REVIEWS

B.C.'s Quest For Tires

James V. Trunzo

My three-year-old daughter, in her innocence, looked at the screen and asked, "What cartoon are you watching, Daddy?" My six-year-old daughter, with her newfound first grade sophistication, said, "Great graphics, huh Dad!" My wife said, "Maybe I can beat this one."

The subject of all this admiration and anticipation is a new arcade-style game released by Sierra On-Line entitled *Quest for Tires*. Each of the statements is accurate.

Eleven Screens

B.C.'s Quest for Tires is an eleven-screen visual delight based upon the cartoon script B.C. The cast of Thor, Fat Broad, Dooky Bird, the dinosaur, and Sweet Chick combine to make *Quest for Tires* a standard for future graphic arcade games. The program is available for Apple, Atari, Commodore 64, and Coleco computers.

The theme of *Quest for Tires* is as old as, well, prehistoric times. Thor, tooling around on his ancient, stone-made unicycle, must overcome a host of obstacles to rescue Sweet Chick from the dinosaur. Some of the obstacles are logs and low-hanging branches that threaten to unseat Thor unless he leaps or ducks in time to avoid disaster, lava pits that must be crossed with the aid of Dooky Bird, and an erupting volcano that spews boulders.

Pushing the joystick forward makes Thor jump, while pulling the stick towards you makes him duck. Moving the stick to the right moves Thor forward at increasing speed, while moving it to the left decreases Thor's forward speed, though it does not stop him.

A Different Speed Level

Those are the basics and they are accomplished with the fire button released. Depressing the fire button adds two more factors to the game play. With the fire button depressed, the results of right and left joystick movement are enhanced. Thor's speed will increase at a much greater rate and he will slow up much quicker too.

However, while the basic play procedures are very simple, after several plays one begins to realize that things aren't as simple as they seem. By moving the joystick towards the upper right diagonal, for example, you can make Thor jump longer distances rather than straight up and down. Mastering this tactic quickly becomes a necessity if you wish to conquer all the screens, especially at the more advanced levels of play.

For Beginners And Experts

How difficult is *Quest for Tires*? My wife's initial reaction was correct with respect to the game's easiest level. The game can be enjoyed by the casual gamer who really isn't interested in totally mastering games. At level one, it is also a game that younger children can play without becoming frustrated by its difficulty and complexity.

This isn't to say, however, that the game is unchallenging to true arcade addicts who have mastered the intricacies of those games requiring four hands. At its more advanced levels, *Quest for Tires* requires a refined sense of timing and quick reactions if one hopes to rescue Sweet Chick before losing all five wheels with which the player begins the game (each wheel equivalent to one "life," of course).

The game has well-planned variety in its levels of difficulty. It's versatile. But the highlight isn't the play of the game itself. It's the great graphics that make the program something special.

Excellent Graphics

Simply put, *Quest for Tires* has graphics and animation which approach cartoon standards. The scrolling of the different screens is done so smoothly that the illusion of movement is complete, and the colors and highlighting of each scene are sharp and vivid.

The game's creators used small details to good effect, such as the way Thor's hair waves in the wind as he pedals his wheel and builds up speed; the look on his face when he fails to jump a pit; and the way Sweet Chick blows Thor a kiss that turns into a heart and floats to Thor when he successfully completes a level.

Quest for Tires has the usual features found in all first-rate arcade programs. It offers four levels of play, one or two player option, and a pause feature. It also provides a vanity board which allows the player to record his initials for posterity upon achieving one of the ten top scores to date. Scoring, incidentally, is based upon avoiding obstacles; the more difficult the obstacle the more points earned by overcoming it. More variation in the scoring is achieved by awarding greater point values to obstacles when they are cleared at higher speeds.

B.C.'s Quest for Tires
Sierra On-Line, Inc.
Sierra On-Line Building
Coarsegold, CA 93614
(209) 683-6858
Apple, Atari, or Commodore 64 disk, $34.95
Atari, Commodore 64, or Coleco cartridge, $39.95

©

Radiotap For VIC And 64

Dan Carmichael, Assistant Editor

The Radiotap by Kantronics is an impressive package that allows you to interface your Commodore 64 or VIC to a shortwave or ham radio receiver. It will automatically interpret communications such as Morse code and radioteletype (RTTY) and display them on your monitor. RTTY is used by various services, from foreign embassies to news wire services. The Radiotap can interpret both standard RTTY and the newer ASCII RTTY.

Several Modes Are Available

In the Morse code mode, the Radiotap reads the incoming signals, and automatically adjusts the receiving speed of the computer. It has the ability to adjust to speeds ranging from 0 to 99 words per minute. All you have to do is turn on the Radiotap, tune in your radio, sit back, and "read the mail."

In the RTTY and ASCII modes, you also have the ability to decipher coded or scrambled messages. These options include bit switching, inverting bit patterns, and the ability to vary the baud rate (receiving speed) to nonstandard settings. In these modes, however, receiving speeds must be set manually.

The Radiotap also has a *scope* feature. It is very useful when analyzing RTTY signals coming in at nonstandard baud rates or which are encoded. You can display the incoming signals on your screen in graphics form, which allows you to measure the timing of the incoming signals (bits) and helps you determine which ciphering techniques, if any, are being used.

Another nice feature is the printer option. By pressing a single key, you can produce a printout in addition to the monitor display. This feature, however, has hardware limitations. It is designed for the Commodore 1525 printer, and may not work with other printer/interface combinations. It did not work with the Epson printer interfaced to a 64 with a Micro-Electronix serial interface. However, it is reported to work with the Cardco interface. For more information, contact Kantronics at the address below.

The Radiotap also has a 24-hour clock, set by the user, that is always displayed at the top of the screen.

Small Inconveniences

There were a few small inconveniences encountered while setting up the Radiotap. First, you must construct the patch cord to connect the Radiotap interface to the shortwave receiver. This requires a little light soldering to attach two plugs to the cord. Second, the dual voltage power supply included with the Radiotap is switch-selectable to either six or nine volts. However, no mention was made of this in the instruction manual. After experimenting for some time with both voltages, we selected nine volts, the correct setting.

The Radiotap comes with the radio-to-computer interface, software in cartridge form, and all the necessary wiring and plugs to connect it up (except the cord to the radio).

Also included is a complimentary copy of the book *Confidential Frequency List* published by Gilfer Associates in Parkridge, New Jersey. But because this book deals largely with items that are not related to either Morse code or RTTY, a better choice might have been *RTTY Frequencies*, also published by Gilfer Associates.

All in all, the Radiotap is a fine product, and you can spend many enjoyable hours reading foreign embassy and wire service messages. The Radiotap is for receiving only; there are no transmit options.

Radiotap
Kantronics
1202 East 23rd St.
Lawrence, KS 66044
(913) 842-7745
$199.95

©

Catie's Grandmother Goes To Camp

Last summer my mom went to computer camp. The camp, The Computer Tutor, was located in Avalon, a small seashore resort town in southern New Jersey. My daughter Catie was also enrolled in a computer camp—Computer FUNdamentals, at Hollins College here in Roanoke, Virginia. (See my column "The World Inside Computer Camps," in last month's COMPUTE! for more on computer camps for children.)

My mom is 58. My daughter Catie is 7. Catie and her grandmother are more than half a century apart in age. Yet, by coincidence, they were enrolled in computer camp at the same time, learning the same thing. I thought that was neat. And significant.

The Candy Store

Millions of adults find themselves on the outside of the computer revolution and they don't know how to get in. They are like a little kid with his face squashed against the display window of a fantastic candy store. They would love to join the other kids inside the store, but they can't find the door.

Computer camps can be one door into computer literacy and computer intimacy for fearful but interested adults. Full-fledged camps for adults are springing up all over the U.S. Adults enroll in the camps for a period of one or two weeks. Many of the camps combine indoor computer instruction with outdoor exercise.

Adult campers usually begin computing slowly, but they quickly pick up the pace. By the end of camp they spend up to ten hours a day in marathon keyboard sessions. And they get hooked. According to the daughter of one 68-year-old camper, "We had to drag mom away from the machine just to make sure she got nourishment."

Computer Day Camp

Not all computer camps are so intense. My mother's computer camp, The Computer Tutor, was only a day camp. My mother attended the camp for three hours a day for five days. Each night when she finished computing, she returned home to my father.

There were four computers, with two people on each computer. The instructor was a teacher at Avalon's lone elementary school. There was one 9-year-old girl enrolled in the camp and a 14-year-old-boy with his father in tow. The rest of the campers were women.

The camp's goal was to teach the basics of computers, including the terminology, a little bit of BASIC programming, and use of the computer keyboard. According to my mother: "Our instructor tried to teach us a lot in a short time. I learned how to do some programming and some graphics. We learned about the disk and about copying and saving."

The Joy Of Flying Solo

My mother's most exciting day came when she got to use the computer on her own. "One day," she said, "I worked by myself. That was great! I felt that if I had done that every day I would have gotten more out of the class."

According to my mother, one of the drawbacks of the course was that the instructor had so much to teach and so little time. "You're trying to jam so much into a few hours that it gets very confusing. Having a computer to practice things on would have made things easier."

It was interesting for my mother to watch her classmates' reaction to computers and compare their reactions with her own. "My college secretarial courses on touch-typing helped me a lot," she said. "Being a typist eliminated some of my fear of computers right away. But I was still cautious.

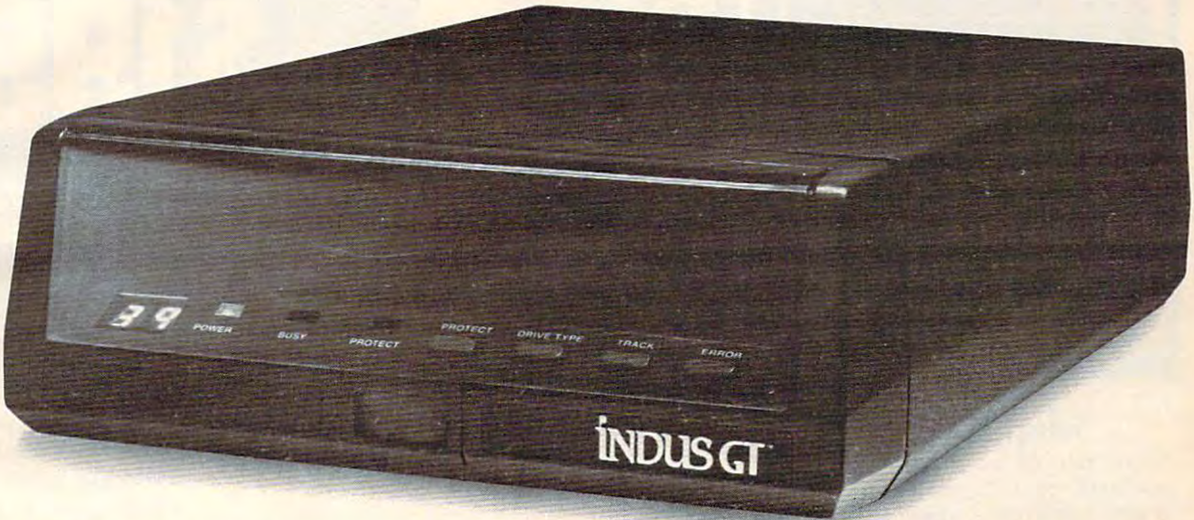
"There were others who were more willing to jump right in," she admitted. "Some of the people got right to work on the computer and tried to invent things right away. Others were quite leery and wouldn't do things until they were taught. This group included me."

Life After Computer Camp

"Computer camp was great," my mother said. "It whetted my appetite for computers. After it was over, I wanted to continue learning more.

"The only problem was I couldn't find any

Looks like a Ferrari. Drives like a Rolls. Parks like a Beetle.



Ask your computer dealer
to take the cover off a world-class disk drive.

The all new, 1984 Indus GT.™

The most advanced, most handsome disk drive in the world.

A flick of its power switch can turn an Atari into a Ferrari.

Or an Apple into a Red Hot Apple.

Looks like a Ferrari.

The Indus GT is only 2.65" high. But under its front-loading front end is slimline engineering with a distinctive European-Gran flair.

Touch its LED-lit CommandPost™ function control AccuTouch™ buttons. Marvel at how responsive it makes every Atari or Apple home computer.

Drives like a Rolls.

Nestled into its soundproofed chassis is the quietest and most powerful disk drive power system money can buy. At top speed, it's virtually unhearable. Whisper quiet.

Flat out, the GT will drive your Atari track-to-track 0-39 in less than one second. Increasing data transfer 400%. (Faster than any other drive. And as fast as any Apple disk drive.)

And each GT comes with the exclusive *GT DrivingSystem*™ of software programs.* World-class word processing is a breeze with the *GT Estate WordProcessor*.™ And your dealer will describe the two additional programs that allow GT owners to accelerate their computer driving skills. *Included as standard equipment.

Also, the 1984 Indus GT is covered with the *GT PortaCase*.™ A stylish case that conveniently doubles as a 80-disk storage file.

Parks like a Beetle.

The GT's small, sleek, condensed size makes it easy to park.

And its low price makes it easy to buy.

$449 for Atari. $329 for Apple.

So see and test drive the incredible new 1984 Indus GT at your nearest computer dealer soon.

The drive will be well worth it.

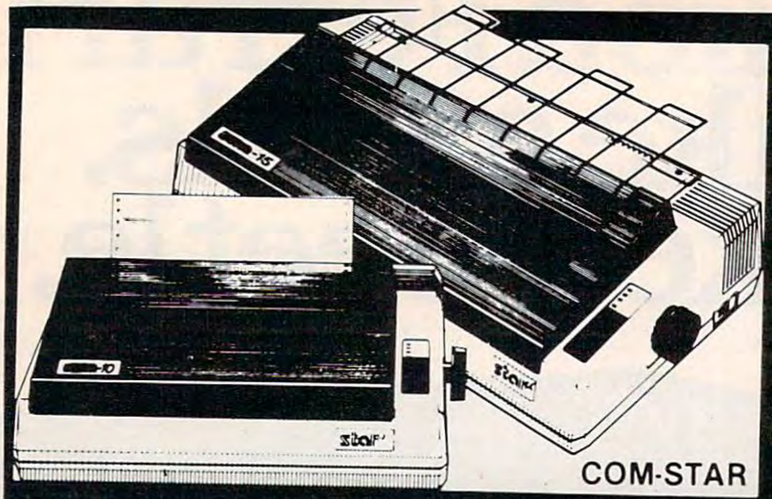
INDUS™

The all-new 1984 Indus GT Disk Drive.

The most advanced, most handsome disk drive in the world.

For dealer information, call 1-800-33-INDUS. In California, 1-800-54-INDUS, 213/882-9600.

80 COLUMN PRINTER SALE—$149.00*



COM-STAR T/F

Tractor Friction Printer

only $199**

•15 Day Free Trial -180 Day Immediate Replacement Warranty

- Lowest Priced, Best Quality, Tractor-Friction Printers in the U.S.A.
- Fast 80-120-160 Characters Per Second
- 40, 46, 66, 80, 96, 132 Characters Per Line Spacing
- Word Processing
- Print Labels, Letters, Graphs and Tables
- List Your Programs
- Print Out Data from Modem Services
- "The Most Important Accessory for Your Computer"

*STX-80 COLUMN PRINTER—$149.00

Prints full 80 columns. Super silent operation, 60 CPS, prints Hi-resolution graphics and block graphics, expanded character set, exceptionally clear characters, fantastic print quality, uses inexpensive thermal paper! Best thermal printer in the U.S.A.! (Centronics Parallel Interface).

**DELUXE COMSTAR T/F 80 CPS PRINTER—$199.00

The COMSTAR T/F (Tractor Friction) PRINTER is exceptionally versatile. It prints 8½" x 11" standard size single sheet stationary or continuous feed computer paper. Bi-directional, impact dot matrix, 80 CPS, 224 characters. (Centronics Parallel Interface).

Premium Quality—120 CPS COMSTAR T/F SUPER-10X PRINTER—$289.00

COMSTAR T/F (Tractor Friction) SUPER-10X PRINTER gives you all the features of the COMSTAR T/F PRINTER plus a 10" carriage, 120 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image (120 x 144 dot matrix), underlining, back spacing, left and right margin settings, true lower decenders with super and subscripts, prints standard, italic, block graphics and special characters, plus 2K of user definable characters! The COMSTAR T/F SUPER-10X PRINTER was Rated No. 1 by "Popular Science Magazine." It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) (Better than Epson FX 80).

Premium Quality—120 CPS COMSTAR T/F SUPER-15½" PRINTER—$379.00

COMSTAR T/F SUPER 15½" PRINTER has all the features of the COMSTAR T/F SUPER-10X PRINTER plus a 15½" carriage and more powerful electronics components to handle large ledger business forms! (Better than Epson FX 100).

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 10" PRINTER—$489.00

SUPER HIGH SPEED COMSTAR T/F (Tractor Friction) PRINTER has all the features of the COMSTAR SUPER-10X PRINTER plus SUPER HIGH SPEED PRINTING—160 CPS, 100% duty cycle, 8K buffer, diverse character fonts, special symbols and true decenders, vertical and horizontal tabs. RED HOT BUSINESS PRINTER at an unbelievable low price!! (Serial or Centronics Parallel Interface)

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 15½" PRINTER—$579.00

SUPER HIGH SPEED COMSTAR T/F 15½" PRINTER has all the features of the SUPER HIGH SPEED COMSTAR T/F 10" PRINTER plus a 15½" carriage and more powerful electronics to handle larger ledger business forms! Exclusive bottom paper feed!!

PARALLEL INTERFACES

For VIC-20 and COM-64—$49.00

For All Apple Computers—$79.00

NOTE: Other printer interfaces are available at computer stores!

Double Immediate Replacement Warranty

We have doubled the normal 90 day warranty to 180 days. Therefore if your printer fails within "180 days" from the date of purchase you simply send your printer to us via United Parcel Service, prepaid. We will IMMEDIATELY send you a replacement printer at no charge, prepaid. This warranty, once again, proves that WE LOVE OUR CUSTOMERS!

Add $17.50 for shipping, handling and insurance. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)

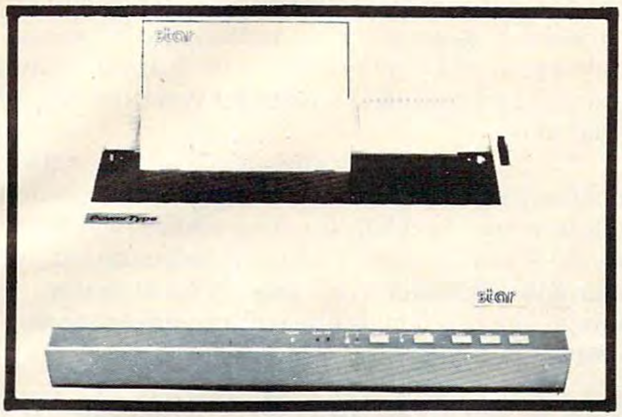
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SUPER-10" ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

EXECUTIVE LETTER QUALITY

DAISY WHEEL PRINTER SALE $379.00

COMSTAR 13"

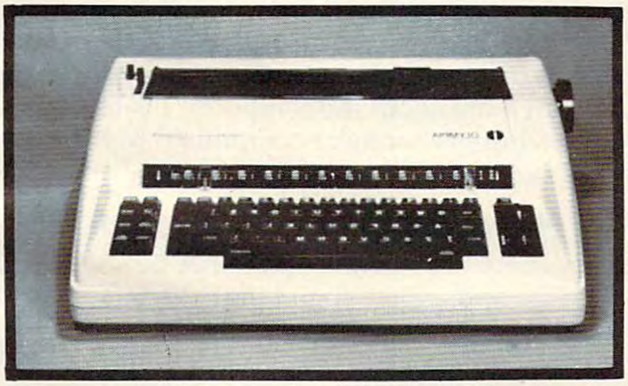


computer printer

COMSTAR 13" "DAISY WHEEL" POWER TYPE PRINTER is typewriter friendly. It uses a simple drop in cassette ribbon. Just turn on the COMSTAR 13" for Crip executive quality correspondence at 18 CPS with a daisy wheel that prints 96 power type flawless characters, bidirectional. Designed for personal and business applications. COMSTAR 13" carriage accepts paper from letter to legal size, continuous computer paper or single sheets, you can set right and left margins, vertical and horizontal tabs. LIST PRICE $599.00 **SALE PRICE $379.00**

OLYMPIA "DAISY WHEEL" COMBINATION

PRINTER/TYPEWRITER SALE $489.00



Olympia (WORLD'S FINEST)

THE **Olympia COMPUTER PRINTER ELECTRONIC TYPEWRITER** is the ultimate for Home, Office, and Word Processing. You get **the best Electronic Typewriter made** and used by the world's largest corporations (better than IBM Selectric) plus a **Superb Executive Correspondence Computer Printer!!** (Two machines in one!) Just flick the switch for the option you want to use. The extra large carriage allows 14⅛" printer paper width. It has cassette ribbon lift off correction. Baud rates, Jumper selectable 75 through 19,200 (serial or parallel interface) LIST $799.00 **SALE $489.00**

• 15 DAY FREE TRIAL — 90 DAY FREE REPLACEMENT GUARANTEE

Add $17.50 for shipping and handling!!

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

PROTECTO

ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

more courses that fit my schedule. I began to worry that I would forget everything I had learned at camp. I decided that the only way to keep learning was to get my own computer. My husband was very supportive. He said he'd get me a low-cost computer as a combined birthday and anniversary gift. He was so proud that I had gone to the camp and that I was learning about computers and having fun."

Grandmother Computer Chic

When my mother got her computer home, at first she couldn't decide where to put it. She ended up putting it on the kitchen table. With its software, manuals, and electric cables, the computer took up almost the whole table.

My mother and father now have to eat their meals on a tiny corner of the table opposite the computer. My mother eats her meals while she is studying the gift certificates she got with her computer. While my father is eating, he just stares at the back of the computer.

My mother is very proud of her new computer. When she first got it, she invited all her friends and family over to see it. The reactions were diverse.

Her grandchildren, Shannon (8) and Laurel (5), were fascinated with the computer's voice synthesizer and the Sesame Street computer games my mother bought. Both grandchildren use computers at school and arrived with the impression that they were "way ahead of Mom Mom" on the computer. But Mom Mom held her own. She got a great deal of pleasure out of being an honorary Whiz Kid.

Mom's friends have had all sorts of reactions. They are proud of her and are amazed that she went out and bought a computer. They are even more amazed that she actually seems to be using it—to write letters using *Bank Street Writer* and to create Christmas-card mailing labels using *PFS:File*.

This year Mom's friends are bragging that they all got Christmas cards addressed by Mom's computer. "I could tell it was a computer that did it," one friend told me, "because it put my last name first on the envelope."

Mom's friends think that she uses her computer all the time. When they call her on the phone, if she sounds vague or distracted, they say, "Libby, are you playing with your computer again?"

Most of Mom's friends are very proud of what she is doing, but they are reluctant to follow in her footsteps. "I know computers are the up and coming thing," one friend told her. "But I could never work one. I'd be lost."

My mother laughs at this kind of reaction. "The biggest thing I have found," she says, "is how easy computers are. I thought they would be much harder."

Then she shrugs and frowns. "It's funny, though," she says. "They are easy, but you still need someone to call on—just for help on the simple things. The most frustrating part is when you are working by yourself and you come to a standstill because a certain button doesn't work. It always worked before, but now it doesn't work, and you feel completely lost and you don't know what to do.

"A person needs someone to call when they feel frustrated—someone to help them, someone to follow up. And half the time advice over the phone is not enough. You need someone looking over your shoulder to see exactly what you're doing. The problem is always something small, but it's enough to stop you in your tracks."

How Grandmothers Get Intimate With Their Computers

According to my mother, "The most exciting time I had with my computer was when I made the mailing labels come out on the printer. 'Look!' I shouted. 'It works! I figured it out myself!'

"Using a computer is like eating Chinese food," my mother told me. "You eat Chinese food, and it tastes great. Then, soon after you are done eating, you are hungry for more.

"Computers are the same way. You do something on it, and it feels great. Then you run out of things to do and you say, 'What can I do now?' It's very frustrating. I have so much fun working on it, but then I finish, and I think I have to invent something new. But to do something new I have to learn more about the computer. It's like I am always hungry for more computing. My appetite keeps growing."

I'm Not Leaving My Computer!

Each winter my mother and father fly south along with the birds. They spend the cold, bitter months in a little resort town tucked away in the Florida keys. My mother flees the Philadelphia area around Thanksgiving, usually after the first cold, soggy November rainstorm.

But this year is different. When Thanksgiving came and went and my mother was still in the Philadelphia area, her friends became puzzled and asked her why she wasn't heading south.

"I'd like to get away from this cold weather," my mother replied. "But I'm not going to leave my computer!"

Mom's Bright Idea

By mid-November my mother had so many manuals, cables, software boxes, and add-ons she would have needed to rent a U-Haul trailer to get her computer down to Florida.

She checked into renting a station wagon and having my brother drive her computer to Florida. Then she realized that her cottage there was barely large enough for her and my father. She and my father have a loving but stormy relationship. She could imagine what would happen if my father arrived in Florida and found that he had been replaced by a computer.

What was my mother to do? She really wanted to go to Florida, but she couldn't bear to leave her computer in Pennsylvania. Yet she didn't dare take it with her.

My mother is no dumb bunny. She wanted a computer in Florida, so if her Pennsylvania computer wouldn't fit, the answer was obvious: She needed a new computer—a smallish kind of computer that she could squeeze into the Florida cottage along with her and my father.

Once my mother has a good idea she doesn't waste any time putting it into action. As I write this article, she is busy saving up for a new computer—a portable.

And she's keeping busy on her old computer, using her spreadsheet to chart her stocks and bonds and keeping an inventory of furniture and other household possessions. And she's churning out a snowstorm of letters to me on her word processor.

It All Started With Computer Camp

I've been away from home for seventeen years, yet I have gotten more letters from my mother in the last couple of months than in all those seventeen years combined. And all the letters are chock full of motherly advice (not to mention nagging).

Each morning I go to the mailbox and find two more letters from my mother—all generated on her infernal (excuse me, Mom) word processor.

I'm beginning to wonder if I did the right thing. Was it really such a good idea to get my mother turned on to computers?

She's beginning to talk about electronic mail and linking us up on a Bulletin Board System. This way she can download ideas and advice anytime she wants, maybe even several times a day. It will be just as if she lived next door.

I'm very proud of Mom. And I'm glad that she went to computer camp. The only thing that worries me is where is this going to end?

My advice to those of you out there with mothers and grandmothers is simple: Once you start them computing, watch out! ©

READERS' FEEDBACK

The Editors and Readers of COMPUTE!

What's A Cassette Buffer?

What is the cassette buffer used for?

James Kenny

A buffer is a temporary storage area. The cassette buffer is a special set of Random Access Memory (RAM) locations where the computer's operating system places information which is to be transferred to or from a cassette storage device. Because the operating system only uses this buffer during tape operations, it is available for other uses between tape operations.

If you switch from tape to disk storage, the cassette buffer no longer has a special function, and memory in the buffer can be used the same as other RAM locations.

In Commodore computers the cassette buffer is in locations 828–1019. In Atari, locations 1021–1151 are used.

Changing The 1541 Device Number

I just purchased my second Commodore 1541 disk drive. I want to change the device number using the hardware method, but I am having trouble doing it.

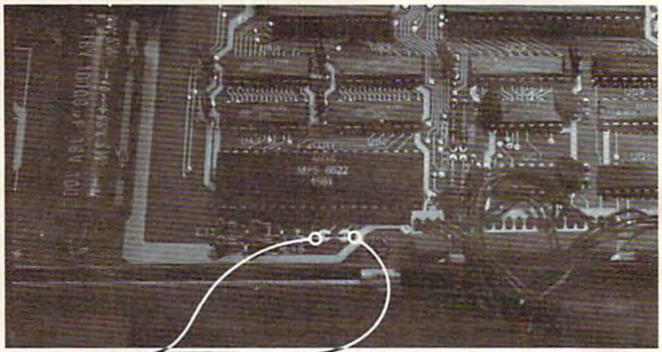
The directions in the 1541 user's manual give ten steps to follow, but they leave out the most important part. Step number seven states: "Locate device number jumpers. If facing the front of the drive, it's on the left edge in the middle of the board." Direction number eight states: "Cut either or both of jumpers 1 and 2."

But once you have removed the case as directed, you find five plugs labeled P4 through P8 with a total of 21 wires. The instructions don't tell which wire is which.

Can you tell me which are the correct two wires to cut so I can change the device number?

Pat Cardinal

Some of this confusion is due to the fact that the jumpers which are to be cut are not wires. The jumpers are actually two mounds of solder, each about 1/4 inch in diameter. If you face the front of the disk drive, they are located on the left edge of the circuit board behind the plugs and wires you described. As shown in the picture, each round drop of solder is separated into halves, with a thin strand of solder connecting each half. You cut the jumpers by taking a sharp, pointed object and scraping away, or breaking, the connecting strand. Be careful to avoid damaging any other components on the circuit board.



The jumpers to be cut can be found on the left side of the circuit board.

As stated in the user's manual, cutting jumper one or jumper two, or both, produces different device numbers. Here is a chart showing which device number is produced when the jumpers are cut:

Jumper Cut	New Device Number
None	8
1	9
2	10
1 and 2	11

One important note about changing the disk drive device number: Once the jumpers are cut, it's difficult to reverse. Also, this sort of modification should be left to an experienced electronics technician unless you know exactly what you are doing. And be sure the drive is not plugged into the wall if you remove the cover to attempt this fix.

Commodore 6502 And 6510 Chips

I am somewhat confused. Is a 6510 microprocessor compatible with the 6502 microprocessor? The box my Commodore 64 came in says that this is true, but I have read differently. If this is true, can I buy a book on programming the 6502 and use it with the 6510 that is in the 64?

Shawn Carnell

Yes, you can. The only differences between the 6502 and the 6510 are locations 0 and 1. In the 6510, these two locations act as an eight-bit parallel input/output port. By POKEing values into memory locations 0 and 1 on the 64, you can do such things as switch out BASIC and Kernal ROM and turn them into usable RAM. For more details on the memory bank-switching technique,

COMPUTE!'s
First Book of VIC Games

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Table of Contents

Special Requirements: J = joystick M = memory expansion

iii

Special Requirements: J = joystick M = memory expansion

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

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

ORDER NOW

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

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

see "6510 Microprocessor Chip Specifications" (Appendix L), and "Memory Management on the Commodore 64," page 260 of the Commodore 64 Programmer's Reference Guide.

Other than these two locations, the 6502 and the 6510 are the same.

Atari Compatibility

Is the BASIC in the new Ataris the same as the BASIC in the 400 and 800? In other words, will the games and programs for the 400 and 800 run on the new XL and XLD computers? Also, how many player/missiles do the new Ataris have?

Dennis Heckman

Although some changes have been made in the newer models, BASIC is still essentially the same. Programs written for the older models should run on the XLs, provided the programmer has followed Atari's rules regarding the ROM jump tables. Some ROM routines have been moved in the newer models, and a direct call to a ROM routine which has been moved will yield unpredictable results.

The new Atari XL computers have four players and four missiles, as did the older models.

A 64 Keyboard Tone

Will you publish a positive stroke key tone generator for the 64, like the one for the VIC in the November 1983 issue of COMPUTE!?

Andrew Predoehl

Try this:

```
100 PRINT"{CLR}{RVS}BEEP-KEY":PRINT"
    {2 DOWN}READING MACHINE LANGUAGE..."
110 FOR I=49152 TO 49228:READ A:CK=CK+A:P
    OKE I,A:NEXT
120 IF CK<>9872 THEN PRINT"ERROR IN DATA
    {SPACE}STATEMENTS. CHECK TYPING":END
130 SYS49152:PRINT"{HOME}{RVS}";TAB(8);"
    {SPACE}NOW ACTIVATED.{6 DOWN}"
140 END
49152 DATA 120,169,013,141,020,003
49158 DATA 169,192,141,021,003,088
49164 DATA 096,165,251,197,197,208
49170 DATA 004,201,064,208,026,165
49176 DATA 197,201,064,240,020,162
49182 DATA 006,189,070,192,157,000
49188 DATA 212,202,016,247,169,015
49194 DATA 141,024,212,169,004,133
49200 DATA 252,165,197,133,251,198
49206 DATA 252,165,252,208,008,173
49212 DATA 004,212,041,254,141,004
49218 DATA 212,076,049,234,000,050
49224 DATA 000,000,129,064,244
```

Different Number Systems

When programming books mention a memory location, many times they put a $ in front of the number. Why is this done?

Marc Foglia

Humans are accustomed to counting by tens, using decimal (base 10) numbers. Computers count by twos, using binary numbers. But binary numbers, such as 10111010, are not easy to read, so machine language programmers commonly use hexadecimal (base 16) numbers because they are easier to read than binary numbers, and easier to translate into the computer's binary system than are decimal numbers. BASIC programmers, however, stick with ordinary decimal.

Since all three number systems can be used in computer programming, we must be able to differentiate between them. For example, if you see the number 0100, you must know if it means 100 (as a decimal number) or 4 (as a binary number) or 256 (as a hexadecimal number).

To prevent this confusion, programmers commonly identify binary and hexadecimal numbers with special symbols:

%0100	**(binary)**
$0100	**(hexadecimal)**
100	**(no symbol for decimal)**

A 1540 Disk Formatting Disaster

I apparently made a big mistake! I own a VIC-20 and a 1540 disk drive. I accidentally reformatted a disk that already had programs on it. I used the following syntax: OPEN 15,8,15,"N0:DISK1 SD1,1B". I noticed that I didn't hear the usual "rattling" sound. I want to know if there is any way of recovering the lost or erased files. I almost had that disk full.

Jeff Lovell

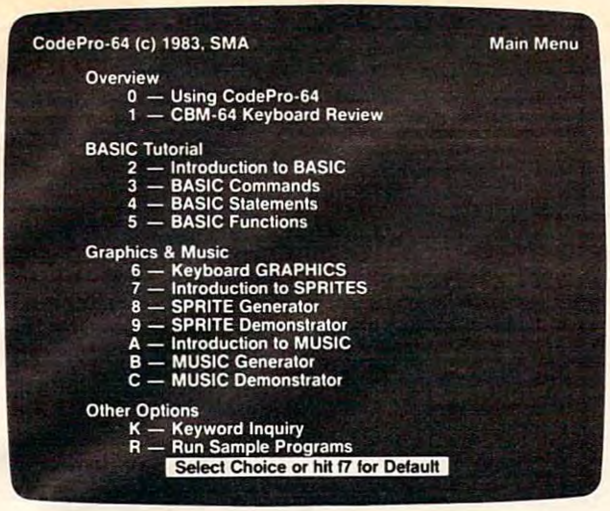
Unfortunately, your programs were erased.

A number of things happen when you format a disk. First, the entire disk is erased, then the directory and BAM (Block Availability Map) are created. Next, timing and block markers are created, and the two-character disk identification code is written on every block of the disk.

To format a disk, use the syntax:

OPEN 15,8,15: PRINT#15,"N0:DISK NAME,ID": CLOSE 15

where DISK NAME is any name up to 16 characters



A new concept in interactive visual learning.

CodePro-64™

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

SEE PROGRAM EXECUTION

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

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

EXTENSIVE TUTORIAL

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

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

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

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

SPRITE GENERATOR & DEMONSTRATOR

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

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

MUSIC GENERATOR & DEMONSTRATOR

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

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

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

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

OUR GUARANTEE

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

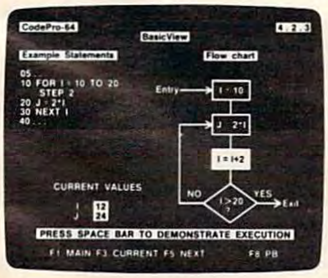
HOW TO ORDER

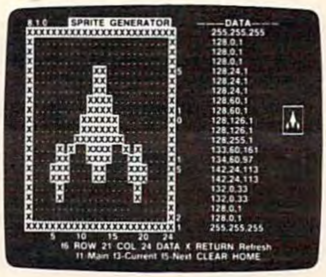
Order your copy of CodePro-64 today by mail or phone. Send only $59.95 plus $3.00 shipping and handling to:

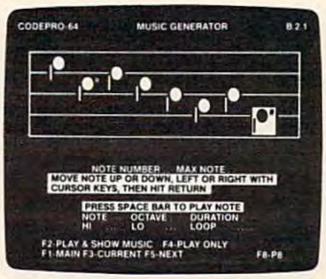
SYSTEMS MANAGEMENT ASSOCIATES
3700 Computer Drive, Dept. G-1
Raleigh, N.C. 27609

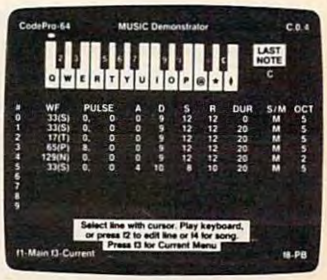
Available on diskette only. MasterCard/-VISA accepted. For faster service on credit card orders, call toll free 1-800 SMA-RUSH. (1-800-762-7874). *Dealer inquiries invited.*

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









SYSTEMS MANAGEMENT ASSOCIATES

long and ID is a two-character identification code. Because of the way the format command works, this should be done only when the disk is new, or when the files on the disk are no longer needed.

As you can see, when a disk is formatted, any programs that were on that disk are lost.

TI LIST And RUN Protector

I would like to know if there is a way to put an access code on a program for the TI-99/4A. In my family I do all the work and everybody else does all the playing; but they cannot play if it has a code on it.

Aaron Rawn

C. Regena replies:

At the beginning of your program you could try these lines:

```
100 A$="HELLO"
110 INPUT "ENTER CODE: ":C$
120 IF C$=A$ THEN 200
130 PRINT "SORRY, CODE NOT ACCEPTED
.."
140 STOP
200 PRINT "PROGRAM CONTINUES."
```

When you RUN your program, the computer first asks you to enter the code. If you type in the correct code name the program will continue; otherwise, the program will stop. We used HELLO for the code name. Note that anybody can just LIST the program and get your code name. Instead of HELLO between the quote marks type in your own code word, but hold down the CONTROL key (to the left of the space bar) while you are typing. Now if someone LISTs the program they can't read what your word is—it's either spaces or some funny-looking characters.

VIC Memory Requirements

Say a VIC program requires 11K, and you have only a 16K memory expansion cartridge. Will the program run with the extra memory, or are you required to use the exact memory needed?

My second question is about word processing. When typing with or using a word processor, do you actually need a printer? All I'd really like to do is store my information on disk.

Dwain Young

When a program or article states requires 11K, *this means it needs 11K bytes of memory* or more. *Both the VIC and 64 have pointers that manage BASIC memory. Examples are memory locations 45 and 46, which point to the start of BASIC variables (the end of the BASIC program proper), and bytes 55 and 56, which point to the highest address used by BASIC. These pointers automatically manage the extra memory for you.*

However, on the VIC you have another thing to consider—screen and color memory locations. In the unexpanded VIC, the screen, color, and BASIC memory areas start at locations 7680, 38400, and 4096, respectively. When you plug in a 3K expander (like the Super Expander cartridge), these locations do not change. However, when you plug in one of the larger expansion cartridges (like the 8K or the 16K), these memory locations do change, to 4096, 37888, and 4608, respectively.

This is important with programs that perform POKEs or PEEKs to color or screen memory. For example, a program that POKEs or PEEKs screen or color memory and states needs 3K expansion *might not work with an 8K expander because of the relocation of these memory blocks.*

On the other hand, a program written for the unexpanded VIC that contains no POKEs or PEEKs to screen or color memory should run with any memory expansion. If you LOAD a program and it doesn't run properly, simply turn off your VIC, change memory expansion cartridges, and reLOAD your program. This will not harm the BASIC program. Just be sure you SAVE your BASIC programs before turning off your VIC.

No, you don't need a printer to use that word processor, if onscreen viewing and storing the text to disk or tape are all you require.

Extra Functions For SuperBASIC 64

The "SuperBASIC 64" program (COMPUTE!, December 1983) is very interesting and helpful. One of its good features is the clarity with which it is laid out. This makes it easy to write and test modifications.

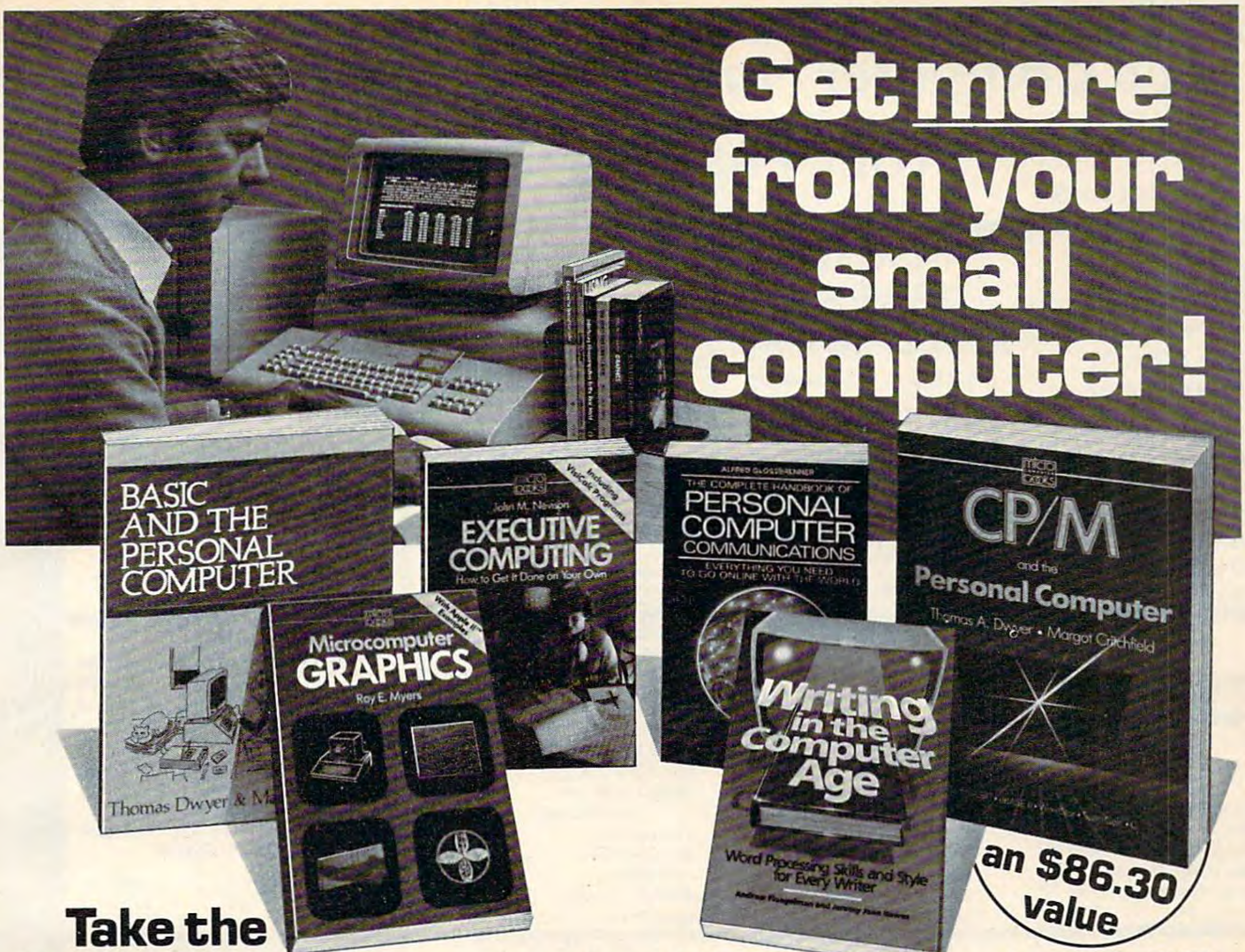
Two sample modifications are given below. To use them, simply LOAD and RUN SuperBASIC 64, then enter and RUN the modification as a normal BASIC program.

The first program creates a new function, [SCRN. [SCRN sets the background, border, and text colors in one operation. For example, [SCRN12,11,1 sets up a light gray background, dark gray border, and white letters.

```
1 REM CREATE [SCRN FUNCTION
10 FORI=0TO9:READA:POKE52168+I,A:NEXT
20 DATA32,57,195,32,66,195,32,218,197,96
30 FORJ=0TO5:READA:POKE49892+J,A:NEXT
40 DATA83,67,82,78,199,203
```

The second program modifies the [DRAW command, allowing it to either draw or erase lines. In its new version, [DRAW must be followed by either 1 or 0. [DRAW1,x1,y1,x2,y2 will draw a line from x1,y1 to x2,y2. [DRAW0,x1,y1,x2,y2 will erase the line between those coordinates.

```
1 REM MODIFY [DRAW TO ALLOW ERASURE
```



Take the 6-volume Microcomputing Library for only $2.95

when you join the SMALL COMPUTER BOOK CLUB

You simply agree to buy 3 more books—at handsome discounts—within the next 12 months.

This set of six carefully selected books—an $86.30 value—will be your basic reference source for small computing.

1. BASIC AND THE PERSONAL COMPUTER.
Thomas A. Dwyer and Margot Critchfield. A down-to-earth introduction to programming in BASIC. Includes: computer graphics, program development, data bases, and more. Also, many step-by-step examples of word processing, computer games, and computer simulations. **Publisher's Price: $15.55.**

2. MICROCOMPUTER GRAPHICS.
Roy E. Myers. Provides the essential mathematics and programming techniques you will need for computer graphics applications in BASIC, and many other topics. **Publisher's Price: $12.45.**

3. THE COMPLETE HANDBOOK OF PERSONAL COMPUTER COMMUNICATIONS: Everything You Need to Go Online with the World.
Alfred Glossbrenner. This digested collection of detailed information on the far-flung aspects of networking and data bases shows you how to gain access to a universe of information with your small computer. **Publisher's Price: $14.95.**

4. WRITING IN THE COMPUTER AGE: Word Processing Skills and Style for Every Writer.
Andrew Fluegelman and Jeremy Joan Hewes. A manual for everything from types of equipment to the fine points of formatting the finished work. **Publisher's Price: $10.95.**

5. CP/M AND THE PERSONAL COMPUTER.
Thomas A. Dwyer and Margot Critchfield. A crisp, detailed tutorial that will easily turn the average reader into a skilled CP/M user. Features lab exercises in each chapter. **Publisher's Price: $19.95.**

6. EXECUTIVE COMPUTING.
John M. Nevison. 25 business methods illustrated by computer programs, in a step-by-step casebook. You'll find details on pricing techniques . . . long-range planning . . . corporate data bases . . . and much more. **Publisher's Price: $12.45.**

SMALL COMPUTER BOOK CLUB is a division of the Library of Computer and Information Sciences, the oldest, largest and most respected computer book club in the world. The SMALL COMPUTER BOOK CLUB will keep you up-to-date with the latest developments in software, improvements in hardware, programming languages, and much more—all at handsome discounts.

So start enjoying the club's benefits today.

4 Good Reasons to Join

1. The Finest Books. Of the hundreds of books submitted to us each year, only the very finest are selected and offered. Moreover, our books are always of equal quality to publishers' editions, *never* economy editions.
2. Big Savings. In addition to getting the Microcomputing Library for $2.95 when you join, you keep saving substantially—up to 30% and occasionally even more. (For example, your total savings as a trial member—including this introductory offer—can easily be over 50%. That's like getting every other book free!)
3. Bonus Books. Also, you will immediately become eligible to participate in our Bonus Book Plan, with savings up to 70% off the publishers' prices.
4. Convenient Service. At 3-4 week intervals (16 times per year) you will receive the Book Club News, describing the Main Selection and Alternate Selections, together with a dated reply card. If you want the Main Selection, do nothing and it will be sent to you automatically. If you prefer another selection, or no book at all, simply indicate your choice on the card, and return it by the date specified. You will have at least 10 days to decide. If, because of late mail delivery of the News, you should receive a book you do not want, we guarantee return postage.

If the reply card has been removed, please write to the **Small Computer Book Club, Dept. Y-AH7, Riverside, N.J. 08075** to obtain membership information and an application.

Compute 4/84

Lyco Computer Marketing & Consultants

TO ORDER CALL US **TOLL FREE 800-233-8760**

In PA 1-717-327-1824

PERCOM

FOR ATARI COMPUTERS

AT88S1 ....$299.00
AT88S2 ....$535.00
AT**S1PD...$439.00
RFD40S1...$399.00
RFD40S2...$675.00
RFD44S1...$449.00
AT88 doubler

HARD DISK DRIVES for APPLE IBM-PC

5MEG.....$1349.00
10MEG ...$1599.00
15MEG ...$1999.00
20MEG ...$2359.00
Add $30.00 for TRS 80 Drives

TEXAS INSTRUMENT

Disk Drive...$245.00

TRAK DISK DRIVES

AT-D1 ....$379.00
AT-D2 ....$399.00
PRINTER CABLE $22.95
Software for ATD-2 ....$22.95

RANA DISK DRIVE

COMPUTER CARE

BIB
5¼ DISK DRIVE CLEANER...........$12.75
COMPUTER CARE KIT $19.75

BLANK DISKETTES

ELEPHANT

Single Side SD (10).......$17.75
Single Side DD (10).......$21.75
Double Side DD (10)......$26.75

MAXELL

MD I (10)....................$28.75
MD II (10) $38.75

CERTRON CASSETTES

CC-10 12 for $15.99
CC-20 12 for $17.99

INNOVATIVE CONCEPTS

Disk Storage (holds 10)....$4.95
Disk Storage (holds 15)....$9.95
Disk Storage (holds 50)....$26.95

apple®

SSI

Battle of Shilo $26.75
Tigers in the Snow.... $26.75
Cosmic Balance $26.75
Knights of the Desert . $26.75
Battle for Normandy .. $26.75
Germany 1985 .... $36.75

RANA DISK DRIVES

Elite 1.............$295.00
Elite 2.............$449.00
Elite 3.............$559.00

MICRO-SCI

A2 ....call
A40 ....call
A70 ....call

MUSE

Castle Wolfenstein $20.75
Caverns of Frietag $20.75
Robot War..............$26.75

CONTINENTAL

Home Accountant $51.75
Book of Apple Software . $16.75

BRODERBUND

Bank Street Writer....... $49.75
AE $24.75
LODE RUNNER D $24.75
Choplifter $24.75
David's Midnight......... $24.75

SPINNAKER

Kindercomp.............. $21.75
Story Machine $23.75
FaceMaker............... $23.75
Snooper Trooper......... $29.75
Delta Drawing............ $34.75

EPYX

Temple of Apshai.......... $26.95
Star Warrior $26.95
Crush, Crumble & Chomp . $22.75

ADVENTURE

Saga#1 Adventureland ...$29.95
Saga#2 Pirate Adventure.$29.95
Saga#3 Secret Mission...$29.95
Stone of Sisyphus $24.95

ALIEN GROUP

Atari Voice Box $99.00
Apple Voice Box......... $129.00

commodore

1212 Programmers Ad.$44.75
1213 Vicmon...........$44.75
Vic 20 dust cover........$6.99
Vic 64 dust cover........$6.99

TIMEWORKS

INVENTORY..............$59.75
ACCOUNTS REC. $59.75
ACCOUNTS PAY. $59.75
GENERAL LEDGER.......$59.75
PAYROLL.................$59.75
CASH FLOW $59.75
SALES ANALYSIS.........$59.75
ELEC. CHECKBOOK......$59.75
MONEY MANAGER $59.75
DATA MANAGER..........$59.75
WALL STREET $59.75

HES 64

Sound Box $9.95
64Forth $55.75
Hesmon................$25.75
Turtle Graphics.......$37.75
Heswriter $28.75
Gridrunner............$19.75
Attack of M C $22.75
Turtle Trainer..........$22.75
Turtle Tutor............$22.75
Paint Brush $22.75
Benji $25.75
Home Manager........$28.75
Time Money Mgr $44.97
OmniCalc $33.75
Sword Point............$19.95
Hes Modem...........$52.75

CARDCO

Cardprinter / LQ1....... $499.00
Cardprint DM1 .... $109.00
5 Slot Expansion 64 ..... $54.00
64 Write NOW............ $39.00
64 Mail NOW............. $29.00
2J Write NOW............ $29.00
64 Keypad $29.00
Universal Cass. Int....... $29.75
Printer Utility............. $19.75
6 Slot Expansion......... $79.95
3 Slot Expansion......... $24 95
PRINTER INTERFACE........ $39.75
PRINTER INTERFACE with full graphics $65.75
LIGHT PEN $29.75

PARKER 20

Frogger (ROM) $33.75
QBert (ROM) $33.75
Tutankham (rom) $33.75

SPINNAKER 64

Kindercomp................$21.75
Story Machine $23.75
Face Maker $23.75
Snooper Trooper...........$29.75
Delta Drawing..............$34.75
Shamus II c/d.............. $24.95
Pinhead c/d $22.95

QUICK BROWN FOX

QBF Word Processor $49.95

LJK

Letter Perfect............ $105.00
Data Perfect.............. $95.00

ADVENTURE INTERNATIONAL

S. Adams Adventure..... $28.75

VIC-64

Household Finance C/D...$24.75

VIC 20

King Arthurs Heir Cass $24.75
Monster Maze Rom $24.75

ATARI

Computers for people.

600XL ...$CALL
800XL........for
1400XL...Lowest
1450 .....Prices

1020 PRINTER...........NOW
1025 PRINTER..............IN
1027 PRINTER ....STOCK
1050 DISK DRIVE ....$SAVES
1010 RECORDER......$74.75

SYNAPSE

BLUE MAX C/D $24.75
Ft. APOCALYPSE C/D...$24.75
PHAROAH'S CURSE C/D...$24.75

FIRST STAR

ASTRO CHASE C/D....$22.75
BRISTOLS C/D.........$22.75
FLIP FLOP C/D $22.75

ALIEN GROUP

Voice Box 2 . $99.75

DON'T ASK

Sam $41.75
Abuse $15.95
Teleatri...... $27.95
Poker Sam .. $24.95

APX

3R Math $19.95
Typo Attack $24.95
Family Budget $19.95
F. Cash Flow.............. $19.95

BRODERBUND

Bank Street Writer D...... $49.75
AE D....................... $24.75
Apple Panic D............ $23.75
Choplifter ROM $32.75
David's Midnight.......... $24.75
Stellar Shuttle C/D........ $18.75
Ft. Apocalypse $24.75

EPYX

GATEWAY TO ASPHI R..............$28.75
JUMPMAN JR R $28.75
PIT STOP R $28.75
GATEWAY TO

SSI

Battle of Shilo C/D.. ......$26.75
Tigers in the Snow C/D....$26.75
Battle for Normandy C/D ..$26.75
Knights of the Desert C/D .$26.75
Cosmic Balance C/D $26.75

ON-LINE

Frogger $24.95
Wizard & Prin $26.95

ROKLAN

Wizard of War.............$29.75
Gorf $29.75
Delux Invader..............$27.95

BIG 5

Miner 2049................ $32.75

EASTERN HOUSE

Monkey Wrench 2 ....$52.75

BRODERBUND

LODE RUNNER D......$24.75
OPERATION WIRLWIND D $29.75
DROL D $24.75

PARKER BROTHERS

Tutankham R............$33.75
Super Cobra R $33.75
Astro Chase R...........$33.75
Frogger R $33.75
QBert R $33.75
Popeye R................$33.75
Risk R $42.75
Chess R $42.75

SPINNAKER

Story Machine R $26.75
Face Maker R............$24.75
Kinderomp R.............$20.75
Fraction Fever R........$24.75
Delta Drawing R.........$26.75

Lyco Computer Marketing & Consultants

TO ORDER CALL US — TOLL FREE **800-233-8760**
In PA 1-717-327-1824

PRINTER INTERFACING
PRINTER PAPER AVAILABLE
Available for IBM PC, Apple, Atari, Vic 20 & Vic 64

SAVE on these in-stock PRINTERS

LETTER QUALITY

SMITH CORONA TP2...$449.00

DIABLO 630 ..$1719.00

ALPHACOM 42	$89.00
ALPHACOM 81	$129.00
NEC 8023	$369.00
NEC 8025	$699.00
NEC PC-8200 COMPUTER	$CALL

EPSON

RX-80	$SAVE$
RX-80FT	ON
FX-80	In-Stock
FX-100	EPSON
MX-80FT	PRINTERS
MX-100	$$CALL$$

MANNESMANN TALLY

SPIRIT 80	$CALL
MT 160L	$CALL

OKIDATA

80	$SAVE$
82A	CALL for
83A	LOWEST
84	PRICES
92	on these
93	In-Stock
PACEMARK 2350	PRINTERS

ATARI 850 REPLACEMENTS IN-STOCK

CITOH

GORILLA GX100	$179.00
PROWRITER 8510	$339.00
PROWRITER II	$659.00
8600	$1025.00
STARWRITER	$1099.00
PRINTMASTER	$1499.00

STAR MICRONTICS

GEMINI 10X	$269.00
GEMINI 15X	$CALL
DELTA 10	$479.00

MODEMS

ANCHOR MARK I	$79.00
ANCHOR MARK II	$79.00
HAYES SMART	$239.00
HAYES MICRO II	$309.00
Micro Bit MPP-1000	$129.75
NOVATION	
CAT	$144.00
D-CAT	$155.00
J-CAT	$115.00
APPLE CAT II	$279.00
212 APPLE CAT	$589.00

CORDLESS TELEPHONES from...$69.75

MONITORS

Sakata Color	$229.00
Amdek Color I	$275.00
Amdek 300 Green	$149.00
Amdek 300 Amber	$149.00
Gorilla Green	$99.00

DUST COVERS

800	$3.99
400	$3.99
1200	$3.99
410	$3.99
810	$3.99
1050	$5.99
PROWRITER	$5.99
GEMINI 10X	$5.99
PERCOM DISK	$5.99
VIC 20/64	$5.99

apple

APPLE DUMPLING GX	$99.75
APPLE DUMPLING 64 (16 Buffer)	$179.75
INFOCOM	
Zork I, II, or III	$26.75
Deadline	$33.75

commodore

HES 20

HES MON Rom	$25.75
Turtle Graphics	$25.75
HES Writer	$25.75
Shamus	$25.75
Protector	$25.75

VIC 64/20

PACMAN	$33.75
DONKEY KONG	$33.75
DIG DUG	$33.75
DEFENDER	$33.75
CENTIPEDE	$33.75

FIRST STAR 64

BRISTOLS C/D	$22.75
FLIP FLOP C/D	$22.75

SYNAPSE 64

ZEPPELIN C/D	$24.75
BLUE MAX C/D	$24.75
DIMENSION X C/D	$24.75

EPYX 64

ASPHI R	$28.75
JUMPMAN JR R	$28.75
PIT STOP R	$28.75

BRODERBUND 64

BANK STREET WRITER	$49.75
CHOPLIFTER	$24.75
LODE RUNNER	$24.75
DROL	$24.75

KOALA TOUCH TABLET...$69.75

PERCOM

AT88S1 ..$299.00
RDF44SI $449.00

GEMINI 10 .... $269.00

	IBM	ATARI	APPLE
PACMAN	$27.95	$29.75	$27.95
DONKEY KONG	$27.95	$29.75	$27.95
DIG DUG	$27.95	$29.75	$27.95
DEFENDER	$27.95	$29.75	$27.95
CENTIPEDE	$27.95	$29.75	$27.95
KOALA TOUCH TABLET	$99.75	$69.75	$84.75
DEADLINE	$34.75	$34.75	$34.75
ENCHANTER	$34.75	$34.75	$34.75
INFIDEL	$34.75	$34.75	$34.75
PLANETFALL	$34.75	$34.75	$34.75
STAR CROSS	$34.75	$34.75	$34.75
SUSPENDED	$34.75	$34.75	$34.75
WITNESS	$34.75	$34.75	$34.75
ZORK I	$34.75	$34.75	$34.75
ZORK II	$34.75	$34.75	$34.75
ZORK III	$34.75	$34.75	$34.75

48K RAM $75.00
64K RAM $99.00

ATARI WRITER $79.00

BUSINESS

Visicalc	$159.75
Letter Perfect	$89.75
Letter Perfect	$89.75
Data Perfect	$89.75
TEXT WIZZARD	$34.75
SPELL WIZZARD	$34.75
File Manager	$69.75
Home File Mgr	$69.75
Bookeeper	$119.75
C.R.I.S.	$199.75

ATARI
Computers for people.

MasterCard VISA

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 1774C

POLICY

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

```
10 FORI=0TO32:READA:POKE52128+I,A:NEXT
20 DATA32,0,192,165,20,141,174,203,32
30 DATA0,192,76,101,199,0,32,14,198,174
40 DATA174,203,240,3,76,132,200,73,255
50 DATA49,251,76,134,200
60 J=51042:K=51329
70 POKEJ,76:POKEJ+1,160:POKEJ+2,203
80 POKEK,76:POKEK+1,175:POKEK+2,203
```

Charles Tyson

Atari Mystery Commands

I frequently see statements in programs that just don't seem to make sense. For example, in "Diamond Drop," Atari version (COMPUTE!, September 1983), lines 5430 and 5750 have the statement Q=1∧1. As I understand it, that means 1 raised to the first power, which is 1 times 1, which is just 1. So why write it this other way? Also, if you remove it, and just change the statements to Q=1, then the diamonds drop faster, but why?

Also, in some programs there will be a loop FOR X=0 TO 1 STEP 0. What is the meaning of this? If you start at zero with a STEP of 0, what happens?

Roy R. Valantine

The Atari takes a significant amount of time to calculate powers and roots. Matt Giwer, the author of "Diamond Drop," took advantage of this and used Q=1∧1 in place of a delay loop to slow down the game. This statement takes up less memory than an empty FOR-NEXT loop. To answer your second question, remember that the STEP value is added to the original value in a FOR-NEXT loop at every NEXT. With a STEP of zero, X will never become equal to one. This is just a faster way of writing an endless loop. Instead of:

```
10 PRINT I:I=SQR(I)
20 GOTO 10
```

you can use:

```
5 FOR X=0 TO 1 STEP 0
10 PRINT I:I=SQR(I)
20 NEXT X
```

Execution will continue until you press BREAK or SYSTEM RESET.

64 SID Register Images

In the 64 memory map there is a sizable chunk of memory labeled "SID Images" ($D500–$D7FF). Presumably the audio synthesizer needs this memory to function, but no explanation is offered by any manual. Could this memory be used to further manipulate SID or is it strictly hands-off?

Rob House

The 64 uses a method known as memory-mapped I/O *to let you access a hardware chip's registers with PEEK and POKE (or load and store in machine language). This is convenient and fast, but some portion of your computer's memory must be dedicated to the chip. No actual memory (RAM or ROM) is used or needed, but the memory addresses are set aside for the chip's use only. The SID chip has only 26 registers, but to make interfacing easy, Commodore allocated it 1K of address space. It's a quirk of the interface chips which causes "reflections" of the SID chip's actual registers to appear from $D500–$D7FF. You can read or change these locations, and they will be redirected to the SID chip. This is not recommended simply because it is nonstandard—no one else, perhaps even you at a later date, will be able to figure out why you are POKEing to these image locations. Also, there is RAM beneath the ROMs and hardware chips, so you can switch out the SID chip to make full use of the underlying 1K of RAM if you want.*

Machine Language To BASIC?

I would like to know if there is a program to translate a machine language program into BASIC. I realize that this sounds a bit impractical, but I think it would be a tremendous aid to us BASIC programmers who are trying to learn machine language.

It seems to me that if a BASIC Interpreter can translate BASIC instructions into machine language, that the opposite is equally possible. If it cannot be done, please explain why.

D. W. Bruce

The BASIC Interpreter doesn't actually translate BASIC into machine language. All the operations that can be performed from BASIC are permanently stored as machine language routines in Read Only Memory (ROM). The Interpreter first converts each BASIC keyword into a one-byte number called a token. *Then, depending on the value of the token, certain of the permanent machine language routines are executed in a predetermined order.*

It is not possible to automatically translate machine language programs into BASIC equivalents. You could look at a machine language program and design a BASIC version of the same thing, but BASIC runs so much slower that it would serve no purpose other than being the educational exercise you mention.

Listing The 1541 Directory

I own a Commodore 64 and a 1541 disk drive, and I have a question: How do I go about listing the directory of a disk?

Danny Chu

After making sure that your computer and disk drive are on, and no BASIC programs are in memory, insert

CHART YOUR PROGRESS

SPECIALS OF THE MONTH

Atari 600XL	$189
Atari 800XL	$299
Atari 1030 Modem	$115
Rana 1000 Disk Drive	$339
1027 Letter Quality Printer	$299
Commodore SX64 Portable Computer	$959
Commodore Automodem	$ 95
Commodore Graphics Printer	$269
Verbatim Twin Pak	$ 5.25
Elephant Disks S/S	$19.50

HARDWARE SPECIALS

Printers

Leading Edge Gorilla	$189
NEC 8023A	$459
Okidata ML 82A	Call
Okidata ML 83A	Call
Okidata ML 84P	Call
Okidata ML 92P	Call
C. ITOH 8510 Prowriter	$399
C. ITOH 8510 SP	$665
Mannesmann Tally 160L	$629
Gemini 10X	$315
Mannesmann Tally Spirit	$345
Alphacom 81	$189

Monitors

Leading Edge Green Hi-Res 12"	$ 89
USI Amber	$169
NEC Green Screen	$169
Leading Edge Amber Hi-Res 12"	$ 89
Amdek Color 1+	$315

RECREATION

Q Bert/CRT	$ 35
Frogger/CRT	$ 35
Popeye/CRT	$ 35
Deadline	$ 35
Zork I, II, III ea.	$ 27
Witness	$ 35
Planetfall	$ 35
Enchanter	$ 35
Infidel	$ 35
Temple of Apshai	$ 27
Pitstop	$ 27
Starleague Baseball	$ 23
Starleague Football	$ 23
Blue Max	$ 23
Joust	$ 43
Robotron 2084	$ 33
Pole Position	$ 36
Ms. Pac Man	$ 36
B-1 Nuclear Bomber (C)	$ 12
Midway Campaign (C)	$ 12
Nukewar (C)	$ 12
Telengard (C)	$ 16
Jumpman Jr./CRT	$ 27
Pitstop/CRT	$ 27
Flight Simulator II/D	$ 39
Night Mission Pinball	$ 20
Sammy Light Foot (CRT)	$ 20
Apple Cider Spider	$ 23

EDUCATION

Atari

Harcourt Brace SAT/D	$59
Chaterbee/D	$29
Match Wits/D	$22
Early Games/D	$22
Early Games Music/D	$22
Spellicopter	$29
Creature Creator	$29
Math Maze	$29
Alien Addition/D	$25
Meteor Multiplication	$25
Compu-Read/D	$22
Compu-Math/Fractions	$29
Compu-Math/Decimals	$29
Spelling Bee Games	$29
Speed Read +/D	$45
Spider Eater/Koala	$22
Bumble Bee/Learning Co.	$29
Snooper Troops 1, 2 ea.	$32
Face Maker/CRT	$25
Kids on Keys/CRT	$25
Alphabet Zoo/CRT	$25
Kid Writer/CRT	$25
Delta Drawing/CRT	$29

Commodore

CBS Addition/Subt.	$17
CBS Multiplication/Division	$17
Speed Reader II/D	$49
Word Attack/D	$36
Mathblaster/D	$36
Spellakazan/D	$29
Crypto Cube/D	$29
Master Type/D-CRT	$29
Songwriter/D	$29
Alphabet Zoo/CRT	$22
Fraction Fever/CRT	$22
Delta Drawing/CRT	$22
Facemaker/CRT	$25
Trains/D	$29
Kidwriter/CRT	$25
Dungeons of Algebra Dragons	$17
Juggles Rainbow	$22
Bumble Bee	$29
Early Games Piece of Cake	$22
Early Games Match Maker	$22

BUSINESS

Atari

Atariwriter	$ 79
Visicalc	$179
Financial Wizard	$ 45
Letter Perfect	$ 69
Data Perfect	$ 79
Spell Perfect	$ 65
Spell Wizard	$ 45
Text/Spell Wizard Combo	$ 65
Bank Street Writer	$ 49
Bank Street Speller	$ 49
Homeword	$ 49
Syn File +	$ 65
Syn Text	$ 23
Syn Mail	$ 34
Home Accountant	$ 48
Tax Advantage	$ 39
Complete Personal Accountant	$ 65
Atspeller	$ 33
Diskette Mailing List	$ 21
Miles Payroll System	$119
Atari Accounting	$169
Family Finance	$ 36
File Manager 800+	$ 65

Commodore

Easy Script 64/D	$ 35
Easy Spell 64/D	$ 16
Easy Finance I, II ea./D	$ 16
Accts. Receivable/D	$ 35
Payroll/Checkwriting/D	$ 35
Assembler 64/D	$ 15
Logo/D	$ 49
Simon's Basic/CRT	$ 17
Bank Street Writer/D	$ 49
Home Accountant/D	$ 48
Tax Advantage	$ 45
FCM	$ 35
Paper Clip Word Processor	$ 89
Delphi's Oracle	$ 89
Practicalc 64	$ 40
Calc Result/Easy-CRT	$ 59
Calc Result/Advanced-D	$109
Multiplan/D	$ 65
General Ledger/D	$ 59
Data Manager/D	$ 17
Electronic Checkbook	$ 17
Swif tax/D	$ 35
Quick Brown Fox/D	$ 59
Quick Brown Fox/CRT	$ 56

COMPUTER OUTLET

COMPUTER OUTLET

1095 East Twain, Las Vegas, NV 89109 • Mon.-Fri. 8 a.m. to 6 p.m., Sat. 9 a.m. to 5 p.m.

1-800-634-6766

VISA

Information and Inquiries 702/796-0296 • Order Status 702/369-5523

Prices reflect cash discount only and are subject to change.

SHIPPING: For fast delivery, cashier checks, money orders or direct bank wires. Personal and company checks, allow 3 weeks to clear. C.O.D. charges: $3 minimum or 1% on orders over $300. Nevada residents add 5¾% sales tax. Shipping charges based on weight. $3 minimum. APO and FPO orders: $10 minimum and 15% on all orders over $100. School and business purchase orders welcome. All returns must be accompanied by return authorization number. Call 702/369-5523 before returning goods for replacement.

your disk into the drive, and type LOAD "$",8 and press RETURN. The disk drive will whir and spin a bit. After the drive has stopped and the READY with the blinking cursor is displayed on the screen, type LIST and press RETURN. The directory will be displayed on the screen. To slow down the listing, press CTRL on the keyboard.

However, there is one thing to watch out for. Listing the directory using this method puts it in the BASIC programming area, so any BASIC program in memory will be overwritten. Also, when you're through looking at the directory, enter NEW to clear the BASIC memory pointers before loading a BASIC program.

If you are searching through several disks for a particular filename, you can have the LOAD command display only that filename. Let's say you're looking for a game called "Space Game." Here's the format:

```
LOAD "$0:SPACE GAME",8
```

Type LIST, and if the filename is found it will be displayed on the screen. If the filename is not on that disk, only the disk header (name of the disk) will be displayed. If you forget how you spelled the name of the file you want, use the wildcard feature, represented by an asterisk. For example, if you enter LOAD "$0:SPACE ",8 it will display all filenames starting with the letters SPACE such as SPACE GAME, SPACEMAN, etc. If you enter LOAD "$0:S*",8 you'll get all the filenames starting with an S.*

PEEKing The VIC SHIFT Keys

I own a VIC and would like to know the PEEK command for the right SHIFT key. I know the left SHIFT key is PEEK(145). I realize PEEK(653) could be used, but I only want to know when the right SHIFT key is pressed, not the left. Also, what is the PEEK command for the RESTORE key?

Chris Stroud

The following information, excerpted from the upcoming COMPUTE! book Mapping the VIC-20 *by G. Russ Davies, should be of help.*

"Each time the jiffy clock TIME is updated by the Kernal, the contents of VIA2PA2 (VIA2—Port A) are copied to location 145.

"Every other key on the bottom row of the keyboard may be tested for in this location, without using a GET command in BASIC.

"Here's an explanation of the different values you'll find in this location:

PEEK	Hex Number
255	($FF) = no key pressed
254	($FE) = STOP key pressed; STOP routine will find and act on
253	($FD) = left SHIFT key pressed. This may be the most useful returned value, especially for game programmers. It allows the program to distinguish between the left and right SHIFT keys by checking location 653, then location 145, bit 1 for left/not left.
251	($FB) = X key pressed
247	($F7) = V key pressed
239	($EF) = N key pressed
223	($DF) = comma key pressed
191	($BF) = slash key pressed
127	($7F) = cursor-down/up key pressed

"You can examine the values in this location when the bottom row keys are pressed by entering and running this short program:"

```
10 ?PEEK(145):GOTO 10
```

In other words, if location 653 says SHIFT is down, but location 145 doesn't, then the right SHIFT key is down. Otherwise, the left SHIFT key is down (or both left and right SHIFT keys together).

The RESTORE key is not a normal keyboard key. It is not wired through the VIA, but directly to the 6502's NMI line. When RESTORE is pressed, a Non-Maskable Interrupt is generated, which makes the 6502 jump to a special NMI routine. This routine checks to see if the RUN/STOP key is being held down, and if so, executes the soft reset normally performed by RUN/STOP—RESTORE. If RUN/STOP is not held down, the interrupt will occur when you press RESTORE, but nothing will seem to happen. It's not easy to trap and re-vector the RESTORE key, but the vector for the routine we've mentioned is at 792 and 793.

The Atari Speaker

I would like to know if it is possible to easily turn the Atari internal speaker off and on, without opening the computer case.

Paul Stach

On the XL series, you can toggle the speaker off and on by holding down the Control key and pressing the F3 key. There is no easy way to turn off the internal speaker on the 400 or 800.

Screen POKEs On The 64

I recently bought a second Commodore 64 and found that if you try to POKE a character to the screen, it won't appear. For example, POKE 1024,1 should put the letter A in the first position of the screen, but it doesn't. I took the new 64 back and got another one, but it had the same bug. My old 64 from December 1982 works OK. Has Commodore changed the design of the new units? If so, why not put an addendum in the user's manual? In the meantime, I've returned both defective 64s. Can you elaborate on these differences in the 64s?

Neil Hoover

Commodore has made design changes since the very first versions of the 64, but by no means is this a bug.

In the early versions of the 64, color memory was automatically filled with 1's—the value for the color white. You could then POKE characters to the screen without worrying about the color.

In the newer versions of the 64, there is no default filling of color memory. When you POKE characters to the screen (locations 1024–2047) in the newer 64s, you also have to POKE values to the corresponding location in color RAM (bytes 55296–56319). For example, if you POKE 1025,1 (the second position in screen RAM) you also need to POKE 55297,x (the corresponding second position in color RAM). The x can be a value from 0 to 15, corresponding to the 16 character colors available on the 64.

It's good practice to always POKE color memory when POKEing screen memory. That way, your programs will run on any version of the Commodore 64.

Modem Tariff Update

I am happy to report that Southwestern Bell has eliminated the requirement that modem users obtain Information Terminal Service for their modems. While the problem has been taken care of here in Oklahoma, I have not heard anything which rescinds rumors of similar tariffs in other states. Therefore, I would like to extend an invitation to your readers to telephone me should they experience similar problems with their local telephone company.

Robert Braver
MCI Mail: RBRAVER
Source: STY801
Telex: 650-116-9625
Telephone: (405) 360-7462

*Robert Braver is the author of "Guest Commentary: The High Cost Of Personal Telecommunications" (*COMPUTE!*, November 1983). His commentary dealt with the then-unresolved question of whether Southwestern Bell, and other telephone companies, should charge home computerists a higher rate (tariff) to connect modems to their residential telephone lines.*

Automatic LOAD And RUN For Commodore

I own a Commodore 64 and would like to know if, after LOADing a program from tape or disk, there is a way to make it automatically RUN without having to actually type in the RUN command?

Jeffrey Stevens

On Commodore computers, you can LOAD and automatically RUN a program from disk or tape. For tape LOADs, simply hit the RUN/STOP key while the SHIFT key is depressed. The first program that is found on the tape will LOAD and RUN. If you wish to LOAD and RUN a program other than the first one, type in LOAD "Program Name",1: and then press the RUN/STOP key with the SHIFT key down. In this case, the program designated by Program Name will LOAD and automatically RUN.

A similar procedure can be followed if you are using a disk drive. Type LOAD "",8: and then press SHIFT RUN/STOP to LOAD and RUN the last program accessed (if no program resides in memory, the first program listed in the directory will be LOADed). For programs other than the first one on the disk, type LOAD "Program Name",8: and press SHIFT RUN/STOP.*

TI Cartridge Loading Problem

In answer to Charles Smith's inquiry (COMPUTE!, January 1984) regarding difficulty in repeated loading of TI-99/4A cartridges, several of us in a user group have had the same problem. It is the result of TI's use of a right-angle adapter for the cartridge socket.

The horizontal socket is mounted 90 degrees to a vertical connector board, which then plugs downward into the motherboard. Continued horizontal motion eventually loosens the pin tension in the motherboard socket. Unfortunately, the solution requires disassembly of the computer to gain access to the motherboard socket.

Charles Smith will have to decide whether he wants to tackle this on his own (possibly voiding his warranty), or refer the job to an authorized service center.

Donald C. McMahon

Long Commodore Tape Saves

With long programs it is often not possible to finish typing in a listing in one sitting. I have been saving part of the program on tape and when I continue the second part, I erase the first part on the tape and rerecord the whole thing.

I would like to just continue recording with the next line number of the program. Please advise if there is a way.

J. R. Blundin

When you SAVE to tape, the computer first writes a header which contains information such as program name and starting address. Then the program is saved twice, and an end-of-file marker is written when the tape file is closed. (Commodore computers save programs twice as a safety measure.)

These SAVE procedures make it impossible to position the tape and append the second half of the program onto the first. Attempting this would probably result in a DATA error, and both halves might be lost.

During long typing sessions, you should SAVE your incomplete program occasionally (every hour or so). This way, if some sort of catastrophe happens, you won't lose the entire program. You can load the last version you saved and continue typing.

Also, it's not a good idea to continually rewind the tape and keep saving programs over the same section of the tape. Saving programs consecutively will afford another safety measure—you'll always have backup

ATARI®

600XL	CALL
800XL	CALL
1200XL	$409*

*Reflects $100 Atari Rebate

1050 DRIVE	$335	MICROBITS INFC	$78
1025 PRINTER	$399	80 COLUMN BD	$249
1020 COLOR PTR	$219	TECHNICAL NOTES	$25
1027 PRINTER	$279	REAL TIME CLK	$38
1010 RECORDER	$72	810 DRIVE	$419

DRIVES

ASTRA 1620	$469
RANA 1000	$295
PERCOM 88-S1	$298
PERCOM 40-S1	$418
PERCOM 40-S2	$718
PERCOM 44-S1	$510
PERCOM 44-S2	$929

MEMORIES

48K RAM (INTEC)	$95
64K RAM (INTEC)	$119
48K RAM (MOSAIC)	$109
64K RAM (MOSAIC)	$145
128K RAM DISK	$299
32K RAM (MOSAIC)	$68

SPECIALS

Gemini 15X Printer	$399
Axiom AT-100 Printer (with interface	$229
Astra Double Density Dual Drive	$469
Rana 1000 Drive	$319
Bit-3 80 Column Board	$245
Mannesmann Talley 160L Printer	$589
Atari 400 Keyboard (In Home)	$35
Programmer Kit $48 / Entertainer Kit	$64
Wico Joystick $23 / Wico Trackball	$49

ASTRA 1620

DISK DRIVE SYSTEM

MORE FOR YOUR MONEY
DOUBLE OR SINGLE DENSITY
TWO DRIVES

SPECIAL $469

COMMODORE

CBM 64	CALL
1541 DISK DRIVE	$239

1701 Color Monitor	$255	1530 Recorder	$59
1525 Printer	$239	1600 Modem	$59
1520 Color Ptr	$169	1650 Auto Modem	$89
Card ? (Infc)	$60	CMB 64 Ref Guide	$18
Light Pen	$29	The Connection (Infc)	$85
Cassette Infc	$29	MSD Disk Drive	$339
Card ? Software	$16	PTI 45 Lot Board	$59

ATARI SOFTWARE

ADVENTURE INT'L
Adv. 1-12 each (C) $18
Preppie (C/D) $20
Preppie II (C/D) $23
Diskey (D) $33
Sea Dragon (C/D) $23

APX
Eastern Front (C/D) .. $23
747 Land Sim. (C/D) .. $17
Fig-Forth (C) $30

ATARI INC.
Microsoft Basic II (R) . $62
Mickey in Great Outdoors (C/D) $36
Paint (D) $30
Speed Reading (C) ... $54
Qix (R) $30
Dig Dug (R) $30
Atari Writer (R) $68
Donkey Kong (R) $30
Time Wise (D) $23
Visicalc (D) $139
Juggles House (C/D) . $22
Juggles Rnbw (C/D) .. $22
Pilot (Home) $55
Galaxian $30
Defender $30
ET $34
Microsoft Basic (D) .. $62
Macro Ass. & Edit (D) . $62
Assembler Editor (R) . $42
Basic Cartridge (R) ... $45
Pac Man (R) $30
Centipede (R) $30
Caverns of Mars (D) .. $28
Star Raiders (R) $30
Conv. Lang. Ea. (C) ... $42
Music Composer (R) .. $31
Super Breakout (R) ... $26
My First Alphabet (D) . $26
Prog. 2 & 3 (ea.)(C) $21
Word Processor (D) . $102
Pilot (Educ.) $92
Touch Typing (C) $19
Home File Mngr (D) .. $36

AUTOMATED SIMUL.
Hellfire Warrior (C/D) . $27
Kng Arthr's Heir (C/D) $20
Invasion Orion (C/D) .. $17
Temple of Aps. (C/D) . $27
Star Warrior (C/D) $27
Dragon's Eye (D) $20
Crush Crumble (C/D) . $20

AVALON HILL
VC (D) $17
B-1 Nuc. Bomber (C) .. $12
Legionnaire (C) $23

BRODERBUND
Sky Blazer (D) $22
Bank St. Writer (D) ... $46
A.E. (D) $23
Arcade Machine (D) .. $39
Choplifter (D) $23

CBS
Mountain King (R) ... $27
Boulders & Bombs (R) $27
Krazy (each) $34

CONTINENTAL SOFT.
Home Accountant (D) $48
Tax Advantage (D) ... $39

DATASOFT
Text Wizard (D) $65
Graphic Master (D) ... $27
Micro Painter (D) $23
Lisp Interpreter (D) ... $79
Graphics Gen. (D) $17
Basic Compiler (D) ... $65
Zaxxon (C/D) $27

DON'T ASK
Sam (D) $39
P.M. Animator (D) $23
Teletari (D) $27

EDU-WARE
Prisoner II (D) $27
Spelling Bee (D) $27
Compu-Read (D) $20
Compu-Math Fr. (D) .. $27
Compu-Math Dec. (D) . $27

EDUCATIONAL SOFT.
Tricky Tutorial 1,2,3 or 4 (C/D) $15
Tricky Tutorial 5,6 or 7 (C/D) $22

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

JV SOFTWARE
Jrny to Plnts (C/D) $20
Action Quest (C/D) ... $20
Ghost Encount. (C/D) . $20

LJK
Letter Perfect (D) $74
Data Perfect (D) $74

ON-LINE
Ultima II (D) $39
Marauder (D) $23
Lunar Leeper (D) $20
Wiz & Princess (D) ... $22
Frogger (C/D) $23
Crossfire (R) $23

OPTIMIZED SYSTEMS
C-65 (D) $58
Bug-65 (D) $23
Max-65 (D) $58
Basic A + (D) $58

ROKLAN
Gorf (D) $27
Gorf (R) $30
Wizard of Wor (D) $27
Wizard of Wor (R) $30

SIRIUS
Alpha Shield (R) $27
Wavy Navy (D) $23
Bandits (D) $23

SPINNAKER
Snooper Troop 1,2 (D) .. $30
Kindercomp (D) $20
Rhymes & Riddles (D) $20
Hey Diddle Diddle (D) . $20
Srch Amzng Thngs (D) $27
Story Machine (D) $23
Face Maker (D) $23

STRATEGIC SIM.
Cosmic Balance (D) .. $27
Cosmic Balance II (D) $27
Tigers In Snow (C/D) .. $27
Battle of Shiloh (C/D) . $27
Battle of Norm. (C/D) . $27
Galactic Gladiator (D) $27
Cytron Masters (D) ... $27

SYNAPSE SOFTWARE
File Mngr 800 + ... $65
Protector II (D) $23 (R) $29
Shamus ... (D) $23 (R) $29
Fort Apocalypse (C/D) $23
Shamus II (C/D) $23
Necromancer (C/D) ... $23
Pharoh's Curse (C/D) . $23

THORN EMI
Soccer (R) $34
Jumbo Jet (R) $34
Submarine Comm. (R) $34

USA
Atari World (D) $39
3-D Sprgrphcs (C/D) .. $27

MISCELLANEOUS
Sargon II .. (C) $20 (D) $23
Financial Wizard (D) .. $41
Castle Wolfenstein (D) $20
Master Type (D) $27
Millionaire (D) $52
Astro Chase (D) $22
Ali Baba (D) $22
Miner 2049er (R) $34
Sammy Sea Serp. (C) . $13
Pinball (D) $20

Printers/Etc.

GEMINI 10X	$279	PROWRITER	$345
GORILLA	$199	SMITH TPI	$488

CITOH
Prowriter $345
Prowriter II $629
Starwriter $1149
Printmaster $1448

NEC
8023 A-C $409
3510 .... $1375
3530 .... $1579
3550 .... $1779
7710/7730 .... $1998

SILVER REED P $669
QUME 11/40 + ... $1299

OKI-DATA
Microline 82A $398
Microline 83A $638
Microline 84P $958
Microline 92 .... $488
Microline 93 .... $858

DIABLO
620R $939
630R $1719

MONITORS

AMDEK
Color I $289
V300 .... $139
V300A $149
Color II $449

NEC
GRN (JB1260) $115
GRN (JB1201) $155
Color Composite ... $298
RGB Color $598

MODEMS

HAYES
Smartmodem $209
Smartmodem 1200 .. $498
Micromodem II $259

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

D = Disk
R = ROM (Cartridge)
C = Cassette

COSMIC COMPUTERS
UNLIMITED

727 BREA CANYON RD., SUITE 16
WALNUT, CA 91789
ORDER LINES OPEN MON-SAT 8 am - 8 pm

(800) 626-7642

PLEASE FOR ORDERS ONLY
SORRY, NO COD'S

(714) 594-5204

FOR TECHNICAL INFO, ORDER INQUIRIES, OR FOR CALIFORNIA ORDERS

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.

64 SOFTWARE 64

Multiplan .... CALL
Script 64 .... $77
Calc Result Prof. $114
Calc Result Easy $68
The Home Accountant $48
Delphis Oracle $89
Word Pro 3 with Spell $78

ACCESS SOFTWARE
Neutral Zone (C/D) ... $26
Sprite Master (C/D) ... $27

AVALON HILL
Nukewar (C) $12
Planet Miners (C) $12
Androm. Conquest (C) $14
Midway Campaign (C) $12
North Atl. Convoy (C) . $12
Comp. Stcks/Bnds (C) $15
Computer Football (C) $18
Telengard (C) $16

BATTERIES INCLUDED
Paper Clip (D) $85
Delphis Oracle (D) 89

BRODERBUND
Choplifter (R) $29
Serpentine (R) $27
Seafox (R) $27
David's Midnight (D) .. $23

COMMODORE
Easy File (D) $75
Easy Finance (D) $38
Easy Mail (D) $38
Easy Script (D) $75
Easy Schedule (D) ... $59
Logo (R) $75
Pilot (D) $75
Assembler (D) $38
Music Machine (D) ... $25
Music Composer (D) .. $25
Meza Music (D) $75
Video/Music Supt. (D) $38
Jupiter Lander (R) $25
Radar Rat Race (R) ... $25
Sea Wolf (R) $25
Kickman (R) $25

COMM-DATA
Pakacuda . (C) $14 (D) $18
Escp. MCP . (C) $14 (D) $18
Centropods (C) $14 (D) $18

COMPUTERMAT
Arcade-Pak (C) $18
Education-Pak (C) $18

CREATIVE SOFTWARE
Moondust (R) $25
Trashman (R) $25
Save New York (R) ... $25
Astroblitz (R) $25
Household Fin. (D) ... $25

DATA 20
Video Pak 80 .... $139
Z80 Video Pak $229

EN-TECH
Finance Calc 64 .... $34
Data Base 64 .... $56
Invoice Ease 64 .... $56

EPYX
Temple of APS (D) ... $27
Upper Reach. APS (D) $14
Jumpman (D) $27

HES
HES Modem $59
6502 Prof.Dev.Sys.(C) . $22
Hesmon 64 (R) $27
Turtle Grapics II (R) .. $41
Heswriter 64 (R) $32
Gridrunner (R) $27
Retroball (R) $27

INFOCOM
Zork I, II or III (D) $27
Deadline (D) $35
Starcross (D) $27

JIN SAM
Mini-Jini (R) $75

LITTLE WIZARD
Pro.Mail.List (C)$22 (D)$25
Stockmaster (Inventory) (C)$25 (D)$28

LOGISTIC
Datacalc 64 (C) $55 (D) $59
Home Journal (D) $65

MICROSPEC
Payroll System (D) ... $73
Inventory Pkg (D) $73
General Ledger (D) ... $73
Disk Data Mgr (D) $62
Mail List Mgr (D) $41
Checkbook Mgr (D) ... $39

M-SOFT
M-File (D) $89

ON-LINE
Frogger (D) $23
Jawbreaker (D) $20

PACIFIC COAST SOFT.
PCS (80 Col BD, Word Proc, D.Base, Spreadsheet) CALL
Account PAC (C/D) ... $34
File PAC (D) $30
Editor PAC (D) $39
Inquire PAC (D) $57
Happy Tutor Typng (D) $18

PROFESS. SOFTWARE
Wordpro 3 + /64 (D) ... $68

QUICK BROWN FOX
Prof.Word Proc. (R) ... $50

RAINBOW
Writers Assistant $95
Spreadsheet Assist. .. $95
File Assistant $95

SIRIUS
Blade/Blackpoodle (D) $27
Type Attack (D) $27
Repton (D) $27
Critical Mass (D) $27
Snake Byte (D) $23
Way Out (D) $27
Fast Eddie (D) $23
Turmoil (D) $23
Spider City (D) $27
Squish'Em (D) $23
Final Orbit (D) $27
Alpha Shield (D) $27

SKYLES ELEC. WORKS
Busicalc (C/D) $52
Busiwriter (D) $72

SPINNAKER
Snooper Troops 1 (D) . $29
Facemaker (D) $23
Kindercomp (D) $20
Hey Diddle (D) $20
Most Amaz. Thing (D) . $27

SYNAPSE
Fort Apocalypse (C/D) $23
Survivor (C/D) $23
Drelbs (C/D) $23
Pharoh's Curse (C/D) . $23
Protector II (D) $23
Morgal (D) $23
Shamus (D) $23

TAYLORMADE
Touch Typing Tutor 3.0 (D) $21

TIMEWORKS
Rbbrs/Lost Tomb (C/D) $21
Wall Street (C/D) $21
Money Manager (C/D) $21
Data Master (C/D) $21
Dungeons of Alg. Dragons (C/D) $21

TOTL
Text 2.6 ... (C) $32 (D) $34
Label 2.6 .. (C) $15 (D) $17
Time Manager 2.6 (C) . $24
Time Manager 2.6 (D) . $27
Resrch Assist. 2.0 (C) . $24
Resrch Assist. 2.0 (D) . $27

UMI
Motor Mania (C) $20
Renaissance (C) $27

VICTORY
Annihilator (C/D) $16
Kongo Kong (C/D) $16
Trek (C/D) $14
Adv. Pack #1 (C/D) ... $16
Adv. Pack #2 (C/D) ... $16
Grave Robbers (C/D) .. $13
Chomper Man (C/D) . $18

versions. If, for example, a tape becomes crinkled in one spot, you can rewind to the previously saved version.

Finally, when you SAVE a program, the cassette will perform the save, then stop. The place where it stops is where you want to start the next SAVE. Knowing this, you can save the time and worry involved in positioning or rewinding the tape.

Saving Incomplete Programs

I recently purchased a Commodore 64 along with a 1541 disk drive. I know how to format a disk, SAVE a program, LOAD a program, and RUN it. However, I'm not sure how to SAVE a program that is not complete, then LOAD it back in to finish work on it. Also, if I LOAD a program and it has mistakes in it, I'm not clear on exactly how to correct these errors.

William H. Boothe

An incomplete program is SAVEd the same way a complete program is:

```
SAVE "filename",devno
```

(on Commodore computers, for example) where filename is a name you assign to the file (it can contain up to 16 characters), and devno *is the device number (1 = tape, 8 = disk). The computer does not know (or care) if your program is complete or not.*

To correct errors and make other changes to program lines, display the line you wish to change on the screen by typing LIST followed by the line number. When the line is displayed, move the flashing cursor to the desired place on the BASIC line using the cursor movement keys in the lower right of the keyboard.

To insert a character or characters into the line, move the cursor to the position where you want to insert, then hold down SHIFT and press INST/DEL. This will insert a space each time you press the INST/DEL key. Now type your change, and press RETURN. To delete an unwanted character, move the cursor one character past the one to be removed, press DEL (without SHIFT), then press RETURN.

When you edit a program line and press RETURN, the corrected line is entered into the BASIC program in its proper place. Consult your user manual or Programmer's Reference Guide *for further details.*

Using A VIC Printer With Atari

I own a Commodore VIC-20 and a 1525 Graphic Printer. I am contemplating purchasing a second computer. I realize I could use my printer with a Commodore 64, but is there any way to interface it with another brand of computer, like the Atari 800XL? If so, where can I get such an interface?

Chris Hill

Unfortunately, Commodore and Atari use proprietary serial interfaces. They cannot talk to each other or to each other's peripherals. An interface to convert from Commodore serial and Atari serial is conceivable, but we know of none.

There is one way the two computers can communicate—over a common interface, such as RS-232. Both Commodore and Atari computers can send and receive data over an RS-232 cable or through a modem. All that's required is an RS-232 interface for each computer. You could set up a program on the VIC that sends anything coming in from the RS-232 port right to the printer. The Atari would then send things out to the VIC over its RS-232 port. You could even operate the VIC 1540 disk drive (with the right program in the VIC) to handle the Atari requests for program saves and loads. The latter method would be rather more difficult to implement, however.

Clearing The 64 Screen From ML

Our machine language class has been working on several small routines written in machine language for the 64. One of the subroutines we've written to clear the screen is somewhat lengthy—49 bytes. We feel there must be a shorter way. Can you help?

Art Lipina

Yes, there is a shorter, and an easier, way. There are many routines in both the 64's BASIC ROM ($A000–$BFFF) and Kernal ROM ($E000–$FFFF) that you can use. For example, there is a screen clearing routine at $E544 (decimal 58692). Here are two ways to clear the screen using ROM routines:

```
JSR $E544
```

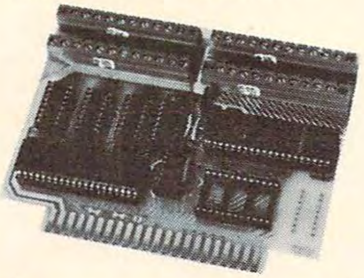
This Kernal ROM routine clears both screen and color memory, and then resets the screen line link table. Clearing the screen using this technique uses only three bytes.

```
LDA  #$93
JSR  $FFD2
```

This method (which uses five bytes) first loads the accumulator with the ASCII character that clears the screen, then calls the Kernal PRINT routine ($FFD2). This is equivalent to the BASIC statement PRINT "{CLR}".

COMPUTE! welcomes questions, comments, or solutions to issues raised in this column. Write to: Readers' Feedback, COMPUTE! Magazine, P.O. Box 5406, Greensboro, NC 27403. COMPUTE! reserves the right to edit or abridge published letters. ©

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



- 16 channel 8-bit A/D converter with 100 microsecond sampling time.
- 1 D/A output.
- 16 high voltage/high current discrete outputs.
- 1 EROM socket.
- Use multiple boards for additional channels up to 6 boards.

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

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

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

MW-302 .... $119.95

Dealer inquiries invited.

VISA MasterCard

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

How to make your computer look as smart as it is.

Store it in a beautiful piece of furniture specifically designed for the proper operation and storage of your home computer equipment.

- Upper unit shelf adjusts to most computers.
- Keyboard shelf at correct typing height with plenty of work surface.
- Monitor placement at proper height and viewing distance eliminates fatigue.
- Lower unit shelves for storage.

- Desk shelf swings up to close off unit when not in use.
- Compact design: 34"w x 36"h x 24"d.
- Indestructible natural oak or walnut woodgrain finish.
- Ready to assemble with only a screwdriver.

ONLY $149.00

To order call toll free
1-800-426-5301
In Washington call
(206) 423-7277
VISA & MasterCard accepted.

THE FURNITURE BYTE
P.O. Box 1757 9 Judith Place Longview, WA 98632

SAVE BIG ON COMPUTER PRODUCTS

MODEMS

HAYES

Smartmodem 300	209
Smartmodem 1200	489
Smartmodem 1200B	425
Micromodem II E	240
Smartcom II Software (for IBM PC)	90
NOVATION	
J. Cat 300B direct	99
103 SmartCat 300B Smart	159
103/212 Smart 300/1200 B	379
AppleCat II 300 baud for Apple	200
212 AppleCat 300/1200B for Apple	390
Access 1-2-3 1200B for IBM	379

MONITORS

GORILLA

12" Green	88.00
12" Amber	94.00
TAXAN	
12" KG-12N (green)	120
12" KG-12N-UY (amber)	125
RGB vision III	499
AMDEK	
V300G 12" green	130
V300A 12" amber	145
V310A 12" amber (for IBM)	160
Color I+ 13" composite	280
Color II 13" RGB	425
Color IV 13" RGB analog	740

QUADRAM

MICROFAZER

QRMP-8 Par/Par	135
QRMSP-8 Ser/Par	145
QRMMSS-8 Ser/Ser	145
QRMPS-8 Par/Ser	145
QUADBOARD (for IBM PC)	
QR5064 64K	270
QR4064 64K	270
QUADLINK	
QR 3000 for IBM	475
QR 3010 for Compaq	475
QR 3020 for Columbia	475
QR 8201 Quadcolor-1	200
QR 8202 Quadcolor-2 (upgrade kit)	200
e Ram 80 — 80 col. card for Apple IIe	115

wabash maxell Dysan

DISKETTES

10 — 5¼" Floppy Diskettes (packed with a storage box)

	SS /SD	SS / DD	DS / DD
Wabash	$19	$22.50	$30
Maxell	NA	27.00	39
Dysan	NA	27.00	39
10 — 8" floppy diskettes			
Maxell	NA	36.00	42

PRINTERS

Tremendous Discounts
GEMINI
DELTA
EPSON
COMREX
DIABLO • TTX
MANNESMANN TALLY
Most Models IN STOCK
CALL FOR PRICES

HEWLETT-PACKARD CALCULATORS

HP-11C	70	HP-15C	90
HP-12C	90	HP-16C	90
HP-41C	145	HP-41CX	245
HP-41CV	200	HP-97	560

all software & accessories too

PORTABLE COMPUTERS

HP-71B	399	HP-75	719

PERSONAL COMPUTATION DEALER

CABLES — INTERFACES

accessories for Computer Printers

GRAPPLER PLUS	130
16K BUFFERED GRAPPLER	165
Apple Dumpling	70
Cardco G	70
10 ft. Par. Cable for IBM	32
10 ft. 36x36 Parallel	32
10 ft. 25x25 RS-232	25
Y Cable (for TI-99/4A)	25
6 ft. TI-99/4A parallel cable	25
13x16 printer stand for 80 col. printers)	15
16x22 printer stand (heavy duty) for wide carriage printers	40

RIBBONS, PAPER

Paper — 1000 Sheet Pack

9½ x 11 white, 20 lb.	12
9½ x 11 green bar, 20 lb.	12
9½ x 11 trim edge, 20 lb.	15
14⅞ x 11 green bar, 20 lb.	19
Ribbon Cartridges	
for Epson 80 col.	4
for Epson 132 col.	7
for Mann. Tally 160	11
for Mann. Tally 180	14
for M.T. Spirit 80	7.50
for Panasonic 1090	6.50
for Diablo Hytype II	5
for C. Itoh Starwriter	5

Twin spool ribbons for Gemini 10/10X, 15/15X; Okidata 80, 82, 83
6 for 15.00 or 12 for 24.00

CALL TOLL FREE 800-621-1269 EXCEPT Illinois, Alaska, Hawaii

Corp. Accts. Invited. Min. Ord. $15.00. Mastercard or Visa by mail or phone. Mail Cashier's Check, Mon. Ord., Pers. Check (2 wks. to clr.) Add $4.00 1st item. **(AK, HI, P.R., Canada add $10.00 first item)** $1.00 ea. add'l shpg. & handl. Shipments to IL address add 7% tax. Prices subj. to change. **WRITE for free catalog. RETURN POLICY: Defectives Only: Most products replaced within 30 days of purchase with identical merchandise only. Computer and large peripherals replaced only when defective on arrival (within 3 work days of purchase). Other problems covered by mfr. warranty. ALL ELEK-TEK MERCHANDISE IS BRAND NEW, FIRST QUALITY AND COMPLETE.**

ELEK-TEK, inc. 6557 N. Lincoln Ave., Chicago, IL 60645
(312) 631-7800 (312) 677-7660

Jim Butterfield, Associate Editor

A Program Critique

Part 1

Over the next few columns, I'll be going through a program by Bud Rasmussen. This program performs a single disk file copy on the Commodore 64. It works well; I'll abridge the program slightly to save space.

The program is exceedingly well organized; it's a pleasure to read. It has good operator interface (there are lots of messages) and performs many useful functions in machine language that readers may wish to study and use.

The program was written using an assembler, in this case the PAL assembler by Brad Templeton. Other assemblers would do the job equally well, but they might require slight changes in syntax.

My role here will be (with Rasmussen's permission) to critique and comment. Since the program works, the criticism is one of style rather than of substance.

Different But Still Right

I should note that Rasmussen wrote this program, not for commercial purposes or publication, but for his own satisfaction and use. I'll be somewhat unfairly criticizing his program based on its appropriateness for general usage. For example, I may comment adversely on such things as his use of a BRK instruction to terminate the program on certain error conditions, because this would be undesirable in a public program. But in the final analysis, it's Rasmussen's personal program and it works the way he planned it.

Accept these comments as ideas on how to organize your own work. You'll find a lot of good machine language programming techniques in the program.

```
;
; ML FILE COPIER
;
```

```
;
;
; IMMEDIATE VALUES
;
;
RK      = 13     ; RETURN KEY
H10     = 16     ; HEX TEN
DK      = 20     ; DEL KEY
C       = 44     ;',' (COMMA)
EOFI    = 64     ; END OF FILE INDICATOR
W       = 87     ;'W' (WRITE)
CH      = 147    ; CLEAR/HOME
;
```

Many programmers like to *equate* constant values to labels. This way, if you want to print RETURN, you can write LDA #RK instead of LDA #$0D followed by JSR $FFD2 to print.

My preference is to skip the symbol and use the $0D value, as long as it is the same on all machines. It seems to me that the symbolic values are useful only when different machines use different values for the same function. In such a case, you would indeed equate the appropriate value to a label and save yourself work.

The next section (not given here) defines zero page, working storage, and Kernal addresses. You will be able to pick these out directly from the program, as needed.

```
C000    *=  $C000
```

A Program Counter

Many assemblers use the asterisk character (*) as a *program counter*. In this case, the program will start at hexadecimal C000. The asterisk is often read as "here"; so a programmer may verbalize this line as "Here is hex C000."

```
;
;
; SCREEN ROUTINE
;
```

UNDERWARE

Software that's priced UNDER the competitions'

COMMODORE 64 & ATARI COMPUTER CLASSICS

ENTERTAINMENT
Just for You!

EPYX
- Temple of Apshai (D&C) $26
- Upper Reaches Apshai (D&C) $16
- Gateway to Apshai (CT) $26
- DragonRiders of Pern (CT) $26
- Jumpman (CT) $26
- Jumpman Junior (CT) $26
- PitStop (CT) $26
- Broderbund Choplifter (CT) $29
- Broderbund Lode Runner (D) $26
- First Star Flip Flop (D&C) $22
- First Star Bristles (D&C) $22
- HesWare Gridrunner (CT) $19
- Infocom Zork I, II, III (D) ea. $26
- Infocom Witness (D) $33
- Infocom PlanetFall (D) $33

PARKER
- Parker Q Bert (CT) $37
- Parker Star Wars (CT) $37
- Parker James Bond (CT) $37
- Parker Gyruss (CT) $37
- Parker Popeye (CT) $37
- Parker Frogger (CT) $37

SYNAPSE
- Synapse Blue Max (D&C) $22
- Synapse Fort Apocalypse (D&C) $22
- Synapse Necromancer (D&C) $22
- Synapse Pharh Curse (D&C) $22
- Synapse Zeppelin (D&C) $22
- Sega Star Trek (CT) $25
- Sega Congo Bongo (CT) $25
- Sega Buck Rogers (CT) $25
- Screenplay DunZhin (D) $29
- SubLogic Pinball (D&C) $22
- SubLogic Flight Simulator $37

EDUCATION

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

HAYDEN (ages 4-10)
- MicroAddition (D&C) $20
- MicroSubtraction (D&C) $20
- MicroDivision (D&C) $20
- MicroMultiplication (D&C) $20
- Monkey See-Spell (D&C) $20

AMERICAN EDUCATIONAL COMPUTER (DISK ONLY)
- Learn About Sounds (grades K-3) $39
- Words In Reading I (grades 1-3) $39
- Words in Reading II (grades 1-3) $39
- Reading Compreh I (grades (1-4) $37
- Reading Compreh II (grades 3-7) $37
- Reading Compreh III (grades 6-8) $37
- Vocabul Word Blder (grades 2-8) $29
- Grammer Word Skill (grades 2-8) $29
- US Geography Facts (grades 2-8) $29
- World Geography (grades 2-8) $29
- Spanish Vocabulary (grades 2-8) $29

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

PERSONAL FINANCE
- Continental Home Accountant (D) $47
- Continental Tax Advantage (D) $33
- Continental FCM First Class Mail $29

ELECTRONIC ARTS
- Pinball Construction Set (D) $39
- M.U.L.E. (D) $39
- Worms? (D&C) $34
- Archon (D&C) $39
- Hard Hat Mack (D&C) $34
- Murder on the Zinderneuf (D) $39
- The Cut & Paste Word Processor (D) $49
- D Bug (D) $34
- Axis Assassin (D) $34
- Music Construction Set (D) $39

FREE DISKETTE with each purchase of Electronic Arts software

COM 64 EXCLUSIVES!

PERSONAL FINANCE
- Timeworks SwifTax (D) $39
- Creative Household Finance (D) $23
- Creative Household Finance (C) $19
- HesWare Finance Manager (D) $49
- Cardco Tax Survival Program (D) $33

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

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

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

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

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

GAMES
Just for You!
- Acess Beach Head (D&C) $23
- Avalon Hill TelenGuard (C) $15
- Avalon Hill B-1 Bomber (C) $12
- Creative Crisis Mountain (CT) $25
- Synapse Sentinel (D&C) $22
- Synapse ZAXXON (D&C) $28
- ScreenPlay Pogo Joe (D&C) $19
- ScreenPlay Asylum (D&C) $23
- ScreenPlay K. Uston BlackJack (D) $57
- Quick Silva Ring Power (D&C) $23
- Quick Silva AquaPlane (D) $23
- First Star Boulder Dash (D&C) $25

VIC 20 HITS
- Creative Home Office (D) $22
- Creative Home Office (C) $19
- MSI Practicalc Plus (D) $35
- MSI Practicalc Plus (T) $33
- MSI Practicalc (D) $32
- MSI Practicalc (T) $29
- HesWare Vic Forth (CT) $39
- HesWare Hesmon (CT) $26
- Cardco Write Now $27
- Epyx Fun with Art (CT) $26
- Epyx Fun with Music (CT) $26
- KinderComp (CT) $23
- Story Machine (CT) $23
- Face Maker (CT) $23
- Kids On Keys (CT) $23
- Alphabet Zoo (CT) $23
- HesWare Turtle Graphics (CT) $26
- WizWare Turtle Tracks (C) $19
- Creative Choplifter (CT) $19
- Epyx Tempe of Apshai (C) $24
- Sega Congo Bongo (CT) $25
- Sega Star Trek (CT) $25

PERIPHERALS

GRAPHICS TABLET
- CHALKBOARD POWERPAD (C64 & Atari) $79

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

MONITORS
- BMC 13" Composite Color Plus $229
- BMC 12" Low Res Green $85
- USI 12" Green Hi-Hi Res $129
- USI 12" Amber Hi-Hi Res $139
- TAXAN 12" Green $99
- TAXAN 12" Amber $109
- NEC JB 1205 Amber $155
- Amdek Color I $287

PRINTERS
- NEC 8023A $379
- STAR GEMINI 10X $287
- ALPHA COM 81 $169
- Cable with AlphaCom FREE
- Okidata 82A $389
- Okidata 83A $589
- Transtar 1315 Color $489

WICO JOYSTICKS
- The Boss $14
- Bat Stick $19
- Red Ball $21
- Track Ball $30
- 3 Way Grip $21

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

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

DISK DRIVES

ATARI
- Rana 1000 176K $329
- Concorde Master 176K $329
- Concorde Slave 176K $269
- Concorde Master 348K $389
- Concorde Slave 348K $309

COMMODORE
- Concorde Master 174K $299
- Concorde Master 348K $369
- Commodore 1541 174K $269

BLANK DISKETTES

MAXELL DISKS
- MD1 SS DD Box of Ten $27
- MD2 DS DD Box of Ten $40

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

$uch A Deal
12629 N. Tatum Blvd.
Suite 138
Phoenix, AZ 85032

CALL TOLL FREE 1-800-431-8697/602-957-3619
For Customer Service Call: 602-955-3857

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

```
            ;
            ; CLEAR SCREEN
            ;
            ;
            CS   =   *
            ;
C000  A9 93         LDA  #CH       ; LOAD CLR/HOME
C002  20  D2 FF     JSR  CHROUT    ; PUT IT
            ;
            ;
            ; CHG COLORS
            ;
            ;
            CC   =   *
            ;
C005  A9 07         LDA  #7        ; SET
C007  8D 20  D0     STA  BCA       ; BORDER COLOR
C00A  A9 05         LDA  #5        ; SET
C00C  8D 21  D0     STA  BGCA      ; BACK GROUND COLOR
C00F  A9 01         LDA  #1        ; SET
C011  8D 86  02     STA  CCA       ; CHARACTER COLOR
            ;
```

As you can see, the documentation is extensive. The program is placed at address $C000, the spare RAM block in high memory.

"CS=*" means "label CS is this point," or "CS is here"; it's a way of defining symbolic locations so that they stand out. Rasmussen uses this type of label definition extensively. Placing the label on the next program line will work just as well; to save space, I'll do this in most cases.

Character color could also be set by calling the CHROUT routine, $FFD2, with the appropriate ASCII color character in the A register.

```
            ;
            ; POSITION CURSOR
            ;
            ;
C014  A2 03      PC    LDX  #3       ; ROW=3
C016  A0 00            LDY  #0       ; COLUMN=0
C018  18               CLC           ; AND
C019  20  F0  FF       JSR  PLOT     ; SET CURSOR
            ;
```

I'd just as soon print three cursor-down or return characters.

```
            ;
            ; CLEAR FILE NAME AREA
            ;
            ;
C01C  A9 00      CFNA  LDA  #0       ; SET 'A'=0
C01E  AA               TAX           ; SET 'X'=0
            ;
C01F  9D 40  03  CFNL  STA  FNA,X    ; CLEAR FILE NAME
C022  E8               INX           ; INCR INDEX
C023  E0  15           CPX  #21      ; IS X=21 (16+4+1)
C025  F0  02           BEQ  PM       ; IF SO, EXIT
C027  D0  F6           BNE  CFNL     ; ELSE, LOOP
            ;
```

This is probably overkill, since the area will be filled with appropriate characters before it is used. I would opt for insertion of a prefix "0:" which here would appear ahead of the file name. Sometimes the disk seems to work better if drive 0 is explicitly identified.

BNE Alone Also Works

I would have put the BNE ahead of the BEQ. Then I would have dropped the BEQ since the program would proceed to the next statement, PM, anyway.

```
                  ;
                  ; PUT MESSAGE
                  ;
                  ;
C029  A2  F4      PM    LDX  #ML     ; LOAD LENGTH
C02B  A0  C0            LDY  #>IM    ; LOAD HI BYTE
C02D  A9  35            LDA  #<IM    ; LOAD LO BYTE
C02F  20  75  C1        JSR  PR      ; PRINT MSG
                  ;
C032  4C  29  C1        JMP  GI      ; GOTO GET INPUT
                  ;
                  ;
                  ; INFORMATION MESSAGE
                  ;
                  ;
C035  12          IM    .BYTE$12
C036  20  20  4B        .ASC" KEY IN THE FILE NAME OF "
C050  0D  0D  12        .BYTE$0D,$0D,$12
C053  20  20  54        .ASC" THE FILE TO BE COPIED, "
C06C  0D  0D  12        .BYTE$0D,$0D,$12
C06F  20  20  41        .ASC" AS N.....,T-"
C081  0D  0D  12        .BYTE$0D,$0D,$12
C084  20  20  4E        .ASC" N=NAME, T=TYPE (P OR S) "
C09E  0D  0D  12        .BYTE$0D,$0D,$12
C0A1  20  20  4D        .ASC" MAXIMUM NAME=16 BYTES "
C0BB  0D  0D  12        .BYTE$0D,$0D,$12
C0BE  20  20  4B        .ASC" KEY <RET> WHEN FINISHED."
C0D8  0D  0D  12        .BYTE$0D,$0D,$12
C0DB  20  20  49        .ASC" IF YOU MAKE A MISTAKE;"
C0F5  0D  0D  12        .BYTE$0D,$0D,$12
C0F8  20  20  55        .ASC" USE THE DELETE KEY. "
C10E  0D  0D  12        .BYTE$0D,$0D,$12
C111  20  20  20        .ASC" C H E E R S!!!"
C126  0D  0D  0D        .BYTE$0D,$0D,$0D
                  ML    =   *-IM
                  ;
                  ;
                  ; GET INPUT
                  ;
                  ;
C129              GI    =   *
                  ;
```

Note how the message length is calculated automatically by the assembler (ML=* – IM). The end of message plus one ("here") minus the start of message gives the length.

Subroutine PR is shown later. As can be seen from the program segment above, the high and low parts of the message address are loaded into registers Y and A respectively, the length into register X; then PR is called. We'll look at that subroutine when it comes up.

An Unusual Place For Messages

The message text is thrown in-line directly behind the program segment that uses it. This is unusual: It's more common for all text, tables, and variables to be placed at the end of the program. During the program development phase things might be out of order, but it's usual to clean that up later. No big deal: It costs us a JMP instruction leap over the message to get to address GI. In the meantime, it's convenient for us, the readers, since as we read the code which prints the message, the message is right there for us to see.

Rasmussen shows exceptional modesty. Even though the user sees a lengthy opening message, the author's identity is not included.

Next comes a "friendly" input routine:

```
                     ;
                     ;
                     ; GET INPUT
                     ;
                     ;
C129  A2 00     GI   LDX  #0        ; SET INDEX=0
C12B  8E A8 02  SI   STX  SIV       ; STORE INDEX VALUE
                     ;
                     ;
                     ; GET NEXT CHARACTER
                     ;
                     ;
C12E  20 E4 FF  GNC  JSR  GETIN     ; GET A CHARACTER
C131  F0 FB          BEQ  GNC       ; IF NONE, TRY AGAIN
C133  AE A8 02       LDX  SIV       ; LOAD INDEX VALUE
C136  C9 0D          CMP  #RK       ; IS THIS RETURN
C138  F0 1C          BEQ  FNE       ; KEY
C13A  C9 14          CMP  #DK       ; IS THIS DEL.
C13C  F0 03          BEQ  PDR       ; KEY
C13E  4C 4C C1       JMP  AI        ; GOTO ACCEPT INPUT
                     ;
```

We look for a character, and go back to try again if no character is there. If the character is RETURN (RK), we're finished and go to FNE. If it's DELETE (DK), we go the special delete routine. Otherwise, we go to "accept input." The Accept Input (AI) routine could have been inserted at this point to save the JMP instruction.

```
                     ;
                     ; PROCESS DEL REQUEST
                     ;
                     ;
C141  E0 00     PDR  CPX  #0        ; INDEX VS ZERO
C143  F0 E6          BEQ  SI        ; IF SO, BYPASS
C145  CA             DEX            ; DEL.
C146  20 D2 FF       JSR  CHROUT    ; CHARACTER
C149  4C 2B C1       JMP  SI        ; GOTO STORE INDEX
                     ;
```

X counts the input characters. If we see a DELETE character, we must decrease X provided it's greater than zero. We should print the delete using CHROUT ($FFD2) in order to erase the previous character on the screen.

```
                     ;
                     ; ACCEPT INPUT
                     ;
                     ;
C14C  9D 40 03  AI   STA  FNA,X     ; STORE FILE NAME BYTE
C14F  20 D2 FF       JSR  CHROUT    ; PUT IT
C152  E8             INX            ; INCR POINTER
C153  4C 2B C1       JMP  SI        ; GOTO STORE INDEX
                     ;
```

An Alternative To JMP

An "ordinary" character is stored and printed. The X counter is increased and we return to get more input. It would be safe to use BNE instead of JMP here, since X will always be nonzero.

```
                     ;
                     ; FILE NAME END
                     ;
                     ;
C156  20 D2 FF  FNE  JSR  CHROUT    ; DOUBLE
C159  20 D2 FF       JSR  CHROUT    ; SPACE
                     ;
                     ;
                     ;
                     ; ADD THE REST OF THE
                     ; FILE NAME FOR WRITE ( ,W )
                     ;
                     ;
C15C  A9 2C          LDA  #C        ; LOAD AND
C15E  9D 40 03       STA  FNA,X     ; STORE COMMA
C161  E8             INX            ; INCR POINTER
C162  A9 57          LDA  #W        ; LOAD AND
C164  9D 40 03       STA  FNA,X     ; STORE 'W'
C167  E8             INX            ; INCR POINTER
C168  8E AB 02       STX  OFNL      ; STORE OUTPUT FILE NL
C16B  38             SEC            ; SUBTRACT 4
C16C  8A             TXA            ; FOR
C16D  E9 04          SBC  #4        ; INPUT FILE
C16F  8D AA 02       STA  IFNL      ; NAME LENGTH
C172  4C 8A C1       JMP  DIOR      ; GOTO DISK I/O ROUTN
                     ;
```

The program trusts the user to correctly type in a name such as DFILE,S or LPROG,P. It's dangerous to depend upon a user to input exactly the right thing: At the very least, I'd have the program check that the last two characters typed in were a comma followed by P or S.

Indirect Addressing

Now we reach the print subroutine previously used by the program.

```
                     ;
                     ; PRINT ROUTINE
                     ;
                     ;
C175  8E A9 02  PR   STX  SLV       ; STORE LENGTH
C178  85 22          STA  LB        ; STORE LO BYTE
C17A  84 23          STY  HB        ; STORE HI BYTE
C17C  A0 00          LDY  #0        ; SET INDEX=0
                     ;
C17E  B1 22     PRL  LDA  (LB),Y    ; GET CHARACTER
C180  20 D2 FF       JSR  CHROUT    ; PUT IT
C183  C8             INY            ; INCR INDEX
C184  CE A9 02       DEC  SLV       ; DECR LENGTH
C187  D0 F5          BNE  PRL       ; IF NOT 0, CARRY ON
C189  60             RTS            ; RETURN
                     ;
```

A quite straightforward use of indirect addressing for a print subroutine. I might have used CPY SLV instead of DEC SLV, but it works out the same.

```
C18A  A9 00     DIOR  =*
```

In the next session, the program opens the command channel and an input data file, and reads the selected file into memory. We'll continue with critical comments on this program. ©

64 EXPLORER

Larry Isaacs

In this column I will show how to create line drawings on a bitmapped display. In addition, we will look at an example written in BASIC and begin converting it to machine language routines which you can use in your programs. For the line-drawing routine to be really effective, conversion to machine language is required. The BASIC routine is much too slow.

There are a number of different methods for drawing lines in machine language. One way appeared in the August 1981 issue of *BYTE* magazine. In the article (page 414), Mike Higgins presented an algorithm which requires only integer adds and subtracts to generate the point along a line specified by two end points. The algorithm was an improvement on one presented in the book *Principles of Interactive Computer Graphics* by Newman and Sproull, which is widely known in the field of computer graphics.

Basically, the method involves setting up a loop which will compute each point along the line. Since we would expect to generate one point each time through the loop, the number of points in the line gives us the number of times the loop should be executed. Remember that the display coordinates are always integer values. The horizontal or vertical step between adjacent dots is always 1. To calculate the number of points in the line, simply take the absolute value of the larger of the differences between the X coordinates and the Y coordinates.

The Primary Axis

To identify the axis with the larger difference in a general way, we will call it the primary axis. For example, if we were drawing a line from 0,0 to 100,50, the X axis has the larger difference and would be the primary axis. The number of points in the line, after plotting the first one, would be 100.

Since there are two possibilities for the primary axis, we will need two loops. At this point, we could write some of the statements that will be required in our BASIC line-drawing routine:

```
<plot first point>
DX=X1-X0:DY=Y1-Y0
IF ABS(DX) < ABS(DY) THEN <1>
FOR I=1 TO ABS(DX)
...
NEXT
...
<1> FOR I=1 TO ABS(DY)
...
NEXT
```

where X0,Y0 is the starting point, X1,Y1 is the ending point, and ‹1› is a symbolic replacement for an appropriate line number. The first point must be plotted first since the loops generate successive points based on the preceding one.

Setting The Increment

Since the number of points is equal to the coordinate difference on the primary axis, that coordinate should be incremented by one each time through the loop. The coordinate for the other axis, which has a smaller difference, is incremented at a rate smaller than one. If the differences along each axis are the same, either axis can be used as the primary axis, and both coordinates are incremented by one each time through the loop.

The next step in implementing our drawing routine is to come up with a method of properly incrementing the nonprimary axis. We'll want the loop to generate integer values for both Y and X. The loop could be written:

```
X=X0:Y=Y0:SX=1:R=.5*SX
FOR I=1 TO ABS(DX)
X=X+1
R=R+DY/DX*SX
IF R>=SX THEN Y=Y+1:R=R-SX
PLOT X,Y
NEXT
```

Easier Conversion

In this version of the loop, I have the variable SX to show where the X increment of 1 is involved in the incrementing of Y. The variable R is initialized to .5 so that rounding will occur as it is incremented by DY/DX. It is also important to note that Y will be properly incremented even if SX isn't the normal value of 1. The very handy improvement Mike Higgins suggested was, in effect, to replace SX with DX. When you do this, note what happens to the loop.

```
X=X0:Y=Y0:R=DX/2
FOR I=1 TO ABS(DX)
X=X+1
R=R+DY
IF R>=DX THEN Y=Y+1:R=R-DX
PLOT X,Y
NEXT
```

Only addition and subtraction are required within the loop, making it much more suitable for conversion to machine language. The only division is a division by 2, which is also easily done in machine language. Naturally, a similar loop could be written to handle the case when Y is the primary axis.

Now that we have a routine for calculating the points along a line, we need a routine to plot these points. There are a number of books which discuss this in detail: the *Commodore 64 Programmer's Reference Guide*, and two recent releases from COMPUTE!, *COMPUTE!'s Reference Guide to Commodore 64 Graphics* and *COMPUTE!'s First Book of Commodore 64 Sound and Graphics*.

The Plotting Routine

Before getting into a short discussion of the point-plotting routine, we will have to decide whether we will be plotting points in the hi-res bitmap mode or the multicolor bitmap mode, or both. To help simplify the discussions, I will be looking at just the hi-res bitmap mode. In next month's column, I will tie up loose ends from this article and include discussion of the multicolor mode.

Briefly, the point-plotting routine must calculate the byte address and the bit within the byte for the point to be plotted. The address of the byte will be dependent on the location of the bit-mapped graphics RAM and the X,Y coordinates of the point. Once the byte and bit have been identified, there are three things we can do with the point.

First, we can set the bit to a 1, causing the point on the screen to take on the foreground color. Second, we can set the bit to a 0, causing the point on the screen to take on the background color. This is equivalent to erasing the point. Third, we can change the bit to the opposite of its current state. This is called "flipping" the point. Applying these to line drawing, we can get three drawing modes, DRAW, ERASE, and FLIP.

Erasing With FLIP

For those who have not encountered FLIP mode before, it has a couple of unique properties. First, lines drawn in FLIP mode are always visible. If a line is drawn through an area that is already set to the foreground color, you can still see the path of the line because that portion of the line will be set to the background color. Second, if a line drawn in FLIP mode is redrawn in FLIP mode, the line will be erased. The screen will be exactly the same as if the line had never been drawn.

The disadvantage of FLIP mode is that gaps appear where lines intersect. FLIP mode is probably most useful when the object being drawn will undergo editing. Once the editing is complete, the screen can be redrawn in the normal DRAW mode, filling the gaps in the lines.

Also described in the books mentioned above are the details on how to initialize the bitmap RAM and the associated screen memory. Program 1 illustrates what we have discussed so far. There is one slight addition to the drawing loop which was shown last. For ERASE and FLIP mode drawing to be most effective, it is desirable that a given line be drawn the same way regardless of which end is its starting point.

To this end, the line-drawing subroutine at line 1000 always draws from left to right. The X increment will always be +1 with the Y increment being +1 or –1, depending on the sign of DY. Also, the point-plotting routine at line 800 does not handle changing the foreground or background colors when plotting a point. This could be easily added if desired. Unfortunately, changing the colors for one point changes the colors for all points within that bitmap cell.

Selecting The Commands

In the example program, the drawing mode is specified by the variable M. If M=0, ERASE mode is selected. M=1 selects DRAW mode, and M=2 selects FLIP mode. The main routine at line 2000 is a simple loop which performs commands specified by DATA statements. Each command consists of a command number followed by one or two argument numbers. A command number of 1 performs a move to the X,Y coordinates specified by the two arguments which follow. Command number 2 performs a draw to the X,Y coordinates specified by the two arguments which follow.

Command number 3 sets the drawing mode to the value of the number that follows. Command number 0 terminates execution of the commands. The graphics display will remain on the screen until a key is pressed. Running this example program should make it clear that BASIC isn't suited for the task of drawing lines. For iterative tasks which require many calculations, the fact that BASIC is interpreted will severely slow up the task.

Conversion To Machine Language

Now we are ready to begin the conversion to machine language. First we will need two routines to replace the BASIC subroutines at lines 100 and 900, which save and restore the text screen, respectively. These will be called SVSCRN and RSSCRN. Next come the BASIC subroutines at lines 200, 300, 400, and 700 which set up the bitmap display. We will combine these into a single routine called GRSCRN to enable the graphics screen.

Since we are working with machine language, we can place the bitmapped memory under the OS ROM and screen memory just below where the

DOS Wedge goes, at $E000 and $C800, respectively. This way no user memory is lost from BASIC. Also, the text screen will not be disturbed by drawing in the graphics screen. A slight disadvantage of this arrangement is that interrupts must be turned off while the OS ROM is disabled. This will cause the 64 clock timer to be slightly off.

The final routine we will look at this month will be the one which clears the bitmap and screen memories. This will replace the BASIC subroutines at lines 500 and 600 and will be called CLRSCR. Actually, this routine will rely on another routine called FILL to do the clearing. Rather than fetch arguments from BASIC, this routine fills the screen memory with a constant. This constant sets the background to white and the foreground to red. Next month we'll upgrade this routine to accept arguments specifying the colors.

The result of this set of machine language routines is shown in Program 2. As written, these routines would assemble at location $C000. Like the location of the graphics RAM, this doesn't reduce BASIC user RAM. It also won't interfere with the DOS Wedge. This may conflict with some other machine language routines you find handy, but finding locations which don't conflict has become impossible.

A Jump Table

There are a couple of things worth noting in the machine language listing. First, a jump table is placed at the beginning. This table will contain JMP instructions to each of the routines that will be called from BASIC. Even though the starting locations of the various routines may change, the jump table and the locations to SYS to will stay put. To call one of the routines from BASIC, simply set a variable equal to $C000 (49152), then SYS to this variable plus the appropriate offset. For example: TB=49152:SYS TB+9 would call the CLRSCR routine.

Second, you may want to take a look at the FILL routine. Its function is to fill an area of memory with a specified byte. This function could have been implemented a number of different ways. For example, it could have been written a little more simply by using a single loop. Instead I implemented it using two loops, one to fill whole pages (groups of 256 bytes) and the other to fill the final partial page. By doing this, the core loops can be written with just STA, INY, and BNE instructions, making them very fast. It would be difficult to make the routine much faster without increasing its size quite a bit and making it more complex.

When implementing almost any routine, you will usually be faced with trade-offs between size, simplicity, and speed. Here I tried to maximize speed without sacrificing size and simplicity too much.

Next month we'll continue with the conversions of the point-plotting and line-drawing routines, and look into the multicolor bitmap mode. I want to provide information you can put to some use. Please write me with any comments or suggestions concerning topics you would like to see discussed.

Program 1: BASIC Drawing Program

```
10 REM DRAW PROGRAM                    :rem 139
20 V1=56576:V2=53272:V3=53265          :rem 58
30 M(7)=1:M(6)=2:M(5)=4:M(4)=8         :rem 34
40 M(3)=16:M(2)=32:M(1)=64:M(0)=128
                                       :rem 21
50 GOTO 2000                           :rem 96
100 REM SAVE DISPLAY BYTES             :rem 65
110 S1=PEEK(V1)                        :rem 80
120 S2=PEEK(V2)                        :rem 83
130 S3=PEEK(V3)                        :rem 86
140 RETURN                             :rem 117
200 REM SET GRAPHICS RAM BANK          :rem 175
210 REM BASED ON BITMAP ADDRESS, BA
                                       :rem 229
220 T=INT(BA/16384):T=3-T              :rem 152
230 POKE V1,(PEEK(V1)AND252)OR T       :rem 38
240 RETURN                             :rem 118
300 REM SET BITMAP 8K OFFSET           :rem 106
310 REM BASED ON BITMAP ADDRESS, BA
                                       :rem 230
320 T=BA:IF T>32767 THEN T=T-65536:rem 87
330 T= -((T AND 8192) <> 0)            :rem 155
```



```
340 POKE V2,(PEEK(V2)AND247)OR (T*8)
                                       :rem 225
350 RETURN                             :rem 120
400 REM SET COLOR MEMORY OFFSET        :rem 131
410 REM BASE ON COLOR ADDRESS, CA:rem 102
420 T=INT(CA/16384):T=CA-T*16384   :rem 28
430 T=INT(T/1024)                      :rem 174
440 POKE V2,(PEEK(V2)AND15)OR (T*16)
                                       :rem 218
450 RETURN                             :rem 121
500 REM CLEAR BITMAP MEMORY            :rem 118
510 FOR A=BA TO BA+7999                :rem 177
520 POKE A,0:NEXT                      :rem 220
530 RETURN                             :rem 120
600 REM CLEAR SCREEN MEMORY            :rem 122
610 REM FC=ON COLOR, BC=OFF COLOR:rem 165
620 T=FC*16+BC                         :rem 243
630 FOR A=CA TO CA+999                 :rem 127
640 POKE A,T:NEXT                      :rem 3
650 RETURN                             :rem 123
700 REM ENABLE BITMAP GRAPHICS         :rem 48
710 POKE V3,PEEK(V3) OR 32             :rem 129
720 RETURN                             :rem 121
800 REM PLOT POINT X,Y WITH MODE M
                                       :rem 208
810 TY=199-Y:RO=INT(TY/8):CO=INT(X/8)
                                       :rem 24
820 YB=TY AND 7:XB=X AND 7             :rem 156
830 A=BA+RO*320+CO*8+YB                :rem 12
840 B=PEEK(A)                          :rem 210
850 IF M=1 THEN POKE A,B OR M(XB):RETURN
                                       :rem 231
860 IF M=0 THEN POKE A,B AND (255-M(XB)):
    RETURN                             :rem 51
870 IF B AND M(XB) THEN POKE A,B-M(XB):RE
    TURN                               :rem 7
```

```
880 POKE A,B OR M(XB):RETURN      :rem 113
900 REM RESTORE DISPLAY           :rem 183
910 POKE V1,S1                    :rem 0
920 POKE V2,S2                    :rem 3
930 POKE V3,S3                    :rem 6
940 RETURN                        :rem 125
1000 REM THE LINE DRAWING ROUTINE:rem 224
1010 REM DRAWS FROM X0,Y0 TO X1,Y1
                                  :rem 122
1020 DX=ABS(X1-X0):DY=ABS(Y1-Y0) :rem 124
1030 IF DX=0 AND DY=0 THEN RETURN :rem 72
1040 X=X0:Y=Y0:IY=SGN(Y1-Y0)     :rem 205
1050 IF DX < DY THEN 1300        :rem 189
1200 IF X1<X0 THEN X=X1:Y=Y1:IY=-IY
                                  :rem 46
1210 GOSUB 800:R=DX/2            :rem 162
1220 FOR I=1 TO DX               :rem 162
1230 X=X+1:R=R+DY                :rem 242
1240 IF R>=DX THEN Y=Y+IY:R=R-DX :rem 142
1250 GOSUB 800:NEXT               :rem 89
1260 X0=X1:Y0=Y1:RETURN          :rem 187
1300 IF Y1<Y0 THEN X=X1:Y=Y1:IY=-IY
                                  :rem 49
1310 GOSUB 800:R=DY/2            :rem 164
1320 FOR I=1 TO DY               :rem 164
1330 Y=Y+IY:R=R+DX               :rem 101
1340 IF R>=DY THEN X=X+1:R=R-DY   :rem 30
1350 GOSUB 800:NEXT               :rem 90
1360 X0=X1:Y0=Y1:RETURN          :rem 188
2000 REM THE MAIN ROUTINE        :rem 210
2010 BA=16384:CA=BA+8192:FC=2:BC=1:M=1
                                  :rem 90
2020 GOSUB 100:GOSUB 200:GOSUB 300
                                  :rem 110
2030 GOSUB 400:GOSUB 700          :rem 42
2040 GOSUB 600:REM CLEAR COLOR MEM
                                  :rem 191
2050 GOSUB 500:REM CLEAR BITMAP   :rem 30
2060 REM LOOP DRAWING DATA        :rem 12
2070 READ S:ON S GOTO 2200,2300,2400
                                  :rem 66
2100 REM END PROGRAM             :rem 150
2110 GET Z$:REM CLEAR KEYBOARD QUE
                                  :rem 227
2120 IF Z$<>"" THEN 2110         :rem 131
2130 GET Z$:IF Z$="" THEN 2130   :rem 225
2140 GOSUB 900:END               :rem 241
2200 REM MOVE                    :rem 223
2210 READ X0,Y0:GOTO 2060         :rem 89
2220 X=X+IX:R=R+DY                :rem 98
2230 IF R>=DX THEN Y=Y+IY:R=R-DX :rem 142
2240 GOSUB 800                   :rem 224
2250 NEXT                          :rem 8
2260 X0=X1:Y0=Y1:RETURN          :rem 188
2300 REM DRAW                    :rem 215
2310 READ X1,Y1:GOSUB 1000:GOTO 2060
                                 :rem 215
2400 REM SET MODE                :rem 187
2410 READ M:GOTO 2060            :rem 107
3000 DATA 3,2                    :rem 110
3010 DATA 1,50,50                 :rem 49
3020 DATA 2,100,100              :rem 139
3030 DATA 1,50,100                :rem 95
3040 DATA 2,100,50                :rem 97
3050 DATA 2,100,100              :rem 142
3060 DATA 2,50,100                :rem 99
3070 DATA 2,50,50                 :rem 56
3080 DATA 2,100,50               :rem 101
3090 DATA 2,50,100               :rem 102
3100 DATA 0                       :rem 14
```

Program 2: Machine Language Drawing Routines

```
;
; MACHINE LANGUAGE DRAWING ROUTINES
;
; EQUATES
;
TIMACT  = $DC0E ;TIMER A CONTROL
MEMCTL  = $DD00 ;C64 MEMORY CONTROL
VICCTL  = $D011 ;VIC CONTROL REGISTER
VICMCT  = $D018 ;VIC MEMORY CONTROL
BMBASE  = $E000 ;BIT-MAP BASE
BMOFFS  = $08   ;8K OFFSET BYTE
SMBASE  = $C800 ;SCREEN MEMORY BASE
SMOFFS  = $20   ;SCREEN MEMORY OFFSET
BMMODE  = $20   ;BIT-MAP ENABLE BIT
;
; PAGE ZERO EQUATES
;
ROMCTL  = $1   ;ROM CONTROL REGISTER
TMP1    = $FB ;TEMP 1
TMP2    = $FD ;TEMP 2
;
; JUMP TABLE
;
        *= $C000
        JMP SVSCRN ;SAVE SCREEN PARMS
        JMP RSSCRN ;RESTORE PARMS
        JMP GRSCRN ;ENABLE GRAPHICS
        JMP CLRSCR ;CLEAR GR. SCREEN
;
; LOCAL STORAGE
;
S1       .BYTE 0 ;SAVE MEMCTL
S2       .BYTE 0 ;SAVE VICMCT
S3       .BYTE 0 ;SAVE VICCTL
;
; SAVE SCREEN PARMS
;
SVSCRN  LDA MEMCTL
        STA S1
        LDA VICMCT
        STA S2
        LDA VICCTL
        STA S3
        RTS
;
; RESTORE SAVED SCREEN PARMS
;
RSSCRN  LDA S1
        STA MEMCTL
        LDA S2
        STA VICMCT
        LDA S3
        STA VICCTL
        RTS
;
; ENABLE GRAPHICS SCREEN
;
GRSCRN  LDA MEMCTL ;SET GRAPHICS BANK
        ORA #$03
        EOR #BMBASE/$4000
        STA MEMCTL
        LDA VICMCT   ;SET OFFSETS
        AND #$07     ;CLEAR OLD BITS
        ORA #BMOFFS ;SET BIT-MAP OFFSET
        ORA #SMOFFS ;SET SCREEN OFFSET
        STA VICMCT
        LDA VICCTL   ;ENABLE BIT-MAP
        ORA #BMMODE
        STA VICCTL
```

```
        RTS
;
; SUBROUTINE: TURN OS ROM OFF
;
OSOFF   PHA
        LDA TIMACT ;TURN OF IRQ'S
        AND #$FE
        STA TIMACT
        LDA ROMCTL ;TURN OF OS ROM
        AND #$FD
        STA ROMCTL
        PLA
        RTS
;
; SUBROUTINE: TURN BASIC ROM ON
;
OSON    PHA
        LDA ROMCTL ;TURN ON OS ROM
        ORA #$Ø2
        STA ROMCTL
        LDA TIMACT ;ENABLE IRQ'S
        ORA #$Ø1
        STA TIMACT
        PLA
        RTS
;
; SUBROUTINE: FILL AN AREA OF MEMORY
;
; ON ENTRY: A= FILL BYTE
; TMP1 = POINTER TO AREA
; TMP2 = # BYTES TO FILL
;
; ON RETURN: A AND X PRESERVED.
; Y, TMP1, AND TMP2 CLOBBERED.
;
FILL    LDY TMP2+1 ;FILL WHOLE PAGES
        BEQ FILL3  ;BR IF NONE
FILL1   LDY #Ø
FILL2   STA (TMP1),Y
        INY
        BNE FILL2
        INC TMP1+1 ;INCREMENT POINTER
        DEC TMP2+1 ;DECREMENT # PAGES
        BNE FILL1  ;BR IF MORE PAGES
FILL3   LDY TMP2    ;CHECK PARTIAL PAGE
        BEQ FILL6  ;BR IF DONE
        DEY        ;CLEAR PARTIAL PAGE
        BEQ FILL5  ;GO CLEAR LAST BYTE
FILL4   STA (TMP1),Y
        DEY
        BNE FILL4
FILL5   STA (TMP1),Y ;THE LAST BYTE
FILL6   RTS
;
; CLEAR BIT-MAP AND SCREEN MEMORY
;
CLRSCR  LDA #<SMBASE ;CLEAR SCREEN MEM
        STA TMP1
        LDA #>SMBASE
        STA TMP1+1
        LDA #<1ØØØ ;1ØØØ BYTES
        STA TMP2
        LDA #>1ØØØ
        STA TMP2+1
        LDA #$21    ;GET COLORS
        JSR FILL
        LDA #<BMBASE ;CLEAR BIT-MAP
        STA TMP1
        LDA #>BMBASE
        STA TMP1+1
        LDA #<8ØØØ ;8ØØØ BYTES
        STA TMP2
        LDA #>8ØØØ
        STA TMP2+1
        LDA #Ø
        JSR OSOFF     ;TURN OS ROM OFF
        JSR FILL      ;CLEAR
        JMP OSON      ;TURN OS ROM ON
                      ;AND RETURN
        .END
```

©

INSIGHT: Atari

Bill Wilkinson

Well, here it is April again. Those of you who read my April column last year may recall that I devoted much of my space to an April Fool announcement of the "new Atari COBOL." Would it surprise you to learn that this year I will also "announce" some new products from Atari?

We're going to be exploring some medium hefty programming logic (in machine language) next month: How to use your 1050 disk drive in enhanced density with Atari DOS 2.0s. For now, though, let's plunge into some wild speculations, rumors, and April fun.

Incredible Integration

Since last year's April announcement was about some literally unbelievable software, it seems only fitting that this year we make a hardware announcement that's almost as doubt-provoking.

By the time you read this article, Atari will be shipping at least the first two of three magnificent new machines. These machines, while maintaining almost full compatibility with existing Atari hardware and software, add the full power of an intelligent peripheral expansion bus. Imagine an Atari computer hooked up to a 5- or 10-megabyte hard disk drive, a true parallel printer interface, a high-speed modem, and maybe even a CP/M emulation package.

I mean, we're talking about possibly moving data to and from a disk drive at 30,000 to 60,000 bytes per second! Imagine taking less than two seconds to load the largest possible programs. And perhaps talking via a serial interface (or, better yet, a local network) to one or more other computers at the same time—at data rates perhaps three to ten times what an 850 Interface Module is capable of.

Software Compatibility, Too

Of course, if you are a realist, you will say, "Okay for the hardware. But what about software and software compatibility?" Would you settle for a smart peripheral bus that intercepts the OS (CIO) if you are trying to do I/O via the old serial bus. You know, the cable that links your 400/800/1200 to the disk, printer, etc.? Could you accept the fact that it checks to see if you have (for example) moved "D1:" to your hard disk drive and sends the request there instead of to your 810? All automatically?

Not enough? How about if these new machines even provided ways for third-party hardware vendors to add their own boards and automatically link in device handlers for them? Imagine a music synthesizer accessed as simply "M:", thus easily callable from even Atari BASIC. Could *any* computer manufacturer possibly design such a well-integrated system?

How about if the new machines even came with a faster math package, so that they were the fastest computers in the home computer marketplace? (I choose to define a home computer as one which costs less than $1000, including at least a disk drive.)

April Fool

Well, you knew it couldn't last, didn't you? Sigh. But it was nice to dream for a paragraph or two, wasn't it? Now, are you ready for the bitter reality?

Surprise! This is my April Fool gag for this year: Almost everything you just read about the new machines is the absolute truth. Honest. No gag.

In fact, as I write this article in January, the machines I have described to you are arriving in stores by the truckfuls. And why, you ask, haven't you seen these wonder computers advertised? Ah, but you have. They are called the 600XL and the 800XL (with big brother 1450XLD still to come). But if all I claimed is true, why hasn't Atari proclaimed it to the computer world as the greatest advance ever in home computers? Now *there* is the April Fool question.

If Atari can solve some advertising and delivery problems, I think you will see a wealth of capabilities added to the new XL machines second only to the selection available to Apple II owners.

Oh, yes. I did throw one April Fool joke into the description above. Can you guess what it is?

Final Foolishness

The descriptions of the Atari super machines were accurate except for one April Fool joke. Sigh. Unfortunately, the part about the advanced, fast BASIC being built-in is still just a dream. ©

C. Regena

Part 2: File Processing

I have had *lots* of requests for programs that use TI Extended BASIC and peripherals (printers, the RS-232 interface, and disk drives). In these next two columns I will try to satisfy those readers, plus the readers who have requested file processing, by discussing a report writer that uses disk files and a printer.

Keep in mind that there are many ways to program (as many ways as there are programmers). I'm going to show you one method I use to set up a file, then generate several reports from that file. For an example, I'm going to use a hypothetical situation—a teacher has divided the class into three reading groups. The students are given the goal of presenting one oral book report each week of the term (for this example the term is ten weeks). The teacher will grade the students on how well the goal is attained. These reports keep a tally of the book presentations. As part of the necessary school paperwork, the teacher also needs to keep track of each student's address and phone number. As a personal touch, the teacher also wants to be able to acknowledge the students on their birthdays. We'll get to these last two programs next month.

Writing A Data File

First, a data file is set up using Program 1. The student information is listed in DATA statements within this program, and the program will generate a data file on diskette. You could use DATA statements within a regular program which also writes the report, but with many students a shortage of computer memory could be a problem. Also, I'm going to use this data for several programs, and I'll only have to type it once—in this data file generator program. All reports will use this one data file.

Program 1 READs the information in from DATA statements, then writes it directly out on a disk file. Line 130 is the key statement to set up the disk system for storing the data. The OPEN statement says device #3 (you can use any number) is disk drive number one, and the file of data will be called SAMPLE. The data will be stored in internal format as output and can have a variable length of 192.

To use this program with cassette instead of diskette, use this line:

```
130 OPEN #3:"CS1",INTERNAL,OUTPUT,F
    IXED 192
```

In later programs change any DSK1 and titles to CS1 within quotes, and change the VARIABLE to FIXED since cassettes can handle only FIXED-length files.

Reading The Data

Line 140 reads the data in the following order for our situation: group G, last name N$, first name F$, address A$, phone number P$, birthday expressed as a number BD, report R$, and comments C$. Line 150 prints the same information onto the diskette. Lines 160–180 stop the process if the name read is "ZZZ", which indicates the end of the file. Line 170 counts the names.

Lines 190–490 contain the data. Notice that the last DATA statement contains "ZZZ" for N$, the last name. These names are sample names only and are not meant to represent any real people. The other information is also made up for purposes of illustration. The DATA statements are in order by last name.

The first number in the DATA statement is a group number—the sample class is divided into three reading groups. We're assuming the students all live in the same city, so only the street address is stored as A$. The phone number P$ is stored as a four-digit number because they all have the same prefix. You can change this if you wish. Since the variable name is P$, you may include a hyphen in the phone number. The birthday BD is expressed as a number which consists of the month number then two digits representing the day number. For example, September 24 is month 9 and day 24 for 924. November 25 is 1125. October 5 is 1005 for month 10 and day 05.

Encode The Special Cases

The next series of numbers represents whether the student presented a book report or not. I combined all the weeks into one ten-digit number. A 1 means the student gave a report that week and 0 means he or she did not. A dash means the student was not enrolled. You may have a report that uses other symbols for other purposes—an asterisk for a different assignment, for example. The comments C$ are just to illustrate more versatility in later reports. I used AUDIT to represent a student who will not be counted for credit, and MOVED indicates the student is no longer in the class.

To try the sample programs this month, start with a new initialized diskette. Type in Program 1 then save it with a command such as SAVE DSK1.LIST1 or SAVE DSK1.NAMES or whatever label you wish. Next RUN this program with the diskette in the disk drive. The drive light will blink on and off as data is being recorded. The screen shows how many names have been processed. This program only sets up the data files—it does not write any reports.

Generating A Roster

Now go on to Program 2. This program will read information from the data file we created previously with Program 1 and print out a roster of information—the student, which group the student is in, the address, the phone, the birthdate, and any comments.

Lines 130–150 contain a procedure that reads the month names, which are later used in printing the birthdays. Line 160 initializes some variables. Note that Extended BASIC allows several variables at once to be set equal to zero. The variable I is used to count the total number of names. N is the number of auditing students, and MM is the number of students who have moved. L is a line counter.

Printer Features

My printer has a feature that will skip over perforations at the end of a page. In case your printer does not automatically do that, I have included this method. L is used to count the lines that have been printed. Line 360 can then check to see if it is time to change pages. PRINTing CHR$(12) goes to the top of a new page. If your printer cannot PRINT CHR$(12), you can put in a pause so you can reset to the top of a new page. (The sample data included here does not go more than one page. Add more names if you want to see this part work.)

Line 180 defines the printer configuration for device #2 (use any device number). Use the parameters you need for your particular printer. Consult the RS-232 manual and your printer manual to figure out your configuration. Line 190 is the OPEN statement for device #3 (any number) to read in the data. Notice that this statement matches the OPEN #3 statement in Program 1, except we use INPUT instead of OUTPUT because we will be reading in data.

Lines 200–240 print the heading. Line 250 is an INPUT #3 statement that tells the computer to READ data from device #3. This statement is just like using READ and DATA statements within the program, only we use data files instead of program statements to store the data. The items listed may be read in as needed or all on one line, but must be in the same order as we previously saved them.

Ordering The Data

Line 260 skips the name on the roster if the student has moved and counts the number of people who have moved. Line 270 skips a line when the name starts with a different letter of the alphabet. By listing the original data in alphabetical order, this roster and other reports will automatically be in alphabetical order. A blank line is printed between groups of names starting with different letters. Line 280 checks for the last data item.

Line 290 combines the last and first names. Lines 300–310 determine the birthday from the number BD. Line 330 combines the common prefix 586- with the data P$ for the phone number.

Lines 320 and 410 are IMAGE statements, a feature of Extended BASIC which makes the module worth its price if you do lots of reports. An IMAGE statement allows you to specify how a line will be printed.

You can also use IMAGE statements PRINT USING statements to line up columns of money. For example, $###.## will print a number in dollar format with the cents rounded off. Line 340 uses line 320 to print the information.

Lines 350 and 360 increment the number of names and number of lines printed. Line 370 checks for a student who is just auditing the class. Line 380 causes the computer to branch back to the INPUT statement to read the next data items.

Typing Errors

If you get a data error, the most likely cause would be mistyping the DATA statements in Program 1. When the program stops with an error, you can PRINT N$ to see which was the last name that was accepted. Use that information to try to pinpoint a typing error in Program 1. (Be sure to SAVE Program 2 before you go back to Program 1.) You also need to make sure the data items in line 250 of Program 2 are in the same order as the data items in line 150 of Program 1. You may use different variable names if you wish, as long as the strings and numeric variables are in the right order.

Lines 390–460 print the totals information. Line 470 uses CHR$(12) to tell the printer to go to a new page. Line 480 CLOSEs both devices used in the program.

Program 1: Write Data Files

```
80 REM TI EXTENDED BASIC
90 REM DISK, PRINTER
```

```
100 REM WRITE DATA FILES
110 CALL CLEAR
120 DISPLAY AT(12,5):"WRITING DATA
    FILES"
130 OPEN #3:"DSK1.SAMPLE",INTERNAL,
    OUTPUT,VARIABLE 192
140 READ G,N$,F$,A$,P$,BD,R$,C$
150 PRINT #3:G,N$,F$,A$,P$,BD,R$,C$
160 IF N$="ZZZ" THEN 180
170 I=I+1 :: DISPLAY AT(23,3):I ::
    GOTO 140
180 CLOSE #3 :: STOP
190 DATA 1,ADAMS,JENNIFER,1207 W 80
    0 S,2314,924,1001110111,""
200 DATA 2,ALLEN,MICHAEL,204 N 300
    W,3273,1125,0010001101,""
210 DATA 3,ANDERSON,PAM,112 S 350 E
    ,1122,129,1101101101,""
220 DATA 2,BAKER,MICHELLE,310 S 350
     E,4054,1005,1101110111,""
230 DATA 2,BROWN,JUSTIN,971 EVERGRE
    EN,9656,212,1111111111,""
240 DATA 1,CARTER,JODI,918 JUNIPER,
    8803,502,0111011111,""
250 DATA 3,CHRISTENSEN,WES,1804 S 8
    00 W,3102,726,1110111011,""
260 DATA 1,DAINES,BRETT,123 S 350 E
    ,4765,1020,1001110111,""
270 DATA 1,GIFF,BEAU,222 CABBAGE LA
    NE,5733,901,--11011111,""
280 DATA 2,HANSEN,GRANT,209 N 300 W
    ,8996,425,111100111-,MOVED
290 DATA 3,JENSEN,CINDY,932 EVERGRE
    EN,7532,415,1011101101,""
300 DATA 2,JENSEN,NATALIE,420 S 100
     E,3487,512,,""
310 DATA 1,JOHNSON,ROCHELLE,355 S 1
    00 E,0217,110,1101111100,""
320 DATA 1,JONES,CHERY,1502 CEDAR,0
    157,802,1111111111,""
330 DATA 3,LARSEN,BOB,120 S 350 E,8
    674,510,---1111111,""
340 DATA 1,LARSEN,BILL,56 S 300 E,1
    299,520,1111111111,""
350 DATA 3,MORRIS,TERRY,375 E 100 N
    ,5607,114,1111011110,AUDIT
360 DATA 2,NELSON,ROGER,1362 N 1700
     E,4355,1005,11001111--,MOVED
370 DATA 1,NIELSEN,TOM,236 E 500 N,
    5670,806,0111111011,""
380 DATA 2,OLSON,RICHARD,801 W 300
    S,4587,0,1100001111,""
390 DATA 3,PETERSON,RANDY,233 E 500
     N,9007,302,1111111111,""
400 DATA 2,RICH,BRITTANY,725 E 525
    S,5683,407,1110110111,AUDIT
410 DATA 2,SAWYERS,JILL,806 W 200 N
    ,,611,1010101011,""
420 DATA 3,SMITH,JEFF,305 S 350 E,7
    789,315,1110111110,""
430 DATA 2,SMITH,TROY,855 S 300 E,6
    834,722,111--11111,""
440 DATA 3,SORENSON,GREG,315 E 525
    S,4379,104,1111111111,""
450 DATA 2,TAYLOR,CAMILLE,225 E 525
     S,3225,826,1111001101,""
460 DATA 1,THATCHER,DON,534 SPRUCE,
    2758,627,1110011101,""
470 DATA 1,WHITE,DEANNA,289 CEDAR,8
    453,121,1110100100,""
480 DATA 3,WILSON,ED,342 S 300 E,21
    55,0,1001111011,""
490 DATA 5,ZZZ,ZZZ,,,0,,""
```

Program 2: Roster

```
80 REM TI EXTENDED BASIC
90 REM DISK, PRINTER
100 REM ROSTER
110 CALL CLEAR
120 DISPLAY AT(12,5):"PRINTING ROST
    ER"
130 DIM M$(12)
140 FOR I=1 TO 12 :: READ M$(I):: N
    EXT I
150 DATA JAN,FEB,MAR,APR,MAY,JUN,JU
    L,AUG,SEP,OCT,NOV,DEC
160 I,N,MM,L=0 :: L$="A"
170 REM PRINTER CONFIGURATION
180 OPEN #2:"RS232.BA=600"
190 OPEN #3:"DSK1.SAMPLE",INTERNAL,
    INPUT ,VARIABLE 192
200 PRINT #2:TAB(34);"SAMPLE CLASS"
210 PRINT #2: :TAB(37);"ROSTER"
220 PRINT #2: :TAB(33);"APRIL 15, 1
    984"
230 PRINT #2: : :"{3 SPACES}NAME";T
    AB(25);"PHONE";TAB(34);"ADDRESS
    ";TAB(61);"BIRTHDAY";TAB(71);"C
    OMMENTS"
240 PRINT #2:"-  ----";TAB(25);"---
    --";TAB(34);"-------";TAB(61);"
    --------";TAB(71);"--------": :
250 INPUT #3:G,N$,F$,A$,P$,BD,R$,C$
260 IF C$="MOVED" THEN MM=MM+1 :: G
    OTO 250
270 IF SEG$(N$,1,1)<>L$ THEN PRINT
    #2 :: L$=SEG$(N$,1,1):: L=L+1
280 IF N$="ZZZ" THEN 390
290 N$=N$&", "&F$
300 IF BD=0 THEN BD$="---" :: D=0 :
    : GOTO 330
310 BD$=STR$(BD):: D=VAL(SEG$(BD$,L
    EN(BD$)-1,2)):: M=VAL(SEG$(BD$,
    1,LEN(BD$)-2)):: BD$=M$(M)
320 IMAGE #  #################  ##
    ######  #######################
    ##{3 SPACES}### ##{3 SPACES}###
    #######
330 P$="586-"&P$
340 PRINT #2,USING 320:G,N$,P$,A$,B
    D$,D,C$
350 I=I+1
360 L=L+1 :: IF L=49 THEN PRINT #1:
    CHR$(12):: L=0 :: GOTO 200
370 IF SEG$(C$,1,5)="AUDIT" THEN N=
    N+1
380 GOTO 250
390 PRINT #2: : :
400 T=I-N
410 IMAGE ###  #################
420 PRINT #2,USING 410:I,"NAMES"
430 PRINT #2
440 PRINT #2,USING 410:N,"AUDIT"
450 PRINT #2,USING 410:MM,"MOVED"
460 PRINT #2,USING 410:T,"PRESENTLY
     ENROLLED"
470 PRINT #2:CHR$(12)
480 CLOSE #2 :: CLOSE #3
490 END
```

©

The Automatic Proofreader

For VIC, 64, And Atari

Charles Brannon, Program Editor

At last there's a way for your computer to help you check your typing. "The Automatic Proofreader" will make entering programs faster, easier, and more accurate.

The strong point of computers is that they excel at tedious, exacting tasks. So why not get your computer to check your typing for you?

With "The Automatic Proofreader" nestled in your VIC-20, Commodore 64, or Atari computer, every line you type in will be verified. It displays a special code, called a *checksum*, at the top of the screen. The checksum, either a number (VIC/64) or a pair of letters (Atari), corresponds to the line you've just typed. It represents every character in the line summed together. A matching code in the program listing lets you compare it to the checksum which the Proofreader displays. A glance is all it takes to confirm that you've typed the line correctly.

Entering The Automatic Proofreader

Commodore (VIC/64) owners should type in Program 1. Program 2 is for Atari users. Since the Proofreader is a machine language program, be especially diligent. Watch out for typing extra commas, or a letter O for a zero, and check every number carefully. If you make a mistake when typing in the DATA statements, you'll get the message "Error in DATA statements" when you RUN the program. Check your typing and try again.

When you've typed in The Automatic Proofreader, SAVE it to tape or disk at least twice *before running it for the first time*. If you mistype the Proofreader, it may cause a system crash when you first run it. By SAVEing a copy beforehand, you can reLOAD it and hunt for your error. Also, you'll want a backup copy of the Proofreader because you'll use it again and again—every time you enter a program from COMPUTE!.

When you RUN the Proofreader, the program will be POKEd safely into memory, then it will activate itself. If you ever need to reactivate it (RUN/STOP—RESTORE or SYSTEM RESET will disable it), just enter the command SYS 886 (VIC/64) or PRINT USR(1536) for the Atari.

Using The Proofreader

Now, let's see how it works. LIST the Proofreader program, move the cursor up to one of the lines, and press RETURN. If you've entered the Proofreader correctly, a checksum will appear in the top-left corner of your screen.

Try making a change in the line and hit RETURN. Notice that the checksum has changed. All VIC and 64 listings in COMPUTE! now have a number appended to the end of each line, for example, :rem 123. *Don't enter this statement*. It is just for your information. The rem is used to make the number harmless if someone does type it in. It will, however, use up memory if you enter it, and it will cause the checksum displayed at the top of the screen to be different, even if you entered the rest of the line correctly.

The Atari checksum is found immediately to the left of each line number. This makes it impossible to type in the checksum accidentally, since a program line must start with a number.

Just type in each line (without the printed checksum), and check the checksum displayed at the top of the screen against the checksum in the listing. If they match, go on to the next line. If they don't, there's a mistake. You can correct the line immediately, instead of waiting to find the error when you RUN the program.

The Proofreader is not picky with spaces. It will not notice extra spaces or missing ones. This is for your convenience, since spacing is generally not important. Occasionally proper spacing is important, but the article describing the program will warn you to be careful in these cases.

Nobody's Perfect

Although the Proofreader is an important aid, there are a few things to watch out for. If you enter a line by using abbreviations for commands, the checksum will not match up. This is because the Proofreader is very literal: It looks at the individual letters in a line, not at tokens such as PRINT. There is a way to make the Proofreader check such a line. After entering the line, LIST it. This makes the computer spell out the abbreviations. Then move the cursor up to the line and press RETURN. It should now match the checksum. You can check whole groups of lines this way. Atari users should beware of using ? as an abbreviation for PRINT—they're not the same thing in the Proofreader's eyes.

The checksum is a sum of the ASCII values of the characters in a line. VIC and 64 owners may wonder why the numbers are so small, never exceeding 255. This is because the addition is done only in eight bits. A result over 255 will roll over past zero, like an odometer past 99999. On the Atari, the number is turned into two letters, both for increased convenience and to make the Proofreader shorter. For the curious, the letters correspond to the values of the left and right nybbles added to 33 (to offset them into the alphabet). This number is then stored directly into screen memory.

Due to the nature of a checksum, the Proofreader will not catch all errors. Since $1+3+5=3+1+5$, the Proofreader cannot catch errors of transposition. In fact, you could type in the line in any order, and the Proofreader wouldn't notice. Anytime the Proofreader seems to act strange, keep this in mind. Since the ASCII values of the number 18 (49+56) and 63 (54+51) both equal 105, these numbers are equal according to the Proofreader. There really is no simple way to catch these kinds of errors. Fortunately, the Proofreader will catch the majority of the typing mistakes most people make.

If you want the Proofreader out of your way, just press SYSTEM RESET or RUN/STOP—RESTORE. If you need it again, enter SYS 828 (VIC/64) or PRINT USR(1536) (Atari). You must disable the Proofreader before doing any tape operations on the VIC or 64.

Hidden Perils

The Proofreader's home in the VIC and 64 is not a very safe haven. Since the cassette buffer is wiped out during tape operations, you need to disable the Proofreader with RUN/STOP—RESTORE before you SAVE your program. This applies only to tape use. Disk users or Atari owners have nothing to worry about.

Not so for VIC and 64 owners with tape drives. What if you type in a program in several sittings? The next day, you come to your computer, LOAD and RUN the Proofreader, then try to LOAD the partially completed program so you can add to it. But since the Proofreader is trying to hide in the cassette buffer, it is wiped out!

What you need is a way to LOAD the Proofreader after you've LOADed the partial program. The problem is, a tape load to the buffer destroys what it's supposed to load.

After you've typed in and RUN the Proofreader, enter the following lines in direct mode (without line numbers) *exactly* as shown:

```
A$="PROOFREADER.T": B$="{10 SPACES}": FOR
   X = 1 TO 4: A$=A$+B$: NEXTX
FOR X = 886 TO 1018: A$=A$+CHR$(PEEK(X)):
   NEXTX
OPEN 1,1,1,A$:CLOSE1
```

After you enter the last line, you will be asked to press record and play on your cassette recorder. Put this program at the beginning of a new tape. This gives you a new way to load the Proofreader. Anytime you want to bring the Proofreader into memory without disturbing anything else, put the cassette in the tape drive, rewind, and enter:

```
OPEN1:CLOSE1
```

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

You can now reload the Proofreader into memory whenever LOAD or SAVE destroys it, restoring your personal typing helper.

Incidentally, you can protect the cassette buffer on the Commodore 64 with POKE 178,165. This POKE should work on the VIC, but it has caused numerous problems, probably due to a bug in the VIC operating system. With this POKE, the 64 will not wipe out the cassette buffer during tape LOADs and SAVEs.

Program 1: VIC/64 Proofreader

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

Program 2: 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":END

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
```

©

1540/1541 Disk Housekeeping

Michael Maione

This simple utility will help you clean up the clutter on your 1540 or 1541 disk drive. For VIC and Commodore 64.

If you experiment with different programming techniques and save each enhancement along the way, your disks tend to get cluttered with outdated routines. Using programming techniques employed by Jim Butterfield (COMPUTE!, April 1982 and March 1983) and Steven Smith (COMPUTE!, March 1983), this short program will help with your housekeeping chores.

Type in the program, SAVE it to disk, and then give it a try. To prevent a disaster, try it first on a disk which does not include any important programs or files.

Scratching And Unscratching Files

If you choose the Scratch option, a portion of the disk directory will be displayed—just enough to fit comfortably on the VIC screen along with the query "Scratch program?" If you do not wish to scratch any of the programs listed, press the N key and another portion of the directory will be presented. Repeat this procedure until the entire directory has been displayed.

If you want to scratch a file on the list, press the Y key. Then, type in the name of the file to be scratched and press RETURN. The file will be scratched automatically, and the program will restart from the beginning. Continue this process until all unwanted files have been removed from the disk.

When the entire disk directory has been presented, you can end the program by pressing the N key in response to the scratch question.

If the Unscratch option is chosen, the program collects all free blocks off the disk and displays the names of any previously scratched files. You are then prompted with a scratched file. Enter Y to unscratch it. Sometimes the file will be partially scrambled because other files have been written over the original file. In that case, a message is displayed indicating that the file is unrecoverable.

Abbreviated Directory Listing

Lines 10–40 set the screen color, display the title and begin the program. Lines 50–190 read eight filenames from the disk directory and print them to the screen. The file sizes and types have been eliminated from the screen display to make it clearer and more concise. If you wish to include them, make the necessary changes by referring to Jim Butterfield's article "The Confusing Catalog" (COMPUTE!, March 1983).

Lines 200–240 branch the program depending on whether or not you wish to scratch a file. Line 250 ends the program when all files have been displayed and the N key is pressed. Line 260 returns to the directory for more filenames.

Line 270 gets the filename which is to be scratched and ends the program if you accidentally hit RETURN before you type a filename.

The subroutine in lines 340–380 examines the filename you enter. If the filename is longer than ten characters, it abbreviates the name and adds "*" to the end. This is done so that the filename and the scratch command together will not be longer than one VIC screen line.

Finally, lines 290–330 use the "dynamic keyboard" technique described by Steven Smith in "Automatic Commodore Program Selector" (COMPUTE!, March 1983), to scratch the file and run the program again from the beginning.

Use PRINT# Abbreviation

If you have a VIC, be sure to abbreviate the command PRINT# (P,SHIFT-R) in line 310, to insure that the filename and the command together do not exceed the 22-character line length of the VIC screen. If they are too long, the RETURNs which are POKEd into memory in line 330 are not entered properly when the END statement is reached.

The Scratch portion of the program runs on the VIC (any memory configuration) and 64. Since the line length of the 64 screen is 40 characters, abbreviating the filename when it is longer than 10 characters should not be necessary. You may wish to modify or simply eliminate the subroutine in lines 340–380. Commodore 64 users who wish to display more than eight filenames on the screen at one time can adjust line 190 accordingly.

With a little experimentation, VIC users should be able to eliminate the necessity for the subroutine which abbreviates the longer filenames. Try using branch statements and a second routine for printing the "OPEN," "scratch," "RUN," and "cursor up" instructions in lines 290–330. Also try adding lines to validate the disk and reorganize the directory. Finally, add a few lines to read the error channel, to make the program more complete.

1540/1541 Disk Housekeeping

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

```
0 REM IF YOU HAVE AN UNEXPANDED VIC, DELE
  TE 10, 11, AND 400-10020          :rem 162
10 DIMA(255),C%(77,28),D%(1),T%(224,1),S%
   (224,1),L%(224),R%(77)            :rem 38
11 D%(0)=58:D%(1)=42:Z$=CHR$(0):B$=CHR$(3
   ):D$="0"                         :rem 234
20 PRINT"{CLR}{6 DOWN}{8 SPACES}SCRATCH":
   PRINT                            :rem 251
30 PRINT"{6 DOWN}S=SCRATCH, U=UNSCRATCH"
                                    :rem 155
40 GETQ$:IFQ$=""THEN40               :rem 11
45 IFQ$="U"THEN400                    :rem 6
46 IFQ$<>"S"THEN40                   :rem 18
50 PRINT"{CLR}{DOWN}{RVS}DISK"       :rem 23
60 OPEN1,8,0,"$0"                    :rem 77
70 N$=CHR$(0)                       :rem 152
80 GET#1,A$,A$                      :rem 190
90 F$="":B=0:GET#1,A$,A$            :rem 205
95 IFC=0THENPRINT"{RVS}";           :rem 250
100 IFA$=""THENX=1:GOTO200            :rem 0
110 GET#1,A$,B$                     :rem 233
120 PRINTTAB(5);                    :rem 184
130 GET#1,A$                         :rem 89
140 IFA$=""THENPRINT:A=A+1:GOTO190:rem 40
150 IFA$=CHR$(34)THENB=1:A$=""      :rem 127
160 IFB=1THENF$=F$+A$:PRINTA$;       :rem 13
170 IFLEN(F$)>16THENC=C+1:GOTO90    :rem 144
180 GOTO130                         :rem 102
190 IFA<8THEN90                     :rem 118
200 PRINT"SCRATCH PROGRAM? Y/N"     :rem 152
210 GETZ$:IFZ$=""THEN210            :rem 123
220 IFZ$="Y"THEN270                  :rem 67
230 IFZ$="N"THEN250                  :rem 55
240 GOTO210                          :rem 98
250 IFX=1THENCLOSE1:GOTO390         :rem 209
260 PRINT"{CLR}{2 DOWN}":A=0:GOTO90
                                    :rem 226
270 CLOSE1:INPUT"{DOWN}WHICH PROGRAM";P$:
    IFP$=""THENEND                   :rem 93
280 GOSUB340                        :rem 177
290 PRINT"{CLR}{8 DOWN}"            :rem 135
300 PRINT"OPEN15,8,15"              :rem 242
310 PRINT"{2 DOWN}PR15,"CHR$(34)"S0:"X$CH
    R$(34)"                          :rem 76
320 PRINT"{2 DOWN}RUN":PRINT"{10 UP}"
                                     :rem 50
330 POKE631,13:POKE632,13:POKE633,13:POKE
    198,3:END                       :rem 147
340 FORA4=1TOLEN(P$):R$=MID$(P$,A4,1)
                                     :rem 86
350 X$=X$+R$                        :rem 110
360 NEXT                            :rem 216
370 IFLEN(X$)>10THENX$=LEFT$(X$,10)+"*"
                                    :rem 158
380 RETURN                          :rem 123
390 CLOSE15:RUN                     :rem 167
400 PRINT"{3 DOWN}LOADING FREE SECTORS":O
    PEN15,8,15,"I"+D$:GOSUB3020     :rem 110
410 OPEN3,8,3,"$"+D$:GOSUB3020       :rem 99
420 A0=1:GET#3,A$:A=ASC(A$+Z$)      :rem 100
430 READA1:IFA=A1THEN470            :rem 168
440 F%=F%+1:IFF%=3THEN510           :rem 112
450 READA1:IFA1=0THEN430            :rem 149
460 GOTO450                         :rem 108
470 READA1:IFA1=0THEN490            :rem 157
480 READB1:FORJ=A0TOA1:R%(J)=B1:NEXTJ:A0=
    J:GOTO470                        :rem 88
490 IFA=1ORA=65THEND1=1:T9=35:S9=3:D9=18
                                    :rem 118
500 IFA=67THEND1=257:T9=77:S9=4:D9=39
                                    :rem 151
510 IFT9=0THENCLOSE3:PRINT"?? DISK NOT RE
    COGNIZED ??":STOP               :rem 132
520 FORJ=1TOD1:GET#3,A$:NEXTJ        :rem 18
530 FORJ=1TOT9:T1=0                 :rem 147
540 IFJ=51THENGET#3,A$,A$,A$,A$     :rem 190
550 GET#3,A$:C=ASC(A$+Z$)            :rem 81
560 K1=0:FORK=0TOS9-1:GET#3,A$:A=ASC(A$+Z
    $)                              :rem 217
570 FORL=0TO7:A%=A/2:D1=A-A%*2:IFK1<=R%(J
    )THENC%(J,K1)=D1                 :rem 13
580 A=A%:T1=T1+D1:K1=K1+1:NEXTL,K:rem 169
590 NEXTJ                            :rem 39
600 CLOSE3                           :rem 63
610 OPEN2,8,2,"#0":GOSUB3020        :rem 255
900 K=0:PRINT"{CLR}LOOKING FOR SCRATCHED"
    :PRINT"FILES..."                :rem 130
910 T=D9:S=1                        :rem 163
920 GOSUB 2000                      :rem 221
930 FORD=2TO255STEP32:IFA(D)<>0ORA(D+1)=0
    THEN980                         :rem 111
940 IFK=0THENPRINT"DO YOU WANT TO RECOVER
    :"                              :rem 161
950 GETX$:FORK=D+3TOD+18:PRINTCHR$(A(K));
    :NEXTK:PRINT"? ";                :rem 58
960 GETX$:IFX$<>"Y"ANDX$<>"N"THEN960
                                    :rem 128
```

```
97Ø PRINTX$:IFX$="Y"THEN1Ø1Ø          :rem 185
98Ø NEXTD                             :rem 36
99Ø T=A(Ø):S=A(1):IFT=D9THEN92Ø       :rem 35
1ØØØ PRINT"THAT'S ALL ":GOTO127Ø      :rem 83
1Ø1Ø T6=T:S6=S:D6=D:T=A(D+1):S=A(D+2):L%(
     Ø)=A(D+28)+A(D+29)*256:L%=Ø      :rem 169
1Ø2Ø GETX$:PRINT"IS THIS FILE:"       :rem 88
1Ø3Ø PRINT" 1. SEQUENTIAL"            :rem 239
1Ø4Ø PRINT" 2. PROGRAM"               :rem 14
1Ø5Ø PRINT" 3. USR"                   :rem 242
1Ø6Ø IFA(D+19)=ØTHEN1Ø8Ø              :rem 38
1Ø7Ø PRINT" 4. RELATIVE"              :rem 87
1Ø8Ø PRINT"{2 SPACES}WHICH NUMBER? ";
                                      :rem 8Ø
1Ø9Ø GETX$:IFX$=""THEN1Ø9Ø            :rem 229
11ØØ X=ASC(X$)-48:IFX<1ORX>4GOTO1Ø9Ø
                                      :rem 144
111Ø PRINTX$:X=X+128                  :rem 185
112Ø IFX=132THENT%(Ø,1)=A(D+19):S%(Ø,1)=A
     (D+2Ø):IFT%(Ø,1)=ØTHEN1Ø2Ø       :rem 91
113Ø IFT>T9ORS<ØTHENT=Ø               :rem 195
114Ø IFT<1ORS>R%(T)THENPRINT" BAD CHAIN!"
     :GOTO126Ø                        :rem 235
115Ø IFC%(T,S)=ØTHENPRINT" ALLOCATED BLOC
     KS!":GOTO126Ø                    :rem 243
116Ø GOSUB3ØØØ:L%=L%+1                :rem 192
117Ø FORJ=ØTO1:PRINT#15,"M-R";CHR$(J);B$:
     GET#15,A$                        :rem 115
118Ø A(J)=ASC(A$+Z$):NEXTJ            :rem 22Ø
119Ø T4=T:S4=S:T=A(Ø):S=A(1):IFT<>ØTHEN11
     3Ø                               :rem 1Ø
12ØØ T=T%(Ø,1):S=S%(Ø,1):T%(Ø,1)=Ø:IFT<>Ø
     THEN113Ø                         :rem 132
121Ø IFL%<>L%(Ø)THENPRINT" INCORRECT BLOC
     K COUNT!":GOTO126Ø               :rem 42
122Ø T=T6:S=S6:D=D6                   :rem 1Ø4
123Ø GOSUB 3ØØØ                       :rem 9
124Ø PRINT#15,"M-W";CHR$(D);B$;CHR$(1);CH
     R$(X)                            :rem 51
125Ø PRINT#15,"U2:2,";D$;T;S:GOSUB3Ø2Ø:GO
     TO13ØØ                           :rem 227
126Ø PRINT"SORRY - IT WON'T WORK":rem 181
127Ø CLOSE2                           :rem 114
13ØØ CLOSE2:PRINT#15,"VØ":CLOSE15:FORQW=1
     TO1ØØØØ:NEXT:RUN                 :rem 59
2ØØØ REM GRAB FULL DISK BLOCK         :rem 139
2Ø1Ø GOSUB3ØØØ                        :rem 6
2Ø2Ø FORJ=ØTO255:PRINT#15,"M-R";CHR$(J);B
     $:GET#15,A$                      :rem 217
2Ø3Ø A(J)=ASC(A$+Z$):NEXTJ:RETURN:rem 241
3ØØØ REM READ BLOAD                   :rem 37
3Ø1Ø PRINT#15,"B-R"2;VAL(D$);T;S      :rem 49
3Ø2Ø REM GET ERROR STATUS             :rem 247
3Ø3Ø INPUT#15,E,E$,E1,E2              :rem 42
3Ø4Ø IFE<>ØTHENPRINT"{RVS}DISK ERROR:
     {OFF}"E;E$,E1;E2                 :rem 21
3Ø5Ø RETURN                           :rem 168
1ØØØØ DATA 1,17,2Ø,24,19,3Ø,17,35,16,Ø
                                      :rem 44
1ØØ1Ø DATA 65,17,2Ø,24,18,3Ø,17,35,16,Ø
                                      :rem 1Ø2
1ØØ2Ø DATA 67,39,28,53,26,64,24,77,22,Ø
                                      :rem 126
```

Hidden Atari DOS Commands

Jason Lex Thomas

Even if you have only one disk drive, you can use COPY FILE and a few one-line programs to add power and convenience to your Atari DOS. There are a number of short examples to type in and try out.

The Atari Disk Operating System (DOS) menu is a very powerful tool. It provides many selections to help both the beginner and the advanced user get the most out of the Atari computer system. This article explores some special uses of selection C, COPY FILE.

COPY FILE is mainly used by owners of two or more disk drives who wish to transfer a file or series of files between drives. This is a powerful alternative to using the DUPLICATE FILE selection (option O). But COPY FILE also can be used in other ways, even if you have only one drive.

The Key Is Device Names

Before we get into the details, though, let's examine a few aspects of Atari DOS. When you select COPY FILE from the menu, the system asks you for source and destination filenames. The interesting fact is that you can also enter a device name here. (Not all device names are legal, but quite a few are.) This is not that surprising, because the Atari is an input/output based machine. To prove it, this BASIC program

```
10 PRINT "enter your name"
20 INPUT A$
```

can easily be written as

```
10 OPEN #1,4,0,"E:"
20 PRINT #1;"enter your name"
30 INPUT #1;A$
40 CLOSE #1
```

The result is about the same. And the Atari BASIC interpreter translates the two programs into similar code. The Atari uses these input/output control blocks (also called channels) for the editor (E:), keyboard (K:), and screen (S:) without the user having to know anything about them.

Confidential Input

Note that in line 10 of the second program above, the input and output are processed through the editor. We could have opened the keyboard (K:) instead, but then the name we keyed in would not have appeared on our screen. If we were incorporating a password routine into our program, we could open the keyboard rather than the editor, so that the password we enter will not be echoed to the screen.

To relate all this to COPY FILE, you can get a listing of a program or a printout of a sequential disk file if you select COPY FILE and then type:

D:*filename*,P:

when it asks you for the source and destination filenames. This will print the information that is in D:*filename* on the printer. This works well for programs you have LISTed to a diskette.

It also works for sequential text files that you have created for uploading or downloading on data base networks like CompuServe. Unfortunately, the cassette handler (C:) is not supported by the COPY FILE selection.

Copy Keyboard Entries To Disk

Perhaps the most intriguing device combination that you can enter is

E:,D:*filename*

which will let you copy anything that you type into a disk file.

The power of this technique derives from the fact that Atari has a direct-mode BASIC. That is, some BASIC instructions can be entered and executed immediately without a line number. Of course, instructions such as GOTO, GOSUB, and a few others which require line numbers are not as useful as if they are within a program.

The next few programs must be typed in using the abbreviations. To type these in, do the following:

1. Type DOS to go into the DOS menu.
2. Select option C. COPY FILE.
3. When it asks for filenames, type in E:,D:filename (for the programs that follow).
4. Type in the one-line program, followed by a carriage return.
5. Type a CTRL 3 to tell the system that you have finished.
6. At this point, you return to the DOS menu.

To use these programs from BASIC, simply type in

EN."D:filename

and the program will begin execution automatically. The best thing of all about these programs is that they occupy only one disk sector each and simulate many of the DOS commands without actually going to the DOS menu. Also, they require little memory and leave your program intact. It's great to be in the middle of program editing and only need to type EN. "D:DIR to get a directory.

These examples are all one-line programs, but you can have multiple lines. You'll have to be a little careful. Remember, no GOTOs or GOSUBs allowed!

Following is a series of programs which are keyed in using the COPY FILE function to copy from the editor to the disk. These programs are just a few examples of what can be done. All of the programs are to be keyed in followed by a single carriage return and a CTRL 3 as explained in the body of the text.

Disk Directory

This program displays a directory of the files on the disk in drive 1. Note that an Error 136 will occur at the end of the listing if you have fewer than 64 files on the diskette. This occurs because you have reached the end of file and can't TRAP anywhere. Simply type END and <RETURN> when through. Here's D:DIR:

```
CLR:T.40000:CL.#1:O.#1,6,Ø,"D:*.*":
DIMA$(3Ø):F.X=1TO64:I.#1;A$:?A$:N.X:
 CLR:CL.#1:END
```

Erase File

This program will ask you which file you wish to delete from the disk, and then delete it without going to DOS. The program D:KILL follows:

```
CLR:T.40000:CL.#1:O.#1,4,Ø,"E:": DIM
A$(3Ø):?"killfile-":I.#1;A$: CL.#1:
O.#1,4,Ø,A$:XIO33,#1,Ø,Ø,A$:CL.#1:CL
R:END
```

Protect File

This program, D:LOCK, performs the LOCK FILE function of the DOS menu:

```
CLR:T.40000:CL.#1:O.#1,4,Ø,"E:": DIM
A$(3Ø):?"lockfile-":I.#1;A$:CL.#1: O
.#1,4,Ø,A$:XIO35,#1,Ø,Ø,A$:CL.#1:CLR
:END
```

Disable File Protection

D:UNLOCK will perform the UNLOCK FILE selection of the DOS menu:

```
CLR:CL.#1:O.#1,4,Ø,"E:":DIMA$(3Ø): ?
"unlockfile":I.#1;A$:CL.#1:O.#1,4,Ø,
A$: xio36,#1,Ø,Ø,A$:CL.#1:CLR:END
```

Format Disk

D:FORMAT performs the FORMAT DISK option of the DOS menu:

```
CLR:CL.#1:O.#1,4,Ø,"E:":?"insert dis
k to format and hit any key":GET#1,A
:CL.#1:XIO254,#1,Ø,Ø,"D1:":END
```

Write DOS

This program performs the WRITEDOS function of the DOS menu. However, D:WRITEDOS will write DOS.SYS only:

```
CLR:CL.#1:O.#1,4,Ø,"E:":?"Insert dis
k to write DOS & hit any key":GET#1,
A:CL.#1:O.#1,8,Ø,"D:DOS.SYS":CL.#1
```

Change Colors

This program will allow you to alter the screen color as you desire. Try different number combinations for the POKEs into the color registers until you find a combination that suits you. This is D:COL:

```
POKE 7Ø9,Ø:POKE 71Ø,1Ø:POKE 712,114
```
©

Function Keys For The Apple

Ilan Reuben

Frequently used commands (or sequences of commands) can be input more easily by function keys defined by the programmer. This machine language program lets you define up to five function keys for the Apple; you can enter the program using the Apple's monitor or the BASIC loader provided.

A standard feature of many newer computers (for example, the Commodore 64, the VIC-20, and many Japanese computers) is the function key. It allows the user to easily enter frequently used commands, or even sequences of commands, and get it right every time. (How many times have you rushed your fingers over the keyboard, only to see you've typed CATALOF or LISY?)

Since the Apple has no special function keys, the only way to get them is through software. "Function Keys" is a machine language utility which provides up to five programmable function keys. A little over half a page (128 bytes) in length, the program resides at location $300 (decimal 768). It also uses a range of memory from $9000 to $94FF. It can be entered directly using the Apple's built-in monitor (Program 1) or with a BASIC loader (Program 2).

Defining Functions

Function Keys allows you to define or use up to five function keys in direct mode. Each function can represent a series of commands of up to 256 characters in length. When you press CTRL-F, the computer will wait for you to press the number of the function you would like to use (1 through 5) or press 0 to define a function. Pressing any other number will produce a CTRL-X.

Once you have indicated what function you want to use, press RETURN and the computer will act as if you'd typed in the function you used. Pressing the key of an undefined function will just give a carriage return.

To define a function, press 0. The computer will respond with the message FN#? (1–5, 0=EXIT). It is asking you what function number you would like to define. Pressing 0 here will abort the define procedure. When the appropriate function number is pressed, the computer will input what you want it to record as that function. Type it in as if you were actually giving the computer that command. When you are done, press RETURN. If you wish to abort while you are typing it in, just press RESET.

Here is a step-by-step example of how to use function keys. Suppose you want to turn CATALOG into a function key:

1. Type CTRL-F.

2. Press 0 since you want to define a function.

3. When you see the computer's prompt (FN#?...), press 1, since you want to define function number one.

4. Type CATALOG and press RETURN.

5. Now, whenever you press ‹CTRL-F› 1 ‹RETURN› you should get a disk catalog.

Putting The Program On Disk

Entering Function Keys through the monitor will be a lot quicker than typing in the whole BASIC loader. In any case, when you are done, save it by typing:

```
BSAVE FUNCTION KEYS 1.2,A$300,L$9C
```

However, if you do use the BASIC loader, make sure you SAVE it before trying it out, in case you made any mistakes.

Once you have the machine language for "Function Keys 1.2" saved on disk, enter the setup routine (Program 3) and save it. Now, whenever you want to use Function Keys, run the setup program. It's a good idea to use the setup program as part of a HELLO program so that every time you boot that disk, Function Keys will be automatically enabled. If for some reason you want to disable Function Keys, enter 9D04:BD 9E from the monitor if DOS is enabled, and if it isn't, enter 36:F0 FD. You might need to disable Function Keys before running certain programs.

Program 1: Data For Function Keys

```
0300- C9 86 D0 1C 18 20 5B 03
0308- B0 19 A0 FF C8 C0 FF F0
0310- 42 B1 06 99 00 02 C9 8D
0318- D0 F2 84 06 A6 06 CA 60
0320- 4C BD 9E A0 10 B9 8B 03
0328- 20 F0 FD 88 10 F7 38 20
0330- 5B 03 A9 F0 8D 04 9D A9
0338- FD 8D 05 9D 20 67 FD A9
0340- 00 8D 04 9D A9 03 8D 05
0348- 9D A0 00 B9 00 02 91 06
0350- C8 D0 F8 A2 00 A9 88 8D
0358- 00 02 60 08 20 0C FD C9
0360- B6 B0 10 C9 B0 90 0C F0
0368- 18 E9 21 85 07 A9 00 85
0370- 06 28 60 28 A9 87 20 F0
0378- FD A9 98 9D 00 02 68 68
0380- 60 28 B0 02 38 60 68 68
0388- 4C 53 03 A9 D4 C9 D8 C5
0390- BD B0 AC B5 AD B1 A8 A0
0398- BF A3 CE C6
```

Program 2: BASIC Loader

```
10  FOR ADDR = 768 TO 923
20  READ BYTE
30  POKE ADDR,BYTE
40  NEXT
50  DATA 201,134,208,28,24,32,91
60  DATA 3,176,25,160,255,200,192
70  DATA 255,240,66,177,6,153,0,2
80  DATA 201,141,208,242,132,6
90  DATA 166,6,202,96,76,189,158
100  DATA 160,16,185,139,3,32,240
110  DATA 253,136,16,247,56,32,91
120  DATA 3,169,240,141,4,157,169
130  DATA 253,141,5,157,32,103
140  DATA 253,169,0,141,4,157,169
150  DATA 3,141,5,157,160,0,185,0
160  DATA 2,145,6,200,208,248,162
170  DATA 0,169,136,141,0,2,96,8
180  DATA 32,12,253,201
185  DATA 182,176,16
190  DATA 201,176,144,12,240,24
200  DATA 233,33,133,7,169,0,133
210  DATA 6,40,96,40,169,135,32
220  DATA 240,253,169,152,157,0,2
230  DATA 104,104,96,40,176,2,56
240  DATA 96,104,104,76,83,3,169
250  DATA 212,201,216,197,189,176
260  DATA 172,181,173,177,168,160
270  DATA 191,163,206,198
```

Program 3: Setup Routine

```
10 D$ =  CHR$ (4): REM   CTRL-D
20  PRINT D$"BLOADFUNCTION KEYS 1.2"
30 OUTVEC = 9 * 4096 + 13 * 256 + 4: REM
     $9D04
40  POKE OUTVEC,0: POKE OUTVEC + 1,3
50  PRINT : PRINT "FUNCTION KEYS ENABLE
     D."
```

©

VARIABLE LISTER

Kenneth D. Day

"Variable Lister" displays a list of current values for all variables and array elements, while your program is running. This can be a most valuable debugging tool. This relocatable subroutine is written in machine language for VIC. Versions are also included for the 64 and Atari. Refer to the "Automatic Proofreader" article before typing in these programs.

"Variable Lister" is a 6502 machine language subroutine which prints the values of variables in BASIC programs. The program makes it easy to inspect the values of all variables in a BASIC program at various preselected points during execution or after the program has been aborted by the BASIC interpreter. The subroutine is called by SYS followed by the decimal address of the entry point into the subroutine (SYS 65336). The subroutine does not interfere with the execution of the BASIC program, so it may be called several times within a program.

When called, the subroutine clears the screen and prints the Variable Lister heading. It then begins to scroll through the values of all simple (nonarray) variables. The order in which the variables are listed is determined by their order of occurrence during the BASIC program's execution.

The variable name is printed, followed by an equal sign and the value of the variable. String variable values are enclosed in quotes so the user can distinguish blank strings (" ") from the *null string* (""). Integer (numbers with no decimal point) and floating-point values are printed exactly as they would be from BASIC, because the subroutine uses the output routines of the BASIC interpreter.

Each Array Element Is Listed

After simple variables are displayed, the program begins scrolling through arrays. A listing of an array begins with a display of the array name and its dimensions. The program then scrolls through each element, printing the element name followed by an equal sign and the value of the element. Values are printed using the same conventions as those for simple variables. Array elements are printed with leftmost subscripts changing the most quickly.

A pause feature is included so you can temporarily halt the scrolling display. To pause, press any key except HOME and the down arrow. Scrolling is restarted by pressing RETURN or any printable character key.

To speed up your search through long variable lists, you can press the down-arrow key for a rapid scroll rate. To clear the screen and skip to the next array, press the HOME key. You can also use this feature while examining simple variables, to skip to the first array.

RETURN Ends The Listing

When you're ready to return from the subroutine, simply press the RETURN key.

Please note that the special functions of the HOME and RETURN keys are not available while the subroutine is in pause mode. Pressing these keys while the program is pausing only restarts the scrolling.

The relocating loader (Program 1) will load Variable Lister into any VIC-20 with or without memory expansion. Variable Lister requires 867 bytes, so the BASIC loader program will just fit into memory on an unexpanded VIC-20. The DATA statements in the loader program contain the hexadecimal equivalent of the unrelocated machine code for the Variable Lister subroutine. Had I separated the pairs of hexadecimal numbers representing each byte in memory with commas, the loader program could not have been run on an unexpanded VIC-20.

Memory Pointers Are Reset

Lines 6–7 move the top-of-memory pointers so that the Variable Lister subroutine will be protected from the operating system and BASIC interpreter.

Lines 9–13 POKE the unrelocated machine language into the area made available at the top of memory.

Lines 14–17 relocate addresses within the subroutine to be consistent with its location in the machine on which it is loaded.

Line 18 tells you how to call Variable Lister by printing SYS followed by the decimal address of the entry point of the machine language subroutine. This line also POKEs the entry point address into the address area used by the USR function. If you forget the decimal address of the entry point for Variable Lister, SYS PEEK(1) + PEEK (2)*256 instead.

You can also call the subroutine with a statement like X = USR(0), but on return from the subroutine the BASIC interpreter will abort with a TYPE MISMATCH error. (However, it's not good programming style to call this subroutine as if it were a function which returns a value.) Line 18 also deletes the loader by issuing the NEW command, so be sure to SAVE a copy before you RUN the program.

To use VIC Variable Lister, first LOAD and RUN the loader (Program 1), then LOAD and RUN the program for which you wish the variables listed. After the program has run, give the SYS to start Variable Lister. If you need to check the value of variables at some point before the end of the program, simply include a SYS*xxxx* statement at the desired point in your program, where xxxx is the proper SYS address for your memory configuration.

If you would rather not enter the program yourself, I will send you a cassette tape copy (VIC version *only*). Send $3, a blank cassette tape, and a self-addressed, stamped mailer to:

Prof. Kenneth D. Day
Baldy 537
Department of Communication
SUNY at Buffalo
Buffalo, NY 14260

Program 1: VIC Variable Lister

```
6 L=PEEK(644)*256+PEEK(643)-867:POKE644,I
  NT(L/256):POKE643,L-PEEK(644)*256
                                 :rem 183
7 FORI=56TO52STEP-2:POKEI,PEEK(644):POKEI
  -1,PEEK(643):NEXT              :rem 110
8 PRINT"{CLR}{6 DOWN}{3 SPACES}LOADING PR
  OGRAM"                          :rem 24
9 P=L+34:FORI=1TO21:READS$:FORJ=1TO40:C$=
  LEFT$(S$,2):S$=RIGHT$(S$,80-J*2):V=0
                                  :rem 62
10 IFLEFT$(C$,1)="X"THEN14       :rem 193
11 D1=ASC(LEFT$(C$,1)):D2=ASC(RIGHT$(C$,1
   ))                            :rem 108
12 V=-(D1>64)*(D1-55)*16+-(D1<65)*(D1-48)
   *16+-(D2>64)*(D2-55)+-(D2<65)*(D2-48)
                                  :rem 39
13 POKEP,V:P=P+1:NEXT:NEXT       :rem 202
14 P=P-1:IFP=L+43THEN18          :rem 108
15 IFPEEK(P)<48ORPEEK(P)>51THEN14  :rem 2
16 IFP=L+80ORP=L+108ORP=L+405ORP=L+412THE
   N14                           :rem 174
17 V=PEEK(P)*256+PEEK(P-1)+L-12288:POKEP,
   INT(V/256):POKEP-1,V-PEEK(P)*256:P=P-1
   :GOTO14                        :rem 99
18 L=L+43:PRINT"{CLR}TYPE ";"{RVS}SYS";L;
   "{OFF} TO USE":POKE2,INT(L/256):POKE1,
   L-PEEK(2)*256:NEW              :rem 85
50 DATA0D504D554420524156A99320D2FFA207A9
   1D20D2FFCAD0FAA209BD213020D2FFCAD0F7A9
   0D20D2FF                      :rem 132
51 DATAA52F8D0930A5308D0A30A52D8D0030A52E
   8D0130189035AD0030C52FD00AAD0130C530D0
   034C8C31                      :rem 223
52 DATA20AD30A93D20D2FF200C31A90D20D2FF20
   D2FFA205AC0030C8D003EE0130CAD0F78C0030
   A9042082                       :rem 43
53 DATA32201D331890C0AE01308E0A30AE00308E
   0930A0008C08308C0230AD00308526AD013085
   27B12610                      :rem 101
54 DATA05297FEE02308D063020D2FFC8B1261008
   297FEE0230EE02308D073020D2FFAE0230F013
   CAF010CA                       :rem 27
```

Commodore 64 And Atari Version Notes To Variable Lister

Kevin Martin, Editorial Programmer

The 64 version of "Variable Lister" works just like the VIC version, except that it is stored in the free memory above address $C000, safe from BASIC. Install the lister by loading and running Program 2. You can initiate the listing of variable values by typing SYS 50000. As an aid in program debugging, you can also include SYS 50000 as a statement in your program wherever you want to see the current values of all the variables.

If you have typed in the "MLX" machine language editor program from a previous issue of COMPUTE!, you can use it to make a copy of the machine language portion of Variable Lister. LOAD and RUN Program 2, then LOAD and RUN MLX. Specify 49991 as the starting address and 50824 as the ending address, then use the MLX Save option to store the machine language on tape or disk. The program thus created can be reloaded without erasing a BASIC program already in memory by typing:

LOAD "filename",1,1 for tape

or

LOAD "filename",8,1 for disk.

As before, the variable listing is initiated with SYS 50000.

The Atari version of Variable Lister is a combination of two programs from *The Atari BASIC Sourcebook* (COMPUTE! Books). The programs are found on pages 123 and 125–26.

Before typing this program in, be sure to type NEW. Type this program in and, instead of SAVEing it, LIST the program to disk (LIST"D:VARLIST") or to cassette (LIST"C:"). Then type NEW in again and LOAD the program for which you want to list variables. Finally, use ENTER to add the Variable Lister subroutine from disk (ENTER"D:VARLIST") or cassette (ENTER"C:"). Then type GOTO 32700 to obtain the listing of the variables.

```
55 DATAFØØ5A925189ØØ2A9248DØ83Ø2ØD2FFAEØØ
   3ØE8DØØ3EEØ13ØE88EØØ3ØDØØ3EEØ13Ø6ØAEØ2
   3ØDØ14AC                          :rem 218
56 DATAØ13ØADØØ3Ø2ØA2DB2ØDDDAØØ1A9ØØ2Ø1E
   CB6ØCADØØ16ØCACAFØ41A9222ØD2FFADØØ3Ø85
   26ADØ13Ø                           :rem 23
57 DATA8527AØØØB1268DØD3ØC8B1268529C8B126
   852AACØD3ØB1298DØ53ØA9ØØ9129A42AA5292Ø
   1ECBACØD                            :rem 9
58 DATA3ØADØ53Ø9129A9222ØD2FF6ØAØØØADØØ3Ø
   8526ADØ13Ø8527B1268DØ53ØC8B126A8ADØ53Ø
   2Ø91D318                          :rem 168
59 DATA9Ø8EA9Ø22Ø8232ADØØ3ØC531DØØ8ADØ13Ø
   C532DØØ16Ø2ØA13ØA92Ø2ØD2FFA9442ØD2FFA9
   492ØD2FF                          :rem 228
6Ø DATAA94D2ØD2FFA92Ø2ØD2FFA9282ØD2FFAØØØ
   ADØØ3Ø8526ADØ13Ø8527B126186DØ93Ø8DØ93Ø
   C8B1266D                           :rem 13
61 DATAØA3Ø8DØA3ØC8B1268DØB3ØØAAAAØØ3B126
   99ØB3ØC8CADØF7ADØB3ØØAAABDØD3ØA8CA8EØ3
   3ØBDØD3Ø                          :rem 1Ø8
62 DATA882Ø8631AEØ33ØCADØØDA9292ØD2FFA9ØD
   2ØD2FF189ØØ8A92C2ØD2FF189ØD6ADØB3ØØA18
   69Ø36DØØ                           :rem 5Ø
63 DATA3Ø8DØØ3ØA9ØØ6DØ13Ø8DØ13ØA9Ø42Ø8232
   2Ø93322ØA4322ØØC31A9ØD2ØD2FF2Ø1D33C91F
   FØØ5A9Ø2                          :rem 1Ø9
64 DATA2Ø82322Ø1D33A9Ø538EDØ23Ø186DØØ3Ø8D
   ØØ3ØA9ØØ6DØ13Ø8DØ13ØCDØA3ØDØCCADØØ3ØCD
   Ø93ØDØC4                          :rem 162
65 DATAA9ØD2ØD2FF4C8C31A2FFAØFF88DØFDCADØ
   F838E9Ø1DØF1186ØADØB3ØØAA8A2ØØA9ØØ9D18
   3ØE888DØ                          :rem 172
66 DATAF96ØADØ63Ø2ØD2FFADØ73Ø2ØD2FFADØ83Ø
   2ØD2FFA9282ØD2FFADØB3ØØAAACABD183ØA8CA
   BD183Ø8E                          :rem 187
67 DATAØC3Ø2Ø8631AEØC3ØFØØ8A92C2ØD2FF189Ø
   E4A9292ØD2FFA93D2ØD2FFADØB3ØØAAACABD18
   3Ø1869Ø1                           :rem 99
68 DATA9D183ØCAA9ØØ7D183Ø9D183ØDDØE3ØFØØ1
   6ØE8BD183ØDDØE3ØFØØ16ØA9ØØ9D183ØCA9D18
   3ØDØØ16Ø                          :rem 243
69 DATA189ØCEA5C5C91FDØØ16ØA6C6FØFB2ØE4FF
   CADØFAC911DØØ3A9ØØ6ØC9ØDDØØ368686ØC913
   DØ16ADØ9                          :rem 122
7Ø DATA3Ø8DØØ3ØADØA3Ø8DØ13ØA9932ØD2FF6868
   4C8C31A6C6FØFC2ØE4FFCADØFAA9ØØ6ØXXXXXX
   XXXXXXX                            :rem 29
```

Program 2: 64 Variable Lister

```
1Ø I=49991                            :rem 247
2Ø READ A:IF A=256 THEN 4Ø             :rem 54
3Ø POKE I,A:CK=CK+A:I=I+1:GOTO 2Ø:rem 129
4Ø IF CK<>1Ø8449 THEN PRINT"CHECK FOR ERR
   OR IN DATA STATEMENTS":STOP          :rem 61
5Ø PRINT"{DOWN}DATA LOADED . . .":PRINT"
   {DOWN}TYPE {RVS}SYS 5ØØØØ{OFF} TO STAR
   T":END                              :rem 2Ø4
49991 DATA 13,84,83,73,76,32          :rem 125
49997 DATA 82,65,86,169,147,32        :rem 24Ø
5ØØØ3 DATA 21Ø,255,162,16,169,29       :rem 4Ø
5ØØØ9 DATA 32,21Ø,255,2Ø2,2Ø8,25Ø      :rem 77
5ØØ15 DATA 162,9,189,7Ø,195,32        :rem 2Ø8
5ØØ21 DATA 21Ø,255,2Ø2,2Ø8,247,169
                                      :rem 136
5ØØ27 DATA 13,32,21Ø,255,165,47       :rem 243
5ØØ33 DATA 141,46,195,165,48,141       :rem 46
5ØØ39 DATA 47,195,165,45,141,37         :rem 6
5ØØ45 DATA 195,165,46,141,38,195       :rem 57
5ØØ51 DATA 24,144,53,173,37,195       :rem 252
5ØØ57 DATA 197,47,2Ø8,1Ø,173,38         :rem 4
5ØØ63 DATA 195,197,48,2Ø8,3,76        :rem 218
5ØØ69 DATA 177,196,32,21Ø,195,169:rem 112
5ØØ75 DATA 61,32,21Ø,255,32,49        :rem 196
5ØØ81 DATA 196,169,13,32,21Ø,255       :rem 44
5ØØ87 DATA 32,21Ø,255,162,5,172       :rem 246
5ØØ93 DATA 37,195,2ØØ,2Ø8,3,238       :rem 252
5ØØ99 DATA 38,195,2Ø2,2Ø8,247,14Ø:rem 1Ø3
5Ø1Ø5 DATA 37,195,169,4,32,167        :rem 2Ø9
5Ø111 DATA 197,32,66,198,24,144         :rem 1
5Ø117 DATA 192,174,38,195,142,47       :rem 56
5Ø123 DATA 195,174,37,195,142,46       :rem 54
5Ø129 DATA 195,16Ø,Ø,14Ø,45,195       :rem 25Ø
5Ø135 DATA 14Ø,39,195,173,37,195       :rem 56
5Ø141 DATA 133,38,173,38,195,133       :rem 47
5Ø147 DATA 39,177,38,16,5,41          :rem 11Ø
5Ø153 DATA 127,238,39,195,141,43       :rem 51
5Ø159 DATA 195,32,21Ø,255,2ØØ,177      :rem 94
5Ø165 DATA 38,16,8,41,127,238         :rem 157
5Ø171 DATA 39,195,238,39,195,141       :rem 61
5Ø177 DATA 44,195,32,21Ø,255,174       :rem 49
5Ø183 DATA 39,195,24Ø,19,2Ø2,24Ø       :rem 44
5Ø189 DATA 16,2Ø2,24Ø,5,169,37        :rem 2Ø5
5Ø195 DATA 24,144,2,169,36,141        :rem 2Ø2
5Ø2Ø1 DATA 45,195,32,21Ø,255,174       :rem 38
5Ø2Ø7 DATA 37,195,232,2Ø8,3,238       :rem 254
5Ø213 DATA 38,195,232,142,37,195       :rem 5Ø
5Ø219 DATA 2Ø8,3,238,38,195,96        :rem 218
5Ø225 DATA 174,39,195,2Ø8,2Ø,172       :rem 49
5Ø231 DATA 38,195,173,37,195,32         :rem 4
5Ø237 DATA 162,187,32,221,189,16Ø      :rem 99
5Ø243 DATA 1,169,Ø,32,3Ø,171           :rem 86
5Ø249 DATA 96,2Ø2,2Ø8,1,96,2Ø2        :rem 2Ø3
5Ø255 DATA 2Ø2,24Ø,65,169,34,32       :rem 248
5Ø261 DATA 21Ø,255,173,37,195,133      :rem 94
5Ø267 DATA 38,173,38,195,133,39        :rem 13
5Ø273 DATA 16Ø,Ø,177,38,141,5Ø        :rem 195
5Ø279 DATA 195,2ØØ,177,38,133,41       :rem 52
5Ø285 DATA 2ØØ,177,38,133,42,172       :rem 45
5Ø291 DATA 5Ø,195,177,41,141,42       :rem 251
5Ø297 DATA 195,169,Ø,145,41,164         :rem 6
5Ø3Ø3 DATA 42,165,41,32,3Ø,171        :rem 185
5Ø3Ø9 DATA 172,5Ø,195,173,42,195       :rem 53
5Ø315 DATA 145,41,169,34,32,21Ø       :rem 242
5Ø321 DATA 255,96,16Ø,Ø,173,37        :rem 2ØØ
5Ø327 DATA 195,133,38,173,38,195       :rem 61
5Ø333 DATA 133,39,177,38,141,42       :rem 253
5Ø339 DATA 195,2ØØ,177,38,168,173:rem 111
5Ø345 DATA 42,195,32,145,179,24         :rem 2
5Ø351 DATA 144,142,169,2,32,167       :rem 249
5Ø357 DATA 197,173,37,195,197,49       :rem 77
5Ø363 DATA 2Ø8,8,173,38,195,197        :rem 15
5Ø369 DATA 5Ø,2Ø8,1,96,32,198         :rem 163
5Ø375 DATA 195,169,32,32,21Ø,255       :rem 5Ø
5Ø381 DATA 169,68,32,21Ø,255,169       :rem 57
5Ø387 DATA 73,32,21Ø,255,169,77         :rem 9
5Ø393 DATA 32,21Ø,255,169,32,32       :rem 248
5Ø399 DATA 21Ø,255,169,4Ø,32,21Ø       :rem 43
5Ø4Ø5 DATA 255,16Ø,Ø,173,37,195       :rem 251
5Ø411 DATA 133,38,173,38,195,133       :rem 47
5Ø417 DATA 39,177,38,24,1Ø9,46        :rem 215
5Ø423 DATA 195,141,46,195,2ØØ,177      :rem 99
5Ø429 DATA 38,1Ø9,47,195,141,47        :rem 1Ø
5Ø435 DATA 195,2ØØ,177,38,141,48       :rem 52
5Ø441 DATA 195,1Ø,17Ø,16Ø,3,177       :rem 245
5Ø447 DATA 38,153,48,195,2ØØ,2Ø2       :rem 47
5Ø453 DATA 2Ø8,247,173,48,195,1Ø       :rem 53
5Ø459 DATA 17Ø,189,5Ø,195,168,2Ø2:rem 11Ø
5Ø465 DATA 142,4Ø,195,189,5Ø,195       :rem 58
5Ø471 DATA 136,32,171,196,174,4Ø       :rem 47
5Ø477 DATA 195,2Ø2,2Ø8,13,169,41       :rem 51
5Ø483 DATA 32,21Ø,255,169,13,32       :rem 247
5Ø489 DATA 21Ø,255,24,144,8,169         :rem 6
```

```
50495 DATA 44,32,210,255,24,144   :rem 248
50501 DATA 214,173,48,195,10,24   :rem 245
50507 DATA 105,3,109,37,195,141   :rem 249
50513 DATA 37,195,169,0,109,38    :rem 210
50519 DATA 195,141,38,195,169,4    :rem 13
50525 DATA 32,167,197,32,184,197   :rem 62
50531 DATA 32,201,197,32,49,196   :rem 255
50537 DATA 169,13,32,210,255,32   :rem 247
50543 DATA 66,198,201,31,240,5    :rem 199
50549 DATA 169,2,32,167,197,32    :rem 216
50555 DATA 66,198,169,5,56,237    :rem 228
50561 DATA 39,195,24,109,37,195    :rem 11
50567 DATA 141,37,195,169,0,109     :rem 6
50573 DATA 38,195,141,38,195,205   :rem 59
50579 DATA 47,195,208,204,173,37   :rem 63
50585 DATA 195,205,46,195,208,196:rem 118
50591 DATA 169,13,32,210,255,76   :rem 255
50597 DATA 177,196,162,255,160,255
                                  :rem 167
50603 DATA 136,208,253,202,208,248
                                  :rem 142
50609 DATA 56,233,1,208,241,24    :rem 197
50615 DATA 96,173,48,195,10,168    :rem 12
50621 DATA 162,0,169,0,157,61     :rem 145
50627 DATA 195,232,136,208,249,96:rem 114
50633 DATA 173,43,195,32,210,255   :rem 44
50639 DATA 173,44,195,32,210,255   :rem 51
50645 DATA 173,45,195,32,210,255   :rem 49
50651 DATA 169,40,32,210,255,173   :rem 42
50657 DATA 48,195,10,170,202,189   :rem 55
50663 DATA 61,195,168,202,189,61   :rem 60
50669 DATA 195,142,49,195,32,171   :rem 64
50675 DATA 196,174,49,195,240,8    :rem 19
50681 DATA 169,44,32,210,255,24   :rem 252
50687 DATA 144,228,169,41,32,210   :rem 50
50693 DATA 255,169,61,32,210,255   :rem 52
50699 DATA 173,48,195,10,170,202   :rem 54
50705 DATA 189,61,195,24,105,1    :rem 204
50711 DATA 157,61,195,202,169,0   :rem 251
50717 DATA 125,61,195,157,61,195   :rem 59
50723 DATA 221,51,195,240,1,96    :rem 199
50729 DATA 232,189,61,195,221,51   :rem 55
50735 DATA 195,240,1,96,169,0     :rem 159
50741 DATA 157,61,195,202,157,61   :rem 50
50747 DATA 195,208,1,96,24,144    :rem 213
50753 DATA 206,165,197,201,31,208 :rem 96
50759 DATA 1,96,166,198,240,251    :rem 13
50765 DATA 32,228,255,202,208,250 :rem 95
50771 DATA 201,17,208,3,169,0     :rem 146
50777 DATA 96,201,13,208,3,104    :rem 200
50783 DATA 104,96,201,19,208,22   :rem 252
50789 DATA 173,46,195,141,37,195   :rem 70
50795 DATA 173,47,195,141,38,195   :rem 69
50801 DATA 169,147,32,210,255,104 :rem 89
50807 DATA 104,76,177,196,166,198:rem 122
50813 DATA 240,252,32,228,255,202 :rem 87
50819 DATA 208,250,169,0,96,256    :rem 10
```

Program 3: Atari Variable Lister

```
JE 32500 Q=WW-1
JD 32505 Q=PEEK(134)+256*PEEK(135)+(Q-
         128)*8
OL 32510 PRINT :PRINT "VARIBLE NUMBER
         ";PEEK(Q+1),
JJ 32515 ON INT(PEEK(Q)/64) GOTO 32600
         ,32650
IO 32520 PRINT "IS A NUMBER, ":PRINT ,
         "VALUE ";
KC 32525 QEXP=PEEK(Q+2):IF QEXP>127 TH
         EN PRINT "-";:QEXP=QEXP-128
LA 32530 QNUM=0:FOR QQ=Q+3 TO Q+7
FB 32535 QNUM=QNUM*100+PEEK(QQ)-6*INT(
         PEEK(QQ)/16):NEXT QQ
PJ 32540 QEXP=QEXP-68:IF QEXP=0 THEN 3
         2555
AB 32545 FOR QQ=QEXP TO SGN(QEXP) STEP
          -SGN(QEXP)
AM 32550 QNUM=(QEXP>0)*QNUM*100+(QEXP<
         0)*QNUM/100:NEXT QQ
LD 32555 PRINT QNUM:PRINT :RETURN
MH 32570 IF PEEK(Q)/2<>INT(PEEK(Q)/2)
         THEN 32580
MH 32575 PRINT ,"AND IS NOT YET DIMENS
         IONED":POP :RETURN
DB 32580 PRINT ,"ADDRESS IS ";PEEK(Q+2
         )+256*PEEK(Q+3):RETURN
FN 32600 PRINT "IS AN ARRAY, ":GOSUB 3
         2570
BK 32610 PRINT ,"DIM 1 IS ";PEEK(Q+4)+
         256*PEEK(Q+5)
CE 32615 PRINT ,"DIM 2 IS ";PEEK(Q+6)+
         256*PEEK(Q+7)
NN 32620 RETURN
GM 32650 PRINT "IS A STRING, ":GOSUB 3
         2570
NG 32660 PRINT ,"LENGTH IS ";PEEK(Q+4)
         +256*PEEK(Q+5)
PH 32665 PRINT ,"{3 SPACES}DIM IS ";PE
         EK(Q+6)+256*PEEK(Q+7)
OC 32670 RETURN
IC 32700 WW=128
IL 32710 FOR W=PEEK(130)+256*PEEK(131)
          TO PEEK(132)+256*PEEK(133)-1
JH 32720 IF PEEK(W)<128 THEN PRINT CHR
         $(PEEK(W));:NEXT W:STOP
KG 32730 PRINT CHR$(PEEK(W)-128):WW=WW
         +1:GOSUB 32500:NEXT W:STOP
```

©

Qwikload/Save For VIC And 64

Richard L. Witkover

Here is a BASIC program which can drastically reduce your waiting time when loading or saving large blocks of data.

Have you ever sat staring at your television set while saving or loading large blocks of data? If you use the GET# or INPUT# commands, chances are you have. It may have taken only a few minutes, but it seems like forever. You can do it ten times faster by using the Kernal routines built into your Commodore computer.

First, you must tell the Kernal routine where to load the data, or what section of memory to save. The Kernal looks for this information in the microprocessor's internal registers. These registers can be accessed from BASIC via memory locations 780–782. The SYS command transfers the contents of these locations into the registers before it jumps to the machine language routine. Location 780 corresponds to the accumulator, 781 to the X register, and 782 to the Y register. So, all we need to do is POKE the information into these locations and SYS to the Kernal routine. "Qwikload/Save" uses this technique to access the Kernal routines SETLFS, SETNAM, LOAD, and SAVE.

Qwikload/Save allows you to save any section of memory in the 64 and up to location 32766 ($7FFE) in the VIC. Any files saved by Qwikload/Save can also be loaded by the program into any area in RAM. Either tape or disk may be used. Just type in the program, SAVE it, and RUN it.

How It Works

Lines 100–110 ask the user whether to use tape or disk and store the answer in B$. Lines 120–130 ask whether to save or load, storing the answer in A$. Line 140 INPUTs the filename and stores it in F$. Line 144 INPUTs the starting address of the block to be saved or loaded. Then the high byte (AH) and low byte (AL) of the starting address are calculated. Line 147 branches to line 500 if disk was chosen. Line 150 branches to line 300 if load was chosen.

Lines 160–200 save a block of memory to tape. Line 160 INPUTs the end address of the block and calculates the high byte (BH) and low byte (BL). Line 190 jumps to the subroutine at line 400 to open a file. Then the high byte and low byte of the starting address are POKEd into zero page. Line 200 POKEs location 780 (accumulator) with the zero-page address of the low byte used in line 190. This creates a pointer which tells the computer where to find the starting address. The low byte of the end address is POKEd into location 781 (X register) and the high byte into location 782 (Y register). Then the block is saved by SYSing to the SAVE routine in the Kernal. The file is then closed by jumping to line 330.

Lines 300–330 load a file from tape. Line 300 opens the file. Line 310 specifies a LOAD by

POKEing 0 into location 780 (0=LOAD; 1=VERIFY). The low byte and high byte of the starting address are POKEd into locations 781 and 782, respectively. Then the file is loaded by SYSing to the LOAD routine. Line 320 checks bits 4 and 5 of the STATUS variable. If either bit is set, the file was not loaded correctly and the message ?LOAD ERROR is printed. Line 330 closes the file and ends the program.

Lines 400–440 comprise a subroutine which opens a file to the cassette recorder similar to the BASIC command OPEN 1,1,0,F$. Line 400 POKEs the length of the filename into location 183. The end-of-arrays pointer is calculated and stored in S. Line 410 POKEs the filename into the free RAM area just above the arrays. Line 420 sets up the logical file by POKEing the file number into location 780, POKEing the device number into location 781, POKEing the secondary address into location 782, and SYSing to the SETLFS routine. Line 430 sets up the filename by POKEing the filename length into location 780, POKEing the low byte of the end-of-arrays pointer into location 781, POKEing the high byte into location 782, and SYSing to the SETNAM routine. Line 440 turns on the tape messages (SEARCHING, FOUND, etc.) by setting bit 7 of location 157.

Line 500 branches to line 700 if a disk load is chosen. Lines 530–660 save a block of memory to disk. Line 530 INPUTs the end address, adds 1 to it, and calculates the low byte and high byte. It is necessary to add 1 to the end address in order to save the last byte of the block. Line 540 OPENs the disk error channel. Line 550 OPENs a program file for writing. Line 560 checks for errors by reading the error channel. Line 570 branches to line 650 if no error occurs. Line 580 prints the error information and jumps to line 760 to end the program if the error number is not 63 (FILE EXISTS ERROR).

Lines 590–610 ask whether the user wants to replace the file on disk with the new file. If not, the program ends by jumping to line 760. Otherwise, the file is replaced by scratching the file on disk and saving the new file. Line 620 scratches the old file. Line 630 returns to line 540 to save the new file. Lines 650–660 save the file by POKEing the starting and ending addresses and SYSing to the SAVE routine.

Lines 700–760 load a file from disk. Line 700 OPENs the error channel. Line 710 OPENs the program file for reading. Line 720 reads the error channel. If any error occurs, line 730 prints the error information and ends the program. Line 750 enables a relocatable load by POKEing a 0 into location 185. Then the file is loaded by POKEing the necessary information and SYSing to the LOAD routine as described for tape. Line 760 closes the files and ends the program.

Qwikload/Save For VIC And 64

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

```
100 PRINT"{CLR}TAPE OR DISK (T/D)?";
                                    :rem 125
105 GETB$:IFB$=""THEN105             :rem 81
110 IFB$<>"T"ANDB$<>"D"THEN105      :rem 153
120 PRINTB$:PRINT"{DOWN}SAVE OR LOAD (S/L
    )?";                             :rem 43
125 GETA$:IFA$=""THEN125             :rem 83
130 IFA$<>"S"ANDA$<>"L"THEN125      :rem 162
140 PRINTA$:INPUT"{DOWN}FILENAME";F$
                                    :rem 140
144 INPUT"{DOWN}STARTING ADDRESS";X:AH=IN
    T(X/256):AL=X-AH*256            :rem 188
147 IFB$="D"THEN500                  :rem 26
150 IFA$="L"THEN300                  :rem 25
159 REM TAPE SAVE                   :rem 220
160 INPUT"{DOWN}END ADDRESS";X:X=X+1:BH=I
    NT(X/256):BL=X-BH*256           :rem 171
190 GOSUB400:POKE251,AL:POKE252,AH:rem 31
200 POKE780,251:POKE781,BL:POKE782,BH:SYS
    65496:GOTO330                   :rem 247
299 REM TAPE LOAD                   :rem 210
300 GOSUB400                        :rem 167
310 POKE780,0:POKE781,AL:POKE782,AH:SYS65
    493                             :rem 131
320 IF(ST AND 48)THENPRINT"{DOWN}?LOAD":P
    RINT"ERROR"                      :rem 96
330 CLOSE1:END                       :rem 78
399 REM OPEN TAPE CHANNEL           :rem 222
400 L=LEN(F$):POKE183,L:S=256*PEEK(50)+PE
    EK(49)                          :rem 174
410 FORX=1TOL:POKES+X-1,ASC(MID$(F$,X,1))
    :NEXT                            :rem 53
420 POKE780,1:POKE781,1:POKE782,0:SYS6546
    6                               :rem 209
430 POKE780,L:POKE781,PEEK(49):POKE782,PE
    EK(50):SYS65469                  :rem 77
440 POKE157,128:RETURN               :rem 69
500 IFA$="L"THEN700                  :rem 28
529 REM DISK SAVE                   :rem 222
530 INPUT"{DOWN}END ADDRESS";X:X=X+1:BH=I
    NT(X/256):BL=X-BH*256           :rem 172
540 OPEN15,8,15,"I0"                 :rem 16
550 OPEN3,8,1,"0:"+F$+",P,W"        :rem 157
560 INPUT#15,EN,EM$,ET,ES           :rem 222
570 IFEN=0THEN650                   :rem 245
580 IFEN<>63THENPRINTEN;EM$;ET;ES:GOTO760
                                    :rem 153
590 PRINT"FILE EXISTS.{2 SPACES}REPLACE (
    Y/N)?";                          :rem 58
600 GETA$:IFA$=""THEN600             :rem 79
610 PRINTA$:IFA$<>"Y"THEN760        :rem 154
620 PRINT#15,"S0:"+F$+",P,W"        :rem 222
630 CLOSE15:CLOSE3:GOTO540          :rem 100
650 POKE157,128:POKE251,AL:POKE252,AH
                                    :rem 159
660 POKE780,251:POKE781,BL:POKE782,BH:SYS
    65496:GOTO760                     :rem 8
699 REM DISK LOAD                   :rem 215
700 OPEN15,8,15,"I0"                 :rem 14
710 OPEN3,8,0,"0:"+F$+",P,R"        :rem 149
720 INPUT#15,EN,EM$,ET,ES           :rem 220
730 IFENTHENPRINTEN;EM$;ET;ES:GOTO760
                                    :rem 179
750 POKE157,128:POKE185,0:POKE780,0:POKE7
    81,AL:POKE782,AH:SYS65493       :rem 187
760 CLOSE3:CLOSE15:END              :rem 109
```

©

VIC/64 Screenprint

Frank C. Gutowski

To copy the contents of your screen to the printer, just insert "Screenprint" into your program as a one-line subroutine. Refer to the "Automatic Proofreader" article before typing these programs in.

Using just one line of BASIC and less than 80 bytes of memory, "Screenprint" opens the needed files, captures the entire screen, sends it to the printer, and closes the files.

Most screen print routines follow this logic:

1. Find where the screen memory is.
2. Start PEEKing screen locations.
3. Convert screen code to ASCII.
4. Assemble an ASCII line and send it to the printer.
5. Repeat until the entire screen has been displayed.

Screenprint takes a different approach. By treating the screen as an input device, we can combine the first three steps into one, creating a compact yet powerful subroutine that is easy to insert into any program.

Input From Screen Memory

The screen is normally the default output device. The VIC and 64 operating systems automatically send keyboard input to the screen unless you redirect it with a CMD command. However, you also can OPEN a screen file for input and read data from screen memory, using some of the same commands and statements which control input from tape or disk.

Screenprint opens a screen file with OPEN3,3. This does not in any way affect normal PRINTs to the screen; it just opens another path (File 3) to the screen (device 3). Now, statements like GET#3 or PRINT#3 can direct input or output from or to the screen. Once the screen file is OPENed, Screenprint uses GET#3,A$ to input a screen character. The character under the cursor at the time of the GET#3 is assigned to the string variable A$, and the cursor moves one space to the right.

Dealing With Screen Wraparound

A good feature of this screen GET is that when it gets to the end of a line, it will read a CHR$(13), and cause a carriage return to be sent to the printer. This means that a line that wraps around on the screen will appear as one line on the printer. If this is not desired, it can be eliminated by inserting an IF-THEN statement that will cause carriage returns to be ignored.

This version of One-Line Screenprint is for VIC:

```
10 OPEN3,3:OPEN4,4:PRINT"{HOME}";:FORI=0T
   O505:GET#3,A$:PRINT#4,A$;:NEXT:CLOSE3:
   CLOSE4
```

Program Explanation

OPEN3,3: opens the screen file.
OPEN4,4: opens the print file.
PRINT"[HOME]";: places the cursor in the home position for the screen read.
FORI=0TO505: sets the number of characters to capture from the screen.
GET#3,A$: captures a character from the screen and moves the cursor right one space.
PRINT#4,A$;: prints the captured screen character with no RETURN.
NEXT:CLOSE4:CLOSE3:RETURN: ends the loop, closes the files and returns to the caller.

For the 64, use this version of One-Line Screenprint:

```
10 OPEN3,3:OPEN4,4:PRINT"{HOME}";:FORI=0T
   O999:GET#3,A$:PRINT#4,A$;:NEXT:CLOSE3:
   CLOSE4
```

To get all this on one line in the 64, you'll need to abbreviate some of the BASIC keywords. For example, O SHIFT-P can stand for OPEN, a question mark can be substituted for PRINT, P SHIFT-R for PRINT#, etc.

Printing Reversed Characters

The GET statement puts ASCII numbers into A$, not the screen code, so you can send the characters to the printer without translation. However, reversed characters do not have distinct ASCII numbers; these are returned as normal characters. If you must have a screen print with reversed characters, you will need another, longer routine. We must find where screen memory is (on the VIC), then look at each screen code to see if it is greater than 127 (reversed character). If the character is reversed, we have to send the appropriate commands to make the printer go into, then out of, reverse mode.

Those of you who want an exact copy of the screen, including low-resolution graphics and reversed characters, can use the screen copy routines below. The control characters sent are set up for the 1515 and 1525 printers, but may be changed for other printers.

VIC Screen Copy

```
100 PRINT"{HOME}";:SS=(PEEK(21Ø))*256:OPE
    N3,3:OPEN4,4                       :rem 236
11Ø FORR=ØTO22:B$=""                    :rem 96
12Ø FORC=ØTO21:A$=""                    :rem 8Ø
13Ø IFPEEK(SS+((R*22)+C))>127THEN:GET#3,A
    $:B$=B$+CHR$(18)+A$+CHR$(146):GOTO16Ø
                                        :rem 61
14Ø GET#3,A$:IFA$=CHR$(13)THEN:A$=" "
                                       :rem 2Ø4
15Ø B$=B$+A$                            :rem 47
16Ø NEXTC:PRINT#4,B$:NEXTR:CLOSE4:CLOSE3
                                        :rem 91
```

64 Screen Copy

```
1ØØ PRINT"{HOME}";:SS=(PEEK(21Ø))*256:OPE
    N3,3:OPEN4,4                       :rem 236
11Ø FORR=ØTO24:B$=""                    :rem 98
12Ø FORC=ØTO39:A$=""                    :rem 89
13Ø IFPEEK(SS+((R*4Ø)+C))>127THEN:GET#3,A
    $:B$=B$+CHR$(18)+A$+CHR$(146):GOTO16Ø
                                        :rem 61
14Ø GET#3,A$:IFA$=CHR$(13)THEN:A$=" "
                                       :rem 2Ø4
15Ø B$=B$+A$                            :rem 47
16Ø NEXTC:PRINT#4,B$:NEXTR:CLOSE4:CLOSE3
                                        :rem 91
```

Lines 100–120: put the cursor to home position, find the start of screen memory (SS), open the needed files, and set the row and column pointers.

Line 130: look at the screen memory. When a screen character greater than 127 is found, it must be a reversed character. The character is then captured and added to string B$ between a [RVS] and [OFF] character.

Lines 140–150: capture all nonreversed characters, remove CHR$(13), and add the captured character to the print string B$.

Line 160: send the strings to the printer, close the files and return.

Screenprint In Action

Programs 1 and 2 are demonstration programs which use both subroutines. Once LOADed and RUNning, just key in anything you wish, even create a graphics picture using graphics characters. When you desire a copy, the f1 key will call One-Line Screenprint. The f3 key will call Screen Copy. This will print the screen including reversed characters and no line wrap.

Necessary Changes

To add either routine to your existing programs, add these lines any place you want the Screenprint option:

```
1ØØ GETA$:IFA$=CHR$(133)THEN GOSUBxxx:A$=
    " "
11Ø IFA$=""THEN1ØØ
```

These lines call the Screenprint subroutine if the f1 key is depressed. Any other key will cause the program to continue. Replace the xxx with the line number of the selected Screenprint subroutine.

Don't forget, you can use the disk or tape as an output or input device just as easily as the screen. With a few modifications to Screenprint, you could be transferring screens of data to and from any device. That could make for some interesting and powerful applications, using only a few program lines.

Program 1: VIC Screenprint Demo

```
2ØØ OPEN3,3:OPEN4,4:PRINT"{CLR}KEY IN ANY
    THING"                             :rem 214
21Ø PRINT"{RVS}F1=ONE LINE SCREEN":PRINT"
    {3 RIGHT}{RVS}PRINT SUB{OFF}":rem 113
22Ø PRINT"{RVS}F3=SCREEN COPY{OFF}"
                                       :rem 186
23Ø GETA$:IFA$=CHR$(133)THENGOSUB26Ø:A$="
    "                                  :rem 149
24Ø IFA$=CHR$(134)THENGOSUB27Ø:A$=""
                                        :rem 25
25Ø PRINTA$;:A$="":GOTO23Ø             :rem 236
26Ø PRINT"{HOME}";:FORI=ØTO5Ø5:GET#3,A$:P
    RINT#4,A$;:NEXT:RETURN              :rem 73
27Ø PRINT"{HOME}";:SS=(PEEK(21Ø))*256
                                       :rem 246
28Ø FORR=ØTO22:B$=""                   :rem 1Ø4
29Ø FORC=ØTO21:A$=""                    :rem 88
3ØØ IFPEEK(SS+((R*22)+C))>127THEN:GET#3,A
    $:B$=B$+CHR$(18)+A$+CHR$(146):GOTO33Ø
                                        :rem 59
31Ø GET#3,A$:IFA$=CHR$(13)THEN:A$=" "
                                       :rem 2Ø3
32Ø B$=B$+A$                            :rem 46
33Ø NEXTC:PRINT#4,B$:NEXTR:RETURN:rem 173
```

Program 2: 64 Screenprint Demo

```
2ØØ OPEN3,3:OPEN4,4:PRINT"{CLR}KEY IN ANY
    THING"                             :rem 214
21Ø PRINT"{RVS}F1=ONE LINE SCREEN PRINT S
    UB{OFF}"                           :rem 253
22Ø PRINT"{RVS}F3=SCREEN COPY{OFF}"
                                       :rem 186
23Ø GETA$:IFA$=CHR$(133)THENGOSUB26Ø:A$="
    "                                  :rem 149
24Ø IFA$=CHR$(134)THENGOSUB27Ø:A$=""
                                        :rem 25
25Ø PRINTA$;:A$="":GOTO23Ø             :rem 236
26Ø PRINT"{HOME}";:FORI=ØTO999:GET#3,A$:P
    RINT#4,A$;:NEXT:RETURN              :rem 9Ø
27Ø PRINT"{HOME}";:SS=(PEEK(21Ø))*256
                                       :rem 246
28Ø FORR=ØTO24:B$=""                   :rem 1Ø6
29Ø FORC=ØTO39:A$=""                    :rem 97
3ØØ IFPEEK(SS+((R*4Ø)+C))>127THEN:GET#3,A
    $:B$=B$+CHR$(18)+A$+CHR$(146):GOTO33Ø
                                        :rem 59
31Ø GET#3,A$:IFA$=CHR$(13)THEN:A$=" "
                                       :rem 2Ø3
32Ø B$=B$+A$                            :rem 46
33Ø NEXTC:PRINT#4,B$:NEXTR:RETURN:rem 173
```

©

To receive additional information from advertisers in this issue, use the handy reader service cards in the back of the magazine.

VIC® 20 OWNERS

Fulfill the expansion needs of your computer with the

RAM-SLOT MACHINE

This versatile memory and slot expansion peripheral for the Commodore Vic-20 Computer consists of a plug-in cartridge with up to 24KBytes of low power CMOS RAM and 3 additional expansion slots for ROM, RAM and I/O. The cartridge also includes a reset button (eliminates using the power-on switch) and an auto start ROM selection switch.

#RSM-8K, 8K RAM + 3 slots....... **$ 84.50**
#RSM-16K, 16K RAM + 3 slots .... **$ 99.50**
#RSM-24K, 24K RAM + 3 slots **$119.50**

We accept checks, money order, Visa/Mastercard. Add $2.50 for shipping, an additional $2.50 for COD. Michigan residents add 4% sales tax. Personal checks—allow 10 days to clear.

® Trademark of Commodore.

K2 ELECTRONICS DESIGN CORPORATION
3990 Varsity Drive • Ann Arbor, MI 48104 • (313) 973-6266

Commodore 64 and VIC-20

SuperTerm

$149.95

Telecommunications with a difference!

Unexcelled communications power and compatibility, especially for professionals and serious computer users. Look us over; **SuperTerm** isn't just "another" terminal program. Like our famous Terminal-40, **it's the one others will be judged by.**

- **EMULATION**—Most popular terminal protocols: cursor addressing, clear, home, etc.
- **EDITING**—Full-screen editing of Receive Buffer
- **UP/DOWNLOAD FORMATS**—CBM, Xon-Xoff, ACK-NAK, CompuServe, etc.
- **FLEXIBILITY**—Select baud, duplex, parity, stopbits, etc. Even work off-line, then upload to system!
- **DISPLAY MODES**—40 column; 80/132 with side-scrolling
- **FUNCTION KEYS**—8 standard, 52 user-defined
- **BUFFERS**—Receive, Transmit, Program, and Screen
- **PRINTING**—Continuous printing with Smart ASCII interface and parallel printer; buffered printing otherwise
- **DISK SUPPORT**—Directory, Copy, Rename, Scratch

Options are selected by menus and EXEC file. Software on disk with special cartridge module. **Compatible with CBM and HES Automodems**; select ORIG/ANS mode, manual or autodial.

Write for the full story on SuperTerm; or, if you already want that difference, order today!

Requires: Commodore 64 or VIC-20, disk drive or Datasette, and compatible modem. VIC version requires 16K memory expansion. Please specify VIC or 64 when ordering.

Smart ASCII Plus . . . $59.95

The only interface which supports streaming —sending characters simultaneously to the screen and printer — with SuperTerm.

Also great for use with your own programs or most application programs, i.e., word processors. **Print modes:** CBM Graphics (w/many dot-addr printers), TRANSLATE, DaisyTRANSLATE, CBM/True ASCII, and PIPELINE.

Complete with printer cable and manual. On disk or cassette.

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

(816) 333-7200

Send for a free brochure.

MIDWEST MICRO inc.

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

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

Advanced Disk Logging On The 64

Jim Butterfield, Associate Editor

"Disk Log 64" provides you with access to some useful data in the disk directory which is not available via the usual BASIC commands.

There's information on disk that you can't get with an ordinary catalog sequence

```
LOAD "$",8
LIST
```

Often it doesn't matter. At other times, you find yourself wishing that you could get at the data.

"Disk Log 64" allows you to do that. It's set up specifically for the 64, and will give you a lot of information.

Programs

On program files, you'll get start and end addresses, in hexadecimal. That's the most convenient way to read them. You might use these addresses to help you make a copy of special overlay files that are often used in music and graphics. (See "Complex Disk Copies For The 64" in this issue.) Note that the "end" address is actually one location higher than the true end; that's the way they are most commonly used. So if you see that a program goes from \$0801 to \$1039, you know that the last byte is actually at \$1038; \$1039 is the first available memory location. Don't forget that these are hexadecimal numbers.

If you have a mixed disk of programs, you can often guess what type of systems the programs were written on. PET/CBM programs will start at \$0401; programs written on the minimum VIC will start at \$1001. A VIC with a 3K expansion will write programs that start at \$0401, like the PET; but if 8K or more is added, then programs will start at \$1201. You'll note that conventional 64 programs begin at \$0801, but there can be exceptions. Character sets, high-resolution graphics, music, and machine language programs can be loaded into unusual memory locations with a LOAD "NAME",8,1 type of command; and then it will be useful to know where such data blocks or programs are being used.

In normal circumstances, programs relocate when loaded into the 64 (or VIC, for that matter). If you use a conventional LOAD from disk, the start address doesn't matter too much: The program will be moved over to the start-of-BASIC location—normally \$0801.

Sequential And User Files

On this type of file, Disk Log 64 counts the bytes for you. No big deal, but sometimes this is more useful than the related coarse measure, block count. There are about 254 characters stored in each disk sector, but the last sector is often partially full. An exact character count can also help you refine file estimates.

A user (USR) file is identical to a sequential (SEQ) file, except that it is named as type USR when it is written. Why would a programmer bother? Usually to mark that there is something unusual about this file. For example, a file written as *packed binary* should probably be written as a USR file to warn programmers that they cannot list it in the usual way.

Relative Files

Relative files may be written on the 1541, and if they are found there, Disk Log 64 gives some very useful information which is not available elsewhere. First, it gives the file's *record length*—the standard size set for each record of this file. For example, L=45 would mean that each record is set to not more than 45 characters on the relative file. Following this, the program gives the number of records existing on the file.

Machine Language Aids

This program builds machine language aids into the cassette buffer. It then calls them in two different ways.

Relative file records are counted using the SYS command in line 740. The machine language program places the record count that it finds in locations 139 and 140, where BASIC picks it up.

Files which are not relative are measured for size using the USR command in line 760. The machine language program carefully places the answer—the number of characters in the file—into a series of locations called the Floating Point Accumulator. This value is given to BASIC the moment that the USR function returns. Note that the USR vector is set up in line 760 with POKEs to addresses 785 and 786: The Commodore 64 differs from all previous machines in this area, since PET, CBM, and VIC would use addresses 1 and 2 for the same purpose.

The program has been checked out with a standard 64 equipped with disk and printer. It should work on interfaces which give access to the IEEE bus, but not all combinations have been tested.

Disk Log 64

```
100 DIMT$(4)                          :rem 104
110 PRINT"DISK FILE LOG"              :rem 144
130 C$=CHR$(0)                        :rem 186
140 DATA169,0,162,4,149,98,202,16,251
                                      :rem 78
145 DATA169,160,133,97,162,2,32,198,255
                                      :rem 192
150 DATA230,101,208,10,230,100,208,6,230,
    99                                :rem 91
155 DATA208,2,230,98,32,228,255,165,144
                                      :rem 185
160 DATA240,235,32,204,255,198,97:rem 146
165 DATA6,101,38,100,38,99,38,98,16,244,9
    6                                 :rem 91
170 DATA169,0,133,139,133,140         :rem 190
180 DATA230,139,208,2,230,140         :rem 183
190 DATA162,15,32,201,255,169,80,32,210,2
    55                                :rem 111
200 DATA169,4,32,210,255,165,139,32,210,2
    55                                :rem 109
205 DATA165,140,32,210,255            :rem 38
210 DATA169,1,32,210,255,32,204,255
                                      :rem 221
215 DATA162,15,32,198,255,32,228,255
                                      :rem 37
220 DATA72,32,204,255,104,201,48,240,200,
    96                                :rem 101
230 FORJ=860TO977:READX:T=T+X:POKEJ,X:NEX
    TJ                                :rem 53
240 IFT<>16312THENSTOP                :rem 101
250 DATA"XXX","SEQ","PRG","USR","REL"
                                      :rem 108
260 FORJ=0TO4:READT$(J):NEXT          :rem 239
270 INPUT"PRINTER";Z$                 :rem 74
280 Z=3:IFASC(Z$)=89THENZ=4:INPUT"DATE";D
    $                                 :rem 74
290 U=8:REM UNIT 8                    :rem 251
300 D=0:REM DRIVE 0                   :rem 12
330 OPEN4,Z:OPEN1,U,15,"I"+CHR$(D+48):CLO
    SE1                               :rem 36
340 G$="{17 SPACES}"                  :rem 131
350 OPEN15,U,15                       :rem 67
360 OPEN1,U,3,"$"+CHR$(D+48)          :rem 200
370 GET#1,A$:A=ASC(A$+" ")            :rem 19
380 IFA=1ORA=65THENL1=141:L2=89:GOTO410
                                      :rem 80
390 IFA=67THENL1=3:L2=735:GOTO410:rem 206
400 CLOSE1:PRINT"???":STOP            :rem 131
410 PRINT#4,"*** DISK LOG ***{2 SPACES}";
    D$                                :rem 149
420 FORJ=1TOL1:GET#1,A$:NEXTJ         :rem 23
430 PRINT#4,"{2 SPACES}";:FORJ=1TO23:GET#
    1,A$:PRINT#4,A$;:NEXTJ            :rem 179
440 PRINT#4:FORJ=1TOL2:GET#1,A$:NEXTJ
                                      :rem 56
450 M=M+1:GET#1,K$,T$,S$              :rem 28
460 L7=-1:Z$=CHR$(160):F$="":FORJ=1TO16:G
    ET#1,A$                           :rem 157
470 IFA$=Z$THENL7=0                   :rem 105
480 IFL7THENF$=F$+A$                  :rem 126
490 NEXTJ                             :rem 38
500 GET#1,A$,A$,A$:L%=ASC(A$+C$)      :rem 131
510 FORJ=1TO6:GET#1,A$:NEXTJ          :rem 208
530 GET#1,A$:L=ASC(A$+C$)             :rem 63
550 GET#1,A$:L=L+256*ASC(A$+C$):IFM<8THEN
    GET#1,A$,A$:GOTO570               :rem 157
560 M=0                               :rem 85
570 SW=ST:IFK$=""GOTO820              :rem 182
580 K=ASC(K$)-128:IFK<1ORK>4THENK=0
                                      :rem 74
620 PRINT#4,T$(K)                     :rem 188
630 PRINT#4,RIGHT$(" "+STR$(L),3);" ";
                                      :rem 222
640 PRINT#4,LEFT$(F$+G$,17);          :rem 25
650 IFK=0GOTO810                      :rem 180
660 IFK=4THENPRINT#4,"L=";MID$(STR$(L%),2
    );                                :rem 245
670 OPEN2,U,4,CHR$(D+48)+":"+F$+","+T$(K)
                                      :rem 83
680 A=0:IFK<>2GOTO730                 :rem 223
690 GET#2,A$,B$:A=ASC(A$+C$)          :rem 206
700 B=ASC(B$+C$)                      :rem 54
710 GOSUB840                          :rem 180
730 IFK<>4GOTO760                     :rem 248
740 SYS915:A=PEEK(139)+PEEK(140)*256-1
                                      :rem 95
750 PRINT#4,",";MID$(STR$(A),2);"R";:GOTO
    800                               :rem 202
760 POKE785,92:POKE786,3:A=A+USR(0)
                                      :rem 19
770 IFK<>2THENPRINT#4,A;"BYTES";:GOTO800
                                      :rem 240
780 PRINT#4," ";:A%=A/256:A=A-A%*256:B=B+
    A%                                :rem 247
790 GOSUB840                          :rem 188
800 CLOSE2                            :rem 64
810 PRINT#4                           :rem 125
820 IFSW=0GOTO450                     :rem 18
830 INPUT#15,A:CLOSE1:PRINT#4,CHR$(13):CL
    OSE4:CLOSE15:END                  :rem 13
840 X=B/16:GOSUB850:X=A/16            :rem 6
850 FORJ=1TO2:X%=X:X=(X-X%)*16:IFX%>9THEN
    X%=X%+7                           :rem 149
860 PRINT#4,CHR$(X%+48);:NEXTJ:RETURN
                                      :rem 44
```

COMPLEX DISK COPIES For The 64

Jim Butterfield, Associate Editor

Copying unusual disk files is easier if you know what to look for and where to find it.

Conventional programs can be easily copied, even if you have only one disk drive. You just LOAD the program, insert another disk, and SAVE. But sometimes it doesn't work quite that easily.

For example, the 64 DOS Wedge seems to give trouble. The LOAD goes easily, but the SAVE seems to hang up. Many demonstration programs come in pieces—and one program, often labeled BOOT in the name, does the job of bringing in all the various chunks so the whole program can be staged.

If you have a dual disk, the job is easy. You give the COPY command, and the disk moves the program you have named from one drive to the other. You don't need to know where the program will be located when it loads—an identical copy will be made.

On a single drive, a backup program might be used. It copies everything and takes quite a while to run. Then you'd need to delete the stuff you didn't want, and you'd need to start with a fresh disk. But you could work it that way.

You could use a copying utility. With a pair of 1541 disks, you might run COPY-ALL64. With one disk, you'd need some other kind of program.

Or you could learn a little more about how programs are loaded into memory, and arrange to do your own custom copying.

Special Cases

The first thing you'd need to do is to spot the special programs which need to be copied with care. Often, the word BOOT in the name suggests that something is afoot (no pun intended). If you are instructed to LOAD in nonrelocatable format (LOAD "PROG",8,1), you may be sure that there's something nonstandard about the memory addresses involved. If, when you run a program, the disk starts to run, there's a good chance that the program is loading other "overlay" program segments. Or, if you make a copy with LOAD and SAVE and it doesn't work, it's a safe bet that there's something you need to look into.

I don't plan to talk about protection schemes here. If you have a piece of commercial software that won't copy, this article won't help. If you're unhappy about not being able to make a backup copy, write the software supplier, because I'm not going to help you on that problem.

First, you need to identify if there is one program or a group that needs special treatment. If there's a BOOT program, load it and list it. You'll see that this program's job is to load in the whole set of programs to do the job. For example, the program CHAR BOOT on the Commodore Disk Bonus Pack shows that three other programs are to be loaded in:

```
LOAD " ROTATE.DATA",8,1
LOAD " STANDARD.SET",8,1
LOAD " CHAR EDITOR",8
```

Note that the first two in the list are to be loaded without relocation (the extra ,1). CHAR BOOT and CHAR EDITOR are to be loaded in the normal, relocatable way.

Conclusion: We may copy CHAR BOOT and CHAR EDITOR with conventional LOAD and SAVE commands (be careful of the names). But we must use special techniques to copy ROTATE.DATA and STANDARD.SET.

Extracting Vital Statistics

Once you've found a program that needs special handling to copy, you must find out more about it.

The best approach is to use the program "Disk Log 64" (elsewhere in this issue). It will neatly give you the start and end addresses that you will need to make a good copy. The addresses are in hexadecimal. Don't worry about the alphabetic letters: Just note the details down carefully.

If for some reason you can't use Disk Log 64, you can try the following quick procedure. Enter this program:

```
100 INPUT "PROGRAM NAME";N$
110 OPEN 1,8,2,N$
120 GET#1,A$,B$:CLOSE 1
130 C$=CHR$(0)
140 A=ASC(A$+C$)/4096+ASC(B$+C$)/16
150 PRINT "PROGRAM START ADDRESS IS: "
160 FOR J=1 TO 4
170 A%=A:A=(A-A%)*16:IF A%>9 THEN
    A%=A%+7
180 PRINT CHR$(48+A%);:NEXT J
```

This will give you the start address, which is half of the story.

To get the end address, you must have a monitor loaded into your system (you'll need one later, anyway).

LOAD the program in question and go directly to the monitor (SYS 8 usually does the trick). Now inspect hexadecimal addresses 002D and 002E. You do this with the command ".M 002D 002E" which will give you a line starting with address $002D and containing eight two-digit hexadecimal numbers. Write down the first two numbers, but in reverse order. In other words, if the numbers are 33 4A, write down 4A33. That's the end address—one location past the end of the program.

But it's much easier to use Disk Log 64.

Making The Copy

Now you have the vital start and end addresses, the rest is easy. Be sure you have a machine language monitor in place.

Load the program in question, using the nonrelocate ",1" suffix (LOAD "NAME",8,1). Place the new disk into the drive, and be sure it's formatted. Go to the monitor. Save the program using the start and end addresses. The format for saving to disk on most monitors is:

```
.S '0:PROGNAME",08,1234,4A33
```

The 0: ahead of the program name specifies drive 0; this doesn't hurt, and sometimes it helps. Make sure you get the program name right. The disk device number, 8, must be typed in as two digits, 08. The start and end addresses should be typed in as shown.

That's all there is to it. Return to BASIC with the command ".X".

Wrinkles

Once in a while, you get a program that resides in exactly the same part of memory as the monitor. For example, Supermon64 normally takes up residence at addresses $97ED to $9FFF. If you are using this monitor and a program is loaded to the same memory area, you'll have trouble with the monitor. Change monitors or set up the monitor in a new place (Supermon can be moved by doing a little intelligent fiddling with addresses 55 and 56 [decimal] before running the Supermon builder).

You may run across programs on cassette tape that you'd like to transfer to disk using the same techniques. Not hard: LOAD the cassette tape program (use the extra ",1" again), and then PEEK addresses 829, 830, 831, and 832. Ask someone who knows about hexadecimal numbers to translate these decimal contents into hex for you; the first two give the start address, the last two give the end address. Or use the machine language monitor to display addresses $033D to $0340—it's the same place and you'll get the hex values right away. Again, reverse each pair of bytes. If the locations show as 24 68 25 69, then start is at $6824 and end is at $6925. If the program won't LOAD without automatically running, it's probably copy-protected and you shouldn't try to copy it anyway.

If you don't have a monitor, get one. Supermon64, for example, has been published in COMPUTE!, is on the Disk Bonus Pack, and is in many club libraries. Other monitors are available for sale. Even if you don't quite know how to use it yet, it's handy to have around.

We've looked at methods of spotting and copying difficult programs. And we've also had a chance to look a little further into the inner workings of the computer. ©

Atari Display List Interrupts

Karl E. Wiegers

This tutorial explains how to use the Atari Display List Interrupt to create sophisticated graphics displays, with multicolored screens and special character sets.

The Display List Interrupt (DLI) is one of the most powerful graphics features of the Atari 400/800/1200 computers. DLIs can be used to create sophisticated displays, with elaborate vertical screen architecture and the simultaneous display of many colors. DLIs also permit the display of multiple character sets in different parts of the screen, and they can also be used to create sound effects. Creative use of DLIs is the reason for the complexity of many Atari games.

The Display List Interrupt takes advantage of the way the television display operates. After the TV's electron beam scans horizontally across the width of the screen, it is turned off very briefly while it returns to the opposite edge and moves down slightly to draw the next scan line. During the few microseconds that the electron beam is off, it is possible to execute a very short machine language program, a Display List Interrupt routine. In this brief time, such routines can do little more than change the value stored in a color or sound register, but this can create some exciting effects.

DLI From BASIC

Atari computers control video output with a special microprocessor chip called ANTIC. ANTIC obtains its instructions from a small area in RAM containing the *display list*. The display list is simply a list of numbers which tells ANTIC what graphics mode to use for each of the 192 TV scan lines in the standard Atari display, where to find the information to be displayed, and so on.

As an example, the display list for graphics mode 0 contains 32 numbers, of which the seventh through twenty-ninth are instructions to display a mode 0 *mode line*, each of which consists of eight TV scan lines.

The display list instructions are not the same as the BASIC graphics mode numbers; for BASIC mode 0, the display list instruction is a "2". The display list is different for each graphics mode, and unique display lists can be designed to combine graphics modes in a single display (see "How to Design Custom Graphics Modes," *COMPUTE!'s First Book of Atari Graphics*). The address of the beginning of the display list in RAM is located by:

```
DL=PEEK(560)+256*PEEK(561)
```

Setting Up The Interrupt

To tell ANTIC that a Display List Interrupt is to be performed, several things must be done.

1. Select the vertical screen position where you want the interrupt to take place, for example, where you want the background color to change. Setting bit 7 of a number in the display list tells ANTIC to perform a DLI following the display of that mode line. Setting bit 7 can be done by simply adding 128 to the value in any memory location. Suppose you wish to have a DLI take place at line 10 of a graphics mode 0 display. You need to add 128 to the display list instruction for line 9, which happens to be in the memory location 13 bytes past the beginning of the display list:

```
POKE DL+13,PEEK(DL+13)+128
```

Do not add 128 to any number in the display list before DL+6, which corresponds to the second mode line.

2. Write your DLI routine and store it someplace safe in memory. The familiar READ/DATA/POKE combination works well for storing ML routines from within a BASIC program. I like to use Page 6 of RAM (locations 1536–1791) for the DLI, although other RAM locations or character strings can also be used.

3. Load the starting address of the DLI routine into locations 512 (low-byte) and 513 (high-byte). For a routine at the beginning of Page 6 this is accomplished with POKE 512,0 and POKE 513,6.

4. POKE 54286,192 to enable the DLI.

Writing The DLI Routine

There are several rules to be followed when writing the DLI routine.

First, it must be short. Only 14 to 61 machine cycles are available for execution, depending on the graphics mode, so keep your total routine down to about 25–30 cycles.

The first operation in your routine must be to save on the stack any of the 6502 registers (accumulator, X, and/or Y) which you use in the DLI. You must also restore these registers by pulling them off the stack at the end of the routine. In between, they can be used for executing the DLI instructions. Also, the final machine language instruction must be a ReTurn from Interrupt (RTI).

You will want to insure that any changes your DLI makes in the video display are properly synchronized with the TV electron gun's operation. A register is provided for this purpose at location 54282 ($D40A). Addressing this location, called WSYNC (Wait for horizontal SYNChronization), will provide a sharp separation between pre- and post-interrupt regions on the screen. So, early in your DLI routine, you should store some number in WSYNC (any number will do).

Use ROM For Changes

Another point is that changes in color registers, audio registers, and the like should be made directly into the hardware locations in ROM, rather than into the corresponding shadow locations in RAM. For example, the five playfield color registers are at locations 53270 through 53274 ($D016–$D01A) in ROM, and their shadows are at locations 708 through 712 in RAM. Your DLI should store values only in locations 53270–53274.

A Simple Color Change

Program 1, in BASIC, calls a DLI to change the screen background color from blue to red after six mode lines have been displayed. The POKE 512,0 and POKE 513,6 in line 70 load the DLI into the beginning of Page 6 of RAM. The DLI routine is stored as decimal numbers in the DATA statement. You can change the third number in this DATA statement to any number less than 256, to change the color displayed in the bottom portion of the screen. This color number is the same value you would POKE into location 710, the color register 2 shadow location, to change the background color. It is computed by taking 16* HUE + LUMINANCE. To return to the normal background from this modified display, simply type GRAPHICS 0.

A Two-Tone Text Window

Graphics modes 1 through 8 are really mixed mode displays, with a mode 0 text window of four lines at the bottom of the screen.

Program 2 shows a way to get three background colors on the screen. The default black for mode 2 is displayed at the top, green goes into the top two lines of the text window, and the bottom half of the text window is yet a different color (pink). In addition, the DLI sets color register 1, which controls the luminance of characters in mode 0, to zero, so the characters are now black.

These techniques will work with any custom-mixed mode display list, allowing almost unlimited variation in graphics displays.

All This And Sound, Too

DLIs need not be used only to brighten up your TV screen. Program 3 shows a way to get a repeating sound using a DLI and audio channel 1. (See *Mapping the Atari*[COMPUTE! Books] by Ian Chadwick, p. 121, for more information about directly addressing the audio registers.) In this program, the loop in line 50 sets the DLI instruction on several lines in the mode 0 display list; the exact number depends on the STEP size. A smaller step size gives a higher rate of sound repetition because the DLI is called more often.

The DLI routine in this example uses an additional RAM byte at location 1600 ($0640) to store a value which governs the pitch of the sound being generated. This location is initially set to 121 (middle C) by line 60.

The DLI decrements this value, then stores it in location 53760 ($D200) to produce a sound on audio channel 1. This value is then compared to a preset limiting pitch value, decimal 60 ($3C) in this example, corresponding to one octave above middle C. If not equal, then the routine terminates. If equal, then 121 ($79) is loaded back into location 1600 to restart the cycle the next time the DLI is called. This routine can be changed, as shown in the REM statements, to give increasing or decreasing pitch, fast or slow repeat rate, and any desired starting and ending pitches.

DLI For Special Characters

Many programs use specially designed character sets in place of the standard Atari characters. A DLI can be used to simultaneously display characters from more than one set on different lines of the screen.

This technique is used in some character set generation programs, such as "SuperFont" (COMPUTE!, January 1982). Program 4 shows how it works, without getting into the details of alternate character set creation (see *COMPUTE!'s First Book of Atari Graphics*, Chapter 3, for more information).

First, 1084 bytes are reserved for the redefined characters, and a DLI is set on line 10 of the graphics mode 0 display. The first five DATA statements contain redefined values of capital letters A, B, C, D, and E, which are loaded into the correct locations in the reserved RAM space. The DLI routine loads the most significant byte of the starting address of the new character set (156, $9C) into both WSYNC and location 54281 ($D409), the hardware character base address (RAM shadow is 756, which would be used by BASIC).

The routine also changes the color of the lower part of the screen. You will see both the regular ABCDE and the redefined ABCDE on the screen simultaneously.

Moving The Invisible Cursor

If you delete the last line in Program 4, you can move the invisible cursor and print in immediate mode in either the top (regular) or bottom (redefined) part of the screen. With this limited sample, only the letters A–E will be shown as nonblanks in the redefined screen area. However, this concept obviously allows great flexibility for creating elaborate screen displays with multiple character sets.

A Demonstration Of Multiple DLIs

So far, our examples have used one DLI routine per BASIC program. But you can also show several colors at once, or change both character sets and colors in separate operations. Obviously, multiple DLIs are needed in such cases. A problem arises, because you can only tell the operating system about one DLI at a time using locations 512 and 513.

There are several possible solutions to this problem. Perhaps the simplest is to load several DLIs into memory, and have each routine load the starting address of the next one to be called into locations 512 and 513. This chaining of DLIs is illustrated in Program 5.

This program will place five bands of color in a mode 1 screen, together with different-colored characters. The same basic DLI is used four times, stored 32 bytes apart in Page 6. The only changes are to the actual color value stored in the background color register 53274 ($D01A) and to the low byte of the Page 6 address of the next DLI to use, which gets stored in location 512 ($0200).

Avoiding Screen Flicker

DLI instructions are placed on four mode lines (line 40). The BASIC program contains a POKE 559,0 in line 30 and a POKE 559,34 in line 230. These statements simply turn off the TV display briefly while the DLI routines are stored and changed, to avoid screen flicker. You can remove these statements and watch the action, if you like.

By experimenting with the BASIC programs and DLI routines in these five programs, you can better understand the principles involved and the ease of using DLIs in your own programs. The book *Mapping the Atari* contains the most detailed available memory map of the Atari, and is an invaluable reference for the programmer wishing to use DLIs effectively.

Program 1: Color Change Using DLI

```
OC 1Ø GRAPHICS Ø
JI 2Ø DL=PEEK(56Ø)+PEEK(561)*256:REM F
      ind start of display list
MI 3Ø POKE DL+1Ø,PEEK(DL+1Ø)+128:REM E
      nable DLI for desired line in di
      splay list
JA 4Ø FOR I=Ø TO 1Ø:READ A:POKE 1536+I
      ,A:NEXT I:REM Poke DLI routine i
      nto page 6
PM 5Ø REM Change 68 to desired color v
      alue
NA 6Ø DATA 72,169,68,141,1Ø,212,141,24
      ,2Ø8,1Ø4,64
HO 7Ø POKE 512,Ø:POKE 513,6:REM Set ad
      dress for DLI (LSB,MSB)
KG 8Ø POKE 54286,192:REM Enable DLI
KF 9Ø LIST
```

Program 2: Two-Tone Text Window

```
OE 1Ø GRAPHICS 2
JI 2Ø DL=PEEK(56Ø)+256*PEEK(561):REM F
      ind start of display list
OM 3Ø POKE 71Ø,198:REM Set color regis
      ter 2 to green
KC 4Ø POKE DL+18,13Ø:REM set DLI for m
      iddle of text window
JK 5Ø FOR I=Ø TO 19:READ A:POKE 1536+I
      ,A:NEXT I:REM Poke DLI routine i
      nto page 6
AN 6Ø REM Change 88 to desired color v
      alue for bottom of text window
NA 7Ø DATA 72,138,72,169,Ø,162,88
HP 8Ø DATA 141,1Ø,212,141,23,2Ø8
EO 9Ø DATA 142,24,2Ø8,1Ø4,17Ø,1Ø4,64
GI 1ØØ POKE 512,Ø:POKE 513,6:REM Set a
       ddress for DLI
NA 11Ø POKE 54286,192:REM Enable DLI
MI 12Ø PRINT #6;" GRAPHICS mode two"
GF 13Ø PRINT "Top half of text window"
PG 14Ø PRINT :PRINT "Another color for
        the bottom half!"
CH 15Ø PRINT "(Press any key to go on)";
NK 16Ø OPEN #1,4,Ø,"K:":GET #1,A:GRAPH
       ICS Ø:LIST
```

Program 3: Creating Sound With DLI

```
OC 1Ø GRAPHICS Ø
JI 2Ø DL=PEEK(56Ø)+PEEK(561)*256:REM F
      ind start of display list
KM 3Ø POKE 53768,Ø:POKE 53775,3:REM Eq
      uivalent of SOUND Ø,Ø,Ø,Ø
PA 4Ø POKE 53761,168:REM Set audio cha
      nnel 1 volume=8, distortion=1Ø
OI 5Ø FOR I=6 TO 23 STEP 3:POKE DL+I,1
      3Ø:NEXT I:REM Change step to cha
      nge speed
NJ 6Ø POKE 16ØØ,121:REM Change 121 to
      desired initial pitch
JE 7Ø FOR I=Ø TO 2Ø:READ A:POKE 1536+I
      ,A:NEXT I:REM Poke DLI routine i
      nto page 6
BE 8Ø REM Change 2Ø6 to 238 to decreas
      e pitch from intial setting
EJ 9Ø REM Change 6Ø to desired ending
      pitch
KN 1ØØ DATA 72,2Ø6,64,6,173,64,6,141,Ø
AF 11Ø DATA 21Ø,2Ø1,6Ø,2Ø8,5,169,121,1
       41
NA 12Ø DATA 64,6,1Ø4,64
```

```
KL 130 POKE 512,0:POKE 513,6:REM Set address for DLI (LSB,MSB)
ND 140 POKE 54286,192:REM Enable DLI
JL 150 PRINT "PRESS ANY KEY TO GO ON"
LE 160 OPEN #1,4,0,"K:":GET #1,A:CLOSE #1:SOUND 0,0,0,0
NE 170 LIST
```

Program 4: Special Characters Using DLI

```
EG 10 CHRBASE=256*(PEEK(106)-4):REM Reserve 4 pages below RAMTOP
PD 20 POKE 106,156:REM Set new RAMTOP (156 is for 48K machine)
HP 30 GRAPHICS 0:REM Put new RAMTOP into effect
JE 35 POKE 710,198:REM Change background color to green
BB 40 POKE 752,1:REM Inhibit cursor
JL 50 DL=PEEK(560)+256*PEEK(561):REM Find start of display list
GA 60 POKE DL+14,PEEK(DL+14)+128:REM Set DLI
PH 70 FOR I=1 TO 40:REM Load bit maps for redefined characters (A,B,C,D,E)
JA 80 READ A:POKE CHRBASE+264+I,A:NEXT I
MH 90 DATA 60,102,198,198,198,102,59,0
HB 100 DATA 62,99,99,99,126,99,126,0
AP 110 DATA 60,99,96,96,96,99,62,0
GJ 120 DATA 60,99,99,99,99,102,124,0
FO 130 DATA 60,98,96,124,96,127,60,0
LB 140 FOR I=0 TO 15:READ A:POKE 1536+I,A:NEXT I:REM Load DLI into page 6
KL 150 DATA 72,169,156,141,10,212,141,9,212,169,148,141,24,208,104,64
JG 160 POKE 512,0:POKE 513,6:POKE 54286,192:REM Enable DLI
AD 170 POSITION 15,4:? "ABCDE":REM Normal characters at top of screen
GN 180 POSITION 15,12:? "ABCDE":REM Redefined characters below DLI line
DL 190 POSITION 7,8:? "PRESS ANY KEY TO GO ON"
JM 200 OPEN #1,4,0,"K:":GET #1,A:CLOSE #1:POKE 106,160:GRAPHICS 0:LIST
```

Program 5: Multiple DLI Routines

```
KE 10 GRAPHICS 17:REM Mode 1 screen will have 5 separate colored sections
JI 20 DL=PEEK(560)+PEEK(561)*256:REM Find start of display list
OJ 30 POKE 559,0:REM Turn off video display while loading DLI routines
DB 40 POKE DL+9,134:POKE DL+14,134:POKE DL+20,134:POKE DL+25,134
AF 50 REM Set DLI instruction on all desired mode lines (134=128+6 for mode 1)
HP 60 FOR I=0 TO 20:READ A:POKE 1536+I,A:NEXT I:REM First DLI routine
DE 70 REM 68 means red:32 sets address for start of next DLI routine
FI 80 DATA 72,169,68,141,10,212,141,26,208,169,32,141,0,2,169,6,141,1,2,104,64
PF 90 RESTORE 80
AD 100 FOR I=0 TO 20:READ A:POKE 1568+I,A:NEXT I:REM Second DLI routine
ND 110 POKE 1570,198:REM Change color to green
MI 120 POKE 1578,64:REM Change starting address for next DLI routine
CA 130 RESTORE 80
JJ 140 FOR I=0 TO 20:READ A:POKE 1600+I,A:NEXT I:REM Third DLI routine
AE 150 POKE 1602,102:REM Change color to lavender
ME 160 POKE 1610,96:REM Change starting address for next DLI routine
CE 170 RESTORE 80
BP 180 FOR I=0 TO 20:READ A:POKE 1632+I,A:NEXT I:REM Fourth DLI routine
CP 190 POKE 1634,28:REM Change color to yellow
BL 200 POKE 1642,0:REM Change starting address back to first DLI routine
LP 210 POKE 512,0:POKE 513,6:REM Set address back to first DLI
NC 220 POKE 54286,192:REM Enable DLI
DM 230 POKE 559,34:REM Turn video display back on
MP 240 POSITION 1,2:? #6;"look at all the"
LA 250 POSITION 1,7:? #6;"COLORS YOU CAN MAKE"
NM 260 POSITION 1,12:? #6;"in mode one"
HH 270 POSITION 1,18:? #6;"PRESS ANY KEY NOW"
DG 280 POSITION 1,21:? #6;"TO GO ON"
CH 290 OPEN #1,4,0,"K:":GET #1,A:CLOSE #1:LIST
```
©

TI Tricks And Tips

Michael A. Covington

Here are 13 ways to get more out of your TI-99/4A and Extended BASIC.

Here is a collection of handy hints for TI-99/4A programmers.

1. You can get white characters on a black screen, for the duration of a program run, by executing statements such as:

```
10 FOR J = 1 TO 14
20 CALL COLOR(J,16,1)
30 NEXT I
40 CALL SCREEN(2)
```

The results look rather good on a color TV, but bad on a black-and-white set, because scan lines break each letter up into separate dots. (The most readable black-and-white display is obtained by executing a CALL SCREEN(15), making the screen gray while leaving the characters black.)

2. In both TI BASIC and Extended BASIC, you can use * for the logical operator AND and + for OR. For instance, the statement IF (X=0)+(Y=0) THEN 1500 means "if X = 0 or Y = 0 then go to line 1500." The parentheses are essential to show that you don't want to add 0 to Y. Extended BASIC allows you to use the alternative notation IF X=0 OR Y=0 THEN 1500.

3. In Extended BASIC, but not in TI BASIC, pressing any key while a program is being LISTed temporarily halts the listing; pressing any key then causes the listing to resume. In each case the key must be held down for half a second or so in order to get any response.

4. If you RESEQUENCE a program that contains references to nonexistent lines, those references will be changed to references to line 32767. For instance, if you have a GO TO 500 and there isn't a line 500, a RES command will change that statement to GO TO 32767.

5. In Extended BASIC, the command RUN "CS1" loads a program from the cassette drive and immediately runs it. It is equivalent to OLD CS1 followed by RUN. With a disk drive and Extended BASIC, you can use RUN "DSK1. filename", where *filename* is the name of the program on disk that you want to LOAD and RUN.

6. TI BASIC gives you 608 more bytes of available memory than Extended BASIC. However, you can usually write your program more compactly in Extended BASIC, so the difference is of little practical consequence.

7. Built-in subprograms that require integer arguments, such as CALL HCHAR, CALL VCHAR, CALL SOUND, and the CHR$ function, will in fact accept numbers that are not integers. The argument is rounded to the nearest integer before being used, so that for instance CHR$(10.8) is the same as CHR$(11). CHR$(10.4) would be equivalent to CHR$(10).

8. In TI BASIC, you can include multiple colons (for example, :::::) in PRINT statements to produce multiple line skips. A TI BASIC program using this feature which is loaded from disk or cassette under Extended BASIC will run correctly, but you cannot type multiple colons while in Extended BASIC unless you want them to be taken as statement separators (::). Put spaces between the colons, as in PRINT A : : B rather than PRINT A :: B, and they will work correctly.

9. In Extended BASIC, you cannot have more than four sprites visible on the same line at the same time; additional sprites will be temporarily invisible. The problem is worse with double-size sprites (CALL MAGNIFY(3) or (4)), since then only part of the sprite generally disappears, distorting its appearance.

10. When you execute a CALL SPRITE statement, the sprite will sometimes momentarily pop into existence at a random screen location and then jump to the location that you specified. To prevent this, create the sprite with a color of 1 (transparent) and then alter its color with a CALL COLOR statement.

11. The loss of resolution on the screen that occurs with certain color combinations is inherent in the way color is encoded onto the video signal and does not represent a defect in the TV set or modulator. For greatest sharpness, use black on gray or cyan.

12. The TI-99 sound generator will produce frequencies from 110 to 44733 hertz (cycles per second), well above the limit of human hearing. However, the response of the sound section of most TV sets falls off markedly above 2000 (or, at best, above 10,000) hertz. This means that you cannot, as is sometimes suggested, use the TI-99 to test the upper frequency limit of your hearing. It also means that tones above 2000 Hz—still well within the range of human hearing—will sound markedly different on different TV sets.

13. If you want to transmit lines of more than 80 characters to the printer, open it as "RS232.CR" (or "PIO.CR") rather than "RS232" (or "PIO"). You must still end your line of output (by executing a PRINT statement that does not end in a comma or semicolon) before more than 80 characters have been transmitted, but doing so will not cause the printer to start a new line; the printer will stay on the same line until you explicitly transmit a carriage return, CHR$(13), and a line feed, CHR$(10).

This is particularly useful when you are using a dot-matrix printer in graphics mode, using each character code for a single vertical row of dots and putting hundreds of them on a line. ©

Commodore

Disk Datamaker

A confusing and tedious aspect of machine language programming, sprite making, or character set redefinition is converting the contents of memory into BASIC DATA statements that can be typed in easily. This program—for any Commodore computer with a disk drive—will do all the work for you. It also provides valuable insight into how the computer stores program files on disk.

In the past there have been programs that created DATA statements by PEEKing the contents of memory and using the dynamic keyboard technique to fool the computer into thinking that someone had typed in the lines. The problem with these programs was that occasionally the resulting DATA statements, or perhaps even the datamaker program itself, would need to occupy the area of memory being PEEKed, destroying the data that needed to be transformed.

If, however, the data is stored on disk, it would not need to be in memory as well. It could just be read from a file. Then I decided there was no reason to use the dynamic keyboard technique either, since a BASIC program could be created right on the disk. The result is "Disk Datamaker" (Program 1); it causes no memory conflicts—because neither the original chunk of data, nor the BASIC program created by Datamaker, is ever in the computer's memory.

In addition to creating DATA statements, the Datamaker program creates the necessary lines of BASIC to READ and POKE the DATA items back into memory. Program 2 is a short example of a program created by the Datamaker. It consists of the DATA statements for one sprite shaped like a Commodore 64.

Applications

Suppose you have written a machine language (ML) program. Sophisticated assembler programs such as MAE, the Commodore assembler, and the PAL assembler all allow you to store the object code on disk. If you use a simple assembler like Supermon or Micromon, you can SAVE a copy of the section of memory that contains your ML program. Then all you have to do is LOAD the Disk Datamaker program, give it the name of your object and program files, and it will take care of everything.

Suppose you have just created sprite data with a sprite editor. If you have the Commodore Sprite Editor, all you have to do is press the S key to invoke the SAVE command, which puts your sprite data on disk. If you are using some other editor, you could use a monitor to SAVE a copy of the section of memory containing the sprite data. Disk Datamaker could then turn that file into DATA statements.

Manipulating PRG Files

All this is possible because PRG (program) files on disk can be OPENed and used just like SEQ (sequential) data files. To OPEN a PRG file for writing, you can either put a ",P,W" after the filename, or you can OPEN with a secondary address of 1, which is reserved for SAVE. These two lines produce identical results:

```
OPEN 8,8,1,"filename"
OPEN 8,8,8,"filename,P,W"
```

To OPEN a PRG file for reading, just OPEN another ",P" file or use a secondary address of 0. These two lines are also identical:

```
OPEN 8,8,0,"filename"
OPEN 8,8,8,"filename,P,R"
```

Using this knowledge, it is possible to read the PRG file that holds the data, and to write a PRG file that contains a BASIC program.

BASIC Basics

The other knowledge necessary to understand Disk Datamaker is the structure of a BASIC program. Each line of BASIC contains four header bytes. The first two are a pointer to the beginning of the next BASIC line in memory. The address is in standard low-byte/high-byte format. The next two bytes are the line number, also in low-byte/high-byte format. Next is the tokenized version of the BASIC line, followed by a zero to denote the end of the line in memory. If the pointer to the next line is two zeros in a row, the computer knows it has reached the end of the program.

Program Operation

Lines 10–90 OPEN the PRG file from which the data will come and the one to which the BASIC program will be written. The subroutine at 10000 will print the appropriate error message if either of the files cannot be OPENed.

Line 100 GETs the first two bytes from the file containing the data. In Commodore PRG files, these bytes always contain the starting address (again in low-byte/high-byte format) of the area of memory where the data was located when it was originally saved. Line 110 writes the first two bytes to the file which will be the new BASIC program. The numbers chosen, 1 and 4, specify a starting address of 1 + (4*256) or 1025. This starting address will allow the BASIC program created to LOAD correctly into a PET/CBM. The starting address is not critical for the VIC and 64 since they automatically relocate any program to their particular starting addresses when it is LOADed.

Line 120 defines the values for the pointer bytes to the next program line. Rather than attempting to calculate the proper addresses for each pointer, Datamaker arbitrarily sets all these bytes to ones. This can be done because Commodore computers automatically redo these pointers whenever a BASIC program is LOADed. Line 130 writes the pointer for the first BASIC line to the disk, and line 140 writes the first four numbers from the DATA statement in line 170. These provide the line number (10) and the tokens for I=. Line 150 uses the value for the starting address found in line 100 to write the value for I to the disk. If you want to relocate your data, simply change the value for I in line 10 of the BASIC loader program once it is created.

Line 160 reads the rest of the DATA from lines 170–200 of Program 1 and writes it to the disk, which creates lines 20 and 30 of the new BASIC program (see Program 2, for example). These lines READ the DATA statements created from your data and POKE the values back into memory.

Line 210 calculates the line number to be used for the first DATA statement. This is the same as the address in memory of the first DATA item. Note this feature of Datamaker: The line number of each DATA line is equal to the address into which the first number in that line will be POKEd, unless you change the value of I to relocate the data. Line 220 writes out the pointer bytes and the line number for each DATA line, and line 230 writes the tokens for DATA and a space.

Lines 250–300 constitute a loop to read eight bytes from the data file and write them as eight DATA items, separated by commas, to the BASIC program file. Line 310 checks the variable S, set in line 250 to the value of the built-in status variable ST, to detect whether the end of the data file on disk has been reached. (See Larry Isaacs' "64 Explorer" column in the October and November 1983 issues of COMPUTE! for more information on detecting an end-of-file with the ST variable.)

If the end has not been reached, line 320 calculates the next DATA line number, and line 330 writes out a zero to mark the end of the current BASIC line. If the end has been reached, line 340 adds a DATA item with the value of 256 to the end of the DATA statements. This is the value checked for as an end-of-data marker in line 20 of the created BASIC program. The three zeros in a row written by line 340 mark the end of the BASIC program. Line 350 CLOSEs the two PRG files and logical file 15, used to detect disk errors, for a clean exit from the program.

Program 1: Disk Datamaker

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

```
10 PRINT"{CLR}MAKE BASIC LOADER OUT OF"
                                :rem 137
20 PRINT"DISK OBJECT FILE.{3 DOWN}"
                                :rem 150
30 INPUT"{DOWN}ENTER FILENAME";FL$
                                :rem 248
40 OPEN15,8,15                  :rem 242
50 OPEN8,8,8,FL$+",P,R"         :rem 226
60 GOSUB 370                    :rem 128
70 INPUT"FILENAME TO CREATE";FC$ :rem 187
80 OPEN9,8,9,"0:"+FC$+",P,W"    :rem 188
90 GOSUB 370                    :rem 131
100 GET#8,LO$,HI$               :rem 152
110 PRINT#9,CHR$(1)+CHR$(4);     :rem 22
120 LINK$=CHR$(1)+CHR$(1)        :rem 83
130 PRINT#9,LINK$;               :rem 54
140 FOR I=1 TO 4:READ BYTE:PRINT#9,CHR$(B
    YTE);:NEXT I                :rem 102
150 PRINT#9,MID$(STR$(ASC(LO$+CHR$(0))+AS
    C(HI$+CHR$(0))*256),2);      :rem 17
160 FOR I=1 TO42:READ BYTE:PRINT#9,CHR$(B
    YTE);:NEXT I                :rem 154
170 DATA 10,0,73,178,0,31,4,20  :rem 171
180 DATA 0,135,32,65,58,139      :rem 50
190 DATA 32,65,178,50,53,54,32,167,32,128
                                :rem 233
200 DATA 0,52,4,30,0,151,32,73,44,65,58,7
    3,178,73,170,49,58,137,32,50,48,0
                                 :rem 51
```

```
210 LO=ASC(LO$+CHR$(Ø)):HI=ASC(HI$+CHR$(Ø
    ))                               :rem 145
220 PRINT#9,LINK$+CHR$(LO)+CHR$(HI);
                                     :rem 92
230 PRINT#9,CHR$(131)CHR$(32);      :rem 131
240 COUNT=1                         :rem 141
250 GET#8,BYTE$:S=ST                :rem 199
260 PRINT#9,MID$(STR$(ASC(BYTE$+CHR$(Ø)))
    ,2);                             :rem 48
270 COUNT=COUNT+1                    :rem 68
280 IF COUNT=9 OR S<>Ø THEN 310     :rem 137
290 PRINT#9,",";                     :rem 91
300 GOTO 250                         :rem 99
310 IF S THEN 340                    :rem 60
320 LO=LO+8:IF LO>255 THEN LO=LO-256:HI=H
    I+1                              :rem 10
330 PRINT#9,CHR$(Ø);:GOTO220        :rem 111
340 PRINT#9,",256"+CHR$(Ø)+CHR$(Ø)+CHR$(Ø
    );                              :rem 251
350 CLOSE8:CLOSE9:CLOSE15            :rem 69
360 END                             :rem 112
370 INPUT#15,E1,E2$,E3,E4           :rem 101
380 IF E1=Ø THEN RETURN              :rem 28
390 PRINT"ERROR - "E1;E2$;E3;E4     :rem 215
400 CLOSE8:CLOSE9:CLOSE15            :rem 65
```

Program 2:
Sample Program Created By Datamaker

```
10 I=12864
20 READ A:IF A=256 THEN END
30 POKE I,A:I=I+1:GOTO 20
12864 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø
12871 DATA Ø,Ø,Ø,Ø,Ø,15,255
12878 DATA 255,23,224,49,16,Ø,1
12885 DATA 35,102,51,102,219,51,75
12892 DATA 108,103,199,254,102,128,Ø
12899 DATA 14,255,255,252,255,255,248
12906 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø
12913 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø
12920 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø
12927 DATA Ø,165,256
```

©

To receive additional information from advertisers in this issue, use the handy reader service cards in the back of the magazine.

Super Directory

Michael Contino

"Super Directory" is an invaluable utility which displays the disk directory on the screen and allows you to use the cursor control keys to automatically select, LOAD, and RUN any program. Originally written for 64 and PETs with Upgrade and 4.0 ROM; versions are also included for VIC, Atari, and IBM PC/PCjr.

As much as we might not want to admit it, there are still many people who have had little or no experience with computers. And many of them still harbor a fear of the machines. In writing programs, most of us keep this in mind and attempt to make the programs as friendly as possible. Of course, there is still the problem of getting the program loaded and running.

Since menus are the standard solution for friendly programs, it is natural to write a menu program that will: present the user with the choice of programs; allow for the selection of a program, usually by number; and then load the program. Problems with this approach include the updating of disks every so often and typing all those names into DATA lines.

"Super Directory" solves these problems, providing menu selection using the cursor controls, with an automatic LOAD and RUN of the selected program.

I developed the program on my CBM 8032 with an 8050 disk. However, it also works on PETs with Upgrade and 4.0 BASIC, with any of the three Commodore disk formats, and with the 64.

This program must be the first program on the disk. If you don't know how to do that, simply COPY the first file or program on the disk back to the same disk under another name, and then save this program. Don't forget to then rename the copied file back to its correct name (or do another copy).

Once the program is properly located, a simple ‹shift RUN› (on the PET) will load and execute the program. If you do not have the 4.0 PET, you will have to LOAD "*",8 and then type RUN for other Commodore machines. The program will read through the entire disk and store all PRG files into an array, skipping all nonprogram files. It then presents a menu in two columns (four on the 8032) of all programs on that disk. The first option is "next page," in case there are more programs than can fit on one page. If this choice is continually taken until there are no more selections, you're offered a choice to go back to the beginning, to access a different disk, or to end the process.

Selection By Cursor

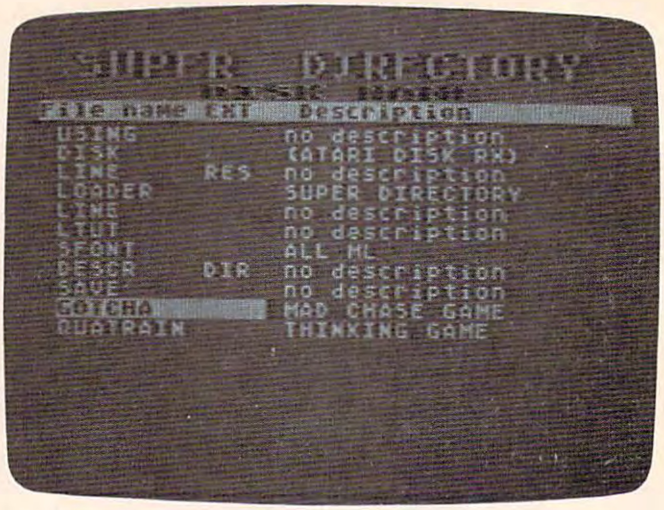
It is in selecting your choice that this menu program is different from most. I have often seen students hesitate for a long time in converting their choice to a number, pressing the correct number, and then pressing RETURN. I have also seen the opposite problem: people moving quickly, watching the screen, and hitting the wrong number. To correct for this in another program, I developed a subroutine which allows for menu selection by control of the cursor keys. When the menu appears, one default choice is highlighted (in RVS reverse print). Pressing the cursor control keys causes the highlight to move up or down, right or left, through the list. I added a wraparound, so that a cursor down from the last item in a column will send the highlight to the top of that column. This provides an almost foolproof method of input. To make it just about perfect you might want to also disable the STOP key.

I like to refer to this subroutine as a light pen—without the pen. I have used it in many programs, and I encourage you to excerpt the subroutine for use in your programs. After we have looked at the program in some detail, I will explain the initialization steps needed before calling the routine.

Simulated WHILE Loop

Before we look at the details of the program, a word about another interesting feature you may want to use in your programs. In structured programming, WHILE and UNTIL loops are considered very nice. Unfortunately, the FOR/NEXT loop does not quite fill the gap in BASIC. It controls a loop where a variable is counting for a specified number of executions, but it does not work as

In this 64 directory, the cursor is on "TINDCAD.VIC". Pressing RETURN will load and run that program.



Atari "Super Directory" allows you to store descriptions along with the program names.

well when some other condition is meant to control the repetition. We usually settle this problem by adding a line such as: IF condition THEN GOTO line nn. That works, but can lead to an unclosed FOR/NEXT loop, which brings on its own problems. You can simulate the WHILE structure by a line such as

```
300 FOR I=0 TO 1 STEP 0
```

which will count by zeros for a very long time, until it reaches one. The real way out is tested for in the body of the loop. When discovered, simply set I equal to 1. For example,

```
360 IF Z=13 THEN I=1: REM CARRIAGE RETURN
```

Getting Around GET

As the program was originally written, the GET statement was used to retrieve the characters of the program names one at a time from the directory sectors of the disk. This made the program rather slow. In a worst case, a Commodore 8050 disk drive containing the maximum 224 programs for that format took almost two minutes to read the directory. I tried writing RETURN characters, CHR$(13), into the directory so that I could use the faster INPUT statement instead. However, this caused funny-looking directories and created problems where there were 13's in the directory that were not RETURNs (such as references to track 13 or sector 13).

I considered storing the directory entries in a sequential file on the disk, and then having my program read the file instead of the directory sectors. This approach would have required an additional program to create the sequential file and to update the file whenever a new program was saved on the disk. In addition to these complications, it seemed unappealing to use up disk space with a file containing information that was already on the disk.

The only other approach in BASIC was to read in the directory sectors and create DATA statements to be added to the Super Directory program, but this again required a second program to generate the DATA statements and update them as new programs were saved to the disk.

The only way to achieve a truly significant increase in speed was to read the directory information from the disk with machine language. Fortunately, I did not have to write the machine language program to do this. Jim Butterfield's "String Thing" (COMPUTE!, November 1982) is a machine language (ML) routine which reads information from the disk into a variable. It functions like a very fast INPUT#. I modified the routine to read the entire contents of a disk sector into a string variable (lines 6000–6100). Lines 6000–6010 create a 254 character-long variable called IN$. The ML routine puts data into the first variable in the program, so it's important that IN$ be set up before any other variables are mentioned. Lines 6020–6030 let the program know whether it's operating in a 64 or a PET/CBM, then lines 6040–6060 load the DATA for the modified String Thing, adjusting it as necessary for the particular computer.

The increase in speed using this approach is dramatic. With a typical PET disk, the time required to load the directory was cut from about 45 seconds to only 8 seconds. The worst case (full 8050 disk) time dropped from 2 minutes to less than 15 seconds. Loading times are slightly longer for the 1541 disk used with the 64, due to its serial communications, but the increase in speed provided by the machine language is even more significant.

On To The Directory

First a quick run-through of the program blocks. The first line jumps to line 1000. This is designed as a time-saver, allowing room at the top for frequently used subroutines. It also mimics the Pascal requirement that all routines and procedures be defined before the body of the program. The most common routine, lines 1 and 2, is the get-character routine. Disk error check, lines 10–20, is next and uses the error channel, not DS$, in order to be compatible with earlier versions.

The main video selector (or penless light) starts at line 100. It in turn calls supportive routines at line 500 and line 600. Before this can be used, the clear display routine at line 900 is called by the main program.

Two other routines are called only once and they have been placed after the program body, to help with readability. The routine at line 2000 reads the disk header and determines the type of disk being used. It then goes to the beginning of the directory and reads the program names into an array. If you have an 8050 disk drive, you'll need to replace the following lines:

```
2Ø4Ø FO=ASC(IN$):IF (FO AND 3)=3 THEN 21Ø
     Ø
21ØØ HE$=MID$(IN$,5,16)+", 8Ø5Ø FORMAT"
211Ø SYS 896:SYS 896
212Ø SE=28
```

Line 2900 allows you to define the disk drive to be used. This defaults to drive 0, unit 8. If you delete line 2910, lines 2920–2940 will then allow you to select the device and drive number each time the program is run.

The program has two possible outcomes, and thus concludes in the sections beginning at either line 3000 or line 5000. If the search was unsuccessful, lines 3000–3050 give the options of restarting the current disk (the directory does not have to be read again from disk, as the array with program names has not been disturbed); of starting over with a new disk; or of quitting the search. A successful search takes us to line 5000, which loads and runs the chosen program via the dynamic keyboard method.

If you don't want to type in the program, I'll make copies (PET/CBM/64 version only). Send $3, a disk (8050 or 4040/1541 format), and a stamped, self-addressed mailer to:

Michael A. Contino
Dept. of Mathematics & Computer Science
Cal State University, Hayward
Hayward, CA 94542

Program 1: Super Directory For PET/64

64 users should refer to the "Automatic Proofreader" article before typing this program in. PET/CBM owners should ignore the :rem at the end of each line.

```
Ø GOTO 1ØØØ                              :rem 42
1 GETZ$:IF Z$="" THEN 1                 :rem 183
2 RETURN                                 :rem 18
1Ø INPUT#15,ER,ER$:IF ER=Ø THEN RETURN
                                          :rem Ø
2Ø PRINT"DISK ERROR #"ER:PRINT ER$:END:RE
   TURN                                  :rem 79
1ØØ PRINT"{HOME}{3 DOWN}":FOR I=Ø TO 15:F
    OR J=Ø TO NC:PRINTTAB(J*2Ø);V$(I+J*16
    );                                  :rem 233
11Ø NEXT J:PRINT:NEXT I                 :rem 164
12Ø POKE XB,Ø:RL=Ø:CL=Ø:GOSUB 6ØØ:rem 131
13Ø FOR I=Ø TO 1 STEP Ø:GOSUB 1:Z=ASC(Z$)
    :TC=CL:TR=RL:FOR J=Ø TO 1 STEP Ø
                                         :rem 67
14Ø IF(ZAND127)=29 THEN CL=(CL+1+2*(Z=157
    ))AND NC                            :rem 139
15Ø IF(ZAND127)=17 THEN RL=(RL+1+2*(Z=145
    ))AND 15                            :rem 121
16Ø IF Z=13 THEN I=1                      :rem 7
17Ø IF V$(RL+CL*16)<>B$ THEN J=1 :rem 162
18Ø NEXT:IF(TC<>CL)OR(TR<>RL) THEN GOSUB
    {SPACE}5ØØ                          :rem 134
19Ø NEXT:RETURN                         :rem 243
5ØØ POKE XA,TR+3:PRINT:PRINTTAB(TC*2Ø);V$
    (TR+16*TC)                          :rem 1Ø1
6ØØ POKE XA,RL+3:PRINT:PRINTTAB(CL*2Ø);"
    {RVS}";V$(RL+16*CL):RETURN    :rem 241
9ØØ FOR I=1 TO NV:V$(I)=B$:NEXT:RETURN
                                          :rem 2
1ØØØ PRINT"{2 HOME}{CLR}":GOSUB 6ØØØ
                                        :rem 2Ø3
1Ø1Ø NC=1:IF PEEK(213)=79 THEN NC=3
                                        :rem 115
1Ø2Ø NV=(NC+1)*16-1:GOSUB 2ØØØ:B$="
     {19 SPACES}":DIMV$(NV)           :rem 241
11ØØ NP=Ø:SL=1:V$(Ø)="NEXT PAGE":PRINT"
     {CLR}{DOWN}{RVS}"HE$             :rem 152
111Ø SL=SL+NP:NP=MA-SL+1:IF NP>NV THEN NP
     =NV                               :rem 2Ø1
112Ø IF NP=Ø THEN 3ØØØ                 :rem 8Ø
113Ø GOSUB 9ØØ:FOR I=1 TO NP:V$(I)=PR$(SL
     +I-1):NEXT                        :rem 131
114Ø GOSUB 1ØØ:IF CL+RL=Ø THEN 111Ø
                                         :rem 87
115Ø GOTO 5ØØØ                         :rem 197
2ØØØ GOSUB 29ØØ:MA=Ø                   :rem 66
2Ø1Ø OPEN 15,UN,15,"I"+DR$:GOSUB 1Ø
                                        :rem 117
2Ø2Ø OPEN 1,UN,3,"$"+DR$              :rem 2Ø6
2Ø3Ø SYS 896                           :rem 1Ø7
2Ø4Ø FO=ASC(IN$):IF (FO AND 3)=1 THEN 21Ø
     Ø                                 :rem 144
2Ø5Ø CLOSE 1:CLOSE 15:PRINT"BAD DISK FORM
     AT":END                            :rem 91
21ØØ HE$=MID$(IN$,143,16)              :rem 18
211Ø IF FO>1 THEN SE=18:HE$=HE$+", 1541/4
     Ø4Ø FORMAT":GOTO 22ØØ           :rem 25Ø
212Ø IF FO=1 THEN SE=19:HE$=HE$+", 2Ø4Ø F
     ORMAT"                            :rem 2ØØ
22ØØ DIM PR$(SE*8):FOR I=1 TO SE:SYS 896
                                        :rem 162
221Ø FOR J=1 TO 254 STEP 32:A=ASC(MID$(IN
     $,J)) AND 127                      :rem 16
222Ø IF A=2 THEN MA=MA+1:PR$(MA)=MID$(IN$
     ,J+3,16)                          :rem 118
223Ø NEXT J                             :rem 8Ø
224Ø IF ST THEN I=SE                    :rem 75
225Ø NEXT I                             :rem 81
226Ø CLOSE 1:CLOSE 15:RETURN          :rem 161
29ØØ UN=8:DR=Ø:DR$="Ø"                :rem 197
291Ø RETURN                            :rem 172
292Ø PRINT"UNIT #{2 SPACES}";UN;:INPUT"
     {4 LEFT}";UN                      :rem 122
293Ø PRINT"DRIVE #{2 SPACES}";DR;:INPUT"
     {4 LEFT}";DR:IF DR*(DR-1) THEN 293Ø
                                        :rem 1Ø2
294Ø DR$=STR$(DR):RETURN              :rem 228
3ØØØ GOSUB 9ØØ:PRINT"{CLR}{DOWN}NO MORE P
     ROGRAMS ON DISK.{2 SPACES}OPTIONS AR
     E:                                 :rem 216
```

```
3020 V$(0)="SEE NEW DISK":V$(2)="RESTART
     {SPACE}THIS DISK":V$(4)="QUIT"
                                        :rem 112
3030 GOSUB 100:IF RL=0 THEN RUN  :rem 207
3040 IF RL=2 THEN 1100              :rem 84
3050 PRINT"{CLR}":END               :rem 61
5000 PRINT"{CLR}{2 DOWN}LOAD";CHR$(34);V$
     (RL+16*CL);CHR$(34);",";UN:PRINT"
     {4 DOWN}RUN{HOME}";             :rem 93
5010 IF XA=214 THEN POKE 631,13:POKE 632,
     13:POKE XB,2:END               :rem 80
5020 POKE 623,13:POKE 624,13:POKE 158,2:E
     ND                             :rem 44
6000 IN$="ZZ":FOR I=1 TO 6:IN$=IN$+IN$:NE
     XT                             :rem 179
6010 IN$=MID$(IN$,2):IN$=IN$+IN$ :rem 250
6020 IF PEEK(65535)=255 THEN XA=214:XB=19
     8:GOTO 6040                    :rem 58
6030 XA=216:XB=158                  :rem 231
6040 FOR I=896 TO 929:READ X$:IF LEFT$(X$
     ,1)<>"A" THEN 6060             :rem 51
6050 X$=STR$(VAL(RIGHT$(X$,LEN(X$)-1))+3*
     (XA=216))                      :rem 19
6060 POKE I,VAL(X$):NEXT            :rem 153
6070 RETURN                         :rem 173
6080 DATA 160,2,177,A45,153,A137,0,200,19
     2,6,208,246,162                :rem 226
6090 DATA 1,32,198,255,32,228,255,164,A14
     2,145,A140,200                 :rem 179
6100 DATA 132,A142,196,A139,208,242,76,20
     4,255                          :rem 254
```

Program 2: Super Directory For VIC

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

```
0 GOTO1000                          :rem 42
1 GETZ$:IFZ$=""THEN1                :rem 183
2 RETURN                            :rem 18
10 INPUT#15,ER,ER$:IFER=0THENRETURN:rem 0
30 PRINT"DISK ERROR #"ER:PRINTER$:END:RET
   URN                              :rem 80
50 FORI=1TOSK:GET#1,A$:NEXT:RETURN
                                    :rem 214
70 NA$="":FORK=1TO16:GET#1,A$:NA$=NA$+A$:
   NEXTK:RETURN                     :rem 201
100 PRINT"{HOME}{3 DOWN}":FORI=0TO15:PRIN
    TV$(I):NEXT                     :rem 220
180 POKE198,0:RL=0:GOSUB600         :rem 91
300 FORI=0TO1STEP0:GOSUB1:Z=ASC(Z$):TR=RL
    :FORJ=0TO1STEP0                 :rem 165
340 IF(ZAND127)=17THENRL=(RL+1+2*(Z=145))
    AND15                           :rem 122
360 IFZ=13THENI=1                   :rem 9
370 IFV$(RL)<>B$THENJ=1             :rem 89
380 NEXT:IF(TR<>RL)THENGOSUB500  :rem 246
410 NEXT:RETURN                     :rem 238
500 POKE214,TR+3:PRINT:PRINTV$(TR)
                                    :rem 138
600 POKE214,RL+3:PRINT:PRINT"{RVS}"V$(RL)
    :RETURN                         :rem 235
900 FORI=1TONV:V$(I)=B$:NEXT:RETURN:rem 2
1000 PRINT"{2 HOME}{CLR}":IFPEEK(50003)=0
     THENNEW                        :rem 8
1030 NV=15:GOSUB2000:B$="{19 SPACES}":DIM
     V$(NV)                         :rem 43
1150 NP=0:SL=1:V$(0)="NEXT PAGE":PRINT"
     {CLR}{DOWN}{RVS}"HE$           :rem 157
1200 SL=SL+NP:NP=MA-SL+1:IFNP>NVTHENNP=NV
                                    :rem 201
1220 IFNP=0THEN3000                 :rem 81
1250 GOSUB900:FORI=1TONP:V$(I)=PR$(SL+I-1
     ):NEXT:GOSUB100:IFCL+RL=0THEN1200
                                    :rem 81
1650 GOTO5000                       :rem 202
2000 GOSUB2900:MA=0:OPEN15,UN,15,"I"+DR$:
     GOSUB10:OPEN1,UN,3,"$"+DR$  :rem 114
2060 GET#1,FO$:FO=ASC(FO$+""):SK=141:IFFO
     =67THENSK=3                    :rem 45
2110 GOSUB50:GOSUB70:HE$=NA$:SK=96:SE=19+
     (FO>1):IFFO=67THENSK=742:SE=28
                                    :rem 74
2140 GOSUB50:SK=11:DIMPR$(SE*8)  :rem 72
2330 FORI=1TOSE:FORJ=1TO8:IFSTTHENI=SE:J=
     8:GOTO2440                     :rem 72
2360 GET#1,B$:GET#1,A$:GET#1,A$:GOSUB70:I
     FB$=CHR$(130)THENMA=MA+1:PR$(MA)=NA$
                                    :rem 181
2420 GOSUB50:IFJ<>8THENGET#1,A$:GET#1,A$
                                    :rem 101
2440 NEXTJ,I:CLOSE1:CLOSE15:RETURN
                                    :rem 217
2900 INPUT"UNIT #{2 SPACES}8{3 LEFT}";UN
                                    :rem 239
2950 INPUT"DRIVE #{2 SPACES}0{3 LEFT}";DR
     :IFDR*(DR-1)THEN2950           :rem 230
2960 DR$=STR$(DR):RETURN            :rem 230
3000 GOSUB900:V$(4)="QUIT":PRINT"{CLR}
     {DOWN}NO MORE PROGRAMS ON DISK.
     {2 SPACES}OPTIONS ARE:         :rem 213
3030 V$(0)="SEE NEW DISK":V$(2)="RESTART
     {SPACE}THIS DISK":GOSUB100:IFRL=0THE
     NRUN                           :rem 183
3080 IFRL=2THEN1150                 :rem 93
3090 PRINT"{CLR}":END               :rem 65
5000 PRINT"{CLR}{2 DOWN}LOAD"CHR$(34)V$(R
     L)CHR$(34)","UN:PRINT"{5 DOWN}RUN
     {HOME}";                       :rem 252
5020 POKE631,13:POKE632,13:POKE198,2:END
                                    :rem 46
```

Program 3: Super Directory For Atari

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

```
JJ 100 REM  SUPER DIRECTORY
OB 105 DIM DN$(19)
BL 107 DN$="DISK NAME"
DC 110 GRAPHICS 0:ENT=0:CHANGED=0
NM 115 GOSUB 5000
NE 120 DL=PEEK(560)+PEEK(561)*256+4
GD 130 POKE DL-1,7+64:POKE DL+2,6:POKE
        DL+4,13:POKE DL+24,13
LI 140 SETCOLOR 4,9,4:SETCOLOR 1,0,10
HP 150 SETCOLOR 0,1,10:POKE 752,1
BO 160 ? "super  directory"
IL 170 DIM FL$(40*64),T$(40),F$(40),DS
       C$(40*64),CDN$(20),FREE$(17)
GC 180 TRAP 200
LJ 190 OPEN #1,6,0,"D:*.*":TRAP 40000:
       GOTO 210
OP 200 ? "Can't read directory.  Error
        #";PEEK(195):END
PG 210 INPUT #1,T$:IF LEN(T$)<17 THEN
       250
FF 220 NF=NF+1
GH 230 FL$(NF*40-39,NF*40-26)=T$:FL$(N
       F*40-25)="{30 SPACES}"
GC 240 GOTO 210
JG 250 CLOSE #1:FREE$=T$
MC 260 TRAP 270:OPEN #1,4,0,"D:DESCR.D
       IR":TRAP 40000:GOTO 280
```

Super Directory For Atari

Charles Brannon, Program Editor

"Super Directory" is an easy-to-use menu program for selecting and running BASIC programs from disk. In addition, it alleviates the limitations of Atari's eight-character filename by storing a 20-character description of each filename.

Type in Program 3 and SAVE a copy of Super Directory on every disk you want to use it with (you may want to call it MENU). You can change line 107 to name your disk. This name will appear at the top of the menu program when RUN.

When you first RUN Super Directory, it will read in the disk directory and display it on the screen. A large, wide cursor will be resting on top of the first filename in the directory. You can move the cursor up or down with the arrow keys, but you do not have to hold down CTRL. You can also use a joystick to move the cursor up or down.

If the directory will not fit on one screen, it will scroll upward as you push the cursor "past" the bottom of the screen. You can also scroll the screen down when the cursor is at the top of the screen. Press any key to select the file, or use the joystick trigger button.

You will switch to another screen, where you are given three choices: press START to RUN the program, press SELECT to change the description, and OPTION to save the descriptions. You can also press RETURN to skip these choices and return to the menu.

Making Sense Of Filenames

You probably noticed that the second column on the screen said "no description" for all the filenames. This is because you haven't entered any yet. If you press SELECT while you are on the other screen, you can enter a description, up to 20 characters. You can enter anything you want here that will help you make sense of filenames like ASKRD.TXT, WMAKER, HAWKMEN, or EASMD.COM.

The description file is saved to the disk under the filename DESCR.DIR. If you delete it, your descriptions are gone. Every time you RUN Super Directory, it will match up each description with each directory entry. If you move a file around on the directory, it will still be matched up with the proper description. Super Directory also has to skip over descriptions that once applied to deleted files. This correlation process takes a few seconds before the menu first appears.

Verify Your Update

You can press OPTION on the second screen to insure that the description file is updated after you change it. It will also be automatically written out if you press START to run a program.

There are some files, like DOS.SYS, that you cannot run, obviously. Super Directory only lets you run BASIC programs that have been SAVEd (not LISTed) to disk. If you try to run any other kind of file, or if there is some kind of disk error, Super Directory will so inform you, then reRUN itself.

Thanks to the description file, Super Directory is more than a mere menu program. It can help you catalog your disks, and get around the eight-character filename limitation.

```
OJ 270  FOR I=1 TO NF:FL$(I*40-22,I*40)
        ="no description  ":NEXT I:CLOS
        E #1:GOTO 310
NJ 280  TRAP 290:INPUT #1,F$:ENT=ENT+1:
        DSC$(ENT*40-39)="{38 SPACES}"
FB 285  DSC$(ENT*40-39,ENT*40)=F$:GOTO
        280
GF 290  CLOSE #1
KM 291  FOR I=1 TO NF:IX40=I*40:IX2=IX4
        0-39
CM 292  T$=FL$(IX2)
MH 293  FOR J=1 TO ENT
AG 294  F$=DSC$(J*40-39)
HM 295  IF T$(1,13)<>F$(1,13) THEN NEXT
        J:GOTO 297
FJ 296  FL$(IX2,IX40)=F$:GOTO 300
HC 297  FL$(IX40-22,IX40)="no descripti
        on"
BL 300  NEXT I
EE 310  ENT=NF:LIM=NF:IF LIM>19 THEN LI
        M=19
HJ 311  CDN$=" ":CDN$(19)=" ":CDN$(2)=C
        DN$:Z=11-LEN(DN$)/2:CDN$(Z,Z+LE
        N(DN$))=DN$
HG 330  POSITION 20,0:? CDN$;
BE 335  ? "File name EXT{3 SPACES}Descr
        iption{10 SPACES}"
NN 337  COLOR 21:PLOT 2,2:DRAWTO 38,2:P
        LOT 2,22:DRAWTO 38,22:POSITION
        2,3
LK 340  FOR I=1 TO LIM
LG 350  F$=FL$(I*40-39)
KO 360  ? " ";F$(3,10);" ";F$(11,13);"
        ";F$(18,38)
AG 370  NEXT I:CURR=1
GE 380  Y=3
```

```
ME 390 F$=FL$(CURR*40-39):T$=F$(3,10):
       T$(9)=" ":T$(10)=F$(11,13):T$(1
       3)="  ":T$(15)=F$(18,38):F$=T$
FC 400 A=USR(ADR(M$),ADR(T$),13)
DH 425 POSITION 2,Y:? " ";T$
NH 430 ST=STICK(0):S=PEEK(764):IF S=25
       5 AND PEEK(53279)=7 AND ST=15 A
       ND STRIG(0) THEN 430
NE 435 SOUND 0,20,10,8
CG 440 POSITION 2,Y:? " ";F$
JK 445 IF PEEK(53279)<7 OR S=12 OR STR
       IG(0)=0 THEN POKE 764,255:GOTO
       500
DB 450 IF (S=15 OR ST=13) AND CURR+1<=
       ENT THEN Y=Y+1:CURR=CURR+1:IF Y
       >LIM+2 THEN ? "{INS LINE}";:POS
       ITION 2,3:? "{DEL LINE}":Y=Y-1
IF 460 IF (S=14 OR ST=14) AND CURR>1 T
       HEN Y=Y-1:CURR=CURR-1:IF Y<3 TH
       EN POSITION 2,3:? "{INS LINE}";
       :POSITION 2,22:? "{DEL LINE}";:
       Y=Y+1
DK 470 SOUND 0,0,0,0:POKE 764,255
HC 490 GOTO 390
GC 500 SOUND 0,0,0,0
MD 505 ? "{CLEAR}";F$(1,12):POSITION 2
       ,20:? FREE$:POSITION 2,8
OL 510 ? :? "Press  START  to run pro
       gram."
CI 520 ? :? "Press  SELECT  to do desc
       ription."
JF 525 ? :? "Press  OPTION  to save de
       scriptions."
BA 527 ? :? "Press  RETURN  to return
       to menu."
CN 530 POKE 53279,0
EC 540 CONSOLE=PEEK(53279):K=PEEK(764)
AA 550 IF CONSOLE<7 OR K=12 THEN 600
GJ 560 IF PEEK(20)<30 THEN 540
AO 570 POKE 755,2-PEEK(755)
JF 580 POKE 20,0:GOTO 530
JG 600 POKE 755,2:POKE 764,255
NM 605 IF K=12 THEN 710
LD 610 IF CONSOLE<>6 THEN 650
NC 620 T$="D:":T$(3)=F$(1,8)
EL 625 FOR I=1 TO LEN(T$):IF T$(I,I)<>
       " " THEN NEXT I
LO 627 POP :IF ASC(F$(10))>32 THEN T$(
       I)=".":T$(I+1)=F$(10,12)
EJ 628 GOSUB 1000:IF T$="D:DOS.SYS" TH
       EN DOS
MM 630 TRAP 640:GRAPHICS 0:RUN T$
OL 640 T$="Can't run program. Error #"
       :T$(LEN(T$)+1)=STR$(PEEK(195)):
       A=USR(ADR(M$),ADR(T$),LEN(T$))
CE 645 POSITION 2,18:? T$;:FOR W=1 TO
       500:NEXT W:RUN
LE 650 IF CONSOLE<>5 THEN 720
FC 660 ? "{CLEAR}";F$(1,12)
LB 670 ? "{5 DOWN}Edit message below (
       max 20 characters)":POKE 752,0
DC 680 T$=F$(15,34):? " ";T$:? " -----
       ---------------{2 UP}":INPUT T$
PG 685 IF T$="" THEN 660
PB 690 IF LEN(T$)<20 THEN T$(LEN(T$)+1
       )="{20 SPACES}"
OH 695 IF LEN(T$)>20 THEN T$=T$(1,20)
GC 697 IF T$<>F$(15,34) THEN CHANGED=1
AI 700 FL$(CURR*40-22,CURR*40)=T$:POKE
        752,1
KF 710 ? "{CLEAR}super  directory":GOT
       O 330
DF 720 GOSUB 1000:IF  NOT CHANGED THEN
        POSITION 2,4:? "Descriptions u
       nchanged"
GP 725 GOTO 530
GL 1000 REM SAVE DESCRIPTIONS
DP 1005 IF  NOT CHANGED THEN RETURN
LN 1010 TRAP 1030
BC 1020 OPEN #1,8,0,"D:DESCR.DIR":TRAP
         40000:GOTO 1040
HD 1030 CLOSE #1:T$="Error #":T$(8)=ST
        R$(PEEK(195)):A=USR(ADR(M$),AD
        R(T$),LEN(T$)):POSITION 2,6:?
        T$:RETURN
EH 1040 FOR I=1 TO ENT:CURR=I-1
JM 1050 PRINT #1;FL$(CURR*40+1,CURR*40
        +39)
EP 1060 NEXT I
KM 1070 CLOSE #1:RETURN
MF 5000 DIM M$(40)
LJ 5010 M$="h{I}{B}{P}{I}{P}{E}hh{JP}{A}'h
        {E}{L}h{E}{K}"
KI 5020 M$(LEN(M$)+1)="hh{E}{M}{,}{IK}
        {I}{J}{Q}{KHDMPU}'{2 ,}"
KI 5030 RETURN
```

Program 4: Super Directory For PC And PCjr

```
100 'Super Directory for The IBM Persona
l Computer
110 'for monochrome or color adaptor, 80
 columns
120 SCREEN 0,0,0:WIDTH 80:COLOR 7,0:CLS:
DEFINT A-Z:KEY OFF:FOR I=1 TO 10:KEY I,"
":NEXT
130 CR$=CHR$(17)+CHR$(196)+CHR$(217)
140 PRINT"Welcome to ";:COLOR 15:PRINT"S
uper Directory":COLOR 7
150 'Remove the word REM from following
line for automatic use with drive A
160 REM DRIVE$="A:":FSPEC$="A:*.*":GOTO
200
170 PRINT:PRINT "Select Drive: (";:COLOR
 16,15:PRINT"A B";:COLOR 7,0:PRINT CHR$(
29);CHR$(29);"/";CHR$(28);")"
180 DRIVE$=INKEY$+":":A=ASC(DRIVE$):IF (
A OR 32)<97 OR (A OR 32)>98 THEN 180
190 DRIVE$=CHR$(A AND 223)+":":FSPEC$=DR
IVE$+"*.*"
200 GOSUB 5000:CLS:COLOR 23:PRINT"Readin
g description file"
210 DIM D$(ENTRIES):FOR I=0 TO ENTRIES:D
$(I)=CHR$(9)+"--" :NEXT
220 ON ERROR GOTO 310
230 OPEN DRIVE$+"DESCR.DIR" FOR INPUT AS
 #1
240 LINE INPUT #1,DISKNAME$:LINE INPUT#1
,A$:NUMREC=VAL(A$)
250 FOR ITEMS=0 TO NUMREC
260   LINE INPUT #1,F$:LINE INPUT#1,D$
270   FOR I=0 TO ENTRIES
280     IF F$=F$(I) THEN D$(I)=D$
290     NEXT:NEXT
300 GOTO 320
310 RESUME 320
320 CLOSE#1:ON ERROR GOTO 0
330 PAGES=INT(ENTRIES/10)
340 CURR=0
350 START=CURR*10:FINISH=START+9:IF FINI
SH>ENTRIES THEN FINISH=ENTRIES
```

IBM Notes: Super Directory

Charles Brannon, Program Editor

To use Super Directory, you'll need an IBM PC with at least 64K, and either a monochrome or color adaptor. Super Directory will also work on the Expanded PCjr with Cartridge BASIC.

If you're not a programmer, you may find working with your PC to be a bit perplexing at times. You turn on your system, wait 45 seconds, face the cryptic A> prompt, enter BASIC, then RUN "filename" to start a BASIC program. It would be much easier if you had a list of all the programs and could run any one by just pressing a function key.

Super Directory is your solution. When you RUN it, it will give you a list of all the files on your disk. To run any displayed program, press the appropriate function key, then press the enter key (designated with a crooked arrow). Super Directory only displays ten files at a time. If there are more than ten files, you can press Pg Dn (the number 3 on the numeric keypad, if you have NUMLOCK on) to go on to the next page. You can also page backwards with Pg Up.

There's more to Super Directory, though. The 11-character filename length offered by PC-DOS does not allow very descriptive names for your files. How can you make sense of names like QTESTV1.BAS? Using Super Directory, you can label each filename with a 61 character description. Super Directory keeps the description with the filename, and displays it every time you go back to Super Directory.

To enter a description, press the appropriate function key for the file you wish to describe, then press the SPACE bar. The last line of the screen will always tell you what to do. The first time you try to describe a file, you will be asked for the disk name. Once you've entered a descriptive name for the disk, Super Directory remembers it and will no longer ask you for it. You can then enter or edit the description.

After you enter a description, the screen will be re-drawn, and you will be able to see the description you've given to a file name. If you go back to redo a description, it will be displayed. You can type over it, or move the cursor to edit the description. Remember to put the cursor at the end of the line when you are through editing (you can use the END key to skip to the end of the line).

When you run a program from the menu, the descriptions will be written to disk first. If you just want to write out the descriptions without running a program, you can press ESC from the main menu. You will see the line:

> 1. Exit to BASIC 2. Exit to DOS 3. Re-Run
> 4. Save Descriptions 5. Menu

Press 4 to save the descriptions to disk. The descriptions will be saved to disk under the filename "DESCR.DIR". Don't DELete this or you will lose your descriptions. You can also use the other options to return to BASIC, DOS, or re-RUN Super Directory. Pressing 5 will take you back to the main menu (if you pressed ESC by mistake, say).

Super Directory will detect errors and prompts you to press ENTER. You'll usually be returned to the main menu. Don't try to run a program which is not BASIC, however. You'll probably get the message "Direct Statement in File" and find that Super Directory has disappeared.

You can make Super Directory completely automatic. If you don't have BASIC on your disk, you can use COPY from DOS. Then enter this one line command to have BASIC and Super Directory come up automatically when you turn on your PC:

OPEN "AUTOEXEC.BAT" FOR OUTPUT AS #1:PRINT#1,"BASIC SUPERDIR":CLOSE#1

This assumes that you've saved Super Directory to the same disk with the command SAVE "SUPERDIR". Super Directory will normally ask you which drive you want to list from. If you only have one drive, or want to always look at drive A:, remove the keyword REM from line 160. Leave the rest of the line in place. Now you can add a flexible, easy to use menu to any disk. Super Directory makes it easy enough for a child to use!

```
360 CLS:COLOR 0,15:PRINT STRING$(80,32):
LOCATE 1,2:PRINT"Super Directory";TAB(70
);"Drive ";DRIVE$:LOCATE 1,40-LEN(DISKNA
ME$)/2:PRINT DISKNAME$:PRINT
370 FOR I=START TO FINISH
380 COLOR 0,15:PRINT "F";LEFT$(MID$(STR$
(1+I-START),2)+" ",2);:COLOR 15,0:PRINT
" ";F$(I);TAB(18);:COLOR 7:PRINT D$(I):P
RINT STRING$(80,196);
390 NEXT
```

```
400 LOCATE 25,1:COLOR 15,0:PRINT"Press "
;:COLOR 0,15:PRINT"F1";:COLOR 15,0:PRINT
" to ";:COLOR 0,15:PRINT "F";MID$(STR$(1
+FINISH-START),2);:COLOR 15,0:PRINT" to 
select program.  Press PgUp or PgDn to p
age, ESC to quit.";
410 LOCATE 23,32:PRINT"Page #";CURR+1;"o
f";PAGES+1
420 A$=INKEY$:IF A$="" THEN 420
430 IF A$<>CHR$(27) THEN 540
440 LOCATE 25,1:PRINT SPACE$(79);:LOCATE
 25,1:PRINT"  1. Exit to BASIC  2. Exit 
to DOS  3. Re-RUN  4. Save descriptions 
 5. Menu";
450 A$=INKEY$:IF A$<"1" OR A$>"5" THEN 4
50
460 ON VAL(A$) GOTO 470,480,490,500:GOTO
 350
470 COLOR 7:CLS:END
480 SYSTEM
490 RUN
500 ON ERROR GOTO 510:GOSUB 1000:GOTO 35
0
510 BEEP:LOCATE 25,1:PRINT SPACE$(79);:L
OCATE 25,1:COLOR 31:PRINT"Can't save des
criptions.  ";:COLOR 7:PRINT"Press ";CR$
;" to continue.";
520 IF INKEY$<>CHR$(13) THEN 520
530 RESUME 350
540 IF A$=CHR$(0)+CHR$(81) THEN CURR=-(C
URR+1)*(CURR<PAGES):GOTO 350
550 IF A$=CHR$(0)+CHR$(73) THEN CURR=CUR
R-1:CURR=CURR-(PAGES+1)*(CURR<0):GOTO 35
0
560 A=ASC(MID$(A$+"0",2))-59:IF A<0 OR A
>FINISH-START THEN BEEP:GOTO 420
570 LOCATE 25,1:PRINT SPACE$(79);:LOCATE
 25,1:PRINT"Press ";CR$;" to run program
, ESC to return to menu, SPACE to do des
cription.";
580 LOCATE 3+A*2,5:COLOR 31:PRINT F$(STA
RT+A);:COLOR 15
590 A$=INKEY$:IF A$<>CHR$(13) AND A$<>CH
R$(27) AND A$<>CHR$(32) THEN 590
600 IF A$=CHR$(27) THEN LOCATE 3+A*2,5:P
RINT F$(START+A);:GOTO 400
610 IF A$<>CHR$(32) THEN 670
620 IF DISKNAME$="" THEN LOCATE 25,1:PRI
NT SPACE$(79);:LOCATE 25,1:LINE INPUT;"E
nter name of disk : ";DISKNAME$ :GOTO 62
0
630 LOCATE 25,1:PRINT SPACE$(79);:LOCATE
 25,1:Z=START+A:PRINT "Description :";D$
(Z);:LOCATE 25,15:LINE INPUT ;D$(Z):D$(Z
)=LEFT$(" "+D$(Z),62):GOTO 350
640 LOCATE 25,1:PRINT SPACE$(79);:BEEP:C
OLOR 31:LOCATE 25,1:PRINT"Cannot save de
scriptions to disk.  ":COLOR 7:PRINT"Run
 program anyway? (Y/N):";:COLOR 7
650 A$=INKEY$:IF A$<>"y" AND A$<>"Y" AND
 A$<>"n" AND A$<>"N" THEN 650
660 IF A$="y" OR A$="Y" THEN RESUME 680 
ELSE RESUME 350
670 ON ERROR GOTO 640:GOSUB 1000
680 ON ERROR GOTO 690:COLOR 7:CLS:RUN DR
IVE$+F$(START+A)
690 LOCATE 25,1:PRINT SPACE$(79):COLOR 2
3:BEEP:PRINT"Cannot run ";F$(A);".  ";:C
OLOR 7:PRINT"Press ";CR$;" to continue..
.";
700 IF INKEY$<>CHR$(13) THEN 700
710 RESUME 350
720 END
1000 'Save descriptions to disk
1010 OPEN DRIVE$+"DESCR.DIR" FOR OUTPUT 
AS #1
1020 PRINT#1,DISKNAME$;CHR$(13);ENTRIES;
CHR$(13);
1030 FOR I=0 TO ENTRIES:PRINT#1,F$(I);CH
R$(13);D$(I);CHR$(13);:NEXT
1040 CLOSE #1:ON ERROR GOTO 0:RETURN
1050 '
5000 'This subroutine reads disk directo
ry into a string array
5010 'Enter with FSPEC$, the file spec f
or the FILES command
5020 'Exits with array F$, and NUMFILES,
 the number of files
5030 'uses a temporary array, TT$, which
 is ERASEd after use
5040 '
5050 DEF SEG=0:WIDTH 80
5060 HEAD=1050:TAIL=1052:BUFFER=1054
5070 CLS:COLOR 23,0,0:PRINT"Reading disk
 directory"
5080 COLOR 0:ON ERROR GOTO 5100
5090 FILES FSPEC$:ON ERROR GOTO 0:GOTO 5
110
5100 BEEP:COLOR 31:CLS:PRINT"Cannot read
 directory":COLOR 7:ON ERROR GOTO 0:END
5110 DIM TT$(24):LOCATE 3,1:COLOR 7:ROWS
=0
5120 'Put code for End, Enter into keybo
ard buffer:
5130 POKE HEAD,30:POKE TAIL,34:POKE BUFF
ER,0:POKE BUFFER+1,79:POKE BUFFER+2,13:P
OKE BUFFER+3,28
5140 LINE INPUT TT$(ROWS)
5150 IF TT$(ROWS)<>"" THEN ROWS=ROWS+1:G
OTO 5130
5160 IF NOT DIMMED THEN DIM F$(ROWS*4-1)
:DIMMED=1
5170 ROWS=ROWS-1
5180 FOR I=0 TO ROWS
5190  FOR J=0 TO 3
5200     T$=MID$(TT$(I),J*18+1,12)
5210     IF T$<>"" THEN F$(ENTRIES)=T$:EN
TRIES=ENTRIES+1
5220  NEXT J
5230 NEXT I
5240 ERASE TT$:ENTRIES=ENTRIES-1
5250 DEF SEG:RETURN
```

©

A Beginner's Guide To Typing In Programs

What Is A Program?

A computer cannot perform any task by itself. Like a car without gas, a computer has *potential*, but without a program, it isn't going anywhere. Most of the programs published in COMPUTE! are written in a computer language called BASIC. BASIC is easy to learn and is built into most computers (on some computers, you have to purchase an optional BASIC cartridge).

BASIC Programs

Each month, COMPUTE! publishes programs for many machines. To start out, type in only programs written for your machine, e.g., "TI Version" if you have a TI-99/4. Later, when you gain experience with your computer's BASIC, you can try typing in and converting certain programs from one computer to yours.

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

Braces And Special Characters

The exception to this typing rule is when you see the braces, such as DOWN}. Anything within a set of braces is a special character or characters that cannot easily be listed in a printer. When you come across such a special statement, refer to the appropriate key for your computer. For example, if you have an Atari, refer to the "Atari" section in "How To Type COMPUTE!'s Programs."

About DATA Statements

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

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

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

Get To Know Your Machine

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

A Quick Review

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

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

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

We regret that we are no longer able to respond to individual inquiries about programs, products, or services appearing in COMPUTE! *due to increasing publication activity. On those infrequent occasions when a published program contains a typo, the correction will appear on the CAPUTE! page, usually within eight weeks. If you have specific questions about items or programs which you've seen in* COMPUTE!, *please send them to Readers' Feedback, P.O. Box 5406, Greensboro, NC 27403.* ©

How To Type COMPUTE!'s Programs

Many of the programs which are listed in COMPUTE! contain special control characters (cursor control, color keys, inverse video, etc.). To make it easy to tell exactly what to type when entering one of these programs into your computer, we have established the following listing conventions. There is a separate key for each computer. Refer to the appropriate tables when you come across an unusual symbol in a program listing. If you are unsure how to actually enter a control character, consult your computer's manuals.

Atari 400/800

Characters in inverse video will appear like: INVERSE VIDEO
Enter these characters with the Atari logo key, {⅄}.

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

Graphics characters, such as CTRL-T, the ball character ● will appear as the "normal" letter enclosed in braces, e.g. {T}.

A series of identical control characters, such as 10 spaces, three cursor-lefts, or 20 CTRL-R's, will appear as {10 SPACES}, {3 LEFT}, {20 R}, etc. If the character in braces is in inverse video, that character or characters should be entered with the Atari logo key. For example, {♥} means to enter a reverse-field heart with CTRL-comma, {5U} means to enter five inverse-video CTRL-U's.

Commodore PET/CBM/VIC/64

Generally, any PET/CBM/VIC/64 program listings will contain words within braces which spell out any special characters: {DOWN} would mean to press the cursor down key. {5 SPACES} would mean to press the space bar five times.

To indicate that a key should be *shifted* (hold down the SHIFT key while pressing the other key), the key would be underlined in our listings. For example, S would mean to type the S key while holding the shift key. If you find an underlined key enclosed in braces (e.g., {10 N}), you should type the key as many times as indicated (in our example, you would enter ten shifted N's). Some graphics characters are inaccessible from the keyboard on CBM Business models (32N, 8032).

For the VIC and 64, if a key is enclosed in special brackets, [< >], you should hold down the *Commodore key* while pressing the key inside the special brackets. (The Commodore key is the key in the lower left corner of the keyboard.) Again, if the key is preceded by a number, you should press the key as many times as indicated.

Rarely, you'll see in a Commodore 64 program a solitary letter of the alphabet enclosed in braces. These characters can be entered by holding down the CTRL key while typing the letter in the braces. For example, {A} would indicate that you should press CTRL-A.

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

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

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

Use the following tables when entering special characters:

VIC And 64

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

All Commodore Machines

Clear Screen	{CLR}	Cursor Left	{LEFT}
Home Cursor	{HOME}	Insert Character	{INST}
Cursor Up	{UP}	Delete Character	{DEL}
Cursor Down	{DOWN}	Reverse Field On	{RVS}
Cursor Right	{RIGHT}	Reverse Field Off	{OFF}

Apple II / Apple II Plus

All programs are in Applesoft BASIC, unless otherwise stated. Control characters are printed as the "normal" character enclosed in braces, such as {D} for CTRL-D. Hold down CTRL while pressing the control key. You will not see the special character on the screen.

Texas Instruments 99/4

The only special characters used are in PRINT statements to indicate where two or more spaces should be left between words. For example, ENERGY {10 SPACES} MANAGEMENT means that ten spaces should be left between the words ENERGY and MANAGEMENT. Do not type in the braces or the words 10 SPACES. Enter all programs with the ALPHA LOCK on (in the down position). Release the ALPHA LOCK to enter lowercase text.

CAPUTE!

Modifications Or Corrections To Previous Articles

Calorie Cop For Atari And TI

In the Atari version of this activity planning program from the December 1983 issue (p. 52), the GOTO 470 in lines 540 and 552 should be changed to GOTO 475. Also, lines 250 and 420 can be deleted.

In the TI version (Program 3), line 360 can be deleted and the following lines should be added or changed:

```
445 PRINT : :
447 PRINT "(OR HIT ENTER FOR MORE)"
750 IF MIN=0 THEN 870
```

Atari Quatrainment

The characters which appear as : : in line 20020 of Program 1 (February 1984, p. 78) should be the vertical line character, obtained by pressing SHIFT and the = key.

The Beginner's Page: Program Forms

Line 560 from Program 1 of this February column (p. 102) should read:

```
560 DATA GALLONS,3.785,LITERS
```

VIC 3-D Drawing Master

Users of this program (February 1984, p. 146) will encounter a syntax error in line 1803 when attempting to load a picture file from tape. The line should read:

```
1803 IFLEFT$(IN$,1)="T"THENIN=1:GOTO1805
```

64 Sound Tester

There were typographical errors in the corrections to this program (November 1983, p. 187) which appeared in the February "CAPUTE!" page. The modified program lines should have read:

```
310 FOR I=1TO15STEP2: POKEW,X: POKEHF,SO(I
    ,A(2)): POKELF,SO(I+1,A(2))
311 O=O+1: FORN=1TOD(O): NEXT: POKEW,X-1:
    NEXT: FORI=1TO10000: NEXT
```

Hidden 64 Memory

If you use the techniques outlined in this article (January 1984, p. 172) to access the RAM under the Kernal ROM, you should be aware that at least one of the 64's Kernal routines writes to the RAM beneath it. Raymond Quiring notes that the Kernal routine RESTOR at address 65418 ($FF8A) writes to the RAM from 64816–64846 ($FD30–$FD4E). The RESTOR routine is called when the RESTORE key is pressed, or when a BRK instruction is encountered in machine language.

Making Cents

The February "Readers' Feedback" suggestion for a program line which formats numbers as dollars and cents (p. 10) produces improper values in those cases where the number is negative or zero. David Gamache suggests instead the following line, which works for both positive and negative numbers and zero:

```
V$=STR$(X+(.001*SGN(X))):V$=LEFT$(V$,LEN(
V$)-1):IFX=.THENV$="0.00"
```

Program Line Addresses For VIC And 64

The instructions given in January "Readers' Feedback" (p. 10) for appending this program to an existing BASIC program work only for the Commodore 64. For the unexpanded VIC, you should use POKE 44,16 instead of POKE 44,8 in step 4. The proper value for the 3K expanded VIC is POKE 44,4, and with 8K or more expansion you should use POKE 44,18.

VIC Modem Save And Download

There are still bugs in this program (November 1983, p. 215) after the corrections in the January "CAPUTE!" page. Load your version of the terminal program (Program 2 with the data from Program 1 added). Type POKE 45,49:POKE 46,20 and RETURN, then make the following changes and additions, suggested by Larry Flohaug:

```
130 PRINT"{CLR}{DOWN}1-SAVE TO TAPE":PRIN
    T"2-PRINTER":PRINT"3-TAPE & PRINTER"
140 PRINT:PRINT"WHICH?"
170 OPEN4,4:FORK=6656TOPEEK(0)*256+PEEK(1
    ):PRINT#4,CHR$(PEEK(K)AND127);:NEXT
175 PRINT#4:CLOSE4
200 GOTO225
```

After making the modifications, type POKE 45,1:POKE 46,26 and save a copy of the revised program. The test of the Program 1 data in the original article (p. 216) will insure that the data has not been affected by these changes.

Colorbot

When you lose your last man in the VIC or 64 version of this game from the January issue (p. 84), the screen clears immediately. Cliff Tener suggests adding the following line to either version to create a sufficient delay to check your final score:

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

Atari XL Compatibility Problems

Based on mail we have received, we suspect that the "Polycopy" program from the November 1983 issue will not run properly on any of the new Atari XL computers. The following programs work on the 1200XL, but apparently not on the 600 or 800XL: "Chopperoids" (December 1983), "Demons Of Osiris" (January 1984), and "Circus" (February 1984). ©

NEWS&PRODUCTS

Color Computer Cassette-Based Games

Radio Shack has produced several cassette-based computer games for the TRS-80 Color Computer with Extended BASIC for children seven years of age and older. The three games require joysticks and encourage players to work together.

Peanut Butter Panic! is a two-person game in which players jump for stars to make peanut butter sandwiches. By working together, players are able to catch the most valuable stars.

In *Taxi*, players get behind the wheel of a cab. By maneuvering around the street grids based on city maps, players try to deliver as many passengers as possible before time runs out. When played cooperatively, the game encourages communication and division of labor.

Star Trap is a maze game in which players attempt to trap stars by blocking their paths. Two players working as a team are more effective than one player.

Each of these games is priced at $19.95.

Another game now available from Radio Shack for the Color Computer on tape is *ZAXXON*. The player becomes the pilot of a fighter spacecraft on a mission to meet and destroy the deadly *ZAXXON* robot. Points are scored by destroying a variety of threats from enemy planes, base missiles, firing gun emplacements, and radar towers.

The game sells for $34.95 and requires a joystick.

Tandy Corporation/Radio Shack
1800 One Tandy Center
Fort Worth, TX 76102
(817) 390-3300

Games For Apple II, Atari

Kangaroo, Inc., has released two new games, *Jeepers Creatures* and *My House My Home*, both for the Apple II and Atari.

Jeepers Creatures takes children through a zoo of 30 basic animals, with interchangeable heads, torsos, legs, and tails. More than 20,000 different combinations are possible for children to create.

My House My Home is an electronic playhouse. A moving conveyor belt with furniture, pets, and people can be controlled by either a joystick or the keyboard. The user can furnish a cutaway view of a house in a variety of different ways.

Jeepers Creatures and *My House My Home* are priced at $34.95 on disk.

Kangaroo, Inc.
322 South Michigan Avenue
Suite 700
Chicago, IL 60604
(312) 987-9050

Atari Tape Interfaces

RC Systems, Inc., has introduced two models of tape interfaces which allow a standard cassette recorder to be used for loading and saving programs with any of the Atari computers.

The models are the AA-2 and AA-1, the second of which operates with the Atari 400 and 800 computers. The AA-1 has all the features of the AA-2, but incorporates additional circuitry to duplicate and rejuvenate program tapes (a second recorder is required for this).

The assembled and tested AA-2 cassette interface board is priced at $27.95. In kit form, it is available for $19.95. The AA-1 is priced at $39.95; in kit form, $29.95.

RC Systems, Inc.
121 West Winesap Road
Bothell, WA 98012
(206) 771-6883

Filing, Mailing Systems For TI

TI File, *TI Mail*, and *TI File Junior* are three packages developed for the TI-99/4A computer system by Kinetic Designs.

TI File is a multipurpose filing system which allows the user to construct, sort, maintain, and print out a variety of files for home or business.

TI File requires Extended BASIC, 32K memory expansion, and cassette deck. Options include disk drive and printer. It is available on tape for $14.95 and on disk for $17.95. A $2 postage fee should be added.

TI Mail is a mailing list management system which also requires Extended BASIC, 32K memory expansion, and cassette deck. An optional heading allows the user to keep track of expiration dates, scores, or anything else related to those listed. *TI Mail* is available on cassette for $10.95 or on disk for $13.95. A postage fee of $2 is additional.

TI File Junior is similar to the *TI File*, but requires only a standard 16K TI-99/4A and a cassette deck. It is available on cassette for $9.95, plus a $2 postage fee.

Kinetic Designs
P.O. Box 1585
Orange Park, FL 32067
(904) 264-6777

Desert Game For Apple

Sands of Egypt, a desert adventure game from Datasoft, has been introduced for the Apple II, IIe, and II+ computers.

The game incorporates action with riddles using a split screen image. It follows Lord Charles Buckingham III, a desert traveller, as the player attempts to find hidden treasure by answering a series of riddles and by dodging dangers and obstacles.

The game is priced at $29.95, and is also available for the Atari 400, 800, and 1200 computers.

Datasoft, Inc.
9421 Winnetka Avenue
Chatsworth, CA 91311
(213) 701-5161

Basketball Simulation Game

Electronic Arts has introduced *One-on-One*, a basketball simulation game that pits Boston Celtics star Larry Bird against Julius (Dr. J) Erving of the Philadelphia 76ers.

It is available on disk for the Apple II, II+, and IIe, and will be available this year for the Atari and Commodore 64 computers.

Statistical tables reflecting the players' strengths all over the court have been built into the game. The two players also advised the game creators on their individual strategies and maneuvers. Among the features are a 24-second-shot clock, a game clock, and a scoreboard that shatters when a slam dunk is made. The game also has hot and cold shooting functions, instant replays, and visible measures of player fatigue.

One-on-One has a suggested retail price of $40.

Electronic Arts
2755 Campus Drive
San Mateo, CA 94403
(415) 571-7171

Commodore 64 Checkwriting, Accounts Package

COMP-U-CHECK is a checkwriting and accounts servicing package designed for personal or small business applications by Hot Data Software Development.

The package requires a Commodore 64 computer, disk drive, and a printer.

COMP-U-CHECK provides checkwriting, check account balancing, credit account management, letter writing, automatic bill paying, and form letters. Also included are tax time reports, constant balance and totals paid to date, new year accounts reset, and user support through a direct-help phone number.

The package is available on disk for $64.95.

Hot Data Software Development
1021 Lincoln Boulevard
Santa Monica, CA 90403
(213) 393-6405

Popular Games Released

Atari, Inc., has released 12 of its games for competing computers and videogame consoles in a new line of software called ATARISOFT.

The initial entries in the new line include *Centipede*, *Defender*, *Dig-Dug*, *Donkey Kong*, *Pac-Man, Robotron, Stargate, Picnic Paranoia, Protector, Shamus, Super Storm,* and *Galaxian.*

The first seven games will run on the Apple II and IIe, IBM PC, Commodore 64, VIC-20, and Texas Instruments 99/4A computers. The remaining games, except *Galaxian,* will operate on the Texas Instruments machine. *Galaxian, Centipede,* and *Defender* will run on ColecoVision, while *Centipede, Defender,* and *Pac-Man* will operate on Intellivision.

The computer games carry suggested retail prices of $34.95 for disks and $44.95 for cartridges. Suggested retail for the ColecoVision and Intellivision versions is $40.95.

Atari Incorporated
1265 Borregas Avenue
P.O. Box 427
Sunnyvale, CA 94086
(408) 743-4810

War Strategy Games For Apple, Atari, 64

Strategic Simulations Inc. has produced several war strategy games for the Apple, Atari, and Commodore 64 computers.

Carrier Force is a simulation of the four major aircraft carrier battles fought in the Pacific during World War II—Coral Sea, Midway, Eastern Solomons, and Santa Cruz. Every major warship and plane is accounted for and rated in the game. Weather, time of day, visibility, inaccurate sightings, and other aspects of actual naval warfare conditions are recreated.

The game comes with two maps, rule book, and 48K disk for the Apple II with Applesoft ROM, II+, IIe, and III. It is also available on 40K disk for the Atari 400/800/1200. The price is $59.95.

Battle for Normandy is a simulation of the D-Day invasion and the 24 days that followed. Now available for the Commodore 64 computer in a 64K disk and a cassette format, the game includes all historical details. The player may be the Supreme Allied Commander or commander of the German defenses. A solitaire option is available, with the computer directing the German defenses.

The game comes with two player-aid cards, maps and rule book. Versions are also available for Atari, TRS-80 models I and III, Apple, and IBM PC computers. *Battle for Normandy* is priced at $39.95.

Tigers in the Snow is a recreation of the World War II Battle of the Bulge, Germany's last major counteroffensive. The game is now available for the Commodore 64 computer.

In the simulation, the Germans attempt to destroy the Allies' stronghold while the outnumbered Allies try to hold until their reinforcements arrive. The forces are division/regiment scale. Both sides command infantry, artillery, and air power.

The price for the game is $39.95 on 64K disk for the Commodore 64. It is also available for the Apple, Atari, and TRS-80 computers.

Strategic Simulations Inc.
883 Stierlin Road, Building A-200
Mountain View, CA 94043
(415) 964–1353

Color Graphics Packages For 64

Sophisticated Software of America has released *Grafix-Artist,* a high-resolution color-graphics package for the Commodore 64, and *Grafix-Printer,* a high-resolution graphics printer-dump for use with printers/interfaces which emulate the Commodore 1525E printer.

Grafix-Artist was created to introduce children and adults to computer graphics and to programming by allowing them to use either a specially designed language or a joystick to control the onscreen graphics capabilities. The product can be used to create detailed screens which can then be included with other programs written in either BASIC or machine language. Design layouts, storyboarding, and ad displays are among the uses of *Grafix-Artist.*

Available on disk, *Grafix-Artist* has a price of $39.95. *Grafix-Printer* is available for $29.95.

Another Sophisticated Software product, *Lesson Designer,* which allows free-form lesson design, will be available this spring.

Sophisticated Software of America
198 Ross Road
King of Prussia, PA 19406
(215) 265-2277

Educational Games On Disk

Unicorn Software has produced four educational games available on disk for the Atari, Commodore 64, IBM PC, and Apple computers.

Ten Little Robots is composed of five learning games that introduce children, from ages two to seven, to the computer. They include Little Robot Story, an interactive nursery tale that presents the concept of subtraction and aids in reading preparation; Robot Letter Match, which teaches upper- and lowercase letters; Count The Robots; Robot Addition; and Robot Sketch.

Ships Ahoy is a series of four games for children from 5 through 13 years of age. These four games include Ships Ahoy and Mine Sweeper, two math skills programs; Treasure Hunt, a maze game included as a reward; and Sailing Sketch, a screen painting program.

Race Car 'Rithmetic, for those five years old to adult, is a math game designed for the entire family to play together. An action race game is included as a reward.

Funbunch is a language arts program available on three levels—*Elementary* (grades 1–6), *Intermediate* (junior high school), and *College Prep* (high school to adult). Each level includes over 2000 words and phrases with which to work.

The four games sell for $39.95 each.

Unicorn Software
Liberace Plaza, Suite 8
1775 East Tropicana Avenue
Las Vegas, NV 89109
(702) 798-2727

Word Processing Program

HomeWord, a word processing program produced by Sierra On-Line, has been introduced on disk for the Apple, Commodore 64, and Atari computers.

The program uses icons, or symbols, as a guide to the writer. There are six major options with appropriate icons that are shown on the screen—filing (a file cabinet), editing (a page of print), printing (a printer), format design (an unorganized page with an arrow pointing to an organized page), customizing (a question mark), and disk utilities (a floppy disk). Within each of these categories there are subcategories, also with icons.

Another feature of the program is the ability to divide the screen into three sections. The upper and largest portion reveals the working text, while the lower right side provides a replica of the entire page as it will be printed. To the left of this insert is a chart which keeps the user appraised of available memory and disk space.

Other standard features include optional joystick control, an outline format, boldface and underlining, text block movement, file merging, headers and footers, and other functions.

HomeWord is priced at $49.95.

Sierra On-Line, Inc.
36575 Mudge Ranch Road
Coarsegold, CA 93614
(209) 683-6858

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

COMPUTE! *welcomes notices of upcoming events and requests that the sponsors send a short description, their name and phone number, and an address to which interested readers may write for further information. Please send notices at least three months before the date of the event, to: Calendar, P.O. Box 5406, Greensboro, NC 27403.* ©

COMPUTER MAIL ORDER

APPLE/FRANKLIN DISK DRIVES

MICRO-SCI

A2	$219.00
A40	$299.00
A70	$319.00
C2 Controller	$79.00
C47 Controller	$89.00

RANA

Elite 1	$279.00
Elite 2	$389.00
Elite 3	$569.00

APPLE IIe STARTER PACK
64K Apple IIe, Disk Drive & Controller, 80 Column Card, Monitor II & DOS 3.3
COMPLETE $1199.00

FRANKLIN

ACE 1000 Color Computer	CALL
ACE Family Pack System	CALL
ACE PRO PLUS System	CALL
ACE 1200 Office Mgmt. System	CALL

"NOT IIE EXPENSIVE"

ATARI HOME COMPUTERS

600XL $189
800XL $299
1200XL .... CALL
1400XL .... CALL

1010 Recorder	$74.00
1020 Color Printer	$249.00
1025 Dot Matrix Printer	$449.00
1027 Letter Quality	$299.00
1030 Direct Connect Modem	CALL
1050 Disk Drive	$379.00
CX30 Paddle	$12.00
CX40 Joystick ... each	$8.00
CX77 Touch Tablet	$64.00
CX80 Trak Ball	$48.00
CX85 Keypad	$105.00
488 Communicator II	$229.00
4003 Assorted Education	$47.00
4011 Star Raiders	$33.00
4012 Missile Command	$29.00
4013 Asteroids	$29.00
5049 VisiCalc	$159.00
7097 Logo	$79.00
7101 Entertainer	$69.00
7102 Arcade Champ	$75.00
8026 Dig Dug	$33.00
8030 E.T. Phone Home	$33.00
8031 Donkey Kong	$39.00
8033 Robotron	$35.00
8034 Pole Position	$39.00
8036 Atari Writer	$79.00
8040 Donkey Kong, Jr.	$39.00
8043 Ms. PacMan	$39.00
8044 Joust	$39.00

PERCOM

AT 88-S1	$369.00
AT 88-A2	$259.00
AT 88-S1PD	$429.00
AT 88-DDA	$119.00
RFD 40-S1	$449.00
RFD 40-A1	$269.00
RFD 40-S2	$699.00
RFD 44-S1	$539.00
RFD 44-S2	$869.00

TEXAS INSTRUMENTS

TX 99-S1	$279.00

RANA

1000	$319.00

TRAK

AT-D2	$389.00

MEMORY BOARDS

Axlon 32K	$59.00
Axlon 48K	$99.00
Axlon 128K	$299.00
Intec 32K	$59.00
Intec 48K	$85.00
Intec 64K	$99.00
Intec Real Time Clock	$29.00

ALIEN VOICE BOX

Atari	$119.00
Apple	$149.00

KOALA PAD

Atari	$75.00
Apple	$85.00
IBM	$95.00
CBM 64	$75.00

CONTROLLERS & JOYSTICKS

WICO

Joystick	$21.99
3-way Joystick	$22.99
Famous Red Ball	$23.99
Power Grip	$21.99
BOSS Joystick	$17.99
ATARI/VIC Trak Ball	$34.99
Apple Trak Ball	$54.99
Apple Adapter	$15.99
Apple Analog	$37.99

KRAFT

Joystick	$41.99
Atari Single Fire	$12.99
Atari Switch Hitter	$15.99
Apple Paddles	$34.99
IBM Paddles	$34.99
IBM Joystick	$46.99

AMIGA

3100 Single	$13.99
3101 Pair	$19.99
Joyboard	$37.99

TG

Atari Trak Ball	$47.99
Apple Joystick	$47.99
Apple Trak Ball	$47.99

commodore

CBM 8023 ... $599

CBM 4032	$599.00
CBM 8096	$869.00
CBM 9000	$999.00
B128-80	$769.00
CBM 64K Memory Board	$269.00
8032 to 9000 Upgrade	$269.00
2031LP Disk Drive	$299.00
8050 Disk Drive	$949.00
8250 Disk Drive	$1199.00
4023 Printer	$379.00
8023 Printer	$569.00
6400 Printer	$1399.00
Z-RAM	$499.00
Silicon Office	$699.00
The Manager	$199.00
Soft ROM	$125.00
VisiCalc	$159.00

PROFESSIONAL SOFTWARE

Word Pro 2 Plus	$159.00
Word Pro 3 Plus	$189.00
Word Pro 4 Plus/5 Plus ... each	$279.00
InfoPro	$179.00
Administrator	$399.00
Power	$79.00

SX-64 PORTABLE $869.00

VIC 20 .... CALL
CBM 64 .... $199

MSD SD1 Disk Drive (C-64)	$349.00
C1541 Disk Drive	$249.00
C1530 Datasette	$69.00
C1520 Color Printer/Plotter	$169.00
M-801 Dot Matrix Printer	$219.00
C1526 Dot Matrix/Serial	$279.00
C1702 Color Monitor	$249.00
C1311 Joystick	$4.99
C1312 Paddles	$11.99
C1600 VIC Modem	$59.00
C1650 Auto Modem	$89.00
Logo 64	$49.00
Pilot 64	$39.00
Simon's Basic	$19.00
Word Pro 64 Plus	$59.00
Parallel Printer Interface	$49.00
Calc Result 64	$129.00
Codewriter 64	$75.00
Quick Brown Fox	$49.00
Word Pro 64 Plus	$59.00

We stock a full inventory of software for Commodore, such as: Artworx, Broderbund, Commercial Data, Creative Software, Epyx, HES, MicroSpec, Nufekop, Romox, Sirius, Synapse, Thorn EMI, Tronix, UMI, Victory, Spinnaker, Rainbow & Timeworks!
CALL FOR DETAILS!

DISKETTES

MAXELL

5¼" MD-1	$29.00
5¼" MD-2	$39.00
8" FD-1 (SS/DD)	$39.00
8" FD-2 (DS/DD)	$49.00

VERBATIM

5¼" SS/DD	$26.99
5¼" DS/DD	$36.99

ELEPHANT

5¼" SS/SD	$18.49
5¼" SS/DD	$22.99
5¼" DS/DD	$28.99

HEAD

5¼" Disk Head Cleaner	$14.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Flip-n-File 10	$3.99
Flip-n-File 50	$17.99
Flip-n-File (400/800ROM) Holder	$17.99

LJK ENTERPRISES

Atari Letter Perfect-Disk (40/80)	$79.99
Atari Letter Perfect-ROM (40 col)	$79.99
Atari Letter Perfect-ROM (80 col)	$79.99
Atari Data Perfect-ROM (80 col)	$79.99
Atari Spell Perfect-DISK	$59.99
Atari Utility/MailMerge	$21.00
Apple Letter Perfect	$99.00
Apple Data Perfect	$75.00
Apple LJK Utility	$21.00
Apple Lower Case Generator	$19.00

HANDHELD COMPUTERS

hp

41CX $249.99
41CV $199.99
41C $144.99

HP 10C	$51.99
HP 11C	$69.99
HP 12C	$88.99
HP 15C	$88.99
HP 16C	$88.99
HP 75C	$749.99
HPIL Module	$98.99
HPIL Cass. or Printer	$359.99
Card Reader	$143.99
Extended Function Module	$63.99
Time Module	$63.99

NEC

PC-8201 Personal Computer	$599.00
PC-8221A Thermal Printer	$149.00
PC-8281A Data Recorder	$99.00
PC-8201-06 8K RAM Chips	$105.00
PC-8206A 32K RAM Cartridge	$329.00

SHARP

PC-1500A ... $165.99
PC-1250A $88.99

CE-125 Printer/Cassette	$128.99
CE-150 Color Printer/Cassette	$171.99
CE-155 8K RAM	$93.99
CE 161 16K RAM	$134.99
CE 500 ROM Library	$29.99

TIMEX/SINCLAIR

Timex/Sinclair 1000	$24.99
Timex/Sinclair 2086	CALL
16K Memory	$25.00
2040 Printer	$99.99
VuCalc	$17.99
Mindware Printer	$99.99

WEST
800-648-3311
In NV call (702)588-5654, Dept 0406
Order Status Number: 588-5654
P.O. Box 6689, Stateline, NV 89449

CANADA
800-268-4559
In Toronto call (416)828-0866 Dept 0406
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 1B
Mississauga, Ontario, Canada L5L1T1

EAST
800-233-8950
In PA call (717)327-9575 Dept 0406
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. Third St., Williamsport, PA 17701

VISA

CANADIAN ORDERS: All prices are subject to shipping, tax and currency fluctuations. Call for exact pricing in Canada.
INTERNATIONAL ORDERS: All shipments outside the continental United States must be pre-paid by certified check only. Include 3% (minimum $5.00) shipping and handling. EDUCATIONAL DISCOUNTS: Additional discounts are available to qualified Educational Institutions.
APO & FPO: Add 3% (minimum $5.00) shipping and handling.

COMMODORE 64 SOFTWARE

Products are Noted [D]-Disk, [C]-Cartridge or [T]-Cassette Tape

Access
Beach-Head .... (D)26.95
Neutral Zone .... (D)26.95

Broderbund
Choplifter .... (C)27.00
Sea Fox .... (C)27.00
Serpentine .... (C)27.00
David's Midnight Magic .... (D)27.00

CDS
Frantic Freddy .... (D)26.95
Pegasis .... (D)26.95

Commodore
Assembler 64 .... (D)16.90
Bonus Pack .... (D,T)16.90
CP/M 2.2 System .... (C)69.00
Easy Script .... (D)43.00
Easy Mail 64 .... (D)16.90
Easy Spell 64 .... (D)16.90
Easy Calc 64 .... (D)73.75
Easy Lesson/Quiz .... (D)16.90
Easy Finance(1-5) .... (D)19.35
Logo .... (D)47.90
Pilot .... (D)47.90
Pet Emulator .... (D)16.90
Intro to Basis 1 .... (T)19.35
Screen Editor .... (D)16.90
Super Expander VSP .... (D)19.50
The Manager .... (D)43.00
Simons Basic .... (C)19.50
Music Machine .... (C)16.90
Music Composer .... (D)16.90
Word/Name Machine .... (D)16.90
Zortek and the Microchips .... (T)24.80
Inventory Mgmt. .... (D)43.00
Payroll Checkwriting .... (D)43.00
Accts. Payable .... (D)43.00
Accts. Receivable .... (D)43.00
Gen. Ledger .... (D)43.00
Nevada Cobol .... (D)59.95
Magic Desk .... (C)74.95
Zork(1-3) .... (D)29.50
Suspended .... (D)29.50
Starcross .... (D)29.50
Deadline .... (D)29.50
Dragons Den .... (C)24.95
Star Ranger .... (C)24.95

Continental
Home Accountant .... (D)50.75
Tax Advantage .... (D)40.60
F.C.M. .... (D)76.50

Creative Software
Spitball .... (C)23.65
Pipes .... (C)23.65
Decision Maker .... (T)10.15(D)13.50
Household Finance .... (T)13.50(D)16.90
Home Inventory . (T)10.15(D)13.50
Loan Analyzer .. (T)10.15(D)13.50
Moondust .... (C)23.65
Astroblitz .... (C)23.65
Trashman .... (C)23.65
Save New York .... (C)23.65

Cyber
Cyberchess .... (D) 34.95

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

Datasoft
Moonshuttle .... (D,T)20.30
Poo Yan .... (D,T)20.30
Genesis .... (D)27.00
O'Riley's Mine .... (D)20.30
Dallas Quest .... (D)27.00

Epyx
Pitstop .... (C)27.00
Temple of Apshai .... (D)27.00
Upper Reaches of Upshai .... (D)13.55
Curse of Ra .... (D)13.55
Sword of fargoal .... (D)20.30
Jumpman .... (D,T)27.00
Gateway to Apshai .... (D)27.00
Jumpman Jr. .... (C)27.00
Oil Baron .... (D)27.00
Lunar Outpost .... (D)27.00

Finesse
Global Risk .... (D)34.95

Hes Software
Oubliette .... (D)20.45
The Factory .... (D)24.25
Missing Links .... (D)20.45
Type 'N Write .... (C)20.45
Paint Brush .... (C)17.00
Hes Mon 64 .... (C)27.00
Turtle Graphics II .... (C)40.60
Attack of the Mutant Camels .... (C)20.45
Time Money Mgr. .... (D)51.00
Omniwriter .... (D)24.20
Synthesound 64 .... (D)25.50
Omni Calc .... (D)58.25
Multiplan .... (D)79.95
Hes Writer 64 .... (C)30.45
Gridrunner .... (C)20.25
Benji Space Rescue .... (D)30.45
The Pit .... (C)20.25
64 Forth .... (C)49.95
Turtle Toyland Jr. .... (D,T)23.95
Chameleon .... (D)17.00
Ghost Manor/Spikes Peak .... (D)17.00
HES Games '84 .... (D)27.25
Minnesota Fats Pool .... (C)20.45

Odesta
Chess 7.0 .... (D)69.95
Checkers 2.1 .... (D)49.95
Odin .... (D)49.95
How About...Chess .... (D)29.95

Micro Prose
Floyd of the Jungle .... (D)23.95
Hellcat Ace .... (D)23.95
Solo Flight .... (D)26.95
Nato Commander .... (D)26.95

Sierra on Line
Frogger .... (D,T)23.65
New Jawbreaker .... (D)20.30
Sammy Lightfoot .... (D)21.80
Threshold .... (D)21.80
Lunar Leeper .... (D)21.80
B.C. Quest for Tires .... (D)25.50
Oil's Well .... (D)21.80
Learning W/Leeper .... (D)21.80

Sirus Software
Turmoil .... (D)27.00
Snake Byte .... (D)27.00
Squish 'Em .... (D)23.65
Repton .... (D)27.00
Blade of Blackpoole .... (D)27.00
Type Attack .... (D)27.00
Way Out .... (D)27.00
Critical Mass .... (D) 27.00

subLogic
Flight Simulator II .... (D)44.95

Spinnaker
Alf in the Color Caves .... (C)27.95
Bubble Burst .... (C)27.95
Juke Box .... (C)27.95
Kidwriter .... (D)23.95
Face Maker .... (D,C)23.65
Hey Diddle Diddle .... (D)20.30
Kindercomp .... (C,D)20.30
Snooper Troops .... (D)27.00
Fraction Fever .... (C)23.65
Amazing Thing .... (D)27.00
Alphabet Zoo .... (D)23.65
Kids on Keys .... (C,D)23.65
Delta Drawings .... (C)27.25
Up for Grabs .... (C) 27.25
Cosmic Life .... (C) 23.65
Story Machine .... (C)27.25
Trains .... (D)27.95
Aerobics .... (D)30.95

Scarborough
Songwriter .... (D)35.95
Master Type .... (D)35.95

Synapse
Ft. Apocalypse .... (D,C)23.70
Shamus .... (D,C)23.70
Blue Max .... (D,C)23.70
Dimension X .... (D,C)23.70

SSI
Combat Leader .... (D)29.00
Knights of the Desert .... (D)29.00
Computer Baseball .... (D)29.00
Eagles .... (D)29.00
Ring Side Seat .... (D)29.00

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

CALL FOR CURRENT PRICES

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

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

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

Koala
Touch Tablet (D)79.95

GEMINI Printers
O'SULLIVAN Furniture
MAXTRON Datasettes

CALL FOR INFORMATION

Terms and Ordering Information: To order call 1-800-527-8698 and send certified checks, money orders or personal checks [allow 2 weeks to clear], or use your Visa, Master Card or American Express. Inside Texas call 1-800-442-8717. Include $2 for P & H [C.O.D. orders add $1.65] UPS Blue Label $4, Canada $6 call for shipping charges on Hardware. Other countries include 10% for P & H. All products factory sealed with manufacturer's warranty. All returns require R.A.#. Prices subject to change without notice. Order desk hours: Mon-Fri 9-6, Sat 9-1 CST.

ATARI, APPLE, VIC 20, SOFTWARE LISTS AVAILABLE
For further information and inquires call 214-753-0485
FOR ORDERS ONLY CALL TOLL FREE

Outside Texas **1-800-527-8698** Inside Texas **1-800-442-8717**

SoftWare Warehouse outlet

P.O. BOX 2511 LONGVIEW, TX. 75606

INSTANT SUCCESS WITH INSTANT TUTOR!*

For IBM PC

Learn how to use your own Personal Computer (PC), WordStar, SuperCalc or dBase II with Instant Tutor!™

You can learn faster and easier than you believed possible—even if you have never used a computer before. Instant Tutor!™ is a quick and easy, step-by-step **computer assisted** learning package that contains a Reference Guide and diskette. The **Easy Reference Guide** shows you how to turn your computer on and insert the **diskette.** From this point on, you begin to interact with your PC. The diskette teaches you how to use the software of your choice by giving you practical information and **real-life** exercises. At any time, you can turn to the Easy Reference Guide for lesson reviews and practice problems.

***Instant Tutor!™ is your Interactive Trainer.**

Instant Tutor!™ is a trademark of Data Processing Educational Corp. WordStar is a registered trademark of MicroPro; dBase II of Ashton-Tate; SuperCalc of Sorcim.
© 1984 Data Processing Educational Corp.

SATISFACTION GUARANTEED!

Instant Tutor!™ will save you days of frustration, indecision and mistakes. If you are not completely satisfied, return it to us within 10 days for a complete and courteous refund. You take no risk.

Credit Card orders only call
(800) 227-1617 EXT. 115

In California call
(800) 772-3545 EXT. 115

Specify which course you wish to learn, **and** type of diskette required. $59.95 per Course plus $2.50 shipping and handling. Ohio residents add sales tax. Visa, MasterCard, AmX, money orders and personal checks accepted.

Data Processing Educational Corp.
4588 Kenny Road, Dept. 22
Columbus, OH 43220

FLY Your Computer

FLIGHT SIMULATOR GAMES

NEW **COCKPIT 64**
For the Commodore 64
- 100% Machine Language
- Windshield View
- 7 Airports

$30.

Runway 64 (Commodore 64)	$25
Runway 20 (VIC-20)	$25
Sky Pilot (VIC-20)	$18

ADD $2.00 FOR DISK VERSION

BAR GRAPH 64
FOR THE COMMODORE 64
- 100% Machine Language
- Full Color Display
- Saves To Disk
- Supports All Parallel Printers*

$30.

COD ORDER PHONE
WE SHIP WITHIN 48 HOURS
(312) 394-5165

SUSIE SOFTWARE
709 Wilshire Dr.
Mt. Prospect, IL 60056

ATARI® NEW LOWER PRICES!!

ATARI is a trademark of ATARI, INC.

ATARI 800XL .... Call

Atari 1050 Disk Drive	Call
Atari 1010 Recorder	77.00
Atari 1027 Printer	Call
Atari 850 Interface	Call

MOSAIC

64K Ram/400	149.00
64K Ram +Cable Kit 400/800	169.00
48K Ram Kit	94.00
16/32 Expander	64.95
32K Ram	77.95
Mosaic Adaptor	49.95

Memory Expansion for Atari 600XL .. Call

3RD Party Printer Interfaces

Interfast I	139.00
Apeface w/cable included	69.95

PRINTERS

Axiom AT-100	229.00
Axiom AT-550	329.00
Riteman	Call
Gemini 10X	Call
Alphacom	Call
Okidata 92A	Call

Call for assorted Printer Packages

DISK DRIVES

Percom, Trak, Concorde, Indus GT .... Call For Prices

RS232 Modem Adaptor (Through serial port) 39.95

Koala Touch Tablet Cart or Disk ... 69.95

MONITORS

USI	Call
AMDEK	Call

STIMUTECH SUBLIMINAL SOFTWARE FOR ATARI AND CBM 64

Expando-Vision Interface w/one **Free** Cart 99.00

- Weight Control
- Study Habits
- Stress Control
- Drinking Control
- Smoking Control
- Career Success
- Sexual Confidence

Addit'l. Rom Carts $29.95 ea.

A T A R I   S O F T W A R E

ATARI

Conversational Lang. - T	44.95
My First Alphabet - D	26.95
Touch Typing - T	19.95
Home Filing Mgr. - D	37.95
Star Raiders - Cart	32.95
Assembly Editor - Cart	46.95
Macroassembler - D	67.95
Invitation to Program I T	19.95
Basketball - Cart	26.95
Graph-it - T	15.95
Qix - Cart	32.95
Dig Dug - Cart	32.95
Atariwriter - Cart	74.95
Donkey Kong - Cart	37.95
Ms. Pac-Man - Cart	39.95
Tennis - Cart	35.95
Eastern Front - Cart	32.95
Donkey Kong Jr. - Cart	39.95
Pengo - Cart	35.95
Logo - Cart	79.95
Robitron - Cart	35.95
Pole Position - Cart	39.95
Microsoft Basic II - C	67.95
Paint - D	33.95
Caverns of Mars - Cart	32.95
Joust - Cart	39.95
Visicalc - D	159.95

SIERRA ON-LINE

Homeword - D	49.95
Dark Crystal - D	27.95
Frogger - D/T	23.95
Quest For Tires - D	23.95
Ultima I - D	23.95
Ultima II - D	41.94
Wizard/Princess - D	22.95

INFOCOM

Deadline - D	34.95
Witness - D	34.95
Zork I, II, III - D	27.95
Starcross - D	27.95
Planetfall - D	34.95
Suspended - D	34.95
Enchanter - D	34.95
Infidel - D	34.95
Sorcerer - D	34.95

BRODERBUND

AE - D	23.95
Arcade Machine - D	41.95
Bank St. Writer - D	49.95
Lode Runner - D	23.95
Drol - D	23.95
Spare Change - D	23.95
Choplifter - D	23.95

PARKER BROS.

Astrochase - Cart	34.95
Frogger - Cart	34.95
Q-Bert - Cart	34.95
Popeye - Cart	34.95

EPYX

Gateway To Apshai - Cart	27.95
Pitstop - Cart	27.95
Swat Rescue - Cart	27.95
Silicon Warrior - Cart	27.95
Fun With Music - Cart	27.95
Jumpman Jr. - Cart	27.95
Dragonriders of Pern - D/T	27.95
Temple of Apshai - D/T	27.95
Hellfire Warrior - D/T	20.95
Fun With Art - Cart	27.95

DATA SOFT

Money Wizard - D	49.95
Letter Wizard - D	49.95
Spell Wizard - D	34.95
Genesis - D/T	20.95
Heathcliff - D/T	20.95
Pooyan - D/T	20.95
Zaxxon - D/T	27.95
Micropainter - D	23.95

SPINNAKER

Up For Grabs - Cart	27.95
Cosmic Life - Cart	23.95
Facemaker - Cart	23.95
Alphabet Zoo - Cart	23.95
Fraction Fever - Cart	23.95
Kids on Keys - Cart	23.95
Delta Drawing - Cart	27.95
Story Machine - Cart	27.95
Rhymes & Riddles - D	20.95
Amazing Things - D	27.95
Hey Diddle Diddle - D	20.95
Snooper Troops 1 or 2 - D	30.95
Trains - D	27.95
Aerobics - D	30.95

LJK

Letter Perfect - D	74.95
Data Perfect - D	74.95
Spell Perfect - D	64.95

MISCELLANEOUS FOR ATARI

Castle-Wolfenstein - D	20.95
Home Accountant - D	52.95
Master Type - D/Cart	27.95
Spelunker - D	27.95
Flight Simulator II - D	37.95
Zombies - D/T	23.95
Mr. Robot - D	23.95
River Raid - Cart	31.95
Diskey - D	34.95
Chatterbee - D	27.95
Rally Speedway - Cart	34.95
Ultima III - D	41.95
Miner 2049'er - Cart	34.95
Scraper Caper - Cart	34.95
Basic XL - Cart	74.95
Monkey Wrench II - Cart	49.95
Omnimon	82.95
Star Trek - Cart	27.95
Starbowl Football - D/T	22.95
Hockey - Cart	27.95
Computer War - Cart	27.95
Flip Flop - D/T	20.95
Boulder Dash - D/T	20.95
Carrier Force - D	41.95
Combat Leader - D/T	27.95
Encounter - D/T	23.95
Quasimoto - D/T	23.95
N.Y.C. - D/T	23.95
Compute Books	Call

Hundreds of items available for the Atari and CBM 64 ... please call.

COMMODORE 64

CBM 64	**Call**
1541 Disk Drive	**249.00**
1525 80 Column Printer	**229.00**
1530 Datasette	**66.00**
1702 Color Monitor	**249.00**
1650 AD/AA Modem	**89.00**
RS 232 Interface	**44.00**

COMMODORE 64 SOFTWARE

Program Ref. Guide	19.95
Assembler - D	17.95
Easy Finance I, II, III, IV - D	19.95
Easy Calc - D	64.95
Easy Mail - D	17.95
Easy Script - D	39.95
Easy Spell - D	19.95
Logo - D	39.95
The Manager - D	37.95
General Ledger -	37.95
Accts. Rec. - D	37.95
Accts. Pay. - D	37.95
Code Writer - D	37.95
Zork I, II or III D	29.95
Suspended - D	29.95
Starcross - D	29.95
Deadline - D	29.95

C O M M O D O R E   6 4   S O F T W A R E

ATARISOFT

Pac-Mac	34.95
Centipede	34.95
Defender	34.95
Dig Dug	34.95
Donkey Kong	34.95
Stargate	34.95
Robotron	34.95

ACCESS

Neutral Zone - D/T	26.95
Spritemaster - D/T	27.95
Beachhead - D/T	26.95

BLUE SKY

Calc Result/Easy - Cart	Call
Calc Result/Advanced - D	Call
Script 64 - D	Call

BATTERIES INCLUDED

Paper Clip	99.95
Delphi Oracle	114.95

DATA 20

Video Pak 80	139.95
Z80 Video Pack	229.95

HES

Multiplan - D	74.95
64 Forth - Cart	41.95
HES Writer - Cart	32.95
HES Mon - Cart	27.95
Turtle Graphics - Cart	41.95
Mutant Camels	20.95
HES Modem	59.95

CIMMARON

Instaspeed - D/T	74.95
Instawriter - Cart	55.95
Instamail - D/T	29.95

SIRIUS

Blackpoole - D	27.95
Repton - D	27.95
Type Attack - D	27.95
Wayout - D	27.95
Critical Mass - D	27.95

SYNAPSE

Blue Max - D/T	23.95
Zaxxon - D/T	23.95
Morgul - D/T	23.95
Shamus II - D/T	23.95
Sentinal - D/T	23.95
Pharoah's Curse - D/T	23.95
Slamball - D/T	23.95

MISCELLANEOUS FOR CBM 64

Flight Simulator II - D	37.95
Night Mission/Pinball - D	20.95
Praticalc PS - D	59.95
M-File - D	84.95
Word Pro 3+/Spell D	74.95
Home Accountant - D	52.95
Step by Step - D/T	44.95
Barron's Sat - D	59.95
Money Manager - D/T	19.95
Bristles - D/T	20.95
Telestar 64 - Cart	37.95
Star League Baseball - D/T	22.95
Zombies - D/T	23.95
Castle Wolfenstein - D	20.95
Swashbuckler - D	23.95
Sorcerer - D	34.95
Computer Baseball - D	27.95
Ringside Seat - D	27.95
Professional Golf - D	27.95
Congo Bongo - Cart	27.95
Mastertype - Cart	27.95
Solo Flight - D	23.95
Vic Switch	124.95
Cardco + G	64.95

ACCESSORIES

WICO Joystick	Call
Flip'n' File Trunks	20.95
Flip'n' File Cart	20.95
Joysensor	24.95
Elephant Disks (Box of 10)	20.95
WICO Trakball	37.95
KRAFT Joystick	15.95

D - Disk
T - Cassette
Cart - Cartridge

ComputAbility™

To Order Call Toll Free
800-558-0003

For Technical Info, Order Inquiries, or for Wisc. Orders -
414-351-2007

NO SURCHARGE FOR MASTERCARD OR VISA

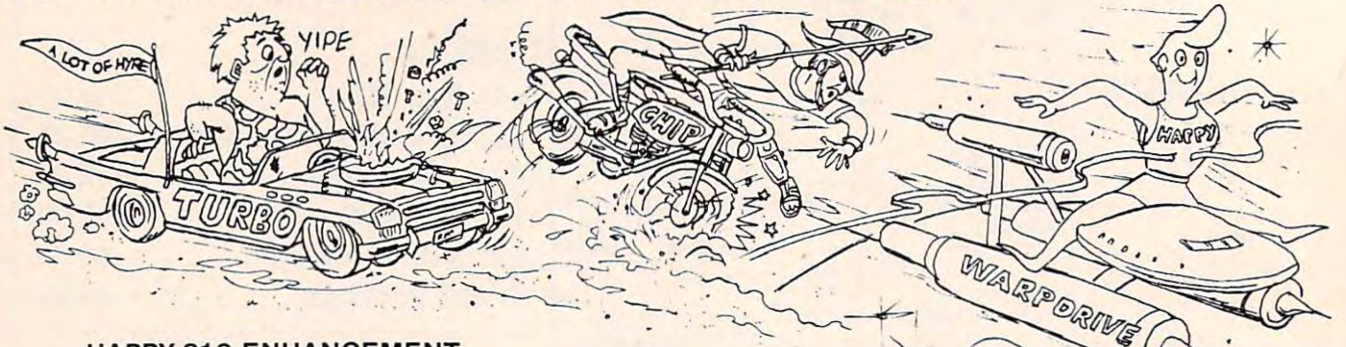
COMPUTABILITY
P.O. Box 17882
Milwaukee, WI 53217

MasterCard   VISA

ORDER LINES OPEN
Mon-Fri 11 AM - 7 PM CST
Sat 12 PM - 5 PM CST

ORDERING INFORMATION For fast delivery send cashier's check, money order or direct bank transfers Personal and company checks allow 2 weeks to clear. Charges for COD are $3.00. School Purchase Orders welcome. In CONTINENTAL USA, include $3.00 shipping per software order. Include 3% shipping on all Hardware orders, minimum $3.00. Mastercard & Visa please include card # and expiration date. WI residents please add 5% sales tax. HI, AK, FPO, APO, Canadian orders - add 5% shipping, minimum $5.00. All other foreign orders, please add 15% shipping, minimum $10.00. All goods are new and include factory warranty. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call 414-351-2007 to obtain an RA# or your return will NOT be accepted for replacement or repair. Prices and availability are subject to change without notice.

HAPPY WINS THE RACE WITH WARP DRIVE SPEED!



HAPPY 810 ENHANCEMENT

- The only change needed to run all WARP DRIVE software
- Plug in P.C. board requires no permanent modifications
- Proven reliable in thousands of installations, reduces disk drive wear
- Comes completely assembled and tested, just plug in and use
- Full one year parts and labor guarantee, compatible with existing software
- High quality printed circuit board with gold connectors

NO ONE ELSE HAS THIS PERFORMANCE
Unenhanced whole disk (ATARI rev B format) read time: 112 seconds
Unenhanced whole disk (ATARI rev C fast format) read time: 89 seconds
ENHANCED 810 whole disk (any format) read time with standard software: 68 seconds
ENHANCED 810 whole disk (any format) read time with WARP DRIVE software: 43 seconds
Standard software whole disk write and verify time: 238 seconds
WARP DRIVE software whole disk write and verify time: 62 seconds

NEW HAPPY WARP DRIVE SOFTWARE

WARP SPEED HAPPY BACKUP PROGRAM
- Completely automatic: nothing to figure out, insert disks and press return
- Only program on the market guaranteed to backup any disk
- Can write to a blank disk: format write and verify in one operation
- Automatic program tracing: copies only the tracks that are used
- Efficient memory utilization: reduces the number of disk insertions
- Requires only one ENHANCED disk drive, backups will work on a standard drive

WARP SPEED MULTI DRIVE HAPPY BACKUP PROGRAM
- Same features as above plus support of multiple ENHANCED drives
- Can be used with up to 4 ENHANCED drives
- Source and all destination drives read and write in parallel
- Format write and verify 3 complete disks in less than 3 minutes

WARP SPEED HAPPY COMPACTOR PROGRAM
- Reduces the number of disks required to backup your library
- Combines up to 8 self booting disks into 1 disk with a menu
- Compacted disks run only on an ENHANCED drive
- Pays for itself by saving on disks
- Single or dual ENHANCED drive operation

HAPPY WARP DRIVE DOS
- Improves ATARI DOS 2.0S to use warp speed reading and write with verify
- Use all features of BASIC, PILOT, FMS, and DUP at top warp speed
- Warp speed I/O software module available separate from DOS

HAPPY WARP DRIVE SECTOR COPY PROGRAM
- Standard format whole disk read, write and verify in 105 seconds
- Use with sngle or dual drives, mix ENHANCED and NON-ENHANCED drives

HAPPY CUSTOMIZER PROGRAM (sold separately $99.95)
- Creates custom format disks of any specification
- Any type bad sector, duplicate sector numbers, or interleave
- Easy to use but requires an advanced level user to interpret the results

REVIEWED IN POPULAR MAGAZINES

A.N.A.L.O.G. COMPUTING—July/August 1983 *"...The installation instructions for the Happy 810 Enhancement are among the best I have ever seen. ...The Happy 810 Enhancement is one of the most powerful hardware modifications available to ATARI computer owners."*

ANTIC—July 1983 *"The difference between a normal ATARI 810 disk drive and one equipped with Happy is like the contrast between mass transit and the automobile. A car costs you more initially, but improves the quality of your life. Similarly, if you use your disk drive a lot, installing Happy will markedly enhance your programming life."*

SPECIAL SUGGESTED RETAIL PRICE BEFORE FEBRUARY 28, 1984: *Get the HAPPY 810 ENHANCEMENT with the single and multi drive HAPPY BACKUP PROGRAM, plus the HAPPY COMPACTOR PROGRAM, plus the HAPPY DRIVE DOS, plus the HAPPY SECTOR COPY, all with WARP DRIVE speed, including our diagnostic for $249.95. Existing registered ENHANCEMENT owners may upgrade to WARP DRIVE speed for $15.00 with no hardware changes!*

Price includes shipping by air mail to U.S.A. and Canada. Foreign orders add $10.00 and send an international money order payable through a U.S.A. bank. California orders add $16.25 state sales tax. Cashiers check or money order for immediate shipment from stock. Personal checks require 2-3 weeks to clear. Cash COD available by phone order and charges will be added. No credit card orders accepted. ENHANCEMENTS for other ATARI compatible drives coming soon, call for information. Please specify -H model for all drives purchased new after February 1982, call for help in ENHANCEMENT model selection. Dealers now throughout the world, call for the number of the dealer closest to you.

ATARI 810 is a registered trademark of Atari, Inc.

HAPPY COMPUTERS, INC. • P. O. Box 1268 • Morgan Hill, California 95037 • (408) 779-3830

PRODUCT MART

Advertisers Index

You asked for it . . .
CARDCO has it!

Another exclusive from Cardco!

The LQ/1 (Letter Quality) PRINTER

In response to your demand and need, CARDCO now has available for the **first** time . . . a "Commodore-ready "LETTER QUALITY PRINTER". Just plug it in and print.

Offering standard friction-feed and optional tractor-feed, the CARDCO LQ/1 PRINTER comes complete with built-in interfacing for all Commodore Personal Computers, as well as compatible input for PC, PC jr., TRS-80 and other personal computers.

"CARDPRINT" C/?+G PRINTER INTERFACE with Graphics

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

"WRITE NOW" WORD PROCESSOR SOFTWARE

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

"MAIL NOW" MAILING LIST SOFTWARE

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

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

cardco, inc.

313 Matheson Wichita, Kansas 67214 (316) 267-6525

"The world's largest manufacturer of Commodore accessories."

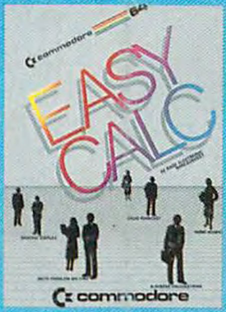
commodore 64

INDISPENSABLE SOFTWARE

For Your Most Important Computing Needs

EasyScript 64
Displays 764 lines × 240 characters. Prints to 130 columns. Works with EasySpell 64.

EasySpell 64
20,000 word Master Dictionary and automatic spelling checker. Works with EasyScript 64.



EasyCalc 64
Multiple electronic spreadsheet with color bar graph feature. 63 columns × 254 rows.

The Manager
Sophisticated database system with 4 built-in applications, or design your own. Text, formulas, graphics.

SuperExpander 64
21 special commands. Combine text with high resolution graphics. Music and game sounds.

Easy Finance I— Loan Analysis
12 loan functions. Bar graph forecasting as well as calculation.

Easy Finance II— Basic Investment Analysis
16 stock investment functions. Investment bar graph.

Easy Finance III— Advanced Investment Analysis
16 capital investment functions. Bar graphs.

Easy Finance IV— Business Management
21 business management features. Bar graphs.

Easy Finance V— Statistics and Forecasting
Assess present/future sales trends with 9 statistics and forecasting functions.

Accounts Payable/ Checkwriting
11 functions. Automatic billing. 50 vendors/disk.

Accounts Receivable/Billing
11 billing functions. Printed statements.

General Ledger
8 general ledger options. Custom income statement, trial balances, reports.

Inventory Management
1000 inventory items. Full reports.

Payroll
24 different payroll functions. Integrated with G/L system.

commodore COMPUTERS

First In Quality Software