You never know who you will meet or what will be said when you're on American People/Link.

AMERICAN PEOPLE/LINK™ users meet a lot of different people. That's because PEOPLE/LINK — the nation's first all entertainment videotex network — allows anyone with a word processor, personal computer or terminal, and a modem to communicate with other users throughout the country from the privacy of their own homes.

Sometimes these electronic conversations are serious...sometimes they're outrageous...but they will always keep your interest as you talk to friends and make new ones. PEOPLE/LINK's recreational programs include:

PARTYLINE — Meet people and talk live with other PEOPLE/LINK members throughout the country in groups or privately.

CLUB-LINK — Join or start a club or group devoted to a favorite hobby, rock group, lifestyle, etc.

WHO-IS-WHO — locate other users with similar interests.

And there's much more with programs like NETMAIL, our person-to-person electronic mail, PEOPLESCAN, the national bulletin board system, and play games such as poker, blackjack, checkers, chess, backgammon, and bridge (color graphics are available for most popular computers).

And the greatest thing is that you don't have to be a computer whiz or even know how to type to use PEOPLE/LINK...just be smart enough to subscribe now!

Talk live nationwide for $2.95/hour . . . an exciting and inexpensive way to meet new friends.*

AMERICAN PEOPLE LINK

Use your Visa, Mastercard, or American Express.
Be On-Line Today! Call Us Toll Free: **1-800-524-0100**

Illinois Residents Call: **1-312-870-5200**

American PEOPLE/LINK
Arlington Ridge Office Center
3215 N. Frontage Road—Suite 1505
Arlington Heights, IL 60004

*Subject to change

SpeedScript 3.0

All Machine Language Word Processor For Expanded VIC-20

Charles Brannon, Program Editor

COMPUTE! continues its SpeedScript 3.0 series this month with our enhanced version for the Commodore VIC-20 (with at least 8K memory expansion). Written entirely in machine language, SpeedScript contains nearly every command and convenience you'd expect from a quality word processor. First introduced in the January 1984 issue of our companion magazine, COMPUTE!'s GAZETTE, SpeedScript incorporates a year's worth of improvements, readers' suggestions, and additional debugging. Because the VIC version is so similar to the Commodore 64 version, refer to last month's article for a full tutorial-style explanation. This month's article is an abbreviated description. Look for the Atari and Apple versions of SpeedScript 3.0 in coming issues.

SpeedScript 3.0, though compact in size (6K), has many features found on commercial word processors. *SpeedScript* is also very easy to learn and use. You type in everything first; preview and make corrections on the screen; insert and delete words, sentences, and paragraphs; then print out an error-free draft, letting *SpeedScript* take care of things like margins, centering, headers, and footers.

Entering *SpeedScript*

SpeedScript is one of the longest machine language programs we've ever published, but the MLX entry system helps you type it right the first time. MLX can detect most errors people make when entering numbers. (See the MLX article elsewhere in this issue.) MLX also lets you type *SpeedScript* in more than one sitting. (Unfortunately, if you have an earlier version of *SpeedScript*, you cannot just make certain changes to bring it up to version 3.0. You have to type it from scratch.) Although the program listing is lengthy, we guarantee the effort will be worthwhile.

Before you begin typing *SpeedScript* (or begin a subsequent session of typing if you enter *SpeedScript* in more than one sitting), you must enter the following POKEs *before* you load and run the MLX program. These POKEs are essential to protect *SpeedScript* from BASIC while you are typing it in. Again, these POKEs should be performed *before* you load MLX, but are not necessary to run the finished *SpeedScript* program:

POKE 44,42:POKE 10752,0:NEW

Now load and run the VIC version of MLX (remember that you need at least 8K memory expansion to run VIC MLX). Answer the first two questions like this:

Starting Address? 4609
Ending Address? 10482

The screen will then show the first prompt, the number 4609 followed by a colon. Type in each three-digit number shown in the listing. You do not need to type the comma shown in the listing. MLX inserts the comma automatically.

The last number you enter in each line is a *checksum*. It represents the values of the other numbers in the line summed together. If you make a mistake while entering the line, the checksum calculated by MLX and displayed on the screen should not match that of the listing, and you will have to retype the line. MLX is not foolproof, though. It's possible to fool the checksum by exchanging the position of the three-digit numbers. Also, an error in one number can be offset by an error in another (just as $3 + 4 + 7 = 1 + 4 + 9$). Keep this in mind. MLX will help catch your errors, but you still must be careful.

Typing In Multiple Sittings

If you want to stop typing the listing at some point and pick up later, press SHIFT-S and follow the screen prompts. Remember to note the line number of the last line you typed in. When you are ready to continue typing, enter the POKEs mentioned above, load MLX, answer the starting and ending address prompts, then press SHIFT-L. MLX asks for the filename you gave to the partially typed program. After the LOAD is complete, press SHIFT-N and tell MLX the line number you stopped at. Now continue typing as before. When you finish all typing, MLX automatically prompts you to save the program.

At this point MLX has saved a program file on tape or disk. If you load it and list it, you'll see that it looks like a normal one-line BASIC program, with a line number and a SYS command. The machine language program that is *SpeedScript* starts in memory just after the SYS command. The simulated BASIC line is included so you can load *SpeedScript* like any BASIC program and enter RUN to start it. You don't need to add the ",1" like you do when loading many machine language programs. Just LOAD "SPEEDSCRIPT" (or whatever filename you called it) for tape, or LOAD "SPEEDSCRIPT",8 for disk, then enter RUN. Once *SpeedScript* is in memory, you can save it from BASIC like a BASIC program. If *SpeedScript* is running, press RUN/STOP–RESTORE to exit to BASIC.

Before using *SpeedScript*, you should generally unplug all cartridges such as the *Super Expander*. You must have a memory expansion cartridge plugged in that provides at least an additional 8K, although *SpeedScript* can take advantage of up to 24K of memory. *SpeedScript* cannot take advantage of any custom hardware configurations except those that do not interfere with normal operations.

Entering Text

When you run *SpeedScript*, the screen colors change to black on white. The first two lines on the screen are black with white letters. *SpeedScript* presents all messages on these *command lines*. The remaining 21 lines of the screen are used to enter, edit, and display your document. The cursor shows where the next character you type will appear on the screen. *SpeedScript* lets you move the cursor anywhere within your document, making it easy to find and correct errors.

To begin using *SpeedScript*, just start typing. When the cursor reaches the right edge of the screen, it automatically jumps to the beginning of the next line, just as in BASIC. But unlike BASIC, *SpeedScript* never splits words at the right edge of the screen. If a word you're typing won't fit at the end of one line, it's instantly moved to the next line. This feature, called *word wrap* or sometimes *parsing*, makes it much easier to read your text on the screen.

Scrolling And Screen Formatting

When you finish typing on the last screen line, *SpeedScript* automatically scrolls the text upward to make room for a new line at the bottom. Imagine the screen as a 21-line window on a long continuous document. In total, there's room for 3072 characters of text with an 8K expander; up to 19,456 with a 24K expander. To check at any time how much space is left, press **CTRL-=** (hold down the CTRL key while pressing the = key). The number which appears in the command line indicates how much room remains for characters of text.

If you're used to a typewriter, you'll have to unlearn some habits. Since the screen is only 22 columns wide, and most printers have 80-column carriages, it doesn't make sense to press RETURN at the end of each line as you do on a typewriter. *SpeedScript*'s word wrap takes care of this automatically. Press RETURN only when you want to force a carriage return to end a paragraph or limit the length of a line. A *return-mark* appears on the screen as a left-pointing arrow.

Using The Keyboard

Most features are accessed with control-key commands—you hold down CTRL while pressing another key. In this article, control-key commands are abbreviated **CTRL-*x*** (where *x* is the key you press in combination with CTRL). An example is the CTRL-= mentioned above to check on free memory. CTRL-E means hold down CTRL and press E. Sometimes you have to hold down both SHIFT and CTRL as you type the command key, as in SHIFT-CTRL-H. Other keys are referenced by name or function, such as *back-arrow* for the left-pointing arrow in the top-left corner of the keyboard, *pound sign* for the British pound sign (£), CLR/HOME for the home cursor key, SHIFT–CLR/HOME for the clear screen key, f1 for special function key 1, and *up-arrow* for the upward-pointing arrow to the left of the RESTORE key. See Figure 1 for a complete quick-reference chart of all keyboard commands.

Some keys let you move the cursor to different places in the document to make corrections or scroll text into view. You can move the cursor by character, word, sentence, or paragraph. Here's how to control the cursor:

- The **left/right cursor key** works as usual; pressing this key by itself moves the cursor right (forward) one space, and pressing it with SHIFT moves the cursor left (backward) one space.
- The **up/down cursor key** moves the cursor forward to the beginning of the next sentence. Pressing it with SHIFT moves the cursor backward to the beginning of the previous sentence.
- The **f1 special function key** moves the cursor forward to the beginning of the next word. The **f2 key** (hold down SHIFT and press f1) moves the cursor backward to the beginning of the previous word.
- The **f3 special function key** moves the cursor forward to the beginning of the next sentence (just like the up/down cursor key). The **f4 key** (hold down SHIFT and press f3) moves the cursor backward to the beginning of the previous sentence (just like pressing SHIFT and the up/down cursor key).
- The **f5 special function key** moves the cursor forward to the beginning of the next paragraph. The **f6 key** (hold down SHIFT and press f5) moves the cursor backward to the beginning of the previous paragraph.
- The **CLR/HOME key,** pressed once by itself, moves the cursor to the top of the screen without scrolling. Pressed twice, it moves the cursor to the beginning of the document.
- **CTRL-Z** moves the cursor to the bottom of the document.

Correcting Your Typing

Sometimes you'll have to insert some characters to make a correction. Use **SHIFT–INST/DEL** to open up a single space, just as in BASIC. Merely position the cursor at the point where you want to insert a space, and press SHIFT–INST/DEL.

It can be tedious to use the SHIFT–INST/DEL key to open up enough space for a whole sentence or paragraph. For convenience, *SpeedScript* has an insert mode that automatically inserts space for each character you type. In this mode, you can't type over characters; everything is inserted at the cursor position. To enter insert mode, press **CTRL-I.** To cancel insert mode, press CTRL-I again. To let you know you're in insert mode, the normally black command lines at the top of the screen turn blue.

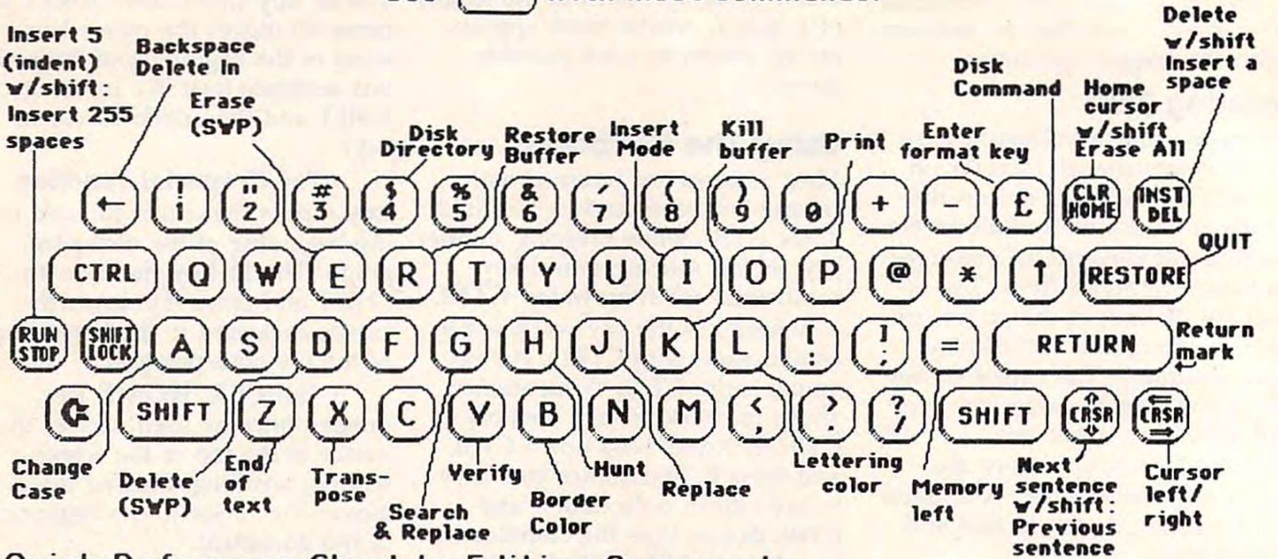
Insert mode is the easiest way to insert text, but it can become too slow when working with a very long document because it must move *all* the text following the cursor position. So *SpeedScript* has even more ways to insert blocks of text.

One way is to use the **RUN/STOP** key. It is programmed in *SpeedScript* to act as a five-space margin indent. To end a paragraph and start another, press

Figure 1:

VIC SpeedScript 3.0 Keyboard Map

Use CTRL with most commands.



Quick Reference Chart to Editing Commands

* Notes commands changed or added since Version 2.0

- CTRL A — Change case
- CTRL B — Change Border Color
- CTRL D — Delete (S,W,P)
- CTRL E — Erase (S,W,P)
- *CTRL G — Auto. Search & Replace
- *CTRL H — Hunt. w/SHIFT: Select Hunt Phrase
- CTRL I — Enter/Exit Insert Mode
- *CTRL J — Replace. w/SHIFT:Select Replace Phrase
- CTRL K — Kill Buffer
- CTRL L — Change Lettering Color
- CTRL P — Print
- CTRL R — Restore Buffer
- CTRL V — Verify
- CTRL X — Transpose Characters
- CTRL Z — Go to End of Text
- CTRL = — Display free memory
- CTRL ↑ — Send disk command/read error
- CTRL 4 — Display Disk Directory
- CTRL £ — Enter Format (printer) command
- CLR HOME — Press once to go to top of screen; hold down to go to top of text. w/SHIFT: Erase ALL
- CRSR ⇐⇒ — Cursor left/right

- * RUN STOP — Indent 5 spaces w/SHIFT:Insert 255 spaces
- RESTORE — Exit SpeedScript
- * ← — Backspace w/CTRL:Delete In w/SHIFT & CTRL: Delete spaces
- Return mark
- * RETURN — w/SHIFT: End Paragraph
- INST DEL — Delete w/SHIFT:Insert space
- CRSR ⇑⇓ — Go to next sentence w/SHIFT: Goto previous sentence.

- f1 — Word Right
- f2 — Word Left
- f3 — Next Sentence
- f4 — Previous Sentence
- f5 — Next Paragraph
- f6 — Previous Paragraph
- f7 — Load
- f8 — Save

RETURN twice and press RUN/STOP. You can use RUN/STOP to open up more space than SHIFT–INST/DEL. No matter how much space you want to insert, each insertion takes the same amount of time. So the RUN/STOP key can insert five spaces five times faster than pressing SHIFT–INST/DEL five times.

There's an even better way, though. Press **SHIFT–RUN/STOP** to insert 255 spaces. You can press it several times to open up as much space as you need. And SHIFT–RUN/STOP is *fast*. (You don't want to be in insert mode when you use this trick; that would defeat its purpose.)

Since the INST/DEL key also is slow when working with large documents (it, too, must move all text following the cursor), you may prefer to use the back-arrow key to backspace. The back-arrow key by itself moves the cursor left one space and blanks out that position. It's more like a backspace than a delete.

After you're done inserting with these methods, there will probably be some inserted spaces left over that you didn't use. Just press **SHIFT-CTRL-back arrow.** This instantly deletes all extra spaces between the cursor and the start of following text.

Erasing Text

Press the **INST/DEL key** by itself to erase the character to the left of the cursor. All the following text is pulled back to fill the vacant space.

Press **CTRL-back arrow** to delete the character on which the cursor is sitting. Again, all the following text is moved toward the cursor to fill the empty space.

These keys are fine for minor deletions, but it could take all day to delete a whole paragraph this way. So *SpeedScript* has two commands that can delete an entire word, sentence, or paragraph at a time. **CTRL-E** erases text *after* (to the right of) the cursor position, and **CTRL-D** deletes text *behind* (to the left of) the cursor.

To use the **CTRL-E erase mode,** first place the cursor at the beginning of the word, sentence, or paragraph you want to erase. Then press CTRL-E. The command line shows the message "Erase (S,W,P): RETURN to exit." Press S to erase a sentence, W for a word, or P for a paragraph. Each time you press one of these letters, the text is quickly erased. You can keep pressing S, W, or P until you've erased all the text you wish. Then press RETURN to exit the erase mode.

The **CTRL-D delete mode** works similarly, but deletes only one word, sentence, or paragraph at a time. First place the cursor after the word, sentence, or paragraph you want to delete. Then press CTRL-D. Next, press S, W, or P for sentence, word, or paragraph. The text is immediately deleted and you return to editing. You don't need to press RETURN to exit the CTRL-D delete mode unless you pressed this key by mistake. *(In general, you can escape from any command in* SpeedScript *by simply pressing RETURN.)* CTRL-D is most convenient when the cursor is already past what you've been typing.

Figure 2: Quick Reference Chart
Format (Printer) Commands
Enter with CTRL-£

Command	Description	Default	Command	Description	Default
a	True ASCII	off	n	Next Page	
b	Bottom Margin	58	p	Page Length*	66
c	Centering		r	Right Margin	75
e	Edge Right		s	Spacing	1
f	Footer		t	Top Margin	5
g	Goto Linked File*		u	Underline toggle	
h	Header		w	Page Wait	
i	Information*		x	Columns across*	80
j	Select linefeeds*		@	Initial page #*	1
l	Left Margin	5	?	Skip pages*	
m	Margin Release*		#	Print page number	

h c SpeedScript/#←	Centered Header with page number
l 10 r 70 s 2←	Left margin 10, right margin 70, double spacing.
g D:SpeedScript.2←	Goto and continue printing with filename "SpeedScript.2"

* Notes command changed or added since Version 2.0

The Text Buffer

When you erase or delete with CTRL-E and CTRL-D, the text isn't lost forever. *SpeedScript* remembers what you've removed by storing deletions in a separate area of memory called a *buffer*. The buffer is a fail-safe device. If you erase too much, or change your mind, just press **CTRL-R** to restore the deletion. However, be aware that *SpeedScript* remembers only the last erase or delete you performed.

Another, more powerful, use of this buffer is to move or copy sections of text. To move some text from one location in your document to another, first erase or delete it with CTRL-E or CTRL-D. Then move the cursor to where you want the text to appear and press CTRL-R. CTRL-R instantly inserts the contents of the buffer at the cursor position. If you want to copy some text from one part of your document to another, just erase or delete it with CTRL-E or CTRL-D, restore it at the original position with CTRL-R, then move the cursor elsewhere and press CTRL-R to restore it again. You can retrieve the buffer with CTRL-R as many times as you like.

Important: The CTRL-E erase mode lets you erase up to the maximum size of the buffer (1K, or 1024 characters), and CTRL-E also removes the previous contents of the buffer. Keep this in mind if there's something in the buffer you'd rather keep. If you don't want the buffer to be erased, press **SHIFT-CTRL-E**. This preserves the buffer contents and adds newly erased text to the buffer.

If you ever need to erase the contents of the buffer, press **CTRL-K** (*kill buffer*).

The Wastebasket Command

If you want to start a new document, or simply obliterate all your text, press **SHIFT-CLR/HOME**. *SpeedScript* asks, "ERASE ALL: Sure? Y/N." This is your last chance. If you *don't* want to erase the entire document, press N or any other key. Press Y to perform the irreversible deed. There is no way to recover text wiped out with Erase All.

If you press RUN/STOP–RESTORE, you'll find yourself back to BASIC's READY prompt. Once in BASIC you still have one chance to reenter *SpeedScript* without losing your text—simply enter RUN (but your chances decrease if you execute other commands in BASIC).

Search And Replace

SpeedScript has a Hunt command that searches through your document to find a selected word or phrase. A Replace option lets you automatically change one word to another throughout the document.

SHIFT-CTRL-H activates the Hunt feature, **SHIFT-CTRL-J** (J is used because it's next to the H) lets you selectively hunt and replace, and **CTRL-G** (also next to the H) is for automatically searching and replacing.

Searching is a two-step process. First you need to tell *SpeedScript* what to search for, then you trigger the actual search. Press SHIFT-CTRL-H. The command lines say "Hunt for:". Type in what you'd like to search for, the *search phrase*. If you press RETURN alone without typing anything, the Hunt command is canceled.

When you are ready to search, press **CTRL-H.** *SpeedScript* looks for the next occurrence of the search phrase *starting from the current cursor position*. If you want to hunt through the entire document, press CLR/HOME twice to move the cursor to the very top before beginning the search. Each time you press CTRL-H, *SpeedScript* looks for the next occurrence of the search phrase and places the cursor at the start of the phrase. If the search fails, you'll see the message "Not Found."

CTRL-J (Replace) works together with CTRL-H. After you've specified the search phrase with SHIFT-CTRL-H, press **SHIFT-CTRL-J** to select the replace phrase. (You can press RETURN alone at the "Replace with:" prompt to select a *null* replace phrase. When you hunt and replace, this deletes the located phrase.) To manually search and replace, start by pressing CTRL-H. After *SpeedScript* finds the search phrase, press CTRL-J if you want to replace the phrase. If you don't want to replace the phrase, don't press CTRL-J. You are not in a special search and replace mode. You're free to continue writing at any time.

CTRL-G links CTRL-H and CTRL-J together. It first asks "Hunt for:", then "Replace with:", then automatically searches and replaces throughout the document starting at the cursor position.

Storing Your Document

Just press f8 (SHIFT-f7) to store a document. You'll see the prompt "Save:". Type in a filename up to 16 characters long, but do not use question marks or asterisks. You cannot use the same name for two different documents on a single disk. To replace a document already on disk using the same filename, precede your filename with the characters **@0:** or **@:**. You can also precede the filename with either **0:** or **1:** if you use a dual disk drive. *SpeedScript* cannot access a second disk drive with a device number of 9.

After entering the filename, answer the prompt "Tape or Disk" by pressing either the **T** or **D** key. You can cancel the SAVE command by pressing RETURN without typing anything else at either the "Save:" or "Tape or Disk?" prompt.

When the SAVE is complete, *SpeedScript* reports "No errors" if all is well, or reads and reports the disk error message if not. It is not possible to detect errors during a tape SAVE, so if you want peace of mind, use the Verify command. Rewind the tape, press **CTRL-V**, then type the filename. Press T for tape, then press PLAY on the recorder. *SpeedScript* compares the file on tape with that in memory and reports "No errors" if the verify succeeds, or "Verify Error" if not. You can also verify disk files.

Loading A Document

To recall a previously saved document, press **f7**. Answer the "Load:" prompt with the filename. Insert the tape or disk, rewind the tape, then answer T or D. Press PLAY on tape. *SpeedScript* loads the file and should display "No errors." Otherwise, *SpeedScript* reads the error channel of the disk drive or simply reports "Load error" for tape.

The position of the cursor is important before loading a file. *SpeedScript* starts loading at the cursor position, so be sure to press CLR/HOME twice or SHIFT–CLR/HOME (Erase All) to move the cursor to the start of text space, unless you want to merge two documents. When you press f7 to load, the command lines turn green to warn you if the cursor is not at the top of the text space.

To merge two or more files, simply load the first file, press CTRL-Z to move the cursor to the end of the document, and then load the file you want to merge. Do not place the cursor somewhere in the middle of your document before loading. A LOAD does not insert the text from tape or disk, but overwrites all text after the cursor position. The last character loaded becomes the new end-of-text pointer, and you cannot access any text that appears ahead of this pointer.

Disk Commands

Sometimes you forget the name of a file, or need to scratch or rename a file. *SpeedScript* gives you full control over the disk drive. Just press **CTRL-up arrow**, then type in a 1541 disk command. You don't need to type PRINT#15 as you do in BASIC, just the actual command. If you press RETURN without typing a disk command, *SpeedScript* displays the disk status. It also displays the status after completing a disk command.

Additional Features

SpeedScript has a few commands that don't do much, but are nice to have. **CTRL-X** exchanges the character under the cursor with the character to the right of the cursor. Thus you can fix transposition errors with a single keystroke. **CTRL-A** changes the character under the cursor from uppercase to lowercase or vice versa.

Press **CTRL-B** to change the background and border colors. Each time you press CTRL-B, one of 16 different background colors appears. Press **CTRL-L** to cycle between the eight character (lettering) colors. The colors are preserved until you change them. If you resave *SpeedScript* from BASIC as described above, the program will load and run with your color choice in the future.

PRINT!

To begin printing, simply press **CTRL-P.** If your printer is attached, powered on, and selected (online), *SpeedScript* begins printing immediately. To cancel printing, hold down the **RUN/STOP** key until printing stops, then release it when the border color changes to white. *SpeedScript* assumes a left margin of five, a right margin of 75, single-spacing, and continuous-feed paper. You can change these default settings if you want (see below).

Before printing, be sure the paper in your printer is adjusted to top-of-form (move the paper perforation just above the printing element). CTRL-P assumes a Commodore printer, so it's helpful if your interface simulates the modes and codes of the Commodore 1525, MPS-801, or 1526 printer. CTRL-P prints with a device number of 4 and a secondary address of 7 (uppercase/lowercase mode).

If CTRL-P doesn't work for you, try another variation, **SHIFT-CTRL-P.** Answer the prompt "Print to: Screen, Disk, Printer?" with the single letter **S**, **D**, or **P**. Press any other key to cancel the command.

If you press P for printer, *SpeedScript* requests two more keystrokes. First answer "Device number" with a number from 4 to 7. This lets you print to one of several printers addressed with different device numbers. Next answer "Secondary Address?" with a number from 0 to 9.

Printing To Screen And Disk

SHIFT-CTRL-P prints to the screen when you press S. The screen colors change to white letters on a black background, and what appears on the screen is exactly what would print on the printer. It takes about four screen lines to hold one 80-column printed line, of course. If you use double-spacing (see below), it's much easier to see how each line is printed. With this screen preview, you can see where lines and pages break. To freeze printing, hold down either SHIFT key or engage SHIFT LOCK. When printing is finished, press any key to return to editing.

SHIFT-CTRL-P prints to a disk file when you press D. Enter the filename when requested. *SpeedScript* sends out all printer information to a sequential file. You can use other programs to process this formatted file (see last month's *SpeedScript* article for details).

Formatting Commands

The print-formatting commands must be distinguished from normal text, so they appear onscreen in reverse field with the text and background colors switched. You enter these reverse-field letters by pressing **CTRL-£** (pound sign). Answer the prompt "Enter format key:" by pressing a single key. This key is inserted into text in reverse-field. All lettered printer commands should be entered in lowercase (unSHIFTed). During printing, *SpeedScript* treats these characters as printing commands.

There are two kinds of printing commands, which we'll call Stage 1 and Stage 2. Stage 1 commands usually control variables such as left margin and right margin. Most are followed by a number, with no space between the command and the number. Stage 1 commands are executed before a line is printed.

Stage 2 commands, like centering and underlining, are executed while the line is being printed. Usually Stage 1 commands must be on a line of their own, although you can group several Stage 1 commands together on a line. Stage 2 commands are by nature embedded within a line of text.

Stage 1 Commands

l Left margin. Follow with a number from 0 to 255. Use 0 for no margin. Defaults to 5.

r Right margin position, a number from 1 to 255. Defaults to 75. Be sure the right margin value is greater than the left margin value, or *SpeedScript* will go bonkers.

t Top margin. The position at which the first line of text is printed, relative to the top of the page. Defaults to 5. The header (if any) is always printed on the first line of the page, before the first line of text.

b Bottom margin. The line at which printing stops before continuing to the next page. Standard 8½ × 11-inch paper has 66 lines. Bottom margin defaults to the fifty-eighth line. The footer (if any) is always printed on the last line of the page, after the last line of text.

p Page length. Defaults to 66. If your printer does not print six lines per inch, multiply lines-per-inch by 11 to get the page length. European paper is usually longer than American paper—11⅝ or 12 inches. Try a page length of 69 or 72.

s Spacing. Defaults to single-spacing. Follow with a number from 1 to 255. Use 1 for single-spacing, 2 for double-spacing, 3 for triple-spacing.

@ Start numbering *at* page number given. Page numbering normally starts with 1.

? Disables printing until selected page number is reached. For example, a value of 3 would start printing the third page of your document. Normally, *SpeedScript* prints starting with the first page.

x Sets the page width, in columns (think *a cross*). Defaults to 80. You need to change this for the sake of the centering command if you are printing in double-width or condensed type, or are using a 40-column or wide-carriage printer.

n Forced paging. Normally, *SpeedScript* prints the footer and moves on to the next page only when it has finished a page, but you can force it to continue to the next page by issuing this command. It requires no numbers.

m Margin release. Disables the left margin for the next printed line. Remember that this executes before the line is printed. It's used for outdenting.

a True ASCII. Every character is assigned a number in the ASCII

(American Standard Code for Information Interchange) character set. Most printers use this true ASCII standard, but Commodore printers exchange the values for uppercase and lowercase to match Commodore's own variation of ASCII. Some printer interfaces do not translate Commodore ASCII into true ASCII, so you need to use this command to tell *SpeedScript* to translate. Also, you will sometimes want to intentionally disable your interface's emulation mode in order to control special printer features that would otherwise be rejected by emulation. Place this command as the first character in your document, even before the header and footer definitions. Don't follow it with a number.

w Page wait. Like the true ASCII command, this one should be placed at the beginning of your document before any text. With page wait turned on, *SpeedScript* prompts you to "Insert next sheet, press RETURN" when each page is finished printing. Insert the next sheet, line it up with the printhead, then press RETURN to continue. Page wait is ignored during disk or screen output.

j Select automatic linefeeds after carriage return. Like **a** and **w**, this command must be placed before any text. Don't use this command to achieve double-spacing, but only if all text prints on the same line.

i Information. This works like REM in BASIC. You follow the command with a line of text, up to 255 characters, ending in a return-mark. This line will be ignored during printing, and is handy for making notes to yourself such as the filename of the document.

h Header define and enable. The header must be a single line of text (up to 254 characters) ending in a return-mark. The header prints on the first line of each page. You can include Stage 2 commands such as centering and page numbering in a header. You can use a header by itself without a footer. The header and footer should be defined at the top of your document, before any text. If you want to prevent the header from printing on the first page, put a return-mark by itself at the top of your document before the header definition.

f Footer define and enable. The footer must be a single line of text (up to 254 characters) ending in a return-mark. The footer prints two lines prior to the last line of each page. As with the header, you can include Stage 2 printing commands, and you don't need to set the header to use a footer.

g GOTO (link) next file. Put this command as the last line in your document. Follow the command with the letter D for disk or T for tape, then a colon (:), then the name of the file to print next. After the text in memory is printed, the link command loads the next file into memory. You can continue linking in successive files, but don't include a link in the last file. Before you start printing a linked file, make sure the first of the linked files is in memory. When printing is finished, the last file linked to will be in memory.

Stage 2 Commands

These commands either precede a line of text, or are embedded within one.

c Centering. Put this at the beginning of a line you want to center. This will center only one line ending in a return-mark. Repeat this command at the beginning of every line you want centered. Centering uses the page-width setting (see above) to properly center the line. To center a double-width line, either set the page width to 40 or pad out the rest of the line with an equal number of spaces. If you use double width, remember that the spaces preceding the centered text will be double-wide spaces.

When *SpeedScript* encounters this command, it prints the current page number. You usually embed this within a header or footer.

u A simple form of underlining. It does not work on Commodore printers, but only on printers that recognize CHR$(8) as a backspace and CHR$(95) as an underline character. Underlining works on spaces, too. Use the first **u** to start underlining, and another one to turn off underlining.

Fonts And Styles

Most dot-matrix printers are capable of more than just printing text at ten characters per inch. The Commodore MPS-801 can print in double width and reverse field. Some printers have several character sets, with italics and foreign language characters. Most can print in double width (40 characters per line), condensed (132 characters per line), and in either pica or elite. Other features include programmable characters, programmable tab stops, and graphics modes. Many word processors customize themselves to a particular printer, but *SpeedScript* was purposely designed not to be printer-specific. Instead, *SpeedScript* lets you define your own Stage 2 printing commands.

You define a programmable *printkey* by choosing any character that is not already used for other printer commands. The entire uppercase alphabet is available for printkeys, and you can choose letters that are related to their function (like D for double width). You enter these commands like printer commands, by first pressing **CTRL-£**.

To define a printkey, just press CTRL-£, then the key you want to assign as the printkey, then an equal sign (=), and finally the ASCII value to be substituted for the printkey during printing.

Here's how you could program reverse-video printed text. Reverse-on, a value of 18, prints all text in reverse video until canceled by reverse-off (a value of 146) or a carriage return. So define SHIFT-R as 18 and SHIFT-O as 146 (CTRL-£ SHIFT-R=18). Anywhere you want to print a word in reverse, bracket the word with printkey R and printkey O.

You can similarly define whatever codes your printer uses for features like double width or emphasized mode. For your convenience, four of the printkeys are predefined, though you can change them. The keys 1–4 are defined as 27, 14, 15, and 18, common values for most printers.

We hope *SpeedScript* is as valuable to you as it has been for thousands of existing users. Again, for more information, see the article accompanying the Commodore 64 version in the March 1985 issue of COMPUTE!. And keep sending in your suggestions and criticisms—someday they may help make *SpeedScript 4.0* a reality.

SpeedScript 3.0 For VIC-20

Please refer to the "MLX" article before entering this listing.

```
4609 :011,018,010,000,158,052,250
4615 :054,050,049,000,000,000,160
4621 :032,131,019,169,203,205,004
4627 :109,044,141,109,044,240,194
4633 :003,032,050,019,032,195,100
4639 :019,076,038,020,165,038,131
4645 :141,067,018,165,039,141,096
4651 :068,018,165,158,141,070,151
4657 :018,165,159,141,071,018,109
4663 :166,181,240,032,169,000,075
4669 :141,000,041,160,000,185,076
4675 :000,000,153,000,000,200,164
4681 :204,000,041,208,244,238,240
4687 :068,018,238,071,018,224,204
4693 :000,240,007,202,208,224,198
4699 :165,180,208,222,096,165,103
4705 :181,170,005,180,208,001,074
4711 :096,024,138,101,039,141,130
4717 :139,018,165,038,141,138,236
4723 :018,024,138,101,159,141,184
4729 :142,018,165,158,141,141,118
4735 :018,232,164,180,208,004,165
4741 :240,013,160,255,185,000,218
4747 :000,153,000,000,136,192,108
4753 :255,208,245,206,139,018,192
4759 :206,142,018,202,208,234,137
4765 :096,169,044,133,195,133,159
4771 :020,169,016,133,196,169,098
4777 :148,133,021,173,252,040,168
4783 :133,251,173,253,040,133,134
4789 :252,173,255,040,032,014,179
4795 :020,162,002,160,000,173,192
4801 :020,023,145,020,177,251,061
4807 :153,008,041,200,041,127,001
4813 :201,031,240,019,192,022,142
4819 :208,235,136,177,251,041,235
4825 :127,201,032,240,005,136,190
4831 :208,245,160,021,200,132,165
4837 :059,136,185,008,041,145,035
4843 :195,136,016,248,164,059,029
4849 :024,152,101,251,133,251,129
4855 :165,252,105,000,133,252,130
4861 :224,002,208,003,140,251,057
4867 :040,192,022,240,008,169,162
4873 :032,145,195,200,076,004,149
4879 :019,024,165,195,105,022,033
4885 :133,195,133,020,144,004,138
4891 :230,196,230,021,232,224,136
4897 :023,240,003,076,190,018,071
4903 :165,251,141,006,041,165,040
4909 :252,141,007,041,096,173,243
4915 :243,040,133,251,141,252,087
4921 :040,141,002,041,133,057,215
4927 :173,244,040,133,252,141,022
4933 :253,040,141,003,041,133,168
4939 :058,056,173,246,040,237,117
4945 :244,040,170,169,032,160,128
4951 :255,198,252,145,251,200,108
4957 :230,252,145,251,200,208,099
4963 :251,230,252,202,208,246,208
4969 :145,251,096,133,059,132,153
4975 :060,160,000,177,059,240,039
4981 :006,032,210,255,200,208,004
4987 :246,096,032,228,255,240,196
4993 :251,096,169,000,141,255,017
4999 :040,141,243,040,141,245,217
5005 :040,141,247,040,141,249,231
5011 :040,141,155,041,141,196,093
5017 :041,169,045,024,105,001,026
5023 :141,244,040,056,165,056,093
5029 :233,001,141,250,040,056,118
5035 :233,004,141,248,040,056,125
5041 :233,001,141,246,040,169,239
5047 :255,141,153,041,032,202,239
5053 :023,169,147,076,210,255,045
5059 :169,128,141,138,002,133,138
5065 :157,173,005,023,032,241,064
5071 :022,173,243,040,133,057,107
5077 :173,244,040,133,058,032,125
5083 :234,019,169,072,160,039,144
5089 :032,108,019,238,254,040,148
5095 :076,134,021,032,250,019,251
5101 :169,054,160,039,032,108,031
5107 :019,169,000,141,254,040,098
5113 :096,162,043,169,160,157,012
5119 :000,016,202,016,250,169,140
5125 :019,032,210,255,169,018,196
5131 :076,210,255,141,134,002,061
5137 :162,043,157,000,148,202,217
5143 :016,250,096,072,041,128,114
5149 :074,133,059,104,041,063,247
5155 :005,059,096,160,000,177,020
5161 :057,133,002,160,000,177,058
5167 :057,073,128,145,057,032,027
5173 :158,018,173,141,002,041,074
5179 :004,240,009,165,197,201,107
5185 :064,240,003,076,216,020,172
5191 :032,228,255,208,013,165,204
5197 :162,041,016,240,229,169,166
5203 :000,133,162,076,044,020,006
5209 :170,160,000,165,002,145,219
5215 :057,224,095,208,012,032,211
5221 :069,022,169,032,160,000,041
5227 :145,057,076,038,020,173,104
5233 :254,040,240,007,138,072,096
5239 :032,234,019,104,170,138,048
5245 :201,013,208,002,162,095,038
5251 :138,041,127,201,032,144,046
5257 :100,224,160,208,002,162,225
5263 :032,138,072,160,000,177,210
5269 :057,201,031,240,005,173,088
5275 :255,040,240,003,032,015,228
5281 :026,104,032,026,020,160,017
5287 :000,145,057,032,158,018,065
5293 :056,165,057,237,002,041,219
5299 :133,059,165,058,237,003,066
5305 :041,005,059,144,014,165,101
5311 :057,105,000,141,002,041,025
5317 :165,058,105,000,141,003,157
5323 :041,230,057,208,002,230,203
5329 :058,032,134,021,076,038,056
5335 :020,160,000,165,002,145,195
5341 :057,024,165,197,105,064,065
5347 :170,132,162,165,162,201,195
5353 :010,208,250,132,198,138,145
5359 :174,016,021,221,016,021,196
5365 :240,006,202,208,248,076,201
5371 :038,020,202,138,010,170,061
5377 :169,020,072,169,037,072,028
5383 :189,057,021,072,189,056,079
5389 :021,072,096,039,029,157,171
5395 :137,133,099,085,138,134,233
5401 :020,148,082,019,076,147,005
5407 :135,139,113,136,140,091,017
5413 :145,017,121,074,090,097,069
5419 :077,070,118,072,081,108,057
5425 :107,110,003,131,084,141,113
5431 :083,059,022,068,022,079,132
5437 :022,133,022,229,022,005,238
5443 :023,020,023,122,023,175,197
5449 :024,014,026,227,024,039,171
5455 :025,116,026,146,026,181,087
5461 :026,214,026,049,027,063,234
5467 :029,048,028,148,029,020,137
5473 :023,122,023,191,029,203,176
5479 :030,095,031,201,022,235,205
5485 :031,029,029,131,036,202,055
5491 :024,111,031,201,023,028,021
5497 :037,027,039,203,025,195,135
5503 :025,191,037,243,025,251,131
5509 :036,032,228,021,056,165,159
5515 :057,237,252,040,165,058,180
5521 :237,253,040,176,032,056,171
5527 :173,252,040,237,243,040,112
5533 :133,059,173,253,040,237,028
5539 :244,040,005,059,240,013,252
5545 :165,057,141,252,040,165,221
5551 :058,141,253,040,032,158,089
5557 :018,056,173,006,041,229,192
5563 :057,133,251,173,007,041,081
5569 :229,058,133,252,005,251,097
5575 :240,002,176,024,024,173,070
5581 :252,040,109,251,040,141,014
5587 :252,040,173,253,040,105,050
5593 :000,141,253,040,032,158,073
5599 :018,076,182,021,096,056,160
5605 :173,002,041,237,245,040,199
5611 :133,059,173,003,041,237,113
5617 :246,040,005,059,144,012,235
5623 :173,245,040,141,002,041,121
5629 :173,246,040,141,003,041,129
5635 :056,165,057,237,243,040,033
5641 :133,059,165,058,237,244,137
5647 :040,005,059,176,011,173,223
5653 :243,040,133,057,173,244,143
5659 :040,133,058,096,056,165,063
5665 :057,237,002,041,133,059,050
5671 :165,058,237,003,041,005,036
5677 :059,176,001,096,173,002,040
5683 :041,133,057,173,003,041,243
5689 :133,058,096,230,057,208,071
5695 :002,230,058,076,134,021,072
5701 :165,057,208,002,198,058,245
5707 :198,057,076,134,021,165,214
5713 :057,133,251,165,058,133,110
5719 :252,198,252,160,255,177,101
5725 :251,201,032,240,004,201,254
5731 :031,208,003,136,208,243,160
5737 :177,251,201,032,240,008,246
5743 :201,031,240,004,136,208,163
5749 :243,096,056,152,101,251,248
5755 :133,057,165,252,105,000,067
5761 :133,058,076,134,021,160,199
5767 :000,177,057,201,032,240,074
5773 :008,201,031,240,004,200,057
5779 :208,243,096,200,208,011,089
5785 :230,058,165,058,205,003,104
5791 :041,144,002,208,025,177,244
5797 :057,201,032,240,236,201,108
5803 :031,240,232,024,152,101,183
5809 :057,133,057,165,058,105,240
5815 :000,133,058,076,134,021,093
5821 :173,002,041,133,057,173,000
5827 :003,041,133,058,076,134,128
5833 :021,169,000,141,252,040,056
5839 :173,003,041,056,233,004,205
5845 :205,244,040,176,003,173,030
5851 :244,040,141,253,040,032,201
5857 :158,018,076,189,022,238,158
5863 :005,023,173,005,023,041,245
5869 :015,141,005,023,010,010,185
5875 :010,010,133,059,173,005,121
5881 :023,041,007,024,105,008,201
5887 :101,059,141,015,144,096,043
5893 :001,238,020,023,173,020,224
5899 :023,041,007,141,020,023,010
5905 :076,158,018,000,165,057,235
5911 :133,251,165,058,133,252,247
5917 :198,252,160,255,177,251,042
5923 :201,046,240,012,201,033,000
5929 :240,008,201,063,240,004,029
5935 :201,031,208,004,136,208,067
5941 :235,096,177,251,201,046,035
5947 :240,027,201,033,240,023,055
5953 :201,063,240,019,201,031,052
5959 :240,015,136,208,235,198,079
5965 :252,165,252,205,243,040,210
5971 :176,226,076,110,023,132,058
5977 :059,198,059,200,240,010,087
5983 :177,251,201,032,240,247,219
5989 :136,076,119,022,164,059,165
5995 :076,055,023,173,243,040,205
6001 :133,057,173,244,040,133,125
6007 :058,076,134,021,160,000,056
6013 :177,057,201,046,240,029,107
6019 :201,033,240,025,201,063,126
6025 :240,021,201,031,240,017,119
6031 :200,208,235,230,058,165,215
6037 :058,205,003,041,240,226,154
6043 :144,224,076,189,022,200,242
6049 :208,014,230,058,165,058,126
6055 :205,003,041,144,005,240,037
6061 :003,076,189,022,177,057,185
6067 :201,032,240,233,201,046,108
6073 :240,229,201,033,240,225,073
6079 :201,063,240,221,201,031,124
6085 :240,217,076,174,022,173,075
6091 :247,040,141,119,041,173,196
6097 :248,040,141,120,041,032,063
6103 :250,019,169,093,160,039,177
6109 :032,108,019,169,001,141,179
6115 :254,040,096,056,165,057,127
6121 :237,243,040,133,059,165,086
6127 :058,237,244,040,005,059,114
6133 :208,003,104,104,096,165,157
6139 :057,133,038,165,058,133,067
6145 :039,096,056,165,057,133,035
6151 :158,073,255,101,038,141,005
6157 :123,041,165,058,133,159,180
6163 :073,255,101,039,141,124,240
```

```
6169 :041,165,038,141,125,041,064
6175 :165,039,141,126,041,165,196
6181 :158,141,127,041,133,038,163
6187 :165,159,141,128,041,133,042
6193 :039,056,173,124,041,109,079
6199 :120,041,205,250,040,144,087
6205 :020,032,250,019,169,108,147
6211 :160,039,032,108,019,169,082
6217 :001,141,254,040,169,000,166
6223 :133,198,096,173,119,041,071
6229 :133,158,173,120,041,133,075
6235 :159,173,123,041,133,180,132
6241 :024,109,119,041,141,119,138
6247 :041,173,124,041,133,181,028
6253 :109,120,041,141,120,041,169
6259 :032,035,018,173,125,041,027
6265 :133,038,173,126,041,133,253
6271 :039,173,127,041,133,158,030
6277 :173,128,041,133,159,056,055
6283 :173,002,041,229,158,133,107
6289 :180,173,003,041,229,159,162
6295 :133,181,032,035,018,056,094
6301 :173,002,041,237,123,041,006
6307 :141,002,041,173,003,041,052
6313 :237,124,041,141,003,041,244
6319 :096,032,230,023,032,069,145
6325 :022,032,003,024,056,173,235
6331 :119,041,233,001,141,119,073
6337 :041,173,120,041,233,000,033
6343 :141,120,041,096,173,141,143
6349 :002,201,005,208,003,076,188
6355 :081,025,032,060,022,032,207
6361 :230,023,032,069,022,032,113
6367 :003,024,076,185,024,032,055
6373 :202,023,169,002,032,014,159
6379 :020,032,250,019,169,120,077
6385 :160,039,032,108,019,032,119
6391 :125,019,072,032,234,019,236
6397 :104,041,191,201,023,208,253
6403 :009,032,230,023,032,080,153
6409 :022,076,003,024,201,019,098
6415 :208,009,032,230,023,032,037
6421 :021,023,076,003,024,201,113
6427 :016,208,009,032,230,023,033
6433 :032,215,026,076,003,024,153
6439 :096,056,165,057,237,252,134
6445 :040,133,059,165,058,237,225
6451 :253,040,005,059,240,011,147
6457 :173,252,040,133,057,173,117
6463 :253,040,133,058,096,173,048
6469 :243,040,133,057,173,244,191
6475 :040,133,058,076,134,021,025
6481 :165,057,133,251,133,158,210
6487 :165,058,133,252,133,159,219
6493 :160,000,177,251,201,032,146
6499 :208,030,200,208,247,165,133
6505 :252,205,000,041,144,015,253
6511 :173,002,041,133,251,173,116
6517 :003,041,133,252,160,000,194
6523 :076,131,025,230,252,076,145
6529 :095,025,024,152,101,251,009
6535 :133,038,169,000,101,252,060
6541 :133,039,056,173,002,041,073
6547 :229,158,133,180,173,003,255
6553 :041,229,159,133,181,056,184
6559 :165,038,229,158,141,123,245
6565 :041,165,039,229,159,141,171
6571 :124,041,032,035,018,056,221
6577 :173,002,041,237,123,041,026
6583 :141,002,041,173,003,041,072
6589 :237,124,041,141,003,041,008
6595 :096,169,255,141,148,041,021
6601 :076,222,025,169,005,141,071
6607 :148,041,032,222,025,177,084
6613 :057,201,032,208,001,200,144
6619 :076,174,022,169,000,141,033
6625 :149,041,032,037,026,169,167
6631 :032,174,148,041,160,000,018
6637 :145,057,200,202,208,250,019
6643 :096,032,015,026,032,015,203
6649 :026,169,031,160,000,145,012
6655 :057,200,145,057,032,158,136
6661 :018,032,060,022,032,060,229
6667 :022,076,204,025,169,001,252
6673 :141,148,041,169,000,141,145
6679 :149,041,032,037,026,169,221
6685 :032,160,000,145,057,076,243
6691 :134,021,024,173,002,041,174
6697 :109,148,041,173,003,041,044
6703 :109,149,041,205,246,040,069
6709 :144,005,104,104,076,116,090
6715 :026,024,165,057,133,038,246
6721 :109,148,041,133,158,165,051
6727 :058,133,039,109,149,041,088
6733 :133,159,056,173,002,041,129
6739 :229,038,133,180,173,003,071
6745 :041,229,039,133,181,032,232
6751 :095,018,024,173,002,041,193
6757 :109,148,041,141,002,041,071
6763 :173,003,041,109,149,041,111
6769 :141,003,041,096,173,255,054
6775 :040,073,006,141,255,040,162
6781 :096,169,135,160,039,032,244
6787 :108,019,032,228,255,240,245
6793 :251,201,147,240,247,041,240
6799 :127,201,089,096,169,002,059
6805 :032,014,020,032,250,019,004
6811 :169,148,160,039,032,108,043
6817 :019,032,126,026,240,003,095
6823 :076,234,019,162,250,154,038
6829 :032,050,019,032,195,019,008
6835 :076,038,020,160,000,177,138
6841 :057,201,031,240,017,200,163
6847 :208,247,230,058,165,058,133
6853 :205,003,041,144,238,240,044
6859 :236,076,189,022,200,208,110
6865 :002,230,058,076,174,022,003
6871 :165,057,133,251,165,058,020
6877 :133,252,198,252,160,255,191
6883 :177,251,201,031,240,017,120
6889 :136,192,255,208,245,198,187
6895 :252,165,252,205,244,040,117
6901 :176,236,076,110,023,056,154
6907 :152,101,251,133,251,169,028
6913 :000,101,252,133,252,056,027
6919 :165,251,229,057,133,059,133
6925 :165,252,229,058,005,059,013
6931 :208,018,132,059,024,165,113
6937 :251,229,059,133,251,165,089
6943 :252,233,000,133,252,076,209
6949 :233,026,165,251,133,057,134
6955 :165,252,133,058,076,134,093
6961 :021,173,141,002,041,001,172
6967 :208,003,032,202,023,032,043
6973 :250,019,169,158,160,039,088
6979 :032,108,019,160,000,177,051
6985 :057,073,128,145,057,032,053
6991 :158,018,160,000,177,057,137
6997 :073,128,145,057,169,002,147
7003 :032,014,020,032,125,019,077
7009 :009,064,201,087,208,009,163
7015 :032,144,027,032,134,022,238
7021 :076,159,027,201,083,208,095
7027 :009,032,144,027,032,123,226
7033 :023,076,159,027,201,080,175
7039 :208,009,032,144,027,032,067
7045 :182,026,076,159,027,032,123
7051 :134,021,076,234,019,165,020
7057 :057,133,158,141,113,041,020
7063 :165,058,133,159,141,114,153
7069 :041,096,056,165,057,133,193
7075 :038,237,113,041,141,123,088
7081 :041,165,058,133,039,237,074
7087 :114,041,141,124,041,032,156
7093 :026,024,173,113,041,133,179
7099 :057,173,114,041,133,058,251
7105 :032,158,018,076,070,027,062
7111 :169,044,229,211,141,004,229
7117 :041,160,000,169,166,032,005
7123 :210,255,169,157,032,210,220
7129 :255,140,005,041,032,125,047
7135 :019,172,005,041,133,059,140
7141 :169,032,032,210,255,169,072
7147 :157,032,210,255,165,059,089
7153 :201,013,240,050,201,020,198
7159 :208,015,136,016,004,200,058
7165 :076,208,027,169,157,032,154
7171 :210,255,076,208,027,165,176
7177 :059,041,127,201,032,144,101
7183 :192,204,004,041,240,187,115
7189 :165,059,153,048,041,032,007
7195 :210,255,169,000,133,212,238
7201 :133,216,200,076,208,027,125
7207 :032,210,255,169,000,153,090
7213 :048,041,152,096,032,250,152
7219 :019,169,214,160,039,032,172
7225 :108,019,032,148,028,176,056
7231 :032,173,243,040,133,251,167
7237 :173,244,040,133,252,174,061
7243 :002,041,172,003,041,169,247
7249 :251,032,216,255,176,009,252
7255 :165,144,041,191,208,003,071
7261 :076,130,029,240,036,173,009
7267 :147,028,201,008,144,006,121
7273 :032,202,036,076,128,028,095
7279 :173,147,028,201,001,240,133
7285 :249,032,250,019,169,220,032
7291 :160,039,032,108,019,169,138
7297 :001,141,254,040,096,032,181
7303 :250,019,169,231,160,039,235
7309 :032,108,019,076,128,028,020
7315 :000,032,199,027,240,022,155
7321 :169,006,160,040,032,108,156
7327 :019,032,125,019,162,008,012
7333 :201,068,240,012,162,001,081
7339 :201,084,240,006,032,234,200
7345 :019,104,104,096,142,147,021
7351 :028,169,001,160,000,032,061
7357 :186,255,160,000,224,001,247
7363 :240,049,185,048,041,201,191
7369 :064,208,014,185,049,041,250
7375 :201,058,240,035,185,050,208
7381 :041,201,058,240,028,169,182
7387 :048,141,088,041,169,058,252
7393 :141,089,041,185,048,041,002
7399 :153,090,041,200,204,005,156
7405 :041,144,244,240,242,200,068
7411 :076,002,029,185,048,041,112
7417 :153,088,041,200,204,005,172
7423 :041,208,244,140,112,041,017
7429 :032,250,019,169,048,160,171
7435 :041,032,108,019,173,112,240
7441 :041,162,088,160,041,032,029
7447 :189,255,169,013,076,210,167
7453 :255,032,250,019,169,196,182
7459 :160,039,032,108,019,032,169
7465 :125,019,032,026,020,009,016
7471 :128,072,173,255,040,240,187
7477 :003,032,015,026,032,234,139
7483 :019,104,076,166,020,056,244
7489 :165,057,237,243,040,133,172
7495 :251,165,058,237,244,040,042
7501 :005,251,240,007,169,005,242
7507 :133,251,032,014,020,032,053
7513 :250,019,169,026,160,040,241
7519 :032,108,019,032,148,028,206
7525 :165,251,201,005,240,003,198
7531 :032,050,019,169,000,166,031
7537 :057,164,058,032,213,255,124
7543 :144,003,076,096,028,142,096
7549 :002,041,140,003,041,032,128
7555 :234,251,032,231,255,032,142
7561 :250,019,169,252,160,039,002
7567 :032,108,019,076,128,028,022
7573 :032,250,019,169,032,160,043
7579 :040,032,108,019,032,148,022
7585 :028,169,001,174,243,040,048
7591 :172,244,040,032,213,255,099
7597 :165,144,041,191,240,207,137
7603 :032,250,019,169,239,160,024
7609 :039,032,108,019,076,128,075
7615 :028,169,147,032,210,255,008
7621 :169,013,032,210,255,032,140
7627 :236,029,169,013,032,210,124
7633 :255,169,040,160,040,032,137
7639 :108,019,032,228,255,201,034
7645 :013,208,249,076,234,019,252
7651 :032,204,255,169,001,032,152
7657 :195,255,096,032,231,255,017
7663 :169,001,162,008,160,000,227
7669 :032,186,255,169,001,162,026
7675 :069,160,040,032,189,255,228
7681 :032,192,255,176,221,162,015
7687 :001,032,198,255,032,081,094
7693 :030,032,081,030,032,081,043
7699 :030,032,081,030,240,202,122
7705 :032,204,255,032,228,255,007
7711 :201,032,208,003,032,125,120
7717 :019,162,001,032,198,255,192
7723 :032,081,030,072,032,081,115
7729 :030,168,104,170,152,032,193
7735 :205,221,169,032,032,210,156
7741 :255,032,081,030,240,006,193
7747 :032,210,255,076,062,030,220
```

```
7753 :169,013,032,210,255,076,060
7759 :017,030,032,207,255,072,180
7765 :165,144,041,191,240,006,104
7771 :104,104,104,076,227,029,223
7777 :104,096,162,000,142,115,204
7783 :041,142,116,041,142,117,190
7789 :041,142,118,041,056,177,172
7795 :251,233,048,144,042,201,010
7801 :010,176,038,014,115,041,003
7807 :046,116,041,014,115,041,244
7813 :046,116,041,014,115,041,250
7819 :046,116,041,014,115,041,000
7825 :046,116,041,013,115,041,005
7831 :141,115,041,200,208,212,044
7837 :230,252,076,113,030,248,082
7843 :173,115,041,013,116,041,150
7849 :240,028,056,173,115,041,054
7855 :233,001,141,115,041,173,111
7861 :116,041,233,000,141,116,060
7867 :041,238,117,041,208,003,067
7873 :238,118,041,076,163,030,091
7879 :173,117,041,216,096,056,130
7885 :173,119,041,237,247,040,038
7891 :141,121,041,173,120,041,080
7897 :237,248,040,141,122,041,022
7903 :013,121,041,208,016,032,142
7909 :250,019,169,078,160,040,177
7915 :032,108,019,169,001,141,193
7921 :254,040,096,024,165,057,109
7927 :133,038,109,121,041,133,054
7933 :158,165,058,133,039,109,147
7939 :122,041,133,159,056,173,175
7945 :002,041,229,038,133,180,120
7951 :173,003,041,229,039,133,121
7957 :181,024,101,159,205,246,169
7963 :040,144,016,032,250,019,016
7969 :169,070,160,040,032,108,100
7975 :019,169,001,141,254,040,151
7981 :096,032,096,018,024,173,228
7987 :121,041,133,180,109,002,125
7993 :041,141,002,041,173,122,065
7999 :041,133,181,109,003,041,059
8005 :141,003,041,165,057,133,097
8011 :158,165,058,133,159,173,153
8017 :247,040,133,038,173,248,192
8023 :040,133,039,032,035,018,128
8029 :076,134,021,160,000,177,149
8035 :057,170,200,177,057,136,128
8041 :145,057,200,138,145,057,079
8047 :096,160,000,177,057,041,130
8053 :063,240,010,201,027,176,066
8059 :006,177,057,073,064,145,133
8065 :057,076,060,022,133,059,024
8071 :041,063,006,059,036,059,143
8077 :016,002,009,128,112,002,154
8083 :009,064,133,059,096,005,001
8089 :075,066,005,058,001,001,103
8095 :001,000,001,000,080,027,012
8101 :014,015,018,141,154,041,036
8107 :138,072,152,072,056,173,066
8113 :138,041,237,140,041,173,179
8119 :139,041,237,141,041,144,158
8125 :025,173,154,041,032,210,056
8131 :255,173,141,002,041,001,040
8137 :208,249,165,145,201,127,016
8143 :208,006,032,136,034,076,187
8149 :168,033,104,168,104,170,192
8155 :173,154,041,096,032,250,197
8161 :019,169,173,160,040,076,094
8167 :108,019,076,168,033,169,036
8173 :000,032,189,255,173,020,138
8179 :023,141,134,002,169,004,204
8185 :141,149,041,160,007,173,152
8191 :141,002,041,001,208,003,139
8197 :076,152,032,032,250,019,054
8203 :169,097,160,040,032,108,105
8209 :019,032,125,019,041,127,124
8215 :162,003,142,149,041,201,209
8221 :083,240,086,162,008,142,238
8227 :149,041,201,068,240,034,000
8233 :201,080,208,188,032,250,232
8239 :019,169,127,160,040,032,082
8245 :108,019,032,125,019,056,156
8251 :233,048,201,004,144,168,089
8257 :201,080,176,164,141,149,208
8263 :041,076,118,032,032,250,108
8269 :019,169,163,160,040,032,148
8275 :108,019,032,199,027,240,196
8281 :143,172,005,041,169,044,151
8287 :153,048,041,200,169,087,025
8293 :153,048,041,200,140,005,176
8299 :041,173,005,041,162,048,065
8305 :160,041,032,189,255,173,195
8311 :149,041,168,201,004,144,058
8317 :026,201,008,176,022,032,078
8323 :250,019,169,142,160,040,143
8329 :032,108,019,032,125,019,216
8335 :056,233,048,168,016,003,155
8341 :076,233,031,169,001,174,065
8347 :149,041,032,186,255,032,082
8353 :223,031,169,001,032,195,044
8359 :255,032,192,255,162,001,040
8365 :032,201,255,144,003,076,116
8371 :168,033,162,000,142,130,046
8377 :041,142,129,041,142,150,062
8383 :041,142,151,041,142,110,050
8389 :045,189,152,031,157,131,134
8395 :041,232,224,012,208,245,141
8401 :169,255,141,145,041,141,077
8407 :143,041,162,004,189,163,149
8413 :031,157,029,042,202,208,122
8419 :247,173,243,040,133,251,034
8425 :173,244,040,133,252,160,211
8431 :000,140,144,041,204,143,143
8437 :041,240,006,173,131,041,109
8443 :141,144,041,177,251,016,253
8449 :003,076,150,034,201,031,240
8455 :240,044,153,109,042,200,027
8461 :238,144,041,173,144,041,026
8467 :205,132,041,144,230,140,143
8473 :001,041,177,251,201,032,216
8479 :240,020,206,144,041,136,050
8485 :208,244,172,001,041,076,011
8491 :056,033,200,177,251,201,193
8497 :032,240,001,136,140,001,087
8503 :041,152,056,101,251,133,021
8509 :251,165,252,105,000,133,199
8515 :252,160,000,173,145,041,070
8521 :201,255,208,003,032,057,061
8527 :034,173,143,041,240,003,201
8533 :032,101,034,056,046,143,241
8539 :041,173,001,041,141,000,232
8545 :041,169,109,133,253,169,203
8551 :042,133,254,032,103,038,193
8557 :032,118,034,173,145,041,140
8563 :205,135,041,144,003,032,163
8569 :199,033,056,165,251,237,038
8575 :002,041,133,059,165,252,011
8581 :237,003,041,005,059,240,206
8587 :056,144,054,173,130,041,225
8593 :240,011,169,000,141,129,067
8599 :041,141,134,041,032,199,227
8605 :033,173,149,041,201,003,245
8611 :208,003,032,125,019,032,070
8617 :225,255,240,251,173,255,032
8623 :040,141,134,002,169,001,150
8629 :032,195,255,032,231,255,157
8635 :162,250,154,032,234,019,014
8641 :076,038,020,076,238,032,161
8647 :056,173,133,041,237,145,216
8653 :041,168,136,136,240,008,166
8659 :048,006,032,136,034,136,091
8665 :208,250,173,130,041,240,235
8671 :017,141,000,041,169,110,189
8677 :133,253,169,044,133,254,191
8683 :032,101,034,032,103,038,063
8689 :032,136,034,032,136,034,133
8695 :032,136,034,238,138,041,098
8701 :208,003,238,139,041,173,031
8707 :137,041,208,050,173,149,249
8713 :041,201,003,240,043,201,226
8719 :008,240,039,056,173,138,157
8725 :041,237,140,041,173,139,024
8731 :041,237,141,041,144,024,143
8737 :032,204,255,032,250,019,057
8743 :169,188,160,040,032,108,224
8749 :019,032,125,019,032,223,239
8755 :031,162,001,032,201,255,221
8761 :173,129,041,240,017,141,030
8767 :000,041,169,109,133,253,000
8773 :169,043,133,254,032,101,033
8779 :034,032,103,038,032,136,194
8785 :034,172,134,041,140,145,235
8791 :041,136,136,240,008,048,184
8797 :006,032,136,034,136,208,133
8803 :250,096,169,032,172,131,181
8809 :041,140,144,041,240,006,205
8815 :032,168,031,136,208,250,168
8821 :096,172,136,041,024,152,226
8827 :109,145,041,141,145,041,233
8833 :032,136,034,136,208,250,157
8839 :096,169,013,032,168,031,132
8845 :173,110,045,240,003,032,232
8851 :168,031,096,141,147,041,003
8857 :041,127,032,133,031,174,179
8863 :225,034,221,225,034,240,114
8869 :009,202,208,248,206,144,158
8875 :041,076,242,035,202,138,137
8881 :010,170,140,146,041,169,085
8887 :034,072,169,196,072,189,147
8893 :245,034,072,189,244,034,239
8899 :072,096,056,173,146,041,011
8905 :101,251,133,251,165,252,074
8911 :105,000,133,252,076,238,243
8917 :032,177,251,201,031,240,121
8923 :001,136,140,146,041,096,011
8929 :018,087,065,076,082,084,125
8935 :066,083,078,072,070,064,152
8941 :080,063,088,077,073,071,177
8947 :074,084,035,102,035,111,172
8953 :035,121,035,131,035,141,235
8959 :035,151,035,161,035,176,080
8965 :035,210,035,058,035,074,196
8971 :035,042,035,032,035,023,213
8977 :035,235,035,020,036,093,215
8983 :035,200,169,000,141,143,199
8989 :041,076,214,034,200,032,114
8995 :099,030,141,142,041,076,052
9001 :214,034,200,032,099,030,138
9007 :141,140,041,173,118,041,189
9013 :141,141,041,076,214,034,188
9019 :200,032,099,030,141,138,187
9025 :041,173,118,041,141,139,206
9031 :041,076,214,034,200,032,156
9037 :099,030,141,133,041,076,085
9043 :214,034,169,000,141,137,010
9049 :041,200,076,214,034,169,055
9055 :010,141,110,045,200,076,165
9061 :214,034,200,169,001,141,092
9067 :150,041,076,214,034,200,054
9073 :032,099,030,141,131,041,075
9079 :076,214,034,200,032,099,006
9085 :030,141,132,041,076,214,247
9091 :034,200,032,099,030,141,155
9097 :134,041,076,214,034,200,068
9103 :032,099,030,141,135,041,109
9109 :076,214,034,200,032,099,036
9115 :030,141,136,041,076,214,025
9121 :034,172,146,041,200,152,138
9127 :072,032,199,033,104,168,007
9133 :140,146,041,096,032,203,063
9139 :035,136,140,129,041,160,052
9145 :001,177,251,153,108,043,150
9151 :200,204,129,041,144,245,130
9157 :240,243,200,076,214,034,180
9163 :200,177,251,201,031,208,247
9169 :249,096,032,203,035,136,192
9175 :140,130,041,160,001,177,096
9181 :251,153,109,044,200,204,158
9187 :130,041,144,245,240,243,246
9193 :076,214,034,032,203,035,059
9199 :076,214,034,200,177,251,167
9205 :201,061,240,007,136,173,039
9211 :147,041,076,009,033,200,245
9217 :032,099,030,072,173,147,042
9223 :041,041,127,170,104,157,135
9229 :237,041,032,214,034,076,135
9235 :197,034,200,162,008,177,029
9241 :251,041,063,201,004,240,057
9247 :009,162,001,201,020,240,152
9253 :003,076,233,031,142,147,157
9259 :028,200,177,251,201,058,190
9265 :240,003,076,233,031,200,064
9271 :177,251,201,031,240,009,196
9277 :032,133,031,153,085,041,024
9283 :076,054,036,152,056,233,162
9289 :003,162,088,160,041,032,047
9295 :189,255,032,204,255,169,159
9301 :002,032,195,255,169,002,228
9307 :174,147,028,160,000,032,120
9313 :186,255,032,050,019,169,040
9319 :000,166,057,164,058,032,068
9325 :213,255,144,003,076,233,009
9331 :031,142,002,041,140,003,218
```

```
9337 :041,104,104,162,001,032,053
9343 :201,255,076,228,032,032,183
9349 :231,255,169,000,032,189,241
9355 :255,169,015,162,008,160,140
9361 :015,032,186,255,032,192,089
9367 :255,144,011,169,015,032,009
9373 :195,255,032,231,255,076,177
9379 :234,019,032,250,019,169,118
9385 :055,160,040,032,108,019,071
9391 :032,199,027,240,022,162,089
9397 :015,032,201,255,176,223,059
9403 :169,048,160,041,032,108,233
9409 :019,169,013,032,210,255,123
9415 :032,204,255,032,231,255,184
9421 :169,000,032,189,255,169,251
9427 :015,162,008,160,015,032,091
9433 :186,255,032,192,255,176,033
9439 :186,032,250,019,162,015,119
9445 :032,198,255,032,199,027,204
9451 :032,204,255,169,015,032,174
9457 :195,255,032,231,255,169,098
9463 :001,141,254,040,096,032,043
9469 :036,037,173,155,041,240,167
9475 :022,032,199,037,032,074,143
9481 :037,173,153,041,201,255,101
9487 :240,009,032,234,037,032,087
9493 :158,018,076,007,037,076,137
9499 :234,019,173,141,002,201,029
9505 :005,208,038,032,250,019,073
9511 :169,209,160,040,032,108,245
9517 :019,032,199,027,141,155,106
9523 :041,208,003,076,234,019,120
9529 :160,000,185,048,041,153,132
9535 :156,041,200,204,005,041,198
9541 :208,244,076,234,019,165,247
9547 :057,133,251,165,058,133,104
9553 :252,169,255,141,153,041,068
9559 :160,001,162,000,173,155,226
9565 :041,240,080,189,156,041,072
9571 :032,026,020,209,251,240,109
9577 :002,162,255,200,208,011,175
9583 :230,252,165,252,205,003,194
9589 :041,240,002,176,054,232,094
9595 :236,155,041,208,224,024,243
9601 :152,101,251,133,059,165,222
9607 :252,105,000,133,060,173,090
9613 :002,041,197,059,173,003,104
9619 :041,229,060,144,024,056,189
9625 :165,059,237,155,041,133,175
9631 :057,141,152,041,165,060,007
9637 :233,000,133,058,141,153,115
9643 :041,032,134,021,096,032,015
9649 :250,019,169,219,160,040,010
9655 :032,108,019,169,001,141,141
9661 :254,040,096,173,141,002,127
9667 :201,005,208,035,032,250,158
9673 :019,169,229,160,040,032,082
9679 :108,019,032,199,027,141,221
9685 :196,041,240,014,160,000,096
9691 :185,048,041,153,197,041,116
9697 :200,204,005,041,208,244,103
9703 :076,234,019,056,165,057,070
9709 :133,158,237,152,041,133,067
9715 :059,165,058,133,159,237,030
9721 :153,041,005,059,208,101,048
9727 :169,255,141,153,041,024,014
9733 :173,155,041,101,057,133,153
9739 :038,169,000,101,058,133,254
9745 :039,056,173,002,041,229,045
9751 :158,133,180,173,003,041,199
9757 :229,159,133,181,032,035,030
9763 :018,056,173,002,041,237,050
9769 :155,041,141,002,041,173,082
9775 :003,041,233,000,141,003,212
9781 :041,173,196,041,240,041,017
9787 :141,148,041,169,000,141,187
9793 :149,041,032,037,026,160,254
9799 :000,185,197,041,032,026,040
9805 :020,145,057,200,204,196,131
9811 :041,208,242,024,165,057,052
9817 :109,196,041,133,057,165,022
9823 :058,105,000,133,058,076,013
9829 :134,021,160,000,204,000,108
9835 :041,240,032,177,253,048,130
9841 :029,032,133,031,032,252,110
9847 :038,032,168,031,173,151,200
9853 :041,240,010,169,008,032,113
9859 :168,031,169,095,032,168,026
9865 :031,200,076,105,038,096,171
9871 :140,146,041,041,127,141,011
9877 :147,041,032,133,031,201,222
9883 :067,208,027,056,173,142,060
9889 :041,237,000,041,074,056,098
9895 :237,131,041,168,169,032,177
9901 :032,168,031,136,208,250,230
9907 :172,146,041,076,138,038,022
9913 :201,069,208,017,056,173,141
9919 :132,041,237,000,041,056,186
9925 :237,131,041,168,169,032,207
9931 :076,173,038,201,085,208,216
9937 :008,173,151,041,073,001,144
9943 :141,151,041,201,035,208,224
9949 :018,140,146,041,174,138,110
9955 :041,173,139,041,032,205,090
9961 :221,172,146,041,076,138,003
9967 :038,174,147,041,189,237,041
9973 :041,032,168,031,076,138,219
9979 :038,174,150,041,240,026,152
9985 :133,059,041,127,201,065,115
9991 :144,018,201,091,176,014,139
9997 :170,165,059,041,128,073,137
10003 :128,074,074,133,059,138,113
10009 :005,059,096,032,250,019,230
10015 :056,173,245,040,237,002,016
10021 :041,170,173,246,040,237,176
10027 :003,041,032,205,221,169,202
10033 :001,141,254,040,096,008,077
10039 :014,211,080,069,069,068,054
10045 :211,067,082,073,080,084,146
10051 :032,051,046,048,000,013,001
10057 :018,066,089,032,195,072,033
10063 :065,082,076,069,083,032,230
10069 :194,082,065,078,078,079,149
10075 :078,000,194,085,070,070,076
10081 :069,082,032,195,076,069,108
10087 :065,082,069,068,000,194,069
10093 :085,070,070,069,082,032,005
10099 :198,085,076,076,000,196,234
10105 :069,076,069,084,069,032,008
10111 :040,211,044,215,044,208,121
10117 :041,000,058,032,211,085,048
10123 :082,069,063,032,217,047,137
10129 :206,058,000,197,210,193,241
10135 :211,197,032,193,204,204,168
10141 :000,197,082,065,083,069,141
10147 :032,040,211,044,215,044,237
10153 :208,041,013,018,208,082,227
10159 :069,083,083,032,146,210,030
10165 :197,212,213,210,206,018,213
10171 :032,084,079,032,069,088,059
10177 :073,084,000,208,082,069,197
10183 :083,083,032,070,079,082,116
10189 :077,065,084,032,075,069,095
10195 :089,058,000,211,065,086,208
10201 :069,058,000,212,065,080,189
10207 :069,032,197,210,210,207,124
10213 :210,000,211,084,079,080,125
10219 :080,069,068,000,214,069,223
10225 :082,073,070,089,032,197,016
10231 :082,082,079,082,000,206,010
10237 :079,032,069,082,082,079,164
10243 :082,083,000,147,032,018,109
10249 :212,146,065,080,069,032,101
10255 :079,082,032,018,196,146,056
10261 :073,083,075,063,000,204,007
10267 :079,065,068,058,000,214,255
10273 :069,082,073,070,089,058,218
10279 :000,208,082,069,083,083,052
10285 :032,018,210,197,212,213,159
10291 :210,206,146,000,196,073,114
10297 :083,075,032,067,079,077,214
10303 :077,065,078,068,058,000,153
10309 :036,206,079,032,210,079,199
10315 :079,077,000,206,079,032,036
10321 :084,069,088,084,032,073,255
10327 :078,032,066,085,070,070,232
10333 :069,082,046,000,147,018,199
10339 :211,146,067,082,069,069,231
10345 :078,044,032,018,196,146,107
10351 :073,083,075,044,032,018,180
10357 :208,146,082,073,078,084,020
10363 :069,082,063,000,196,069,090
10369 :086,073,067,069,032,078,022
10375 :085,077,066,069,082,063,065
10381 :000,211,069,067,079,078,133
10387 :068,065,082,089,032,[illegible],164
10393 :068,068,082,069,083,[illegible],094
10399 :032,035,063,000,198,[illegible],048
10405 :076,069,078,065,077,[illegible],087
10411 :058,000,147,208,082,[illegible],227
10417 :078,084,073,078,071,046,095
10423 :046,046,013,013,000,206,251
10429 :069,088,084,032,083,072,105
10435 :069,069,084,044,032,146,127
10441 :210,197,212,213,210,206,169
10447 :018,000,200,085,078,084,160
10453 :032,070,079,082,058,000,022
10459 :206,079,084,032,198,079,129
10465 :085,078,068,000,210,069,223
10471 :080,076,065,067,069,058,134
10477 :000,209,213,201,212,000,048
```

193

©

IBM Graphics Printer Switch Settings

Michael A. Covington

Although neither the IBM PC reference manuals nor the instructions that come with the printer mention them, the IBM Graphics Printer has a set of internal DIP switches which allow you to control how it operates.

The switch settings within the IBM Graphics Printer determine the defaults that apply when the printer is first turned on; almost all of them can be overridden by sending appropriate escape codes to the printer. But there may be situations in which you'll want to change the defaults.

To get at the switches, unplug the printer, disconnect the interface cable, remove the plastic cover and wire-grid paper guide, and turn the printer upside down. Unscrew the four Phillips-head screws at the corners, then put tape over the deep holes they sit in so they won't fall out. Now turn the printer right side up, pull off the paper advance knob, and carefully lift off the cover, maneuvering it clear of the knob shaft.

On the main circuit board you should find two sets of DIP switches under removable plastic dust covers. Using a ballpoint pen or similar tool, set them according to your preference (see accompanying table), put the dust covers back in place, and reassemble the printer.

The most useful thing the switches can do for you is give you access to the full character set. The *IBM PC Guide to Operations* lists two character sets for the printer; in character set 1, ASCII codes 128 to 159 are duplicates of codes 0 to 31, but in character set 2, they are accented letters for foreign languages. (Both character sets include a variety of mathematical symbols and box-drawing characters.)

A few programs may not work properly with character set 2; if you have this problem, you can either set the switch back to its original setting, or set the printer back into character set 1 by sending it ASCII codes 27 and 55 as an initialization sequence.

IBM Graphics Printer Internal Switch Settings

Asterisks mark how switches are set at the factory.

A. Large set of 8 switches:

1 Not used; normally on.

2 Off: Printer generates a linefeed of its own after every carriage return.
***On: Printer does not advance to next line until it receives a linefeed character (ASCII 10).**

3 *Off: When more characters are received than will fit on a line, printer begins a new line.
On: When more characters are received than will fit on a line, printer overprints on same line.

4 *Off: ASCII code 24 clears the printer buffer.
On: ASCII code 24 has no effect.

5 Not used; normally on.

6 Off: Buzzer on printer will not sound.
***On: Buzzer sounds when out of paper or when ASCII code 7 is received.**

7 *Off: Character set 1.
On: Character set 2.

8 Off: Computer sends "Select" signal to activate printer.
***On: Printer is always ready to receive input.**

B. Small set of 4 switches:

1 *Off: Paper length is 11 inches.
On: Paper length is 12 inches.

2 *Off: Lines are spaced 6 to the inch.
On: Lines are spaced 8 to the inch.

3 *Off: Paper feeding is controlled by computer.
On: Paper automatically advances after printing.

4 *Off: Printer does not skip over the perforation where pages join.
On: Printer skips 1 inch where pages join. ©

Creating Atari Machine Language Strings

Tom Sak

This clever utility program converts a machine language subroutine into fast-executing BASIC string statements and stores them on disk for later use. Requires at least 16K RAM.

The most common way to use a machine language subroutine in a BASIC program is to convert the object code into decimal numbers, put the numbers into DATA statements, then READ the numbers and POKE them into memory.

However, if you'd like your programs to initialize faster, or if you're running short of memory, there's a better technique you should consider: converting the machine language into strings. Using string assignment statements instead of DATA statements not only saves the time required to POKE the numbers into memory, it also consumes only about one-third as much RAM. The main limitation of this technique is that the machine language routine must be completely relocatable—not a serious handicap for short (under 256-byte) routines.

The listing following this article, "ML String Creator," is a self-modifying BASIC program that automatically creates string assignment statements from your object code and LISTs them to disk for inclusion in other BASIC programs.

Direct Execution From A String

The string technique works because, essentially, these statements are equivalent:

```
CJ 1Ø DATA 33,37,1Ø6,47,122,65
OD 3Ø A$="!%j/zA"
```

If your subroutine contains internal JMPs or JSRs, which are not relocatable, you must use the conventional DATA statement technique. Until a BASIC program runs, you don't know where a certain string will end up in memory; therefore, if you encode your machine language (ML) into a string, it will end up at an unpredictable memory address. However, when the ML is relocatable, it is possible to execute the subroutine directly from the string with a statement like this:

```
50 X=USR(ADR(A$))
```

The ADR() function lets you find the beginning address of the string (and therefore of your subroutine). Of course, this assumes you have previously encoded the ML into the string variable A$ with ML String Creator.

The string assignment statement also is preferable when you're trying to squeeze a few more bytes into limited memory. Each ML byte has a decimal value in the range of 0–255. Representing this in a decimal DATA statement requires as many as three bytes, plus a comma to separate the entries. In a string assignment, each ML byte is represented as a single character.

There are a few other limitations, however. It's not possible to represent the decimal values 155 or 34 inside quotes in a string assignment. The value 155 represents a carriage return or end-of-line marker which cannot be embedded in the assignment statement, even as part of an escape sequence. The value 34 represents the double-quote character used as a delimiter in the assignment statement.

Stringing It All Together

Keeping these limitations in mind, you can use ML String Creator to locate an ML subroutine somewhere in memory, turn it into one or more string assignment statements, and LIST the statements to disk. It is your responsibility to initially load the ML into memory. If you're using an assembler that lets you switch back to BASIC without erasing memory, you can assemble directly to memory and then load ML String Creator to convert the object code into strings.

The program begins by requesting that you supply the first and last memory addresses (in decimal) of your routine, the name of the string variable to be created, and a line number for the first string assignment statement. A maximum of 80 bytes can be contained in a single statement string, and the maximum ML program length accommodated by the program is 256 bytes.

The string variable name is limited to seven characters, including the trailing $ symbol which must be present. Finally, the line number for the first string assignment statement must be greater than 190. Subsequent lines are numbered in increments of ten.

ML String Creator is self-modifying; the string assignment statements become a part of the program. However, the part of the program which is taking care of business protects itself from modification. The program can be used repeatedly without being reloaded, but it will grow in size.

The self-modification feature is also used to produce a LIST statement at line 150. In the listing below it appears as a REM statement, but after the string assignment statements are created it will be modified.

Finally, ML String Creator will prompt you for the filename of the disk file in which it will store the assignment statements. This filename and the first and last statement numbers of the created statements are concatenated with 150 LIST, in addition to the appropriate commas and double quotes, to form a genuine LIST statement.

Checking For Quotes And Carriage Returns

Before retiring, the program will indicate the memory locations, if any, at which a decimal value of either 155 or 34 was encountered. The program substitutes a value of zero in these instances. If more than ten occurrences of 155 or 34 are detected, the program stops with an error message.

The technique used to create the strings consists of printing string assignment statements on a previously cleared screen, just as you would do from the keyboard if you were typing in a BASIC program. After the last string assignment statement is placed on the screen, a CONT statement is written on the screen in immediate mode (that is, with no statement number).

Another feature of the program is its automatic RETURN. Normally when you press RETURN after typing a BASIC statement, the statement is either immediately executed (for example, LIST) or incorporated into your BASIC program (for example, 10 A=B*C). The Atari has a switch which makes pressing the RETURN key optional. The switch is location 842, which usually contains a 12. POKE 842,13 switches to automatic RETURN.

Brace Yourself For Fast Action

Processing takes place rapidly when the computer presses RETURN, so be prepared. The commands to be processed must be both correct and in the right place on the screen, and the cursor must be positioned on or above the first statement. If an error is detected, a message will be written on the screen, but the Atari, using the automatic RETURN, will process the error message as a command and a syntax error will result.

Lines 50 and 70 write the string assignment statements onto the screen. Line 85 places CONT on the screen and positions the cursor at the top, well above the first statement to be processed. The switch at location 842 is set at line 90. Then the program is stopped. When you are entering BASIC statements from the keyboard, you don't have one of your BASIC programs executing, and that is what is happening here, except that the text is "typed," the cursor is positioned, and RETURN supplied by the computer.

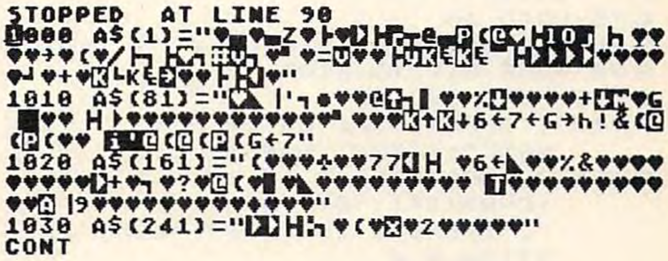
Watching The Atari Type

If you want to watch this action, you can see most of it by looking at the screen carefully. Insert the following statement to see what the screen looks like immediately before processing:

```
86 GOTO 86
```

Press BREAK to regain control; a STOPPED AT LINE 86 message will be displayed, destroying portions of the information which you are attempting to view.

The figure below depicts a typical screen image immediately following the STOP statement in line 90 and just before the automatic RETURN. (Of course, the actual string characters will vary depending on the ML subroutine you are reading.) Don't forget to delete line 86 when you've seen enough.



The CONT statement is the last one executed by the flying cursor before it returns control to your program. (The immediate execution of GOTO 100 would have the same effect.) The same technique is used to create and incorporate the LIST statement.

With a little imagination, you can modify this program to accept other forms of input of decimal or hexadecimal values to be converted to character strings, or to accept an ML object file from disk.

If you are interested in adapting some of these techniques to your own programs, there are a few things to watch out for. First, when placing the cursor at the top of the screen prior to activating the automatic RETURN, be sure to allow sufficient room so the screen text produced by the STOP statement won't overwrite the statements which your program placed on the screen. Second, be sure to turn off the automatic RETURN (POKE 842,12) when you're done.

ML String Creator

Please refer to "COMPUTE!'s Guide To Typing In Programs" before entering this listing.

```
PF 1 REM ML STRING MAKER
HA 2 REM Writes string assignment sta
     tements from up to 256 memory lo
     cations and LISTs them on disk.
AJ 1Ø DIM NAME$(1Ø),RTN(11),RTN1(11)
EC 2Ø ? CHR$(125);"ENTER -":? "START
      ADDRESS";:INPUT FBA:? " END ADD
      RESS";:INPUT LBA
JN 25 ? " STRING NAME";:INPUT NAME$:?
       " FIRST STATEMENT NUMBER";:INP
      UT FSN:SN=FSN-1Ø:I=LEN(NAME$)
CC 3Ø IF LBA<FBA OR LBA-FBA>255 OR I<
      2 OR I>7 OR NAME$(I,I)<>"$" OR
      FSN<191 THEN ? CHR$(253):GOTO 2
      Ø
OM 35 ? CHR$(125):? :DISP=-79:FBA=FBA
      -8Ø
DJ 4Ø SN=SN+1Ø:FBA=FBA+8Ø:DISP=DISP+8
      Ø:IF FBA>LBA THEN GOTO 85
FL 45 RANGE=79:IF LBA-FBA<79 THEN RAN
      GE=LBA-FBA
FO 5Ø ? SN;" ";NAME$;"(";DISP;")=";CH
      R$(34);:FOR I=FBA TO FBA+RANGE:
      J=PEEK(I)
DC 6Ø IF J=155 THEN J=Ø:K=K+1:RTN(K)=
      I:IF K=11 THEN 19Ø
DI 65 IF J=34 THEN J=Ø:L=L+1:RTN1(L)=
      I:IF L=11 THEN 19Ø
AL 7Ø ? "{ESC}";CHR$(J);:NEXT I:? CHR
      $(34)
AJ 75 GOTO 4Ø
FP 85 ? "CONT":POSITION Ø,Ø
EG 9Ø POKE 842,13:STOP
ON 1ØØ POKE 842,12
CN 11Ø ? CHR$(125);"ENTER -":? " FILE
       NAME";:INPUT NAME$
LA 12Ø ? CHR$(125):? :? :? "15Ø LIST"
       ;CHR$(34);"D:";NAME$;CHR$(34);
       ",";FSN;",";SN-1Ø:? "CONT":POS
       ITION Ø,Ø
HB 13Ø POKE 842,13:STOP
PB 14Ø POKE 842,12
NM 145 ? CHR$(125);"LISTING ";NAME$
DK 15Ø REM LIST statement will be ins
       erted here.
LP 16Ø ? CHR$(125):IF K>Ø THEN ? "Zer
       o substituted for 155 @":FOR I
       =1 TO K:? " ";RTN(I);:NEXT I
NF 17Ø IF L>Ø THEN ? :? "7ero substit
       uted for 34 @":FOR I=1 TO L:?
       " ";RTN1(I);:NEXT I
HA 18Ø END
ED 19Ø ? CHR$(125):? "TOO MANY 155s A
       ND/OR 34s":END
```

©

Commodore File Protector

John Dearinger

You won't have to worry about accidentally erasing important files off your disks with "Commodore File Protector." It lets you protect individual files or entire disks. The program works on any Commodore 64, VIC-20 with at least 3K RAM expansion, Plus/4, or 16 with a 1541 or 1541-compatible disk drive.

Have you ever scratched a program on a disk and then realized you just deleted the wrong one? Perhaps it was a mental error, or maybe you used a filename with a wild card (* or ?) and got rid of more than you bargained for.

Some computers, such as Apple and Atari, allow you to lock and unlock disk files, offering some protection. Commodore computers, however, don't have any such commands. Neither does the Commodore 1541 disk drive. Yet, interestingly enough, the 1541 does have the routine built into its Disk Operating System (DOS). In fact, the disk drive actually uses the routine to check for a locked file during a write operation.

Here's what happens. Whenever the 1541 starts to scratch a file, it first must find the file on the disk to make sure it exists. Once it is found, the disk drive knows several things about the file, because this information is stored with the filename in the directory on track 18. It knows the track and sector where the first block of the file is stored on the disk. It knows how long the file is and the file type (PRG, SEQ, etc.) by reading the byte stored in the first location of each file entry. The first byte normally is a number from 128 to 132 decimal. (See the charts on pages 56 and 57 of the *1541 User Guide.*) Another DOS routine also checks this location to tell if a file is locked or not. If bit 6 is set, DOS knows the file is locked and won't modify it in any way. For example, if the 1541 finds the number 194 decimal instead of 130 decimal, it knows that a PRG file is locked.

"Commodore File Protector" uses the direct access disk commands to lock the files on a disk so they cannot be deleted—until, of course, they've been unlocked.

Disk Command Menu

To make File Protector compatible with your computer, only one line must be added to the program listing. If you have a Commodore 64, add this line:

```
2Ø F1=4:F7=3:POKE53281,12:POKE5328Ø,6
```

If you have a VIC-20, add this line:

```
2Ø F1=39:F7=63:NS=4:U$=LEFT$(U$,23)
```

If you have a Plus/4 or 16, add this line:

```
2Ø FORA=1TO8:KEYA,"":NEXT:KB=239:SF=1347:
   KL=198:F1=4:F7=3
```

Once File Protector is running, you'll have several options on a menu. First, you can view a directory. This option is offered within several of the routines as well.

You have the option to lock all the files on a disk at once. This will save you a great deal of typing and time when you first use the program on a disk.

You can choose to lock or unlock one specific file at a time, in case you later want to scratch a file or modify a file and replace the old version.

When a file is locked, a less-than sign appears to the right of the file type whenever you list the directory—whether you LOAD "$",8, use DOS 5.1, or choose option 1 on the File Protector menu.

Scratching a file is another option on the menu, and the only one that allows wild cards (* or ?). All the other options require you to enter the exact filename. Some interesting possibilities arise from this. For example, by locking some files and not others, you could clear a disk of unwanted files with many different names (and save a lot of typing) just by specifying an asterisk (*) for a filename to delete.

The last option on the File Protector menu allows you to lock or unlock the entire disk itself. It's best to use this command only on full or completed disks, though, because once the disk is locked, it cannot be written on again until it is unlocked. Locked files on a disk don't prevent the rest of the disk from being used.

Not Totally Foolproof

There are three normal ways to remove files from a disk:

1. Scratch the file
2. Clear the directory with **OPEN15,8,15,"N0:filename"**
3. Reformat the disk with **OPEN15,8,15,"N0:filename, ID#"**

Files locked with Commodore File Protector will withstand number 1 but not numbers 2 and 3. A locked disk will withstand numbers 1 and 2 but not number 3.

One final word of warning: I strongly recommend that you do *not* use this program on any commercial software. These programs often use parts of track 18 in their copy protection, and since File Protector alters that track, it may change something that shouldn't have been changed.

No More Sticky Tabs

The method for locking the entire disk is similar to that for locking a file. In track 18, sector 0, the Block Allocation Map (BAM) is stored. The first two locations (bytes 0 and 1) tell the 1541 where it can find the first directory block containing the first eight filenames on the disk. The third location (byte 2) denotes on which drive this particular disk was formatted. This location should contain the hexadecimal number $41 (65 decimal), which indicates 1541 and 4040 format. If it doesn't, the 1541 will assume that the disk was formatted on a different disk drive and will read the disk, but refuse to write on it.

So, by writing a different number at this location, the disk can be effectively write-protected. No more of those sticky little tabs that are always coming off anyway.

The program also changes location 166 from a $41 (65 decimal) to a $42 (66 decimal). This has no effect on whether the disk is write-protected or not, but is done only to visually indicate a locked disk. The directory header will read 0 "Diskname" ID 2B—note the 2B instead of the normal 2A.

Commodore File Protector

Please refer to "COMPUTE!'s Guide To Typing In Programs" before entering this listing.

```
5 PRINT"{CLR}":Q$=CHR$(13)                 :rem 59
10 U$="{RVS}{40 SPACES}":KB=198:SF=653:KL
   =203:NS=12                              :rem 223
50 GOSUB1240:GOTO1080                      :rem 232
60 PRINT#15,"U1";2;0;T;S:GOTO670           :rem 119
70 PRINT#15,"B-P 2 0":PRINT#15,"U2";2;0;T
   ;S:GOTO670                              :rem 90
80 T=PEEK(681):S=PEEK(682):RETURN          :rem 8
90 REM:::LOCK ALL FILES:::                 :rem 30
100 PRINT"{CLR}"SPC(NS)"LOCK ALL FILES":P
    RINTU$:GOSUB660:T=18:S=1               :rem 75
110 P=2:GOSUB60                            :rem 113
120 GET#2,A$:T1=ASC(A$+CHR$(0)):GET#2,A$:
    S1=ASC(A$+CHR$(0))                     :rem 196
130 FORI=0TO7:PRINT#15,"B-P";2;P+32*I
                                           :rem 133
140 GET#2,A$:A=ASC(A$+CHR$(0)):IFA=.THEN1
    70                                     :rem 137
150 IFAAND64THEN170                        :rem 106
160 PRINT#15,"B-P";2;P+32*I:PRINT#2,CHR$(
    AOR64);                                :rem 248
170 NEXT:GOSUB70:IFETHENRETURN             :rem 21
180 PRINT"TRACK"T"SECTOR"S"IS LOCKED":T=T
    1:S=S1:IFTTHEN110                      :rem 168
190 GOSUB730:GOSUB740:RETURN               :rem 35
200 REM:::READ DIRECTORY:::                :rem 163
210 H$=CHR$(18):PRINT"{CLR}{RVS}HOLD [SHI
    FT] TO PAUSE{DOWN}                     :rem 20
220 GOSUB660:PRINT#15,"M-R"CHR$(144)CHR$(
    7)CHR$(23)                             :rem 103
230 FORI=0TO22:GET#15,A$:H$=H$+(A$+CHR$(0
    )):NEXT:PRINTH$"{BLK}":POKE140,1:T=18
    :S=1                                   :rem 84
240 GOSUB60:SYS828:GOSUB80:IFPEEK(SF)THEN
    WAITSF,1,1                             :rem 8
250 IFTTHEN240                             :rem 63
260 PRINT#15,"M-R"CHR$(250)CHR$(2):GET#15
    ,LO$:PRINT#15,"M-R"CHR$(252)CHR$(2)
                                           :rem 224
270 GET#15,HI$:PRINTASC(LO$+CHR$(0))+256*
    ASC(HI$+CHR$(0))" BLOCKS FREE:rem 188
280 GOSUB730:GOSUB740:POKE140,0:RETURN
                                           :rem 125
290 REM:::INPUT NAME:::                    :rem 140
300 PRINT"{DOWN}[F1] EXIT{13 SPACES}[F7]
    {SPACE}DIRECTORY                       :rem 166
310 PRINT"{DOWN}FILENAME? "CU$;:POKEKB,0:
    F$=""                                  :rem 100
320 KQ=PEEK(KL):GETA$:IFA$=""ANDKQ=64THEN
    320                                    :rem 131
330 IFKQ=F1ORKQ=F7THENPRINTCHR$(20):RETUR
    N                                      :rem 246
340 IFA$=CHR$(20)ANDF$=""THEN320 :rem 254
350 IFA$=CHR$(13)ANDF$<>""THENPRINTCHR$(2
    0):RETURN                              :rem 4
360 IFA$=CHR$(13)ANDF$=""THEN320   :rem 2
370 IFA$=CHR$(20)THENPRINTCHR$(20)A$CU$;:
    F$=LEFT$(F$,LEN(F$)-1):GOTO320:rem 98
380 PRINTCHR$(20)A$CU$;:F$=F$+A$:GOTO320
                                           :rem 27
390 REM:::LOCK A FILE:::                   :rem 102
400 PRINT"{CLR}";                          :rem 51
405 PRINTSPC(NS)"LOCK A FILE":PRINTU$
                                           :rem 12
410 GOSUB300:IFKQ=F1THENRETURN             :rem 208
420 IFKQ=F7THENGOSUB210:GOTO405            :rem 201
430 POKE679,1:GOSUB780:GOSUB660:T=18:S=1
                                           :rem 167
440 GOSUB60:SYS828:A=PEEK(252):IFATHEN470
                                           :rem 244
450 GOSUB80:IFTTHEN440                     :rem 101
460 PRINT"{DOWN}{RED}FILE NOT FOUND{BLK}
    {DOWN}":GOTO410                        :rem 206
470 IF(AAND64)THENPRINTF$:PRINT" IS ALREA
    DY LOCKED":GOSUB730:GOSUB740:GOTO510
                                           :rem 44
480 P=PEEK(255):GOSUB60:PRINT#15,"B-P";2;
    P:PRINT#2,CHR$(AOR64);                 :rem 243
490 GOSUB70:IFETHEN400                     :rem 85
500 GOSUB730:PRINTF$:PRINT"IS LOCKED":GOS
    UB740                                  :rem 142
510 POKE679,0:GOTO400                      :rem 206
520 REM:::UNLOCK A FILE:::                 :rem 4
530 PRINT"{CLR}";                          :rem 55
535 PRINTSPC(NS)"UNLOCK A FILE":PRINTU$
                                           :rem 179
540 GOSUB300:IFKQ=F1THENRETURN             :rem 212
550 IFKQ=F7THENGOSUB210:GOTO535            :rem 209
560 POKE679,1:GOSUB780:GOSUB660:T=18:S=1
                                           :rem 171
```

```
570 GOSUB60:SYS828:A=PEEK(252):IFATHEN600
                                 :rem 243
580 GOSUB80:IFTTHEN570           :rem 109
590 PRINT"{DOWN}{RED}FILE NOT FOUND{BLK}
    {DOWN}":GOTO540              :rem 214
600 IF(AAND64)=.THENPRINTF$Q$"IS ALREADY
    {SPACE}UNLOCKED":GOSUB730:GOSUB740:GO
    TO640                        :rem 231
610 P=PEEK(255):GOSUB60:PRINT#15,"B-P";2;
    P:PRINT#2,CHR$(AAND135);      :rem 79
620 GOSUB70:IFETHEN530            :rem 84
630 GOSUB730:PRINTF$:PRINT"IS UNLOCKED":G
    OSUB740                       :rem 53
640 POKE679,0:GOTO530            :rem 214
650 REM:::OPEN FILE:::            :rem 45
660 CLOSE2:CLOSE15:OPEN15,8,15,"I0":OPEN2
    ,8,2,"#"                     :rem 254
670 INPUT#15,E,EM$,ET,ES         :rem 146
680 IFE=73ORE=26THENGOSUB730:GOTO710
                                 :rem 248
690 IFETHENPRINT"{RED}{RVS}ERROR:":PRINTE
    ","EM$","ET","ES"{BLK}":GOSUB730:END
                                 :rem 225
700 RETURN                       :rem 119
710 IFE=73THENPRINT"DISK IS LOCKED":GOSUB
    740:RETURN                   :rem 251
720 PRINT"REMOVE WRITE PROTECT TAB":GOSUB
    740:RETURN                    :rem 42
730 CLOSE2:CLOSE15:RETURN        :rem 114
740 PRINT"{RVS}{DOWN}PRESS ANY KEY{DOWN}"
    :POKEKB,0                     :rem 80
750 KQ=PEEK(KL):GETA$:IFA$=""ANDKQ=64THEN
    750                          :rem 145
760 RETURN                       :rem 125
770 REM:::STORE NAME FOR ML:::    :rem 12
780 IFLEN(F$)<16THENF$=F$+CHR$(160):GOTO7
    80                           :rem 209
790 FORI=1TOLEN(F$):POKE683+I,ASC(MID$(F$
    ,I,1)):NEXT:POKE700,0:RETURN :rem 199
800 REM:::LOCK ENTIRE DISK:::    :rem 243
810 PRINT"{CLR}"SPC(NS)"LOCK ENTIRE DISK"
    :PRINTU$                     :rem 116
820 PRINT"{DOWN}INSERT DISK IN DRIVE
    {2 SPACES}[F1] TO ABORT":FORTD=1TO900
    :NEXT:GOSUB740                 :rem 8
830 IFKQ=F1THENRETURN            :rem 137
840 GOSUB660:T=18:S=0:GOSUB60:PRINT#15,"B
    -P 2 2"                      :rem 233
850 PRINT#2,CHR$(66);:PRINT#15,"B-P 2 166
    ":PRINT#2,CHR$(66);:GOSUB70:IFETHEN81
    0                             :rem 42
860 GOSUB730:GOSUB660:GOSUB730    :rem 98
870 PRINT"THE DISK IS NOW WRITE PROTECTED
    ":GOSUB740:RETURN            :rem 176
880 REM:::SCRATCH A FILE:::       :rem 73
890 PRINT"{CLR}";                 :rem 64
895 PRINTSPC(NS)"SCRATCH A FILE":PRINTU$
                                 :rem 248
900 GOSUB300:IFKQ=F1THENRETURN   :rem 212
910 IFKQ=F7THENGOSUB210:GOTO895  :rem 218
920 INPUT"{DOWN}ARE YOU SURE";A$:IFA$<>"Y
    "THEN890                      :rem 73
930 GOSUB660:PRINT#15,"S0:"+F$:INPUT#15,E
    ,EM$,ET,ES:IFE>1THENGOSUB680:GOTO890
                                 :rem 215
940 GOSUB730:IFET=.THENPRINT"{RED}FILE IS
     LOCKED OR NOT ON DISK{BLK}":GOSUB740
    :GOTO890                      :rem 97
950 IFET>1THENPRINTET;EM$:GOSUB740:GOTO89
    0                            :rem 228
960 PRINTF$:PRINT"IS SCRATCHED":GOSUB740:
    GOTO890                       :rem 55
970 REM:::UNLOCK A DISK:::        :rem 24
980 PRINT"{CLR}"SPC(NS)" UNLOCK{2 SPACES}
    DISK":PRINTU$                 :rem 88
990 PRINT"{DOWN}INSERT DISK IN DRIVE
    {3 SPACES}[F1] TO ABORT":FORTD=1TO900
    :NEXT:GOSUB740                :rem 16
1000 IFKQ=F1THENRETURN           :rem 175
1010 GOSUB660:T=18:S=0:GOSUB60    :rem 44
1020 PRINT#15,"M-W";CHR$(1);CHR$(1);CHR$(
     1);CHR$(65);                :rem 135
1030 PRINT#15,"B-P 2 2":PRINT#2,CHR$(65);
     :PRINT#15,"B-P 2 166"       :rem 251
1040 PRINT#2,CHR$(65);:GOSUB70:IFETHEN980
                                 :rem 202
1050 GOSUB60:GOSUB730:PRINT"DISK IS UNLOC
     KED                           :rem 5
1060 GOSUB740:RETURN             :rem 252
1070 REM:::MAIN MENU:::           :rem 98
1080 PRINT"{CLR}{BLK}{DOWN}"SPC(NS-2)"<<F
     ILE PROTECTOR>>"             :rem 31
1090 PRINT"{DOWN} 1) DIRECTORY   :rem 153
1100 PRINT" 2) LOCK ALL FILES     :rem 65
1110 PRINT" 3) LOCK A FILE        :rem 88
1120 PRINT" 4) UNLOCK A FILE     :rem 253
1130 PRINT" 5) LOCK ENTIRE DISK  :rem 237
1140 PRINT" 6) UNLOCK A DISK      :rem 12
1150 PRINT" 7) SCRATCH A FILE     :rem 63
1160 PRINT" 8) QUIT               :rem 27
1170 PRINT"{2 DOWN} ENTER YOUR CHOICE"
                                  :rem 52
1180 GETA$:IFA$=""THEN1180       :rem 183
1190 C=VAL(A$):IFC<1ORC>8THEN1180:rem 176
1200 ONCGOSUB210,100,400,530,810,980,890,
     1220                         :rem 71
1210 GOTO1080                    :rem 198
1220 CLOSE2:CLOSE15:END          :rem 148
1230 REM:::VARIABLES & M/L:::    :rem 141
1240 POKE679,0:POKE140,0:CU$=CHR$(31)+CHR
     $(161)+CHR$(144)            :rem 102
1250 FORQZ=828TO1006:READZQ:POKEQZ,ZQ:NEX
     T:RETURN                     :rem 34
1260 DATA 169,0,141,168,2,133,253,162,2,3
     2,198,255,32                 :rem 46
1270 DATA 228,255,141,169,2,32,228,255,14
     1,170,2,230,253             :rem 192
1280 DATA 230,253,32,228,255,133,252,165,
     253,133,255,32,228           :rem 91
1290 DATA 255,32,228,255,230,253,230,253,
     160,0,32,228,255            :rem 241
1300 DATA 230,253,153,189,2,200,24,192,16
     ,144,242,160,243            :rem 230
1310 DATA 32,228,255,230,253,165,144,240,
     3,141,168,2,200             :rem 178
1320 DATA 208,241,165,252,208,7,173,168,2
     ,208,22,240,191             :rem 194
1330 DATA 165,140,240,3,32,199,3,173,167,
     2,240,3,32                  :rem 191
1340 DATA 175,3,173,168,2,240,171,32,204,
     255,96,160,0                 :rem 42
1350 DATA 185,172,2,240,8,217,189,2,208,7
     ,200,208,243                 :rem 47
1360 DATA 238,168,2,96,169,0,133,252
                                 :rem 245
1370 DATA 96,160,0,185,189,2,240,6,32,210
     ,255,200,208                 :rem 41
1380 DATA 245,165,252,41,64,240,10,169,14
     4,32,210,255,169,60,32,210,255,169
                                 :rem 111
1390 DATA 13,32,210,255,169,144,32,210,25
     5,96                        :rem 167
```

C. Regena

Matching Quiz

This month's column presents a general matching-quiz program that can be adapted to any topic. It contains no graphics or sound, so it should be easy to translate to other computers. Feel free to add your own graphics and sound to enhance your particular quiz.

The sample program is a quiz of terms and their definitions. This particular quiz can be used in a computer literacy class for learning general computer terminology.

First the program prints a definition on the screen followed by 12 possible terms. The user must press the letter corresponding to the term defined. If the answer is correct, the program continues and that definition will not appear again. If the answer is incorrect, the program gives the correct answer and the definition *will* appear again.

The score is kept by keeping track of how many times an answer is attempted. A perfect score in this case would be 12. Each time a definition is shown, the score is incremented.

If you want to use this matching quiz for several different topics, type in and save the program consisting of lines 100 through 710. Now, to build a custom program, start with this basic structure and then add DATA statements starting at line 720. Then save the quiz on a different tape or with a different name on the disk. Different quizzes will simply have different DATA statements. You may also need to change the instructions.

Creating DATA Statements

Notice that each DATA statement contains two items separated by a comma. The first item is the term, and the second item is the corresponding definition. If the definition contains a comma, it must be surrounded by quotation marks. Otherwise, the computer will mistake the characters after the comma for another DATA element.

On a quiz for a different topic, use the same idea—put matching parts in the same DATA statement.

Line 110 DIMensions arrays for the quiz. Since this quiz has 12 definitions and terms, the numbers in the DIM statement are 12. You will need to adjust this for the number of items in your own quiz. Line 120 sets the variable N to 12 for the 12 items in this example program. If you have a different number of items, be sure to change this line.

Lines 130–200 clear the screen and print the instructions. Lines 210–230 READ from the DATA the 12 words (W$) and their corresponding definitions (D$). Within the FOR-NEXT loop, a counter with the variable name A varies from 1 to 12. Line 220 looks for DATA statements and reads in order first a word W$(A), then the definition D$(A). The number A keeps them matched up properly. Make sure when you type your DATA statements that you have matched pairs of items (separated by commas).

Program Setup

Lines 240–270 wait for the user to press ENTER before clearing the screen to start the quiz. Line 280 initializes the score (SC) to zero at the beginning of each quiz.

Lines 290–310 set up a temporary word file array, T$(A), which is the same as the original W$ array. This temporary array is used in choosing the terms for the quiz.

Lines 320–550 perform the quiz for the number of items to be matched, N, or in this case 12. Line 330 increments the score SC for each time a definition is shown.

Line 340 clears the screen. Lines 350–370 randomly choose one of the terms which has not previously been matched correctly. The term chosen is denoted by the number R. Line 380 prints the definition D$(R) corresponding to the term chosen.

Lines 390–420 print all of the terms possible for answers with a letter to indicate the answer. Line 430 sounds a prompting tone. Lines 440–460 accept the user's answer, making sure the key pressed is an acceptable letter of one of the terms, then prints the letter chosen.

Evaluating The Answer

Line 470 tests the user's response with the correct answer stored in R. If the answer is incorrect, lines 480–510 print the correct answer, wait for the user to press ENTER, then branch back to line 330 to increment the score and print the next definition. If the answer is correct, lines 520–540 print the message CORRECT!, set T$(R) equal to the null string so the term cannot be chosen again, and then wait for the user to press ENTER. Line 550 increments P for the loop counter to go to the next problem.

After the quiz is complete and all terms have been correctly matched, line 560 clears the screen. Lines 570–580 print the possible score and the user's score. Lines 590–600 print a message if there is a perfect score.

Lines 610–670 present the option to try the quiz again or to end the program.

Lines 680–710 contain the subroutine to wait for the user to press the ENTER key before continuing the program.

Lines 720–840 in this program contain the data for the quiz. Notice that some of the definitions contain extra spaces. These are used to print the definition on the 28-column screen without splitting words.

Customizing The Quiz

Now to change the topic of the quiz. Decide how many items will need to be matched. Keep in mind how it will look when printed on the 24-row screen. Change the DIMension statement of line 110 and the definition of N in line 120 to reflect the number of items.

Next add the DATA statements starting with line 720. For example, if you want a quiz on BASIC programming commands, a typical DATA statement might be:

720 DATA GOTO,Command to transfer program control

A history quiz might contain:

720 DATA 1492,Columbus discovered America.

An algebra quiz could use:

720 DATA x=2,x+5=5x−3

A states and capitals quiz could use:

720 DATA Providence,Rhode Island

When typing the DATA statements, make sure there are matching pairs. If there are short words, you may put more than one matching pair in a DATA statement—just be sure to use commas to separate each item. With longer phrases, make sure you use spaces to print the phrase properly on the screen without splitting words.

Remember that you can add your own sound effects and graphics for positive reinforcements on correct answers. You may also wish to use graphics and sound as part of the matching process.

If you wish to save typing effort and obtain a copy of this program, send a blank cassette or disk, a stamped, self-addressed mailer, and $3 to:

C. Regena
P.O. Box 1502
Cedar City, UT 84720

Please be sure to specify the title of the program and the type of computer you use.

Matching Quiz For TI

Please refer to "COMPUTE!'s Guide To Typing In Programs" before entering this listing.

```
100 REM  MATCHING QUIZ
110 DIM W$(12),T$(12),D$(12)
120 N=12
130 CALL CLEAR
140 PRINT TAB(9);"CHAPTER 1"
150 PRINT :::"A DEFINITION WILL BE
    GIVEN."
160 PRINT :"CHOOSE THE TERM WHICH"
170 PRINT :"MATCHES THE DEFINITION.
    "
180 PRINT :"PRESS THE LETTER OF THE
    "
190 PRINT :"ANSWER."
200 PRINT :"THERE WILL BE";N;"PROBL
    EMS."
210 FOR A=1 TO N
220 READ W$(A),D$(A)
230 NEXT A
240 PRINT ::"PRESS <ENTER> TO START
    ."
250 CALL KEY(0,K,S)
260 IF K<>13 THEN 250
270 CALL CLEAR
280 SC=0
290 FOR A=1 TO N
300 T$(A)=W$(A)
310 NEXT A
320 FOR P=1 TO N
330 SC=SC+1
340 CALL CLEAR
350 RANDOMIZE
360 R=INT(N*RND)+1
370 IF T$(R)="" THEN 360
380 PRINT D$(R)::
390 FOR A=1 TO N
400 PRINT CHR$(64+A);"  ";W$(A)
410 NEXT A
420 PRINT
430 CALL SOUND(150,1500,2)
440 CALL KEY(0,K,S)
```

```
450 IF (K<65)+(K>64+N)THEN 440
460 PRINT CHR$(K)::
470 IF K-64=R THEN 520
480 PRINT "THE CORRECT ANSWER IS"
490 PRINT CHR$(R+64);"--";W$(R)
500 GOSUB 680
510 GOTO 330
520 PRINT "CORRECT!"
530 T$(R)=""
540 GOSUB 680
550 NEXT P
560 CALL CLEAR
570 PRINT "THERE WERE";N;"DEFINITIO
    NS."
580 PRINT :"YOUR SCORE: ";SC;"ANSWE
    RS":::
590 IF SC<>N THEN 610
600 PRINT "GOOD WORK!":::
610 PRINT "PRESS 1 TO TRY AGAIN"
620 PRINT "{6 SPACES}2 TO END PROGR
    AM"
630 CALL KEY(0,K,S)
640 IF K=49 THEN 270
650 IF K<>50 THEN 630
660 PRINT ::"2  END":::
670 STOP
680 PRINT :"PRESS <ENTER>.";
690 CALL KEY(0,K,S)
700 IF K<>13 THEN 690
710 RETURN
720 DATA DOCUMENTATION,THE BOOKS AN
    D MANUALS THAT  ACCOMPANY A COM
    PUTER-RELATEDPRODUCT
730 DATA SYSTEM,A SET OR ARRANGEMEN
    T OF{5 SPACES}PARTS ACTING TOGE
    THER TO{4 SPACES}PERFORM A FUNC
    TION
740 DATA INFORMATION SYSTEM,"A SYST
    EM THAT TAKES INPUT,  PROCESSES
     IT, AND PRODUCES  INFORMATION
    AS OUTPUT"
750 DATA COMMUNICATION SYSTEM,"A SY
    STEM THAT CONSISTS OF A SENDER,
    A PHYSICAL CHANNEL,  AND A RECE
    IVER"
760 DATA HARDWARE,THE PHYSICAL COMP
    ONENTS{5 SPACES}ASSOCIATED WITH
     A COMPUTER  OR OTHER SYSTEM
770 DATA SOFTWARE,PROGRAMS THAT CON
    TROL THE{3 SPACES}FUNCTIONS OF
    SYSTEMS
780 DATA NETWORK,TWO OR MORE COMMUN
    ICATING{3 SPACES}DEVICES THAT A
    RE CONNECTED  TOGETHER
790 DATA APPLICATION,WHAT IS DONE W
    ITH COMPUTERS
800 DATA CIRCUIT,AN INTERCONNECTED
    SET OF{4 SPACES}COMPONENTS THAT
     PERFORM AN ELECTRONIC FUNCTION
810 DATA BINARY SIGNAL,A COMPUTER C
    IRCUIT THAT IS  REPRESENTED BY
    TWO DIFFERENTLEVELS OF CURRENT
820 DATA DATA,"FACTS, NUMBERS, AND
    SYMBOLS PROCESSED BY A COMPUTER
     TO  PRODUCE INFORMATION"
830 DATA BINARY DIGIT (BIT),A BASIC
     BUILDING BLOCK OR{3 SPACES}UNI
    T OF INFORMATION USED IN COMPUT
    ER SYSTEMS
840 END
```

©

THE BEGINNER'S PAGE

Tom R. Halfhill, Editor

Programs Within Programs

Imagine what your life would be like if every time you had to perform a routine task—such as starting your car or switching on a TV—you had to think really hard about it, almost as if you were learning the task for the first time. Starting a car doesn't seem too difficult, but it does require you to execute a number of smaller tasks in exactly the same sequence each time. You have to find the right key, unlock the door, grasp the handle, pull open the door, climb into the seat, stick the key into the ignition, twist the key, and press the gas pedal.

Yet, unless the car is brand-new or belongs to someone else, you can probably do all of this with your eyes closed, like a blindfolded soldier reassembling his rifle. That's because you've performed the actions so many times that they're carved into your unconscious. You just think *start the car,* and a little "program" takes over.

When you think about it, your brain stores thousands of such tiny programs. They let you perform everyday tasks almost on autopilot. Without them, every routine action would be like a new learning experience. Life might be more interesting, like a young child's, but you'd be a lot less efficient.

Computer programs can benefit from the same sort of efficiency. After all, a program at its most basic level is just a list of instructions telling the computer how to perform some kind of job. That job might be something as simple as adding two numbers or something as complex as modeling the economy of a large nation. Still, even simple jobs can often be broken down into several smaller tasks which are executed repeatedly. So why make the computer do things the hard way? Why not equip your programs with the same kind of subprograms that your brain seems to use to automate routine tasks?

This concept of smaller programs within larger programs is so powerful that virtually every computer language offers some way to do it. By identifying these repetitive tasks and turning them into subprograms or *subroutines*, you can write programs that run faster, consume less memory, and are easier to understand and modify.

When To Use A Subroutine

Your brain acquires a subroutine by rote—it subconsciously memorizes a task that you perform over and over again. Today's computers aren't quite intelligent enough to learn this way, so you have to spell it out for them more literally with BASIC commands.

First you have to decide when to take a piece of a program and make it into a subroutine. This judgment comes naturally after a while, but as a general rule, any small task which is performed more than once in a program is a candidate for a subroutine.

Once you've identified this task, you write the little routine and make the program detour to those lines whenever you need to perform that task. At the end of each subroutine, you use the command RETURN to automatically go back into the main program and proceed with other things.

Let's try an example. Assume you're writing a program that frequently pauses and asks the user to press a key. With no subroutines, this is how clumsy the program would be:

```
90 DIM A$(1):REM This line for Atari only
100 PRINT "During the Civil War,"
110 PRINT "more American soldiers died"
120 PRINT "than in all other"
130 PRINT "American wars combined."
140 PRINT "PRESS C AND RETURN TO
    CONTINUE";
150 INPUT A$
160 IF A$<>"C" THEN GOTO 140
170 PRINT "Poor medical care accounted"
180 PRINT "for many casualties,"
190 PRINT "but outmoded military tactics"
200 PRINT "were also to blame."
210 PRINT "PRESS C AND RETURN TO
    CONTINUE";
220 INPUT A$
230 IF A$<>"C" THEN GOTO 210
...
```

Notice how the lines which ask the user to press a key (lines 140–160 and 210–230) are simply repetitious; only the line number references are different.

In each case these lines keep printing the prompt PRESS C AND RETURN TO CONTINUE until the user presses the C key. (Make sure to press a capital C if you try running this example. If you have a TI-99/4A, change every occurrence of THEN GOTO to THEN in this and all following examples.) A little three-line routine like this one might not seem like much, but if it's repeated throughout a long program, considerable space and programming time would be wasted. This is an ideal candidate for a subroutine.

Why Not GOTO?

At this point, you might be thinking about building a subroutine with the GOTO command. After all, a subroutine requires a detour from the main program, and GOTO is a programming detour (see last month's column). Why not just jump to the subroutine with GOTO and then exit from it the same way? The program might look like this:

```
90 DIM A$(1):REM This line for Atari only
100 PRINT "During the Civil War,"
110 PRINT "more American soldiers died"
120 PRINT "than in all other"
130 PRINT "American wars combined."
140 GOTO 1000
150 PRINT "Poor medical care accounted"
160 PRINT "for many casualties,"
170 PRINT "but outmoded military tactics"
180 PRINT "were also to blame."
190 GOTO 1000
200 PRINT "For instance, many battles"
210 PRINT "were fought with mass charges"
220 PRINT "of infantry and cavalry."
230 GOTO 1000
...
1000 PRINT "PRESS C AND RETURN TO
     CONTINUE";
1010 INPUT A$
1020 IF A$<>"C" THEN GOTO 1000
1030 GOTO 150
```

At first this seems to fit the bill. The lines which await the user's keystroke are grouped together in a neat subroutine at the end of the program. All it takes is a simple instruction—GOTO 1000—to activate (or *call*) the subroutine.

If you try running the program, however, a problem soon becomes apparent. The subroutine works great the first time it's called. The first paragraph of text appears on the screen, followed by the prompt, and the program continues printing when you press C. But after the second time the subroutine is called, the program prints the second paragraph all over again! In fact, it keeps printing the same paragraph no matter how many times you press C—it never reaches the third paragraph at all.

GOTO is the culprit. GOTO 1000 works okay for *calling* the subroutine, because the routine is always at line 1000. But GOTO doesn't work so well when *returning* from the subroutine. The line number in the routine's final GOTO statement is fixed (GOTO 150), but the line number where the program should continue after calling the routine keeps changing. What's needed is a substitute for GOTO that always knows how to pick up where the program left off. That substitute is the pair of commands GOSUB and RETURN.

GOSUB: A GOTO With Brains

If you understood how the above programs work, you'll have no trouble at all grasping GOSUB and RETURN. GOSUB (which means *GOto SUBroutine*) is merely a smarter version of GOTO. The statement GOSUB 1000 does the same thing as GOTO 1000—it detours the program to line 1000. However, it also makes the computer remember *where it detoured from.* Then, when a RETURN statement is encountered, the program automatically returns from the subroutine and begins executing the statement which immediately follows the original GOSUB.

Here's how the previous example would look after GOSUB and RETURN are substituted for the GOTO statements that caused the problem:

```
90 DIM A$(1):REM This line for Atari only
100 PRINT "During the Civil War,"
110 PRINT "more American soldiers died"
120 PRINT "than in all other"
130 PRINT "American wars combined."
140 GOSUB 1000
150 PRINT "Poor medical care accounted"
160 PRINT "for many casualties,"
170 PRINT "but outmoded military tactics"
180 PRINT "were also to blame."
190 GOSUB 1000
200 PRINT "For instance, many battles"
210 PRINT "were fought with mass charges"
220 PRINT "of infantry and cavalry."
230 GOSUB 1000
240 END
...
1000 PRINT "PRESS C AND RETURN TO
     CONTINUE";
1010 INPUT A$
1020 IF A$<>"C" THEN GOTO 1000
1030 RETURN
```

Think how much memory (and programming time) you could save by simply inserting a GOSUB 1000 statement whenever you want the user to press a key to continue, instead of redundantly entering the routine itself each time you need it. The memory savings are even more dramatic with longer subroutines.

For that reason alone, GOSUB and RETURN are worth their weight in RAM chips. Yet memory conservation is only one advantage of using subroutines in your programs. We already mentioned how they can increase execution speed and help make programs easier to understand and modify. But they can also drastically reduce the time you spend writing and debugging a program. Once you get a subroutine up and running without bugs, you can call it with confidence whenever necessary. If an error does result, you can be fairly certain that something outside the subroutine is causing the error. This narrows down your search for the elusive bug.

Subroutines can also make it less intimidating to write large, complex programs. By breaking a big job down into many smaller jobs, and then tackling them one at a time, the program seems to fall together much more easily. In fact, many programmers keep a library of frequently used subroutines and stick them into new programs wherever needed.

Questions Beginners Ask

Q In manuals, books, and articles, I keep seeing the term "default." What does default mean?

A *Default* means the way something starts out, its normal condition. For example, many computer games default to one-player mode. If there are two players, you have to let the game know by pressing a special key.

In computer terminology, default can refer to the standard setting of a switch, the screen colors when you first turn on the computer, the number stored in a memory location before it's altered by a program, and many other things. For example, the LOAD command on a Commodore 64 or VIC-20 defaults to tape instead of disk. If you type:

```
LOAD"PROGRAM NAME"
```

the computer assumes you are loading from the cassette recorder and responds PRESS PLAY ON TAPE. To load a program from the disk drive, you have to add a device number to the command which overrides the default:

```
LOAD"PROGRAM NAME",8
```

Another example is a dot-matrix printer which defaults to a standard typeface. To print in a special typeface such as bold or italics, you must send the printer a command (usually from within a program) which overrides the default setting. ©

INSIGHT: Atari

Bill Wilkinson

Atari Acquires Apple!

As I write this, the Winter Consumer Electronics Show (CES) in Las Vegas has just ended. By now you have probably read in the papers and magazines just what real marvels the new Atari Corporation introduced at CES. While I didn't get a chance to attend CES (though others from my company were there), I did have the privilege of getting some preshow information about Atari's new products. Also, thanks to being just a bit nosey, I learned a little about how Atari developed their remarkable new computers and even a little bit of what's yet to come.

Purchase Obvious In Retrospect

(An important aside: The issue of COMPUTE! which will carry this article is dated April 1985. However, since this issue will most likely appear on newsstands and in subscribers' mail by about mid-March, you might be reading this before April. If so, be sure to keep all of what I am about to reveal secret until at least the first of April.)

Reveals Other Buys

Anyway, as I started to say, I was lucky enough to be privy to some early information and (thanks to my nosey nature) overhear even more. One thing I overheard was a simple question, "Should we take the Mac with us?" (An obvious reference to an Apple Macintosh.) It seems that in the process of designing the 130ST and 520ST computers, the engineers at Atari looked at several existing computers. Now, no rival companies were about to be so generous as to donate machines. So, looking back, it seems obvious that Atari had to go out and buy several—including the Mac, of course.

IBM Failure Described

In the process of evaluating the various computers, Atari also was able to look at the microprocessors (CPUs) which they used. It comes as no surprise that the 8/16 bit 8088 used by the IBM PC was rejected early on as being unable to achieve the speed Atari desired. So what processor got the nod for the 130ST and 520ST?

Leonard Tramiel Departs Company

Although I have managed to enjoy Leonard Tramiel's company in several meetings, the one time we managed to get in a really interesting discussion of processors he had to depart early (for another meeting, probably). Before he left, he did seem to indicate that his personal choice for a CPU might be the National Semiconductor 32016 and 32032 processors. They are very powerful and very orthogonal machines, but (and this is speculation on my part) the fact that they are available only from National Semi probably makes choosing them difficult for any company.

In any case, Atari chose to go with the tried and true Motorola 68000 series of processors, the same one used in the Apple Macintosh and Lisa computers. (An aside: The official meaning of the *ST* designation is "Sixteen/Thirty-two" for the 16-bit bus and 32-bit registers of the 68000 chip. *XE* implies XL compatibility, but *E*xtended.)

Future Plans Fall Flat

What about all the loyal Atari 400/800/1200XL/600XL/800XL owners? Has Atari completely forgotten them? *No way!* Apple has Mac and Lisa, both built around a 68000 chip, in its "sort of 32-bit" division, and the IIe and IIc, both using a 650x CPU, in its 8-bit division.

Lo and behold! We already saw that Atari has the 130ST and 520ST built around the 68000. Does it really surprise you to learn that the 65XE and 130XE will be produced using a 650x processor? And we were even given the privilege of having a set of drawings for a portable computer (in the 650x line) dropped flat on the table in front of us!

Original Projections Unrealized

The same day we saw those plans for the portable, we also got to see some of the features that the new machines will be sporting. On that day I decided that my predictions of success for Atari, which I made in this column in December, could very well have been ridiculous underestimates.

Operations Shut Down

What kind of features impressed me? I think it will be obvious to you when you read a spec sheet at your local dealer or the other CES coverage in this issue. In the meantime, I'll give a brief list of what I think are the best features of each machine at the end of this column. I tried to ask some of my contacts at Atari about a couple of things I am not quite clear on, but the lure of CES left the software and engineering departments virtually shut down for these four days.

Long-Term Outlook Bright

If there is any area of concern to those of us here at Optimized Systems Software, it is about those products where our software sales overlap those of Atari Corporation. New prices on Atari software have made us rethink some of our plans, but we think that there will always be sophisticated and/or advanced users out there who will be willing to pay a little more for higher quality. And we are not alone: The number of companies showing Atari-compatible software or hardware at CES was almost amazing. Will we stay in the Atari software market? How could we not?

At Last

"What the heck," you ask, "was all that about?" The answer: Every word that you just read was true. Even the subheadlines are properly explained in the text. Oh, I may have bent some words here and there to make the headlines more spectacular, but that was the whole purpose of this exercise. I always wanted to show how you can take an innocuous and/or positive review and generate sensational *National Enquirer*-type headlines.

If you're an acrostics fan, you may have already caught the significance of the first letter of each headline. (Go back and reread them if you want a minor laugh.) This is, of course, my annual attempt at some humor. It's not very subtle or well-hidden this year, because I thought it would be fun to find out how many COMPUTE! readers actually plow through all my verbiage. If you got to here unscathed, congratulations. Time for a complete change of pace.

New Machine Features

This is just a simple table of what I feel are the most important features of four of the new Atari machines. I am sure that more info will be available by the time you read this, but maybe these specs will whet your appetite.

65XE

- **6502-series processor.**
- **64K of RAM.**
- **Very, very compatible with 800XL.**
- **Nicely sculptured case and keyboard.**
- **Cartridge port on rear (where our ugly orange cartridges won't be so obtrusive).**
- **About $100.**

130XE

- **Identical to 65XE *plus*:**
- **128K of RAM (supported as a ramdisk by new DOS 2.5).**
- **Expansion port on rear (used in conjunction with cartridge slot).**
- **About $150.**

130ST

- **68000-series processor.**
- **128K of RAM.**
- **192K of ROM.**
- **Uses Digital Research's GEM windowing and display system—virtually identical in form and function to Apple's Macintosh system.**
- **Built-in RS-232 interface.**
- **Built-in parallel printer interface.**
- **Built-in disk controller handles up to four floppy disk drives (designed to use *very* inexpensive 3.5-inch drives, 360K each—priced perhaps as low as $100!).**
- **DMA-capable expansion port (designed for *very* fast hard disk drives).**
- **Three-voice sound chip.**
- **Color graphics (640 × 400 in black and white, 640 × 200 with four colors, 320 × 200 with 16 colors).**
- **Cartridge slot (up to 128K ROM in cartridge).**
- **10 special function keys.**
- **MIDI interface (for music synthesizers and ???).**
- **About $400.**

520ST

- **All the features of the 130ST *plus*:**
- **512K of RAM instead of 128K.**
- **About $600 (Yep . . . that gives you a color "Fat Mac" at around $1,000).**

Information Please

It's time, once again, to respond to some letters. I may have made a mistake in publishing the P.O. box where you can write me directly, since I find myself with about five or six times as much mail to answer as I had before. Until I get adjusted to answering this much correspondence, please bear with me.

For this month, I have decided to select some letters which (I think) really *need* answers. Surprisingly, for such varied topics, the answers to all may be much the same.

Bob Dorn, of College Park, Georgia, was the first of three or four to ask me how to use an Atari 1030 direct-connect modem to upload and download files. Well, you got caught in the great Atari let's-protect-the-poor-dumb-user game. For reasons best understood only by now-extinct marketing people at the old Atari, neither the 835 or 1030 modem came with software support for uploading and downloading programs, text files, and so on. I guess those marketers never used a computer with a modem, so they couldn't see any use for the capabilities.

Luckily, many other people, including a few software gurus, found themselves in the same fix you are in. One commercial company which seems to be doing a lot of work with these modems is Gardner Computing, P.O. Box 388, Holbrook, NY 11741. I am *not* endorsing them (I have never used any of their products—I have only read their ads), and I apologize in advance for inadvertently slighting any other companies supplying similar software.

There are other solutions. See the "Readers' Feedback" letter headlined "Atari Modem Update" in the February 1985 issue of COMPUTE!. There are also some programs floating around in public domain user group libraries which allow upload/download and more. As a general rule, such programs come without documentation (or, at most, with a few paragraphs on the disk with the program), so you may need to do a little detective work to use them.

Good Local Support

Again, though, there may be another solution. *Join your local user group*. Come on now, what will it cost you? One evening and a couple of dollars a month will probably be the best investment you ever made in computing. And so many user groups have people who know the answers. To almost anything you ask!

Another practical reason for joining such a group is that Atari has already announced that its primary means of providing programming support to users will be through the user group network. The toll-free phone lines are gone, and the support group is decimated. This may be the *only* way to get technical answers in the future (aside from writing to me or "Readers' Feedback").

All of this, and we haven't even mentioned the fact that most user groups have literally *hundreds* of programs available for next to nothing. Okay, okay. Some of the programs don't work right, are poorly written, are too slow, etc. So what? You are getting what you paid for and more. If nothing else, a cruddy little Atari BASIC subroutine may lead your computer to uses you hadn't thought of yet.

So join, join, join. Why wait five months for my answer to appear in this magazine when help is available two miles from your home?

How do you know where/who/when/what your local group is? Well, try asking at local computer stores, even those that don't sell Atari products. Look in your local paper. Look in Atari-oriented magazines, which sometimes have listings of clubs. If you are really desperate, send me a *self-addressed and stamped* card or envelope. No guarantees, because I don't know where *all* the clubs are, but if there's one on my list I will tell you. *Please* use me only if all else fails, because (1) I'm always too busy, (2) it may take me some time to answer, and (3) if I ask my kids to help me with this, they will charge me.

Deluged With Information

From going to users who can't find what they need, we go to a couple of readers who have found too much. Jamie Patterson, of Hooker, Oklahoma, sent me a well-argued plea for some help in choosing material about his three-month-old baby, an 800XL computer. I quote: "How does a three-month-old know which books to choose?"

Darned good question. My usual answer, when I want to choose a new computer book, is to go to two or three bookstores that carry a couple of hundred computer books each and browse. This works because there are at least a dozen such bookstores within reasonable distance of my house. Now, I have to admit I don't know where Hooker, Oklahoma, is, but if it isn't within 20 miles of a major computer bookstore, my method won't work for Jamie. What can he do?

The editors of COMPUTE! might like me to answer, "Buy a COMPUTE! book." But whatever book you buy, you must choose one which is at the right level for you. From COMPUTE! Books, the most general material may be found in the *First, Second,* and *Third Book of Atari*, along with the two books on *Atari Graphics*. Some, but not all, of this material is relevant to someone who has learned the fundamentals of Atari BASIC.

Suppose, though, that you aren't even to that level yet. You don't know a PRINT from a PLOT statement. Where do you turn? Since Atari stopped shipping copies of *Inside Atari BASIC* with the XL computers, buyers have been left to choose their own tutorial. And what should they choose?

My trouble is that every time I look at a book that purports to teach BASIC (or word processing or assembly language or . . .), I find something wrong. I don't like the order of presentation of the topics. There are mistakes in the section on how to speed up your programs. The author encourages poor programming style. The list goes on and on. So I refuse to make a firm recommendation.

The Great Book Survey

What, then, can Jamie Patterson and others like him do? What else? Join a user group. Ask other Atari owners. Ask to look at their books. Okay, so maybe none of the over-200 user groups is close enough to Jamie. And, besides, he asked *me* for an answer. I guess I should do something, right?

So here it comes. I am asking you, my readers, to make some comments on the books *you* have learned from. Don't stick to learning BASIC. Any aspect of Atari computers is eligible, even manufacturers' manuals. To make life easier for me, just send the title(s) of the book(s), the level (1 to 10, with 1 being rank beginner), and your overall rating (0 for trash to 10 for perfection). A postcard will do fine.

I don't want any experts evaluating these books; I can mishandle that aspect myself. Instead, I want actual real-life experiences. Did or did not the book teach you what it said it would? If it did, was it an uphill battle or did the style make it downright easy for you? I can't respond personally to these rating cards, but I will report the results received by April 20 in the August or September issue (sorry, but that's the fastest turnaround possible).

Translators, Again

Robert Glover, of Cleveland, Tennessee, has been the proud owner of an Atari 400, an 800, and now an 800XL. He asks me why he can't simply use the binary save option of Atari DOS to make a copy of the 800's operating system ROMs and then load that file into his 800XL as a home-brew translator disk. He suggests that I perform this service in my column.

Well, in theory, and with some modifications to his method, I *might* be able to do so. Why won't I? First, there are several problems to overcome. Two of the simpler examples: (1) You can't write/save ROM directly with DOS 2.0S; you have to copy it down to RAM first. (2) Joystick ports 3 and 4 are used for *output* in an 800XL and for *input* in an 800.

Also, how many readers have access to both an 800 and 800XL? And, finally, why go to that kind of trouble when the translator disks are so available?

Ah, but that last point was raised by Mr. Glover. He says he cannot find the translator disks anywhere. Hmmmm. Guess where I am going to suggest he look? Right. Ask your local user group. And that brings us back to the quandary of the last reader: What if there is no user group nearby?

I have a couple of partial solutions. First, there are a few mail-order organizations which, in addition to selling commercial software, sell public domain programs for reasonably low prices. Right now, LotsaBytes (15445 Ventura Blvd., Suite 10, Sherman Oaks, CA 91413) seems to be the leader in this category, but I should also mention DynaComp, *Antic*, and *ANALOG* (the latter two offer primarily games and BASIC utilities from their magazines).

Perhaps even better, many user groups (especially the larger ones) allow mail-order memberships. Since there are so many of these groups just crying for members, I hesitate to recommend one over another. But because their newsletter has been around the longest and may have the greatest number of readers, I will at least mention the very friendly people of ACE (3662 Vine Maple Dr., Eugene, OR 97405).

So my message this month is clear: Atari is very, very, very much alive and well. Keep your interest in your machine similarly healthy by joining a user group. ©

IBM BASIC's Undocumented SHELL Command

Michael A. Covington

With DOS 3.0, IBM has announced a number of new features for disk BASIC. At least one of them is actually present in DOS 2.0 and 2.1 as well, though the manuals do not mention it. That feature is a command called SHELL that allows you to execute DOS commands from within BASIC. (The technique does not work with PCjr Cartridge BASIC.)

The SHELL command in IBM BASIC takes one parameter, a character string containing the DOS command to be executed. SHELL works by loading, from drive A, a second copy of COMMAND.COM (the DOS command processor) and invoking it as a subprocess. (Note that this implies that COMMAND.COM must be present on the disk in drive A when the SHELL command is executed.) The top level COMMAND.COM and the BASIC interpreter are in suspended animation until the subprocess finishes; then control returns to BASIC.

SHELL handles the cursor somewhat awkwardly. When the SHELL command is executed, the screen is cleared from the current cursor position to the bottom; DOS writes its output there, scrolling as needed (the twenty-fifth line scrolls along with the others). But when control returns to BASIC, the cursor suddenly appears one line below where it was when the subprocess started, ignoring all screen activity that took place under the subprocess.

The best way to prevent chaos on the screen is to execute a CLS (clear-screen) immediately after each SHELL, or as soon afterward as you're done looking at the output.

Not A Child

The one command that SHELL cannot issue, either directly or indirectly, is BASIC (or BASICA). If you try to do this, you get the message "You cannot run Basic as a Child of Basic"—naturally enough, you can't run BASIC in the subprocess because most of BASIC is in ROM and there's only one copy of it in the machine. If you issue a SHELL and COMMAND.COM is not on drive A, you get a "File not found" error within BASIC.

The most useful SHELL commands are probably:

```
SHELL "A:"
SHELL "B:"
```

and the like, to change logged disks. These are foolproof commands; they produce no messages to clutter up the screen, and they can't terminate abnormally.

You can also use SHELL without parameters, in immediate mode, to enter the DOS command mode. The advantage of this over SYSTEM is that when you're done issuing DOS commands, you can type EXIT and return to BASIC with your program undisturbed.

Most kinds of errors in the subprocess will return you to BASIC with no problem, but a few, such as typing A in response to "Abort, Retry, Ignore," will leave you in the DOS command level of the subprocess, in which case you must type EXIT to get back to BASIC.

One At A Time

Don't issue several SHELL commands in succession if you can avoid it; each of them loads COMMAND.COM all over again. Instead, if you have a series of commands to issue, write them onto a .BAT file from within BASIC, and give one command to run the whole file.

The accompanying program demonstrates

one way to use SHELL to create a menu-driven user interface for DOS. Naturally, a practical program would include many more options and more error-checking.

Purpose:	Executes a DOS command from within BASIC. This is done by loading a second copy of COMMAND.COM and invoking it as a subprocess.
Versions:	Cassette no · Disk yes · Advanced yes · Compiler ?
Format:	SHELL or SHELL X$
Remarks:	X$ is a character string constant, variable, or expression containing any valid DOS command.
	In order for SHELL to work, COMMAND.COM must be present on disk A. If it is not, the message "File not found" is displayed.
	X$ can be an internal DOS command or invoke a .COM, .EXE, or .BAT file. However, the BASIC interpreter cannot be invoked using SHELL; if this is attempted, the message "You cannot run Basic as a Child of Basic" is displayed.
	The amount of memory available in the subprocess is markedly less than is available in DOS by itself.
	If X$ is omitted, the user is placed at the DOS command level of the subprocess. To return to the calling BASIC program, type the command EXIT.
	Certain fatal errors in the subprocess may also leave the user at the DOS command level of the subprocess; again, typing EXIT returns control to BASIC. However, most errors in the subprocess return control to the calling BASIC program automatically.
Examples:	SHELL (to go temporarily into command mode)
	SHELL "B:" (to change logged disk)
	SHELL "DIR A: : SORT : MORE"
	SHELL "MYFIL" (to invoke MYFIL.COM, MYFIL.BAT, or MYFIL.EXE, as the case may be)

Demo of SHELL Command

```
GI 1Ø ' COMMAND.COM must be on drive A
IA 2Ø ' MORE.COM and CHKDSK.COM must be
ME 3Ø '   on the current default disk
KN 4Ø CLS: KEY OFF
BD 5Ø PRINT "Welcome to menu-driven DOS."
FI 6Ø PRINT
LO 7Ø PRINT "Available functions are:"
NG 8Ø PRINT "  1  Directory of disk A"
AG 9Ø PRINT "  2  Directory of disk B"
QJ 1ØØ PRINT "  3  Disk and memory inform
   ation"
BD 11Ø PRINT "  4  Copy a file"
AB 12Ø PRINT "  5  View a file"
JK 13Ø PRINT "  6  End this program"
JI 14Ø PRINT
KA 15Ø INPUT "Choose one...";N
BL 16Ø IF N=6 THEN CLS: END
OF 17Ø IF (N<1) OR (N>5) THEN BEEP: GOTO
   15Ø
BN 18Ø CLS
KA 19Ø ON N GOTO 21Ø,24Ø,27Ø,32Ø,37Ø
MD 2ØØ ' directory of A
AH 21Ø SHELL "dir a:"
CP 22Ø GOTO 4ØØ
ON 23Ø ' directory of B
CN 24Ø SHELL "dir b:"
CF 25Ø GOTO 4ØØ
OG 26Ø ' disk & memory info.
CG 27Ø INPUT "Drive to check ";A$
BI 28Ø IF A$="a"OR A$="A" THEN SHELL "chk
   dsk a:"
FG 29Ø IF A$="b"OR A$="B" THEN SHELL "chk
   dsk b:"
CM 3ØØ GOTO 4ØØ
ML 31Ø ' copy a file
MC 32Ø INPUT "File to copy from ";A$
CM 33Ø INPUT "File to copy onto ";B$
EO 34Ø SHELL "copy "+A$+" "+B$
CG 35Ø GOTO 4ØØ
KD 36Ø ' view a file
OA 37Ø INPUT "Name of file ";A$
CG 38Ø SHELL "more <"+A$
LJ 39Ø ' finish up
DN 4ØØ LOCATE 25,1
MD 41Ø WHILE INKEY$<>"": WEND
OB 42Ø PRINT "(Press any key to continue.
   ..)";
LA 43Ø WHILE INKEY$="": WEND
OF 44Ø GOTO 4Ø
```

©

Apple SuperFont

Custom Character Set Graphics For The Apple

Tim Victor, Editorial Programmer

Here's a significant enhancement for graphics on Apple II-family computers. With "Apple SuperFont," you can now place upper- and lowercase text anywhere on the high-resolution screen. In addition, you're not limited to the built-in character set, either—you can easily define foreign character sets, italics, boldface, and underline fonts, as well as shapes for high-speed animated games in BASIC. Apple SuperFont is an all-new, original version of the SuperFont series of programs published by COMPUTE! *for Atari, Commodore 64, and TI computers and adds several new features especially for the Apple. It requires a 48K or 64K Apple II+, Apple IIe, or Apple IIc, with either DOS 3.3 or ProDOS.*

Without resorting to machine language, programming high-speed graphics is difficult on the Apple. High-resolution graphics look nice, but shape tables are too slow for most animation purposes. One alternative is to use character graphics for animation. Characters can move a whole block (character position) at a time, and can be placed on the screen with a simple PRINT statement. Unfortunately, ordinary Apple characters aren't very suitable for games or even business charts.

But now there's a way around these problems. With "Apple SuperFont" and its accompanying utility programs, you can easily redefine a character into practically any shape you want and print it directly on the hi-res graphics screen. Custom character sets are a snap to design, and fast animation is as simple as printing a character, erasing it, and printing it again in a new location.

Several programs already exist for printing characters on the hi-res screen, including HRCG (High Resolution Character Generator), which is part of the Apple DOS Toolkit. The Apple SuperFont HROUT program works much like HRCG, putting characters on the high-resolution screen from a table of character images, but the Apple SuperFont system is much more versatile.

The Apple SuperFont Editor makes it easy for you to create character sets (fonts) for use with HRCG or HROUT. Special features help you design multicharacter shapes and allow you to see the effects of the Apple's unusual use of color in hi-res graphics. Once you've created or customized a character set, you can easily use these fonts in your own programs.

Typing Apple SuperFont

To run SuperFont, you need to have four files on the same disk: APPLEFONT, APPLEFONT2, HROUT, and NORMAL.SET. There are two different versions of APPLEFONT. Program 1 is for using SuperFont with DOS 3.3. Program 2 shows the changes necessary to use Program 1 with ProDOS. The other three files need no changes to be used with either disk operating system.

APPLEFONT2 (Program 3, the Apple SuperFont Editor), NORMAL.SET (Program 4), and HROUT (Program 5) are all machine language binary files and must be entered with the Apple's built-in machine language editor (monitor). It's easy; you don't need to understand machine language to use these programs.

Here's how to type them in. To enter the monitor, type CALL −151. The Applesoft prompt (normally a]) will be replaced by the monitor's prompt, an asterisk (*). To enter a line from the listing, first type in the four-digit hexadecimal number, then type a colon (:) instead of the hyphen shown in the listing produced by the monitor. This is the address where you'll enter the rest of the line. Type in the rest of the line after the colon, leaving a space between each two-digit number. After eight numbers, press RETURN and enter the address for the next line.

Again, use a colon instead of the hyphen shown in the program listing. If you want to review what you've entered to check for accuracy, you can list a block of data by typing the address of the first location in the range, then a period, then the last address, and pressing RETURN.

Once you've entered one of the machine language programs, save it to disk using the BSAVE command. This command can be used either from BASIC or from the monitor (you can exit the monitor and return to BASIC by pressing CTRL-C, then RETURN). To BSAVE Program 3 (APPLEFONT2), the command is:

BSAVE APPLEFONT2,A$1000,L$FE0

Save Program 4 (NORMAL.SET) by entering:

BSAVE NORMAL.SET,A$8D00,L$300

Save Program 5 (HROUT) by entering:

BSAVE HROUT,A$300,L$58

Because of the length of APPLEFONT2, typing mistakes could be difficult to find. As a check, BLOAD APPLEFONT2 and enter the following line, then hit RETURN:

S=0 : FOR I=4096 TO 8159 : S=S+PEEK(I) : NEXT : PRINT S

If the result of this calculation is not 365090, there is at least one error in your copy of APPLEFONT2. To help locate errors, we've included a small checksum program (Program 6). To use it, BLOAD APPLEFONT2, then run Program 6. If you have mistyped some data, it will tell you where to look to find the mistake.

When all the files are entered and saved to disk, type RUN APPLEFONT. APPLEFONT first checks to see which operating system is in your Apple. If the correct operating system for this version of APPLEFONT is present, it will BLOAD the other three files, and connect HROUT to the standard character output routine. APPLEFONT2, the SuperFont Editor, is started with a CALL to 4096. From then on, the SuperFont Editor is in complete control except when it needs to access the disk drive. If you ask to load or save a character set, control returns to the BASIC program, the file is transferred using BASIC's disk access commands, and the SuperFont Editor program is CALLed again.

Using The SuperFont Editor

Characters are designed and edited on a grid that represents 32 (vertical) × 55 (horizontal) pixels. Each cell in the grid is a fourfold enlargement of actual size. Individual cells can be turned on (white) or off (black) with the bit-editing functions, and blocks of cells can be copied from one place to another on the screen. Patterns of 7 × 8 cells can be saved from the screen to the character set being edited with the Put command. The

The main editing screen of "Apple SuperFont," showing the design grid, an option menu, and the Apple's built-in character set (NORMAL.SET).

Get command does just the reverse, pulling a character from the character set onto the editing screen.

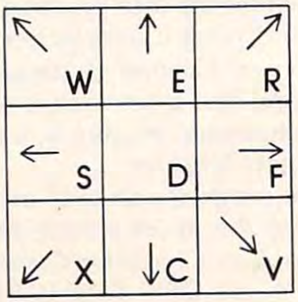
All of the features of the Editor are controlled with a series of four menus, entitled Bit Edit, Charsets, Utility, and Display. Each of these menus contains three to six selections. Only one menu is displayed on the screen at a time.

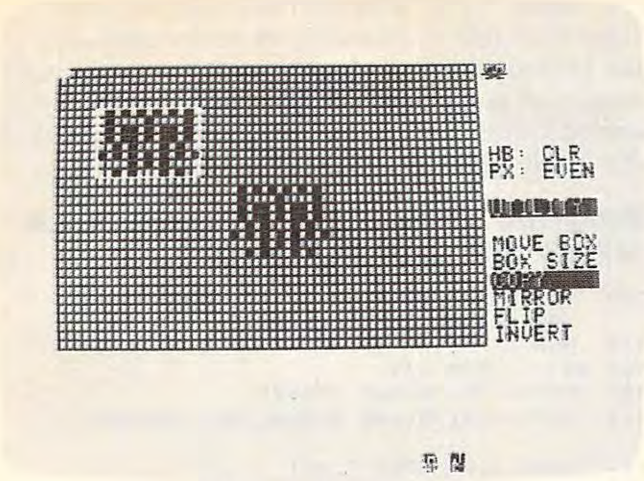
To change menus, press the space bar. The next menu title will be printed on the screen, along with its menu selections. The top selection will be printed in inverse characters to indicate that it has been chosen. To select a different menu item, use the left- and right-arrow keys. The large cursor bar moves up or down the menu to show you which selection is active.

Some menu items, like Clear Screen or Save Set, wait for you to press the RETURN key before performing their functions.

Three Cursors

You will be using three visually distinctive cursors in the SuperFont Editor: the bit cursor, the box cursor, and the character cursor. When a menu item is selected, one of the cursors may begin to flash, indicating that it can be moved. The cursors are controlled by a keypad centered on the D key:





Using the Copy command, you can duplicate shapes on the editing grid quickly and easily, as demonstrated with this Space Invaders*-type character.*

The bit cursor is a 1 × 1 cell box displayed on the editing screen. It flashes whenever the Bit Edit menu is displayed. Moving the bit cursor around on the editing screen sets (white) or clears (black) the cells that the cursor passes over. In other words, the bit cursor leaves a trail of black or white behind it. Selecting Black or White changes the color drawn when the bit cursor is moved. If you want to move the bit cursor without drawing on the screen, select the Move option.

The box cursor is a box displayed on the editing screen, but its size can be changed. It can be as small as a 1 × 1 cell, or as large as the entire editing screen. When you're using a utility such as Copy or Flip, the box cursor outlines the area on which the utility will operate. These utilities can be used on a character, part of a character, on shapes made up of several characters, or on a portion of a character, simply by changing the size of the box. Pressing the RETURN key when Flip is selected turns the contents of the box cursor upside down, and the Mirror function reverses left and right sides of the box. The Invert function changes all of the white cells inside the box to black cells, and all black cells to white. When Copy is selected, the cursor pad controls a second box cursor, which initially appears on top of the original box. Pressing the RETURN key copies the contents of the original box to the second box.

You can also use the box cursor to select the 7 × 8 cell character pattern for the Put and Get functions. The character cursor, located in the character set displayed at the bottom of the screen, flashes when the Get or Put function is selected. Use it to select the character that is the source of the Get or the destination of the Put.

The contents of the box cursor are displayed at actual size (one cell = one pixel) in the upper-right corner of the screen. Two parameters, HB and PX, affect how colors are presented. Pressing the RETURN key when the High Bit menu entry is selected changes the setting of HB. In Apple hi-res graphics, the status of seven one-bit pixels is stored in the lower seven bits of a byte in memory. The eighth bit, the most significant bit, controls the colors in which these bits will be drawn. When drawing on the high-resolution screen in BASIC, the high bit is clear when HCOLOR is between zero and three, and is set when HCOLOR is between four and seven. The display is in blue and orange when the high bit is set, or green and violet when the high bit is clear.

The Even/Odd menu entry controls whether this display starts on an even or an odd pixel (PX). When a shape is shifted by one bit, the colors in the display are reversed (blue for orange or green for violet). The alignment of the shape is changed by pressing RETURN when Even/Odd is selected.

At the bottom of the screen, all of the characters in a 96-character set are shown. With the RAM/ROM function in the Display menu, the character set displayed can be either the set you are currently editing or the hardware character set in your Apple. Get and Put operate only on the RAM character set no matter which set is being displayed.

HROUT, The Character Generator

Apple SuperFont uses a machine language graphics utility called HROUT, for high-resolution output. HROUT links into the standard character output vector and permits text to be displayed on either hi-res screen. Because the standard text output routine also remains active, the PRINT command, and any other text commands, can be used to create hi-res text. HROUT's only limitation is that it cannot perform screen scrolls at the bottom of the screen.

To use HROUT in your own programs, BLOAD it into memory. It can be loaded anywhere in memory, but to make things simpler, we'll use location $300. First, let HROUT know which character set to use by POKEing the address of the character set into locations 6 and 7, low byte first. If you put your character set at $8D00, the POKEs are:

```
POKE 6,0 : POKE 7,141
```

If you are using DOS 3.3, you can activate HROUT by entering:

```
POKE 54,0 : POKE 55,3 : CALL 1002
```

When in immediate mode, these commands have to be entered together on a multistatement line (separated by colons). They can be on separate lines in a BASIC program, but the three commands should be executed one after another. Since locations 54 and 55 are being POKEd with the low and high bytes of the address of HROUT, these POKEs will be different if you put HROUT somewhere other than $300.

From ProDOS, it's easier to turn on HROUT. Just type

PR# A$300

Avoiding Screen Scrolls

Since HROUT concludes by calling the standard ROM routine for displaying a character on the text screen, all cursor control remains the same. You can move to any location on the screen by using the HTAB and VTAB commands. HOME still moves the cursor to the upper left of the screen, but will not clear the hi-res screen. To get the equivalent of a text HOME, use HOME : CALL −3092. The routine at −3092 clears the current hi-res screen and turns on hi-res graphics.

If you need to know what's where on the screen, you can PEEK to the text screen. By taking a couple of precautions, both text and hi-res screens should be the same. First of all, make sure that you clear both screens at the same time, as mentioned above. Second, don't let the text screen scroll. In order to make HROUT as small (88 bytes) and fast as possible, no provision was made for scrolling the screen. This could even be to your advantage for many applications, but you have to be careful if you want the text and graphics screens to agree.

The biggest problem arises when you print to the last character on the twenty-fourth line. Even if you follow the PRINT statement with a semicolon, the cursor will wrap onto the twenty-fifth line and the screen will scroll. There is a solution: Fool the computer into thinking the screen has 25 lines by using POKE 35,25. The output routine will then have no qualms at all about advancing the cursor to the twenty-fifth line, leaving it there, and even printing there. A lot of responsibility now rests on your shoulders, because the twenty-fifth line doesn't really exist. Printing something there is the same thing as POKEing out of the range of the text screen. That could cause significant problems.

If you change the text attribute with the INVERSE or FLASH commands, the bit patterns will be reversed before they are plotted on the screen, inverting the character. The NORMAL command also works, canceling inverted printing.

Now you can label high-resolution charts and graphs with a choice of any font, and you can design these fonts yourself with the Apple SuperFont Editor. Also, letters of the alphabet can become detailed shapes, permitting fast high-resolution game graphics in BASIC. In fact, we've started using this technique ourselves for some of the Apple games published in COMPUTE!.

Program 1: Apple SuperFont For DOS 3.3 (APPLEFONT)

```
100  IF  PEEK (978) <  > 157 THEN  PRINT
     "DOS 3.3 NOT FOUND": END
110  HGR
120 D$ =  CHR$ (4)
130  PRINT D$;"BLOAD HROUT"
140  PRINT D$;"BLOAD NORMAL.SET,A$8D00"

150  POKE 6,0: POKE 7,141
160  POKE 54,0: POKE 55,3: CALL 1002
170  PRINT D$;"BLOAD APPLEFONT2":CE = 4
     096
180  ONERR  GOTO 280
190  CALL CE
200 NA$ = "":I = 14 * 256
210  IF  PEEK (I) = 141 THEN 260
220 NA$ = NA$ +  CHR$ ( PEEK (I)):I = I
      + 1: IF  PEEK (I) <  > 141 THEN 2
     20
230  IF  PEEK (14 * 256 + 32) THEN 250
240  PRINT D$;"BLOAD";NA$;",A$8A00": GOTO
     260
250  PRINT D$;"BSAVE";NA$;",A$8A00,L$30
     0"
260  GOSUB 320
270  CALL CE + 3: GOTO 200
280  GOSUB 320: VTAB 18: HTAB 1:EN =  PEEK
     (222)
290  IF EN = 6 OR EN = 7 THEN  PRINT "C
     OULDN'T FIND "NA$: GOTO 270
300  IF EN = 13 THEN  PRINT NA$" ISN'T
     A CHARACTER SET": GOTO 270
310  PRINT "DISK ERROR": GOTO 270
320  VTAB 18: HTAB 1: FOR I = 1 TO 80: PRINT
     " ";: NEXT : RETURN
```

Program 2: Apple SuperFont ProDOS Modifications

```
100  IF  PEEK (978) <  > 190 THEN  PRINT
     "PRODOS NOT FOUND": END
160  PRINT D$;"PR# A$300"
```

Program 3: Apple SuperFont Editor (APPLEFONT2)

```
1000- 4C 53 12 4C 6D 12 41 00
1008- 00 05 05 07 08 00 00 00
1010- 00 07 08 00 01 00 00 00
1018- 04 03 05 06 06 00 1A 50
1020- 82 00 01 00 00 00 00 00
1028- 00 00 00 00 00 00 00 04
1030- 00 00 00 A9 00 85 1C A9
1038- 20 85 E6 20 F6 F3 A9 02
1040- 20 09 1D A0 00 A2 00 18
1048- 20 00 1D A2 DC 20 03 1D
1050- C8 C8 C8 C8 C0 84 90 ED
1058- A2 00 A0 00 18 20 00 1D
1060- A0 80 20 06 1D E8 E8 E8
1068- E8 E0 E0 90 ED 60 A9 FF
1070- 85 32 A9 8A 85 07 A9 A0
1078- 8D 23 10 A0 15 98 20 5B
```

```
1080- FB A2 00 AD 23 10 86 24
1088- 20 ED FD EE 23 10 E8 E0
1090- 20 D0 F0 C8 C0 18 D0 E5
1098- A9 8D 85 07 60 20 56 1C
10A0- B9 00 0C CE 2D 10 30 05
10A8- 1D BA 1C D0 03 3D B2 1C
10B0- 99 00 0C EE 2D 10 AD 2D
10B8- 10 F0 02 A9 03 20 09 1D
10C0- 18 AD 2A 10 0A 0A AA E8
10C8- AD 2C 10 0A 0A A8 C8 18
10D0- 8A 20 02 11 20 02 11 20
10D8- 02 11 AD 2A 10 CD 09 10
10E0- 90 1F ED 0B 10 90 05 CD
10E8- 09 10 B0 15 AD 2C 10 CD
10F0- 0A 10 90 0D ED 0C 10 90
10F8- 05 CD 0A 10 B0 03 20 2B
1100- 11 60 20 00 1D E8 E8 20
1108- 03 1D AA C8 60 A9 00 A8
1110- 99 00 0C C8 D0 FA 20 56
1118- 11 60 20 56 1C B9 00 0C
1120- 3D BA 1C F0 02 A9 01 8D
1128- 2D 10 60 AD 2D 10 F0 02
1130- A9 03 18 6D 2F 10 20 09
1138- 1D AD 2C 10 38 ED 0A 10
1140- A8 A9 E0 38 ED 09 10 18
1148- 6D 2E 10 6D 2A 10 AA 20
1150- 00 1D 20 03 1D 60 AC 0A
1158- 10 8C 2C 10 A0 00 8C 23
1160- 10 AE 09 10 8E 2A 10 A2
1168- 00 8E 24 10 20 1A 11 20
1170- 2B 11 EE 2A 10 EE 24 10
1178- AE 24 10 EC 0B 10 D0 EC
1180- EE 2C 10 EE 23 10 AC 23
1188- 10 CC 0C 10 D0 D3 60 AD
1190- 27 10 F0 50 30 28 AD 09
1198- 10 18 6D 0B 10 E9 00 8D
11A0- 2A 10 AD 0A 10 8D 2C 10
11A8- AC 0C 10 8C 23 10 20 1A
11B0- 11 20 2B 11 EE 2C 10 CE
11B8- 23 10 D0 F2 F0 26 AD 09
11C0- 10 18 6D 0B 10 8D 2A 10
11C8- AD 0A 10 8D 2C 10 AC 0C
11D0- 10 8C 23 10 A9 00 8D 2D
11D8- 10 20 2B 11 EE 2C 10 CE
11E0- 23 10 10 F5 AD 28 10 F0
11E8- 4F 30 27 AD 0A 10 18 6D
11F0- 0C 10 E9 00 8D 2C 10 AD
11F8- 09 10 8D 2A 10 AC 0B 10
1200- 8C 23 10 20 1A 11 20 2B
1208- 11 EE 2A 10 CE 23 10 D0
1210- F2 60 AD 0A 10 18 6D 0C
1218- 10 8D 2C 10 AD 09 10 8D
1220- 2A 10 AC 0B 10 8C 23 10
1228- A9 00 8D 2D 10 20 2B 11
1230- EE 2A 10 CE 23 10 10 F5
1238- 60 AD 2F 10 20 09 1D A0
1240- 00 A2 E0 18 20 00 1D A2
1248- 17 38 20 03 1D C8 C0 20
1250- 90 EF 60 20 58 FC 20 33
1258- 10 20 1A 19 2C 52 C0 20
1260- 0D 11 A9 C1 8D 06 10 20
1268- 9E 1A 20 DA 1B 20 6E 10
1270- A9 00 8D 13 10 8D 27 10
1278- 8D 28 10 20 7D 19 AD 13
1280- 10 49 02 8D 13 10 A2 00
1288- A0 80 AD 00 C0 30 08 E8
1290- D0 F8 C8 D0 F5 10 E4 48
1298- A0 02 8C 13 10 20 7D 19
12A0- 68 2C 10 C0 C9 A0 D0 1B
12A8- AD 21 10 38 69 00 CD 18
12B0- 10 D0 02 A9 00 8D 21 10
12B8- A9 01 8D 22 10 20 9E 1A
12C0- 4C 01 13 C9 88 D0 17 AD
12C8- 22 10 18 E9 00 D0 06 AC
12D0- 21 10 B9 19 10 8D 22 10
12D8- 20 9E 1A 4C 01 13 C9 95
12E0- D0 1F AD 22 10 38 69 00
12E8- 8D 22 10 AC 21 10 B9 19
12F0- 10 CD 22 10 B0 05 A9 01
12F8- 8D 22 10 20 9E 1A 4C 01
1300- 13 C9 D7 D0 09 CE 27 10
1308- CE 28 10 4C 5A 13 C9 C5
1310- D0 06 CE 28 10 4C 5A 13
1318- C9 D2 D0 09 CE 28 10 EE
1320- 27 10 4C 5A 13 C9 D3 D0
1328- 06 CE 27 10 4C 5A 13 C9
1330- C6 D0 06 EE 27 10 4C 5A
1338- 13 C9 D8 D0 09 CE 27 10
1340- EE 28 10 4C 5A 13 C9 C3
1348- D0 06 EE 28 10 4C 5A 13
1350- C9 D6 D0 06 EE 28 10 EE
1358- 27 10 AE 21 10 D0 03 4C
1360- 75 13 CA D0 03 4C D5 13
1368- CA D0 03 4C C5 15 CA D0
1370- 03 4C 37 18 60 48 20 31
1378- 19 A9 01 8D 14 10 68 C9
1380- C4 F0 08 AD 27 10 0D 28
1388- 10 F0 47 AD 22 10 C9 03
1390- F0 14 69 FF 8D 2D 10 AD
1398- 07 10 8D 2A 10 AD 08 10
13A0- 8D 2C 10 20 9D 10 AD 07
13A8- 10 18 6D 27 10 C9 FF D0
13B0- 02 A9 36 C9 37 D0 02 A9
13B8- 00 8D 07 10 AD 08 10 18
13C0- 6D 28 10 C9 FF D0 02 A9
13C8- 1F C9 20 D0 02 A9 00 8D
13D0- 08 10 4C 70 12 48 AE 22
13D8- 10 CA D0 03 4C F8 13 CA
13E0- D0 03 4C 74 14 CA D0 03
13E8- 4C DA 14 CA D0 03 4C 55
13F0- 15 CA D0 03 4C 64 15 00
13F8- 20 31 19 A9 01 8D 15 10
1400- 20 3F 19 68 4C 70 12 20
1408- 31 19 A9 01 8D 16 10 AD
1410- 0B 10 CD 11 10 D0 08 AD
1418- 0C 10 CD 12 10 F0 2A AD
1420- 11 10 8D 0B 10 AD 12 10
1428- 8D 0C 10 20 39 12 20 56
1430- 11 AD 09 10 C9 31 90 05
1438- A9 30 8D 09 10 AD 0A 10
1440- C9 19 90 05 A9 18 8D 0A
1448- 10 AD 27 10 0D 28 10 F0
1450- 22 AD 28 10 F0 05 0A 0A
1458- 0A 0A 0A 18 6D 27 10 18
1460- 6D 06 10 C9 A0 10 02 69
```

```
1468- 60 C9 00 30 03 38 E9 60
1470- 8D 06 10 60 20 07 14 68
1478- C9 8D D0 5B 20 38 15 AD
1480- 0A 10 8D 2C 10 AD 0C 10
1488- 8D 24 10 A9 00 8D 31 10
1490- AD 09 10 8D 2A 10 AD 0B
1498- 10 8D 23 10 A9 00 8D 32
14A0- 10 20 1A 11 AD 2D 10 F0
14A8- 02 38 24 18 6E 32 10 EE
14B0- 2A 10 CE 23 10 D0 EA AD
14B8- 2F 10 F0 02 A9 80 6E 32
14C0- 10 0D 32 10 AC 31 10 91
14C8- 1A EE 2C 10 EE 31 10 CE
14D0- 24 10 D0 BC 20 6E 10 4C
14D8- 70 12 20 07 14 68 C9 8D
14E0- D0 53 20 38 15 AD 0A 10
14E8- 8D 2C 10 AD 0C 10 8D 24
14F0- 10 A9 00 8D 31 10 AD 09
14F8- 10 8D 2A 10 AD 0B 10 8D
1500- 23 10 AC 31 10 B1 1A 8D
1508- 32 10 4E 32 10 A9 00 69
1510- 00 8D 2D 10 20 9D 10 EE
1518- 2A 10 CE 23 10 D0 EB AD
1520- 32 10 0A 0A 8D 2F 10 EE
1528- 2C 10 EE 31 10 CE 24 10
1530- D0 C4 20 DA 1B 4C 70 12
1538- AD 06 10 38 E9 A0 85 1A
1540- A9 00 85 1B A2 03 06 1A
1548- 26 1B CA D0 F9 A5 1B 18
1550- 69 8A 85 1B 60 20 31 19
1558- 68 C9 8D D0 04 A9 00 F0
1560- 12 4C 70 12 20 31 19 68
1568- C9 8D D0 04 A9 01 D0 03
1570- 4C 70 12 8D 20 0E A9 A0
1578- A2 1F 9D 00 0E CA 10 FA
1580- A9 FF 85 32 A9 11 20 5B
1588- FB A9 00 85 24 A0 00 B9
1590- A7 15 F0 06 20 ED FD C8
1598- D0 F5 20 6A FD BD 00 02
15A0- 9D 00 0E CA 10 F7 60 C5
15A8- CE D4 C5 D2 A0 CE C1 CD
15B0- C5 A0 CF C6 A0 C3 C8 C1
15B8- D2 C1 C3 D4 C5 D2 A0 D3
15C0- C5 D4 BA 8D 00 48 AE 22
15C8- 10 CA D0 03 4C F8 13 CA
15D0- D0 03 4C EE 15 CA D0 03
15D8- 4C 37 16 CA D0 03 4C EB
15E0- 16 CA D0 03 4C 66 17 CA
15E8- D0 03 4C E1 17 00 20 31
15F0- 19 A9 01 8D 15 10 68 AD
15F8- 27 10 0D 28 10 F0 35 AD
1600- 27 10 18 6D 0B 10 D0 02
1608- A9 01 8D 0B 10 18 6D 09
1610- 10 C9 38 D0 03 CE 0B 10
1618- AD 28 10 18 6D 0C 10 D0
1620- 02 A9 01 8D 0C 10 18 6D
1628- 0A 10 C9 21 D0 03 CE 0C
1630- 10 20 8F 11 4C 70 12 20
1638- 31 19 A9 01 8D 17 10 AD
1640- 27 10 0D 28 10 F0 32 AD
1648- 27 10 18 6D 0D 10 10 02
1650- A9 00 8D 0D 10 18 6D 0F
1658- 10 C9 38 D0 03 CE 0D 10
1660- AD 28 10 18 6D 0E 10 10
1668- 02 A9 00 8D 0E 10 18 6D
1670- 10 10 C9 21 D0 03 CE 0E
1678- 10 68 C9 8D D0 6A 20 70
1680- 1C AD 0A 10 8D 23 10 AD
1688- 0E 10 8D 25 10 AD 09 10
1690- 8D 24 10 AD 0D 10 8D 26
1698- 10 AD 23 10 8D 2C 10 AD
16A0- 24 10 8D 2A 10 20 56 1C
16A8- B9 00 0D 3D BA 1C F0 02
16B0- A9 01 8D 2D 10 AD 25 10
16B8- 8D 2C 10 AD 26 10 8D 2A
16C0- 10 20 9D 10 EE 24 10 EE
16C8- 26 10 AD 09 10 18 6D 0B
16D0- 10 CD 24 10 D0 C3 EE 23
16D8- 10 EE 25 10 AD 0A 10 18
16E0- 6D 0C 10 CD 23 10 D0 A5
16E8- 4C 70 12 20 31 19 A9 01
16F0- 8D 15 10 20 3F 19 68 C9
16F8- 8D D0 68 20 70 1C AD 0A
1700- 10 8D 23 10 8D 25 10 AD
1708- 09 10 8D 24 10 18 6D 0B
1710- 10 E9 00 8D 26 10 AD 23
1718- 10 8D 2C 10 AD 24 10 8D
1720- 2A 10 20 56 1C B9 00 0D
1728- 3D BA 1C F0 02 A9 01 8D
1730- 2D 10 AD 25 10 8D 2C 10
1738- AD 26 10 8D 2A 10 20 9D
1740- 10 EE 24 10 AD 26 10 CD
1748- 09 10 F0 05 CE 26 10 B0
1750- C5 EE 23 10 EE 25 10 AD
1758- 0A 10 18 6D 0C 10 CD 25
1760- 10 D0 A4 4C 70 12 20 31
1768- 19 A9 01 8D 15 10 20 3F
1770- 19 68 C9 8D D0 68 20 70
1778- 1C AD 0A 10 8D 23 10 18
1780- 6D 0C 10 E9 00 8D 25 10
1788- AD 09 10 8D 24 10 8D 26
1790- 10 AD 23 10 8D 2C 10 AD
1798- 24 10 8D 2A 10 20 56 1C
17A0- B9 00 0D 3D BA 1C F0 02
17A8- A9 01 8D 2D 10 AD 25 10
17B0- 8D 2C 10 AD 26 10 8D 2A
17B8- 10 20 9D 10 EE 24 10 EE
17C0- 26 10 AD 09 10 18 6D 0B
17C8- 10 CD 24 10 D0 C3 EE 23
17D0- 10 AD 25 10 CD 0A 10 F0
17D8- 05 CE 25 10 B0 AA 4C 70
17E0- 12 20 31 19 A9 01 8D 15
17E8- 10 20 3F 19 68 C9 8D D0
17F0- 43 20 70 1C AD 0A 10 8D
17F8- 2C 10 AD 09 10 8D 2A 10
1800- 20 56 1C B9 00 0D 3D BA
1808- 1C D0 03 A9 01 2C A9 00
1810- 8D 2D 10 20 9D 10 EE 2A
1818- 10 AD 09 10 18 6D 0B 10
1820- CD 2A 10 D0 DB EE 2C 10
1828- AD 0A 10 18 6D 0C 10 CD
1830- 2C 10 D0 C6 4C 70 12 48
1838- 20 31 19 AE 22 10 CA D0
1840- 03 4C 63 18 CA D0 03 4C
1848- 7A 18 CA D0 03 4C 94 18
1850- CA D0 03 4C B2 18 CA D0
```

```
1858- 03 4C F2 18 CA D0 03 4C
1860- 0C 19 00 68 C9 8D D0 0F
1868- A9 04 38 ED 2F 10 8D 2F
1870- 10 20 56 11 20 DA 1B 4C
1878- 70 12 68 C9 8D D0 12 A9
1880- 01 38 ED 2E 10 8D 2E 10
1888- 20 39 12 20 56 11 20 DA
1890- 1B 4C 70 12 68 C9 8D D0
1898- 16 A9 01 38 ED 30 10 8D
18A0- 30 10 F0 05 2C 53 C0 B0
18A8- 06 2C 52 C0 20 DA 1B 4C
18B0- 70 12 A9 01 8D 15 10 20
18B8- 3F 19 68 C9 8D D0 30 A9
18C0- 00 8D 2D 10 AD 0A 10 8D
18C8- 2C 10 AD 0C 10 8D 23 10
18D0- AD 09 10 8D 2A 10 AD 0B
18D8- 10 8D 24 10 20 9D 10 EE
18E0- 2A 10 CE 24 10 D0 F5 EE
18E8- 2C 10 CE 23 10 D0 E1 4C
18F0- 70 12 68 C9 8D D0 12 20
18F8- 58 FC 20 33 10 20 9E 1A
1900- 20 6E 10 20 DA 1B 20 0D
1908- 11 4C 70 12 68 C9 8D D0
1910- 06 20 1A 19 20 6E 10 4C
1918- 70 12 A9 8A 85 1B A9 00
1920- 85 1A A2 03 A0 00 91 1A
1928- C8 D0 FB E6 1B CA D0 F6
1930- 60 48 A2 00 8A 9D 14 10
1938- E8 E0 04 90 F8 68 60 AD
1940- 27 10 0D 28 10 F0 35 AD
1948- 27 10 18 6D 09 10 10 02
1950- A9 00 8D 09 10 18 6D 0B
1958- 10 C9 38 D0 03 CE 09 10
1960- AD 28 10 18 6D 0A 10 10
1968- 02 A9 00 8D 0A 10 18 6D
1970- 0C 10 C9 21 D0 03 CE 0A
1978- 10 20 56 11 60 A9 00 20
1980- 09 1D AD 14 10 F0 06 AD
1988- 13 10 20 09 1D AD 07 10
1990- 0A 0A AA AD 08 10 0A 0A
1998- A8 18 20 00 1D 8A 69 04
19A0- AA 20 03 1D 98 69 04 A8
19A8- 20 06 1D 8A 38 E9 04 18
19B0- AA 20 03 1D 98 38 E9 04
19B8- A8 20 06 1D A9 00 20 09
19C0- 1D AD 15 10 F0 06 AD 13
19C8- 10 20 09 1D AD 09 10 0A
19D0- 0A AA AD 0A 10 0A 0A A8
19D8- 18 20 00 1D AD 09 10 6D
19E0- 0B 10 0A 0A AA 20 03 1D
19E8- AD 0A 10 6D 0C 10 0A 0A
19F0- A8 20 06 1D AD 09 10 0A
19F8- 0A AA 20 03 1D AD 0A 10
1A00- 0A 0A A8 20 06 1D A9 3F
1A08- 85 32 A9 8A 85 07 AD 06
1A10- 10 29 1F 85 24 AD 06 10
1A18- 29 60 A2 05 4A CA D0 FC
1A20- 69 14 20 5B FB AD 16 10
1A28- F0 09 AD 13 10 F0 04 A9
1A30- FF 85 32 AD 06 10 20 ED
1A38- FD A9 8D 85 07 AD 17 10
1A40- F0 43 AD 13 10 20 09 1D
1A48- AD 0D 10 0A 0A AA AD 0E
1A50- 10 0A 0A A8 18 20 00 1D
1A58- AD 0D 10 6D 0F 10 0A 0A
1A60- AA 20 03 1D AD 0E 10 6D
1A68- 10 10 0A 0A A8 20 06 1D
1A70- AD 0D 10 0A 0A AA 20 03
1A78- 1D AD 0E 10 0A 0A A8 20
1A80- 06 1D 4C 9D 1A AD 09 10
1A88- 8D 0D 10 AD 0A 10 8D 0E
1A90- 10 AD 0B 10 8D 0F 10 AD
1A98- 0C 10 8D 10 10 60 A9 08
1AA0- 20 5B FB A9 3F 85 32 AC
1AA8- 21 10 B9 19 10 8D 23 10
1AB0- 18 69 01 38 ED 22 10 8D
1AB8- 24 10 B9 1D 10 A8 20 FF
1AC0- 1A A9 8D 20 ED FD A9 FF
1AC8- 85 32 AD 24 10 CD 23 10
1AD0- D0 04 A9 3F 85 32 20 FF
1AD8- 1A CE 23 10 D0 E8 A9 FF
1AE0- 85 32 AC 21 10 AD 19 10
1AE8- 8D 23 10 A2 08 A9 20 85
1AF0- 24 20 11 1B EE 23 10 A9
1AF8- 08 CD 23 10 D0 ED 60 A2
1B00- 08 A9 20 85 24 B9 1F 1B
1B08- F0 07 20 ED FD CA C8 D0
1B10- F4 C8 E0 00 F0 08 A9 A0
1B18- 20 ED FD CA D0 FA 60 C2
1B20- C9 D4 A0 C5 C4 C9 D4 00
1B28- C2 CC C1 C3 CB 00 D7 C8
1B30- C9 D4 C5 00 CD CF D6 C5
1B38- 00 C3 C8 C1 D2 D3 C5 D4
1B40- D3 00 CD CF D6 C5 A0 C2
1B48- CF D8 00 D0 D5 D4 A0 C3
1B50- C8 C1 D2 00 C7 C5 D4 A0
1B58- C3 C8 C1 D2 00 CC CF C1
1B60- C4 A0 D3 C5 D4 00 D3 C1
1B68- D6 C5 A0 D3 C5 D4 00 D5
1B70- D4 C9 CC C9 D4 D9 00 CD
1B78- CF D6 C5 A0 C2 CF D8 00
1B80- C2 CF D8 A0 D3 C9 DA C5
1B88- 00 C3 CF D0 D9 00 CD C9
1B90- D2 D2 CF D2 00 C6 CC C9
1B98- D0 00 C9 CE D6 C5 D2 D4
1BA0- 00 C4 C9 D3 D0 CC C1 D9
1BA8- 00 C8 C9 A0 C2 C9 D4 00
1BB0- C5 D6 C5 CE AF CF C4 C4
1BB8- 00 D2 C1 CD AF D2 CF CD
1BC0- 00 C3 CC D2 A0 C2 CF D8
1BC8- 00 C3 CC D2 A0 D3 C3 D2
1BD0- CE 00 C3 CC D2 A0 D3 C5
1BD8- D4 00 A9 05 20 5B FB A9
1BE0- 20 85 24 A9 FF 85 32 A0
1BE8- 00 A9 04 20 2C 1C AD 2F
1BF0- 10 D0 09 A0 04 A9 03 20
1BF8- 2C 1C F0 07 A0 07 A9 03
1C00- 20 2C 1C A9 8D 20 ED FD
1C08- A9 20 85 24 A0 0A A9 04
1C10- 20 2C 1C AD 2E 10 D0 09
1C18- A0 0E A9 04 20 2C 1C F0
1C20- 0A A0 12 A9 04 20 2C 1C
1C28- AD 2E 10 60 8D 29 10 A2
1C30- 00 B9 3F 1C 20 ED FD C8
1C38- E8 EC 29 10 D0 F3 60 C8
1C40- C2 BA A0 C3 CC D2 D3 C5
```

```
1C48- D4 D0 D8 BA A0 C5 D6 C5
1C50- CE CF C4 C4 A0 00 AD 2A
1C58- 10 0A 0A 8D 2B 10 AD 2C
1C60- 10 29 07 AA AD 2C 10 4A
1C68- 4A 4A 18 6D 2B 10 A8 60
1C70- AD 0A 10 8D 2C 10 AD 0C
1C78- 10 8D 23 10 AD 09 10 8D
1C80- 2A 10 AD 0B 10 8D 24 10
1C88- 20 1A 11 B9 00 0D CE 2D
1C90- 10 30 05 1D BA 1C D0 03
1C98- 3D B2 1C 99 00 0D EE 2D
1CA0- 10 EE 2A 10 CE 24 10 D0
1CA8- DF EE 2C 10 CE 23 10 D0
1CB0- CB 60 FE FD FB F7 EF DF
1CB8- BF 7F 01 02 04 08 10 20
1CC0- 40 80 8D A0 A0 A0 A0 A0
1CC8- A0 A0 A0 A0 A0 A0 A0 A0
1CD0- A0 A0 A0 A0 A0 A0 A0 A0
1CD8- A0 A0 A0 A0 A0 A0 A0 A0
1CE0- A0 A0 00 FF 00 FF 00 FF
1CE8- 00 FF B7 FF 00 FF 00 FF
1CF0- 00 FF 00 FF 00 FF 00 FF
1CF8- 00 FF 00 FF 00 FF 00 FF
1D00- 4C 05 1F 4C 22 1E 4C C4
1D08- 1D 4C F6 1E 00 00 00 00
1D10- 00 00 00 00 00 A5 1C 51
1D18- 26 25 30 51 26 91 26 60
1D20- 85 45 86 46 84 47 60 A5
1D28- 45 A6 46 A4 47 60 A5 1C
1D30- 4A 4A 4A 4C 40 1D A5 1C
1D38- 4A 4C 40 1D A5 1C 4A 4A
1D40- 29 0F A8 B9 4F 1D 24 1C
1D48- 10 02 09 80 85 1C 60 00
1D50- 11 22 33 44 55 66 77 08
1D58- 19 2A 3B 4C 5D 6E 7F 00
1D60- 04 08 0C 10 14 18 1C 00
1D68- 04 08 0C 10 14 18 1C 01
1D70- 05 09 0D 11 15 19 1D 01
1D78- 05 09 0D 11 15 19 1D 02
1D80- 06 0A 0E 12 16 1A 1E 02
1D88- 06 0A 0E 12 16 1A 1E 03
1D90- 07 0B 0F 13 17 1B 1F 03
1D98- 07 0B 0F 13 17 1B 1F 81
1DA0- 82 84 88 90 A0 C0 81 83
1DA8- 87 8F 9F BF FF FF FE FC
1DB0- F8 F0 E0 C0 00 2A 55 7F
1DB8- 80 AA D5 FF 22 11 77 5D
1DC0- A2 91 F7 BB 08 20 20 1D
1DC8- C0 C0 90 03 4C B8 1F AC
1DD0- 0D 1D B9 9F 1D 85 30 A5
1DD8- 27 29 1F 05 E6 85 27 A5
1DE0- 47 38 ED 10 1D AA 6E 11
1DE8- 1D 10 17 E8 AC 0C 1D 20
1DF0- 15 1D CA F0 23 AD 11 1D
1DF8- 18 20 D3 F4 20 3C 1D 4C
1E00- EC 1D CA AC 0C 1D 20 15
1E08- 1D E8 F0 0C AD 11 1D 20
1E10- D3 F4 20 3C 1D 4C 03 1E
1E18- A5 47 8D 10 1D 20 27 1D
1E20- 28 60 08 20 20 1D 90 07
1E28- E0 18 90 03 4C B8 1F A0
1E30- 00 8C 0E 1D 8E 0F 1D 28
1E38- 08 90 03 EE 0E 1D A5 27
1E40- 29 1F 05 E6 85 27 AD 0F
1E48- 1D A2 E0 8E 12 1D AE 0E
1E50- 1D 8E 13 1D 4E 13 1D A2
1E58- 06 90 03 69 1F 38 2E 13
1E60- 1D CD 12 1D 90 06 EE 13
1E68- 1D ED 12 1D 4E 12 1D CA
1E70- D0 EC 8D 14 1D AE 0D 1D
1E78- AC 0C 1D CC 13 1D D0 1E
1E80- EC 14 1D 90 0B BD A6 1D
1E88- AE 14 1D 3D AD 1D B0 09
1E90- BD AD 1D AE 14 1D 3D A6
1E98- 1D 85 30 4C EB 1E 90 27
1EA0- BD A6 1D 85 30 20 15 1D
1EA8- 20 36 1D CE 0C 1D AC 0C
1EB0- 1D CC 13 1D F0 06 A5 1C
1EB8- 91 26 B0 EC AE 14 1D BD
1EC0- AD 1D 85 30 4C EB 1E BD
1EC8- AD 1D 85 30 20 15 1D 20
1ED0- 2E 1D EE 0C 1D AC 0C 1D
1ED8- CC 13 1D F0 06 A5 1C 91
1EE0- 26 90 EC AE 14 1D BD A6
1EE8- 1D 85 30 20 15 1D 8E 0D
1EF0- 1D 20 27 1D 28 60 08 20
1EF8- 20 1D 29 0F A8 B9 B4 1D
1F00- 85 1C 4C 9B 1F 08 20 20
1F08- 1D 90 07 E0 18 90 03 4C
1F10- B8 1F C0 C0 90 03 20 B8
1F18- 1F A9 01 2D 10 1D F0 03
1F20- 20 3C 1D A9 03 2D 0C 1D
1F28- F0 07 AA 20 36 1D CA D0
1F30- FA A5 47 8D 10 1D A5 46
1F38- 8D 0F 1D A0 00 8C 0E 1D
1F40- 28 08 90 03 EE 0E 1D 2C
1F48- 10 1D 10 02 A0 50 50 02
1F50- A0 28 84 26 A9 08 2D 10
1F58- 1D F0 06 A9 80 05 26 85
1F60- 26 AD 10 1D 29 3F A8 B9
1F68- 5F 1D 85 27 AD 0F 1D A2
1F70- E0 8E 12 1D AE 0E 1D 8E
1F78- 0C 1D 4E 0C 1D A2 06 90
1F80- 03 69 1F 38 2E 0C 1D CD
1F88- 12 1D 90 06 EE 0C 1D ED
1F90- 12 1D 4E 12 1D CA D0 EC
1F98- 8D 0D 1D A9 01 2D 10 1D
1FA0- F0 03 20 3C 1D A9 03 2D
1FA8- 0C 1D F0 07 AA 20 2E 1D
1FB0- CA D0 FA 20 27 1D 28 60
1FB8- A0 00 B9 C6 1F 20 ED FD
1FC0- C8 C0 19 D0 F5 00 D3 C3
1FC8- D2 C5 C5 CE A0 C2 CF D5
1FD0- CE C4 C1 D2 D9 A0 C5 D8
1FD8- C3 C5 C5 C4 C5 C4 8D 00
```

Program 4: Apple SuperFont NORMAL.SET

```
8D00- 00 00 00 00 00 00 00 00
8D08- 08 08 08 08 08 00 08 00
8D10- 14 14 14 00 00 00 00 00
8D18- 14 14 3E 14 3E 14 14 00
8D20- 08 3C 0A 1C 28 1E 08 00
8D28- 06 26 10 08 04 32 30 00
8D30- 04 0A 0A 04 2A 12 2C 00
8D38- 08 08 08 00 00 00 00 00
8D40- 08 04 02 02 02 04 08 00
```

```
8D48- 08 10 20 20 20 10 08 00
8D50- 08 2A 1C 08 1C 2A 08 00
8D58- 00 08 08 3E 08 08 00 00
8D60- 00 00 00 00 08 08 04 00
8D68- 00 00 00 3E 00 00 00 00
8D70- 00 00 00 00 00 00 08 00
8D78- 00 20 10 08 04 02 00 00
8D80- 1C 22 32 2A 26 22 1C 00
8D88- 08 0C 08 08 08 08 1C 00
8D90- 1C 22 20 18 04 02 3E 00
8D98- 3E 20 10 18 20 22 1C 00
8DA0- 10 18 14 12 3E 10 10 00
8DA8- 3E 02 1E 20 20 22 1C 00
8DB0- 38 04 02 1E 22 22 1C 00
8DB8- 3E 20 10 08 04 04 04 00
8DC0- 1C 22 22 1C 22 22 1C 00
8DC8- 1C 22 22 3C 20 10 0E 00
8DD0- 00 00 08 00 08 00 00 00
8DD8- 00 00 08 00 08 08 04 00
8DE0- 10 08 04 02 04 08 10 00
8DE8- 00 00 3E 00 3E 00 00 00
8DF0- 04 08 10 20 10 08 04 00
8DF8- 1C 22 10 08 08 00 08 00
8E00- 1C 22 2A 3A 1A 02 3C 00
8E08- 08 14 22 22 3E 22 22 00
8E10- 1E 22 22 1E 22 22 1E 00
8E18- 1C 22 02 02 02 22 1C 00
8E20- 1E 22 22 22 22 22 1E 00
8E28- 3E 02 02 1E 02 02 3E 00
8E30- 3E 02 02 1E 02 02 02 00
8E38- 3C 02 02 02 32 22 3C 00
8E40- 22 22 22 3E 22 22 22 00
8E48- 1C 08 08 08 08 08 1C 00
8E50- 20 20 20 20 20 22 1C 00
8E58- 22 12 0A 06 0A 12 22 00
8E60- 02 02 02 02 02 02 3E 00
8E68- 22 36 2A 2A 22 22 22 00
8E70- 22 22 26 2A 32 22 22 00
8E78- 1C 22 22 22 22 22 1C 00
8E80- 1E 22 22 1E 02 02 02 00
8E88- 1C 22 22 22 2A 12 2C 00
8E90- 1E 22 22 1E 0A 12 22 00
8E98- 1C 22 02 1C 20 22 1C 00
8EA0- 3E 08 08 08 08 08 08 00
8EA8- 22 22 22 22 22 22 1C 00
8EB0- 22 22 22 22 22 14 08 00
8EB8- 22 22 22 2A 2A 36 22 00
8EC0- 22 22 14 08 14 22 22 00
8EC8- 22 22 14 08 08 08 08 00
8ED0- 3E 20 10 08 04 02 3E 00
8ED8- 3E 06 06 06 06 06 3E 00
8EE0- 00 02 04 08 10 20 00 00
8EE8- 3E 30 30 30 30 30 3E 00
8EF0- 00 00 08 14 22 00 00 00
8EF8- 00 00 00 00 00 00 00 7F
8F00- 04 08 10 00 00 00 00 00
8F08- 00 00 1C 20 3C 22 3C 00
8F10- 02 02 1E 22 22 22 1E 00
8F18- 00 00 3C 02 02 02 3C 00
8F20- 20 20 3C 22 22 22 3C 00
8F28- 00 00 1C 22 3E 02 3C 00
8F30- 18 24 04 1E 04 04 04 00
8F38- 00 00 1C 22 22 3C 20 1C
8F40- 02 02 1E 22 22 22 22 00
8F48- 08 00 0C 08 08 08 1C 00
8F50- 10 00 18 10 10 10 12 0C
8F58- 02 02 22 12 0E 12 22 00
8F60- 0C 08 08 08 08 08 1C 00
8F68- 00 00 36 2A 2A 2A 22 00
8F70- 00 00 1E 22 22 22 22 00
8F78- 00 00 1C 22 22 22 1C 00
8F80- 00 00 1E 22 22 1E 02 02
8F88- 00 00 3C 22 22 3C 20 20
8F90- 00 00 3A 06 02 02 02 00
8F98- 00 00 3C 02 1C 20 1E 00
8FA0- 04 04 1E 04 04 24 18 00
8FA8- 00 00 22 22 22 32 2C 00
8FB0- 00 00 22 22 22 14 08 00
8FB8- 00 00 22 22 2A 2A 36 00
8FC0- 00 00 22 14 08 14 22 00
8FC8- 00 00 22 22 22 3C 20 1C
8FD0- 00 00 3E 10 08 04 3E 00
8FD8- 38 0C 0C 06 0C 0C 38 00
8FE0- 08 08 08 08 08 08 08 08
8FE8- 0E 18 18 30 18 18 0E 00
8FF0- 2C 1A 00 00 00 00 00 00
8FF8- 00 2A 14 2A 14 2A 00 00
```

Program 5: Apple SuperFont HROUT

```
0300- D8 78 85 45 86 46 84 47
0308- A6 07 0A 0A B0 04 10 3E
0310- 30 04 10 01 E8 E8 0A 86
0318- 1B 18 65 06 85 1A 90 02
0320- E6 1B A5 28 85 08 A5 29
0328- 29 03 05 E6 85 09 A2 08
0330- A0 00 B1 1A 24 32 30 02
0338- 49 7F A4 24 91 08 E6 1A
0340- D0 02 E6 1B A5 09 18 69
0348- 04 85 09 CA D0 E2 A5 45
0350- A6 46 A4 47 58 4C F0 FD
```

Program 6: APPLEFONT2 Checksum

```
100  PRINT "CHECK THESE BLOCKS:";
110  FOR I = 8160 TO 8191: POKE I,0: NEXT
120  FOR I = 0 TO 63:S = 0
130  PRINT ".";
140  FOR J = 0 TO 63:S = S +  PEEK (409
     6 + I * 64 + J): NEXT
150  READ A:S = S - 256 * INT (S / 256
160 AD = 4096 + I * 64: GOSUB 200:A1$ =
     H$
170 AD = 4096 + I * 64 + 63: GOSUB 200:
     A2$ = H$
180  IF A <  > S THEN  PRINT : PRINT "$
     ";A1$;" TO $";A2$;
190  NEXT : END
200 H$ = "": FOR K = 0 TO 3:X =  INT (A
     D / 16):H$ =  MID$ ("0123456789ABC
     DEF",AD - X * 16 + 1,1) + H$:AD =
     X: NEXT : RETURN
210  DATA 33,162,119,1,233,214,30,118
220  DATA 36,37,152,145,189,208,216,189
230  DATA 80,127,70,70,155,173,185,120
240  DATA 74,176,171,163,153,216,210,18
250  DATA 191,187,202,174,159,187,87,125
260  DATA 41,199,222,202,111,195,52,127
270  DATA 178,145,237,85,215,171,8,97
280  DATA 123,137,102,120,22,29,223,142
```

©

The Home Computer Revolution: Another False Start?

Fred D'Ignazio, Associate Editor

In my recent columns I have written about the overselling of the home computer. (See "The Morning After: Anti-Computer Backlash And The Arrival Of The Mass-Market Home Computer," COMPUTE!, May and June 1984; and "Is The Computer A Home Appliance?," COMPUTE!, August 1984.)

Now it seems that a genuine backlash against home computers has appeared. In publication after publication, and on TV and radio, we hear that the "home computer revolution" was a fluke. Commentators and reporters tell us that computers are still too difficult, too finicky, and too expensive to be a mass-market "appliance." And, unlike the TV, the telephone, and the toaster oven, there is no compelling reason to own a computer.

There is some truth to all of these charges, and, collectively, they have chipped away at the glossy high-tech image that home computers have enjoyed for the last couple of years. As a result, the glamour has worn off the home computer, and this has caused the industry to sag.

History Repeats Itself

But this is not the first time it's happened. In 1975, when the first computer kit (the Altair) appeared, there was a lot of discussion in the media about a "home computer revolution." This discussion was short-lived, however, because the first computers were strictly hobbyist devices. They had very little memory, almost no software, and were not built, distributed, serviced, or supported as consumer products.

The home computer hype started again in 1977 when Apple introduced its Apple II, Radio Shack came out with the TRS-80 Model I, and Commodore introduced its PET. Again we heard claims about how computers would soon be in everyone's homes. Unfortunately, these claims were just as premature as they were before. Like the machines before them, these new computers were suitable only for hobbyists and students as do-it-yourself educational devices.

We are now at the end of a third wave of claims that the home computer has arrived. This wave, like the others, has subsided and turned sour because our computer technology is still not mature enough to create a true, mass-market consumer product.

There have been three false starts in launching the home computer revolution, and there are sure to be more. Home computers are now in five million homes, but they're used daily in only a minority of those homes. It will be a long time before computers appear in 100 percent of people's homes and become a way of life like telephones or TV sets.

Fred D'Ignazio is a computer enthusiast, the father of two children, and the author of several books on computers for young people. His books include Katie and the Computer *(Creative Computing),* Working Robots *(Hayden),* The Star Wars Question and Answer Book about Computers *(Random House), and* Computing Together: A Parents and Teachers Guide to Using Computers with Young Children *(COMPUTE! Publications).*

Fred appears regularly as the "family computing" commentator on "The New Tech Times," a half-hour public-TV program on consumer electronics that airs weekly on over 240 stations across the country.

Fred's column appears monthly in COMPUTE!.

The Digital Utility Center

Experts predict that a real home computer will not appear until computers are integrated into all aspects of people's lives, including banking, shopping, working, communicating, and entertainment. A real home computer will not sit alone on a desktop and look like a typewriter plugged into a TV set. Instead, it will be a hybrid machine—part TV, part telephone, part videocassette recorder, and part stereo system. It will be the brains of a general-purpose digital utility center that a family operates to hear music, watch movies and TV, make phone calls, control household appliances, and pay bills.

The home computer of the present is made up of awkward, ill-fitted, and confusing components. The day its components fuse together into a single digital utility center that is sold at discount supermarkets, it will truly become a mass-market device.

The digital utility center will come in a single box and plug into the wall with a single cord. The center's audio, video, and computer software will be uniform and standardized (in some kind of optical or magnetic format), and will play everything—from educational games to Bruce Springsteen to the latest Burt Reynolds movie.

All the recordings will be digital and capable of being stored on a single, high-density storage device. All programming will be in English and will consist of making simple choices from a menu of selections that appears on a screen and are read to the user aloud by the center's synthesized voice. Input will be from a keyboard, light pen, mouse, microphone, or touch screen, depending on the individual's preference. No technical knowledge whatsoever will be needed to operate the center. And the center will come with one- to five-year warranties, full service contracts, and modular, replaceable parts.

Like The Electric Motor

When the digital utility center arrives, the home computer will really be a mass-market appliance. But when computers have become digital utility centers, they will no longer be computers. To paraphrase Joseph Weizenbaum, a digital utility center to a computer is the same as a vacuum cleaner to an electric motor.

Before we see consumers going wild over digital utility centers, a lot of separate developments have to take place. Audio, video, communications, and computer hardware must evolve much further and become more integrated, digital, compatible, and inexpensive. Software for the separate devices has to be integrated under a single multimedia operating system and has to adopt a standardized storage and data interchange format.

In addition, the software must have a friendly, human-like mouthpiece that deals with us in our natural, spoken language and is not only user-friendly but also user-forgiving. The software will have to fill in the gaps in people's commands, correct their typos and misspellings, not let them make any serious mistakes, hold their hand as they work their way through a task, and anticipate what they will want to do next.

Most important of all, a mass-market home computer will require a reliable, universal communications network that links the digital utility center into very-high-speed satellite channels that support two-way instantaneous transmission of voices, music, video images, computer-generated pictures, text, and numerical data. This network, too, must be standardized, instantly available at the push of a CALL button on the digital utility center, and invisible to the user.

Only when such a network is in place will the digital utility center become popular with a majority of consumers. Only then will all the pie-in-the-sky promises of computer enthusiasts become possible.

Such a network will make it possible to do home banking, telecommuting, shopping at home, and attending courses and classes at home. People will be able to purchase all the new records, movies, computer software, and books over the network and have them downloaded into their local mass-storage device or into a portable computer that they can detach from the main unit and carry with them when they travel.

The Computer As Translator And Terminal

The lesson in all this is that our vision of the home computer has been too limited, and that's why we keep having false starts. Our vision has been limited by the fact that we are still too close to the computer's birth; we are still too familiar with the computer's early stages and functions to see what it may ultimately become.

We are only now beginning to move beyond the image of the computer as a computing engine that juggles numbers and processes paychecks. But we must go much further. We must see the computer as only a part of the digital revolution of all human media—voice, music, art, graphics, film, literature, and so on. As all science, art, technology, and communications are digitized, the computer assumes a central role as a translator among the media, and as a terminal linking human beings to the media and to each other.

The computer should enable the average person to enter information in any medium (pictures, voice, text, whatever) and instantly translate it (at the discretion of the person) into any other medium—or into several different media. It should then enable the person to send the package to any other person. Likewise, anyone who uses a computer should have instant access to all media in any format they wish.

This sounds extremely abstract, so picture the home computer of the future as the United Nations Building. It will have two major functions: translator and terminal. It will house all the disparate streams of digitized information representing all the different media, and it will translate them back and forth at the needs and whims of the user. And it will be plugged into the outside world (of cultures, peoples, nations, and institutions) and capable of vital two-way communication with that world in any language that is appropriate.

Next Month: Redefining Computer Literacy ©

CAPUTE!

Modifications Or Corrections To Previous Articles

Atari Chess

Atari owners who use OSS DOS 2.20 from Optimized Systems Software must first select Q from the DOS menu to quit to DOS XL, then select T to go to cartridge before attempting to load this game from the December 1984 issue (p. 99).

Atari Acrobat

Due to a printing error in line 2030 of this game from the February 1985 issue (p. 60), the STRIG function to read the joystick button appears as STPIG. Also, lines 20115, 23500, and 27035 are too long to type in as listed. To enter these lines, simply omit all spaces between BASIC statements and variables. For example, POKE BC,14 can be typed as POKEBC,14.

Machine Language Multiplication

In Part 2 of the series on multiplication in the "Machine Language" column (p. 121, February 1985 issue), the high and low bytes of the product are switched in the example program. The last few instructions of the example should read as follows:

```
        LDA   $0380
        ADC   $0382
        STA   $0380
        LDA   $0381
        ADC   #$00
        STA   $0381
NOADD   DEX
        BNE   NXBIT
```

Thanks to Karl Schmitt, Norman Sprock, and other readers who wrote in with the correction.

IBM Illegal Function Errors

A number of readers have reported problems with illegal function call errors in COMPUTE!'s graphics games for the IBM, such as "Horse Racing" (October 1984) and "Paratrooper" (January 1985). If you receive an Illegal Function Call error message in a line containing a PUT statement (such as line 1220 of Paratrooper), it most likely means that you have made a typing error in the DATA statements that define the graphics displayed by the PUT. When you see that error message in a line involving PUT, check all your DATA items carefully.

Proofreading The IBM Proofreader

Many readers have had problems getting the "IBM Automatic Proofreader" to work properly. The program is correct as listed, but if it's not typed in correctly, you may receive the cryptic message Error #2. The Proofreader traps all errors, even syntax errors. Instead of getting the usual "Syntax error in ..." message, you get the error number (2 is syntax error) with no hint as to where the error might be. To help you find your typos, change the 650 in line 140 to 0. This turns off the error trapping so you'll get the usual error messages if you have any errors.

Before using the Proofreader to type in programs, it's a good idea to test all the Proofreader commands, especially the SAVE command, just to make sure there are no bugs lurking in some obscure place in the program. To test the Proofreader's SAVE command, run the Proofreader and type in one line, say 10 REM. Now save this test program. If you didn't get an error message, you can safely type in a complete listing without fear of losing all your typing due to a bug in the SAVE command. When you think you have all the bugs out, type BASIC to exit the Proofreader, change line 140 back to normal, and save this bug-free version of the Proofreader.

David D. Thornburg, Associate Editor

Visual Computing, Part 1

In January 1984 Apple launched the Macintosh—a computer that would accelerate a revolution in computing that had already been gathering momentum for some time. This revolution was not in the computer hardware itself, although this certainly played a role. The revolution was in the way we communicate with our computational technology.

The Macintosh was the first low-cost personal computer to incorporate a primarily pictorial user interface. Rather than having to deal with words and phrases to convey information or desires to the computer, you can select small images (icons) that represent the object with which you want to work. To edit a document with the word processor, for example, you simply place the cursor over the document (shown as a page with a label beneath it) using a pointing device called a mouse. Once the cursor is over the document, two clicks of the mouse is all that's needed to load the document (and the word processor!) into the computer.

The difference between loading a program or text file in this fashion and loading it in by typing commands from the keyboard is subtle. To understand the nature of this difference, and why the visual interface appeals to some users and not to others, we need to explore different ways that people "think."

David D. Thornburg feels comfortable working across the text-picture boundary, and has written a dozen books on computing, including the KoalaPad Book *(Addison-Wesley) and* 101 Ways to Use a Macintosh *(Random House). His most recent book,* Beyond Turtle Graphics, *describes the nongraphics aspects of the computer language Logo. This book is an introduction to artificial intelligence and will be available soon from Addison-Wesley. Thornburg is currently working on his first novel.*

The Two Brains

Several years ago it was in vogue to think of human thinking style as being lateralized to the two hemispheres of the brain. Thinking that takes place in the left hemisphere is linear and analytical. Thinking that takes place in the right hemisphere is parallel, visual, and creative. This model of mental activity became so popular that we found ourselves referring to artists as "right-brained" people and to analytical thinkers as being "left-brained."

In fact, we all have the ability to think with both sides of our brain—to be both analytical and to be creative—to think linearly and in parallel. It is true that many of us spend more time in one mode of thought than the other. It is also true that our society seems to develop and encourage our analytical linear thinking at the expense of our creative mind. But it is both unfair and inaccurate to suggest that any individual is purely "left-brained" or "right-brained."

When interactive computer systems were first developed for mass production, it was decided that people should communicate with these machines through the typewriter keyboard and that the computer should respond primarily through a text-based display. Interestingly, the dedicated videogame computers that were being developed at the same time chose to use non-keyboard devices such as joysticks and game paddles instead of the keyboard, and to produce colorful graphic images rather than text displays.

Anyone who remembers the fads of the late 1970s will recall that videogame consoles outsold personal computers many times over. This extremely high ratio of game to computer sales was not based on price alone. The fact was that purchasers of game machines knew exactly what to do with them as soon as they were plugged in. The videogame was extremely easy to use—intuitively easy, perhaps.

Nothing Automatic

Personal computers, on the other hand, seemed designed for the linear analytical mode of thought. Nothing happened automatically—the keyboard had to be used for everything, including loading a program in the first place.

For example, suppose we look at the process of starting a game with the Atari 2600 Video Computer System and with the Commodore 64 computer. In the case of the Atari game machine, one needs only to insert the game cartridge and switch on the power. While this same process applies to the Commodore 64 with cartridge games, the story is quite different when the program is provided on disk. You then must enter:

```
LOAD "*",8
RUN
```

to get the game into the computer.

This difference in the user interface has nothing to do with technology differences between the two machines. The fact that the Commodore 64 has more RAM, or a disk drive, or can be used with thousands of different programs, is not the issue. In fact, most personal computer users expect to have to type strings of textual information into their computer to make it do something useful.

Mainly The Keyboard

For those of us who have used computers for a long time, none of this represents any hardship—it is simply "how things are done." Of course we are happy when the interface is simplified. Almost all Apple II owners, for example, equip their computers with "autostart ROMs" that will let a program boot from the disk automatically when the computer is turned on.

But still, the keyboard has maintained its role as the primary communication tool, even when the information to be communicated is nontextual.

This restriction in interface technology has kept many people from using computers. A major typing tutor program was promoted with the slogan "If you can't type you can't compute." For the vast majority of potential computer users in the world, this amounts to disfranchisement.

Fortunately, the slogan was wrong. Typing has nothing whatsoever to do with computing. All that is needed is a variety of communication tools across the man-machine interface to make computers accessible to any who would want to use them.

What made the Macintosh different was that it provided another type of interface—one that was primarily visual rather than textual.

A Step Back?

Of course, there are critics who would argue that the visual interface is a giant step backwards—that we gave up iconographic writing many years ago in favor of building words from an alphabet of letters. These same people might argue that those cultures whose language is still recorded in iconographic form are burdened with a cumbersome writing system that has hampered their development.

The visual computer interface has nothing to do with how we write. I am not arguing that we should do away with our alphabet or with words or with writing. I am not suggesting that we should use nothing but pictures in our next letter to Aunt Elsinore. What I am suggesting is that, when we are referring to the operations to be performed by a computer, it is only a matter of convention that we refer to these operations in written form. The convention to build programming languages from a vocabulary of English words was completely arbitrary. It was done, in part, because computer systems were provided with keyboards.

In fact, the first computer programs devised by Lady Lovelace for Babbage's Analytical Engine were patterns of holes in punched cards.

Any Symbols Will Do

Because most of us don't think of programming as a nontextual activity, it is hard for us to realize that one can communicate information to a computer in many different ways. A computer is, after all, just a symbol manipulation tool. The use of letters and numbers as symbols is arbitrary—it could work as easily with any other symbols we may devise.

The reason for exploring this topic at all is simple: Without being consciously aware of it, we have been overtaken by symbolic nontextual programming languages and have embraced them wholeheartedly. We have, in fact, become a nation of programmers without knowing it.

Anyone who builds a new level of *Lode Runner,* designs a new game with *Pinball Construction Set,* creates a new spreadsheet with *Multiplan,* or who works with any of the myriad construction set systems that represent one of the best-selling classes of software that has ever existed, is, in fact, creating computer programs with a minimum of typing. In fact, many of these programs are created by people with no typing whatsoever.

So, it is mildly amusing to hear many of these same construction set users suggest that programming is a "typing" activity.

Free Choice

Again, it is not typing that is the issue. I will argue that the nature of our communication medium determines the nature of the ideas we communicate. Some of us express ourselves quite well in linear textual form, and others of us are more comfortable with pictures and diagrams. There is nothing wrong with either approach to expression. What is important is that our technology has advanced to the point where people are free to choose their communication form, and even to switch back and forth between the two if they so desire. Any choice between the two has to be based on personal preference, not on the assumption that there is one "right" way to communicate.

Judging from the popularity of the visual interface (there is even a version of a Macintosh-like graphics program available for the PCjr!), the development of visual interfaces is opening up computer access to many thousands of people who would never have otherwise been interested in using this technology.

But, just because this new communication mode has been made available to the general public, this is no reason to think that we already know all of its consequences. As I gaze into my cloudy crystal ball, I see a future in which much of our programming will be done without the labor of typing—where we will write programs by constructing flow charts that indicate graphically what it is we want the computer to do for us.

These visual programming environments will let us express a goal without also requiring that we tell the computer how to achieve that goal.

Next month we will explore a visual programming environment in depth and compare it to text-based programming. Our visual programming language will be the database language HELIX, developed by Odesta for the Macintosh. ©

IBM Personal Computing

Donald B. Trivette

Spreadsheets For The Home

Remember when you were growing up and your pals used a word you'd never heard before? Were you too embarrassed to ask for a definition—to admit you didn't know what they were talking about (and maybe even doing)? Did you fake it as best you could?

Now that you're an adult, are you still faking? Do you really know what a *spreadsheet* program is? Don't be embarrassed. There are lots of well-adjusted, computer-literate people who have only a vague notion of what spreadsheet software is all about. You may have thought that spreadsheets were something only an accountant could appreciate and understand. Not true. Although spreadsheets *were* born of the accounting world, they have dozens of uses for those of us who have trouble balancing a checkbook. Yes, spreadsheets can actually be fun. First we'll look at their fascinating history, then at a typical numerical spreadsheet, and finally at some unusual nonnumerical applications.

Let There Be *VisiCalc*

It can be argued that the personal computer era really began with the invention of spreadsheet software. Before then, a few personal computers were around, but most were owned and used by hobbyists and tinkerers. In general, personal computer software was primitive in those days—back in the late 1970s.

It was in 1978 that Dan Bricklin was sitting in a classroom at Harvard Business School watching his professor laboriously create a model budget on the blackboard. Every time the professor changed a number in one column, all the related numbers in the other columns had to be recalculated and changed, too. (This is a familiar concept to those who adjust their income tax returns until they fall into the lowest possible tax bracket.)

Suddenly, in a flash (lightning striking and all that), Bricklin imagined an *electronic* blackboard that would, when one number was changed, automatically recalculate all the other numbers derived from it. Was such a thing possible? Bricklin didn't know, but he took the idea to his neighbor and friend Robert Frankston. Frankston, an experienced computer programmer and designer, was at first reluctant but finally agreed to pursue the project along with Dan Fylstra, a fledgling software publisher. Thus was born *VisiCalc*, the visible calculator.

The first *VisiCalc* program was sold in January 1979; it ran on a 24K RAM Apple II computer. The *VisiCalc* program was so useful that it helped sell Apple computers, and Apple in turn promoted *VisiCalc*—software that in essence turned a computer screen into an electronic blackboard for budget planning, financial forecasting, and virtually any task involving columns and rows of data. Nothing succeeds like a best-selling computer program, and it wasn't long before a dozen other companies were marketing spreadsheet programs, too. (Mercifully some are no longer with us.)

Today, there are spreadsheet programs for virtually every business, personal, and home computer. *VisiCalc* lives on in a much improved version that is available for several machines; *Multiplan* is another favorite; and *Lotus 1-2-3*, an integrated package that includes a spreadsheet, is one of the most popular computer programs of all time.

A Screenful Of Cells

A look at an actual spreadsheet application will help you grasp what Bricklin hath wrought. All spreadsheet programs start with a screen that looks like the blank spreadsheet in Figure 1. The columns, across the top, are lettered; the rows, down the edge, are numbered. The maximum size of the spreadsheet—the number of rows and columns—depends on the program and the amount of memory in the computer.

Each combination of a row and column forms a *cell* or box where data may be entered. Thus, the upper-left cell is referred to as A1—column A, row 1. The current cell—the place the computer will put the data when you type—is usually shown as a white box. That box is the spreadsheet's cursor. Just like a regular cursor, it can be moved up and down, left and right, by the arrow keys on the computer's keyboard.

Figure 1: Typical Spreadsheet Layout

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						

Let's create a supersimple spreadsheet for a make-believe company. An entry in a spreadsheet cell may be one of three types: characters, numbers, or a formula which the program will turn into numbers. By typing characters in some cells, you can create headings. In other cells—B3, B4, B5, B7—we'll put numbers to represent sales. As always when entering numbers in a computer program, omit the commas. (See Figure 2.)

To get the subtotal for divisional sales, you don't add the numbers manually and enter the result. That would defeat the purpose of the spreadsheet. Instead, you tell the computer to do it for you—to always add up column B, row 3, row 4, and row 5 and then put the total in column B, row 6. You do that by typing the *formula*—instead of a *number*—directly in the cell. The exact format differs slightly from one spreadsheet program to another, but generally you'd type B3+B4+B5 in cell B6. That is, B6 is always the sum of B3, B4, and B5. Makes sense, doesn't it?

Figure 2: A Sample Spreadsheet

	A	B	C
1		January	February
2	Sales		
3	Division 1	1000.00	
4	Division 2	1400.00	
5	Division 3	5000.00	
6	Subtotal	7400.00	
7	Mail Order	1200.00	
8	Grand Total	8600.00	

Figure 2 doesn't show the formula in cell B6—it only shows the value that the formula has calculated. The actual formula for any cell may be displayed at the top of the spreadsheet, but it is invisible in the spreadsheet itself. A similar formula is entered in cell B8 for the grand total.

Next, let's say you want to estimate the sales for February through December. Just enter a formula in the February cell C3. We'll project that each of the three divisions will sell a half-percent more than in the previous month. For example, C3 will be B3 multiplied by 1.005. There's a way to enter a formula so that it's automatically repeated for every month that remains in the year. And there's a way to copy a formula from one row to another, so only a few keystrokes are needed to generate the spreadsheet in Figure 3.

Figure 3: Projecting Sales With A Spreadsheet

	A	B	C
1		January	February
2	Sales		
3	Division 1	1000.00	1005.00
4	Division 2	1400.00	1407.00
5	Division 3	5000.00	5025.00
6	Subtotal	7400.00	7437.00
7	Mail Order	1200.00	1206.00
8	Grand Total	8600.00	8643.00

Spreadsheets As Big As Bedsheets

There's more to most spreadsheets than can be shown on a screen. In our sample, the columns for March through December will scroll into view when we move the cursor to the right side of the screen; similarly, rows below the "Grand Total" label can be scrolled into view. The screen is just a window onto a portion of the spreadsheet.

Most spreadsheet programs have commands to delete and insert rows and columns, to move entire rows and columns to other locations, to make hardcopy printouts, and to save the spreadsheet on disk.

Now, here's what makes spreadsheets so wonderful: To see how the numbers change when Division 2 sales increase, all you have to do is move the cursor to B4 and enter a new number. Instantly, the subtotal in B6 and the grand total in B8 are recalculated and replaced in the spreadsheet. Since a change in B4 alters some of the numbers for February through December, the spreadsheet automatically recalculates those values, too.

This is a typical numerical spreadsheet. But spreadsheets can also be useful and fun even for those who hate math. I know one woman who uses a spreadsheet to record her family tree. Each cell represents one of her ancestors; each column is a generation. Her spreadsheet has no formulas or mathematical calculations—just lots of names and dates. (See Figure 4.)

Some folks use spreadsheets in place of word processing programs. My architect friend uses *Lotus 1-2-3* to compose and print the schedules on his blueprints. He says it's much faster and easier than using a word processor. Once the schedule is entered, he prints it on a transparent film instead of paper and then sticks it to the drawing. He does schedules for doors, hardware, electrical fixtures, plumbing fixtures, and even shrubbery. Figure 5 shows part of a shrubbery schedule.

You could use a schedule like this to keep track of when you fertilized your plants, and what kind of fertilizer you used. Or to keep an inventory of your wine cellar. Or to record the expiration dates of your magazine subscriptions (especially if, like me, you think the magazines are always wrong). In fact, spreadsheet programs are ideal for any situation where you need to organize and record data in lists or tables.

Figure 4: Family Tree Spreadsheet

	A	B	C
1			
2		Helen (1840–1841)	
3			
4	John Smith (1810–1880)	John, Jr. (1850–1865)	
5	m. Mary Jones (1820–1860)		
6		Tim (1860–1930)	John Mason (1885–1914)
7		m. Betty Mason (1865–1925)	m. Sara Penny (1890–1964)

Figure 5: Shrubbery-Scheduling Spreadsheet

	A	B	C	D
1	Plant Name	Quantity	Height	Remarks
2	Pittosporum tobria	24	6′ 0″	Plant on centers shown
3	Juniperus conferta	30	12′ 15″	Plant on 3′ centers
4	Yeddo-Hawthorne	10	6′ 0″	Furnished by owner
5	Yucca gloriosa	15	24′ 30″	Transplant from corner

©

MLX Machine Language Entry Program For VIC-20

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost fail-safe entry of machine language programs published in COMPUTE!. You need to know nothing about machine language to use MLX—it was designed for everyone. At least 8K expansion memory is required.

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

Using MLX

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

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

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

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

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

MLX Commands

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

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

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

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

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

VIC MLX: Machine Language Entry

```
100 PRINT"{CLR}{PUR}";CHR$(142);CHR$(8);
                                    :rem 181
101 POKE 788,194:REM DISABLE RUN/STOP
                                    :rem 174
110 PRINT"{RVS}{14 SPACES}"         :rem 117
120 PRINT"{RVS} {RIGHT}{OFF}Ɛ*Ǝ£{RVS}
    {RIGHT} {RIGHT}{2 SPACES}Ɛ*Ǝ{OFF}Ɛ*Ǝ
    £{RVS}£{RVS} "                :rem 191
130 PRINT"{RVS} {RIGHT} ƐGƎ{RIGHT}
    {2 RIGHT} {OFF}£{RVS}£Ɛ*Ǝ{OFF}Ɛ*Ǝ
    {RVS} "                         :rem 232
140 PRINT"{RVS}{14 SPACES}"         :rem 120
```

```
200 PRINT"{2 DOWN}{PUR}{BLK}A FAILSAFE MA
    CHINE":PRINT"LANGUAGE EDITOR{5 DOWN}"
                                  :rem 141
210 PRINT"{BLK}{3 UP}STARTING ADDRESS":IN
    PUTS:F=1-F:C$=CHR$(31+119*F)  :rem 97
220 IFS<256ORS>32767THENGOSUB3000:GOTO210
                                  :rem 2
225 PRINT:PRINT:PRINT:PRINT       :rem 123
230 PRINT"{BLK}{3 UP}ENDING ADDRESS":INPU
    TE:F=1-F:C$=CHR$(31+119*F)    :rem 158
240 IFE<256ORE>32767THENGOSUB3000:GOTO230
                                  :rem 234
250 IFE<STHENPRINTC$;"{RVS}ENDING < START
    {2 SPACES}":GOSUB1000:GOTO 230 :rem 176
260 PRINT:PRINT:PRINT             :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S    :rem 56
310 PRINTRIGHT$("0000"+MID$(STR$(AD),2),5
    );":";:FORJ=1TO6              :rem 234
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                  :rem 228
390 IFN=-211THEN 710              :rem 62
400 IFN=-204THEN 790              :rem 64
410 IFN=-206THENPRINT:INPUT"{DOWN}ENTER N
    EW ADDRESS";ZZ                :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}OUT OF RANGE":GOSUB1000:GOTO410
                                  :rem 225
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                  :rem 238
420 IF N<>-196 THEN 480           :rem 133
430 PRINT:INPUT"DISPLAY:FROM";F:PRINT,"TO
    ";:INPUTT                     :rem 234
440 IFF<SORF>EORT<SORT>ETHENPRINT"AT LEAS
    T";S;"{LEFT}, NOT MORE THAN";E:GOTO43
    0                             :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";    :rem 30
455 FORK=0TO5:N=PEEK(I+K):IFK=3THENPRINTS
    PC(10);                       :rem 34
457 PRINTRIGHT$("00"+MID$(STR$(N),2),3);"
    ,";                           :rem 157
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                  :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310                     :rem 50
480 IFN<0 THEN PRINT:GOTO310      :rem 168
490 A(J)=N:NEXTJ                  :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT   :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20)
                                  :rem 234
515 IFN=CKSUMTHEN530              :rem 255
520 PRINT:PRINT"LINE ENTERED WRONG":PRINT
    "RE-ENTER":PRINT:GOSUB1000:GOTO310
                                  :rem 129
530 GOSUB2000                     :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:rem 80
550 AD=AD+6:IF AD<E THEN 310      :rem 212
560 GOTO 710                      :rem 108
570 N=0:Z=0                       :rem 88
580 PRINT"{+}";                   :rem 79
581 GETA$:IFA$=""THEN581          :rem 95
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN670                 :rem 229
590 IFA>128THENN=-A:RETURN        :rem 137
600 IFA<>20 THEN 630              :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    {LEFT} {LEFT}";:GOTO690       :rem 172
620 GOTO570                       :rem 109
630 IFA<48ORA>57THEN580           :rem 105
640 PRINTA$;:N=N*10+A-48          :rem 106
650 IFN>255 THEN A=20:GOSUB1000:GOTO600
                                  :rem 229
660 Z=Z+1:IFZ<3THEN580            :rem 71
670 IFZ=0THENGOSUB1000:GOTO570    :rem 114
680 PRINT",";:RETURN              :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                  :rem 149
692 FORI=1TO3:T=PEEK(S%-I)        :rem 68
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                  :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1);:RETURN
                                  :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***{3 DOWN}"
                                  :rem 236
720 INPUT"{DOWN} FILENAME";F$     :rem 228
730 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK: (T/D)"        :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"THEN740
                                  :rem 36
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                  :rem 158
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256              :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                  :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                            :rem 69
765 POKE254,S/256:POKE253,S-PEEK(254)*256
    :POKE780,253                  :rem 12
766 POKE782,E/256:POKE781,E-PEEK(782)*256
    :SYS65496                     :rem 124
770 IF(PEEK(783)AND1)OR(ST AND191)THEN780
                                  :rem 111
775 PRINT"{DOWN}DONE.":END        :rem 106
780 PRINT"{DOWN}ERROR ON SAVE.{2 SPACES}T
    RY AGAIN.":IFDV=1THEN720      :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO720          :rem 103
782 GOTO720                       :rem 115
790 PRINT"{CLR}{RVS}*** LOAD ***{2 DOWN}"
                                  :rem 212
800 INPUT"{2 DOWN} FILENAME";F$   :rem 244
810 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK: (T/D)"        :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"THEN820
                                  :rem 34
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                  :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256              :rem 2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                  :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                            :rem 70
850 POKE780,0:SYS65493            :rem 11
860 IF(PEEK(783)AND1)OR(ST AND191)THEN870
                                  :rem 111
865 PRINT"{DOWN}DONE.":GOTO310    :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.{2 SPACES}T
    RY AGAIN.{DOWN}":IFDV=1THEN800
                                  :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO800          :rem 102
1000 REM BUZZER                   :rem 135
1001 POKE36878,15:POKE36874,190   :rem 206
1002 FORW=1TO300:NEXTW            :rem 117
1003 POKE36878,0:POKE36874,0:RETURN:rem 74
2000 REM BELL SOUND               :rem 78
2001 FORW=15TO0STEP-1:POKE36878,W:POKE368
     76,240:NEXTW                 :rem 22
2002 POKE36876,0:RETURN           :rem 119
3000 PRINTC$;"{RVS}NOT ZERO PAGE OR ROM":
     GOTO1000                     :rem 89
```

COMPUTE!'s Guide To Typing In Programs

Before typing in any program, you should familiarize yourself with your computer. Learn how to use the keyboard to type in and correct BASIC programs. Read your manuals to understand how to save and load BASIC programs to and from your disk drive or cassette unit. Computers are precise—take special care to type the program *exactly* as listed, including any necessary punctuation and symbols. To help you with this task, we have implemented a special listing convention as well as a program to help check your typing—the "Automatic Proofreader." Please read the following notes before typing in any programs from COMPUTE!. They can save you a lot of time and trouble.

Since programs can contain some hard-to-read (and hard-to-type) special characters, we have developed a listing system that spells out in abbreviated form the function of these control characters. You will find these special characters within curly braces. For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. A symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CONTROL key and press A. Commodore machines have a special control key labeled with the Commodore logo. Graphics characters entered with the Commodore logo key are enclosed in a new kind of special bracket. A graphics character can be listed as [<A>]. In this case, hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S̲. One exception is {SHIFT-SPACE}. Hold down SHIFT and press the space bar.

If a number precedes a symbol, such as {5 RIGHT}, {6 S̲}, or [<8 Q>], you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (printed in white on black) should be entered with the Atari logo key. Since spacing is sometimes important, any more than two spaces will be listed, for example, as: {6 SPACES}. A space is never left at the end of a line, but will be moved to the next printed line as {SPACE}. There are no special control characters found in our IBM PC/PCjr, TI-99/4A, and Apple program listings. For your convenience, we have prepared this quick-reference key for the Commodore and Atari special characters:

Atari 400/800/XL

When you see	Type	See	
{CLEAR}	ESC SHIFT <		Clear Screen
{UP}	ESC CTRL -	↑	Cursor Up
{DOWN}	ESC CTRL =	↓	Cursor Down
{LEFT}	ESC CTRL +	←	Cursor Left
{RIGHT}	ESC CTRL *	→	Cursor Right
{BACK S}	ESC DELETE		Backspace
{DELETE}	ESC CTRL DELETE		Delete character
{INSERT}	ESC CTRL INSERT		Insert character
{DEL LINE}	ESC SHIFT DELETE		Delete line
{INS LINE}	ESC SHIFT INSERT		Insert line
{TAB}	ESC TAB		TAB key
{CLR TAB}	ESC CTRL TAB		Clear tab
{SET TAB}	ESC SHIFT TAB		Set tab stop
{BELL}	ESC CTRL 2		Ring buzzer
{ESC}	ESC ESC		ESCape key

Commodore PET/CBM/VIC/64

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

The Automatic Proofreader

Also, we have developed a simple, yet effective program that can help check your typing. Type in the appropriate Proofreader program for your machine, then save it for future use. On the VIC, 64, or Atari, run the Proofreader to activate it, then enter NEW to erase the BASIC loader (the Proofreader will still be active, hidden in memory, as a machine language program). Pressing RUN/STOP–RESTORE or SYSTEM RESET deactivates the Proofreader. You can use SYS 886 to reactivate the VIC/64 Proofreader, or PRINT USR(1536) to reenable the Atari Proofreader. The IBM Proofreader is a BASIC program that lets you enter, edit, list, save, and load programs that you type. It simulates the IBM's BASIC line editor.

Using The Automatic Proofreader

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a number (on the Commodore) or a pair of letters

(Atari or IBM) appears. The number or pair of letters is called a *checksum*. Try making a change in the line, and notice how the checksum changes.

All you need to do is compare the value provided by the Proofreader with the checksum printed in the program listing in the magazine. In Commodore listings, the checksum is a number from 0 to 255. It is set off from the rest of the line with *rem*. This prevents a syntax error if the checksum is typed in, but the REM statements and checksums need *not* be typed in. It is just there for your information.

In Atari and IBM listings, the checksum is given to the left of each line number. Just type in the program, a line at a time (without the printed checksum) and compare the checksum generated by the Proofreader to the checksum in the listing. If they match, go on to the next line. If not, check your typing: You've made a mistake. On the Commodore and Atari Proofreader, spaces are not counted as part of the checksum, and no check is made to see that you've typed in the characters in the right order. If characters are transposed, the checksum will still match the listing. Because of the checksum method used, do not use abbreviations, such as ? for PRINT. However, the Proofreader does catch the majority of typing errors most people make. The IBM Proofreader is even pickier; it *will* detect errors in spacing and transposition. Also, be sure you leave Caps Lock on, except when you need to enter lowercase characters.

Special Proofreader Notes For Commodore Cassette Users

The Proofreader resides in the cassette buffer, which is used during tape LOADs and SAVEs. Be sure to press RUN/STOP–RESTORE before you save or load a program, to get the Proofreader out of the way. If you want to use the Proofreader with tape, run the Proofreader, then enter these two lines *exactly* as shown, pressing RETURN after each one:

```
A$="PROOFREADER.T":B$="{10 SPACES}"
  :FORX=1TO4:A$=A$+B$:NEXT
FORX=886TO1018:A$=A$+CHR$(PEEK(X))
  :NEXT:OPEN 1,1,1,A$:CLOSE1
```

Then press RECORD and PLAY on a blank tape, and a special version of the Proofreader will be saved to tape. Anytime you need to reload the Proofreader after it has been erased, just rewind the tape, type OPEN1:CLOSE1, then press PLAY. When READY comes back, enter SYS 886.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you enter NEW, the Proofreader will prompt you to press Y to be especially sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program into the normal BASIC environment (this will replace the Proofreader in memory). You can now run the program, but you may want to resave it to disk. This will shorten it on disk and make it load faster, but it can no longer be edited with the Proofreader. If you want to convert a program to Proofreader format, save it to disk with SAVE "filename",A.

VIC/64 Proofreader

```
100 PRINT"{CLR}PLEASE WAIT...":FORI=886TO10
    18:READA:CK=CK+A:POKEI,A:NEXT
110 IF CK<>17539 THEN PRINT"{DOWN}YOU MADE
    {SPACE}AN ERROR":PRINT"IN DATA STATEMEN
    TS.":END
120 SYS886:PRINT"{CLR}{2 DOWN}PROOFREADER A
    CTIVATED.":NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 DATA 150,141,036,003,169,003
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251
958 DATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003
```

Atari Proofreader

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:READ A:POKE I
    ,A:CK=CK+A:NEXT I
120 IF CK<>19072 THEN ? "Error in DA
    TA Statements.  Check Typing.":E
    ND
130 A=USR(1536)
140 ? :? "Automatic Proofreader Now
    Activated."
```

```
150 END
1536 DATA 104,160,0,185,26,3
1542 DATA 201,69,240,7,200,200
1548 DATA 192,34,208,243,96,200
1554 DATA 169,74,153,26,3,200
1560 DATA 169,6,153,26,3,162
1566 DATA 0,189,0,228,157,74
1572 DATA 6,232,224,16,208,245
1578 DATA 169,93,141,78,6,169
1584 DATA 6,141,79,6,24,173
1590 DATA 4,228,105,1,141,95
1596 DATA 6,173,5,228,105,0
1602 DATA 141,96,6,169,0,133
1608 DATA 203,96,247,238,125,241
1614 DATA 93,6,244,241,115,241
1620 DATA 124,241,76,205,238,0
1626 DATA 0,0,0,0,32,62
1632 DATA 246,8,201,155,240,13
1638 DATA 201,32,240,7,72,24
1644 DATA 101,203,133,203,104,40
1650 DATA 96,72,152,72,138,72
1656 DATA 160,0,169,128,145,88
1662 DATA 200,192,40,208,249,165
1668 DATA 203,74,74,74,74,24
1674 DATA 105,161,160,3,145,88
1680 DATA 165,203,41,15,24,105
1686 DATA 161,200,145,88,169,0
1692 DATA 133,203,104,170,104,168
1698 DATA 104,40,96
```

IBM Proofreader

```
10 'Automatic Proofreader Version 2.00 (Lines 270,510,515,517,620,630 changed from V1.0)
100 DIM L$(500),LNUM(500):COLOR 0,7,7:KEY OFF:CLS:MAX=0:LNUM(0)=65536!
110 ON ERROR GOTO 120:KEY 15,CHR$(4)+CHR$(70):ON KEY(15) GOSUB 640:KEY (15) ON:GOTO 130
120 RESUME 130
130 DEF SEG=&H40:W=PEEK(&H4A)
140 ON ERROR GOTO 650:PRINT:PRINT"Proofreader Ready."
150 LINE INPUT L$:Y=CSRLIN-INT(LEN(L$)/W)-1:LOCATE Y,1
160 DEF SEG=0:POKE 1050,30:POKE 1052,34:POKE 1054,0:POKE 1055,79:POKE 1056,13:POKE 1057,28:LINE INPUT L$:DEF SEG:IF L$="" THEN 150
170 IF LEFT$(L$,1)=" " THEN L$=MID$(L$,2):GOTO 170
180 IF VAL(LEFT$(L$,2))=0 AND MID$(L$,3,1)=" " THEN L$=MID$(L$,4)
190 LNUM=VAL(L$):TEXT$=MID$(L$,LEN(STR$(LNUM))+1)
200 IF ASC(L$)>57 THEN 260 'no line number, therefore command
210 IF TEXT$="" THEN GOSUB 540:IF LNUM=LNUM(P) THEN GOSUB 560:GOTO 150 ELSE 150
220 CKSUM=0:FOR I=1 TO LEN(L$):CKSUM=(CKSUM+ASC(MID$(L$,I))*I) AND 255:NEXT:LOCATE Y,1:PRINT CHR$(65+CKSUM/16)+CHR$(65+(CKSUM AND 15))+" "+L$
230 GOSUB 540:IF LNUM(P)=LNUM THEN L$(P)=TEXT$:GOTO 150 'replace line
240 GOSUB 580:GOTO 150 'insert the line
260 TEXT$="":FOR I=1 TO LEN(L$):A=ASC(MID$(L$,I)):TEXT$=TEXT$+CHR$(A+32*(A>96 AND A<123)):NEXT
270 DELIMITER=INSTR(TEXT$," "):COMMAND$=TEXT$:ARG$="":IF DELIMITER THEN COMMAND$=LEFT$(TEXT$,DELIMITER-1):ARG$=MID$(TEXT$,DELIMITER+1) ELSE DELIMITER=INSTR(TEXT$,CHR$(34)):IF DELIMITER THEN COMMAND$=LEFT$(TEXT$,DELIMITER-1):ARG$=MID$(TEXT$,DELIMITER)
280 IF COMMAND$<>"LIST" THEN 410
290 OPEN "scrn:" FOR OUTPUT AS #1
300 IF ARG$="" THEN FIRST=0:P=MAX-1:GOTO 340
310 DELIMITER=INSTR(ARG$,"-"):IF DELIMITER=0 THEN LNUM=VAL(ARG$):GOSUB 540:FIRST=P:GOTO 340
320 FIRST=VAL(LEFT$(ARG$,DELIMITER)):LAST=VAL(MID$(ARG$,DELIMITER+1))
330 LNUM=FIRST:GOSUB 540:FIRST=P:LNUM=LAST:GOSUB 540:IF P=0 THEN P=MAX-1
340 FOR X=FIRST TO P:N$=MID$(STR$(LNUM(X)),2)+" "
350 IF CKFLAG=0 THEN A$="":GOTO 370
360 CKSUM=0:A$=N$+L$(X):FOR I=1 TO LEN(A$):CKSUM=(CKSUM+ASC(MID$(A$,I))*I) AND 255:NEXT:A$=CHR$(65+CKSUM/16)+CHR$(65+(CKSUM AND 15))+" "
370 PRINT #1,A$+N$+L$(X)
380 IF INKEY$<>"" THEN X=P
390 NEXT :CLOSE #1:CKFLAG=0
400 GOTO 130
410 IF COMMAND$="LLIST" THEN OPEN "lpt1:" FOR OUTPUT AS #1:GOTO 300
420 IF COMMAND$="CHECK" THEN CKFLAG=1:GOTO 290
430 IF COMMAND$<>"SAVE" THEN 450
440 GOSUB 600:OPEN ARG$ FOR OUTPUT AS #1:ARG$="":GOTO 300
450 IF COMMAND$<>"LOAD" THEN 490
460 GOSUB 600:OPEN ARG$ FOR INPUT AS #1:MAX=0:P=0
470 WHILE NOT EOF(1):LINE INPUT #1,L$:LNUM(P)=VAL(L$):L$(P)=MID$(L$,LEN(STR$(VAL(L$)))+1):P=P+1:WEND
480 MAX=P:CLOSE #1:GOTO 130
490 IF COMMAND$="NEW" THEN INPUT "Erase program - Are you sure";L$:IF LEFT$(L$,1)="y" OR LEFT$(L$,1)="Y" THEN MAX=0:GOTO 130:ELSE 130
500 IF COMMAND$="BASIC" THEN COLOR 7,0,0:ON ERROR GOTO 0:CLS:END
510 IF COMMAND$<>"FILES" THEN 520
515 IF ARG$="" THEN ARG$="A:" ELSE SEL=1:GOSUB 600
517 FILES ARG$:GOTO 130
520 PRINT"Syntax error":GOTO 130
540 P=0:WHILE LNUM>LNUM(P) AND P<MAX:P=P+1:WEND:RETURN
560 MAX=MAX-1:FOR X=P TO MAX:LNUM(X)=LNUM(X+1):L$(X)=L$(X+1):NEXT:RETURN
580 MAX=MAX+1:FOR X=MAX TO P+1 STEP -1:LNUM(X)=LNUM(X-1):L$(X)=L$(X-1):NEXT:L$(P)=TEXT$:LNUM(P)=LNUM:RETURN
600 IF LEFT$(ARG$,1)<>CHR$(34) THEN 520 ELSE ARG$=MID$(ARG$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34) THEN ARG$=LEFT$(ARG$,LEN(ARG$)-1)
620 IF SEL=0 AND INSTR(ARG$,".")=0 THEN ARG$=ARG$+".BAS"
630 SEL=0:RETURN
640 CLOSE #1:CKFLAG=0:PRINT"Stopped.":RETURN 150
650 PRINT "Error #";ERR:RESUME 150
```

ATARI®

600XL .... CALL
800XL .... CALL

DISK DRIVES
Rana 1000 .... $239
Astra 2001 .... $549
Indus GT $298
Trak AT-D2 .... $329
Trak AT-1 .... $319
Trak AT-D4 .... Call
Astra 1620 (Dual) ... $499
Percom .... Call
Atari 1050 .... $249

MEMORIES
Microbits 64K (XL) .. $115
Mosaic 48K (400) $98
Mosaic 64K (400/800) Call
Mosaic 32K $68
Atari 64K (600XL) .... Call

INTERFACES
Axiom 846 .... Call
Ape Face .... Call
Atari 850 (In Stock) .. $169
Interfast 1 .... $150
Microbits 1150 ...... Call
R-Verter .... Call

DIRECT PRINTERS
Axiom AT-100 ...... $195
Atari 1027 .... $269
Axiom 550 AT $259
Axiom 700 AT $469
Atari 1025 .... Call

DIRECT MODEMS
Microbits 1000C $109

OTHER ATARI
400 Keyboard .... Call
Koala Pad $67
Chalkboard Pad $75
Bit-3 80 Column $228
ATR-8000 (64K) $489
ATR-8000 (16K) $359
Alien Voice Box $98
1010 Recorder $55

ATARI SOFTWARE

MISCELLANEOUS
Syn Calc (D) $48
Syn File (D) $48
Syn Trend (D) $48
Syn Com (D) $29
Syn Chron (D) $29
Decathlon (R) $29
Drols (D) $23
Gyruss (R) $31
Heist (D) $23
Bruce Lee (C/D) $27
Universe (D) .... Call
Questron (D) $34
Koala Coloring I $20
Koala Logo Design $27
Bumble Games (D) $27
Miles Accounting .... Call
World Gtst. Baseball .. $23
Gridrunner (R) $20
Sargon II (C/D) $23
Millionaire (D) $34
Castle Wolfenstein (D) $20
Odesta Chess (D) $46
Financial Wizard (D) .. $41
Ultima III (D) $39

ADVENTURE INT'L
Ultra Disassembler (D) $33
Diskey (D) $33
Adv. 1-12 (each) (C) $18
Saga 1-12 (each) (D) .. $27

ATARI
Atari Writer (R) $68
Paint (D) $30
Microsoft Basic II (R) . $64
Visicalc (D) $139
Home File Mgr (D) $36
Assembler Editor (R) . $44
Qix (R) $32
Dig Dug (R) $32
Atari Logo (R) $72
Ms. Pac Man (R) $33
Joust (R) $33
Donkey Kong Jr. (R) .. $35
Computer Chess (R) .. $24

AVALON HILL
Telengard . (C) $16 (D) $19
Close Assault (C) 20 (D) 23
TAC (D) $27

BRODERBUND
Arcade Machine (D) .. $39
Bank St. Writer (D) ... $43
Oper. Whirlwind (D) .. $27
Choplifter . (D) $23 (R) $29

CBS SOFTWARE .... Call

CONTINENTAL
Home Accountant (D) $44
Tax Advantage (D) ... $45

COUNTERPOINT SW Call

DATASOFT
Pooyan (C/D) $20
Teletalk (D) $33
Graphic Master (D) ... $23
Micropainter (D) $23
Zaxxon (C/D) $27

EASTERN HOUSE
Monkey Wrench II ... $51

EDUCATIONAL SW
Tricky 1,2,3 or 4 ...... $15
Tricky 5-13 .... $22

EPYX
Dragon Riders (C/D) .. $27
Temple APS (C/D) $27
Jumpman (C/D) $27

FIRST STAR
Boulder Dsh (C/D) 20 (R) 27
Bristles (C/D) $20
Flip Flop (C/D) $20

GAMESTAR
Football (C/D) $21
Baseball (C/D) $21

INFOCOM
Zork I, II or III (D) $27
Deadline (D) $34
Starcross (D) $27
Suspended (D) $34
Witness (D) $34
Planetfall (D) $34
Enchanter (D) $34
Infidel (D) $34

KRELL SAT .... Call

INTELL. STATEMENTS
Prof. Blackjack (D) ... $46

LJK
Letter Perfect (D) $58
Data Perfect (D) $74
Spell Perfect (D) $56
Letter Perfect (R) $74

MICROPROSE
Solo Flight (D) $23
Hellcat Ace (C/D) $20

MONARCH
ABC Compiler (D) $55

OPTIMIZED SYSTEMS
Action (R) $65
Basic XL (R) $65
Mac 65 (D) $58
C-65 (D) $58
Bug 65 (D) $23

PARKER BROS
Astrochase (R) $33
Death Star (R) $33
Q-Bert (R) $33
Popeye (R) $33

QUALITY SW
Return of Hercules (D) $22
Ali Baba (D) $22

RESTON
Moviemaker (D) $45

SCARBOROUGH SYS.
Mastertype (D/R) $25
Songwriter (D) $25

SCHOOL WIZWARE . Call

SIERRA ON-LINE
Homeword (D) $46
Ultima II (D) $39
Dark Crystal (D) $26
Wiz. & Princess (D) ... $22

SPINNAKER
Snooper Troop 1,2 (D) . $30
Most Amazing (D) $27
Kids on Keys (D) $20
Trains (D) $27
Delta Drawing (R) $27
Aerobics (D) $34

STRATEGIC SIM.
Broadsides (D) $27
Carrier Force (D) $39
Combat Leader (D) $27
Rails West (D) $27
Epidemic (D) $23
Eagles (D) $27
Cosmic Bal I or II (D) .. $27

SUBLOGIC
Flight Simulator II (D) . $36
Pinball (C/D) $20

SYNAPSE
File Manager (R) $54
Fort Apocalypse (C/D) $23
Dimension X (C/D) $23
Blue Max (C/D) $23
Encounter (D/R) $23
Zepplin (C/D) $23
Pharoah's Curse (C/D) $23

TRONIX
S.A.M. (D) $39
P.M. Animator (D) $29
Juice (C/D) $20
Chatterbee (D) $27

Printers/Etc.

DIABLO
630 Letter Qual. $1559

SILVER REED
EXP 400 Ltr. Qual. $288
EXP 500 Ltr. Qual. $355
EXP 550 Ltr. Qual. $419
EXP 770 Ltr. Qual. $849

STAR
Gemini 10X $239
Gemini 15X $355
Delta 10 .... $339
Delta 15 .... $449
Radix 10 .... $498
Radix 15 .... $588
Power Type $319

TOSHIBA
1340 .... $739
1351 .... $1249

NEC
3510 .... $1215
3530 .... $1215
3550 .... $1498
7710/7730 .... $1648

CITOH
Prowriter $309
Prowriter II $498
Starwriter $909
Printmaster $1189

OKIDATA
82A $298
84P $669
92 .... $368
93 .... $579

DAISYWRITER
2000 .... $985

MANNESMANN
160L $559
Spirit $267

JUKI
6100 .... $389
6300 .... Call

PANASONIC
1090 .... $228
1091 .... $288
1092 .... $439

MONITORS

AMDEK
V300 G $119
V300 A $139
V310 G (IBM) $155
V310 A (IBM) $159
Color I+ $269
Color II+ $399
Color III $349
Color IV (IBM) $699

PRINCETON GRAPHICS
MAX 12 (Amber) $178
HX 12 (RGB) $449
SR 12 (RGB) $595

SUPER 5
100A (Amber) $99
500G (IBM with tilt) .. $126
500A (IBM with tilt) .. $126

SAKATA
SC100 (Color) $239
1000G (Green) $99

TAXAN
100 Green $115
105 Amber $125
210 RGB/Composite $259
400 RGB Med-Res. ... $296
415 RGB Hi-Res. $429
420 RGB Hi-Res.(IBM) $449

NEC
JB 1260 (Grn) $99
JB 1201 (Grn) $145
JB 1205 (Amber) $145

ZENITH
Green $85
Amber $95

MODEMS

NOVATION
J-Cat $99
Apple Cat II $259
D-Cat $149

HAYES
Smartmodem $199
Smartmodem 1200 .... $469
Micromodem II $259
Micromodem IIe $225

PROMETHIUS
Promodem 1200 ...... $329

Incredible Prices

COSMIC COMPUTERS

727 BREA CANYON RD., SUITE 16
WALNUT, CA 91789

OPEN MON-SAT

(800) 626-7642

PLEASE FOR ORDERS ONLY
SORRY, NO COD'S

IN CALIF. **(714) 594-5204**
FOR TECHNICAL INFO, ORDER INQUIRIES,

Add $2.50 shipping per software order in continental U.S. Add $5.00 shipping per software order for AK, HI, FPO-APO. Add $10.00 or 15% (whichever is greater) per software order for non-U.S. Call for cost of hardware shipping. Calif. residents add 6½% sales tax. Cashiers checks or money orders filled within 24 hours for items in stock. Personal checks require 4 weeks to clear. MasterCard and Visa OK for software only within continental U.S., add 3% surcharge. Include card no., expiration date and signature. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call to obtain one before returning goods for replacement or repair. Prices & availability subject to change.

COMMODORE

INTERFACES
The Connection $85
Bus Card $138
Cardco G+ $65
Cardco B .... Call
MSD (IEEE) $98
Cardco 5 Slot $48
Grappler C D $109

DIRECT MODEMS
Hesmodem $53
1650 Automodem $99
1600 Modem .... Call
Westridge Modem .... Call

80 COLUMN BDS
Batteries 80 Col $138
Video Pak 80 .... $129

DISK DRIVES
MSD (170K) $309
MSD (Dual) (170Kx2) . $498
Indus GT .... Call
Laser (170K) $325
Commodore 1541 .... $239

RECORDERS
Cardco Recorder $48
1530 Commodore .... Call
Cassette Interface ... $29
Phonemark Rec $37

DIRECT PRINTERS
MPS 801 .... $219
Commodore 1526 .... $288
Cardco LQ/I $498
1520 Color Printer ... $129

64 SOFTWARE 64

MISCELLANEOUS
MAE Assembler (D) .. $47
VIP Terminal (D) $38
Star Wars (R) $33
Super Base 64 (D) .. $68
Doodle City (D) $27
Summer Games (D) .. $27
Pitfall II (D) $25
Decathlon (D) $25
50 Mission Crush (D) . $27
IFR. (C/D) $20
Master Composer (D) . $27
Donkey Kong (R) $29
Bruce Lee (D) $23
Pro Football Stat. (D) . $56
Seastalker (D) $27
Koala Coloring I $20
Koala Logo Design $27
Rockys Boots (D) $33
Bumble Games (D) ... $27
Beyond Wolfenstein .. $23
Peachtree Account .. Call
Odesta Chess (D) $46
Ultima III (D) $39
Prof. Blackjack (D) ... $46
Homeword (D) $46
Pers. Accountant (D) . $23
Karate Devils (D) $27
Final Flight (D) $22
Diskey (D) $33
Barrons Sat (D) $59
Millionaire (D) $39
Sargon II (D) $23
B-Graph (D) $59
Castle Wolfenstein (D) $20

ACCESS SOFTWARE
Beached (D) $24
Neutral Zone (C/D) .. $24
Spritemaster (D) $25

AVALON HILL
Nuke War (C) $12
Androm. Conquest (C) $14
Midway Campaign (C) $13
Computer Football (C) $13
Telengard . (C) $16 (D) $19
Flying Ace (C) $15
Moon Patrol (C) $17

BATTERIES INCLUDED
Paper Clip (D) $59
Consultant (D) $64
Paper Clip w/Spell (D) $79
Spell Pack (D) $34
Organizer Series (Ea) . $22

BLUE SKY
Calc Result Adv. $99
Calc Result Easy $57

BRODERBUND
Bank St. Writer (D) ... $43
Operat. Whrlwnd (D) .. $27
Choplifter (R) $27
Lode Runner . (D) 23 (R) 27

CBS SOFTWARE
Success with Math (D) $17
Wbstr Word Game (D) $20

COMMODORE
Simons Basic (R) $29
Magic Desk (R) $48
Logo (D) $50
Assembler 64 (D) $36
Easy Script 64 (D) ... $52

CONTINENTAL S.W.
Home Accountant (D) $44
Tax Advantage (D) ... $45
FCM (D) $34

COUNTER POINT SW
Early Games (Ea) $20

CREATIVE SOFTWARE
Moondust (R) $23
Save New York (R) $23
Pipes (R) $23
Househld Finance (C) $20

DATASOFT
Pooyan (C/D) $20
Moon Shuttle (D) $20

ENTECH
Studio 64 (C/D) $28
Database 64 (D) $45

EPYX
Temple of APS (C/D) .. $27
Jumpman (C/D) $27
Dragonriders (C/D) ... $27
Gateway to APS (R) .. $27

FUTURE HOUSE
Comp. Pers. Account. $56

HES
Omniwriter (D) $45
Hesmon (R) $27
64 Forth (R) $39
Multiplan (D) $65
Turtle Graphics (R) ... $39

INFOCOM
Planet Fall (D) $34
Enchanter (D) $34
Infidel (D) $34

JINSAM
Mini Jini (R) $75

MICROSOFTWARE INT'L
Practicalc 64 . (C) 34 (D) 36
Spreadsheet . (C) 49 (D) 52
Practifile 64 (D) $36

MIRAGE CONCEPTS
Data Base (D) $68
Word Processor (D) .. $68

MICROSPEC
Database (D) $44
Mailing List (D) $32
Checkbook Mgr (D) ... $47
G/L (D) $44
A/P (D) $47
A/R (D) $47

PARKER BROS
Q-Bert (R) $33
Popeye (R) $33
Frogger (R) $33

PROFESSIONAL SW
Wordpro W/Spell (D) .. $68
Spellright (D) $45

QUIKTEX
Quick Br. Fox (R) $49

RAINBOW
File Assistant (D) $46
Writers Assistant (D) . $46
Spreadsht Assist. (D) . $56

SCARBOROUGH
Mastertype (D/R) $27
Song Writer (D) $27

SOFTSMITH
Touch Typing (C/D) $21

SOUTHERN SOLUTIONS
Businessman (D) $48
Bill Payer (D) $48
Bill Collector (D) $48
Paymaster (D) $48

SPINNAKER
Snooper 1 or 2 (D) $27
Aerobics (D) $34
Kids on Keys (D) $20
Most Amazing (D) $27
Kindercomp (D) $20
Alphabet Zoo (D) 20
Trains (D) $27
Delta Drawing (R) $27

SUBLOGIC
Flight Simulator II (D) . $36
Pinball (C/D) $22

SYNAPSE
Ft. Apocalypse (C/D) . $23
Necromancer (C/D) ... $23
Zaxxon (D) $23
Blue Max (D) $23
Pharoah's Curse (C/D) $23

SSI
Combat Leader (C/D) . $27
Computer Baseball (D) $27
Eagles (D) $27
Ringside Seat (D) $27
Tigers In Snow (C/D) .. $27
Battle Normandy (C/D) $27

TIMEWORKS
Dungeons of Alg. (C/D) $17
Robbers Lost (C/D) ... $17
Money Mgr. (C/D) $17
Wall Street (C/D) $17
Data Manager (C/D) .. $17
Elec. Checkbook (C/D) $17

TOTL
Totl Text .. (C) $32 (D) $34
Label ...... (C) $15 (D) $17
Time Mgr. .. (C) $24 (D) $27
Rsrch Asst. (C) $24 (D) $27

TRONIX
S.A.M. (D) $39
Juice (D) $23
Chatterbee (D) $27

Lyco Computer Marketing & Consultants

"PEOPLE WHO KNOW WHAT THEY WANT AND KNOW HOW TO USE IT RECEIVE THE LOWEST PRICES AT LYCO"

SAVE ON THESE IN-STOCK PRINTERS

GEMINI 10X $229 ★

MANNESMANN TALLY
SPIRIT 80 .... $255.00
MTL-160L $549.00
MTL-180L $739.00

JUKI
Juki 6100 .... $389
Tractor Kit $119

Epson
RX80 .... $229
RX80FT $269
RX100 .... $369
FX80 .... $369
FX100 .... $555
JX80 .... $1089
LQ1500P(includes kit) ... $1149
LQ1500S(includes kit) ... $529

Citoh
Prowriter8510A $289
8510BC2 .... $399
8510BP1 .... $349
8510SP $399
8510SR $409
8510SCP $419
8510SCR $499
1550P $489
1550BCD $539
A10-20P $469
F1040PU or RDU $899
F1055PU or RUD $1099

DIGITAL DEVICES
16K printer buffer 99.75
32K printer buffer 119.75
64K printer buffer 169.95

BLUE CHIPS
M12010 .... $275
M12010 C-64 .... $275
D4015 .... $1389

OKIDATA
Okimate 10 .... CALL
82A 299.00
83A $549
84 .... $649
92 .... $359
93 .... $569

LEGEND
880 .... $229.00
1000 .... $279
1200 .... CALL
1500 .... CALL
1081 .... CALL

CARDCO
LQ1 Printer 379.00
LQ3 Printer 299.00
C/?6 C-64 Graphics 65.75
C/?B C-64 .... 39.75
C/?AT Atari 55.00

Smith Corona
Fastext 80 .... 189.00
D100 .... 219.00
D200 .... 399.00
D300 .... 519.00
L1000 .... 339.00

Corona
LP300 Laser Printer .... 2699.
200361 Toner Cartridge .... 89.

NEC
NEC 8025 .... $699
NEC 8027 .... $359

STAR MICRONICS
Gemini 10x $229
Gemini 15x $345
Delta 10 .... $339
Delta 15 .... $449
Radix 10 .... $499
Radix 15 .... $589
Powertype $309
Sweet P 100 .... $549

PANASONIC
1090 .... $219
1091 .... $279
1092 .... $415
1093 .... $599
3151 .... $469

PANASONIC 1090 $219

NOW STOCKING CITIZEN & OLIVETTIC PRINTERS

OVER 2000 SOFTWARE TITLES IN STOCK

COMPUTER CARE

BIB
DISK DRIVE CLEANER $12.75
COMPUTER CARE KIT $19.75

NORTRONICS DISK DRIVE CLEANER with software for IBM-PC, Atari, Vic,
DISK DRIVE CLEANER with software for IBM-PC, Atari, Vic, Apple, TI $29.75
DISK CLEANER REFILL $14.75
CASSDRIVE CLEANER $9.95
MEDIA BULK ERASER $46.75

NEC
PC8201 Portable $429
NECB1 64K Computer System $1049
NECB2 128 K Computer System $1299
PC8221 Thermal Printer ... $ 139
PC8201 8K RAM Chip ... $ 99
PC8206 32K RAM Cart ... $ 299
PC300 Modem ..... $ 65
PC8801 MSDOS 16 Bit Card ..... $ 339

PRINTING PAPER
3000 SHEETS FANFOLD $42.75
1000 SHEETS FANFOLD $19.75
1000 SHEET LETTER $21.95
200 SHEETS LETTER $8.99
150 RAG STATIONARY $10.99
MAILING LABELS (1in.) ... $9.95
14 x 11 1000 FANFOLD $24.75

INNOVATIVE CONCEPTS
FLIP-N-FILE 10 ..... $3.75
FLIP-N-FILE 15 ..... $8.95
FLIP-N-FILE 25 ..... $18.95
FLIP-N-FILE 50 ..... $17.75
FLIP-N-FILE (ROM HOLDER) $17.75

DRIVES

MSD
SD1 DRIVE $259.00
SD2 DRIVE $475.00

INDUS
GT Atari 269
GT Atari 269
GT Commodore ..... CALL
GT Apple w/controller 219
GT Apple 169

DISKETTES

SKC
(Box 10)
SKC-SSSD $12.99
SKC-SSDD $15.99
SKC-DSDD $18.99

ELEPHANT
(Box 10)
5¼"SSSD $14.99
5¼"SSDD $16.99
5¼"DSDD $21.99

MAXELL
(Box 10)
5¼"MD-1 ..... $17.95
5¼"MD-2 ..... $23.95

IBM-PC SOFT-WARE

Scarborough
Songwriter $24.75
Picturewrite $32.75
Phi Beta F $32.75
Mastertype $32.75
Run f Money $32.75
Net Worth $54.75

Spinnaker
Alphabet $18.75
Kids on Keys $19.75
Grandma $19.75
Kindercomp $17.75
Facemaker $19.75
Kidwriter $19.75

IBM-PC COMPATABLE

CORONA
PPC22A Portable 256K-Amber ... $1995
PPC22G Portable 256K-Green ... $1995
PPCXTA Portable 256K-10Meg ... $3295
COR128K 128K RAM ... $ 159

Zenith
Z-150 .... Call

Columbia Data
1600 .... Call

Televideo
TS1605 .... Call

Leading Edge
PC Compatable .... Call

Microprose
Solo Flight $22.75
NATO $22. 75
Spitfire $22.95

Graphics Tablet
Supersketch $49.95
Kolala $99.95
Illustrator $99.95
Logo Design $27.95
Grams Spell $27.95

TOLL FREE 1-800-233-8760

TO ORDER

CALL TOLL FREE

800-233-8760

Customer Service 1-717-327-1825

or send order to
Lyco Computer
P O Box 5088
Jersey Shore PA 17740

RISK FREE POLICY
In-stock item shipped within 24 hours of order. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the Continental U.S. PA residents add sales tax. APO, FPO, and International orders add $5.00 plus 3% for priority mail service. Advertised prices show 4% discount for cash, add 4% for Master Card or Visa. Personal checks require 4 weeks clearance before shipping. All items subject to change without notice.

TAKE THE FEAR OUT OF USING THE TOOLS OF MODERN MAN.

You're a modern person who wants to use a computer. But primitive training methods don't fill your needs. You need a solid training system that shows you, in the privacy of your own home, office... or cave, how to get the most out of your computer and software.

INTRODUCING COMPUTUTOR™
CompuTutor, developed by Chase Scientific, is a remarkable series of plain language computer training programs on videocassette. By providing real life personal computing applications, CompuTutor gives you an immediate hands-on experience. The method is simple. Using your TV and videocassette recorder side by side with your personal computer, you set the pace. Start and stop the video program at any point. Skip sections you already know and review others as many times as you want. Within minutes you'll begin to use your computer with ease and confidence.

As entertaining as they are informative, CompuTutor hardware and software training systems are designed specifically for your IBM®-PC...Apple® IIe... TRS-80® Model 4 and many compatibles. Popular software like Lotus™ 1-2-3™...dBASE II®...dBASE III™ ...WordStar®...SuperSCRIPSIT™ and VisiCalc® perform even better because CompuTutor makes them easier to understand, through state-of-the-art video technology—with a warm, human touch.

Start learning today...Start using tonight, with CompuTutor from Chase Scientific. And put the tools of modern man to work for you.

Order CompuTutor Now. For Visa and MasterCard orders Call Toll Free: (In California) 1 (800) 528-2554. (Outside California) Call Collect: 1 (213) 395-7884. Or, if more convenient fill in the order form below.

ORDER FORM

PLEASE SELECT PROGRAMS:

Title	Qty
Using Your Machine	______
Using VISICALC	______
Using WORDSTAR	______
Using SuperSCRIPSIT (for TRS-80, Model 4 only)	______
Using dBASE II (for Apple IIe only)	______
Using dBASE III (For IBM PC only)	______
Using LOTUS 1-2-3 (For IBM PC only)	______

Video Format: ☐ VHS ☐ BETA
Your Computer: ☐ IBM-PC ☐ Apple IIe ☐ TRS-80 Model 4 (Call About Compatibles)

______ **CompuTutor Programs at $69.95 each** (Please add $5 shipping and handling for each item. California residents add 6% sales tax.)

Total enclosed $______________

☐ **I am interested, but send further information:** ______________________________ (Program Titles)

☐ Check ☐ Money Order ☐ Visa ☐ MasterCard

Card # ______________________ Exp. Date __________

Name ______________________

Address ______________________

City ______________ State ________ Zip ________

Signature: ______________________

Send Check or Money Order Payable To:
CHASE SCIENTIFIC 1311 Colorado Ave., Santa Monica, CA 90404
(Dealer and Distributor Inquiries Welcome) Allow 4-6 weeks for delivery.

NOW! YOU CAN GET THOUSANDS OF FREE PROGRAMS, AND PUT YOUR TELEPHONE TO WORK

With The New ATARI Modem/Software Package For Only $79.95!

To get more out of your ATARI, whether you're a brand-new owner or a database expert—this offer is for you. The ATARI 1030 is the easiest-to-use modem on the market. And since the experts at ATARI designed it, you're guaranteed that it works with your ATARI Computer System.

The perfect modem package for everyone, it has all the necessary software built right in. All you need is a 16K ATARI computer and a telephone line to get started! If you're a disk drive owner, this package includes additional software (on disk—selected by ANTIC Magazine) that will give your 1030 all the power you'll ever need!

NEW!

- Upload/Download Files With Your Disk Drive
- Auto Dial Telephone Number Database
- Easy Downloading Of Programs From Compuserve's ATARI SIG.
- Easy Access To All Bulletin Board Systems
- Simple ATARI-To-ATARI "MacIntosh-Like" Terminal Software

and more...

You'll love the hi-tech design of the 1030 modem. It'll look great next to your computer and peripherals! And hidden inside is the most sophisticated circuitry on the market. This means 100% accurate file transmissions the first time—even over voice-grade phone lines anywhere in the country. Your 1030 modem is built almost to military specs—guaranteed to have less than 1 bit-error out of every 100,000 bits—the lowest in the industry.

And, you'll also receive free introductory subscriptions to Compuserve (access to hundreds of great free programs), and Dow Jones News Retrieval Service (get stock quotes as fast as your stock broker), with FREE TIME ON EACH!

Now ATARI quality at a lower price THAN ANY OTHER MODEM!

YES! I want this extraordinary communications value!

I'm ordering now so I can receive:

- 1 ATARI 1030 300 baud modem with built in software
- Free Introductory Time on: Dow Jones, Compuserve
- Disk Communications Software

The suggested retail value is $199.95 MY PRICE IS ONLY $79.95

Send me ______ number of packages at $79.95 per package to:

Name

Address

City State Zip

Please make check payable to ADD-ON Systems.
Payment enclosed ☐ check ☐ money order
Bill my ☐ Mastercard ☐ Visa

ID# Expires

Credit Card Orders Only Call Toll Free
800 227 1617 X133
800 772 3545 X133 (inside CA)

California residents add 6½% sales tax.
Add shipping charges of $2.75 per modem.

Canadian residents please send U.S. dollars • Allow 2-4 weeks for delivery • Prices subject to change without notice • Delivery subject to availability.

Send coupon to:

ADD-ON Systems
524 2nd St.
San Francisco, CA 94107

PRODUCT MART

Now, the lowest prices ever on

3M Scotch® DISKETTES

$1.53 ea. 5¼" SSDD QTY. 20 — **$2.06** ea. 5¼" DSDD QTY. 20

5¼" SSDD-96TPI → $2.29 ea. 5¼" DSDD-96TPI → $2.85 ea.

SOFT SECTOR ONLY! MINIMUM ORDER: 20 DISKETTES

These are factory-fresh 3M diskettes packed in boxes of 10 with Tyvek sleeves, reinforced hubs, identification labels and write-protect tabs.

3.5" MICRO-DISKETTES—SS-135 TPI → $2.89 ea.

LIFETIME WARRANTY ON ALL 3M SCOTCH DISKETTES!

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois: 1-312-944-2788)
INFORMATION & INQUIRIES: 1-312-944-2788

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
Suite 4806 • 30 East Huron Street • Chicago, Illinos 60611

DISK WORLD! Authorized Distributor Information Processing Products 3M

FANTASTIC LOW PRICES ON

BASF QUALIMETRIC DISKETTES!

$1.41 ea. 5¼" SSDD Qty. 20 — **$1.59** ea. 5¼" DSDD Qty. 20

PACKED IN PLASTIC LIBRARY CASES!

BASF QUALIMETRIC DISKETTES have a LIFETIME WARRANTY and are packed in PLASTIC LIBRARY CASES with Tyvek sleeves, reinforced hubs, user identification labels and write-protect tabs.

SOFT SECTOR ONLY! MINIMUM ORDER: 20 DISKETTES

BASF 3.5" MICRO-FLOPPIES SSDD-135 TPI → $3.02 ea.
BASF 5¼" HIGH DENSITY FOR IBM PC-AT DSDD-HD → $4.72 ea.

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois: 1-312-944-2788)
INFORMATION & INQUIRIES: 1-312-944-2788

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD! Authorized Reseller Information Processing Media BASF

Incredible value!

Nashua™ Diskettes

LIFETIME WARRANTY!

$1.05 ea. 5¼" SSDD Qty. 50 — **$1.15** ea. 5¼" DSDD Qty. 50

These are poly-bagged diskettes packaged with Tyvek sleeves, reinforced hubs, user identification labels and write-protect tabs. NASHUA Corporation is a half-billion dollar corporation and a recognized leader in magnetic media.

SOFT SECTOR ONLY! Sold in multiples of 50 only!

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois: 1-312-944-2788)
INFORMATION & INQUIRIES: 1-312-944-2788

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
Suite 4806 • 30 East Huron Street • Chicago, Illinos 60611

DISK WORLD! Authorized Distributor NASHUA MAGNETIC MEDIA

Save 20% to 60% Or More

On all your OFFICE & COMPUTER SUPPLIES!

Now, you can enjoy DISK WORLD! savings on more than 21,000 office and computer supply products! You name it, we got it…at tremendous savings.

Everything from Scotch™ Tape to Post-It Notes™ to paper clips and rubber bands…and thousands of computer products as well!

Our catalog is huge…more than 700 pages, listing more than 21,000 items.

We have to charge for it: $10.00 to be exact.

But we include a $50.00 worth of discount coupons that you can use on future orders.

Now, it's DISK WORLD! for every office or computer supply need…and always at tremendous savings!

This offer supercedes all prior catalog offers.
Not responsible for typographical errors.

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois: 1-312-944-2788)
INFORMATION & INQUIRIES: 1-312-944-2788

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
Suite 4806 • 30 East Huron Street • Chicago, Illinos 60611

DISK WORLD!

DISK WORLD!

Ordering & Shipping Instructions

Shipping: 5¼" & 3.5" DISKETTES—Add $3.00 per each 100 or fewer diskettes. **Other Items:** Add shipping charges as shown in addition to other shipping charges. **Payment:** VISA and MASTERCARD accepted. **COD Orders:** Add additional $3.00 Special Handling charge. **APO, FPO, AK, HI & PR Orders:** Include shipping charges as shown **and** additional 2% of total order amount to cover PAL and insurance. **Taxes:** Illinois residents only, add 8% sales tax.

Prices subject to change without notice.
This ad supercedes all other ads.
Not responsible for typographical errors.
MINIMUM TOTAL ORDER: $35.00

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois: 1-312-944-2788)
INFORMATION & INQUIRIES: 1-312-944-2788

HOURS: 8AM-5PM Central Time Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
Suite 4806 • 30 East Huron Street • Chicago, Illinos 60611

DISK WORLD!

ATHANA DISKETTES

The great unknown!

99¢ ea. ← 5¼" SSDD Qty. 50 — **$1.09** ea. 5¼" DSDD → Qty. 50

You've used these diskettes hundreds of times…as copy-protected originals on some of the most popular software packages. They're packed in poly-bags of 25 with Tyvek sleeves, reinforced hubs, user identification labels and write-protect tabs.

LIFETIME WARRANTY!

SOFT SECTOR ONLY! Sold in multiples of 50 only.

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois: 1-312-944-2788)
INFORMATION & INQUIRIES: 1-312-944-2788

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
Suite 4806 • 30 East Huron Street • Chicago, Illinos 60611

DISK WORLD! Authorized Distributor ATHANA MAGNETIC MEDIA

DISKETTE STORAGE CASES

AMARAY MEDIA-MATE 50: A REVOLUTION IN DISKETTE STORAGE

Every once in a while, someone takes the simple and makes it elegant! This unit holds 50 5¼" diskettes, has grooves for easy stacking, inside nipples to keep diskettes from slipping and several other features. We like it! **$10.95** ea. + $2.00 Shpng.

DISKETTE 70 STORAGE: STILL A GREAT BUY.

Dust-free storage for 70 5¼" diskettes. Six dividers included. An excellent value. **$11.95** + $3.00 Shpng.

DISK CADDIES The original flip-up holder for 10 5¼" diskettes. Beige or grey only. **$1.65** ea. + 20¢ Shpng.

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois: 1-312-944-2788)
INFORMATION & INQUIRIES: 1-312-944-2788

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
Suite 4806 • 30 East Huron Street • Chicago, Illinos 60611

DISK WORLD! The value leader in Computer supplies And accessories.

PRINTER RIBBONS:

at extraordinary prices!

Brand new ribbons, manufactured to Original Equipment Manufacturer's specifications, in housings. (Not re-inked or spools only.)

LIFETIME WARRANTY!

Epson MX-70/80 .. $3.58 ea. + 25¢ Shpng.
Epson MX-100 .... $4.95 ea. + 25¢ Shpng.
Okidata Micro83 .. $1.48 ea. + 25¢ Shpng.
Okidata Micro84 .. $3.66 ea. + 25¢ Shpng.

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois: 1-312-944-2788)
INFORMATION & INQUIRIES: 1-312-944-2788

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
Suite 4806 • 30 East Huron Street • Chicago, Illinos 60611

DISK WORLD!

PAPER: Save 50% or more!

We buy paper by the TRUCKLOAD from the two biggest mills in the country.

Therefore, we charge a lot less than you've been paying!

For the complete DISK WORLD paper catalogs and price lists, call 1-312-944-2788 or write us. (Please do not use the "800" line for paper catalog requests, unless you are also placing an order.

Don't spend more than you have to for top-quality computer printer paper. Call DISK WORLD! today.

INFORMATION & INQUIRIES:
1-312-944-2788

HOURS: 8AM-5PM Central Time
Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
Suite 4806 • 30 East Huron Street • Chicago, Illinos 60611

DISK WORLD! Where paper Costs less!

STOP SOFTWARE PIRACY . . .

... with one of our new, sophisticated copy prevention products.

for Disk Security

- **SAFEGUARD DISK*** for IBM PC/Compatibles
- **SAFEGUARD 64 DISK*** for Commodore 64

*Renders Bit and Nibbler Copiers Ineffective Software is encrypted and protected on each disk.

for Data Security

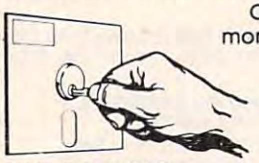
- **DATA PADLOCK**
- **PC CRYPT**

for Telecommunication Security

- **PRIVATE TALK**

Why should your valuable data or useful software program become available in the Public Domain?

Call or write for more information.



GLENCO ENGINEERING INC.
3920 Ridge / Arlington Hts., IL 60004
(312) 392-2492

PRICE BUSTERS

software * books * supplies
for
apple ** atari ** commodore ** ibm-pc ** pc-jr
games * education * home management
business * utilities * systems
on
disks cartridges cassettes

Our prices are 25% to 50% or more below retail.
We have thousands of different computer items.
Please call or write for our current price list.

>CODE for this PRICE LIST<

A=Apple II+ D=Atari Disk G=C/64 Disk J=IBMpc
B=Apple IIC E=Atari Cart H=C/64 Cart K=PCjr
C=Apple IIE F=Atari Tape I=C/64 Tape

e = Educational h = Home Use r = Recreational

Type of Computer:	Store Price		Our Price	Name of Program/Item
ABCDE.GH..JK	$33	e	$23	Alphabet Zoo (Spinnaker)
ABCD..G..JK	$40	e	$26	Agent USA.48k(Atari $22)
ABCD..G..JK	$40	r	$28	Archon (Electronic Arts)
ABCD..G....	$40	r	$28	Archon II: Adept (E.Art)
ABC...G....	$35	r	$23	Beyond Castle Wolfenstein
ABCD..G....	$40	r	$26	Bruce Lee. 48k
ABCD..G..J.	$30	r	$20	Castle Wolfenstein (32k)
ABC.E..H.J.	$35	r	$16	Centipede (IBM $21)
.BCD..G..JK	$50	h	$35	Cut & Paste Word Process
ABCD..G..JK	$50	r	$32	Deadline (Infocom)
ABC.E..H.J.	$35	r	$16	Donkey Kong (IBM $21)
ABCD..G..JK	$35	e	$23	Early Games-Young Chldrn
ABCD..G..JK	$50	h	$35	Financial Cookbook
ABCD..G.IJK	$50	r	$35	Flight Simulator II
AB.DEFGHIJ.	$35	r	$23	Frogger (Cartridges $30)
ABCD..G..JK	$35	r	$25	Hard Hat Mac (Elec.Arts)
ABCD..G....	$75	h	$49	Home Acct.(Arrays/Cont)
.........JK	$150	h	$99	Home Acct.Plus (Array/C)
ABC...G....	$70	h	$33	Homeword (Sierra On Line)
...D.....J.	$70	h	$45	Homeword (Sierra On Line)
ABCDE.GH.J.	CALL	h	CALL	Koala Pad + Software
ABCD..GH.JK	$35	r	$23	Lode Runner (Cart.$27)
ABCD..G..JK	$40	e	$26	MasterType (Scarborough)
ABC...G..JK	$50	e	$32	Math Blaster (Davidson)
ABCD.FG.I..	$25	e	$18	Memory Bldr:Concentration
...D..G....	$40	r	$28	M.U.L.E (Electronic Arts)
ABCD..G....	CALL	e	CALL	Muppet Learning Keys
ABCD..G..JK	$40	r	$28	Murder On the Zinderneuf
ABCD..G..JK	$40	r	$28	Music Construction Set
ABCD..G..JK	$40	r	$28	1 on 1: L.Bird & J.Erving
ABC....H.J.	$35	r	$16	Pac Man (IBM $21)
ABCD..G..JK	$40	r	$28	Pinball Construction Set
ABCD..G....	$50	h	$32	Print Shop (Broderbund)
AB..E..H.J.	$35	r	$19	Robotron (IBM $21)
ABCD.FG.I..	$20	r	$15	Sargon II (Hayden)
ABC...G..JK	$50	r	$32	Sargon III (Hayden)
ABC...G..JK	$90	e	$60	SAT: Barron's Study Prog.
ABCD..G..JK	$40	r	$28	7 Cities of Gold (64k)
ABC........	$40	r	$28	Sky Fox (Electronic Arts)
ABC...G..JK	$70	e	$45	Speed Reader II(Davidson)
ABCD..G..JK	$40	e	$26	Spellicopter (Designware)
ABCD..G..JK	$40	e	$26	Summer Games (Epyx)
ABC...G..JK	$40	r	$26	Trivia Fever (48k)
ABCD..G..JK	$60	r	$39	Ultima II,III.(each).48k
ABCD..G....	$80	h	$52	Your Personal Net Worth
.........JK	$100	h	$65	Your Personal Net Worth
ABCD..G..JK	$40	r	$26	Zork 1,2,3.(Apple $29)ea.

* THE FINE PRINT *

California Buyers: Please add 6% Sales Tax.
Shipping Cost for Software within Cont.48 states:
UPS Ground: 1 Item = $3. Thereafter, $1.50 ea.
UPS Air : 1 Item = $5. Thereafter, $2.00 ea.
Air Mail : 1 Item = $5. Thereafter, $3.00 ea.
Alaska, Hawaii, FPO/APO, Canada: Same as Air Mail
VISA & M/CARD: Please add 3% to the above cash prices. Also include: Card #; Expiration Date; and Cardholder's signature. Sorry, no credit cards if outside the continental USA.
Cashier Check/Credit Card/Money Orders usually shipped out within 24 hrs. Other checks: 2 wks.
Public/Parochial School Purchase Orders Accepted.
This ad was submitted on January 10th, therefore, Price/Availabilty subject to change. In case of problems, your phone # will help us notify you.

FAMILY DISCOUNT COMPUTER PRODUCTS
250-2 So.Orange Ave.#540
Escondido, CA 92025

Phone: (619) 489-1040

>We REBATE $1.on Phone Orders<

RATED #1 FOR SERVICE & RELIABILITY

J&R MUSIC WORLD

23 PARK ROW, NEW YORK, N.Y. 10038

ORDER TOLL FREE 800-221-8180 IN NEW YORK STATE CALL (212) 732-8600

SPECIALS OF THE MONTH

maxell GOLD STANDARD FLOPPY DISKS
Avoid floppy disk trial and error Start with Maxell Performance you can depend on, with a lifetime warranty
MAXELL MD1D 5¼" Single sided/double density 10-pack $19.95/10-pack
MAXELL MD2D 5¼" Double sided/double density 10-pack $24.95/10-pack

DISCWASHER SPIKEMASTER

Superior protection for your electronics •Multimode Multistage suppression •Guarantees protection against fast rise time surges •RFI filtering isolates equipment from radio signals on the power line
$49.95

'TOP 10' COMPUTER HARDWARE

•Famous Commodore 64 computer & keyboard •Excellent graphics & sound capabilities •Built in BASIC •Programmable function keys
$169.95

Atari 800XL 64K COMPUTER	$119.95
Atari 1050 DISK DRIVE	$169.95
Westridge 6420 MODEM	$89.95
Epson QX10 COMPUTER SYS	$1595.95
IBM PCjr. PERSONAL COMPUTER	$799.95
Zenith ZVM-123 12" MONITOR	$99.95
Taxan 210 COLOR MONITOR	$239.95
Commodore C1702 MONITOR	$239.95
Commodore 1541 DISK DRIVE	$229.95

'TOP 10' IBM COMPATIBLE SOFTWARE

LOTUS 1-2-3
•Integrated software package •Spreadsheet •Database •Business graphics •Industry leader
$289.95

Simon/Schuster 'TYPING TUTOR 3'	$34.95
Sorcim 'SUPERCALC III'	$229.95
Microsoft 'FLIGHT SIMULATOR'	$39.95
Microrim 'Rbase 4000'	$279.95
Ashton-Tate 'FRAMEWORK'	$379.95
Lotus 'SYMPHONY'	$449.95
MULTIMATE 3.2	$269.95
Ashton-Tate 'dBase III'	$379.95
Elect. Arts 'ONE-ON-ONE'	$34.95

'TOP 10' COMPUTER PRINTERS

OKIDATA OKIMATE 10 COLOR PRINTER

•Over 26 beautiful colors •Letter quality text printing •Commodore 64/Atari compatible (Please specify computer when ordering)
$199.95

Juki 6100 LETTER QUALITY	$399.95
Epson RX80 F/T DOT MATRIX	$289.95
Atari 1027 LETTER QUALITY	$234.95
Silver-Reed EXP500P	$319.95
Okidata 92P 80 COLUMN	$319.95
Commodore MPS801 MATRIX	$229.95
Gemini 10X FRICTION/TRACTOR	$269.95
Epson RX-80 DOT MATRIX	$249.95
Juki 6300 HI-SPEED DAISY	$779.95

'TOP 10' APPLE COMPUTER SOFTWARE

LOGIC 'FLIGHT SIMULATOR II'
•Pilot a Piper 181 Cherokee Archer •Color 3D graphics •Over 80 airports •Variable weather and more!
$34.95

Hayden 'SARGON III'	$34.95
Monogram 'DOLLARS & SENSE'	$69.95
Infocom 'ZORK I'	$34.95
Simon/Schuster 'TYPING TUTOR 3'	$29.95
Microsoft 'MULTIPLAN'	$179.95
Visicorp 'VISICALC PACKAGE'	$179.95
Broderbund 'BANK STREET WRITER'	$49.95
THINK TANK	$99.95
Software Pub 'PFS:FILE'	$89.95

'TOP 10' COMPUTER ACCESSORIES

WIRE TREE SURGE PROTECTOR
•4 outlet filtered power source •Mounting bracket •Illuminated power switch •Nine foot cord
$49.95

Head 5¼" DISK DRIVE CLEANER	$17.95
Cardco ? + G PRINTER INTERFACE	$69.95
Classic COMMODORE 64 COVER	$6.95
Ring King 070 DATA DEFENDER	$17.95
Maxell CP-15 DATA CASSETTES	$4.95
Hayes MICRO IIE MODEM	$279.95
MPP 1000C DIRECT CONNECT	$119.95
Verbatim MD525 DISKETTES 10-pk	$19.95
TDK M2DS 5¼" DISKETTES 10-pk	$24.95

'TOP 10' MACINTOSH SOFTWARE

MICROSOFT 'MULTIPLAN' SPREADSHEET SOFTWARE

•Performs what-if analysis •Budgeting •Scheduling •Planning •Many other uses •Now available for Macintosh
$129.95

Hayden 'DA VINCI BUILDING BLOCKS'	$49.95
Microsoft 'CHART'	$89.95
Blue Chip 'MILLIONAIRE'	$44.95
Broderbund 'LODERUNNER'	$34.95
Software Pub 'PFS:FILE'	$89.95
Simon/Schuster 'TYPING TUTOR 3'	$39.95
Microsoft 'BASIC INTERPRETER'	$109.95
Monogram 'DOLLARS & SENSE'	$119.95
Mainstreet 'MAIN STREET FILER'	$124.95

'TOP 10' COMMODORE 64 SOFTWARE

BATTERIES INCLUDED WORD PROCESSOR WITH SPELL PACK
•80 column display •15,000 word dictionary •Powerful search functions •Easy to use commands
$79.95

Epyx 'BASEBALL'	$29.95
Sub-Logic 'FLIGHT SIMULATOR II'	$39.95
Access 'RAID ON MOSCOW'	$29.95
Commodore 'SIMON'S BASIC'	$39.95
Activision 'GHOSTBUSTERS'	$29.95
Scarborough 'MASTERTYPE'	$26.95
Creative Software 'I AM THE 64'	$24.95
Precision 'SUPERBASE'	$69.95
Access 'BEACH HEAD'	$24.95

'TOP 10' ATARI COMPUTER SOFTWARE

HAYDEN 'SARGON II' CHESS
•Considered one of the best computer chess programs •Suggested openings •Multi-play levels •Hi-res display
$15.95

MPP 'MICROFILE'	$34.95
Electronic Arts 'ARCHON II'	$29.95
Broderbund 'BANK STREET WRITER'	$44.95
Sub-Logic 'FLIGHT SIMULATOR II'	$39.95
Elect. Arts MUSIC CONSTRUCTION	$29.95
Scarborough 'MASTERTYPE'	$24.95
Origin Sysytem 'ULTIMA III'	$44.95
Infocom 'ZORK I'	$24.95
Elect. Arts '7 CITIES OF GOLD'	$29.95

Remember this is only a sampling of our stock of computers, software, peripherals and accessories. If what you need is not listed please call or write for prices and availability

NEW COMMODORE 128 & ATARI ST & XE COMPUTERS
Call us for availability and price!

NOTE: WE CARRY A FULL LINE OF AUDIO, VIDEO, AND COMPUTERS IN EVERY MAJOR BRAND. THIS LISTING IS JUST A SMALL SAMPLE IN OUR $10,000,000 INVENTORY. PLEASE CALL US TOLL FREE OR WRITE TO US FOR PRICES ON ITEMS NOT LISTED IN THIS AD.

23 PARK ROW, DEPT. C4, NYC, NY 10038

HOW TO ORDER BY MAIL: FOR PROMPT AND COURTEOUS SHIPMENT, SEND MONEY ORDER, CERTIFIED CHECK, CASHIER'S CHECK, MASTERCARD/VISA (include card number, interbank No., expiration date and signature) DO NOT SEND CASH. PERSONAL AND BUSINESS CHECKS MUST CLEAR OUR BANK BEFORE PROCESSING. $25 MINIMUM ORDER. Shipping, Handling & Insurance Charge is 5% of Total Order with a $3.95 minimum. WE SHIP TO CONTINENTAL U.S., ALASKA, HAWAII, PUERTO RICO, AND CANADA ONLY. (Canadian Orders Add 15% Shipping, with a $9.95 minimum charge.) For shipments by air, please double these charges. SORRY, NO C.O.D.'s. NEW YORK RESIDENTS PLEASE ADD SALES TAX. ALL MERCHANDISE SHIPPED BRAND NEW, FACTORY FRESH, AND 100% GUARANTEED. WE ARE NOT RESPONSIBLE FOR ANY TYPOGRAPHICAL ERRORS.

* FREE GIANT CATALOGS: CALL TOLL FREE 800-426-6027
•368 PAGE AUDIO/VIDEO/COMPUTER CATALOG •80 PAGE RECORD AND CASSETTE CATALOG •80 PAGE VIDEO MOVIE CATALOG *Please Send $1.00 For Shipping & Handling

WHAT IS THIS GIRL DOING?

THE: TH AX PAS | GIRL: GG3 ER2 LL PAS | IS: IH SS
29 15 04 34 52 45 04 12 55

FEEDING: PAS FF IY PA3 OO2 IH NG PAS BB1
04 40 19 02 33 12 44 04 28

BREAD: RR1 EH EH PA1 OO1 PAS | TO: TT2 UW2
14 07 07 00 21 04 13 31

A: PAS EY PAS | DUCK: OO2 AA KK2 PAS
04 20 04 33 24 41 04

You can now have your computer speak to you . . . with the inevitable "Kilobaud Monster." This remarkable breakthrough in engineering is now being offered to the general public by Computel, and is made possible at a fraction of the cost you'd expect to pay anywhere else. You will also learn how computers speak, what speech synthesis is, and the basics behind phonemics. It's a great educational tool for the individual and for the institution alike!

The Kilobaud Monster is a revolutionary item for the hobbyist, because now you can have many new resources available for experimentation. No longer is this field of science limited to the lab, and no longer is its cost limited to governmental research projects. You can make your computer speak with little effort, a lot easier than you think!

The quality of speech pronunciation is excellent. Anything imaginable may be spoken, from scientific thru slang words—we include detailed instructions showing you exactly how to do this. You can now have all the satisfaction of doing this yourself, at a fraction of the cost, and having something really useful to show for your efforts!

VISA® MasterCard

Devant si je Puis

KILOBAUD MONSTER?

YOUR COMPUTER WILL TELL YOU! DON'T THEY ALL NOW "SPEAK?". . .

- Able to work with any computer!
- Complete instructions included!
- User-programmable vocabulary of virtually unlimited size!
- Program it to say anything you want!
- Guaranteed to hold your interest and delight for hours!

ONLY **$9.95** *regular price**

Introductory package includes everything you need to get started!

** SAVE—Computel members only $ 7.95.*

The Kilobaud Monster is the basis of a series of projects that can make your computer talk, and is necessary for later projects. The Kilobaud Monster will set you up with everything you need to know in order to attach it to your computer and enable it to speak words. All additional information is included with this beginner's package. The built-in editor in the Wordmaker enables you to easily "fine tune" every word you create to make it sound exactly as you wish. Or you can set up a "word bank" with the Message Talker that includes all the phrases and statements that you'll ever need! The Artificial Intelligence accessory will permit your computer to produce totally random sentences, following all the rules of correct grammar within a pre-defined vocabulary-set, all day long!

If you're new to the computer field, or if you're an old-time "hacker," you'll want to subscribe to our publication "Computel," which covers everything that YOU want to see! Encompassing the subjects of Computers, Electronics, and Telecommunications, this publication gives you a "new meaning" of the free enterprise system and the monopoly of existing organizations. You'll need to become part of this international hobbyist phenomenon really soon. SAVE $2.00 on EACH ITEM you buy if you subscribe to Computel now! And this is IN ADDITION to the total package price for the Complete KILOBAUD MONSTER Collection—you'll SAVE OVER $18. Dealer and educational institution inquiries invited. Circle our Reader Service Number for additional information.

You may order these products from us by sending a check, money order, postage or cash in U.S. negotiable funds only, to the address indicated below. Or you may charge your order to your Visa or Master Card. Please include $2.00 shipping and handling on all orders. California residents please add sales tax. Allow three weeks for delivery. Add $5.00 for next-day *shipment* by U.P.S.; $15.00 for next-day *delivery* by U.S.P.S. We guarantee your satisfaction for 90 days with our products, and offer a money-back guarantee if not fully satisfied during that period.

COMPUTEL PUBLISHING SOCIETY
6354 VAN NUYS BLVD., SUITE 161 / VAN NUYS, CA 91401-2696

YES, I'm interested in making my computer talk. Please send me the following items:

- ☐ INTRODUCTORY PACKAGE $9.95
- ☐ WORDMAKER $19.95
- ☐ MESSAGE TALKER $14.95
- ☐ ARTIFICIAL INTELLIGENCE.......... $24.95
- ☐ COMPLETE COLLECTION SET (SAVE $10) $59.50
- ☐ One year subscription to Computel,. $ 14.00
- ☐ including all Kilobaud Monster items $65.50

PLEASE ADD $2.00 SHIPPING & HANDLING TO YOUR ORDER. C2

CREDIT CARD__________ EXP. DATE______

NAME__________

ADDRESS__________

CITY__________ STATE______ ZIP______

Your Computer__________

SEND ALL PAYMENT IN U.S. FUNDS DRAWN ON U.S. BANKS • SORRY, NO C.O.D.'S

Make checks payable to: COMPUTEL PUBLISHING SOCIETY
Post Office Drawer 7765 / VAN NUYS, CA 91409-7765

Advertisers Index

Reader Service Number	Advertiser	Page
102	Abacus Software	58
103	Abacus Software	65
104	Abacus Software	110
105	AB Computers	39
106	Access Software Incorporated	35
107	ADD ON Systems	155
108	American PEOPLE/LINK	99
	Apropos Technology	13
	Batteries Included	37
109	Cardco, Inc.	IBC
	CBS Software	2,3
110	Chase Scientific	154
	Commodore	BC
111	CompuServe	29
	ComputAbility	51
112	Compute! Publishing Society	159
113	Computer Mail Order	88,89
114	Cosmic Computers	151
115	Davidson & Associates, Inc.	4
116	DesignWare, Inc.	7
	Disk World!	157
	Eastman Kodak Company	14,15
117	Family Discount Computer Products	158
118	General Electric	21
119	Glenco Engineering Inc.	158
	Happy Computers, Inc.	114
120	Harmony Video & Computers	126
121	Indus Systems	47
122	Inforunner Corporation	IFC,1
123	Innovative Technology Ltd.	156
124	Insight	62
	ITT Systems, Inc.	27
125	J & R Music World	158
	Jason-Ranheim	94
126	Legend Peripheral Products	87
127	Lyco Computer	152,153
128	Micro Roots	156
129	Mimic Systems, Inc.	23
130	Mindscape, Inc.	41
131	Nibble Notch Computer Products	128
	North Hills Corporation	156
	North Hills Corporation	156
	NRI Schools	39
	Oberon International	31
	Ortho Information Services	19
132	Pacific Exchanges	156
132	Pacific Exchanges	156
133	Precision Data Products	156
134	Protecto Enterprizes	66
135	Protecto Enterprizes	67
136	Protecto Enterprizes	68–81
136	Protecto Enterprizes	83
137	Protecto Enterprizes	85
138	PSI	11
139	Quinsept, Inc.	65
140	SubLOGIC Corporation	45
141	Tape World	156
142	Timeworks, Inc.	9
143	Witt's End	94

COMPUTE! Back Issues	96,97
COMPUTE! Books	55
COMPUTE! Commodore Collection	49
COMPUTE!'s First, Second & Third Book of Commodore	64
COMPUTE!'s Programming the VIC	53
COMPUTE! Subscriber Services	65

PLUG INTO COMPUTE! AND TURN ON THE POWER.

SAVE 32% OFF THE COVER PRICE!

Accomplish more—enjoy more—with less effort! That's the beauty of your computer and COMPUTE!, the leading magazine of home, educational and recreational computing, filled with up to 30 games and programs each month.

☐ Enjoy 1 year (12 issues) for only $24. Save 32% off the cover price!

☐ Save even more! Enjoy 2 years (24 issues) for only $45.

Name______________________________
(please print)

Address______________________________

City________________ State________ Zip________

☐ Payment enclosed ☐ Bill me later **354101**

Indicate make and model of computer you use:

APPLE ☐ 01 ATARI ☐ 02 64 ☐ 03 VIC 20 ☐ 04 IBM ☐ 05
TEX INS ☐ 06 OTHER__________ 99

PLEASE ALLOW 4-6 WEEKS FOR DELIVERY OF YOUR FIRST ISSUE. FOREIGN AND CANADIAN
PLEASE ADD $6 (U.S.) PER YEAR POSTAGE. OFFER SUBJECT TO CHANGE WITHOUT NOTICE.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 2312 GREENSBORO, NC

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!
P.O. Box 914
Farmingdale, NY 11737

COMPUTE!'s APPLE APPLICATIONS ISSUE

☐ Volume 1 **$3.95**
(magazine only)

☐ I would like to order the companion *Disk* for Volume 1 **$12.95**

(Add $1.00 shipping/handling inside U.S. Add $2.00 shipping/handling outside U.S.) All orders must be prepaid. To order send in this card with your payment or call toll-free **1-800-334-0868.**

NAME ______________________________

ADDRESS ______________________________

CITY ______________________ STATE ____________ ZIP ______________

☐ Payment enclosed. Charge: ☐ Visa ☐ MasterCard ☐ American Express

____

Acct. # Exp. Date

____

Signature

454884C

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 2312 GREENSBORO, NC

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!'s
APPLE APPLICATIONS ISSUE
Post Office Box 5058
Greensboro, NC 27403

The Editor's Feedback:

Computer: ☐ PET ☐ Apple ☐ Atari ☐ VIC-20 ☐ Commodore 64
☐ TI-99/4A ☐ Timex/Sinclair ☐ Radio Shack Color Computer
☐ Other ________________ ☐ Don't yet have one . . .

Are you a COMPUTE! Subcriber? ☐ Yes ☐ No

I would like to see:

More	Just Right	Fewer	
☐	☐	☐	Specific applications programs.
☐	☐	☐	BASIC programs
☐	☐	☐	Machine language programs.
☐	☐	☐	Tutorials.
☐	☐	☐	Educational articles.
☐	☐	☐	Detailed explanations of programs.

More	Just Right	Fewer	
☐	☐	☐	Games.
☐	☐	☐	Reviews of game software.
☐	☐	☐	Reviews of business software.
☐	☐	☐	Reviews of educational software.
☐	☐	☐	Reviews of hardware.

What do you like best about COMPUTE!?

What do you like least?

4 5 6 7 8 9 10 11 12 C

Place
Postage
Here

COMPUTE! Magazine
P.O. Box 5406
Greensboro, NC 27435-0406

COMPUTE!'s
FREE Reader Information Service

Use these cards to request FREE information about the products advertised in this issue. Clearly print or type your full name and address. Only one card should be used per person. Circle the numbers that correspond to the key number appearing in the advertisers index.

Send in the card and the advertisers will receive your inquiry. Although every effort is made to insure that only advertisers wishing to provide product information have reader service numbers, COMPUTE! cannot be responsible if advertisers do not provide literature to readers.

Please use these cards *only* for subscribing or for requesting product information. Editorial and customer service inquiries should be addressed to: COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Check the expiration date on the card to insure proper handling.

Use these cards and this address only for COMPUTE!'s Reader Information Service. Do not send with payment in any form.

COMPUTE!

101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117
118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151
152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185
186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202
203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219
220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236
237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253

Circle 101 for a one year new U.S. subscription to COMPUTE!: you will be billed for $24.

Please let us know. Do you own: / plan to buy:

own		plan to buy
☐ 270	Apple ______	☐ 271
☐ 272	Atari ______	☐ 273
☐ 274	Commodore ______	☐ 275
☐ 276	IBM ______	☐ 277
☐ 278	TI-99/4A ______	☐ 279
☐ 280	Other ______ (specify model)	☐ 281

Please print or type name and address.
Limit one card per person.

Name ______________________
Address ______________________
City ______________________
State/Province ______________ Zip ______
Country ______________________

Please include zip code. Expiration 6/30/85. C0485

SUBSCRIBE TO COMPUTE!

For Fastest Service, Call Our **Toll-Free** US Order Line **800-334-0868**
In NC call 919-275-9809

My Computer Is:
01 ☐ Apple 02 ☐ Atari 03 ☐ Commodore 64
04 ☐ VIC-20 05 ☐ IBM 06 ☐ TI-99/4A
99 ☐ Other ______________ ☐ Don't yet have one.

☐ $24.00 One Year US Subscription
☐ $45.00 Two Year US Subscription

(Readers outside of the US, please see our foreign readers subscription card or inquire for rates).

Name ______________________
Address ______________________
City ______________ State ______________ Zip ______

☐ Payment Enclosed ☐ Bill me
Charge my: ☐ VISA ☐ MasterCard ☐ American Express
Account No. ______________ Expires ____ / ____

Your subscription will begin with the next available issue. Please allow 4-6 weeks for delivery of first issue. Subscription prices subject to change at any time.
The COMPUTE! subscriber list is made available to carefully screened organizations with a product or service which may be of interest to our readers. If you prefer not to receive such mailings, please check this box ☐ [illegible]

Place
Stamp
Here

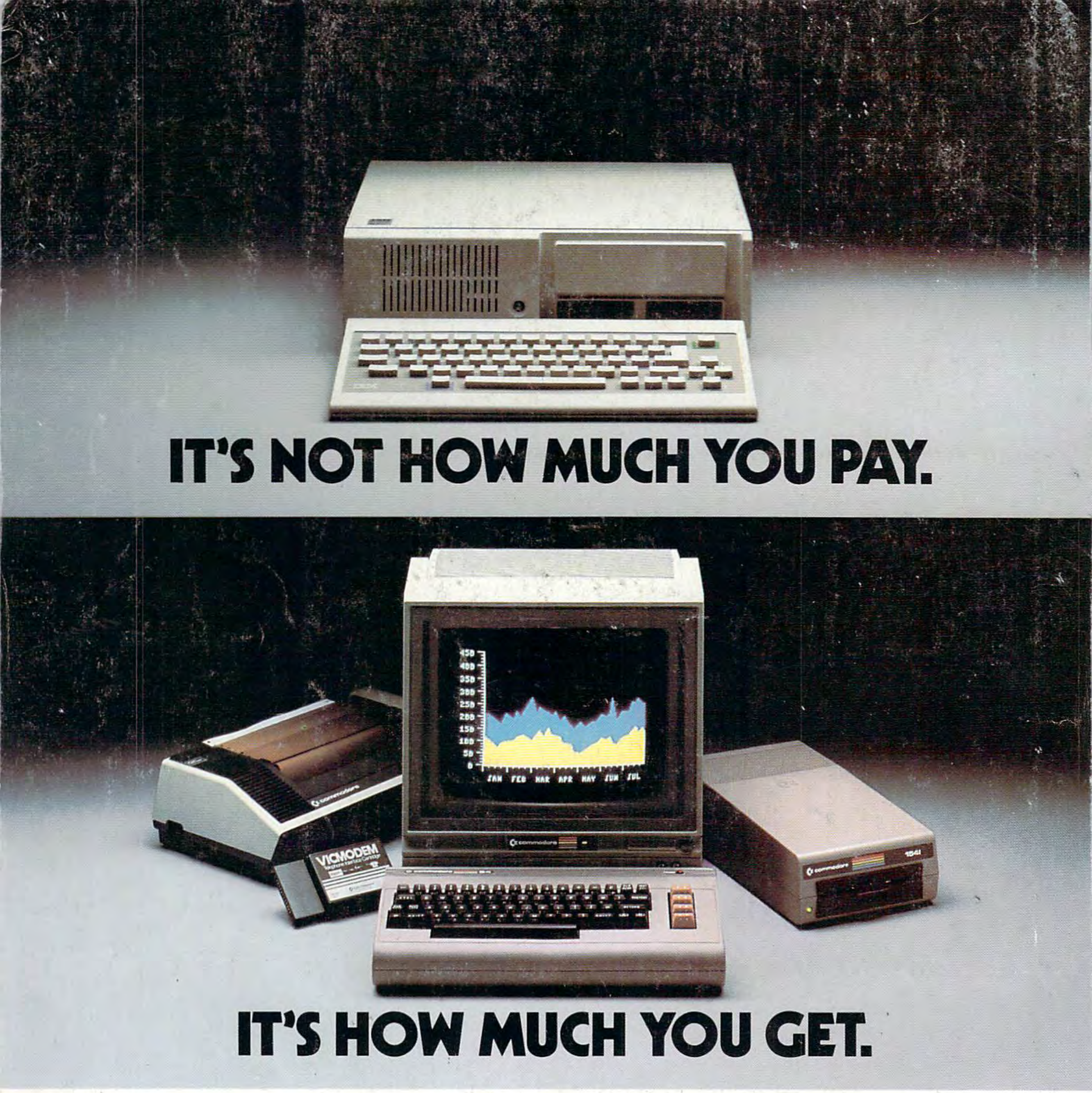
COMPUTE! Reader Service
P.O. Box 2141
Radnor, PA 19089

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 2312 GREENSBORO, NC

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!
P.O. Box 914
Farmingdale, NY 11737



The computer at the top has a 64K memory.

It has the initials I, B, and M. And you pay for those initials.

The Commodore 64™ has a 64K memory.

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

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

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

It has 9-octave high fidelity sound.

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

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

Or buy a lot of computer for the money.

COMMODORE 64

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