Q1:POKEQ+ZZ,QQ:Q=Q+22+FNA(3)-1
     :IFQ=AORQ=A+1ORQ=A+22ORQ=A+23THENGOT
     O3100                            :rem 120
2171 Q1=PEEK(Q)                        :rem 81
2180 POKEQ,0:POKEQ+ZZ,6               :rem 162
2190 IFQ>8185THENR1=0:POKEQ,32:POKEQ+ZZ,0
     :C(R)=0:DE=DE+1                  :rem 145
2195 IFQ=BTHENPOKEQ,4:POKEQ+ZZ,2       :rem 54
2196 IFDE=PSTHEN9200                  :rem 196
2200 IFQ=BTHENFORT=170TO255:POKEV,15:POKE
     S4,T:POKES2,T:POKES1,T:POKES3,T:NEXT
     :POKEQ,32                        :rem 112
2201 IFQ=BTHENPOKEQ+ZZ,0               :rem 14
2202 POKEV,0:POKES4,0:POKES1,0:POKES3,0:P
     OKES2,0                          :rem 209
2210 IFQ=BTHENSC=SC+50:C(R)=0:R1=0:F=0
                                      :rem 240
2222 RETURN                           :rem 168
3100 POKEA,12:POKEA+1,13:POKEA+22,16:POKE
     A+23,17:POKEA+ZZ,7:POKEA+ZZ+1,7:POKE
     A+ZZ+22,7                        :rem 162
3101 POKEA+ZZ+23,7                     :rem 7
3102 POKEV,15:POKES2,158:FORT=1TO600:NEXT
     :POKES2,0:FORT=1TO60:NEXT        :rem 217
3103 POKEV,15:POKES2,158:FORT=1TO300:NEXT
     :POKES2,0:FORT=1TO60:NEXT        :rem 215
3104 POKEV,15:POKES2,181:FORT=1TO1050:NEX
     T:POKES2,0:FORT=1TO60:NEXT        :rem 7
3105 POKEV,0:POKES1,0:POKES2,0         :rem 61
3200 TY=TY-1:IFTY=0THEN9200           :rem 151
3201 TY$=CHR$(58)+CHR$(TY+48)          :rem 84
3203 R1=0                             :rem 184
3210 IFTY<>0THENF=0:POKEB,32:POKEA,32:POK
     EA+1,32:POKEA+22,32:POKEA+23,32
                                      :rem 49
3220 IFTY<>0THENPOKEB+ZZ,0:POKEA+ZZ,0:POK
     EA+1+ZZ,0:POKEA+22+ZZ,0:POKEA+23+ZZ,
     0:GOTO17                         :rem 114
7000 PRINTCHR$(147):PRINT:PRINT:PRINT:PRI
     NT:PRINT:PRINT:PRINT             :rem 179
7001 POKE36869,240:FORT=1TO6:POKEV,15:POK
     ES2,239                          :rem 142
7002 PRINTCHR$(5);CHR$(145);SPC(4);CHR$(1
     8);                              :rem 219
7003 PRINT"NEXT LEVEL":FORP=1TO150:NEXT:P
     OKES2,0                          :rem 45
7005 PRINTCHR$(145);SPC(4);"NEXT LEVEL":F
     ORP=1TO150:NEXTP:NEXTT           :rem 207
7006 POKE36869,255:PRINTCHR$(5)        :rem 34
7007 IFQW=1THENQW=2                   :rem 185
7010 R=0:POKE7679,TY:QW=QW-1:POKE7678,QW
                                      :rem 228
7011 IFPS=1THEN7020                    :rem 97
7012 PS=PS-1                          :rem 171
7020 R=0:DE=0:GOTO3                    :rem 94
9000 J=PEEK(197):POKEA,32:POKEA+1,32:POKE
     A+22,32:POKEA+23,32              :rem 206
9001 POKEA+ZZ,0:POKEA+ZZ+1,0:POKEA+ZZ+22,
     0:POKEA+23+ZZ,0                  :rem 159
9003 IFF=1THEN9010                     :rem 8
9005 IFJ=32THENF=1:B=A-22:FORT=1TO20:POKE
     V,15:POKES4,180:NEXT:POKES4,0
                                      :rem 106
9010 IFJ=44THENX=X+1                  :rem 192
9011 IFJ=20THENX=X-1                  :rem 189
9015 IFX>21THENX=20                   :rem 125
9016 IFX<0THENX=0                      :rem 23
9020 IFJ=36THENY=Y+1                  :rem 196
9021 IFJ=12THENY=Y-1                  :rem 193
9025 IFY>21THENY=21                   :rem 129
9026 IFY<17THENY=17                   :rem 138
9030 RETURN                           :rem 172
9040 POKEB,2:POKEB+ZZ,4               :rem 134
9050 POKEB,32:POKEB+ZZ,0:B=B-22        :rem 66
9070 IFB<7680THENF=0                  :rem 148
9075 IFB=QTHEN2195                     :rem 52
9080 IFPEEK(B)<>0THEN9090             :rem 195
9085 POKEB,4:FORT=225TO170STEP-1:POKEV,15
     :POKES4,T:POKES2,T:NEXT:SC=SC+10
                                      :rem 158
9090 POKES2,0:POKES4,0:POKEV,0         :rem 73
9159 IFPEEK(B)<>32THENPOKEB,32:POKEB+ZZ,0
     :F=0:GOTO9166                    :rem 75
9165 POKEB,2:POKEB+ZZ,4               :rem 142
9166 REM                              :rem 186
9180 RETURN                           :rem 178
9200 POKE36869,240:PRINTCHR$(147);CHR$(5)
     ;"THE SKY HAS FALLEN."           :rem 86
9210 POKE198,0                        :rem 249
```

Program 3: 64 Version

```
100 REM CHICKEN                       :rem 106
110 :                                 :rem 204
120 PRINT"{CLR}":POKE53280,0:POKE53281,0:
    FORI=0TO27:POKE54272+I,0:NEXT :rem 31
130 POKE54277,16:POKE54278,242:POKE54296,
    15                                :rem 155
136 POKE54284,16:POKE54285,242:POKE54291,
    16:POKE54285,242                  :rem 206
140 POKE214,11:PRINT:PRINTSPC(13)"{6 SPACE}CH
    ICKEN LITTLE"                     :rem 178
150 POKE56,48:CLR:FORI=0TO62:READA:POKE49
    152+I,A:NEXT:SYS49152             :rem 98
160 GOSUB1410:GOSUB1120               :rem 91
170 POKE905,9:POKE907,0:POKE908,4:POKE909
    ,3                                 :rem 3
180 TY=3:QW=4:DIMC(30)                :rem 20
190 POKE53272,28:PS=PEEK(905):ZZ=54272
                                      :rem 191
200 PRINT"{CLR}"                      :rem 246
210 TY=PEEK(909)                      :rem 149
220 QW=PEEK(908)                      :rem 144
230 S1=54272:S2=54279:S4=54286:XL=53248:X
    H=53264:YL=53249                  :rem 199
240 POKE53280,0:POKE53281,0           :rem 235
250 DEFFNA(Z)=INT(RND(1)*Z)            :rem 73
260 FORT=1TO30:C(T)=FNA(640)+1063:POKEC(T
    ),0:POKEC(T)+ZZ,6:NEXT            :rem 19
270 X=20:Y=20:POKE53269,1:POKE2040,11:POK
    E53287,?7                         :rem 35
280 :                                 :rem 212
290 REM MAIN LOOP                     :rem 222
300 IFQW=1THENM=1:GOTO330              :rem 43
310 FORM=1TOQW                        :rem 129
330 PRINT"{HOME}{CYN}SCORE:"SC;TAB(27)"
    {PUR}CHICKENS:"TY                 :rem 64
335 IFPEEK(56320)<>127THENGOSUB870 :rem 2
350 XP=X*8+24:POKEXL,XPAND255:POKEXH,XP/2
    56:POKEYL,Y*8+50                  :rem 70
355 A=INT(X)+INT(Y)*40+1024           :rem 239
```

```
390 IFF=1THENGOSUB980                    :rem 47
400 IFM<>1THEN450                        :rem 227
410 IFR1=1THENGOSUB520                   :rem 91
420 IFR=31THEN750                        :rem 227
430 IFR1=0THENGOSUB480                   :rem 97
450 IFQW=1THEN300
460 NEXT:GOTO300                         :rem 223
470 :                                    :rem 213
480 R=R+1:IFR=31THEN750                  :rem 96
490 R1=1:IFC(R)=0ORPEEK(C(R))=32THEN480
                                         :rem 20
500 Q=C(R):QQ=6                          :rem 88
510 IFQ1<>0THENQ1=32:QQ=0                :rem 237
520 POKEQ,Q1:POKEQ+ZZ,QQ:Q=Q+40+FNA(3)-1:
    IFQ=AORQ=A+1ORQ=A+40ORQ=A+41THEN640
                                         :rem 226
530 Q1=PEEK(Q)                           :rem 30
540 POKEQ,0:POKEQ+ZZ,6                   :rem 112
550 IFQ>1864THENR1=0:POKEQ,32:C(R)=0:DE=D
    E+1                                  :rem 103
560 IFQ=BTHENPOKEQ,4:POKEQ+ZZ,2
570 IFDE=PSTHEN1090                      :rem 141
580 IFQ<>BTHEN610                        :rem 255
585 POKES4+4,129:POKES1+4,17:POKES2+4,33:
    FORT=8TO90                           :rem 52
586 POKES4+1,T:POKES2+1,T*2:POKES1+1,T:NE
    XT:POKEQ,32                          :rem 232
590 POKEQ+ZZ,0                           :rem 89
600 SC=SC+50:C(R)=0:R1=0:F=0             :rem 51
610 POKES4+4,128:POKES1+4,16:POKES2+4,32
                                         :rem 48
620 RETURN                               :rem 120
630 :                                    :rem 211
640 FORI=0TO15:POKE53287,(I+7)AND15:FORT=
    1TO100:NEXT:NEXT                     :rem 28
660 POKES2+4,17:POKES2+1,21:FORT=1TO600:N
    EXT:POKES2+4,16:FORT=1TO60:NEXT
                                         :rem 246
670 POKES2+4,17:POKES2+1,21:FORT=1TO300:N
    EXT:POKES2+4,16:FORT=1TO60:NEXT
                                         :rem 244
680 POKES2+4,17:POKES2+1,25:FORT=1TO1050:
    NEXT:POKES2+4,16:FORT=1TO60:NEXT
                                         :rem 44
700 TY=TY-1:IFTY=0THEN1090               :rem 104
710 R1=0                                 :rem 136
725 F=0:POKEB,32                         :rem 141
740 GOTO270                              :rem 109
744 :                                    :rem 217
750 PRINT"{CLR}{7 DOWN}"                 :rem 119
760 POKE53272,21:FORT=1TO6:POKES2+1,31:PO
    KES2+4,17                            :rem 213
770 PRINTSPC(15)"{WHT}{UP}{RVS}NEXT LEVEL
    ":FORP=1TO150:NEXT:POKES2+4,16
                                         :rem 220
780 PRINTSPC(15)"{UP}NEXT LEVEL":FORP=1TO
    150:NEXT:NEXT                        :rem 95
790 PRINT"{CLR}":POKE53272,28             :rem 6
800 IFQW=1THENQW=2                       :rem 131
810 R=0:POKE909,TY:QW=QW-1:POKE908,QW
                                         :rem 63
820 IFPS=1THEN840                         :rem 5
830 PS=PS-1                              :rem 124
840 R=0:DE=0:GOTO190                     :rem 152
850 :                                    :rem 215
860 REM MOVE PLAYER                      :rem 134
870 J=PEEK(56320)                        :rem 156
890 IFF=1THEN910                         :rem 173
900 IF(JAND16)THEN910                    :rem 198
905 F=1:B=A-40:POKES4+4,129:POKES4+1,20:F
    ORT=1TO20:NEXT:POKES4+4,128          :rem 69
910 IF(JAND8)=0THENX=X+1:IFX>39THENX=39
                                         :rem 175
920 IF(JAND4)=0THENX=X-1:IFX<0THENX=0
                                         :rem 52
930 IF(JAND2)=0THENY=Y+1:IFY>21THENY=21
                                         :rem 157
940 IF(JAND1)=0THENY=Y-1:IFY<17THENY=17
                                         :rem 167
950 RETURN                               :rem 126
960 :                                    :rem 217
970 REM MOVE ROCK                        :rem 234
980 POKEB,2:POKEB+ZZ,4                    :rem 90
990 POKEB,32:B=B-40                       :rem 48
1000 IFB<1024THENF=0                     :rem 119
1010 IFB=QTHEN560                        :rem 235
1020 IFPEEK(B)THEN1040                   :rem 254
1030 POKEB,4:POKES2+4,17:POKES4+4,129
                                         :rem 141
1035 POKEB,4:FORT=33TO10STEP-1:POKES4+1,T
     :POKES2+1,T:NEXT:SC=SC+10           :rem 142
1040 POKES2+4,16:POKES4+4,128            :rem 129
1050 IFPEEK(B)<>32THENPOKEB,32:F=0:RETURN
                                         :rem 36
1060 POKEB,2:POKEB+ZZ,4                  :rem 128
1070 RETURN                              :rem 168
1080 :                                    :rem 3
1090 PRINT"{CLR}":POKE53272,21:PRINT"
     {WHT}THE SKY HAS FALLEN."           :rem 205
1095 PRINT"{DOWN}YOUR SCORE WAS:"SC
                                         :rem 55
1100 POKE53269,0:END                     :rem 103
1110 :                                   :rem 253
1120 PRINT"{CLR}{3 DOWN}"SPC(13)"{6}CHI
     CKEN LITTLE"                        :rem 150
1130 PRINT"{GRN}{DOWN} YOU ARE CHICKEN LI
     TTLE.{2 SPACES}YOU MOVE IN"         :rem 86
1140 PRINT"{DOWN} THE BOTTOM PART OF THE
     {SPACE}SCREEN WHILE"                :rem 68
1150 PRINT"{DOWN} PIECES OF SKY (BLOWN RA
     NDOMLY BY THE"                      :rem 122
1160 PRINT"{DOWN} WIND) FALL TO THE GROUN
     D.{2 SPACES}TO SCORE"               :rem 196
1170 PRINT"{DOWN} POINTS, TOSS ROCKS AT T
     HE FALLING"                         :rem 242
1180 PRINT"{DOWN} PIECES OF SKY WITH THE
     {SPACE}FIRE BUTTON."                :rem 62
1190 PRINT"{2 DOWN} YOU CAN ALSO HIT NON-
     FALLING PIECES."                    :rem 157
1200 PRINTSPC(9)"{6}{2 DOWN}PRESS FIRE
     {SPACE}TO CONTINUE":GOSUB1320
                                         :rem 250
1210 :                                   :rem 254
1220 PRINT"{CLR}{GRN}{DOWN} USE A JOYSTIC
     K IN PORT TWO TO MOVE":PRINT"{DOWN}
     {SPACE}ABOUT THE SCREEN."           :rem 12
1230 PRINT"{DOWN} YOU GET 50 POINTS FOR H
     ITTING FALLING"                     :rem 194
1240 PRINT"{DOWN} SKY, AND 10 POINTS FOR
     {SPACE}OTHERS."                     :rem 199
1250 PRINT"{DOWN} EACH TIME YOU CLEAR THE
      SCREEN AND"                        :rem 194
1260 PRINT"{DOWN} ADVANCE TO THE NEXT LEV
     EL THE PIECES":PRINT"{DOWN} OF SKY F
     ALL FASTER."                        :rem 44
1270 PRINT"{DOWN} IF MORE THAN EIGHT PIEC
     ES OF SKY HIT"                      :rem 52
1280 PRINT"{DOWN} THE GROUND, THE GAME IS
      OVER."                             :rem 138
1290 PRINTSPC(15)"{2 DOWN}{6}GOOD LUCK!
     ":PRINTSPC(10)"{DOWN}PRESS FIRE TO B
     EGIN":GOSUB1320                     :rem 93
```

```
1300 PRINT"{CLR}":RETURN                :rem 66
1310 :                                  :rem 255
1320 IF(PEEK(56320)AND16)=0THEN1320
                                        :rem 184
1330 IFPEEK(56320)AND16THEN1320         :rem 251
1340 RETURN                             :rem 168
1350 :                                    :rem 3
1360 DATA 173,14,220,41,254,141,14,220,16
     5,1,41,251,133,1,169,208           :rem 103
1370 DATA 133,252,169,48,133,254,169,0,13
     3,251,169,0,133,253,160,0          :rem 174
1380 DATA 162,8,177,251,145,253,200,208,2
     49,230,254,230,252,202,208,242
                                        :rem 165
1390 DATA 165,1,9,4,133,1,173,14,220,9,1,
     141,14,220,96                       :rem 77
1400 :                                  :rem 255
1410 READA:IFA<0THENFORI=0TO62:READA:POKE
     704+I,A:NEXT:RETURN                :rem 127
1420 FORI=12288+A*8TO12288+A*8+7:READJ:PO
     KEI,J:NEXT                          :rem 32
1430 GOTO1410                           :rem 199
1440 DATA 0,48,44,100,131,129,195,36,24
                                        :rem 119
1450 DATA 1,12,18,233,33,241,14,9,16
                                        :rem 219
1460 DATA 2,0,0,28,34,36,24,0,0         :rem 215
1470 DATA 4,36,227,1,0,0,195,65,82
                                        :rem 131
1480 DATA 6,0,0,0,0,0,1,3,133           :rem 104
1490 DATA 7,16,8,4,2,1,0,0,1             :rem 70
1500 DATA 10,121,1,1,2,252,68,130,1
                                        :rem 149
1510 DATA 12,1,1,6,4,10,5,2,1           :rem 105
1520 DATA 13,192,48,8,12,9,58,228,8
                                        :rem 195
1530 DATA 16,41,42,60,72,164,135,68,56
                                         :rem 81
1540 DATA 17,252,2,1,1,1,253,3,1,-1
                                        :rem 146
1541 DATA 0,0,0,30,0,0,107,0,0,209:rem 87
1542 DATA 128,0,97,128,0,25,128,2,12
                                        :rem 227
1543 DATA 255,141,12,0,113,12,31,1,12
                                        :rem 252
1544 DATA 97,131,12,131,6,12,4,12,6,0
                                          :rem 7
1545 DATA 56,3,3,224,1,254,0,1,152
                                        :rem 119
1546 DATA 0,1,152,0,1,152,0,1,152       :rem 53
1547 DATA 0,1,152,0,3,108,0              :rem 24
1550 END                                :rem 162
```

Martian Prisoner

(Article on page 68.)

BEFORE TYPING...

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

Program 1: VIC/64 Martian Prisoner

```
10 GOSUB5000                            :rem 166
100 GOSUB1000:IFR=6ANDU=0THENR=1:GOTO100
                                        :rem 232
110 GOSUB2000:PRINT                     :rem 155
120 ONVGOSUB3000,3000,3000,3000,3100,3200
    ,3300,3400,3500,3600,3700,3800,3900
                                         :rem 42
125 IFV=14THEN3900                       :rem 26
130 GOTO100                              :rem 94
1000 PRINT:ONRGOSUB1100,1200,1300,1400,15
     00,1600,1700,1800                  :rem 87
1005 IFI(4)=-1ANDI(6)=-1THENPRINTNS$
                                        :rem 125
1010 PRINT:PRINT"{CYN}OBJECTS:{WHT}";
                                        :rem 125
1020 FORL=1TO8:IFI(L)=RTHENPRINTTAB(8);N$
     (L)                                :rem 227
1030 NEXT:PRINT:RETURN                  :rem 228
1100 PRINT"YOU ARE IN A PRISON{3 SPACES}C
     ELL."                              :rem 105
1110 IFRND(1)>.25THENRETURN             :rem 154
1120 G=1:PRINT"A GUARD HAS TURNED OFFTHE
     {SPACE}FORCE FIELD AND{3 SPACES}ENTE
     RED THE CELL."                     :rem 126
1130 C%(1,1)=2:RETURN                   :rem 149
1200 PRINT"YOU ARE IN A N/S HALL.":RETURN
                                        :rem 122
1300 PRINT"YOU ARE IN THE ENGINE ROOM.":R
     ETURN                               :rem 29
1400 PRINT"YOU ARE IN A SMALL{4 SPACES}RO
     OM. A LARGE SIGN IS ON THE WALL."
                                        :rem 124
1410 IFI(4)=-1THENPRINTNS$               :rem 63
1420 IFI(6)=-1THEN4500                  :rem 185
1430 RETURN                             :rem 168
1500 PRINT"YOU ARE IN THE SUPPLY ROOM.":R
     ETURN                               :rem 86
1600 PRINT"YOU ARE IN THE NORTH{2 SPACES}
     SIDE OF THE HALL."                  :rem 90
1610 IFU=1THENPRINT"THE GUARDS DON'T
     {6 SPACES}NOTICE YOU."              :rem 10
1620 IFU=0THENPRINT"THE GUARDS TAKE YOU
     {3 SPACES}BACK TO THE CELL.":G=0
                                        :rem 180
1630 RETURN                             :rem 170
1700 PRINT"YOU ARE IN A LARGE{4 SPACES}RO
     OM.":RETURN                         :rem 54
1800 PRINT"YOU ARE IN A STRANGE{2 SPACES}
     GARDEN WHERE FOOD IS{2 SPACES}GROWN
     {SPACE}FOR THE CREW."              :rem 255
1810 IFI(4)=-1THENPRINTNS$               :rem 67
1820 RETURN                             :rem 171
2000 C$="":N=0:V=0:PRINT:INPUT"COMMAND
     {GRN}";C$:PRINT"{WHT}":IFC$=""THEN20
     00                                 :rem 226
2015 P=0:IFLEN(C$)<2THEN2050             :rem 73
2020 FORL=2TOLEN(C$)-1                  :rem 254
2030 IFMID$(C$,L,1)=" "THENP=L          :rem 104
2040 NEXT                                 :rem 5
2050 IFP=0THENV$=C$:N$=""               :rem 141
2060 IFP>0ANDP=LEN(C$)THENV$=C$:N$=""
                                        :rem 134
2070 IFP>0ANDP<LEN(C$)THENV$=LEFT$(C$,P-1
     ):N$=RIGHT$(C$,LEN(C$)-P)          :rem 86
2080 FORL=1TO14:IFLEFT$(V$,3)=V$(L)THENV=
     L                                  :rem 23
2100 NEXT:FORL=1TO8:IFLEFT$(N$,3)=A$(L)TH
     ENN=L                              :rem 55
2120 NEXT:IFN>0ANDV>0THENRETURN          :rem 47
2130 IFN=0ANDV>0ANDN$=""THENRETURN
                                        :rem 124
2135 IFN=0ANDV=5THENRETURN              :rem 191
2140 PRINT:PRINT"I DON'T UNDERSTAND.":GOT
     O2000                              :rem 95
```

```
3000 N$=V$:GOTO3110                          :rem 36
3100 N$=LEFT$(N$,1)                          :rem 226
3110 IFR=1ANDN$="E"ANDG=0THENPRINT"THE FO
     RCE FIELD STOPS YOU.":RETURN            :rem 56
3120 IFR<>1ORN$<>"E"ORG=0THEN3130:rem 179
3125 PRINT"AS YOU LEAVE THE CELL THE FORC
     E FIELD IS{4 SPACES}ACTIVATED, TRAPP
     ING"                                    :rem 17
3127 PRINT"THE GUARD."                       :rem 32
3130 IFR=2ANDN$="E"ANDC%(2,1)=0ANDI(8)>-1
     THENPRINT"THE LOCKED DOOR STOPS YOU.
     "                                       :rem 118
3135 IFR=2ANDN$="E"ANDC%(2,1)=0ANDI(8)>-1
     THENRETURN                              :rem 255
3140 IFR<>2ORN$<>"E"ORC%(2,1)>0THEN3150
                                             :rem 186
3145 PRINT"YOU UNLOCK THE DOOR{3 SPACES}W
     ITH THE KEY.":C%(2,1)=5:N$(7)="OPEN
     {SPACE}DOOR"                            :rem 191
3150 IFN$="N"THEND=0                         :rem 121
3160 IFN$="E"THEND=1                         :rem 114
3165 IFN$="S"THEND=2                         :rem 134
3170 IFN$="W"THEND=3                         :rem 135
3175 IFC%(R,D)=0THENPRINTCN$:RETURN
                                             :rem 210
3180 PRINT"OK":R=C%(R,D):RETURN              :rem 149
3200 IFN=1ORN=2ORN=3ORN=7THENPRINT"YOU CA
     N'T LIFT IT!":RETURN                    :rem 129
3203 IFI(N)<>RTHENPRINT"IT'S NOT HERE.":R
     ETURN                                   :rem 127
3205 IFN=5THEN3720                           :rem 20
3210 PRINT"OK":I(N)=-1:RETURN                :rem 8
3300 PRINT"OK":I(N)=R:RETURN                 :rem 252
3400 PRINT"YOU ARE CARRYING:"                :rem 6
3410 FORL=1TO8:IFI(L)=-1THENPRINTTAB(3);N
     $(L)                                    :rem 239
3420 NEXT:RETURN                             :rem 34
3500 IFN<>3ORR<>4THENPRINTCN$:RETURN
                                             :rem 126
3510 PRINT"ATOMIC FUEL NEARBY.{3 SPACES}D
     ON'T BRING ANY RADIO-ACTIVE":rem 128
3520 PRINT"MATERIALS INTO THIS{3 SPACES}R
     OOM.":RETURN                            :rem 52
3600 IFN<>7ORR<>2ORI(8)>-1THENPRINTCN$:RE
     TURN                                    :rem 144
3610 N$="E":GOTO3145                         :rem 66
3700 IFI(5)<>RTHENPRINTCN$:RETURN:rem 127
3720 PRINT"YOU ARE NOW WEARING A UNIFORM.
     ":I(5)=-1:U=1:RETURN                    :rem 189
3800 IFN<>6THENPRINTRI$:RETURN               :rem 237
3810 PRINT"YOU QUICKLY BECOME{4 SPACES}SI
     CK AND DIE.":GOTO4600                   :rem 161
3900 IFN=2THENPRINT"THE GUARD SHOOTS YOU.
     ":GOTO4600                              :rem 180
3910 PRINTCN$:RETURN                         :rem 41
4500 PRINT                                   :rem 86
4505 PRINT"THE RADIOACTIVE PLANT EMITS EN
     OUGH NEUTRONS TO START A"               :rem 98
4510 PRINT"CHAIN REACTION. THE{3 SPACES}S
     HIP EXPLODES."                          :rem 40
4515 PRINT"YOU ESCAPE IN A LIFE- CRAFT."
                                             :rem 17
4520 PRINT:PRINT"{PUR}YOU WIN!":GOTO4610
                                             :rem 73
4600 PRINT:PRINT"{PUR}YOU LOSE!"             :rem 79
4610 PRINT:PRINT:PRINT"{GRN}PLAY AGAIN?"
                                             :rem 29
4620 GETK$:IFK$="Y"THENRUN                   :rem 81
4630 IFK$="N"THENEND                         :rem 160
4640 GOTO4620                                :rem 211
5000 PRINT"{HOME}{CLR}":POKE36879,8:PRINT
     TAB(3)"{GRN}{RVS}MARTIAN PRISONER
     {OFF}":PRINT                            :rem 135
5080 DIMV$(14),C%(8,3),I(8),N$(8),A$(8)
                                             :rem 146
5090 R=1:FORL=1TO14:READV$(L):NEXT:rem 87
5100 FORL=1TO8:READC%(L,0),C%(L,1),C%(L,2
     ),C%(L,3):NEXT                          :rem 31
5110 FORL=1TO8:READN$(L),A$(L),I(L):NEXT
                                             :rem 97
5115 CN$="YOU CAN'T":RI$="DON'T BE SILLY!
     "                                       :rem 87
5120 NS$="GEIGER COUNTER IS{5 SPACES}CLIC
     KING.":RETURN                           :rem 9
6000 DATAN,E,S,W,GO,GET,DRO,INV,REA,OPE,W
     EA,EAT,KIL,HIT                          :rem 217
6010 DATA0,0,0,0,6,0,3,0,2,4,0,0,0,0,0,3,
     0,0,0,2,7,0,2,0,0,8,6,0,0,0,0,7
                                             :rem 103
6020 DATAFORCE FIELD,FOR,1,GUARDS,GUA,6,S
     IGN,SIG,4,GEIGER COUNTER,GEI,5,UNIFO
     RM                                      :rem 251
6030 DATA UNI,5,PLANT,PLA,8,LOCKED DOOR,D
     OO,2,MAGNETIC KEY,KEY,3                 :rem 151
```

Program 2: 64 Formatter

```
100 PRINT "{CLR}{4 SPACES}{RVS}22-COLUMN
    {SPACE}PRINT FORMATTER FOR 64":PRINT
                                             :rem 191
110 PRINT "READING DATA"                     :rem 119
120 FORI=828TO881:READA:CK=CK+A:POKEI,A:N
    EXT:POKE179,883AND255                    :rem 92
130 IF CK<>6032 THEN PRINT "ERROR IN DATA
    :CHECK TYPING.":END                      :rem 227
140 PRINT"{DOWN}BEFORE...":SYS 828:PRINT"
    AFTER..."                                :rem 150
150 PRINT "{DOWN}PRESS RUN/STOP-RESTORE";
    :PRINT"TO REGAIN 40 COLUMNS"             :rem 228
160 PRINT "{DOWN}ENTER {RVS}SYS 828{OFF}
    {SPACE}TO":PRINT"REACTIVATE, IF":PRIN
    T"NECESSARY."                            :rem 115
170 PRINT "{DOWN}DO NOT EDIT ANY":PRINT"L
    INES WHILE IN 22 COL-UMN MODE."
                                             :rem 84
1000 DATA169,71,141,38,3,169,3,141
                                             :rem 180
1010 DATA39,3,96,72,152,72,138,72:rem 141
1020 DATA56,32,240,255,192,9,176,3
                                             :rem 185
1030 DATA76,100,3,192,31,144,15,169
                                             :rem 226
1040 DATA13,32,202,241,56,32,240,255
                                             :rem 9
1050 DATA160,9,24,32,240,255,104,170
                                             :rem 14
1060 DATA104,168,104,76,202,241              :rem 30
```

Munchmath

(Article on page 76).

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: Munchmath, VIC Version

```
10 PRINT"{CLR}":POKE36878,15:S=36874:SM=7
   702:CM=38422:L=1:BC=27          :rem 240
15 POKE52,29:POKE56,29:S$="{RVS}{HOME}
   {30 DOWN}":CB=36879             :rem 96
20 J$="{22 I}":P=3:GOTO305        :rem 66
25 FORT=1TO300:NEXTT:RETURN        :rem 45
30 FORT=1TO30:NEXTT:RETURN        :rem 249
35 FORT=1TO70:NEXTT:RETURN          :rem 2
40 D=VAL(AN$):IFASC(AN$)=81THEN280
                                   :rem 223
45 IFD=CTHENP=P+1:R=R+1:M=M+1:SC=SC+10:PO
   KES+2,220:FORT=1TO5:NEXTT:POKES+2,0:GO
   TO240                           :rem 22
50 M=M+1:W=W+1:PRINTLEFT$(S$,11)SPC(11-LE
   N(C$))"{RED}"C:POKES,200:GOSUB25:POKES
   ,0:GOTO250                      :rem 66
55 FORI=1TOLEN(A$):PRINTMID$(A$,I,1);:POK
   ES,250:GOSUB30:POKES,0:NEXTI:A$=""
                                   :rem 94
60 RETURN                          :rem 70
65 PR=PR+1:A=INT(RND(1)*(5*L))+1   :rem 36
70 B=INT(RND(1)*(5*L))+1:IFB>ATHENA=A+B
                                   :rem 239
75 E=A*B:A$=STR$(A):B$=STR$(B)     :rem 179
80 IFQ=1THENC=A+B:X=43:GOTO100     :rem 77
85 IFQ=2THENC=A-B:X=45:GOTO100     :rem 87
90 IFQ=3THENC=A:GOTO120           :rem 175
95 C=E:X=88                       :rem 114
100 C$=STR$(C):PRINTLEFT$(S$,8)SPC(11-LEN
    (A$))A                         :rem 196
105 PRINTLEFT$(S$,9)SPC(9-LEN(B$))" ";CHR
    $(X)B"{DOWN}{3 LEFT}FFF"       :rem 187
110 PRINTLEFT$(S$,11)SPC(10-LEN(C$)):INPU
    TAN$:IFAN$=""THEN110           :rem 91
115 D=VAL(AN$):GOTO40             :rem 214
120 PRINTLEFT$(S$,13)SPC(7)"{9 SPACES}"
                                   :rem 121
125 PRINTLEFT$(S$,11)SPC(9)"{8 SPACES}"
                                   :rem 126
130 PRINTLEFT$(S$,12)SPC(9)"{N}{4 T}
    {DOWN}{5 LEFT}{N}"            :rem 125
135 PRINTLEFT$(S$,13)SPC(9-LEN(B$))B;E
                                   :rem 194
140 PRINTLEFT$(S$,11)SPC(10-LEN(C$)):INPU
    T"{RVS}";AN$:IFAN$=""THEN140  :rem 242
145 GOTO40                         :rem 55
150 FORI=130TO254:POKES+2,I:GOSUB30:NEXTI
    :POKES+2,0                     :rem 183
155 POKECM+P,2:GOSUB30:POKECM+P,5:GOSUB30
                                   :rem 66
160 POKES+3,220:POKESM+P,69:FORI=1TO290:N
    EXTI                           :rem 46
165 POKESM+P,64:FORI=1TO250:NEXTI  :rem 80
170 POKESM+P,32:POKES+3,0:SC=SC-50:IFSC<0
    THENSC=0                       :rem 27
175 L=L-1:IFL=0THENL=1             :rem 59
180 P=3:M=0:PRINT"{CLR}":GOTO445  :rem 251
185 POKESM+(M-1),32:FORI=1TO6:POKECM+M,3:
    POKESM+M,60:POKECM+P,5:POKES+2,200
                                   :rem 101
190 POKESM+P,62:GOSUB30:POKESM+P,58:POKEC
    M+M,6:POKESM+M,61:GOSUB30      :rem 182
195 POKESM+M,32:POKESM+P,32:POKES+2,0:P=P
    -1:M=M-1:NEXTI                 :rem 167
200 FORI=12TO9STEP-1:POKECM+I,5:POKESM+I,
    62:POKECM+9,6:POKESM+9,60:GOSUB30
                                   :rem 146
205 POKES+2,200:POKECM+9,3:POKESM+I,58
                                   :rem 6
210 GOSUB30:POKES+2,0:POKESM+I,32:NEXTI
                                   :rem 187
215 FORI=1TO5:PRINT"{HOME}{RVS}{DOWN}
    {RED}"TAB(7)"**100**":POKES+2,220:GOS
    UB25:POKES+2,0                 :rem 182
220 PRINT"{HOME}{DOWN}{RVS}"TAB(6)"
    {8 SPACES}":GOSUB25:NEXTI:L=L+1
                                   :rem 139
225 SC=SC+100:P=3:M=0:BC=BC+1:IFBC>31THEN
    BC=27                          :rem 211
230 POKECB,BC:PRINT"{CLR}{BLU}":PRINTLEFT
    $(S$,4)J$                      :rem 99
235 POKE36869,255:PRINT"{HOME}{DOWN}{PUR}
    {4 SPACES}????????????????{RVS}{RED}S"
                                   :rem 223
240 POKESM+(P-1),32:POKECM+P,5:POKESM+P,5
    9:GOSUB25:POKESM+P,58          :rem 81
245 IFSM+P=SM+18THEN185           :rem 131
250 POKESM+(M-1),32:POKECM+M,2:POKESM+M,6
    1:GOSUB25:POKECM+M,6:POKESM+M,60
                                     :rem 8
255 IFPEEK(SM+M)=PEEK(SM+P)THEN150:rem 76
260 PRINTLEFT$(S$,17)SPC(7)"{CYN}LEVEL:"L
    "{BLU}"                        :rem 130
265 PRINTLEFT$(S$,18)J$            :rem 68
270 PRINTLEFT$(S$,20)" "N$"'S SCORE:"SC
                                   :rem 139
275 PRINTLEFT$(S$,8)SPC(8)"{4 SPACES}
    {DOWN}{4 LEFT}{4 SPACES}{2 DOWN}
    {5 LEFT}{8 SPACES}":GOTO65     :rem 239
280 POKE36869,240:POKECB,254:PRINT"{CLR}
    {RVS}{DOWN}{2 RIGHT}"N$"' SCOREBOARD
    {OFF}","{2 DOWN}{RIGHT}PROBLEMS:"PR-1
                                   :rem 235
285 PRINT"{2 DOWN}{RIGHT}{GRN}RIGHT ANSWE
    RS:"R,"{2 DOWN}{RIGHT}{RED}WRONG ANSW
    ERS:"W,"{2 DOWN}{RIGHT}{BLK}GRADE:"IN
    T(100/(PR-1))*R"%"             :rem 20
290 INPUT"{2 DOWN}{RIGHT}PLAY AGAIN (Y/N)
    ";A$                           :rem 11
295 IFA$="Y"THENRUN                :rem 146
300 PRINT"{CLR}":FORI=1TO8:PRINT"{DOWN}
    {2 RIGHT}{RED}G O O D B Y E ! !":GOSU
    B25:NEXTI:END                  :rem 91
305 FORF=7632TO7679:READA:POKEF,A:NEXTF
                                   :rem 198
310 DATA24,60,110,126,126,126,60,24
                                   :rem 215
315 DATA56,124,95,248,224,248,127,56
                                    :rem 48
320 DATA60,126,255,219,255,255,169,169
                                   :rem 146
325 DATA60,126,255,219,255,255,90,180
                                    :rem 89
330 DATA120,116,30,14,30,124,120,0
                                   :rem 148
335 DATA0,0,0,14,14,14,0,0          :rem 8
340 FORK=7424TO7431:READA:POKEK,A:NEXTK
                                   :rem 197
345 DATA,,,,,,,0                    :rem 26
350 POKECB,250:POKE36869,255:PRINTLEFT$(S
    $,11)"{2 RIGHT}{BLU}M {GRN}U{RED} N
    {BLU}C {BLK}H {GRN}M {RED}A {BLU}T
    {CYN}H"                        :rem 223
355 READF:IFF=-1THEN375            :rem 119
360 POKES+2,F:GOSUB30:POKES+2,0:GOSUB30:G
    OTO355                         :rem 169
365 DATA195,209,219,225,225,225,225
                                   :rem 250
370 DATA219,219,219,209,219,209,195,-1
                                   :rem 139
```

```
375 FORI=ØTO15:PRINTMID$(S$,2,1Ø)SPC(I)"
    {SPACE}{CYN}= {RED}<{2 SPACES}{GRN}:"
    ;:GOSUB35                          :rem 106
380 PRINTMID$(S$,2,1Ø)SPC(I)" {BLU}<
    {PUR}={2 SPACES}{GRN};":POKES+2,2ØØ:G
    OSUB35:POKES+2,Ø:NEXTI             :rem 90
385 PRINTMID$(S$,2,1Ø)SPC(15)"{8 SPACES}"
                                       :rem 191
390 POKECB,254:POKE36869,24Ø:PRINT"{CLR}
    {3 DOWN}{BLU}":A$="WHAT IS YOUR NAME"
    :GOSUB55                           :rem 198
395 PRINTLEFT$(S$,9):PRINT"{GRN}":INPUTN$
    :PRINT"{BLU}"                      :rem 57
4ØØ PRINT"{CLR}"LEFT$(S$,6):A$="WHAT WOUL
    D YOU":GOSUB55                     :rem 113
405 PRINTLEFT$(S$,8):A$="LIKE TO PRACTICE
    ,":GOSUB55:PRINTLEFT$(S$,1Ø):A$=N$+":
    "                                  :rem 134
41Ø GOSUB55:PRINTLEFT$(S$,12):A$="{RED}1)
    {GRN}ADDITION":GOSUB55             :rem 16
415 PRINTLEFT$(S$,14):A$="{RED}2){GRN}SUB
    TRACTION":GOSUB55                  :rem 246
42Ø PRINTLEFT$(S$,16):A$="{RED}3){GRN}DIV
    ISION":GOSUB55                     :rem 12
425 PRINTLEFT$(S$,18):A$="{RED}4){GRN}MUL
    TIPICATION{BLU}":GOSUB55           :rem 176
43Ø GETQ:IFQ<1ORQ>4THEN43Ø             :rem 121
435 PRINT"{CLR}{2 DOWN}{RIGHT}LEVEL (1-9)
    ?"                                 :rem 222
44Ø GETL:IFL<1ORL>9THEN44Ø             :rem 113
445 GOTO23Ø                            :rem 1Ø7
```

Program 2: Munchmath, 64 Version

```
1ØØ POKE56,48:CLR:PRINT"{CLR}":SM=1Ø73:CM
    =55345:L=1:BC=3                    :rem 142
11Ø FORI=ØTO27:POKE54272+I,Ø:NEXT:POKE542
    96,15:POKE54277,18:POKE54278,165
                                       :rem 56
12Ø S$="{HOME}{21 DOWN}":SF=54272:WV=5427
    6                                  :rem 67
13Ø J$="99999999999999999999":P=3:GOTO9
    ØØ                                 :rem 111
14Ø :                                  :rem 207
15Ø FORT=1TO3ØØ:NEXT:RETURN             :rem 8
16Ø FORT=1TO4Ø:NEXT:RETURN             :rem 218
17Ø FORT=1TO9Ø:NEXT:RETURN             :rem 224
18Ø :                                  :rem 211
19Ø D=VAL(AN$):IFASC(AN$)=81ANDPR>1THEN8Ø
    Ø                                  :rem 247
2ØØ IFINT(D)<>INT(C)ORAN$>"9"THEN23Ø
                                       :rem 1Ø9
21Ø P=P+1:R=R+1:M=M+1:SC=SC+1Ø:POKESF,223
    :POKESF+1,29:POKEWV,17             :rem 67
22Ø FORT=1TO5:NEXT:POKEWV,16:GOTO7ØØ
                                       :rem 65
23Ø M=M+1:W=W+1:PRINTLEFT$(S$,1Ø)SPC(2Ø-L
    EN(C$))"{RVS}{RED}"C"{5 SPACES}"
                                       :rem 1Ø
24Ø POKESF+1,8:POKESF,1ØØ:POKEWV,33:GOSUB
    15Ø:POKEWV,32:GOTO72Ø              :rem 136
25Ø :                                  :rem 2Ø9
26Ø POKESF+1,4Ø:POKEWV,17              :rem 246
27Ø GOSUB16Ø:POKEWV,16:GOSUB16Ø:RETURN
                                       :rem 196
28Ø :                                  :rem 212
29Ø PR=PR+1:A=INT(RND(1)*5*L)+1          :rem 3
3ØØ B=INT(RND(1)*5*L)+1:IFB>ATHENA=A+B
                                       :rem 2Ø2
31Ø E=INT(A*B):A$=STR$(A):B$=STR$(B)
                                       :rem 23
32Ø IFQ=1THENC=A+B:X=43:GOTO36Ø        :rem 13Ø
33Ø IFQ=2THENC=A-B:X=45:GOTO36Ø        :rem 136
34Ø IFQ=3THENC=A:GOTO41Ø               :rem 223
35Ø C=E:X=88                           :rem 156
36Ø C$=STR$(C):PRINTLEFT$(S$,7)SPC(2Ø-LEN
    (A$))"{RVS}"A                      :rem 33
37Ø PRINTLEFT$(S$,8)SPC(18-LEN(B$))"{RVS}
     "CHR$(X)B"{OFF}{DOWN}{3 LEFT}999"
                                       :rem 176
38Ø PRINTLEFT$(S$,1Ø)SPC(19-LEN(C$)):GOSU
    B123Ø:IFAN$=""THEN38Ø              :rem 12Ø
39Ø D=VAL(AN$):GOTO19Ø                  :rem 17
4ØØ :                                  :rem 2Ø6
41Ø PRINTLEFT$(S$,12)SPC(16)"{9 SPACES}"
                                       :rem 17Ø
42Ø PRINTLEFT$(S$,1Ø)SPC(18)"{8 SPACES}"
                                       :rem 171
43Ø PRINTLEFT$(S$,11)SPC(19)"7777":PRINTS
    PC(18)"8"                          :rem 1Ø9
44Ø PRINTLEFT$(S$,12)SPC(18-LEN(B$))"
    {RVS}"B;E                          :rem 7Ø
45Ø PRINTLEFT$(S$,1Ø)SPC(19-LEN(C$)):GOSU
    B123Ø:IFAN$=""THEN45Ø              :rem 116
46Ø GOTO19Ø                            :rem 1Ø9
47Ø :                                  :rem 213
48Ø POKEWV,17:FORI=4TO33:POKESF+1,I:GOSUB
    16Ø:NEXT:POKEWV,16                 :rem 54
49Ø POKECM+P,2:GOSUB16Ø:POKECM+P,5:GOSUB1
    6Ø                                 :rem 172
5ØØ POKESF+1,14:POKEWV,33:POKESM+P,69:FOR
    I=1TO23Ø:NEXT                      :rem 154
51Ø POKESM+P,64:FORI=1TO25Ø:NEXT       :rem 1
52Ø POKESM+P,32:POKEWV,32:SC=SC-5Ø:IFSC<Ø
    THENSC=Ø                           :rem 75
53Ø L=L-1:IFL=ØTHENL=1                 :rem 54
54Ø P=3:M=Ø:PRINT"{CLR}":GOTO121Ø      :rem 34
55Ø :                                  :rem 212
56Ø POKESM+M-1,32:FORI=1TO6:POKECM+M,3:PO
    KESM+M,6Ø:POKECM+P,5               :rem 58
57Ø ?POKESF+1,14:POKEWV,129             :rem 47
58Ø POKESM+P,62:GOSUB16Ø:POKESM+P,58:POKE
    CM+M,6:POKESM+M,61:GOSUB16Ø        :rem 33
59Ø POKESM+M,32:POKESM+P,32:POKEWV,128:P=
    P-1:M=M-1:NEXT                     :rem 197
6ØØ FORI=12TO9STEP-1:POKECM+I,5:POKESM+I,
    62:POKECM+9,6:POKESM+9,6Ø:GOSUB16Ø
                                       :rem 2Ø2
61Ø POKESF+1,14:POKEWV,129:POKECM+9,3:POK
    ESM+I,58                           :rem 252
62Ø GOSUB16Ø:POKEWV,128:POKESM+I,32:NEXT
                                       :rem 19
63Ø FORI=1TO5:PRINT"{HOME}{RVS}{DOWN}
    {RED}"TAB(15)"** 1ØØ **":POKESF+1,15:
    POKEWV,33                          :rem 15
64Ø GOSUB15Ø:POKEWV,32                 :rem 87
65Ø PRINT"{HOME}{DOWN}{RVS}"TAB(15)"
    {9 SPACES}":GOSUB15Ø:NEXT:L=L+1
                                       :rem 168
66Ø SC=SC+1ØØ:P=3:M=Ø:BC=BC+1:IFBC>31THEN
    BC=7                               :rem 164
67Ø REM SETUP                           :rem 18
68Ø POKE53280,BC:POKE53281,1:PRINT"{CLR}
    {BLU}":PRINTLEFT$(S$,3)SPC(9)J$
                                       :rem 33
69Ø POKE53272,28:PRINT"{HOME}"SPC(13)"
    {DOWN}{PUR}????????????{RVS}{RED}S"
                                       :rem 132
7ØØ POKESM+P-1,32:POKECM+P,5:POKESM+P,59:
    GOSUB15Ø:POKESM+P,58               :rem 48
71Ø IFSM+P=SM+18THEN56Ø                :rem 125
72Ø POKESM+M-1,32:POKECM+M,2:POKESM+M,61:
```

```
    GOSUB150:POKECM+M,6:POKESM+M,60
                                    :rem 232
730 IFPEEK(SM+M)=PEEK(SM+P)THEN480:rem 80
740 PRINTLEFT$(S$,16)SPC(16)"{RVS}{CYN}LE
    VEL:"L"{BLU}"                   :rem 198
750 PRINTLEFT$(S$,17)SPC(9)J$       :rem 178
760 PRINTLEFT$(S$,19)"{RVS}"SPC(12)N$"'S
    {SPACE}SCORE:"SC                :rem 67
770 PRINTLEFT$(S$,7)SPC(17)"{4 SPACES}":P
    RINTSPC(17)"{4 SPACES}":PRINTSPC(13)"
    {DOWN}{8 SPACES}"               :rem 233
780 GOTO290                         :rem 115
790 :                               :rem 218
800 POKE53272,21:POKE53280,6:POKE53281,7
                                    :rem 245
810 PRINT"{CLR}{DOWN}{RVS}"SPC(13-LEN(N$)
    /2)N$"'S SCOREBOARD"            :rem 255
820 PRINTSPC(14)"{2 DOWN}PROBLEMS:"PR-1
                                    :rem 199
830 PRINTSPC(12)"{2 DOWN}{GRN}RIGHT ANSWE
    RS:"R:PRINTSPC(12)"{2 DOWN}{RED}WRONG
     ANSWERS:"W                     :rem 151
840 PRINTSPC(14)"{2 DOWN}{BLK}GRADE:"INT(
    R/(PR-1)*100)"%"                :rem 138
850 PRINTSPC(12)"{2 DOWN}PLAY AGAIN (Y/N)
    ?":POKE198,0                    :rem 141
860 GETA$:IFA$<>"Y"ANDA$<>"N"THEN860
                                    :rem 57
870 IFA$="Y"THENPR=0:R=0:W=0:SC=0:GOTO110
    0                               :rem 20
880 END                             :rem 119
890 :                               :rem 219
900 FORF=55TO63:FORI=0TO7:READA:POKEF*8+I
    +12288,A:NEXT:NEXT              :rem 213
910 FORI=0TO7:POKE32*8+I+12288,0:NEXT
                                    :rem 186
920 :                               :rem 213
930 POKE53281,2:POKE53281,7         :rem 251
940 PRINTLEFT$(S$,10)SPC(11)"{BLU}M {GRN}
    U{RED} N {BLU}C {BLK}H {GRN}M {RED}A
    {SPACE}{BLU}T {GRN}H"           :rem 207
950 POKE56334,PEEK(56334)AND254:POKE1,PEE
    K(1)AND251:Z=13312:Y=53248      :rem 96
960 FORI=0TO519:POKEI+Z,PEEK(I+Y):NEXT:FO
    RI=664TO671:POKEI+Z,PEEK(I+Y):NEXT
                                    :rem 68
970 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334
    )OR1                            :rem 143
980 POKE53272,28:PRINTLEFT$(S$,10)SPC(11)
    "{RVS}{BLU}M {GRN}U{RED} N {BLU}C
    {BLK}H {GRN}M {RED}A {BLU}T {GRN}H"
                                    :rem 231
990 :                               :rem 220
1000 READF,G:IFF=-1THEN1040          :rem 52
1010 POKESF+1,F:POKESF,G:POKEWV,33:GOSUB1
     60:POKEWV,32:GOSUB160          :rem 190
1020 GOTO1000                       :rem 189
1030 :                              :rem 254
1040 GOSUB150:FORI=4TO24:PRINTLEFT$(S$,10
     )SPC(I)" {CYN}= {RED}<{2 SPACES}
     {GRN}:";:GOSUB170               :rem 7
1050 PRINTLEFT$(S$,10)SPC(I)" {BLU}<
     {PUR}={2 SPACES}{GRN};"        :rem 72
1060 POKESF,195:POKESF+1,17:POKEWV,17:GOS
     UB170:POKEWV,16:NEXT           :rem 106
1070 PRINTLEFT$(S$,10)SPC(24)"{8 SPACES}"
                                    :rem 218
1080 POKE53280,5:POKE53281,7:POKE53272,21
                                    :rem 37
1090 PRINT"{CLR}"SPC(8)"{3 DOWN}{BLU}WHAT
     IS YOUR NAME";:GOSUB260:INPUTN$
                                    :rem 60
1100 PRINT"{CLR}{BLU}"SPC(13)"{5 DOWN}WHA
     T WOULD YOU":GOSUB260          :rem 135
1110 PRINTSPC(11)"{DOWN}LIKE TO PRACTICE,
     {DOWN}":GOSUB260               :rem 224
1120 PRINTSPC(20-LEN(N$)/2)N$":":GOSUB260
                                    :rem 92
1130 PRINTSPC(14)"{DOWN}{RED}1){GRN}ADDIT
     ION":GOSUB260                  :rem 117
1140 PRINTSPC(14)"{DOWN}{RED}2){GRN}SUBTR
     ACTION":GOSUB260               :rem 121
1150 PRINTSPC(14)"{DOWN}{RED}3){GRN}DIVIS
     ION":GOSUB260                  :rem 146
1160 PRINTSPC(14)"{DOWN}{RED}4){GRN}MULTI
     PICATION{BLU}":GOSUB260        :rem 48
1170 GETA$:Q=VAL(A$):IFQ<1ORQ>4THEN1170
                                    :rem 82
1180 PRINTLEFT$(S$,Q*2+10)SPC(14)"{RVS}"M
     ID$(STR$(Q),2)                 :rem 49
1190 PRINTLEFT$(S$,20)SPC(14)"LEVEL (1-9)
     ?"                             :rem 124
1200 GETA$:L=VAL(A$):IFL<1ORL>9THEN1200
                                    :rem 60
1210 GOTO680                        :rem 155
1220 :                              :rem 255
1230 PRINT"{RVS}? ";:AN$="":POKE198,0
                                    :rem 248
1240 GETZA$:IFZA$=""THEN1240        :rem 101
1250 ZL=LEN(AN$):IFZA$=CHR$(20)ANDZLTHENP
     RINTZA$;:AN$=LEFT$(AN$,ZL-1):rem 227
1260 IFZA$=CHR$(13)THENPRINT:RETURN
                                    :rem 224
1270 IFZA$<>"Q"AND(ZA$<"0"ORZA$>"9")ORZL=
     5THEN1240                      :rem 132
1280 PRINTZA$;:AN$=AN$+ZA$:GOTO1240
                                    :rem 83
1290 :                               :rem 6
1300 DATA 0,0,0,0,0,0,255,255       :rem 106
1310 DATA 3,3,3,3,3,3,3,3           :rem 171
1320 DATA 0,0,255,255,255,255,0,0 :rem 68
1330 DATA 24,60,110,126,126,126,60,24
                                    :rem 10
1340 DATA 56,124,95,248,224,248,127,56
                                    :rem 95
1350 DATA 60,126,255,219,255,255,169,169
                                    :rem 198
1360 DATA 60,126,255,219,255,255,90,180
                                    :rem 137
1370 DATA 120,116,30,14,30,124,120,0
                                    :rem 201
1380 DATA 0,0,0,14,14,14,0,0        :rem 57
1390 :                               :rem 7
1400 DATA 16,195,22,96,28,49,33,125,33,12
     5,33,125,33,125                :rem 195
1410 DATA 28,49,28,49,28,49,22,96,28,49,2
     2,96,16,195,-1,0               :rem 10
```

64 Timepiece

(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.

```
10 S=54272                                :rem 245
12 FORL=STOS+24:POKEL,0:NEXT              :rem 55
14 POKES+5,9:POKES+15,30:POKES+12,9
                                          :rem 55
20 POKE657,0:POKE53280,6                  :rem 92
30 DIMA(40),B(48)                         :rem 94
40 POKE53280,1:POKE53281,1:C=54272
                                          :rem 121
50 PRINT"{CLR}{RVS}{BLK}"TAB(13)"64 TIMEP
   IECE{OFF}":PRINTTAB(7)"{DOWN}TO SET TI
   ME,PRESS ANY KEY"                      :rem 143
60 PRINT"{DOWN}{3 SPACES}WHEN{2 SPACES}EN
   TERING{2 SPACES}TIME, GIVE HOURS AND M
   INUTES IN THIS MANNER:";               :rem 234
70 PRINT"{RED}{2 DOWN}{9 LEFT}915{RVS}RET
   URN{OFF} OR{13 LEFT}";                 :rem 243
80 PRINT"{RED}{DOWN}1231{RVS}RETURN{OFF}"
   :PRINTTAB(92)"(PRESS ANY KEY)":rem 254
83 POKE198,0                              :rem 152
85 GETA$:IFA$=""THEN85                    :rem 253
90 GOSUB1000                              :rem 170
100 REM DRAW CLOCK                        :rem 15
101 PRINT"{HOME}{BLK}{8 SPACES}{RVS}
   {24 SPACES}{OFF}"                      :rem 170
102 FORI=1TO23:PRINTTAB(8)"{RVS} {OFF}"TA
   B(31)"{RVS} {OFF}":NEXT                :rem 53
103 PRINT"{8 SPACES}{RVS}{24 SPACES}{OFF}
   ";                                     :rem 68
105 PRINT"{HOME}{2 DOWN}{RED}{RVS}"TAB(21
   )"{9 SPACES}"                          :rem 153
110 PRINT"{BLK}{3 DOWN}{9 RIGHT}{RVS}
   {2 SPACES}{UP}{LEFT}{OFF}{RED}{RVS}
   {DOWN}1234{RED}{8 SPACES}"             :rem 178
120 PRINT"{4 DOWN}{9 RIGHT}{BLK}{RVS} 112
   2334455{RED}{5 SPACES}"                :rem 105
130 PRINT"{9 RIGHT}{BLK}{RVS}505050505
   {RED}{5 SPACES}"                       :rem 86
140 PRINT"{9 RIGHT}{BLK}{4 DOWN}{RVS}
   {9 SPACES}111{RED}{5 SPACES}"
150 PRINT"{9 RIGHT}{BLK}{RVS}1234567890 12
   {RED}{4 SPACES}{OFF}£"                 :rem 213
160 PRINTTAB(26)"{RED}{RVS}£{DOWN}
   {2 LEFT}£{OFF}£{DOWN}{3 LEFT}{RVS}
   £{OFF}£"                               :rem 220
170 PRINTTAB(9)"{DOWN}{BLU}{5 SPACES}QQQQ
   QQQQQQQQQQQQ{5 UP}";:POKE1974,81:POKE
   1974+C,6                               :rem 251
200 U=0                                   :rem 84
210 FORI=1974TO1094STEP-40:A(U)=I:U=U+1:N
   EXT                                    :rem 176
212 FORI=1093TO1084STEP-1:A(U)=I:U=U+1:NE
   XT                                     :rem 118
214 FORI=1123TO1243STEP40:A(U)=I:U=U+1:NE
   XT                                     :rem 117
216 FORI=1242TO1239STEP-1:A(U)=I:U=U+1:NE
   XT                                     :rem 120
218 U=1                                   :rem 94
220 FORI=1235TO1247:B(U)=I:U=U+1:NEXT
                                          :rem 219
222 FORI=1287TO1447STEP40:B(U)=I:U=U+1:NE
   XT                                     :rem 134
224 B(U)=1448:B(U+1)=1449:U=U+2           :rem 153
226 FORI=1489TO1689STEP40:B(U)=I:U=U+1:NE
   XT                                     :rem 150
227 B(U)=1690:B(U+1)=1730:B(U+2)=1770:B(U
   +3)=1809                               :rem 48
228 B(U+4)=1848:B(U+5)=1887:B(U+6)=1926:G
   OTO900                                 :rem 42
300 REM START OF BALL MOVEMENT ROUTINE
                                          :rem 70
310 FORX=0TO40                            :rem 71
320 POKEA(X),81:POKEA(X)+C,6              :rem 102
325 IFX>22THENPOKEA(X-1),32               :rem 90
330 IFX>1ANDX<24THENPOKEA(X-1),99:POKEA(X
   -1)+C,6                                :rem 126
340 IFX>2THENPOKEA(X-2),32                :rem 38
350 GOSUB2000:NEXT:POKE1974,81:POKE1974+C
   ,6                                     :rem 52
400 REM CONTINUE TO MINUTES               :rem 165
410 L=A(40)                               :rem 20
420 IFPEEK(L-1)=81THEN500                 :rem 175
430 L=L-1:POKEL+1,32:POKEL,81:POKEL+C,6:I
   FPEEK(L-1)=81THEN900                   :rem 146
440 IFL=A(40)-4THEN900                    :rem 207
450 GOSUB2010:GOTO430                     :rem 230
460 POKES+4,20:POKES+11,20:POKES+24,0
                                          :rem 137
500 REM MINUTES FULL,{9 SPACES}GOTO 5'S
                                          :rem 229
505 Z=1447                                :rem 1
510 FORX=5TO22:POKEB(X),81:POKEB(X)+C,6:P
   OKEB(X-5),32:GOSUB2040:NEXT            :rem 155
520 FORX=23TO38:POKEB(X),81:POKEB(X)+C,6
                                          :rem 147
525 Z=Z-1:IFPEEK(Z)<>81ANDZ>1432THENPOKEZ
   ,81:POKEZ+C,6:POKEZ+1,32               :rem 44
530 POKEB(X-4),32:GOSUB2080:NEXT:POKES+4,
   20:POKES+11,20:POKES+24,0              :rem 202
600 IFPEEK(1444)<>81THEN900:REM OR 5'S AR
   E FULL                                 :rem 140
605 Z=1435                                :rem 255
610 POKE1444,81:POKE1444+C,6:GOSUB2080
                                          :rem 178
612 FORX=1445TO1447:POKEX,81:POKEX+C,6:PO
   KEX-12,32:GOSUB2080:NEXT               :rem 28
620 FORX=18TO28:POKEB(X),81:POKEB(X)+C,6
                                          :rem 151
630 Z=Z+1:POKEZ,32:GOSUB2060:NEXT         :rem 53
640 FORX=29TO37:POKEB(X),81:POKEB(X)+C,6:
   POKEB(X-12),32                         :rem 16
650 GOSUB2040:NEXT                        :rem 90
659 Z=B(37-11)                            :rem 199
660 FORX=37TO44:POKEB(X),81:POKEB(X)+C,6:
   POKEB(X-11),32                         :rem 14
670 Z=Z-1:IFPEEK(Z)<>81THENPOKEZ,81:POKEZ
   +C,6:POKEZ+1,32                        :rem 248
680 GOSUB2040:NEXT                        :rem 93
690 IFZ-1>1672ANDPEEK(Z-1)<>81THEN692
                                          :rem 168
691 GOTO695                               :rem 125
692 Z=Z-1:POKEZ,81:POKEZ+C,6:POKEZ+1,32:G
   OSUB2015:GOTO690                       :rem 31
695 POKES+4,20:POKES+11,20:POKES+24,0
                                          :rem 147
700 IFPEEK(1685)<>81THEN900               :rem 27
710 REM HOURS ARE FILLED UP               :rem 58
720 FORX=1686TO1690:POKEX,81:POKEX+C,6:PO
   KEX-12,32                              :rem 38
730 GOSUB2100:NEXT                        :rem 86
740 FORX=1TO12:POKEB(X+26),81:POKEB(X+26)
   +C,6:POKE1678+X,32                     :rem 212
750 GOSUB2100:NEXT                        :rem 88
760 FORX=1TO7:POKEB(27+X),32:GOSUB2110:NE
   XT                                     :rem 148
900 REM TIME FOR ACTION                   :rem 81
905 POKES+4,20:POKES+11,20:POKES+24,0
                                          :rem 141
910 IFRIGHT$(TI$,2)>"57"THEN300           :rem 235
920 GETA$:IFA$<>""THENRUN                 :rem 240
930 GOTO900                               :rem 110
1000 REM TIME SETTING ROUTINE             :rem 24
1010 PRINT"{CLR}"TAB(53)"TIME?{HOME}"TAB(
```

```
    90)"(IN 3 OR 4 DIGITS)"          :rem 4
1020 INPUTA$                         :rem 184
1030 A$="0"+A$:A$=RIGHT$(A$,4)       :rem 254
1040 A=VAL(LEFT$(A$,2)):B=VAL(RIGHT$(A$,2
     )):D=INT(B/5):B=B-5*D           :rem 229
1045 IFA>12ORD>11THEN1000            :rem 176
1050 REM SETTING CLOCK               :rem 52
1055 PRINT"{CLR}"                    :rem 47
1060 FORX=1TOA:POKE1672+X,81:POKE1672+X+C
     ,6:NEXT:REM HOURS               :rem 93
1070 X=0                             :rem 141
1080 X=X+1:IFX>BTHEN1100             :rem 164
1090 POKE1234+X,81:POKE1234+X+C,6:GOTO108
     0:REM MINUTES                   :rem 224
1100 X=0                             :rem 135
1110 X=X+1:IFX>DTHEN1130             :rem 163
1120 POKE1432+X,81:POKE1432+X+C,6:GOTO111
     0                               :rem 145
1130 GOTO100                         :rem 143
2000 REM SOUND                       :rem 47
2005 GOTO2020                        :rem 196
2010 IFX=37THEN2190                  :rem 76
2015 POKES+24,10:POKES+11,21:FORV=1TO4:PO
     KES+8,130                       :rem 219
2017 FORW=1TO2:NEXT:POKES+8,240:NEXT:RETU
     RN                              :rem 56
2020 IFX<24THENFORY=1TO55:NEXT:RETURN
                                     :rem 202
2030 IFX>33ANDX<37THENPOKES+4,20:POKES+11
     ,20:RETURN                      :rem 176
2035 GOTO2010                        :rem 198
2040 IFPEEK(1447)=81THENGOTO2190     :rem 117
2060 IFPEEK(1689)=81THENGOTO2190     :rem 127
2080 IFPEEK(B(29))=81THENGOTO2190:rem 167
2085 IFPEEK(B(32))=81THENGOTO2190:rem 166
2090 IFPEEK(B(26))=81THENGOTO2190:rem 165
2100 IFPEEK(1770)=81THENGOTO2190     :rem 113
2110 IFPEEK(1926)=81THENGOTO2190     :rem 117
2120 GOTO2015                        :rem 198
2180 IFX>4THEN2015                   :rem 27
2190 POKES+24,15:POKES+4,21:POKES+1,130:F
     ORY=1TO9:NEXT                   :rem 48
2195 POKES+4,20:POKES+11,20:FORY=1TO60:NE
     XT:RETURN                       :rem 155
```

64 Aardvark Attack

(Article on page 82.)

BEFORE TYPING...

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

```
100 PRINT"{CLR}":S=54272:FORI=0TO24:POKES
    +I,0:NEXT:S1=S+7:S2=S+14:POKES+24,15
                                     :rem 22
110 POKE53280,2:POKE53281,14:GOSUB720
                                     :rem 113
120 PRINT"{CLR}{2 DOWN}{BLK} INSTRUCTIONS
    (Y/N)?"                          :rem 196
130 GETG$:IFG$=""THEN130             :rem 87
140 IFG$="Y"THENGOSUB880             :rem 184
150 PRINT"{CLR}"SPC(9)"{DOWN}{BLK}{RVS}
    {3 SPACES}AARDVARK{2 SPACES}ATTACK
    {3 SPACES}"                      :rem 101
160 PRINTSPC(10)"{2 DOWN}{WHT}ALPHANUMERI
    C WEAPONS"                       :rem 189
170 PRINTSPC(14)"{4 DOWN}{WHT}WHICH LEVEL
    ?":PRINTSPC(10)"{DOWN}{BLK}(1:NOVICE
    {2 SPACES}9:EXPERT)"             :rem 135
180 GETHF$:IFHF$<"1"ORHF$>"9"THEN180
                                     :rem 45
190 HF=INT(2.5↑(9-VAL(HF$)))         :rem 49
200 SC=0:GOSUB790                    :rem 239
210 :                                :rem 205
220 REM NEW LETTER                   :rem 50
230 GOSUB820:POKE56334,PEEK(56334)AND254:
    POKE1,PEEK(1)AND251              :rem 10
240 FORI=0TO3:FORJ=0TO3:C(I,J)=0:NEXT:NEX
    T                                :rem 163
250 L=INT(RND(1)*26+1):B=53248+L*8:F=0
                                     :rem 53
260 FORI=0TO3                        :rem 11
270 F=ABS(F-1):T=PEEK(B+2*I+1-F)     :rem 9
280 FORJ=0TO3:M=(FNM(J)ANDT)/(4↑J):C(I,J)
    =C(I,J)+INT(M*(4↑F)):NEXT        :rem 191
290 IFFTHEN270                       :rem 56
300 NEXT:POKE1,PEEK(1)OR4:POKE56334,PEEK(
    56334)OR1                        :rem 251
310 :                                :rem 206
320 REM FLASHING                     :rem 197
330 POKES+5,0:POKES+6,240:POKES,0:POKES+1
    ,0:POKES+4,33                    :rem 236
340 FORY=1TO70:Z=INT(RND(1)*10):ZZ=1037+Z
    :POKEZZ+54272,1:POKEZZ,160       :rem 81
350 POKES+1,RND(1)*15+10:POKEZZ+54272,0:N
    EXT:POKES+4,8                    :rem 196
360 FORI=0TO3:FORJ=0TO3:POKE1135-J+40*I,S
    (C(I,J)):NEXT:NEXT               :rem 123
370 LF=0:FH=0:SH=0                   :rem 16
380 :                                :rem 213
390 REM START ATTACK                 :rem 198
400 POKES1+5,0:POKES1+6,240:POKES1,0:POKE
    S1+1,0:POKES1+4,17:POKE198,0     :rem 72
410 FORX=1TO18:POKES1+1,(19-X)*7     :rem 100
420 GETG$:IFG$=""THEN480             :rem 97
430 IFLFTHEN460                      :rem 129
440 IF(ASC(G$)-64)<>LTHEN480         :rem 51
450 LF=1:FH=X:N=S:GOSUB1010          :rem 138
460 IFVAL(G$)-1<>ZAND(G$<>"0"ORZ=9)THEN48
    0                                :rem 179
470 SH=X:POKE198,0:GOTO590           :rem 67
480 FORY=1TOHF:NEXT:POKE(1037+Z+X*40),81+
    LF*128:NEXT:POKES1+4,8           :rem 79
490 :                                :rem 215
500 REM LOSE ROUND                   :rem 52
510 POKE198,0:N=S2:GOSUB1010:FORI=2TO18:P
    OKE53280,IAND15:NEXT
520 FORXX=0TO3:PB=(1797+Z+(40*XX)):IFPEEK
    (PB)<>32THEN540                  :rem 90
530 NEXT:GOTO620                     :rem 226
540 IFXX=3ORLFTHEN560                :rem 69
550 POKEPB+40,32                     :rem 123
560 POKEPB,32:GOTO230                :rem 245
570 :                                :rem 214
580 REM SCORING                      :rem 150
590 N=S2:GOSUB1010:POKES1+4,8:SC=SC+INT((
    500-5*FH-2*SH)/SQR(HF)):GOTO230
                                     :rem 20
600 :                                :rem 208
610 REM LOSE ROUTINE                 :rem 212
620 FORY=1TO100:POKE53280,RND(1)*255:POKE
    53281,RND(1)*255:NEXT            :rem 122
630 POKE53280,2:POKE53281,14:PRINT"{CLR}
    {DOWN}{BLK} ";:FORI=1TO38:PRINT"*";:N
    EXT                              :rem 110
```

```
640 PRINTSPC(12)"{DOWN}{WHT}SORRY, YOU LO
    ST."                              :rem 83
650 PRINTSPC(13)"{3 DOWN}{BLU}YOUR SCORE"
    ;SC                               :rem 245
660 PRINTSPC(15)"{DOWN}AT LEVEL ";HF$
                                      :rem 21
670 PRINTSPC(11)"{3 DOWN}{BLK}PLAY AGAIN
    {SPACE}(Y/N)?"                    :rem 198
680 GETY$:IFY$=""THEN680              :rem 143
690 IFY$="N"THENPRINT"{CLR}":END      :rem 30
700 GOTO150                           :rem 102
710 :                                 :rem 210
720 DIMS(15),C(3,3)                   :rem 207
730 DEFFNM(X)=3*(4↑X)                 :rem 200
740 FORX=0TO15:READS(X):NEXT:RETURN
                                      :rem 53

750 DATA32,108,123,98,124,225,255,254
                                      :rem 86
760 DATA126,127,97,252,226,251,236,160
                                      :rem 139
770 :                                 :rem 216
780 REM SCREENS                       :rem 150
790 PRINT"{CLR}"                       :rem 4
800 POKE214,18:PRINT:PRINTSPC(10)"{BLK}
    {Q}{2 SPACES}1234567890{2 SPACES}
    {W} LEVEL "HF$                    :rem 130
810 FORX=1TO4:PRINTSPC(10)"{Q}
    {2 SPACES}{BLU}{RVS}{10 SPACES}{OFF}
    {BLK}{2 SPACES}{W}":NEXT          :rem 234
820 PRINT"{HOME}"SPC(10)"{BLK}{Q}
    {2 SPACES}{RED}**********{2 SPACES}
    {BLK}{W}"                         :rem 152
830 PRINT"{HOME}":FORI=1TO6:PRINTSPC(27)"
    {RED}{RVS}{6 SPACES}":NEXT        :rem 130
840 PRINT"{HOME}{DOWN}":FORI=1TO4:PRINTSP
    C(28)"{BLK}{RVS}{4 SPACES}":NEXT
                                      :rem 7
850 PRINT"{HOME}":FORI=1TO18:PRINTSPC(10)
    "{BLK}{Q}{2 SPACES}{CYN}{10 SPACES}
    {BLK}{2 SPACES}{W}":NEXT          :rem 158

860 POKE214,16:PRINT:PRINTSPC(27)"SCORE"S
    C:RETURN                          :rem 149
870 :                                 :rem 217
880 REM INSTRUCTIONS                   :rem 57
890 PRINT"{CLR}{DOWN}{BLK} ";:FORI=1TO38:
    PRINT"*";:NEXT:PRINT              :rem 119
900 PRINT"{DOWN}ANDROMEDAN AARDVARKS ARE
    {SPACE}ATTACKING EARTH"           :rem 213
910 PRINTSPC(6)"{DOWN}{WHT}THEY HAVE 26 K
    INDS OF BOMBS":PRINTSPC(18)"{DOWN}AND
    "                                 :rem 196
920 PRINTSPC(6)"{DOWN}THEY'RE ATTACKING 1
    0 CITIES!"                        :rem 187
930 PRINT"{3 DOWN}{BLK} YOUR ALPHANUMERIC
     RADAR CAN SAVE EARTH"            :rem 215
940 PRINTSPC(4)"{3 DOWN}{WHT}FIRST: RECOG
    NIZE THE BOMB (A-Z)"              :rem 147
950 PRINTSPC(7)"{DOWN}THEN: SAVE THE CITY
     (0-9)"                           :rem 135
960 PRINTSPC(9)"{2 DOWN}{BLK}{RVS}HIT ANY
     KEY TO PLAY"                      :rem 51
970 GETG$:IFG$=""THEN970              :rem 111
980 RETURN                            :rem 129
990 :                                 :rem 220
1000 REM EXPLOSION                    :rem 102
1010 POKEN+5,37:POKEN+6,252:POKEN,100:POK
    EN+1,5:POKEN+4,129:POKEN+4,128
                                      :rem 181
1020 RETURN                           :rem 163
```

Connect The Dots

(Article on page 88.)

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:

Connect The Dots — VIC Version

```
1 PRINT"{CLR}{3 SPACES}QQ{3 SHIFT-SPACE}Q
  Q{2 SHIFT-SPACE}QQQ{2 SHIFT-SPACE}QQ
  {6 SHIFT-SPACE}Q{SHIFT-SPACE}Q
  {SHIFT-SPACE}Q{2 SHIFT-SPACE}Q
  {2 SHIFT-SPACE}Q{2 SHIFT-SPACE}Q
  {8 SHIFT-SPACE}Q{SHIFT-SPACE}Q
  {SHIFT-SPACE}Q{2 SHIFT-SPACE}Q
  {2 SHIFT-SPACE}Q{3 SHIFT-SPACE}Q:rem 88
2 PRINT"{3 SPACES}Q{SHIFT-SPACE}Q
  {SHIFT-SPACE}Q{2 SHIFT-SPACE}Q
  {2 SHIFT-SPACE}Q{4 SHIFT-SPACE}Q
  {6 SHIFT-SPACE}QQ{3 SHIFT-SPACE}QQ
  {3 SHIFT-SPACE}Q{2 SHIFT-SPACE}QQ
                                      :rem 126
3 A$="Z":POKE36878,15:GOSUB910        :rem 210
6 PRINT"{5 SPACES}{2 DOWN}{RVS}{PUR}INSTR
  UCTIONS":PRINT"{DOWN}WHEN THE GRID APPE
  ARS,FIND THE SOLID DOT.             :rem 137
7 PRINT"{DOWN}TYPE THE NUMBER OF THEROW A
  ND HIT RETURN.{3 SPACES}{DOWN}TYPE THE
  {SPACE}LETTER OF THE";              :rem 238
8 PRINT"COLUMN AND HIT RETURN.":PRINT"
  {2 DOWN}{BLK} HIT ANY KEY TO PLAY.
                                      :rem 245
9 GETS$:IFS$=""THEN9                  :rem 185
10 PRINT"{CLR}{2 DOWN}{RED}{2 SPACES}WHAT
    WOULD YOU LIKE"SPC(9)"TO DRAW?":PRINT
   "{BLU}{DOWN}{RVS}A{OFF}=BUTTERFLY (22
   {SPACE}DOTS)                       :rem 147
11 PRINT"{DOWN}{RVS}B{OFF}=MUSHROOM (12 D
   OTS)":PRINT"{DOWN}{RVS}C{OFF}=DOG (20
   {SPACE}DOTS)                       :rem 180
12 PRINT"{DOWN}{RVS}D{OFF}=HEART (10 DOTS
   )                                  :rem 168
18 GETA$:IFA$=""THEN18                :rem 245
19 L=7746:C=38466:POKE36879,26:CD=30720
                                      :rem 152
20 PRINT"{CLR}{2 DOWN}{2 SPACES}ABCDEFGHI
   JKLMNOPQR                          :rem 241
30 FORT=1TO418:POKEL,79:L=L+1:POKEC,3:C=C
   +1:NEXT                            :rem 229
40 PRINT"{HOME}{21 DOWN}{2 SPACES}ABCDEFG
   HIJKLMNOPQR{2 SPACES}";            :rem 19
50 PRINT"{HOME}{3 DOWN} 1"SPC(18)"1
   {2 SPACES}2"SPC(18)"2{2 SPACES}3"SPC(1
   8)"3{2 SPACES}4"SPC(18)"4{2 SPACES}5"S
   PC(18)"5 ";                        :rem 41
60 PRINT" 6"SPC(18)"6{2 SPACES}7"SPC(18)"
   7{2 SPACES}8"SPC(18)"8{2 SPACES}9"SPC(
   18)"9 10"SPC(18)"1011";            :rem 198
65 PRINTSPC(18)"11"                    :rem 62

70 PRINT"{UP}12"SPC(18)"1213"SPC(18)"1314
   "SPC(18)"1415"SPC(18)"1516"SPC(18)"161
   7"SPC(18)                          :rem 55
```

```
80 PRINT"1718"SPC(18)"18{HOME}"  :rem 106
89 RESTORE                       :rem 149
90 READB$:IFB$=A$THEN92          :rem 86
91 GOTO90                        :rem 12
92 IFA$="A"THEN500               :rem 229
93 IFA$="B"THEN550               :rem 236
94 IFA$="C"THEN600               :rem 234
95 IFA$="D"THEN650               :rem 241
99 FORT=1TO500:NEXT              :rem 204
100 READA,B,E$,S:IFA=0THEN900    :rem 175
101 POKEA,81:POKEA+CD,B          :rem 97
105 PRINT"{HOME}{43 SPACES}{UP}" :rem 11
110 INPUT"{BLU}{HOME}←NUMBER";N$:rem 109
115 FORT=1TO500:NEXT             :rem 241
120 PRINTSPC(11)"{2 UP}↑LETTER";:INPUTL$
                                 :rem 194
130 IFE$=N$+L$THEN200            :rem 151
140 PRINT"{UP}{6 SPACES}TRY AGAIN":POKE36
    874,209:FORT=1TO500:NEXT:POKE36874,19
    5                            :rem 102
150 FORT=1TO500:NEXT:POKE36874,0 :rem 193
160 FORT=1TO1500:NEXT:GOTO105    :rem 43
200 FORT=1TO1000:NEXT:FORT=1TO18:POKESQ,1
    60:POKESQ+CD,B:IFSQ=ATHEN220 :rem 178
210 SQ=SQ+S:NEXT                 :rem 15
220 FORZ=200TO235:POKE36876,Z:NEXT:POKE36
    876,0:GOTO99                 :rem 15
500 POKE7799,77:POKE7799+CD,0:POKE7801,78
    :POKE7801+CD,0:POKE7822,160:POKE7822+
    CD,5                         :rem 254
510 SQ=7822:GOTO99               :rem 47
550 POKE7976,160:POKE7976+CD,4:POKE7977,1
    60:POKE7977+CD,4:POKE8058,160:POKE805
    8+CD,5                       :rem 118
560 SQ=8058:GOTO99               :rem 54

600 POKE7903,74:POKE7903+CD,0:POKE7904,75
    :POKE7904+CD,0:POKE7882,85:POKE7882+C
    D,0                          :rem 196
610 POKE7783,160:POKE7783+CD,2:SQ=7783:GO
    TO99                         :rem 141
650 PRINT"{8 DOWN}{7 RIGHT}{GRN}MATTHEW
    {2 DOWN}{8 LEFT}{BLU}JONATHAN{2 DOWN}
    {6 LEFT}{YEL}EMILY"          :rem 97
660 POKE7822,160:POKE7822+CD,2:SQ=7822:GO
    TO99                         :rem 128
800 PRINT"{HOME}{6 SPACES}GOOD JOB!
    {12 SPACES}DRAW AGAIN?{27 SPACES}"
                                 :rem 110
810 Y$="":GETY$:IFY$=""THEN810   :rem 189
820 IFY$="Y"THEN1                :rem 224
830 IFY$="N"THENPRINT"{CLR}{8 DOWN}
    {7 SPACES}BYE NOW!":FORT=1TO3000:NEXT
    :END                         :rem 89
840 GOTO810                      :rem 110
900 PRINT"{HOME}{6 SPACES}GOOD JOB!
    {12 SPACES}DRAW AGAIN?{27 SPACES}"
                                 :rem 111
910 READB$:IFB$="Z"THEN930       :rem 241
920 GOTO910                      :rem 110
930 READP,D:IFP=-1ANDA$="Z"THENPOKE36875,
    0:RETURN                     :rem 32
940 IFP=-1THENPOKE36875,0:GOTO960:rem 240
950 POKE36875,P:FORT=1TOD:NEXT:GOTO930
                                 :rem 168
960 Y$="":GETY$:IFY$=""THEN960   :rem 201
970 IFY$="Y"THEN10               :rem 22
980 IFY$="N"THENPRINT"{CLR}{8 DOWN}
    {7 SPACES}BYE NOW!":FORT=1TO3000:NEXT
    :END                         :rem 95
990 GOTO960                      :rem 122
1000 DATAA,7888,5,7I,22,7804,2,3M,-21
                                 :rem 137
1010 DATA7806,2,3O,1,7852,2,5Q,23,7940,2,
     9Q,22,7982,2,11O,21,8028,7,13Q,23
                                 :rem 4
1020 DATA8072,7,15Q,22,8093,7,16P,21,8090
     ,7,16M,-1,7998,7,12I,-23,8082,7,16E,
     21                          :rem 241
1030 DATA8079,7,16B,-1,8056,7,15A,-23,801
     2,7,13A,-22,7970,7,11C,-21,7924,2,9A
     ,-23                        :rem 10
1040 DATA7836,2,5A,-22,7794,2,3C,-21,7796
     ,2,3E,1,7888,2,7I,23,8042,5,14I,22,0
     ,0,0,0                      :rem 122
1500 DATAB,8063,5,15H,1,7975,4,11H,-22,79
     72,4,11E,-1,7950,4,10E,-22,7908,4,8G
     ,-21                        :rem 47
1510 DATA7913,4,8L,1,7959,4,10N,23,7981,4
     ,11N,22,7978,4,11K,-1,8066,4,15K,22
                                 :rem 100
1520 DATA8071,5,15P,1,8063,5,15H,-1,0,0,0
     ,0                          :rem 132
2000 DATAC,7867,2,6J,21,7862,2,6E,-1,7883
     ,2,7D,21,7993,2,12D,22,8039,2,14F,23
                                 :rem 123
2010 DATA8040,2,14G,1,8018,2,13G,-22,7995
     ,2,12F,-23,7973,2,11F,-22,7977,2,11J
     ,1                          :rem 184
2020 DATA8046,2,14M,23,8047,2,14N,1,8003,
     2,12N,-22,7980,2,11M,-23,7892,2,7M,-
     22                          :rem 209
2030 DATA7850,2,5O,-21,7852,0,5Q,1,7830,0
     ,4Q,-22,7829,0,4P,-1,7783,2,2N,-23,0
     ,0,0,0                      :rem 139
2500 DATAD,7753,2,1F,-23,7750,2,1C,-1,779
     2,2,3A,21,7924,2,9A,22,8108,2,17I,23
                                 :rem 94
2510 DATA7940,2,9Q,-21,7808,2,3Q,-22,7762
     ,2,1O,-23,7759,2,1L,-1,7822,2,4I,21,
     0,0,0,0,0                   :rem 188
4000 DATAZ,225,300,231,100,235,200,240,20
     0,235,200,231,200,225,200,231,200,23
     5,300                       :rem 203
4010 DATA231,100,225,200,215,200,225,100,
     0,300,225,100,-1,0          :rem 60
```

Program 2:

Connect The Dots — 64 Version

```
100 REM TITLE PAGE               :rem 20
110 PRINT"{CLR}{7}":POKE53280,0:POKE532
    81,0                         :rem 31
120 PRINTSPC(10)"QQQ{3 SHIFT-SPACE}QQ
    {2 SHIFT-SPACE}QQQQQ{2 SHIFT-SPACE}QQ
    "                            :rem 40
130 PRINTSPC(10)"Q{SHIFT-SPACE} Q
    {SHIFT-SPACE}Q{2 SHIFT-SPACE}Q
    {2 SHIFT-SPACE} Q{SHIFT-SPACE}
    {SHIFT-SPACE}Q{2 SPACES}Q"   :rem 180
140 PRINTSPC(10)"Q {SHIFT-SPACE}Q
    {SHIFT-SPACE}Q{2 SHIFT-SPACE}Q
    {2 SHIFT-SPACE} Q{SHIFT-SPACE}
    {2 SHIFT-SPACE}Q"            :rem 132
150 PRINTSPC(10)"Q {SHIFT-SPACE}Q
    {SHIFT-SPACE}Q{2 SHIFT-SPACE}Q
    {2 SHIFT-SPACE} Q{SHIFT-SPACE}
    {2 SPACES}{2 SHIFT-SPACE}Q"  :rem 133
160 PRINTSPC(10)"Q{SHIFT-SPACE} Q
    {SHIFT-SPACE}Q{2 SHIFT-SPACE}Q
    {2 SHIFT-SPACE} Q{SHIFT-SPACE}
    {2 SPACES}Q{2 SHIFT-SPACE}Q"  :rem 87
```

```
170 PRINTSPC(10)"QQQ{2 SHIFT-SPACE} QQ
    {3 SHIFT-SPACE} Q {SHIFT-SPACE}
    {SHIFT-SPACE}QQ"                  :rem 233
180 L=1114:C=55386:CD=54272:WV=54276
                                      :rem 220
190 A$="Z":POKE54296,15:POKE54277,22:POKE
    54278,165:GOSUB840                :rem 114
200 :                                 :rem 204
210 REM INSTRUCTIONS                  :rem 44
220 PRINTSPC(13)"{2 DOWN}{WHT}INSTRUCTION
    S:"                               :rem 22
230 PRINTSPC(9)"{DOWN}WHEN THE GRID APPEA
    RS,":PRINTSPC(10)"FIND THE SOLID DOT.
    "                                 :rem 141
240 PRINTSPC(9)"{DOWN}TYPE THE NUMBER OF
    {SPACE}THE":PRINTSPC(10)"ROW AND HIT
    {SPACE}RETURN."                   :rem 171
250 PRINTSPC(6)"{DOWN}THEN TYPE THE LETTE
    R OF THE"                         :rem 126
260 PRINTSPC(9)"COLUMN AND HIT RETURN."
                                      :rem 109
270 PRINTSPC(10)"{2 DOWN}{RVS}HIT ANY KEY
     TO PLAY.":POKE198,0              :rem 90
280 GETS$:IFS$=""THEN280              :rem 123
290 :                                 :rem 213
300 REM DRAW SELECTION                :rem 75
310 POKE53281,6:PRINT"{CLR}"SPC(6)"
    {5 DOWN}{3}WHAT WOULD YOU LIKE TO D
    RAW?"                             :rem 83
320 PRINTSPC(8)"{7}{3 DOWN}A - BUTTERFL
    Y (22 DOTS)"                      :rem 192
330 PRINTSPC(9)"{DOWN}B - MUSHROOM (12 DO
    TS)":PRINTSPC(11)"{DOWN}C - HORSE (20
     DOTS)"                           :rem 82
340 PRINTSPC(10)"{DOWN}D - HEART (10 DOTS
    )":POKE198,0                      :rem 73
350 GETA$:IFA$<"A"ORA$>"D"THEN350     :rem 95
360 :                                 :rem 211
370 REM DRAW BOARD                    :rem 20
380 PRINT"{CLR}{3}"SPC(11)"{2 DOWN}ABCD
    EFGHIJKLMNOPQR{HOME}"             :rem 210
390 FORRH=1TO18:FORT=1TO18:POKEL+T+RH*40,
    79:POKEC+T+RH*40,14:NEXT:NEXT:rem 170
400 PRINTSPC(11)"{3}{20 DOWN}ABCDEFGHIJ
    KLMNOPQR"                         :rem 19
410 PRINT"{HOME}{3 DOWN}{7}";         :rem 129
420 FORI=1TO18:PRINTSPC(8)RIGHT$(STR$(I),
    2)SPC(19)"{G}"RIGHT$(STR$(I),2):NEX
    T                                 :rem 137
430 :                                 :rem 209
440 REM FIND DATA                     :rem 183
450 RESTORE                           :rem 189
460 READB$:IFB$<>A$THEN460            :rem 243
470 ONASC(A$)-64GOTO650,690,730,770
                                      :rem 139
480 FORT=1TO500:NEXT                  :rem 246
490 READA,B,E$,S:IFA=0THEN830         :rem 189
500 POKEA,81:POKEA+CD,B               :rem 100
510 PRINT"{HOME}{39 SPACES}"          :rem 122
520 PRINT"{7}{HOME} (→) NUMBER";:GOSUB
    930:N$=IN$                        :rem 195
530 PRINT"{HOME}"SPC(20)"(↑) LETTER";:GOS
    UB930:L$=IN$                      :rem 11
540 IFE$=N$+L$THEN610                 :rem 161
550 PRINT"{HOME}{BLK}{15 SPACES}TRY AGAIN
    {10 SPACES}"                      :rem 109
560 POKECD,48:POKECD+1,11:POKEWV,33:POKEW
    V,32                              :rem 18
570 FORT=1TO400:NEXT:POKECD,195:POKECD+1,
    16:POKEWV,33:POKEWV,32            :rem 222
580 FORT=1TO400:NEXT                  :rem 246
590 FORT=1TO1200:NEXT:GOTO510         :rem 47
600 :                                 :rem 208
610 FORT=1TO700:NEXT:FORT=1TO18:POKESQ,16
    0:POKESQ+CD,B:IFSQ=ATHEN630       :rem 146
620 SQ=SQ+S:NEXT                      :rem 20
630 POKEWV,17:FORZ=9TO26:POKECD+1,Z:POKEC
    D,0:NEXT:POKEWV,16:GOTO480        :rem 84
640 :                                 :rem 212
650 POKE1242,77:POKE1242+CD,0:POKE1244,78
    :POKE1244+CD,0                    :rem 126
660 POKE1283,160:POKE1283+CD,5        :rem 166
670 SQ=1283:GOTO480                   :rem 91
680 :                                 :rem 216
690 POKE1563,160:POKE1563+CD,4:POKE1564,1
    60:POKE1564+CD,4                  :rem 241
700 POKE1717,160:POKE1717+CD,5        :rem 165
710 SQ=1717:GOTO480                   :rem 88
720 :                                 :rem 211
730 POKE1436,74:POKE1436+CD,0:POKE1437,75
    :POKE1437+CD,0                    :rem 137
740 POKE1397,85:POKE1397+CD,0         :rem 130
750 POKE1208,160:POKE1208+CD,2:SQ=1208:GO
    TO480                             :rem 146
760 :                                 :rem 215
770 PRINT"{HOME}{7 DOWN}"             :rem 249
780 PRINTSPC(16)"{GRN}MATTHEW"        :rem 70
790 PRINTSPC(16)"{2 DOWN}{7}JONATHAN"
                                      :rem 30
800 PRINTSPC(17)"{2 DOWN}{3}EMILY"
                                      :rem 64
810 POKE1283,160:POKE1283+CD,2:SQ=1283:GO
    TO480                             :rem 152
820 :                                 :rem 212
830 PRINT"{HOME}{10 SPACES}GOOD JOB! DRAW
     AGAIN?{3 SPACES}"                :rem 113
840 READB$:IFB$<>"Z"THEN840           :rem 48
850 READPL,PH,D:IFPL=-1ANDA$="Z"THENPOKEW
    V,0:RETURN                        :rem 29
860 IFPL=-1THENPOKEWV,0:GOTO890       :rem 223
870 POKECD,PL:POKECD+1,PH:POKEWV,33:FORT=
    1TOD*75:NEXT:POKEWV,32            :rem 85
880 GOTO850                           :rem 118
890 GETY$:IFY$<>"Y"ANDY$<>"N"THEN890
                                      :rem 135
900 IFY$="Y"THEN310                   :rem 66
910 :                                 :rem 212
920 PRINT"{CLR}";:END                 :rem 75
930 PRINT"? ";:IN$=""                 :rem 93
940 PRINT"{RVS} {OFF}{LEFT}";         :rem 234
950 GETA$:IFA$=""THEN940              :rem 94
960 ZL=LEN(IN$):IFA$=CHR$(20)ANDZLTHENPRI
    NTA$;:IN$=LEFT$(IN$,ZL-1)         :rem 30
970 IFA$=CHR$(13)ANDZLTHENPRINT" ":RETURN
                                      :rem 26
980 IF(A$<"0"ORA$>"R")OR(A$>"9"ANDA$<"A")
    ORLEN(IN$)=2THEN950               :rem 67
990 PRINTA$;:IN$=IN$+A$:GOTO940       :rem 92
1000 :                                :rem 251
1010 DATA A,1403,5,7I,40,1247,2,3M,-39
                                      :rem 119
1020 DATA 1249,2,3O,1,1331,2,5Q,41,1491,2
     ,9Q,40,1569,2,11O,39,1651,7,13Q,41
                                      :rem 236
1030 DATA 1731,7,15Q,40,1770,7,16P,39,176
     7,7,16M,-1,1603,7,12I,-41,1759,7,16E
     ,39                              :rem 235
1040 DATA 1756,7,16B,-1,1715,7,15A,-41,16
     35,7,13A,-40,1557,7,11C,-39      :rem 69
1050 DATA 1475,2,9A,-41               :rem 115
1060 DATA 1315,2,5A,-40,1237,2,3C,-39,123
     9,2,3E,1,1403,2,7I,41,1683,5,14I,40
                                      :rem 216
1070 DATA 0,0,0,0                     :rem 38
```

```
1080 :                                  :rem 3
1090 DATA B,1722,5,15H,1,1562,4,11H,-40,1
     559,4,11E,-1,1519,4,10E,-40        :rem 75
1100 DATA 1441,4,8G,-39                 :rem 118
1110 DATA 1446,4,8L,1,1528,4,10N,41,1568,
     4,11N,40,1565,4,11K,-1,1725,4,15K,40
                                        :rem 53
1120 DATA 1730,5,15P,1,1722,5,15H,-1,0,0,
     0,0                                :rem 118
1130 :                                  :rem 255
1140 DATA C,1364,2,6J,39,1359,2,6E,-1,139
     8,2,7D,39,1598,2,12D,40,1680,2,14F,4
     1                                  :rem 111
1150 DATA 1681,2,14G,1,1641,2,13G,-40,160
     0,2,12F,-41,1560,2,11F,-40,1564,2,11
     J,1                                :rem 136
1160 DATA 1687,2,14M,41,1688,2,14N,1,1608
     ,2,12N,-40,1567,2,11M,-41,1407,2,7M,
     -40                                :rem 206
1170 DATA 1329,2,5O,-39,1331,0,5Q,1,1291,
     0,4Q,-40,1290,0,4P,-1,1208,2,2N,-41
                                        :rem 244
1180 DATA 0,0,0,0                       :rem 40
1190 :                                   :rem 5
1200 DATA D,1160,2,1F,-41,1157,2,1C,-1,12
     35,2,3A,39,1475,2,9A,40,1803,2,17I,4
     1                                  :rem 56
1210 DATA 1491,2,9Q,-39,1251,2,3Q,-40,116
     9,2,1O,-41,1166,2,1L,-1,1283,2,4I,39
                                        :rem 47
1220 DATA 0,0,0,0                       :rem 35
1230 :
1240 DATA Z,195,16,3, 31,21,1, 30,25,2, 1
     35,33,2                            :rem 169
1250 DATA 30,25,2, 31,21,2, 195,16,2, 31,
     21,2, 30,25,3, 31,21,1             :rem 233
1260 DATA 195,16,2                      :rem 115
1270 DATA 143,12,2, 195,16,1            :rem 36
1280 DATA 0,0,3, 195,16,1, -1,0,0:rem 205
```

VIC Super Expander Graphics

(Article on page 80.)

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.

```
40 B=250:L=250:PI=3.14159265            :rem 230
50 C=3*PI:INC=PI/50                      :rem 89
60 DIM X(151),Y(151)                    :rem 237
70 GRAPHIC2:COLOR6,6,7,7                :rem 174
100 FOR Z=2TO12                          :rem 71
110 FOR A=0 TO C STEP INC                :rem 35
120 R=B*COS(A*Z)+L                        :rem 0
130 S=S+1                               :rem 211
140 X(S)=.8*R*COS(A)+512                 :rem 20
150 Y(S)=R*SIN(A)+512                   :rem 139
160 POINT2,X(S),Y(S)                    :rem 164
170 NEXT                                :rem 215
200 FOR K=50 TO 15STEP-5                 :rem 13
210 FOR S=1 TO 100 STEP 5               :rem 224
220 DRAW2,X(S),Y(S)TO X(S+K) ,Y(S+K ):NEX
    T                                   :rem 114
230 FOR TM=1 TO 500:NEXT                :rem 60
240 SCNCLR:NEXT K:S=0                   :rem 25
250 NEXT Z:GOTO 100                     :rem 52
```

VIC-20 Character Developer

(Article on page 130.)

```
10 POKE 52,24: POKE 56,24: PRINT CHR$(147
   ) "ONE MOMENT PLEASE"                :rem 84
20 FORI=6144 TO 7679: POKE I, PEEK(I+2662
   4): NEXT                             :rem 99
30 POKE 36869,254: PRINT CHR$(147);
                                        :rem 89
40 PRINT: PRINT: FORI=1 TO 8: PRINT "
   {2 SPACES}--------": NEXT            :rem 87
100 PRINT CHR$(19); : PRINT: PRINT: FORI=
    1 TO 8                              :rem 86
110 M=8: N=0                            :rem 73
120 INPUT A$                           :rem 136
130 FORJ=0 TO 7: M=M-1                 :rem 123
140 IF MID$(A$,J+1,1)="-" THEN 160
                                       :rem 123
150 N=N+2↑M: N(I)=N                     :rem 36
160 NEXTJ                               :rem 32
170 M=4: MS=0                          :rem 157
180 FORJ=0 TO 3: M=M-1                 :rem 124
190 IF MID$(A$,J+1,1)="-" THEN 210
                                       :rem 124
200 MS=MS+2↑M                           :rem 23
210 NEXTJ                               :rem 28
220 IF MS<10 THEN X$=STR$(MS): MS$(I)=MID
    $(X$,2,1): GOTO 240                :rem 186
230 MS$(I)=CHR$(MS+55)                 :rem 183
240 M=4: LS=0                          :rem 154
250 FORJ=4 TO 7: M=M-1                 :rem 130
260 IF MID$(A$,J+1,1)="-" THEN 280
                                       :rem 129
270 LS=LS+2↑M                           :rem 28
280 NEXTJ                               :rem 35
290 IF LS<10 THEN X$=STR$(LS): LS$(I)=MID
    $(X$,2,1): GOTO 310                :rem 188
300 LS$(I)=CHR$(LS+55)                 :rem 179
310 NEXTI                               :rem 28
320 PRINT CHR$(19);: PRINT , " DEC
    {3 SPACES}HEX":PRINT                :rem 200
330 FORI=1 TO 8: X$ = STR$(N(I)): X=LEN(X
    $)                                 :rem 211
340 PRINT , N(I);: PRINT SPC(6-X) MS$(I)
    {SPACE}LS$(I): NEXT                 :rem 96
350 FORI=0 TO 7: POKE 6144+I, N(I+1): NEX
    T: PRINT: PRINT "{6 SPACES}@": PRINT
                                       :rem 125
360 PRINT: PRINT " PRINT-OUT (Y/N)?"
                                        :rem 73
370 A$= "": GETA$: IFA$= "" THEN 370
                                       :rem 119
380 IFA$= "Y" THEN 500                  :rem 45
400 FORI=1 TO 21: PRINT CHR$(157); : NEXT
    : PRINT "ARE YOU DONE (Y/N)?" :rem 14
410 A$= "": GETA$: IFA$="" THEN 410
                                       :rem 109
420 IF A$="N" THEN 440                  :rem 32
430 END                                :rem 110
440 FORI=1 TO 12: PRINT CHR$(157); : NEXT
    I: PRINT CHR$(145)                 :rem 162
450 PRINT "NEW CHARACTER (Y/N)?";:rem 130
460 A$= "": GETA$: IF A$= "" THEN 460
                                       :rem 119
```

```
470 IF A$="Y" THEN RUN 30          :rem 240
480 PRINT CHR$(19): FORI=1 TO 9: PRINT "
    {SPACE}" , "{10 SPACES}": NEXT
                                   :rem 141
490 FORI=1 TO 12: PRINT"{20 SPACES}": NEX
    T: RUN 100                     :rem 133
500 OPEN1,4: CMD1                  :rem 151
510 SC=PEEK(648)*256: FOR P=SC TO SC+330
                                   :rem 134
520 CH=PEEK(P): C$="": IF (P-SC)/22 = INT
    ((P-SC)/22) THEN PRINT CHR$(13);
                                   :rem 36
530 IF CH<32 THEN CH=CH+64: GOTO 550
                                   :rem 119
540 IF CH>95 THEN CH=32            :rem 187
550 C$=CHR$(CH): PRINT C$; : NEXT: PRINT:
     PRINT: PRINT#1: CLOSE1,4      :rem 231
560 GOTO 400                       :rem 104
```

How To Make Custom Characters On The 64

(Article on page 120.)

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:

Character Set Transfer To RAM

```
10 POKE 56334,0:REM TURN OFF INTERRUPTS
                                   :rem 83
20 POKE 1,51:REM TURN OFF VIDEO CHIP TO E
   XPOSE CHARACTER GENERATOR       :rem 220
30 FOR ADDRESS=2048 TO 6143        :rem 204
40 POKE ADDRESS,PEEK(ADDRESS+51200):REM C
   OPY CHARACTERS TO RAM           :rem 32
50 NEXT ADDRESS                    :rem 170
60 POKE 1,55:REM TURN ON VIDEO CHIP
                                   :rem 251
70 POKE 56334,129:REM TURN ON INTERRUPTS
                                   :rem 135
80 END                             :rem 63
```

Program 2: Chred 64

```
100 REM "CHRED 64"                 :rem 137
120 POKE53280,11:POKE53281,0:PRINT"{5}"
                                   :rem 189
130 V=53248:SC=1024:CB=2048:CC=SC+40*21+9
                                   :rem 222
140 SZ=7:FP=0:FO=0:TP=0:TY=0:SL=0:R$="0":
    C$="0"                         :rem 199
150 CO$="* {RIGHT}{LEFT}{DOWN}{UP}{HOME}
    {CLR}{F1}{F3}{F5}{F7}SLRQ"     :rem 85
160 DN$="{HOME}{17 DOWN}"          :rem 2
170 BL$="{18 SPACES}"              :rem 203
180 NU$="0123456789ABCDEF"         :rem 131
190 DEFFNA(F)=SC+62+40*CY+CX       :rem 86
200 DEFFNB(F)=CB+((FO+TP)*8)+TY    :rem 251
210 PRINT"{CLR}{5 DOWN}{9 SPACES}{YEL}CHA
    RACTER SET LOADING"            :rem 97
220 GOSUB1500                      :rem 218
225 POKE 49276,8:REM ENTER THIS LINE FOR
    {SPACE}DISK ONLY -- OMIT FOR CASSETTE
                                   :rem 94
230 SYS49152:POKEV+24,19           :rem 127
240 POKE 53281,1:PRINT"{CLR}":POKE 53281,
    0:GOSUB1060                    :rem 12
250 CY=0:CX=0                      :rem 226
260 POKEFNA(0),PEEK(FNA(0))OR128   :rem 81
270 POKE198,0                      :rem 198
280 GETCH$:IFCH$=""THEN280         :rem 235
290 FORCH=1TOLEN(CO$):IFMID$(CO$,CH,1)=CH
    $THEN310                       :rem 149
300 NEXTCH:GOTO270                 :rem 105
310 POKEFNA(0),PEEK(FNA(0))ANDNOT128
                                   :rem 112
320 ONCHGOTO330,370,410,430,450,470,490,5
    00,590,770,800,1040,910,900,840,1020
                                   :rem 119
330 POKEFNA(0),42                  :rem 173
340 GOSUB550                       :rem 177
350 POKEFNB(0),PEEK(FNB(0))OR2↑(ABS(TX-7)
    )                              :rem 208
360 GOTO260                        :rem 106
370 POKEFNA(0),32                  :rem 176
380 GOSUB550                       :rem 181
390 POKEFNB(0),PEEK(FNB(0))ANDNOT2↑(ABS(T
    X-7))                          :rem 247
400 GOTO260                        :rem 101
410 CX=CX+1:IFCX>SZTHENCX=0        :rem 234
420 GOTO260                        :rem 103
430 CX=CX-1:IFCX<0THENCX=SZ        :rem 236
440 GOTO260                        :rem 105
450 CY=CY+1:IFCY>SZTHENCY=0        :rem 242
460 GOTO260                        :rem 107
470 CY=CY-1:IFCY<0THENCY=SZ        :rem 244
480 GOTO260                        :rem 109
490 GOTO250                        :rem 109
500 FORCY=0TOSZ:FORCX=0TOSZ:POKEFNA(0),32
                                   :rem 158
510 GOSUB550                       :rem 176
520 POKEFNB(0),0                   :rem 121
530 NEXTCX,CY                      :rem 58
540 GOTO250                        :rem 105
550 TP=FP:TX=CX:TY=CY:IFTX>7ANDTY<8THENTP
    =TP+1:TX=TX-8                  :rem 177
560 IFTX<8ANDTY>7THENTP=TP+2:TY=TY-8
                                   :rem 134
570 IFTY>7ANDTX>7THENTP=TP+3:TY=TY-8:TX=T
    X-8                            :rem 189
580 RETURN                         :rem 125
590 PRINTDN$"{RVS}{YEL}ROW, COLUMN?{OFF}
    {5} ";                         :rem 107
600 POKE198,0                      :rem 195
610 GETR$:IFR$=""THEN610           :rem 115
620 IFVAL(R$)<>0ANDVAL(R$)<40RR$="0"THENR
    =VAL(R$):GOTO640               :rem 165
630 GOTO600                        :rem 104
640 PRINTR$",";                    :rem 72
650 POKE198,0                      :rem 200
660 GETC$:IFC$=""THEN660           :rem 95
670 IFC$=CHR$(20)THENPRINT"{2 LEFT}
    {2 SPACES}";:GOTO590           :rem 10
680 IFASC(C$)>64THENC=ASC(C$)-55:IFC>15TH
    EN600                          :rem 174
690 IFVAL(C$)<>0ORC$="0"THENC=VAL(C$)
                                   :rem 118
700 IFC>15THEN650                  :rem 215
710 PRINTC$                        :rem 140
720 FP=R*16+C                      :rem 189
730 IFSZ=15ANDFP>60THENFP=60:C$="C":C=12
                                   :rem 112
740 GOSUB1290                      :rem 231
750 PRINTDN$;BL$                   :rem 204
760 GOTO250                        :rem 109
```

```
770 IFFO<191THENFO=FO+64:GOTO790 :rem 215
780 FO=0                         :rem 161
790 FP=0:R$="0":C$="0":GOSUB1240:GOTO250
                                 :rem 225
800 IFSZ=15THENSZ=7:GOTO830      :rem 213
810 IFFP>60THENFP=60:C$="C":C=12  :rem 76
820 SZ=15                        :rem 234
830 POKE 53281,1:PRINT"{CLR}":POKE 53281,
    0:CX=0:CY=0:GOSUB1060:GOTO250:rem 160
840 PRINTDN$;"{RVS}{YEL}ARE YOU SURE? ";
                                 :rem 156
850 POKE198,0                    :rem 202
860 GETCH$:IFCH$="N"THENPRINTDN$;"{OFF}
    {5}";BL$:GOTO250            :rem 134
870 IFCH$<>"Y"THEN860            :rem 193
880 PRINT"YES{OFF}{5}"          :rem 140
890 SYS49152:GOSUB1310:PRINTDN$;BL$:GOTO2
    50                           :rem 152
900 SL=1                         :rem 166
910 PRINTDN$;:INPUT"{RVS}{YEL}FILE NAME";
    NA$                           :rem 5
920 POKE253,LEN(NA$)             :rem 115
930 IFLEN(NA$)=0THEN970           :rem 74
940 FORL=1TOLEN(NA$)             :rem 196
950 POKE49359+L,ASC(MID$(NA$,L,1))
                                 :rem 125
960 NEXT                         :rem 222
970 SYS49269                     :rem 173
980 PRINTDN$;BL$;DN$;"{6 UP}"    :rem 167
990 IFSL=0THENSYS49292:GOTO1010  :rem 170
1000 SYS49310                    :rem 193
1010 POKE 53281,1:PRINT"{CLR}{5}":POKE
    {SPACE}53281,0:GOSUB1060:SL=0:GOTO25
    0                            :rem 32
1020 POKEV+24,21                 :rem 104
1030 PRINT"{CLR}{3 DOWN}":END    :rem 108
1040 IFPEEK(V+24)=19 THEN POKEV+24,21:GOT
    O260                         :rem 55
1050 POKEV+24,19:GOTO260         :rem 125
1060 PRINT"{HOME}{RVS}{YEL}CHARACTER EDIT
    OR{OFF}"                     :rem 65
1070 PRINT"{DOWN}{RVS}{YEL}F1{OFF}{5} E
    DIT NEW CHAR."               :rem 87
1080 PRINT"{RVS}{YEL}F3{OFF}{5} NEXT CH
    AR. BLOCK"                   :rem 227
1090 PRINT"{RVS}{YEL}F5{OFF}{5} BLOCK S
    IZE"                         :rem 150
1100 PRINT"{RVS}{YEL}F7{OFF}{5} FLIP CH
    ARACTER SET"                 :rem 142
1110 PRINT"{RVS}{YEL} R{OFF}{5} RESTORE
     FONT"                       :rem 27
1120 PRINT"{RVS}{YEL} S{OFF}{5} SAVE CH
    AR. SET"                     :rem 41
1130 PRINT"{RVS}{YEL} L{OFF}{5} LOAD CH
    AR. SET"                     :rem 20
1140 PRINT"{RVS}{YEL} Q{OFF}{5} QUIT"
                                  :rem 5
1150 PRINT"{HOME}{19 DOWN}{RVS}"TAB(21);"
     ";NU$;" {OFF}"              :rem 29
1160 FORL=1TO4:PRINTTAB(21)"{RVS}"MID$(NU
    $,L,1);SPC(16);" ":NEXT      :rem 164
1170 PRINTTAB(21)"{RVS}{18 SPACES}{OFF}
    {2 UP}"                      :rem 235
1180 PRINT"{HOME}"TAB(21);       :rem 116
1190 PRINT"{RVS} ";MID$(NU$,1,SZ+1);:PRIN
    T" {OFF}"                    :rem 105
1200 FORL=1TOSZ+1                 :rem 16
1210 PRINTTAB(21)"{RVS}"MID$(NU$,L,1);SPC
    (SZ+1) ;" {OFF}"             :rem 169
1220 NEXTL                       :rem 80
1230 PRINTTAB(21)"{RVS}";:FORL=0TOSZ+2:PR
    INT" ";:NEXT:PRINT"{OFF}"    :rem 82
1240 CH=FO                       :rem 36
1250 FORY=1TO4                   :rem 77
1260 FORX=1TO16                  :rem 128
1270 POKESC+781+X+Y*40,CH:CH=CH+1:rem 143
1280 NEXTX,Y                     :rem 231
1290 PRINT"{HOME}{19 DOWN}{5 SPACES}{RVS}
    EDITING ";R$","C$"{OFF}":POKECC,FP+F
    O                            :rem 216
1300 IFSZ=15THENPOKECC+1,FP+FO+1:POKECC+4
    0,FP+FO+2:POKECC+41,FP+FO+3 :rem 125
1310 X=0:Y=0:CX=0:CY=0           :rem 15
1320 GOSUB1390                   :rem 19
1330 IFSZ<>15THEN1380            :rem 222
1340 X=8:Y=0:FP=FP+1:GOSUB1390    :rem 27
1350 X=0:Y=8:FP=FP+1:GOSUB1390    :rem 28
1360 X=8:Y=8:FP=FP+1:GOSUB1390    :rem 37
1370 FP=FP-3                     :rem 148
1380 RETURN                      :rem 172
1390 TP=FP:TX=CX:TY=CY:IFTX>7ANDTY<8THENT
    P=TP+1:TX=TX-8               :rem 228
1400 IFTX<8ANDTY>7THENTP=TP+2:TY=TY-8
                                 :rem 176
1410 IFTY>7ANDTX>7THENTP=TP+3:TY=TY-8:TX=
    TX-8                         :rem 231
1420 TE=8*(FO+TP)+CB:REM CHAR. POINTER
                                 :rem 239
1430 POKE251,TE-INT(TE/256)*256  :rem 233
1440 POKE252,INT(TE/256)          :rem 94
1450 TE=FNA(0)+X+40*Y:REM SCREEN LOC.
                                 :rem 117
1460 POKE253,TE-INT(TE/256)*256  :rem 238
1470 POKE254,INT(TE/256)          :rem 99
1480 SYS49209                    :rem 212
1490 RETURN                      :rem 174
1500 FORL=49152TO49319           :rem 232
1510 READD:POKEL,D:NEXT          :rem 197
1520 RETURN                      :rem 168
1530 REM FONT COPIER ROUTINE     :rem 204
1540 DATA120,169,51,133,1,169,1,141,13,22
    0,169,0,133,251,133,253,169,208,133
                                 :rem 189
1550 DATA252,169,8,133,254,160,0,177,251,
    145,253,230,251,230,253,208,246,230
                                 :rem 205
1560 DATA252,230,254,165,252,201,225,208,
    236,169,129,141,13,220,169,55,133,1
                                 :rem 205
1570 DATA88,96                   :rem 242
1580 REM CHAR EXPAND AND DISPLAY :rem 121
1590 DATA160,0,162,0,169,128,133,250,177,
    251,37,250,208,4,169,32,208,2,169,42
                                  :rem 3
1600 DATA145,253,24,102,250,240,8,230,253
    ,208,2,230,254,208,229,230,251,208,2
                                 :rem 230
1610 DATA230,252,165,253,24,105,33,133,25
    3,165,254,105,0,133,254,232,224,8,20
    8                            :rem 33
1620 DATA201,96                  :rem 17
1630 REM SAVE AND LOAD ROUTINES  :rem 73
1640 DATA169,128,133,157,169,1,162,1,160,
    1,32,186,255,165,253,162,208,160,192
                                 :rem 11
1650 DATA32,189,255,96,169,0,133,251,169,
    8,133,252,169,251,162,16,160,25
                                 :rem 33
1660 DATA32,216,255,96           :rem 116
1670 DATA169,0,162,0,160,8,32,213,255,96
                                 :rem 226
```

Binary Castle

(Article on page 38.)

BEFORE TYPING...

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

Program 1: VIC Version

```
10 DIMBI$(15):D=21:D$="{20 DOWN}":PRINTCH
   R$(147)CHR$(152)                  :rem 213
20 DATA"L£2 @3@","LLL£+3","LL£+3@",
   "LL£2 +3","L£+3@@","L£+3L£+3",
   "L£2 +3@","L£3 +3","£+3@@@","
   £+3@@£+3","£+3L£+3@"   :rem 148
30 DATA"£+3@£2 +3","£2 +3@@","
   £2 +3L£+3","£3 +3@","£4 +3"
                                     :rem 239
35 POKE53281,1:POKE53280,1           :rem 191
40 FORI=0TO15:READBI$(I):NEXT:POKE53281,1
   :POKE53280,1                      :rem 179
50 PRINT"{HOME}"LEFT$(D$,D):RAN=INT(15*RN
   D(0)):PRINTSPC(8);BI$(RAN):INPUT"
   {HOME} WHAT NUMBER";N             :rem 67
60 IFN<>RANTHENPRINTLEFT$(D$,D)"SORRY,":P
   RINT"WRONG NUMBER":FORT=1TO1000:NEXTT
                                     :rem 86
70 IFN<>RANTHENPRINTD$:D=21:GOSUB200:PRIN
   TCHR$(147):GOTO50                 :rem 186
80 D=D-1:IFD<=1THENPRINT"{HOME} YOU DID I
   T!!!":GOSUB 200:D=21:PRINT CHR$(147)
                                     :rem 66
90 PRINT"{HOME}{16 SPACES}":GOTO50:rem 37
100 END                              :rem 104
200 REM BELLS & WHISTLES             :rem 129
205 FOR J=30 TO 60 STEP 10           :rem 14
210 POKE54296,15:POKE54277,5:POKE54278,5:
    FORI=0TO15:POKE53280,I:POKE53281,15-I
                                     :rem 57
220 POKE 54272,0:POKE54273,J+I:POKE54276,
    16+GT:GT=1-GT:NEXT:NEXT          :rem 66
230 POKE53281,1:POKE53280,1:POKE54276,16:
    POKE54296,0:RETURN               :rem 217
```

Program 2: 64 Version

```
10 DIMBI$(15):D=21:D$="{20 DOWN}":PRINTCH
   R$(147)                           :rem 235
20 DATA"L£2 @3@","LLL£+3","LL£+3@",
   "LL£2 +3","L£+3@@","L£+3L£+3",
   "L£2 +3@","L£3 +3","£+3@@@","
   £+3@@£+3","£+3L£+3@"   :rem 148
30 DATA"£+3@£2 +3","£2 +3@@","
   £2 +3L£+3","£3 +3@","£4 +3"
                                     :rem 239
40 FORI=0TO15:READBI$(I):NEXT        :rem 34
50 PRINT"{HOME}"LEFT$(D$,D):RAN=INT(15*RN
   D(0)):PRINTSPC(8);BI$(RAN):INPUT"
   {HOME} WHAT NUMBER";N             :rem 67
60 IFN<>RANTHENPRINTLEFT$(D$,D)"SORRY,":P
   RINT"WRONG NUMBER":FORT=1TO1000:NEXTT
                                     :rem 86
70 IFN<>RANTHENPRINTD$:D=21:GOSUB200:PRIN
   TCHR$(147):GOTO50                 :rem 186
80 D=D-1:IFD<=1THENPRINT"{HOME} YOU DID I
   T!!!":GOSUB 200:D=21:PRINTCHR$(147)
                                     :rem 66
90 PRINT"{HOME}{16 SPACES}":GOTO50:rem 37
100 END                              :rem 104
200 REM BELLS & WHISTLES             :rem 129
210 FORI=8TO24:POKE36879,I:POKE36878,10:P
    OKE36876,I+128:FORT=1TO200:NEXT:NEXT
                                     :rem 242
220 POKE36878,0:RETURN               :rem 73
```

Understanding Sound On The 64

(Article on page 136.)

```
100 I=52992                          :rem 34
110 READ A:IF A=256 THEN 190         :rem 156
120 POKE I,A:I=I+1:GOTO 110          :rem 226
130 DATA 24,5,6,0,1,2,3              :rem 77
140 DATA 21,12,13,7,8,9,10           :rem 234
150 DATA 11,19,20,14,15,16,17        :rem 128
160 DATA 23,4,11,18,162,0,188        :rem 135
170 DATA 0,207,185,0,192,153,0       :rem 182
180 DATA 212,232,224,25,208,242,96,256
                                     :rem 80
190 POKE53281,1:POKE53280,1          :rem 241
200 POKE650,128                      :rem 35
210 F$="{19 SPACES}"                 :rem 126
220 S=49152:D=0:Q=54272:P=53017:M$="VOICE
    ":Z$="{4 SPACES}{4 LEFT}":KE=197
                                     :rem 158
230 FORT=STOS+30:POKET,0:NEXT:SYSP
                                     :rem 255
240 PRINT"{CLR}";:FI$=" NONE "       :rem 211
250 FORA=1TO11:ON A GOSUB500,510,520,530,
    540,550,560,570,590,600,610:NEXT
                                     :rem 138
270 GETE$:U=PEEK(KE):IFU=64ANDPEEK(S+4)TH
    ENPOKES+4,PEEK(S+4)AND254:SYSP
                                     :rem 207
280 IFU=64ANDPEEK(S+7+4)THENPOKES+7+4,PEE
    K(S+7+4)AND254:SYSP              :rem 161
290 IFU=64ANDPEEK(S+14+4)THENPOKES+14+4,P
    EEK(S+14+4)AND254:SYSP           :rem 44
300 IFU=62THENSYSP:GOTO1330          :rem 212
310 IFE$="1"ORE$="2"ORE$="3"THEND=(ASC(E$
    )-49)*7:PRINT"{CLR}";TAB(25);M$;E$:GO
    TO250                            :rem 8
320 IFD>7THENPOKES+24,(PEEK(S+24)AND127):
    SYSP                             :rem 236
330 IFU=4THENPOKES+4+D,33:SYSP       :rem 133
340 IFU=5THENPOKES+4+D,17:SYSP       :rem 137
350 IFU=6THENPOKES+4+D,129:SYSP      :rem 191
360 IFU=3THENPOKES+4+D,65:SYSP       :rem 140
370 IF U=39THENPOKES+24,(PEEK(S+24)AND255
    ):FI$=" NONE{6 SPACES}":POKES+23,0:SY
    SP                               :rem 126
380 IF U=60 THENFORT=0TO14STEP7:POKES+4+T
    ,PEEK(S+4+T)OR1:NEXT:SYSP        :rem 181
390 IFU=57THENPOKES+4+D,PEEK(S+4+D)OR3:SY
    SP                               :rem 200
400 IFU=54THENPOKES+4+D,21:SYSP      :rem 181
410 V=2↑(D/7)                        :rem 179
420 IFU=42THENFI$=" LOWPASS ":POKES+23,V:
    POKES+24,(PEEK(S+24)OR16):SYSP:rem 44
430 IFU=29THENFI$=" HIGHPASS ":POKES+23,V
    :POKES+24,(PEEK(S+24)OR64):SYSP
                                     :rem 99
440 IFU=28THENFI$=" BANDPASS ":POKES+23,V
    :POKES+24,(PEEK(S+24)OR32):SYSP
                                     :rem 83
```

```
450 N$="ADSROYTVFPW":FORJ=1TO LEN(N$):G$=
    MID$(N$,J):IF LEFT$(G$,1)=E$THEN480
                                    :rem 206
460 NEXT                            :rem 217
470 GOTO270                         :rem 109
480 ONLEN(G$)GOSUB610,600,590,570,560,550
    ,540,530,520,510,500            :rem 155
490 GOTO270                         :rem 111
500 PRINT"{BLK}{HOME}{RVS}A{OFF}TTACK
    {2 SPACES}RATE +-":GOSUB620:RETURN
                                    :rem 85
510 PRINT"{BLU}{HOME}{2 DOWN}{RVS}D{OFF}E
    CAY{2 SPACES}RATE +-":GOSUB700:RETURN
                                    :rem 180
520 PRINT"{RED}{HOME}{4 DOWN}{RVS}S{OFF}U
    STAIN LEVEL +-":GOSUB770:RETURN
                                    :rem 232
530 PRINT"{GRN}{HOME}{6 DOWN}{RVS}R{OFF}E
    LEASE RATE{2 SPACES}+-":GOSUB840:RETU
    RN                              :rem 153
540 PRINT"{13}{HOME}{8 DOWN}{RVS}O{OFF}V
    ERALL VOLUME +-":GOSUB910:RETURN
                                    :rem 221
550 PRINT"{2}{HOME}{10 DOWN}PITCH (HIGH
     B{RVS}Y{OFF}TE)+-":GOSUB970:RETURN
                                    :rem 74
560 PRINT"{PUR}{HOME}{12 DOWN}PI{RVS}T
    {OFF}CH (LOW BYTE)+-":GOSUB1030:RETUR
    N                               :rem 106
570 IFD>0THENPRINT"{HOME}{14 DOWN}NO RING
    /SYNC FOR VOICES TWO AND THREE":RETUR
    N                               :rem 38
580 PRINT"{7}{HOME}{14 DOWN}PITCH {RVS}
    V{OFF}OICE 3 (FOR RING)+-":GOSUB1090:
    RETURN                          :rem 44
590 PRINT"{4}{HOME}{16 DOWN}{RVS}F{OFF}
    ILTERS{2 SPACES}CUTOFF{2 SPACES}+-":G
    OSUB1150:RETURN                 :rem 160
600 PRINT"{3}{HOME}{18 DOWN}{RVS}P{OFF}
    ULSE WAVE HIGH{2 SPACES}+-":GOSUB1210
    :RETURN                         :rem 178
610 PRINT"{2}{HOME}{20 DOWN}PULSE {RVS}
    W{OFF}AVE LOW{3 SPACES}+-":GOSUB1270:
    RETURN                          :rem 172
620 POKE198,0:GETA$:IF A$<>""THEN620
                                    :rem 247
630 IF PEEK(KE)<>40ANDPEEK(KE)<>43ANDPEEK
    (KE)<>1THEN680                  :rem 23
640 IFPEEK(KE)=40ANDX1<15THENX1=X1+1
                                    :rem 168
650 IFPEEK(KE)=43ANDX1>0THENX1=X1-1
                                    :rem 122
660 IFPEEK(KE)=1THENPOKE197,0:POKE198,0:F
    ORT=1TO500:NEXT:POKE198,0:PRINT:RETUR
    N                               :rem 61
670 PRINT"{RVS}";LEFT$(F$,X1);"{OFF}";RIG
    HT$(F$,15-X1);Z$;(PEEK(S+D+5)AND240);
    "{2 UP}"                        :rem 82
680 POKES+D+5,(X1*16)+(PEEK(S+D+5)AND15):
    POKEQ+D+5,(PEEK(S+D+5))         :rem 113
690 GOTO630                         :rem 113
700 POKE198,0:IF PEEK(KE)<>40ANDPEEK(KE)<
    >43ANDPEEK(KE)<>1THEN750        :rem 122
710 IFPEEK(KE)=40ANDX2<15THENX2=X2+1
                                    :rem 169
720 IFPEEK(KE)=43ANDX2>0THENX2=X2-1
                                    :rem 123
730 IFPEEK(KE)=1THENPOKE197,0:POKE198,0:F
    ORT=1TO500:NEXT:POKE198,0:PRINT:RETUR
    N                               :rem 59
740 PRINT"{RVS}";LEFT$(F$,X2);"{OFF}";RIG
    HT$(F$,15-X2);Z$;(PEEK(S+D+5)AND15);"
    {UP}"                           :rem 145
750 POKES+D+5,X2+(PEEK(S+D+5)AND240):POKE
    Q+D+5,PEEK(S+D+5)               :rem 109
760 GOTO700                         :rem 109
770 POKE198,0:IF PEEK(KE)<>40ANDPEEK(KE)<
    >43ANDPEEK(KE)<>1THEN820        :rem 127
780 IFPEEK(KE)=40ANDX3<15THENX3=X3+1
                                    :rem 179
790 IFPEEK(KE)=43ANDX3>0THENX3=X3-1
                                    :rem 133
800 IFPEEK(KE)=1THENPOKE197,0:POKE198,0:F
    ORT=1TO500:NEXT:POKE198,0:PRINT:RETUR
    N                               :rem 57
810 PRINT"{RVS}";LEFT$(F$,X3);"{OFF}";RIG
    HT$(F$,15-X3);Z$;(PEEK(S+D+6)AND240);
    "{UP}"                          :rem 194
820 POKES+D+6,(X3*16)+(PEEK(S+D+6)AND15):
    POKEQ+D+6,PEEK(S+D+6)           :rem 34
830 GOTO770                         :rem 114
840 POKE198,0:IF PEEK(KE)<>40ANDPEEK(KE)<
    >43ANDPEEK(KE)<>1THEN890        :rem 132
850 IFPEEK(KE)=40ANDX4<15THENX4=X4+1
                                    :rem 180
860 IFPEEK(KE)=43ANDX4>0THENX4=X4-1
                                    :rem 134
870 IFPEEK(KE)=1THENPOKE197,0:POKE198,0:F
    ORT=1TO500:NEXT:POKE198,0:PRINT:RETUR
    N                               :rem 64
880 PRINT"{RVS}";LEFT$(F$,X4);"{OFF}";RIG
    HT$(F$,15-X4);Z$;(PEEK(S+D+6)AND15);"
    {UP}"                           :rem 155
890 POKES+D+6,X4+(PEEK(S+D+6)AND240):POKE
    Q+D+6,PEEK(S+D+6)               :rem 120
900 GOTO840                         :rem 110

910 POKE198,0:IF PEEK(KE)<>40ANDPEEK(KE)<
    >43ANDPEEK(KE)<>1THEN960        :rem 128
920 IFPEEK(KE)=40ANDX5<15THENX5=X5+1
                                    :rem 181
930 IFPEEK(KE)=43ANDX5>0THENX5=X5-1
                                    :rem 135
940 IFPEEK(KE)=1THENPOKE197,0:POKE198,0:F
    ORT=1TO500:NEXT:POKE198,0:PRINT:RETUR
    N                               :rem 62
950 PRINT"{RVS}";LEFT$(F$,X5);"{OFF}";RIG
    HT$(F$,15-X5);Z$;(PEEK(S+24)AND15);"
    {UP}"                           :rem 92
960 POKES+24,(X5+(PEEK(S+24)AND240)):SYSP
    :GOTO910                        :rem 145
970 POKE198,0:IF PEEK(KE)<>40ANDPEEK(KE)<
    >43ANDPEEK(KE)<>1THEN1020       :rem 170
980 IFPEEK(KE)=40ANDX6<15THENX6=X6+1
                                    :rem 190
990 IFPEEK(KE)=43ANDX6>0THENX6=X6-1
                                    :rem 144
1000 IFPEEK(KE)=1THENPOKE197,0:POKE198,0:
     FORT=1TO500:NEXT:POKE198,0:PRINT:RET
     URN                            :rem 98
1010 PRINT"{RVS}";LEFT$(F$,X6);"{OFF}";RI
     GHT$(F$,15-X6);Z$;PEEK(S+D+1);"{UP}"
                                    :rem 50
1020 POKES+1+D,16*X6:POKEQ+1+D,PEEK(S+1+D
     ):GOTO970                      :rem 179
1030 POKE198,0:IF PEEK(KE)<>40ANDPEEK(KE)
     <>43ANDPEEK(KE)<>1THEN1080     :rem 212
1040 IFPEEK(KE)=40ANDX7<15THENX7=X7+1
                                    :rem 229
1050 IFPEEK(KE)=43ANDX7>0THENX7=X7-1
                                    :rem 183
1060 IFPEEK(KE)=1THENPOKE197,0:POKE198,0:
```

```
FORT=1TO500:NEXT:POKE198,0:PRINT:RET
URN                                 :rem 104
1070 PRINT"{RVS}";LEFT$(F$,X7);"{OFF}";RI
GHT$(F$,15-X7);Z$;PEEK(S+D);"{UP}"
                                    :rem 222
1080 POKES+D,16*X7:POKEQ+D,PEEK(S+D):GOTO
1030                                :rem 202
1090 POKE198,0:IF PEEK(KE)<>40ANDPEEK(KE)
<>43ANDPEEK(KE)<>1THEN1140  :rem 215
1100 IFPEEK(KE)=40ANDX8<15THENX8=X8+1
                                    :rem 229
1110 IFPEEK(KE)=43ANDX8>0THENX8=X8-1
                                    :rem 183
1120 IFPEEK(KE)=1THENPOKE197,0:POKE198,0:
FORT=1TO500:NEXT:POKE198,0:PRINT:RET
URN                                 :rem 101
1130 PRINT"{RVS}";LEFT$(F$,X8);"{OFF}";RI
GHT$(F$,15-X8);Z$;PEEK(S+15+D);"{UP}
"                                   :rem 110
1140 POKEQ+24,PEEK(S+24)OR128:POKES+15+D,
X8*16:POKEQ+15+D,X8*16:GOTO1090
                                    :rem 230
1150 POKE198,0:IF PEEK(KE)<>40ANDPEEK(KE)
<>43ANDPEEK(KE)<>1THEN1200  :rem 209
1160 IFPEEK(KE)=40ANDX9<15THENX9=X9+1
                                    :rem 238
1170 IFPEEK(KE)=43ANDX9>0THENX9=X9-1
                                    :rem 192
1180 IFPEEK(KE)=1THENPOKE197,0:POKE198,0:
FORT=1TO500:NEXT:POKE198,0:PRINT:RET
URN                                 :rem 107
1190 PRINT"{RVS}";LEFT$(F$,X9);"{OFF}";RI
GHT$(F$,15-X9);Z$;PEEK(S+22);"
{6 RIGHT}";FI$;"{UP}"               :rem 32
1200 POKES+21,X9/2:POKES+22,(X9*16):POKEQ
+21,7:POKEQ+22,(X9*16):GOTO1150
                                    :rem 168
1210 POKE198,0:IF PEEK(KE)<>40ANDPEEK(KE)
<>43ANDPEEK(KE)<>1THEN1260  :rem 212
1220 IFPEEK(KE)=40ANDXA<15THENXA=XA+1
                                      :rem 3
1230 IFPEEK(KE)=43ANDXA>0THENXA=XA-1
                                    :rem 213
1240 IFPEEK(KE)=1THENPOKE197,0:POKE198,0:
FORT=1TO500:NEXT:POKE198,0:PRINT:RET
URN                                 :rem 104
1250 PRINT"{RVS}";LEFT$(F$,XA);"{OFF}";RI
GHT$(F$,15-XA);Z$;PEEK(S+D+2);"{UP}"
                                     :rem 79
1260 POKES+D+2,XA*16:POKEQ+D+2,PEEK(S+D+2
):GOTO1210                          :rem 235
1270 POKE198,0:IF PEEK(KE)<>40ANDPEEK(KE)
<>43ANDPEEK(KE)<>1THEN1320  :rem 215
1280 IFPEEK(KE)=40ANDXB<15THENXB=XB+1
                                     :rem 12
1290 IFPEEK(KE)=43ANDXB>0THENXB=XB-1
                                    :rem 222
1300 IFPEEK(KE)=1THENPOKE197,0:POKE198,0:
FORT=1TO500:NEXT:POKE198,0:PRINT:RET
URN                                 :rem 101
1310 PRINT"{RVS}";LEFT$(F$,XB);"{OFF}";RI
GHT$(F$,15-XB);Z$;PEEK(S+D+3);"{UP}"
                                     :rem 79
1320 POKES+D+3,XB*16:GOTO1270       :rem 169
1330 REM SAVE ROUTINE
1340 S=49152:CO=52992               :rem 113
1350 PRINT"{CLR}":DIMQ(45),ML(45):rem 203
1360 FORT=0TO44:Q(T)=PEEK(S+T):ML(T)=PEEK
(CO+T):NEXT                         :rem 231
1370 PRINT"1 RP=52992:FORR=RPTORP+44:READ
GP:POKER,GP:NEXT"                   :rem 197
1380 PG=0:FORA=0TO4:PG=PG+3         :rem 121
1390 PRINT PG"DATA";:FORT=0TO8 :PRINTML(T
+9*A);:IF T<8 THENPRINT"{LEFT},";
                                    :rem 235
1400 NEXT:PRINT:NEXT                :rem 68
1410 PRINT"20S=49152:FORT=STOS+24:POKET,0
:NEXT:P=53017{2 SPACES}"            :rem 115
1420 PRINT"30FORT=STOS+25:READDS:POKET,DS
:NEXT:SYSP{3 SPACES}"               :rem 189
1430 PO=30:FORW=0TO2:PO=PO+10        :rem 2
1440 PRINTPO"DATA";:FORT=0TO8:PRINTQ(T+9*
W);:IFT<8THENPRINT"{LEFT},";:rem 189
1450 NEXT:PRINT:NEXT                :rem 73
```

Machine Language For Beginners

(Article on page 164.)

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.

The Assembler

```
10 H=1:REM IF H = 0 THEN ASSEMBLY IS IN D
   ECIMAL                           :rem 42
50 HE$="0123456789ABCDEF":SZ=1:ZO$="000"
                                    :rem 166
100 PRINT"{3 SPACES}SIMPLE{3 SPACES}ASSEM
    BLER{2 SPACES}CONVENTIONS:"     :rem 90
110 DIMM$(56),TY(56),OP(56)        :rem 181
120 FORI=1TO56:READM$(I)           :rem 160
122 ROP$=MID$(M$(I),4,1):TY(I)=VAL(ROP$)
                                     :rem 5
124 OP$=RIGHT$(M$(I),3):OP(I)=VAL(OP$)
                                    :rem 155
126 M$(I)=LEFT$(M$(I),3)           :rem 235
140 NEXTI: PRINT                   :rem 228
150 PRINT"IMMEDIATE{5 SPACES}LDA #15
                                    :rem 46
155 PRINT"ABSOLUTE{6 SPACES}LDA 1500
                                    :rem 64
160 PRINT"ZERO PAGE{5 SPACES}LDA 15
                                    :rem 218
165 PRINT"ACCUMULATOR{3 SPACES}ASL
                                    :rem 107
170 PRINT"INDIRECT X{4 SPACES}LDA (15X)
                                    :rem 209
175 PRINT"INDIRECT Y{4 SPACES}LDA (15)Y
                                    :rem 216
177 PRINT"ZERO PAGE X{3 SPACES}LDA 15X
                                    :rem 146
179 PRINT"ZERO PAGE Y{3 SPACES}LDX 15Y
                                    :rem 173
180 PRINT"ABSOLUTE X{4 SPACES}LDA 1500X
                                    :rem 238
185 PRINT"ABSOLUTE Y{4 SPACES}LDA 1500Y
                                    :rem 245
189 PRINT:PRINT"{4 SPACES}ENTER ALL NUMBE
    RS IN ";                        :rem 127
190 IFH=1 THENPRINT"HEX":GOTO200 :rem 201
195 PRINT"DECIMAL"                  :rem 95
```

```
200 PRINT:PRINT"PLEASE INPUT STARTING ADD
    RESS FOR ML PROGRAM":INPUT SA$
210 IFH=1THENH$=SA$:GOSUB5000:SA=DE:GOTO2
    20                              :rem 130
215 SA=VAL(SA$)                     :rem 85
220 TA=SA:PRINT"{CLR}":REM CLEAR THE SCRE
    EN                              :rem 190
230 IFH=1THENDE=SA:SZ=3:GOSUB4000:PRINTH$
    ;:GOTO240                       :rem 175
235 PRINTSA" ";                     :rem 58
240 INPUTMN$:PRINT"{UP}"SPC(20);:REM GO U
    P ONE LINE AND OVER 20 SPACES:rem 232
241 REM ADD NEW PSEUDO-OPS HERE     :rem 65
242 IFRIGHT$(MN$,7)="FORWARD"THENFB=SA
                                    :rem 90
243 IFRIGHT$(MN$,7)="RESOLVE"THENFR=SA-FB
    :POKEFB+1,FR-2:PRINT"{2 SPACES}OK":GO
    TO230                           :rem 72
244 IFRIGHT$(MN$,4)="POKE"THENPRINT"ADDR,
    NUMBER(DEC)";:INPUTADR,NUM:POKEADR,NU
    M:GOTO230                       :rem 116
250 IFMN$="END"THENPRINT:PRINT"{6 SPACES}
    PROGRAM IS FROM"TA"TO"SA:END    :rem 13
260 L=LEN(MN$):L$=LEFT$(MN$,3)      :rem 181
270 FORI=1TO56:IFL$=M$(I)THEN300    :rem 136
280 NEXTI                           :rem 34
290 GOTO850                         :rem 113
300 REM PRIMARY OPCODE CATEGORIES   :rem 59
301 TY=TY(I):OP=OP(I)               :rem 20
305 IFFB=SATHENTN=0:GOTO2010        :rem 244
310 IFTY=0THENGOTO1000              :rem 102
320 IFTY=3THENTY=1:IFL=3THENOP=OP+8:GOTO1
    000                             :rem 81
330 R$=RIGHT$(MN$,L-4):IFH=1THENGOSUB6000
                                    :rem 200
340 LR$=LEFT$(R$,1):LL=LEN(R$):IFLR$="#"T
    HEN480                          :rem 184
350 IFLR$="("THEN520                :rem 88
360 IFTY=8THEN600                   :rem 15
370 IFTY=3THENOP=OP+8:GOTO1000      :rem 135
380 IFRIGHT$(R$,1)="X"ORRIGHT$(R$,1)="Y"T
    HEN630                          :rem 210
390 IFLEFT$(L$,1)="J"THEN820        :rem 44
400 TN=VAL(R$):IFTN>255THEN430      :rem 40
410 IFTY=1ORTY=3ORTY=4ORTY=5THENOP=OP+4
                                    :rem 133
420 GOTO2000                        :rem 145
430 H%=TN/256:L%=TN-256*H%:IFTY=2ORTY=7TH
    ENOP=OP+8:GOTO470               :rem 92
440 IFTY=1ORTY=3ORTY=4ORTY=5THENOP=OP+12:
    GOTO470                         :rem 197
450 IFTY=6ORTY=9THEN470             :rem 214
460 GOTO850                         :rem 112
470 GOTO3000                        :rem 151
480 TN=VAL(RIGHT$(R$,LL-1))         :rem 58
490 IFTY=1THENOP=OP+8:GOTO2000      :rem 137
500 IFTY=4ORTY=5THENGOTO2000        :rem 44
510 GOTO850                         :rem 108
520 IFRIGHT$(R$,2)=")Y"THEN540      :rem 184
530 IFRIGHT$(R$,2)="X)"THEN570      :rem 187
540 TN=VAL(MID$(R$,2,LL-3))         :rem 243
550 IFTY=1THENOP=OP+16:GOTO2000     :rem 181
560 GOTO850                         :rem 113
570 TN=VAL(MID$(R$,2,LL-3))         :rem 246
580 IFTY=1THENGOTO2000              :rem 113
590 GOTO850                         :rem 116
600 TN=VAL(R$):TN=TN-SA-2:IFTN<-128ORTN>1
    27THENPRINT"TOO FAR ";:GOTO850
                                    :rem 154
610 IFTN<0THENTN=TN+256             :rem 172
620 GOTO2000                        :rem 147
630 IFRIGHT$(R$,2)=")Y"THEN540      :rem 186
640 IFRIGHT$(R$,1)="X"THEN720       :rem 144
650 REM *ZERO Y                     :rem 66
660 TN=VAL(LEFT$(R$,LL-1)):IFTN>255THEN68
    0                               :rem 249
670 IFTY=2ORTY=5THEN730             :rem 209
675 IFTY=1THEN760                   :rem 24
680 GOSUB770:IFTY=1THENOP=OP+24:GOTO710
                                    :rem 230
690 IFTY=5THENOP=OP+28:GOTO710      :rem 151
700 GOTO850                         :rem 109
710 GOTO3000                        :rem 148
720 TN=VAL(LEFT$(R$,LL-1)):IFTN>255THENGO
    SUB770:GOTO780                  :rem 136
730 IFTY=2THENOP=OP+16:GOTO760      :rem 145
740 IFTY=1ORTY=3ORTY=5THENOP=OP+20:GOTO76
    0                               :rem 10
750 GOTO850                         :rem 114
760 GOTO2000                        :rem 152
770 H%=TN/256:L%=TN-256*H%:RETURN:rem 187
780 IFTY=2THENOP=OP+24:GOTO810      :rem 145
790 IFTY=1ORTY=3ORTY=5THENOP=OP+28:GOTO81
    0                               :rem 19
800 GOTO850                         :rem 110
810 GOTO3000                        :rem 149
820 TN=VAL(R$)                      :rem 35
830 GOSUB770                        :rem 185
840 GOTO710                         :rem 109
850 PRINT"{RVS} ERROR ":GOTO230     :rem 18
1000 REM 1 BYTE INSTRUCTIONS        :rem 191
1010 POKESA,OP:SA=SA+1:IFH=1THEN 1030
                                    :rem 189
1020 PRINTOP:GOTO230                :rem 247
1030 DE = OP:GOSUB4000:PRINTH$:GOTO230
                                    :rem 226
2000 REM 2 BYTE INSTRUCTIONS        :rem 193
2005 IFTN>256THENPRINT" INCORRECT ARGUMEN
     T. (#5 IN HEX IS #05)":GOTO230
                                    :rem 94
2010 POKESA,OP:POKESA+1,TN:SA=SA+2:IFH=1T
     HEN2030                        :rem 231
2020 PRINTOP;TN:GOTO230             :rem 213
2030 DE = OP:GOSUB4000:PRINTH$" ";:rem 90
2040 DE = TN:GOSUB4000:PRINTH$:GOTO230
                                    :rem 231
3000 REM 3 BYTE INSTRUCTIONS        :rem 195
3010 POKESA,OP:POKESA+1,L%:POKESA+2,H%:SA
     =SA+3:IFH=1THEN3030            :rem 172
3020 PRINTOP;L%;H%:GOTO230          :rem 77
3030 DE = OP:GOSUB4000:PRINTH$" ";:rem 91
3040 DE = L%:GOSUB4000:PRINTH$" ";:rem 46
3050 DE = H%:GOSUB4000:PRINTH$:GOTO230
                                    :rem 180
4000 REM{2 SPACES}DECIMAL TO HEX (DE TO H
     $)                             :rem 8
4010 H$="":FORM=SZTO0STEP-1:N%=DE/(16↑M):
     DE=DE-N%*16↑M:H$=H$+MID$(HE$,N%+1,1)
                                    :rem 179
4020 NEXT:SZ=1:RETURN               :rem 116
5000 REM{2 SPACES}HEX TO DECIMAL (H$ TO D
     E)                             :rem 9
5010 D=0:Q=3:FORM=1TO4:FORW=0TO15:IFMID$(
     H$,M,1)=MID$(HE$,W+1,1)THEN5030
                                    :rem 221
5020 NEXTW                          :rem 93
5030 D1=W*(16↑(Q)):D=D+D1:Q=Q-1:NEXTM:DE=
     INT(D):RETURN                  :rem 41
6000 REM ACCEPT HEX OPCODE INPUT AND TRAN
     SLATE IT TO DECIMAL            :rem 57
6010 IFLEFT$(R$,1)="#"THENH$="00"+RIGHT$(
```

```
     R$,2):GOSUB5000:R$="#"+STR$(DE):RETU
     RN                                :rem 234
6020 LS=LEN(R$):AZ$=LEFT$(R$,1):ZA$=MID$(
     R$,LS,1):IFAZ$<>"("THEN6050 :rem 126
6030 IFZA$="Y"THENH$="00"+MID$(R$,2,2):GO
     SUB5000:R$="("+STR$(DE)+")Y":RETURN
                                       :rem 30
6040 IFZA$=")"THENH$="00"+MID$(R$,2,2):GO
     SUB5000:R$="("+STR$(DE)+"X)":RETURN
                                       :rem 238
6050 IFZA$="X"ORZA$="Y"THEN6070    :rem 40
6060 H$=LEFT$(ZO$,4-LS)+R$:GOSUB5000:R$=S
     TR$(DE):RETURN                    :rem 44
6070 IFLS=5THENH$=LEFT$(R$,4):GOTO6090
                                       :rem 253
6080 H$="00"+LEFT$(R$,2)           :rem 186
6090 GOSUB5000:R$=STR$(DE)+ZA$:RETURN
                                       :rem 252
20000 DATAADC1097,AND1033,ASL3002,BCC8114
      ,BCS8176,BEQ8240,BIT7036,BMI8048
                                       :rem 93
20010 DATABNE8208,BPL8016,BRK0000,BVC8080
      ,BVS8112,CLC0024,CLD0216,CLI0088
                                       :rem 114
20020 DATACLV0184,CMP1193,CPX4224,CPY4192
      ,DEC2198,DEX0202,DEY0136,EOR1065
                                       :rem 184
20030 DATAINC2230,INX0232,INY0200,JMP6076
      ,JSR9032,LDA1161,LDX5162,LDY5160
                                       :rem 200
20040 DATALSR3066,NOP0234,ORA1001,PHA0072
      ,PHP0008,PLA0104,PLP0040,ROL3034
                                       :rem 185
20050 DATAROR3098,RTI0064,RTS0096,SBC1225
      ,SEC0056,SED0248,SEI0120,STA1129
                                       :rem 216
20060 DATASTX2134,STY2132,TAX0170,TAY0168
      ,TSX0186,TXA0138,TXS0154,TYA0152
                                       :rem 79
50000 PRINTX:POKE5,X:GOTO530        :rem 7
```

One-Touch Commands For The 64

(Article on page 159.)

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 POKE56,208:POKE55,0:F=0:C=PEEK(55)-120
  :IFC<0THENC=C+256:F=-1
2 D=PEEK(56)+F:POKE55,C:POKE56,D
3 S=828:I=146:GOSUB100
10 DATA32,198,3,165,55,133,251,133,253,1
   65,56,133,252,133,254,169
15 DATA49,133,167,169,133,133,168,169,13
   ,32,210,255,169,70,32,210
20 DATA255,165,167,32,210,255,169,61,32,
   210,255,169,63,32,210,255
25 DATA169,32,32,210,255,32,207,255,72,1
   60,0,165,168,145,55,104
30 DATA32,198,3,201,13,240,14,201,95,208
   ,2,169,13,145,55,32
35 DATA207,255,76,124,3,230,167,165,167,
   41,1,208,10,24,165,168
40 DATA105,4,133,168,76,170,3,56,165,168
   ,233,3,133,168,165,167
45 DATA201,57,144,163,120,169,L0,141,20,
   3,169,H0,141,21,3,88
50 DATA169,0,133,167,32,68,166,76,116,16
   4,166,55,208,2,198,56
55 DATA198,55,96
56 S=PEEK(55)+256*PEEK(56):I=120:GOSUB10
   0
57 SYS(828)
58 END
60 DATA165,167,240,59,160,0,177,251,32,L
   99,H0,176,12,165,55,197
65 DATA251,208,21,165,56,197,252,208,15,
   169,0,133,167,165,253,133
70 DATA251,165,254,133,252,76,49,234,166
   ,198,177,251,157,119,2,230
75 DATA198,32,L111,H0,165,198,201,11,144
   ,204,230,167,76,49,234,165
80 DATA215,32,L99,H0,176,3,76,49,234,165
   ,8,41,1,208,247,160
85 DATA0,177,251,197,215,208,6,32,L111,H
   0,76,L6,H0,32,L111,H0
90 DATA76,L81,H0,201,133,144,6,201,141,1
   76,2,56,96,24,96,166
95 DATA251,208,2,198,252,198,251,96,0,0
100 F=0:FORD=STOS+I:READA$:IFASC(A$)<58T
    HENA=VAL(A$):GOTO115
105 IFASC(A$)=76THENA=VAL(RIGHT$(A$,LEN(
    A$)-1))+PEEK(55):IFA>255THENA=A-256:
    F=1
110 IFASC(A$)=72THENA=VAL(RIGHT$(A$,LEN(
    A$)-1))+PEEK(56)+F:F=0
115 POKED,A:NEXT:RETURN
```

The Beginner's Corner

(Article on page 150.)

Bake A Cake

```
1 DIMI$(24),B$(24,1),M(11),C$(11):Z=24
                                       :rem 126
2 POKE 53281,1                         :rem 193
3 FORN=0TOZ:READA$,B$(N,0):I$(N)=A$+" "+B
  $(N,0):NEXT                          :rem 118
5 PRINT"{CLR}{BLU}{5 DOWN}{5 RIGHT}BAKE A
   CAKE"                                :rem 6
7 PRINT"{2 DOWN}CHOOSE:":PRINT"{DOWN} I N
  EED TO KNOW":PRINT"{3 SPACES}WHAT CAN B
  E MADE."                             :rem 77
9 PRINT"{DOWN} 2 WANT TO SEE":PRINT"
  {3 SPACES}A CERTAIN RECIPE":PRINT"
  {DOWN} 3 END PROGRAM"                :rem 79
11 GETE$:IFE$="3"THEN200               :rem 82
13 IFE$="2"THEN61                      :rem 165
15 IFE$<>"1"THEN11                     :rem 222
17 PRINT"{CLR}{BLU}IN THE FOLLOWING LIST,
    PRESS":PRINT"{2 SPACES}'Y' IF YOU HAV
   E THE INGREDIENT"                   :rem 53
19 PRINT"{2 SPACES}'N' IF YOU DO NOT
   {DOWN}":PRINT"{2 SPACES}'S' TO START O
   VER.{2 DOWN}"                       :rem 97
```

```
21 Y=0:FORN=0TOZ:PRINTB$(N,0);"{2 SPACES}
   {RED}--{BLK}";:GOSUB150          :rem 58
23 GETE$:IFE$="S"THEN17             :rem 75
25 IFE$="N"THENPRINT"N{BLU}":GOTO31
                                    :rem 114
27 IFE$<>"Y"THEN23                  :rem 12
29 PRINT"Y{BLU}":Y=Y+1              :rem 57
31 B$(N,1)=E$:NEXTN:C=0:PRINT"{2 DOWN}YOU
    CAN MAKE:"                      :rem 113
33 IFB$(1,1)="N"ORB$(2,1)="N"ORB$(5,1)="N
   "THEN37                          :rem 17
35 IFY>7THEN47                      :rem 95
37 PRINT"NOTHING TODAY.":PRINT"YOU NEED M
   ORE SUPPLIES"                    :rem 205
39 PRINT"{2 DOWN}{GRN}PRESS RETURN{BLU}";
   :GOSUB150                        :rem 148
41 GETE$:IFE$=""THEN41              :rem 245
43 IF ASC(E$)<>13THEN41             :rem 249
45 GOTO5                            :rem 215
47 RESTORE:FORN=0TOZ*2+1:READE$:NEXT:READ
   A$                               :rem 19
49 FORN=0TOZ:READE$:IFE$=""ORE$="0"THEN53
                                    :rem 208
51 IFB$(N,1)="N"THENFORI=N+1TOZ:READE$:NE
   XTI:GOTO55                       :rem 199
53 NEXTN:PRINTA$;" CAKE":C=C+1      :rem 13
55 READA$:IFA$<>"Z"THEN49           :rem 205
57 IFC=0THEN37                      :rem 68
59 PRINT"{DOWN}GO AHEAD AND BAKE!":GOTO39
                                    :rem 31
61 PRINT"{CLR}{BLU}{DOWN}CHOOSE:{2 DOWN}"
   :PRINT"A BANANA CAKE":PRINT"B CHERRY C
   AKE":PRINT"C CHOCOLATE CAKE"     :rem 59
63 PRINT"D DEVIL'S FOOD CAKE":PRINT"E GOL
   D LAYER CAKE":PRINT "F OATMEAL CAKE"
                                    :rem 23
65 PRINT"G RED VELVET CAKE":PRINT"H SAUER
   KRAUT CAKE":PRINT"I SPICE CAKE"
                                    :rem 146
67 PRINT"J TWO-EGG CAKE":PRINT"K WACKY CA
   KE":PRINT"L WHITE CAKE":GOSUB150
                                    :rem 187
69 GETE$:IFE$=""THEN69               :rem 9
71 A=ASC(E$):IFA<65ORA>76THEN69     :rem 83
73 RESTORE:PRINT"{CLR}{BLU}";:FORN=0TO2*Z
   +1:READE$:NEXT                   :rem 79
75 IFA=65THEN79                     :rem 131
77 FORN=1TOA-65:READA$:FORI=0TOZ:READE$:N
   EXTI,N                           :rem 23
79 READA$:PRINTA$;" CAKE{DOWN}":I=0
                                    :rem 177
81 FORN=0TOZ:READE$:IFE$=""ORVAL(E$)=0THE
   N85                              :rem 193
83 M(I)=VAL(E$):C$(I)=I$(N):PRINTM(I);TAB
   (6);C$(I):I=I+1                  :rem 152
85 NEXTN                            :rem 250
87 PRINT"{2 DOWN}{RED}CONVERT RECIPE? (Y/
   N){BLU}"                         :rem 220
89 GETE$:IFE$="N"THEN39             :rem 86
91 IFE$<>"Y"THEN89                  :rem 25
93 PRINT"{DOWN}MULTIPLY BY WHAT":PRINT"NU
   MBER OR DECIMAL?":PRINT"{RED} --{BLU}"
   ;                                :rem 27
95 INPUTF:IFF<=0THENPRINT"SORRY, F>0":GOT
   O93                              :rem 90
97 F=INT(F*100)/100:PRINT"{CLR}{BLU}";F;"
   TIMES ORIGINAL{DOWN}":PRINTA$;" CAKE
   {DOWN}"                          :rem 101
99 FOR N=0 TO I-1:PRINT INT(F*M(N)*100)/1
   00;TAB(6);C$(N):NEXT:GOTO 87     :rem 109
101 DATAC.,SHORTENING,C.,FLOUR,C.,SUGAR,C
    .,BROWN SUGAR,TSP.,BAKING PDR,TSP.
                                    :rem 25
103 DATASALT,TSP.,SODA,C.,CHERRY JUICE,,C
    HERRIES,C.,BANANAS,C.,SAUERKRAUT,C.,M
    ILK                             :rem 188
105 DATAC.,BUTTERMILK,,EGGS,,EGG WHITES,T
    SP.,RED COLOR,0Z.,CHOCOLATE,TBSP.,COC
    OA,TSP.                         :rem 187
107 DATAVANILLA,TSP.,CINNAMON,TSP.,NUTMEG
    ,TSP.,VINEGAR,C.,SALAD OIL,C.,WATER,C
    .,OATMEAL                       :rem 137
109 DATABANANA,.67,2.5,1.67,,1,1,1,,,1,,,
    .67,2,,,,,,,,,,,CHERRY,.5,2.25,1.33,
    ,3,.5,,.25                      :rem 161
111 DATA16,,,.5,,,4,,,,,,,,,,CHOCOLATE,.
    67,2.5,1.75,,,.5,1,,,,,,2,,,2,,1,,,,
    ,1.25,0                         :rem 143
113 DATADEVIL'S FOOD,.67,2.25,2,,1,1,1,,,
    ,,1.25,,3,,1,3,,,,,,,,GOLD LAYER,.5,
    2.25,1.5,,3                     :rem 49
115 DATA1,,,,,,1.67,,2,,,,,1.5,,,,,,,OATM
    EAL,.5,1.5,1,1,,.5,1,,,,,,,2,,,,,1,.7
    5,.25,,,1.25                    :rem 76
117 DATA1,RED VELVET,.5,2.75,1.5,,,.5,1.5
    ,,,,,,1,2,,6,,2,1,,,1,,,,SAUERKRAUT,.
    67,2.25,1.25                    :rem 211
119 DATA0,1,.25,,,,,.67,,,3,,,,8,1,,,,,1.
    25,,SPICE,.75,2.25,1,,1,1,1,,,,,,1,3,
    ,,,,,1,,.5,,0                   :rem 37
121 DATA0,,TWO-EGG,.5,2.25,1.5,,2.5,1,,,,
    ,,1,,2,,,,,1,,,,,,,WACKY,,2.5,1.5,,,1
    ,1,,,,,,,,0                     :rem 142
123 DATA0,,6,1,,,1.5,.75,1.5,,WHITE,.75,2
    .25,1.5,,3,1,,,,,,1,,,5,,,,1.5,,,,,,,
    Z                               :rem 103
150 FORI=0TO9:GETE$:NEXTI:RETURN :rem 110
200 PRINT"{CLR}{BLU}":END           :rem 38
```

VIC/64 Program Lifesaver

(Article on page 132.)

UNNEW

```
10 I=525
20 READ A:IF A=256 THEN 40
30 POKE I,A:I=I+1:GOTO 20
40 POKE 43,525 AND 255:POKE 44,2:REM SET
   {SPACE}BOTTOM OF MEMORY
50 POKE 45,578 AND 255:POKE 46,2:REM SET
   {SPACE}TOP OF MEMORY
60 CLR : SAVE"0:UNNEW",8
70 REM FOR TAPE USE SAVE"UNNEW",1,1
525 DATA 160,3,200,177,43,208,251
532 DATA 200,200,152,160,0,145,43
539 DATA 165,44,200,145,43,133,60
546 DATA 160,0,132,59,162,0,200
553 DATA 208,2,230,60,177,59,208
560 DATA 245,232,224,3,208,242,200
567 DATA 208,2,230,60,132,45,164
574 DATA 60,132,46,96,256
```

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.

PRODUCT MART

64 DISK-a-ZINE

• For Commodore 64 Users •

The magazine on a disc!
Menu driven • ready to run programs with games • articles • Kids Korner • business programs, and more!

SEND: CHECK OR MONEY ORDER
to
3118 Gilbert Dr.
Shreveport, La. 71104

$8. per issue per year $60.

Commodore 64 is a trademark of Commodore Electronics Limited.

Software For
Commodore 64

DATAFILE $15.95
A data-base management program which lends itself well to address lists, collections, club lists, etc. Records can be Sorted, Found, Saved, Edited, Updated, and sent to optional printer.

FLIGHT 64 $15.95
Your mission is to make an IFR flight, avoiding treacherous mountains and other hazards. Full color graphic instrument display. Topography changes with every flight. Use keyboard or joystick.

SPELLATHON $19.95
Unique spelling tutor for all ages. Make up your own word lists and save them to disk or tape. After drilling play "WORDATHON", a letter scramble game included in the package.

•Prices are for cassette, add $2 for disk.
•Check, MO, or COD. Add $2 for shipping.
•See your dealer or order direct

FANTASY COMPUTERWARE
BOX 481
SIOUX FALLS, SO. DAK. 57101
(605) 335-7684

C-64®/VIC-20® CONSOLE UNIT!

Combine your C-64*/VIC-20* into one compact unit. The C-64*/VIC-20* console unit is designed to house keyboard, disc drive, datasette, and monitor. Sturdy ¼" acrylic.

Smoke Tint	Clear
$69.95	$62.95

Ohio residents add 6% sales tax, $4.00 for shipping and handling. C.O.D.'s, cashiers check or money order. **Call (216) 722-0770 for Visa or MasterCard.** Allow six weeks for delivery.

Name _____
Address _____
City _____ State ____ Zip ____

Specify **clear** or **tint** when ordering. Pat. applied for.

THE VIDEO & COMPUTER PLACE INC
884 North Court Sq.
Medina, OH 44256

VIC-20/64

$48

CASSETTE TO CASSETTE

Did you ever lose an expensive program because the tape stretched or the copy became weak?

NEVER AGAIN!

Because, if it's on tape, **CAS-20** can make a **BACK-UP COPY** for you!

What Makes CAS-20 So Popular?

- A faster way to run off copies of your programs.
- Gold-plated fingers!
- No computer needed.
- Back-up copy as good as original.
- On-board power supply.
- Includes Save Block 5(A000) tape!
- Very low price!
- Simple to use.
- 90 day warranty.

Dealer Inquiries Welcome!

ALSO! . . . SAVE BLOCK 5(A000)

$26.00 FOR VIC-20 Programs saved with this program not only load back into Block 5 but, (for most programs including AUTO START) they start automatically. Great for cartridge back-up copies. (8K expansion required).

Please write or send check or money order to:

COMPUTRONIX
P.O. Box 404 • Clawson, Michigan 48017

Michigan Residents add 4% sales tax. Add $3.00 for postage and handling.
This Unit Requires Two Cassettes For Back-Up Copying.
VIC-20 & 64 ARE TRADEMARKS OF COMMODORE.

VIC-20 EXPANDER

MORE MEMORY — MORE GAMES

FOUR-SLOT BUS EXPANDER BOARD

- PREMIUM SOLDER-MASKED BOARD
- GOLD-PLATED CONNECTORS
- BLOCK SELECT SWITCHES
- ON-BOARD RESET SWITCH
- SAFETY FUSE

$19.00 BARE BOARD (instructions)

Bare Board / instructions	$19.00
Kit board / all parts	$33.00
Complete assembled, tested	$39.00

ADD $2.00 shipping

FROM: The Data Toolbox
P.O. BOX 4808 V
Las Vegas, Nevada
89127

PHONE ORDERS: (702) 648-3258

QUAD-SLOT

VIC-20* KidBit Software Present:

THE PLAY SCHOOL TAPES

Preschool learning programs, on cassettes for the unexpanded VIC

☐ Counting ☐ Little letter/Big Letter
☐ Alphabet Sequence ☐ Same/Different

$9.95 ea $29.95 for all four
Free Brochure ☐

Name _____
Address _____
City _____ St _____ Zip _____

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

KidBit Software
6116 Merced Ave #175
Oak. Ca. 94611 (415) 339-2303

COMMODORE 64* OWNERS ONLY

• SHARE • LEARN • ENJOY •

• Monthly Newsletter
• Public Domain Software
• Reports of Recent 64 Articles
• Local Chapter Meetings
• Service Advice
• Bi-Monthly Magazine • Advice on Training

Send Name, address, phone no. and annual dues ($25) to:

The Commodore 64 Users Group
P.O. Box 572
Glen Ellyn, Illinois 60137

Or Call:

(312)790-4320 (weekdays 9:00 am - 5:00 pm - Central Time)

"An independent not-for-profit organization".

Send for your

A *Programmer's* ***FREE***

POKE'R CARD

Handy 3"x6" reference card makes graphics programming a breeze! EZ to use format includes graphics, poke #'s, etc. Every ACE programmer should have one!

SEND NO MONEY

For prompt reply, send a stamped, self-addressed business-size envelope to:

PM PRODUCTS
4455 TORRANCE BLVD., #177
TORRANCE, CA 90503

or, mark the Reader Response Card.

FOR THE BEST DEAL AROUND. . .

get **PAL** (Programmer's Aids and Logs)

95 pages of programming aids, design worksheets, and info logs. Everything you need to create, design, and document your own programs. **INVALUABLE!** Beginner to Expert. Specify PAL-20 or PAL-64.

Send **$9.95** plus $2 shipping (CA add 6%)

EVERYBODY NEEDS A PAL!

A

WORDPROCESSING FOR THE VIC-20 AND C-64

Scribe

Be more creative and efficient by using the *Scribe's* easy-to-learn menu-driven features. Within minutes use:

- Move
- Insert
- Delete
- Scroll
- Word wrap
- Center
- Formatted printing; Margin adjust; right justify; Single, double, or triple space
- With some printers: graphics, underline, and special characters
- Screen edit
- Up to 10 tabs
- Search and replace
- Save/load/merge files
- 1024 color combinations

For a limited time you can obtain the *Scribe* at a *Special Introductory Price:*

VIC-20 (Specify 16K or 24K version) ~~$20.95~~ ***$16.95***

Commodore 64 version ~~$24.95~~ ***$19.95***

Add $3.00 for discs. Specify your type and model printer.

Send a check or money order (including $3.00 for mailing) to:

***Scribe* Associates**
P.O. Box 292647, Dayton, OH 45429

Allow 3 weeks for personal checks
Ohio residents add 6% sales tax

VIC-20 and C-64 are trademarks of Commodore Business Machines, Inc.
Dealer Inquiries Welcomed

LEARN

Having trouble learning to use your computer?

Reference manuals don't teach. Most BASIC texts don't cover specific personal computers.

TIS solves these problems with step-by-step books tailored for your machine.

For PET/CBM

Understanding your PET/CBM $14.95
Vol. 1: Basic Programming
PET Graphics ..... $ 4.95

For VIC

Understanding Your VIC $11.95
Vol. 1: Basic Programming

Software for PET, VIC, & C64

Software products on cassette or floppy disk with complete instruction manual.
Cassette $24.95
Diskette $29.95

SW-1 MAIL B mailing list system
SW-2 CHECKBOOK record
SW-3 ACCOUNTS keep track of who owes you how much
SW-4MEDIT create and maintain data files
SW-5 CALENDAR appointments, meetings at-a-glance

Money Back Guarantee. VISA/MC accepted.
All prices include UPS or 1st Class postage.

TIS INC

Total Information Services, Inc.
Box 921, Dept. G
Los Alamos, NM 87544

NEW FOR *VIC 20*

16k with Battery Backup **$89**

NO MORE DATA LOSS!

If power fails – data and programs are retained for up to 2 weeks!

No other 16k expansion can make this claim!

PLUS:

- FULL BLOCK SWITCHING
- WRITE PROTECT
- RESET SWITCH, AND MORE

FULLY GUARANTEED

To order, or for free detailed application literature, write to:

ABARIS, INC.
PO Box 2501
Vancouver, WA 98668

Add $3 shipping

Washington residents, add 6% sales tax.

VIC & 64

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

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

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

Send check or money order plus $2.00 (postage and handling)
PA residents add 6% sales tax

Name _____
Address _____
City _____ State _____ ZIP _____

(1) Product of Commodore Business Machines, Inc. (2) Product of United Microware Industries, Inc. (3) Product of Hudson Digital Electronics, Inc. (4) is a Trademark of Commodore Business Machines, Inc.

CHEATSHEET PRODUCTS™
P.O. Box 8299 Fittsburgh, PA. 15218

CGH83

YOUR NIFTY LITTLE C-64 JUST BECAME A GRAPHICS POWERHOUSE

PROSCREEN 64

Finally! Unlock the powerful hi-res and multi-color hi-res graphics potential of your C-64. Now you can create professional graphics for games, graphs, or just for fun!

- Use all of Commodore's characters and/or characters of your own design. Use both C-64 fonts simultaneously.
- Characters may be normal or reverse, positive or negative, upright or rotated. All combinations may be used.
- Place characters anywhere on the screen, not just in the usual character cells. Overlap or erase without changing the background!
- Call hi-res routines from your BASIC program or create graphics in the immediate mode. Switch at will from hi-res screen to standard screen and back.
- Draw or erase with a joystick using any character as the stylus.
- Lines, circles, and arcs drawn automatically. PROSCREEN spline connects discrete dots with a smooth line!
- Machine code program uses not a single byte of user BASIC RAM.

If you're not convinced, send $5.00 (refundable upon Purchase) for users' guide to get the inside scoop. If you're sold, $34.95 gets the whole thing on tape, $37.95 on disk. Check or M.O. to:

WETHERBEE SOFTWARE
P.O. Box 4512, Ann Arbor, MI. 48106

RAINBOW SALES & SERVICES of HOUSTON

1Ø for 1Ø for COMMODORE 64

- 10 PROGRAMS FOR $10.00, POSTAGE PAID ANYWHERE ON EARTH.
- PROGRAMS CAN BE USED AS POINT OF PURCHASE DISPLAY ADVERTISING AS THEY SHOW COMMODORE 64's AUDIO/VIDEO CAPABILITIES BLENDED INTO UNIQUELY PERPETUAL PRESENTATIONS OF SOUND & COLOR.
- A BARGAIN FOR COMPUTER ARTWORK APPLICATIONS AS PROGRAMS EDUCATE USER AND EXPLICITLY SHOW ALL POSSIBLE COMBINATIONS OF CHARACTER, COLOR, VOICE AND SPRITE CAPABILITIES IN A RANDOM GENERATED FORMAT FOR YOUR COSMIC IMPLICATIONS...
- AVAILABLE IN CASSETTE ONLY.

$10.00

SEND CHECK OR MONEY ORDER.
RAINBOW SALES & SERVICES
P.O. BOX 772253
HOUSTON, TEXAS 77215

3-D OTHELLO

OJAI SOFTWARE

Offers for the unexpanded

VIC-20

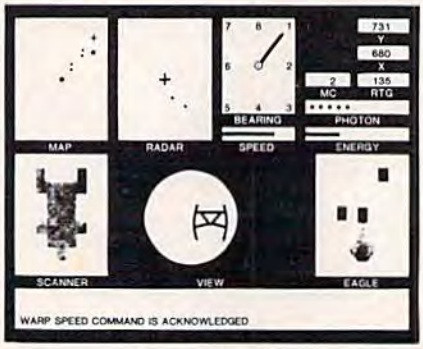
An added dimension to the game of Othello.

Play against friend or computer

$15 per cassette
(Joystick required)

Dealer inquiry invited.
Calif. Res. please add 6% sales tax.

OJAI SOFTWARE
P.O. BOX 1860
OJAI, CALIF. 93023



QUEST OF THE EAGLE

YOUR MISSION THIS TIME...SECURE PLUTONIUM AND DELIVER TO PLANET JEMSIO...ENERGY LOW... RADAR SHOWS CRAFT APPROACHING...TRANSPORT ROOM TOO DAMAGED TO BEAM UP ENERGY...IS CRAFT FRIEND OR FOE?...

CAN YOU COMPLETE THIS MISSION AND ADVANCE TO MORE CHALLENGING MISSIONS OR WILL THEY DESTROY YOU?

COMMODORE 64	FERIN ENTERPRISES 6310 UNDERWOOD AVE. S.W. CEDAR RAPIDS IA. 52404	$28.75 SPECIFY CASSETTE OR DISKETTE

GET THE MOST FROM YOUR VIC-20/C64

CASSETTE INTERFACE

- USE ANY STANDARD CASSETTE RECORDER
- CONNECTS TO THE CASSETTE PORT
- CONTROLS THE CASSETTE MOTOR
- NEEDS NO BATTERIES
- COPY FROM RECORDER TO RECORDER
- THIS IS THE BEST ONE FOR LESS $$$
- ONLY $34.95 POSTPAID. ADD $2.50 FOR SHIPPING OUTSIDE US, CANADA, MEXICO

THE MODEM INTERFACE

- CONNECTS TO THE USER I/O PORT
- NEEDS NO BATTERIES SOLDERING
- CONNECTS ANY STANDARD MODEM
- USE MODEM AUTODIAL, ANSWER FEATURES
- COMES WITH A FREE TYPE IN BASIC TERMINAL PROGRAM
- ONLY $24.95 POSTPAID. ADD $2.50 FOR SHIPPING OUTSIDE US, CANADA, MEXICO

CHARGE OR COD ORDERS CALL

1-800-227-3800

ASK FOR OPERATOR 225

BYTESIZE MICRO TECHNOLOGY
PO BOX 12309 DEPT HD
SEATTLE, WA 98111
(206) 236-BYTE

CALL OR WRITE FOR DEALER INFORMATION

Micro Match™ SAVES MONEY

Don't Let Programs Gather Dust!

TRADE Original SOFTWARE with Documentation for ALL MICROS . . .

WE MATCH TRADERS
for only $5.00
GUARANTEED SATISFACTION

FOR UNLIMITED TRADING:
Ask about SOFTRADERS Worldwide Trading Directories

Send your system description, titles of all software wanted, titles of all software to trade and S.A.S.E. Send no money.

SOFTRADERS™ International
1610 Shomaker Drive
Murphysboro, IL 62966

SPEECH SYNTHESIZER FOR VIC-20

- Speech
- Educational Software
- Game Software
- Robot, Machines
- Easy To Use
- Affordable

The **"CHATTERBOX"** is a standard cartridge with a DIN plug that attaches between the VIC-20 and the TV. Simply follow the principal of allphones to pronounce any word and utilize the simple program to construct a sentance or a speech. Endless inexpensive applications such as Interactive Education program response with speech, game software, robot, announcer, home control center, alarm station, collection gate, vending machine etc. **Order Now, CHATTERBOX $99.00. Also data cassette for VIC-20 or VIC-64 $55.00.** Visa & Master card with exp. date accepted. CA res. add 6.5% sales tax. Send for free brochure.

MAXTRON
7807A Telegraph Road, Montebello, CA 90640
(213) 724-4491 • 724-0829

VIC-20 SOFTWARE

ROTORCOMMAND — An original, exciting space game like no other you have played before. You control a ship armed with a revolving forcefield. Flying around the screen, your mission is to destroy all intruders who try to get to your base. This one is quite addicting. Needs 3K and joystick.

HANGPERSON — This great new version of the classic word game features a vocabulary of over 500 words. The Vic's sound and color capabilities are used well here. Hangperson is great fun for every member of the family. Needs 3K.

STARFIGHTER — One of the best games we've seen for the Vic! You must seek and destroy enemy fighters in deep space. Check the map, pick your quadrant, and go . . . but watch your fuel. Docking with fuel ships will win some back. This game has it all. You may never turn it off. Needs 3K and joystick.

*** This set of three games is only $10.00 postage included ***

*****EDUCATIONAL SOFTWARE*****

Also available is a set of two fun and educational games. One game specializes in teaching the four basic math skills. The other is a vocabulary builder for elementary school students. Both are enjoyable and will hold any child's attention for hours. Both games on one cassette for only $10.00.

GREAT DEAL

Buy both the educational pack and the games pack for only $18.00 and receive another super game program free of charge!!

MAIL ALL ORDERS TO: LOMBO SOFTWARE
914-769-6541
90 Birch Drive
Pleasantville, N.Y. 10570

WE ACCEPT CHECKS OR MONEY ORDERS
All Prices Include Postage
New York State residents must add sales tax

FAMILY TREE

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

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

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

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

GENEALOGY SOFTWARE
PHONE 519-344-3990

P.O. BOX 1151
PORT HURON, MICHIGAN 48061

1046 PARKWOOD AVE
SARNIA, ONTARIO N7V 3T9

Writers, programmers... *COMPUTE!'s Gazette* is looking for articles on the VIC and 64.

```
$ PAYROLL $
* USER Friendly
* PRINTS Employee  Pay
Stub
* W-2's At the end  of
year.
* FEDERAL & F.I.C.A. &
STATE Calculations
* TOTALS FOR QUARTERLY
TAX  REPORTS:  Prints
all emoloyees on  file
and  gives  you  their
combined totals.
* FULL DOCUMENTATION
     ONLY  $89.95
   C-64 DISK ONLY
J.O.Warren
P.O.Box 297
Clallam Bay, Wa.98326
PHONE 206-963-2693
```

CANADIANS!

IF YOU ADORE YOUR 64 — YOU'LL LOVE C.S.S.

CANADIAN SOFTWARE SOURCE provides you with Commodore 64 Home, Business, and Game Software. Compare our prices and amazingly fast* delivery to your door.

	Canadian $ Retail	C.S.S.
• Paperclip 64 (Batteries)	$ 150.00	$129.95
• Frogger (Sierra) (D)	$ 49.95	$36.95
• Temple of Apshai (Epyx) (T D)	$ 54.95	$44.95
• Home Accountant (Continental) (D)	$ 93.95	$74.75
• Jumpman (Epyx) (T D)	$ 54.95	$44.95
• Quick Brown Fox (QBF)	$ 79.95	$68.85
• Forbidden Forest (Cosmi) (T D)	$ 25.95	$21.95
• Fire Command-joy stick	$ 69.95	$59.95
• Acc t Rec Pay (Timeworks) ea (D)	$149.95	$99.95

GIGANTIC SELECTION of over 200 more popular Commodore 64 programs from the top 20 manufacturers. Write or phone for FREE catalogue or to place order.

CANADIAN SOFTWARE SOURCE
Box 340, Station "W", Toronto, Ont.
M6M 5B9
VISA **(416) 491-2942** MasterCard

Ontario residents add 7% sales tax. Send certified cheque or money order. VISA & MasterCard, please include card number, expiry date, and signature. Add $2.00 for shipping and handling.

*Delivery by U.P.S. within 3 days of order date if stocked by local suppliers.

MAKE YOUR OWN CARTRIDGES!

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

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

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

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

The Banner Machine™
- For the **Commodore 64** with tape or disk
- For the **VIC-20** with 24K memory and disk
- Five fonts available; more coming
- Print on any Epson RX, FX, or MX with Graftrax; CBM 1525; Okidata 82A with Okigraph; or 8510 Prowriter
- Uses dot graphics—not Commodore block graphics
- Menu-driven program operates like a word processor
- Makes signs up to 13" tall by any length
- Makes borders of variable width up to ¾ inch
- 8 sizes of letters from ¾" to 8½" high
- Proportional spacing
- Automatic centering; Right and left justifying; Tab

Order The Banner Machine™—$49.95 from

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

Advertisers Index

Reader Service Number/Advertiser	Page
102 A-1 Computer Services	170
103 Aardvark Action Software	135
Abacus Software	62
Abaris, Inc.	206
104 The Abnel Company	93
105 Academy Software	112
106 Access Software Inc.	85
Accolade Computer Products	148
107 Advanced Processor Systems	129
108 Adventure International	70
109 Adventure International	111
The Alien Group	65
Alphacom	55
American Peripherals	161
110 Animax Computer	44
111 Apropos Technology	155
Assembly Technology	145
112 Avalon Hill Game Company	41
B. Dalton Bookseller	66
Bible Software	145
113 Boone Corp.	67
114 Bröderbund Software	IFC
115 Byte-Ryte	174
116 Bytes & Pieces	173
117 Bytes & Pieces	93
118 Bytesize Micro Technology	206
Canadian Software Source	207
119 Cardco, Inc.	IBC
Castle Software	62
120 Century Micro Products	177
Cheatsheet Products	206
Comm*Data Computer House, Inc.	89
121 Commercial Data Systems Ltd.	71
Commodore 64	BC
Commodore 64 Users Group	204
122 Compatible Systems Incorporated	174
Compuscope	167
CompuServe	47
Computer Mail Order	151
123 Computer Marketing Services, Inc.	37
124 ComputerMat	153
125 Computer Outlet	169
126 Computer Place	175
Computer Software Assoc., Inc.	119
Computronix	204
127 ComStar	178
128 Continental Software	43
Cosmopolitan Software Services	21
Creative Software	4
Crown Computer and Software	177
129 Cyberia	20
Data Equipment Supply Corp.	158
The Data Toolbox	204
130 Davka Corporation	121
Eastern House	172
131 EduCare	145
Elcomp Publishing, Inc.	53
Electronic Arts	23
Entech	86
EPYX	25
EPYX	27
132 Estes Engineering, Inc.	171
Fantasy Computerware	204
Ferin Enterprises	206
133 First Star Software Inc.	65
Foxfire Systems, Inc.	170
134 French Silk	109
135 Genealogy Software	207
Genesis Computer Corporation	112
Grapevine Software, Inc.	77
Human Engineered Software	75
136 Hytec Systems	91
Infocom	28,29
Institutional Computer Development Corp.	167
Interesting Software	139
J. O. Warren	207
137 Jay Sun Enterprises, Incorporated	94
138 Jini-Micro-Systems, Inc.	134
Jou Laboratories	205
Kidbit Software	204
139 Kyan Software	167
Limbic Systems Inc.	118
Lombo Software	207
Magic Carpet	139
Mariner Systems, Corp.	205
Marshmallow Software	205
Maxtron	207
140 MicroDigital	45
141 Micro-80 Inc.	171
Micro-Ware Distributing	57
Micro World Electronix, Inc.	165
142 Midwest Micro, Inc.	147
Mirage Concepts	117
MMG Micro Software	69
Mosaic Electronics, Inc.	94
143 New Leaf Inc.	78
Nova Software Incorporated	15
Nüfekop	35
OJAI Software	206
Olympic Sales Company	171
Otto Systems	207
Parallel Systems	205
144 Parsec Research	131
PM Products	204
Powerbyte	174
145 Professional Software, Inc.	9
146 Program Design, Inc.	94
The Program Store	101
Pro-Line Software	161
Pro-Line Software	171
147 Protecto Enterprizes	163
148 Protecto Enterprizes	99
149 Protecto Enterprizes	106
150 Protecto Enterprizes	105,107
151 Public Domain, Inc.	205
Purvis Systems Incorporated	165
152 Pyramid Software International	161
Quality Computer	86
Queue Products	205
Quicksilva	133
153 Rainbow Computer Corporation	137
Rainbow Sales & Services	206
Romox	51
SAVE	175
Scribe Associates	206
Sierra On-Line, Inc.	31
Sierra On-Line, Inc.	81
Sierra On-Line, Inc.	7
Sierra On-Line, Inc.	91
154 Sim Computer Products	20
64 Disk-a-Zine	204
SJB Distributors Inc.	157
155 Skyles Electric Works	87
156 Skyles Electric Works	97
Soft-Guide	176
Softsync, Inc.	205
157 Softraders International	207
Softron, Inc.	61
158 The Software Connection	175
159 Software Gallery	172
Software To Go	124
160 Software Warehouse Outlet	160
Sophware	171
Southwest Micro Systems, Inc.	158
161 Space Shuttle Software	176
Spinnaker	16,17
Spinnaker	19
162 Star Micronics	33
163 subLOGIC	39
164 Such-A-Deal! Software	79
Sunsoft	173
Synapse	49
165 Systems Management Associates	123
166 Systems Management Associates	125
167 T & F Software Company	1
3G Company, Inc.	95
168 Tenex Computer Marketing Systems	103
169 Timeworks, Inc.	63
170 Toronto Pet Users Group	93
171 Total Information Services, Inc.	206
172 Toll Software, Inc.	147
173 Tri Micro	115
Tronix	12,13
Tronix	11
174 Umbrella Software Incorporated	44
175 United Microware Industries	83
Viasala Inc.	52
Valorum	205
Vic Flic	145
176 Victory Software Inc.	95
The Video & Computer Place Inc.	204
Virginia Micro Systems	207
177 WESoft	
Wetherbee Software	206
York 10 Computerware	177
Zytek	178

COMPUTE!'s First Book of VIC	127
COMPUTE!'s First Book of VIC Games	143

New From Cardco

Five Slot Expansion Interface for the C-64

The CARDBOARD/5 (CB/5) is an enclosed five slot, fully switch selectable, expansion interface for the Commodore 64™. This quality product allows the user to switch select any cartridge slot or combination of cartridge slots. Twenty-two color coded light emitting diodes give status indication. Each slot has four LEDs and two toggle switches for indication and control. Two master toggle switches allow the user to manually override any situation.

All Cardco products are individually tested to insure quality and reliability.

- full support under the board to prevent flexing
- full plastic enclosure to insure safety
- fused to protect your computer
- convenient reset button
- CARDCO, Inc.'s exclusive Lifetime Guarantee

Some of the features of the CARDBOARD/5 are:

- high quality glass/epoxy circuit board
- gold plated contacts
- logic lines are switched by solid state IC switches
- full LED status indication
- convenient toggle switches

See a complete line of American made Cardco Products at a computer store near you, today.

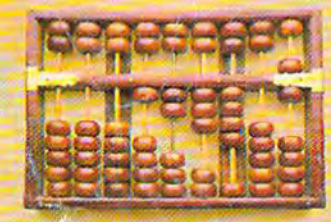
313 Mathewson • Wichita, Kansas 67214 • (316) 267-6525

It writes, rates, creates, even telecommunicates. Costs less, does more- the Commodore 64.

When Commodore introduced the 64, the industry suddenly realized that there would be a computer in every home, school and business years before anyone ever dreamed.

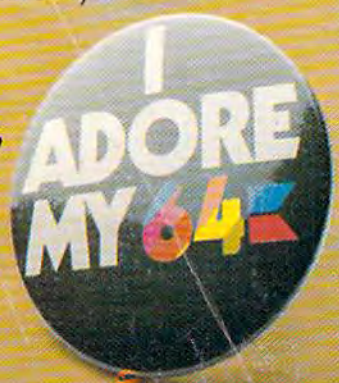
That's because Commodore 64 halved the price of high technology: while you can compare the 64's capabilities with those of any sophisticated business PC, you can compare its price with that of an average television.

What can you do with it? Create with



its high resolution Sprite Graphics. Add a printer and type with it. Add a disk drive to use spread sheets and other financial programs. Learn and play music through your home sound system on the 64's professional quality music synthesizer.

Add a modem, and hook up with the vast computer networks through your telephone. In short, the Commodore 64 is the ultimate personal computer, at a price you can afford.



COMMODORE 64