COMPUTE!'s GAZETTE

For Owners And Users Of **Commodore VIC-20** And **64** Personal Computers

Balloon Blitz

An armed anti-tank hot air balloon? That's what you operate in this multilevel action-strategy game for the VIC and 64.
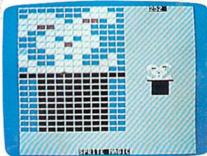
Campaign Manager

Manage your candidate's presidential campaign with advertising, polls, and personal appearances in this colorful, all machine language election simulation for the 64.

Also In This Issue:

Home Telecommunications: Uploading

Computing For Families

Selecting A Printer Interface

Choosing the right interface can make all the difference to your printer. An in-depth look at what's available.

Sprite Magic

Animate your 64 with this feature-packed all machine language sprite editor.

THANKS TO COMPUSERVE'S CB SIMULATOR, "DIGITAL FOX" ACCESSED "DATA HARI" AND PROCEEDED TO AN "ALTARED" STATE.

The CB Simulator, where CompuServe Subscribers can Access Friends and Influence People on 72 Different Channels.

Just pick your handle and get on line. From math to matrimony, there's always someone out there who speaks your language. Friends from all over the U.S. and Canada are at it 24 hours a day. Talking tech or just having fun. And if you've got a secret, just use the CB Scrambler.

That'll fool the "lurkers," those CB "see it alls" who get their kicks by watching. Or you can always use the private talk mode for guaranteed one-to-one conversation.

The CB Simulator is just one of CompuServe's many electronic communications options that include a National Bulletin Board, Professional Forums and Electronic Mail. Plus, there's a world of on-line information and entertainment all for the price of a local phone call plus connect time.

You can access CompuServe with almost any computer and modem, terminal or communicating word processor.

To receive your illustrated guide to the CompuServe Information Service and learn how to subscribe, call or contact:
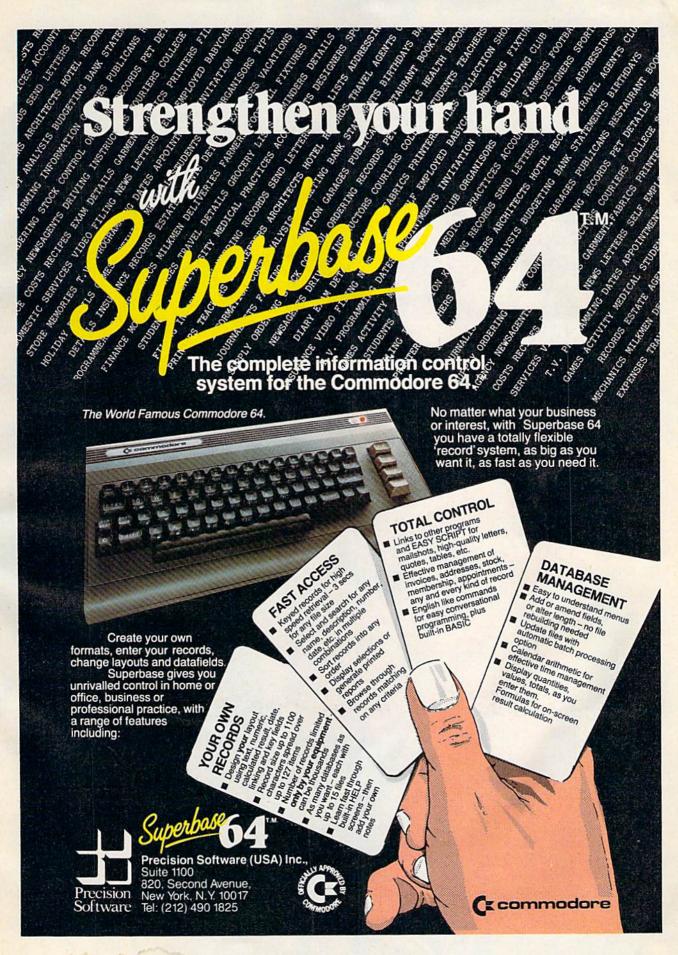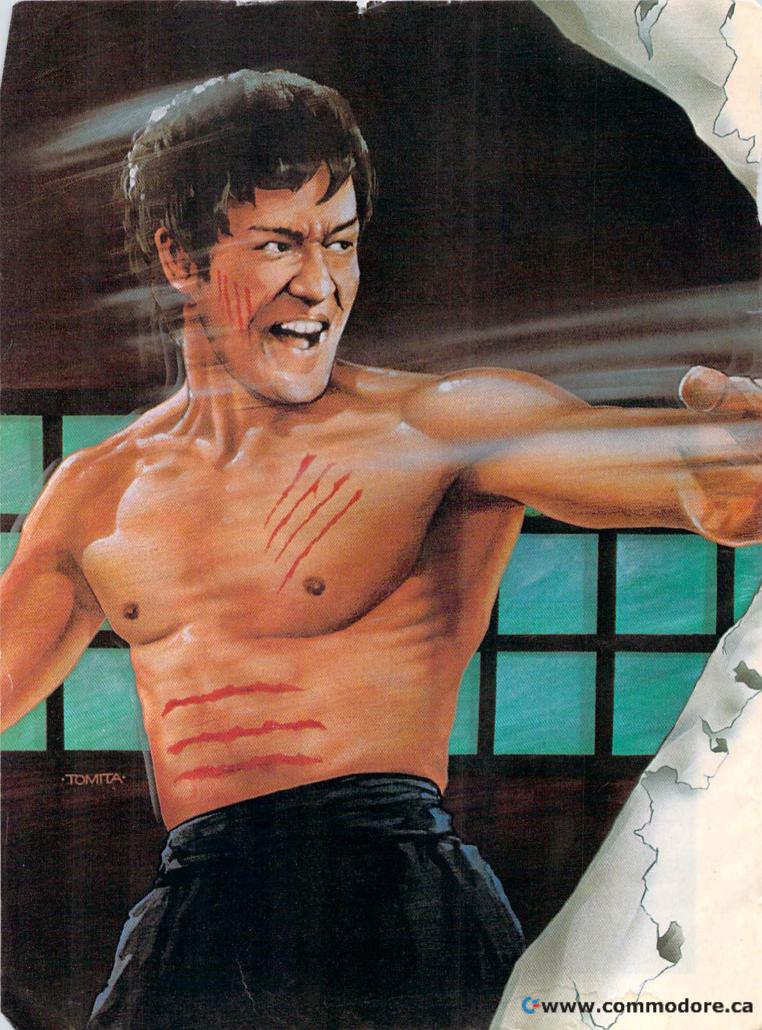
CompuServe

Consumer Information Service. P.O. Box 20212
5000 Arlington Centre Blvd., Columbus, OH 43220
800-848-8199
In Ohio call 614-457-0802

An H&R Block Company

℃ www.commodore.ca

FEATURES

GAMES

REVIEWS

EDUCATION/HOME APPLICATIONS

PROGRAMMING

DEPARTMENTS

PROGRAM LISTINGS

*=General, **V**=VIC-20, **64**=Commodore 64.

www.commodore.ca

THE EDITOR'S
notes

The Summer Consumer Electronics Show in Chicago is the first major trade show since Commodore underwent major changes in management. As you'll recall, these changes were triggered by the departure in late February of Commodore founder and visionary, Jack Tramiel. He was replaced by Marshall Smith, a manager with extensive financial and manufacturing experience, but a computer industry novice.

Commodore, known for its aggressive leadership in the home computer market, this March began to experience a steady loss of its battle-seasoned middle managers.

Rumors from within Commodore revealed a growing conflict between recently hired (and probably more traditional) managers versus the younger, but Commodore-seasoned, veterans of the company's past battles. These veterans were accustomed to reacting quickly to market changes and demands.

Although the real impact, if any, of this potential change in corporate personality won't be visible to the public for many months, we were anxious to see what Commodore did at the June CES show.

Things seem to be moving ahead slowly and wisely. As expected the VIC-20 was significantly de-emphasized. In fact, one highly placed Commodore official said that the company

had stopped manufacturing the VIC this spring and any VICs now being sold are surplus inventory. That surplus is expected to be gone by August. When the last VIC is sold, it will conclude a remarkable chapter in personal computing: approximately 2½ million VICs were sold worldwide, 1½ million in the U.S. alone.

The Commodore 64 continues to dominate the company's product line. A new computer family, lead by the Commodore 264 that was announced at the Winter CES last January, was reintroduced this June, but there were changes. Fortunately, these changes seem to reflect a responsiveness to dealer feedback from the winter introduction. The first 264s were to have included four versions of built-in, optional applications programs—a spreadsheet, a word processor, a graphics package, and a data base manager.

Dealers evidently objected to having to stock these various ROM chips and Commodore has now decided to include the four applications programs as part of the standard machine. Renamed the Commodore +/4, there will now be only one version of the computer which should simplify things for both dealers and consumers.

Another member of this new family of machines, the Commodore 16, comes with

16K of RAM memory, but is expandable to 64K.

Both the 16 and the +/4 models are housed in a charcoal gray case which is otherwise quite similar to the familiar VIC and 64 keyboards. Neither of the new machines will include sprites or SID chips. And although the new computers are expected to go on sale this fall, prices have not yet been established.
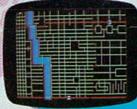
All in all, Commodore seems to be charting a measured, thoughtful (though some would argue overcautious) course through the choppy seas of the personal computer marketplace. Time will tell.

Robert Lock

Editor In Chief

www.commodore.ca

GAZETTE FEEDBACK

Do you have a question or a problem? Have you discovered something that could help other VIC-20 and Commodore 64 users? Do you have a comment about something you've read in COMPUTE!'s GAZETTE? We want to hear from you. Write to Gazette Feedback, COMPUTE!'s GAZETTE, P.O. Box 5406, Greensboro, NC 27403.

Stacked Disk Drives

Is it safe for me to stack my 1541 disk drives on top of each other? I know the disk drive's motor generates a lot of heat, and I'd hate to burn something out.

Tom McSweeney

It's best not to. If you take a look at your 1541, you'll notice that the vents are located on the top rear and the bottom rear. Because heat rises, the drive on the bottom will vent its heat into the drive above.

Whether you have one disk drive or ten, it's best to find separate shelf space for each. A disk drive should be placed where the ventilation holes are unobstructed, and air can circulate freely.

Commodore Key

I'd like to know if there is any way to program the Commodore key located at the bottom left of the keyboard. Is there a CHR$ code for it?

Andrew Smith

No, there isn't a CHR$ code for the Commodore key, but there's a way you can check to see if this key is pressed. Memory location 653 is used to detect if the SHIFT, CTRL, or the Commodore keys are pressed. Enter and RUN this one-line BASIC program:

10 PRINTPEEK(653):GOTO10

While the program is running, press each of the three keys mentioned above. As each key is pressed, a different value will be returned. Pressing the SHIFT returns a 1, the Commodore key a 2, and the CTRL key a 4. Thus, you could determine whether the Commodore key was being pressed during a program by using a line like:

100 IF PEEK(653)=2 THEN 200

Magic Cursor

I would like to know if it's possible to make the cursor "write." For example, when you use the PRINT command, the statement inside the quotes just appears. Is it possible to make the cursor go across the screen and in its trail write the message you desire?

Chris Rust

Yes, it is possible, and it can be done a number of different ways. Using the PRINT statement and string manipulation, you can do it this way on the VIC or 64:

Cursor Write Routine

```
100 A$="THIS IS A TEST"
110 FORI=1TOLEN(A$)
120 PRINTMID$(A$,I,1)"{RVS} {OFF}{LEFT}";
130 FORJ=1TO50:NEXTJ
140 NEXTI:PRINT" "
```

Another method involves POKEs, and the code is considerably longer. First, enter one of the following lines, which will POKE a solid cursorlike block (the cursor is a reversed space, CHR$ 160) onto your screen:

    POKE 1536,160:POKE 55808,1 *(Commodore 64)*
    POKE 7936,160:POKE 38656,6 *(unexpanded VIC)*

Each of these lines first POKEs the reversed space to the screen, then POKEs the letter you desire to the same position. You would then move to the next space, and start the cycle over again. The speed that the cursor moves could be controlled by a delay loop. As an example, enter and run the following program on your VIC or 64.

```
10 SC=1024:CO=55296:INC=40:SYS65517:IFPEE
   K(781)=22THENSC=7680:CO=38400:INC=22
                                    :rem 25
15 PRINT"{CLR}":CH=PEEK(646)        :rem 226
20 FORA=COTOCO+200:POKEA,CH:NEXT    :rem 37
30 READD:IFD>255THENGOTO50          :rem 119
40 POKESC,160:FORT=1TO50:NEXTT:POKESC,D:S
   C=SC+1:CT=CT+1:GOTO30            :rem 233
50 SC=SC+INC-CT:CT=0:IFD=999THENPRINT:END
                                    :rem 151
60 GOTO30                           :rem 2
100 DATA 8,5,12,12,15,44,256,20,8,9,19,32
    ,9,19,32,25,15,21,18,256,3,15,13,16,2
    1                               :rem 16
110 DATA 20,5,18,32,19,16,5,1,11,9,14,7,4
    6,999                           :rem 208
```

In this routine, the cursor speed can be easily changed. In line 40, change T=1TO50 to T=1TO5, and rerun the program. As you can see, the cursor speed is much faster.

Incidentally, an interesting technique is found in line 10. This line determines if the program is being run on a VIC or 64, and sets up the screen and color memory POKE locations accordingly. This

is done with two statements.

Type and enter the following line:

SYS65517:PRINTPEEK(781)

If you're using a VIC, the value printed will be 22; on a 64, it will be 40. This is the number of columns on the screen of each machine. The SYS to memory location 65517 is the start of the screen kernal routine which is used to determine which machine is in use. If you're writing a program for both machines, this technique is very handy.

Scratching Files

I have many disk files that I am unable to scratch using the S0: command. I have a 1541 disk drive and have read the user's manual, but every time I try, I get a FILE NOT OPEN ERROR or a SYNTAX ERROR. Can you please help?

A. Padgeh

The command format for scratching files on the 1541 is:

OPEN 15,8,15: PRINT#15,"S0:filename":CLOSE 15

where filename is the exact name of the file you wish to be scratched.

When using the scratch command, you may also use the wildcard (*). For example, if you wish to scratch a program from your disk named SPACE GAME, you could use the format:

OPEN 15,8,15:PRINT#15,"S0:SPACE*":CLOSE 15

However, when using the wildcard, more than one file may be scratched. For example, when using the above format (SPACE*), any other files on the same disk starting with SPACE (SPACE CADET, SPACE MAN, SPACE.HEATER, etc.) would also be scratched.

Thus, if you have a file that you can't seem to scratch in the normal way, try using a wildcard. For example, if you can't seem to get rid of a program called MYPROG, try a command like

OPEN 15,8,15,"S0:MYP*":CLOSE 15

Just make sure that there are no programs you want to retain on the disk which have the same pattern (for the example above, MYP as the first three letters).

Lost Forever

Once you've typed in the NEW command on disk, is there any way to retrieve the programs that were lost?

Jason Whitley

Sorry, the programs on that NEWed disk are lost forever. When you format a diskette (with the NEW command), a 1 is written to almost every byte on the diskette.

This means that any programs on the disk were erased and overwritten.

NEW is used to prepare newly purchased diskettes for use (called formatting). The syntax for the NEW command is as follows:

OPEN 15,8,15: PRINT#15,"N0:diskname,id":CLOSE 15

where diskname is any name up to 16 characters in length, and id is any two-character identifier. It's a good idea to use different two-character id's on different disks.

64 Cold Starts

In your April Feedback reply ("Cold Starting"), you mentioned you could reset the 64 by entering SYS 64738. However, the user's guide says to enter SYS 64759. What is the difference between the two, and what do these cold starts do?

Steven Wiberg

The correct address to initiate a system cold start (which resets pointers, vectors, "NEWs" BASIC, etc.) is 64738.

The address 64759 is apparently a misprint. $FCE2 (64738) is the beginning of the cold start routine. If you disassemble the routine in ROM ($FCE2-$FD01), you'll see that $FCF7 (64759) is in the middle of a JSR command. Since 64759 does not point to the beginning of an instruction, it is not a good entry point for the cold start routine.

Damaged Disk Drives?

I recently purchased a game on disk. Although there was no warning on the outside of the package, at the very end of the instruction booklet there was written: "Caution: ..... is protected against unauthorized copying. Attempting to copy this disk may result in damage to your disk drive." Is this possible, and if so, how?

Bill Winterling

Making copies of copyrighted software without permission of the distributor is against the law. However, the warning about damaging your disk drive is probably just a scare tactic.

When you make copies of diskettes, the computer and disk drive are under the control of the copy program, not the software being copied.

On the other hand, if you do try to copy "protected" diskettes, you can cause unusual wear on your drive. Many software companies create bad (error laden) tracks and sectors on their diskettes as a method of protection. You might have heard the disk drive "chatter" when you were loading one of these diskettes.

When the copy program you're running hits the bad sectors, the drive will chatter while trying to read the errors. If the diskette has more than one track full of errors, the drive could go through the chatter routine 20 times or more. This constant vibration on the read/write head can cause it to become misaligned, necessitating repair. ◐

SIMPLE ANSWERS TO COMMON QUESTIONS

QA

Tom R. Halfhill, Staff Editor

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

Q. *Are all word processors compatible with all printers?*

A. No they aren't, and this can cause many problems when you're trying to assemble a word processing system made up of components from several different companies. Not only does the word processing program have to be compatible with the computer, but it also must be compatible with the printer and printer interface. It can be maddening, sometimes, to discover that the program you like won't work well with the printer you bought, or that neither works well with your printer interface. Putting together a well-matched system calls for some very careful planning—*before* you buy.

Of course, if you want to be safe, you could purchase all the components from the same company and see them demonstrated by the dealer before you take them home. This would be like buying a matched-component stereo system; you're assured of compatibility.

But perhaps you've seen or heard of another word processing program which you prefer for some reason. Or maybe you want a printer that is faster or more versatile than the one offered by the company which makes the computer. Or maybe you already own a printer and you want to build your system around it. Now the responsibility for making sure everything is compatible is up to you.

Remember that the word processing program and the printer must complement each other. The printer might be capable of printing in special typefaces such as condensed, expanded, boldface, underlining, or italics, but it won't do you much good if the program can't send the printer control codes to activate those features. Likewise, a program that has commands for underlining or italics can't add those features to a printer which lacks them. In some cases, the program's commands for a feature such as underlining might not even work on a printer that does have underlining. Special printer features are switched on and off by codes sent from the computer, and the codes vary from printer to printer. The program must be capable of sending exactly the right codes.

So how can you be sure if a certain word processing program will work with a certain printer? First, try to find someone who is already using the same setup (check with your local Commodore user group). If that fails, perhaps the dealer who is selling the printer or word processing program can answer your questions. If not, you'll have to do some research. Before buying a word processor, read the manual. Look for a section on printers. See if the program can send a wide variety of control codes to the printer. Then check the printer's manual to see if the features you want can be switched on and off with those codes. Finally, investigate the printer interface to be certain it will function properly with the printer you're considering.

When assembling a system this way, you'll probably end up making a few compromises. For instance, the word processor you choose might have every feature you've dreamed of, yet it may not support underlining on your printer. You may conclude that underlining is worth giving up for the other features.

If all this sounds like a lot of work, it is. Sometimes you won't be able to find out how well the various parts match together until you get them home, plug everything in, and try them out. Ask the dealers who sell you each component for return privileges in case you encounter major problems.

Q. *I need to know when, or when not, to insert a diskette in a disk drive. I am very confused about statements in the user's manual. Page 8 indicates you should always remove the diskette before the drive is turned on or off. Okay. But the next sentence says never to remove it when the green light is on. The green operational light is always on. Could they have meant the red light? When is the best time to remove or change diskettes?*

A. The paragraph you're referring to, on page 8 of the Commodore *VIC-1541 User's Manual*, reads:

"Remember to always remove the diskette before the drive is turned off or on. Never remove the diskette when the green drive light is on! Data can be destroyed by the drive at this time!"

As you surmised, the manual is in error. The green light is a power indicator LED which should always be glowing when the disk drive is switched on. Commodore meant to warn you against fiddling with the diskette when the *red* LED is glowing. The red LED is called a *busy light*, and it indicates when the disk drive is accessing the disk (either reading or writing). Naturally, if you interrupt this process by popping open the drive door, you'll probably lose some data or even mess up the disk. Interrupting a write operation (when saving a program, for instance) would leave the file unclosed, and the next file you save could overwrite it.

Commodore is, however, correct in warning you against switching the drive on or off with a disk inserted. Although you could probably get away with this most of the time, there's a chance that a power surge caused by switching the drive on or off could destroy some data on the disk.
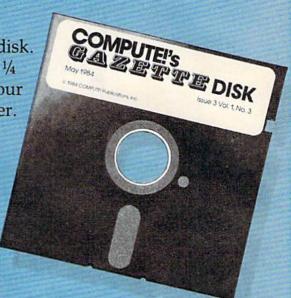
Here's the correct procedure for handling disks with 1541 disk drives:

1. Switch on the computer.

2. Switch on the disk drive.

3. When the drive stops its initial whirring, insert the disk and proceed as usual.

4. When you're ready to end the session, remove the disk before switching off the drive.

It doesn't matter if the disk drive door is open or closed when switching the power on or off. However, many people prefer to close the door when the drive's not in use to keep out dust. **Ⓖ**

COMPUTE!'s GAZETTE DISK

COMPUTE!'s GAZETTE is now available on disk. Each month you can receive a fully tested 5¼ inch floppy disk which will run on either your Commodore 64 or VIC-20 personal computer. Each issue of COMPUTE!'s GAZETTE DISK will contain all the programs which appear in the corresponding issue of COMPUTE!'s GAZETTE magazine. You'll save hours of typing time and gain hours of enjoyment with all the quality programs found each month in COMPUTE!'s GAZETTE.

COMPUTE!'s GAZETTE DISK
May 1984
© 1984 COMPUTE! Publications Inc.
Issue 3 Vol. 1, No. 3

Here are just a few of the quality programs for the VIC and 64 which appeared in the July 1984 issue:

- *Ultrafont+*—A feature-packed character editor for the 64 written in machine language. Lets you concentrate on your artwork instead of programming.
- *Space Patrol*—Engaged in a battle in space, you must prevent aliens from landing. An action-strategy game for the VIC and 64.
- *Robot Math*—An excellent learning tool that's fun for children. Included is a menu that lets you tailor the program to the child's learning level. For the VIC and 64.

- *Power BASIC: Color Chart*—A short and very handy utility for the VIC and 64 that displays all possible combinations of character and background colors.

All the programs included in each issue of COMPUTE!'s GAZETTE are available on disk. Order yours today.

Ordering Information

To Subscribe:
Return the attached postpaid card or call COMPUTE!'s GAZETTE TOLL FREE 800-334-0868 (in North Carolina 919-275-9809). All orders must be prepaid—send check or money order or charge to Visa, MasterCard, or American Express.

Individual Issues:
Individual issues of the GAZETTE DISK can be ordered for $7.95 (in the US and Canada add $1 per issue for shipping and handling. Outside the US and Canada add $3 per issue). Individual issues can be ordered by calling TOLL FREE 800-334-0868 (in North Carolina call 919-275-9809), or by sending your prepaid order to GAZETTE DISK, P.O. Box 5406, Greensboro, NC 27403.
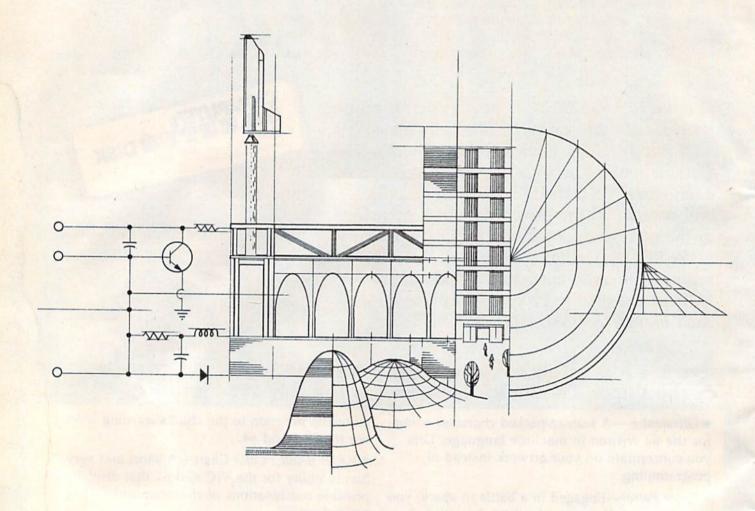
Subscription Rates:
One year (12 disks) subscription $69.95
(add $36 shipping and handling outside the US and Canada)

Two year (24 disks) subscription $129.95
(add $72 shipping and handling outside the US and Canada)

C= www.commodore.ca

An Introduction To Plotters

Robert Sims, Assistant Editor

There was a time, back in the days before computers, when clerks with ink-stained fingers functioned as human machines, using brass and stainless steel implements to create mechanical drawings of humankind's inventions. Drawing precision was measured in thirty-seconds of an inch, because that's about as small a scale as the human eye and hand can manage.

In those low-tech days, people thought of plotters as malcontents who skulked about in crumbling basements and argued over the best way to get rid of the king.
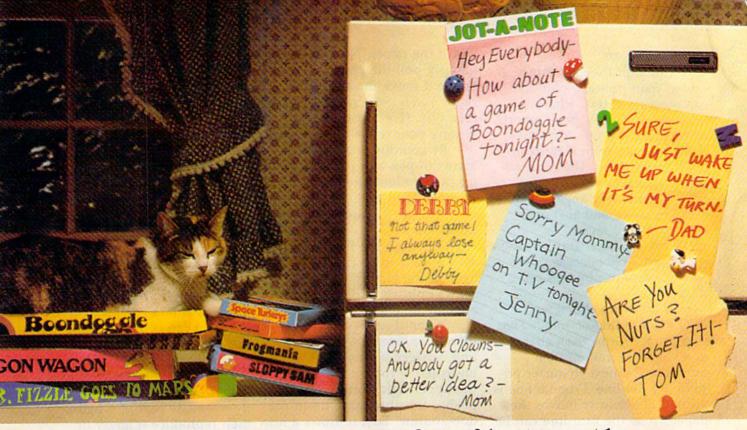
Today, clerks have cleaner fingers because plotters are merely machines which can draw two curved lines, exactly parallel and a few thousandths of an inch apart.

In their own quiet, dull way, these computer peripherals are as revolutionary as their anarchistic namesakes.

Plotters make it possible to turn out contractor-ready architectural drawings on a home computer. Students can turn in figures of such symmetry, composed of such smoothly curving lines and exact angles, as would tempt a geometry teacher to pass out A+ grades left and right.

Artists, if they can arrange a marriage between the rigors of trigonometry and the freedom of creativity, can use plotters to expand the horizons of

reproductive art.

Although they are much slower than printers, plotters can even print text, whether it be business letters, book reports, or program listings. Not only that, they can print it horizontally or vertically. The more sophisticated plotters can print text at nearly any angle (even upside down), and also offer italic and bold faces.

This precision and versatility are possible because of the way plotters put images on paper.

Printers are character-oriented and line-oriented. That is, a printer forms each character in a single operation, then moves to the next character position on the line. When a line is printed, the printer moves the paper up so the next line can be printed. But the printer doesn't keep track of its current position in relation to what has already been printed.

Plotters, on the other hand, are point-oriented and direction-oriented. Their function requires that they keep track of current position in relation to every other point on the paper.

To understand how plotters draw, you must think back to eighth grade geometry, when the teacher tried to interest you in the Pythagorean theorem, Cartesian coordinates, and other angular mysteries.

Plotters work on an X-Y coordinate system, in which each plottable point is identified by a pair of numbers. The plotter positions its pen and paper according to these number pairs. To draw horizontally, it moves the pen back and forth; to draw vertically, it moves the paper; to draw diagonally, it moves both paper and pen at the same time.

In order to appreciate the precision required to draw diagonally in this manner, hold the bottom edge of a piece of paper with one hand. With the other, press a pen to the lower-left corner of the paper. Now, simultaneously pull the paper toward you and move the pen to the right. You'll get a ragged diagonal line. (For a real test, try writing your name this way.)

By coordinating the movements of pen and paper, a plotter can produce any shape, from a single point to a complete set of engineering plans for a geodesic dome.

Although the final result may appear to be curved or diagonal, the plotter actually draws only vertically or horizontally, in a stairstep.

When it receives a command, for example, to draw a line between a point in the lower-left corner to a point in the upper-right corner, it accomplishes this through a fill-in-the-dots routine. First it draws a line from the original point to the nearest adjacent point which is between the original and end points. Then it draws a line from that point to the next point, and repeats this process until the end point is reached.

The result is not one straight line, but a series of tiny interconnected lines. Whether this conglomeration looks like a single line depends on the plotter's resolution, or how far apart the individual points are.

Most plotters which can be interfaced with the VIC and Commodore 64 will have a resolution of between .0078 inch and .001 inch.

Although the difference may seem quite small, it is critical to the apparent unity of a line. A resolution of .0078 inch will leave a visible stairstep effect, while a line drawn with .001 resolution will appear unbroken to the eye.

Other plotter features also vary widely among models.

Generally, the less expensive plotters are "dumber" than the higher-priced models. The Commodore 1520 Printer/Plotter (see the product review elsewhere in this issue) can only draw from point to point. It knows no formulas for making geometric shapes, and it recognizes only the basic commands which position the pen, place it onto the paper, and lift it again.

On the other hand, the Bausch & Lomb DMP-40 Plotter (which costs about $800 more than the 1520) recognizes an entire language of commands which not only move the pen, but also tell it to draw complete figures such as circles and ellipses. In fact, if you give it a few key points in a complex curved shape, it can compute the rest of the points and draw the shape automatically.

The DMP-40 includes other advanced features, like the ability to isolate part of a figure, then reproduce it larger or smaller in the same proportion as the original.
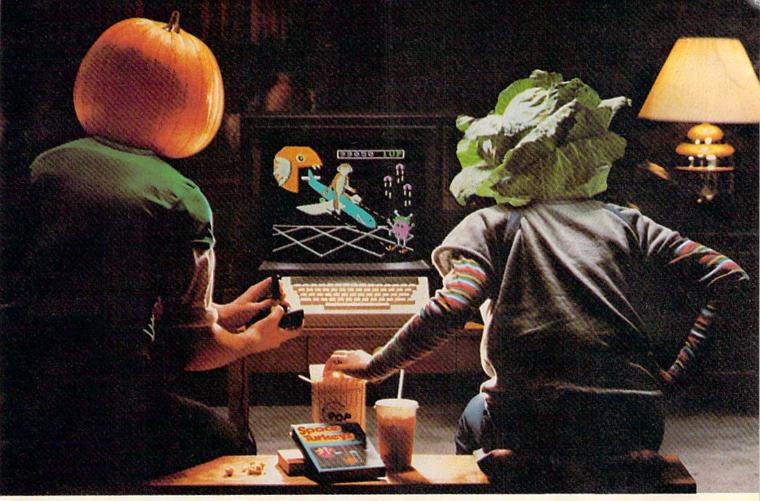
This difference in sophistication has a direct bearing on how much math the user must know and how much programming is required to make the plotter work. Since the DMP-40 has a built-in computational ability, the user is primarily concerned with learning the commands that tell the plotter the general shapes in the design, and where to place the design on the page. These commands are then sent to the plotter as character strings in BASIC or machine language programs.

With the 1520, the user also must work out the formulas, and write a program to calculate the points and transmit the results to the plotter.

Some features do not depend on price. Both the 1520 and DMP-40 can draw in color, for example. And both have a problem with pen quality.

The 1520 uses special ballpoint pens which draw a thin, uneven line, and which tend to skip. The DMP-40 uses special felt-tip pens with soft points that are quickly blunted as the

You bought a computer to cultivate your kids' minds. Make sure it's bearing fruit, not growing vegetables.

Introducing a whole crop of Learning Adventure games from Spinnaker.

When it comes to cultivating adventurous young minds, the computer's potential is endless.

Unfortunately, the search for software that makes the most of that potential has been endless, too.

That is, until Spinnaker created the Learning Adventure Series. A unique collection of games that reward curiosity with

hours of adventure and learning. So the time kids spend with our games will help them develop valuable skills. Instead of just tired thumbs.

But what really makes our Learning Adventure games unique — educational value aside — is how much fun they are. Which isn't too surprising when you consider you can do things like bargain with aliens, search a haunted house, or build your own railroad empire.

In fact, our games are so much fun, kids will really enjoy developing some very important skills. Deductive reasoning, note taking, and problem solving, for instance.

So, if you're in the market for software that will truly cultivate young minds, pick the Spinnaker Learning Adventure Series.

It's the best way to be sure your search will be fruitful.

Spinnaker Learning Adventure games are available for Apple,® Atari,® IBM® and Commodore 64™ home computers.

It's New! TRAINS.™
You're in charge of an old-time railroad — and whether it turns into a bonanza or a bust depends on how well you run it. But either way you'll find that working on this railroad is a challenge — and a lot of fun! **Ages 10-Adult.**

It's New!
ADVENTURE CREATOR.™
Design a challenging adventure game that you or a friend can tackle — or let the computer design one for you. It's complex, exciting — utterly addictive! **Ages 12-Adult.**

IN SEARCH OF THE MOST AMAZING THING.™
It isn't easy to find — even in your B-liner. But you'll have help from your Uncle Smoke Bailey as you search the universe to find the Most Amazing Thing. **Ages 10-Adult.**

SPINNAKER™
We make learning fun.

Disks for: Apple, Atari, IBM, and Commodore 64.
Cartridges for: Atari and Commodore 64 — (ADVENTURE CREATOR only).

www.commodore.ca

pen is dropped on the paper, lifted, and dropped again. Both of these problems are most noticeable when the plotters are printing text because the plotter uses many short, jerky motions to form the letters, and picks up and drops the pen after drawing each letter.

However, interfacing is one area where the 1520 has the DMP-40 plotter beat—and any other plotter, for that matter. Since the 1520 is made by the same company that manufactures the VIC and 64, it is fully compatible with your computer. Just plug it into the serial port as if it were a Commodore printer.

Other plotters (unless they come with a special interface) are usually connected to the computer through an interface cartridge plugged into the user

port (RS-232). This is not a simple task.

First, the use of plotters with Commodore home computers is not widespread. Manufacturers are not familiar with the user port's wiring, so they can't help much with interface questions. Hooking up a third-party plotter requires the user to know at least the basics of how the Commodore user port, the interface cartridge, and the plotter's RS-232 port are wired, and the control codes used to transmit and receive data.

In addition, there is a bug in Commodore's RS-232 handshaking routine, that portion of the operating system which allows the computer to communicate automatically with devices on the user port. Because of this bug, the computer sometimes doesn't send all its data to the RS-232 device.

Even after you answer the wiring and signal questions, you must include routines in your plotter programs which bypass the bug. The only sure way to deal with the problem is to keep checking the RS-232 output buffer to make sure it's empty after each batch of data is sent, and if it's not empty, to send null bytes until it empties. You also must be familiar with the plotter's input buffer size, how fast data moves through it, and how the plotter signals that it's ready to receive more data.

All of these technical and programming obstacles can be daunting to the beginner, and help is not easy to find. Still, if you have a practical need to produce hard copy of high-resolution, computer-generated graphics, the result is well worth the work. ⬛

www.commodore.ca

A Survey Of Printers For The VIC-20 And Commodore 64

Kathy Yakal, Editorial Assistant

I f you've ever browsed for printers at a store where the clerks were less-than-knowledgeable and more-than-technophobic, you may be a little nervous about buying a printer that will "work" with your VIC-20 or Commodore 64.

Granted, you might have to do a bit of investigation to get the right kind of interface (see "Selecting A Printer Interface," elsewhere in this issue) if you're interested in a non-Commodore printer. And if you already own a lot of software, you will want to check on the compatibility of those packages with print capabilities.

But don't be put off by anyone who tells you you have to be pretty technical to hook up anything but a Commodore printer to a VIC-20 or Commodore 64. The fact is that your Commodore computer can work with any printer with a Centronics parallel or RS-232 serial interface capability.

The following chart is limited to printers that cost less than $500. There are more expensive printers that can be used with Commodore comput-

Cardco, Inc.'s newest printer, the LQ/3, offers letter quality print for less than $500.

ers, but it's not likely that you'll want to spend more on a printer than you paid for your computer and disk drive put together.

We have tried to be as comprehensive as possible in these listings. If any manufacturer has been left out, we regret the omission.

Here's an explanation of the chart specifications:

Manufacturer/Distributor: In some cases, this is actually the company that makes the printer. In others, it's the company that markets or distributes it.

Type of Interface Standard: Commodore computers are equipped to communicate with

"Now Your Commodore 64™ Can Print Like a Pro!"

Grappler™ CD
Printer Interface

The Revolutionary Printer Interface for the Commodore 64™

A New Era in Commodore Printing Power.

Grappler CD offers the first complete answer to your printer interfacing requirements, with many powerful capabilities unique in the Commodore marketplace. Complete signal translation allows many popular name brand printers to operate perfectly with the Commodore 64, or to imitate Commodore's own printer. Even Commodore's graphic character set can be reproduced on Epson, Okidata, Star, ProWriter and other popular printers.

Exclusive Grappler CD features provide a variety of graphic screen dumps, text screen dumps and formatting. No other Commodore interface can offer this.

If you own a Commodore 64...

If you're serious about quality, trouble free printing... You need the Grappler CD.

Contact your nearest Commodore dealer or call Orange Micro for a dealer near you.

A Uniquely Intelligent Interface:

- Prints Screen Graphics Without Software
- Graphics Screen Dump Routines Include Rotated, Inversed, Enhanced and Double Sized Graphics.
- Full Code Translation From Commodore's PET ASCII to Standard ASCII, the Language of Most Printers.
- Complete Emulation of the Commodore 1525 Printer for printing of Commodore's Special Characters.
- Dip Switch Printer Selection for Epson, Star, Okidata, ProWriter and other popular printers.
- Conversion Mode for Easy Reading of Special Commodore Codes.
- Text Screen Dump and Formatting Commands
- 22 Unique Text and Graphics Commands

Orange Micro inc.

1400 N. LAKEVIEW AVE., ANAHEIM, CA 92807 U.S.A.
(714) 779-2772 TELEX: 183511CSMA

© Orange Micro, Inc., 1983

Commodore 64 and Commodore 1525 are trademarks of Commodore Electronics Limited.
Epson is a registered trademark of Epson America, Inc.

www.commodore.ca

IT'S YOUR TIME AND BOTTOM LINE!

COMMODORE OWNERS now have an alternative to the frustrating LOAD ... wait ... UNLOAD ... wait ... RELOAD ... wait procedure. The NEW MSD DUAL DISK DRIVE ... SUPER DISK II will eliminate those hours wasted during task performance and time lags that inflate expensive labor costs and deflate your bottom line.

One SUPER DISK II will copy, format, and verify in less than two minutes ... a procedure that normally takes 30 to 40 minutes using 2 single drives. Yes, SUPER DISK II is TWENTY TIMES FASTER!

MSD provides quality craftsmanship that is backed by an exceptional six month warranty. The SUPER DISK II workhorse drive never overheats, so why be aggravated by the loss of time and money?

Invest today in SUPER DISK II, the SMART disk drive which contains its own microprocessor and memory enabling the computer to command the drive and then continue. SUPER DISK II does the work quickly so that you can get on with your program.

As for the money you will save in unproductive labor costs ... perhaps you will invest it in a daisychain of SUPER DISKS. These workhorse drives will allow you to connect up to four drives for maximum speed and reliable, continued performance.

SUPER DISK II DESIGN ASSETS

- Sturdy sheet metal casing and heat sink
- Vertical load for ergonomic space efficiency
- Key latch secures disk position
- Smart disk contains microprocessor and memory in each drive
- Status lights to display which drive is in use
- Three steel bands for repeated accuracy of alignment
- Flexible use of serial or parallel communication
- Access flexibility configured as drive 0 or drive 1

MSD
SYSTEMS, INC.

Dealer and distributor inquiries invited.

Good graphics capabilities, a variety of character sizes, and a speed of 120 cps make the Gemini 10X, from Star Micronics, a popular printer for Commodore owners.

Blue Chip Electronics, Inc., has introduced the M120/10, a correspondence-quality dot matrix printer that interfaces directly with the Commodore 64. An RS-232 serial interface is optional.

printers through their serial ports. If you want to buy any kind of printer other than one that is "Commodore-ready" (one that can be connected directly to the Commodore serial port), you will have to buy an interface.

Basically, there are two kinds of interfaces. One translates Commodore serial into standard RS-232 serial; the other turns Commodore serial into standard Centronics parallel.

Some manufacturers offer both serial and parallel versions of a particular printer.

This column on the printer chart tells what interface(s) is standard and what, if anything, is optional.

Print Technology: In this price range, there are two types: *impact* and *thermal*.

Impact printers transfer characters to paper by actually making contact with it. *Dot-matrix* printers form characters and graphics with a set of wires that strike the paper in the desired configuration. *Daisywheel* printers have a printhead that contains fully-formed characters like those in a typewriter, except that the characters are arranged in a circle like the petals of a flower—hence the name daisywheel. Both use inked ribbons.

Daisywheel printers generally offer better print quality, while dot-matrix printers' biggest selling points have traditionally been low cost and graphics capabilities. Not necessarily so for much longer, though: Some dot-matrix printers now approach letter quality printing, and daisywheel printers can be had for less than $500.

Thermal printers burn off a special aluminum coating on thermal paper so the black surface under the coating will show through in the pattern of desired characters. This technology is quiet and cheap, but the special paper required often ages

READER'S DIGEST INTRODUCES PLAY-AND-LEARN SOFTWARE FOR A VERY TOUGH AUDIENCE.

In the eyes of any red-blooded, American four-five-or-six-year-old, computer software is nothing more than a new toy. And when it comes to a new toy, kids have never been known to fool around. It's either good or bad, fun or boring, worth playing with or not, right from the start. Nothing in between and no bones about it.

So when we created our early learning series, we did it without any illusions. And, judging from the kids who've tried our games so far, there's not a dull learning tool in the group.
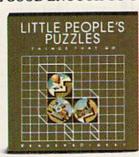
Each game is fun. And worth playing over and over. So our programs continually reinforce early learning skills along with basic computer commands. And unlike some other early learning programs, Reader's Digest Software games are designed so that even nonreaders can follow the fun.

Whether it's Little People's Puzzles™ for pleasure, Micro Habitats,™ the creative construction set, or kooky, unspooky Alphabet Beasts and Company,™ there's a lot of color, animation, sound and even music to turn our toughest customers into a captive audience. Look for it now at your software store or call Customer Service at 1-800-431-8800 (NY: 1-800-262-2627; AK, HI: 914-769-7000; Canada: 514-934-0751).

SOFTWARE GOOD ENOUGH TO GO OUT AND BUY A COMPUTER FOR.
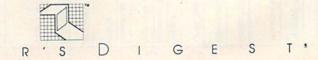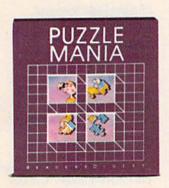
Little People's Puzzles, Alphabet Beasts and Company and Micro Habitats are available for Apple® II, II Plus, IIe, IIc: 48 K with Disk Drive, Commodore® 64 Machine with Disk Drive, IBM® PC and PCjr.™ Color monitor required for Little People's Puzzles, preferred for Alphabet Beasts and Company. Color monitor and joystick preferred for Micro Habitats.

R E A D E R ' S D I G E S T ™

www.commodore.ca

Model Name	Manufacturer/Distributor	Type of Interface Standard	Print Technology	Speed	Pitch	Buffer	True Descenders	Feed Type	Max. Paper Width, In.	Suggested Retail Price
Abati LQ-20	Micro D	Parallel Standard; serial optional	Impact (daisywheel)	18 cps	120–180 cpl	1.5K	Yes	Friction standard; pin optional	13	$479
Alphacom 42	Alphacom, Inc.	Parallel and serial cables available; also Commodore, Atari, TI	Thermal	80 cps	10 cpi	One line	Yes	Friction	4.5	$119.95
Alphacom 81	Alphacom, Inc.	Parallel and serial cables available; also Commodore, Atari, TI, Apple	Thermal	80 cps	10 cpi	One line	Yes	Friction	8.75	$169.95
Cardco LQ-2	Cardco, Inc.	Parallel standard; built-in interface for Commodore computers	Impact (daisywheel)	12–20 cps	Max. 80 cpl	One line	Yes	Friction	8.7	$349.95
Commodore 1526	Commodore Business Machines	Commodore-ready	Impact (dot-matrix)	45 inches per minute	80 cpl	One line	Yes	Friction and pin	8.5	under $300
CP-80 Type 1	Everett/Charles Marketing Service, Inc.	Parallel standard; serial optional	Impact (dot-matrix)	80 cps	40–142 cpl	None	Yes	Pin and tractor	10	$275
Dataport	Dataport	Parallel	Impact (daisywheel)	15 cps	80 cpl	None	Yes	Friction	8.5	$295
Epson MX-80	Epson America, Inc.	Parallel	Impact (dot-matrix)	80 cps	80 cpl	One line	Yes	Friction and pin	10	$494
Facit 4510	Facit Data Products	Both parallel and serial	Impact (dot-matrix)	120 cps	10–17 cpi	2K	Yes	Friction and pin standard	11	$495
Gemini 10X	Star Micronics	Parallel standard; serial optional	Impact (dot-matrix)	120 cps	6–17 cpi	4K or 8K	Yes	Both friction and pin	9.5	$399
IT-4010	Blue Chip Electronics	Serial and parallel standard; no special interface required for Commodore 64	Thermal transfer	120 cps	10–15 cpi	256 bytes	Yes	Both pin and friction	9.5	$399
KX-P1090	Panasonic	Parallel standard; serial optional	Impact (dot-matrix)	80 cps		1K standard; 4K optional	Yes	Pin and friction	10	$399
KX-P1091	Panasonic	Parallel standard; serial optional	Impact (dot-matrix)	120 cps		1K standard; 4K optional	Yes	Pin and friction	10	$499
Legend 800/1000	Legend Peripheral Products	Parallel standard; serial optional	Impact (dot-matrix)	Legend 800: 80 cps; Legend 1000: 100 cps	40–142 cpl	1K	Yes	Friction and pin standard	10	800:$349 1000:$359
LQ-3	Cardco, Inc.	Parallel; built-in Commodore interface	Impact (daisywheel)	13 cps	N.A.	N.A.	Yes	Friction standard; pin optional	N.A.	$449.95
M120/10	Blue Chip Electronics	Serial and parallel standard; no special interface required for Commodore 64	Impact (dot-matrix)	120 cps	10–15 cpi	256 bytes	Yes	Both pin and friction	9.5	$349
Microline 80	Okidata	Parallel	Impact (dot-matrix)	80 cps	80–132 cpl	None	Yes	Pin and friction; tractor optional	9.5	$449
MPS-801	Commodore Business Machines	Commodore-ready	Impact (dot-matrix)	50 cps	80 cpl	One line	No	Pin	8.5	under $300
NEC PC-6021	NEC Home Electronics (USA), Inc.	Parallel	Thermal	40 cps	40 cpl	None	No	Friction	4.5	$249.95
NEC PC-8023A	NEC Home Electronics (USA), Inc.	Parallel standard; serial optional	Impact (dot-matrix)	100 cps	80–136 cpl	2K	Yes	Both pin and friction	10	$499
Printelex	Computer Peripherals	Parallel and serial standard; interface cables available for Commodore, IBM PC, Radio Shack	Impact (dot-matrix)	160 cps	40 cpl	One line	No	Friction only	4.25	$145
Prowriter 8510-AP	C. Itoh	Parallel	Impact (dot-matrix)	120 cps	10–17 cpi, 80–136 cpl	1K	Yes	Both friction and pin	13	$495
Seikosha GP-100A	Axiom Corporation	Parallel and serial	Impact (dot-matrix)	64 cps	32 cpl	None			10	$389
Seikosha GP-250X	Axiom Corporation	Parallel and serial available	Impact (dot-matrix)	50 cps	80 cpl	80 bytes			10	$499
Thin-Print 80	Axonix Corporation	Parallel and serial available	Thermal	40 cps	80–136 cpl	2K	No	Friction feed	8.5	$279
Thinkjet (HP2225)	Hewlett-Packard	Parallel, HP-1B, and HP-IL available	Ink-jet	150 cps	40–142 cpl	1000 bytes	Yes	Pin and friction	9.5	$495

Major Printer Manufacturers And Distributors

If you are interested in finding out more about a particular printer, it's best to check with a local computer dealer first. If they don't have the information you need, contact the manufacturer or distributor listed here.

Alphacom, Inc.
2323 S. Bascom Ave.
Campbell, CA 95008

Apple Computer
20525 Mariani Ave.
Cupertino, CA 95014

Axiom Corporation
1014 Griswold Ave.
San Fernando, CA 91340

Axonix Corporation
417 Wakara Way
Salt Lake City, UT 84108

Blue Chip Electronics
7406 E. Butherus Dr.
Scottsdale, AZ 85260

CAL-ABCO
Legend Peripheral Products
14722 Oxnard St.
Van Nuys, CA 91401

Cardco, Inc.
300 S. Topeka
Wichita, KS 67202

Commodore Business
    Machines
1200 Wilson Dr.
West Chester, PA 19380

Computer Peripherals
6400 Canoga Ave.
Suite 305
Woodland Hills, CA 91367

Comrex
3701 Skypark Dr.
Torrance, CA 90505

Dataport
Computer Products Research
    Facility
5525 Olinda Rd.
Bldg. A
El Sobrante, CA 94803

Epson America, Inc.
3415 Kashiwa St.
Torrance, CA 90505

Everett/Charles Marketing
    Services, Inc.
6101 Cherry Ave.
Fontana, CA 92335

Facit Data Products
235 Main Dunstable Rd.
Nashua, NH 03060

Fujitsu America, Inc.
3055 Orchard Rd.
San Jose, CA 95134

Hewlett-Packard
3000 Hanover St.
Palo Alto, CA 94304

Integral Data Systems
Milford, NH 03055

Leading Edge
225 Turnpike St.
Canton, MA 02021

Micro Peripherals, Inc.
4426 S. Century Dr.
Salt Lake City, UT 84123

NEC Home Electronics
    (U.S.A.), Inc.
Personal Computer Division
Elk Grove Village, IL 60007

Okidata
Mt. Laurel, NJ 08054

Panasonic Company
One Panasonic Way
Secaucus, NJ 07094

Star Micronics
200 Park Ave.
Pan Am Building
New York, NY 10166

Swintec Corporation
23 Poplar St.
P.O. Box 421
East Rutherford, NJ 07073

Tandy Corporation/Radio Shack
1800 One Tandy Center
Fort Worth, TX 76102

Transtar
P.O. Box C-96975
Bellevue, WA 98009

---

quickly. Some new *thermal transfer* printers do not require special paper.

Speed: How fast the printer prints, usually measured in characters per second (cps).

Pitch: How many characters per inch (cpi) or characters per line (cpl). This will vary in printers that are capable of printing different sizes of characters, like double-wide and compressed.

Buffer: Most printers can "hold" a certain amount of text while printing what's directly ahead of it. In printers of this price range, this buffer is not significant, averaging around one line (80 characters).

The advantage of having a larger buffer is that it frees up the computer for use while the printer is printing.

True Descenders?: An important consideration if you're going to use your printer for anything more than informal home applications. Do the "tails" on lower-case letters like q, y, and p actually descend below the line?
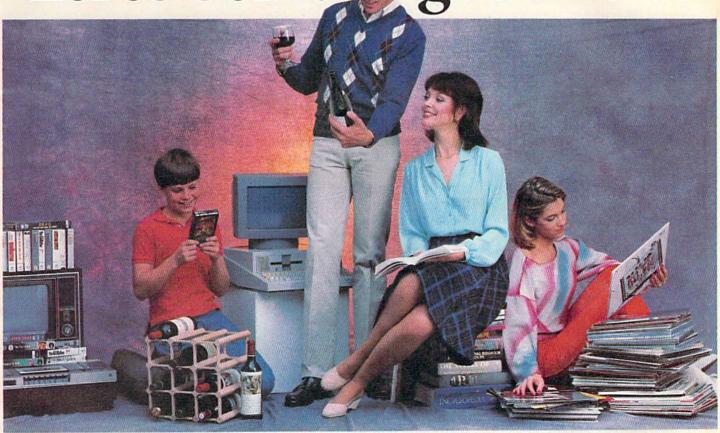
Feed Type: Two kinds are available. *Friction* feed is similar to the way typewriters hold paper. You can either use single sheets or continuous-feed rolls.

Tractor feed printers (sometimes called *pinfeed*) grip special perforated-edged paper with little toothed wheels on either end of the platen. Tractor-feed paper is 9.5" wide, standard 8.5 × 11" paper with an extra ½" on either side for the tear-away perforations.

Maximum Paper Width: You may never have occasion to use anything other than standard typewriter or pinfeed paper. Some printers, though, accommodate wider paper, and a few aren't wide enough for regular paper.

Suggested Retail Price: Manufacturers' suggested price at press time. Prices may vary from dealer to dealer. Shop around. ⬛

C=www.commodore.ca

If you're having a hard time getting organized, here's some good news.

The Home Cataloger.™

With The Home Cataloger, you can set up a complete home inventory system to cross-file your wine, book and record collections. Or even keep your favorite recipes.

Remember birthdays, anniversaries and your Christmas card list effortlessly. You won't even have to try to recall who sent you a Christmas card last year—The Home Cataloger will tell you.

And since it's such a flexible and easy to use program, you'll be able to put it to work almost immediately. Even if you're new to computers.

To help you out, The Home Cataloger comes with 10 ready-to-use cataloging formats for some of the most common uses: telephone list, inventory, travel plans, restaurants, insurance policies, coins, growth (height and weight), running, studies, and book list. All you have to do is plug in your own information. If you want additional categories, just add them on—simple as that.

Your imagination is the only limit to

More ways to use The Home Cataloger:

Organize your stamp collection according to country, commemoratives, people, places, mint, cancelled, or first day categories.
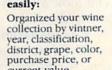
Catalog your coin collection using categories such as date, mint, country, type of coin, or condition.

And, find things easily:

Organized your wine collection by vintner, year, classification, district, grape, color, purchase price, or current value.

Videotapes can be filed in a number of ways: comedy, drama, mystery, adventure, documentary, stars, director, or year made.

ways you can use The Home Cataloger.

Stop by your nearest dealer and check out The Home Cataloger. You'll be amazed at all the different ways The Home Cataloger will work for you. You'll also be amazed at the low price.

If fact, you'll probably wonder what you ever did without it.

Available for: Apple II/IIe, Commodore 64, IBM PC/XT and PCjr.

Price: from $49.95.

For your free 64 page booklet, "Tips for Buying Software," and complete product catalog, please write Continental Software, Dept. GAZ, 11223 S. Hindry Avenue, Los Angeles, CA 90045, 213/410-3977.

The Home Cataloger is a registered trademark of Continental Software. Apple II/IIe, Commodore 64, IBM PC/XT and PCjr are registered trademarks, respectively, of Apple Computer, Inc., Commodore Business Machines, Inc. and IBM Corp.

Continental
Software
A Division of Arrays, Inc.

www.commodore.ca

Selecting A Printer Interface

J. Blake Lambert, Assistant Editor

Perhaps the easiest way to set up a printer system for your VIC-20 or Commodore 64 is to purchase a Commodore printer. This way, you can just plug it in and have the special Commodore graphics symbols and reverse-video characters. But if your needs are different—if you want to print foreign language characters, use a wide variety of printing fonts and styles, define a large special character set, print scientific symbols,—you may decide to purchase another brand of printer. To use a non-Commodore printer with a VIC or 64, you'll probably need an interface. Some printers have built-in interfaces, but most do not.

Most non-Commodore printers need an interface to work with a VIC or 64 because the computers send signals in a form the printers can't understand. The VIC and 64 communicate with peripheral devices (like printers and disk drives) through the serial port—all data is sent and received over one wire. Think of the cables from the computer to the interface as a highway. The cars on this highway are the individual pieces of data the computer is sending to the printer—each one represents a *bit* of data. The computer sends out the signals in groups of eight, called *bytes*; each byte represents one character. We can think of the bytes as eight-car caravans.

While the serial line from the computer is like a one-lane road, most printers receive data through a parallel port, which is like an eight-lane highway. That is, the printer wants to receive all eight bits of the data at once over eight separate wires. The interface is the junction where the number of lanes increases from one to eight. At this point in the road, each car chooses a different lane; the cars continue travelling as a group, but now they are side-by-side. So, while the computer sends out the eight bits of data that make up a byte sequentially (one after the other), they arrive at the printer side-by-side and at the same time.

Interfaces are tools used for communication between your computer and printer. And they allow you to print your work in the form you want. Like wrenches, interfaces come in many different forms. Some are adjustable and some are not. Some are designed for special purposes, and therefore are not as versatile. The key to selecting the right tool, though, is knowing what you plan to use it for. The most common features found in interfaces for the VIC and 64 are transparent interfacing, emulation, text handling, listing, and special ROM (Read-Only Memory) character sets. Let's look at each of these in greater detail:

• **Transparency.** This is a standard feature of printer interfaces. It converts the serial data to parallel data without altering any values. This is sometimes called the *graphics* feature, since it is used most often to print high-resolution graphics. It allows you to access the special character sets of your printer, and to send codes to your printer to make it print special fonts or styles. Often this mode can be used just after power-up to put the printer into a special state (for example, emphasized print or double-strike) before listing a program or printing a memo. In addition, some word processors recommend using the transparent mode to be able to access all of the features of your printer. A slight drawback to transparency is that if your computer is sending out Commodore ASCII and the printer expects standard ASCII, the results may not be what you expect. Capital and lowercase letters may be switched, for example. You can usually avoid this problem with a short conversion program.

• **Emulation.** This is a must for any printer interface unless you plan to never use software written by others for your Commodore. Most commercial software written for the VIC and 64 assumes that you will be using a Commodore printer. If you have a different printer and your interface does not provide emulation, some strange results can occur. Emulation allows your printer to pretend it's a Commodore 1515 or 1525 by translating the codes sent from the computer into the codes that will work with your printer. This is similar to translating a passage in British English to American English: Most things remain unchanged, but a few need to be

translated.

• **Text.** This function varies from one interface to another. In general, this function prints text normally, and handles special codes differently. Some interfaces print special mnemonic representations of these codes (see "Listing" below); others ignore the nontext data altogether.

• **Listing.** Most interfaces offer a special mode for listing your programs, so you can see where things like cursor moves and color changes are included in the program. If it were not for the listing mode, some character combinations would trigger the special features of the printer; you might suddenly go into another print mode, or even lock up in the middle of a listing. (Selecting the wrong mode is an easy mistake to make, so SAVE your work before you print it. This way, you won't lose the program if the system locks up.)

• **Special ROM characters.** The simplest printer interfaces do not provide a method of printing the Commodore graphics characters on other printers, but more expensive interfaces do. They are able to do this because the graphic character set is permanently stored in ROM in the interface. This way you can access the Commodore characters at times, then switch modes and still be able to access the standard character set in the printer's ROM (or even access a RAM set that you have defined beforehand). Note that not all interfaces that offer Commodore graphics allow you to print reverse-video characters as well. One interesting application of special ROM characters is the "correspondence quality" printing mode found on the lowest-priced XETEC (pronounced "z-tek") interface described later.

With all these features to choose from, it's easy to see that choosing a printer interface can be as difficult and as

important as choosing a printer. But if you know how you plan to use your printer, the decision will be easier. If you plan to do mostly word processing and don't mind using the graphics characters in your printer's ROM, a simple interface will do. Some printers already have characters which closely resemble many of the Commodore graphics characters, and some allow you to define as many as 96 *downloadable* characters. (These characters are created by sending codes to the printer, redefining the pattern of dots for one or more characters. The printer then places the new pattern of dots where it would have put the standard character.) Remember that most word processors which allow you to select a specific printer from the menu (and most programs with special graphics) use the transparent feature anyway.

On the other hand, if you want to print Commodore graphics with no hassles, or if you want special features like built-in screen dumping, you will need to get a more expensive interface. The key to making the right decision lies in knowing what you want, what is available, and what you can afford. If at all possible, see the printer and interface at work together before making a purchase. Also, consider whether or not the interface manufacturer offers a trade-up policy, and investigate what software is available for use with the interface. Some simple interfaces are able to print Commodore graphics with the proper software.

There are many good interfaces on the market specifically for the VIC and 64. For many VIC and 64 owners, the most practical option is to purchase an interface in the $50 to $60 range. These are the simple interfaces which don't provide Commodore graphics printing without additional software.

Perhaps the best known of these interfaces is the CARDCO CARD/?A (pronounced "card-print"). The CARD/?A has recently been replaced by the CARD/?B, which performs the same functions but is in a smaller housing. The CARD/?A is a versatile interface which provides listing mode, emulation, transparent mode, and allows for sending or omitting automatic line feed codes.

Mode selection is easy from the keyboard or within a program, and the interface works well with word processing programs in both the emulation and graphics modes. To list a program in upper/lowercase with line feeds added, for example, you simply

OPEN4,4,6:CMD4:LIST

The 6 in the OPEN command is a secondary address. This interface checks the secondary address to determine what mode to use. It also allows you to lock it into a mode, and stay in that mode until you turn off the power. After the listing is complete,

PRINT#4:CLOSE4

will redirect output to the monitor screen.

The manual for the CARD/?A covers the basics of printing, including a section on sending control characters to the printer to initiate special print features. It also contains a few short programs including a subroutine that allows you to print a Commodore graphics character if you wish. CARDCO also markets an excellent printer utility program for use with the CARD/?A, which may work with other simple interfaces. CARDCO says that the utility program, which provides screen dump functions, Commodore graphics, and banner poster printing, will work with the CARD/?B and the CARD/?+G.

The CARDCO and XETEC interfaces draw power from the cassette port. However, this

doesn't keep you from using the cassette unit because CARDCO and XETEC provide a special plug for tape users. If you don't plan to use a tape unit, though, you should wrap the contacts on the special cassette port plug to prevent damage to the computer from accidentally shorting the connections. Also, if you plan to use the interface with the portable SX-64 computer, the CARDCO and XETEC products will not work in their present forms because the SX-64 has no cassette port.

One of the major complaints made about dot-matrix printers is the print quality. Options like double-strike and emphasized print help, but many still feel this is not good enough. XETEC attempts to resolve this problem with what they call "correspondence quality" dot-matrix print.

The "correspondence quality" print mode, which works with Epson and Gemini printers, uses ROM characters within the interface to perform double pass printing. But it is not the same as double-strike because the dots that form the character on the second pass are not in the same configuration as those in the first pass. Between passes, the platen shifts upward 1/144 of an inch, so that the final character is a composite of the two dot patterns.

The XETEC interface has essentially the same capabilities as the CARD/?B, and XETEC offers an optional 2K print buffer ($10 extra, installed). A more expensive XETEC interface offers the 2K buffer as standard, and adds Commodore graphics and reverse graphics capabilities, but does not offer the "correspondence quality" option. Both XETEC interfaces offer transparent, emulation, and text-handling options.

Three popular interfaces which will print Commodore graphics are the CARDCO CARD/?+G, the Tymac Connection, and the Orange Micro Grappler CD. Each has special capabilities, so the choice is yours. All of these interfaces have transparent, total emulation, and text-handling capabilities.

The CARD/?+G is similar in appearance to the CARD/?A, but it has ROM within to provide Commodore graphics, including reverse-video characters. Printing graphics using the interface is not as easy as it first would seem, though, and many times the spacing of a printout will not match that of the original on-screen representation. This is because the CARD/?+G leaves extra space between the graphics characters with some computers. So, if you print a picture with writing in it, the results are likely to be poor, unless you find a way to correct for the spacing problem.

The CARD/?+G has several nice features not found on the CARD/?A, though. Program listings are easier to generate, because the listing feature is incorporated into the text modes. This reduces the number of times you need to OPEN and CLOSE channels and change secondary addresses to select the mode you need.

Another helpful feature of the CARD/?+G is the special mode that will print all characters sent through the interface as their hexadecimal equivalent. This is an especially useful feature for machine language (ML) programmers, since ML programs are generally assembled into hex code. The CARD/?+G also allows the use of the special commands available with the VIC 1525 printer, including dot-addressable graphics.

The internal DIP switches of the CARD/?+G can easily be reset to allow changing the device number of the printer, locking the interface in the "no ASCII correction" mode, and selecting automatic line feeds.

Some of the switches tell the computer what printer is being used, and these should be set before printing the first time. The DIP switches also insure that if you need to use the interface with another compatible printer, you will get good results.

Like the CARD/?+G, the Tymac Connection contains a ROM set of characters for producing Commodore graphics. In addition, the spacing between the graphics characters is better. The Connection does not allow printing reverse-video text characters, although it does print Commodore graphics characters in reverse-video.

Giving up the reverse-video characters has its advantages, though, as the Connection has an internal 2K buffer, which holds data until the printer is ready. In many cases, this will free up the computer sooner, so that you can go on using the computer while the printout finishes.

The Connection draws its power through the printer connection, so it will work with the SX-64. This method of drawing power is also less awkward in terms of the physical arrangement of your computer system's components.

Unlike the CARD/?+G, the Tymac Connection is *printer specific*. This means you need to order the Connection that contains the ROM chip for your particular printer. Tymac sells replacement ROM chips, in case you later change printers.

The manual for the Connection is brief but thorough. Short programs illustrate the features of the interface and explain the additional commands that are available in the emulate mode. These commands allow you to examine and change the device number of the printer, skip the perforation on the paper, and set left and right margins.

The Connection also has a

Share the Olympic Experience.

"Standing on the top of the Olympic victory stand is like stretching one's body on the top of the world. It is a moment where the individual man or woman gets introduced to the whole planet. It is a moment that is his or hers alone."

—Olga Connolly
Gold Medalist, 1956 Olympics

This summer, the Olympic torch will return to Los Angeles after 52 years. The stage is set. Some 10,000 athletes from 150 countries will battle for the gold in the historic Games of the XXIII Olympiad. And whether or not you plan to attend the Games, you can participate in the drama of this once-in-a-lifetime spectacle.
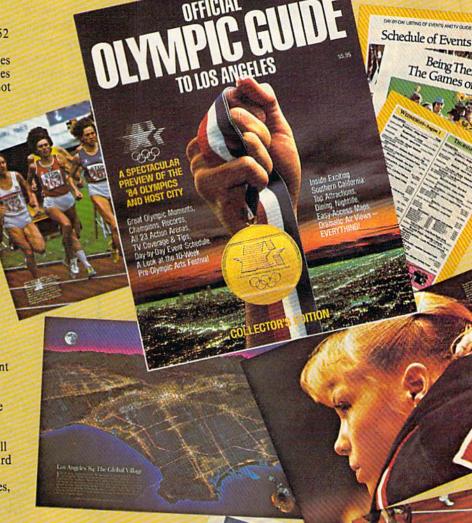
You and your friends will have virtual front-row tickets with the *Official Olympic Guide to Los Angeles*, a lavish 300-page magazine that captures it all—the thrills of great Olympic moments past and present, future stars and hopefuls, records to break, maps, TV and day-by-day events schedules.

Everything from the magic of opening ceremonies to the final lap of the marathon, along with an indispensable guide to Southern California, including top attractions, dining, nightlife, tips and trips. All in breathtaking full-color photography and illustrations.

You, and your out-of-town friends, will share all the action of this memorable event with this special collector's issue that will become a cherished keepsake of the legendary rekindling of the Olympic flame in Los Angeles.

Order now and be ahead of the crowd. Send in your check or money order. Or call 557-7587 and charge it to your MasterCard or Visa. Official Olympic Guide, 1888 Century Park East, Suite 920, Los Angeles, California 90067.

OFFICIAL OLYMPIC GUIDE TO LOS ANGELES

$5.95

A SPECTACULAR PREVIEW OF THE '84 OLYMPICS AND HOST CITY

Great Olympic Moments, Champions, Records, All 23 Action Arenas, TV Coverage & Tips, Day-by-Day Event Schedule, A Look at the 10-Week Pre-Olympic Arts Festival

Inside Exciting Southern California: Top Attractions, Dining, Nightlife, Easy-Access Maps, Dramatic Air Views — EVERYTHING!

COLLECTOR'S EDITION

Schedule of Events

Being There: The Games on TV

Los Angeles 84 The Global Village

Track & Field Layout of 1984

OFFICIAL OLYMPIC GUIDE TO LOS ANGELES

Games of the XXIIIrd Olympiad Los Angeles 1984

Send me _____ copy/copies of the Official Olympic Guide to Los Angeles at $5.95 each, plus $1.00 shipping and handling. Enclosed is my check or money order.

NAME

ADDRESS

CITY

STATE

☐ Yes! I'd like the beautifully designed gift box for an additional $1.00. Please send me _____ boxes. I've enclosed an extra $ _____ .

ready light, a reset button, and a printer test. The reset button allows you to recover from a system lock-up without having to turn the computer off.

Orange Micro's Grappler CD is a sophisticated interface for the 64 only, which performs the transparent, listing, and text-handling functions and has special ROM characters. The Grappler CD plugs into the expansion port of the 64, but provides its own expansion port, so games can use the port as well.

The graphics generated by the Grappler CD are better than those on any of the other interfaces we've seen. The interface removes the space between the graphics characters when printing them, and there is also a text screen dump available, which leaves the spaces in, so the letters don't run together. The Grappler CD does not print reverse-video graphics, but prints a normal graphic character wherever a reverse one should be.

The Grappler CD is controlled by DIP switches and OPEN statements like other interfaces, but also allows you the option of sending commands in PRINT statements using what are called "Control-A" commands. The Grappler CD responds to Control-A commands rather than sending them to the printer. This way, many text formatting and graphics commands are available. For example, the Grappler CD allows you to fill the monitor screen with text or graphics, then print an inverse, rotated double-size image of the screen. Few if any other interfaces allow you to do so much so simply. The interface is easy to use and the accompanying manual is excellent.

If you often share programs with friends, you should each consider purchasing similar interfaces if possible. Since the Grappler CD uses commands embedded in PRINT statements,

the special functions are only available to Grappler CD users. Programs written to use with other interfaces will work fine with the Grappler CD. But if you use the Control-A commands in your programs, you will need to remove them before running those programs with the CARDCO interface, for example. For this reason, the Control-A commands will be most helpful when used from the keyboard or in your personal programs.

Machine language programmers can try out their skill with the Grappler CD easily, since the manual contains an Appendix of ML entry points. This way, you could add a section of code in your ML program to perform a graphics screen dump (by loading the accumulator and X-register with the proper values, then JSRing to the starting address of the graphics screen dump subroutine).

The Grappler CD does have

www.commodore.ca

STAR WARS:

THE ARCADE GAME

products are BusCard II (see review elsewhere in this issue), INTERPOD, and C64 LINK. Amazingly, these units allow the use of parallel, serial, and RS-232 (another standard method of transmitting signals) devices. Since these products are much more than printer interfaces, we'll look at them in more detail in a future issue.

Cardco, Inc.
313 Mathewson
Wichita, KS 67214
CARD/?B $49.95
CARD/?+G $89.95

Micro Ware
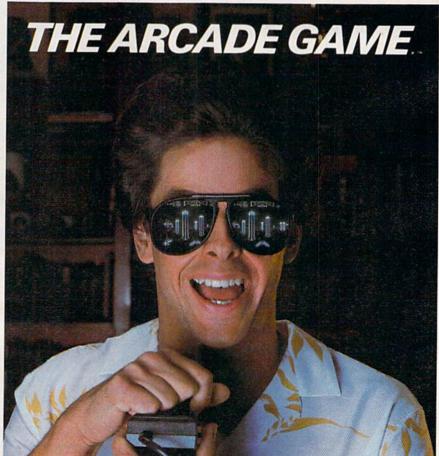*1342B Rt. 23*
Butler, NJ 07405
The Connection $119

Orange Micro, Inc.
1400 N. Lakeview Ave.
Anaheim, CA 92807
Grappler CD $139

XETEC, Inc.
3010 Arnold Rd.
Salina, KS 67401
XETEC $49.95
(with graphics, $79.95)

its limitations. It uses about 8K of the 64's RAM to store hi-res screens, and the screen dumps will not print sprites, because of the way sprites are stored in memory. Also, the screen dump feature can't be used when another expansion cartridge is being used. It seems impossible to get a screen dump without also printing the commands that generated it, so your picture of the sunset will have PRINT"AG" somewhere in it. It is possible to screen dump from within a program without these commands appearing on the screen, but only if the RUN/STOP key is still enabled.

The Grappler CD comes in a well-designed housing, and draws power safely with no exposed connections.

In addition to the parallel printer interfaces we've covered, there are several other products for the VIC and 64 which serve as interfaces but offer many other features. Three of these

User Group Update

Kathy Yakal, Editorial Assistant

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

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

Northwest Arkansas Computer Users Association
Mike Mahoney
1556 N. Leverett #11
Fayetteville, AR 72701

Commodore Computer Club
Joe Ragsdale
P.O. Box 6000
So. Station
Ft. Smith, AR 72906

Harrison Users Group
Dennis C. Loiselet
Rt. 2, Liar, Apt. #3
Harrison, AR 72601
(501) 741-3425

Canyon De Chelly—Four Corners Users Group
Larry DiLucchio
Calumet Consulting
Box 1945
Chinle, AZ 86503
(602) 674-3421

Valley Computer Club
Marcia Esparza
P.O. Box 310
Denair, CA 95316

Lowell HS Commodore 64 Users' Group
Ben Lee
1566 9th Ave.
San Francisco, CA 94122

Commodore Users Group of Santa Cruz
Elli Gould
P.O. Box 8068
Santa Cruz, CA 95061-8068
(408) 335-2082
(408) 476-0294

VICDore Users Group
Wayne Sundstrom
326 Emery Drive
Longmont, CO 80501
(303) 772-2821

Ram Rom 84
Nancy Kenneally
1620 Morning Dove Lane
Englewood, FL 33533
(813) 474-9450

Citrus Commodore Users Group
Ralph Juliano
P.O. Box 1494
Inverness, FL 32651
(904) 344-2793

Commodore Club of Augusta
David Dumas
929 N. Willow Wick Drive
Grovetown, GA 30813
Board # (404) 863-7733

Stone Mountain User's Group-64 (SMUG-64)
John Chambers
P.O. Box 382
Snellville, GA 30278

Pocatello Commodore Users Group
Gary R. North
82 Mountain Drive
Pocatello, ID 83204

Springfield Pet User Group (SPUG)
Bill Eardley
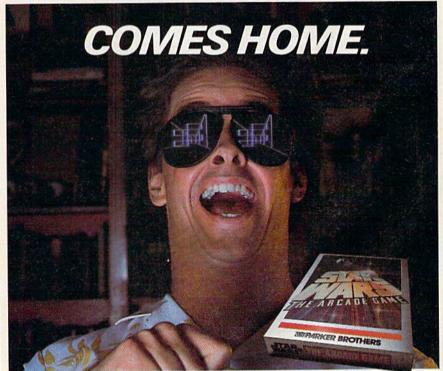3116 Concord
Springfield, IL 62704
(217) 753-8500

Wilmington Area Computer Users Group
John Marshal
P.O. Box 301
Wilmington, IL 60481
(815) 476-6575

Parsons Commodore User Group
Bonnie
3300 Main
Parsons, KS 67357
(316) 421-9210

Baltimore Area Commodore Users Group (Bay-CUG)
Michael M. Broumberg
4605 VOGT Avenue
Baltimore, MD 21206
(301) 325-2156

Southern Maryland Commodore Computer Users Group (SMCUG)
Stephen Quintis
4115 Cassell Blvd.
Pr. Frederick, MD 20678
(301) 535-5872
(301) 586-0386

Suburban 64 Users Group
Ken Partridge
P.O. Box 919
Islington, MA 02090

COM-RADES
Best Business Equipment
Joan M. Evanosky
269 Lincoln Street
Worcester, MA 01605
(617) 829-2344

www.commodore.ca

Tri-C.U.G.
Mark D. Meyers
P.O. Box 45
Bay City, MI 48706
(517) 893-6999

Michigan Commodore 64 Users Group, Inc.
Jan Rooks
P.O. Box 539
East Detroit, MI 48021
(313) 773-6302

COM/VIDEO
Jason Shuster
124 East Maple Street
Gladwin, MI 48624
(517) 426-3520

Young Peoples' Computer User Group (YP-CUG)
Dave Grost
200 North Foster
Lansing, MI 48917
(517) 321-4359

The Commodore Club
David Daniel Smalldon
7761 South Williams Road
St. Johns, MI 48879
(517) 669-5963

Coastal Commodore Club
Chris Elliott
Waldoboro Village, Apt. 19
Waldoboro, ME 04572

Platte Valley Commodore Users Group (PVCUG)
Jim Parks
1720 O St.
Gering, NE 69341
(308) 436-3211

Silver State Commodore Users Group
Karen Douglas
5109 South Reeder Circle
Las Vegas, NV 89119

Connecticut River Commodore User Group
Ronald Greenleaf
Charlestown Community Room
Charlestown, NH
(603) 826-4183

Grey Locker User Group
Mark Ford
Littleton High School
Littleton, NH 03561

Bordentown Area Commodore Users Group
Joe Griner
10 Spring St.
Bordentown, NJ 08505
(609) 298-6275

Cape Atlantic Commodore Users Group
B. J. Chadwick
1440 Old Stagecoach Road
Ocean View, NJ 08230
(609) 398-4044

Commodore 64 NJ Users Group
Anthony Germinario
87 North Queen Street
Bergenfield, NJ 07621

Rancocas Valley User Group
M. Eisenbacher
P.O. Box 234
Mt. Laurel, NJ 08054
(609) 267-1912

Olympian Computer Club
Paul A. Mullens
Box 4277
APO New York 09223

Commodore 64 Software Group
Michael S. Kimmel
3028 Verity Lane
Baldwin, NY 11510

Deltacom Users Group 64
Joe Pacifico
Delta Video and Computer Center
248A North Avenue
New Rochelle, NY 10801
(914) 632-5500

Eden Area C-64 Users Group
P.O. Box 3053
Eden, NC 27288

Akron Commodore Software Exchange
Daniel Shibley
P.O. Box 9243
Akron, OH 44305

Port Clinton—Commodore 64 Users Group
2158 North East Catawba Road
Port Clinton, OH 43452

Tulsa Area Commodore Users Group
Craig Bowman
7804 N. 117th E. Ave.
Owasso, OK 74055
(918) 272-9755

Scranton CUG
Mark Davis
P.O. Box 211
Clarks Summit, PA 18411

Wyoming Valley Commodore Users Group
Lee Zielen
304 East Oriole Drive
Larksville, PA 18704
(717) 288-7949

Oxford Circle 64 Users Group
Roger Nazeley
4921 Castor Ave.
Philadelphia, PA 19124
(215) 743-8999
(215) 535-9021

C-64 Users Group of Florence
Scott Warren
115 Allen Road
Darlington, SC 29532
(803) 395-0270

EL PASO 64's
Joe Grossinger
1713 Dean Martin St.
El Paso, TX 79936
(915) 855-1107

CEN-TEX Commodore Users Group
Chris Peltier
1406 Alta Mira
Kileen, TX 76541
(817) 699-8983

Commodore Users Group
David M. Boyter
Rt. 2, Box 278B
Boydton, VA 23917

Burlington Area Commodore Users Group
Steve Lippert
6 Mayfair
South Burlington, VT
(802) 658-4160

RCC Club
Jeff Andrie
1316 Hunt
Richland, WA 99352

Central Washington CUG
Sam Cox
P.O. Box 10937
Yakima, WA 98909
(509) 248-8193

S.W.I.T.C.H.
Pat Gaylord
W153 N8009 Meadowland Drive
Menomonee Falls, WI 53051

Campaign Manager

Todd Heimarck, Assistant Editor

This two-player national election simulation ranks as one of the best games we've published. With the right strategy, your candidate can make it to the White House. For the Commodore 64.

In this game, the Republican player chooses the senator from Nebraska, who has excellent charisma and intelligence.
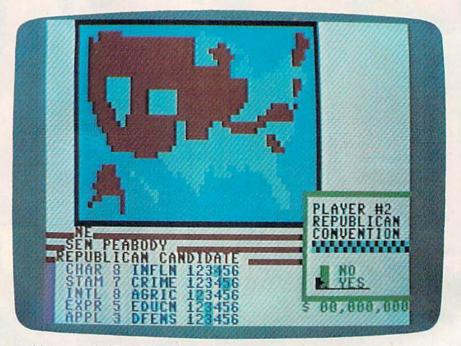
he Democratic delegates are gathered in Moscone Center, wearing straw hats, carrying balloons and signs. The floor fights are done. The time has come to nominate.

"Maryland?"

"Mister Chairman—the great state of Maryland, The Free State, Home of the World Champion Baltimore Orioles, casts all of its votes for the senator from Arizona."

The chairman pounds his gavel. The din of cheers and jeers subsides. The convention is deadlocked. And you control a large block of uncommitted delegates. It's all up to you.

The vice president from Rhode Island has good charisma and intelligence, but you know his health is poor. The reverend from Arkansas is attractive, but a bit conservative. Although the senator from Arizona is experienced, he's not very smart. Perhaps the New Jersey doctor? No, the Ohio senator has the best combination of personality and issues, plus you'll get a home region advantage in the populous Heartland.

Now it's the Republican's turn. Of the five choices, the woman from South Carolina is the best all-around candidate. She has high charisma and fundraising appeal, which translates well into television ads.

It's time to hit the campaign trail.

The Democratic senator starts with $9 million and 59 health points. He rests two days (to build up his health), then spends two days fundraising. Campaign stops in Illinois and Texas sway the voters slightly to the Democratic side.
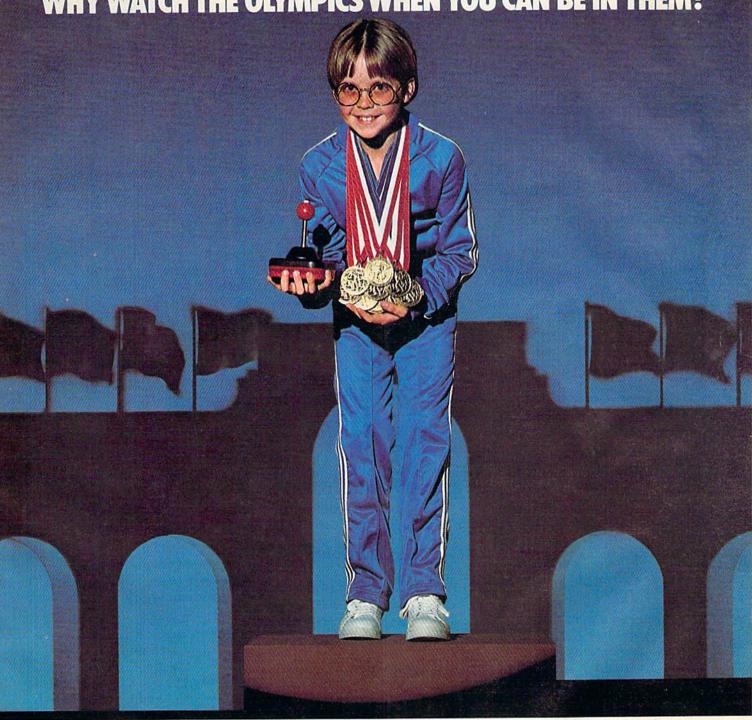
The Republican campaigns in her home state of South Carolina. She then moves on to

North Carolina, Virginia, and Florida, followed by a couple of days resting.

As the campaign progresses, the Democrat concentrates on personal appearances in the industrial northeast, plus forays into the larger states such as Texas, California, and Florida. The Republican candidate does less actual campaigning, preferring to spend more time on fundraising to pay for the (expensive) television ads.

In the crucial eighth week, both candidates rest and fundraise in preparation for the last minute campaigning. The Democrat does a media blitz in the Pacific, Southern, and Atlantic

SUMMER GAMES.
WHY WATCH THE OLYMPICS WHEN YOU CAN BE IN THEM?

You're an Olympic athlete competing in eight key events at the Summer Games. How well can you score in track, swimming, diving, shooting, gymnastics and more? So realistic, there's even an opening ceremony and awards presentation after each event.

Unlike other "Olympics-Like" games, Summer Games has incredible realism, superb state-of-the-art graphics and sound effects (including national anthems from 18 countries), and it is a true action-strategy game. In each event you must plan and execute your game strategy in order to maximize your score. It is not just a matter of how fast you can move the joystick.

So change into your running shoes, grab your joystick and GO FOR THE GOLD!

One or more players; joystick controlled.

states. The Republican hits the Heartland, Arklatex, and the Urban Northeast.

Initial returns from New England show the Republicans sweeping the region, but the large states of New York and Pennsylvania went Democratic. The Republicans won most states from Ohio to the Great Plains, but the Democrats picked up the Southern Atlantic states (except Florida). Texas voted for the GOP, while the rest of the region went Democratic. The Rocky Mountain states were solid Republican. The Democrats won the Pacific States.

The final results show the Republicans winning six of nine regions and capturing the presidency, with 315 electoral votes to the Democrats' 223. Three of the four biggest states voted Democratic, but Ohio and Illinois (with 47 electoral votes between them) made the difference. The TV ads in the last week moved these two key states into the Republican camp.

Written entirely in machine language, "Campaign Manager" pits you against an opponent. Each of you manages the campaign of your candidate. The player who makes the right decisions gets his or her candidate elected.

You have nine weeks to campaign. Each week you plan your moves and enter them via the menu on the itinerary. You have two defensive moves, resting and fundraising, and two ways to gain votes, campaigning (personal appearances) and advertising on television.

At the beginning of each turn you see a medium-resolution map of the U.S. which indicates which way each state is leaning. The MAP option allows you to move a cursor around the country, to identify which states are which. If the Republicans are ahead, the state is red. Democratic states are cyan (light blue). If you're using a black

and white television, the Republican states are the darker ones. You may notice that states occasionally switch back and forth, even though neither candidate campaigned or advertised there. This indicates that the voters in that state are split down the middle, and because of slight errors in polling, seem to be leaning one way or the other.

Since you only have 63 days (nine weeks of seven days), you have enough time to campaign in each state once or twice. But in terms of electoral votes, California (with 47) is far more important than some of the smaller (three vote) states like North Dakota or Vermont.

Generally, it makes more sense to campaign more heavily in the ten biggest states, sometimes called "megastates".

State	Electoral Votes
CA	47
NY	36
TX	29
PA	25
IL	24
OH	23
FL	21
MI	20
NJ	16
NC	13

Winning the election requires 270 electoral votes (of a possible 538). The ten biggest states account for 254, just 16 short of a majority.

At the beginning of the campaign, each state has a large pool of undecided voters. As the game progresses, they make up their minds and the pool diminishes. It's possible, but unlikely, for all of a state's voters to decide before the end of the campaign. You would have to go to the state at least eight times before the undecided points were used up.

Each state has a built-in bias toward one party, based on past elections for president, senator, governor, etc. The District of Columbia, for example, is staunchly Democratic, so the Democratic candidate will automatically get seven campaign

points there, compared to a Republican's two.

Since the Republicans have won three of the last four elections (including a landslide victory in 1972), you might expect them to begin the game with a huge advantage. But if you look at non-presidential elections, you will find a lot of states that elect Democratic governors, senators, and representatives and then vote for a Republican president. And a lot of those basically Democratic states were split by third-party campaigns (Wallace in '68, Anderson in '80).

To even things up, and make the game more playable, the Democrats begin with an electoral vote advantage of 282 to 256, although four of the megastates (PA, OH, FL, and NC) are barely leaning to the Democratic side. The Republicans have the advantage of beginning with 29 of the 51 states (since DC has three electoral votes, it counts as a state). Most of the states west of the Mississippi are Republican, while the Democrats have most of the industrial Northeast and the South.

In addition to the natural political leanings, each state believes certain things about five general issues:
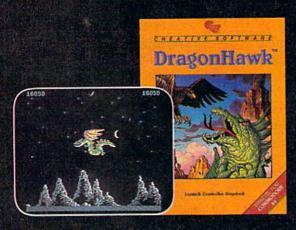
1) unemployment/inflation, 2) poverty/crime, 3) agriculture, 4) education, and 5) defense.

(The issues are based on census reports, almanacs, etc.) A very urban state might be conservative on crime, but not care much about agriculture, for example. Each candidate has certain stands on these issues. When you campaign or advertise in a state, you can get up to three extra campaign points for each issue, if you agree with the citizens there.

Finally, the candidate you choose has a campaign effectiveness rating based on charisma and intelligence. This factor translates to votes each time you campaign in a state.

DragonHawk™

You are the DragonHawk, soaring to attack — and escape from — a host of flying monsters. Your mission is to destroy the giant fire-breathing serpent that has wreaked havoc on an entire mountain range.

Can you conquer the flying hordes of spellbound monsters? Can you survive the lightning storms? Can you discover the serpent's Achilles' Heel and survive to become Master of the Mountain Range?

DragonHawk is a fast action, fantasy game. Multiple difficulty levels insure challenging play for even the most adept fast action fans.

For the Commodore 64.

CREATIVE SOFTWARE

"Commodore 64" is a trademark of Commodore Electronics, Ltd.
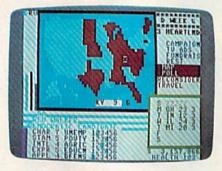
© 1984 Creative Software
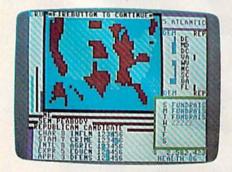
Week 2: Reverend Walker rests, raises money, and campaigns in three Great Plains states.

Week 4: Senator Peabody hits Texas twice and then jets to the West coast.

Week 6: After visiting the Heartland, the Democrat checks the map.

Week 8: The Republican uses a poll to discover that most of the Atlantic states are barely leaning to the Democratic side.

To start the game, choose which party will go first. You might want to flip a coin, the winner choosing either a party or to go first or second. In testing, we found that the second player has the very slight advantage of making the last move. Next, decide if one of you will start out as the campaign manager for the president running for a second term. Being incumbent gives you some extra campaigning strength, and is not recommended if you want an even game.

Note that all choices can be made with a joystick in either port. Move the pointer to a menu item and press the fire button twice to make your choice. If you don't own a joystick, use I, J, K, and L for up, left, down, and right respectively. Press M in place of the fire button.

Players then pick which candidate will represent their party. Five randomly chosen candidates are available. To the right of the candidate's stats is the YES/NO counter. Before making your choice, pick NO for each possibility until you have seen all five. They will cycle around again so you can make your choice.

The heart of the game is the actual campaign, but in some ways the convention is more important. Nominate a terrible candidate and you'll spend most of your campaign trying to catch up.

A candidate's personality greatly affects the outcome of the election. In the lower left corner you'll see a list of five attributes, each associated with a number from one (worst) to eight (best). With a couple of exceptions, the ideal candidate is the one with straight eights.

First is charisma (CHAR), which is personal magnetism, panache, the ability to influence and excite people. This is the most important personality trait

because it is part of both campaign effectiveness and advertising effectiveness.

Stamina (STAM) rates your candidate's health. A candidate with low stamina will have to rest frequently to regain health and strength.

Intelligence (INTL) adds points to campaign effectiveness and last minute campaigning.

Experience (EXPR) helps you with fundraising. If your candidate has lots of experience, he or she has more contacts and connections for raising money. Since experience comes with age, it counts against your health, although stamina counts for more health points.

Appeal (APPL) also contributes to fundraising appeals. But if you have maximum appeal (eight) you may be tainted by your affiliations with special interest groups, and there is a backlash when you advertise. It's best to have an appeal of six or seven.

The candidates' attributes are generated by adding three random numbers, so candidates are more likely to have a middle number (four or five) than one of the extremes.

The personality traits translate into these five campaign factors:

Campaign Effectiveness (CHAR*2 + INTL): the key factor in campaign stops.

Strength/Health (STAM*4 +9 − EXPR): determines the effectiveness of a rest day.

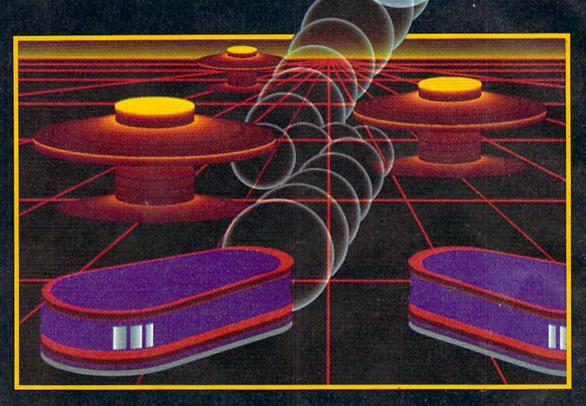**Fundraising Appeal** (EXPR*3 + APPL): determines how much money can be raised in a day.

TV Ads (APPL OR 8 + CHAR): translates into votes when advertising.

Last Minute Campaigning (INTL + STAM): wins last-minute votes to your side after the ninth week.

The significance of each factor is discussed later.

Next to the personality factors are the candidate's stands on various issues. You see five issues, each with a sliding scale of one (at the far left, representing liberal) to six (conservative). A Republican who wants to get tough on crime, for example, will have a rank of six. A Democrat who wants to solve the unemployment problem will have a rating of one.

Candidates will range from two to five on the issues of agriculture and education. On the other three issues, the Democrats will have stands from one to four; the Republicans will go from three to six.

You will generally get more votes with middle of the road beliefs. Look for a candidate with twos or threes if you're the Democrat. Fours and fives are best for the Republican. The exception is agriculture and education, where you do best with a three or a four.

Common sense tells you which issues are important in most states. Agriculture is a major issue in the farming states. Your stand on defense makes a difference in states with a lot of military-related industry.

The candidate's personality is generally more crucial than the stands on issues. If you have a lot of charisma, intelligence, and appeal, it doesn't matter that you may have radical views on one or two issues.

If you have five very bad candidates, press RUN/STOP–RESTORE and try again. It's not much fun to run a campaign you are destined to lose.

After the nominees have been chosen, the first week begins. You may notice that some states have changed colors. That's because each nominee gets the equivalent of campaigning once in each state. Some people make up their minds before the campaign even starts. If one candidate is much more charismatic, or happens to hit

the right issues, a state may jump over to his or her side. In addition, each gets a home state and home region advantage.

You should develop a strategy. If your appeal and charisma are strong, concentrate on television ads. If your candidate has a strong anti-crime stance, visit the more urban states. At the very least, you should plan to visit each of the megastates.

You begin in your home state where it is traditional to campaign once (but not twice). And the first week usually means some fundraising and resting as purely defensive moves.

Under the week's itinerary are two numbers representing money and health. At the beginning of each week, your treasurer tells you how much money you have, up to a maximum of $25 million. Your personal physician figures out how healthy you are. At most you'll have 255 health points.

If you fall below $4 million any time during the week, television advertising will be useless until you replenish the campaign coffers. If you have less than one million, you won't be able to pay the pollster (the bar graph to the left of the map will disappear). When your bank account falls to zero, the campaign is paralyzed until you sponsor a fundraiser. You can't even afford to pay your doctor or staff.

It takes time away from campaigning, but you have to raise money once in a while. Each fundraising point (experience times three plus appeal) is worth $200,000.

Campaigning takes a lot out of you, so you have to occasionally take a day to rest and relax. When you decide to catch some Zs, the itinerary will be filled with (you guessed it) Zs. Each day of rest adds double your strength factor, plus campaign effectiveness, plus the number of states you are winning to the

health you have. A high campaign effectiveness gives you optimism; you rest better. If you're behind, you lose sleep worrying about it. Resting two days in a row gets you 16 extra health points.

There are two reasons to keep your health up. First, when you campaign in a state, you get an extra campaign point for every 32 health points you possess. Second, if your health falls below eight you look haggard and stutter; campaigning does you no good.

The treasurer counts dollars, the doctor counts your health, and your pollster counts votes.

The pollster does three things. First, you get a bar chart that shows how many electoral votes would go to the Democrats and Republicans if the election were held at that time. You can see it to the left of the map. The gray bar marked U represents undecided states too close to call. Second, you have a map of the U.S. to show you, at a glance, which way each state is leaning. Republican states are red; Democratic states are blue. These first two services are part of the pollster's contract, and cost you nothing. Of course, if your money drops lower than one million, you have to stop paying the pollster; all you get is the map.

The third service is the most important—regional polls. To get a poll of all states in a region, move to POLL on the main menu and press the fire button twice. You'll see a bar chart showing which way each state in the region is leaning, from one (half a character wide) to four (two characters). The poll reflects the political situation at the beginning of the week; whatever campaigning you have planned for the week is not included. A state with a thin bar can usually be taken with a single campaign stop.

Don't use polls in the first

www.commodore.ca

Although Walker won California and New York, the Republicans have captured the Presidency, 325 to 213.

couple of weeks because most states start out fairly even and you won't learn much. But polling can be a powerful tool towards the end of the game. If New York is firmly committed to you, forget about further efforts in that state. And if you find a whole region weakly supporting your opponent, you can hit them with TV ads and score a few dozen electoral votes.

Regional polls cost $100,000 and are not available if you begin the week with less than $1 million.

The final character (although transparent) in your entourage is the jet pilot. Your jet can carry you on short hops within a region for almost nothing. But if you travel to a new region, you shell out $100,000 for fuel, maintenance, etc. As long as you're in a region, you might as well stay there a few days to avoid a lot of travel expenses. Again, you don't actually move to a new region until you have campaigned in one of the states. You can use the travel option to conduct regional polls; you'll pay $100,000 for the poll, and another $100,000 if you decide to campaign in a region. If you travel to a region to poll and decide not to campaign, you won't be charged for traveling.

Benjamin Franklin once said that after three days, guests and fish begin to smell. The same principle applies to campaigning.

Campaign once and you gain some votes. Stay for a second day and the voters of a state are flattered; you gain a couple of bonus votes. But stick around for a third or fourth day and you have overstayed your welcome. Do not campaign in a state more than two days in a row.

Each state begins with 255 undecided voter points. Your main goal is to use campaigning and television advertising to sway the undecided. And you have to maintain your health and money.

The effects of a personal appearance can vary. You get up to three points for each issue (if the state agrees with you), one point for every 32 health points, and up to 24 for your campaign effectiveness (intelligence plus double charisma), and a two point bonus if it's your second day in the state.

If your money is down to zero, you get no campaign points. If your health is below eight, you get a single vote.

Each campaign stop decreases your health and money. It's possible to run out in the middle of the week, making each succeeding visit ineffective until you rest or raise money. Let's say you go to Connecticut and impress 23 of the 255 undecideds. The pool of available voters is reduced by that number. Half of 23 (11 points) is charged against your health. Half again (5 points) times $100,000 is subtracted from your money. In addition, each state has some people who don't agree with you, so a quarter of your total (five points) goes to your opponent as a reaction against your speech. If you had previously been in a different region, travel expenses of $100,000 are subtracted.

Television advertising is a little different. It affects every state in the region, and quickly swings voters to your side. To advertise, first travel to the region and make at least one campaign stop to establish your presence. You can then place the cursor on TV ADS and press the fire button twice. After campaigning once, advertise as much as you like.

Unlike resting and campaigning, the effects of advertising do not accumulate from day to day. If you advertise two days in a row, you don't get bonus points. Advertising does grow in strength from week to week, however, and will be more effective towards the end of the campaign.

If you flood the region with ads, it's possible to bring a whole section of the country to your side. But it is costly. In each state, advertising credits you with half your campaign effectiveness, half your TV ads effectiveness rating, points for issues, plus two times the week number (in week seven, for example, you get 14 extra campaign points).

The cost is the usual one-fourth of campaign points gained, plus double the TV ads' effectiveness. The large regions can cost a lot. Going on TV in the Atlantic States (all nine) or in the Rocky Mountains (eight) can deplete your treasury.

On the day you plan to advertise, you must have at least four million dollars. If you don't, you waste the day and gather no new votes. So, if you begin the week with $5 million, and campaign in six states, it's likely you'll have less than $4 million by Saturday. Your ad campaign will do you no good.

There is one more item you can choose: RECONSIDER. If you make a mistake, this option wipes your itinerary clean so you can start the week anew. Your choices are not permanent until you fill out the seventh day and press the fire button. (If you pull down on the joystick, your slate will be wiped clean—a quicker way to reconsider.)

Main Menu Command Summary

CAMPAIGN—allows you to make a personal appearance in one of the states of the region you're visiting. Results depend on campaign effectiveness, built-in party bias of the state, health, and issues. Does not work if you have zero health or money, or if all undecided voters have been claimed. Gains votes, costs health and money.

TV ADS—blankets the region with advertising. Reduces health and costs a lot of money, but can quickly deliver a big chunk of votes. Net votes based on TV advertising effectiveness, campaign effectiveness, and issues. Does not work if you have less than $4 million.

FUNDRAIS—raises money for your campaign based on fundraising ability. Takes a day, gains no votes, costs nothing.

REST—builds up your health points, according to strength factor. Extra points if you rest two days in a row. Gains no new votes, costs nothing.

MAP—moves the cursor around the map, prints the state name, electoral votes, and region number. For information only, costs nothing.

POLL—provides a bar graph showing which way the states in the region are leaning. Costs $100,000 (immediately). Not available if money falls below $1 million.

RECONSIDER—erases the week's itinerary if you make a mistake.

TRAVEL—takes you to a new region of the country. Costs $100,000 (not charged to you until you actually campaign there).

The ninth week is usually the most hectic. If you sponsored some fundraisers in week eight, you will want to spend a lot on TV advertising in the regions where you have a chance. Polls can tell you which states are most vulnerable.

After both candidates have finished their last week of campaigning, a couple of things happen. The last region to be visited by a candidate gives a few extra votes to him or her. And the last-week routine goes into action, as all the undecided voters make up their minds. Each candidate gets his or her last-minute campaigning points (intelligence plus stamina) added to each state in the country. The undecided voters are split between the candidates and ties are resolved (based on the built-in bias to one party or

the other).

The map is drawn for the final time. The final bar chart appears to the left (which should indicate at a glance which candidate won). Beginning with region one (New England), the electoral votes are displayed, with region totals below.

The winner is the candidate with the most electoral votes. There is a slight chance that there will be a tie, in which case you'd have to flip a coin. If you want to play again, press RUN/STOP–RESTORE and type RUN.

Here are a few rules of etiquette which help to make a fairer game.

First, if you're playing with two joysticks, try to avoid interfering with your opponent's choices. Remember, the joystick routine reads *both* joysticks.

Second, when you have filled out your itinerary and the prompt PRESS FIREBUTTON TO CONTINUE appears, let your opponent study what moves you made, and he or she can then press the fire button.

Third, since polls cost money, they should be kept private. When the other player is taking a poll, avoid looking at the screen.

Special Instructions For Entering Campaign Manager
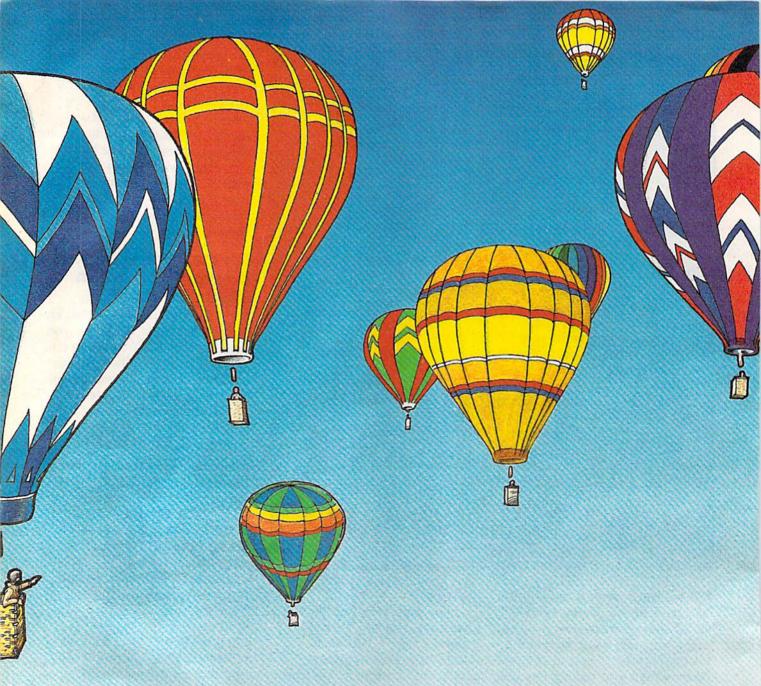
Since the program is written entirely in machine language, you must use the MLX machine language editor (elsewhere in this issue) to enter it. Before loading MLX, you have to protect part of BASIC memory by typing the following line:

POKE 642,50: SYS 58260

You'll then see the usual start-up message, but you'll notice less than the normal 39K RAM. Next LOAD MLX using a start address of 2049 and ending address of 9518 and begin typing. The program uses about 10K, which was crunched down to about 7K to make typing it in a little easier. Since it's such a long program, you may want to enter it in parts. If you choose to do so, make sure you follow the MLX instructions for loading and saving, and enter the above POKE and SYS before you resume using MLX. The newest version of MLX has a numeric keypad, which should save you some time.

When you have finished typing Campaign Manager, make sure to save it to tape or disk (maybe a couple of backup copies as well). Turn your 64 off and then on, LOAD the program (as if it were BASIC), and type RUN. The first few bytes look like a BASIC program with the command SYS 2061. But you don't have to remember the SYS; it's built into the program. *See program listing on page 141.*

Balloon Blitz

Michael T. Bohn

Floating along in your hot-air balloon is a nice way to spend a beautiful summer morning. But you've got a mission: To destroy the enemy's tanks, which are threatening your force's flank. Originally written for the 64, we've added a version for the VIC with at least 8K expansion. A joystick is required.
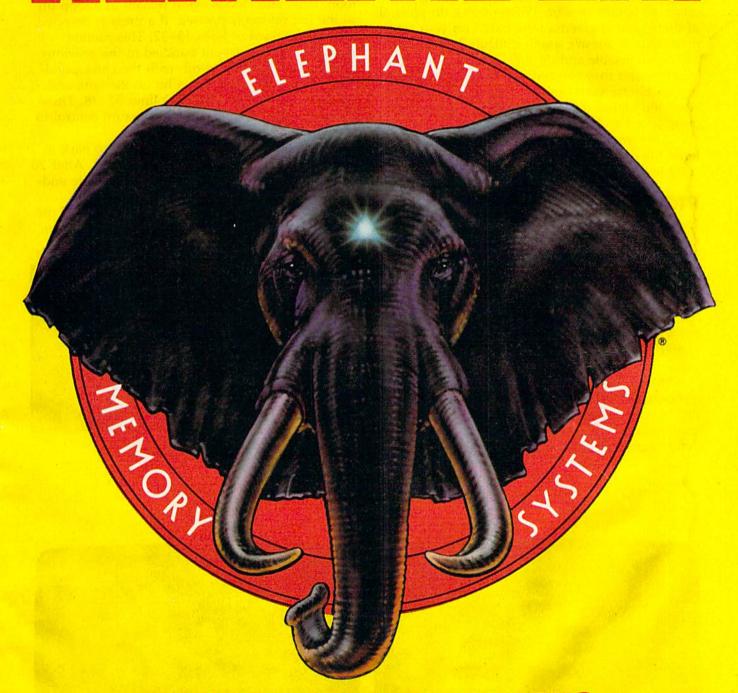
Alone in your anti-tank hot-air balloon, you're drifting along and enjoying the tranquility of a

beautiful summer morning. And hoping the enemy stays out of sight—but you know they won't. They're sending their best tanks (which are wily and evasive) to get to your troop's southern flank, and your job is to destroy them. If you're skillful enough to stop the first wave, you're ready for the next game level.

Elusive Targets

At the beginning of each game, you are asked to choose a difficulty level from 1 (easy) to 6 (hard). After your selection, the action begins.

REMEMBER.

ELEPHANT

MEMORY SYSTEMS

ELEPHANT™ NEVER FORGETS.

www.comm _ _ _ e.ca

With a joystick (port 2 for the 64 version), move your balloon right and left. You may increase or decrease your flight speed at any time during the game by pressing any number from 1 (slowest) to 9 (fastest).

Drop bombs by guiding your balloon over an enemy tank and pressing the fire button. You're given 20 bombs to start. When you've dropped all of them, you're given a rank based on your hit/miss ratio. It doesn't sound difficult, but the tanks are unpredictable and do their best to avoid your bombs—and they can hide in the trees making it impossible for your bombs to do any damage. Also, the higher difficulty levels give the tanks more speed and direction options.

After your 20 bombs are expired, you're offered the option to play again and choose a difficulty level.

VIC Notes

The VIC version requires at least 8K memory expansion. Before loading the game (right after the computer is turned on), carefully enter the following two lines in direct mode (without line numbers):

```
POKE 43,1:POKE 44,32:POKE 8192,0:NEW
POKE 36869,240:POKE 36866,150:POKE 6
  48,30:PRINT"{CLR}"
```

The original version, written for the 64, uses sprites, which of course are unavailable on the VIC. As a substitute, four short machine language routines are used to move the balloon, tank, cloud, and bomb smoothly around the screen.

The 64 Version

The initialization routine for the 64 version is found in lines 100–165. Sprites, screens, and variables are set up in this section. The program then executes lines 2–14, the main movement routine. These lines read the joystick and move the balloon, tank, and clouds—and remain in a loop until the fire button is pressed. If a press is detected, control is passed to lines 15–32. This routine maintains all movement handled in the previous routine, but adds the bomb with the corresponding sound. It also tests for a hit on the tank and, if one is made, passes control to lines 33–38. These lines make the tank explode and return control to line 2.

If the tank is not hit, lines 15–32 go back to line 2 without going through lines 33–38. After 20 shots, the program executes lines 50–80, the end-of-game routine and new game option.

The sprites are set up at the beginning of the program so execution is faster. Here's a rundown on each of the eight sprites:

Sprite	Description	DATA Lines
0	tree	217-219
1	bomb	210-212
2	cloud	220-222
3	tank	207-209
	explosion	213-216
4	tree	217-219
5	tree	217-219
6	balloon	202-206
7	cloud	220-222

See program listings on page 149.

This bomb is right on target (VIC version).

This evasive tank has gone into reverse as the bomb was released (64 version).

HINTS&TIPS

64 LIST Lockup

Alan King

If you've discovered a clever, timesaving technique, or a brief but effective programming shortcut, send it to "Hints & Tips," c/o COMPUTE!'s GAZETTE. If we use it, we'll pay you $35.

The Commodore 64 is subject to something called the "strange lockup bug." To make it happen, move the cursor to the line at the bottom of the screen. Now type anything to fill up two complete screen lines. After the cursor has wrapped around twice, scrolling the screen twice, press the DELete key. If you have a program in memory, it will run and the screen will say READY, with a blinking cursor. But you won't be able to type anything. The computer is locked up. (Commodore's new portable SX-64 does not have this problem, which suggests that a solution has been found.)

This bug is not a problem if you do it on purpose. But if you have been developing a program for three hours, and have not backed up your work, it can be very annoying to lose everything you just did.

One way to escape, if you own a Datassette, is to press the left SHIFT key and 3 at the same time. You will see the prompt, PRESS PLAY ON TAPE. Press the play button, hit RUN/STOP and the computer will be back to normal.

Let's take advantage of this bug.

There are occasions when, for whatever reason, you don't want people to LIST your program. And with just a couple of program lines,

you can make the computer lock up when someone tries to look at your program. Put these two lines at the beginning of your program:

Line 1:
Type the line number and REM
Type two quotes (SHIFT-2)
Delete the second quote
Type CTRL-9 (RVS ON) and eight T's
Type SHIFT-M
Type back arrow (the key above CTRL)
Type 25 Q's
Press RETURN

Line 2:
Type the line number and REM
Type seven SHIFT-Y's
Type two quotes and delete the second one
Enter RVS ON and SHIFT-M
Type 15 T's and Press RETURN

Now SAVE the program. If you try to list it before saving, you'll get the lockup.
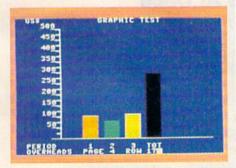
You should probably save a listable version for yourself, in case you want to make changes some day. And note that while this prevents listings, it does not affect LOAD, RUN, or SAVE. People can still make copies of your program. And if you use this trick on lines one and two, the user can get around it with LIST 3—. Your best bet is to sprinkle these two lines throughout the program.

The first line makes sure the cursor is at the bottom of the screen. The second causes the lockup to happen. A REM followed by a quotation mark puts the computer into quote mode. A reverse-T is then interpreted as a delete, reverse-SHIFT-M is a carriage return, and so on. **G**

Handic —A Complete Line of Products to Keep Your Commodore 64 Busy —Everyday!
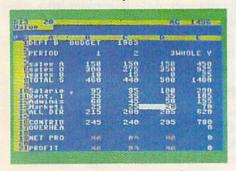
Handle your home budget, stock portfolio, loans and mortgages with Calc Result

Calc Result Easy is a simple-to-use spreadsheet program for the Commodore 64. It includes 254 lines × 64 columns, built-in graphics, and flexible printout formats. Plug-in cartridge... just plug it in and its ready. Perfect for cash flow analysis, personal net worth, IRA analysis, travel expenses, credit card expenditures, gas and electricity bills, etc.

Calc Result Easy $49.95

Calc Result Advanced gives you 32 pages of interrelated information. The three-dimensional feature allows you to consolidate calculations in summary format. Calc Result Advanced comes on plug-in cartridge and disk. Disk drive required.

Calc Result Advanced $99.95

A complete database for the home
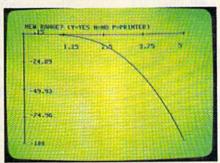
Addresses, telephone numbers, appointments, birthdays, or records—whatever you want to remember—put it on DIARY, an electronic notebook for home use. DIARY comes on a plug-in cartridge. It's easy to use and easy to learn, giving you the flexibility to design a personal calendar or address book.

Diary $29.95

Turn statistical information into graphic format
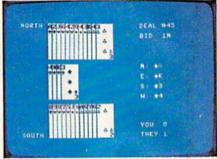
GRAF 64 converts mathematical functions into graphical analysis on the Commodore 64. An ideal program for studying math. Define a function, set the limits of an axis, plot a graph and display the extreme points, intersection values, etc.

Graf 64 $29.95

Develop your bridge skills

Whether you're an experienced bridge player or a beginner, polish your skills or learn the game with BRIDGE 64. Play North-South, then switch to East-West in the same deal, the return to that deal again and test your skill with a different strategy.

Bridge $39.95

Handic—for the broadest range of Commodore products

As the largest independent developer of Commodore software and accessories, Handic's broad range of business, education and recreation products are designed exclusively for the Commodore user who demands quality and reliability.

The Catalogue

Software & Hardware accessories for Commodore Computers

handic
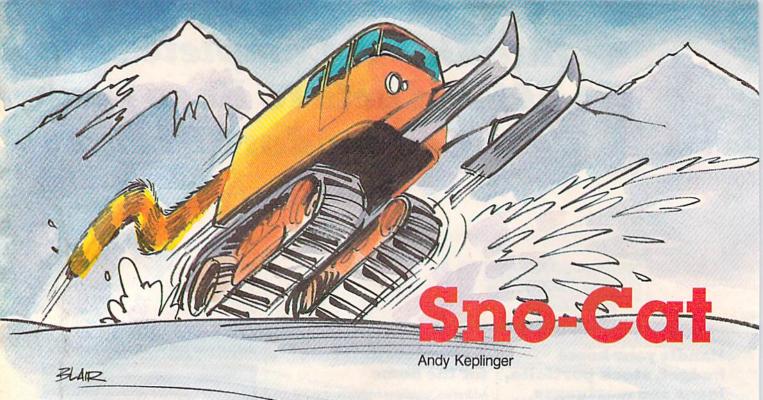
For more information and a catalogue of our products, see your nearest Commodore dealer, or call us direct.

handic software

Handic Software, Inc.
Fellowship Business Center
520 Fellowship Road, B 206
Mount Laurel, NJ 08054
Phone (609) 663-0660

Commodore 64 is a registered trademark of Commodore Electronics, Ltd.

www.commodore.ca

Sno-Cat

Andy Keplinger

BLAIR

Nobody ever climbed Everest like this. See if you can drive your super fast turbo-driven tractor to the top in this game for the VIC and 64.

The Sno-Cat is a super-powered turbo-driven tractor that can go zero to fifty in less than a second. It has a minor problem, however: It shifts its own gears. It starts in second, shifts into third automatically after ten seconds, then into fourth after fifty seconds. It can't go any higher, but it doesn't need to. It's difficult to maintain control in third gear, let alone fourth.

Fortunately you have brakes, *very powerful* brakes that will slow you down to almost zero in tricky situations. To stop overuse, the brakes (controlled by the fire button or space bar) cause the distance meter to stop as long as the brakes are pressed. But the timer still runs, so your time may be impaired by using the brakes too much.

Interrupt-Controlled

If you look at the main routine, you may notice there are no statements for controlling the player or the trees. These are controlled by an *interrupt routine* in machine language, those first few lines of data (lines 10–64). The routine automatically moves sprites 1 through 7 down the screen and moves sprite 0 (the Sno-Cat) according to the joystick's position.

An interrupt routine is a special program that is run every $1/60$ second. The computer's normal interrupt routine is used to read the keyboard and update the values in the timer. It's called an interrupt routine because it stops whatever the

computer is doing, checks for a keypress, adds 1 to the timer, and lets the computer continue.

I've changed this a little. The new order is to go to the sprite movement routine, then continue with the normal interrupt routine functions. So now, every $1/60$ second, it will move every sprite down a little and move the Sno-Cat.

Two Things At Once

If you press RUN/STOP during the game you can move the joystick around and see that the modified interrupt routine is still functioning. Pressing RUN/STOP and RESTORE together returns things to normal.
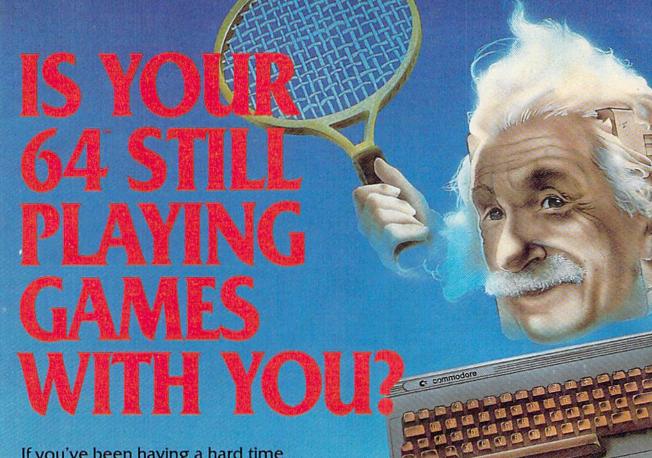
The first reason for an interrupt is speed. With an interrupt routine, the computer can process a BASIC program and still execute the sprite movement routine at the same time—in effect, doing two things at once.

The second reason is for smoothness in movement. This routine is performed every $1/60$ second while the BASIC portion of the program is completing a loop about ten times a second. Without an interrupt routine, the sprites would blink and jump around the screen.

The machine language data is broken into two parts, but only the first part is called from BASIC. It simply tells the computer to add the sprite routine to the normal interrupt sequence.

Sprite Movement

The second part is the sprite movement routine itself. It starts at memory location 841 and is broken into two more parts. The first is the machine language equivalent of this BASIC program:

IS YOUR 64 STILL PLAYING GAMES WITH YOU?

If you've been having a hard time teaching your newly-adopted computer there's more to life than fun and games, you're not alone.

Now, you can introduce your Commodore 64™ to the Work Force: affordable, easy-to-use software and hardware that will unleash the power you always expected from your Commodore 64™, but thought you might never see.

PaperClip™

is simply the best word processing program of its kind—loaded with advanced features, yet so easy to use even a novice can get professional results. With **SpellPack™**, it even corrects your spelling! Once you've tried it, you'll never use a typewriter again.

The Consultant™
(formerly Delphi's Oracle)

is like a computerized filing cabinet with a brain. Organize files for recipes, albums, or the membership of your service club. Then search, sort, arrange and analyze your information with speed and flexibility that's simply astounding.

SpellPack™

teaches your 64 to spell. It checks an entire document in 2 to 4 minutes against a dictionary of over 20,000 words. And you can add up to 5,000 of your own specialized terms. Type letter perfect every time!

BusCard II™

is a magic box that lets you transform your humble home computer into a powerful business machine. It gives you the added power of BASIC 4.0, and lets you add IEEE disk drives, hard disk, virtually any parallel printer, and other peripherals without extra interfaces. Completely software invisible.

B.I.-80™ Column Adaptor
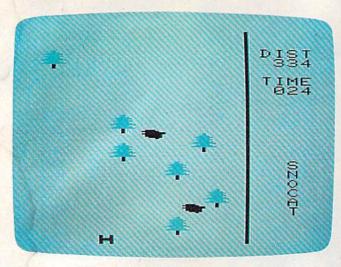
gives you crystal clear 80 column display. Using the highest quality hardware, we've eliminated the problems of snow, fuzziness and interference. Basic 4.0 commands greatly simplify disk drive access. Switches easily from 40 to 80 column display.

Discover the true power of your Commodore 64™. Ask your dealer about the Commodore 64™ Work Force, from Batteries Included—the company that doesn't leave anything out when it comes to making things simple for you.

BATTERIES INCLUDED
"Excellence in Software"

These products have been developed specifically for Commodore computers by Batteries Included and are totally compatible with each other. For a full color brochure write to:

186 Queen Street West, Toronto, Canada M5V 1Z1 (416) 596-1405 / 3303 Harbor Blvd., Costa Mesa, CA 92626 (714) 979-0920

64 AND COMMODORE 64 ARE REGISTERED TRADE MARKS OF COMMODORE BUSINESS MACHINES

Distance and time are displayed as you maneuver the Sno-Cat up the mountainside (VIC version).

```
10 MEMSTART = 53251:REM VERTICAL POSITION
   OF SPRITE 1
20 FOR X=1 TO 7:REM COUNT FROM 1 TO 7
30 A=PEEK(MEMSTART):REM GET SPRITE X'S VE
   RTICAL POSITION
40 A=A+2:IF A<256 THEN 60
50 A=251:REM A IS 251 IF A WAS >255 IN LI
   NE 40
60 POKE MEMSTART,A:REM PUT A IN SPRITE X'
   S VERTICAL POS.
70 MEMSTART=MEMSTART+2:REM GET NEXT SPRIT
   E POS.
80 NEXT X:END:REM BACK AROUND UNTIL SPRIT
   E 7 IS REACHED
```

The second part moves a sprite right or left according to the joystick input. If you don't have a joystick, use the CTRL key for left and the 2 key for right. The space bar can be used to apply the brakes. There is no special provision for this in the program; it is built into the computer's keyboard reading routine.

Pressing the fire button puts you immediately in the rescue chase (64 version).

Programmer's Notes: VIC Version

Kevin Mykytyn, Programming Assistant

The VIC version of "Sno-Cat" is divided into two parts to fit into the unexpanded VIC (be sure to remove or disable any memory expanders). The first, Program 2, POKEs in machine language and redefined characters and then loads the second part, Program 3, which is the main portion of the game. For the autoload feature to work properly, Program 3 must be saved on tape or disk with the name SC. Tape users must change the 8 in line 100 to a 1, and should save Program 3 immediately following Program 2 on the same tape.

This version includes a slope littered with rocks in addition to the trees, so there are more obstacles to be avoided. Large multicolor characters are used for the trees, rocks, and the Sno-Cat, in place of the 64's sprites. The Sno-Cat character is moved with an interrupt routine as described in the 64 version, but the trees and rocks are scrolled in BASIC. The Sno-Cat is steered with the Z (left) and X (right) keys.

The major difference from the 64 version is that the VIC Sno-Cat has no brakes. Instead, you select from ten difficulty levels at the start of the game. This way you don't have to deal with a runaway tractor when you're just learning how to play, but you can still increase the challenge of the game as you become more proficient.

BASIC Program Parts

The BASIC part of the program is broken into five parts: the main routine in lines 150–180, the opening screen in lines 400–880, the instruction screen in lines 1000–1180, the YOU MADE IT screen in lines 1500–1700, and the YOU CRASHED routine in lines 3000–3240. All of these, except the instruction screen, call the tree scroll routine.

If you would like a copy of the program on disk (64 version only), send $3, a formatted disk, and a self-addressed, stamped mailer to:

Andy Keplinger
251 Upper Grassy Hill Rd.
Woodbury, CT 06798

See program listings on page 134.

www.commodore.ca

www.commodore.ca

What Makes Good Software?

Fred D'Ignazio, Associate Editor

Courseware Report Card

What does a parent look for when shopping for software for the family? What kind of software should a teacher look for?

One of the best guides to educational software is the *Courseware Report Card*, published in Compton, California. (The *Report Card* was recently absorbed by PC Telemart of Washington, DC.)

When parents and teachers go shopping for new software they can ask the same questions asked by the *Report Card*'s reviewers.

For example, what is the software's:

- subject area
- age/grade level
- medium (tape, disk)
- publisher's address
- type of program (drill, tutorial, etc.)
- type of computer (computer, memory size, etc.)
- price

Next a parent or teacher should evaluate a program's:

- performance
- error handling
- documentation
- ease of use
- appropriateness
- educational value

"Documentation" and "ease of use" are self-explanatory categories. "Performance" means how flexible the program is, how quickly it fills screens full of words and pictures, how quickly it saves and recalls information, etc. "Error handling" is what the computer does when a person types an unexpected button (for example, RESET). "Appropriateness" is how appropriate the software's methods are for teaching a certain subject. For example, it would not be especially appropriate to teach map-making with an all-text program. "Educational value" means what learning the program actually promotes. Does it help a child learn how to follow directions, determine cause and effect, learn how to spell or do addition problems, tell his left hand from his right?
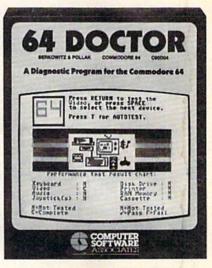
MAKE YOUR COMMODORE 64† WORK LIKE CRAZY, WHILE YOU KEEP IT FROM DRIVING YOU NUTS.

PractiCalc 64™ Only $54.95*

PractiFile™ Only $54.95*

64 Doctor™ Only $29.95*

One way to make your Commodore work like crazy is to give it a shot in the arm with PractiCalc 64. It's the most potent electronic spreadsheet you can buy at the least cost. You can track expenses, inventories, investments. Make charts and graphs. Keep mailing lists. Project profits. Sort alphabetically or numerically, instantly and easily, of course. And at the price, it can pay for itself the first time you use it.

Another way to make your computer system work like crazy is to boost it with PractiFile. It's like having a library full of information you can call on for just about everything while you pay practically nothing. A fully professional data base for your Commodore, it can handle mailing list entries by the thousands. You can change records, numbers, methods

of filing, and do plenty more—all at the touch of a key and the blink of an eye. And all at a price to make your eyes light up. And, it integrates with PractiCalc.
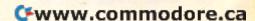
The way to keep <u>you</u> from going crazy when something in your computer system goes haywire is to treat it with 64 Doctor. You know that maddening feeling you get when something's wrong but you don't know exactly what? Well, kiss it goodbye with this powerful medicine. It's an inexpensive and versatile diagnostic program that takes the guesswork out of troubleshooting your computer system. With simple, plain-English instructions. Use it to test your Commodore's RAM memory, RS-232 port, keyboard, video, audio, joystick, printer, data set and disk drive. Think of it as low-cost health insurance for your computer.

PRACTICORP™
No-Nonsense Software

The Silk Mill, 44 Oak St., Newton Upper Falls, MA 02164 • (617) 965-9870

*Prices are suggested retail for disk versions of these programs. Actual prices can vary. Tape versions are slightly lower.
†Commodore 64 is a trademark of Commodore Business Machines Inc.   © 1984 Practicorp International, Inc.

C-www.commodore.ca

A Look At Tess

The EPIE (Educational Products Information Exchange) and Teachers College of Columbia University have just published *TESS* (*The Educational Software Selector*). *TESS* is the size of our Roanoke, VA, phone book, and is filled with descriptions and evaluations of educational software. Many of the evaluation and description categories are the same as those employed in the *Courseware Report Card*. However, *TESS* also recommends that you look at a program's:

- uses (home, classroom, both? remedial, curriculum?)

- grouping (is the product best used alone or in a group?)

- lesson length (time to complete a lesson or session with the computer)

- record keeping of the learner's progress

- copy protection

- printed components—workbooks, stickers, maps, etc.

- user site (other users of the product who can be contacted)

More Basic Guidelines

To this list of basic guidelines I would like to add some of my own: First, when you shop for software, you should look for a *warranty* card. A warranty gives you some guarantee that you can return the software or get a full refund, in case the software is defective. Most warranties range from 30 to 90 days.

Second, you should look for a *replacement disk offer*. If, for some reason, the disk gets damaged, you will not want to pay full price for a replacement disk. You should be able to order a disk from the company at a low price.

Third, you should look carefully at the package and the documentation. The writing in both should be clear and low-key. It should explain, in simple language, what the program does. Both should have *full-color screen shots* that show you exactly how the program looks when it is running on the computer.

Fourth, look to see if the software offers *self-teach* and *help* features. Newer programs are substituting these features for lengthy printed manuals and guidebooks.

When you turn on the computer, a menu of program options should appear on the screen. One of the options is a tutorial. If you choose this option, the program teaches you about itself. Advanced programs feature several tutorials that gradually teach you more and more sophisticated features of the program. You learn by hands-on

experience with increasingly sophisticated and complex versions of the program, until, at last, you are using the program at the "expert" level.

Newer programs also offer a HELP key. Whenever you are stymied and don't know what to do next, you just type HELP. The program hopefully will show you just what you need to know to make your next move. The instructions or hints should be in regular English, not in cryptic programmer jargon.

Last, you should look for a *hotline* phone number. If the program does something unexpected, or if it looks like you have made some kind of disastrous error, the dealer (where you bought the software) may not be able to help you. In that case, you will need to go back to the source—the program's manufacturer—to advise you on what to do next.

Do-It-Yourself Guidelines

Software is a swiftly evolving medium whose potential we have barely begun to tap—or understand. Software is hard to evaluate because we are not sure what it can already do.

All the guidelines I've given you are basic rules of thumb that you can use when you purchase a new program. But these are not the only guidelines you should follow. In fact, with a little effort, you can create your own.

If software were a static form of art, entertainment, instruction, and communication, we could draw neat boundary lines around it. Critics and experts would quickly emerge and describe what lies within the boundary lines and what lies without. Fixed standards to help us tell good software from bad could be created and need never be changed.

But experts and standards, though already in great supply, are only of limited use. Software is moving and growing too quickly for us to accept any rigid standards or any critic's pronouncements as gospel.

This is why it is important for you to be your own critic. How do you and your family react to the software? What do you notice that's good (or bad) about it? What are your gut feelings? What do you notice about the software that is a surprise—something you never read about and didn't expect?

Getting Intimate With Your Software

As serious computer users, we are all becoming on-the-job software critics and software experts. We have never been to software school or taken a course in "Software Appreciation," but we are gradually discovering software we like and soft-

ware we don't like. We are learning to recognize features in a program that we find attractive.

Many of our opinions and observations about software are highly personal. That's because software is not like a head of lettuce or a light bulb. There are many different kinds of lettuce and light bulbs, but most of us agree on what makes a head of lettuce rotten or delicious, or a light bulb bright or burnt out.

But software is different. It is more like music, movies, and books. Or like statues, paintings, and plays. Twenty people might work with the same program and have twenty different reactions. Ultimately, our impression of a given program will be very personal, and, in part, subjective. Our means of judging the software will be based on our gut reaction and an intuitive comparison between this particular program and the dozens of other programs we alone have used.

And perhaps "used" isn't even the right word. We don't *use* software the way we use tissue paper or scotch tape. And we don't *consume* software the way we consume potato chips or soda pop.

Instead, we establish a *relationship*. We meet the software, get acquainted, then play with it or work with it. As we become more familiar with the software, we become more intimate with it. Our relationship ceases to be conscious and becomes almost second nature. Our relationship with the software evolves each time we get to know it at a new level.

Charm, Humor, And Wonder

My family and I have used dozens of software packages in the seven years since we got our first microcomputer. Most of the programs we have used have been educational. Here are some of the informal criteria I have come up with from watching the way my family interacts with software.

First, *charm*. Charm is one of my most important criteria for evaluating new programs. Charm is something intangible, but when a program has it, you know it. It makes you smile; it makes you feel good when you use the program. It makes the program delicious. It might be a cute little bit of music, letters that look like puffy doughnuts, or the way the program "talks" to you in a voice all its own.

Second, *wonder*. I like programs that surprise me, startle me with their intelligence and their imaginative responses to my stumbles and bumbles, my muddling thought, and my queries. Programs provoke a sense of wonder in me when they outpace my expectations. The wonder might come from the speed of a program's footwork, or the neat, original way it handles a mundane task, or from watching it do a backward handspring I

didn't think was possible.

Third, *humor*. I always prefer a light, airy program to one that is somber, dreary, and dull. I'm not sure that I want a program to be a wise guy all the time. That would get tedious and irritating. But a program that makes me and my kids laugh, even occasionally, is a definite hit in our house.

Fourth, *process vs. product*. I used to tell people that the computer is just an immature *appliance*. That one day, when it has grown up, it will be just as invisible as a refrigerator is when we want to get at something inside.

But I'm not sure I believe that anymore. A lot of the fun of computing, for me, is not in getting the job done but in the doing. It is in process, not product. Programs that charm me the most don't just reward me if I get the right answer, they reward me even before I get the right answer. The reward I get isn't just in completion of the task, it is the fun and joy I feel on the way.

Last, *treating me like a human being*. I think that the fault with many programs is that their ideal user is not a human but a *computer*. These programs treat the human being like a computer instead of like a person. Exchanges between the person and the computer are reduced to transfers of data—dry, stuffy streams of bits and tidbits that turn computers on but leave people cold.

But I like a program with pizzazz. The more personality a program gives a computer, the more human, warm, humorous, and intimate the exchange—and the more I like it. After all, I am a human being. I am not a computer that operates on only one, dry, cerebral wavelength. Instead I am a creature of many wavelengths and many dimensions. I am a sensing, feeling being, and I like to be treated as such—by other people *and* by programs. A program that recognizes my human nature makes me more productive, and also happier and easier to get along with.

What Do You Think?

I would like to hear your response to this month's column.

What are important criteria that *you* use to evaluate new software? Did you find those criteria here?

What makes programs extraordinary, and what are some extraordinary programs? What are some features you think *should* be included in programs but still haven't been invented?

What do you think about the "do-it-yourself" method of software evaluation? What guidelines do you recommend?

Send your comments to:

Fred D'Ignazio
2117 Carter Road, SW
Roanoke, VA 24015

Sprite Magic:
An All-Machine-Language Sprite Editor

Charles Brannon, Program Editor
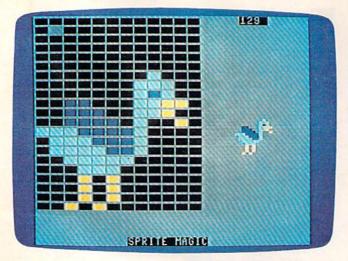
Sprites make animation on the 64 fun and easy to program. But actually drawing and creating sprites with graph paper can be tedious. "Sprite Magic" simplifies their creation, and lets you concentrate on the artistic aspects of sprite design.
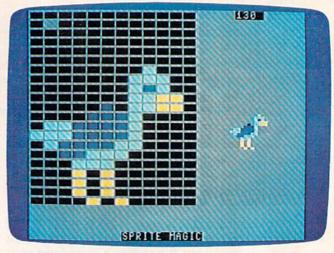
Most of the what you've read about sprites covers how to program them: setting them up, protecting memory, moving and animating them, and using them in games. But sprite design is usually left up to you.

A sprite is defined by 63 binary numbers. The one bits (on) represent solid pixels. Zeros (off) represent blank areas in which the screen background is visible. Normally, you sketch a sprite on a grid 24 squares across and 21 squares high. This is three bytes per row (8 bits*3 bytes=24 bits) and 21 rows of bytes (3*21=63 bytes). But after you've drawn the sprite, you have to convert the squares into binary, and then into decimal so that you can put the numbers in DATA statements.

There are utility programs that will do the conversion for you, even editors that let you clear and set squares with a joystick. Since you're using a computer, other functions can be supported to let you clear, invert, reflect, reverse, shift, and test out your sprite. The more work the computer does, the less you have to think in terms of binary numbers.

Sprite Magic offers the best features of most sprite editors, including true multicolor mode,

Alternating between two similar shapes creates the illusion of motion.

and pulls it off with the speed and power of an all-machine language program. Sprite Magic's style (and even some of the coding) is similar to "Ultrafont +," which appeared in last month's issue. Many of the commands are the same, so you can get up to speed quickly. If you've learned how to use Ultrafont +, it won't take much to become comfortable with Sprite Magic.

Typing It In

Since Sprite Magic is an all-machine-language program, you cannot enter it as you do a BASIC program. We've included MLX, a machine language editor, in this issue for use with this program. If you haven't used it before, read the explanation of its use and commands.

After you've typed in MLX, run it, and answer the prompts of Starting Address and Ending Address with 49152 and 51851, respectively. You'll then be ready to start typing in Sprite Magic. Type in each line from the program listing. The last number in each line is a checksum, so type it carefully. If the checksum you've typed matches the checksum computed from the line you typed, a pleasant bell tone tells you you've typed the line correctly. If the number doesn't match, a buzzer warns you to re-enter the line. This way, you should be able to type in Sprite Magic correctly the first time.

Assuming you've typed and saved Sprite Magic, here's how you get it up and running. If you used the filename "SPRITE MAGIC", type:

LOAD "SPRITE MAGIC",8,1   (for disk)
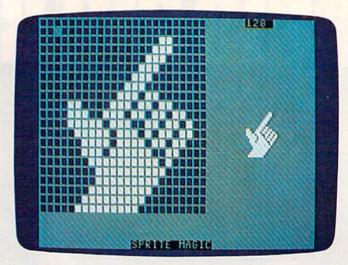
or

LOAD "SPRITE MAGIC",1,1   (for tape)

Be sure to add the ,1 to the end. After the computer comes back with the READY message, type NEW and press RETURN. This resets some important memory locations, but leaves Sprite Magic in its protected cubbyhole at $C000.

Doodle

Activate Sprite Magic with SYS 49152. Instantly, the main screen should appear, with a large 24 × 21 grid. The grid is a blow-up of the sprite you are editing. The actual sprite will be seen to the right of the grid. The flashing square within the large grid is your cursor. Move the cursor with either the cursor keys or with a joystick plugged into port 2. To light up a blank spot (in other words, to turn that pixel on), press either the space bar or the joystick fire button. If the square is already lit, it will turn dark. This signifies that the pixel has been turned off. The button or space bar thus toggles points on or off. You can draw your sprite quite easily in this

Sprites can be used as custom cursors and pointers.

manner. One fine point: With the joystick, you can hold down the fire button and move the cursor. If the first point you change was set, then the fire button will continue to set points as you move the joystick, regardless of the other points' original state. If the first point you change was empty, then you can hold down the fire button and move about, clearing anything the cursor passes over. Notice how any changes are immediately visible in the actual sprite.

If you've just entered Sprite Magic, the grid is probably full of garbage pixels. To clear out the grid for a new picture, press SHIFT-CLR/HOME. You now have an empty area (a fresh canvas, so to speak) to draw upon. You can press CLR/HOME without holding down SHIFT to home the cursor to the upper-left corner of the grid.

Does the cursor move too slow or too fast? To change the velocity (speed) of the cursor, press V. Answer the prompt with a number key from 0 (slow) to 9 (very fast).

Shift, Expansion, And Symmetry

Sometimes when you're drawing, it's necessary to reposition the shape within the grid. The first two function keys let you shift the sprite shape around within the grid. If you shift something out of the grid, it wraps around to the opposite side. The f1 key shifts right, f3 shifts down. Use the SHIFT key along with the function key to move in the opposite direction: f2 moves the sprite shape left, f3 up.

After you've drawn something, press F. Instantly, the sprite is flipped upside-down. Press it again to flip it back over. Remember F as the command for Flip. Now try M, for Mirror. The shape you've drawn is mirrored left to right. Of course, if you've drawn something symmetrical,

you may not see any change.

Now try CTRL-R or CTRL-9. The sprite will become reversed. Every square that was on is now turned off, and vice versa.

A sprite can also be expanded or contracted either horizontally or vertically, or both horizontally and vertically. The X and Y keys on the keyboard let you do this. Press X to switch from wide to narrow, or vice versa. Press Y to switch from tall to short, or vice versa. Regardless of your choices, the main grid will not change size or proportion.

An unusual command is Symmetry. I added this command after some suggestions that many shapes are symmetrical from left to right, as if a mirror were put in the middle of the grid. To enter the Symmetry mode, press the back-arrow (←) key (found in the upper-left corner of the keyboard). Now, every square drawn on one side will be instantly mirrored to the left. Blank squares are not copied over, though, so you cannot erase in this mode. This command is not only quite useful, but is also a great deal of fun to play with. To return to normal editing, press the back-arrow key again.

Notice the number in the upper-right corner of the screen. This is the sprite page number, which can range from 0 to 255. You start out at the top of the sprite memory. The + and − keys are used to go forward or backward through sprite shapes. Press the minus key and see how you now have a new shape in the grid.

There is a limit to how far back you can go. If you have no BASIC program in memory, you can step back to sprite page number 32. However, character information resides in sprite pages below 128. You can still clear the page and draw a sprite shape on pages below 128, but it won't really register. To be safe, use only the sprite pages from 128 on up. If you have a program in memory, Sprite Magic will not let you step back past its end. This protects your program from being accidentally overwritten by a sprite shape. If you want maximum space available for sprite shapes, be sure to NEW out any BASIC program before you SYS 49152. You'll sometimes want to keep a program in memory, however. We'll show you why a bit later.

Programming note: The sprite page number, when multiplied by 64, gives you the starting memory location for the 63 numbers representing the sprite.

Put It In The Buffer

You might use Flip to design two views of a shape, such as a spaceship pointing in two directions. Draw one freehand, then do the other with Flip. Mirror can be used to design separate left

and right views as well. But what you first need is a way to copy the original shape to another sprite area. One way to do this is to copy the sprite shape to an area of memory (a buffer). You can use + or − to step to another sprite page, then copy the buffer to the sprite. This, you may remember, is the way you copy characters with Ultrafont +. The same keys are used in Sprite Magic. Press f7 to copy the sprite to the buffer. The grid flashes to affirm this. Then go to the sprite page where you want to put the copy and press f8 (SHIFT-f7). The shape in the buffer replaces any shape already in the sprite grid. You can also use the buffer as a fail-safe device. Before modifying an existing sprite, press f7 to save it in the buffer. Then, if you mangle the sprite, or accidentally erase it, you can recall the previous shape from the buffer.

Computer Disney?

The buffer is also useful for animation. Since you can change sprite pages so easily, you can also use Sprite Magic as an animation design tool. Cartoons make only minor changes between frames. Too much change makes the animation jerky. So put the first frame into the buffer, copy it to the next area, then make a change. Put the new image into the buffer, copy it again to a new area, then make another small change. Continue in this fashion as you build up a whole series of frames. Put different but similar shapes on adjacent pages, then hold down plus or minus to step through the shapes. As with cartoon animation, you will get the illusion of motion. Use a cursor velocity of 9 for maximum speed. So even if you don't care to program sprites, Sprite Magic is a fun tool for making moving cartoons.

A Bit Of Color

The normal drawing mode lets you set or clear points, but in only one color. If you're willing to give up half as many horizontal points, you can have four colors to work with. Multicolor mode lets any square be one of four colors, but gives you only 12 pixels across instead of 24. This is because two dots are grouped together to give four combinations. The colors come from four memory locations:

Pattern	Color location	
00	53281	Background color register
01	53285	Sprite multicolor register 0
10	53287- 53294	Sprite color registers
11	53286	Sprite multicolor register 1

There are two multicolor sprite registers, which are shared between all sprites (in programming, but not in Sprite Magic, you can have eight sprites on the screen at the same time). The bit

An illusion, done with symmetry. Is it a vase or talking heads?

pattern marked 10 is unique to each sprite, and comes from that sprite's own color register. Pattern 00 is blank, and whatever is underneath the sprite shape will show through.

The reason for this sojourn into bits and addresses is that only the 10 bit pattern has a unique color for that sprite. If you're designing several sprites for a game, remember that anything drawn in that color can be changed individually for each sprite. Squares drawn with bit pattern 01 or 11 will be colored from two locations shared by all sprites.

Many sprite editors let you see how the sprite would look in multicolor, but you still have to pair up the pixels yourself, and keep track of binary bit pairs. No fun! Instead, Sprite Magic offers a multicolor mode. When you press f5, the screen instantly changes. Each square in the grid is now rectangular, two squares wide. The cursor has also been enlarged, and can be moved about as before in the new grid. But the way you set and clear points has been changed, since you are now working with four colors.

Multicolor Palette

The fire button or the space bar always sets a point, but you have to tell Sprite Magic which color you are currently drawing in. The number keys 1 to 4 select the drawing color. The number you press is one number higher than the binary value of the bit pairs in the table above. The 1 key, for instance, chooses the 00 bit pair, which represents the background color. In practice, you are choosing from a palette of four colors. The 1 key can be used when you want to erase, although the fire button can still be used to toggle points on and off.

When you press a number key from 1 to 4, the border color changes to remind you which

color you're drawing with. If you want to change one of the four colors, hold down SHIFT while you type the number. The prompt ENTER COLOR KEY appears. Now you have to enter another key combination. Press CTRL and one of the number keys from 1 to 8, or hold down the Commodore key and one of the number keys from 1 to 8. These are the same key combinations you use to change the text color in BASIC. You can also change the screen background color by pressing the letter B on the keyboard until the color you want appears.

Some Sprite Magic commands act strangely in multicolor mode. For example, a shift left or shift right (done with the f1 and f2 keys respectively) moves the sprite over by only one bit, which changes the color assignments. In general, you must press f1 or f2 twice to preserve the same colors. Pressing the M key (for Mirror) reverses the bit pairs, so that every 01 becomes a 10. The effect is that colors 2 and 3 are exchanged. The R key (Reverse) also inverts the bits, so that 01 becomes 10, 10 becomes 01, 00 becomes 11, and 11 becomes 00. Colors 2 and 3 are switched, as well as colors 1 and 4.

If you want to go back to normal (non-multicolor) mode, press the f6 key (SHIFT-f5). There's nothing to prevent you from designing both normal and multicolor sprites on different pages.

If you changed colors in the multicolor mode, some of the colors in the normal mode may have been changed. You can alter these colors as in multicolor mode. Press SHIFT-1 to change the color of the empty pixels, and SHIFT-2 to change the color of the on pixels. (You'll be prompted to press a color number key after each SHIFT-1 or SHIFT-2 combination. Remember to press either CTRL or Commodore simultaneously with the color key.)

Mobilizing Your Sprite

If you want to try out your sprite in action, press J (for Joystick). You can now move the actual sprite around with the joystick. The speed of movement depends on the current cursor velocity. When you've finished putting your sprite through its paces, press the fire button to return to Sprite Magic. Also, if you want to test the animation while you are moving about, hold down the SHIFT key to step forward through the pages of your defined sprites, or the Commodore key to step backward. You can lock the SHIFT key to keep the animation happening while you move around.

Saving Your Sprites

After all your work, you surely want to save your

www.commodore.ca

creations on tape or disk for future use. You can save an individual shape, or all the sprites. Press S (for Save), then either D (Disk) or T (Tape). Next, enter the filename. You'll be asked if you want to "Save all from here?" If you press N, for No, then only the current sprite you are working on is saved. If you press Y for Yes, then every sprite from the current sprite to sprite 255 will be saved. Thus, if you want to save a range of sprites, be sure to use the minus key to step back to the first sprite you want saved.

To recall your sprites, press L. The Load command loads everything that was saved. If you're loading in more than one sprite, be sure you step backward far enough with the minus key so that all the sprites will fit between the current sprite and sprite 255. The sprites load starting at the current sprite page number. After you press L, enter T or D for Tape or Disk.

Making Sprite DATA

If you're a programmer, you're probably more interested in DATA statements. That way, you can use BASIC to READ and POKE the numbers into memory. If you have some kind of "DATA maker," you can run it on the memory used by the sprite in Sprite Magic (again, the memory location is the sprite number times 64). But Sprite Magic has a special DATA maker of its own. It's similar to the Create DATA option in Ultrafont +, but it's been enhanced.

Press CTRL-D to create a series of DATA statements from the current sprite in memory. Just tap the key, or you'll get hundreds of DATA statements as the key repeats. Sprite Magic will create eight DATA statements, with eight bytes per line. The last byte is not strictly used. Sprite shapes are made from 63 bytes, but the sprite areas are padded out so they will conveniently fall in 64-byte ranges. To create DATA statements for another sprite, use the + or – key to move to the correct sprite page, then press CTRL-D again.

If you have a program already in memory, the DATA statements are appended to the end of the program, starting with the next available line number. To add DATA statements to an existing program, then, first load Sprite Magic. Type NEW. Load your BASIC program, and SYS 49152 to enter Sprite Magic. You can then load in sprite shapes and use CTRL-D to add those DATA statements to the end of the BASIC program in memory.

You can check to see that these DATA statements were added by exiting Sprite Magic (press CTRL-X) and typing LIST. Your program should have eight new DATA lines for each sprite pattern. If there was no program in memory, the DATA statements form a program all their own,

starting with line 1. If you want, you can save just the DATA statements to tape or disk, using the normal SAVE command.

To exit Sprite Magic and return to BASIC, press CTRL-X. You can also use RUN/STOP-RESTORE.

Quick Reference Chart

B: Cycle through background colors
F: Flip sprite upside-down
J: Move sprite with joystick. Press button when done.
L: Load sprite from tape or disk
M: Mirror sprite from left to right
S: Save sprite(s) to tape or disk
V: Set cursor velocity
X: Toggle X expansion on/off
Y: Toggle Y expansion on/off

CTRL-D: Create DATA statements
CTRL-R or CTRL-9: Reverse sprite
CTRL-X: Exit to BASIC

+: Next sprite page
–: Previous sprite page
CLR/HOME: Home sprite editing cursor
SHIFT-CLR/HOME: Erase grid
Space bar or fire button: Set/clear points
CRSR keys or joystick in port 2: Moves cursor
Back arrow: Symmetry mode
Keys 1-4: Select drawing color for multicolor mode
SHIFT 1-4: Change a drawing color

f1: Shift right
f2: Shift left
f3: Shift down
f4: Shift up
f5: Multicolor mode
f6: Normal mode
f7: Store sprite to buffer
f8: Recall sprite from buffer

See program listing on page 138. ⬛

Robert Sims, Assistant Editor

UPLOADING

On-line data bases were created to disseminate large amounts of information to a broad segment of the population. Before the advent of the computer, public and private libraries served this function.

Most commercial data bases, in fact, still are organized along the same lines as a library. Material is published, collected and collated, then it is catalogued and made available for downloading.

Tapping into the vast amount of information in data bases, while undoubtedly valuable, is essentially a passive activity: Someone else has made available this data which you retrieve.

If you want to move from being a passive information receiver to being an active information provider, then according to the established order you must compose your message, find a publisher who agrees that the world needs to know, and get the work distributed to libraries and data bases. Only then will the world get your message.

Direct Delivery

In the less structured domain of home telecommunications, the path from creator to user is more direct, and more immediate. If you have a home computer and a modem, all you need in order to tell the world is a short course on how to upload the message.

Whether your message is a private note to a friend, a checkbook balancing program, or the first chapter of your new novel, there are three ways to upload it.

The simplest, and slowest, way is to log on to a bulletin board or network and type the data in manually.

Another way is to prepare the data before going on-line, then load it into the terminal software's capture buffer and transmit it after you log on.

The third way is to prepare the data off-line, and use the automatic upload feature of your terminal software to upload the file directly from disk.

By coincidence, these three methods are used to compose and send three forms of communication. If you want to tell everybody something, you can leave a message on a bulletin board. If your message is private, send it as electronic mail, which is just a private letter delivered electronically. These two forms are informal and temporary. A message will stay on a bulletin board for only a few days, and electronic mail usually is erased as soon as it is read.

To deliver a long program or a long text file which will be available to others on a more permanent basis, you upload it to special upload/download areas set aside on the networks and bulletin boards.

On-Line Word Processors

To allow you to compose a bulletin board message or electronic mail on-line, bulletin boards and networks have built-in word processing routines. Most bulletin boards have a line editor, which permits entry and editing of one message line at a time. The simplest line editor will only allow you to edit by deleting your mistakes and retyping the line. If you want to change word order or insert text, you must delete everything back to the point where you want to insert, then retype the rest of the message.
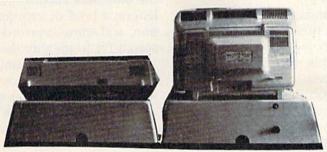
This limited word processing capability is a function of available memory and processing speed. Bulletin boards are usually run on home computers, and memory is at a premium. A full-featured word processor would take up too much of the RAM needed to hold the main bulletin board software.

The information networks such as CompuServe, Delphi, and The Source use banks

of mainframe computers. Since memory is not a major limitation, the networks provide full-featured word processors which can be used to compose messages on-line.

Even though these word processors are much more sophisticated than line editors, it's still not possible to edit your text by simply moving the cursor around on the screen of your computer, inserting or deleting words at will. Because of the relatively slow transmission rates of 300 and 1200 bps (bits per second) involved in telecommunications, your computer and the host cannot interact fast enough to allow editing text directly on the screen.

For that reason, on-line word processors require that you edit your message in pieces, without seeing the whole. Each editing operation, whether deleting a single letter or inserting a paragraph, is performed by sending a unique command in an exact syntax. The process is necessarily time-consuming and ungainly. The inexperienced user can call up help files which briefly describe these commands, but to become proficient a user must buy the printed manuals.

Composing Off-Line

Because of these disadvantages, experienced users prefer to compose and edit their messages off-line, then upload them after they log on.

The usual way to do this is to compose the message using a word processor. Then the message file is converted to a form which the terminal program and the host computer can process, and this upload file is stored on disk. (For a discussion of the conversion process, see last month's column.)

The upload file is loaded into the terminal software's capture buffer. Then, after logging on and accessing the on-line word processor, the user waits for the host's prompt to enter the message text, then uploads the file from the capture buffer instead of typing it in.

Uploading The Message Text

Generally, the on-line word processor will accept text a line at a time, and will send a special prompt character when it is ready to receive the next line. If your text file is sent all at once, the host will receive only one line, and the rest will be lost. For that reason, the file must be uploaded one line at a time.

Some terminal programs provide a function key which, when pressed, uploads one line of text. The user waits for the prompt character and pushes the key, repeating the process until the end of the file is reached.

Other terminal software does this automatically, if the user knows which prompt character

the host sends. The user doesn't have to put the upload file into the capture buffer; the terminal software will take it directly from the disk. The user selects one-line-at-a-time uploading, and the software asks for the host prompt character.

The user types in the prompt character, and the terminal software then takes over. Every time the host sends a character the terminal software compares it to the prompt character provided by the user. If the characters are the same, the terminal software sends a line of text from the upload file. After the file has been uploaded, the terminal software returns program control to the user.

Uploading Programs

Uploading programs and text files into data base areas where they will be downloaded by other users is more complicated than uploading messages.

First, the program must be converted to a format which can be uploaded (see July's column) and stored on disk.

Then, when the user is on-line and accesses the upload area of the bulletin board or network, the host will ask for information to be included in the data base catalog. This may include file length, a brief description of the program's function, and some keywords which other users can search for to identify the file.

The user who is uploading may be asked to provide a filename by which the uploaded file will be stored on the host system.

All this information must be complete and correct in order for other users to locate, download, and make use of the uploaded program.

File Extensions

Often, the host will give the uploader a choice of file extensions, or suffixes appended to the filename proper. These extensions identify the format in which the file is stored. If the wrong extension is attached to a filename, it may make the file impossible to download. (For a discussion of file types, see last month's column.)

For example, an extension of .BIN should mark a file as a program which has been converted to ASCII representations of hexadecimal numbers. It may be a BASIC program or a machine language program.

If an uploader mistakenly labels a program file with a .DOC or .TXT extension, a user who downloads this file will convert it as though it were text, and the end result will be useless garbage.

After the catalog is entered, the user will be prompted by the bulletin board or network as to the actual upload procedure. This can be one line

at a time, or the entire file may be uploaded automatically.

Providing Documentation

Besides the basic catalog information for a file, a responsible uploader will provide documentation for any program he uploads.

This documentation should include detailed instructions for using the program, and should be uploaded as a text file with a .TXT or .DOC extension. Its filename should be similar to the program's name so downloaders can connect the two, and reference to the program it documents should be included in the descriptive notes that go with the documentation file's catalog entry.

The documentation file can be a text file created with a word processor, or it can be generated as a program listing consisting entirely of REMark statements, converted, and uploaded as a program file with a .BIN or .IMG extension. Such a file can be downloaded, converted, and LISTed, allowing the end user to get a printout without using a word processor.

But however the documentation is handled, it is important that it accompany the uploaded program. Failure to document software before offering it to other users is a major problem in home telecommunications. There are thousands of pro-

grams in the public domain for the VIC and 64, free for the taking, but too often users obtain a program and then find to their dismay that the program contains no instructions.

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

ⓒ www.commodore.ca

C. Regena

Using A Printer

A printer is one of the more important peripherals you'll buy or use. To get the most out of your printer, it's best to know how to use it with your VIC or 64, and to understand what it can do.

This month, we'll work with the Commodore 1525 printer. Much of our discussion, though, applies to other printers as well.

Talking To The Printer

To use any printer, you must first OPEN a communication channel to the printer. OPEN1,4 is the command to open file number 1 to the printer, which is device number 4. You may use any number from 1 to 255 for the file number, but always use 4 for the device number. If you use a file number over 127, the printer will double space. I often use the number 4 for both, so I only have to remember one number for the command—OPEN4,4.

To get a program listing on the printer, use CMD*n* (where *n* is the same number as the file number in the OPEN command) to transfer the control from the computer to the printer. Now add the command LIST. Instead of listing on the screen, the listing will appear on the printer. This can be accomplished in one line:

OPEN 4,4:CMD4:LIST

You may list certain sections of the program by using the standard LIST commands:

LIST −200	all lines up to 200
LIST 300–320	lines numbered 300 to 320
LIST 800–	lines from 800 to the end of the program

To turn off the printer and return to the screen, type:

PRINT#4:CLOSE4

which properly closes the file.

A Printer Is For More Than Listings

A printer should serve for more than listings, however. Other common applications are word processing, mailing labels, graphics, or anything you'd use a typewriter for.

To access your printer from within a program, use the OPEN and CLOSE commands we've already discussed. Then, to write something or PRINT with the printer, use PRINT#*n* where *n* is that same file number. We'll use 4 in our example, but you can use any number from 1 to 255. Just make sure that you use the same number in the OPEN*n*,4 command, PRINT#*n* command, and CLOSE*n* command. PRINT#*n* is typed with no spaces and cannot be abbreviated. Here is a sample of how the commands are used.

```
10 REM PRINTER 1
20 OPEN4,4
30 PRINT#4,"HELLO THERE"
40 PRINT#4
50 CLOSE4
60 END
```

PRINT#4 is similar to the PRINT command you're already familiar with, except the printing will go to the printer instead of the screen. PRINT#4 alone prints a blank line. If you want to print a specific message, type a comma after PRINT#4 and then your message in quotes. You may also print variables and functions and use TAB and SPC. (Note: A bug in the VIC's operating system prevents the use of TAB or SPC *immediately* following a PRINT# command. The result is a SYNTAX ERROR. To remedy this, print anything—for example, a null character ("")—following the PRINT# command but preceding a TAB or SPC.) Here are some sample PRINT# commands.

```
10 PRINT#4,X
20 PRINT#4,N$;" AND ";M$
```

C-www.commodore.ca

C-www.commodore.ca

```
30 PRINT#4,A;B,A+B
40 PRINT#4,TAB(12);SIN(C)
```

If you look on a chart of ASCII codes, you'll see the character code numbers for various symbols. PRINT CHR$(65) will give you the symbol corresponding to number 65, or A. You can use the same code when using the printer, PRINT#4,CHR$(65). However, some of the character codes are used to change some of the printer procedures. For example, CHR$(13) is a carriage return. (Note: Some of the codes may be different with different printers. The printer manual should have a list of character codes with their functions.)

On the 1525, we can use expanded print (double-width) character mode. To enable this larger printing, PRINT#4,CHR$(14). Now anything sent to the printer will be in the expanded print. To get back to standard printing, PRINT#4,CHR$(15). The following program illustrates these two styles of printing.

```
10 REM PRINTER 2
20 OPEN4,4
30 PRINT#4,CHR$(14)"TITLE"
40 PRINT#4,CHR$(15)
50 PRINT#4,"BACK TO STANDARD PRINTING"
60 PRINT#4
70 CLOSE4
80 END
```

Printing Math Tests

Earlier in the school year, my daughter's class was reviewing multiplication facts. One of the drills was to see how many problems could be done in one minute. As the children were practicing the speed drills at home, I noticed that the test used was always the same; the students were actually memorizing the answers in order—not really doing the multiplication. But with a computer and a printer, a different test can be produced each time by printing the problems in a random order.

Program 1 prints a randomly generated test of multiplication problems. The multiplicands and multipliers may be numbers from 0 to 9. The program simply prints random numbers across the top, then random numbers below, and then draws lines under the problems.

Line 20 opens file number 4 for the printer and prints a blank line. Line 30 makes sure the printer is in standard size printing and prints a place for the student's name. Line 40 prints a blank line, then line 50 prints a place for the date and a place for the score. SPC(30) means to leave 30 spaces horizontally. Line 60 prints blank lines. Line 70 sets the printer to expanded print to print the title then tabulates over 12 spaces. Line 80 returns the printer to standard printing size.

The FOR-NEXT loop in lines 90–200 sets up the printing of ten rows of problems. The loop in

lines 100–150 prints two lines of numbers. The loop in lines 110–140 prints ten random numbers with spaces between them across the printed line. Lines 160–180 draw the lines under the problems. Line 210 closes the file to the printer, and line 220 ends the program.

Program 1: Multiplication Test

Refer to the "Automatic Proofreader" article before typing this program in.

```
10 REM MULTIPLICATION TEST         :rem 179
20 OPEN4,4:PRINT#4                  :rem 70
30 PRINT#4,CHR$(15)"NAME[30 P]"     :rem 18
40 PRINT#4                          :rem 72
50 PRINT#4,"DATE[18 P]"SPC(30)"SCORE
   [10 P]"                          :rem 85
60 PRINT#4:PRINT#4                  :rem 104
70 PRINT#4,CHR$(14);TAB(12);"MULTIPLICATI
   ON"                              :rem 161
80 PRINT#4,CHR$(15):PRINT#4         :rem 78
90 FOR ROW=1 TO 10                  :rem 186
100 FOR I=1 TO 2                    :rem 4
110 FOR P=1 TO 10                   :rem 59
120 A=INT(10*RND(0))                :rem 61
130 PRINT#4,"{3 SPACES}"A"{2 SPACES}";
                                    :rem 168
140 NEXT P:PRINT#4                  :rem 66
150 NEXT I                          :rem 30
160 FOR P=1 TO 10                   :rem 64
170 PRINT#4,"{3 SPACES}[2 T]{3 SPACES}";
                                    :rem 109
180 NEXT P                          :rem 40
190 PRINT#4:PRINT#4:PRINT#4         :rem 186
200 NEXT ROW                        :rem 201
210 CLOSE4                          :rem 61
220 END                             :rem 107
```

Since Program 1 uses random numbers to create the problems, it's likely that problems could be repeated and not all possible combinations used. Suppose you want a test that has all combinations of multipliers but in a random order. Since the numbers can be from 0 to 9, there are 100 combinations. (Refer to the sample test on the following page.)

There are several ways to accomplish the task. The way I did it was to use a two-dimensional array, as shown in Program 2. The numbers in the subscripts of the array are the two numbers to be multiplied. Line 20 initializes all elements of the array to be zero. As a problem is chosen randomly, the element is changed to 1 so that problem will not be chosen again.

This time I have printed the problems horizontally, four problems to a line, and 25 rows of problems with double spacing between lines. Also, I used an asterisk to indicate multiplication. If you prefer an X, change the symbol in line 130. You can easily change this to an addition test by changing the title in line 80 and putting a plus sign in line 130.

NAME_____
DATE_____   SCORE_____

MULTIPLICATION

```
0   9   5   2   8   3   1   5   8   4
6   4   0   2   5   5   0   9   2   5

0   5   5   7   8   7   6   3   0   1
0   6   7   0   0   0   2   6   3   7

9   1   9   0   7   4   0   7   2   7
4   6   6   0   9   9   1   3   5   1

8   4   2   5   1   9   6   5   4   2
8   4   8   3   4   9   6   0   4   6

3   5   3   1   7   4   3   8   3   2
8   6   3   0   2   2   3   9   3   3

1   1   2   7   4   7   8   5   1   7
2   5   8   9   2   8   8   7   5   9

1   4   4   7   8   0   4   0   3   2
4   1   7   5   6   4   0   0   7   9

6   1   5   6   0   6   8   5   3   0
2   6   8   0   3   0   0   7   1   8

1   6   8   8   4   5   8   3   7   6
6   8   5   8   8   9   9   3   7   8

2   1   8   5   9   1   0   4   3   8
2   0   7   4   9   1   3   4   7   0
```

NAME_____
DATE_____   SCORE_____

SUBTRACTION

```
3 - 0 =        8 - 6 =        9 - 2 =
4 - 4 =        3 - 5 =        7 - 0 =
2 - 2 =        9 - 3 =        4 - 0 =
2 - 1 =        3 - 3 =        8 - 8 =
9 - 7 =        4 - 2 =        6 - 5 =
3 - 2 =        9 - 4 =        4 - 3 =
7 - 1 =        8 - 1 =        6 - 3 =
5 - 4 =        8 - 9 =        7 - 3 =
2 - 0 =        3 - 1 =        7 - 2 =
5 - 2 =        5 - 5 =        1 - 1 =
8 - 2 =        5 - 5 =        8 - 4 =
7 - 7 =        8 - 8 =        8 - 5 =
6 - 0 =        9 - 8 =        5 - 0 =
5 - 3 =        8 - 3 =        5 - 1 =
1 - 0 =        6 - 6 =        7 - 4 =
8 - 7 =        6 - 2 =        6 - 1 =
4 - 1 =        6 - 1 =        5 - 1 =
9 - 6 =        7 - 5 =        7 - 6 =
```

Program 2: Revised Multiplication Test

Refer to the "Automatic Proofreader" article before typing this program in.

```
10 REM MULTIPLICATION TEST 2      :rem 229
20 FOR I=0 TO 9:FOR J=0 TO 9:N(I,J)=0:NEX
   T J,I                          :rem 204
30 OPEN4,4:PRINT#4                :rem 71
40 PRINT#4,CHR$(15)"NAME[30 P3"   :rem 19
50 PRINT#4                        :rem 73
60 PRINT#4,"DATE[18 P3"SPC(30)"SCORE
   [10 P3"                        :rem 86
70 PRINT#4:PRINT#4                :rem 105
80 PRINT#4,CHR$(14);TAB(12)"MULTIPLICATIO
   N"                             :rem 103
90 PRINT#4,CHR$(15):PRINT#4       :rem 79
100 FOR ROW=1 TO 25               :rem 232
110 FOR P=1 TO 4                  :rem 14
120 A=INT(10*RND(0)):B=INT(10*RND(0)):IF
    {SPACE}N(A,B)=1 THEN 120      :rem 105
130 PRINT#4,A"*"B"="SPC(12);      :rem 235
140 N(A,B)=1:NEXT P               :rem 26
150 PRINT#4:PRINT#4:NEXT ROW      :rem 9
160 PRINT#4:CLOSE4                :rem 95
170 END                          :rem 111
```

The subtraction program is similar to the previous program; however, the number to be subtracted must be less than or equal to the first number. I used numbers from 1 to 9 for the first number, so there will be 54 combinations. This time the printing is done with 18 rows of three problems each.

Program 3: Subtraction Test

Refer to the "Automatic Proofreader" article before typing this program in.

```
10 REM SUBTRACTION                :rem 147
20 FOR I=1 TO 9:FOR J=0 TO I:N(I,J)=0:NEX
   T J,I                          :rem 221
30 OPEN4,4:PRINT#4                :rem 71
40 PRINT#4,CHR$(15)"NAME[30 P3"   :rem 19
50 PRINT#4                        :rem 73
60 PRINT#4,"DATE[18 P3"SPC(30)"SCORE
   [10 P3"                        :rem 86
70 PRINT#4:PRINT#4                :rem 105
80 PRINT#4,CHR$(14);TAB(13)"SUBTRACTION"
                                  :rem 136
90 PRINT#4,CHR$(15):PRINT#4       :rem 79
100 FOR ROW=1 TO 18               :rem 234
110 FOR P=1 TO 3                  :rem 13
120 A=INT(9*RND(0)+1):B=INT((A+1)*RND(0))
    :IF N(A,B)=1 THEN 120         :rem 42
130 PRINT#4,"{3 SPACES}"A"-"B"="SPC(15);
                                  :rem 53
140 N(A,B)=1:NEXT P               :rem 26
150 PRINT#4:PRINT#4:NEXT ROW      :rem 9
160 PRINT#4:CLOSE4                :rem 95
170 END                          :rem 111
```

The procedure for the division test is similar to the multiplication test. Two numbers are chosen randomly. The problem for division will be the product of the two numbers chosen divided by the first number. There are 81 possibilities, so the test is printed in 27 rows of three problems each. Since the dividend may be either a two-digit or a one-digit number, I changed the product A*B to a string variable Q$, then adjusted Q$ so the problems would line up properly. If A*B is a one-digit number, the length of Q$ (which is a space plus the number) will be 3, and I put an-

other space in front of the number. Line 150 adjusts Q$.

The 1525 does not have a "divide by" symbol, so I designed my own symbol. If you have a different printer, you may have the symbol already, or you can define graphics in a different manner. If you use a 1525, refer to pages 21–22 of the manual. You can define your own character by coloring in dots of a 7 × 7 matrix, converting the graph to numbers, then using DATA statements in the program to define the character. Lines 20–30 of Program 4 define a character (D$) to be the division symbol. In line 160, before using D$, I set the printer to graphics mode with CHR$(8). After printing the special graphics character, CHR$(15) returns the printer to standard printing.

Program 4: Division Test

Refer to the "Automatic Proofreader" article before typing this program in.

```
10 REM DIVISION                        :rem 170
20 DATA 136,136,136,170,136,136,136
                                       :rem 184
30 FOR I=1 TO 7:READ D:D$=D$+CHR$(D):NEXT
                                       :rem 246
40 FOR I=1 TO 9:FOR J=1 TO 9:N(I,J)=0:NEX
   T J,I                               :rem 208
50 OPEN4,4:PRINT#4                     :rem 73
60 PRINT#4,CHR$(15)"NAME[30 P]"        :rem 21
70 PRINT#4                             :rem 75
```

```
80 PRINT#4,"DATE[18 P]"SPC(30)"SCORE
   [10 P]"                             :rem 88
90 PRINT#4                             :rem 77
100 PRINT#4,CHR$(14);TAB(15)"DIVISION"
                                       :rem 202
110 PRINT#4,CHR$(15):PRINT#4           :rem 120
120 FOR ROW=1 TO 27                    :rem 236
130 FOR P=1 TO 3                       :rem 15
140 A=INT(9*RND(0)+1):B=INT(9*RND(0)+1):I
    F N(A,B)=1 THEN 140                :rem 213
150 Q$=STR$(A*B)+" ":IF LEN(Q$)=3 THEN Q$
    =" "+Q$                            :rem 117
160 PRINT#4,"{3 SPACES}"Q$;CHR$(8)D$;CHR$
    (15)A"="SPC(15);                   :rem 26
170 N(A,B)=1:NEXT P                    :rem 29
180 PRINT#4:NEXT ROW                   :rem 238
190 PRINT#4:CLOSE4                     :rem 98
200 END                                :rem 105
```

Making A Letterhead

Another use for your printer is to make your own letterhead. You can use different styles of print to make the letterhead more interesting. If your printer doesn't have graphics capabilities, you can make up a design using regular symbols. If you have a printer that can support graphics, consult your printer manual to see how to draw something. The 1525 has the graphics symbols displayed on the keys of the VIC and 64. It can print any of these symbols with the standard PRINT# command.

Program 5 uses the standard Commodore graphics symbols to make a design for a letterhead. I used the symbols just as if they were in standard PRINT statements. B$ is defined in line 30 as the reverse space, which is a solid block. TAB() is used to start the printing in a different column (rather than the very first column at the left). SPC() spaces over the specified number of spaces.

CEDAR HACKERS USERS GROUP
P. O. BOX 1502
CEDAR CITY, UTAH 84720

The name used in this letterhead is fictional and for illustration purposes only, but it might provide some ideas for a letterhead for your users group, or a family or Christmas newsletter.

Program 5: Letterhead

Refer to the "Automatic Proofreader" article before typing this program in.

```
10 REM LETTERHEAD                      :rem 39
20 OPEN4,4:PRINT#4,CHR$(15)            :rem 42
30 B$="{RVS} {OFF}":S$=""              :rem 32
```

www.commodore.ca

```
40 PRINT#4                                :rem 72
50 PRINT#4,S$TAB(21)"{RVS}{3 SPACES}{OFF}
   "SPC(6)B$SPC(5)B$SPC(4)B$SPC(5)B$SPC(6
   )"{RVS}{3 SPACES}{OFF}";               :rem 55
60 PRINT#4,S$SPC(17)"{RVS}£ E*3":rem 106
70 PRINT#4,"{3 SPACES}{RVS}{10 SPACES}
   {OFF}"SPC(7)B$SPC(3)B$SPC(5)B$SPC(5)B$
   SPC(4)B$;                              :rem 179
80 PRINT#4,S$SPC(5)B$SPC(5)B$SPC(3)B$SPC(
   7)"{RVS}{5 SPACES}{OFF}"SPC(4)"E*3
   {RVS} {OFF}£"                          :rem 148
90 PRINT#4,"{2 SPACES}{RVS}{2 SPACES}
   {OFF}{3 SPACES}{RVS}{2 SPACES}{OFF}
   {3 SPACES}{RVS}{2 SPACES}{OFF}"SPC(5)B
   $SPC(5)B$SPC(4)B$SPC(5)B$SPC(4)B$;
                                          :rem 252
100 PRINT#4,S$SPC(5)B$SPC(4)B$SPC(5)B$SPC
    (7)"{RVS} {OFF}{2 SPACES}{RVS} {OFF}"
    SPC(5)B$                              :rem 89
110 PRINT#4,"{2 SPACES}{RVS}{2 SPACES}
    {OFF}{3 SPACES}{RVS}{2 SPACES}{OFF}
    {3 SPACES}{RVS}{2 SPACES}{OFF}"SPC(5)
    B$SPC(10)B$SPC(5)B$SPC(4)B$; :rem 128
120 PRINT#4,S$SPC(5)B$SPC(4)B$SPC(13)"
    {RVS} {OFF}{2 SPACES}{RVS} {OFF}
    {2 SPACES}{RVS} {OFF}{2 SPACES}{RVS}
    {SPACE}{OFF}{2 SPACES}EF3"   :rem 231
130 PRINT#4,"{2 SPACES}{RVS}{2 SPACES}
    {OFF}{3 SPACES}{RVS}{2 SPACES}{OFF}
    {3 SPACES}{RVS}{2 SPACES}{OFF}"SPC(5)
    B$SPC(10)B$SPC(7)B$SPC(4)"
    B$SPC(5)B$;                   :rem 4
140 PRINT#4,S$SPC(4)B$SPC(13)"{RVS}
    {12 SPACES}"                 :rem 222
150 PRINT#4,"{RVS} {OFF} {RVS}{12 SPACES}
    {OFF}"SPC(5)B$SPC(10)B$SPC(5)B$SPC(4)
    B$SPC(5)B$;                  :rem 178
160 PRINT#4,S$SPC(4)B$SPC(3)"{RVS}
    {3 SPACES}{OFF}{2 SPACES}{RVS}
    {17 SPACES}{OFF}E-3"         :rem 193
170 PRINT#4,"{RVS}{15 SPACES}{OFF}"SPC(4)
    B$SPC(5)B$SPC(4)B$SPC(5)B$SPC(4)B$;
                                 :rem 226
180 PRINT#4,S$SPC(5)B$SPC(4)B$SPC(5)B$;"
    {2 SPACES}{RVS}{7 SPACES}{OFF}UCI
    {RVS}{7 SPACES}"             :rem 43
190 PRINT#4,"{RVS}{15 SPACES}{OFF}E4 I3
    {RVS} {OFF}{3 SPACES}{RVS} {OFF}"SPC(
    5)B$SPC(5)B$SPC(5)B$;        :rem 19
200 PRINT#4,S$SPC(3)B$SPC(6)B$SPC(3)B$"
    E2 O3{RVS}{7 SPACES}{OFF}B H{RVS}
    {7 SPACES}"                  :rem 130
210 PRINT#4,"{3 SPACES}Q{SHIFT-SPACE}Q
    {4 SPACES}Q{SHIFT-SPACE}Q"SPC(8)"
    {RVS}{3 SPACES}{OFF}"SPC(6)B$SPC(5)B$
    SPC(6)"{RVS}{3 SPACES}{OFF}";:rem 247
220 PRINT#4,S$SPC(8)"{RVS}{3 SPACES}{OFF}
    "SPC(6)"Q Q{2 SPACES}J*K{3 SPACES}Q
    {SHIFT-SPACE}Q{SHIFT-SPACE}M":rem 201
230 FOR I=1 TO 3:PRINT#4:NEXT I   :rem 233
240 PRINT#4,S$TAB(15)CHR$(14)"CEDAR HACKE
    RS USERS GROUP"               :rem 37
250 PRINT#4,CHR$(15)                :rem 95
260 PRINT#4,S$TAB(33)"P. O. BOX 1502"
                                   :rem 157
270 PRINT#4                        :rem 125
280 PRINT#4,S$TAB(28)"CEDAR CITY, UTAH
    {2 SPACES}84720"              :rem 242
290 PRINT#4:PRINT#4               :rem 157
300 CLOSE4                         :rem 61
310 END
```

REVIEWS

The Commodore 1520 Printer/Plotter

Robert Sims, Assistant Editor

The 1520 is almost a great product. It's an inexpensive plotter which uses special ballpoint pens to draw in four colors on a roll of paper 4½ inches wide.

In its printer mode, the 1520 can print upper- and lowercase text left to right or top to bottom in four sizes, with up to 80 characters per line.

It is capable of plotting intricate designs by addressing 959,040 X-Y plotting points, with .2 millimeters (.0078 inch) between points.

But using the 1520 is a lot like putting a size 9 shoe on a size 9½ foot; you can make it work, but don't plan to do a lot of dancing.

Plotting On Note Paper

The most important drawback is the paper width. While 4½ inches is a good size for note paper and memo pads, it's not much good for anything else.

If you want to use the 1520 for something other than high-resolution doodling, you will have to transfer your designs (by photocopying or some printing process) to full-size paper.

The transfer will bring out the next problem: line quality. Ball point pens produce a thin, uneven line which does not reproduce well. Also, they tend to dry out and skip, leaving gaps in the lines.

To get the best quality, every line must be drawn twice. This is merely inconvenient, however, because the 1520 is easily capable of drawing the same line twice in exactly the same place.

A less important problem is the distance between plotting points. Although .0078 inch seems small on a ruler, it is wide enough to cause a visible stairstep effect that is most pronounced in lines which are almost horizontal or almost vertical. If your needs lean more toward art than precision, this can be considered an interesting effect rather than a shortcoming.

Programming The 1520

Whether the 1520 is easy to use depends on your BASIC programming skills.

Plot and print features are selected by using certain secondary addresses in OPEN statements, in either the immediate or program modes. For example, a secondary address of 0 tells the plotter to print characters:

OPEN 4,6,0

To select character size, you must open a second file with a secondary address of 3:

OPEN3,6,3
and
PRINT#3,0

to select 80 characters per line. To print the text, you then use

PRINT#4, A$
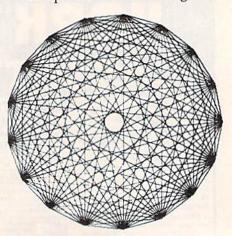
Plotting Commands

A secondary address of 1 tells the 1520 to draw a figure:

OPEN 1,6,1

Plotting is handled by PRINT#1 statements which send commands to move the pen, put it down on the paper, and pick it up.

Your BASIC programming skills will also be needed to debug the manual. While it contains a detailed, illustrated explanation of how to set up the 1520, the manual is seriously flawed in its demonstration programs. For example, the most impressive demonstration is Program 8, Geometric Designs, which plots this circular figure:

Unfortunately, the program contains two errors which cause it to crash:

```
150 PRINT#,"M";0,-260: REM
    ORIGIN PT
220 X2=240+L*SIN(2/N*J* π )
```

Anyone familiar with the syntax of the PRINT# statement could spot the bug in line 150; there is no logical file number following PRINT#. But in line 220, one

REVIEWS

trigonometric function has been erroneously substituted for another. Users who don't know trigonometry are in trouble.

To get the figure above, you must change lines 150 and 220 to read:

```
150 PRINT#1,"M";0,-260: REM
    ORIGIN PT
220 X2=240+L*COS(2/N*I*π )
```

Of the five other demonstration programs I tested, Programs 4,7, and 10 worked. However, in Program 1, Concentric Circles, line 170 should read:

```
170 C=C+1:IF C>=4 THEN C=0
```

In Program 11, Changing Forms, substitute these lines:

```
110 OPEN 4,6 :REM PRINT CHAR
300 XX(I)=X1(I)+K*AX/(M+1)
```
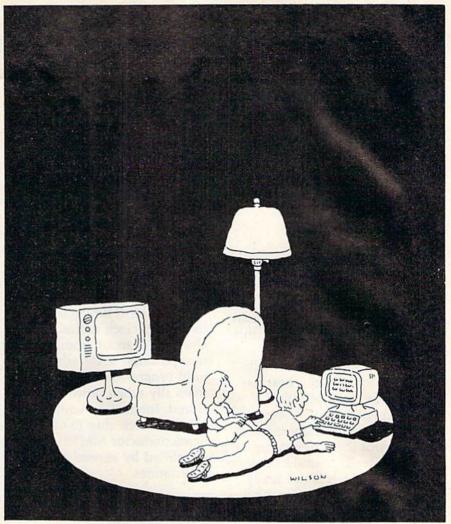
A Lot For The Price

The 1520 is a worthwhile product, despite the inaccuracies in the manual, the small paper size, and the visible stairsteps in plotted lines.

If your needs call for an 80-column printer and you don't care about full-size paper, the 1520 will serve admirably, and for far less than even an inexpensive dot-matrix printer. Although the 80-column text size is small, it is legible and superior to dot-matrix print quality.

If you want to experiment with graphics on a plotter, you won't find another product at anywhere near the price.

*The 1520 Printer/Plotter
Commodore Business Machines
1200 Wilson Drive
West Chester, PA 19380
About $125*

www.commodore.ca

VIC Auto-Clock

Harvey B. Herman
Associate Editor

The VIC Auto-Clock is quite unlike most other plug-in cartridges. If you need a peripheral which can be used as the heart of an intelligent controller, this is just the ticket.

What exactly is an intelligent controller? I can best describe it by an application I heard about recently. Someone wanted to periodically monitor and control the temperature of a greenhouse. This could be done manually, which is not really practical, or by constructing a circuit that performs only one fixed task. A better way, however, would be to use a computer-controlled circuit (intelligent controller) which would not be limited to one task and whose function could be easily modified by software changes.

Several items are needed to build an intelligent controller for a greenhouse. You need something which measures temperature and converts the signal into a digital value. You need a control circuit which can heat or cool on demand. Finally, you need a clock which keeps track of real time, so you know when to take measurements.

Where does the Auto-Clock fit in? As the name implies, it keeps track of time, but that's not all. It can be used to turn the VIC on at any future date and time, and then run a program which would perform the monitor and control functions.

Of course, you must supply the additional hardware necessary for the latter functions.

Basically, the Auto-Clock is a timer which remembers the day, date and time, even with AC power off, and can turn on and off controller accessories with its alarm function, much like many clock radios. Also included is a small amount of battery powered RAM which can hold programs that run automatically when power is restored. Interaction with the internal clock chip (National Semiconductor MM58167A) is simplified by several ROM subroutines.

Running It Through The Paces

A little preliminary work was required before I could test the Auto-Clock. Following the directions in the manual, I cut an old extension cord and wired it to the in/out pads inside the cartridge. When the VIC is plugged into the extension socket and the extension plugged into the wall, the Auto-Clock can turn the VIC on and off under computer control. Doing it this way, rather than an alternative method in the manual, required no permanent change in the VIC.

For review purposes, I wrote a program which illustrates some of the Auto-Clock features, but isn't particularly useful. The program was saved in the Auto-Clock RAM with a ROM SYS call. Any program in its RAM is run automatically whenever the VIC is activated, unless RETURN is pressed. Because its RAM is battery backed, the program will be there as long as the battery lasts, even with no AC power supplied.

```
10 GOSUB 100:REM SET ALARM 10 SECO
   NDS AHEAD
20 A=PEEK(45080):REM RETRIEVE A FR
   OM STORAGE
30 A=A+1
40 PRINTA
50 POKE 45080,A:REM STORE A
60 FOR I=1 TO 2000:NEXTI:REM DELAY
70 SYS 41023:REM VIC OFF NOW/VIC O
   N 10 SECONDS FROM SET
80 END
100 REM SET ALARM
110 X=45074:REM BEGIN ALARM REGIST
    ERS
120 POKEX,0:POKEX+1,0:POKEX+2,0:RE
    M SET DELAYS,HOURS,MINUTES
130 POKEX+3,10:REM SET SECONDS
140 POKEX+4,0:REM ALARM RELATIVE T
    O REAL TIME
150 SYS 41002:REM SET ALARM NOW!
160 RETURN
```

The program turns on the VIC every ten seconds, adds one to a variable, prints the variable on the screen and then turns the VIC off.

Progressive Peripherals has done a nice job with this uncommon piece of hardware. I could not test everything, but, with one exception, all the functions I tried worked. February 29 did not come up normally in 1984. However, the company claims that leap years are handled correctly by the resident firmware. Perhaps I have an earlier version of their program.

Also, I would rather have seen the hardware schematics included with the manual. They are indispensable if service is ever needed. Other than that, I recommend it to anyone interested in intelligent controllers.

VIC Auto-Clock
Progressive Peripherals And Software
6340 West Mississippi Ave.
Lakewood, CO 80226
$129.95

BusCard II: The Magic Box

Ian A. Wright

The BusCard II, a small rectangular box slightly larger than a cigarette package, allows the Commodore 64 to access IEEE drives and serial drives, IEEE printers and parallel printers—all via the same interface.

I've been using the BusCard as an interface to connect my 64 to 4040, 2031, 2031sl, and 1541 drives (in various combinations) without any problems. Attaching both a Commodore 4023 (IEEE) and an Epson MX-80 (parallel) printer to the 64 and then printing to either printer also worked without problem.

Even beyond that, I hooked up two printers, two drives (both IEEE and serial), the 64 and a 4032/8032 PET, and was able to selectively work with each of these machines linked together. This configuration can be a real time saver for software designers, reviewers, or programmers.

Monitor And BASIC 4.0 Included

Another useful feature provided by the BusCard is the built-in monitor. The monitor is activated by SYS 8, and provides register and memory display, a simple assembler and disassembler with automatic addressing, save, hunt, fill, load, and transfer memory, and printing of the disassembly. Although far from a full-featured assembler this monitor does allow rapid access for those little "fixes" that are part of programming in machine language.

BusCard also has BASIC 4.0 built-in so that commands are simplified. For example, to scratch a file you now type: sC "filename". All the commands of the later PET/CBM machines are enabled by SYS61000, and for a student (or anyone) who works with various machine types this is a boon.

The 36-page manual that accompanies the BusCard uses photos and text to explain how to attach the interface, how to set the switches, how to use the monitor, and explains the BASIC 4.0 commands in detail. There is even a lengthy description of how the BusCard works that will be of use to the advanced programmer.

A common question that arises when discussing IEEE interface units for the 64 is: "Will it load program X?". I've loaded and run hundreds of commercial and public domain programs from my 4040 drive and there were very few that would not load through the BusCard. Any problems that occurred could invariably be traced to excessive disk checking in programs that were heavily protected.

Upgrade Improvements

For those who own the earlier version of the BusCard, there are a few visible changes on the new BusCard II. The mini-switches, which allow you to select various devices, have been moved so as to be more accessible, although most users will set them only once for their specific system.

The cartridge slot has also been moved to the right side of the board, and the IEEE slot is now at the rear. According to Batteries Included, there is less strain on the 64 motherboard while changing cartridges with this configuration. The IEEE and printer connector wiring now comes from the rear of the 64.

One change quickly becomes apparent when the new BusCard II is installed because there is now only one clip that has to be connected to the inside of the 64. This wiring change is an indication that the latest version of the BusCard II is even more transparent than its predecessors.

Similarly, this new version also lets you reset the mini-switches without resetting the machine (turning off/on). If you should decide to change input/output between IEEE and serial devices, you do not lose whatever program you have in memory.

Having BASIC 4.0, a monitor with assembler and disassembler, a complete manual, the ability to load programs such as *Blue Max* with one keypress, a loading speed four to five times faster than the 1541, and the ability to use the "brainpower" of a 4040 dual drive—all of these features come from the BusCard II. This is truly a magic box.

Music Writer III For The VIC-20
David Florance, Programming Assistant

Music Writer III, the latest music software from Nüfekop, turns your VIC-20 into a smart musical instrument. The program, designed by David Henry Funte, lets you write, edit, review, save, or load songs and melodies.

People who wonder about the memory limitations of an unexpanded VIC will be surprised at what this package can do in less than 4K. *Music Writer III* is written entirely in machine language, which makes it fast, powerful, and versatile. And it's available on tape or disk.

Easy To Use
After loading and running the program you see a window in the center of your screen. This is the measure of music being played or written. Pressing f1 loads an example piece into memory, which is a smorgasbord of familiar tunes. The example demonstrates the versatility of the program.

To write music, first clear memory by pressing f8. A star indicates which note is current. The program needs to know four things: Is the note flat or sharp? (Default is natural); What is the note? (letter name); the octave (1 or 2); and the note value (1/4, 1/8, or 1/16).

Quick Editing
After a note is selected, the program asks if the information input is correct. If it is, the note

goes into memory and the process is repeated until the tune is finished. If it is not to your liking, the previous input is erased and the process is started again. The program erases one note at a time, so only the values just put in are erased.

After your melody is written, you can review the work by using the cursor keys. This quick editing feature allows change of any note, octave, or duration.

A Musical Tutorial
If you have a background in music, this program is very friendly. But suppose you know almost nothing about music. Does the program lend itself to the person without musical training? It does.

With a few hours of practice, your VIC can sound like a real song machine. Just keep in mind that a quarter note (1/4) is twice as long in duration as an eighth note (1/8), which is twice as long in duration as a sixteenth note (1/16). You can put your favorite songs on your VIC, or write your own original compositions.

Naturally there are some limitations. It is possible to save the tunes to tape or disk, but should you want to use the music in a program of your own, data statements will have to be generated.

Also, in writing tunes that are slower in tempo (speed) the program will not display note values more than the quarter

note. This is easily overcome by adding values (e.g., three quarter notes equal one dotted half note). Since the program has a capacity for 500 notes, there is plenty of room for elaborate melodies.

Playing With Music
Music Writer III can do a number of things for both experienced and novice musicians. It can increase a novice's ability to write tunes, understand basic theory, and hear and sharpen listening skills.
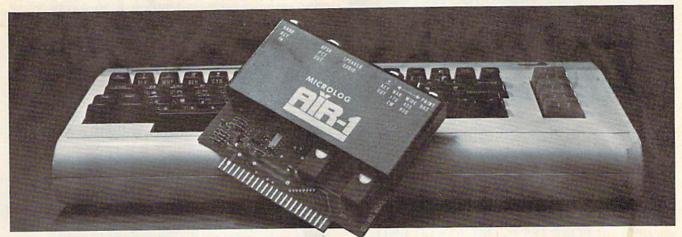
If you're a more experienced composer, it provides a fast way to compose and edit a melody when you may not be able to get to another musical instrument or a piece of staff paper. And the program lets you hear your melody immediately.

The versatility of *Music Writer III* makes it one of the better music programs available for the VIC-20.

Music Writer III
Nüfekop
P.O. Box 156
Shady Cove, OR 97539
$16.95 (tape)
$19.95 (disk)

Make Your Commodore Radio Active

The MICROLOG AIR-1 cartridge will turn your VIC-20 or C-64 into a complete Radio Teletype and Morse code terminal. Connect a shortwave radio and you'll be watching text readout from weather stations, news services, ships and HAM radio operators all over the world. A whole new use for your home computer. The AIR-1 contains both program in ROM and radio interface circuit to copy Morse code and all speeds and shifts of radio teletype. Plus the on screen tuning indicators mean you never have to take your eyes off the video for perfect tuning.

For Ham radio use, the Air-1 will also send and receive RTTY/CW with AFSK/PTT & ± CW keying outputs.

The AIR-1 will even tell you what Morse speed you're copying and provide built-in send/receive code practice!

With keyword or manual printer control for permanent paper copy, you won't miss a single bit of the action.

If you've been looking for something to spice-up your computing, try the ultimate "peripheral" and connect your computer to the AIR-1.

The complete AIR-1 for the VIC-20 or "64" is $199. (With 4 mode AMTOR, $279.) See it at your local dealer or call Microlog Corporation, 18713 Mooney Drive, Gaithersburg, Maryland 20879.
TEL: (301) 258-8400. Telex: 908153.

MICROLOG
INNOVATORS IN DIGITAL COMMUNICATION

C-www.commodore.ca

Bug-Swatter:
Modifications And Corrections

● The VIC version of "Frantic Fisherman" (June) works as listed, but the instructions for tape users on page 60 contain a typo. The second POKE should be 631 (the keyboard buffer) rather than 531 (the middle of the input buffer). Line 700 should read POKE 198,1:POKE 631,131:END.

● "Sea Route To India" (March) runs as listed. However, readers Harry R. Meyer and David W. Kenvin discovered that it is possible for the water supply to run out, and even fall to a negative number, with no ill effects. Adding the following line will allow you to perish more gracefully when the water runs out:

```
11015 IF WT<1 THEN PRINT"{CLR}OUT OF WATE
      R":PRINT"{DOWN}YOU DIE OF THIRST.":
      GOTO 17000                    :rem 57
```

● "File Converter" from "Speedscript Revisited" (May) works as described in the article, but will occasionally report an I/O error where there is none. If you check your disk directory, you should find that the file was converted, in spite of the message to the contrary. To fix the program change line 380:

```
380 SYS(ADR):IF(PEEK(144)AND191)=0THENPRI
    NT"{DOWN}DONE.":GOTO280        :rem 184
```

● Two characters were accidentally cut from line 30 of the 64 version of "Spelling Critter" (June). The correct line is:

```
30 S=54272:V=54296:AD=54277:SR=54278:HF=5
   4273:LF=S:SD=54276              :rem 117
```

● A printer's gremlin deleted an apostrophe from line 1030 of "Castle Dungeon/64" (June). Because it happened to a PRINT statement, it does not affect the running of the program, but does cause the Proofreader checksum number to appear to be incorrect. The line should have been:

```
1030 PRINT"{2 RIGHT}A BEAST. PRESS THE 'L
     ' KEY FOR A "                 :rem 198
```

● Line 80 of the VIC version of "Cassette Beeper" (May) was misprinted. The second to the last number (a nine) should be a 169. In addition, the −1141 in line 75 should have a comma between the first two 1s (−1,141).

● "3-D Tic-Tac-Toe" (June) contains no programming bugs, but suffers from a conceptual flaw. Several mathematically-inclined readers have informed us that if the first player takes the middle position and makes no mistakes in subsequent moves, the second player always loses. One way of correcting this would be to rewrite the game to use a 4 × 4 × 4 board, rather than a 3 × 3 × 3. An easier fix would be for both players to agree that neither will place a marker in the middle space, unless it is necessary to block (or complete) a tic-tac-toe.

● "Ultrafont +" (July) contains a few inconsistencies. The original Ultrafont, published in COMPUTE!'s First Book of 64 Sound and Graphics, could only redefine the uppercase/graphics character set. Since many people design entirely new character sets, Ultrafont + also lets you alter the lowercase character set. Two commands that are case-sensitive were not adjusted, however.

The Fix command, which restores a character from its image in ROM, will always replace the character with its uppercase image, even if you are working in lowercase. More troublesome is that in lowercase, the DATAmaker (CTRL-D) will list almost every character pattern, even those that you haven't changed. This is because Ultrafont + compares the redefined character set to the uppercase/graphics set in ROM, so naturally most of the lowercase set does not match. If you redefine the entire character set, this problem is of no import. ⬚

www.commodore.ca

ML Mailbag

This month we'll answer some more reader questions:

Automatic RUN

Q: *I've noticed that commercial ML programs often use an "auto-start" technique where the mere act of loading the program causes it to run. Could you please explain how this works?*

Bob Burdick

A: Aside from the degree of copy protection that this technique permits, it also has real value to the programmer. If you use a disk drive, you can simply load in one of your own customized *boot* programs and it can then call in another program automatically and start it running. (The term *boot* refers to the phrase "pulling itself up by its own bootstraps.")

There are several ways to take control of the computer following a disk LOAD. One way is to alter the *stack*. This is the section of memory between addresses 256-511 which holds all active RTS (ReTurn from Subroutine) addresses as two-byte numbers. Each time you JSR (Jump to Sub-Routine) in ML or GOSUB in BASIC, the address following the JSR/GOSUB is pushed onto the top of the stack. If you change the address on the top of the stack, you change where the computer will go when it encounters the next RTS/RETURN at the end of a subroutine.

By placing a new address on top of the stack, you can force the computer to execute your own ML routine. A boot program can be saved in a special way so that it includes the stack. Then, when loaded, the altered stack covers over the normal one and, when the LOAD is finished, the computer obediently pulls off the artificial return address and goes there (where your special ML load-and-run routine awaits). For specific information on how to set this up, see Dan Carmichael's article "Autoload" in an upcoming issue.

A second way to boot is to put your custom ML loading routine into addresses $02A7-$02FF and then make addresses $0304-5 point to $02A7. All this is then saved. $0304 is a special "warm start" pointer. Following a LOAD, the computer always goes where this address tells it to—normally that will be a routine which prints READY and puts the machine in BASIC standby mode, waiting for further instructions. However, you can tamper with this pointer, giving your own ML routine control of the computer.

DATA In Machine Language

Q: *Data tables. I've tried every combination of numbers, dollar signs, and numbers signs, but my monitor, Supermon, keeps giving me error messages. What is the proper way to enter data tables?*

Chris Solar

A: *Data tables* are to ML programming what *DATA statements* are to BASIC. They are zones of non-instructions, places where pure information is stored for the program's use. For example, suppose your program needed to know the ages of your three children:

```
10 DATA 5,7,9
20 PRINT "LAURIE IS";:READ X:PRINT X
20 PRINT "TOMMY IS";:READ X:PRINT X
20 PRINT "BILL IS";:READ X:PRINT X
```

In BASIC, the computer ignores any information following the word DATA unless it comes upon a READ statement. In ML, you've got to know where your program starts and ends and then locate data tables *outside* of the program itself. There is no automatic sliding past data. Coming upon a data table located *within* an ML program, the computer will try to interpret the table as instructions. The consequences are unpleasant. Likewise, the computer does not keep track of which data items have been read. That, too, is up to you. By convention, ML data tables are stored at the end of an ML program.

To print the childrens' ages (not their names) in ML:

```
10 PRINT = $FFD2
20 LDY #0
30 LOOP LDA TABLE,Y; LOAD ITEM
40 BEQ END; ZERO MEANS FINISHED
50 JSR PRINT
60 INY
70 JMP LOOP
80 ;             END OF LOOP
90 END RTS
100 ;      -- DATA TABLES --
110 TABLE .BYTE "5 7 9: .BYTE 0; THE
     CHILDREN'S AGES
```

Obviously, this ML isn't using Supermon or any other monitor. It's an advanced assembler which allows variable names (see line 10), comments, and simplified data entry (any numbers or letters following the .BYTE command tell the assembler to put them directly into memory as is—they're not to be assembled as ML commands). You write your ML the same way you'd write a BASIC program, using line numbers, etc. (This is the LADS assembler from my new book, *The Second Book of Machine Language*.)

Since ML written and listed in this form is far easier to read and study, we'll be using these conventions from now on in this column.

However, if you're assembling from a monitor, the .BYTE instruction is not available and you must enter the direct memory mode via the .M command. After you finish writing your ML program, exit the assembly mode and type .M XXXX XXXX to display a section of memory following the ML program itself. Then you can enter your data items directly:

.M 0378 35 37 39 00 FF FF FF FF
($35 is the ASCII code for 5.)

Your error was trying to enter data while in the .A assembler mode of the monitor. Simple assemblers will try to turn any information into

6502 commands; they cannot know that you want certain numbers to remain as pure numbers, pure information.

To enter letters of the alphabet with LADS, you would program:

10 TABLE .BYTE "ABCDE

and to do the same thing in a monitor assembler, you would enter the .M memory mode and type the ASCII code for the letters:

.M 0378 41 42 43 44 45 00 FF FF

When using a monitor assembler, how do you know where the table is located, where to LDA from? You can either plan where you're going to put the table before writing the program, or go back and reassemble over the instructions which reference it after you find out where the table will be located.

If you have any questions that you'd like to see answered in this column, please write to Machine Language For Beginners, P.O. Box 5406, Greensboro, NC 27403.

Enhancing Your VIC With The Super Expander

Commodore's Super Expander for the VIC-20 is a versatile cartridge that allows you to do hi-resolution and multicolor mode drawings, adds eight more colors, adds function key and music commands, and provides an additional 3K of RAM memory. If you don't have a Super Expander, or if you seldom use the one you own, here's a look at what you've been missing.

After inserting the Super Expander cartridge into the expansion port, you've got not only an additional 3K of usable RAM, but much, much more. It adds extra commands to your BASIC vocabulary which allow you to create works of video art in the hi-res mode. With new commands such as DRAW, CIRCLE, PAINT, POINT, and CHAR, you can draw boxes, circles, or just about any shape. The PAINT command allows you to fill your work of art with color. And the CHAR command lets you mix text with your hi-res drawings. You can also set four sound tones and the volume simultaneously.

With the RPOT, RPEN, and RJOY commands you can read the paddles, light pen, and the joystick with a single command. These commands save programming time and memory because you no longer have to write your own subroutines for these applications.

Using The Super Expander

As with all cartridges, turn off your VIC, carefully insert the cartridge into the expansion port, then turn the VIC on. The first message displayed is 6519 BYTES FREE.

Unlike the 8K or 16K expanders, the Super Expander does not affect screen and color memory locations. It uses memory locations 1024 to 4095. Thus, screen memory (7680–8191) and color (38400–38911) are unchanged.

Any programs written for the unexpanded VIC which POKE or PEEK screen or color memory will run successfully with the Super Expander. The beginning of BASIC is moved from 4096 to 1024. But, because of the VIC's ability to relocate BASIC programs, this shouldn't present any problems with programs written for the unexpanded VIC.

Let's look at the Super Expander commands and see what they do:

GRAPHIC—prepares the screen for graphics use.

SCNCLR—clears the graphic screen area.

COLOR—allows you to set the screen, border, character, and auxiliary colors.
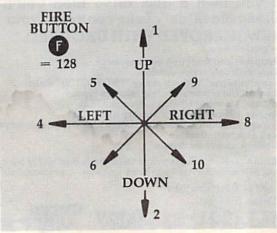
REGION—sets character color only.

DRAW—draws a straight line between two points.

POINT—allows you to turn on a single dot (pixel).

CIRCLE—draws a circle, ellipse, or arc.

Directional Values With The RJOY Command

C-www.commodore.ca

PAINT—fills in an enclosed area with color.

CHAR—allows you to put text on the graphics screen.

SOUND—lets you set four tones and the speaker volume simultaneously.

These additional functions will read the screen or a memory location and return a value:

RGR—displays the current graphic mode.

RCOLR—returns the value in a color register.

RDOT—returns the color of a single point (pixel) on the screen.

RPEN—reads the current position of the light pen.

RJOY—reads the position of the joystick.

RSND—returns the value in a sound register.

Hi-Res And Multicolor Modes

With the GRAPHIC command, you can draw in multicolor mode, hi-resolution mode, or a combination of the two. A major difference between each of these modes is horizontal resolution. To demonstrate, enter the following one-line program:

```
10 GRAPHIC2:DRAW2,512,300TO512,700
```

This program draws a vertical line down the center of the screen in the hi-res mode. Note the width of the line. When drawing in the hi-res mode, the resolution is fine (the lines are thin). Now change the GRAPHIC2 to read GRAPHIC1 and run it again. This will draw the same line in the multicolor mode.

You'll see that the horizontal resolution is halved (the line is twice as thick). Although multicolor mode lets you choose from four colors (as opposed to two in hi-res mode), you do sacrifice resolution.

The differences in resolution in these modes also affect text printed to the screen. Enter this line to print ABC on the screen using the hi-res mode:

```
10 GRAPHIC2:CHAR9,9,"ABC"
```

In this mode, the letters print correctly. Now change the GRAPHIC2 to GRAPHIC1 and run the program again. This changes the mode to multicolor, and as you can see, the letters are somewhat distorted. It's best not to mix multicolor graphics and text on the same screen. However, you can use GRAPHIC3—mixed hi-res and multicolor—and text at the same time.

Reading The Joystick

Reading a joystick, paddle, or light pen is simple with the Super Expander. For example, RJOY reads the joystick, and returns a value. Enter and RUN the following BASIC program:

```
10 PRINTRJOY(0):GOTO10
```

Now move your joystick in various directions to see how this command works.

Reading the paddles and light pen is done in much the same way as the joystick. RPOT is used to read the paddles, and RPEN to read the position of the light pen.

A Musical Expander

With the Super Expander, you can also play music by typing directly on the keyboard, or with PRINT statements within a BASIC program. A nice feature of the music option is the fact that it is *interrupt driven*. This means that music can be played while the program is doing other things—almost like running two programs simultaneously.

Other features include the ability to display the note characters on the screen as they are played, selection of octaves, tempo, and rests, and playing the notes sharp or flat.

Programmable Function Keys

The eight function keys are already programmed with the Super Expander. When a function key is pressed, a Super Expander keyword (such as GRAPHIC, COLOR, CIRCLE, etc.) is printed. Although each of the keys is assigned a keyword, they can be programmed to suit your own needs.

With the use of the KEY command, you can assign any string or command to one of the function keys up to 128 characters long, including cursor or color controls.

To illustrate some of the Super Expander's capabilities, try this demonstration program. It will show you how your VIC can be a much more powerful machine.

Super Expander Demo

```
10 GRAPHIC2:SCNCLR:COLOR1,3,0,2    :rem 147
20 CHAR1,6,"CIRCLES"               :rem 136
30 CIRCLE 2,512,512,200,280:GOSUB500
                                   :rem 162
40 REGION4:CHAR1,5,"COLOR FILL"    :rem 92
50 REGION2:PAINT2,512,512:GOSUB500 :rem 26
60 SCNCLR:REGION0:CHAR1,7,"LINES"  :rem 48
70 REGION6:DRAW2,200,200TO800,200:GOSUB50
   0                               :rem 191
80 REGION0:CHAR1,7,"BOXES"         :rem 57
90 REGION6:DRAW2,800,200TO800,800TO200,80
   0TO200,200:GOSUB500             :rem 185
100 REGION5:CHAR1,1,"INDIVIDUAL PIXELS"
                                   :rem 158
200 Q=15:R=250:S=500:P=400:O=300:FORA=1TO
    300                            :rem 255
210 CO=INT(RND(1)*Q)+0:X=INT(RND(1)*S)+R:
    Y=INT(RND(1)*P)+O              :rem 171
220 REGIONCO:POINT2,X,Y:NEXT       :rem 98
230 REGION0:CHAR9,6,"THE END"      :rem 164
499 END                           :rem 125
500 FORT=1TO1500:NEXTT:RETURN      :rem 142
```

String Search

Glen Colbert

This timesaving machine language utility searches through string arrays looking for a match much faster than its BASIC equivalent. For the VIC and 64.

Although machine language is fast, BASIC is generally preferable when you are writing a program to handle lots of strings: names, addresses, recipes, lists in general. BASIC has built-in string and array functions that make it easy to handle large volumes of information.

It is frustrating, however, to have to wait while the program searches through a few hundred entries looking for a match. The longer the list, the slower BASIC becomes.

String Search is fast because it is written in machine language (ML), although you don't have to understand ML to use it.

Special Instructions

There are a couple of things you have to do before using the program:

1. The first and second variables defined in the target program must be strings. To be safe, put them in the first few lines. And the second string must be the "match" you're looking for.

2. The string array to be searched must be the first array DIMensioned. An integer array containing the same number of elements must be the second array DIMensioned. The integer array will contain flags that indicate a match was found.

Program 1 is the BASIC loader for String Search. When you RUN it, the ML routine is located to the top of BASIC memory and the pointers are reset, protecting it from BASIC. Program 1 can be incorporated into your own pro-

grams or loaded and run as a separate program before loading your own data management program.

To access the search routine, SYS (PEEK (55)+256*PEEK(56)). The ML routine is relocatable. If you prefer, you can put it up at $C000 (49152) on the Commodore 64.

Program 2 is a test of String Search. After you enter, SAVE, and RUN Program 1, RUN Program 2. First, an array containing 300 elements is set up. BASIC then searches for a match and you see how many jiffies it took (a jiffy is one sixtieth of a second). Next, the ML routine is used. You may be surprised at how much faster you get the results.

How It Works

The search method used in this routine is quite simple. When it is called, the first operation is to swap out a portion of the zero page ($D9-$E9) into the cassette buffer. The length of the string to be checked for is put into $D9 and the address of the string is set into $DA-$DB. Next, addresses $DC-$DD are set to point to the zero element of the integer array. Addresses $E0-$E1 are set to point to the three bytes of string array information (length, low byte of address, and high byte of address) for the zero element of the string array. Things are now in order for the processing loop.

The first step in the processing loop is to increment the pointers for the arrays that are being worked to the next element. For this reason, the zero element is not searched. The information for the string array element being worked is moved to $E5-$E7. $E5 is checked for a null (string = "") and if it is null, the zero page information is put back in and returned to BASIC. A counter for the search string ($E2) and one for the searched

string ($E3) are set to zero and the search begins.

Whether there is a match or not is determined by these counters. If the search string counter is equal to the length of the search string, then there has been a match. If the searched string counter is equal to the length of the searched string, then there was no match. In either case, the routine sets the value in the integer array and returns to the main loop to try the next element of the array.

If the counters do not match, the accumulator is loaded with the first character of the search string. This is compared against each element of the searched string until a match is found. Then the second character of the search string is compared against the next character in the searched string and so on until the counter equals the length of the search string. If a match is not found, then the search string counter is reset (but not the searched string counter) and the program loops back.

See program listings on page 133.

www.commodore.ca

Disk Purge
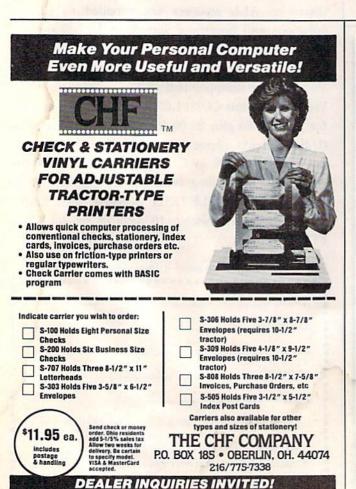
Daniel Weiner

Get rid of unwanted disk files or recover deleted files with this utility for the VIC and 64. "Disk Purge" works with Commodore 2031, 4040, 1540, and 1541 drives.

Like many other computer users, I go to a local user group meeting once a month and come home with three or four disks full of great software. Un-

fortunately, some of the programs are not really great. There are times when the next day is spent typing "S0:FILENAME", deleting useless files and programs.

Disk Purge simplifies this process. It can be used on all Commodore computers, including the 64 and the VIC, and it works with 2031, 4040, 1541, and 1540 disk drives. It should not be used on the 8050 or 8250 because, unlike the other drives, these have more than one directory track.

www.commodore.ca

Looking At The Directory

If you study the program listing, you'll learn a bit about how the disk drive stores information about programs. First, the program asks which drive you want to purge, and initializes that drive. It then reads in the first sector of the directory, which is track eighteen, sector one.

These directory sectors contain vital information about the file, such as where the file is on the disk, how large the file is, the filename, and what type of file it is, which is what we're concerned with here.

The first byte of the directory sector is the file type ORed with 128. A 128 is a DELeted file, 129 is a SEQuential file, 130 a PRoGram file, 131 a USeR file, and 132 a RELative file. Using this information, we can restore deleted files, or change the filetype and concatenate two program files.

Deleting Files

We are only concerned with file types that are greater than 128, so line 170 checks to see that a file does indeed exist there. If so, a loop prints out the filename. If you wish to delete the file, press Y; otherwise press N. It is not necessary to press
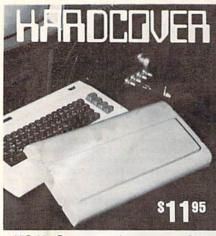
RETURN after hitting one of the above keys. Also note that this program is able to delete illegal filenames, such as commas.

Each directory sector can hold only eight files, so every once in a while the program will let you know that it has gone to a new directory sector by printing out the next track and sector.

When it has gone through all of the files in the directory, the program validates the disk. If this is not done, the files will not be removed from the directory. Instead, they will show up as DELeted files, wasting directory space.

See program listings on page 133. 🔲

Error Trapping

Alejandro A. Kapauan

It's frustrating to encounter an ERROR message when programming and then spend a lot of time scanning your listing in search of the culprit line. With this short subroutine added to your programs, you can pinpoint the type of error and the line number.

Many versions of BASIC include a command which allows errors in a program to be trapped by the program itself. For example, in some BASICs the statement ON ERR GOTO 200 or TRAP 200 tells BASIC that if any error occurs in the program, then branch to line 200.

This is useful for programs in which errors may result from certain values of data input. A spreadsheet program, for instance, may allow formulas to be entered which instruct it to divide the entries in one column by the entries of another. Unless special precautions are taken in the program, a divide by zero, or even an overflow error might take place. The ON ERR or TRAP statement can allow the programmer to trap such errors.

Unfortunately, Commodore BASIC does not include such a statement. However, the accompanying subroutine can be incorporated in your own programs to allow automatic error trapping.

Setting The Trap

Include lines 200–4020 in your program. To set an error trap, set the variable ET to the line number of the error handling routine you want to jump to when an error occurs, then GOSUB 3050. In your error handling routine, you can call the subroutine at line 4000 which sets EN, LN, and ER$ to the error number, line number, and error message string.

The subroutine at line 3050 POKEs a small wedge program into the cassette buffer and installs its address in the error message printing routine vector. If the wedge program is ever executed, it issues the command GOTO 200; in effect, the equivalent of ON ERR GOTO 200.

For safety, the wedge restores the original error message routine vector when it is executed, so that if an error occurs in the error handler, a normal exit from the program is taken. If you want to reactivate the error handler again, you can set ET

Number	Error
1	Too many files
2	File open
3	File not open
4	File not found
5	Device not present
6	Not input file
7	Not output file
8	Missing filename
9	Illegal Device number
10	NEXT without FOR
11	Syntax
12	RETURN without GOSUB
13	Out of data
14	Illegal quantity
15	Overflow
16	Out of memory
17	Undefined statement
18	Bad subscript
19	ReDIM'd array
20	Division by zero
21	Illegal direct
22	Type mismatch
23	String too long
24	File data
25	Formula too complex
26	Can't continue
27	Undefined function
28	VERIFY
29	LOAD
30	Break

to its line number then GOSUB 3060 (line 3050 can be skipped if the wedge is already in place).

At any time, you can change the error handler line number by setting ET to the line number and GOSUB 3060. If an error occurs in immediate mode, the normal error message routine is called.

Program 2 for the Commodore 64 is slightly different in that an error number of 128 is not really an error, but a normal program END.

The lines shown in Program 3 can be added to either Program 1 or 2 to provide a simple demonstration of the error trapping routine.

Triggering The Trap

Lines 10–50 (Program 3) compute the quotient 1/X with X varying from 10 to zero. Of course, everything will be okay until the quotient 1/0 is computed, when a division by zero error occurs. This would normally cause the program to quit and an error message to be printed. However, in line 20, the variable ET is set to 200 and the subroutine at line 3050 is executed.

Therefore, when the quotient 1/0 is computed at line 40, the program immediately jumps to line 200, which is the error handling routine. In order to determine what type of error occurred, you can PEEK location 889 in your error handling routine. The value you get will be a number from 1 to 30

which corresponds to the errors listed in the table.

The line number where the error occurred can be found in locations 890 and 891 in low byte/high byte format. I have supplied a subroutine which starts at line 4000 in Programs 1 and 2 which sets the variable EN to the error number, the variable LN to the line number, and the string variable ER$ to the error message string corresponding to the error number EN.

In this sample program, the error handling routine does nothing but print a message and stop. In your own programs, you might want to recover from the error and resume execution of your program.

See program listings on pages 156.

Using The GET Statement

Alfred J. Bruey

GET has many uses, from controlling user input to interrupting program execution. Here are some examples of when to use, and when to avoid, this versatile command.

Every program manipulates data. And most require the user to input some of the data.

The usual way to do this is with an INPUT statement. But a user can enter anything in response to an INPUT statement: a number, a letter, a special character, even just a cursor movement. He might even press the RETURN key without entering any data at all.

If RETURN is pressed, you might not get the results you want. If, for example, you have programmed:

```
200 INPUT"ENTER WORD ";N$
210 PRINT N$
```

the answer you will get if you press RETURN depends on what value N$ had earlier in the program. If N$ had been set equal to the string "HELLO" earlier with the line

```
100 N$="HELLO"
```

then you would get HELLO after pressing RETURN. If N$ hadn't been assigned earlier, it would be set equal to the null (nothing) character.

An Alternative

In many cases, GET is preferable to INPUT. The operation of the GET statement is easy to understand: When the computer encounters a GET statement, it assigns the first character in the keyboard buffer to the variable named in the GET statement and then goes on.

The keyboard buffer is ten memory locations (631–640) which remember which keys were pressed by the user. Sixty times a second the computer checks the keyboard. If a key is being pressed, the ASCII value of that character is put in the keyboard buffer. Most of the time the letter is printed on the screen and the buffer cleared. But when a program is running, the buffer can fill up. The computer will use the letters when the program ends or when an INPUT or GET occurs. If there is no value in the keyboard buffer, it goes on without waiting for a character to be entered. If you execute the statement

```
100 GET A$
```

and there is nothing in the keyboard buffer, A$ will be set equal to the null string. If there is nothing in the buffer when the line

```
100 GET N
```

is executed, N will be set equal to 0. To see how this works, enter and run these two short programs:

```
10 A$="HELLO"
20 GET A$
30 PRINT A$
```

and

```
10 N=10
20 GET N
30 PRINT N
```

A String Is Safer

If the only input you want to use in a GET statement is an integer from 1 to 9, a numeric variable can be used in the GET statement, but normally string variables are used with the GET statement. If you use a line like GET N and type any key other than the number keys 0–9, the program will stop with a SYNTAX ERROR message.

Since the GET statement doesn't wait for you to enter a character, you have to do something to

UNIQUE HARDWARE
For Your Commodore or Vic

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

You're not alone. Thousands of Commodore 64 owners have "fuzzy" color on their TVs.

Most have interference lines crowding out their great graphics. Many have bought expensive monitors or new TVs, and often even that hasn't helped. But, most of us just lived with the problem. Now the engineers at Bytes & Pieces have four simple, inexpensive solutions.

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

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

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

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

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

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

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

Computer dust covers **$9.95**

Disk drive dust covers **$8.95**

Dataset dust covers **$7.95**

Why Blank "Cheat" Sheets?

Because They're Better Blank

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

A few companies do offer a solution...a die cut "cheat" sheet that attaches to your keyboard with all the commands of one program printed on it. Great idea, unless you need them for 10 or 20 programs. You could purchase another disk drive for the same investment. Our solution? Simple. A pack of 12 lined cards, die cut to fit your keyboard and just waiting to be filled with those problem commands **you** forget most often. Simple? Yes, but effective. Now you can have **all** your program commands right at your finger tips on YOUR VERY OWN, custom designed "cheat" sheets. Order a couple packs today!
12 for **$15.95**

Is Your Commodore Disk Drive Hot and Bothered?

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

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

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

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

Muffin fan **$54.95**

Fan with surge protector **$79.95**

Order Today!

Rush me the following:

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

Qty.	Item	Amount
Muffin Fans		
___	Muffin Fans @ $54.95	$ ___
___	Muffin Fans with Surge Protector @ $79.95	$ ___
	Shipping & Handling	$ 2.00
	Sub Total	$ ___
	5% State Tax (Wisconsin Residents only)	$ ___
	TOTAL	$ ___

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

VISA # _____

MasterCard # _____

Inner Bank # _____

Expiration Date _____

Signature _____

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

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

www.commodore.ca

slow the statement down. This can be done as follows. (Be sure you don't have any spaces between the quote marks in line 100.)

```
100 GET A$: IF A$="" THEN 100
200 PRINT A$
```

Note that when you run this short program, nothing happens until you press a character. Then this character is printed on the screen by the PRINT command. You can see how it works: the IF–THEN instruction puts the GET statement into a one-statement loop. Execution stays in line 100 until you press a key.

You may have noticed that the GET statement does not prompt you with a question mark or message the way an INPUT statement does. It's a good idea to put a PRINT statement in front of the GET statement to tell the user when it's time to enter data.

Numeric Input

One use of the GET statement is to count characters. One example of this might be where you require someone to enter a social security number, always nine digits long, or a zip code, which is always five digits long.

Let's look at an example:

```
100 PRINT"ENTER 5 DIGIT ZIP CODE ":
200 FOR K=1 TO 5
300 GET Z$(K): IF Z$(K)="" THEN 300
900 PRINT Z$(K);
1000 NEXT K
```

The semicolon in line 900 is necessary for the five digits to be printed on the same line.

At first appearance, this program looks just like what we needed. But there are a lot of shortcomings. For example, the program will allow you to enter the five characters QWERT as a valid zip code. Try using the DELETE key to correct a mistake; you'll see that the program counts the delete character as one of the five characters. Note, also, that cursor movements count as characters.

Adding Error Correction

Let's put in the changes that we need to allow the user to correct a typing error with the DELETE key and restrict input so the user can only enter digits from 0 to 9.

First we change line 200 to:

```
200 FOR K=1 TO 6
```

and line 100 to:

```
100 PRINT"ENTER 5 DIGIT ZIP CODE AND PRES
    S RETURN "
```

These two lines together allow the user to correct the fifth digit before going on. This requires the

user to press RETURN to signal that he is done. Next, add:

```
400 IF (Z$(K)>"9" OR Z$(K)<"0") AND Z$(K)
    <>CHR$(20) AND Z$(K)<>CHR$(13) THEN 3
    00
```

This line checks the value of Z$(K). It will be accepted as a valid character if, and only if, it is a digit from 0 to 9, a DELETE character (CHR$(20)), or a RETURN character (CHR$(13)). If it is none of these, control is returned to line 300 to wait for a valid character.

A RETURN Bug

Now run the program again. There is still a problem, because you can enter the RETURN key even if you haven't entered five characters yet. So let's add this line:

```
500 IF K<6 AND Z$(K)=CHR$(13) THEN 300
```

This allows the RETURN key to be accepted only as the sixth character.

Next we need to add a line to make sure someone doesn't try to delete a character when there isn't one there to delete:

```
600 IF K=1 AND Z$(K)=CHR$(20) THEN 300
```

There's still a problem because the DELETE character counts as one of the characters allowed for input. We can stop this with the line:

```
700 IF Z$(K)=CHR$(20) THEN PRINT CHR$(20)
    ;:K=K-1:GOTO 300
```

This statement checks for the delete character. If it finds it, it prints a delete character. Then it reduces the character count by 1 and goes back to wait for another character.

The Final Test

Now run this program. You'll see that you can enter and delete characters at will. It almost works. You might have discovered that this coding will allow you to enter a six-digit number. Earlier we put in a check to make sure that the RETURN character could only be entered as the sixth character; now we need to make sure that, if the sixth character isn't a DELETE, it's a RETURN character:

```
800 IF K=6 AND Z$(K)<>CHR$(13) THEN 300
```

Your final program should look like this:

```
100 PRINT"{RIGHT}ENTER FIVE DIGIT ZIP COD
    E AND":PRINT"{RIGHT}PRESS RETURN"

200 FOR K=1 TO 6
300 GET Z$(K):IF Z$(K)=""THEN 300:
400 IF (Z$(K)>"9" OR Z$(K)<"0") AND Z$(K)
    <>CHR$(20) AND Z$(K)<>CHR$(13)THEN 30
    0
```