

C-Notes for the New Tandy Model 100 computer appeared in 80-Micro from July 1983 to July 1984.

1983: JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER

1984: JANURARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, and JULY

"C-Notes Take a Look at Tandy's Model 100"

Tandy's Towering Totable An in-depth disassembly in words and pictures...	7/83:158	John Berman
People are Talking And so they were, about he first of a kind...	7/83:166	John P. Mello
But Will It Fly? What About Using It In The Air...	7/83:169	John P. Mello
Still Up in The Air	9/83:290	Eric Grevsted
Nothing's Perfect	8/83:209	Ken Barbier
Model 100 Conversions: Easy as I, II, III...	9/83:280	Beve Woodbury

This is a listing of the programs and articles that appeared in  
80 micro's C-Note section from July 1983 to July 1984.

(use <- menu key to return)

Article Title	Issue:Page	Debug	Author
Program Name	Description		
Winning Numbers	11/83:270		Ronald F. Balonis
DAILYN.BA	Increases your chance of winning lottery		
Program Length	12/83:246		Ronald F. Balonis
LENGTH.BA	Computes program length to manage memory		
100 Disks	1/84:171	6/84:168	Ronald F. Balonis
M3 COM.CMD	Transfer M100 files to Model 1/11 storage		
Write Now	2/84:191	6/84:168	Ronald F. Balonis
WRITER.BA	A Model 100 word processor		
Pin Pals	2/84:194		Ronald F. Balonis
	Transfer .BA and .DO files between two M100s		
A New Pair of Shoes	3/84:194	6/84:168	Ronald F. Balonis
SHOOTR.BA	Craps game for the Model 100		
Sort Order	7/84:126		
DOSORT.BA	Alphabetize your data files		Ronald F. Balonis



Nothing's Perfect	8/83:209	Ken Barbeir
BENCH.BA	A closer look at the Model 100	
Uploading Text to WordStar	8/83:210	Ken Barbeir
UPLOAD.BA	Transferring text files to WordStar	
Fast Backwards	3/84:184	Ray Cadmus
CALC.BA	Use reverse Polish notation to do math	
Graphic Results	4/84:188 7/84:137	Emmett Carmody
PIE.BA	Graphics for the Model 100	
BAR.BA	Graphics for the Model 100	
Disassembler	9/83:283 1/84:181	David Cloutier
DISASM.BA	A dis-assembler for the Model 100	
A Real Sketch	5/84:158	John Cornman Aileen Cornman
SKETCH.BA	A sketchpad graphics program	
HEX.DO	A sketchpad graphics program	
LOADER.BA	A sketchpad graphics program	
Form-matters	6/84:156	John Cornman Aileen Cornman
FORMS.BA	Converts M100 DO files to Model I..	Scripsit
Portable Scripsit	7/84:124	John Cornman Aileen Cornman
CSCRIP.BA	Converts Scripsit files to M100.DO	

Remote Robot	11/83:269	Peter W. Deininger Rolf A. Deininger
ROBOT.BA	Program controls radio-controlled robot	
Mod for Autoliner	5/84:173	Jacqueline Davis
AUTMOD.BA	Modification for AUTLIN.BA	
Seven 100 Percent Solutions	2/84:186	Rolf A. Deininger
	Alternative way's to power your Model 100	
Itinerary 100	7/83:173	Bradford N. Dixon
ITINRY.BA	Keeps your day organized	
Foxfighter	8/83:200 12/83:251	Bradford N. Dixon
FOXFGH.BA	Clear the sky of air-mines suspended oyw a city	
Mighty Write	10/83:266 2/84:197	Bradford N. Dixon
PRNXTX.BA	A word processing program for the Model 100	
Cram 100	10/83:283	
CRAM.BA	An exciting drawing game	Bradford N. Dixon
Monitor 100	7/83:178 12/83:251	Bradford N. Dixon
MONITR.BA	Taking a look inside your Model 100	
Thanks for the Memory	3/84:185	Bradford N. Dixon
	Install 8K memory chips in your Model 100	
Consulting the I Ching	10/83:267	Amee Eisenberg
ICHING.BA	Finding guidance to problems with the 100	

Harmony and me MUSIC.BA	A tone generator for the Model 100	11/83:265	Ben Firschein
From Soup to Nuts MENU.BA	Command menu drives your Model 100 programs	5/84:169	Ben Firschein
Calculator Program MCCALC.BA	Short calculator program	12/83:251	Mark Fox
To Err is Confusing ERRORS.BA	Prints the M100 error list	4/84:191	Robert L. Green
Rooting Out the Problem BISECT.BA	Plots data/finds the value of equation roots	1/84:176	William R. Harlow
North by Northwest NAVIGT.BA	Find distance and bearing between locations	4/84:186 9/84:30	Smith Harris
Account for Yourself DATMGR.BA	Generate expense account reports	3/84:187 6/84:168	Mark Hickenbottom
Punch Out TXCLCK.BA	A portable payroll program	7/83:172	Mare-Anne Jarvela
The Final Notice DUEDT.BA	A payment-scheduling progrwn	7/83:176 9/83:22	Mare-Anne Jarvela
Subterra SUBTRA.BA	Richard Ramelia's adventure game in portable	8/83:203	Mare-Anne Jarvela

If It's Tuesday	9/83:285	Mare-Anne Jarvela
CONV.BA	A currency and measurement conversion program	
To Market, To Market	10/83:276	Mare-Anne Jarvela
SHPLST.BA	A Portable Shopping list Program	
Nag Analysis	11/83:264	Mare-Anne Jarvela
HORSES.BA	Program rates horses by a handicapping system	
The Great Escape	1/84:174	Scott Jones
	Explains Model 100 escape codes	
Backing Up the Model 100	2/84:189	Byroan R. Leipper
BKUPDO.BA	Does automatic tape back-up of Model 100 files	
Make Your Own Modem Cable	12/83:244	Carl Oppedahl
	Shows how to build Inexpensive modem cable	
The Searcher	6/84:167	Carl Oppedahl
SEARCH.BA	Search M100 DO files for words and phrases	
Etch-A-Screen	7/84:131	Richard J. Perry
ETCHER.BA	Free-draw graphics program	
Etch-A-Screen	7/84:131	Richard J. Perry
DUMP.BA	Dumps Screen to printer	
Etch-A-Screen	7/84:131	Richard J. Perry
HEXETCH.BA	Binary Hex file conversion utility	

The Year in Review	8/83:207	Richard Ramella
REVIEW.BA	Bar graph displays data for a 12 month period	
Elapsed Time	9/83:284	Richard Ramella
ELAPTX.BA	Marks start/stop times of a program and time	
Autoliner	1/84:178	Richard Ramella
AUTLIN.BA	Automatically numbers Basic program lines	
The Shadow Knows	2/84:187	Richard Ramella
LAMONT.BA	Encode Model 100 files you want kept secret	
Getting Personal	6/84:161	Richard Ramella
HIPAL.BA	Prints personalized form letters	
Wipe Out	3/84:192	Thomas Robb
WIPOUT.BA	File deletion program that simplifies killing	
Calling Wall Street	4/84:178	Thomas Robb
DNJ/R.BA	Calling the Dow Jones News/Retrieval Service	
Calling Wall Street	4/84:178	Thomas Robb
GRAPH.BA	Calling the Dow Jones News/Retrieval Service	
Mnemonic Powers	1/84:170	Dan Robinson
	A table of the Model 100's command mnemonics	
CR Patch	1/84:181	Robi Robinson
CRLFIX.BA	Adds LF to Printer	

Blackjack	12/83:247	Paul Serotta
BLKJCK.BA	A simulation of the casino card game --- 21	
Inside the Model 100	12/83:232	David P. Sumner
	An extensive memory map for the Model 100	
CR Patch	1/84:181	Michael Stanford
CRLFIX.BA	Adds LF to Printer	
It's a Chimer	5/84:172	James M. Stubchaer
BIGBEN.BA	Westminster chimes every hour	
Gas/Oil Mileage	7/83:170 7/84:137	Beve Woodbury
GAS.BA	Keeps track a car's gas and oil consumption	
Traveling Expenses	7/83:171	Beve Woodbury
TRVLEX.BA	program cumulatively records travel expenses	
The Rule of 78	7/83:180	Beve Woodbury
RULE78.BA	Calculating the interest on a loan	
Intel	8/83:201	Beve Woodbury
INTEL.BA	A portable chess-pawn game	
Renumber 100	8/83:206	Beve Woodbury
RNUMBR.BA	A renumbering utility	
Quick On-site Job Estimate	10/83:270	Beve Woodbury
PARTS.BA	The parts and materials list	
Quick On-site Job Estimate	10/83:270	Beve Woodbury
JOBEST.BA	Estimate cost of doing a job	

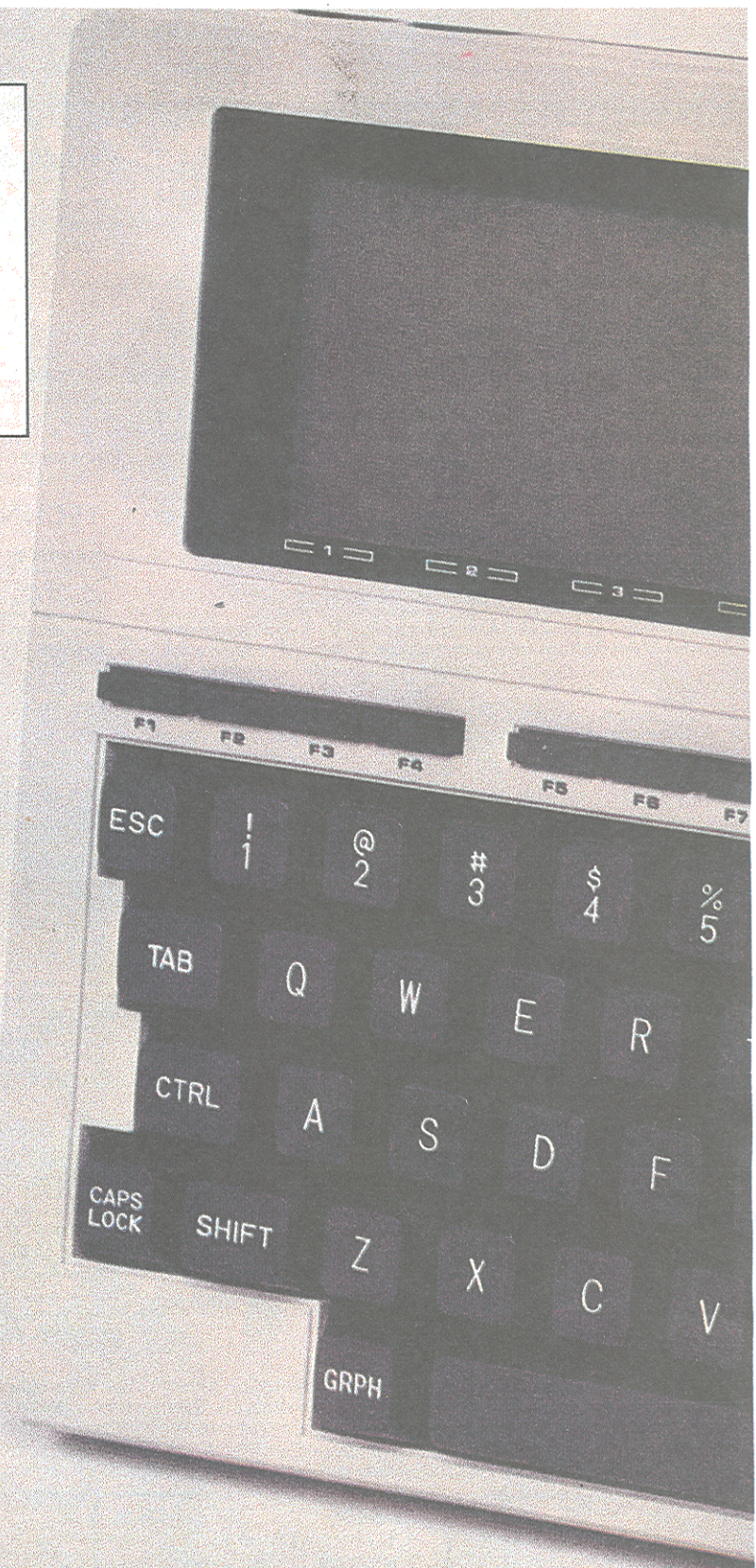
C-NOTE FOR JULY 1983



# C·Notes

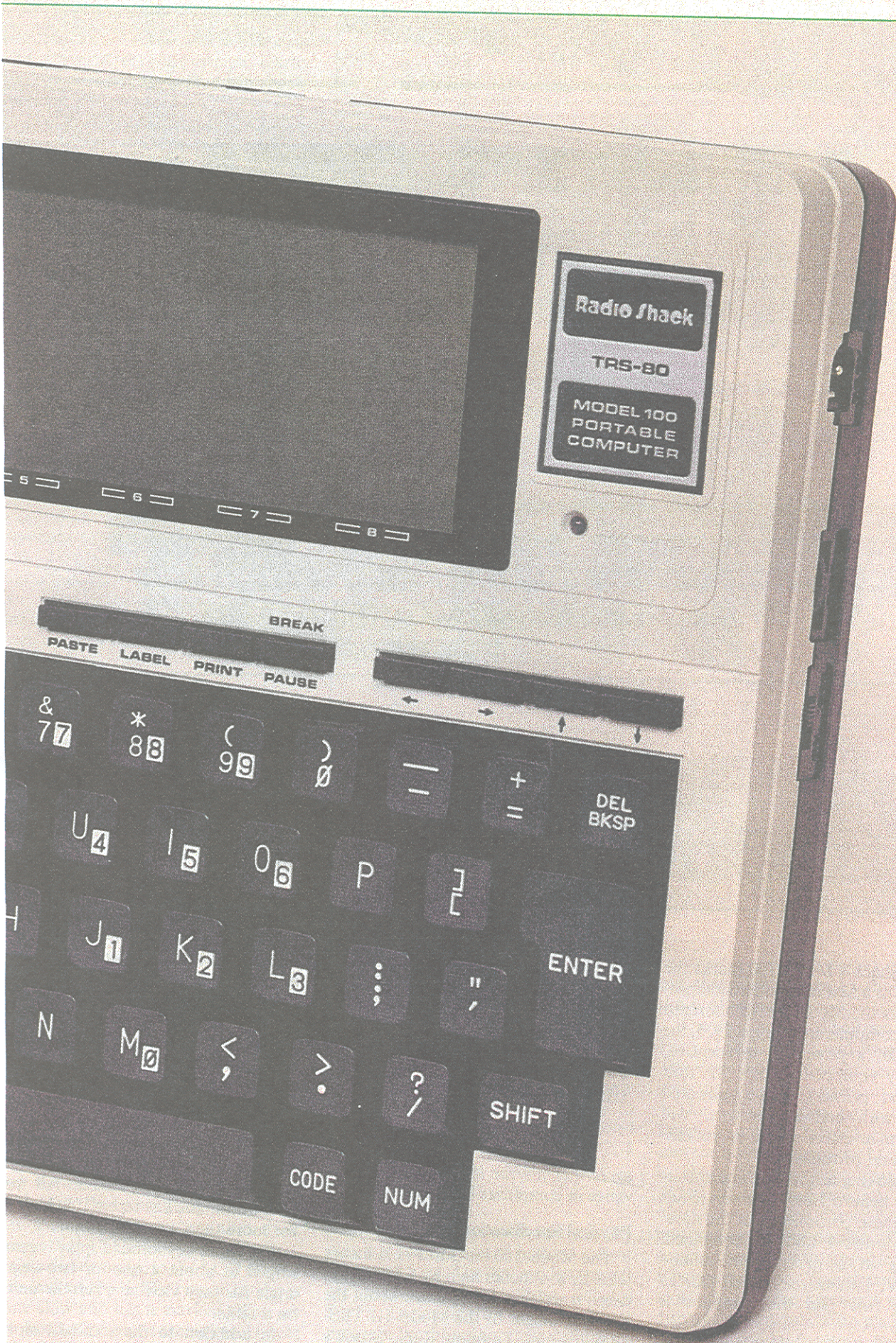
**80 Micro takes an in-depth look at Tandy's Model 100.**

<b>Tandy's Towering Total</b> .....	158
<b>People Are Talking</b> .....	166
<b>But Will It Fly?</b> .....	169
<b>Model 100 Start-up Kit:</b>	
<b>Gas/Oil Mileage</b> .....	170
<b>Traveling Expenses</b> .....	171
<b>Punch Out</b> .....	172
<b>Itinerary 100</b> .....	173
<b>The Final Notice</b> .....	176
<b>Monitor 100</b> .....	178
<b>The Rule of 78</b> .....	180



All photographs in this section  
by Frank Cordelle



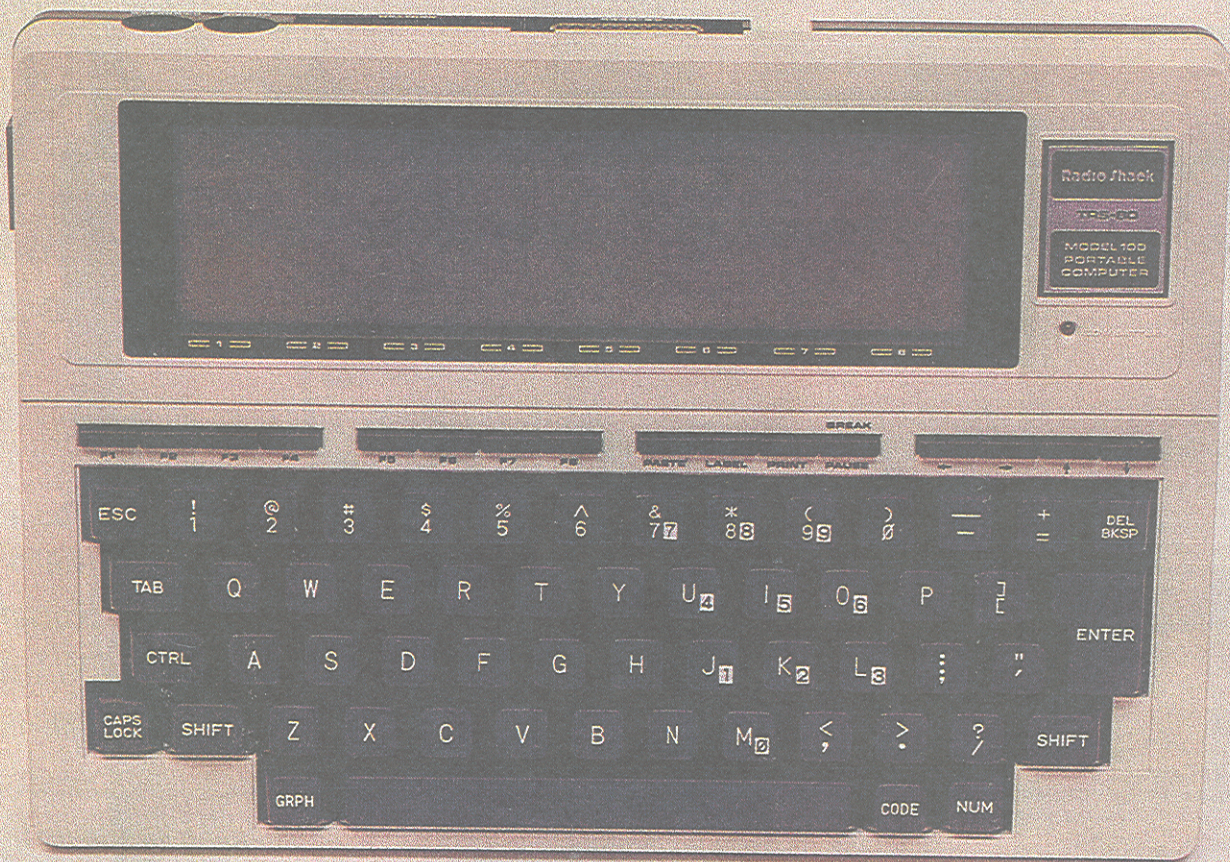


Radio Shack  
TRS-80  
MODEL 100  
PORTABLE  
COMPUTER

BREAK  
PAUSE PRINT LABEL PASTE

& 77 \* 88 ( 99 ) Ø - + DEL BKSP  
U 4 I 5 O 6 P ] [ ENTER  
H J 1 K 2 L 3 ; , " /  
N M 0 < , > . ? / SHIFT  
CODE NUM





# Tandy's Towering Totable

by John Berman

Radio Shack's Model 100 marks the beginning of a new era in microcomputers. Just as the Model I revolutionized the microcomputer industry in 1977, the Model 100, a powerful, inexpensive, completely equipped (except for disk drives and printer), battery-powered portable with built-in software, will have a similar effect on the microcomputer market of today.

After looking at the machine and its market, I believe Tandy is going to sell more of these portables than the number of Models I and III combined, something on the order of one million units. And I think they'll sell half of them the first year the computer is available.

To back up this rather outlandish claim, let me tell you about the computer. When I'm done, I'm sure you'll

---

**The Model 100 is all you could want in a portable—powerful, versatile, and lightweight.**

---

agree that it's the most significant advance in computers this year.

## Physical Specifications

The Model 100 measures 11 3/4 inches wide by 8 1/2 inches high by 1 7/8 inches deep. It weighs four pounds, about the size and weight of the Model III TRS-DOS manual. To quote Radio Shack's publicity announcement, "any smaller and the standard-sized keyboard

wouldn't fit, any lighter and the unit would slide across your worktable."

In fact, the unit's dimensions were dictated almost entirely by human-use (ergonomic) considerations. It features a full-sized typewriter-style keyboard with normal keys (not chicklet keys like the Color Computer, nor membrane keys like the Atari 400).

The keys have a solid feel, and give an easily audible "click" as they hit the bottom-stop when pressed. All keys have automatic key repeat if held down for more than a second.

The entire keyboard/display is gently sloped at about a one- or two-degree angle for easy viewing when the unit is on a table.

In addition to the standard alphanumeric and punctuation keys, there are special keys for computer functions.



These special keys are escape, delete, backspace, tab, caps lock, code, graph, number, and control keys.

The code key generates foreign-language letters with appropriate accent marks for French, Spanish, German, and other languages. The graph key puts graphics blocks, lines, and symbols on the display. The control key generates computer control codes 1-26.

The numeric key converts the U,I,O, J,K,L, and M keys into a numeric keypad, indicated by small number blocks in the lower right corner of each key. While the keys aren't aligned like a standard keypad, and there isn't a telltale bump on the I (5) key, it's better than no keypad at all.

The graph and code keys are also used with the shift key to give you an alternate set of graphics and letters. This means that all 255 displayable characters can be generated from the keyboard by one or another combination of keys. Unfortunately, using the control, code, and graph keys requires both hands. This is a problem for the handicapped since the keyboard is too wide to be spanned with one hand.

### The Display

The display is an eight-line by 40-character liquid crystal display (LCD), four times the size of the Epson HX-20 display. It's large enough to prevent most complaints about LCD displays being too small for serious work, but small enough to maintain the unit's portability.

The LCD features dot-addressable graphics of 64 rows by 240 columns that can be mixed with text for games, charts, and graphs.

The display also includes a dial to adjust the viewing angle of the LCD, so you can put the computer at almost any angle and position the display for maximum visibility. Because the LCD display uses reflected light to illuminate the display, it puts to rest one of the major complaints about video terminals: radiation-induced eye fatigue. (Radio Shack experimented with an LCD display of two 40-character banks, for a total display size of 80 columns by eight lines, but in user field tests discovered that it was a cumbersome system that most people didn't like.)

One disadvantage to the LCD display is its lack of speed. The response time is quite slow and has a marked effect on the speed of program execution. I wrote a simple For...Next loop that counted to 800, and had the program display

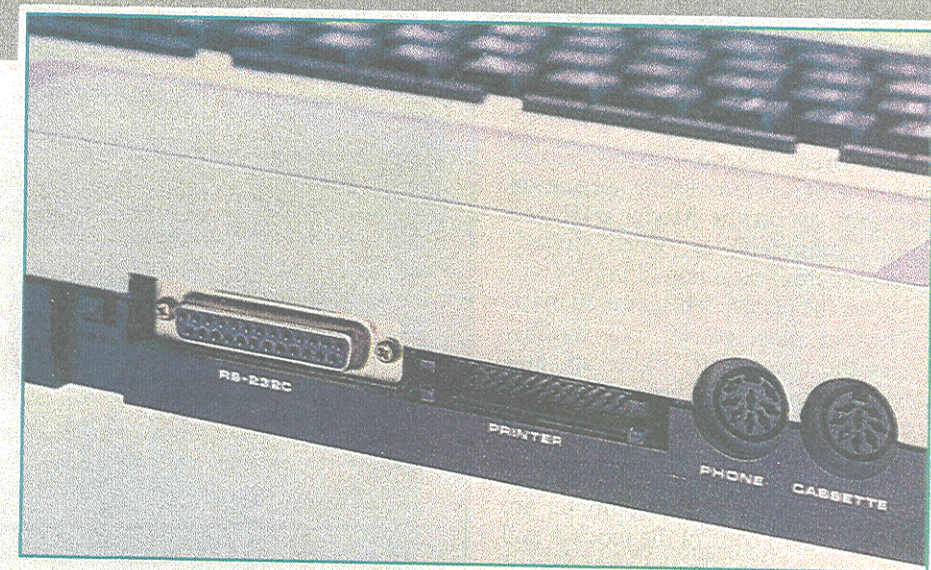


Photo 1. The back side of the Model 100. From left, the components are the reset button, RS-232 port, parallel printer port, phone jack, and cassette jack.

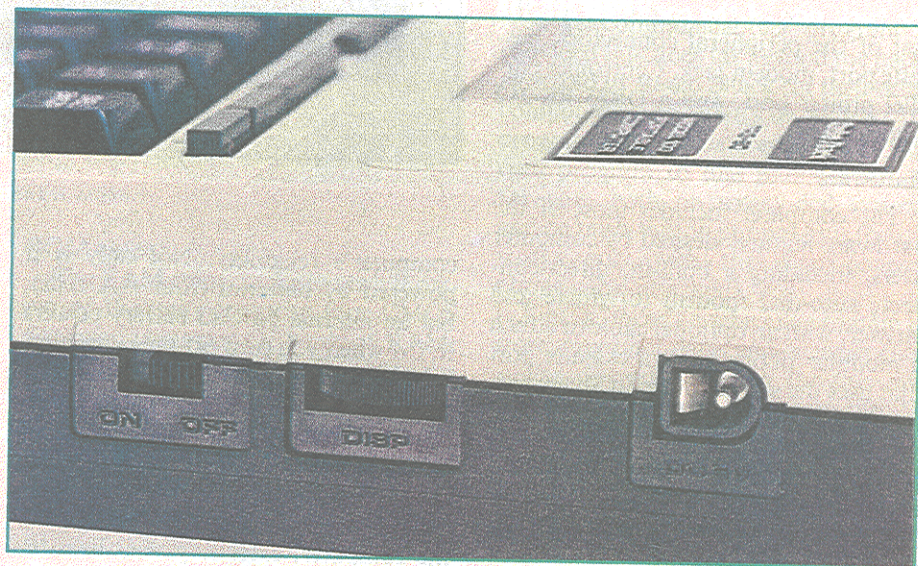


Photo 2. Right side of the Model 100 showing (from left to right) the on/off switch, display adjustment, and 6V dc power jack.

both starting and ending times.

It took approximately one second for the loop to execute. Adding a "PRINT@0,I" in the loop slowed execution time down to 14 seconds. Changing the PRINT@ to simply PRINTI, forcing the entire display to scroll one line for each number, slowed the display down so much that it took 33 seconds just to count to 100! A definite handicap since the CPU operates at 2.4 MHz.

Between the keyboard and the display is a line of 16 small buttons in four groups of four. The two sets on the left are special-function buttons, labeled F1-F8. Depending on which software program you're using, their definitions change, although in Basic you can redefine them to whatever you want (there

are space limitations to those definitions, of course).

The third group of buttons is labeled Paste, Label, Print, and Break/Pause. The Paste button is used in conjunction with the text editor; more on that later.

The Label button pertains to the function buttons labeled F1-F8, and uses the bottom line of the LCD display to show the definitions currently assigned to those buttons. A series of white boxes are evenly spaced across the bottom line of the display. The boxes are numbered 1-8 to correspond to the function buttons. The internal design constraints of the keyboard and LCD display circuitry prevented placement of the eight special-function buttons directly below the LCD.

The Print button sends either the dis-



## C-Notes

play contents or the entire file being edited to the printer port.

The Break/Pause button either pauses program or listing execution (like shift @ on the Model I/III and Color Computers) or, when shifted, breaks program execution (it prints Control-C on the display when you do that).

The rightmost set of four buttons moves the cursor left, right, up, and down. It would have been more convenient if the buttons were arranged in a diamond pattern, but the size of the LCD circuit board precluded that set-up.

On the left side of the computer are two switches to control the modem and a bar-code reader. On the right side of the computer is the on/off switch, the display-angle adjustment, and the external power supply (6V dc) connector. At the back side of the computer are the rest of the peripheral connectors: cassette port, modem jack, Centronics parallel printer port, and a DB25 RS-232 connector.

### Hardware

The computer includes most of the hardware features desired by computer users, including a built-in, direct-connect, auto-dial modem (originate and answer, with an optional cable to attach a modular phone line), a cassette port that uses the standard Radio Shack cassette cable and tape recorders, a 26-pin Centronics-compatible parallel port connector, and an RS-232 port. The computer also has a jack for an acoustic 300-baud modem for non-modular phones.

One feature adds considerably to the

computer's reliability. Not only is it powered by batteries (four AA cells), but an internal Ni-Cad battery powers RAM when you turn off the computer.

The Ni-Cad batteries maintain your memory after the other batteries go dead for up to 30 days with an 8K computer, eight days with a 32K computer. And the optional ac power supply recharges the batteries while you're using the computer, extending their lives indefinitely.

Photo 3. The inside of the Model 100. The main circuit board is on the

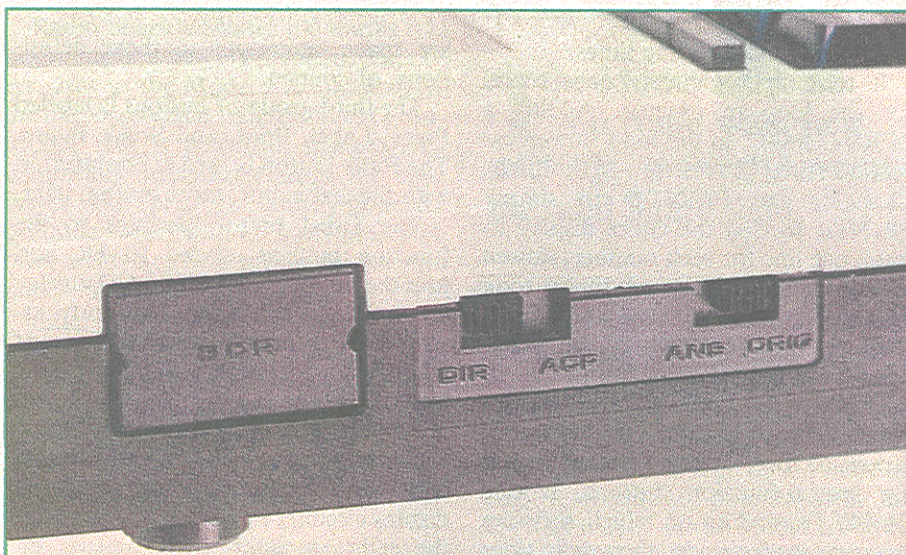
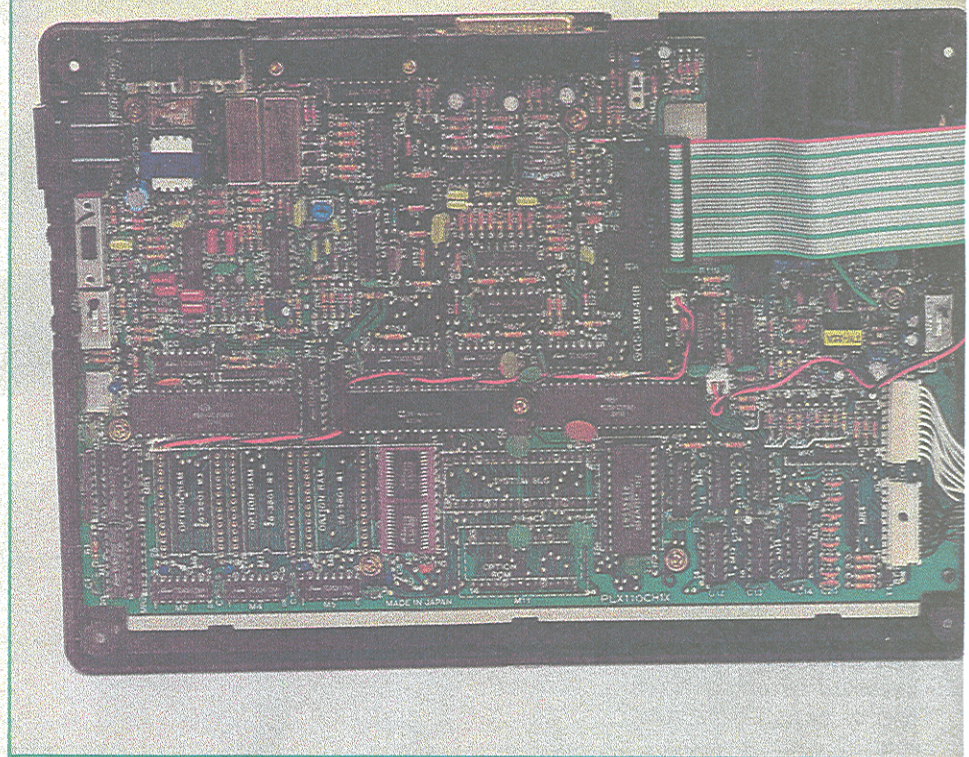


Photo 4. Left side of the Model 100 with (from left to right) the bar code reader port, a direct-connect/acoustic modem switch, and the answer/originate mode switch.

The AA batteries are used primarily to power the LCD, peripheral ports, and CPU. Using standard batteries you have about 20 hours of computing time before you'll have to replace them. When you have less than about 20 minutes of life left in the batteries, a low-power LED comes on. I ran mine for almost 40 minutes before my batteries finally gave out.

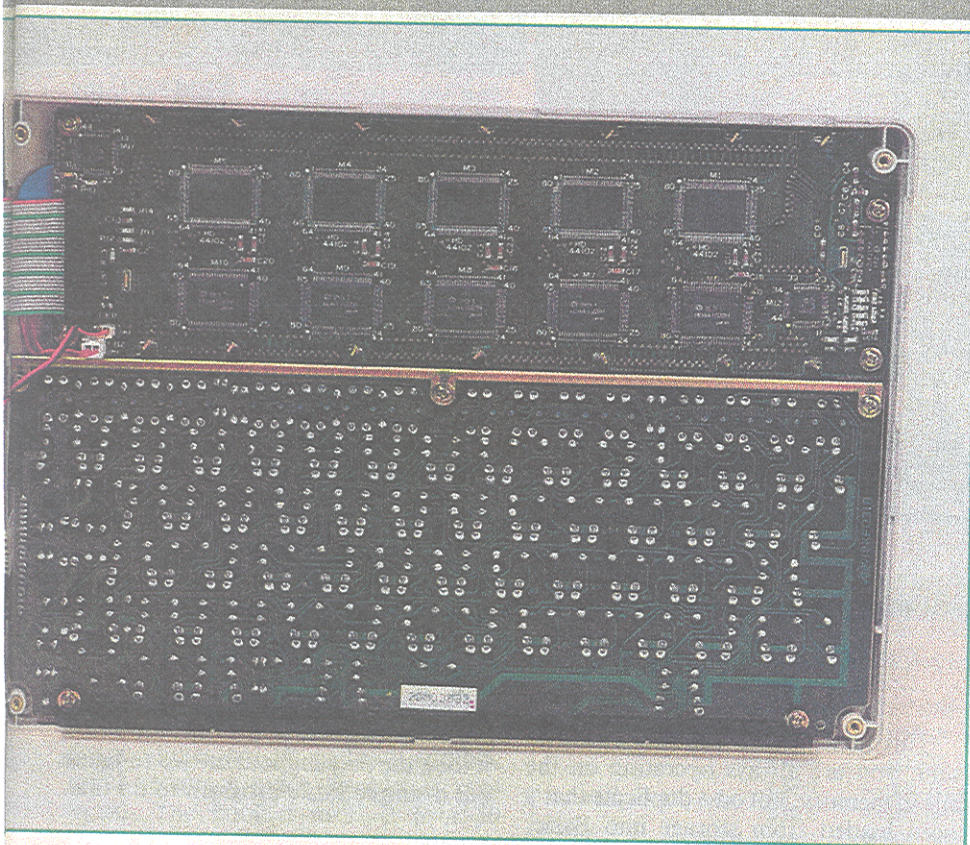
To see what would happen, I tried to CSAVE a program with weak batteries. About halfway through, the computer shut itself down and turned off the remote control on the tape recorder. Replacing the dead batteries with fresh ones restored the display, which still indicated CSAVE, but nothing was happening. I had to use the Break button to regain control of the unit.

CLOADing the file failed, since the power went off in the middle of a line of data, but the intact program remained in my computer's memory.

If you were to use AA Ni-Cad batteries, you'd probably get much more than 20 hours of useful life, and you could recharge them as many times as you wanted (well, almost, since even Ni-Cads get too tired to use after a year or so).

To help the batteries last longer, the computer automatically turns itself off if you don't use the keyboard for a specified time. You can set this time limit to anywhere from 60 seconds to 25 min-





left, the LCD readout (top) and keyboard circuitry are on the right.

utes. The power-off feature is disabled when you're in the term mode of Telcom.

For those interested in expanding the computer's capabilities, there's an expansion port socket under a small door on the bottom of the computer. The same compartment holds an expansion ROM socket to insert software cartridges.

The built-in modems are definitely a good deal, and allow the utmost in flexibility. The Centronics port is a must on almost all computers today, and the RS-232 port is an extension of the modem equipment.

You have complete control over the RS-232. Baud rates of 75, 110, 300, 600, 1200, 2400, 4800, 9600, and 19,200 are acceptable, with 6-, 7-, or 8-bit word lengths; Odd, Even, or no parity is supported as is XON/XOFF status control.

The baud rate is selected by specifying a number from 1-9, or using an M to select the 300-baud modem instead of the RS-232 port. All this is done with software; there aren't any hardware switches to play with.

Examining the inside of the computer is an experience in advanced technology. All the chips are CMOS design (complementary metal-oxide semiconductors), with extremely low power drain. The ROM is a special 32K package of eight smaller flat-pack CMOS

RAM chips.

The RAM chips, in the lower left corner of the unit, comprise four CMOS flat-pack chips per 8K bank of RAM. Two chips are on top and two are on the bottom of each special RAM pack carrier chip.

The direct-connect modem hardware is in the upper left corner of the unit, and the expansion port and expansion ROM socket are in the bottom center of the case.

It's almost incredible that they managed to pack so much capability in little hardware.

### Firmware

The firmware (software stored in ROM) is as impressive as the rest of the machine. In fact, the firmware is what makes the Model 100 such an exceptional computer. It's one of the first computers to supply all the software required by most first-time computer users as an integral part of the computer. The Model 100 gives you a Basic that's more powerful than any I've ever seen, far better than the Basic on Radio Shack's other computers.

The simple word processor takes only a few minutes to learn, but is powerful enough to satisfy most writers' requirements.

A telecommunications program is provided that goes far beyond the packages sold by Radio Shack for their other computers. And there's an address-file program that can be used with the Telcom program to dial phone numbers for you, both on rotary-dial systems and on push-button phones (this feature works only with the direct-connect phone cable attached).

Finally, there's a schedule program to store important dates and engagements.

The operating system for this machine is easy to use. When you turn on the computer you're presented with a menu display of what's in RAM, along with the month, date, year, day, time, and currently available RAM space.

There's room for 24 file names in the menu, five of which are occupied by

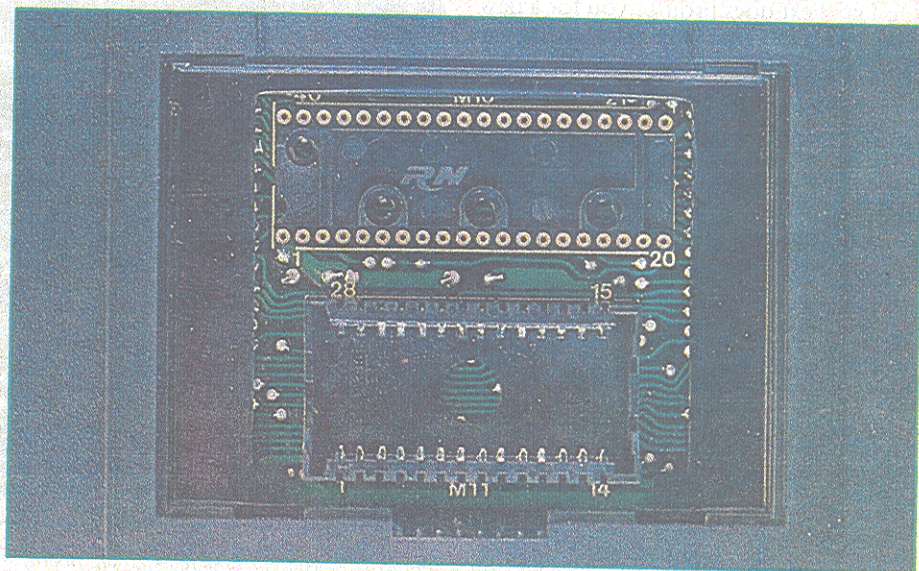


Photo 5. The ROM socket and expansion port on the bottom of the Model 100. (Note: The cover has been removed.)



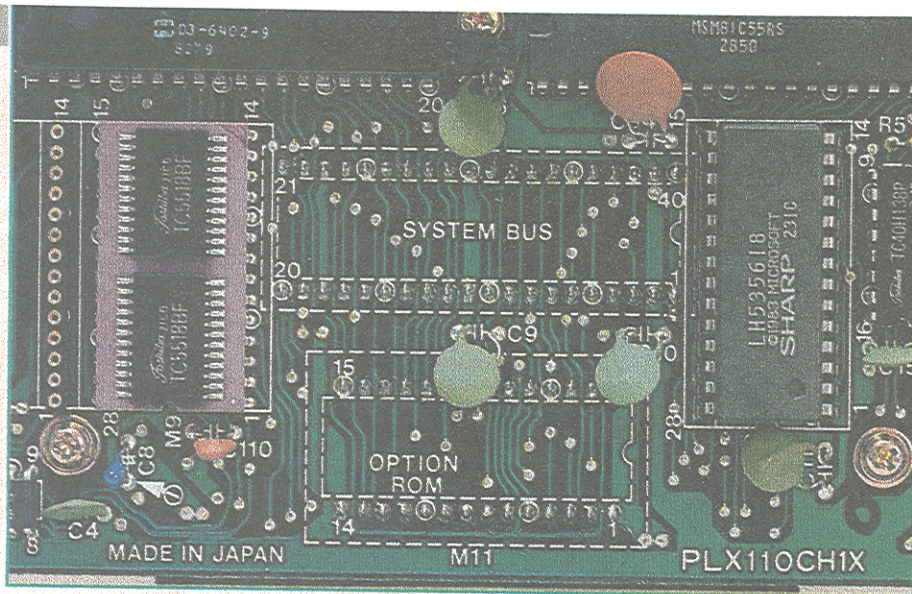


Photo 6. An internal view of the expansion port and ROM socket (see Photo 5).

you want to work on "TEMPB", the ASCII file, you have to make sure you have enough room in RAM for a third program.

ASCII files aren't loaded into Basic; they're duplicated into Basic as tokenized programs! Thus you have "TEMPA", an ASCII file, "TEMPB", your new program, and a tokenized version of "TEMPA" in Basic. This is all fine and good if "TEMPA" is a small program, but if it's 2K, and "TEMPB" is 2K, you won't have enough room left in an 8K computer to load "TEMPA" into Basic. (Remember, you only have 5,460 free bytes, less if you have an address and/or schedule file set up.)

The only solution is to save your precious program on tape, a quick procedure at 1500 baud, fortunately.

Another drawback to the operating system is that you have no way to determine the length of RAM files. However, it took me only an hour or two to figure out a simple Basic program to do that.

### Manual

The manual is a well-written, spiral-bound, 224-page book. It lies flat and measures 8½ inches by 11 inches. It's fairly thorough, but some of the examples are too short to be informative and it's marred by several sins of omission. For example, the command ASC(x) isn't listed although it's a valid Basic command.

The LCD is listed as an output device with no instructions on how to use it, just a list of commands. Similar problems are scattered throughout the book.

Basic, Text, Telcom, Address, and Schedule. That's right, RAM is partitioned into RAM files. You can have up to 19 files in RAM, if you have enough room for them all. (An 8K computer has only 5,460 free bytes; the rest is used for display memory, function-key assignments, port addresses, and other things needed by the operating system.)

To select a program or text file, position the cursor over the file name and press enter. If it's a Basic program, Basic is enabled, your program is loaded, and it automatically begins execution. Text files are automatically loaded into the word processor, with the cursor at the beginning of the file. Machine-language files begin executing immediately.

Placing the cursor over Basic or Text loads them without a resident file, although Text does prompt you for a new file name. That way a text file is automatically saved when you return to the menu or turn off the computer.

Basic programs are left in Basic until you either provide a name to save them in RAM, or type New and wipe them out. Named files aren't destroyed by New; they're saved and Basic pointers are moved to empty RAM.

You can be working on a Basic program, go to a text file and write, and return to your Basic program that is exactly as you left it.

Which brings up the interesting point of how the computer works. When you load a file into Basic, it isn't moved from its current position in RAM. Instead, pointers in Basic are set up to tell Basic where in memory the file sits, and where there's empty RAM for Basic variables, arrays, and other data.

This means that, unlike a disk system, you are always operating on the file in memory, not on a duplicate that's been loaded from storage into Basic. This gets hairy at times and points out a major flaw in this type of system: You can't save a file twice from Basic.

Say you're working on a program called "TEMPA" and you want to save it under a new name so you can edit it without risking it. You can't save it as a Basic file. Instead, you have to save it as an ASCII file, which takes up more memory than a Basic file (untokenized versus tokenized), and memory is at a premium in an 8K system.

This leads to another problem. After you've finished with "TEMPA" and

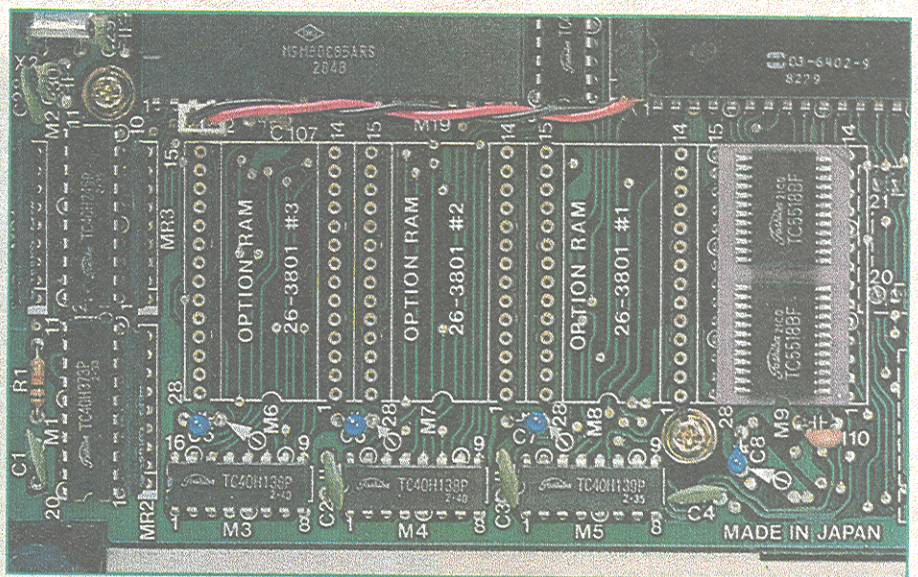


Photo 7. A section of the main circuit board, showing 8K RAM expansion sockets. Above, one 8K RAM module is in place at the far right center of the photo. Three additional 8K RAM module sockets are to the left.



The manual starts off with a brief explanation of how to use the Model 100 for text processing, telecommunications, scheduling, and as an address file. The explanations, while brief, are adequate. Then the manual goes into a description of Basic. Finally, a series of five appendixes gives the technical details on connecting Model 100 accessories, turning the machine on and off, sample work sessions, technical specifications, and maintenance information.

For the hacker, the specifications section is the most interesting since it gives the pin-outs for all the ports, including the expansion port on the unit's base. In addition, it has a display worksheet, character code tables, Basic error codes, and derived functions.

The last item is an index, which should have included the Basic commands but doesn't.

The manual is almost half as thick as the computer itself, but a small, totable, quick-reference guide booklet is provided. The booklet itself is over 50 pages long.

The Basic portion of the manual is definitely not designed for the neophyte. It consists of an alphabetical listing of almost all the Model 100 Basic commands. If you're a novice at Basic programming, you might have to buy a book explaining the commands.

### Accessories

Model 100 accessories currently available include a power supply (\$5.95), a direct-connect modem cable with one hour of CompuServe and Dow

Lease Costs (minimum purchase must total \$1600)		
Computer	Lease/machine	
M100 8K version	\$27.90/month	
M100 24K version	\$34.86/month	
8K upgrades (each)	\$ 4.19/month	
Service Contracts		
Machine	Lease customers	Non-lease
M100 8K	\$6.11/month	\$79/yr
M100 24K	\$8.25/month	\$99/yr
8K upgrades (each)	\$1.00/month	\$12/yr

(Lease customers are leasing the M100)

Table 1. The lease figures listed are based on the formula (Lease = .03492 times purchase price) for monthly payments. Subject to change at the whim of Tandy.

Addresses	Use
0 - 32767	ROM
128 - 600	Basic keyword list
796 - 859	Basic error codes
916 - 1024	Cold start values for RAM
32768 - 40959	Fourth RAM chip (24K-32K)
40960 - 49151	Third RAM chip (16K-24K)
49152 - 57343	Second RAM chip (8K-24K)
57344 - 62960	First RAM chip (8K). Note you only have 5K.
62961 - 65535	Operating system overhead
63109 - 63358	Keyboard input buffer
63369 - 63496	Basic function keys
63369	Files
63385	Load "
63401	Save "
63417	Run
63433	List
63449	(blank)
63465	(blank)
63481	Menu
63498 - 63625	Duplicate of 63369-63496
63841 - 64128	Menu display
63841	BASIC
63853	TEXT
63864	TELCOM
63875	ADDRSS
63886	SCHEDL
63897	Suzuki Hayashki
63908	Available (blank)
64904 - 64927	Menu Time Display
65024 - 65343	RAM duplicate of LCD
65344 - 65535	Telcom video buffer

Fig. 1. A rather primitive memory map, but more than you get from Tandy.

Jones connect time (\$19.95), a printer cable (\$19.95) that fits any Radio Shack parallel printer, and 8K RAM upgrades (\$119.95 plus \$15 installation). A hard-shell carrying case is not yet available.

For the businessman interested in leasing and maintenance, service contracts are available for the Model 100 (see Table 1).

### Basic

As I mentioned earlier, the real power of the Model 100 lies with its firmware. All the files in RAM and the hardware in the machine are available to the programmer. If you'll look at Table 2 you'll see many commands added to the standard Microsoft Basic to take advantage of the Model 100's special features.

The design and treatment of the I/O is exceptionally open-ended. The same I/O commands are used for all the different input and output devices (see Table 3). That is, the command Input# inputs data from the cassette port, the modem, the RS-232, and from RAM files. The command Print# sends data to the cassette, the modem, the RS-232, the RAM, the LCD, and the printer.

The difference is in how the file buf-

fer is opened, as a CAS, MDM, COM, RAM, LCD, or PTR buffer. (Although they don't mention it in the manuals, one of the devices used for I/O is DSX—a disk drive perhaps?) You can even use the Run command to load and run a Basic program from the modem, the RS-232 port, or tape cassette ports.

Making the Basic even more powerful are the ON...GOSUB commands. While running a Basic program you can set up subroutines to handle interrupts from the modem, the RS-232, function keys, and internal clock. That's right, you can put a specific time in the ON...GOSUB statement, and when that time arrives, the program branches to the subroutine automatically!

Another handy feature is the Files command, which displays the file names currently in the menu. If you happen to be using one of the Basic programs, it's indicated by an asterisk at the end of the file name. The Files command can even be included in a program.

Microsoft decided to use the text processor they already had, Text, rather than write a new one just for the 100's Basic. So you can edit a line, a group of lines, or even your whole program just by typing Edit and giving line numbers.

Since you're in the text editor and all the text commands are available to you, you can move lines, merge lines, split lines, and make any other changes you want.

When you're finished, press the F8 key and your alterations are merged back into your program. If you've improperly formatted your edited lines, the Model 100 beeps and remains in the text editor (I'll tell you more about the text editor in a moment).

This warning system doesn't always work, though. When I tried to combine four Basic lines into one, the combined line never made it back into Basic. Although I tried several times, the result was always the same, no original or merged lines in Basic. I still haven't figured out why this happened since I've successfully merged other lines.

Model 100 Basic does have a few disadvantages: It can't do hexadecimal or octal base arithmetic, there's no Auto Line-number command and no Delete command, and the largest line number allowed is 65529 (the manual doesn't mention this anywhere; I discovered it through trial and error).

I also found one serious flaw in Basic: In If...Then...Else statements, if the command Else is misspelled you get no error messages. I discovered this when I had an If...Then statement with the Else at the end of the display

*"The word processor included with the Model 100 is a sharp little program."*

line. The Else was misspelled ELSRETURN, with the Return on the next line. Not only didn't I get an error message, the Return was ignored. It took many hours of hard searching to track down the error.

Model 100 Basic has another little idiosyncrasy: Sending PRINT to the display doesn't automatically clear each line. That is, if you fill the display with

information, then send "PRINT@0," ":PRINT:PRINT:PRINT", you don't get the first four lines of the display cleared, as you do with the Models I/II/III/12/16 and the Color Computer. The only way to clear a line is to use the PRINT@ statement with the STRING\$ statement to send a batch of blank spaces to the display to clear the required number of lines.

There may be many interesting things the display can do, but they aren't listed anywhere in the manual. For example, typing PRINTCHR\$(27);"p" makes everything sent to the display appear as reverse characters (white on black). To switch back, type PRINTCHR\$(27)"q". These must be lowercase letters, although I don't know why.

### Text Processor

The word processor included with the Model 100 is a sharp little program. It's character-oriented and provides most of the important features desired by writers. You can move, duplicate, and delete by the character, word, line, or sentence, everything on the display be-

Table 2. Basic keywords. Note the interrupts allowed from the RS-232, modem, and time feature. Not listed in either the quick reference guide nor the manual are the two commands DSKIS and DSKO\$. I wonder what they could be for?

Command	Definition	EDIT	
ABS	Get absolute value of number.	END	Edit a Basic program, line, or range of lines.
ASC	Get ASCII code for character.	EOF	End program execution.
ATN	Arctangent function.	ERL	Test for end of file.
BEEP	Generates a "beep".	ERR	Get line number of last error.
CALL	Calls M/L routine, passes values to A HL reg.	ERROR	Get number code for last error.
CDBL	Converts numbers to double precision.	EXP	Simulate an error.
CHR\$	Returns ASCII character for number.	FILES	Exponent.
CINT	Truncates decimal number to integer.	FIX	Display all file names in menu.
CLEAR	Clears string space; sets high memory.	FOR...TO...	Truncate real numbers.
CLOAD	Load a Basic program from tape.	STEP...NEXT	
CLOAD?	Verify a Basic tape program.	FRE	Establish program looping.
CLOADM	Load a M/L program from tape.	GOSUB	Free memory space.
CLOSE	Close open data files.	GOTO	Call a Basic subroutine.
CLS	Clear Screen.	HIMEM	Transfer program execution.
COM	Enable/Disable communications (RS-232) from Basic.	IF...THEN...ELSE	Get highest memory address available to Basic.
CONT	Resumes program execution.	INKEY\$	Conditional test expression.
COS	Cosine function.	INP	Scan keyboard once.
CSAVE	Save a Basic program to tape.	INPUT#	Input from a port.
CSAVEM	Save a M/L program to tape.	INPUT\$	Input data from a file.
CSNG	Convert a number to single precision.	INSTR	Input a given number of characters from either the keyboard or from a file.
CSRLIN	Returns verticle cursor position.	INSTR	Search a string for a substring.
DATA	Defines a data set in a program line.	INT	Replace a substring with another.
DATE\$	Set or display current date.	IPL	Get whole number from real number.
DAYS\$	Set or display current day.	KEY	Define warm start program.
DEFDBL	Define double-precision variables.	KEY LIST	Enable/Disable/Define a Basic function keys.
DEFINT	Define integer variables.	KILL	List current function key definitions.
DEFSNG	Define single-precision variables.	LCOPY	Delete a RAM file.
DEFSTR	Define string variables.	LEFT\$	Copy screen to line printer.
DIM	Define array size.	LEN	Return left portion of a string.
			Return length of a string.



fore or after the cursor, and everything in a file before or after the cursor.

The cursor is moved around the file by using the arrow buttons in conjunction with the shift and control keys. Shift/left arrow and shift/right arrow move the cursor to the beginning of either the previous word or the next word.

Shift/up arrow and shift/down arrow move the cursor to the top and bottom of the display, and if pressed again will page through your file, seven lines at a time (not eight because the cursor line is repeated on the new display). If you have the function key labels displayed, the paging is in groups of six lines.

The control key, when used with the arrow buttons, moves the cursor to the appropriate side of the display, or to the beginning or end of your file. As a result, moving around your file is fairly easy and fast.

The special function buttons are pre-defined as:

- Load, for loading files;
- Save, for saving files;
- Copy, for copying text into the Paste buffer for duplicating elsewhere;

- Cut, which cuts text out of your file in the Paste buffer, and stores it in the Paste buffer for moving elsewhere or deleting; and

- Menu, for returning to the same.

The Paste buffer is interesting. When you select a section of text to put in the Paste buffer, the text is duplicated into available RAM. Of course, you need sufficient room in the buffer for the entire text string, unless you use Cut to move the text into the buffer.

Additional room isn't necessary with Cut since you're moving the text from your file into the buffer and not duplicating it. What makes the Paste buffer so useful is that you can store text from one file and put it into another, even a Basic program. However, running a Basic program wipes out any text stored in the Paste buffer.

The Paste buffer has another drawback: You have to remember to clear it out if you need more room for text. Deleting 8K of text just moves it to the buffer. It still occupies 8K until you load the Paste buffer with another, shorter text string.

CLOSE file number  
 EOF(file number)  
 INPUT#file number,  
 INPUT\$(numeric expression, file number)  
 LINEINPUT#  
 LOAD  
 LOADM  
 MERGE  
 OPEN  
 PRINT#  
 PRINT#USING  
 RUN  
 RUNM  
 SAVE  
 SAVEM

*Table 3. Common I/O Commands. Note that not all devices will respond to these commands; for example, you can't input from the LCD.*

Using Text is simple. Just load a file and start typing. You're always in insert mode, so the current cursor position marks where new text is added.

The keyboard is interrupt-driven and won't lose any keystrokes no matter how fast you type. It's a little disconcerting to look up after quickly typing several sentences and watch the display

LET	Assignment statement (optional).	POWER OFF	Turn power off (in program).
LINE	Draw a line (dot graphics).	PRESET	Turn off x,y pixel in display.
LIST	List program on display.	PRINT	Print data on display.
LLIST	List program on line printer.	PRINT#	Print data to a file.
LINEINPUT#	Input a string from a file.	PSET	Turn on x,y pixel in display.
LINEINPUT	Input a string from the keyboard.	READ	Read values from a Data list.
LOAD	Load a Basic program (can be from RAM, CAS, COM, or MDM).	REM	Comment line.
LOADM	Load a M/L program (RAM or CAS).	RESTORE	Reset the data statement pointer.
LOG	Natural logarithm.	RESUME	Continue execution after an error, or after power is turned back on (after a POWER OFF command).
LPOS	Line printer column position.	RETURN	Ends subroutine of GOSUB.
LPRINT	Print data on line printer.	RIGHT\$	Return right portion of a string.
MAXFILES	List or set current number of files.	RND	Return a random number between zero and one.
MAXRAM	Lists current memory size.	RUN	Execute a Basic program.
MDM	Enable/Disable interrupts from modem to Basic.	RUNM	Execute a M/L program.
MENU	Exits Basic.	SAVE	Save a Basic program (to RAM, CAS, COM, or MDM).
MERGE	Merges ASCII program with current program (can be from RAM, CAS, COM, or MDM).	SAVEM	Save a M/L program to RAM or CAS.
MID\$	Get/Replace middle characters of a string.	SCREEN	Lock/Unlock display of function key labels.
MOTOR	Turn cassette motor on/off.	SGN	Algebraic sign.
NAME...AS...	Rename a RAM file.	SIN	Trigonometric sine function.
NEW	Erase current program from Basic buffer.	SOUND	Enable/Disable/Output a tone (1 to 16383).
ONCOMGOSUB	Define Communication interrupt.	SPACE\$	String of spaces.
ONERRORGOTO	Define Error interrupt.	SQR	Square root function.
ONKEYGOSUB	Define Function Key interrupt.	STOP	Halt Basic program execution.
ONMDMGOSUB	Define Modem interrupt.	STR\$	Convert a number to a string.
ONTIME\$GOSUB	Define Clock interrupt.	STRING\$	Define a string of characters.
ON...GOTO	Branch on expression.	TAB	Skip space on printing.
ON...GOSUB	Branch on expression.	TAN	Trig. Tangent function.
OPEN	Open a file for I/O.	TIMES	Set/Display current clock time.
OUT	Output a byte to a CPU port.	TIMES	Enable/Disable time interrupt.
PEEK	Get a value directly from memory.	USING	Format data to be printed.
POKE	Load a value directly into memory.	VAL	Convert strings to numbers.
POS	Get column position of cursor.	VARPTR	Get address of variable.
POWER	Define time to Automatic power off.		

fore or after the cursor, and everything in a file before or after the cursor.

The cursor is moved around the file by using the arrow buttons in conjunction with the shift and control keys. Shift/left arrow and shift/right arrow move the cursor to the beginning of either the previous word or the next word.

Shift/up arrow and shift/down arrow move the cursor to the top and bottom of the display, and if pressed again will page through your file, seven lines at a time (not eight because the cursor line is repeated on the new display). If you have the function key labels displayed, the paging is in groups of six lines.

The control key, when used with the arrow buttons, moves the cursor to the appropriate side of the display, or to the beginning or end of your file. As a result, moving around your file is fairly easy and fast.

The special function buttons are pre-defined as:

- Load, for loading files;
- Save, for saving files;
- Copy, for copying text into the Paste buffer for duplicating elsewhere;

- Cut, which cuts text out of your file in the Paste buffer, and stores it in the Paste buffer for moving elsewhere or deleting; and

- Menu, for returning to the same.

The Paste buffer is interesting. When you select a section of text to put in the Paste buffer, the text is duplicated into available RAM. Of course, you need sufficient room in the buffer for the entire text string, unless you use Cut to move the text into the buffer.

Additional room isn't necessary with Cut since you're moving the text from your file into the buffer and not duplicating it. What makes the Paste buffer so useful is that you can store text from one file and put it into another, even a Basic program. However, running a Basic program wipes out any text stored in the Paste buffer.

The Paste buffer has another drawback: You have to remember to clear it out if you need more room for text. Deleting 8K of text just moves it to the buffer. It still occupies 8K until you load the Paste buffer with another, shorter text string.

CLOSE file number  
 EOF(file number)  
 INPUT#file number,  
 INPUT\$(numeric expression, file number)  
 LINEINPUT#  
 LOAD  
 LOADM  
 MERGE  
 OPEN  
 PRINT#  
 PRINT#USING  
 RUN  
 RUNM  
 SAVE  
 SAVEM

Table 3. Common I/O Commands. Note that not all devices will respond to these commands; for example, you can't input from the LCD.

Using Text is simple. Just load a file and start typing. You're always in insert mode, so the current cursor position marks where new text is added.

The keyboard is interrupt-driven and won't lose any keystrokes no matter how fast you type. It's a little disconcerting to look up after quickly typing several sentences and watch the display

LET	Assignment statement (optional).	POWER OFF	Turn power off (in program).
LINE	Draw a line (dot graphics).	PRESET	Turn off x,y pixel in display.
LIST	List program on display.	PRINT	Print data on display.
LLIST	List program on line printer.	PRINT#	Print data to a file.
LINEINPUT#	Input a string from a file.	PSET	Turn on x,y pixel in display.
LINEINPUT	Input a string from the keyboard.	READ	Read values from a Data list.
LOAD	Load a Basic program (can be from RAM, CAS, COM, or MDM).	REM	Comment line.
LOADM	Load a M/L program (RAM or CAS).	RESTORE	Reset the data statement pointer.
LOG	Natural logarithm.	RESUME	Continue execution after an error, or after power is turned back on (after a POWER OFF command).
LPOS	Line printer column position.	RETURN	Ends subroutine of GOSUB.
LPRINT	Print data on line printer.	RIGHT\$	Return right portion of a string.
MAXFILES	List or set current number of files.	RND	Return a random number between zero and one.
MAXRAM	Lists current memory size.	RUN	Execute a Basic program.
MDM	Enable/Disable interrupts from modem to Basic.	RUNM	Execute a M/L program.
MENU	Exits Basic.	SAVE	Save a Basic program (to RAM, CAS, COM, or MDM).
MERGE	Merges ASCII program with current program (can be from RAM, CAS, COM, or MDM).	SAVEM	Save a M/L program to RAM or CAS.
MIDS	Get/Replace middle characters of a string.	SCREEN	Lock/Unlock display of function key labels.
MOTOR	Turn cassette motor on/off.	SGN	Algebraic sign.
NAME...AS...	Rename a RAM file.	SIN	Trigonometric sine function.
NEW	Erase current program from Basic buffer.	SOUND	Enable/Disable/Output a tone (1 to 16383).
ONCOMGOSUB	Define Communication interrupt.	SPACES	String of spaces.
ONERRORGOTO	Define Error interrupt.	SQR	Square root function.
ONKEYGOSUB	Define Function Key interrupt.	STOP	Halt Basic program execution.
ONMDMGOSUB	Define Modem interrupt.	STR\$	Convert a number to a string.
ONTIME\$GOSUB	Define Clock interrupt.	STRING\$	Define a string of characters.
ON...GOTO	Branch on expression.	TAB	Skip space on printing.
ON...GOSUB	Branch on expression.	TAN	Trig. Tangent function.
OPEN	Open a file for I/O.	TIMES	Set/Display current clock time.
OUT	Output a byte to a CPU port.	TIMES	Enable/Disable time interrupt.
PEEK	Get a value directly from memory.	USING	Format data to be printed.
POKE	Load a value directly into memory.	VAL	Convert strings to numbers.
POS	Get column position of cursor.	VARPTR	Get address of variable.
POWER	Define time to Automatic power off.		

catch up to what you were typing.

The constant insert takes a little getting used to. If you make a mistake, you can't go back and type over it; you have to delete a character every time you type a correction. I usually just back up over

communications hardware is impressive, including a standard RS-232 connector, a direct-connect (for modular plugs) auto-dial modem, and an acoustic modem. One slide switch controls answer or originate mode, and another

directly intercept or place phone calls.

The acoustic modem uses a cable with rubber cups that accept the phone's handset. You can use any standard phone handset with the acoustic modem, even those on pay phones.

When you select Telcom from the menu, you're told the current baud rate, word length, stop bits, and parity. You can change these settings by typing Status followed by the new conditions (F3 prints Status for you). F1 searches through the address file to match a character string you type in (such as CIS for CompuServe). It displays the number so you can use the auto-dial feature or dial it yourself.

Pressing F2 makes the 100 dial the number you've indicated. If you want to tie into a computer network, leave the phone hook down and the 100 automatically goes on-line. If you want to talk to someone, lift the handset before the 100 finishes dialing and you'll be put in voice mode when the computer finishes dialing.

F4 puts you in terminal mode, and F8 returns you to the menu.

---

*"The communications hardware is impressive, including a standard RS-232 connector, a direct-connect auto-dial modem, and an acoustic modem."*

---

the errant character, type the correction, and continue typing, letting all the bad keystroke characters pile up in front of the cursor.

When I'm done with the sentence, I delete the whole batch of accumulated errors. This only works if I catch the mistake as soon as I type it; otherwise I have to go back through the text and remove typos.

There is a backspace key that erases the character you've just typed, but I always forget about it, silly though that is.

The only other commands available are for printing your file. The Print button is used to either print the display or print your entire file. When the Print button is pressed, you're prompted for the line width of your printer, and then the Model 100 starts sending characters to the printer.

There are no printer formatting commands, but you can embed printer control commands in your text by using the control key. This lets you take advantage of any special printer functions your printer may have, such as underline or boldface.

The lack of printer formatting commands isn't as bad as it sounds, since you can write a simple Basic program that formats your printed file. As a matter of fact, the manual has a short program in the appendix that paginates your file every 56 printed lines and prints a page title with the time and date. It won't take much skill to add to the program so it indents paragraphs, prints page numbers, centers lines, and so forth.

The text processor also doesn't have global search and replace functions, multiple block definition for altering text, optional windowing of text (so you can see what text formatted for a 60-column page would look like), and macro definitions.

## Telecom

As I mentioned before, the commu-

selects acoustic or direct-connect modem hardware.

Both the acoustic and the direct-connect modems use the same receptacle on the top of the Model 100. To use the direct-connect modem, you have to use the modem cable with the modular plug and female connector on it. After plugging the cable into the computer, plug the male plug into the wall socket, and your phone's modular plug into the cable's female connector. Now the computer is between your phone and the wall socket, and can

---

## People Are Talking

by John P. Mello Jr.  
80 Micro News Editor

From market analysts to hungry competitors Tandy's book-sized Model 100 micro has attracted accolades, and some observers maintain the Micro Executive Work Station (MEWS) may be the elbow grease the Forth Worth firm needs to polish its ailing image.

"I think it's a very attractive machine," observed Aaron C. Goldberg, research manager for information systems at International Data Corporation in Framingham, MA.

Alex D. Stein, an industry analyst with Dataquest in Cupertino, CA, added, "It is the embodiment of the user friendliness that the market is requiring more and more [these days]."

Clive Smith, a senior analyst with The Yankee Group in Boston, commented, "It's the first real portable incorporating a substantial flat screen into the computer. It's only eight lines—obviously a full page would have been better—but it is a significant step. With further enhancements, which I expect Tandy to introduce, I expect the product to do very well for them."

David Hughes, publisher of Source-trek Magazine on The Source and a leading figure in the Network Nation, also found the 100's screen a drawback, but still praised the micro. "It looks damn good to me," he said. "A chunk of text 8 by 40 is small for a writer, but it does feel like I'm writing and not just getting a data stream in."

"The 100," he said, "fills a real gap between the hand-held and Osborne level portables."

He explained: "Time after time, I'm at some meeting and I want to take running notes and access information in my system by dialing it up without being obtrusive. I've literally taken an Osborne into a meeting, sat in the back of a room, and used it for note taking. It works, but it's awfully awkward."

For John Hemphill, a product analyst for Future Computing in Richardson, TX, the 100 exploited what manufacturers of other "portables" ignored:

"The technology to do this has been around for quite some time, but manufacturers have been slow in putting it out. People that put out portable computers in the past really didn't make them portable and really didn't make

In terminal mode the function buttons are redefined. The first one is used to toggle between the previous eight-line display and the current one, giving you a total of 16 display lines. F2 saves incoming data in a RAM file. F3 ships a RAM file out (when enabled, the labels for F2 and F3 appear in reverse video).

F4 toggles the unit between half- and full-duplex operation, F5 sends incoming data to both the display and your printer, and F8 takes you out of term mode (but it asks if you want to disconnect before it does so).

The auto-dial feature has another important use. When you're signing onto a computer database, it automatically sends any log-on messages you want, including user ID numbers and passwords.

To facilitate its use, and to prevent someone from watching you auto-dial to a service and seeing your ID passwords, you can enclose the important information in greater-than and less-than symbols (" $<$ " and " $>$ ") in the address file, instructing Telcom not to display this information when you use the FIND and CALL function keys.

Overall, Telcom is a simple and easy-to-use system.

That is not to say that Telcom is perfect. It isn't. There's no echo function, so the Model 100 doesn't echo received characters back to the transmitting computer. There's also no automatic line feed when a carriage return is received. This presents problems when communicating between Model 100 computers.

Every time you hit enter, you have to remember to hit control-J or the cursor returns to the beginning of the line, obliterating the line with new, incoming information. Another disadvantage is that there isn't automatic file capture: Sending control-R to a Model 100 doesn't automatically open a storage buffer for an incoming file.

Speaking of buffers, you have no way of knowing the size of your available text buffer. Your only choice is to check the menu for available RAM before you go into Telcom. If you should run out of RAM while receiving a file, the Model 100 beeps once and the download indicator (F2) returns to nor-

mal video from reverse video.

## Uses

Business executives will find the Model 100 a boon to their work. They can prepare reports and charts, enter and analyze data, and download information from data bases, no matter where they are, a car, bus, private plane, or train. Where you would normally waste time, you can now use a portable computer to take advantage of otherwise enforced idleness.

With the expansion ROM socket, Radio Shack will soon have a more powerful word processor available, and I've heard they're working on an electronic spreadsheet for the 100 as well.

Salesmen and their managers will find the Model 100 worth its weight in gold. A salesman can prepare daily sales reports in the field and transmit them to headquarters at the first opportunity. Or he can write up sales orders on the 100 and save them in RAM. At day's end, the salesman calls the home office from his motel room or a public phone, and uploads all the day's sales.

*Review continues*

them battery-powered."

"One of the strong points of this machine," he said, "is the communications interface. It shows [Tandy] understands the importance of the communications-computer relationship."

Even Tandy's competitors admit the Fort Worth firm's MEWS is a winner.

"It's a pretty nifty little system," noted Mike Kennedy, marketing vice president for Grid Systems Corporation of Mountain View, CA. Grid makes a folio-sized computer selling for more than \$7,000.

An executive at a Japanese computer firm rumored to be preparing a market challenge to the 100 told *80 Micro* he purchased a MEWS the day it came out. He explained, "I had to call four or five Radio Shack stores to find one."

"I think it's a great product!" he declared. "I think they did a good job the first time through."

He added, "The way this product came together is the way things should go. You have the best of Japanese technology, Radio Shack's product definition, and Microsoft's software capability combining to make a nice product."

When interviewed about the 100, Herb Feinstein, director of marketing for the Teleram Communications Corporation in White Plains, NY, praised MEWS because it would improve the

market for his firm's \$2,995 book-sized micro. "We like the fact [Tandy's] introduced it," he said. "We think it's going to expand the market and bring more attention to portable computers."

Walt McIntyre, national sales manager for Epson of America, agreed with Feinstein about Tandy's entrance into the book-sized market: "I think it will help this whole segment of the business. They're a recognized name. Even though Epson is a world leader in printer and LCD manufacturing, the name is not associated with the same strength with microcomputers. It's a blessing on the concept."

"I think it looks good," Access Computer Corporation's Marketing Vice President Harry White said of the 100, "but it's not really in the same market we're in."

White's San Jose firm makes an Osborne-sized micro that comes with a printer, two types of modems, lots of software, and weighs more than 30 pounds with its power pack (see *80 Micro*, May 1983, p. 350).

"No one," he contended, "is going to do data-base management or word processing or spreadsheet analysis in any serious way on the little Radio Shack."

But on the contrary, some very serious computing may soon be done on

the 100, according to Steve Leninger, the father of the Tandy's Model I and Color Computer. Leninger, who recently returned to Tandy after a brief stint as a consultant, maintained: "My personal feeling is it's going to be another Model I—a popular machine with a lot of independent software being written for it. I think a giant amount of software will be written for it."

But the 100 is more than just a new machine for Tandy. Not only does it represent one of the largest investments in the firm's corporate history, but it may resuscitate what some experts see as Tandy's flagging image in the micro-computer market.

"I think it's important," IDC's Goldberg said of the 100, "because Tandy's market share's declined significantly. There's a lot of luster off the Tandy brand-name and a lot of people don't care who they are any more."

"To be quite honest, other announcements were pretty ridiculous," he opined. "Now we see they're in the market for real. The price [of the 100] is competitive and they're back in a new part of the marketplace."

Stein of Dataquest, though, disagrees with Goldberg's analysis of Tandy's image.

*News story continues*

The following morning, the manager prints out all the salesmen's reports on the office computer and prepares a summary report for the boss. This can be done with a Model III and currently available BBS software (such as Connection-80 or Forum-80).

With minor modifications, the home-office computer could be programmed to print confirmation orders to mail to customers contacted by the salesmen. The net result: next-day accuracy on the status of your salesmen, up-to-date reports on inventory levels and the fastest-selling items, and the ability to leave messages on the company's BBS, both for general product information and for specific salesmen.

In today's market, daily information on sales and product activity can make or break a company. The Model 100 gives companies access to that information at a modest cost per salesman.

Writers will love the freedom the computer gives them. Up to now, most writers have been limited to writing with typewriters. Even the small portables are difficult to deal with. You need paper and a place to set it up. And they're noisy.

After you've finished a page, you have to either retype the page or literally cut it up for revisions. And if you aren't near a wastepaper basket while typing, what do you do with all the paper you've wasted?

With the new Model 100, your compositions are easily revised. When you have a finished draft, you can save it to tape for later printout. Newspaper reporters will also find it useful for remote reporting, uploading the finished

---

*"In conclusion,  
I think the Model 100  
is the best computer  
to appear on the market  
in a long time."*

---

copy to the newspaper's computer. No more rushing back to the office, or calling it in over the phone.

Students can use the machine for term papers and reports. All they need is a printer, and I'm sure some bright soul will set up a printer rental operation on campus.

#### Miscellaneous Comments

Tandy's advertising implies you can use the Model 100 in an airplane; the computer's box even has a picture of an executive working in flight. Unfortunately, you may not be able to use the Model 100 on a commercial airplane

(see p. 169 for the lowdown on in-flight use of the 100).

Another potential problem is the use of power supplies. Radio Shack, and many other companies, sell 6V dc power supplies for toys and games. These supplies should not be used with the Model 100 because they don't provide any line-current or voltage-filtering protection. The CMOS chips inside the Model 100 are extremely sensitive, so using these other power supplies could eventually ruin your Model 100. Use only the power supply made by Radio Shack expressly for the Model 100. At \$5.95 you can't say they're trying to get more money out of you for their computer.

In conclusion, I think the Model 100 is the best computer to appear on the market in a long time. While disk drives aren't immediately available, they are on the way. In addition, I know someone who's developing a 128K battery-powered bubble-memory device that plugs into the Model 100. It should be ready by September.

Considering what you get for the price, the Model 100 is a great computer. I'm sure many copycat computers will appear on the market, trying to cash in on the boom.

If you're looking for a portable computer, and can do without CP/M and disk drives for now, buy the Model 100. I don't think you'll regret it. ■

New story continued

"I don't think Tandy is in decline at all," Stein said. "I think we see Tandy losing market share because the market is growing and competition is coming in. We're not really seeing them suffering. Financially, they're still doing well. They're profitable."

However, the Dataquest analyst noted, "Tandy has made some decisions in distribution that have pretty much hindered their growing at the rate of the market."

On just how well the 100 may do this year, analysts differ.

Asked if Tandy could sell 300,000 units by the end of calendar 1983, Smith of Yankee responded: "That's plausible. We're not really talking about a lot of money here and we're talking about real utility. I think that's an achievable number."

Goldberg of IDC termed the 300,000 figure "a little high." He suggested the figure would be more around 200,000.

"It all depends on the acceptance of the product," he said. "It's hard to guess that people are going to go back to Radio Shack stores. This product may generate some interest in people, but, of course, it is not going to be available [outside Tandy stores]. I think that's going to hurt Tandy."

Dataquest's Stein also found the 300,000 figure high. "Even for a mass-market product like this, there's a ramp-up time," he said. "I think at the high end, they'll sell 150,000 units by the end of the year."

Predictions by Future Computing's Hemphill were even lower than Stein's. He said the under-\$1,000, book-sized market for 1983 would be 65,000 units, Tandy having 25 percent of that or 15,000 to 20,000 units.

Although Access Marketing Vice President White also found the 300,000 figure high, he conceded, "It might be possible... It's not impossible, but it sounds aggressive."

Asked if the 100 would appeal to sec-

ond- as well as first-time computer buyers, White said, "That's an interesting market for it. I wouldn't mind having one as a terminal for one of my other computers."

Sourcetronaut Hughes added, "You wouldn't necessarily use it as your first computer because it is a little limiting, but linked with other systems, it isn't. It's the water glass and you've always got the pitcher somewhere else."

With the release of MEWS, Tandy has taken an inside track on the book-sized market. Existing competitors are either priced outside the mass market or have fewer features than the 100. But be prepared for at least six new book-sized micros before the end of the year.

"My guess is the thing is going to take off real well and everyone is going to jump behind it," one industry executive surmised. And Clive Smith added, "Eighty-four will be when they face significant challenges from other computer makers. Look to Comdex this winter." ■

# But Will It Fly?

by John P. Mello Jr.  
80 Micro News Editor

The Model 100 has been touted as the ideal micro for perking up life's dead spots—like long plane rides. But if you intend on computing while flying, beware! You're entering a regulatory morass.

When it comes to using electric gadgets on passenger planes, Federal Aviation Administration rule 91-19 bars them all—except tape recorders, hearing aids, pacemakers, and electric shavers—unless an airline approves the device.

Which devices have been anointed by the air carriers? A survey by *80 Micro* of six major airlines revealed none of them have a list of approved devices nor do they have a hard, fast rule on using portable computers on their aircraft.

"It is up to the airline to determine if the device interferes with the communication or navigation aids to the airplane," an FAA spokesman said. "Our belief now is that we don't think portable computers will do that."

Northwest and United spokesmen said their airlines allow computers to be used in flight.

"Generally," the United Airlines spokesman observed, "computers don't bother us as long as they can fit under the seat."

Matt Goring of Northwest Airlines noted, "As long as the computer doesn't transmit anything or receive anything or cause any sort of interference with communications, then we don't have any restrictions on it. Being a relatively new technology, we haven't experienced any problems with it, and a lot of people have had computers on board."

In its flight guide, American Airlines warns passengers: "Portable radios, TVs, and some electronic toys radiate signals that may interfere with the navigation system so we must ask you not to use those in flight."

"We really haven't addressed the subject of computers," American spokesman Paul Haney admitted. "If somebody attempted to use one and there was a question, it would undoubtedly fall on the captain to decide if it would be a problem or not."

At TWA, spokesperson Sally McElwreath explained, "The policy is to let a passenger use the computer until the pilot perceives any interference."

Delta Airlines was emphatic about barring micros from its aircraft. "We cannot take a chance on the safety of that equipment," a spokesman for the airline said. "We can't take a chance on something going awry. If someone came aboard and set up shop, the flight attendant would probably ask them to discontinue."

Yet Richard Shaffer of *The Wall Street Journal* was on a Delta flight when he wrote: "This is being written

---

*"Computers don't bother us as long as they can fit under the seat."*

---

in the air. . . . I'm testing a small computer designed to be used aboard airplanes and almost anywhere else." The small computer he was testing was the Model 100.

"I have yet to have an airline stop me from using anything," Shaffer told *80 Micro*.

"I went to a lot of trouble because it was an unannounced product," he said. "I taped up the name and I masked all the ports and everything that could identify what it was."

"The only one to say anything to me was the guy sitting next to me on the plane. He said, 'Is that one of those minicomputers?' And I said, 'No.'"

"I don't see how it would be any different from using a calculator," Shaffer added. "It's the same circuitry inside."

But on Eastern Airlines, even calculators may be a target of prohibition. "At one time we allowed some electronic calculators, but since they've expanded, we've had to rule those out," said Dale Jones, manager of flight technical services for Eastern at Miami International Airport.

Asked what happens when a person tries to board the plane with a computer, Jones responded: "We try to keep them off. It's a little difficult to say it's enforced without exception. The flight attendants have been instructed to tell anyone using computers not to

use them."

He maintained burdening airlines with regulating electronic devices creates problems. "The only way we can determine any individual device does not cause interference is to test that device on the aircraft that we fly," he said. "That's a long and very expensive process."

"We've had quite a bit of controversy over this," he continued. "We have in the past tried to take certain devices and evaluate them, but the expense and time involved was too much for us."

He added, "Our engineering department has received a number of electronic devices from manufacturers to evaluate, but we've just had to return them. We just can't do it."

Jones cited one instance where an aircraft missed a checkpoint allegedly because someone played a portable Pac-Man game on the plane. The pilot was convinced it was the game, Jones said, but "We can't prove it was Pac-Man. We tried to duplicate it on the ground and couldn't."

Richard Climie, director of avionics for the Airline Electronics Engineering Committee in Annapolis, MD, noted cases have been reported to his organization of devices causing systems on an aircraft to malfunction. He added, however, "I personally don't know of any case where this has created a safety hazard."

In one case, he said, a passenger used a radio telephone on a plane. "It caused an inconvenience to the passengers," he said. "It caused the system that controls the pressure to fluctuate and create discomfort in their eardrums."

Climie, who owns a Model I that "clobbers" TV channels at the lower end of the dial, noted: "Certainly things are better than in the days of the old Model I, but requiring each airline to approve the equipment is not a practical solution to the problem when we have so many devices."

One solution to what will become a growing problem as more and more book-sized totables enter the market is to have the FAA follow the lead of the Federal Communications Commission and adopt RFI standards.

"If they do that," Climie explained, "the testing could be done once. The units could be identified so it will be easy for the customer to know if he's buying something he can use on an airplane or not. It would not be ambiguous." ■

# Model 100 Start-up Kit

What do you do with your new Model 100?  
Here are seven programs to get you going.

## Gas/Oil Mileage

by Beve Woodbury  
80 Micro Technical Editor

Once you buy a Model 100, you'll never travel without it. This program lets you keep a running average and overall average of your car's gas and oil consumption.

Begin the program when your gas tank and oil pan are full. Run Program Listing 1. This program asks for the brand of car and the current mileage. It then establishes the data file, "GASOIL.DO". Listing 1 may now be deleted.

Program Listing 2 prints a menu with three options: record purchase of gas and/or oil, print a miles-per-gallon chart, and print oil use.

The first option, purchase of gas/oil, prompts for the price per gallon of gas, the current mileage, and the number of gallons purchased. The total purchase price is printed. You are given the opportunity to reenter the data if it is incorrect. The miles-per-gallon (mpg) figure from the last tank of gas is printed. You are then asked for the amount of oil added. The program must be run again to record another

### Printing Formats:

- B-8 spaces
- E-3 decimal number
- F-2 decimal number

### Arrays:

- D-dates of purchases
- Y-0, number of gallons purchased
- 1-price per gallon of gas
- 2-mileage
- 3-oil added

### For...Next Loops:

- K and J

### Decisions:

- A-data entered correct?
- H-add oil?
- U-purchases been run?
- Z-menu choice.

### General:

- C-car name
- L-last array dimension number
- M-current mileage
- N-new array dimension number
- O-array location of oil purchase
- P-price/gallon of gas
- Q-number of gallons of gas purchased
- R-total gallons of gas
- S-miles/quart of oil
- T-miles per gallon
- V-number of quarts of oil added
- W-total quarts of oil added
- X-last tank miles per gallon

Table 1. GAS.BA Variables

purchase.

Option two, print miles-per-gallon chart, prints a chart with tank fill-up number, date of purchase, mpg, price per gallon, and the cost of gas per mile. The overall mpg average is also printed.

Option three (oil use) prints a chart of the fill-up number, purchase date, amount of oil added, and number of miles per quart of oil. The overall miles per quart is printed.

Each option returns to the menu. ■

Line	Description
10-40	Formatting
50	Prints date
60-70	Open data file & read car name, last fill-up number
80	Prints heading
90-160	Set up and fill data arrays
170-240	Print menu, execute choice
250-310	Enter gas purchase data
320	Calculates mpg of last tank
340-370	Enter oil purchase data
380-420	Add new data to arrays
430-490	Write data arrays to data file
500-590	Print mpg chart
600-700	Print oil usage chart
710-720	Return to menu routine
730	End

Table 2. GAS.BA Line Descriptions

```

10 PRINT@10,"SET UP ORIGINAL FILE"
20 PRINT@50," FOR GAS & OIL USAGE
30 PRINT
40 OPEN"RAM:GASOIL.DO"FOROUTPUTAS1
50 INPUT"ENTER NAME OF CAR ";C$
60 PRINT
70 INPUT "ENTER CURRENT MILEAGE ";M
80 PRINT#1,C$;" ";M;" ";M;" ";M
90 CLOSE:END
    
```

Program Listing 1. ORIG.BA

### Program Listing 2. GAS.BA

```

10 CLEAR
20 REM * GAS AND OIL MILEAGE *
25 REM BY BEVE WOODBURY
30 DEFSTR-H:B=SPACE$(8):F="#####.###"
40 E="####.###"
50 CLS:PRINT@50,DATE$:PRINT
60 OPEN"RAM:GASOIL.DO"FORINPUTAS1
70 INPUT#1,C,L:N=L
80 PRINT"GAS AND OIL USAGE FOR "C:PRINT
90 DIMDD(L+1),Y(L+1,3):INPUT#1,Y(0,2)
    
```

Listing 2 continues



```

100 IFL=0THEN160
110 FORJ=1TOL
120 INPUT#1,D(J)
130 FORK=0TO3
140 INPUT#1,Y(J,K)
150 NEXTK:NEXTJ
160 CLOSE:GOTO 710
170 CLS:PRINTB;B;"MENU":PRINT
180 PRINTB;"1. PURCHASE OF GAS / OIL"
190 PRINTB;"2. PRINT MILES PER GALLON CH
ART"
200 PRINTB;"3. PRINT OIL USAGE":PRINT
210 INPUT"ENTER # OF YOUR CHOICE";Z
220 IFZ=1ANDU=1THENCLS:PRINT:PRINT"PRESS
<F4> FOR MORE ENTRIES":END
230 IFZ>4ORZ<1THENCLS:GOTO170
240 ONZGOTO250,500,600
250 CLS:N=N+1:INPUT"PRICE/GALLON OF GAS"
;P
260 INPUT"CURRENT MILEAGE";M
270 INPUT"NUMBER OF GALLONS PUCHASED";Q
280 PRINT "COST SHOULD BE";:PRINTUSING"$
###.##";P*Q:U=1
290 INPUT"IS DATA ENTERED CORRECT? Y/N";
A
300 IFA="Y"GOTO320
310 IFA="N"GOTO250ELSE290
320 X=(M-Y(L,2))/Q:PRINT
330 PRINT"LAST TANK AVERAGED";:PRINTUSIN
G"###.##";X;:PRINT"MPG":PRINT
340 INPUT"ADD OIL Y/N";H
350 IFH="Y"GOTO370
360 IFH="N"GOTO380ELSE340
370 INPUT "HOW MANY QUARTS OF OIL ADDED"
;V
380 D(N)=LEFT$(DATE$,5)
390 Y(N,0)=Q
400 Y(N,1)=P
410 Y(N,2)=M

```

```

420 Y(N,3)=V
430 OPEN"RAM:GASOIL.DO"FOROUTPUTAS1
440 PRINT#1,C;"",N;"",Y(0,2)
450 FORJ=1TOL
460 PRINT#1,D(J)
470 FORK=0TO3
480 PRINT#1,Y(J,K)
490 NEXTK:NEXTJ:CLOSE:GOTO710
500 CLS:PRINT"TANK DATE MPG $/GA
L $/MILE
510 FORK=1TON
520 T=(Y(K,2)-Y(K-1,2))/Y(K,0)
530 PRINTUSING"###.##";K;:PRINT" ";D(K);:P
RINTUSINGF;T;:PRINTUSINGE;Y(K,1);
540 IFT=0THENPRINTUSINGE;TELSEPRINTUSING
E;Y(K,1)/T
550 R=R+Y(K,0)
560 NEXTK
570 PRINT:PRINT"OVERALL MPG AVERAGE IS -
";
580 PRINTUSINGF;(Y(N,2)-Y(0,2))/R
590 GOTO710
600 CLS:PRINTB,"OIL USAGE":PRINT
610 FORK=1TO1000:NEXT
620 PRINT " TANK DATE QTS/OIL MI
LES/QT"
630 FORK=1TON
640 IFY(K,3)=0THEN680
650 W=W+Y(K,3):S=(Y(K,2)-Y(0,2))/Y(K,3):
O=K
660 PRINTUSING"###.##";K;:PRINT" ";D(K
);:PRINT" ";Y(K,3);" ";
670 PRINTUSINGF;S
680 NEXTK
690 PRINT:PRINT"AVERAGE MILES/QUART OF O
IL -";
700 PRINTUSINGF;(Y(N,2)-Y(0,2))/W
710 PRINT:PRINT"PRESS ANY KEY FOR MENU"
720 IFINKEY$=""THEN720ELSE170
730 END

```

## Traveling Expenses

by Beve Woodbury  
80 Micro Technical Editor

This handy program cumulatively records all your travel expenses so you'll know if you are getting close to or going over budget. It can also provide a printout of your expenses.

### Running the Program

Set up a new expense account file by running Listing 3. Run this program only to establish a new expense file.

Choose the appropriate option from

the menu by entering the correspond- ing number. A pause occurs after the first menu while the computer reads the old file and prints the new one. Answer prompts as they appear—name, amount, dates, comments. Be as brief as possible in your answers. Do not include commas in your data.

The travel option provides two sub- options—automobile and other. When you choose the automobile option, the program requests mileage. Enter the mileage you have driven and the pro- gram prints the charge. (The charge- per-mile is set at twenty cents. You can change it in line 140.)

Always end the program by going to

the totals-end menu. This step prepares the file for your next trip. If you don't do this, you may lose your data file.

The totals menu totals expenses for

#### Decisions:

- D Menu choice
- DD Has EXPFIL been rewritten to EXPNEW
- H Hardcopy
- J Travel—automobile or other
- V Miscellaneous or specific total

#### General:

- A Category accumulator
- B Grand total accumulator
- C Charge
- E Automobile mileage
- F Misc. special category accumulator
- G Hardcopy number print format
- K Start character of file
- L Category indicator to search for
- N Name
- P Special miscellaneous category
- U Read EXPFIL and write EXPNEW
- X Comments
- Y Dates
- Z Category indicator

Table 3. ORIG.BAII

#### Program Listing 3. ORIG.BAII

```

10 CLS:DEFSTR G-Z:G="#####.###"
20 OPEN"RAM:EXPFIL.DO"FORINPUTAS1:OPEN"R
AM:EXPNEW.DO"FOROUTPUTAS2:INPUT#1,K:PRIN
T#2,K
30 CLS:PRINTTAB(10)"MENU":GOSUB430:PRINT
TAB(5)"6. TOTALS OR END"
40 INPUT"ENTER # OF YOUR CHOICE: ";D
50 IFD<6THENDD=1:GOTO70
60 IFD=6ANDDD=1THEN180ELSE200

```

Listing 3 continues



Listing 3 continued

```

70 IFEOF(1) THEN 90
80 INPUT#1,U:PRINT#2,U:GOTO70
90 ONDGOTO100,110,120,130,170
100 CLS:INPUT"HOTEL NAME";N:Z="H":GOTO410
110 CLS:INPUT"RESTAURANT NAME";N:Z="M":GOTO410
120 CLS:INPUT"ENTERTAINMENT NAME";N:Z="E":GOTO410
130 CLS:PRINTTAB(17)"TRAVEL":PRINT:INPUT"
AUTOMOBILE OR OTHER? A/O: ";J
140 IFJ="A" THEN INPUT"MILEAGE: ";E:C=E*.2
0:PRINT"CHARGE IS ";C:N="AUTO":GOTO160
150 INPUT"MEANS OF TRANSPORTATION ";N:IN
PUT"CHARGE";C
160 INPUT"DATE";Y:INPUT"COMMENTS";X:Z="T
":GOTO420
170 CLS:PRINTTAB(10)"MISCELLANEOUS":PRIN
T:INPUT"MISC. CATEGORY NAME";N:Z="S":GOT
O410
180 CLOSE:KILL"EXPFIL.DO":NAME"RAM:EXPN
EW.DO"AS"RAM:EXPFIL.DO"
190 CLS:A=0:B=0:L=" ":V=" ":P=" ":OPEN"R
AM:EXPFIL.DO"FORINPUTAS1:INPUT#1,K
200 PRINTTAB(12)"TOTALS FOR:":GOSUB430:P
RINTTAB(5)"6. GRAND TOTAL 7. END"
210 INPUT"ENTER # OF YOUR CHOICE: ";D
220 IFD=7 THEN:END
230 INPUT "DO YOU WANT A HARD COPY? Y/N:
";H:IFH="Y" THEN LPRINT:LPRINT
240 ONDGOTO250,270,290,310,330,380
250 CLS:PRINT"CALCULATING LODGINGS TOTAL
":PRINT:IFH="Y" THEN LPRINT"LODGINGS:":LPR
INT
260 L="H":GOSUB450:PRINT"LODGING TOTAL="
;A:INPUT;G:GOTO190
270 CLS:PRINT"CALCULATING MEALS TOTAL":P
RINT:IFH="Y" THEN LPRINT"MEALS:":LPRINT
280 L="M":GOSUB450:PRINT"MEALS TOTAL=";A
:INPUT;G:GOTO190
290 CLS:PRINT"CALCULATING ENTERTAINMENT

```

```

TOTAL":PRINT:IFH="Y" THEN LPRINT"ENTERTAIN
MENT:":LPRINT
300 L="E":GOSUB450:PRINT"ENTERTAINMENT T
OTAL=";A:INPUT;G:GOTO190
310 CLS:PRINT"CALCULATING TRAVEL TOTAL":
PRINT:IFH="Y" THEN LPRINT"TRAVEL:":LPRINT
320 L="T":GOSUB450:PRINT"TRAVEL TOTAL=";
A:INPUT;G:GOTO190
330 CLS:PRINT"ENTER M FOR MISCELLANEOUS
TOTAL":INPUT"ENTER S FOR TOTAL ON A SP
ECIFIC NAME ";V:IFV="M" THEN 360
340 INPUT"CATEGORY NAME: ";P:PRINT:PRINT
"CALCULATING ";P;" TOTAL":IFH="Y" THEN LPR
INTP":LPRINT
350 PRINT:GOSUB450:PRINTP" TOTAL=";A:INP
UT;G:GOTO190
360 CLS:PRINT"CALCULATING MISCELLANEOUS
TOTAL":PRINT:IFH="Y" THEN LPRINT"MISCELLAN
EOUS:":LPRINT
370 L="S":GOSUB450:PRINT"MISCELLANEOUS T
OTAL=";A:INPUT;G:GOTO190
380 CLS:-PRINT"CALCULATING GRAND TOTAL":
PRINT:IFH="Y" THEN LPRINT"TOTAL EXPENSE IS
: ";
390 GOSUB450:PRINT"TOTAL EXPENSE IS ";B:
INPUT;D:IFH="Y" THEN LPRINTB
400 GOTO190
410 INPUT"CHARGE: ";C:INPUT"DATES: ";Y:I
NPUT"COMMENTS";X
420 -
430 PRINTTAB(5)"1. LODGINGS":PRINTTAB(5)
"2. MEALS":PRINTTAB(5)"3. ENTERTAINMENT"
440 PRINTTAB(5)"4. TRAVEL":PRINTTAB(5)"5
. MISCELLANEOUS":RETURN
450 IFEOF(1) THEN CLOSE:RETURN
460 INPUT#1,Z,N,C,Y,X
470 B=B+C:IFL=Z THEN A=A+C:IFH="Y" THEN LPR
INTTAB(3)N;TAB(18);:LPRINTUSINGG;C;:LPRIN
TTAB(3)Y;TAB(5)Y;X
480 IFP=N THEN F=F+C:A=F:IFH="Y" THEN LPRINT
TAB(5)N;TAB(25);:LPRINTUSINGG;C;:LPRINTT
AB(4)Y;TAB(55);X
490 GOTO450

```

each category or provides a grand total of all expenses.

The miscellaneous option lets you print a total of all miscellaneous expenses or a total of any recurring expenses in the miscellaneous file (like tolls). You can obtain a printout of every total. The grand total printout prints only the grand total amount. If you want a printout of individual expenses, print each category's total. ■

## Punch Out

Conversion by Mare-Anne Jarvela  
80 Micro Technical Editor

This payroll program, Listing 4, tallies timecards and calculates certain payroll deductions. It is a conversion of James J. Conroy's program (*80 Micro-computing*, February 1981, p. 198).

The program has a looping routine

based on how many time-in/time-out entries the operator decides to make. The screen first displays "How many periods are there?" After the appropriate number is entered, the computer asks for the time in (including a.m. or p.m.) and the time out. Carefully enter each time without any punctuation. You must also hit enter after each entry.

When done correctly, the computer displays the time worked in hours and minutes and in total minutes. After the last time period is entered, the computer pauses and a list of all the payroll entries appears in total minutes. The total hours and minutes worked are then displayed.

Then the hourly pay rate is entered and the gross pay appears on the screen.

The withholding deductions are calculated and the final deductions and net pay appear on the screen. FICA, federal withholding, and state and local taxes are calculated.

Q	Number of time periods	G	Pay rate
T	Total	GP	Gross pay
T1	Time punched in	F	Federal deduction
T2	Time punched out	Soc	Social Security
Z1	a.m. or p.m. (punch in)	ST	State tax
Z2	a.m. or p.m. (punch out)	CT	Local tax
TMIN	Total minutes left	A\$	INKEY (Y/N)
HTT	Hours until twelve	X	Loop counter
HFT	Hours from twelve to time out	Y	Loop counter
THRS	Total hours	D	Counter

Table 4. Punch Out Variables

```

4 REM PUNCH-TIME CLOCK PROGRAM FOR THE M
ODEL-100
10 CLS: CLEAR
25 INPUT "HOW MANY TIME PERIODS ARE THER
E"; Q
27 DIM L(Q)
30 T=0
35 FOR D=0 TO Q: IF D=Q THEN GOTO 330
40 PRINT: INPUT "TYPE THE TIME PUNCHED IN
"; T1
45 INPUT "ENTER AM OR PM (A/P)"; Z1$
50 INPUT "TYPE THE TIME PUNCHED OUT"; T2
55 INPUT "ENTER AM OR PM (A/P)"; Z2$
60 CLS
70 FOR X=100 TO 1200 STEP 100
75 IF T1-X < 60 THEN GOTO 90
80 NEXT X
90 FOR Y=100 TO 1200 STEP 100
95 IF T2-Y < 60 THEN GOTO 110
100 NEXT Y
110 IF Z1$=Z2$ THEN GOTO 125
115 IF Z1$ <> Z2$ THEN GOTO 215
125 IF T1-X > 0 THEN 160 ELSE 135
135 HRS=Y*.01-X*.01: IF X*.01=12 THEN HRS
=HRS+12
140 PRINT "TOTAL TIME WAS"; HRS; "HOURS AND
"; T2-Y, "MINUTES"
145 PRINT "OR: "; HRS*60+(T2-Y); "TOTAL MINU
TES"
150 L(D)=HRS*60+(T2-Y): NEXT D
160 HRS=Y*.01-X*.01-1: IF X*.01=12 THEN H
RS=HRS+12
165 MIN=60-(T1-X)
180 TMIN=MIN+(T2-Y)
190 IF TMIN > 60 THEN TMIN=TMIN-60: HRS=HR
S+1
195 PRINT "TOTAL TIME WAS: "; HRS; "HOURS A
ND"; TMIN; "MINUTES"
200 PRINT "OR: "; HRS*60+TMIN; "TOTAL MINUTE
S"
205 L(D)=HRS*60+TMIN: NEXT D
215 IF T1-X > 0 THEN 275 ELSE 230
230 HTT=12-X*.01
245 HFT=Y*.01: IF Y*.01=12 THEN HFT=HFT-
12
250 THRS=HTT+HFT
255 PRINT "TOTAL TIME WAS: "; THRS; "HOURS A
ND"; T2-Y; "MINUTES"
260 PRINT "OR: "; THRS*60+(T2-Y); "TOTAL MIN

```

```

UTES"
265 L(D)=THRS*60+(T2-Y): NEXT D
275 HTT=12-X*.01-(1)
280 MIN=60-(T1-X)
290 TMIN=(T2-Y)+MIN
295 HFT=Y*.01: IF Y*.01=12 THEN HFT=HFT-1
2
300 IF TMIN > 60 THEN TMIN=TMIN-60: HFT=HF
T+1
310 THRS=HTT+HFT
315 PRINT "TOTAL TIME WAS"; THRS; "HOURS AN
D"; TMIN; "MINUTES"
320 PRINT "OR: "; THRS*60+(TMIN); "TOTAL MIN
UTES"
325 L(D)=THRS*60+(TMIN): NEXT D
330 FOR R=1 TO 2000: NEXT: CLS: PRINT "CALCU
LATING TOTAL TIME FOR PAY"
335 FOR C=0 TO D-1
340 PRINT L(C): NEXT C
345 FOR C=0 TO D-1
350 T=T+L(C): NEXT C
355 PRINT T; " (TOTAL MINUTES)"
360 PRINT "TOTAL TIME WAS"; T/60; "HOURS"
362 PRINT: PRINT: PRINT "PRESS ENTER TO C
ONTINUE";: INPUT Y$
365 CLS: INPUT "WHAT IS YOUR RATE OF PAY";
G
370 GP=INT((G*(T/60))*100+.5)/100
375 PRINT "GROSS PAY IS: ";: PRINT USING "$$
###.##"; GP
380 INPUT "ENTER THE FED. DEDUCTION"; F
382 PRINT: PRINT: PRINT "PRESS ENTER TO CO
NTINUE";: INPUT Y$
385 CLS: PRINT "***** DEDUCTIONS ARE:"
390 SOC=INT((GP*.0665)*100+.5)/100: PRINT
"1. FICA (SOC. SEC.) IS: "; SOC
395 PRINT "2. FED. WITHHOLDING IS: "; F
400 ST=INT((GP*.022)*100+.5)/100: PRINT "3
. STATE TAX IS: "ST
405 CT=INT((GP*.01)*100+.5)/100: PRINT "4.
LOCAL TAX IS: "CT
410 PRINT "TOTAL DEDUCTIONS ARE"; SOC+ST+F
+CT
415 PRINT "***** NET PAY IS: "; GP-SOC-F-S
T-CT
420 INPUT "DO YOU WANT TO RUN ANOTHER PAY
? (Y/N)"; A$
425 IF A$="Y" THEN CLS: CLEAR: GOTO 25
430 END

```

Program Listing 4. Punch Out

You can change these deductions in the program to suit your individual needs. For users who don't want to look up the federal withholding deduction from a schedule, use the classical percentage method to figure the correct deduction.

The program has four subroutines that encompass all possible combinations of time-in/time-out entries for figuring elapsed work time. The total work time is accurate, since it is figured from the total minutes worked.

If you want a written record, change the Print statements in the program to LPRINT. If you want to calculate overtime pay you can put provisions in to detect and account for it. ■

## Itinerary 100

Conversion by Brad Dixon  
80 Micro Technical Editor

In today's mobile society, applications for portable computers have come to the forefront. This menu-driven itinerary program (Listing 5) is a conversion of a program by Ben Gorsky (*80 Microcomputing*, April 1980, p. 95). It requires at least 10K of usable RAM, including memory space for data files.

Two files store travel information: the first keeps track of flights, hotels, car rentals, and other reservations; the second stores payment records. These files are stored in RAM as listed in the

program, but could be stored on tape by changing the RAM file statement to a tape file statement. The list of variables used in the program appears in Table 5.

Program housekeeping is done in line 10, with the title page and main menu following in lines 20-100. Option 1 in the main menu loads the data recorded during earlier sessions in RAM. Options 2-6 begin by checking the file data for previous entries. If none are found, you are notified and given the option to add file information. If data is found, it is displayed. You can then add or delete file records, or leave the file as it is and return to the main menu.

Option 7 directs the program to the printer routine, giving a hard copy of all

the data in your itinerary in the same order as in the main menu. After all data for your travel itinerary has been entered, Option 8 loads it into RAM files for retrieval as needed.

### Special Instructions and Modifications

When entering hotel addresses under Option 3, you must put slashes (/) between the lines in the address. These signal the printer to start a new line under Option 7. Under the flights option (2), enter the time in four numbers without a colon between the hours and minutes. Enter dates for Options 2, 3, 5, and 6 in the conventional format (MM/DD/YY or MM-DD-YY).

Each session ends when the input information is stored in RAM files. If you press the break key at any time before using Option 8, all your itinerary data for that session will be erased.

Modifications to Itinerary 100 are limited only by your specific needs and the amount of memory available in your Model 100. However, one useful modification is to randomly access the files to list all data pertaining to a particular date or location in your travels. ■

NFT	Number of flights
FDATE(I)	Date of 1st flight
ALN(I)	Airline
DPCIT(I)	Departure city
DEPTM(I)	Departure time
NM(I)	Flight number
ACIT(I)	Arrival city
ATM(I)	Arrival time
NHOT	Number of hotels
ADAT(I)	Arrival date at 1st hotel
HOTNAM(I)	Name
DR(I)	Address
DDAT(I)	Departure date
NO	Number of other reservations
OT(I)	Data for other reservations
NACT	Number of accounting items
WHO(I)	Name of party to pay for 1st item
DATDUE(I)	Date payment due
STSNT(I)	Date payment sent
DOLDEP(I)	Amount of payment
NCAR	Number of rental cars
RSNM(I)	Reservation number for 1st reservation
AGNCY(I)	Rental agency
PDAT(I)	Pick-up date
PLC(I)	Pick-up location
RDAT(I)	Return date
RLC(I)	Return location
TRPNM	Trip name or title

Table 5. Itinerary 100 Variables

### Program Listing 5. Itinerary 100

```

10 CLEAR500:DEFINT I,N,K:DEFSTR
F,A,D,Z,H,O,P,W,R,T
20
CLS:PRINT@135,"ITINERARY":PRINT@171,"FOR
THE MODEL 100"
22 FOR Y=1 TO 1000: NEXT Y
100 CLS:PRINT" YOU MAY SELECT ANY OF
THESE FUNCTIONS":PRINTTAB(8)"1-INPUT
RESERVATION DATA":PRINTTAB(6)"2-FLIGHT
DATA":PRINT@101,"3-HOTEL DATA":PRINTTAB(
2)"4-OTHER RESERVATIONS":PRINT@144,"5-
CAR RENTALS"
110 PRINTTAB(8)"6-DEPOSITS AND
PAYMENTS":PRINTTAB(3)"7-PRINT
ITINERARY":PRINT@222,"8-RECORD DATA"
120 PRINT:INPUT"YOUR CHOICE";K:ON K GOTO
150,200,410,610,810,1000,1200,1510
150 'INPUT DATA FROM FILE
160 CLS:INPUT"HIT ENTER TO LOAD DATA
FILE";Z:OPEN"RAM:ITINER.DO"FORINPUT AS
1:INPUT#1,NFT,NHOT,NO,NACT,NCAR:FORI=1TO
NFT:INPUT#1,FDATE(I),ALN(I),DPCIT(I),
DEPTM(I),NM(I),ACIT(I),ATM(I):NEXT
170 FOR I=1TONHOT:IFEOF(1)
THEN171:INPUT#1,ADAT(I),HOTNAM(I),DR(I)
,DDAT(I):NEXT
171 FOR I=1TONO:IFEOF(1)
THEN172:INPUT#1,OT(I):NEXT
172 FOR I=1TONACT:IFEOF(1)
THEN180:INPUT#1,WHO(I),DATDUE(I),DTSNT(I)
),DOLDEP(I):NEXT
180 FOR I=1TONCAR:IFEOF(1)
THEN190:INPUT#1,RSNM(I),AGNCY(I),PDAT(I)
,PLC(I),RDAT(I),RETL(I):NEXT:CLOSE:GOTO
100
190 CLOSE:GOTO100

```

```

200 CLS:PRINT:IF NFT=0 THEN PRINT@49,"NO
FLIGHTS LISTED":PRINT:GOTO 230
220
FORI=1TONFT:CLS:PRINT@56,"FLIGHTS":PRINT
@80,"#":PRINT@106,"DATE":PRINT@120,"FLT
.#":PRINT@126,"AIRLNE":PRINT@136,"DEPART
":PRINT@150,"ARRIVE"
225 PRINT@82,I:PRINT@112,FDATE(I)
:PRINT@160,NM(I):PRINT@166,ALN(I)
:PRINT@173,DPCIT(I):PRINT@182,DEPTM(I)
:PRINT@187,ACIT(I):PRINT@196,ATM(I)
226 PRINT:INPUT"ENTER <C>ONTINUE OR <R>
ETURN";A$:IF
A$="R"THEN230ELSEIFA$="C"THENNEXTIELSE22
6
230 FOR
S=1TO1000:NEXTS:CLS:PRINT@96,"YOU
MAY":PRINT"<A>DD, <D>ELETE, OR <M>
AINTAIN LISTINGS.":PRINT:INPUT"YOUR
SELECTION";Z$:IFZ$="A"THEN 240 ELSE IF
Z$="D"THEN 320 ELSE IF Z$="M" THEN 100
ELSE 230
240 CLS:PRINTTAB(16)"FLIGHTS":PRINT"TO
ADD A FLIGHT BETWEEN 2
EXISTING":PRINT"FLIGHTS ENTER THE NUMBER
OF THE NEW":PRINT"FLIGHT IN
SEQUENCE.":PRINT"TO ADD A FLIGHT AT THE
END, ENTER THE":PRINT"NEXT NUMBER IN
THESEQUENCE.":PRINT:INPUT"FLIGHT
NUMBER";K
250 IF K>NFT THEN NFT=NFT+
1:K=NFT:GOTO270
260 NFT=NFT+1:FORI=NFT-1 TO K STEP -
1:FDATE(I+1)=FDATE(I):ALN(I+1)=ALN(I)
:DPCIT(I+1)=DPCIT(I):DEPTM(I+1)=DEPTM(I)
:NM(I+1)=NM(I):ACIT(I+1)=ACIT(I):ATM(I+1)
)=ATM(I):FCOM(I+1)=FCOM(I):NEXT

```

Listing 5 continues

Listing 5 continued

```
270 INPUT "ENTER DATE OF FLIGHT";FDATE(K
)
275 INPUT"ENTER AIRLINE NAME";ALN(K)
280 INPUT"ENTER FLIGHT NUMBER";NM(K)
285 INPUT"ENTER DEPARTURE CITY";DPCIT(K)

290 INPUT"ENTER DEPARTURE TIME";DEPTM(K)

295 INPUT"ENTER ARRIVAL CITY";ACIT(K)
300 INPUT"ENTER ARRIVAL TIME";ATM(K)
310 GOTO 200
320 PRINT:PRINT"ENTER THE
NUMBER":INPUT"OF THE FLIGHT TO BE
DELETED";J:NPT=NPT-1:FOR I=JTO NPT:FDATE
(I)=FDATE(I+1):ALN(I)=ALN(I+1):DPCIT(I)
=DPCIT(I+1):DEPTM(I)=DEPTM(I+1):NM(I)=NM
(I+1):ACIT(I)=ACIT(I+1)
330 ATM(I)=ATM(I+1):NEXT:GOTO200
410 CLS:IFNHOT=0THEN PRINT@52,"NO HOTELS
LISTED":GOTO430
420 FORI=1TONHOT:CLS:PRINTTAB(17)
"HOTELS":PRINT:PRINTI;" ";HOTNAM(I)
,"ARRIVE ";ADAT(I):PRINT, "DEPART ";DDAT
(I)
425 PRINT:INPUT"ENTER <C>ONTINUE OR <R>
ETURN";A$:IF A$="C"THEN NEXTIELSE IF
A$="R"THEN 430 ELSE425
430 FOR S=1TO500:NEXTS:CLS:PRINT@86,"YOU
MAY <A>DD, <D>ELETE, OR":PRINT@127,"<M>
AINTAIN HOTEL LISTINGS":PRINT:INPUT"YOUR
SELECTION";Z$:IFZ$="A"THEN440ELSEIF
Z$="D"THEN550 ELSEIFZ$="M"THEN100
ELSE430
440 CLS:PRINT:PRINT"TO ADD A HOTEL
BETWEEN TWO IN THE LIST,":PRINT"ENTER
THE NEWHOTEL NUMBER."
445 PRINT"TO ADD A HOTEL TO THE
END,":PRINT"ENTER THE NEXT NUMBER IN
SEQUENCE.":PRINT:INPUT"HOTEL
NUMBER";K:IFK>NHOT THEN NHOT=NHOT+
1:GOTO460
450 NHOT=NHOT+1:FOR I=NHOT-1 TO KSTEP-
1:HOTNAM(I+1)=HOTNAM(I):ADAT(I+1)-ADAT(I)
):DDAT(I+1)=DDAT(I):DR(I+1)=DR(I)
:NEXT460 INPUT"ENTER NAME OF
HOTEL";HOTNAM(K)
470 PRINT"ENTER ADDRESS OF HOTEL":INPUT"
(PUT '/' BETWEEN ADDRESS LINES";DR(K)
480 INPUT"ENTER ARRIVAL DATE";ADAT(K)
490 INPUT"ENTER DEPARTURE DATE";DDAT(K)
:GOTO410
550 INPUT"ENTER NUMBER OF HOTEL TO BE
DELETED";K
560 FOR I=KTONHOT:HOTNAM(I)=HOTNAM(I+1)
:DR(I)=DR(I+1):ADAT(I)=ADAT(I+1):DDAT(I)
=DDAT(I+1):NEXT:NHOT=NHOT-1:GOTO410
610 CLS:PRINT:IF NO=0 THENPRINT@86,"NO
OTHER RESERVATIONS LISTED":GOTO 630
620 CLS:FORI=1 TO NO:PRINT@45,I;"
";OTH(I):PRINT:PRINT:INPUT"ENTER <C>
ONTINUE OR <R>ETURN";A$:IF
A$="C"THENNEXTIELSEIFA$="R"THEN630
630 FOR S=1TO500:NEXTS:CLS:PRINT@88,"YOU
MAY <A>DD, <D>ELETE,":PRINT@124,"OR <M>
AINTAIN OTHER
RESERVATIONS":PRINT:PRINT:INPUT"YOUR
SELECTION";Z$:IFZ$="A"THEN640ELSEIFZ$="D
"THEN670 ELSEIFZ$="M"THEN100 ELSE 630
640 NO=NO+1:PRINT:INPUT"ENTER
RESERVATION INFORMATION";OTH(NO):GOTO610

670 PRINT:INPUT"ENTER RESERVATION TO BE
DELETED";K:NO=NO-1:FORI=KTONO:OTH(I)
```

```
=OTH(I+1):NEXT:GOTO610
810 CLS:IFNCAR=0THEN PRINT:PRINT@89,"NO
CAR RENTALS LISTED":PRINT:GOTO830
820 FOR I=1TONCAR:CLS:PRINT@54,"CAR
RENTALS":PRINT@77,"#I:PRINT@83,"AGENCY:
"AGNCY(I):PRINT@105,"REG.#: "RSNM(I)
:PRINT@165,"PICKUP:PDAT(I)
:PRINT@185,PLC(I):PRINT@205,"RETURN:
"RDAT(I):PRINT@225,RETLC(I)
825 PRINT:INPUT"ENTER <C>ONTINUE OR <R>
ETURN";A$:IFA$="C"THEN NEXTIELSE
IFA$="R"THEN830ELSE825
830 FOR S=1TO500:NEXTS:CLS:PRINT@88,"YOU
MAY <A>DD, <D>ELETE,":PRINT@127,"OR <M>
AINTAIN CAR
RENTALS":PRINT:PRINT:INPUT"YOUR
SELECTION";Z$:IFZ$="A"THEN840ELSE
IFZ$="D"THEN950 ELSEIFZ$="M"THEN100 ELSE
830
840 CLS:PRINT"TO ADD A RENTAL BETWEEN
TWO ":PRINT"EXISTING ONES, ENTER THE
NUMBER ":PRINT"FOR THE NEW
RENTAL.":PRINT"TO ADD A NEW RENTAL TO
THE END ":PRINT"OF THE LIST, ENTER THE
NEXT ":PRINT"NUMBER IN THE
SEQUENCE.":PRINT
845 INPUT"RENTAL NUMBER";K
850 IFK>NCAR THEN NCAR=NCAR+
1:K=NCAR:GOTO870
860 NCAR=NCAR+1:FORI=NCAR TO KSTEP-
1:RSNM(I+1)=RSNM(I):AGNCY(I+1)=AGNCY(I)
:PDAT(I+1)=PDAT(I):PLC(I+1)=PLC(I):RDAT(I
+1)=RDAT(I):RETLC(I+1)=RETLC(I):NEXT
870 INPUT"ENTER AGENCY NAME";AGNCY(K)
880 INPUT"ENTER RESERVATION NUMBER";RSNM
(K)
890 INPUT"ENTER PICK UP DATE";PDAT(K)
900 INPUT"ENTER PICK UP LOCATION";PLC(K)

910 INPUT"ENTER RETURN DATE";RDAT(K)
920 INPUT"ENTER RETURN LOCATION";RETLC(K)
:GOTO810
950 INPUT"ENTER RENTAL NUMBER TO BE
DELETED";J:NCAR=NCAR-
1:FORI=JTONCAR:RESNM(I)=RESNM(I+1):AGNCY
(I)=AGNCY(I+1):PDAT(I)=PDAT(I+1):PLC(I)
=PLC(I+1):RDAT(I)=RDAT(I+1):RETLC(I)
=RETLC(I+1):NEXT
960 GOTO810
1000 CLS:IFNACT=0THEN PRINT@48,"NO
PAYMENT ITEMS LISTED":GOTO1030
1020 FORI=1TONACT:
CLS:PRINT@56,"PAYMENTS":PRINT@83,"#I:PR
INT@103,"AMOUNT: $"DOLDEP(I)
:PRINT@132,"TO: "WHO(I):PRINT@160,"DATE
DUE: "DATDUE(I):PRINT@180,"DATE SENT:
"DTSNT(I)
1025 PRINT:INPUT"ENTER <C>ONTINUE OR <R>
ETURN";S$:IFSS="C"THEN
NEXTIELSEIFSS="R"THEN1030ELSE1025
1030 FORS=1TO500:NEXTS:CLS:PRINT@88,"YOU
MAY <A>DD, <D>ELETE,":PRINT@129,"OR <M>
AINTAIN PAYMENTS":PRINT:INPUT"YOUR
SELECTION";Z$:IFZ$="A"THEN1040
ELSEIFZ$="D"THEN1100 ELSEIFZ$="M"THEN100
ELSE1030
1040 NACT=NACT+1
1050 PRINT:PRINT"ENTER NAME OF AGENCY
REQUIRING PAYMENT":INPUTWHO(NACT)
1060 INPUT"ENTER AMOUNT DUE";DOLDEP(NACT)
)
1070 INPUT"ENTER DATE DUE";DATDUE(NACT)
1080 INPUT"ENTER DATE SENT";DTSNT(NACT)
```

Listing 5 continues

```

:GOTO1000
1100 INPUT"ENTER PAYMENT NUMBER TO BE
DELETED";J:NACT=NACT-1:FORI=JTONACT:WHO(
I)=WHO(I+1):DOLDEP(I)=DOLDEP(I+1):DATDUE
(I)=DATDUE(I+1):DTSNT(I)=DTSNT(I+1)
:GOTO1000
1200 CLS:PRINT@47,"ENTER TITLE FOR
ITINERARY":PRINT:INPUTTRPNM
1220 INPUT"HIT <ENTER> WHEN THE PRINTER
IS READY";Z:LPRINTSTRING$(4,"-")
1230 LPRINTSTRING$(69,"X")
:LPRINT:LPRINTTAB(72-LEN(TRPNM))/2)
TRPNM:LPRINT:LPRINTSTRING$(69,"X")
:LPRINT:LPRINT:LPRINT
1240 LPRINTTAB(25)"F L I G H T S":LPRINT

1250 FOR I=1TONFT:LPRINT FDATE(I);TAB(20
)ALN(I);TAB(26)"DEP ";DPCIT(I);TAB(50)
"AT ";DEPTM(I):LPRINTTAB(20)"#";NM(I)
;TAB(26)"ARV ";ACIT(I);TAB(50)"AT ";ATM(
I):LPRINT:NEXT
1270 LPRINT:LPRINT:LPRINTTAB(25)"H O T E
L S":LPRINT
1280 FORI=1TONHOT:LPRINT HOTNAM(I);TAB(
20)"ARV ";ADAT(I);TAB(40)"DEP ";DDAT(I)
:A=DR(I)
1290 FORJ=1TOLEN(A):IFMID$(A,J,1)<>"/
"THENNEXT:GOTO1310
1300 F=LEFT$(A,J-1):LPRINT F:A=RIGHT$(

```

```

A,LEN(A)-J):GOTO1290
1310 LPRINT A:LPRINT:NEXT:LPRINT:LPRINT
1320 LPRINTTAB(15)"C A R R E N T A L
S":LPRINT
1330 FORI=1TONCAR:LPRINT AGENCY(I);TAB(15
)"PICK UP ";PDAT(I);" AT ";PLC(I)
:LPRINTRSNM(I);TAB(15)"RETURN ";RDAT(I)
;" AT ";RETL(I)
:LPRINT:NEXT:LPRINT:LPRINT:LPRINT
1340 LPRINTTAB(25)"O T H E R":LPRINT
1350 FORI=1TONO:LPRINT OT(I):LPRINT:NEXT

1360 LPRINT:LPRINT:GOTO1000
1510 CLS:PRINTTAB(20)"RECORDING DATA"
1520 OPEN"RAM:ITINER.DO"FOROUTPUTAS1
1525
PRINT#1,NFT",";NHOT",";NO",";NACT",";NCA
R:FORI=1TONFT:PRINT#1,FDATE(I)",";ALN(I)
",";DPCIT(I)",";DEPTM(I)",";NM(I)",";ACIT
(I)",";ATM(I):NEXT
1530
FORI=1TONHOT:PRINT#1,ADAT(I)",";HOTNAM(I)
)",";DR(I)",";DDAT(I):NEXT
1540 FORI=1TONO:PRINT#1,OT(I):NEXT
1550
FORI=1TONACT:PRINT#1,WHO(I)",";DATDUE(I)
)",";DTSNT(I)",";DOLDEP(I):NEXT
1560
FORI=1TONCAR:PRINT#1,RSNM(I)",";AGENCY(I)
)",";PDAT(I)",";PLC(I)",";RDAT(I)",";RETL
C(I):NEXT:CLOSE:GOTO1000

```

## The Final Notice

Conversion by Mare-Anne Jarvela  
80 Micro Technical Editor

This payment-scheduling program (Listing 6) is a conversion of one by Walter J. Atkins, which appeared in the February 1981 issue of *80 Microcomputing* (p. 200). It displays all accounts due on the 1st, the 15th, or between any other two dates of the month.

The account files are stored in data statements at the end of the program. The format is: Line# DATA "account name", "account #", due date, amount due. See lines 600-680 in the program listing.

After you choose a method of reporting, the program asks if the account numbers are to be displayed. N suppresses the numbers.

At the end of your display, you'll see the total number of accounts due and the total amount due. ■

D	Option variable
D1	Start date
D2	Stop date
A	Amount due
T	Total amount
C	Counter
H	Counter
F	Counter
A\$	Account number
N\$	Name
I\$	INKEY

Table 6. The Final Notice Variables

### Program Listing 6. The Final Notice

```

10 REM PAYMENT DUE DATE
20 REM FOR THE MODEL-100 APRIL 83
70 CLS:PRINTTAB(10)"PAYMENT DUE DATES"
80 PRINT"FUNCTIONS AVAILABLE ARE:"
90 PRINT" 1. ACCOUNTS DUE 1ST OF MONTH
100 PRINT" 2. ACCOUNTS DUE 15TH OF MON
TH
110 PRINT" 3. ACCOUNTS DUE BETWEEN ANY
TWO DATES
120 PRINTTAB(10)"SELECT ONE OPTION ":INP
UT"==>";D:CLS
130 IF D>3 OR D<1 THEN 70
140 PRINT:PRINT
150 PRINT"DO YOU WANT ACCOUNT NUMBERS DI
SPLAYED (Y OR N)";
160 INPUT I$:I$=LEFT$(I$,1)
170 IF I$<>"Y"AND I$<>"N"THEN CLS:GOTO1
40
180 CLS
190 ON D GOTO 200,240,280
200 GOSUB 450
210 IF D<17THEN GOSUB 480
220 IF F<>1 THEN 200
230 GOSUB 540:GOTO 60
240 GOSUB 450

```

```

250 IF D>=17 THEN GOSUB 480
260 IF F<>1 THEN 240
270 GOSUB 540:GOTO 60
280 PRINT:PRINT
290 INPUT"START DATE (MAX=31)";D1
300 IF D1>31 OR D1<1 THEN CLS:GOTO290
310 INPUT"STOP DATE (MAX=31)";D2
320 IF D2>31 OR D2<1 THEN CLS:GOTO 310
330 IF D2<D1 THEN CLS:GOTO290
340 CLS
350 GOSUB 450
360 IF D>=D1 AND D<=D2THEN GOSUB 480
370 IF F<>1 THEN 350
380 GOSUB 540:GOTO60
390 PRINTC;". ";N$;
400 PRINTTAB(18)"DUE DATE ";D;
410 PRINTTAB(45)"AMOUNT $";A
420 IF I$="Y"THEN PRINTTAB(6)"ACCOUNT NU
MBER : ";A$
430 PRINT
440 RETURN
450 READ N$,A$,D,A
460 IF N$="END"THEN PRINT"NUMBER ACCOUNT
S: ";C,"TOTAL DUE $";T:F=1
470 RETURN

```

Listing 6 continues

## MONITOR 100

Conversion by Amee Eisenberg  
80 Micro Technical Editor

You might be curious to find out what's inside the Model 100 and where it's stored. When Sergio Zigras had similar questions about the Color Computer, he wrote a monitor program in Basic (80 Micro, January 1983, p. 252). Here the monitor is modified for the Model 100 (see Listing 7).

The monitor offers five main menu options:

- M (Memory examine and modify)

- D (Dump a block of memory)
- F (Find a character or string of characters within a specified block of memory)
- C (Convert from hex to decimal or vice versa)
- E (End)

To conserve memory space, this program is unforgiving of input format errors. If you enter a command in lowercase characters or without its required parameters, a function call error results.

### Memory Examine and Modify

The input format is MXXXX (enter).

This command displays the contents of a specified hexadecimal address (XXXX) and allows you to change it.

The output format is:

```
HEX ADDRESS=XXXX
BYTE=xx
ASCII CHARACTER='(character)'
Next Action: H(igher), L(ower), R(estart),
(change to)XX?
```

The monitor responds to this command by showing the specified hexadecimal (hex) address (XXXX), the hex byte contained at that address (xx) and the ASCII character equivalent to

### Program Listing 7. Monitor 100

```
5 CLEAR 500:CLS:PRINT@85,"WARNING:Bad Input
Format Will Cause":PRINT@125,"Program To
Fail":PRINT@205,"READ THE MANUAL!"
10 GOTO800
20 H$="":GOSUB60:HB$=H$:RETURN
30 V$=HB$:DA=0:GOSUB90:DB=DA:RETURN
40 ER=0:IFV<48ORV>70THENER=1ELSEIFV<58THEN
V=V-48ELSEIFV>63THENV=V-55
45 RETURN
50 H$="":A=DA/4096:A=INT(A):GOSUB64:DB=DA-
4096*A
56 A=DB/256:A=INT(A):GOSUB64:DB=DB-256*A
58 DB=DB-256*A
60 A=DB/16:A=INT(A):GOSUB64:A=DB-16*A
64 AA=A:IFA>9THENA=A+55ELSEA=A+48
66 A$=CHR$(A):H$=H$+A$:A=AA:RETURN
72 HB$=HA$
74 V=ASC(HB$):GOSUB40:IFER=1THEN110ELSEDA=
4096*V:V$=MID$(HB$,2,1):V=ASC(V$):GOSUB40:
IFER=1THEN110
76 DB=256*V:DA=DA+DB:V$=MID$(HB$,3,1)
90 V=ASC(V$):GOSUB40:IFER=1THEN110ELSEDB=1
6*V:DA=DA+DB:V$=RIGHT$(HB$,1):V=ASC(V$):GO
SUB40:IFER=1THEN110ELSEDA=DA+V:RETURN
110 PRINT"ERROR,NOT HEX":RETURN
```

```
120 PA$="":FORX=1TO4:DB=PEEK(DA):GOSUB20:P
A$=PA$+HB$:DA=DA+1:NEXT:RETURN
130 A$="":FORX=1TO8:DB=PEEK(DA):IFDB<32THE
NDB=46
135 P$=CHR$(DB):A$=A$+P$:DA=DA+1:NEXT:RETU
RN
200 CLS:HA$=MID$(O$,3,4):GOSUB72:IFER=1THE
N10
210 SDA=DA:DB=PEEK(DA):XDB=DB:IFDB<32THENX
DB=46ELSEB$=CHR$(XDB)
215 CLS:GOSUB20:PRINT@46,"HEX ADDRESS=";HA
$:PRINT@133,"BYTE=";HB$:PRINT@202,"ASCII C
HARACTER=";B$;" "
220 INPUT" Next Action: H(igher),L(ower),R
(estart),(change to)XX";HB$
225 IFHB$="L"THEN275ELSEIFHB$="H"THEN260EL
SEIFHB$="R"THEN10
230 GOSUB30:IFER=1THEN255
240 POKE SDA,DB:CDB=PEEK(SDA):IFCDB<>DBTHE
NPRINT"NO CHANGE"
255 DA=SDA
260 DA=DA+1
265 GOSUB50:HA$=H$:GOTO210
275 DA=DA-1:GOTO265
300 BA$=MID$(O$,3,3):BA$=BA$+"0":EA$=MID$(
O$,8,4):HA$=EA$:GOSUB72:LA=DA:IFER=1THEN10
310 HA$=BA$:GOSUB72:IFER=1THEN10
320 IFMO=1THEN405ELSEGOSUBL20:FI$=PA$:GOSU
BL20:SE$=PA$:DA=DA-8:GOSUBL30:PRINTBA$;"
";FI$;" ";SE$;" "A$
340 IFDA>=LATHEN10
345 GOSUB50:BA$=H$:GOTO320
400 PRIN MO=1:GOTO300
405 BE=DA:MO=0:IN$=MID$(O$,13):CNT=LEN(IN$
)/2:CNT=INT(CNT)
415 BL$="":FORZ=1TOCNT:DB=PEEK(DA):GOSUB20
:BL$=BL$+H$:DA=DA+1:NEXT:IFBL$=IN$THEN450
440 BE=BE+1:DA=BE:IFBE=LATHEN10ELSEGOTO415
450 DA=BE:GOSUB50:PRINTIN$;" is at ";H$ "
";DA=DA+CNT:GOTO440
500 CO$=MID$(O$,3,1):IFCO$="D"THEN530ELSEH
$=MID$(O$,5):DA=VAL(H$):IFDA>65535THENPRIN
T"BEYOND RANGE":GOTO10
505 GOSUB50:PRINT@85,DA;"Dec.="H$;"Hex":GO
TO10
530 HB$=MID$(O$,5):L=LEN(HB$):IFL<>4THENPR
INT"4 HEX CHARS ONLY":GOTO10
535 GOSUB74:PRINT@85,HB$ Hex=";DA;"decima
l":GOTO10
800 PRINT:PRINT"MONITOR COMMANDS:M,D,F,C,E
":INPUTO$:C$=LEFT$(O$,1)
810 CLS:IFC$="M"THEN200ELSEIFC$="D"THEN300
ELSEIFC$="F"THEN400ELSEIFC$="C"THEN500ELSE
IFC$="E"THENEND
820 PRINT"WHAT?":GOTO10
```

### Listing 6 continued

```
480 IF N$="END"THEN 530
490 C=C+1
500 T=T+A
510 IF H=0THEN GOSUB 580:H=1
520 GOSUB 390
530 RETURN
540 PRINT:INPUT"CONTINUE (Y OR N) ";I$:I
$=LEFT$(I$,1)
550 IF I$<>"Y" AND I$<>"N"THEN 540
560 IF I$="N"THEN CLS:END
570 RETURN
580 PRINTTAB(15)" A C C O U N T S D U
E ":PRINT
590 RETURN
600 DATA"DOE&CO","A1234-RT",2,12.50
610 DATA"JONES","22233",5,1432.56
620 DATA"SMITH","3344E",22,65.89
630 DATA"TOM","SA22234",25,34.87
640 DATA"UNIVERSAL INDUSTRIES","123HH4",
30,56.54
650 DATA"HARRY","QW223-A",3,34.34
660 DATA"DICK","ABC 123",14,75.75
670 DATA"JANE","23-456",26,67.89
680 DATA"END","",0,0
690 END
```

that byte.

The program now offers four possible actions: H(igher), L(ower), R(estart) and (change to)XX.

H displays the next higher address (XXXX + 1) with its contents and ASCII character.

L displays the next lower address (XXXX - 1) with its contents and ASCII character.

R restarts the program by returning to the main menu.

XX changes the contents of the current memory location, then displays the next higher memory address with its contents. If you attempt a change to a read-only memory (ROM) location, the screen flashes "No Change" and displays the next higher memory location.

### Dump a Memory Block

The input format is D XXXX YYYY (enter).

The Dump command reveals the contents of a block of memory beginning at hex address XXXX and ending with hex address YYYY.

The output format is "xxx→→ HEXBYTES HEXBYTES" (ASCII characters).

The output from the Dump command consists of the hex address of the first byte on the display line (xxxx), the hex characters for the 8 bytes (HEXBYTES HEXBYTES), and their ASCII equivalents. Remember, two hex characters equal 1 byte. This means 8 bytes of information are represented by 16 characters.

If the memory block being dumped is greater than eight lines, use the pause key to slow the display.

When the dump is complete, the program returns to the main menu.

### Find a Character

The input format is F XXXX YYYY

z...z (enter).

Find seeks every occurrence of the specified hex character or group of characters (z...z) within a selected block of memory (from XXXX to YYYY).

This command is slow. One way to make it seem faster is to search through short blocks of memory. While the program is searching, the screen goes blank.

The output format is "z...z is at xx."

When found, the starting hex address (xx) of the string (z...z) is printed on the screen. The program then returns to the main menu. If the specified character or string is not located within the block of memory, the program just returns to the main menu.

### Convert to Hex or Decimal

The input format to Convert to Hex is C H DD (enter). The format to Convert to Decimal is C D XXXX (enter).

Convert to Hex changes a two-character decimal input (DD) into its equivalent hex notation. Using longer decimal numbers can yield incorrect results or function call errors.

Convert to Decimal changes a four-character hex number (XXXX) to the equivalent decimal number. This command requires a four-character input. If the number to be converted has less than four characters, use leading zeros to fill the extra spaces.

The output format to Convert to Hex is "dd Dec. = xxxHex." The output to Convert to Decimal is "xxx HEX = dd decimal." The program returns to the main menu after doing the conversion.

### End

The input format is E (enter).

The End command stops the program and returns to Basic. ■

## The Rule of 78

Conversion by Beve Woodbury  
80 Micro Technical Editor

Two ways to calculate interest on a loan are: add-on interest, and annual percentage rate (APR).

Assume you are interested in an add-on interest rate of 12 percent for a two-year loan of \$8,000 with monthly payments. The following equation calculates your monthly payments:

$$\text{Monthly payment} = \frac{\text{Interest} * \text{Years} * \text{Loan} + \text{Loan}}{12 * \text{Years}}$$

The calculations for the loan specified above are:

$$\text{Monthly payment} = \frac{12 * 2 * 8000 + 8000}{12 * 2}$$

$$\text{Monthly payment} = 413.33$$

Let's compute the value of the annual percentage rate that yields a payment of \$413.33 per month for 24 months on a loan of \$8,000. If payments are monthly, the interest per period is the APR divided by 12.

Under the standard amortization approach, the borrower pays interest for a period (a month) on the amount actually loaned during the period. The difference between the payment and the interest due reduces the remaining balance of the loan.

A number of approximation equations have been developed for the purpose of determining the APR associated with some add-on interest situations. No equation gives an exact answer except the one given below. The problem with this equation is that you must solve it by trial and error. There is no way to solve it directly.

$$\text{Monthly payment} = PV \frac{(i(1+i)^n)}{((1+i)^n - 1)}$$

The computer lends itself to solving the equation by trial and error.

After the APR is determined, it is possible to calculate a standard amortization table.

The Rule of 78 is an alternative method to determine the sum of money needed to pay off a loan. The Rule of 78 assumes (N/sum of the digits times the interest expense) is reduced the first period, ((N - 1)/sum of the digits times the

Line	Description
20	Converts a decimal character to hex
30	Converts a hex character to decimal
40-45	Decimal nibble (4 bits, 1 digit) to hex
50-60	Decimal address to hex
64-66	Pack a hex address (four characters)
72-110	Hex address to decimal
120	Packs 4 hex bytes
130-135	Pack eight ASCII or graphics characters
200-275	M command
300-345	D command
400-450	F command
500-535	C command
800-820	Main routine, command decoding

Table 7. Monitor 100 Line Descriptions



interest expense) is reduced the second period, and so on, where N is the number of periods over which the loan runs.

Listing 8 compares the results of the two approaches. It's a conversion of "The Rule of 300," *80 Micro*, January 1982, p. 116. In general, the amount needed to pay off a loan using the Rule of 78 is greater than that using the normal amortization approach. This is illustrated in the sample run (see Figs. 1 and 2).

### Running the Program

The example in the figures specified a loan of \$8,000 for two years, with add-on interest (ADD) at a rate of 12 percent. The monthly payment is \$413.33.

If you request hard copy, the following appears:

Loan: 8000  
 Add On Interest: 12 percent  
 Payment: 413.33  
 Annual Percentage Rate:  
 21.59999907016754 percent

The APR decimal is carried out this

far because a small difference in the APR produces big variations in the results. If you rerun the program selecting to input the APR, enter 21.60 and you will get a monthly payment of \$413.35.

Options are now given for the Rule of 78 or the regular amortization chart.

The key in this analysis is the difference in the amount needed to pay off a loan under the alternative approaches. In this example, the maximum difference occurs in the eighth month. Under the Rule of 78, \$5,742.96 is needed to eliminate the loan. Only \$5,704.62 is needed with the amortization approach.

From the finance company's point of view the rate of return earned on their money is greater if the loan is paid off early. The Rule of 78 is an alternative method of calculating the dollars necessary to pay off a loan. This approach is sometimes used for auto loans—be aware of the loan agreement details so you don't end up paying for your ignorance. ■

AS-K\$ Word formats for printing  
 XS Number format for printing  
 K-I Loop counters

#### Decisions:

LS ? ADD or APR  
 Y\$ ? hard copy  
 Z\$ ? which chart

#### Rule of 78 chart:

IT Interest  
 RXD Principal reduction  
 RB Balance of loan  
 DD Difference

#### Amortization chart:

IX Interest  
 RD Principal reduction  
 LX Balance of loan  
 DD Difference

#### Calculations:

AD Add-on interest rate  
 AP# Annual percentage rate  
 LO Loan amount  
 M Number of months of loan  
 PAY Monthly payment  
 S Divisor in interest calculation  
 XP Monthly payment calculated  
 YR Number of years of loan

Table 8. The Rule of 78 Variables

LOAN: 8000  
 ADD ON INTEREST: 12 %  
 PAYMENT: 413.33  
 ANNUAL PERCENTAGE RATE 21.59999907016754  
 REGULAR AMORTIZATION

MONTH	INT.	REDUC	BALANCE	DIFF
1	144.00	269.33	7730.67	-9.60
2	139.15	274.18	7456.49	-17.65
3	134.22	279.11	7177.38	-24.23
4	129.19	284.14	6893.24	-29.44
5	124.08	289.25	6603.99	-33.36
6	118.87	294.46	6309.53	-36.09
7	113.57	299.76	6009.77	-37.72
8	108.18	305.15	5704.62	-38.34
9	102.68	310.65	5393.97	-38.06
10	97.09	316.24	5077.73	-36.97
11	91.40	321.93	4755.80	-35.17
12	85.60	327.73	4428.08	-32.76
13	79.71	333.62	4094.45	-29.86
14	73.70	339.63	3754.82	-26.56
15	67.59	345.74	3409.08	-22.97
16	61.36	351.97	3057.11	-19.21
17	55.03	358.30	2698.81	-15.38
18	48.58	364.75	2334.06	-11.60
19	42.01	371.32	1962.74	-7.99
20	35.33	378.00	1584.74	-4.66
21	28.53	384.80	1199.94	-1.73
22	21.60	391.73	808.21	0.67
23	14.55	398.78	409.43	2.41
24	7.37	409.43	0.00	0.00

Fig. 1. Regular Amortization

LOAN: 8000  
 ADD ON INTEREST: 12 %  
 PAYMENT: 413.33  
 ANNUAL PERCENTAGE RATE 21.59999907016754  
 RULE 78

MONTH	INT.	REDUC	BALANCE	DIFF
1	153.60	259.73	7740.27	-9.60
2	147.20	266.13	7474.14	-17.65
3	140.80	272.53	7201.61	-24.23
4	134.40	278.93	6922.68	-29.44
5	128.00	285.33	6637.35	-33.36
6	121.60	291.73	6345.62	-36.09
7	115.20	298.13	6047.49	-37.72
8	108.80	304.53	5742.96	-38.34
9	102.40	310.93	5432.03	-38.06
10	96.00	317.33	5114.70	-36.97
11	89.60	323.73	4790.97	-35.17
12	83.20	330.13	4460.84	-32.76
13	76.80	336.53	4124.31	-29.86
14	70.40	342.93	3781.38	-26.56
15	64.00	349.33	3432.05	-22.97
16	57.60	355.73	3076.32	-19.21
17	51.20	362.13	2714.19	-15.38
18	44.80	368.53	2345.66	-11.60
19	38.40	374.93	1970.73	-7.99
20	32.00	381.33	1589.40	-4.66
21	25.60	387.73	1201.67	-1.73
22	19.20	394.13	807.54	0.67
23	12.80	400.53	407.01	2.41
24	6.40	406.93	0.00	0.00

Fig. 2. Rule of 78

#### Program Listing 8. The Rule of 78

```
10 XS="###"   ##.##   ####.##   #####.##   #
###.##
20 AS="LOAN: " :BS="ADD ON INTEREST: " :CS
="PAYMENT: " :DS="ANNUAL PERCENTAGE RATE"
:E$="RULE 78" :F$="REGULAR AMORTIZATION"
30 GS="MONTH " :HS=" INT. " :IS=" REDUC
BALANCE " :JS=" DIFF" :K$="PAYMENT: "
40 CLS:PRINT:PRINTTAB(15)E$:PRINT@136,"V
```

```
ERSUS":PRINT@210,"AMORTIZATION TABLES":F
ORK=0 TO1000:NEXT:GOTO 90
50 X=(AP*(1+AP)^(YR*12)):PAY=LO*X/((1+
AP)^(12*YR)-1):RETURN
60 X=INT(100*PAY)/100:Y=PAY-X:IFY>.005TH
ENX=X+.01
70 PAY=X:RETURN
80 CLS:PRINT"MONTHLY PAYMENT IS: ";PAY:PR
INT:RETURN
```

Listing 8 continues

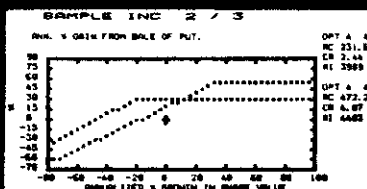


```

90 PRINT:INPUT"LOAN";LO:INPUT"NUMBER OF
YEARS: ";YR
100 PRINT:PRINT"MENU":PRINT  ADD  "
B$
110 PRINT  APR  ";D$
120 INPUTL$:IFL$="APR"THEN 180
130 IFL$<"ADD"THEN 100
140 INPUT"ADD ON INTEREST RATE: ";AD:AD=
AD*.01
150 PAY=(YR*AD*LO+LO)/(12*YR):GOSUB60:GO
SUB80:XP=PAY
160 FORI=1TO999:AP#=.0001*I+AD/12:GOSUB5
0:GOSUB60:IFPAY>XPTHEN210
170 NEXTI:PRINT"ERROR":END
180 INPUT"% RATE";AP#:AP#=AP*.01/12
190 GOSUB50:GOSUB60:XP=PAY
200 ADD=(12*YR*PAY-LO)/(LO*YR)
210 INPUT"HARD COPY Y/N: ";Y$:CLS:PRINTA
$;LO:PRINTB$;100*ADD;"%":IFY$="Y"THEN LP
RINTA$;LO:LPRINTB$;100*ADD;"%"
220 PRINTK$;XP:PRINTD$;1200*AP#;"%":IFY$
="Y"THENLPRINTK$;XP:LPRINTD$;1200*AP#:LP
RINT:LPRINT:LPRINT
    
```

```

230 PRINT:PRINT"ENTER Z FOR RULE 78 CHAR
T":INPUT  A FOR REGULAR AMORTIZATION
";Z$:PRINT
240 IFZ$="A"THEN270ELSEIFZ$<"Z"THEN230
250 PRINTTAB(10)E$:GOSUB350:IFY$="Y"THEN
LPRINTTAB(10)E$ELSEGOTO280
260 GOTO360
270 PRINTTAB(8)F$:GOSUB350:IFY$="Y"THENL
PRINTF$:GOTO360
280 S=0:M=12*YR:FORI=1TOM:S=S+I:NEXTI:RB
=LO:LX=LO
290 FORI=1TOM:IT=(M-I+1)/S*(YR*AD*LO):RX
D=XP-IT:RB=RB-RXD:IFI=MTHENIX=AP#*LX:RD=
LX:LX=0:DD=0:GOTO310
300 IX=AP#*LX:RD=XP-IX:LX=LX-RD:DD=LX-RB
310 IFZ$="A"THENPRINTUSINGX$;I,IX,RD,LX,
DD:IFY$="Y"THENLPRINTUSINGX$;I,IX,RD,LX,
DDELSEGOTO340
320 IFZ$="A"THEN340
330 PRINTUSINGX$;I,IT,RXD,RB,DD:IFY$="Y"
THENLPRINTUSINGX$;I,IT,RXD,RB,DD
340 NEXT:END
350 PRINT:PRINTG$;H$;I$;J$:RETURN
360 LPRINT:LPRINTG$;H$;I$;J$:GOTO280
    
```



**OPTIONS-80**

ANALYZES OPTION INVESTMENTS

HANDLES CALLS, PUTS, SPREADS, BUYING AND SELLING. INCLUDES COMMISSIONS, RISK, COST OF MONEY, DIVIDENDS. PRESENTS RESULTS IN TABULAR OR GRAPHICAL FORM. HARDCOPY AND STORAGE TO DISK. ACCOMPANYING MANUAL A COMPLETE AND UNIQUE GUIDE TO OPTION INVESTING. \$125. M/C AND VISA. MONEY BACK GUARANTEE. SEND FOR FREE BROCHURE.

FOR TRS-80, APPLE & IBM PC.

OPTIONS-80, BOX 471-E  
CONCORD, MASS 01742.

**MEMOREX FLEXIBLE DISCS**

WE WILL NOT BE UNDER-SOLD! Call Free (800)235-4137 for prices and information. Dealer inquiries invited and C.O.D.'s accepted.



VISA

PACIFIC EXCHANGES  
100 Foothill Blvd  
San Luis Obispo, CA  
93401 In Cal call  
(800)592-5935 or  
(805)543-1037



**"CC" RISER**

- Raises & tilts TV monitor for easy viewing
- Allows CC keyboard to slip under monitor for more desk space
- Matching silver-gray fiberglass
- \$37.50 + \$2.50 shipping

**"80" DEMI-CASES**

- Models for LNW or MDX I & II Boards
- \$32.50 + \$2.50 shipping
- N.Y.S. Residents Add 7% Tax

SYRACUSE R & D CENTER  
Box 125, Dewitt, N.Y. 13214  
"Specializing in Electronic Packaging"

**THE ULTIMATE IN SOFTWARE UNPROTECTION**

FPS-3 IS A FRONT PANEL SIMULATOR FOR THE TANDY CORP. TRS80 MODEL III

**JUST FLIP A SWITCH AND!!!**

THE PROGRAM IN MEMORY IS COPIED TO YOUR CHOICE OF DISK OR TAPE. TO RUN THE COPY SIMPLY BOOT THE DISK FROM RESET OR LOAD THE TAPE WITH THE SYSTEM COMMAND. YOU DO NOT NEED ANY TECHNICAL KNOWLEDGE TO USE THE FPS-3. ALL YOU NEED TO INSTALL THE FPS-3 ARE A HALF HOUR OF YOUR TIME AND A SCREWDRIVER. THE COST FOR A COMPLETE FPS-3 IS ONLY \$50.

**WHAT THE SOFTWARE GODS HAVE HIDDEN THIS LITTLE SHALL REVEAL**

J.E.S. GRAPHICS, P.O. BOX 2752,  
TULSA, OK. 74101 CALL 918 742 7104.

**DISKETTES**

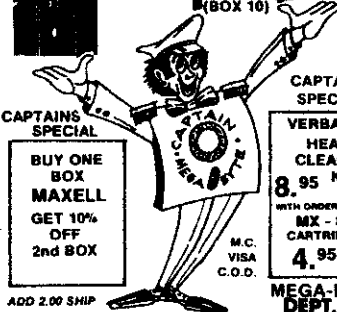
maxell. VERBATIM

100% ERROR FREE

5 1/4 SS/DD 28.95 (BOX 10)

DATALIFE

5 1/4 SS/DD 24.95 (BOX 10)



CAPTAINS SPECIAL

BUY ONE BOX MAXELL GET 10% OFF 2nd BOX

VERBATIM HEAD CLEANING 8.95 KIT

WITH ORDER OF DISKES MX-80 CARTRIDGE 4.95 EA.

M.C. VISA C.O.D.

MEGA-BYTE DEPT. 80  
4699 S.W. 45th STREET  
FORT LAUD., FLA 33314  
305-587-1130



**MAKE YOUR OWN SIGNS IN MINUTES**

Reduction of an actual sign

**The Banner Machine**

- For the TRS-80 I & III with 32K tape or 48K disk
- For use on the Epson MX-80 with Graftrax
- Uses dot graphics instead of TRS-80 block graphics
- Menu-driven program
- Operation similar to a word processor
- Makes signs up to 10" tall by any length
- 10 sizes of letters from 1/4"-8" high
- Mono or proportional spacing
- Automatic centering; Right and left justifying
- Makes borders of variable width up to 1/4"

Order The Banner Machine - \$49.95 from

Virginia Micro Systems  
13646 Jeff Davis Highway  
Woodbridge, Virginia 22191

Phone (703) 491-6502

C-NOTE FOR AUGUST 1983

# Five Portable Programs

## Foxfighter

by Brad Dixon

80 Micro Technical Editor

The Model 100 is fast proving itself a powerful computer with great potential in business and home environments. However, there are times when the drudgery of computing must be replaced with some fast-paced entertainment. Foxfighter (Program Listing 1) is a Model 100 game that uses a meager 3K of memory, yet provides plenty of action.

At the game's start you're given the option of receiving flight instructions or moving directly to your mission. The instructions are provided in lines 390-440. As pilot of a Foxfighter aircraft, your mission is to clear the sky of deadly airmines over an enemy city. To operate the Foxfighter, use the O key to move your aircraft up, the P key to move down, and the space bar to fire the fighter's laser cannon. An INKEY function in line 80 scans the keyboard for these flight instructions. Upward flight is performed in line 90 while downward flight and laser fire are controlled in lines 110 and 190, respectively.

Foxfighter's position is guided by the PRINT@ values produced by the loop in line 70. Once in flight, the fighter moves to the beginning of the next line on the screen when it reaches the end of the previous line. When the fighter reaches number 198 in the loop, it is recycled back to the top left corner of the screen.

If you have little interest in guiding the fighter, but would rather obtain a high score, this feature pilots the craft while you sit back and fire at the appropriate time.

This feature produces an interesting characteristic during laser fire called "wrap around, zap around." If you fire the laser cannon within eight spaces of the end of a row, part of the laser fire wraps around to the beginning of the next row. In effect, you are able to shoot around corners.

Another feature makes the game even more challenging. If an airmine is brushed by the nose or tail of your Foxfighter, the airmine becomes invisible on the screen. Don't let this fool

### The Key Box

Foxfighter and Intel require 4K RAM, The Year in Review needs 8K RAM, and Subterra and Renumber 100 take 24K RAM.

Program Listing 1. Foxfighter.

```

10 DIM X(6):V=3:SCR=0:CLS:PRINT@92,"<<
FOXFIGHTER >>":PRINT:PRINT:INPUT"DO YOU
NEED INSTRUCTIONS (Y OR N)";A$:IF A$="Y
"THEN390 ELSE20
20 DATA 45,79,115,95,102,124,19,83,58,23
,150,128,51,73,140,113,102,31,16,118,88,
67,81,75,100,12,68,95,64,28,116,147,159,
48,95,114,149,140,119,57,107,30
30 FORM=1TO6:READX(M):IFX(M)=30THENRESTO
RE20ELSENEXT
40 P=0:CLS
50 GOSUB220
60 Z$=CHR$(237)+CHR$(232)+CHR$(254)
70 FORB=7TO198:FORM=1TO6:IFB<198ANDB<>X(
M)THENNEXTELSEIFB=198THEN180ELSEIFB=X(M)
THEN380
80 C$=INKEY$:IFC$=""THEN120ELSEIFC$=CHR$
(32)THEN190ELSEIFC$="O"THEN90ELSEIFC$="P
"THEN110ELSE120
90 PRINT@(B-1)," " :B=B-39:IFB<0THENB=B
+40ELSE120
100 GOTOL20
110 PRINT@(B-1)," " :B=B+41:IFB>198THEN
B=B-40
120 PRINT@B,Z$
130 SOUND12538,1
140 IFB=0THEN170ELSEC=B-1
150 IFSCR=500ORSR=1000ORSR=2000THENPRI
NT@0,V+1
160 PRINT@C," "
170 NEXTB
180 PRINT@(B-1)," " :GOTO70
190 PRINT@(B-1),Z$:PRINT@(B+2),".....
":FORT=1TO8:SOUND1479,1:NEXT:PRINT@(B+2)
,
200 FORM=1TO6:IFX(M)>(B+2)ANDX(M)<(B+10)
THEN370ELSENEXT
210 GOTOL40
220 PRINT@200,CHR$(237):PRINT@203,CHR$(2
39):PRINT@208,CHR$(234)+CHR$(233)+CHR$(2
28)+CHR$(233)+CHR$(228)
230 PRINT@216,CHR$(234):PRINT@225,CHR$(2
32)+CHR$(233)+CHR$(238)+CHR$(238)+CHR$(2
39)+CHR$(239)
240 PRINT@233,CHR$(233)+CHR$(238):PRINT@
238,CHR$(237)
250 PRINT@240,CHR$(239):PRINT@242,CHR$(2
27)+CHR$(239)+CHR$(227):PRINT@248,CHR$(2
39)+CHR$(239)+CHR$(239)
260 PRINT@251,CHR$(237)+CHR$(238):PRINT@

```

Listing 1 continues

you. The airmine is still there and will destroy you if you crash into it. Airmine also disappear if the sensor field around each is disturbed by laser fire aimed at nearby airmine.

Crashes and airmine hits provide similar explosions with flashing asterisks and sounds. In each case, the scoreboard is updated in the upper left corner of the screen. When you exhaust all your fighters, the screen clears and your final score is posted. You can play again or exit the program. Each airmine is worth 10 points and additional fighters are awarded at 500, 1000, and 2000 points.

### Enhancements

The cityscape in the program is more for visual effect than for increasing the game's difficulty. Crashes into buildings are

not difficult to add, but in Basic the checks for crashes might inhibit fast game play. Moving targets or anti-aircraft missiles from the surface would also add to the excitement of the game, but, again, the limitations of Basic programming make such refinements impractical.

The available RAM in the Model 100 limits the level of game modification. One nice addition that doesn't sacrifice speed is incorporating a time limit on the game so that a player has to shoot all the airmine within a specified time. If this is not accomplished, points are deducted from the player's score.

I think you'll find Foxfighter a fast and challenging game that's a nice alternative to the Model 100's more serious applications. ■

*Listing 1 continued*

```

254,CHR$(238):PRINT@256,CHR$(239):PRINT@
260,CHR$(238)+CHR$(239)
270 PRINT@262,CHR$(237)+CHR$(232):PRINT@
265,CHR$(239)+CHR$(239)+CHR$(239)+CHR$(2
39)+CHR$(239)+CHR$(239)
280 PRINT@271,CHR$(232)+CHR$(238)+CHR$(2
39)+CHR$(239)+CHR$(232)+CHR$(232)+CHR$(2
32)+CHR$(239)
290 PRINT@280,CHR$(239)+CHR$(239)+CHR$(2
39)+CHR$(239)+CHR$(239)+CHR$(239)+CHR$(2
39)+CHR$(239)+CHR$(239);:
300 PRINT@288,CHR$(239)+CHR$(239)+CHR$(2
39)+CHR$(239)+CHR$(239)+CHR$(239)+CHR$(2
39)+CHR$(239)+CHR$(239);:
310 PRINT@297,CHR$(239)+CHR$(239)+CHR$(2
39)+CHR$(239)+CHR$(239)+CHR$(239)+CHR$(2
39)+CHR$(239)+CHR$(239);
320 PRINT@306,CHR$(239)+CHR$(239)+CHR$(2
39)+CHR$(239)+CHR$(239)+CHR$(239)+CHR$(2
39)+CHR$(239)+CHR$(239);
330 PRINT@315,CHR$(239)+CHR$(239)+CHR$(2
39)+CHR$(239);
340 FORM=1TO6:PRINT@X(M),CHR$(169):NEXT
350 PRINT@0,V":PRINT@3,SCR
360 RETURN
370 FORT=1TO5:PRINT@X(M),****:SOUND9394
,2:NEXTT:PRINT@X(M),":SCR=SCR+10:PRI
NT@3,SCR:P=P+1:IFP=6THEN30ELSEX(M)=290:G
OTO140
380 FORT=1TO5:PRINT@B-1,****?SOUND995
2,2:NEXTT:V=V-1:PRINT@0,V:IFV=0THEN450EL
SE30
390 CLS:PRINT@52,"FOXFIGHTER":PRINT"YOUR
MISSION IS TO DESTROY THE AIRMINE":PRINT
"FIELDS ABOVE THE ENEMY CAPITOL."
400 PRINT"YOUR FIGHTERS ARE GUIDED USING
THE 'O' KEY TO GO UP AND THE 'P' KEY TO
GO DOWN.THE SPACEBAR FIRES THE FIGHTER'S
LASER CANNON."
410 FORT=1TO4000:NEXTT
420 CLS:PRINT@52,*** WARNING ***:PRINT
:PRINT"IF LASER FIRE OR PART OF YOUR FIG
HTER":PRINT"BRUSHES THE SENSOR FIELD ARO
UND"
430 PRINT"EACH AIRMINE, IT WILL BECOME I
NVISIBLE YET REMAINS ARMED. GOOD LUCK!"

440 FORT=1TO4000:NEXT:GOTO20
450 CLS:PRINT@82,"YOU HAVE RUN OUT OF FI
GHTERS!!":PRINT@122,"YOUR FINAL SCORE IS
";SCR:PRINT:V=3:SCR=0:INPUT"PLAY AGAIN (
Y OR N)";A$:IF A$<>"Y"THENENDELSE20

```

Variable	Description
V	Number of Foxfighters
SCR	Score
B	Foxfighter screen position
P	Score counter
Z\$	Foxfighter graphics

*Table 1. Foxfighter variables list.*

Line	Description
10	Initialization and title page
20-30	Data for airmine placements
60-70	Foxfighter setup and screen definitions
80-130	INKEY routine to check steering and laser fire
150	Check score for extra fighter
190-200	Firing routine and check for airmine hits
220-350	City graphics and scoreboard
370	Airmine hit routine and scoreboard update
380	Crash routine and fighter count update
390-440	Instructions for play
450	Final score and replay option

*Table 2. Foxfighter line descriptions.*

## Intel

### Conversion by Beve Woodbury 80 Micro Technical Editor

"4K Intelligence," a game by William M. Lopez, (*80 Micro*, March 1980, p. 55) was designed so that a computer would "learn" from its mistakes; when it is defeated, it doesn't repeat its last move again. Thus the first few games are easy to win, but as play continues, the computer becomes unbeatable.

"4K Intelligence" is played with chess pawns on a 3- by 3-block matrix. Pawns can move one space forward to an empty space, or diagonally to capture an opponent. That player whose pawn reaches the opposite side first, or who blocks his opponent from moving, wins.

When you start the game, you are given the opportunity to see the game instructions. Type and enter a Y for instructions,

or an N to begin the game. Before the game starts, there is a brief pause while arrays are set up.

You always get the first move. Move a pawn forward one space to a blank space, or diagonally to an occupied space to capture the opponent's pawn. When the screen prompts FROM, choose the pawn to be moved, type the pawn location and press the enter key. When the screen prompts TO, type the desired destination and press enter. If you make an invalid move (move forward to occupied space, move diagonally to unoccupied space, skip a space), the computer is unforgiving. You lose the game!

### The Program

Program Listing 2 sets up four arrays. Array M contains values for each board configuration. When a move results in a loss, that move is changed to zero, so that the next time the configuration comes up, the computer uses the next possible move. A 15 appears on the display if the computer cannot make a move. If the computer loses in that way, it changes its previous move to zero. If all three possible moves become zero, column 2 changes to 15 so that next time around the previous move is changed.

Array B contains all possible computer block configurations—those with which the computer has won by blocking all your possible moves.

Array C contains the possible computer moves using the matrix locations. Column 1 has the FROM matrix location and column 2 has the TO matrix location.

Array G contains a summary of the game matrix. The screen location for the pawn is in column 0, the matrix location of the computer pawn in column 1, and your pawn in column 3. A 1 indicates a pawn in that location and zero indicates an empty space. Column 2 contains specific values for the corresponding matrix location for computer pawns, and column 4 has the values for your pawns.

The program occupies almost 4K and contains several REM statements for program clarity. If you have limited memory space, remove all remark lines. ■

#### Arrays

- B() Computer block configurations
- C() Possible computer moves
- G() Game matrix summary
- M() Board configurations and possible moves

#### General

- A Row location of computer move
- BL Human blocked if 1
- C Computer configuration value
- F Matrix location to move computer pawn from
- H Human configuration value
- I For . . . Next loop counter
- J Matrix location to move human pawn from
- K Matrix location to move human pawn to
- L Number of human losses
- LC Column number of previous computer move
- LR Row number of previous computer move
- M Computer move
- NS ? Play again?
- T Matrix location to move computer pawn to
- W Number of human wins
- X Value to check for invalid move
- Y\$ ? Instructions wanted?

Table 3. "4K Intelligence" variables list.

```

10 DIMG(8,4),M(38,4),C(15,1),B(14,1)
20 CLS:PRINT@12,"INTEL":PRINT:W=0:L=0
30 INPUT"DO YOU WANT INSTRUCTIONS? Y/
N":Y$
40 IFY$="Y"THEN1170ELSEIFY$="N"THEN70
50 GOTO30
60
70 DATA4,3,7,5,4,1,2,1,0,6,8,2,11,7,3
80 DATA7,6,2,5,4,3,11,10,5,6,5,0,9,8,0
90 DATA3,2,0,5,4,3,7,6,0,7,6,0,7,0,0
100 DATA11,8,0,2,0,0,8,5,0,3,14,0,8,11,0
110 DATA15,0,0,15,0,0,11,14,0,8,7,6
120 DATA3,11,0,5,11,0,2,8,0,6,14,0,2,10,0
130 DATA1,2,6,15,0,0,15,0,0,6,0,0,6,7,0
140 DATA7,0,0,1,0,0,1,2,14,14,0,0,11,5,0
150 FORI=0TO38:READM(I,2),M(I,3),M(I,4)
:NEXT
160 'set up computer model table
170 DATA392,7,224,7,336,7,272,13,264,22
180 DATA152,5,296,3,96,22,112,14,160,13
190 DATA280,35,104,6,144,6,80,6,136,5
200 DATA32,28,56,1,48,10,24,34,32,25
210 DATA40,5,136,17,8,52,16,12,8,18
220 DATA16,10,8,5,16,36,24,1,176,5
230 DATA16,2,168,21,56,4,272,6,264,5
240 DATA96,5,80,37,136,37,32,18
250 FORI=0TO38:READM(I,0),M(I,1):NEXT
260 'blocked human move table
270 DATA1,8,2,16,4,32,8,64,16,128
280 DATA12,96,33,264,32,256,5,40
290 DATA40,320,21,168,17,136,20,160
300 DATA34,272,9,80
310 FORI=0TO14:READB(I,0),B(I,1):NEXT
320 'possible computer moves
330 DATA0,3,0,4,1,3,1,4,1,5,2,4,2,5
340 DATA3,6,3,7,4,6,4,7,4,8,5,7,5,8
350 FORI=1TO14:READC(I,0),C(I,1):NEXT
360 ' set up graphic array
370 DATA67,1,1,0,0,72,1,2,0,0
380 DATA77,1,4,0,0,147,0,8,0,8
390 DATA152,0,16,0,16,157,0,32,0,32
400 DATA227,0,0,1,64,232,0,0,1,128
410 DATA237,0,0,1,256
420 FORI=0TO8:READG(I,0),G(I,1),G(I,2),G
(I,3),G(I,4):NEXT
430 CLS ' setup board graphic
440 LINE(146,4)-(234,50),1,B:LINE(145,3)-
(235,51),1,B
450 LINE(174,5)-(174,50):LINE(205,5)-
(205,50)
460 LINE(147,20)-(233,20):LINE(147,35)-
(233,35)
470 PRINT@65,"0":PRINT@67,CHR$(128)
480 PRINT@70,"1":PRINT@72,CHR$(128)
490 PRINT@75,"2":PRINT@77,CHR$(128)
500 PRINT@145,"3":PRINT@150,"4"
:PRINT@155,"5"
510 PRINT@225,"6":PRINT@227,CHR$(148)
520 PRINT@230,"7":PRINT@232,CHR$(148)
530 PRINT@235,"8":PRINT@237,CHR$(148)
540 ' get human move
550 GOSUB1300
560 GOSUB750
570 ' get model table value.
580 IFH=0THEN970
590 FORI=0TO14
600 IFC=B(I,0)AND H=B(I,1)THENBL=1
610 NEXT
620 IFBL=1THENPRINT@200,"BLOCKED! I
WIN!":BL=0:GOTO990
630 PRINT@200,"YOUR MOVE HUMAN!"

```

Listing 2 continues

```

640 INPUT"FROM";J:PRINT@250,"TO";:INPUTK
650 X=J-K
660 IFX<2ORX>4THEN1090
670 IFX=3ANDG(K,1)=1THEN1090
680 IFX=4ANDG(K,1)=0THEN1090
690 IFX=2ANDG(K,1)=0THEN1090
700 G(K,3)=1:G(K,1)=0:G(J,3)=0
710 PRINT@G(K,0),CHR$(148):PRINT@G(J,0),"
"
720 IFK<3THEN1120
730 GOSUB740:GOTO810
740 'get array values for comp. move
750 C=0:H=0
760 FORI=0TO8
770 IFG(I,1)=1THENC=C+G(I,2)
780 IFG(I,3)=1THENH=H+G(I,4)
790 NEXT
800 RETURN
810 ' get computer move #
820 FORI=0TO38
830 IFH=M(I,0)ANDC=M(I,1)THENA=I:GOTO860
840 NEXT
850 ' get and make computer move
860 FORI=2TO4
870 IFM(A,I)>0THENM=M(A,I):GOTO900
880 NEXT
890 IFM=0THENM(A,2)=15
900 IFM=15THEN1140
910 F=C(M,0):T=C(M,1)
920 G(F,1)=0:G(T,3)=0:G(T,1)=1
930 PRINT@G(F,0)," ":PRINT@G(T,0),CHR$(
128)
940 IFT>5THEN980
950 LR=A:LC=I
960 GOTO550
970 ' computer win routine
980 GOSUB1300:PRINT@200,"I WIN, HUMAN!"
990 ' ending routine
1000 L=L+1
1010 PRINT"YOU'VE WON";W; "AND LOST";L

```

```

1020 INPUT"DO YOU WANT TO PLAY AGAIN? (Y/
N)";N$
1030 FORI=0TO2:G(I,1)=1:G(I,3)=0:NEXTI
1040 FORI=3TO5:G(I,1)=0:G(I,3)=0:NEXTI
1050 FORI=6TO8:G(I,1)=0:G(I,3)=1:NEXTI
1060 M=0:IFN$="Y"THEN430
1070 END
1080 ' invalid move routine
1090 GOSUB1300:PRINT@200,"INVALID MOVE!
I WIN!"
1100 GOTO990
1110 ' human win routine
1120 PRINT@200,"YOU'VE WON, HUMAN!"
1130 W=W+1:M(A,B)=0:GOTO1010
1140 PRINT@200,"YOU'VE WON, HUMAN!"
1150 W=W+1:M(LR,LC)=0:GOTO1010
1160 ' game instructions
1170 CLS:PRINT:PRINT"INTEL IS PLAYED ON A
3 BY 3 MATRIX."
1180 PRINT"THE PAWNS ARE MOVED ONE SPACE
FORWARD"
1190 PRINT"OR ONE SPACE DIAGONALLY TO
CAPTURE."
1200 PRINT"MY PAWNS ARE ";CHR$(128);"
YOURS ARE ";CHR$(148)
1210 INPUT Y$:CLS:PRINT
1220 PRINT"WIN OCCURS WHEN YOU REACH THE
OPPOSITE"
1230 PRINT"SIDE OF THE BOARD OR"
1240 PRINT"WHEN ALL OPPONENTS PAWNS ARE
BLOCKED."
1250 PRINT"AN INVALID MOVE RESULTS IN A
LOSS."
1260 PRINT"TO MAKE A MOVE, ENTER PRESENT
PAWN"
1270 PRINT"POSITION, THEN ENTER NEW PAWN
POSITION."
1280 INPUT Y$:GOTO70
1290 ' blank screen text
1300 FORI=0TO16:PRINT@(200+I)," ":PRINT@(
240+I)," ":NEXT:RETURN

```

## Subterra

Conversion by Mare-Anne Jarvela  
80 Micro Technical Editor

Subterra, an adventure game by Richard Rämella, first appeared in our Fun House column (80 Micro, November 1982, p. 488). It requires a 24K Model 100.

Variable	Description
A\$	Data from data statements
B	Array counter
I	For...Next counter
X\$	Input variable
C\$	Input variable
K\$	Input variable
I	For...Next counter
Z	Random number
M	Random number
K	Found key flag
H	Going in or out flag
P	Mile counter

Table 4. Subterra variables list.

### Program Listing 3. Subterra.

```

10 REM SUBTERRA
20 CLS
30 DATA NORTH,SOUTH,EAST,WEST,KEY,NOTE-S
OMETIMES EMPTY
40 DATA DESERT,HOME,LOCATION,DO YOU WALK
INTO WALLS A LOT?
50 DATA WALL,CAVE ENTRANCE,TUNNEL OF MIC
A,CRAWL SPACE,VESTIBULE OF EVIL
60 DATA TRIANGLE TUNNEL,ENDLESS PASSAGE,
GROTTO OF GRIEF,DRAGON,WELL
70 DATA FOUR CORNERS,TROLL WAY,ROCK TUNN
EL,THREE CORNERS,ECHO CAVERN
80 DATA RIVER,THREE DOORWAYS,SULPHUR LAN
E,WATERY ELBOW,DARKLING WAY
90 DATA COBWEBBED HALLWAY,CIRCLE CHAMBER
,WHITE WATER,RIVER ROCK,ABYSMAL WATERFAL
L
100 DATA SACRIFICIAL ALTAR,SHORT HALL,SN
AKE CITY,POISON HALL,RAT CHAMBER
110 DATA IT'S ENDLESS SO GO BACK !!
120 DATA I'M SORRY.. YOU WILL NEVER RETU
RN.
130 DATA HER FIERY BREATH DRIVES YOU BAC
K,A DOORWAY,RIVER BANK
140 DATA LIMBO JAUNT,CAVE-IN NOW BLOCKS
ENTRANCE,A ROUND PORTAL

```

Listing 3 continues



In Subterra, you must traverse underground caves and passages in search of a precious idol. Travel in any direction, but tread cautiously—many hidden traps await you. And don't assume you're through when you find the idol—you are in danger until you and the idol are safely above ground.

Pictures add to the fun, and show off some of the Model 100's graphics. I used CHR\$(27);“p” to make the background dark and CHR\$(27);“q” to make it light. Lines 2230-2730 contain all the graphics. (See Program Listing 3.)

Subterra produces sound in conjunction with the graphics.

Change line 2310 to compose a different tune for the key, line 2390 for the doors, line 2500 for the dragon, line 2620 for the well, and line 2700 for the idol.

The Model 100's random function (RND) gives you numbers between zero and one. I used INT(10\*RND(1)) for an integer between zero and nine. RND always gives the same order of random numbers; to avoid that, I used the clock to pick up a different sequence each time (see p. 175 of the Model 100 manual).

Enjoy Subterra—but don't get lost. ■

Listing 3 continued

```

150 CLEAR 500
160 DIM A$(48)
170 FOR B=1 TO 48
180 READ A$(B)
190 NEXT B
200 PRINT@50,"HAVE FUN FINDING THE IDOL!
!!"
210 PRINT@90,"HOLD YOUR EYES OPEN FOR...
..":FOR T=1 TO 1200:NEXT T:CLS
220 GOSUB 2350:CLS:GOSUB 2240:CLS:GO
SUB 2420:CLS:GOSUB 2530:CLS:GOSUB 26
40:CLS
230 REM CAVE
240 L=12:N=7:W=7:S=8:E=13
250 GOSUB 2000
260 IF X$="N" OR X$="W" THEN PRINT "LOST
...NEVER HEARD OF AGAIN.":END
270 IF X$="S" THEN PRINT "WENT HOME AND
DIDN'T EVEN TRY.":END
280 IF X$="E" GOTO 300
290 REM TUNNEL OF MICA
300 L=13:N=11:W=13:S=14:E=13
310 GOSUB 2000
320 IF X$="N" THEN PRINT A$(10):GOTO300
330 IF X$="W" THEN PRINT A$(47):GOTO300
340 IF X$="S" GOTO 370
350 IF X$="E" GOTO 480
360 REM VESTIBULE OF EVIL
370 L=15:N=14:W=11:S=11:E=16
380 SEC = VAL(RIGHT$(TIMES$,2))
390 FOR I=1 TO SEC
400 Z=INT(10*RND(1))
410 NEXT I
420 IF Z<6 THEN GOSUB 2100
430 GOSUB 2000
440 IF X$="W" OR X$="S" THEN PRINT A$(10
):GOTO 370
450 IF X$="N" GOTO 300
460 IF X$="E" GOTO 600
470 REM GROTTO OF GRIEF
480 L=18:W=13:S=22:E=20
490 IF H=2 THEN N=14 ELSE N=19
500 GOSUB 2000
510 IF X$="W"THEN GOTO 300
520 IF X$="E" THEN GOTO 770
530 IF X$="S" GOTO 600
540 IF X$="N" AND H<>2 THEN GOSUB 2420:C
LS
550 IF X$="N" AND H<>2 THEN SEC=VAL(RIGH
T$(TIMES$,2)):FOR I=1 TO SEC:Z=INT(10*RND
(1)):NEXT I
560 IF X$="N" AND H=2 THEN PRINT "YOU EX
IT TO SUNLIGHT. THE IDOL IS YOURS":END
570 IF Z<6 THEN PRINT A$(42):END
580 IF Z>5 THEN PRINT A$(43):Z=0:GOTO
480

```

```

590 REM FOUR CORNERS
600 L=21:N=22:W=16:S=17:E=23
610 GOSUB 2000
620 IF X$="N" GOTO 480
630 IF X$="W" GOTO 370
640 IF X$="E" GOTO 1170
650 IF X$="S" GOTO 660
660 P=2
670 REM -ENDLESS PASSAGE
680 L=17:W=11:S=17:E=11
690 GOSUB 2000
700 IF P=2 AND X$="N" THEN PRINT "YOU LE
AVE ";A$(17):P=0:GOTO600
710 IF X$="S" THEN P=P+2
720 IF X$="N" THEN P=P-2
730 IF P=10 THEN GOSUB 2100
740 PRINT "DISTANCE INTO ";A$(17);":";P;
" MILES"
750 GOTO 680
760 REM WELL
770 GOSUB2530:CLS
780 L=20:N=11:W=11:S=11:E=11
790 GOSUB 2000
800 IF X$="N" OR X$="S"THEN PRINT"YOU CA
N'T GET OUT!!!":END
810 IF X$="W"THEN PRINT"THAT HURT!":FOR
T=1 TO 500:NEXT T:GOTO 770
820 IF X$="E"THEN PRINT"A WALL AGAIN ?":
FOR T=1 TO 500:NEXT T:GOTO 770
830 N=11:W=11:S=25:E=11
840 IF Z=1 THEN L=38
850 IF Z=2 THEN L=39
860 IF Z=3 THEN L=40:E=37
870 SEC = VAL(RIGHT$(TIMES$,2))
880 FOR I=1 TO SEC
890 M=INT(10*RND(1))
900 NEXT I
910 IF M<6 THEN PRINT"FOUND-CHEST WITH A
";
920 IF M>5 GOTO 1000
930 SEC = VAL(RIGHT$(TIMES$,2))
940 FOR I=1 TO SEC
950 M=INT(10*RND(2))
960 NEXT I
970 IF M<6 THEN PRINT A$(5):FOR T=1 TO 5
00:NEXT T:GOSUB 2230:CLS
980 IF M>5 THEN PRINT A$(6)
990 IF M<6 THEN K=1
1000 GOSUB 2000
1010 IF X$="N" OR X$="W" GOTO 1000
1020 IF X$="E" AND Z=3 GOTO 1450
1030 IF X$="E" GOTO 1000
1040 IF X$="S" GOTO 1060
1050 REM ECHO CAVERN
1060 L=25:N=27:W=20:S=28:E=44
1070 GOSUB 2000
1080 IF X$="W" GOTO 770

```

Listing 3 continues

```

1090 IF X$="S" GOTO 1170
1100 IF X$="E" AND K<>1 THEN PRINT "NO "
;A$(5): GOTO 1060
1110 IF X$="E" AND K=1 THEN PRINT A$(5)
;" DOESN'T WORK.":GOTO 1060
1120 GOSUB2350
1130 PRINT@42, "WHICH DOOR - 1,2 OR 3";:
INPUT Z:CLS
1140 IF Z<>1 AND Z<>2 AND Z<>3 GOTO 1130
1150 GOTO 830
1160 REM RIVER BANK
1170 L=45: N=28: W=23: S=11: E=26
1180 GOSUB 2000
1190 IF X$="N" GOTO 1060
1200 IF X$="W" GOTO 600
1210 IF X$="S" THEN PRINT "HURT YOUR HEA
D?":GOTO 1170
1220 REM WATERY ELBOW
1230 L=29: N=26: W=26: S=46: E=11
1240 GOSUB 2000
1250 IF X$="N" GOTO 1390
1260 IF X$="W" GOTO 1170
1270 IF X$="E" THEN PRINT "OOPS,ANOTHER
WALL EH?": GOTO 1230
1280 REM LIMBO JAUNT
1290 L=46: N=29: W=11: S=46: E=11
1300 PRINT "DISTANCE IN:";P;"MILES"
1310 GOSUB 2000
1320 IF X$="S" THEN P=P+2
1330 IF X$="N" THEN P=P-2
1340 IF P<2 THEN PRINT "YOU'RE OUT":GOTO
1230
1350 IF P=10 THEN PRINT A$(41)
1360 IF P=12 THEN PRINT "TOO LATE . LOST
IN THE DARK. THAT'S ALL": END
1370 GOTO 1290
1380 REM RIVER ROCK
1390 L=34: N=33: W=44: S=26: E=31
1400 GOSUB2000
1410 IF X$="W" GOTO 1060
1420 IF X$="S" GOTO 1230
1430 IF X$="E" GOTO 1610
1440 REM SACRIFICIAL ALTAR
1450 L=36: N=35: W=40: S=33: E=11
1460 GOSUB 2000
1470 IF X$="W" THEN Z=3: GOTO 830
1480 IF X$="S" GOTO 1390
1490 IF X$="E" THEN PRINT"ANOTHER BRUISE
": GOTO 1450
1500 IF X$="N" THEN PRINT "YOU KNOW WHAT
AN ";A$(35);" IS AND STILL WANT TO GO?"
1510 INPUT "ANSWER YES OR NO";X$
1520 IF X$<>"YES" AND X$<>"NO" GOTO 1510
1530 IF X$="NO" GOTO 1450
1540 PRINT
1550 PRINT "NOW YOU FALL FOREVER ";:GOTO
1590
1560 PRINT "AND EVER ";
1570 FOR T=1 TO 200
1580 NEXT T
1590 GOTO 1560
1600 REM COBWEBBED HALLWAY
1610 L=31: N=11: W=34: S=11: E=48
1620 PRINT "YOU STAND AT ";A$(48)
1630 IF K<>1 THEN PRINT "NO ";A$(5);" TO
ENTER": GOTO 1390
1640 IF K=1 THEN PRINT "YOUR ";A$(5)" WO
RKS!"
1650 GOSUB 2000
1660 PRINT "YOU NOW ENTER THE ";A$(32)
1670 FOR T=1 TO 500
1680 NEXT T
1690 PRINT "DO YOU RECALL THE MAGIC DIRE

```

```

CTION?"
1700 PRINT "IF YOU DON'T KNOW, GUESS QUI
CKLY!"
1710 PRINT"THE CEILING WILL START TO LOW
ER..."
1720 FOR T=1 TO 1000
1730 NEXT T
1740 PRINT "PRESS ANY KEY IF YOU KNOW."
1750 PRINT
1760 FOR T=1 TO 500
1770 NEXT T
1780 FOR M=10 TO 1 STEP -1
1790 CLS
1800 PRINT M;"SECONDS TO GO"
1810 C$=INKEY$
1820 IF C$<>"GOTO 1860
1830 FOR T=1 TO 200
1840 NEXT T
1850 NEXT M
1860 PRINT "THE MAGIC DIRECTION IS "
1870 INPUT"(NORTH-EAST-SOUTH-WEST)";C$
1880 IF C$=K$ GOTO 1900
1890 PRINT "BAD GUESSING, A CRUSHING DEF
EAT, EH?": END
1900 CLS:GOSUB 2640:CLS
1910 PRINT "YOU FOUND THE GOLDEN IDOL IN
TIME."
1920 PRINT "YOUR WORRIES ARE OVER..."
1930 PRINT "EXCEPT YOU HAVE TO FIND YOUR
WAY OUT."
1940 PRINT"YOU EXIT WITH TREASURE IN HAN
D."
1950 H=2
1960 FOR T=1 TO 1500
1970 NEXT T
1980 CLS
1990 GOTO 1390
2000 PRINT A$(9);": ";A$(L)
2010 PRINT A$(1);": ";A$(N)
2020 PRINT A$(4);": ";A$(W)
2030 PRINT A$(2);": ";A$(S)
2040 PRINT A$(3);": ";A$(E)
2050 PRINT"DIRECTION - (N-W-S-E)";
2060 INPUT X$
2070 IF X$<>"N" AND X$<>"W" AND X$<>"S"
AND X$<>"E" GOTO 2050
2080 CLS
2090 RETURN
2100 IF K$<>" THEN RETURN
2110 SEC = VAL(RIGHT$(TIME$,2))
2120 FOR I=1 TO SEC
2130 Z=INT(10*RND(1))
2140 NEXT I
2150 IF Z<3 AND Z>0 THEN K$=A$(1):GOTO 2
190
2160 IF Z<5 THEN K$=A$(2):GOTO 2190
2170 IF Z<8 THEN K$=A$(3):GOTO 2190
2180 IF Z<10 THEN K$=A$(4)
2190 Z=0
2200 PRINT "A WALL OPENS. A MESSAGE SCRO
LL APPEARS:"
2210 PRINT "REMEMBER THE MAGIC DIRECTION
: ";K$;" "
2220 RETURN
2230 REM KEY
2240 PRINT@59,CHR$(230);CHR$(231);CHR$(2
31);CHR$(229)
2250 PRINT@99,CHR$(229);CHR$(232);CHR$(2
32);CHR$(230)
2260 PRINT@141,CHR$(233)
2270 PRINT@181,CHR$(233)
2280 PRINT@220,CHR$(232);CHR$(233)
2290 PRINT@259,CHR$(232);CHR$(232);CHR$(

```

```

233)
2300 PRINT@301,CHR$(225)
2310 SOUND 2348,25: SOUND 1864,25: SOUND
7456,25: SOUND11172,25
2320 FOR T=1 TO 500:NEXT T
2330 RETURN
2340 REM DOORS
2350 PRINT@282,CHR$(234);" ";CHR$(233);"
";CHR$(234);" ";CHR$(233);" ";CHR$(234
);" ";CHR$(233)
2360 PRINT@242,CHR$(233);"1";CHR$(234);"
";CHR$(233);"2";CHR$(234);" "CHR$(233
);"3";CHR$(234)
2370 PRINT@202,CHR$(233);" ";CHR$(234);"
";CHR$(233);" ";CHR$(234);" ";CHR$(23
3);" ";CHR$(234)
2380 PRINT@162,CHR$(235);CHR$(231);CHR$(
236);" ";CHR$(235);CHR$(231);CHR$(236);
" ";CHR$(235);CHR$(231);CHR$(236)
2390 SOUND 3134,25: SOUND 12538,25: SOUN
D 2348,25: SOUND 9394,25: SOUND 1660,25:
SOUND 6642,25
2400 FOR T=1 TO 500:NEXT T
2410 RETURN
2420 REM DRAGON
2430 PRINT@59,CHR$(229);" ";CHR$(230)
2440 PRINT@100,CHR$(229);CHR$(228);CHR$(
227);CHR$(230)
2450 PRINT@139,CHR$(228);CHR$(238);CHR$(
239);CHR$(239);CHR$(237);CHR$(227)
2460 PRINT@178,CHR$(228);CHR$(238);CHR$(
232);CHR$(239);CHR$(239);CHR$(232);CHR$(
237);CHR$(227)
2470 PRINT@218,CHR$(239);CHR$(239);CHR$(
239);CHR$(254);CHR$(252);CHR$(239);CHR$(
239);CHR$(239)
2480 PRINT@258,CHR$(236);CHR$(167);CHR$(
167);CHR$(167);CHR$(167);CHR$(167);CHR$(
167);CHR$(235)
2490 PRINT@299,CHR$(229);CHR$(232);CHR$(
232);CHR$(232);CHR$(232);CHR$(230)
2500 SOUND 15000,50: SOUND 16383,50: SOU
ND 15000,50
2510 FOR T=1 TO 500:NEXT T
2520 RETURN
2530 REM WELL
2540 PRINT@59,CHR$(252);CHR$(239);CHR$(2
39);CHR$(239);CHR$(239);CHR$(239);CHR$(2
39);CHR$(254)
2550 PRINT@100,CHR$(233);" ";CHR$(234
)
2560 PRINT@140,CHR$(237);CHR$(232);CHR$(
232);CHR$(232);CHR$(232);CHR$(238);CHR$(
238)
2570 PRINT@180,CHR$(235);CHR$(231);CHR$(
231);CHR$(231);CHR$(231);CHR$(236)
2580 PRINT@219,CHR$(239);CHR$(255);CHR$(
239);CHR$(255);CHR$(239);CHR$(255);CHR$(
239);CHR$(255)
2590 PRINT@259,CHR$(255);CHR$(239);CHR$(
255);CHR$(239);CHR$(255);CHR$(239);CHR$(
255);CHR$(239)
2600 PRINT@299,CHR$(239);CHR$(255);CHR$(
239);CHR$(255);CHR$(239);CHR$(255);CHR$(
239);CHR$(255)
2610 FOR T=1 TO 500:NEXT T
2620 SOUND 1567,25: SOUND 3134,20: SOUND
6269,25: SOUND 12538,25
2630 RETURN
2640 REM IDOL
2650 PRINT@99,CHR$(252);CHR$(239);CHR$(2
39);CHR$(239);CHR$(239);CHR$(239);CHR$(2
39);CHR$(254)
2660 PRINT@139,CHR$(253);CHR$(239);CHR$(
27);"p";CHR$(92);CHR$(245);CHR$(245);CHR
$(47);CHR$(27);"q";CHR$(239);CHR$(251)
2670 PRINT@180,CHR$(253);CHR$(239);CHR$(
27);"p";CHR$(92);CHR$(47);CHR$(27);"q";C
HR$(239);CHR$(251)
2680 PRINT@221,CHR$(253);CHR$(27);"p";CH
R$(92);CHR$(47);CHR$(27);"q";CHR$(251)
2690 PRINT@262,CHR$(253);CHR$(251)
2700 SOUND 932,25: SOUND 2793,50: SOUND
7456,25: SOUND 1174,25
2710 FOR T=1 TO 500:NEXT T
2720 RETURN
2730 END

```

## Renumber 100

by Beve Woodbury  
80 Micro Technical Editor

Running out of room to insert additional line numbers in a program is frustrating. This renumber utility solves that problem. It lets you choose line increments and then renumbers the entire program, including all GOTO, GOSUB, Then, and Else lines. (See Program Listing 4.)

The program first reads a Do file (ASCII), then writes a file (NWNUMB.DO) with the corrected line numbers. Because of these file requirements, the number of bytes available to renumber a program must at least equal the number of bytes in the Do file plus 500 bytes for string space and 4 bytes for each line (array). I can renumber a file of approximately 7,000 bytes on a 24K Model 100 with no other files.

When you run the program, it requests the name of the file you want renumbered. Enter the file name, including the .DO extension. The computer pauses while it determines the number of lines in the file and dimensions an array to store the old and new program line numbers.

The program then requests the desired line increments. After you enter this figure, there is another pause while the

Line	Description
10	Clear \$ space for old and renumbered line
20	Title
30	Get name of file to renumber
40-70	Read file to count lines and dimension array
80-120	Set up array of old and new line numbers
130-140	Open files for input and output
150-190	Read old line and replace line number
200-230	Search for GOTO in line
240-270	Search for GOSUB in line
280-310	Search for THEN followed by line number in line
320-360	Search for ELSE followed by line number in line
370	Get next line
380-390	Sound "beep" and end program
400-520	Replace old line number reference with new

Table 5. Renumber line descriptions.

old/new line-number array is set up.

When the program starts to number lines, "Processing, Please Wait" appears on the screen each time a new line is read. If the program is quite long and a line has several line

references, the processing notice remains steady while all the line changes are checked through the array.

It took approximately one minute to renumber the 7,000-byte program mentioned above. This is a long time by computer standards, but it beats renumbering manually. When the program finishes writing the renumber file, it beeps.

```

10 CLEAR500
20 CLS:PRINT@50,"RENUMBER UTILITY":
PRINT
30 INPUT"NAME OF FILE TO RENUMBER: ";F$
40 FF$="RAM:"+F$:OPENFF$FORINPUTAS1
50 IF EOF(1)THENCLOSE:GOTO70
60 LINEINPUT#1,A$:D=D+1:GOTO50
70 D=D-1:DIMT(D,1)
80 OPENFF$FORINPUTAS1:PRINT
90 INPUT" LINE INCREMENT DESIRED: ";I
100 FORK=0TOD:LINEINPUT#1,A$:N=N+I
110 V=VAL(LEFT$(A$,5)):T(K,0)=V:T(K,1)=N

120 NEXT
130 CLOSE:OPENFF$FORINPUTAS1
140 OPEN"RAM:NWNUMB.DO"FOROUTPUTAS2
150 FORK=0TOD:CLS
160 PRINT@125,"PROCESSING, PLEASE WAIT"
170 LINEINPUT#1,A$:V=VAL(LEFT$(A$,5))
180 Y=LEN(A$):H=LEN(STR$(V))
190 A$=STR$(T(K,1))+MID$(A$,H)
200 X=1
210 X=INSTR(X,A$,"GOTO")
220 IFX>0THENGOSUB410ELSE240
230 GOTO210
240 X=1
250 X=INSTR(X,A$,"GOSUB")
260 IFX>0THENGOSUB400ELSE280
270 GOTO250
280 X=1
290 X=INSTR(X,A$,"THEN"):IFX=0THEN320
300 IFVAL(MID$(A$,X+4))>
0THENGOSUB410ELSEX=X+4
310 GOTO290
320 X=1
330 X=INSTR(X,A$,"ELSE"):IFX=0THEN360
340 IFVAL(MID$(A$,X+4))>
0THENGOSUB410ELSEX=X+4
350 GOTO330
360 PRINT#2,A$
370 NEXT
380 CLOSE:SOUND
4697,75:IFUD=1THENPRINT"UNLISTED LINE
390 END
400 X=X+1
410 X=X+3:Y=Y+1
420 M=VAL(MID$(A$,X+1))
430 R=-1
440 R=R+1
450 IFR>DTHENP$=" ****":UD=1:GOTO480
460 IFM=T(R,0)THENPP$=STR$(T(R,1))
:P$=MID$(PP$,2):GOTO480
470 GOTO440
480 IFY<X+4THEN520
490 C$=RIGHT$(STR$(M),2):LC=LEN(C$)
:Q=INSTR(X,A$,C$):C=Q+LC
500 A$=LEFT$(A$,X)+P$+MID$(A$,C)
510 RETURN
520 A$=LEFT$(A$,X)+P$
530 RETURN

```

Program Listing 4. Renumber utility.

This lets you do other tasks while your program is renumbered.

If there is an undefined line number ("Line number is not in program") referenced in a line, the number is replaced with asterisks.

When the program ends, "Unlisted Line" appears on the screen.

The renumbered file is in NWNUMB.DO. The original file is unchanged. ■

Variable	Description
A\$	Old file line input
C	Location to put remainder of line
C\$	Remainder of line
D	Number of lines in program
F\$	Name of file to renumber
FF\$	Proper file name format to open file
H	Length of new line number
I	Increment amount
J	Length of old line number reference
K	Loop counter
LC	Length of remainder of line
M	Value of MIDS
N	New line number
PP\$	New line number reference
P\$	Formatted new line number
Q\$	Location of remainder of line
R	Array row number
T()	Table (array) of old and new line numbers
UD	Indicate presence of undefined line number
V	Value of old line number
X	Location in line
Y	Length of old line
Z\$	New file line output

Table 6. Renumber variables list.

## The Year in Review

by Richard Ramella

This program produces a bar graph that displays data for a 12-month period on the X axis and accommodates up to 10 million units on the Y axis. It requires an 8K Model 100. The display fits comfortably on the screen.

### The Program

When you run the program, you'll see the prompt "Enter title in 40 or fewer characters." It can be anything: 1983 Births at Enloe Hospital, Sales of the South Pacific, and so on. If you exceed the 40-character limit, you're told so and given another try.

Then you are asked to enter data for each month, from January to December. For each prompt in this section, type the number of units for that month and hit the enter key to see the next prompt. If you enter a number higher than 10 million, the program politely ends its participation in your scheme.

After you enter the December figure, the screen blanks and the program draws the bar graph.

At the left of the screen the prompt "Month?" appears. Answer it by typing the first three letters of any month in low-

erage and the program displays the number for that month. To see a figure for another month, press the enter key. The screen goes blank and another prompt appears.

The strange letter groupings in lines 370-390 of Program Listing 5 turn into three-letter month abbreviations displayed vertically. The bar for each month builds upward to the right of the month.

The Y axis has 10 divisions on it. At the bottom left of the screen is a notation telling you what each of these 10 increments represents. The abbreviation "incr." is followed by either "tens," "hundreds," "thousands," "10 thous.," "100

thous.," or "millions."

The program notes the largest number you enter and then scales down all the other numbers so the bars displayed are in proportion to each other.

I arbitrarily set a limit of 10 million units. If you have more than 10 million of something in any one month, you might represent your figures with decimals, for example, 300.200000 representing three hundred million, two hundred thousand. ■

Write to Richard Ramella at 1493 Mountain View Ave., Chico, CA 95926.

Program Listing 5. The Year in Review.

```

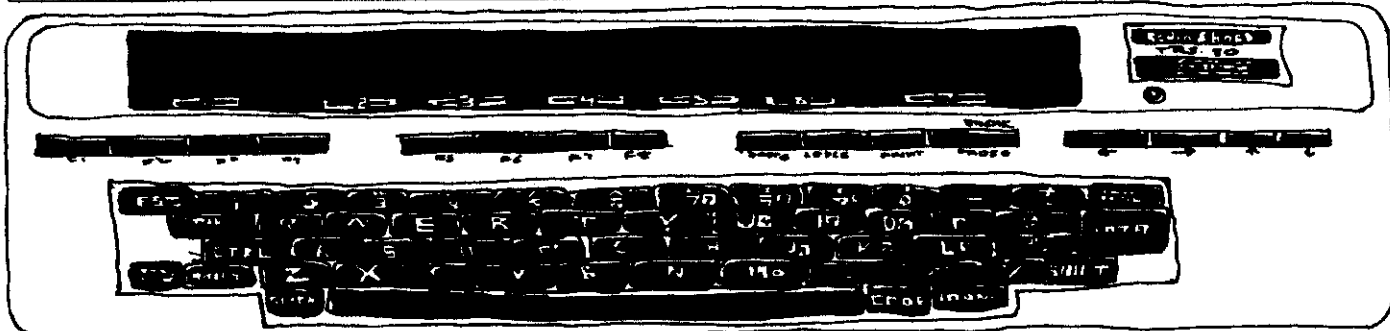
100 REM * Bar Graph * TRS-80 Model 100
8K * Richard Ramella
110 CLS
120 CLEAR 200
130 DIM B(12),B$(12)
140 DATA Jan., Feb., Mar., Apr., May.,
June., July., Aug., Sept., Oct., Nov.,
Dec.
150 FOR A=1 TO 12
160 READ B$(A)
170 NEXT A
180 M$=
"janfebmaraprmayjunjulaugsepoctnovdec"
190 PRINT "Enter title in 40 or fewer
characters"
200 INPUT A$
210 IF LEN(A$)>40 THEN PRINT "Title"LEN(
A$)-40"character too long. Try again?":
GOTO 190
220 CLS
230 FOR A=1 TO 12
240 PRINT B$(A)" figure";
250 INPUT B(A)
260 IF B(A)>10000000 THEN PRINT "I'm
sorry. I only accept numbers up to 10
million (10000000)...": END
270 IF B(A)>Z THEN Z=B(A)
280 NEXT A
290 CLS
300 IF Z>10000000 THEN G=2000000:
Z$="millions": ELSE IF Z>1000000 THEN
G=200000: Z$="100 thous." ELSE IF Z>100000
THEN G=20000: Z$="10 thous." ELSE IF Z>
10000 THEN G=2000: Z$="thousands" ELSE IF
Z>1000 THEN G=200: Z$="hundreds" ELSE G=2:
Z$="tens"
310 FOR A=1 TO 12
320 B(A)=B(A)/G
330 NEXT A
340 PRINT @ 240,"incr:"Z$;

```

```

350 C=1
360 PRINT @ 0,A$
370 PRINT @ 216,"j f m a m j j a s o n d
";
380 PRINT @ 256,"a e a p a u u u e c o e
";
390 PRINT @ 296,"n b r r y n l g p t v c
";
400 T=92
410 F=13
420 FOR W=92 TO 238 STEP 2
430 PSET(W,F)
440 NEXT W
450 FOR W=13 TO 63
460 PSET(T,W)
470 IF L=0 OR L/5=INT(L/5) THEN FOR Q=T
TO T-3 STEP -1: PSET(Q,W): NEXT Q
480 L=L+1
490 NEXT W
500 C=1
510 FOR A=102 TO 242 STEP 12
520 FOR B=63 TO 63-B(C) STEP -1
530 FOR D=A TO A+4
540 PSET(D,B)
550 NEXT D
560 NEXT B
570 C=C+1
580 NEXT A
590 PRINT @ 80,SPACE$(14);
600 PRINT @ 80,"";
610 PRINT "month..."
620 L=0
630 INPUT R$
640 PRINT @ 80,SPACE$(14);
650 PRINT @ 120,SPACE$(14);
660 FOR A=1 TO 34 STEP 3
670 L=L+1
680 IF R$=MID$(M$,A,3) THEN PRINT @
80,R$; PRINT @ 120,B(L)*G;: GOTO 700
690 NEXT A
700 IF INKEY$<>" THEN PRINT @ 80,SPACE$
(14);: PRINT @ 120,SPACE$(14);: GOTO 590
ELSE 700
710 END

```



# Nothing's Perfect

by Ken Barbier

Ah, the benefits of the Model 100: optional 24K bytes of battery back-up RAM, RS-232 serial port, built-in text editor, built-in modem, ability to upload into a larger computer, full-size keyboard, and so easy to carry. But does the computer have any flaws?

## Run Speed

I keyed a little benchmark program calculating the cosine of one 240 times (Program Listing 6) into the 100, and got what I thought was a dead computer. It wasn't dead, but it was slow. It took 28 seconds to run a benchmark that takes less than four seconds on my 4 MHz Z80 homebuilt computer, and seven seconds on a TRS-80 Model I.

The 100 Basic computer functions in 14-digit double-precision calculations, and while that provides precise answers, it does not provide speed. Patching the benchmark program to force the variable X to single precision, and the counter I to integer produced a result 28 seconds later with six digits, not 14.

## Manual Oversights

I spotted a couple of weaknesses in the generally excellent 100 user's manual. The Call statements in the sample programs on pp. 149, 199, and 201 reference machine-language subroutines contained in the Model 100 PROM. Call is documented, but PROM isn't.

The manual doesn't mention that the maximum number of files is limited to the 20 files displayed at sign-on. If you ask Text to create a twenty-first file, you get a beep and another "File to Edit?" prompt, not a warning that the directory is full.

The sample program on p. 199 uses the Basic PRINT@ function without fully explaining it. The inference you must draw is that PRINT 40 causes the next display to start 40 positions from the screen's first character spot.

The Basic initial program load function (IPL) lets you pre-set the computer to load and execute a program the next time you turn on the computer. But if you don't have Basic loaded when you turn the computer off, it powers up the menu instead of your IPL program.

If you ever turn the microcomputer off during the execution

of a program, the program resumes once you restore power. This is handy if you use the 100 with an unattended controller of data-acquisition device powered from the optional ac adapter, but the user's manual unfortunately doesn't ever tell you this happens.

## Terminal Hang-ups

Don't use shortcuts with Telecom in the TERM mode. If you try to step through the procedures for calling another computer on the telephone without actually connecting the modem cable to the phone, you end up in limbo as soon as you press the TERM mode key. The computer is waiting to hear the modem tone from the other end of the phone link before it updates the function key display to that shown on p. 81 of the manual.

Try a dry run before you call another computer. Switch your Model 100 from ORIG to ANS (find the switch on the left) before stepping through the TERM procedures. This makes your computer act like the called machine instead of the caller, and you hear the modem tone as soon as you enter TERM. The computer hangs up at this point, since it's waiting for a call.

Use STAT to change from the modem operation to the RS-232 port to make a dry run all the way through the TERM mode. Enter STAT and key in 88NID. You can then fake an upload of a file from your computer through the serial port, even though there's nothing at the other end.

The computer remembers the STAT assignment, and won't revert to the modem port the next time you power up. Use STAT and the manual's table on p. 85 to restore modem operation.

## The RS-232 Port

You can't connect a standard RS-232 data cable to the 100 because the cutout in the computer's case is too small to accommodate the connector shell. Worse, the serial port doesn't conform to RS-232 standard.

A table on p. 205 of the documentation implies that the Model 100 serial port uses the data send and receive signals on pins 2 and 3, and the handshaking signals on RS-232 connector pins 4, 5, 6, and 20. Unfortunately, the computer ignores the handshaking signals.

The Model 100 is configured as a terminal device (DTE) to be connected to a communications device (DCE). A DTE should assert the handshake line Data Terminal Ready (DTR, pin 20) when it is powered up, and Ready To Send (RTS, pin 4) when it wants to talk to the DCE.

Similarly, the communications device on the other end of the RS-232 cable (such as a modem), asserts Data Set Ready (DSR, pin 6) when the machine is turned on, and Clear To Send (CTS, pin 5) when it is ready to accept data. The DTE (Model 100, in this case), should stop sending data if the DCE is not ready (DSR, Not True), or is busy (CTS, Not Ready). The Model 100, however, ignores both DSR and CTS.

The 100 always asserts DTR when in the TERM mode, and does not use RTS at all. When the computer is used to upload

```
10 PRINT "Rickard Cosine Benchmark"
20 PRINT "START ";TIMES
30 BEEP
40 FOR I=1 TO 240
50 X=COS(1.0)
60 NEXT
70 BEEP
80 PRINT "STOP ";TIMES
90 PRINT X
100 INPUT A$
110 MENU
```

Program Listing 6. Benchmark program.



files to, or to download files from, another computer at higher baud rates, data can be lost.

I can connect the serial port to my big computer and upload text files to it at 9600 baud (see Sidebar: "Uploading Text to WordStar"), or I can use the Model 100 as a full duplex terminal on the other computer if I don't exceed a 300 baud rate. If I run at higher data rates, the Model 100 screen scroll takes too long, and data from my big computer to Model 100-as-terminal is lost.

If I hook up my Olympia daisy wheel typewriter/printer to the serial port of the Model 100 to use it as a 300 baud letter-quality printer, the 100 ignores the DSR signal, and sends data faster than the printer can print, garbaging the printout.

## Uploading Text to WordStar

The TRS-80 Model 100 built-in text editor, Text, stores data in document files (.DO) in a format incompatible with larger microcomputers running WordStar under CP/M. I had to find a simple method to upload Text files to WordStar.

If the Model 100 Telcom facility is used to upload files, it demands that you tell it how long a text line should be, and inserts hard carriage return (CR) characters into the text at the end of each line as the data is uploaded to the target computer (or modem, or printer).

WordStar, on the other hand, inserts soft end-of-lines consisting of a carriage return character (0DH in hexadecimal) with the eighth bit set on (8DH), followed by a line feed (LF) character (0AH). Only paragraph ends are marked with standard CR, LF (0DH, 0AH) sequences in WordStar.

Since WordStar gets confused by plain text with a CR at the end of each line, but no LF, it can't be used to reformat such files. Therefore, the Model 100 Telcom program can't upload files to WordStar. To upload text from the Model 100 to my WordStar-CP/M system, I had to write UPLOAD.BA, shown in Program Listing 7.

This program takes a Text document file named by the operator and sends it out through the serial port to a host computer without inserting CRs at line ends. Paragraph breaks are sent as they exist within a Text file, with a CR and LF, which is already WordStar-compatible. The end of the text file is signaled to the host com-

```

10 PRINT "TRS-80 Model 100 TEXT to WordStar"
20 MAXFILES=2
30 EF$=CHR$(26)
40 INPUT "File to send: ";FIS
50 OPEN FIS FOR INPUT AS 1
60 OPEN "COM:88N2D" FOR OUTPUT AS 2
100 AS=INPUT$(1,1)
110 PRINT #2,AS;
120 IF EOF(1) THEN GOTO 1000
130 GOTO 100
1000 PRINT #2,EF$
1010 MENU
    
```

*Program Listing 7. This program permits uploading Model 100 Text files to a host computer running WordStar under CP/M.*

## The Up-Side

The Model 100 is perfect for remote sites: programmers working in any language can use the 100 to key in, edit, and store their source programs wherever they are. The 14 digits of double-precision accuracy top the accuracy of the single-precision functions of other computers.

If you live in an area with power interruptions, you can avoid computer glitching by keying in text with the 100 running on batteries. The built-in calendar clock and modem provide numerous remote controller possibilities. ■

*Ken Barbier can be reached at P.O. Box 1253, Borrego Springs, CA 92004.*

puter by the CTRL Z character (1AH).

This last special character is defined in the program (see Listing 7) in line 30. Line 40 prompts the operator for the name of the file to send, which should be entered in the form FILE.DO without quotation marks. Line 50 then opens this file as input.

Output to the serial port is established in line 60 with the baud rate set at 9600, 8 data bits, no parity, 2 stop bits, and XON/XOFF disabled (see the manual for details on setting up the serial port).

The document file is then transmitted character by character by the program loop starting at line 100. When the end of the document file is reached, a CP/M-compatible terminator (EF\$, 1AH, CTRL Z) is sent to the host computer, and the Model 100 returns to the menu display.

For this output to be properly received by a CP/M computer, the Model 100 serial port has to be connected to the host computer reader device (RDR:) serial port. Since no serial port handshaking signals are recognized by the Model 100, the text has to be sent in a continuous block.

To receive an uninterruptible input through the RDR:, the host computer inputs and buffers the text from the start to the end-of-file character, before writing it to the disk. This is accomplished under CP/M by instructing the Peripheral Interchange Program (PIP) to create a disk file consisting of reader device input buffered in memory until the end-of-file is received:

```
PIP TEXT.DOC=RDR:[B]
```

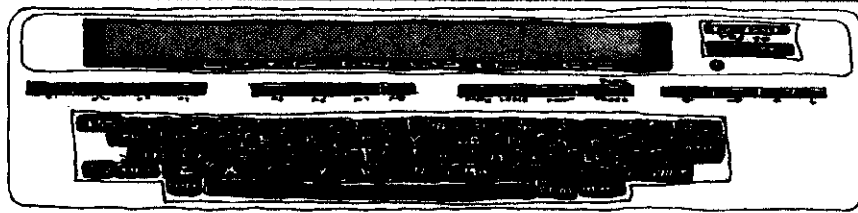
Once Text.DOC (or whatever file name you chose on the receive end) has been written to the disk by PIP, WordStar can then reformat the file and insert its own soft end-of-line characters. Invoke WordStar, select Text.DOC as a document file, and execute the WordStar global reformat with the command sequence:

```
CTRL Q, Q, CTRL B
```

What started as a Model 100 Text file is now a WordStar-compatible document file.

A similar technique can be used to upload Text files to other word processors running under other operating systems. ■

C-NOTE FOR SEPTEMBER 1983



## Model 100 Conversions: Easy as I, II, III

by Beve Woodbury  
80 Micro Technical Editor

You can convert many Model I, II, and III programs for the Model 100 with few modifications. In fact, some programs don't require any changes. This article will help you choose and convert programs appropriate for the Model 100.

### What to Avoid

Avoid programs that require direct access (disk) file input and output. As of now, the 100 offers only sequential file input and output. Since Model 100 programs and files are stored within the computer, your 100's memory size limits the size of the programs and files you can use.

Also avoid programs that use POKEs and PEEKs. If you're not familiar with machine language and the Model 100's memory map, these commands may give you some strange results.

Finally, avoid programs that require large screen displays such as game graphics. You can alter screen displays for the Model 100 providing you keep in mind the limitations imposed by the 40-character by 8-line display. Change any Print@ statements in your conversion so that it prints at the correct location on your 100.

### A Sample Program

Math Program is a Model I/II/III program that I converted to run on the Model 100 as a conversion example (see Program Listing 1).

Once you boot up the program, enter your name and choose from a menu to add, subtract, multiply, divide, or quit. When you choose the function you want, you're given the option of Level 1 (for numbers 0-9) or Level 2 (for numbers 0-100).

The program uses random numbers in the math problems. It prints the problem on the screen and waits for your answer. If you answer correctly, the computer tells you so; if you're wrong, the screen displays WRONG along with the correct answer.

When you quit the program, it reads an input sequential file to see if your name exists in the file. If it does, the program

prints the scores you achieved last time along with your current scores. Scores are calculated for each math function as well as overall.

Your current scores replace your old scores on the output sequential file or, if you aren't already on the file, the program adds your name and scores to the end of the output file.

### Establishing Files

Prior to running Math Program on any TRS-80, you must enter Basic with two files and an established sequential input file.

On Models I/II/III, enter Basic by typing BASIC - F:2. To establish the sequential input file for Math Program, type in:

```
OPEN "O",1,"SCORES":CLOSE
```

and press the enter key.

On the Model 100, the command Maxfile=2 in line 10 opens the Basic file channels. Enter the text mode to establish the sequential input file. When the screen prints "File to edit?", type SCORES, press the enter key, and then press the Model 100's F8 key.

You only need to establish the Scores file before the program's first run or when the Scores file has been killed.

### Model 100 Math Program Changes

Program Listing 2 is Math Program converted for the Model 100. Compare the two programs. I'll explain what changes were made and why. I have kept the line numbers the same in both programs for easy comparison.

### Remark Statements

Remark statements, indicated by REM or an apostrophe, help you understand and debug a program. They illustrate the program's flow, explain what's being done in certain areas of the program, and help locate problem areas. Remarks also define a particularly confusing or unusual command area.

Each character or space in a remark statement takes up 1 byte of memory. Because of the Model 100's memory limitation, it is advisable to remove all spaces and remark statements from the program. But you should keep a printed copy of the complete program for future reference in making changes, debugging, or understanding program flow.

### GOTO and GOSUB Statements

Programmers sometimes have GOTO or GOSUB statements reference a remark statement that explains a routine that follows. When you remove REM lines, be sure to change the line number in any GOTO or GOSUB statement that references, the deleted remark line number to the correct number. See program lines 390 and 810. If you miss one of these corrections, you'll get an undefined line error (UL).

### Altering the Display Command

Print @ statements require the most changes (see lines 290-360 in Listing 1). You have to modify print locations from a

### The Key Box

"Model 100 Conversions: Easy as I, II, III" applies to any Model 100. "Disassembler," "Elapsed Time," and "If It's Tuesday..." all require 8K RAM.

64-character by 16-line format (Model I/III) or an 80-character by 23-line format (Model II) to the Model 100's format. Map the necessary changes by plotting the print locations on the 100's screen graph paper (p. 210 in the Model 100 manual) or by trial and error. The 100's screen limitation may make many programs unusable or unpleasant when converted.

If you have many lines of print that don't fit on the eight lines available, you'll want to control the speed of the screen scroll. You can accomplish this with For...Next loops every eight lines. I prefer to use an INKEY\$ routine (line 270) that allows the user to go to the next screen at his own rate by pressing any key.

**Opening Files**

Model 100 commands to open files are different from those

*Program Listing 1. Math Program for Models I, II, and III.*

```

10 * *****
20 * *
30 * * MATH PRACTICE *
40 * * by Beve Woodbury *
50 * *
60 * *****
70 *
80 *
90 *
100 CLS
110 ON ERROR GOTO 1140
120 FS=" #.## #.## #.## #.## #.##"
130 REM beginning screen
140 PRINT# 215, "MATH PRACTICE"
150 PRINT# 723, "TYPE YOUR NAME, PLEASE."
160 PRINT# 855, " ";
170 INPUT NS
180 CLS
190 PRINT "You will now have a chance to practice your math."
200 PRINT
210 PRINT "You may choose what math function you want to practice then I will choose some number problems for you to solve."
220 PRINT
230 PRINT "I will keep score of how many problems you get right and how many you get wrong. When you QUIT, I will tell you your score."
240 PRINT
250 PRINT "I will also tell you if you did better than the last time you practiced."
260 PRINT# 845, "Press ANY key to continue."
270 IS=INKEY$:IF IS="" THEN 270
280 REM print math menu
290 CLS
300 PRINT#153, "MENU"
310 PRINT#340, "1. ADD"
320 PRINT#404, "2. SUBTRACT"
330 PRINT# 468, "3. MULTIPLY"
340 PRINT# 532, "4. DIVIDE"
350 PRINT# 596, "5. QUIT"
360 PRINT#833, "PRESS NUMBER OF MATH FUNCTION WANTED, THEN PRESS THE ENTER KEY"
370 PRINT# 928, " ";:INPUT F
380 IF F>5 THEN 290
390 IF F=5 THEN 750
400 T=T+1
410 CLS:PRINT:PRINT
420 PRINT " LEVEL 1 numbers 0 to 9"
430 PRINT
440 PRINT " LEVEL 2 numbers 0 to 100"
450 PRINT:PRINT
460 INPUT "PRESS 1 OR 2 THEN PRESS THE ENTER KEY";L
470 IF L>2 THEN 410
480 CLS:ON L GOTO 500, 530
490 REM get level I random numbers for problem
500 A=RND(9)
510 B=RND(9)
520 GOTO 550
530 A=RND(100)
540 B=RND(100)
550 ON P GOTO 570,620,670,710
560 REM addition
570 CLS
580 PRINT#400, A"+"B="";:INPUT C
590 IF C=A+B THEN AR=AR+1:GOTO1120
600 PRINT# 530, A"+"B"="A+B:AW=AW+1:GOTO1130
610 REM subtraction
620 IF B>A THEN H=A:A=B:B=H:verify minuend larger number
630 PRINT#400, A"-B="";:INPUT S
640 IF S=A-B THEN SR=SR+1:GOTO 1120
650 PRINT#530, A"-B"="A-B:SW=SW+1:GOTO 1130
660 REM multiplication
670 PRINT#400, A"*B="";:INPUT M
680 IF M=A*B THEN MR=MR+1:GOTO 1120
690 PRINT#530, A"*B"="A*B:MW=MW+1:GOTO 1130
700 REM division

```

*Listing 1 continued in next column*

*Listing 1 continued from previous column*

```

710 DV=A*B 'verify even division
720 PRINT#400, DV/"B"="";:INPUT D
730 IF D=DV/B THEN :DR=DR+1:GOTO 1120
740 PRINT# 530, DV/"B"="DV/B: DW=DW+1:GOTO1130
750 REM ending routine
760 REM read old and write new file
770 OPEN "I",1,"SCORES"
780 OPEN "O",2,"NNSCORE"
790 IF EOF(1) THEN 930
800 INPUT#1, N1$,A1,A2,S1,S2,M1,M2,D1,D2,T1
810 IF N1$=N$ THEN 840
820 PRINT#2, N1$,"";A1;A2;S1;S2;M1;M2;D1;D2;T1
830 GOTO 790
840 REM get old score
850 PA=A1/(A1+A2)
860 PS=S1/(S1+S2)
870 PM=M1/(M1+M2)
880 PD=D1/(D1+D2)
890 PT=(A1+S1+M1+D1)/T1
900 GOSUB 1100 ' write new score to file
910 Y=1:GOTO 790
920 REM get new percentage scores
930 NA=AR/(AR+AW)
940 NS=SR/(SR+SW)
950 NM=MR/(MR+MW)
960 ND=DR/(DR+DW)
970 NT=(AR+SR+MR+DR)/T
980 REM print scores
990 CLS:PRINT:PRINT
1000 PRINT#153,N$
1010 PRINT:PRINT
1020 PRINT" ADD SUB MULT DIV
TOTAL"
1030 PRINT
1040 IF Y=0 THEN GOSUB 1100:GOTO1070
1050 PRINT "OLD ";:PRINT USING F$;PA,PS,PM,PD,PT
1060 PRINT
1070 PRINT "NEW ";:PRINT USING F$;NA,NS,NM,ND,NT
1080 PRINT:PRINT
1090 CLOSE:KILL"SCORES":CHD"RENAME NNSCORE SCORES":END
1100 PRINT#2,N$,"";AR;AW;SR;SW;MR;MW;DR;DW;T
1110 RETURN
1120 PRINT#530, "RIGHT":GOTO 260
1130 PRINT#420, "WRONG":GOTO 260
1140 IF ERR/2+1=11 THEN RESUME NEXT 'continue if /0 error

```

- A Random number—addend, minuend, multiplicand, or quotient
- AR Total of correct additions
- AW Total of wrong additions
- B Random number—addend, subtrahend, multiplier, or divisor
- C Sum in addition
- D Quotient in division
- DR Total of correct divisions
- DV Dividend in division
- DW Total of wrong divisions
- F Math function choice
- F\$ Print format string
- H Hold variable for making minuend larger than subtrahend
- IS Inkey variable
- L Level wanted
- M Product in multiplication
- MR Total of correct multiplications
- MW Total of wrong multiplications
- N\$ Player's name
- NA New addition percentage
- ND New division percentage
- NM New multiplication percentage
- NS New subtraction percentage
- NT New total percentage
- N1\$ Name in file
- PA Old addition percentage
- PD Old division percentage
- PS Old subtraction percentage
- PT Old total percentage
- S Difference in subtraction
- SR Total of correct subtractions
- SW Total of wrong subtractions
- T Total of problems attempted

*Table 1. Math Program variable list.*

of the other TRS-80s (lines 770 and 780). These commands must appear as shown in Listing 2. Spaces can be inserted between words if desired. Refer to p. 165 in the Model 100 manual for a more detailed explanation of opening files.

**Random Numbers**

The Model 100's random number generator returns numbers between 0 and 1. Therefore, all random numbers are decimal numbers to 14 places. The program produces whole numbers by multiplying the decimal numbers generated (line 1520 in Listing 2).

The random numbers are not truly random, however, as the same number sequence is generated each time a program runs. The Model 100 manual provides a short program on p. 175 that ties the random number generator to real time. This produces a different number sequence for each program execution (lines 1500-1510 in Listing 2).

**Other Conversion Problems**

I am aware of two other problems not demonstrated in Math Program. You must include the Then in an If...Then...GOSUB statement. You cannot use If...GOSUB as you can on the Model III. But in an If...Then...GOTO statement, either the Then or the GOTO can be left out.

Character strings (CHR\$) 32-125 are the same on all TRS-80s. Any other character string may or may not be different on the 100 and must be checked carefully to determine the appropriate CHR\$ command for the Model 100.

**More Changes?**

This covers the changes I've found necessary in converting programs to the Model 100. Give it a try and if you come across any others, let me know about them and I'll include them in a future issue. ■

*Program Listing 2. Math Program for the Model 100.*

```

10 MAXFILES=2
100 CLS
110 ONERRORGOTO1140
120 F$=" #.## #.## #.## #.##
#.##"
140 PRINT@92,"MATH PRACTICE
150 PRINT@167,"TYPE YOUR NAME, PLEASE.
160 PRINT@254," ";:INPUT N$
210 CLS:PRINT"Choose the math function
you want.
220 PRINT"I will print number problems
to solve.
230 PRINT:PRINT"When you quit, I will
print your scores and your last scores.
260 PRINT@243,"Press ANY key to
continue.
270 I$=INKEY$:IFI$=""THEN270
290 CLS:PRINT@56,"1. ADD
320 PRINT@96,"2. SUBTRACT
330 PRINT@136,"3. MULTIPLY
340 PRINT@176,"4. DIVIDE
350 PRINT@216,"5. QUIT
360 PRINT@280,"ENTER NUMBER OF MATH
FUNCTION WANTED";:INPUTF
380 IFF>5THEN290
390 IFF=5THEN770
400 T=T+1

```

```

410 CLS:PRINT:PRINT" LEVEL 1
numbers 0 to 9
430 PRINT:PRINT" LEVEL 2 numbers 0
to 100
460 PRINT:PRINT"ENTER LEVEL SELECTION 1
OR 2 ";:INPUTL
470 IFL>2THEN410
480 CLS:ONLGO500,530
500 GOSUB1500:A=X1:GOSUB1500:B=X1
520 GOTO550
530 GOSUB1500:A=X2:GOSUB1500:B=X2
550 ONFGOTO570,620,670,710
570 CLS:PRINT@95,A"+"B"="";:INPUTC
590 IFC=A+BTHENAR=AR+1:GOTO1120
600 PRINT@175,A"+"B"="A+B:AW=AW+
1:GOTO1130
620 IFB>ATHENH=A:A=B:B=H
630 PRINT@95,A-"B"="";:INPUTS
640 IFS=A-BTHENSR=SR+1:GOTO1120
650 PRINT@175,A-"B"="A-B:SW=SW+
1:GOTO1130
670 PRINT@95,A*"B"="";:INPUTM
680 IFM=A*BTHENMR=MR+1:GOTO1120
690 PRINT@175,A*"B"="A*B:MW=MW+
1:GOTO1130
710 DV=A*B:IFB=0THENB=1
720 PRINT@95,DV/"B"="";:INPUTD
730 IFD=DV/BTHENDR=DR+1:GOTO1120
740 PRINT@175,DV/"B"="DV/B:DW=DW+
1:GOTO1130
770 OPEN"RAM:SCORES.DO"FORINPUTAS1
780 OPEN"RAM:NWSCOR.DO"FOROUTPUTAS2
790 IFEOF(1)THEN930
800 INPUT#1,N1$,A1,A2,S1,S2,M1,M2,
D1,D2,T1
810 IFN1$=N$THEN850
820 PRINT#2,N1$," ";A1;A2;S1;S2;M1;M2;
D1;D2;T1
830 GOTO790
850 PA=A1/(A1+A2)
860 PS=S1/(S1+S2)
870 PM=M1/(M1+M2)
890 PT=(A1+S1+M1+D1)/T1
900 GOSUB1100
910 Y=1:GOTO790
930 NA=AR/(AR+AW)
940 NS=SR/(SR+SW)
950 NM=MR/(MR+MW)
960 ND=DR/(DR+DW)
970 NT=(AR+SR+MR+DR)/T
990 CLS:PRINT@18,N$
1020 PRINT@88,"ADD SUB MULT DIV
TOTAL
1040 IFY=0THENGOSUB1100:GOTO1070
1050 PRINT"OLD
";:PRINTUSINGF$;PA,PS,PM,PD,PT:PRINT
1070 PRINT"NEW
";:PRINTUSINGF$;NA,NS,NM,ND,NT
1090 CLOSE:KILL"SCORES.DO":
NAME"NWSCOR.DO"AS"SCORES.DO":END
1100 PRINT#2,N$," ";AR;AW;SR;SW;MR;MW;
DR;DW;T
1110 RETURN
1120 PRINT@178,"RIGHT":GOTO260
1130 PRINT@114,"WRONG":GOTO260
1140 IFERR=11THENRESUMENEXT
1500 S=VAL(RIGHT$(TIMES,2))
1510 FORI=1TOS:X=RND(1):NEXT
1520 X1=INT(X*10):X2=INT(X*100):RETURN

```

## Disassembler

by David Cloutier

Soon after I got my Model 100, I tried to map its ROM and RAM. I downloaded the two disassemblers on CompuServe to do so, but both were too large to run on an 8K machine. So, I had to write my own program (see Program Listing 3).

Run Listing 3 and enter the memory location you wish to examine. You must enter the location in decimal format. Originally, I wrote the program so you could enter it in either decimal or hexadecimal, but, to conserve memory, I had to remove the subroutine that provided this option.

If your machine has more than 8K, you can reinsert this option by changing line 12520 to:

```
12520 INPUT "Start address (suffix with H if hex)";SS
```

and adding the lines in Fig. 1.

After the memory location you're interested in is mapped, press the R key to enter another memory location. Or, press the Q key to end the program.

One problem with this disassembler is its length. If you have an 8K machine, you'll probably have to remove all other files in memory or save them to tape. Between each mnemonic and its operand is a tab—there are no spaces.

When the program first runs, it arrives at memory location

```
10500 MS$=LEFT$(SS,2):LS$=RIGHT$(SS,2)
10510 N$=MS$:GOSUB10610:LM=L:RM=R
10520 N$=LS$:GOSUB10610
10530 S=LM*16 3+RM*16 2+L*16+R:RETURN

10610 L$=LEFT$(N$,1):IFASC(L$)>57
THENL=(240ORASC(L$))-231:ELSEL=VAL(L$)
10620 R$=RIGHT$(N$,1):IFASC(R$)>57
THENR=(240ORASC(R$))-231:ELSER=VAL(R$)
10630 RETURN
```

```
12525 IFLEN(SS)>5THEN12520
12530 IFRIGHT$(SS,1)="H"ORRIGHT$(SS,1)="
h"THENS$=LEFT$(SS,LEN(SS)-1):GOSUB12700:
GOSUB10500:ELSES=VAL(SS)
12700 IFLEN(SS)<4THENS$="0"+SS:GOTO12700
:ELSERETURN
```

Figure 1. If your Model 100 has more than 8K RAM, add these lines to the disassembler; they give the option of entering the RAM address in decimal or hexadecimal

Program Listing 3. Disassembler.

```
5 CLEAR70:DIMA$(256)
10 DATA "NOP", "LD          BC,#", "LD
(BC),A", "INC  BC", "INC  B"
20 DATA "DEC  B", "LD  B,! ", "RCLA", "EX
AF,AF'", "ADD  HL,BC", "LD  A,(BC)"
30 DATA "DEC  BC", "INC  C", "DEC
C", "LD  C,! ", "RCA", "DJNZ
@"
```

7D33. This isn't surprising since the first command in memory is JP 7D33. When the program reaches locations 7D37-7D3D, it tests for a time delay. If the check comes back negative, the program sends some numbers to port D3.

As it continues through memory, the disassembler eventually arrives at 7EE1. This subroutine tests the memory size by getting the value of the first memory location of a chip and then complementing the value and loading it back into memory. If a read finds the number is the complement of the original, the memory is RAM and the program checks for any remaining RAM chips (see "Monitor 100," 80 Micro, July 1983, p. 178 for a memory map).

If the test fails or if all the chips are accounted for, the program loads the beginning location of all RAM into FAC0.

The subroutine 5A7C copies the default labels from ROM into RAM locations F789-F809. Call 6C93 makes a copy of F789-F809 and puts it into F80A-F88A. Call 6C9C does the reverse—it copies the label line in F80A-F88A to F789-F809.

Call OFE8 retrieves the character in the memory location HL points to; if the character is lowercase, the call converts it to uppercase and returns with the new character in the HL register pair.

Call 2542 is similar to the Z80 command LDIR. It takes the block pointed to by the HL register (block length in register B) and moves it to the memory pointed to by the DE register. ■

Write to David Cloutier at Bullard Road, North Brookfield, MA 01535.

```
40 DATA "LD          DE,#", "LD          (DE),A",
"INC  DE", "INC  D", "DEC D"
50 DATA "LD          D,! ", "RLA", "JR  @", "ADD
HL,DE", "LD  A,(DE)", "DEC  DE"
60 DATA "INC  E", "DEC E", "LD  E,! ", "R
RA", "JR  NZ,@", "LD  HL,#", "LD
(####),HL", "INC HL", "INC  H"
70 DATA "DEC  H", "LD  H,! ", "DAA", "JR
Z,@", "ADD  HL,HL", "LD  HL,(####)
", "DEC HL", "INC  L", "DEC L"
80 DATA "LD          L,! ", "CPL", "JR  NC,@", "L
D          SP,#", "LD          (####),A", "INC
SP", "INC  (HL)", "DEC  (HL)"
90 DATA "LD          (HL),! ", "SCF", "JR
C,@", "ADD  HL,SP", "LD  A,(####)
", "DEC SP", "INC  A", "DEC A"
100 DATA "LD          A,! ", "CF", "LD  B,B", "LD
B,C", "LD          B,D", "LD
B,E", "LD          B,H", "LD          B,L", "LD
B,(HL)
110 DATA "LD          B,A", "LD          C,B", "LD
C,C", "LD          C,D", "LD
C,E", "LD          C,H", "LD          C,L"
120 DATA "LD          C,(HL)", "LD          C,A", "LD
D,B", "LD          D,C", "LD
D,D", "LD          D,E", "LD          D,H"
130 DATA "LD          D,L", "LD          D,(HL)",
"LD          D,A", "LD          E,B", "LD
E,C", "LD          E,D", "LD          E,E"
140 DATA "LD          E,H", "LD          E,L", "LD
E,(HL)", "LD          E,A", "LD
H,B", "LD          H,C", "LD          H,D"
150 DATA "LD          H,E", "LD          H,H", "LD
H,L", "LD          H,(HL)", "LD
```

Listing 3 continued



Listing 3 continued

```

H,A", "LD      L,B", "LD      L,C"
160 DATA"LD      (HL),C", "LD      (HL),D",
"LD      (HL),E", "LD      (HL),H", "LD
(HL),L", "HALT", "LD      (HL),A"
170 DATA"LD      (HL),C", "LD      (HL),D",
"LD      (HL),E", "LD      (HL),H", "LD
(HL),L", "HALT", "LD      (HL),A"
180 DATA"LD      A,B", "LD      A,C", "LD
A,D", "LD      A,E", "LD
A,H", "LD      A,L", "LD      A,(HL)"
190 DATA"LD      A,A", "ADD      A,B", "AD
D      A,C", "ADD      A,D", "ADD
A,E", "ADD      A,H", "ADD      A,L"
200 DATA"ADD      A,(HL)", "ADD      A,A", "AD
C      A,B", "ADC      A,C", "ADC
A,D", "ADC      A,E", "ADC      A,H"
210 DATA"ADC      A,L", "ADC      A,(HL)",
"ADC      A,A", "SUB      B", "SUB      C", "SUB
D", "SUB      E", "SUB      H", "SUB      L", "SUB      (HL)"
220 DATA"SUB      A", "SBC      A,B", "SBC
A,C", "SBC      A,D", "SBC      A,E", "SB
C      A,H", "SBC      A,L", "SBC
A,(HL)", "SBC      A,A"
230 DATA"AND      B", "AND      C", "AND      D", "AND
E", "AND      H", "AND      L", "AND      (HL)", "AND
A", "XOR      B", "XOR      C", "XOR      D", "XOR      E"
240 DATA"XOR      H", "XOR      L", "XOR      (HL)", "X
OR      A", "OR      B", "OR      C", "OR      D", "OR
E", "OR      H", "OR      L", "OR      (HL)", "OR
A"
250 DATA"CP      B", "CP      C", "CP      D", "CP
E", "CP      H", "CP      L", "CP      (HL)", "CP
A", "RET      NZ", "POP      BC"
260 DATA"JP      NZ,#", "JP      #", "CALL
NZ,#", "PUSH      BC", "ADD
A,! ", "RST      0", "RET      Z", "RET"
270 DATA"JP      Z,#", "Unknown", "CALL
Z,#", "CALL      #", "ADC      A,! ", "RST
8", "RET      NC", "POP      DE", "JP      NC,#"
280 DATA"OUT      (I),A", "CALL      NC,#", "P
USH      DE", "SUB      !", "RST      10H", "RE
T      C", "EXX", "JP      C,#"
290 DATA"IN      A,(I)", "CALL      C,#", "Un
known", "SBC      A,! ", "RST      18H", "RE
T      PO", "POP      HL", "JP      PO,#"
300 DATA"EX      (SP),HL", "CALL      PO,#", "P
USH      HL", "AND      !", "RST      20H", "RE
T      PE", "JP      (HL)", "JP      PE,#"
310 DATA"EX      DE,HL", "CALL      PE,#", "U
nknown", "XOR      !", "RST      28H", "RET
P", "POP      AF", "JP      P,#", "DI", "CALL      P,#"
320 DATA"PUSH      AF", "OR      !", "RST      30H", "RE
T      M", "LD      SP,HL", "JP      M,#", "EI
", "CALL      M,#", "Unknown", "CP      !", "RST
38H"
1000 FORX=0TO255:READA$(X):NEXT
1010 X=0:GOTO12500
1012 IFX>65535ORX<0THEN12520
1015 P=PEEK(X):PA$=CHR$(P)
1020 P$=A$(P):ML=X:GOSUB12000
1025 GOSUB1700
1030 P$=ML$+CHR$(9)+P$
1040 IFRIGHT$(P$,1)="!"THENGOSUB11000
1050 IFRIGHT$(P$,1)="#"THENGOSUB11500
1055 IFRIGHT$(P$,1)="@"THENGOSUB1500
1060 GOSUB1900:PRINTP$;TAB(25);PA$

```

```

1065 I$=INKEY$:IFI$="r"ORI$="R"THEN12520
1070 X=X+1:GOTO1012
1500 P$=LEFT$(P$,LEN(P$)-1):X=X+1:C=PEEK
(X):PA$=PA$+CHR$(C):IFC>127THENC=C-256
1520 ML=X+1+C:GOSUB12000:P$=P$+ML$:RETUR
N
1700 IFP<>34ANDP<>50THEN1720ELSEGOSUB180
0
1710 MID$(P$,5,4)=P2$+P1$:RETURN
1720 IFP=42THENGOSUB1800:MID$(P$,8,4)=P2
$+P1$:RETURN
1740 IFP=58THENGOSUB1800:MID$(P$,7,4)=P2
$+P1$:RETURN
1760 IFP=211THENX=X+1:P=PEEK(X):PA$=PA$+
CHR$(P):C=P:GOSUB10000:P1$=C$:MID$(P$,6,
2)=P1$:RETURN
1770 IFP=219THENX=X+1:P=PEEK(X):PA$=PA$+
CHR$(P):C=P:GOSUB10000:P1$=C$:MID$(P$,7,
2)=P1$:RETURN
1800 X=X+2:P1=PEEK(X-1):P2=PEEK(X):PA$=P
A$+CHR$(P1)+CHR$(P2):C=P1:GOSUB10000:P1$
=C$:C=P2:GOSUB10000:P2$=C$:RETURN
1900 FORR=1TOLEN(PA$):IFASC(MID$(PA$,R,1
))<32THENMID$(PA$,R,1)="."
1910 NEXTR:RETURN
10000 H$="0123456789ABCDEF"
10010 L=INT(C/16):R=C-L*16
10020 C$=MID$(H$,L+1,1)+MID$(H$,R+1,1)
10030 RETURN
11000 P$=LEFT$(P$,LEN(P$)-1)
11010 X=X+1:C=PEEK(X):PA$=PA$+CHR$(C):GO
SUB10000
11020 P$=P$+C$:RETURN
11500 P$=LEFT$(P$,LEN(P$)-1)
11510 X=X+2:C=PEEK(X):P2$=CHR$(C):GOSUB1
0000
11520 P$=P$+C$:C=PEEK(X-1):PA$=PA$+CHR$(
C)+P2$:GOSUB10000
11530 P$=P$+C$:RETURN
12000 MS=INT(ML/256):LS=ML-MS*256
12010 C=MS:GOSUB10000:ML$=C$
12020 C=LS:GOSUB10000:ML$=ML$+C$
12030 RETURN
12500 CLS
12520 INPUT"Start address";S
12550 X=S:GOTO1012

```

## Elapsed Time

by Richard Ramella

The Model 100 command PRINT TIMES displays the current time in 24-hour, military format, but the time is embedded in a data string and can't be easily substituted in mathematical equations. Elapsed Time (Program Listing 4) solves this problem by letting you use the enter key to mark start and end times of a program and telling you how much time passes between the two.

I used Elapsed Time as a subroutine in programs. It's helpful when I'm trying to pare the running time of a new program, but it can time any event that lasts fewer than 24 hours. The program has various applications, but since the machine

only states times to the second, it isn't useful for timing events that require split-second accuracy.

The program calculates the hours, minutes, and seconds of start and end times, stores them in separate number variables, and then subtracts the start time from the end time. Remember that the decimal system is not useful when working in 60-second, 60-minute, and 24-hour units.

### Running the Program

The program begins with this prompt: "To start, tap enter." Hitting the enter key sets the start time. Another prompt then appears: "Timing has begun. To end, tap enter." This second tap stops the program timer. Then the start time, the end time, and the elapsed time are given in this format: 2 hours 23 minutes 1 second.

### How It Works

Lines 150 and 200 set A\$(1) and A\$(2) as start and end times. The For . . . Next loop in lines 240-280 isolates the hour, minute, and second totals of the start and end times and assigns their variable values. This is done using the LEFT\$, MID\$, and RIGHT\$ string commands to isolate the correct two characters for hours, minutes, and seconds, then transforming these into numbers with the VAL command.

Here's an example of how this works:

```
10 Z$="4"
20 W=VAL(Z$)
30 PRINT W
```

Line 30 produces the number 4, which is subject to normal mathematical computation on the computer. This example won't work if, for instance, line 10 reads Z\$="four".

The reverse of this process is STR\$:

```
10 W=4
20 Z$=STR$(4)
30 PRINT Z$
```

Line 30 prints a 4 again, but it's a string 4 and subject to string variable rules.

Elapsed Time changes parts of a string into numbers. These numbers are dealt with in lines 330-380. I won't explain all these lines, but I'll translate one.

In effect, line 330 (in Listing 4) says: If the start- and end-seconds numbers are equal, zero seconds is the answer, so go to the next line. If the end number is higher than the start number, the number of elapsed seconds is the end number minus the start number. The only other possibility is that the start number is bigger than the end number, so the elapsed time number for seconds has to be 60 seconds minus the start number plus the end number. If this is the case, you must borrow the 60 seconds from the minute number.

The minute and hour figures are handled in roughly the same manner. Lines 360-370 print the elapsed time. If no hours or minutes have passed, the program doesn't print "0 hours 0 minutes." But, Elapsed Time does pay attention to plurals and adds an S to the words hour, minute, and second if needed.

In using this routine as a timer for a program in progress, I include program lines to set the value of A\$(1) as the timed routine begins and to set the value of A\$(2) when it is complete. Then I send the test to lines 240-380; the lines in Elapsed

Time are renumbered higher than the lines at the end of the program being timed, so no significant overlap exists. Once the elapsed time is stated, you can go back to the timed program to try to shorten its run time. ■

Write to Richard Ramella at 1493 Mt. View Ave., Chico, CA 95926.

```
100 REM * Elapsed Time * TRS-80 Model
100 * Richard Ramella *
110 CLS
130 PRINT "To start, tap Enter."
140 INPUT X
150 A$(1)=TIME$
160 CLS
170 PRINT "Timing has begun."
180 PRINT "To end, tap Enter."
190 INPUT X
200 A$(2)=TIME$
230 CLS
240 FOR A=1 TO 2
250 H(A)=VAL(LEFT$(A$(A),2))
260 M(A)=VAL(MID$(A$(A),4,2))
270 S(A)=VAL(RIGHT$(A$(A),2))
280 NEXT A
290 PRINT "Start: ";A$(1)
300 PRINT "Finish: ";A$(2)
310 PRINT STRING$(20,"-")
320 PRINT "Elapsed time..."
330 IF S(1)=S(2) THEN 340 ELSE IF S(2)>S
(1) THEN S(3)=S(2)-S(1) ELSE S(3)=(60-S
(1))+S(2):M(3)=-1
340 IF M(2)=M(1) THEN 350 ELSE IF M(2)>M
(1) THEN M(3)=M(2)-M(1) ELSE M(3)
=M(2)+(60-M(1))+M(1):H(3)=-1
350 IF H(2)=H(1) THEN 360 ELSE IF H(2)>H
(1) THEN H(3)=H(2)-H(1) ELSE H(3)
=H(2)+(24-H(1))+H(1)
360 IF H(3)>0 THEN PRINT H(3)"hour";:
IFH(3)>1 THEN PRINT "s ";
370 IF M(3)>0 THEN PRINT M(3)"minute";:
IF M(3)>1 THEN PRINT "s ";
380 IF S(3)>0 THEN PRINT S(3)"second";:
IF S(3)>1 THEN PRINT "s"
390 END
```

Program Listing 4. Elapsed Time.

## If It's Tuesday . . .

Conversion by Mare-Anne Jarvela  
80 Micro Technical Editor

You may have noticed that your corner liquor store stocks your favorite brands in half-liters instead of those familiar fifths and pints. And it's just as likely that the local service station now dispenses gas in liters rather than gallons. Despite these minor concessions to metric conversion, the United States, for the most part, stands apart in a world of kilometers, liters, and grams.

But if travel or business takes you abroad, or even across the border to Canada or Mexico, you may need to convert

U.S. measurements and currency into metric equivalents and foreign monies and back again. Conv.BA does all the hard work (see Program Listing 5).

## Measurement Conversions

After boot-up, the program displays the menu shown in Fig. 2. If you choose option 1 or 2, conversions between U.S. and metric measurements, a new menu appears and you choose the conversion you want to make (see Fig. 3). For example, you can convert inches to centimeters or liters to gallons.

After this, enter the number you want converted and the answer appears on the screen. You can then return to the main menu for another conversion.

## Currency Conversion

If you select Foreign Currency from the main menu, a submenu displays two options: to convert U.S. dollars into foreign currency or vice versa.

If you choose to convert U.S. dollars to a foreign currency, the program displays seven countries and their monetary denominations in parentheses (see Fig. 4). Select the country/currency of interest.

After you select the exchange, the program prompts you for

the current currency exchange rate. For instance, if you wanted to convert dollars into French currency, the program asks you the exchange rate for one franc. You'll find exchange rates in the financial section of your local newspaper or *The Wall Street Journal* under the heading "U.S. \$ Equivalents."

When you enter the exchange rate, the program prompts for the number of dollars you want converted. When the answer appears, you can return to the main menu.

If you want to convert foreign currency to U.S. dollars, you would choose option 2 from the submenu. When you choose the currency to which you want your dollar figure converted, the program prompts, "What is the current exchange rate for one U.S. dollar?" Again, this information is found in your newspaper's financial section, under "Currency per U.S. \$." Enter the appropriate number, then enter the number of dollars you want converted and your answer appears on the display.

If for any reason you want to change the calculations for figuring out the currency, alter lines 1040 and 1100.

## Bon Voyage

Have fun converting and don't leave home without your Model 100. Hopefully this program will help you the next time you buy gas by the liter at a German service station. ■

- ```

1. U.S. STANDARD->METRIC
2. METRIC->U.S. STANDARD
3. FOREIGN CURRENCY
ENTER 1, 2 OR 3?
    
```

Figure 2. Conv.BA's main menu.

- ```

1. FAHRENHEIT TO CELSIUS
2. INCHES TO CENTIMETERS
3. FEET TO METERS
4. MILES TO KILOMETERS
5. OUNCES TO GRAMS
6. POUNDS TO KILOGRAMS
7. GALLONS TO LITERS
ENTER NUMBER OF CONVERSION?
    
```

Figure 3. Menu to convert U.S. measurements to foreign measurements.

- ```

1. CANADA (DOLLAR)
2. BRITAIN (POUND)
3. FRANCE (FRANC)
4. MEXICO (PESE)
5. W. GERMANY (MARK)
6. SPAIN (PESETA)
7. JAPAN (YEN)
ENTER NUMBER OF CURRENCY?
    
```

Figure 4. Currency conversion options.

## Program Listing 5. Conv.BA.

```

10 REM CONVERSION
20 CLS:PRINT"1. U.S. STANDARD
";:PRINTCHR$(154);:PRINT" METRIC"
30 PRINT:PRINT"2. METRIC ";:PRINTCHR$(
154);:PRINT" U.S. STANDARD"
40 PRINT:PRINT"3. FOREIGN CURRENCY"
50 PRINT:INPUT"ENTER 1,2 OR 3";N
60 IF N=1 GOTO90
70 IF N=2 GOTO 210
80 IF N=3 GOTO890 ELSE 20
90 CLS:PRINT"1. FARENHEIT TO CELSIUS"
100 PRINT"2. INCHES TO CENTIMETERS"
110 PRINT"3. FEET TO METERS"
120 PRINT"4. MILES TO KILOMETERS"
130 PRINT"5. OUNCES TO GRAMS"
140 PRINT"6. POUNDS TO KILOGRAMS"
150 PRINT"7. GALLONS TO LITERS"
160 INPUT"ENTER NUMBER OF CONVERSION";R
170 ON R GOSUB
330,370,410,450,490,530,570
180 PRINT:INPUT"DO YOU WANT ANOTHER
CONVERSION (Y/N)";A$
190 IF A$="Y" THEN 20
200 IF A$="N" THEN 1210 ELSE 180
210 CLS:PRINT"1. CELSIUS TO FARENHEIT"
220 PRINT"2. CENTIMETERS TO INCHES"
230 PRINT"3. METERS TO FEET"
240 PRINT"4. KILOMETERS TO MILES"
250 PRINT"5. GRAMS TO OUNCES"
260 PRINT"6. KILOGRAMS TO POUNDS"
270 PRINT"7. LITERS TO GALLONS"
280 INPUT"ENTER NUMBER OF CONVERSION";S
290 ON S GOSUB
610,650,690,730,770,810,850
300 PRINT:INPUT"DO YOU WANT ANOTHER
CONVERSION (Y/N)";A$
    
```

Listing 5 continued

|   |                    |     |                       |
|---|--------------------|-----|-----------------------|
| T | Menu choice        | X   | Degrees Celsius       |
| N | Menu choice        | W   | Centimeters           |
| R | Menu choice        | E   | Meters                |
| S | Menu choice        | K   | Kilometers            |
| V | Menu choice        | G   | Grams                 |
| D | Degrees Fahrenheit | B   | Kilograms             |
| I | Inches             | L   | Liters                |
| F | Feet               | C   | Current exchange rate |
| M | Miles              | U   | How many units?       |
| O | Ounces             | A\$ | Y/N Input             |
| P | Pounds             | B\$ | Singular currency     |
| Q | Gallons            | C\$ | Plural currency       |

Table 2. Conv.BA variable list.

| Line      | Description                            |
|-----------|----------------------------------------|
| 10-80     | Menu (Main)                            |
| 90-200    | Menu (U.S. Standard to Metric)         |
| 210-330   | Menu (Metric to U.S. Standard)         |
| 340-600   | Calculations (U.S. Standard to Metric) |
| 610-880   | Calculations (Metric to U.S. Standard) |
| 890-910   | Menu (Main-Foreign Currency)           |
| 920-1010  | Menu (Choice of Currency)              |
| 1020-1070 | U.S. \$ to Foreign Currency            |
| 1080-1130 | Foreign Currency to U.S. \$            |
| 1140-1200 | String Values                          |
| 1210      | End                                    |

Table 3. Conv.BA line descriptions.

Listing 5 continued

```

310 IF A$="Y" THEN 20
320 IF A$="N" THEN 1210 ELSE 300
330 REM F TO C
340 CLS:INPUT"DEGREES FARENHEIT";D
350 PRINT:PRINT D;"DEGREES FARENHEIT
=";5/9*(D-32); " DEGREES CELSIUS"
360 RETURN
370 REM INCHES TO CENTIMETERS
380 CLS:INPUT"HOW MANY INCHES";I
390 PRINT:PRINT I;"INCHES =";I*
2.54;"CENTIMETERS"
400 RETURN
410 REM FEET TO METERS
420 CLS:INPUT"HOW MANY FEET";F
430 PRINT:PRINT F;"FEET =";F*
.3048;"METERS"
440 RETURN
450 REM MILES TO KILOMETERS
460 CLS:INPUT"HOW MANY MILES";M
470 PRINT:PRINT M;"MILES =";M*
1.609;"KILOMETERS"
480 RETURN
490 REM OUNCES TO GRAMS
500 CLS:INPUT"HOW MANY OUNCES";O
510 PRINT:PRINT O;"OUNCES =";O*
28.35;"GRAMS"
520 RETURN
530 REM POUNDS TO KILOGRAMS
540 CLS:INPUT"HOW MANY POUNDS";P
550 PRINT:PRINT P;"POUNDS =";P*
.45;"KILOGRAMS"
560 RETURN
570 REM GALLONS TO LITERS
580 CLS:INPUT"HOW MANY GALLONS";Q
590 PRINT:PRINT Q;"GALLONS =";Q*
3.8;"LITERS"
600 RETURN
610 REM C TO F
620 CLS:INPUT"HOW MANY DEGREES
CELSIUS";X
630 PRINT:PRINT X;"DEGREES CELSIUS =";9*X
/5+32;"DEGREES FARENHEIT"
640 RETURN
650 REM CENTIMETERS TO INCHES
660 CLS:INPUT"HOW MANY CENTIMETERS";W
670 PRINT:PRINT W;"CENTIMETERS =";W*
.39;"INCHES"
680 RETURN
690 REM METERS TO FEET
700 CLS:INPUT"HOW MANY METERS";E
710 PRINT:PRINT E;"METERS =";E*
3.28;"FEET"
720 RETURN
730 REM KILOMETERS TO MILES
740 CLS:INPUT"HOW MANY KILOMETERS";K
750 PRINT:PRINT K;"KILOMETERS =";K*
.62;"MILES"
760 RETURN
770 REM GRAMS TO OUNCES
780 CLS:INPUT"HOW MANY GRAMS";G
790 PRINT:PRINT G;"GRAMS =";G*
.035;"OUNCES"
800 RETURN
810 REM KILOGRAMS TO POUNDS
820 CLS:INPUT"HOW MANY KILOGRAMS";B
830 PRINT:PRINT B;"KILOGRAMS =";B*
2.2;"POUNDS"
840 RETURN
850 REM LITERS TO GALLONS
860 CLS:INPUT"HOW MANY LITERS";L
870 PRINT:PRINT L;"LITERS =";L*
0.264;"GALLONS"
880 RETURN
890 CLS:PRINT"1. U.S.$ ";:PRINT CHR$(154)
";:PRINT" FOREIGN CURRENCY":PRINT" (
U.S.$ EQUIVALENT )"
900 PRINT:PRINT"2. FOREIGN CURRENCY
";:PRINT CHR$(154);:PRINT" U.S.$":PRINT"
( CURRENCY PER U.S.$ )"
910 PRINT:INPUT"ENTER 1 OR 2";T
920 CLS:PRINT"1. CANADA (DOLLAR) "
930 PRINT"2. BRITAIN (POUND) "
940 PRINT"3. FRANCE (FRANC) "
950 PRINT"4. MEXICO (PESO) "
960 PRINT"5. W.GERMANY (MARK) "
970 PRINT"6. SPAIN (PESETA) "
980 PRINT"7. JAPAN (YEN) "
990 INPUT"ENTER NUMBER OF CURRENCY";V
1000 ON V GOSUB
1140,1150,1160,1170,1180,1190,1200
1010 IF T=2 THEN 1080
1020 CLS:PRINT"WHAT IS THE CURRENT

```

Listing 5 continued

Listing 5 continued

```

EXCHANGE RATE FOR 1
":PRINTB$;:INPUT"";C
1030 PRINT:INPUT"HOW MANY U.S.$";U
1040 PRINT:PRINTU;"U.S.$="";:PRINT
USING"#####.##";U/C;:PRINT" ";C$
1050 PRINT:INPUT"DO YOU WANT ANOTHER
CONVERSION (Y/N)";A$
1060 IF A$="Y" THEN20
1070 IF A$="N" THEN1210 ELSE 1050
1080 CLS:PRINT"*** ";C$;" ***":INPUT"WHAT
IS THE CURRENT EXCHANGE RATE FOR 1
U.S.$";C
1090 PRINT:PRINT"HOW MANY
":PRINTC$;:INPUT"";U
1100 PRINT:PRINTU;C$;"="";:PRINT USING
"#####.##";U/C;:PRINT" U.S.$"
1110 PRINT:INPUT"DO YOU WANT ANOTHER
CONVERSION (Y/N)";A$
1120 IF A$="Y" THEN 20
1130 IF A$="N" THEN1210 ELSE 1110
1140 B$="CANADIAN $":C$=B$:RETURN
1150 B$="POUND":C$="POUNDS":RETURN
1160 B$="FRANC":C$="FRANCS":RETURN
1170 B$="PESO":C$="PESOS":RETURN
1180 B$="MARK":C$="MARKS":RETURN
1190 B$="PESETA":C$="PESETAS":RETURN
1200 B$="YEN":C$=B$:RETURN
1210 CLS:END

```

## Still Up in the Air

by Eric Grevstad  
80 Micro News Editor

From the moment Radio Shack introduced the Model 100, people have considered the portable a suitable seatmate for plane trips. After three months on the market, the 100 has logged a lot of flying time, but still hasn't earned its official license.

Hundreds of owners use their micros, with and without asking permission, on commercial flights. A few ask for permission and are turned down, though, as far as anyone knows, no planes have suffered instrumentation interference or near-disaster as a result of airborne computing. Past that, users waiting for an authoritative yes or no on the 100's airworthiness are still waiting.

As reported in *80 Micro* (July 1983, p. 169), Federal Administration Aviation rule 91-19 outlaws all electric devices except hearing aids, pacemakers, tape recorders, and shavers—unless an airline chooses to allow them. From Model 100 owners' chats on CompuServe, there's no doubt that many airlines are tolerating the 100.

The bulletin board contains dozens of messages like E. Brad Meyer's: "Took Northwest Flight 287 from Boston to Chicago today. Asked permission to use the 100, received it, and used the machine intermittently throughout the flight, including during the final approach. No problems reported."

*InfoWorld* gave two pages of its June 20 issue to a rather giddy article by Bob Loudon of Palo Alto, CA, who played with his new Model 100 from San Francisco to New York and back: "Years ago, a friend told me that you aren't a real success in this country until the stewardesses always stop by to say 'Hello,'" Loudon wrote. "You can achieve the same effect with a Model 100, at least until everybody else has one."

Besides impressing flight attendants, Loudon pointed

out an advantage of CMOS RAM over magnetic media: "The Model 100 has now been X-rayed twice by the airlines, with no loss of memory," he wrote.

On his way home, Loudon used the portable's Basic and sound routines to write a program that played five octaves of music in whole, half, quarter, and eighth notes—presumably trying the patience of people sitting nearby, but apparently causing no harm to the plane's communication and navigation equipment.

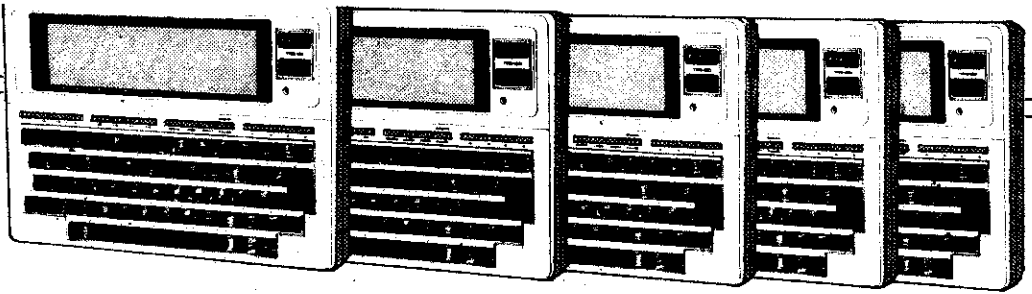
According to John Revelle of Rohnert Park, CA, the first rule of Model 100 air travel is to notify the flight crew of its use: "Although the rule is a federal one and the authority is supposed to be with the airline, the actual final word rests with the captain," wrote Revelle in a letter to *80 Micro*.

Meanwhile, Radio Shack is trying to end the uncertainty by collecting owners' reports and offering test 100s to airlines. Said director of computer merchandising Ed Juge, "I don't think there's been any official word yet. The only cases I've heard of someone not using the 100 or being stopped from using it were on USAir. There were a couple of incidents of the stewardess acting on her own, saying 'It's a rule that you can't have personal computers on the plane.'

"It doesn't look like there's a major problem at all," Juge said. "It's just a matter of history and of accumulating evidence to tell the airlines that people have used it, with the captain's permission, and there've been no problems.

"We're in contact with several airlines, all of whom have been offered units for testing; we've made some inroads with them and will possibly be contacting some more," Juge said of Radio Shack's requests for a definite all-clear. "We're trying to work within the existing law; we're not trying to go to the FAA and get a new law. And the existing law states that it's up to the airlines and the individual captains." ■

C-NOTE FOR OCTOBER 1983



## Mighty Write

by Bradford N. Dixon  
80 Micro Technical Editor

One of the Model 100's most useful firmware utilities is its word processing program. Unfortunately, however, it provides no formatting commands similar to those found in other word processing programs. PRNXTX.BA is a short Basic program that reads text stored in do-files and prints it according to a user-specified format (see Program Listing 1).

### Running the Program

When you run PRNXTX.BA, the first screen displays the programs resident in the Model 100's RAM. The program informs you that it formats text in do-files only.

Select the text file you want printed. Type in the file name and press the enter key. The File to Print prompt disappears and the program asks you to set the right and left margins. Then it asks whether you want the text single- or double-spaced. As soon as you answer, the program goes to work.

### Program Operation

Line 130 opens the file you specified and the program reads each character one by one. Lines 160-280 test for the presence of a character. Lines 160 and 170 check the position of the print head.

When the print head is at its starting point at the left margin and the next character is a space, the program reads the following character. But if the print head encounters a space within 10 characters of the right margin, the program automatically generates a carriage return and line feed to prevent a word break (line 240).

The program also checks to see if a character is a carriage return. A carriage return brings the print head to the start of the next line and inserts the appropriate number of line feeds. Otherwise, the program moves to line 310 to check the line counter, then returns to line 140 to read another character.

The line counter in this program is set to 57 single-spaced lines per page. If you want to use a different sized paper, modify lines 230 and 310. You could also make the line counter a user-defined variable at the start of the program.

When the line counter reaches 57 lines, printing stops, the screen clears, and a short menu appears. For multipage printouts, press the N key to start printing the next page. Lines

350-370 contain a loop that generates 10 line feeds to provide proper spacing at page breaks for fan fold and roll paper, or a top margin of 10 spaces for cut-sheet paper.

Press the enter key to close a previously opened file and send the program back to the first screen at line 40. The E key terminates the print formatting program.

### Enhancements

One advantage inherent to this program is its size. PRNXTX.BA occupies less than 1,400 bytes—small enough so that it doesn't restrict the size of text files.

With a bit more imagination, you could add options such as page numbering, page titles, and top/bottom margins. A modification that allows more than double spacing is also a possibility.

Nevertheless, this simple utility gives you an easy and versatile text formatter. ■

| Variable | Description                |
|----------|----------------------------|
| CR\$     | Carriage return            |
| F\$      | File name                  |
| LC       | Line counter               |
| LM       | Left margin                |
| RM       | Right margin               |
| S        | Spacing (single or double) |

Table 1. Variables list for PRNXTX.BA.

### Program Listing 1. PRNXTX.BA.

```

10 ' Text Formatting Program
20 ' by Bradford N. Dixon
30 ' 80 Micro Technical Editor
40 CR$=CHR$(13):SP$=" "
50 CLS:LC=1:FILES
60 PRINT@170,CHR$(27);"p";" Text files o
nly (.DO) ";CHR$(27);"q";
70 PRINT@240,"File to print: ";:INPUT F$
80 GOSUB 440:PRINT@240,"Left Margin: ";:
INPUT LM
90 GOSUB 440:PRINT@249,"Right Margin: ";
:INPUT RM
100 GOSUB 440:PRINT@244,"Single or Doubl
e Spacing (1/2): ";:INPUT S:S=S-1
110 GOSUB 450
120 LPRINTTAB(LM)
130 OPEN F$ FOR INPUT AS 1
140 IF EOF(1) THEN 420
150 A$=INPUT$(1,1)
160 IF LPOS(0)=LM AND A$=SP$ THEN 140
170 IF LPOS(0) < RM-10 THEN 240
180 IF A$<>SP$ THEN 240
190 LPRINT A$
200 LC=LC+1
210 GOSUB 450
    
```

Listing 1 continued

### Key Box

The programs in "Mighty Write," "Consulting the I Ching," and "Cram 100" will run in 8K RAM. The programs in "Quick On-Site Job Estimate" and "To Market, To Market" require 24K RAM.



Listing 1 continued

```

220 LPRINTTAB(LM)
230 IF LC<57 THEN 140 ELSE GOTO 340
240 LPRINT AS;
250 IF AS<>CR$ THEN 310
260 LC=LC+1
270 AS=INPUT$(1,1)
280 IFA$=SP$ THEN 270
290 GOSUB 450
300 LPRINTTAB(LM);
310 IF LC<57 THEN GOTO140
320 CLS:PRINT@50,"PRESS 'N' to print nex
t page":PRINT@90,"<ENTER> to print anoth
er file":PRINT@130,"'E' to end printing"
330 I$=INKEY$:IF I$="N" THEN 340 ELSE IF
I$=CHR$(13) THEN 430 ELSE IF I$="E" THE
N 400 ELSE 330
340 LC=1
350 FOR I=1 TO 10
360 LPRINT
370 NEXT I
380 LPRINTTAB(LM);
390 GOTO 140
400 CLOSE
410 END
420 CLOSE:GOTO 320
430 CLOSE:GOTO 40
440 PRINT@240,SPACE$(40);:RETURN
450 IF S=1 THEN LC=LC+1 ELSE RETURN
460 LPRINT:RETURN

```

## Consulting the *I Ching*

Conversion by Ameer Eisenberg  
80 Micro Technical Editor

People often seek guidance when they're faced with a tough decision. Some seek out the peace of a forest, others find solace in the rhythm of the sea. But some people need more than nature's solitude. I wrote a conversion of Anthony Scarpelli's *I Ching* program (80 Micro, April 1980, p. 123) so Model 100 owners can find spiritual guidance on land or sea.

### The *I Ching*

The ancient Chinese consulted the *I Ching*, or *Book of Changes*, as an oracle whose messages could align the people with the cosmic forces governing their lives. They believed their fate was the result of a balance between opposing life forces—the dark and the light, the negative and the positive, the receptive and the active, devolution and evolution—what the Chinese generally called yin and yang. The *I Ching* (pronounced Yee Jing) provided guidance in periods of change by revealing the relative influences of these opposing forces.

The Chinese defined 64 patterns of life change in the *I Ching*. In ancient times, they cast yarrow stalks and, from the random pattern the stalks created, developed a hexagram identifying one of those life patterns. By throwing the *I Ching*, the Chinese believed they could determine appropriate responses to present circumstances and future events.

More recently, three coins are used to throw the *I Ching*. The two sides of a coin represent the opposing forces. You throw the coins six times to generate a pattern that corre-

sponds to broken (yin) and solid (yang) lines called a hexagram (see Fig. 1).

Today, Westerners use the ancient book to forecast events and make decisions—not according to principles of cause and effect, but through the dynamics of chance and coincidence.

### Model 100 *I Ching*

The program I converted puts the *I Ching* at your fingertips. Type Program Listing 2 into your Model 100 and save it; I use the file name ICHING.BA. Run the program by positioning the cursor over the menu listing and pressing the enter key. The program doesn't accept lowercase input, so press your 100's caps lock key or remember to answer in uppercase letters.

The program first asks you if you want instructions. If you do, the instructions appear and the display automatically continues to the next screen until all the instructions have appeared.

After the instructions, the computer asks you if you're ready to begin. When you press the Y key, the computer prompts, "What is your question (enter)?" I find it easier to concentrate on my question if I've typed it. But if you don't want your question displayed, press the enter key and the screen goes blank.

You can throw the *I Ching* at any point thereafter. Take your time, concentrate on your question and, when you feel the time is right, press the space bar to simulate the throw of the coins. One of the nicest features of this program is that it lets you determine the moment of the "coin toss," rather than using a random function to throw the *I Ching*. Press the space bar six times, once for each line of the hexagram.

The computer highlights the hexagram lines that indicate change with an arrow. It posts the changed hexagram next to the original. Consult the *I Ching* to read the interpretations for your hexagram, the changing lines and the final hexagram.

Finally, the computer asks you if you wish to cast another *I*

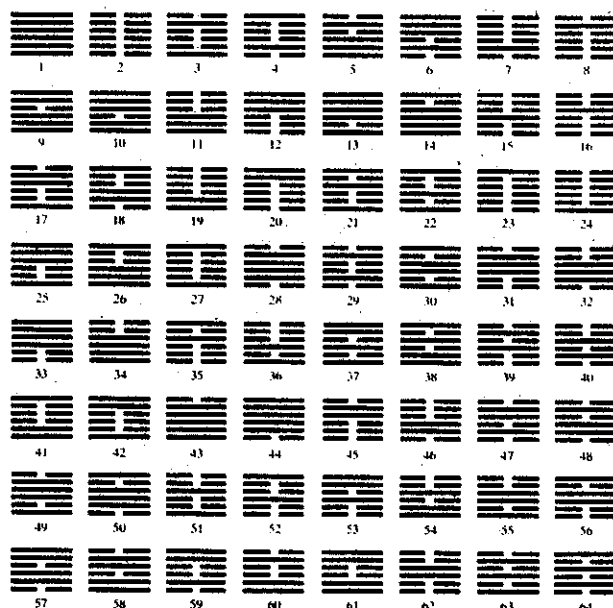


Figure 1. The King Wen sequence of hexagrams (devised in 1143 B.C.). In this arrangement the even numbered hexagram mirrors the previous odd numbered hexagram, reversing or inverting the pattern of broken and unbroken lines.

*Ching* hexagram. If you choose not to continue, press the break key to end the program.

### Program Operation

As you run ICHING.BA, the program repeatedly counts from six to nine until you press the space bar. Six is a changing yin (broken) line that resolves itself to seven, a yang (solid) line. Nine, a changing yang line, resolves itself to eight, a yin line. Wherever the computer is in its count when you press the space bar determines the line cast.

The program counts with a nested For...Next loop in lines 300-400. Line 300 maintains a count from one to six to keep track of which of the six hexagrams it's casting. Line 310 generates the repeating count from six to nine. Line 320 checks to see if you've pressed the space bar. If you have, the program reads line 350, which jumps out of the six to nine counting loop to record which line it generates. Line 370 prints the generated line and line 390 adjusts the video display to accom-

modate the new line. If the hexagram is not complete by line 400, the program goes back to line 300 to start the process again. If you haven't pressed the space bar, lines 330 and 340 send the computer back to continue counting from six to nine.

Variable C acts as a flag to mark whether you've thrown a changing line. If C equals 1 after the program casts the original hexagram, the computer continues to the routine in lines 410-500. These lines write the changed hexagram.

By listing the numbers of the 64 hexagrams in the order of their progress from all yin to all yang lines, it's possible to have the computer calculate which hexagram is cast. After arranging the data, you then use the addressing routine in lines 520-590 (similarly for the changed hexagram in lines 710-810).

Line 520 initializes the variables A and T for the addressing routine. Line 530 begins another loop that counts to six. If the hexagram line is a broken line (a yin), the program skips to line 550. For example, if the hexagram comprises all yin lines, you count only A\*2, or one. Six times through the loop still leaves T equal to 1. So in line 580, when the program counts from zero to T, it takes only one step.

The computer reads H, the first piece of data, and ends its loop. Line 610 prints H under the hexagram displayed on the screen. In the example presented here, the Model 100 displays a hexagram composed of all broken lines shown with the number 2 under it. And in fact, K'un or Natural Response is the second hexagram.

I've included a list of the hexagram names (see Fig. 2). As you become familiar with the *I Ching* and its hexagrams, knowing the name of the hexagram reminds you of its meaning. A simple program modification displays the hexagram name on the screen. Change the data statements in lines 620-670 to include the hexagram's name following its number. For example, DATA 2,Natural Response,...

Then, in lines 590 and 780, change READ H to READ H,H\$. This tells the computer to put the numeric data, 2, in H and the string data, Natural Response, in H\$. Modify PRINT H in lines 610 and 800 to PRINT H, H\$. *Voila!* The computer now names the hexagrams. ■

| Hexagram number | Hexagram name        | Hexagram number | Hexagram name         |
|-----------------|----------------------|-----------------|-----------------------|
| 1.              | Creative Power       | 33.             | Retreat               |
| 2.              | Natural Response     | 34.             | Great Power           |
| 3.              | Difficult Beginnings | 35.             | Progress              |
| 4.              | Inexperience         | 36.             | Censorship            |
| 5.              | Calculated Waiting   | 37.             | Family                |
| 6.              | Conflict             | 38.             | Contradiction         |
| 7.              | Collective Force     | 39.             | Obstacles             |
| 8.              | Unity                | 40.             | Liberation            |
| 9.              | Restrained           | 41.             | Decline               |
| 10.             | Conduct              | 42.             | Benefit               |
| 11.             | Prospering           | 43.             | Resolution            |
| 12.             | Stagnation           | 44.             | Temptation            |
| 13.             | Community            | 45.             | Assembling            |
| 14.             | Sovereignty          | 46.             | Advancement           |
| 15.             | Moderation           | 47.             | Adversity             |
| 16.             | Harmonize            | 48.             | The Source            |
| 17.             | Adapting             | 49.             | Changing              |
| 18.             | Repair               | 50.             | Cosmic Order          |
| 19.             | Promotion            | 51.             | Shocking              |
| 20.             | Contemplating        | 52.             | Meditation            |
| 21.             | Reform               | 53.             | Developing            |
| 22.             | Grace                | 54.             | Subordinate           |
| 23.             | Deterioration        | 55.             | Zenith                |
| 24.             | Returning            | 56.             | Traveling             |
| 25.             | Innocence            | 57.             | Penetrating Influence |
| 26.             | Potential Energy     | 58.             | Encouraging           |
| 27.             | Nourishing           | 59.             | Reuniting             |
| 28.             | Critical Mass        | 60.             | Limitations           |
| 29.             | Danger               | 61.             | Insight               |
| 30.             | Synergy              | 62.             | Conscientiousness     |
| 31.             | Attraction           | 63.             | After the End         |
| 32.             | Continuing           | 64.             | Before the End        |

For further reference:  
*I Ching*, Richard Wilhelm and Cary F. Baynes, Princeton University Press.  
*I Ching*, Raymond Van Over, Mentor Press.  
*Eight Lectures on the I Ching*, Hellmut Wilhelm, Princeton University Press.  
*The I Ching and You*, Diana Pfarington Hook, E.P. Dutton Publishers.  
*The I Ching Workbook*, R.L. Wing, Doubleday and Company Inc.

Figure 2. Hexagram names.

```

Program Listing 2. ICHING.BA

10 REM INSTRUCTIONS AND QUESTION
20 CLS:INPUT "Do you want instructions
(Y/N) ";Y$
30 IF Y$="Y"THEN100ELSEIFY$<>"N"THEN20
40 CLEAR:CLS:PRINT"what is your
question(enter)";:INPUTQ$
60 CLS:PRINTQ$
110 CLEAR 60:DEFINT Y,C,D,I,L,J,A,T,H
120 DIM C(6):DIMD(6):DIML$(9)
150 A$=CHR$(231):C$=CHR$(154):D$="
"
210 L$(6)=A$+A$+A$+D$+A$+A$+D$+C$
220 L$(7)=A$+A$+A$+A$+A$+A$+A$
230 L$(8)=A$+A$+A$+D$+A$+A$+A$
240 L$(9)=A$+A$+A$+A$+A$+A$+D$+C$
250 Y=240
260 YC=255
270 C=0
300 FOR I=1TO6

```

Listing 2 continued

Listing 2 continued

```

310 FOR J=6TO9
320 T$=INKEY$:IFT$=" "THEN350
330 NEXT J
340 GOTO 310
350 L=J
360 D(I)=L
370 PRINT@Y,I;" ";L$(L);
380 IF L=6 OR L=9 THENC=1
390 Y=Y-40
400 NEXT I
410 IF C=0THEN520
420 FOR I=1TO6
430 IF D(I)=6THENL=7:GOTO470
440 IF D(I)=9THENL=8:GOTO470
450 L=D(I)
470 C(I)=L
480 PRINT@YC,L$(L);" ";
490 YC=YC-40
500 NEXT I
520 A=1:T=0
530 FOR I=1TO6
540 IF D(I)=6ORD(I)=8THEN560
550 T=T+A
560 A=A*2
570 NEXT I
580 FOR I=0TOT
590 READ H
600 NEXT I
610 PRINT @286,H;:RESTORE:GOTO700
620 DATA 2,24,7,19,15,36,46, 11,16,51,40,54
630 DATA62,55,32,34,8,3,29,60,39,63,48,5
640 DATA 45,17,47,58,31,49, 28,43,23,27,4,41
650 DATA 52,22,18,26,35, 21,64,38,56,30,50,14
660 DATA20,42,59,61,53,37,57,9,12,25,6
670 DATA10,33,13,44,1
680 PRINT"Press space bar to do
another hexagram";
690 T$=INKEY$:IFT$=" "THEN 40 ELSE
690
700 IF C=0 THEN 680
710 A=1:T=0
720 FOR I=1TO6
730 IF C(I)=8 THEN750
740 T=T+A
750 A=A*2
760 NEXT I
770 FOR I=0TOT

```

```

780 READ H
790 NEXT I
800 PRINT@298,H
810 RESTORE
820 GOTO680
1000 CLS:PRINT:PRINT"The oracle
of the I Ching speaks to you of
the changing patterns and compelling
forces in the cosmos."
1010 FORX=1TO2500:NEXTX:CLS
1020 PRINT:PRINT"Concentrate on
your question.":PRINT"The computer
allows you to write down your
question, if you choose."
1030 FORX=1TO2500:NEXTX:PRINT"'Throw'
your hexagram when the question
is clear in your mind. "
1040 FORX=1TO2500:NEXTX
1050 CLS:PRINT:PRINT"when you
feel it's right, press the space
bar. This throws three symbolic
coins; a line appears."
1060 FORX=1TO2500:NEXTX
1070 CLS:PRINT:PRINT"Press the
space bar six times, once for each
line of the hexagram. There will
be no prompts, just a blank screen"
1080 FORX=1TO2500:NEXTX
1090 CLS:PRINT:PRINT"The six lines
of your hexagram compose the oracle's
answer. The hexagram's num-ber
appears underneath it."
1100 FORX=1TO2500:NEXTX
1110 CLS:PRINT:PRINT"Some of the
lines you throw will be changing.
These are marked with small arrows."
1120 PRINT"The changed hexagram
and its number are shown to the
right of the first."
1130 FORX=1TO2500:NEXTX
1140 CLS:PRINT:PRINT"Consult your
copy of the I Ching for insight
into the meaning of the hexagram..."
1150 FORX=1TO2500:NEXTX
1160 CLS:PRINT:INPUT "Are you ready
(Y)";Y$
1170 IFY$="Y"ORY$="y"THEN40ELSEGOTO1120

```

## Quick On-Site Job Estimate

by Beve Woodbury  
80 Micro Technical Editor

Carry your Model 100 instead of a clipboard for quick on-site job estimates. It's convenient to handle, looks impressive, and relieves some of the tedium of calculating job estimates based on material and labor costs.

### Setting up the Materials File

Program Listing 3 sets up the materials file. The record format begins with a part number that can be any number except zero. Next, enter a brief word description of the part and enter the cost (do not use commas).

The program loads the materials file into an array when it boots up, thus avoiding rewriting the file for each change. You add, delete, or change the cost of the materials from the main

menu. You can also print a listing of all items in the array. When you print the array, all additions and changes appear. Deletions are indicated by a zero in the part number column.

You can change the Parts.DO file (the materials listing) in the Text mode. Changing the file using this method may seem faster, but if you make the slightest error in the format, the estimating program won't function properly.

When the program begins, you're asked how many items you want to add. The program sets up an array with sufficient rows for the requested additions and fills the row with zeros. You can make fewer but no more additions than you requested.

When you add an item, the computer asks for the part name, description, and cost. You can assign a labor part number and a per-hour cost.

When you delete or change an item's cost, you can search by either part number (U) or by part name (A). If the part is not in the array, the program displays Part Not Found and returns to the menu.

| Variable | Description                                    |
|----------|------------------------------------------------|
| A ( )    | Numeric array for part number (1) and cost (2) |
| A\$( )   | String array for part description              |
| AD       | Number of parts to be added                    |
| C        | Cost of part                                   |
| CA       | Number of new parts input                      |
| D        | Length of arrays                               |
| D\$      | Description of part                            |
| DL\$     | Decision—delete or not?                        |
| H        | Loop counter                                   |
| IS       | Pause control                                  |
| K        | Loop counter                                   |
| L        | Line counter                                   |
| N        | Menu choice number                             |
| NA\$     | Name of part for search                        |
| NU       | Number of part for search                      |
| P        | Part number                                    |
| R        | Array row location for new part input          |
| S        | Array row location for file input              |
| S\$      | Decision—search by part or number              |
| X        | Array row location for file output             |
| Z        | Part not found flag                            |

Table 2. Parts set-up variables.

Program Listing 3. Parts set-up.

```

10 MAXFILES=1
20 F$="### \ \ ###.## ###"
30 OPEN"RAM:PARTS.DO" FOR INPUT AS 1
40 CLS:PRINT:PRINT" SETTING UP ARRAY"
50 IF EOF(1) THEN CLOSE:GOTO70
60 INPUT#1,P,D$,C:R=R+1:GOTO50
70 PRINT:INPUT" HOW MANY NEW PARTS WILL
YOU ADD ";AD:D=R+AD
80 OPEN"RAM:PARTS.DO" FOR INPUT AS 1
90 DIM A$(D),A(D,2)
100 IF EOF(1) THEN CLOSE:GOTO140
110 INPUT#1,P,D$,C:S=S+1
120 A$(S)=D$:A(S,1)=P:A(S,2)=C
130 GOTO100
140 NU=0:NA$=" ":Z=0:CLS:PRINT@18,"MENU"
150 PRINT@50,"1. CHANGE PART PRICE"
160 PRINT@90,"2. ADD NEW PART"
170 PRINT@130,"3. DELETE PART"
180 PRINT@170,"4. PRINT ARRAY"
190 PRINT@210,"5. QUIT"
200 PRINT:INPUT"ENTER CHOICE NUMBER:";N
210 IF N>5 THEN140
220 IF N=5 THEN CLOSE:GOTO520
230 CLS:ON N GOTO 240,270,330,380
240 GOSUB420:IF Z=1THEN140
250 INPUT"NEW COST ";A(K,2)
260 GOTO140
270 R=R+1:CA=CA+1
280 IF CA>AD THEN PRINT@88, "BEYOND ADD
LIMIT":FOR H=1TO500:NEXT:GOTO140
290 PRINT:INPUT"PART NUMBER: ";A(R,1)
300 PRINT:INPUT"DESCRIPTION: ";A$(R)
310 PRINT:INPUT"COST: ";A(R,2)
320 GOTO140
330 GOSUB420:IFZ=1THEN140

```

When the program finds the part, it prints the part number, description, and cost. If you choose the Change option, the program prompts you for the new cost. If you choose the Delete option, the program prompts you for a confirmation or cancellation. After each option, the program returns you to the menu.

Option 5, the Quit option, writes the file from the array and ends the program. If you don't use option 5, all additions, changes, and deletions made are lost.

### Getting Estimates

Program Listing 4 is the job estimate program. The program reads the parts file into an array. You are asked for a job title and a job description. A menu gives you the option of entering a part needed, printing an estimate, printing the parts array, performing a special calculation, or ending the program.

Call up a part either by the part number or name. If the program can't find the part, you are notified and returned to the menu.

Enter the number of the parts you need, and the computer prints the quantity, description, and the calculated cost of the part.

The Print Estimate and Print Parts Array listings pause when they fill the screen; press any key to continue.

The Print Estimate option prints a list of the quantity, description, unit cost, and total parts cost of each item you need to complete a job. It then prints a parts estimate and a final estimate.

The Print Parts Array prints all parts in your inventory.

The program adds an overhead/profit markup of 35 percent to the parts estimate. Change this margin by changing the .35 in line 370. Pressing any key returns you to the main menu.

```

340 INPUT"DELETE? Y/N ";DL$
350 IF DL$="N"THEN 140
360 IF DL$="Y"THEN A(K,1)=0:GOTO140
370 GOTO340
380 FOR H=1TOD
390 PRINTUSINGF$;A(H,1),A$(H),A(H,2)
400 L=L+1:IFL=7THEN L=0:INPUTIS
410 NEXT:INPUTIS:GOTO140
420 CLS:INPUT"SEARCH BY PART NUMBER, OR
NAME? U/A:";S$:PRINT
430 IF S$="U" THEN INPUT"PART NUMBER ";N
U:GOTO460
440 IF S$="A" THEN INPUT"PART NAME ";NA$
:GOTO460
450 GOTO420
460 FOR K=1TOD
470 IF NU=A(K,1)OR NA$=A$(K)THEN510
480 NEXT
490 PRINT@210,"PART NOT FOUND":Z=1
500 FORH=1TO500:NEXT:RETURN
510 PRINT:PRINT" "A(K,1)" "A$(K)" "A(
K,2):PRINT:RETURN
520 OPEN"RAM:PARTS.DO" FOR OUTPUT AS 1
530 CLS:PRINT" PRINTING NEW PARTS FILE"
540 FORX=1TOD
550 IF A(X,1)=0 THEN570
560 PRINT#1,A(X,1);", ";A$(X);", ";A(X,2)
570 NEXT:CLOSE:END

```

| Variable | Description                                             |
|----------|---------------------------------------------------------|
| A( )     | Numeric array for part number (1), cost (2), amount (3) |
| A\$( )   | String array for part description                       |
| AD       | Number of parts to be added                             |
| C        | Cost of part                                            |
| CR       | Calculation result                                      |
| C1       | First calculation input                                 |
| C2       | Second calculation input                                |
| C3       | Third calculation input                                 |
| D        | Length of arrays                                        |
| D\$      | Description of part                                     |
| E        | Estimate including markup                               |
| F\$      | Parts print format                                      |
| FF\$     | Job estimate file name                                  |
| G\$      | Estimate print format                                   |
| H        | Loop counter                                            |
| I\$      | Pause control                                           |
| J\$      | Job description                                         |
| K        | Loop counter                                            |
| L        | Line counter                                            |
| N        | Menu choice number                                      |
| NA\$     | Name of part for search                                 |
| NU       | Number of part for search                               |
| P        | Part number                                             |
| PE       | Total parts estimate                                    |
| PP       | Cost of total number of specific part wanted            |
| PX       | Calculated cost of specific part wanted                 |
| S        | Array row location                                      |
| T\$      | Job title                                               |
| WF\$     | Write file print format                                 |
| X        | Number of parts                                         |
| Z        | Part not found flag                                     |

Table 3. Job estimator variables.

Program Listing 4. Job estimator.

```

10 CLS:MAXFILES=3
20 F$="### \ \ ##.## ###"
30 G$="### \ \ ##.## #####"
40 WF$="### \ \ ##.## ### #
###.##"
50 PRINT:INPUT" ENTER JOB TITLE: ";T$
60 PRINT:PRINT" ENTER JOB DESCRIPTION:
":PRINT:INPUT J$
70 OPEN"RAM:PARTS.DO"FOR INPUT AS 1
80 CLS: PRINT "SETTING UP ARRAY"
90 IF EOF(1) THEN CLOSE:GOTO110
100 INPUT#1,P,D$,C:D=D+1:GOTO90
110 OPEN"RAM:PARTS.DO" FOR INPUT AS 1
120 DIM A$(D),A(D,3)
130 IF EOF(1) THEN CLOSE:GOTO170
140 INPUT#1,P,D$,C:S=S+1
150 A$(S)=D$:A(S,1)=P:A(S,2)=C
160 GOTO130
170 CLS:PRINT@18,"MENU"
180 PRINT@50,"1. ENTER PART"
190 PRINT@90,"2. PRINT ESTIMATE"
200 PRINT@130,"3. PRINT PARTS ARRAY"
210 PRINT@170,"4. DO CALCULATION"
220 PRINT@210,"5. WRITE JOB FILE/QUIT"
230 PRINT:INPUT"ENTER CHOICE NUMBER: ";N
240 CLS:ON N GOTO 250,380,500,540,800
250 NU=0:X=0:NA$=" "

```

Do Calculations lets you perform up to five calculations related to your estimates. I included calculations for square footage and cubic footage. Customize this part of the program area to suit your needs.

All calculation variables are set to zero (line 550) when the program prints the calculation menu in lines 540-600. Place your own function names here. Function 1 starts in line 640, function 2 in line 680, and so on (see line 630).

Use the Input command to enter commands to input the variables needed for the function in the first line (see lines 650 and 690). Put any comment that helps you understand the required input inside the quotes. If you use more variables than C1, C2, and C3, be sure to set them to zero in line 550.

The function formula goes in the next line. Use the variable name where you want the variable number put. Set up the formula in the format CR=(write in your own formula):GOTO 790. Line 790 prints the answer, waits for you to enter any key, and returns you to the main menu.

Quit writes the estimate file. H uses the first six letters of the job title and adds the .DO extension to the file name. This file contains the parts and quantity for a specific job, the parts' estimate, and the final estimate with the markup. Don't exit the program without the Quit option, or you'll lose all your data.

Read the estimate file by placing the cursor over the file name and pressing enter. Print a hard copy by using the Model 100's shift-print key function. ■

```

260 PRINT:INPUT" PART NUMBER ";NU
270 IF NU=0 THEN PRINT:INPUT"PART NAME "
;NA$
280 FOR K=1TOD
290 IF NU=A(K,1)OR NA$=A$(K) THEN330
300 NEXT
310 PRINT@210,"PART NOT FOUND":Z=1
320 FORH=1TO500:NEXT:GOTO170
330 PRINT:INPUT"NUMBER OF PARTS ";X
340 PX=X*A(K,2)
350 PRINT:PRINTUSINGF$;X,A$(K),A(K,2)
360 A(K,3)=A(K,3)+X
370 FOR H=1 TO 500:NEXT:GOTO170
380 PRINT:PRINTTAB(10);J$:PRINT:L=0:PP=0
:PE=0
390 FORK=1TOD
400 IFA(K,3)=0THEN 450
410 PP=A(K,2)*A(K,3)
420 L=L+1:IF L=8 THEN INPUTI$:L=0
430 PRINTUSINGG$;A(K,3),A$(K),A(K,2),PP
440 PE=PE+PP
450 NEXT:L=0
460 E=PE+(PE*.35) :E=(INT((E+.005)*100)
/100)
470 PRINT:PRINT" PARTS ESTIMATE IS ";PE
480 PRINT:PRINT" ESTIMATE IS ";E:I
NPUTI$
490 GOTO170
500 FOR H=1TOD
510 L=L+1:IFL=8THEN INPUTI$:L=0
520 PRINTUSINGF$;A(H,1),A$(H),A(H,2),A(H
,3)
530 NEXT:L=0:INPUTI$:GOTO170
540 CLS:PRINT@18,"MENU"
550 C1=0:C2=0:C3=0:CR=0
560 PRINT@50,"1. SQUARE FEET"

```

Listing 4 continued

Listing 4 continued

```

570 PRINT@90,"2. CUBIC FEET"
580 PRINT@130,"3. FUNCTION 3"
590 PRINT@170,"4. FUNCTION 4"
600 PRINT@210,"5. RETURN TO MAIN MENU"
610 PRINT:INPUT"ENTER CHOICE NUMBER:";N
620 IFN>5THEN540
630 CLS:ON N GOTO 640,680,720,760,170
640 CLS:PRINT:PRINTTAB(5)"SQUARE FOOTAGE
CALCULATION":PRINT
650 INPUT"LENGTH ";C1:INPUT"WIDTH ";C2
660 CR=C1*C2:PRINT:PRINT"SQUARE FOOTAGE
IS ";CR
670 GOTO790
680 CLS:PRINT:PRINTTAB(5)"CUBIC FOOTAGE
CALCULATION":PRINT
690 INPUT"LENGTH ";C1:INPUT"WIDTH ";C2:
INPUT"HEIGHT ";C3
700 CR=C1*C2*C3:PRINT:PRINT"CUBIC FOOTAG
E IS ";CR
710 GOTO790
720 'place TITLE for function 3 here
730 'place INPUT for function 3 here
740 'place FORMULA for function 3 here
750 GOTO790
760 'place TITLE for function 4 here
770 'place INPUT for function 4 here
780 'place FORMULA for function 4 here
790 INPUTI$:GOTO170
800 FF$="RAM:"+LEFT$(T$,6)+"DO"
810 OPEN FF$ FOR OUTPUT AS2
820 PRINT" PRINTING JOB ESTIMATE FILE"
830 PRINT#2,T$
840 PRINT#2,J$
850 FORK=1TOD
860 IF A(K,3)=0 THEN880
870 PRINT#2,USINGWF$;A(K,1),A$(K),A(K,2)
,A(K,3),A(K,2)*A(K,3)
880 NEXT
890 PRINT#2,"PARTS ESTIMATE = ";PE
900 PRINT#2,"TOTAL ESTIMATE = ";E
910 CLOSE:END
    
```

**Line Description**

|         |                                                  |
|---------|--------------------------------------------------|
| 10      | Open file channel                                |
| 20      | Print format                                     |
| 40-70   | Find array size for dimensioning                 |
| 80-130  | Dimension and set up array                       |
| 140-230 | Main menu print and choice                       |
| 240-260 | Change part price                                |
| 230     | GOSUB for search                                 |
| 250-260 | Input new cost and return to menu                |
| 270-320 | Add a new part                                   |
| 270     | Calculate row location and number of parts added |
| 280     | Check for beyond add limit                       |
| 290-320 | Input new part information                       |
| 330-370 | Delete part                                      |
| 380-410 | Print array                                      |
| 420-450 | Get search name or number                        |
| 460-500 | Search for part                                  |
| 510     | Print part information                           |
| 520-570 | Write parts file                                 |

Table 4. Parts set-up line descriptions.

**Line Description**

|         |                                             |
|---------|---------------------------------------------|
| 10      | Open file channels                          |
| 20-40   | Print formats                               |
| 50-60   | Title and description inputs                |
| 70-100  | Find array size for dimensioning            |
| 110-160 | Dimension and set up array                  |
| 170-240 | Main menu print and choice                  |
| 250-370 | Enter part wanted                           |
| 250-270 | Input part number or name                   |
| 280-320 | Search for part                             |
| 330-370 | Input number of parts wanted and print cost |
| 380-490 | Print estimate                              |
| 380     | Print heading                               |
| 390-450 | Print parts wanted and costs                |
| 460     | Calculate final estimate                    |
| 470-490 | Print estimate                              |
| 500-530 | Print parts array                           |
| 540-790 | Calculations                                |
| 540-630 | Calculation menu print and choice           |
| 640-670 | Square footage calculation                  |
| 680-710 | Cubic footage calculation                   |
| 720-750 | Third calculation                           |
| 760-780 | Fourth calculation                          |
| 790     | Pause and return to main menu               |
| 800-910 | Print job estimate file                     |

Table 5. Job estimator line descriptions.

**To Market, To Market**

Conversion by Marc-Anne Jarvela  
80 Micro Technical Editor

Most people consider food shopping a drudgery. It would be less bothersome if you had a master list of all the grocery items you usually buy and could quickly get a printout of a shopping list before you go to the store. This Model 100 conversion of Hal Smith's program (80 Micro, March 1981, p. 274) does just that and a little more (see Program Listing 5).

You need at least 16K of memory to run this program. The listing itself is only 5K but you need space for the data file.

When you run the program for the first time it asks you if you have a data file. Answer N and enter your first item. The program now creates the Shop.DO file. Line 20 sets maxfiles to two. After this, the program starts and you can add more items, delete, change, select items to buy, reset, flip pages, and get printouts of your shopping list.

If you already have a data file, answer Y on the first question or the computer writes over your existing file. When you finish changing, selecting, and so on, enter Q (quit) and your data file is automatically updated. If you break out of the program, you lose your changes.

With the Enter option (line 530) you can enter up to 250 items on your master list. Start the program, entering the grocery items you regularly buy. You don't have to reenter the list; pressing Q stores them on the data file. If there's not enough memory, change lines 50 and 60 to fit your computer.

The program stores all items alphabetically with code numbers. When you refer to an item on the list, use the code number assigned. The numbers change as you enter or delete items.

Program Listing 5. Shoplist.

```

10 REM SHOPLIST
20 MAXFILES=2
30 CLS:PRINT@53,"S H O P L I S T"
40 FORX=1TO1000:NEXT
50 CLEAR5000:DEFINTA-Z
60 DIM ML$(250),ML(250),TL$(50),TL(50)
70 R$=STRING$(26," ")
80 F1$="(###) ## "
90 FB$="          ## "
100 F2$="(###) "
110 F3$="      ## "
120 F4$="          ## "
130 CLS:GOSUB2040
140 CLS:PRINT@15,"SHOP LIST"
150 REM READ LIST FROM DISK
160 DATA C,D,E,L,P,S,Q,R
170 OPEN"RAM:SHOP.DO"FORINPUTAS1
180 N=1
190 J=1
200 INPUT #1,ML$(N),ML(N)
210 N=N+1
220 IF NOT EOF(1) THEN 200
230 I=N
240 CLOSE 1
250 START=1:REM ASSIGN 1 AS PAGE TO BE
PRINTED
260 COUNT=14:REM NUMBER OF LINES OF
ITEMS PRINTED (14 HERE)
270 REM MENU PRINT
280 V=0
290 CLS
300 G=INT((I-1)/14)
310 IFG<>((I-1)/14)THENG=INT((I-1)/14)+
1ELSEG=INT((I-1)/14)
320 J=J:IFJ<THENJ=1
330 IFJ>GTHENJ=G
340 PRINT"MASTER LIST:";I-1;"ITEMS
FOR";G;"PAGES,PAGE";J
350 PRINT"FORMAT:(CODE #) (QTY NEEDED) (
ITEM NAME)"
360 PRINT"PRESS ANY KEY TO CONTINUE"
370 IFINKEY$=""THEN370
380 GOSUB1780
390 PRINT:PRINT"PRESS ANY KEY TO
CONTINUE"
400 IFINKEY$=""THEN400
410 CLS:PRINT@200," <C>HANGE "," <D>
ELETE "," <E>NTER "," <L>IST ":PRINT "<P
>AGE",
420 PRINT"<R>ESET "," <S>ELECT "," <Q>
UIT"
430 FORY=1TO8:READB$(Y):NEXT
440 ONERRORGOTO460
450 GOTO480
460 RESUME 470
470 FORT=1TO20:NEXTT
480 PRINT"YOUR CHOICE: "
490 GOSUB 2020:D=1
500 IFB$(D)=A$THEN510:ELSED=D+1:IFD<
10THEN500ELSE490
510 ONDGOSUB 1060,1360,530,860,
720,650,1980,1170
520 GOTO270
530 REM ENTER
540 INPUT"NAME OF ITEM TO ADD TO LIST (0
TO ABORT)";N$
550 IFN$="0"THEN270
560 FOR N=1 TO I-1

```

Listing 5 continued

Use the Change option (line 1060) if you make a mistake entering an item. You provide the code number and enter the new name for that item. You also provide the code number of the item you want to delete (line 1360).

The Page command (line 720) lets you look at different pages (one page is equivalent to 14 items). At the top of the screen you'll see how many items your file contains and the number of the page displayed. To display a different page, hit P and then the page number, or N for next page, P for previous page. This lets you look at your entire file. Since the program stores items in alphabetical order, you'll have a general idea which page to look for.

To start your list for the grocery store, use the Select option (line 650). This lets you enter the quantity you need to buy. Hit S, enter the code number of the item, and then the quantity needed. When you print the list, the amount appears to the left of the item.

When it's time to go to the store, use the List (line 860) command. You have the option to print a full list or a short list. If you answer Y to the question "Is the printer ready?" you'll get a printout. Answer N and your list appears only on the screen.

The short list (line 1490) prompts you to enter the item and amount. It is only a temporary list—the program doesn't save it to the data file. This is a good option when you're in a hurry and want a quick list.

The Reset option (line 1170) lets you change the values to zero after you finish your shopping (answer Y on the first question), or reset the quantity needed for each item. Answer N and you'll see the first item on your shopping list. When all the items on your list are reset, the program returns to the main menu. Remember to hit Q so that all your changes appear on the data file.

If for any reason you want a listing of all the items in your data file, load Shop.DO from the Model 100's menu. It will appear on the screen, and you can print it as a text file.

This is a handy program to use if you have a lot of grocery shopping to do. If you're single, you'll probably be better off with a piece of paper and a pencil.

You can also use this program for other types of inventory control. Use your imagination, and good luck. ■

C-Notes text continued on p. 283

| Variable | Description               | Variable | Description          |
|----------|---------------------------|----------|----------------------|
| X        | Time delay.               | I        | Counter.             |
| ML\$     | Item (full list).         | J        | Counter.             |
| ML       | Code number (full list).  | Y        | Menu choice.         |
| TL\$     | Item (short list).        | T        | Time delay.          |
| TL       | Code number (short list). | D        | Menu choice.         |
| R\$      | String variable.          | A\$      | INKEY.               |
| F1\$     | Print format.             | N\$      | Item to add.         |
| FB\$     | Print format.             | C        | Code number of item. |
| F2\$     | Print format.             | PA\$     | What page ?          |
| F3\$     | Print format.             | K        | Counter.             |
| F4\$     | Print format.             | P        | Print ?              |
| N        | Counter.                  | Q        | Change.              |
| J        | Counter.                  | H        | Page Counter.        |
| V        | Counter.                  | E        | Counter.             |
| G        | Counter.                  |          |                      |

Table 6. Shopping list variables.



Listing 5 continued

```

570 IF N$>ML$(N) THEN NEXT N
580 FOR N1=I TO N+1 STEP -1
590 ML$(N1)=ML$(N1-1)
600 ML(N1)=ML(N1-1)
610 NEXT N1
620 ML$(N)=N$: ML(N)=0
630 I=I+1
640 RETURN
650 REM SELECT
660 INPUT"CODE NUMBER OF ITEM TO BE
BOUGHT (0 TO ABORT)";C
670 IF C>=I OR C<0 THEN PRINT:GOTO660
680 IFC=0THEN270
690 PRINT:PRINT"WHAT QUANTITY OF ";ML$(C
);
700 INPUT ML(C)
710 RETURN
720 REM PAGE COMMAND
730 INPUT"WHAT PAGE # (ENTER N FOR NEXT,
P FOR PREVIOUS, OR PG #) ";PA$
740 IFPA$="P" THENV=1:J=J-1:GOTO760
750 IFPA$="N" THENJ=J+1:GOTO790
760 IFV=1THEN START=START-1*
COUNT:GOTO810
770 J=VAL(PA$)
780 START=(VAL(PA$)-1)*COUNT+1:GOTO810
790 IF(START+1)+COUNT>I THEN RETURN
800 START=START+COUNT
810 IF VAL(PA$)>GTHENPRINT:GOTO730
820 IF I-COUNT<1 THEN RETURN
    
```

```

830 IF START <1 THEN START=1
840 IF START>I-COUNT THEN START=START
850 RETURN
860 REM PRINT LIST TO PRINTER
870 A$="":CLS:PRINT"DO YOU WANT THE FULL
LIST OR A SHORT ONE (F OR S) ? ";
880 A$=INKEY$:IF A$="" THEN880
890 IF A$="S" THENPRINTA$:FORZ=1TO250:
NEXT:GOTO1490
900 IF A$="F" THENPRINTA$:FORZ=1TO250:
NEXT:GOTO920
910 GOTO880
920 PRINT:INPUT"IS THE PRINTER READY (Y/
N)";A$
950 CLS:PRINTTAB(15);"SHOPPING LIST"
960 IF A$="Y"THEN LPRINT TAB(32)
;"SHOPPING LIST"
970 PRINT:IF A$="Y"THENFORK=1TO3:LPRINT"
":NEXTK
980 FOR N=1 TO I-1
990 IF ML(N)=0 THEN 1020
1000 PRINT:PRINTUSING F1$;N,ML(N)
;:PRINTLEFT$(ML$(N)+R$,15);
1005 IF A$="N"THEN1020
1010 LPRINT:LPRINTUSING FB$;ML(N)
;:LPRINTLEFT$(ML$(N)+R$,15);
1020 NEXT N
1030 IF A$="Y"THENLPRINTLEFT$(ML$(I-1))
1040 IFA$="N"THENPRINT:PRINT:INPUT"HIT <
ENTER> TO CONTINUE ";AN$
    
```

Listing 5 continued

## the GOLD kit

New Life for your TRS 80 Model I

- Memory Faults?
- Programs Freezing Up?
- Worn Contacts?

The Model I is a good computer, but it is getting old. Your expensive gold connectors have to transmit data through the Mod I's worn and oxidized PC board contacts. With the GOLD KIT and fifteen minutes you can upgrade these old tin contacts to gold.



These new connectors are soldered onto the tin plated edge contacts of your computer and expansion interface. You need just a screwdriver and soldering iron.

The kit includes all six connectors and detailed instructions.

Send \$29.50 Check or Money Order to:  
\*ADD \$1.50 for Shipping

**i/o tech** 198  
P.O. Box 2167  
2445 N. 50th Street

Milwaukee, WI 53201-2167

Wisconsin Residents add 5% sales tax.

# TRS - 80 MODEL 100

TRS 80 is a trademark of the Tandy Corporation

Monthly magazine loaded with Software and Hardware Reviews, Useful How-to Tips, Program Listings, and Accessory Interface Suggestions for this unique new computer. Whatever your interest in this notebook size portable, this exclusive publication will help you realize its maximum potential while saving you money. Regular subscription rate is \$24/year (12 issues), however charter subscriptions are available for a limited time at \$9.00 for 6 months or \$16.00 for 12 months.



customers call 800-227-1617 ext 425

In California 800-772-3545 ext 425 or send check or Money Order to:

## briefcase portable

560 South Hartz Avenue, Suite 447 • Danville, CA 94526 -165

SOMETHING NEW YOU'VE WAITED FOR THIS BETTER THAN A FORM LETTER  
IT DOES WHAT THE OTHERS DO AND MUCH MORE  
THERE IS TIME TO DO IT BEFORE CHRISTMAS!

Type in all the names, addresses, and personal information. Then you will tell your news to each friend: but not in a form letter! Enter your common interest codes to sort out which topics go to which friends. Then write on each topic to correspond with the codes. It's easy if you have

LET'R'WRYT

Our program lets you individualize your letters by printing on each letter only the presorted paragraphs corresponding to your pre-coded common interests. This way a unique letter can be tailored for each of your friends.

- \* menu driven-no study since you are trained by prompts on screen \*
- \* live keyboard \*
- \* address file with update/replace features \*
- \* topic file (paragraphs) with update features \*
- \* dozens of free options with prompting instructions: format, greeting, closing, nickname, search, sort, etc. \*
- \* no dead ends - errors lead to user friendly menu, not termination \*
- \* continuous file update - almost impossible to lose a file \*

600 names per disk on models 1,3,4

PRICE \$35.99 till NOV 1 then \$41.99 (our breakthrough price is on the rise!)  
To order call 313-683-5094 (MC,VISA) or send your check or M.O. to:  
Meyer's Place, 5132 Kitson Ln, West Bloomfield MI, 48033

-215

Listing 5 continued

```

1050 RETURN
1060 REM CHANGE COMMAND
1070 INPUT"CODE NUMBER OF ITEM TO CHANGE
(0 TO ABORT)";C
1080 IF C>=I OR C<0 THEN PRINT:GOTO1070
1090 IFC=0THEN270
1100 PRINT:PRINT"CHANGE ";MLS(C);" TO
WHAT ";
1110 INPUT N$
1120 Q=ML(C)
1130 GOSUB 1400: REM DELETE PREVIOUS
ENTRY
1140 GOSUB 560: REM ENTER NEW ENTRY
1150 ML(N)=Q
1160 RETURN
1170 REM RESET COMMAND
1180 INPUT"ARE ALL ITEMS TO BE RESET";AS
1190 IF A$="N"GOTO1240
1195 IF A$<>"Y"THEN1180
1200 FOR N=1 TO I-1
1210 ML(N)=0
1220 NEXT N
1230 RETURN
1240 PRINT
1250 CLS:PRINT"HERE IS A LIST OF THE
ITEMS YOU WERE TO BUY."
1260 PRINT"FOR EACH ITEM, HIT ENTER IF
IT WAS PURCHASED,"
1270 PRINT"OR ENTER THE QUANTITY

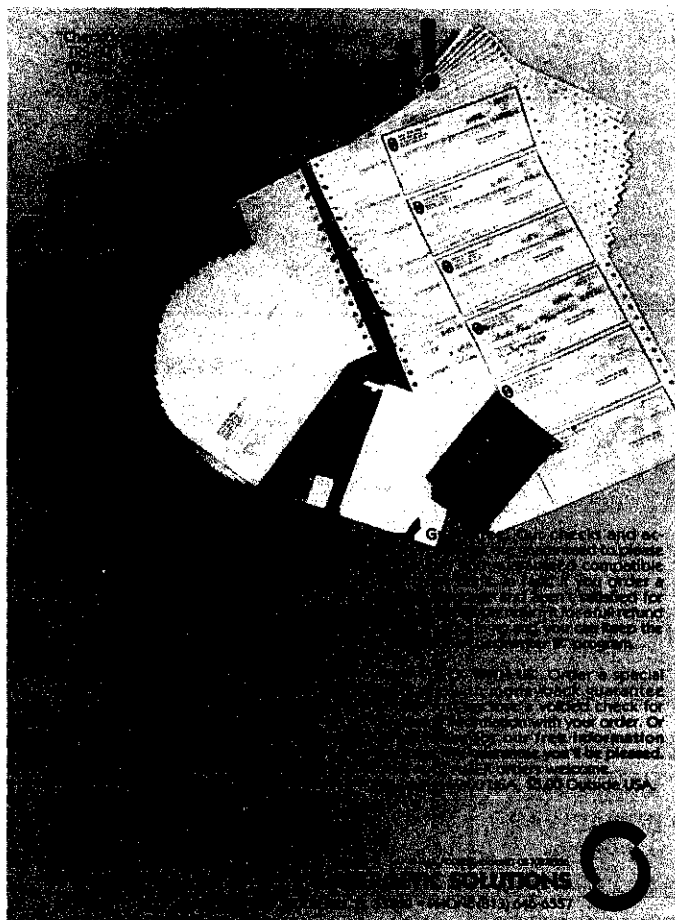
```

```

REMAINING TO BUY."
1280 FOR N=1 TO I-1
1290 IF ML(N)=0 THEN 1340
1300 PRINT:PRINT@200, ML(N);" ";MLS(N);
1310 ML(N)=0
1320 INPUT ML(N)
1330 PRINT CHR$(13)
1340 NEXT N
1350 CLS:PRINT"THERE ARE NO MORE ITEMS
TO BE RESET":FORX=1TO1500:NEXT:RETURN
1360 REM DELETE
1370 INPUT"CODE NUMBER OF ITEM TO DELETE
(0 TO ABORT)";C
1380 IF C>=I OR C<0 THEN PRINT:GOTO1370
1390 IFC=0THENGOTO270
1400 FOR N=C+1 TO I-1
1410 ML$(N-1)=ML$(N)
1420 ML(N-1)=ML(N)
1430 NEXT N
1440 I=I-1
1450 H=INT((I-1)/14)
1460 IFH<>((I-1)/14)THEN RETURN
1470 START=START-14
1480 RETURN
1490 REM TEMPORARY LIST
1500 TI=1
1510 COUNT=14:REM NUMBER OF ITEMS TO
PRINT
1520 CLS

```

Listing 5 continued



# MODEL 100

We're **ALPHAWARE**, and we think computers should be user-friendly, always. That's why, no matter where you are in **MicroEditor II** or **Mail Master**, you have immediate access to **GUARDIAN™**, a full-function help feature ready to assist at any time. Finally, programming that puts you in control: **MicroEditor II** and **Mail Master** by **ALPHAWARE**.

Dealer Inquiries Invited

## MicroEditor II

**\$49.95**

- Pagination
- Right, left, alternate margins
- Control of page width & length
- Line and page numbering
- Clear memory
- Center, justify
- Headings
- File handling
- Form letters
- Tabs

## Mail Master

**\$34.95**

- Cross reference
- Alphabetize
- Control sorting
- Mailing labels
- MicroEditor II compatible

**ALPHAWARE**  
INCORPORATED

2100 South Boulevard • Edmond, OK 73034

Listing 5 continued

```

1530 PRINT"TYPE IN ANY SPECIAL ITEMS
THAT YOU WISH TO BUY."
1540 GOSUB1700
1550 PRINT:PRINT"IS THE PRINTER READY?";
1560 GOSUB1920
1570 P=ANSWER
1580 CLS:PRINTTAB(15);"QUICK LIST"
1590 LPRINTTAB(32);"QUICK LIST"
1600 PRINT:FORZ=1TO3:LPRINT" ";NEXTE
1610 FOR N=1 TO TI-1
1620 PRINT:PRINTUSING F3$;TL(N)
;:PRINTLEFT$(TL$(N)+R$,15);
1630 IF P THEN LPRINT:LPRINTUSING F4$;TL
(N);:LPRINTLEFT$(TL$(N)+R$,15);
1640 NEXT N
1650 LPRINTLEFT$(TL(N)+R$,15)
1660 COUNT=14
1670 PA$="1"
1680 GOSUB 750
1690 RETURN
1700 PRINT:PRINT"HIT <ENTER> TO STOP"
1710 PRINTUSING"NAME OF ITEM ##";TI;
1720 INPUT TL$(TI)
1730 IF LEN(TL$(TI))=0 THEN RETURN
1740 PRINT"WHAT QUANTITY OF ";TL$(TI);
1750 INPUT TL(TI)
1760 TI=TI+1
1770 PRINT:GOTO1710
1780 REM PRINT LIST
1790 FOR N=START TO START+COUNT-1
1800 IF ML(N)=0 THEN 1820
1810 PRINT:PRINTUSING F1$;N,ML(N)
;:PRINTLEFT$(ML$(N)+R$,15);:GOTO1830
1820 PRINT:PRINTUSING F2$;N;:PRINTLEFT$(
ML$(N)+R$,15);
1830 IF N<I-1 THEN NEXT N
1840 RETURN
1850 REM UPDATE DATA FILE
1860 OPEN"RAM:SHOP.DO"FOROUTPUTAS1
1870 FOR N=1 TO I-1
1880 PRINT#1,CHR$(34);ML$(N);CHR$(34)

```

```

;";ML(N)
1890 NEXT N
1900 CLOSE 1
1910 RETURN
1920 REM YES/NO ANSWER
1930 ANSWER=1
1940 AN$=INKEY$:IFAN$=""THEN1940
1950 IF AN$="Y" THEN ANSWER=-
1:PRINTAN$:FORZ=1TO250:NEXT:RETURN
1960 IF AN$="N" THEN
ANSWER=0:PRINTAN$:FORZ=1TO250
NEXT:RETURN
1970 GOTO1940
1980 REM QUIT
1990 GOSUB 1850
2000 END
2010 REM DATA FILE CREATION
2020 A$=INKEY$:IFA$=""THEN2020:RETURN
2030 RETURN
2040 A$="1":PRINT"DO YOU ALREADY HAVE A
DATA FILE CREATED?";
2050 A$=INKEY$:IFA$=""THEN2050 ELSE
PRINTA$:FOR Z=1 TO 250:NEXT
2060 IF A$="Y" THEN RETURN
2070 IF A$="N" THEN 2090
2080 GOTO 2050
2090 CLS:PRINT"NOTE: THIS SECTION WILL
CREATE YOUR SHOP LIST DATA FOR ONE
ITEM."
2100 PRINT"AFTER YOU PLACE IN THE FIRST
ITEM, THE PROGRAM WILL THEN START
RUNNING"
2110 PRINT
2120 INPUT"NAME OF AN ITEM TO PLACE ON
YOUR SHOPPING LIST ";N$
2130 ML$(N)=N$:ML(N)=0
2140 I=I
2150 OPEN"RAM:SHOP.DO"FOROUTPUTAS1
2160 N=0
2170 PRINT#1,CHR$(34);ML$(N);CHR$(34)
;";ML(N)

```


Listing 5 continued

**SCRIBE**  
REAL  
WORD PROCESSING  
CAPABILITY  
for the  
Radio Shack TRS 80  
**MODEL 100**  
Portable Computer  
USES ONLY 2K RAM STORAGE  
**\$24.95**  
PLUS \$2.00 HANDLING  
Chattanooga Systems Associates  
P.O. Box 22261  
Chattanooga, TN 37422  
**DEALERS INQUIRE** ✓225

TRS-80 MODEL 16  
**XENIX**

- CONSULTING
- SYSTEM CONFIGURATION
- INSTRUCTION

**INTERACTIVE  
SYSTEMS DESIGN**  
SUITE 201, 170 BROADWAY  
NEW YORK, N.Y. 10038  
(212)-233-6923 ✓147

 **CHURCH DONATIONS** ✓307

Computer Program for TRS 80  
Models I, II, III, IV, 12 & 16  
IBM/PC

**RECORDS, STORES & REPORTS**  
Gives printout of collections, totals to date as well as  
individual quarterly reports to congregation members

**•NOW AVAILABLE•**  
Additional modules to provide:  
Church Directory  
Word Processing  
Attendance Record  
Membership Profile  
General Ledger

**Radio Shack ★★★★★  
Five-Star Rating**  
Designed to fulfill administrative & financial reporting  
and record keeping needs of churches

CUSTOM DATA 301 Eleventh Street  
Dept. AC P.O. Box 1869  
505-434-1096 Alamogordo, NM 88310

Listing 5 continued

```
2180 CLOSE
2190 RETURN
```

C:Notes text continued from p. 278

## Cram 100

### Conversion by Bradford N. Dixon 80 Micro Technical Editor

Since Cram first appeared in *80 Micro* (August 1982, p. 234) it has been the object of much gaming interest and at least one conversion (Take II, *80 Micro*, August 1983, p. 320). This Model 100 version of Cram uses the portable's graphics and sound capabilities to make for an exciting game that you take wherever you go.

Cram is easy to learn and addictive. The object of this version is to draw a line around the display without running into the screen edge or a previously drawn line.

The line starts in the upper left corner of the screen and runs to the right. Press any key and the line turns down. Press a key again and the line turns left, press a key again to go up, and so on.

The 100 emits a tone whenever you press a key and again when the game ends. During play, the program displays the previous high score and the last score at the top of the screen.

Program housekeeping takes place in lines 30 and 40 of Program Listing 6. The code that moves the line across the screen follows.

A counter keeps track of the line position and acts as a check for collisions with the sides or another part of the line. To get the most out of the program, I used PSET to light the pixels in the line. I could have used characters as in other versions of the game, but I was limited by the 100's screen size.

The Model 100, like the Model II, doesn't have a POINT (X,Y) function to check for lit pixels, so the counters keep track of the line while allowing full use of the 240- by 64-pixel matrix available on the 100's LCD screen. An INKEY\$ routine initiates the line turns and facilitates the game's fast pace.

Program instructions are located in a subroutine at the end of Listing 6. You can bypass the directions after you know how to play. ■

#### Program Listing 6. Cram 100.

```
10 ' CRAM, A game by Hardin Brothers
20 ' ** Conversion by Brad Dixon **
25 ' ** 80 Micro Technical Staff **
30 DEFINT A-Z:GOSUB 270
35 CLS:PRINT@14,"*** CRAM ***"
40 L=0:R=239:T=7:B=64:N=0:GOSUB 250
50 ' MOVE RIGHT
60 J=J+1:IF J=R THEN 200 ELSE PSET(J,I)
70 IF INKEY$="" THEN 60 ELSE
N=N+1:R=J:SOUND 2216,2
80 ' MOVE DOWN
90 I=I+1:IF I=B THEN 200 ELSE PSET(J,I)
100 IF INKEY$="" THEN 90 ELSE
N=N+1:B=I:SOUND 2216,2
110 ' MOVE LEFT
120 J=J-1:IF J=L THEN 200 ELSE PSET(J,I)
```

```
130 IF INKEY$="" THEN 120 ELSE
N=N+1:L=J:SOUND 2216,2
140 ' MOVE UP
150 I=I-1:IFI=T THEN 200 ELSE PSET(J,I)
160 IF INKEY$="" THEN 150 ELSE
N=N+1:T=I:SOUND 2216,2:GOTO 60
190 ' GAME OVER
200 FOR X=1 TO 5:BEEP:NEXT X:CLS:
PRINT@5,"TURNS=";N;" PREVIOUS
HIGH=";M:IF N>M THEN M=N
210 FOR I=1 TO 2000:NEXT I:GOTO 40
250 I=7:J=0:A$=INKEY$:RETURN
260 ' DIRECTIONS
270 CLS:PRINT@134,"*** CRAM
***":PRINT:PRINT:INPUT"DO YOU WANT
INSTRUCTIONS (Y/N)";Y$
280 IF Y$="Y" OR Y$="y" THEN 290 ELSE 35
290 CLS:PRINT:PRINT TAB(5)"THE OBJECT OF
THE GAME IS TO SPIRAL THE LINE AROUND
THE SCREEN WITHOUT IT RUNNING INTO
THE EDGES OR ITSELF. ":PRINT"PRESS
ENTER TO CONTINUE.":GOSUB 320
300 CLS:PRINT:PRINT TAB(5)"PRESS ANY KEY
TO CHANGE THE DIRECTION OF THE
LINE. THE GAME IS OVER WHEN YOU HIT THE
SIDES OR THE LINE."
310 PRINT TAB(5)"TO GET OUT OF THE GAME,
PRESS THE 'BREAK' KEY. WHEN YOU'RE
READY TO PLAY, PRESS 'ENTER'."
320 A$=INKEY$:IF A$="" THEN 320 ELSE RETURN
```

## CONVERT MOD III BASIC TO MOD 4

- Good News!** The Model 4 is twice as fast as the Model III.
- Bad News!** Model III disk BASIC programs only run at half speed on the Model 4.
- Best News!** "CONV3TO4" will automate many of the changes required to convert Model III BASIC programs to Model 4 BASIC. Your Model III BASIC programs will run twice as fast as they do now, and your manual conversion efforts will be vastly reduced.

In minutes this powerful utility can insert all required spaces, recalculate PRINT@ addresses, adjust TAB ( ) addresses, insert correct exponentiation symbols, and flag and list unresolved line numbers.

Other options enable your programs to run even faster - remove REM's, comments, down arrows, and unnecessary spaces.

Or, format your programs and make them easier to read and debug - insert down arrows and indent between multiple instructions. IF, THEN, and ELSE statements.

I agree, it's foolish to run my Model III disk BASIC programs at half speed on the Model 4. Send me "CONV3TO4" today! Enclosed is my check/money order for \$49.95 plus \$2.00 for shipping/handling (NJ res. add 6% sales tax).

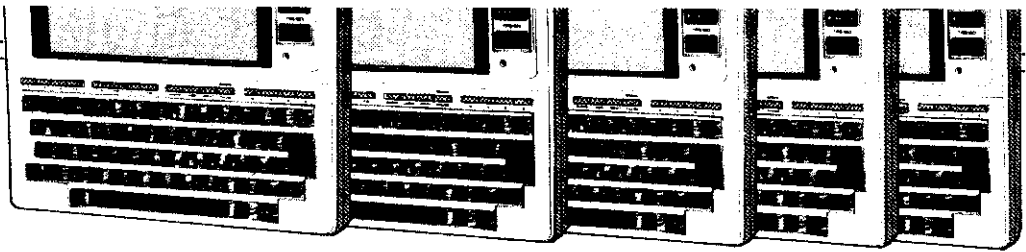
NAME: \_\_\_\_\_

STREET: \_\_\_\_\_

CITY: \_\_\_\_\_ STATE \_\_\_\_\_ ZIP \_\_\_\_\_

**EDUCATIONAL MICRO SYSTEMS, INC.**  
P.O. Box 471, Chester, New Jersey 07930  
201-879-5982

C-NOTE FOR NOVEMBER 1983



## Nag Analysis

Conversion by Mare-Anne Jarvela  
80 Micro Technical Editor

If you've got an eye for the horses, bring your Model 100 to the race track and make big bucks.

This conversion (Program Listing 1) of Dave Crosby's program Nag Analysis (80 Micro, July 1981, p. 243) rates horses by a handicapping system. My conversion runs in 8K RAM.

You enter information about the horse: days between races, race distance, class, gain in stretch, call positions, jockey's weight, speed rating, and earnings, and the horse is given a number rating for each category. After all the information is entered, the score is totaled. Either enter the number of another horse you want evaluated or press the zero key.

When you hit the zero key, the question "Do you want to see a list?" appears. Enter Y and the rating list appears on the screen. If you also want a printout, answer Y to the next question. When you answer the date prompt, you need only enter the day, not the month and year.

If the horse hasn't raced in the last 15 days, the program automatically eliminates it. If the horse has been in a race within 15 days, the program awards it 10 points. Enter the distance: 8 for eight furlongs, 2 for two miles, 4.16 for 4 1/16, and so on. If the distance is the same for both this race and the last, the program adds 10 points to the total score. The class question asks for the horse's claim for this race and the previous race. The horse is awarded points if the difference between races is significant. Stretch gain for the horse's last three races receives 10 points.

When you answer the call position question, remember that it's for the last three races (any calls). If the horse came in first five times, enter 5. A first call is worth 10 points and a second call five. Again, 10 points if the jockey's weight stays the same both races.

After you enter the last three speed ratings, the program

averages them and assigns points accordingly. The last question is about the earnings for the year. The program divides earnings by the number of starts the horse made, and awards the appropriate number of points.

The total score appears on the screen for the post position that you entered and is written to the NAG.DO file. The NAG.DO file also contains all of the post positions and the scores. To rate another horse, enter a new post position number. By loading the NAG.DO file you can get a list of all of the horses on the screen. The NAG.DO file stays the same until the next time you start the program.

Bet on the horse with the highest score and hope to win. ■

### Program Listing 1. Nag Analysis.

```

10 'HORSES
20 MAXFILES=2: CLEAR:CLS:N=0
30 INPUT"TODAYS DATE ";A
40 INPUT"DAYS IN PREVIOUS MONTH ";C
50 INPUT"TODAYS DISTANCE ";B:J=0:L=15
60 INPUT"POST # ";E:IF E=0THEN 510
70 IF E>15 THEN 60
80 D=0:IF E>J THEN LET J=E
90 IF E<L THEN LET L=E
100 CLS:PRINT"DAYS BETWEEN RACES "
110 INPUT"LAST RACE DAY ";F:G=A
120 IF F>G THEN LET G=G+C
130 IF (G-F)>15 THEN PRINT"ELIMINATED":
GOTO 60
140 I=10:GOSUB 480
150 INPUT "DISTANCE LAST RACE ";R
160 IF R=B THENLETI=10:GOSUB480:GOTO180
170 GOSUB 500
180 CLS:PRINT"CLASS CHECK"
190 INPUT"TODAYS CLAIM";S
200 INPUT"LAST CLAIM ";G
210 I=INT ((G-S)/100+.5):GOSUB 480
220 CLS:INPUT"GAIN IN STRETCH (Y/N)";K$
230 IF K$="Y"THEN LET I=10:GOSUB 480:GOT
O 250
240 IFK$="N"THENGOSUB 500ELSEGOTO220
250 CLS:PRINT"RUNNING POSITION"
260 INPUT"1ST ANY CALL ";U
270 INPUT"2ND ANY CALL ";V
280 I=U*10+V*5:GOSUB480
290 CLS:PRINT"ASSIGNED WEIGHT"
300 INPUT"JOCKEY'S WEIGHT THIS RACE ";WT
310 INPUT"JOCKEY'S WEIGHT LAST RACE ";WL
320 IF WT=WLTHENLET I=10:GOSUB480:GOTO34
0
330 GOSUB 500
340 CLS:PRINT"LAST 3 SPEEDS"
350 INPUT "FIRST ";XF
360 INPUT "SECOND ";XS
370 INPUT "THIRD ";XT
380 I=INT ((XF+XS+XT)/3+.5):GOSUB 480
390 CLS:PRINT"AVERAGE EARNINGS"
400 INPUT"YEARS EARNINGS ";Z

```

|     |                      |     |                  |
|-----|----------------------|-----|------------------|
| N   | Counter              | U   | First any call   |
| A   | Today's date         | V   | Second any call  |
| C   | Days last month      | WT  | Weight this race |
| B   | Today's distance     | WL  | Weight last race |
| E   | Post number          | XF  | First speed      |
| D   | Total points awarded | XS  | Second speed     |
| F   | Last race day        | XT  | Third speed      |
| I   | Awarded points       | Z   | Year's earnings  |
| R   | Distance last race   | Y   | Number of starts |
| S   | Today's claim        | T   | Delay variable   |
| G   | Last claim           | P\$ | Y/N input        |
| K\$ | Y/N input            |     |                  |

Table 1. Nag Analysis variables list.

Listing 1 continued

Listing 1 continued

```

410 INPUT"NUMBER OF STARTS ";Y:IF Y=0 TH
EN LET I=0:GOTO 430
420 I=INT(Z/Y*.1):D=D+I
430 PRINT"AWARD ";I:FORT=1TO200:NEXTT
440 CLS:PRINT"SCORE: PP# ";E;" IS ";D
450 IF N>0 THEN GOTO 470
460 OPEN"RAM:NAG.DO"FORINPUTAS1
470 PRINT#1,"PP# ";E;" IS";D:N=N+1:GOTO
60
480 D=D+I:PRINT"AWARD ";I
490 PRINT"SCORE NOW";D: FORT=1TO200:NEXT
T:RETURN
500 PRINT "NO POINTS AWARDED":FORT=1TO20
0:NEXTT:RETURN
510 CLOSE:K$="":CLS:INPUT"DO YOU WANT TO
SEE LIST(Y/N)";K$:IFK$="N"THEN END
520 OPEN"RAM:NAG.DO"FORINPUTAS1
530 INPUT#1,F,E,G,D
540 PRINT"PP# ";E;" IS ";D
550 IFNOT EOF(1)THENGOTO530
560 CLOSE
570 PRINT:PRINT"PRESS ANY KEY TO CONTINU
E"
580 IFINKEY$=" "THEN580
590 INPUT"DO YOU WANT A PRINT OUT(Y/N)";
P$
600 IF P$="Y"GOTO620
610 IF P$="N"THEN END ELSE GOTO 590
620 OPEN"RAM:NAG.DO"FORINPUTAS1
630 INPUT#1,F,E,G,D
640 LPRINT"PP# ";E;" IS ";D
650 IFNOT EOF(1)THENGOTO630
660 CLOSE:END

```

## Harmony and Me

by Ben Firschein

The TRS-80 Model 100's tone generator lets you make music. You can access it easily from Basic with the Sound command. The manual provides a table that matches tone codes with musical notes.

The Model 100 generates notes within a range of five octaves. Unfortunately, there's no software that lets you compose, play back, save, or load a song. To play a song using the tone generator, you must write a program in Basic and use the table in the manual to find the necessary codes. *Harmony, Program Listing 2*, solves this problem. It runs in 8K RAM.

The program's menu displays the commands. You enter the names of the notes rather than tone codes.

The program plays the note as you press the key, much like a musical instrument would, and it also lets you play the complete song when you're done. You can save the song to load and play it later.

### The Main Menu

When you run the program, a menu appears, displaying the amount of free memory and the following options: Make a song, Play back, Save, Load, What are my files, and Quit. Press the appropriate key to select an option. If you don't select an available option, the menu redispays.

If you don't have enough memory to run the program, the program notifies you and requests that you make more space available and terminates its execution. You can either delete

# Continuous Checks, Statements, and Invoices for Desk-Top Computers


- Compatible with software from over 300 sources. Or program to NEBS standard forms yourself.
- Continuous Micro-Perf™ Letterheads and matching continuous Envelopes provide a clean, trim look.
- Also, diskettes, continuous labels, other supplies and accessories.
- Our policy is to process forms printed with your name within 6 working days. Then ship direct to you (We pay shipping charges on prepaid orders).

**QUALITY PRODUCTS  
SMALL QUANTITIES AT LOW PRICES  
MONEY-BACK GUARANTEE**

## FREE Full-color, catalog

fast service by mail or phone

**TOLL FREE 1 + 800-225-9550**  
(Mass. residents 1 + 800-922-8560)

|                                                                                                                                        |                                     |                                             |
|----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|---------------------------------------------|
| NAME/TITLE                                                                                                                             |                                     | PHONE                                       |
| COMPANY                                                                                                                                |                                     |                                             |
| STREET                                                                                                                                 |                                     |                                             |
| CITY, STATE, ZIP                                                                                                                       |                                     |                                             |
| SOFTWARE BRAND                                                                                                                         | PACKAGE #                           | <input type="checkbox"/> DO OWN PROGRAMMING |
| COMPUTER BRAND                                                                                                                         | MODEL                               | PLAN TO PURCHASE WITHIN:                    |
| YOUR LINE OF BUSINESS                                                                                                                  | NUMBER OF EMPLOYEES                 |                                             |
| MOST OFTEN USE MY COMPUTER FOR:                                                                                                        |                                     |                                             |
| <input type="checkbox"/> WORD PROCESSING                                                                                               | <input type="checkbox"/> ACCOUNTING | <input type="checkbox"/> OTHER              |
|                                                                                                                                        |                                     | 15110                                       |
| <br>12 South Street, Townsend, Massachusetts 01469 |                                     | 533                                         |
| A Division of New England Business Service, Inc.                                                                                       |                                     |                                             |



| Lines     | Special Feature | Comments                                                                                                                                                                                                                                                                                                  |
|-----------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | (*)             |                                                                                                                                                                                                                                                                                                           |
| 0-4       |                 | Program declaration.                                                                                                                                                                                                                                                                                      |
| 5         | *               | Model 100 Basic goes to line 200 if there's an error while running the program. (See commentary on lines 200-250 for information on error handling.)                                                                                                                                                      |
| 8         |                 | Sets maximum number of notes. Change this depending on how much memory your machine has.                                                                                                                                                                                                                  |
| 10-26     |                 | Dimension statements.                                                                                                                                                                                                                                                                                     |
| 30        | *               | Reads data on notes. The codes (lines 1000-1220) are taken from the table in the manual under the Sound command. The rows are the notes and the columns are the octaves.                                                                                                                                  |
| 32-39     | *               | Turn the function keys on for use in program.                                                                                                                                                                                                                                                             |
| 40-47     |                 | Initialize octave, duration, and note number.                                                                                                                                                                                                                                                             |
| 50        | *               | Calls Subroutine that assigns function keys. (See commentary on lines 550-665.)                                                                                                                                                                                                                           |
| 52-88     |                 | Main Menu. Tell you the options, scan keyboard for command (line 75) and go to proper subroutine if the command is valid.                                                                                                                                                                                 |
| 90-99     |                 | Compose subroutine.                                                                                                                                                                                                                                                                                       |
| 91        |                 | Prints instructions and initializes.                                                                                                                                                                                                                                                                      |
| 92        |                 | Gets a note from the user and identifies it.                                                                                                                                                                                                                                                              |
| 93        |                 | Tests for command to exit.                                                                                                                                                                                                                                                                                |
| 94        |                 | Plays the note if the note was valid.                                                                                                                                                                                                                                                                     |
| 95        |                 | If entered maximum amount of notes then done.                                                                                                                                                                                                                                                             |
| 96        |                 | New note.                                                                                                                                                                                                                                                                                                 |
| 97        |                 | Stores octave, duration, note.                                                                                                                                                                                                                                                                            |
| 99        |                 | Gets another note.                                                                                                                                                                                                                                                                                        |
| 100-160   |                 | Subroutine to find a note and identify.                                                                                                                                                                                                                                                                   |
| 170-180   |                 | Subroutine to correct a note.                                                                                                                                                                                                                                                                             |
| 200-250   | *               | Error handling subroutine. If there is an error, control is transferred to this part of the program (see line 5).                                                                                                                                                                                         |
| 300-330   |                 | Subroutine to print the key labels and initialize the number of notes to zero. It also gives instructions.                                                                                                                                                                                                |
| 400-440   | *               | Subroutine that lets you see what files you have. This subroutine uses the built-in Basic command called Files, that lists all RAM files.                                                                                                                                                                 |
| 500-520   | *               | Subroutine to play a note. L is the octave. K is the variable that is used to print the name of the note and to look up the code for that tone and octave. N\$(K) is the name of the note. The SOUND statement calls the tone generator. P%(K,L) is the code for that tone and octave. D is the duration. |
| 550-665   | *               | Select the octave. Subroutines 610-650 change the octave (L). Subroutines 655-665 change the duration of the tone (D).                                                                                                                                                                                    |
| 700-770   |                 | This subroutine (called by line 30) loads the array N\$(N) with the names of the notes (A, F, C, etc.) It also loads the matrix P%(N,C) with the codes for the tones. N is the note number, C is the octave. The subroutine loads the codes from the data on lines 1000-1220.                             |
| 800-885   |                 | This subroutine plays back a tune. LV%(NT) is the octave of a given note. DU%(NT) is the duration of a given note. NOS(NT) is the name of the note.                                                                                                                                                       |
| 900-960   | *               | This subroutine saves a tune.                                                                                                                                                                                                                                                                             |
| 970-996   | *               | Load a song.                                                                                                                                                                                                                                                                                              |
| 1000-1220 | *               | Store tone data.                                                                                                                                                                                                                                                                                          |

Table 2. Comments on music program structure and features.

some files or change line 8 of the program to free up memory (line 8 specifies the maximum amount of notes allowed). Currently, M (the maximum number of notes) is set at 100; this is about a page of music.

### Make a Song

You enter this mode with the M key. The program gives you a summary of the instructions and function key labels. Enter a note using its name (C, A, G notes, etc.) Specify sharps by pressing the shift key and the note key simultaneously (C, A, and G stand for C sharp, A sharp, and G sharp). When you hit the key, the note sounds. Use the Model 100's function keys to change the octave or the duration. Function keys 1-5 control the octave, and function keys 6-8 control the duration of the note. The octave and duration don't change again until you use the function keys to change them. Stop keying in notes at any time by hitting the period key. If you never hit the period key, the program tells you when you key in the maximum number of notes allowed. If this happens, the main menu reappears, and the program retains the notes keyed in so far.

If you key in the wrong notes, you can easily correct them; hit the backspace key to delete notes until you arrive at the first note you want to change. As you hit the backspace key, a cursor shows you the note you're on. Retype the note or notes that you want to correct. The original song is maintained up to

### Program Listing 2. Harmony.

```

0 REM music program
1 REM By Ben Firschein
2 REM June,1983
3 REM for use on Radio Shack TRS-80
  model 100 portable computer
4 REM
5 ON ERROR GOTO 200:REM handle error
8 M=100:REM maximum number of notes
10 DIM N$(12):REM notes
20 DIM P%(12,5):REM pitch
22 DIM LV%(M):REM octave
24 DIM DU%(M):REM duration
26 DIM NOS(M):REM note
30 GOSUB 700:REM read data
32 KEY (1) ON:REM turn on func key
33 KEY (2) ON
34 KEY (3) ON
35 KEY (4) ON
36 KEY (5) ON
37 KEY (6) ON
38 KEY (7) ON
39 KEY (8) ON
40 L=1:REM level (octave)
45 D=12:REM duration of tone
47 C=0:REM note #
50 GOSUB 550:REM assign function keys
52 GOTO 60:REM no time delay
55 REM main program
57 FOR ZZ=1 TO 500:NEXT ZZ:REM delay
60 CLS
61 PRINT"menu           ";FRE(0);"bytes
  free"
62 PRINT
64 PRINT"[m]ake a song","[p]layback"
65 PRINT"[s]ave","[w]hat are my files?"
66 PRINT"[l]oad"
67 PRINT"[q]uit           hit key in [ ] to
  choose"

```

Listing 2 continued

Listing 2 continued

```

75 K$=INKEY$
77 IF K$="" THEN 75
80 IF K$="m" THEN GOSUB 90
82 IF K$="p" THEN GOSUB 800
84 IF K$="s" THEN GOSUB 900
85 IF K$="l" THEN GOSUB 970
86 IF K$="q" THEN END
87 IF K$="w" THEN GOSUB 400
88 GOTO 57:REM another command
90 REM compose
91 GOSUB 300:REM print labels and init
92 GOSUB 100:REM get a note & identify
93 IF K$="." THEN RETURN
94 GOSUB 500:REM play note
95 IF C=M THEN PRINT:PRINT"can only
store";M;"notes.":RETURN
96 C=C+1:REM new note
97 LV$(C)=L:DU$(C)=D:NO$(C)=K$
98 REM 97 stores octave,duration,note
99 GOTO 92:REM another note
100 REM get a note and identify
105 PRINT"o";CHR$(8);:REM cursor and
backspace
110 K$=INKEY$
115 IF K$="." THEN PRINT".":RETURN
120 IF K$="" THEN 110
125 IF ASC(K$)=8 THEN GOSUB 170:GOTO
100:REM back space and get note
130 FOR K=1 TO 12:REM note
140 IF K$=N$(K) THEN 160
150 NEXT K
155 GOTO 110:REM illegal note
160 RETURN:REM was the kth note
170 REM correct a note
172 IF C=0 THEN RETURN:REM no notes.
cannot backspace
173 PRINT " ";CHR$(8);
174 PRINT CHR$(8);CHR$(8);CHR$(8);CHR$(8
);
176 REM chr$(8) is backspace
178 C=C-1
180 RETURN
200 REM error handling
210 IF ERR=7 THEN PRINT"out of
memory.please make some space":END
220 IF ERR=52 THEN PRINT"file not

```

Listing 2 continued

the note before the cursor. Thus, if you backspace to correct some notes and then type '.' to exit to the main menu, any notes after the cursor are not saved; you must retype any notes that appear after the cursor if you wish to save them.

### Play Back a Song

To get into this mode, hit the P key. Enter the play back mode to play back a song you composed. As the song plays, the program displays the octaves and names of the notes.

### Save a Song

To save a song type S. The save option writes the song to a file (RAM or cassette). If you type only the file name and don't specify the type of file, the Model 100 assumes it's a RAM file. Since the Model 100 retains its memory even when turned off, files written to RAM remain until you delete them from Basic.

If you enter an illegal file name, the program informs you (in English, not by an error code) and returns you to the main menu. It also alerts you if you attempt to save a RAM file and run out of memory.

### Load a Song

You can load songs that you've previously saved. If you type the name of a RAM file that doesn't exist, or a bad file name, the program informs you of the error and returns you to the main menu. After it finds the file, the program loads it and returns to the main menu. You can then play back the song using the play back option.

### What Are My Files

To see what files you have in RAM, press the W key. This is an important feature; if you have just composed a song and you have forgotten the names of your files, it's desirable to find out what files you have to prevent writing over a file. This feature also lets you verify whether you've stored a RAM file properly.

### Quit

When you finish, leave the program by calling the Quit option (Q). ■

Write to Ben Firschein at 29 Stowe Lane, Menlo Park, CA 94025.

# TRS - 80 MODEL 100

TRS 80 is a Trademark Of Tandy Corp.

Monthly magazine loaded with Software and Hardware Reviews, Useful How-to Tips, Program Listings, and Accessory Interface Suggestions for this unique new computer. Whatever your interest in this notebook size portable, this exclusive publication will help you realize its maximum potential while saving you money. Regular subscription rate is \$24/year (12 issues), however charter subscriptions are available for a limited time at \$9.00 for 6 months or \$16.00 for 12 months.

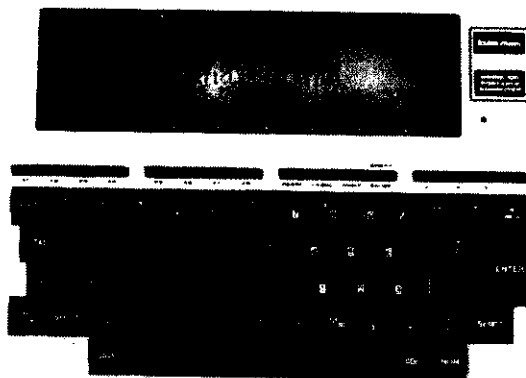


customers call 800-227-1617 ext 425

In California 800-772-3545 ext 425 or send check or Money Order to:

## briefcase portable

560 South Hartz Avenue, Suite 447 • Danville, CA 94526



165

calling all TRS-80  
MODEL IV and  
MAX-80 owners



an innovative word processing system

## NEW 80 X 24 character word processing

- Use with any Model III DOS or TRSDOS 6
- Compatibility with 64 character version
- simple, powerful, forgiving; new computer owners can easily master its one-key editing commands
- special purpose extensions available:
  - LAWYERS: make fast revisions to legal documents with LazyDoc
  - BUSINESSES: let LazyMerge print beautiful form letters after you've gone home
  - EVERYONE WITH AN EPSON OR PROWRITER PRINTER: create and print your logo or letterhead with LazyFont; large selection of free fonts and graphics to get you started

PLUS OTHER EXTENSIONS — SEND FOR FREE INFORMATION. All Lazy Writer owners get free newsletters and low priced updates

"Lazy Writer has long been highly regarded... the best supported word processor for the TRS-80."  
80 Micro, June 1983.

DEALER INQUIRIES WELCOME

Still only \$175.00

AlphaBit Communications, Inc. ✓ 476

Call Now!  
MC/Visa  
accepted

13349 Michigan Ave.  
Dearborn, Michigan 48126  
(313) 581-2896

## NOTES

Listing 2 continued

```

found.":RESUME 55:REM menu
230 IF ERR=55 THEN PRINT"bad file
name.":RESUME 55:REM menu
240 PRINT "error code ";ERR;"in line
";ERL:STOP
250 REM end of error handling
255 REM
300 REM print key lables & initialize
302 C=0:REM no note yet
304 CLS:PRINT"make a song (up
to";M;"notes)"
305 PRINT"hit key of the note to
play.for sharps:"
306 PRINT"shift & the key.'.'=done.hit [
bksp] to"
310 PRINT"change note.fl-f8 change
octave&length:"
312 PRINT
315 PRINT"oct1 oct2 oct3 oct4 oct5 1/4
1/2 -1/1"
330 RETURN
336 PRINT"wait";
400 REM what are my files
410 CLS
430 FILES
431 PRINT
434 PRINT"hit space bar to continue ";
435 W$=INKEY$:IF W$<>" " THEN 435
436 PRINT " wait";
440 RETURN
442 REM
500 REM play the note
510 PRINT L;N$(K);' ';
515 SOUND P$(K,L),D
520 RETURN
550 REM select the octave
600 ON KEY GOSUB
610,620,630,640,650,655,660,665
605 RETURN
610 L=1:RETURN
620 L=2:RETURN
630 L=3:RETURN
640 L=4:RETURN
650 L=5:RETURN
655 D=12:RETURN:REM tone duration
660 D=25:RETURN
665 D=50:RETURN
700 REM load arrays with values
705 REM
710 FOR N=1 TO 12:REM note
720 READ N$(N):REM note name
730 FOR C=1 TO 5:REM octave
740 READ P$(N,C):REM note,octave
750 NEXT C
760 NEXT N
770 RETURN
800 REM playback
801 CLS:PRINT"playback":PRINT
802 IF C>0 THEN 820
804 PRINT"error. no notes to play"
808 RETURN
820 FOR NT=1 TO C:REM stored notes
839 L=LV$(NT):REM octave
845 D=DU$(NT):REM duration
850 K$=NO$(NT):REM note
860 GOSUB 130:REM identify
865 GOSUB 500:REM play the note
870 NEXT NT

```

Listing 2 continued

Listing 2 continued

```

875 PRINT
885 RETURN
900 REM save
905 CLS:PRINT"save":PRINT
906 IF C>0 THEN 910
908 PRINT"error. no notes to save"
909 RETURN:REM to menu
910 INPUT"filename ";N$
920 OPEN N$ FOR OUTPUT AS 1
923 PRINT"saving ";N$
925 FOR NT=1 TO C
930 PRINT #1,LV%(NT),DU%(NT),NO$(NT)
932 PRINT LV%(NT);NO$(NT);" ";
940 NEXT NT
942 PRINT #1,0,0,"*":REM eof
944 CLOSE 1
955 PRINT:PRINT"saved. ";FRE(0);" bytes
free."
960 RETURN
970 REM load
972 CLS:PRINT"load":PRINT
974 INPUT"file ";NA$
976 OPEN NA$ FOR INPUT AS 1
978 PRINT"found ";NA$
980 FOR C=1 TO M:REM m=max
982 INPUT #1,LV%(C),DU%(C),NO$(C)
984 IF NO$(C)="*" THEN 990:REM done
986 NEXT C
990 C=C-1
992 CLOSE 1
994 PRINT"loaded"
996 RETURN
1000 DATA "g",12538,6269,3134,1567,783
1020 DATA "G",11836,5918,2959,1479,739
1030 DATA "a",11172,5586,2793,1396,698
1040 DATA "A",10544,5272,2636,1318,659
1050 DATA "b",9952,4976,2484,1244,622
1060 DATA "c",9394,4697,2348,1174,587
1070 DATA "C",8866,4433,2216,1108,554
1080 DATA "d",8368,4184,2092,1046,523
1090 DATA "D",7900,3728,1975,987,493
1200 DATA "e",7456,3718,1864,932,466
1210 DATA "f",7032,3516,1758,879,439
1220 DATA "F",6642,3321,1660,830,415

```

## Remote Robot

by Peter W. Deininger and Rolf A. Deininger

Radio Shack sells a radio-controlled robot (part number 60-3023A) that you can control with your Model 100. It's a cute, but simple, robot that's only able to move forward, backward, and make turns.

If you open the robot's hand-held remote control, you'll find that pressing the button causes a contact (two copper plates) to close.

The Model 100 controls the robot easily with two resident Basic commands, Motor On and Motor Off. Normally, these commands control a cassette recorder connected to the 100. They turn a relay on the Model 100 on or off. The smallest plug on the cassette interface cable carries the relay status to the recorder.

To control the robot through the Model 100, all you need do is install a subminiature phone jack (Radio Shack part number 274-292) in the robot's remote control with the two

```

10 'MODEL 100 ROBOT CONTROL'
20 'PETER DEININGER JULY 83'
30 PRINT"HOW MANY SECONDS FOR FULL TURN"
40 INPUT FT
50 FS=320*FT :RT=1/4*FS
60 RVS=1/2*FS:LFT=3/4*FS
70 CLS:PRINT"TURN ON ROBOT....."
80 A$=INKEY$:IF A$=""THEN 80:'GET CMND'
90 A=ASC(A$) :A=A-27
100 IF A>5 OR A<1 THEN 80
110 ON A GOSUB 130,160,190,210,240
120 GOTO 80
130 PRINT"RIGHT ";
140 MOTOR ON:FOR I=1 TO RT:NEXT I
150 MOTOR OFF:RETURN
160 PRINT"LEFT ";
170 MOTOR ON:FOR I=1 TO LFT:NEXT I
180 MOTOR OFF:RETURN
190 PRINT"FORWARD ";
200 MOTOR OFF:RETURN
210 PRINT"REVERSE ";
220 MOTOR ON:FOR I=1 TO RVS:NEXT I
230 MOTOR OFF:RETURN
240 PRINT"STOP ";
250 FOR I=1 TO 40
260 MOTOR ON:FOR D=1 TO 60:NEXT D
270 MOTOR OFF:FOR D=1 TO 60:NEXT D
280 NEXT I :RETURN

```

Program Listing 3. Robot Control.

leads connected to the push-button switch. By plugging the cassette cable into this socket, you can simulate pushing the button on the remote controller with your Model 100.

The Motor On command is now equivalent to pushing the button; the Motor Off command is equivalent to releasing the button.

### Robot Control

The following program lets you control the robot from the Model 100 keyboard after you plug in the cassette cable.

```

10 A$=INKEY$:IF A$=""GOTO 10
20 IF A$="F" THEN MOTOR OFF
30 IF A$="R" THEN MOTOR ON
40 GOTO 10

```

Typing an F makes the robot go forward and typing an R makes it go backward.

To make the robot go left and right and stop takes a bit more work (see Program Listing 3). Making a left turn is analogous to trying to turn left on a road that prohibits left turns. By making three right turns, you ultimately head off in the left direction. The same holds true for the robot; the program triggers three right turns to make the robot turn left. It simulates a stop by repeatedly sending commands to move the robot forward and backward in small steps.

The general control program begins by asking how long, in seconds, it takes the robot to make a full, 360-degree turn. The program then uses this estimate to turn the robot 90, 180, and 270 degrees. Use the four arrow keys to move the robot in the appropriate direction and press the space bar to make the robot stop. The program runs in 8K RAM. ■

Write to Peter and Rolf Deininger at 3063 Overridge Drive, Ann Arbor, MI 48104.



## Winning Numbers

by Ronald F. Balonis

If you play a daily numbers game, this program increases your chances for that megabucks win. It helps you choose which daily number to bet on no matter what number selection scheme you prefer.

The Daily Numbers Statistician computes a ranked histogram of the digits used in a daily numbers game (see Program Listing 4). The program ranks digits according to the frequency with which they're drawn.

The program analyzes the numbers drawn from a daily numbers file (DAILY.DO), which you maintain using the resident text editor. Using this information, you can compute the mathematical probability of success for numbers based on the frequency distribution.

### Program Operation

The daily numbers file is organized as an inverted file with the latest entries listed first (see Fig. 1). The file data given here is from the Pennsylvania Daily Number Lottery from April 29, 1982, to April 29, 1983. Change the state by changing the definition of STATE\$ in line 10 and inserting an historical file of the appropriate daily numbers.

Program operation is screen-oriented. Screen 1 asks you if you want to recalculate the statistics (you have to do this for the first run of the program to create the statistics file DNSTAT.DO). Screen 2 prompts you for the number of digits (greater than 10) you select for the recalculate option. Screens

```

157 803 542 833 380 600 823 262 954 414
120 668 501 516 475 119 447 952 149 322
847 485 936 790 236 051 904 120 387 031
641 008 107 552 061 302 830 775 877 306
917 172 580 082 518 603 030 981 583 932
242 013 707 643 438 828 155 572 617 272
751 538 530 777 510 351 469 587 267 790
579 908 428 514 366 991 931 602 837 917
324 291 580 587 437 217 912 306 088 271
513 990 367 271 665 174 034 815 431 724
651 126 936 253 147 560 255 483 619 530
455 787 433 568 007 185 755 829 695 479
075 443 000 551 555 967 160 925 721 277
108 694 151 151 135 430 594 037 569 847
808 515 653 737 861 986 662 621 711 798
995 368 041 540 221 347 952 446 625 735
811 726 594 079 317 357 574 852 023 039
174 842 954 298 706 440 402 308 097 851
520 885 663 700 877 989 215 398 031 481
060 802 289 744 258 961 805 144 064 601
605 731 995 722 645 590 511 243 308 470
329 216 310 302 130 321 668 707 955 468
773 520 096 663 004 688 817 366 444 695
068 577 594 473 218 243 528 180 492 385
186 498 098 195 151 929 806 042 947 690
715 398 235 069 970 006 735 612 336 140
934 485 374 021 028 055 695 696 623 858
522 308 545 590 915 321 522 066 998 600
045 267 356 013 289 477 662 290 524 277
399 855 164 611 882 690 144 517 771 673
329 325 782 678 003 700 230 874 424 189
673 444 649 696
    
```

Figure 1. Daily numbers file. The latest entries are listed first.

3 and 4 notify you of program operation.

Screen 5 displays the analysis. The ranked histogram display shows the random nature of each digit and digit trends in a run of numbers. Enter a number to calculate its probability and press the E key to exit to Screen 1.

Daily Number Statistician requires between 4K and 6K of memory. The program DAILY.N.BA takes up 3.75K with remark statements, 3K without. The statistics file (DNSTAT.DO) uses 360 bytes of memory. The daily numbers file, DAILY.DO, needs 4 bytes of memory for each three-digit number you enter.

To use the program on an 8K machine, key it in without the remark statements. The program's only limitation is the amount of memory you have available. ■

Contact Ronald F. Balonis at 118 Rice St., Trucksville, PA 18708.

### Program Listing 4. DAILY.N.BA.

```

5 'DAILY.N.BA 5/30/83 BY RON BALONIS
10 CLEAR 200:STATE$="PA"
15 TITLE$="* "+STATE$+" DAILY NUMBER
  STATISTICIAN *"
20 CLS:PRINT@4,TITLE$
30 PRINT@130,"RECOMPUTE STATISTICS?"
35 PRINT@209,"<Y>ES, <N>O OR <E>XIT ";
40 POS=233:LG=0:GOSUB 10000:'SCAN KEYS
50 IF KB$="Y" THEN 100
60 IF KB$="N" THEN 800
70 IF KB$="E" THEN MENU ELSE GOTO 20
80 '
90 '---READ THE DAILY NUMBER FILE---
100 CLS:PRINT@4,TITLE$
105 PRINT@125,"NUMBER OF NUMBERS ";:
  INPUT N0
110 IF N0<10 OR N0>999 THEN 100
115 PRINT@125,"**** COMPUTING STATISTICS
  ****"
120 '--READ & SELECT SORT #S OF # FILE--
125 OPEN "DAILY0.DO" FOR INPUT AS 1
130 NX=NX+1
135 FOR I=1 TO 4
140 IF EOF(1) THEN 160
145 N$=INPUT$(1,1):N=VAL(N$):STAT(I,N)
  =STAT(I,N)+1
150 NEXT I
155 IF N0=NX THEN 160 ELSE 130
160 CLOSE 1
180 '
190 '---FORM RANKING INDEX---
200 FOR I=1 TO 4
210 FOR II=1 TO 10
220 RANK(I,II)=II-1
230 NEXT II
240 NEXT I
480 '
490 '---RANK FOR LEFT TO RIGHT DISPLAY--
  -
500 FOR II=1 TO 3:M=10:N=10
510 M=INT(M/2)
520 IF M=0 THEN 590 ELSE I=1:L=N-M
530 J=I
540 K=J+M:JJ=RANK(II,J):KK=RANK(II,K)
550 IF STAT(II,JJ)>=STAT(II,KK) THEN
580
    
```

Listing 4 continued

Listing 4 continued

```

560 R=RANK (II,J):RANK (II,J)=RANK (
II,K):RANK (II,K)=R
570 J=J-M:IF J<1 THEN 580 ELSE 540
580 I=I+1:IF I>L THEN 510 ELSE 530
590 NEXT II
680 '
690 '---SAVE THE STATISTICS IN A FILE---

700 OPEN "DNSTAT.DO" FOR OUTPUT AS 1
705 FOR II=0 TO 9
710 PRINT#1,USING" # ";II;
715 NEXT II
720 PRINT#1,""
725 FOR I=1 TO 4
730 FOR II=0 TO 9
735 PRINT#1,USING"### ";STAT(I,II);
740 NEXT II
745 NEXT I
750 FOR I=1 TO 4
755 FOR II=1 TO 10:
760 PRINT#1,USING"### ";RANK(I,II);:
765 NEXT II
770 NEXT I:CLOSE 1:GOTO 1000
780 '
790 '---READ STATISTICS FILE---
800 CLS:PRINT@4,TITLE$
805 PRINT@126,"**** READING STATISTICS *
***"
810 OPEN "DNSTAT.DO" FOR INPUT AS 1
815 INPUT#1,U0$
820 FOR I=1 TO 4
825 FOR II=0 TO 9
830 INPUT#1,STAT(I,II)
835 NEXT II
840 NEXT I
845 FOR I=1 TO 4
850 FOR II=1 TO 10
855 INPUT#1,RANK(I,II)
860 NEXT II
865 NEXT I
870 CLOSE 1:NX=STAT(4,0)
980 '
990 '---THE HISTOGRAPHIC ANALYSIS---
1000 CLS:PRINT STATES$+" DAILY"
1010 PRINT" NUMBER "
1020 PRINTUSING"LAST ###";NX;
1025 PRINT"# "
1030 PRINT"WIN PROB."
1035 PRINT:PRINT"[ <E>XIT]"
1040 PRINT@240,"[ENTER #]"
1050 PRINT@283,"000";
1055 '---COMPUTE DISPLAY OFFSET---
1060 FOR I=1 TO 3
1070 IF STAT(I,RANK(I,1))-XSET>53 THEN
XSET=XSET+1:GOTO 1070
1080 NEXT I
1090 '---RANKED HISTOGRAPH DISPLAY---
1100 II=0
1110 PSET(53,54)
1120 FOR Y=1 TO 180 STEP 60
1125 II=II+1
1130 LINE(52+Y,0)-(112+Y,53),1,B
1140 PSET(112+Y,54)
1150 FOR I=0 TO 9
1160 Y0=Y+54+I*6
1170 X0=53-STAT(II,RANK(II,I+1))+XSET
1175 IF X0>53 THEN X0=53
1180 LINE(Y0,X0)-(Y0+2,53),1,BF

```

Listing 4 continued

# MODEL 100

We're **ALPHAWARE**, and we think computers should be user-friendly, always. That's why, no matter where you are in **MicroEditor II** or **Mail Master**, you have immediate access to **GUARDIAN™**, a full-function help feature ready to assist at any time. Finally, programming that puts you in control: **MicroEditor II** and **Mail Master** by **ALPHAWARE**.

Dealer Inquiries Invited

## MicroEditor II

**\$49.95**

- Pagination
- Right, left, alternate margins
- Control of page width & length
- Line and page numbering
- Clear memory
- Center, justify
- Headings
- File handling
- Form letters
- Tabs

## Mail Master

**\$34.95**

- Cross reference
- Alphabetize
- Control sorting
- Mailing labels
- MicroEditor II compatible

# ALPHAWARE

INCORPORATED

2100 South Boulevard • Edmond, OK 73034

## UNITED SOFTWARE ASSOCIATES

PRESENTS

# ULTRA TERM

### A FULL FEATURED TERMINAL PROGRAM

The Ultra Term communications package is one of the easiest to use and most versatile communications programs available for the TRS-80. It includes a full featured intelligent terminal program, with all the popular features of competing programs costing two to three times as much, and some new features that can't be found anywhere else at any price. Ultra Term also includes a self relocating host program, and hex conversion utilities for bulletin board downloading. Some of Ultra Term's unique features are:

- Supports both manual and auto dial modems.
- Exclusive Ultra Term direct to disk file transfer mode, allows unattended operation at the receiving computer.
- Exclusive split screen feature allows simultaneous two way communications without confusion.
- Line printer support with a 1K print buffer.
- Half and full duplex support.
- Universal ASCII format file transfer with a 33K Buffer.
- A full featured host program.

|               |         |                   |          |
|---------------|---------|-------------------|----------|
| • ULTRA TERM  | \$59.95 | • LYNX MODEM      | \$249.00 |
| • INFOEX 80   | \$99.95 | • ANCHOR 1-300    | \$99.00  |
| • M TERM      | \$69.95 | • HAYES 300       | \$249.00 |
| • COMMBAT     | \$44.95 | • HAYES 1200      | \$619.00 |
| • MODEM GAMES | \$19.95 | • ORCHESTRA 85/90 | \$89.00  |
|               |         | • PIANO SOFTWARE  | \$34.95  |

ORDERS 305/965-3496 BBS 305/842-2687  
 COD ADD \$3.00 TECHNICAL INFOR. 305/842-8805 (5-9 E.S.T.)  
 734 Flamingo Way, North Palm Beach, FL 33408

-332

Listing 4 continued

```

1190 PRINT@289+I+Y/6,RIGHT$(STR$(RANK
(II,I+1)),1);
1200 NEXT I
1210 NEXT Y:GOTO 1400:'DISP PROB OF 000
1280 '
1290 '--COMPUTE A #'S PROBABILITY--
1300 P0S=283:LG=3:GOSUB 10000:'SCAN KEYS

1390 '--PROB. OF 3 INDEPENDENT EVENTS--
1395 '--BASED ON NUMBERS DRAWN IN PAST-
1400 PROB=(STAT(1,N1)/STAT(4,0))
1410 PROB=PROB*(STAT(2,N2)/STAT(4,0))
1420 PROB=PROB*(STAT(3,N3)/STAT(4,0))
1430 '
1440 PRINT@160,USING" #.#####";PROB;
1450 GOTO 1300:'LOOP ON PROB. COMPUTE
1500 '
9990 '---KEY BOARD---
10000 K1=255:DISP=65024
10100 STRT=P0S:IK=STRT:EN=P0S+LG
10110 K2=PEEK(IK+DISP)
10115 PRINT@IK,CHR$(K1);:IC=0
10120 KB$="":KB$=INKEY$:IC=IC+1
10125 IF IC=20 THEN PRINT@IK,CHR$(K2);:
ELSE IF IC=40 THEN 10115
10135 IF KB$="" THEN 10120
10140 KB=ASC(KB$):PRINT@IK,CHR$(K2);

```

```

10145 IF KB=28 THEN 10220:'CURSOR RIGHT
10150 IF KB=29 OR KB=8 THEN 10240:'BCKSP
10155 '--GET ONE CHARACTER & RETURN--
10160 IF LG=0 THEN PRINT@IK,KB$;:RETURN
10165 IF KB$="E" THEN RUN:'EXIT BY ABORT

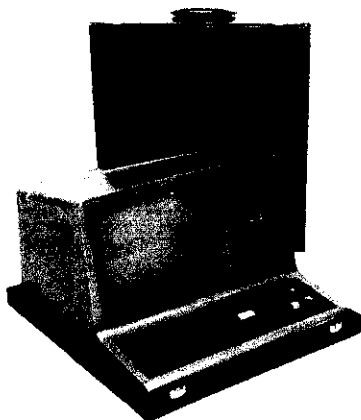
10170 IF KB=13 THEN 10300:'NORMAL EXIT

10175 IF KB<48 OR KB>57 THEN 10110
10180 IF IK=EN THEN 10110
10185 PRINT@IK,KB$;:'--ONLY NUMBERS--
10190 '
10195 '--ADJUST CURSOR POSITON--
10200 IF IK+1>EN THEN 10110 ELSE IK=IK+
1:GOTO 10110
10210 '--CURSOR RIGHT ONE SPACE--
10220 IF IK+1>EN THEN 10110 ELSE POKEIK+
DISP,K2:IK=IK+1:GOTO 10110
10230 '--BACKSPACE CURSOR--
10240 IF IK-1<ST THEN 10110 ELSE POKEIK+
DISP,KZ:IK=IK-1:GOTO 10110
10280 '
10290 '--SET UP VALUES & RETURN--
10300 I=STRT+DISP:N1=VAL(CHR$(PEEK(I)))
10310 I=I+1: N2=VAL(CHR$(PEEK(I)))
10320 I=I+1: N3=VAL(CHR$(PEEK(I)))
10330 '--WAIT FOR KEY UP--
10340 IF INKEY$="" THEN RETURN ELSE
10400

```

**Computer Case Company**

# WE MAKE A GREAT CASE FOR YOUR RADIO SHACK.



**RS204**

One size does not fit all. Our cases are designed for specific hardware configurations. When you put your computer in our case, it fits hand-in-glove. Once your equipment is safely inside the attache-style carrying case, it never has to be taken out again. To operate, simply remove the lid and connect the power. To store your computer, disconnect the power, enclose your disks, working papers, and manuals in the compartments provided, and attach the lid. It's as easy as that.

**CALL TOLL FREE: (800) 848-7548**

- RS201 Model I with expansion unit and drives ..... \$109
- RS204 Model III or IV ..... 129
- AP106 Amdek Color I, II or III Monitor ..... 119
- P401 Paper Tiger Printer (400/445/460) ..... 99
- P402 Centronics 730/737 & Radio Shack Printer ..... 89
- P403 Epson MX70 or MX80, Microline 82A Printer or Color Computer ..... 89
- P404 Epson MX100 Printer ..... 99
- P405 IDS 560 or Prism 132 Printer ..... 109
- P406 C. Itoh Starwriter/Printmaster F-10 Printer ..... 119
- P407 Okidata Microline 83A or 84 Printer ..... 99
- P408 C. Itoh Prowriter 2 Printer ..... 99
- P409 C. Itoh Prowriter (Apple Dot Matrix) or NEC PC8023 Printer ..... 89
- P410 Epson FX80 Printer ..... 109
- IB501 IBM Personal Computer with Keyboard ..... 129
- IB502 IBM Monochrome Monitor ..... 99
- CC50 Case Cart ..... 79
- CC80 Matching Attache Case (5") ..... 85
- CC90 Matching Attache Case (3") ..... 75
- CC91 Matching Accessories Case (5 1/4" Diskettes, Paper, etc.) ..... 95
- CC92 5.25" Diskette Case (Holds 75 Diskettes) ..... 49



✓ 139

**Computer Case Company, 5650 Indian Mound Court, Columbus, Ohio 43213 (614) 868-9464**

# RAM FILES

*RAM Files is our readers' forum for the C•Notes section. Send your questions and comments to: C•Notes Editor, 80 Micro, 80 Pine St., Peterborough, NH 03458.*

## Itinerary 100 Error

Brad Dixon's "Itinerary 100" program (July 1983, p. 173) has a bug that precludes programs from correctly reading data files.

Lines 170-180 all contain the same logic fault; they check for an end-of-file, but don't read the file if EOF is not encountered. Adding an Else statement to these lines allows the program to read all of the data. The corrected lines are in Program Listing 1.

*Jim Gaffney  
10549 Springwood Drive  
El Paso, TX 79925*

```
170 FOR I=1 TO NH: IF EOF(1) THEN 171 EL
SE INPUT#1,AD(I),HO(I),DR(I),DD(I): NEXT
171 FOR I=1 TO NO: IF EOF(1) THEN 172 EL
SE INPUT#1,OT(I): NEXT
172 FOR I=1 TO NA: IF EOR(1) THEN 180 EL
SE INPUT#1,WH(I),DA(I),DT(I),DO(I): NEXT
180 FOR I=1 TO NC: IF EOF(1) THEN 190 ELS
E INPUT#1,RS(I),AG(I),PD(I),PL(I),RD(I),
RE(I): NEXT: CLOSE: GOTO 100
```

*Program Listing 1. Correction for "Itinerary 100" program.*

## Of Protocol and Handshaking

"Nothing's Perfect" by Ken Barbier (August 1983, p. 209) is correctly named. Barbier says that the Model 100's serial port doesn't conform to the RS-232 standard, and then he goes on to say that the 100 ignores RTS and CTS handshaking!

Since when does protocol become part of the RS-232 standard? How the port handles RTS and CTS is a function of the software driver.

I have both a Model I and a Model III with RS-232 ports. I write communication drivers and use the CTS and RTS as I choose in software. The hardware provides the information, but it is up to me to use it.

The fact that the 100 protocol doesn't match Barbier's printer is not the fault of the RS-232 port, but of the incompatible drivers.

I don't use CTS or RTS at all; I use XON/XOFF. When I'm on CompuServe, RTS and CTS are of little value because my modem has no way to send the information on the tones, but CompuServe does respond to XON/XOFF.

I have communicated at the 19,200 baud rate (hard wired) and the handshaking works well. My effective baud rate is slowed by the screen scrolling as mentioned, but I don't lose data.

RTS and CTS are useful in the multiwire and hardware

environment, but when it comes to modem communications, the XON/XOFF protocol provided is better.

If Barbier wants to point to the RS-232 standard, he should point out that the standard defines the 1 and zero states as "greater than +3 and -3 volts." If you check the voltage swing you will find that it's marginal.

Radio Shack has a modification to change the value of some resistors, so if you have an early unit you might wish to have it checked for this free modification.

*C. Warren Andreasen  
P.O. Box 8306  
Van Nuys, CA 91409*

## Barbier's Reply

*C. Warren Andreasen's letter contains a lot of information valuable to any Model 100 owner; however, I think he missed the point I was trying to make about the 100's RS-232 handshaking signals.*

*Since the Radio Shack manual includes a list of interface signals, and since the four signals named in my article are on the list, the computer is defective in design or software if it does not use those signals.*

*The distinction between standard and protocol is not important. Radio Shack says the signals are used, but they aren't: that is what's hurting the users.*

*The Model 100 is not alone in improperly using these handshaking lines, and as Andreasen's letter points out, the XON/XOFF protocol is more reliable when implemented.*

*But what is the user to do when connecting to a device that does not include XON/XOFF? He will look in the manual and be misled into believing that the standard RS-232 handshake signals are implemented and are used in a standard manner.*

*They aren't, and that is the point I was trying to make.*

*Ken Barbier  
Box 1253  
Borrego Springs, CA 92004*

## Nothing's Perfect Revisited

Concerning Ken Barbier's article, "Nothing's Perfect" (August 1983, p. 209), I, too, find the Model 100 to be very slow on a benchmark program involving trigonometric functions.

On the other hand, I find the 100 to be quite fast in calculations involving variables assignment, floating point arithmetic, and so on.

It's 35 percent faster than the standard Model III for these types of calculations with single-precision accuracy on each machine. With double-precision calculations, the difference is even greater.

Regarding the RS-232 port, the standard Radio Shack Model III serial cable (PN 260-1408) fits the Model 100 perfectly. Therefore, I disagree with the concern that a standard cable does not fit.

Barbier had trouble uploading text from the 100 to Word-



Star because of the 100's hard carriage return at the end of each line.

Appendix C of the 100's manual (p. 198) says that to send a text file as is, without carriage returns, press the enter key without specifying a number in response to the Width query when uploading a file from TELCOM.

This procedure works fine for me in uploading text files from the 100 to Scripsit and Superscripts.

*James M. Stubchaer  
869 North Kellogg Ave.  
Santa Barbara, CA 93111*

### Runner's Program

Program Listing 2 may interest your readers who, like myself, are both computer enthusiasts and runners.

Many runners log their daily runs to keep track of distance run and time required. To determine their velocity, they convert their running time into units of minutes per mile.

To use this program, input the distance run and the time elapsed (in minutes and seconds). The program then calculates the rate at which you ran.

*H. Robert Lind  
26 Ferris Hill Road  
New Canaan, CT 06840*

### 100 Screen Dump

Soon after I purchased my Model 100 I wrote to Tandy to ask for information about getting screen dumps showing pixel graphics. They have not been very helpful in releasing such information.

Does anyone have a program that will duplicate the Model 100 screen on a printer?

*William R. Harlow  
340 Halidonhill Drive  
Cincinnati, OH 45238*

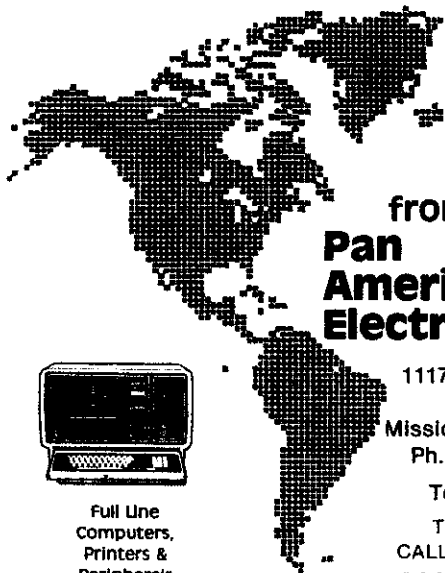
```

10 CLS
20 INPUT"HOW MANY MILES DID YOU RUN TODA
Y?";D
30 CLS
40 INPUT"WHAT WAS YOUR TIME IN MINUTES?"
;M
50 CLS
60 INPUT"AND IN SECONDS?";S
70 CLS
80 P=(S/60+M)/D
90 PRINT INT(P)
100 T=((S/60+M/D-INT(P))*60
110 PRINT INT(T)
120 PRINT "YOU RAN "D" MILES AT "INT(P)"
MIN. AND "INT(T)" SEC.!"
    
```

*Program Listing 2. Runner's program.*

## Radio Shack TRS-80™ Computers

At Guaranteed Savings



from  
**Pan  
American  
Electronics**

1117 Conway Ave.  
Dept. EM  
Mission, Texas 78572  
Ph. 512/581-2766

Telex 767339

TO ORDER  
CALL TOLL FREE  
**800-531-7466**

Full Line  
Computers,  
Printers &  
Peripherals

### Faster Foxfighter

Thanks for the articles on the Model 100. I bought a 24K version to supplement my Model III and use both primarily for civil engineering calculations and word processing.

Just for fun, I keyed in the "Foxfighter" game program from the August 1983 issue (p. 200) and made some minor changes to save memory and make it run faster.

Here are the changes:

```

5 DEFINT A-Z
20 C$ = CHR$(239):D$ = CHR$(238):E$ = CHR$(232):F$ = CHR$(237):
G$ = CHR$(233):H$ = STRING$(9,239)
    
```

The first line change declares all numeric variables to be integers (line 5). This change alone gives a noticeable increase in the speed of the fighter.

The other changes reduce the volume of typing to enter the program, and save 600 bytes of memory. The original program includes about a hundred PRINTCHR\$(xxx) statements.

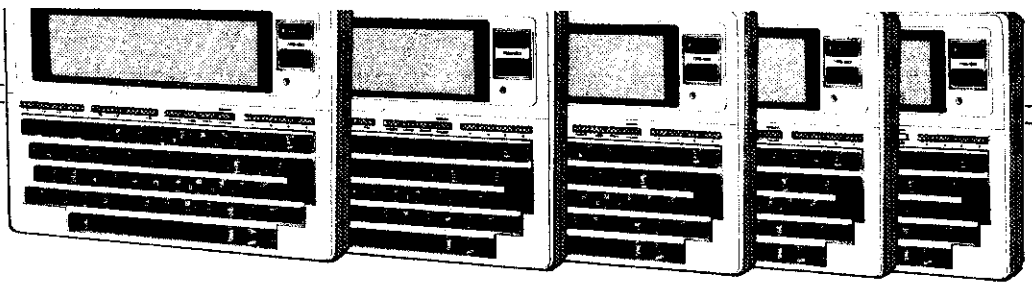
I assign the CHR\$(xxx) statement to string variables (revised line 20). The syntax is C\$ = CHR\$(xxx). Thereafter, two characters replace nine for every occurrence.

Lines 290-320 of the original program print CHR\$(239) nine times. This can be shortened by using the Model 100's STRING\$ function (line 20). The form is H\$ = STRING\$(9,239). This creates a string of nine CHR\$(239)s.

I omitted instructions from my version of the program. The result of all these changes is an 1,843-byte program instead of the original version's 3K.

*James M. Stubchaer  
869 North Kellogg Ave.  
Santa Barbara, CA 93111*

C-NOTE FOR DECEMBER 1983



## Inside the Model 100

by David P. Sumner

To help you write sophisticated programs on your Model 100, I'm providing a fairly extensive memory map, some relevant documentation, and a list of the tokens for the Basic keywords.

One word of caution: It's possible that some versions of the Model 100 have a ROM slightly different from the one in my machine. You should also be aware that some locations have multiple uses, some of which may not be documented here.

### Running a Machine-Language Program

You will find the addresses of several routines provided in the memory map useful (see Table 1) primarily for machine-language programs, but it's important to know how to access them from Basic. The secret is in the Basic command Call. To run a machine-language program that resides at location 30306 (the Model 100's sound routine), type in the Basic statement CALL 30306 and you'll hear a beep. Not all Calls are so direct. Often you'll need to pass data to the routine that you're calling. The Model 100 provides nicely for this. A statement of the form

CALL address,X,Y

initiates the execution of the machine code stored at the given address, but first it assigns the value in X to the accumulator, and the value in Y to the register pair HL. Thus, the value of X can only be between zero and 255 (so that it can fit into the 8-bit accumulator), and the value of Y can be any 16-bit integer from zero to 65535.

For example, a Call to location 32 prints the contents of the accumulator (considered an ASCII character) on the display. If you simply type CALL 32, nothing happens since the accumulator contains zero. However, if you type CALL 32,65 you'll see the letter A appear on the screen. You've put 65, the ASCII code for an A, into the accumulator prior to the Call. The effect of CALL 32,X is the same as printing CHR\$(X). In fact, since PRINT CHR\$(7) produces a beep, CALL 32,7 also results in a beep. Another example: The routine located at 10161 displays the message located at the address pointed to by the register pair HL. If you look at the memory map, you'll see that the 100 stores many messages throughout the ROM. For example, try typing CALL 10161,0,3442. The computer displays the message "extra ignored." Type CALL

### The Key Box

The programs in "Blackjack" and "Program Length" will run in 8K RAM.

Table 1. The memory map.

NOTE: Material in single quotes represents ASCII text. A "+" after a location indicates a 2-byte pointer.

|      |                                                                                                  |
|------|--------------------------------------------------------------------------------------------------|
| 0    | (RST 0) JMP 32051                                                                                |
| 3    | 'Menu'                                                                                           |
| 8    | (RST 1) Test for special character and fall into RST2.                                           |
| 16   | (RST2) Set pointer to next character of BASIC text. (JMP 2136)                                   |
| 24   | (RST3) Compare registers HL,DE directly                                                          |
| 32   | (RST 4) PRINT ASCII contents of accumulator.                                                     |
| 36   | (TRAP) Handles power-down. User interface at 62978.                                              |
| 40   | (RST 5) JMP 4201 Checks variable type.                                                           |
| 44   | (RST 5.5) JMP 62969                                                                              |
| 48   | (RST 6) Returns sign of FAC1. (JMP 13276)                                                        |
| 52   | (RST 6.5) JMP 28076 (62972 RAM vector interface)                                                 |
| 56   | (RST 7) Executes routine indicated by next byte. (JMP 32767)                                     |
| 60   | (RST 7.5) Updates timer, adjusts power-down values etc. (JMP 6962, user can interface at 62975). |
| 64   | BASIC function addresses.                                                                        |
| 128  | BASIC Keywords with high bit set in the first character of each word.                            |
| 610  | BASIC command addresses.                                                                         |
| 750  | Table of BASIC addresses.                                                                        |
| 796  | 2-Byte error codes.                                                                              |
| 858  | Initial values for pointers 62960-63103.                                                         |
| 1003 | 'Error'                                                                                          |
| 1010 | 'in'                                                                                             |
| 1014 | 'ok'                                                                                             |
| 1019 | 'Break'                                                                                          |
| 1094 | Syntax error. Other entry points at 1100, 1103 .. 1115 for other errors.                         |
| 1117 | Error message based on contents of E register.                                                   |
| 1245 | PRINTs error messages (accumulator holds a value from 28 to 58).                                 |
| 1520 | Builds BASIC line pointers.                                                                      |
| 1576 | Enter with DE containing a line number. Exit with BC containing                                  |

Table 1 continued

10161,0,32676, and the TRS-80 logo will appear on the display.

### PEEK, POKE, and Pointers

There is more of interest here than just the addresses of particular routines. Basic uses many locations to store data such as the values of variables, the text of Basic programs, and pointers that determine how the computer behaves. The PEEK command accesses the values in these locations, and the POKE command can change them.

For instance, the location 63368 contains the horizontal position of the cursor. If you let  $X = \text{PEEK}(63368)$  in a Basic program, the effect is the same as if you had used the expression  $X = \text{POS}(0)$ . Try the following short program as an illustration. (Don't overlook the semicolon in line 20.)

```
10 FOR I=1 TO 100
20 PRINT PEEK(63368);
30 NEXT I
40 END
```

If the number stored in location 64173 is zero, then you cannot use the label line. With a POKE command you can put a zero into this location. Try the following experiment. First enter Basic from the main menu and then press the label key.

You'll see the key menu displayed on the label line. Now clear the label line by pressing the label key again. Finally, type POKE 64173,0 and press the enter key. Pressing the label key now has no effect. If you type

POKE 64173,255

things return to normal.

Many of the pointers point to addresses, line numbers, and other 16-bit integers. The Model 100 stores such addresses with the low byte of the integer followed by the high byte. For instance, the computer stores the line number where a break occurred in locations 64426-64427. To determine the line number just type  $\text{PRINT PEEK}(64426) + 256 * \text{PEEK}(64427)$ .

To determine the starting location of the currently active program, type  $\text{PRINT PEEK}(63100) + 256 * \text{PEEK}(63101)$ .

### Floating Point Operations

In order to use the floating point routines, you must first place the proper values in the floating point accumulators. For instance, to find the square root of 2 using a call to location 12378, you must place 2 in the primary floating point accumulator FAC1 prior to the call. After the call, the value of the square root of 2 resides in FAC1, and you must move it to the proper variable.

Table 1 continued

|      |                                                                                                                                                                                             |      |                                                                                           |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------------|
|      | the location of the line, and HL the location of the next line. The carry flag is used to indicate if the line exists.                                                                      | 2902 | PRINT                                                                                     |
| 1606 | Tokenize BASIC text.                                                                                                                                                                        | 3073 | TAB(                                                                                      |
| 1830 | FOR                                                                                                                                                                                         | 3141 | LINE (general)                                                                            |
| 1899 | TO                                                                                                                                                                                          | 3152 | LINE INPUT                                                                                |
| 2112 | BASIC command dispatcher; accumulator holds token.                                                                                                                                          | 3188 | '?Redo from start'                                                                        |
| 2136 | Set pointer to BASIC text.                                                                                                                                                                  | 3225 | INPUT #                                                                                   |
| 2162 | DEF (general)                                                                                                                                                                               | 3235 | INPUT                                                                                     |
| 2177 | DEFDBL                                                                                                                                                                                      | 3289 | READ                                                                                      |
| 2182 | DEFINT                                                                                                                                                                                      | 3442 | 'Extra ignored'                                                                           |
| 2198 | DEFSNG                                                                                                                                                                                      | 3625 | >,=,<                                                                                     |
| 2207 | DEFSTR                                                                                                                                                                                      | 3926 | ERL                                                                                       |
| 2267 | FC error.                                                                                                                                                                                   | 3966 | VARPTR                                                                                    |
| 2283 | Used by GOTO, GOSUB, and RUN to determine the 2-byte value of the line (in ASCII) to branch to. The HL register points to the ASCII string of digits, the 16-bit integer is returned in DE. | 4072 | If character in M is lowercase, the upper case equivalent is returned in the accumulator. |
| 2319 | RUN                                                                                                                                                                                         | 4236 | OR                                                                                        |
| 2334 | GOSUB                                                                                                                                                                                       | 4247 | AND                                                                                       |
| 2358 | GOTO (contains a bug!)                                                                                                                                                                      | 4258 | XOR                                                                                       |
| 2381 | Jump here if UL error.                                                                                                                                                                      | 4269 | EQV                                                                                       |
| 2406 | RETURN                                                                                                                                                                                      | 4277 | IMP                                                                                       |
| 2462 | DATA                                                                                                                                                                                        | 4296 | LPOS                                                                                      |
| 2464 | REM,ELSE                                                                                                                                                                                    | 4302 | POS                                                                                       |
| 2499 | LET                                                                                                                                                                                         | 4305 | Store accumulator as low byte of 16-bit integer.                                          |
| 2607 | ON (general)                                                                                                                                                                                | 4352 | INP                                                                                       |
| 2612 | ON ERROR                                                                                                                                                                                    | 4364 | OUT                                                                                       |
| 2736 | RESUME                                                                                                                                                                                      | 4398 | ASCII text (string/variable) converted to integer.                                        |
| 2831 | ERROR                                                                                                                                                                                       | 4411 | LLIST                                                                                     |
| 2842 | IF                                                                                                                                                                                          | 4416 | LIST                                                                                      |
| 2894 | LPRINT                                                                                                                                                                                      | 4514 | PRINTs from buffer until 0 byte is encountered.                                           |
|      |                                                                                                                                                                                             | 4522 | Put data into buffer until a 0 byte is reached.                                           |
|      |                                                                                                                                                                                             | 4740 | PEEK                                                                                      |
|      |                                                                                                                                                                                             | 4747 | POKE                                                                                      |
|      |                                                                                                                                                                                             | 4811 | Saves registers, waits for                                                                |

Table 1 continued



Table 1 continued

|      |                                                                                       |
|------|---------------------------------------------------------------------------------------|
|      | character of input, and restores registers.                                           |
| 4848 | PASTE.                                                                                |
| 5029 | Toggles the label line.                                                               |
| 5138 | Break routine.                                                                        |
| 5145 | POWER                                                                                 |
| 5169 | Power off /Returns to program on power-up.                                            |
| 5201 | Power off                                                                             |
| 5209 | POWER CONT                                                                            |
| 5225 | Sets power-down values.                                                               |
| 5288 | Preliminary tape I/O routine.                                                         |
| 5290 | Called at the end of tape I/O.                                                        |
| 5296 | Returns byte from tape in accumulator.                                                |
| 5313 | Sends byte in accumulator to tape.                                                    |
| 6281 | EOF                                                                                   |
| 6404 | TIMES                                                                                 |
| 6436 | DATE\$                                                                                |
| 6485 | DAY\$                                                                                 |
| 6520 | ASCII of days stored here.                                                            |
| 6553 | Converts byte pointed to by DE to an ASCII digit. Result placed in M.                 |
| 6571 | TIMES (as command/assignment).                                                        |
| 6589 | DATE\$ (as command/assignment).                                                       |
| 6641 | DAY\$ (as command/assignment).                                                        |
| 6547 | MAX RAM                                                                               |
| 6776 | IPL                                                                                   |
| 6814 | COM,MDM                                                                               |
| 6851 | KEY ON/OFF/STOP                                                                       |
| 6927 | ON TIMES                                                                              |
| 7096 | KEY (general)                                                                         |
| 7101 | KEY LIST                                                                              |
| 7136 | PRINTS B PRINTable-characters starting at address in HL.                              |
| 7182 | Defines function key.<br>HL points to string, accumulator holds key number minus one. |
| 7255 | PSET                                                                                  |
| 7270 | PRESET                                                                                |
| 7277 | Line drawing routine.                                                                 |
| 7519 | Subroutine for PRINT @                                                                |
| 7568 | CSRLIN                                                                                |
| 7579 | MAX (general).                                                                        |
| 7602 | MAX FILES                                                                             |
| 7609 | HIMEM                                                                                 |
| 7619 | WIDTH                                                                                 |
| 7621 | SOUND (general)                                                                       |
| 7653 | SOUND OFF                                                                             |
| 7654 | SOUND ON                                                                              |
| 7660 | MOTOR                                                                                 |
| 7667 | Turns motor on.                                                                       |
| 7669 | Turns motor off.                                                                      |
| 7674 | CALL                                                                                  |
| 7714 | SCREEN                                                                                |
| 7774 | LCOPY                                                                                 |
| 7994 | FILES                                                                                 |
| 8081 | KILL                                                                                  |
| 8247 | NAME                                                                                  |
| 8446 | NEW                                                                                   |
| 8832 | CSAVE (general)                                                                       |
| 8856 | CSAVEs current BASIC program.                                                         |
| 8889 | Sends the DE bytes starting at location in HL to tape.                                |
| 8908 | SAVEM                                                                                 |
| 8925 | CSAVEM                                                                                |
| 9079 | CLOAD (general)                                                                       |

Table 1 continued

The routines that you need for these actions all appear in the map. You can use the machine code below to find the square root of the variable pointed to by the HL register pair.

```

PUSH H      ; We will need this address later.
CALL 12740  ; Put the variable into FAC1.
CALL 12378  ; Take the square root.
POP H       ; I told you we needed that address.
CALL 12746  ; Replace the value by its square root.
RET         ; That's all.
    
```

Access the program above from Basic by using the next program, being sure to protect it first by the command CLEAR 256,61999. This prevents Basic from destroying the machine code starting at location 62000.

```

5 GOSUB 100
10 INPUT "X ";X
20 A = VARPTR(X) + 65536
30 CALL 62000,0,A
40 PRINT X
50 END
100 FOR I=0 TO 11
110 READ V
120 POKE 62000+I,V
130 NEXT I
140 RETURN
150 DATA 229,205,196,49,205,90,48,225,205,202,49,201
    
```

This program assumes that you've placed the machine-language routine at location 62000. Line 20 determines the memory location of the variable X, and line 30 passes this address to the register pair HL before executing the routine at 62000.

Lines 100-150 POKE the machine code described earlier into memory.

Now run the program, and you'll see that the value of the variable X is replaced by its square root.

Some of the other floating point functions seem to require the setting of the variable type flag prior to a Call.

### Interrupts and the RST Instructions

Like the 8080, the Model 100's 8085 chip contains eight restart instructions that are effectively 1-byte Call instructions. (See Table 1.)

Unlike the 8080, the 8085 has several interrupt routines known by the mnemonics RST 5.5, RST 6.5, RST 7.5, and TRAP. The addresses of each of these resides in the low bytes of ROM. However, the Model 100 allows a user to intercept these routines by placing a Jump instruction in a particular location in RAM. These interface vectors are documented in the map.

### The Basic Commands

The memory map contains the addresses of most of the Basic commands and functions. The important thing to realize is that most Basic commands expect the accumulator to contain the next byte of Basic text upon entry to the routine. Some commands, such as Beep and End, aren't modified by additional text. A simple call to the addresses of these routines has the desired effect. On the other hand, the actions of some commands depend upon the text that follows them. For example, consider the Key command. When this statement appears in a Basic program or in immediate mode, several things may

for the TRS-80 from Micro-Mega

## The Original GREEN-SCREEN



The eye-pleasing Green-Screen fits over the front of your TRS-80 Video Display and gives you improved contrast with reduced glare. You get bright luminous green characters and graphics like those featured by more expensive CRT units.

Don't confuse the Original Green-Screen with a piece of thin film stuck to the face of your video tube, such as that advertised by others. The Original Green-Screen is mounted in a full frame perfectly matched to the color and texture of the TRS-80 Video Display. It is attached with adhesive strips which do not mar your unit in any way.

The full frame design of the Original Green-Screen "squares off" the face of your video display and greatly improves the overall appearance of your system.

(Specify whether for Model I, Model III, or Model IV)

**THE GREEN-SCREEN**..... \$15.95  
Add \$1.50 for postage and handling.

Terms: Check or money order, no CODs or credit cards, please. Add amount shown for postage and handling to price of the item. All items shipped within 48 hours by first class or priority mail. Virginia residents, add 4% sales tax. ✓162

Micro-Mega · P.O. Box 6265 · Arlington, Va 22206

## MODEL 100 OWNERS

### Tired of Waiting for Slow Cassettes?

HOLMES ENG. has solved your problem with -- the PMD-100 Portable Micro Drive

The PMD-100 is a HIGH SPEED Storage System for SAVING and LOADING programs and files on the M100 Microcomputer.

USER FRIENDLY menu driven software and a Rom based operating system make the PMD as fast and easy to use as a floppy disk.

#### KEY FEATURES INCLUDE:

- Up to 70K Storage per wafer.
- Fast Data Storage/Retrieval (Nearly as fast as disks)
- Portable & light weight
- ROM based operating system
- Connects to RS232 Port
- Rechargeable Battery
- Replaceable wafer tapes
- 16K Ram Buffer

**\$349.95**  
**DRIVE INCLUDES:**  
Wafer tapes, battery charger, coiled cable, operating instructions

**IM-100**  
**8K RAM MODULES**  
**\$75.00 each**  
**\$70.00 each (2 or more)**  
Up to 32K upgrade  
Plugs into M100 sockets  
Easy to install

**HOLMES**  
ENGINEERING, INC.

Bus. Hours: 8:5-30 (MST) Mon.-Fri.  
10:2:00 (MST) Sat. (sales only) ✓153

5175 Green Pine Drive Salt Lake City, Utah 84123  
(801) 261-5652 or 24 Hr. BBS (801) 263-1103



Call For Dealer Nearest You  
Dealer Inquiries Invited  
Full warranty on all products  
Add shipping/handling \$10.00  
U.S. and Canada. All others add  
15%. Prices subject to change  
without notice.

## U-Notes

follow it. For instance, KEY ON, KEY OFF, KEY LIST, or KEY 5, "PRINT" are all possible. At the time the program executes the command, the computer has already tokenized the keywords, and the Key routine expects to find the appropriate token or text in the accumulator. The token for ON is 151, and so typing CALL 7096,151 produces the effect of KEY ON. Similarly, since the token for LIST is 165, CALL 7096,165 will result in the same action as if you had entered KEY LIST. Unfortunately, you'll find that not all calls to Basic command locations respond as expected. So be cautious when experimenting with these addresses.

### Using the Memory Map

The examples given earlier should help you take advantage of much of this information. You cannot use all of the routines in the ROM directly, however. Many of them, like the floating point operations, require setting up special conditions prior to the call. And, although I would strongly urge you to experiment with your computer, be sure you have saved any important data or programs on tape first. An inappropriate call or POKE can easily garble a program or cause the computer to hang. Of course, you will not damage the computer by such a crash, but you may be unable to restart the computer without using the memory kill button on the bottom of the computer.

### About the Map

Notice that items enclosed in single quotes represent actual ASCII text. For instance, the word 'Error' is stored in memory starting at location 1003. The letters H,L,D,E,B,C, and A refer to registers in the 8085. ■

David P. Sumner can be reached at 1009 Walters Lane, Columbia, SC 29209.

Table 1 continued

|       |                                                                             |
|-------|-----------------------------------------------------------------------------|
| 9235  | Gets DE bytes from tape and stores them at the location pointed to by HL.   |
| 9302  | CLOAD?                                                                      |
| 9345  | 'Verify failed'                                                             |
| 9361  | LOADM, RUNM                                                                 |
| 9383  | CLOADM                                                                      |
| 9538  | Moves B bytes from address in HL to address in DE in an increasingz manner. |
| 9587  | CLOADM?                                                                     |
| 9685  | 'Top End Exe'                                                               |
| 9697  | 'Exe:'                                                                      |
| 9982  | 'Found'                                                                     |
| 9989  | 'Skip'                                                                      |
| 10042 | STR\$                                                                       |
| 10161 | PRINTs message pointed to by HL. Message ends in quote or 0-byte            |
| 10444 | String addition.                                                            |
| 10508 | Moves L bytes from address in BC to address in DE (increasing).             |
| 10563 | LEN                                                                         |
| 10575 | ASC                                                                         |
| 10597 | CHR\$                                                                       |
| 10605 | STRING\$                                                                    |
| 10638 | SPACE\$                                                                     |

Table 1 continued

Table 1 continued

|       |                                                                                      |  |  |
|-------|--------------------------------------------------------------------------------------|--|--|
| 10667 | LEFT\$                                                                               |  |  |
| 10716 | RIGHT\$                                                                              |  |  |
| 10726 | MID\$                                                                                |  |  |
| 10759 | VAL                                                                                  |  |  |
| 10807 | INSTR                                                                                |  |  |
| 11084 | FRE                                                                                  |  |  |
| 11113 | FAC1 <-- FAC1-FAC2.                                                                  |  |  |
| 11128 | FAC1 <-- FAC1+FAC2.                                                                  |  |  |
| 11519 | FAC1 <-- FAC1*FAC2.                                                                  |  |  |
| 11719 | FAC1 <-- FAC1/FAC2.                                                                  |  |  |
| 12006 | Moves C bytes starting from address in HL to address in DE in a decreasing manner.   |  |  |
| 12015 | COS                                                                                  |  |  |
| 12041 | SIN                                                                                  |  |  |
| 12120 | TAN                                                                                  |  |  |
| 12145 | ATN                                                                                  |  |  |
| 12239 | LOG                                                                                  |  |  |
| 12378 | SQR                                                                                  |  |  |
| 12452 | EXP                                                                                  |  |  |
| 12606 | RND                                                                                  |  |  |
| 12686 | MOVE FAC1 to 64633-64640.                                                            |  |  |
| 12692 | Adds memory to FAC1.                                                                 |  |  |
| 12698 | Subtracts memory from FAC1.                                                          |  |  |
| 12704 | FAC1 <-- FAC1*FAC1 (squares FAC1)                                                    |  |  |
| 12707 | Multiplies memory and FAC1.                                                          |  |  |
| 12725 | Moves FAC1 to FAC2.                                                                  |  |  |
| 12728 | Moves number at address HL to FAC2.                                                  |  |  |
| 12737 | Moves FAC2 to FAC1.                                                                  |  |  |
| 12740 | Moves number at address HL to FAC1.                                                  |  |  |
| 12746 | Moves FAC1 to address in HL.                                                         |  |  |
| 12852 | Saves FAC1 on the stack.                                                             |  |  |
| 12892 | Floating point constants.                                                            |  |  |
| 12942 | .25                                                                                  |  |  |
| 12974 | pi/2                                                                                 |  |  |
| 12990 | Square root of 3.                                                                    |  |  |
| 13015 | 1                                                                                    |  |  |
| 13203 | 2*pi.                                                                                |  |  |
| 13298 | ABS                                                                                  |  |  |
| 13309 | Negate FAC1.                                                                         |  |  |
| 13319 | SGN                                                                                  |  |  |
| 13417 | Moves B bytes of memory from address in DE to address in HL in an increasing manner. |  |  |
| 13426 | Moves B bytes of memory from address in DE to address in HL in a decreasing manner.  |  |  |
| 13569 | CINT                                                                                 |  |  |
| 13610 | CSNG                                                                                 |  |  |
| 13754 | COBL                                                                                 |  |  |
| 13893 | FIX                                                                                  |  |  |
| 13908 | INT                                                                                  |  |  |
| 14804 | PRINTs the 16-bit integer stored in the register pair HL.                            |  |  |
| 16288 | TIME\$ ON                                                                            |  |  |
| 16306 | TIME\$ OFF                                                                           |  |  |
| 16313 | TIME\$ STOP                                                                          |  |  |
| 16393 | Clears COM and TIME\$ locations. Clears KEY definitions and sets 63060 to 0          |  |  |

Table 1 continued

**INTRODUCING TELE-STOCK™**  
for TRS-80\* Mod 100

**STOCK MARKET DECISION ASSISTANCE**  
12 to 24 hours ahead of the newspaper at less than the cost of the newspaper. Your Model 100 automatically calls the Dow Jones News Service immediately following the close of the market each day, retrieves the market activity for your investments (it takes about 1 minute and cost 15-30 cents), then displays and prints for each investment:

- **Today's Activity**
- **45 day historical graph showing HIGH, LOW, CLOSE, VOLUME**
- **Return on Investment**

The cost of the daily service, this software and even your TRS-80 may be tax deductible when used for this purpose (consult your tax advisor for details).

**IMPROVE YOUR INVESTMENT PERFORMANCE** for only \$59.95 (Ca. residents add 6% tax) on cassette with manual. Send check, money order or VISA/MC number with expiration date to:

**TELESOFT™**  
P.O. Box 6398  
Thousand Oaks, Ca. 91359 ✓15  
or call 805/499-6271

\*Requires 24K Model 100 with direct connect or acoustic cable. \*TRS-80 is a trademark of the Tandy Corporation.

**IMPACTOR**  
BY ROB GOLDSTONE

YOU ARE THE COMMANDER OF THE IMPACTOR, A VITAL RESEARCH BASE WITHIN THE PLANET THORALL'S INNER CORE. IT IS UP TO YOU TO REPEL THE ENEMY INTO THE OUTER CORE, BEFORE THEY CRUSH THE IMPACTOR. BUT BEWARE OF THE ETERNAL BOUNCING OOT. IMPACTOR IS A HIGH SPEED, MACHINE LANGUAGE GAME.

MODEL I & III 16K TAPE.....\$11.95  
MODEL I & III 32K DISK.....\$13.95

KOPY BY BILL ADDISON:  
KOPY IS A MACHINE LANGUAGE PROGRAM THAT WILL ALLOW YOU TO COPY MOST ANY SINGLE LOAD MACHINE LANGUAGE PROGRAM, ON CASSETTE. KOPY WILL ALLOW THE MODEL III USER TO COPY A PROGRAM IN AT 500/1500 BAUD, AND COPY IT BACK OUT AT 500/1500 BAUD. KOPY SHOWS FILENAME, STARTING ADDR., TOTAL BYTE COUNT, ENTER ADDR., AND ALLOWS FOR MULTIPLE DUMPS.

MODEL I & III 16K TAPE.....\$9.95

**THE SOFTWARE EXCHANGE**  
1615 COMPTON ROAD  
CLEVELAND HTS., OHIO 44116 ✓28

ADD \$1.00 PER ITEM ORDERED FOR SAH  
ALL ORDERS SHIPPED FIRST CLASS MAIL  
INCLUDE CHECK OR M.O. \* COD: ADD \$2.00  
OHIO RESIDENTS ADD SALES TAX  
- DEALERS INVITED -

**LARGE CAPACITY ACCOUNTING PROGRAMS**

**ACCOUNTS RECEIVABLE**  
5000+ CUSTOMERS  
15000+ TRANSACTIONS  
BALANCE FORWARD, 99 TRANSACT. CODES  
30-60-90-120 AGED, STATEMENTS SHOW DATE/INV#/DESCRIP/AMT (WITH AGEING). SELECTIVE FINANCE CHARGES & RATES. FAST ENTRY, POSTING W/AUDIT REPORT. SUB-ACCTS. % OF CREDIT LIMIT. DATE OF LAST PAYMENT. SALES ANALYSIS, SPECIAL 90 DAY ACCOUNTS, LABELS.

**GENERAL LEDGER**  
400+ ACCOUNTS  
5000+ TRANSACTIONS/MONTH  
NO OTHER SYSTEM OFFERS...  
† REPORT FLEXIBILITY/CAPACITY  
† DEPARTMENT P & L (UP TO 9)  
† USE 100+ SUB-TOTALS  
† STATEMENT OF CHANGES (ASSETS)  
† PERCENT P&L + ACCOUNT DETAIL

**149.95 Each ★ Both for 199.95**

Both Programs...  
- Run on Model 1, 3/4, LNW, LOBO  
- Require Only 2 Drives  
- Come with Sample Data  
- ARE AVAILABLE for Hard Disk  
- 100% Written in BASIC  
- Require 80 Column Printer ✓355

Ask About Our Other Programs

**H. D. P.**  
2059 West Lincoln  
Oroville, CA 95965  
916 - 533-5992

Add \$3.00 Shipping & Handling  
Postman Collects Cash If Sent COD  
VISA or M/C

**LARGE CAPACITY SYSTEM LARGE CAPACITY SYSTEM LARGE CAPACITY SYSTEM LARGE CAPACITY SYSTEM LARGE CAPACITY SYSTEM**

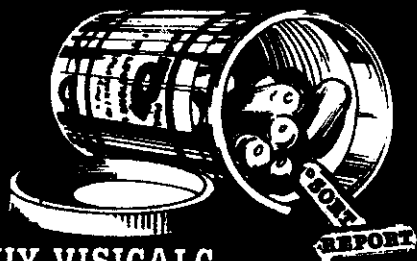
Table 1 continued

|       |                                                                 |                                                                         |
|-------|-----------------------------------------------------------------|-------------------------------------------------------------------------|
| 15758 |                                                                 | printer.                                                                |
| 16511 | RESTORE                                                         | 19360 Sends carriage return to printer                                  |
| 16538 | STOP                                                            | 19434 INKEY\$                                                           |
| 16543 | END                                                             | 19659 OPEN                                                              |
| 16602 | CONT                                                            | 19824 LOAD                                                              |
| 16625 | Determines if HL points to an alphabetic character.             | 19825 MERGE                                                             |
| 16626 | Determines if the accumulator contains an alphabetic character. | 19919 SAVE                                                              |
| 16633 | CLEAR                                                           | 20008 CLOSE                                                             |
| 16756 | NEXT                                                            | 20110 INPUT\$                                                           |
| 16930 | PRINTs CHR\$(13)+CHR\$(10)                                      | 20591 LFILES                                                            |
| 16937 | BEFP                                                            | 20725 'CRT CAS COM WAND LPT MDM RAM'                                    |
| 16941 | PRINTs CHR\$(11)                                                | 20806 TELCOM program.                                                   |
| 16945 | CLS                                                             | 20558 Enter here or one of 20561, 20563...20579 in case of file errors. |
| 17001 | Turns on reverse video.                                         | 20593 DSKO\$                                                            |
| 17006 | Turns off reverse video.                                        | 20595 DSKI\$                                                            |
| 17008 | PRINTs CHR\$(27)+CHR\$ of accumulator.                          | 20860 'Telcom'                                                          |
| 17034 | Clears the label line.                                          | 20869 'STAT'                                                            |
| 17064 | PRINTs label line.                                              | 20874 'TERM CALL RFIND MENU'                                            |
| 17994 | Puts input data into input buffer.                              | 20900 'Find Call Stat Term Menu'                                        |
| 18315 | DIM                                                             | 20906 'x pps'                                                           |
| 18320 | Get variable name and type.                                     | 21060 'Calling'                                                         |
| 18833 | USING                                                           | 21890 'Full Half Echo'                                                  |
| 19268 | PRINTs ASCII character in accumulator.                          | 21909 'Wait'                                                            |
| 19322 | Sends byte in accumulator to the                                | 22353 'File to upload File to download'                                 |
|       |                                                                 | 22385 'aborted'                                                         |
|       |                                                                 | 22396 'No file'                                                         |
|       |                                                                 | 22406 'Disconnect'                                                      |
|       |                                                                 | 22417 PRINTs message. Sends a cr if not at start of a line.             |

Table 1 continued

# This Prescription Will save the life of your VISICALC !

— Don't Buy a New Machine to Run it On  
— Don't Buy a New Spreadsheet Program



**BUY VISICALC ENHANCERS from SOLUTIONS**

October 1983 Popular Computing

InfoWorld September 5, 1983

**SYMPTOM:** Desperate need to sort the rows or columns of a VisiCalc spreadsheet

**R<sub>x</sub>:** SORT™ from SOLUTIONS

**SYMPTOM:** Bleary Eyes from trying to print a good looking report from VisiCalc

**R<sub>x</sub>:** REPORT™ from SOLUTIONS

## SORT

- Sort the rows or columns of your VisiCalc spreadsheet
- Formulas and values move with each row or column
- Sort alphabetically or numerically
- Sort in ascending or descending order
- Use up to four additional keys to break ties or specify secondary sorts

Both are Available for TRS-80® Models III/II/16, III, Apple® II +, IIE, III and for the IBM PC™ and compatibles.

## REPORT

- Print with variable width columns
- Segment large spreadsheets into multipage reports
- Repeat columns and rows and multipage reports
- Eliminate unwanted columns
- Align decimal points
- Center or justify labels and values
- Add titles and page numbers to your report
- Write reports to disk for later printing or transmission

### Solutions, Inc.

13 State Street, Box 989, Montpelier, Vermont, 05602. Telephone (802) 229-0368

Send me a Free Brochure

PLEASE SEND ME THE FOLLOWING SOFTWARE

SORT \$89.00     REPORT \$79.00

Send me BOTH SORT and REPORT for \$158.00  
ALSO AVAILABLE: (SAVE \$10)

Dow Jones Spread Sheet Link \$279.00

GL Bridge \$195.00

Include \$4.00 for US shipping and handling.

Payment Method Preferred:

Check or Money Order Enclosed

Charge My Credit Card     Visa     MasterCard

Card no. \_\_\_\_\_ Exp. Date. \_\_\_\_/\_\_\_\_/\_\_\_\_

Signature \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_

State \_\_\_\_\_ Zip \_\_\_\_\_



Table 1 continued

|       |                                                                                    |             |                                                                                       |
|-------|------------------------------------------------------------------------------------|-------------|---------------------------------------------------------------------------------------|
| 22423 | MENU                                                                               | 27611       | Moves BC bytes from address in HL to address in DE increasing.                        |
| 22574 | PRINTs number of free bytes.                                                       | 27636       | 'BASIC TEXT TELCOM ADDRSS SCHEDL Suzuki Hayashi'                                      |
| 22580 | PRINTs 'Select:' and responds appropriately to input.                              | 27721       | BASIC (as called from MENU)                                                           |
| 23128 | PRINTs starting at address in HL until a zero byte is reached.                     | 27795       | Temporarily saves the function keys.                                                  |
| 23164 | Sets up function keys.                                                             | 27804       | Restores the function keys.                                                           |
| 23273 | Months stored in ASCII.                                                            | 29156       | Adds a character to the keyboard buffer.                                              |
| 23312 | 'Microsoft'                                                                        | 29250       | Returns ASCII of keypress in the accumulator; does not wait-returns 0 if no keypress. |
| 23332 | 'Select:'                                                                          | 29381       | SOUND routine. DE contains pitch, and B contains duration.                            |
| 23366 | Original key definitions stored here. FILES LOAD " SAVE " RUN LIST MENU            | 29772       | Turns on the pixel (x,y) where D contains x and E contains y.                         |
| 23400 | ADDRSS (program)                                                                   | 29773       | Turns off the pixel (x,y) where D contains x and E contains y.                        |
| 23407 | SCHEDL (program)                                                                   | 30326       | Toggles the speaker.                                                                  |
| 23450 | PRINTs 'Not found press space bar for menu'                                        | 30306       | Equivalent to BEEP.                                                                   |
| 23758 | ADDRS.DO                                                                           | 30481       | Character set 5-bytes each.                                                           |
| 23767 | 'Not found'                                                                        | 31729       | Keyboard matrix                                                                       |
| 23852 | 'Call'                                                                             | 32051       | Initialization                                                                        |
| 23920 | PRINTs date and time at top of screen and updates it until a key is pressed.       | 32231       | Initialize pointers (cold start)                                                      |
| 24046 | TEXT (program)                                                                     | 32422       | PRINTs TRS-80 logo.                                                                   |
| 24051 | Requests a file to edit.                                                           | 32428       | PRINTs number of free bytes.                                                          |
| 24085 | 'File to edit'                                                                     | 32523       | MAX FILES=                                                                            |
| 24106 | 'FIND LOAD SAVE COPY CUT SEL MENU'                                                 | 32664       | ' bytes free'                                                                         |
| 24145 | EDIT (program)                                                                     | 32676       | TRS-80 logo stored in ASCII.                                                          |
| 24367 | Waits for a space keypress and then returns.                                       | 40960-62959 | User RAM in 24K machine.                                                              |
| 24376 | 'Text ill-formed'                                                                  | 62964+      | HIMEM value.                                                                          |
| 24395 | 'Press space bar for TEXT'                                                         | 62966       | Code called at 32145 and 32197.                                                       |
| 24753 | 'Memory full'                                                                      | 62969       | RAM vector for RST 5.5.                                                               |
| 26051 | Moves memory starting at address in HL to address in DE until a 0 byte is reached. | 62972       | RAM vector for RST 6.5.                                                               |
| 26062 | 'No Match'                                                                         | 62975       | RAM vector for RST 7.5.                                                               |
| 26071 | 'String'                                                                           | 62978       | JMP 5169                                                                              |
| 26380 | 'Width'                                                                            | 62981       | Code called at 32108 and 32292.                                                       |
| 26421 | 'Save to:'                                                                         | 62991       | Code called at 896.                                                                   |
| 26579 | 'Load from:'                                                                       | 63012       | Code                                                                                  |
|       |                                                                                    | 63024       | 8-bytes KEY ON/OFF flags; ON=1                                                        |

Table 1 continued

## MODEL 100 ASSEMBLER

**\$22.95**

Requires 16K Ram  
Assemble to Tape or Ram  
Available on Tape or Disk  
Extensive Documentation

**CUSTOM SOFTWARE**  
605 North C, # 2  
Wellington, KS 67152  
316/326-6197 ✓ 49

### 8K MEMORY MODULES FOR THE MODEL 100 PORTABLE COMPUTER

**\$85 each**  
**2 or more \$80 each**

Exact Replacement for RS Memory  
Easy to Install  
Warranty Included  
VISA and MC Accepted

**iota systems**

1690 Day Valley Road  
Aptos, CA 95003  
(408) 684-0482

Prices Subject To Change Without Notice  
TRS-80 is a trademark of Tandy Corporation. ✓ 400

## SCRIBE REAL WORD PROCESSING CAPABILITY

for the  
Radio Shack TRS 80

## MODEL 100

Portable Computer  
USES ONLY 2K RAM STORAGE

**\$24.95**

PLUS \$2.00 HANDLING

Chattanooga Systems Associates  
P.O. Box 22261  
Chattanooga, TN 37422

**DEALERS INQUIRE** ✓ 225

Table 1 continued

|        |                                                                                                                                |
|--------|--------------------------------------------------------------------------------------------------------------------------------|
|        | OFF=0                                                                                                                          |
| 63033  | Vertical Print position 0 to 7.                                                                                                |
| 63034  | Next horizontal print position 0-39.                                                                                           |
| 63035  | Number of active lines 0-8.                                                                                                    |
| 63036  | Number of active columns 0-40.                                                                                                 |
| 63037  | Label line flag; 0=unused, 255=used.                                                                                           |
| 63038  | Inhibits return to first line if not zero.                                                                                     |
| 63040  | Cursor line.                                                                                                                   |
| 63041  | Horizontal print position.                                                                                                     |
| 63048  | Reverse video if not zero.                                                                                                     |
| 63054  | x-pixel set.                                                                                                                   |
| 63055  | y-pixel set.                                                                                                                   |
| 63063  | Power-down value (constant).                                                                                                   |
| 63070  | Code (to 63078)                                                                                                                |
| 63079  | OUT/INP self-modifying code.                                                                                                   |
| 63090  | Error code                                                                                                                     |
| 63092  | Value of LPOS                                                                                                                  |
| 63093  | Output flag; 0=display, 1=printer.                                                                                             |
| 63096+ | Top of available RAM.                                                                                                          |
| 63098+ | Current BASIC line number; 65535 stored here if no program is running.                                                         |
| 63100+ | Start of current BASIC program text.                                                                                           |
| 63104  | End of statement marker (: OR 0)                                                                                               |
| 63105  | Multi-purpose buffer area. Tokenized text starts at 63105. Input buffer starts at 63109 and extends to 63362.                  |
| 63368  | Value of POS                                                                                                                   |
| 63369  | Function key definitions currently active. (extends to 63497)                                                                  |
| 63498  | Function key definitions used by BASIC. (extends to 63626)                                                                     |
| 63628+ | Pointer to start of PASTE text.                                                                                                |
| 63639  | Start of work area.                                                                                                            |
| 63785  | Day of the month low digit here, high digit in 63786.                                                                          |
| 63787  | Current day of the week (e.g. 3=wed.)                                                                                          |
| 63788  | Current month - decimal 1 to 12.                                                                                               |
| 63789  | Current year stored with low decimal value here and*the high decimal value in 63790                                            |
| 63791  | Timer; decreases from 125 to 0.                                                                                                |
| 63792  | Timer decreases from 12 to 0.                                                                                                  |
| 63793  | Power-down countdown value (varies).                                                                                           |
| 63795  | The computer stores the current time starting here with the low digit of the number of seconds. A numerical value - not ASCII. |
| 63796  | The high digit of number of seconds.                                                                                           |
| 63797  | Low digit of number of minutes.                                                                                                |
| 63798  | Time and date continues here with one decimal value per location.                                                              |
| 63805  | 6-byte value of TIME\$ for ON TIME\$ statement (in reverse order).                                                             |
| 63812  | COM ON/OFF flag.                                                                                                               |
| 63813+ | Address of COM ON routine.                                                                                                     |
| 63815  | TIME\$ ON/OFF flag.                                                                                                            |
| 63816+ | Address of TIME\$ ON routine.                                                                                                  |
| 63818  | ON/OFF flag and routine address                                                                                                |

Table 1 continued

# Continuous Checks, Statements, and Invoices for Desk-Top Computers

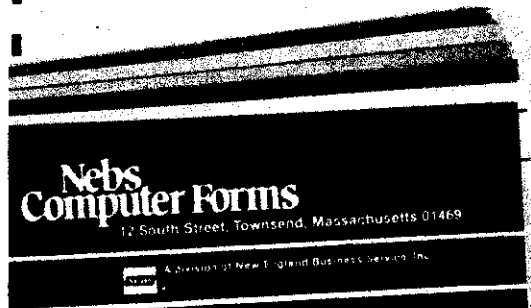
- Compatible with software from over 300 sources. Or program to NEBS standard forms yourself.
- Continuous Micro-Perf™ Letterheads and matching continuous Envelopes provide a clean, trim look.
- Also, diskettes, continuous labels, other supplies and accessories.
- Our policy is to process forms printed with your name within 6 working days. Then ship direct to you (We pay shipping charges on prepaid orders).

**QUALITY PRODUCTS  
SMALL QUANTITIES AT LOW PRICES  
MONEY-BACK GUARANTEE**

**FREE Full-color, catalog**  
fast service by mail or phone  
**TOLL FREE 1 + 800-225-9550**  
(Mass. residents 1 + 800-922-8560)

|                                          |                                     |                                             |
|------------------------------------------|-------------------------------------|---------------------------------------------|
| NAME/TITLE                               |                                     | PHONE                                       |
| COMPANY                                  |                                     |                                             |
| STREET                                   |                                     |                                             |
| CITY, STATE, ZIP                         |                                     |                                             |
| SOFTWARE BRAND                           | PACKAGE #                           | <input type="checkbox"/> DO OWN PROGRAMMING |
| COMPUTER BRAND                           | MODEL                               | PLAN TO PURCHASE WITHIN:                    |
| YOUR LINE OF BUSINESS                    |                                     | NUMBER OF EMPLOYEES                         |
| I MOST OFTEN USE MY COMPUTER FOR:        |                                     |                                             |
| <input type="checkbox"/> WORD PROCESSING | <input type="checkbox"/> ACCOUNTING | <input type="checkbox"/> OTHER              |

15111



533

Table 1 continued

- of function keys; 3 bytes per key. (extends to 63841)
- 63842 Files in the format: address, 6-character name, 2-byte extension
- 63898+ Address of BASIC program that has not been saved to RAM. (Suzuki)
- 63909+ Address of Hayashi, points to the end of documents.
- 64173 Label line enable flag; enabled if not zero.
- 64175 Name of IPL program.
- 64190+ Used for temporary storage of stack pointer.
- 64208+ Length of CLOAded/CSAVEd program
- 64357 BASIC variable type.
- 64404+ Line number of active DATA statement.
- 64409+ Location of BASIC variable for assignment statement.
- 64411+ Start of current BASIC statement
- 64413+ 2 less than value in 63096-63097
- 64415+ Line where error occurred.
- 64417+ Most recently entered/listed line
- 64419+ Location of statement where error occurred.
- 64421+ Location of ON ERROR GOTO line.
- 64423 Error status flag.
- 64424+ End of BASIC expression.
- 64426+ Line where Break occurred.
- 64428+ Location after error (contains 0 or 58).
- 64430+ Start of documents.
- 64434+ Start of variables.
- 64440+ POINTS TO NEXT  
The region from 64536 to 64640 is used for floating point computations.
- 64536 FAC1 (floating point accumulator) 8-bytes.
- 64617 FAC2 (floating point accumulator) 8-bytes.
- 64642 Maxfiles.
- 64659 Name of current BASIC program; 6-bytes.
- 64668 Name of program loaded from tape; 6-bytes.
- 64904 Start of date and time stored in ASCII. Used for Menu display. Not used for TIMES\$.
- 65024 Start of screen memory.
- 65348 Sound flag; on=0, off=175.
- 65349 Cassette on/off flag.
- 65424 Holds the value 2 as long as a noncontrol key is held down.
- 65429 Devoted to the number keys. Also uses location 65430
- 65431 Bits are set here according to which of the following keys are pressed: SPACE,DEL,TAB,ESC,PASTE,LABEL,PRINT,ENTER.
- 65432 Pressing a function key sets the corresponding bit in this location.
- 65441 Behaves like 65432.
- 65442 The following keys set bits in this location: SHIFT,CTRL,GRPH, CODE,NUM,CAPS LOCK.

- 65446 Code (not ASCII) for most recently pressed key.
- 65450 Number of characters in keyboard buffer.
- 65451 Keyboard buffer (32 byte maximum) Odd bytes contain ASCII values. A 255 in an even byte indicates a function key.
- 65515 Used to store 5-byte character code.

| Token | Keyword | Token | Keyword  | Token | Keyword |
|-------|---------|-------|----------|-------|---------|
| 128   | END     | 171   | DATES    | 213   | AND     |
| 129   | FOR     | 172   | DAYS     | 214   | OR      |
| 130   | NEXT    | 173   | COM      | 215   | XOR     |
| 131   | DATA    | 174   | MDM      | 216   | EQV     |
| 132   | INPUT   | 175   | KEY      | 217   | IMP     |
| 133   | DIM     | 176   | CLS      | 218   | MOD     |
| 134   | READ    | 177   | BEEP     | 219   | \       |
| 135   | LET     | 178   | SOUND    | 220   | >       |
| 136   | GOTO    | 179   | LCOPY    | 221   | =       |
| 137   | RUN     | 180   | PSET     | 222   | <       |
| 138   | IF      | 181   | PRESET   | 223   | SGN     |
| 139   | RESTORE | 182   | MOTOR    | 224   | INT     |
| 140   | GOSUB   | 183   | MAX      | 225   | ABS     |
| 141   | RETURN  | 184   | POWER    | 226   | FRE     |
| 142   | REM     | 185   | CALL     | 227   | INP     |
| 143   | STOP    | 186   | MENU     | 228   | LPOS    |
| 144   | WIDTH   | 187   | IPL      | 229   | POS     |
| 145   | ELSE *  | 188   | NAME     | 230   | SQR     |
| 146   | LINE    | 189   | KILL     | 231   | RND     |
| 147   | EDIT    | 190   | SCREEN   | 232   | LOG     |
| 148   | ERROR   | 191   | NEW      | 233   | EXP     |
| 149   | RESUME  | 192   | TAB(     | 234   | COS     |
| 150   | OUT     | 193   | TO       | 235   | SIN     |
| 151   | ON      | 194   | USING    | 236   | TAN     |
| 152   | DSKOS   | 195   | VARPTR   | 237   | ATN     |
| 153   | OPEN    | 196   | ERL      | 238   | PEEK    |
| 154   | CLOSE   | 197   | ERR      | 239   | EOF     |
| 155   | LOAD    | 198   | STRING\$ | 240   | LOC     |
| 156   | MERGE   | 199   | INSTR    | 241   | LOF     |
| 157   | FILES   | 200   | DSKIS    | 242   | CINT    |
| 158   | SAVE    | 201   | INKEY\$  | 243   | CSNG    |
| 159   | LFILES  | 202   | CSRLIN   | 244   | CDBL    |
| 160   | LPRINT  | 203   | OFF      | 245   | FIX     |
| 161   | DEF     | 204   | HIMEM    | 246   | LEN     |
| 162   | POKE    | 205   | THEN     | 247   | STR\$   |
| 163   | PRINT   | 206   | NOT      | 248   | VAL     |
| 164   | CONT    | 207   | STEP     | 249   | ASC     |
| 165   | LIST    | 208   | +        | 250   | CHR\$   |
| 166   | LLIST   | 209   | -        | 251   | SPACE\$ |
| 167   | CLEAR   | 210   | *        | 252   | LEFT\$  |
| 168   | CLOAD   | 211   | /        | 253   | RIGHT\$ |
| 169   | CSAVE   | 212   | ^        | 254   | MID\$   |
| 170   | TIMES   |       |          |       |         |

\* Basic reads ELSE as :LSE, with LSE = 145.

Table 2. Basic keywords and tokens.



## Make Your Own Modem Cable

by Carl Oppedahl

While the Radio Shack modem cable for the Model 100 comes with one free hour each of CompuServe and Dow Jones services, this may be of little use to you if you already subscribe, or if you need more than one modem cable. For these reasons I built my own.

The circuit diagram for the Radio Shack cable appears in Table 3. One end is an exotic 8-pin DIN plug (see Photo 1) whose spacings are slightly different from those of the 8-pin cassette DIN plug. Look closely at the top panel of the Model 100: The difference lies in the placement of pins 6, 7, and 8. In the phone jack pins 7 and 8 sit directly below 1 and 3, while in the cassette jack pins 6, 7, and 8 are somewhat closer together.

Extending from the 8-pin plug are two conventional-looking modular telephone line cords. You would connect the beige cord to the local phone company dial tone, as with a wall jack, and the silver cord to a conventional telephone if you were using the computer as an automatic dialer for voice calls.

Unplug the modem cable from the Model 100, and the telephone connected to the silver cord will go dead. So Radio Shack provides a shorting connector to mate with the 8-pin plug in place of the Model 100. It connects pins 1 and 7, temporarily making the modem cable into a rather expensive telephone extension cord.

But with a bit of drilling and soldering, you can make your own cable at a cost of only a few dollars. Even if you buy all the parts new from Radio Shack and want to duplicate all the functions, including autodialing for voice telephone calls, you'll spend only \$10.88. (See Table 4.)

Since you need only pins 1, 3, and 7 (and you can omit pin 1 if you don't need the silver cord) you can pry loose one of pins

2, 4, or 5 to be used in position 7. However, it's difficult to remove pins from the black plastic. I ended up buying an extra 5-pin DIN plug, and simply cracked apart the plastic to get a spare pin to mount into a newly-drilled hole in the other.

After you extract an extra pin from a DIN plug, drill a hole in the 5-pin plug so that you can insert the new pin and glue it in place. This requires a 5/64-inch drill bit and a steady hand. Before drilling, slip off the plastic sleeve of the DIN plug by lifting the tab above pin 2. Then separate the two halves of the metal barrel inside. This exposes the black plastic carrier containing the five pins, which requires a hole for the new pin 7. The important thing is to drill the hole directly below pin 1, so that the new pin 7 fits into the matching hole in the Model 100 phone jack.

Then, mix up some epoxy glue. (I used a brand that sets in 10 minutes with satisfactory results.) Grasp the extra pin with a tweezer or needle-nose pliers, apply glue to it with a toothpick, and insert it carefully into the hole. (See Photo 2.) Hold it parallel to the other pins until the glue has hardened somewhat. Then let it set for the period recommended in the glue instructions.

Next take the modular phone cord and cut it in half. Taking one half, carefully remove about an inch of the outer jacket. Inside you'll find green, red, and probably black and yellow wires. Clip the yellow and black wires (if any) short and strip the red and green wires, which carry what the telephone com-

| DIN pin | Modular plug                                          |
|---------|-------------------------------------------------------|
| 1       | grey cable, green wire                                |
| 2       | NC                                                    |
| 3       | grey cable, red wire,<br>and beige cable, red<br>wire |
| 4       | NC                                                    |
| 5       | NC                                                    |
| 6       | NC                                                    |
| 7       | beige cable, green wire                               |
| 8       | NC                                                    |

Table 3. Connections in Radio Shack Model 100 modem cable.

| Part Description                         | Current Price |
|------------------------------------------|---------------|
| Two 5-pin DIN plugs, part number 274-003 | \$1.49        |
| Telephone line cord, part number 279-374 | \$4.95        |
| Inline coupler, part number 279-358      | \$2.95        |

Table 4. Parts list for homemade modem cable.

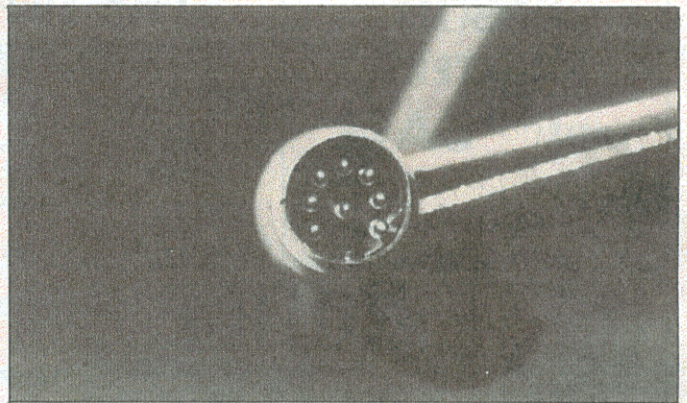


Photo 1. Radio Shack 8-pin DIN plug.

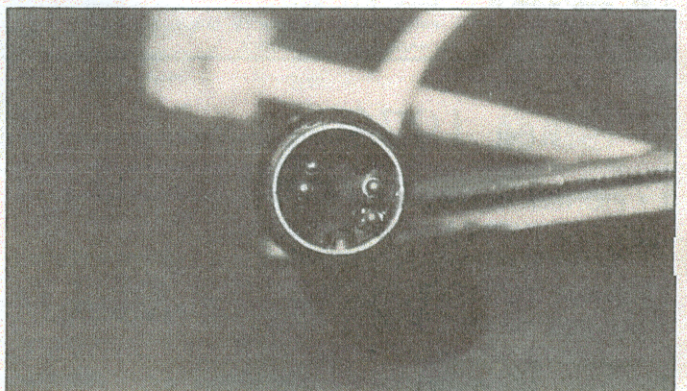


Photo 2. 5-pin DIN plug with new pin 7.



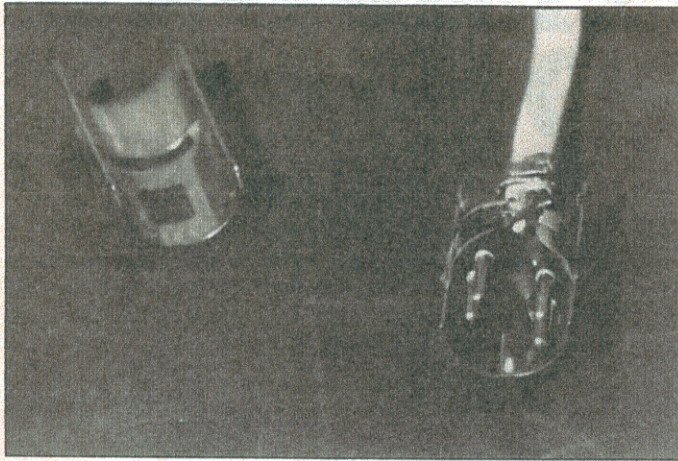


Photo 3. The red wire goes to pin 3, while the green wire goes to pin 7.

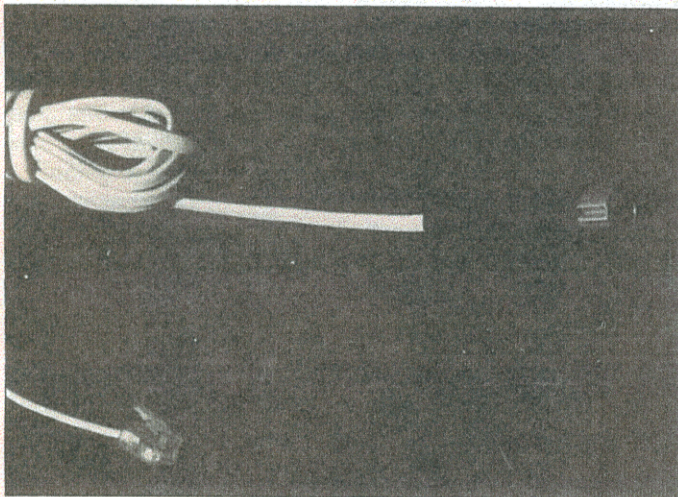


Photo 4. Completed modem cable.

pany calls "ring" and "tip" signals, respectively.

Thread the DIN plastic sleeve onto the phone cord so that later you can slip it onto the DIN plug. Then solder the cord to the plug, connecting the red wire to pin 3 and the green wire to pin 7. Reassemble the metal barrel. (See Photo 3.) Slide the sleeve back on, and the cable should be ready to test. (See Photo 4.)

Testing the cable is easy. If you have access to an ohmmeter, use it to check that none of the DIN pins is shorted to any other, and that none of the modular plug pins is shorted to any other. Then check for continuity along the modem cable—the red wire at the modular plug should connect with pin 3 at the DIN plug, while the green wire should go to pin 7. Then plug the modem cable into the computer and the telephone line.

Call your local Radio Shack store to get the telephone number for Tymnet or Telenet. With the computer in TELCOM, dial that number on an extension phone, and when you hear the high-pitched carrier tone, push F4. Words and letters should appear on the screen. Then push F8, the Bye key, and type Y. The tone should stop. ■

*Carl Oppedahl can be reached at 99 Park Ave., New York, NY 10016.*

## Program Length

by Ronald F. Balonis

Length.BA, an 8K RAM utility program, provides a method to compute program length in order to manage memory space on the Model 100. (See Program Listing 1.)

All memory references are in decimal notation. The user menu directory is located in RAM memory from 63930 to 64139, with each entry using 11 bytes. The first byte of each entry denotes the type of file: 0 is for a killed/empty file, 128 is for Basic program (.BA) files, and 192 for text (.DO) files. The next 2 bytes are the start address of the file in RAM memory, and the remaining 8 bytes are the file name and extension. Disassembly of the file area indicated that 26 is the end of file (EOF) indicator of a text file, and the standard string of three zeros is EOF for a .BA file.

The program conducts a sequential search of the directory for a file match, and then a sequential search of the file for the EOF byte, using PEEKs to read the memory. The menu directory search, while slow, proved acceptable, but the EOF search of the file had a tiresome wait with long files. This was an application in need of some machine language, and so I made my first attempt at a machine-language subroutine on the Model 100.

Borrowing on early Model I techniques, I fashioned a simple machine-language sequential search routine for the EOF search. The program loads it into the Model 100 with data statements, and the Basic PEEK and POKE statements pass the variables to and from the routine. A word or two of warning about my experience with machine language on the Model 100 is in order: Errors, at least of the types I made, cause the computer to do a cold restart, erasing all of the user programs.

### Program Operation

Line 15 clears string space and reserves 50 bytes (from MAXRAM to HIMEM 63960 to 63910) for the machine-language subroutine. Lines 30–50 POKE the machine-language program (see Program Listing 2) into this memory space.

Lines 100–180 prompt for a file name, test that it has a valid formation, then construct a menu directory match string of it. The program stores the file names in the menu directory without the period before the extension, with the name left-justified and the extension right-justified, and with spaces filling the middle if necessary. Lines 200–260 do the sequential search of the menu directory using the PEEK function to create a match string. Line 205 tests for a killed/empty file, and line 225 tests for a match of file names. If a match is found, then lines 230–235 get its address. Line 310 passes this address to the machine-language program for the EOF search called in line 320 or 330. Lines 400–430 compute and display the results.

The utility program is simple to use. Just enter a valid file name and in two to three seconds the program displays its length for logging. Press the space bar for another program or to exit. ■

*Write to Ronald F. Balonis at 118 Rice St., Trucksville, PA 18708.*



```

5 'LENGTH.BA FIND LENGTH OF A PROGRAM
10 'BY R.F.BALONIS JULY 16, 1983
15 CLEAR 100,MAXRAM-50:'JULY 17, 1983
20 TITLES$="LENGTH OF A PROGRAM"
25 '-MACHINE LANGAUGE PROG. END SEARCH-
30 FOR I=62911 TO 62958
40 READ Z:POKEI,Z:B$=B$+" "
50 NEXT I
60 DATA 0,0,229,42,191,245,43,
35,126,254,26,194,198
70 DATA 245,195,234,245,229,42,
191,245,43,35,126,254
80 DATA 0,194,213,245,35,126,254,
0,194,213,245,35
90 DATA 126,254,0,194,213,245,
34,191,245,225,201
95 '
100 B$="":FNMS$="":ADRS=0
110 CLS:PRINT@10,TITLES$
120 PRINT@82,"ENTER NAME AS IT APPEARS
ON THE MENU"
130 PRINT@165,"<ENTER> TO EXIT ";
135 INPUT FNMS$:IF FNMS$="" THEN MENU
140 I=INSTR(FNMS$,"."):TYPES$="":TYPE=0
145 IF I=0 THEN 100
150 TYPES$=MID$(FNMS$,I+1,2)
155 IF TYPES$="DO" THEN TYPE=1
160 IF TYPES$="BA" THEN TYPE=2
165 IF TYPE=1 OR TYPE=2 THEN 170
ELSE 100
170 FILN$=LEFT$(FNMS$,I-1)
175 IF LEN(FILN$)<6 THEN FILN$=FILN$+"
":GOTO 175
180 FILN$=FILN$+TYPES$
185 PRINT@285,"**** SEARCHING DIRECTORY
****";
190 '
195 '--SEQUENTIAL DIRECTORY SEARCH--
200 FOR I=63930 TO 64139 STEP 11
205 B$="":IF PEEK(I)=0 THEN 240
210 FOR II=3 TO 10
215 B$=B$+CHR$(PEEK(I+II))
220 NEXT II
225 IF INSTR(B$,FILN$)<1 THEN 240
230 IL=PEEK(I+1):IH=PEEK(I+2)
235 ADRS=IH*256+IL:I=64139
240 NEXT I
245 IF ADRS=0 THEN 250 ELSE 290
250 PRINT@285," **** NOT IN DIRECTORY *
*** ";
260 FOR I=1 TO 1000:NEXT I:GOTO 100
280 '
290 '---FIND THE LENGTH---
300 PRINT@285," **** COMPUTING LENGTH *
*** ";
310 POKE62911,IL:POKE62912,IH
320 IF TYPE=1 THEN CALL 62913
330 IF TYPE=2 THEN CALL 62928
340 '
400 CLS:PRINT@10,TITLES$
405 IADR=PEEK(62912)*256+PEEK(62911)
410 PRINT@163,"< "FNMS$ > IS "IADR-ADRS"
BYTES LONG"
420 PRINT@287,"**** PRESS <SPACEBAR> ***
**";
430 IF INKEY$="" THEN 100 ELSE 430

```

Program Listing 1. Length.BA—utility.

|           | Memory Address | Source Statement | Object Code |
|-----------|----------------|------------------|-------------|
| L Addr    | 62911          |                  | 0           |
| H Addr    | 62912          |                  | 0           |
| DO Entry  | 62913          | Push HL          | 229         |
|           | 62914          | LD HL, 62911     | 42,191,245  |
|           | 62917          | Dec HL           | 43          |
|           | 62918          | Inc HL           | 35          |
|           | 62919          | LD A, (HL)       | 126         |
|           | 62920          | Cp 26            | 254,26      |
|           | 62922          | JNZ 62918        | 194,198,245 |
|           | 62925          | JP 62954         | 195,234,245 |
| .BA Entry | 62928          | Push HL          | 229         |
|           | 62929          | LD HL, 62911     | 42,191,245  |
|           | 62932          | Dec HL           | 43          |
|           | 62933          | Inc HL           | 35          |
|           | 62934          | LD A, (HL)       | 126         |
|           | 62935          | Cp 0             | 254,0       |
|           | 62937          | JNZ 62933        | 194,213,245 |
|           | 62940          | Inc HL           | 35          |
|           | 62941          | LD A, (HL)       | 126         |
|           | 62942          | Cp 0             | 254,0       |
|           | 62944          | JNZ 62933        | 194,213,245 |
|           | 62947          | Inc HL           | 35          |
|           | 62948          | LD A, (HL)       | 126         |
|           | 62949          | Cp 0             | 254,0       |
|           | 62951          | JNZ 62933        | 194,213,245 |
|           | 62954          | LD 62911, HL     | 34,191,245  |
|           | 62957          | Pop HL           | 225         |
|           | 62958          | Return           | 201         |

Program Listing 2. Length.BA—machine language.

## Blackjack

by Paul Serotta

Blackjack isn't a mere conversion: This program uses the Model 100's unique features—graphics, sound, and interrupt-controlled function keys—to simulate the popular casino card game also known as 21. Like the Vegas version, the game program pits a single player against the house's dealer (the Model 100). The object of the game is to accumulate a hand worth 21 points, or as close to 21 as possible, without going over. The house wins if you draw over 21, or if the dealer's hand is closer to 21 than yours is.

The 8K program starts by asking you to type in your name and press the enter key. From now on, it addresses you personally. The program gives you a stake of \$1,000 and asks you to enter your bet. Should you decide to wager the entire amount on the first hand, the buzzer sounds and the program applauds your bravado with an encouraging "Go for it!"

Conversely, a low bet (under \$100) merits the program's disdainful "You are cheap!"

After you've entered your bet (in full-dollar amounts only), four boxes appear on the screen. The two upper boxes represent your first two cards; a typical deal might be a king of clubs and a six of hearts. In the lower right-hand box the program displays the dealer's first card, for example, a six of spades. You then have the following options, selected with the 100's function keys: hit, double, stay, and quit. Press the F1 key if you want another card, the F2 key if you want to be dealt another card and double your original bet, the F3 key to play a two-card hand against the dealer, or the F4 key to stop the game altogether.

When you stay, you're electing to play your current hand against whatever the house turns up for itself. The program then keeps dealing itself more cards until it has reached 21—an automatic win, beaten your hand, or gone over. If you lose, the program tells you "You are busted." If you win, it admits "I'm busted—you win!" Should the dealer's hand match your own, the program declares the deal a draw or "push" and no one wins or loses.

The program keeps a running tally of your stakes; after each deal it reminds you of how much money you have available and asks you to enter another bet.

It's impossible to cheat at this Blackjack, by the way. Try wagering more money than you have in your purse or doubling at the wrong time, and the program calls you on it.

When you decide to bail out, press the F4 key and the program totals your winnings—or your losses.

There's one born every minute. ■

Contact Paul Serotta at 131 Penrose Drive, Pittsburgh, PA 15208.

Program Listing 3. Blackjack.

```

10 REM *****
20 REM          BLACKJACK
30 REM
40 REM PAUL SEROTTA
45 REM 131 PENROSE DR.
50 REM PITTSBURGH, PA 15208
55 REM
60 REM *****

80 REM
99 REM DIMENSION CARD ARRAY, SET
  PLAYER'S AMOUNT OF MONEY
100 DIM C(52):PM=1000
105 REM CLEAR FUNCTION KEYS (F1-F4)
110 FOR LL=63369 TO 63432:POKELL,0:NEXT LL
112 REM TITLE PAGE
115 CLS:LINE(70,24)-(115,52),1,B:LINE(
  120,24)-(165,52),1,B:PRINT@173,"ACE
  ";:PRINT@181,"JACK ";
125 GOSUB9000:CLS:PRINT@121,"PLEASE
  TYPE YOUR NAME AND PRESS 'ENTER'"
140 PRINT:LINE INPUTN$:GOSUB9000
145 KEY OFF:IFPM=<=0 THEN GOTO3000
147 REM INITIALIZE LOGIC VARIABLES & ASK
  
```

```

FOR BET
150 PP=1:CS=2:PA=0:CA=0:PT=0:CT=0:CR=2:
  PC=81:CP=201:GOSUB9020:PRINT:PRINTN$;"
  YOU HAVE $";PM
160 PRINT@200,"";:INPUT"PLEASE ENTER
  YOUR BET";BET:BET=INT(BET)
170 IF BET >PM THEN GOSUB9050:GOTO160
180 IFSGN(BET)=-1ORSIGN(BET)
  =0 THEN GOSUB9050:GOTO160
190 IFBET=PM THEN PRINT@293,CHR$(27);"p";"
  GO FOR IT !! ";CHR$(27)
  ;"q";:SOUND4000,15:SOUND3000,10:SOUND200
  0,8:FORDL=1TO500:NEXTDL
195 IFBET<100ANDPM>500 THEN PRINT@291,CHR$(
  27);"p";" YOU ARE CHEAP !! ";CHR$(27)
  ;"q";:SOUND4000,30:SOUND12000,30:FORDL=1
  TO500:NEXTDL
200
  GOSUB9020:GOSUB9100:GOSUB9110:PP=1:GOSUB
  9500:ONKEYGOSUB1000,4000,2000,3000
202 PRINT@100,CHR$(155);"
  YOU";:PRINT@220,CHR$(155);"
  DEALER";:FORDL=1TO300:NEXTDL:PRINT@100,S
  PACE$(6);:PRINT@220,SPACE$(9);
210 PRINT@PC,C$;CHR$(156+SU);:PC=PC+
  8:PT=PT+CV:GOSUB9500:PT=PT+
  CV:PRINT@PC,C$;CHR$(156+SU);:PC=PC+8
215 IFPA=2 THEN PA=1:PT=PT-10
220 PP=0:GOSUB9500:HC$=C$+CHR$(156+SU)
  :CT=CT+CV:CP=CP+8:GOSUB9500:CT=CT+
  CV:PRINT@CP,C$;CHR$(156+SU);:CP=CP-8
225 IFCA=2 THEN CA=1:CT=CT-10
230 IFPT=2LANDCT<>
  21 THEN PRINT@CP,HC$;:PRINT@280,"BLACKJACK
  !$!$!$!$!$!";:PM=PM+BET+INT
  (BET/2)
  :FORDL=1TO5:SOUND4000,10:SOUND8000,10:NE
  XTDL:GOTO145
240 IFCT=2LANDPT<>
  21 THEN PRINT@CP,HC$;:PRINT@280,"I HAVE
  BLACKJACK $$!! - YOU LOSE ";:PM=PM-
  BET:SOUND14000,15:SOUND16000,20:FORDL=1T
  O1000:NEXTDL:GOTO145
250
  IFCT=2LANDPT=21 THEN PRINT@CP,HC$;:GOTO221
  0
260 KEY ON
270 PRINT@280,"HIT DBL STAY
  QUIT";:FORWW=1TO200:NEXTWW:PRINT@280,SPA
  CE$(39);:FORWW=1TO200:NEXTWW
280 IFNH=1 THEN NH=0:GOTO145:ELSE GOTO270
999 REM PLAYER HITS
1000 CR=CR+1
1010 IFCR=3 THEN LINE(94,9)-(142,30)
  ,1,B:GOTO1050
1020 IFCR=4 THEN LINE(141,9)-(189,30)
  ,1,B:GOTO1050
1030 IFCR=5 THEN LINE(188,9)-(236,30),1,B
1050 PP=1:GOSUB9500:PRINT@PC,C$;CHR$(
  156+SU):PC=PC+8:PT=PT+CV
1060 IFPT>21ANDPA<=0 THEN PRINT@280,N$;"
  YOU ARE BUSTED
  ";:FORDL=1TO10:SOUND12000,DL:NEXTDL:PM=PM-
  BET:FORDL=1TO500:NEXTDL:NH=1:RETURN
1070 IFPT>21ANDPA>0 THEN PA=PA-1:PT=PT-10
1100 IFCR<>5 THEN RETURN
1200 PRINT@280,"5 CARD CHARLIE - YOU WIN
  !!";:FORDL=1TO8:SOUND5000,DL:SOUND15000,
  DL:NEXTDL:PM=PM+BET+INT(BET/2)
  
```

Listing 3 continued

Listing 3 continued

```

:NH=1:RETURN
1999 REM COMPUTER'S LOGIC
2000 KEY OFF
2005 PRINT@280,SPACE$(38);
2010 PP=0:PRINT@CP,HC$:CP=CP+16
2020 WC=94: IFCT>16THEN2200
2050 CS=CS+1:
IFCS=6THENGOTO2206:ELSELINE(WC,32)-(WC+
48,53),1,B:GOSUB9500
2060 WC=WC+47:CT=CT+CV:PRINT@CP,C$;CHR$(
156+SU);:CP=CP+8
2070 IFCT>21 AND CA>0 THENCA=CA-1:CT=CT-
10
2075 IFCS=6ANDCT<22THENGOTO2206
2080 IFCT<=16THEN2050
2199 REM WHAT HAPPENED?
2200 IFCT>21THENPRINT@280,"I'M BUSTED --
- YOU WIN
!!";:SOUND14000,20:SOUND10000,20:SOUND50
00,20:SOUND3500,20:PM=PM+BET:GOTO2500
2206 IFCS>=5THENPRINT@280,"I'VE GOT A 5
CARD CHARLIE -- YOU LOSE";:PM=PM-
BET:SOUND7500,20:SOUND5000,15:SOUND10000
,20:GOTO2500
2210
IFCT=PTTHENPRINT@280,"PUSH.....
";:FORDL=7000TO10000STEP1000:SOUNDL,8:N

```

```

EXTDL:GOTO2500
2220 IFCT>PTTHENPRINT@280,"I WIN
$$$$$$$$";:SOUND16000,25:PM=PM-
BET:GOTO2500
2230 IFCT<PTTHENPRINT@280,"YOU WON
!!!!";:SOUND3000,20:SOUND8000,20:SOUND3
000,20:PM=PM+BET
2500 FORDL=1TO1250:NEXTDL: IFCO>
25THENGOSUB9100
2505 NH=1: RETURN
2999 REM QUIT
3000 CLS:PRINT@80,"BYE ";N$:IFPM>
=1000THENPRINT@200,"YOU WON $";PM-
1000:END:ELSE
3010 PRINT@200,"YOU LOST $";ABS(PM-1000)
:END
3999 REM PLAYER DOUBLES
4000 IFCR>2THENPRINT@280,"YOU CAN'T
DOUBLE NOW --- DUMMY
!!";:SOUND7500,20:SOUND13000,20:FORDL=1T
01000:NEXTDL:PRINT@280,SPACE$(38)
;:RETURN
4002 IFBET*2>PMTHENPRINT@280,"YOU DON'T
HAVE ENOUGH MONEY ****
";:SOUND3000,10:SOUND15000,10:SOUND8000,
10:FORDL=1TO1000:NEXTDL:PRINT@280,SPACE$
(35);:RETURN

```

Listing 3 continued

# TRS-80\*

100% Radio Shack Equipment

## SAVE A BUNDLE

Order Toll Free 1-800-874-1551

FLA Residents 904-438-6507 collect

EPSON, OKIDATA, CITH, TABCO Printer Switches

**HVAB SALES CO.**

704 W Michigan Ave; P.O. Box 8098  
Pensacola, FLA 32505

-189

\*TRS-80 is a trademark of Tandy Corporation.



Listing 3 continued

```

4005 PP=1:CR=CR+1:LINE(94,9)-(142,30)
,1,B:GOSUB9500
4010 PRINT@PC,C$;CHR$(156+SU):PT=PT+
CV:BET=BET*2
4020 IFPT>21ANDPA=<0THENGOTO1060
4030 IFPT>21ANDPA>0THENPT=PT-10
4050 GOTO2000
8999 REM PRINT WELCOME
9000 FORLL=1TO5:PRINT@50,"WELCOME TO
";CHR$(27);"p";"BLACKJACK";CHR$(27);"q"
9010 FORDL=1TO60:NEXTDL:BEEP:
PRINT@50,SPACE$(20)
;:FORDL=1TO60:NEXTDL,LL:RETURN
9015 REM PRINT NAME OF GAME ON FIRST
LINE
9020 CLS:LINE(0,0)-(239,7)
,1,BF:PRINT@15,"BLACKJACK":LINE(0,7)-(
239,7):RETURN
9030 REM ANY CHEATING ??????
9050 PRINT@280,"NO CHEATING IN THIS GAME
!";SOUND4000,10:SOUND10000,10:SOUND1600
0,10
9060
FORDL=1TO750:NEXTDL:PRINT@223,SPACE$(96)
;:RETURN
9099 REM SHUFFLE THE CARDS
9100 CO=0:FORLL=1TO52:C(LL)

```

```

=1:NEXTLL:RETURN
9105 REM DRAW THE INITIAL 4 CARDS
9110 LINE(0,9)-(48,30),1,B:LINE(47,9)-(
95,30),1,B:LINE(0,32)-(48,53),1,B:LINE(
47,32)-(95,53),1,B
9115 RETURN
9499 REM PICK A CARD
9500 SEC=VAL(RIGHT$(TIME$,2))
:FORI=1TOSEC:DUM=RND(1):NEXTI:RN=INT(RND
(1)*53)
9510 IFC(RN)=0THEN9500
9520 CO=CO+1:C(RN)=0:SU=RNMOD4:
9530 IFRN>4THENGOTO9535:ELSEC$="ACE
":CV=11
9532 IFPP=1THENPA=PA+1:ELSECA=CA+1
9533 GOTO9600
9535 IFRN>40THENGOTO9540:ELSEC$=STR$(INT
(RN/4))+":CV=INT(RN/4)
9537 IFRNMOD4<>0THENC$=STR$(INT(RN/4)+1)
+"":CV=INT(RN/4)+1
9538 GOTO9600
9540 IFRN<45THENC$="JACK
":CV=10:GOTO9600
9545 IFRN<
49THENC$="QUEEN":CV=10:GOTO9600
9550 C$="KING":CV=10
9600 RETURN

```

STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION

1. Title of publication, 80 Micro. A. ISSN 07447868. 2. Date of filing, Oct. 1, 1983. 3. Frequency of issue, Monthly. A. No. of issues published annually, 12. B. Annual subscription price, \$36.00. 4. Location of known office of publication, 80 Pine Street, Peterborough, Hillsboro County, NH 03458. 5. Location of headquarters or general business offices of the publishers, Same. 6. Names and complete addresses of publisher, editor, and managing editor. Publisher and Editor, Wayne Green, 80 Pine Street, Peterborough, NH 03458. Managing Editor, Eric Maloney, 22A Pine Street, Peterborough, NH 03458. 7. Owner, International Data Group, PO Box 1450, 5 Speen Street, Framingham, MA 01701; Patrick J. McGovern, PO Box 1450, 5 Speen Street, Framingham, MA 01701. Stockholders, Wayne Green, Peterborough, NH 03458. 8. Known bondholders, mortgagees, and other security holders owning or holding 1 percent or more of total amount of bonds, mortgages or other securities,

None. 9. For completion by nonprofit organizations authorized to mail at special rates (Section 411.3 DMM). The purpose, function and nonprofit status of this organization and the exempt status for Federal income tax purposes (Check one) Not applicable. 10. Extent and nature of circulation. (X) Average No. copies each issue during preceding 12 months. (Y) Actual No. copies of single issue published nearest to filing date. A. Total No. of copies printed (X) 148,067 (Y) 164,029 B. Paid circulation 1. Sales through dealers and carriers, street vendors and counter sales (X) 48,160 (Y) 73,415 2. Mail subscription (X) 78,634 (Y) 78,179 C. Total paid circulation (X) 124,794 (Y) 149,584 D. Free distribution by mail, carrier or other means, samples, complimentary, and other free copies (X) 807 (Y) 837 E. Total distribution (X) 125,601 (Y) 150,431 F. Copies not distributed 1. Office use, left over, unaccounted, spoiled after printing (X) 7,980 (Y) 13,598 2. Returns from news agents (X) 15,086 (Y) 0 G. Total (X) 148,067 (Y) 164,029.

**ARE YOU TIRED OF HEADACHES AND WATERY EYES FROM STARING AT YOUR TRS-80'S HARSH WHITE VIDEO DISPLAY?**

Well, a **FATIGUE FIGHTER** optical filter changes that display to a nice, easy on the eyes, green. It is made of hard 1/8 inch thick acrylic for durability, attaches in seconds without tools, and matches TRS-80 styling. So, invest in some relief, get a **FATIGUE FIGHTER** for your Model I, II, or III. It will probably be one of the most used accessories you will ever buy.

**P.S. Available direct from us or at computer stores.**

TRS-80 IS A TANDY CORP. TRADEMARK.

**TO ORDER:**

Send Name & Address Typed or Clearly Printed with Check or Money Order (U.S. FUNDS) for \$14.95 Each, Including Shipping. Canadian Orders Add \$1.00 Each. All Other Foreign Orders Add \$3.00 Each for Shipping. NO CREDIT CARD ORDERS. COD's (U.S. ORDERS ONLY) are \$3.00 Additional per Order & are Accepted by Mail or by Calling 904-378-2494 or 9-5 M-F. Florida Residents Add 5% Sales Tax.



**SOUTHERN INNOVATIVE DESIGN**  
3033-15 N. E. 19TH DRIVE  
GAINESVILLE, FL 32601-3326

✓35

**— DEALER INQUIRIES INVITED —**

# RAM FILES

## Monitor 100 Changes

I found the "Monitor 100" program (August 1983, p. 178) particularly interesting, but to make it work I had to make two changes to the printed version.

First, statement 58 is redundant and should be deleted. Also, in line 200, MID\$(0\$,3,4) should be MID\$(0\$,2,4).

The Model 100 is an excellent complement to my big computer (a Kaypro II set up as an RCPM/RBBS). I can prepare draft documents on the go, then capture the incoming text file to disk.

A Basic program on the Kaypro (ADDLF.BAS) adds a line feed at each carriage return (required by CP/M and many non-TRS-80 computers). You can then print the resulting file as is, or process it further on your home computer. You can upload Model 100 Basic programs in a similar manner if you save them in ASCII format.

I use an Epson MX-80 printer. Its ability to skip over perforations by either software command or hardware switch lets you paginate documents created on the Model 100 without the intermediate step of uploading to a more powerful computer.

Anyone buying a printer for the Model 100 should look for this feature, as well as the capability to add line feeds (a hardware switch option on the MX-80).

The usefulness of the Model 100 is enhanced by the series of features initiated in your July issue. Although it is not my primary use of this computer, methods of installing machine-language code would be a good subject for a future article.

*Phil Wheeler  
5539 Towers St.  
Torrance, CA 90503*

## Foxfighter Glitch

I just bought my first copy of *80 Micro*, and I'm delighted to find the C\*Notes section dedicated to the Model 100. I am satisfied with my Model 100, but until recently thought that no one was writing software for it.

I enjoyed the Foxfighter program (August 1983, p. 200), but I did find a couple of glitches in it.

The program always presents one of seven predefined screen displays for the air mines. This becomes routine after a little while and encourages high scoring.

The changes shown in lines 10 and 25-34 in Program Listing 1 display the air mines in random patterns instead. Occasionally this results in an invisible air mine or two, adding to the challenge of the game.

Line 10 sets the RND function to one of 60 different starting points based on the Model 100's built-in clock. Lines 25-34 use the RND function to display the air mines in various screens instead of the seven predefined screens used in the original listing.

Also, line 150 of Foxfighter is supposed to provide an additional fighter plane when the score reaches 500, 1,000, and 2,000. Actually, it only increments the number of fighters displayed by 1 at these three points. Lines 150-158 in Listing 1 change the number of fighters as well as the display.

*Harold Shaver  
509 Mulberry #9  
Suisun, CA 94585*

```
10 DIMX(6),A(42):V=3:SCR=0:
CLS:PRINT@92,"<<FOXFIGHTER>>":
PRINT:PRINT:FORT=1TOVAL(RIGHT$(TIMES,2))
:SEC=RND(1):NEXT:INPUT"DO YOU NEED
INSTRUCTIONS (Y OR N) ";AS:IFAS="Y"
THEN 390 ELSE 20
25 FORI=1 TO 42:READA(I): NEXT
30 FORM=1 TO 6
32 Y=INT(41*RND(1))+1:FORI=1 TO
6:IFX(I)=A(Y) THEN32
34 X(M)=A(Y):NEXTM
150 IFSCR=500 AND
BN=0THENV=V+1:BN=1:GOTO155ELSEIFSCR=1000
AND BN=0THENV=V+1:BN=1:GOTO155ELSE IF
SCR=2000 AND BN=0THENV=V+1:BN=1:GOTO
155
155 IF SCR<> 500 AND SCR <>1000 AND SCR
<> 2000 THEN BN=0
158 PRINT@0,V
```

*Program Listing 1. Adjustments for Brad Dixon's "Foxfighter" program.*

## Calculator Program

The short calculator program in Program Listing 2 is one of the first programs I wrote on my Model 100.

To run the program, input a value and press the enter key. Then input either a plus (P), multiplication (.), subtraction (-), or division (/) sign.

Type in your next number and press the enter key. Now press the equals key to get an answer, or key in another function and continue calculating. Once you have an answer, you can start over, stop, or carry your balance forward.

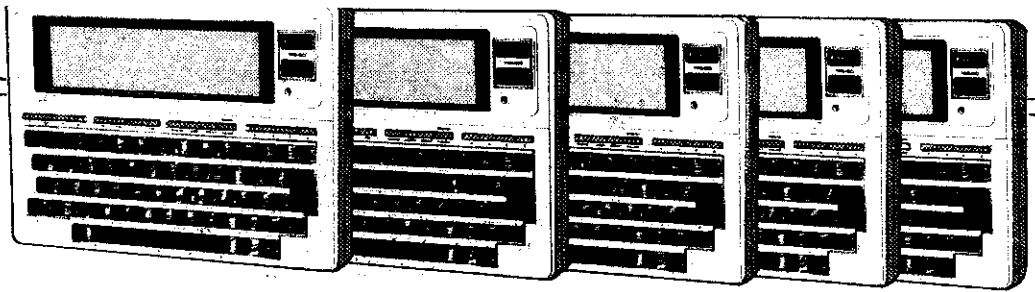
By using P for addition and a period for multiplication, you don't need to use the shift lever.

*Mark Fox  
774 Hazelwood Drive  
North Wales, PA 19454*

```
5 CLS
6 PRINT"CALCULATOR PROGRAM"
10 INPUT Y
11 CLS
15 PRINTY
20 Y$=INKEY$:IF Y$=""THEN GOTO20
21 IF Y$=""THEN100
25 PRINTY$
30 INPUT X
31 IF Y$="P"THEN Y=Y+X:GOTO20
32 IF Y$="-"THEN Y=Y-X:GOTO20
33 IF Y$="."THEN Y=Y*X:GOTO20
34 IF Y$="/"THEN Y=Y/X:GOTO20
100 PRINT"-----"
200 PRINTY;" IS YOUR ANSWER"
210 PRINT
220 PRINT"ANOTHER EQUATION? OR BAL
FWD(Y,N,B) "
223 Z$=INKEY$
230 IF Z$=""GOTO 223
230 IF Z$="Y"THEN5
240 IF Z$="Y"THEN5
250 IF Z$="B"THEN15
260 END
```

*Program Listing 2. Model 100 Calculator program.*

C-NOTE FOR JANUARY 1984



## Mnemonic Powers

by Dan Robinson

Radio Shack's Model 100 is a chip off the old block, so to speak. It uses a low-powered version of the 8085 central processing unit (CPU) and its instruction set is almost identical to that of the 8080 chip.

Since the Z80 CPU that drives the Models I, II, III, and 4 is upwardly compatible with the 8080, most of the 100's 8085 instructions should be familiar to Model I/II/III/4 Assembly-language programmers. But some significant differences exist. I'll go over them and provide you with a table listing the command mnemonics for each chip.

The primary differences between the chips arise from the fact that the 8085 is short a few registers. It has no IX and IY index registers, and no primed alternate register set. Also, the 8085 has no instruction to directly load the interrupt register. Instead, a pair of RIM and SIM instructions use a bit pattern

in the A register to set the interrupt mask and data bus, and to perform a restart instruction. The restart instructions—numbered RST zero through 7—work from locations in the first 64 bytes of memory.

The 8085 doesn't support relative jumps, and is missing the set and reset bit tests. The miscellaneous instructions that compare blocks and moves are also missing, as are the CB, ED, DD, and FD families of instructions.

In theory, it's possible to use Model I/III utilities such as EDTASM to write assembler code for the Model 100, but you'll have to avoid commands that the Model 100 doesn't support. The object is to send code to the Model 100 through a modem while filtering out unsupported loading codes.

Intel uses its own jargon with the 8085 (MOV instead of LD, for example). Since Z80 terms are currently more familiar to TRS-80 owners, you're likely to see many Zilog words applied to the 8085. Table 1 helps you make the translation. ■

You can reach Dan Robinson at 1625 Higgins Way, Pacifica, CA 94044.

Table 1. Translation of Intel 8085 mnemonics to Zilog Z80.

| Intel 8085 | Zilog Z80  | Description                  |          |            |                                                      |
|------------|------------|------------------------------|----------|------------|------------------------------------------------------|
| M          | (HL)       | Contents of HL register pair | JC nn    | JP C,nn    | Jump to nn if carry                                  |
| ACI n      | ADC A,n    | Add n + carry bit to A       | JM nn    | JP M,nn    | Jump to nn if minus                                  |
| ADC r      | ADC r      | Add r + carry bit to A       | JMP nn   | JP M,nn    | Jump to nn if minus                                  |
| ADD r      | ADD r      | Add r to A                   | JNC nn   | JP NC,nn   | Jump to nn if no carry                               |
| ADI n      | ADD A,n    | Add n to A                   | JNZ nn   | JP NZ,nn   | Jump to nn if not zero                               |
| ANA r      | AND r      | Logical AND r & A            | JP nn    | JP nn      | Jump to nn                                           |
| ANI n      | AND n      | Logical AND n & A            | JPE nn   | JP PE,nn   | Jump to nn if parity even                            |
| CALL nn    | CALL nn    | Call routine at nn           | JPO nn   | JP PO,n    | Jump to nn if parity odd                             |
| CC         | CALL C,nn  | Call nn if carry bit set     | JZ nn    | JP Z,nn    | Jump to nn if zero                                   |
| CM         | CALL M,nn  | Call nn if minus             | LDA nn   | LD A,(nn)  | Load A with contents of nn                           |
| CMA        | CPL        | Complement A                 | LDAX     | LD A,(rr)  | Load A with contents of register pair                |
| CMC        | CCF        | Complement carry flag        | LHLD nn  | LD HL,(nn) | Load HL with data contained at address nn            |
| CMP r      | CP r       | Compare r to A               | LXI r,nn | LD rr,nn   | Load register pair with nn                           |
| CNC nn     | CALL NC,nn | Call nn if no carry          | MOV r,r  | LD r,r     | Load register with contents of second register       |
| CNZ nn     | CALL NZ,nn | Call nn if not zero          | MVI r,n  | LD r,n     | Load register with n                                 |
| CP nn      | CALL P,nn  | Call nn if positive          | NOP      | NOP        | No Operation                                         |
| CPE nn     | CALL PE,nn | Call nn if parity even       | ORA r    | OR r       | Logical OR r with A                                  |
| CPI n      | CP n       | Compare n to A               | ORI n    | OR n       | Logical OR n with A                                  |
| CPO nn     | CALL PO,nn | Call nn if parity odd        | OUT n    | OUT n,A    | Send byte in A out port n                            |
| CZ nn      | CALL Z,nn  | Call nn if zero              | PCHL     | JP (HL)    | Jump to address in HL                                |
| DAA        | DAA        | Decimal adjust A             | POP r    | POP rr     | Load register pair from stack                        |
| DAD r      | ADD HL,rr  | Add HL & register pair       | PUSH r   | PUSH rr    | Put register pair on stack                           |
| DCR r      | DEC r      | Decrement register           | RAL      | RLA        | Rotate A left with carry                             |
| DCX r      | DEC rr     | Decrement register pair      | RAR      | RRA        | Rotate A right with carry                            |
| DI         | DI         | Disable interrupts           | RC       | RET C      | Return if carry                                      |
| EI         | EI         | Enable interrupts            | RET      | RET        | Return                                               |
| HLT        | HALT       | Stop processor               | RIM      | ---        | Load A with byte & read interrupt mask & serial port |
| IN n       | IN n       | Read byte from port n to A   | RLC      | RLCA       | Copy A to carry, rotate left                         |
| INR r      | INC r      | Increment register           |          |            |                                                      |
| INX r      | INC rr     | Increment register pair      |          |            |                                                      |

Table 1 continued



Table 1 continued

|         |            |                                                                      |
|---------|------------|----------------------------------------------------------------------|
| RM      | RET M      | Return if minus                                                      |
| RNC     | RET NC     | Return if no carry                                                   |
| RNZ     | RET NZ     | Return if not zero                                                   |
| RP      | RET P      | Return if positive                                                   |
| RPE     | RET PE     | Return if parity even                                                |
| RPO     | RET PO     | Return if parity odd                                                 |
| RRC     | RRCA       | Copy A to carry, rotate right                                        |
| RST     | RST        | Restart instruction                                                  |
| RZ      | RET Z      | Return if zero                                                       |
| SBB r   | SBC A,r    | Subtract r from A with carry                                         |
| SBI n   | SBC A,n    | Subtract n from A with carry                                         |
| SHLD nn | LD (nn),HL | Load HL to address nn                                                |
| SIM     | ---        | Set interrupt mask from A, read byte from serial port reset data bus |
| SPHL    | LD SP,HL   | Load stack pointer with HL                                           |
| STA nn  | LD (nn),A  | Load address nn with A                                               |
| STAX r  | LD (rr),A  | Load address pointed to by register pair with A                      |
| STC     | SCF        | Set carry flag                                                       |
| SUB r   | SUB r      | Subtract r from A                                                    |
| SUI n   | SUB n      | Subtract n from A                                                    |
| XCHG    | EX DE,HL   | Exchange DE & HL registers                                           |
| XRA r   | XOR r      | Logical exclusive OR, r & A                                          |
| XRI n   | XOR n      | Logical exclusive OR, n & A                                          |
| XTHL    | EX (SP),HL | Exchange contents of stack pointer with HL                           |

Program Listing 1. COM/CMD.

```

00100 ; COMKXTY/SOR COM/CMD
00110 ; A MODEL 1 DISK UTILITY
00120 ; FOR FILE TRANSFER BETWEEN
00130 ; THE MODEL 1 AND MODEL 100
00140 ; USING RS232 COMMUNICATION
00150 ; BY RONALD F. BALONIS 8/23/83
00160 ;
00170 ;
7000 00180 ; ORG 07000H
00190 ;
00200 ;----- PROGRAM PARAMETERS
00210 ; RS232 COM STATS
00220 ; 9600 BAUD, 8 BIT WORD,
00230 ; PARITY ENABLED,
00240 ; ONE STOP BIT
00250 ; XON/XOFF ON SEND TO
00260 ; MODEL 100 ONLY
00270 ; ON MODEL 100 USE COM
00280 ; STATS OF 88E1E
00290 ;
00300 ; TO OUTPUT A DISK FILE
00310 ; TYPE: COM O FILENAME
00320 ;
00330 ; TO INPUT A 100 FILE
00340 ; TYPE: COM I FILENAME
00350 ;
00EE 00360 BAUDS EQU 0EEH
00E5 00370 CONFG EQU 0E5H
0011 00380 XON EQU 11H
0013 00390 XOFF EQU 13H
404A 00400 HINEM EQU 404AH
4318 00410 CMD EQU 4318H
431C 00420 FUNCT EQU CMD+4
431E 00430 FILE EQU CMD+6
402D 00440 QUIT EQU 402DH
4420 00450 DINIT EQU 4420H
4424 00460 DOPEN EQU 4424H
4436 00470 DREAD EQU 4436H
4439 00480 DWRITE EQU 4439H
4428 00490 DCLOSE EQU 4428H
0033 00500 VIDEO EQU 33H
00510 ;----- RESET THE UART FIRST
7000 CDDB71 00520 START CALL RSRST
    
```

Listing 1 continued

## 100 Disks

by Ronald F. Balonis

The Model 100's tape file system is great, but this is 1984 and I prefer disk storage. So I wrote a program called COM/CMD (Program Listing 1) that allows easy file transfer between a Model I/III and the Model 100 and adds manual disk storage to your Model 100.

To use the utility, connect the machines' RS-232 ports with a null modem cable or directly to the Model I if the communications terminal (COM/TERM) switch is set to COM.

### File Transfer

To transfer a file from the Model 100 to a Model I/III, get into DOS on the Model I/III, type COM I and the file name, and press the enter key. Then, on the Model 100, type SAVE "COM:88E1E and press the enter key.

After a time, depending on the file type and length, COM/CMD sends the file to the Model 100's RS-232 port and saves it to the Model I/III disk. When the READY prompt and cursors appear on both computers, the transfer is complete.

To transfer a file from the Model I/III to the Model 100, type LOAD "COM:88E1E on the Model 100 and press the

enter key. In DOS on the Model I/III, type COM 0 and the file name and press the enter key.

After a time, again depending on the file type and length, the program reads the disk file and sends it to the Model I/III's RS-232 port. When the READY prompt and cursors appear, the transfer is complete.

### The Program

My program is a collection of familiar Model I Z80 routines (see Table 2 for Model III modifications). The program first reads the DOS command line to select input or output mode and for the file name. The program then reads the files, whether input or output, into memory storage and then outputs them according to the mode of operation. In sequence, the routines are: Disk In, Com Out, Com In, Disk Out, Error messages, and the software for the RS-232 port.

The RS-232 stats are set in the software. It ignores configuration switches and sets communications parameters at 9600 baud, 8-bit word length, even parity, 1 stop bit, and XON/XOFF enabled on send to the Model 100 only.

The XON/XOFF, which enables the Model 100 to turn the Model I/III data stream on and off, is necessary to the communication link. Otherwise, the Model 100's housekeeping routines would lose data.

To get a copy of my utility program running on your Model I/III, if you have a Z80 disk assembler, simply key in the source listing and compile it. I use the Misosys modified version of the Radio Shack Series I Editor Assembler—Cassette. Alternatively, key in the hex listing with Debug, then use the DOS Dump command to put a copy on disk. ■

Contact Ronald F. Balonis at 118 Rice St., Trucksville, PA 18708.

### The Key Box

The programs in "100 Disks" require 8K RAM on the Model 100, 32K RAM on a Model I or III. The programs in "Rooting Out the Problem" and "Autoliner" run in 8K RAM.

Listing 1 continued

```

7003 11E771 00530 LD DE,DCB
7006 211E43 00540 LD HL,FILE
7009 011700 00550 LD BC,23
700C EDB0 00560 LDIR
00570 ;-----
700E 3A1C43 00580 SELECT COM <I>N OR COM <O>T
7011 FE49 00590 LD A,(FUNCT)
7013 287E 00600 CP 'I'
7015 FE4F 00610 JR Z,GETCOM
7017 C20071 00620 CP 'O'
00630 JP NZ,ERCMD
00640 ;-----
701A 11E771 00640 GETDSK PUT THE DISK FILE IN STORAGE
701D 010000 00650 LD DE,DCB
7020 C5 00660 LD BC,0000H
7021 210772 00670 PUSH BC
7024 CD2444 00680 LD HL,BUFFER
7027 C20571 00690 CALL DOPEN
702A 110773 00700 JP NZ,EROPEN
702D C1 00710 LD DE,STORAG
702E C5 00720 POP BC
702F D5 00730 READLP PUSH BC
00740 PUSH DE
7030 11E771 00740 LD DE,DCB
7033 CD3644 00750 CALL DREAD
7036 FE1C 00760 CP 1CH
7038 281C 00770 JR Z,RCLOSE
703A FE1D 00780 CP 1DH
703C 2818 00790 JR Z,RCLOSE
703E B7 00800 OR A
703F C20A71 00810 JP NZ,ERREAD
7042 D1 00820 POP DE
7043 210772 00830 LD HL,BUFFER
7046 010001 00840 LD BC,256
7049 EDB0 00850 LDIR
704B C1 00860 POP BC
704C 0C 00870 INC C
704D 3A4A40 00880 LD A,(HIEM)
7050 BA 00890 CP D
7051 CA2071 00900 JP Z,TOOBIG
7054 18D8 00910 JR READLP
7056 11E771 00920 RCLOSE LD DE,DCB
00930 ;-----
7059 3AF371 00940 PUT FILE LENGTH IN BC
705C 47 00950 LD A,(DCB+12)
705D 3AEP71 00960 LD B,A
7060 4F 00970 LD A,(DCB+8)
LD C,A
    
```

```

7061 CD2844 00980 CALL DCLOSE
7064 C20A71 00990 JP NZ,ERREAD
01000 ;----- SEND FILE TO RS232 COM PORT
7067 210773 01010 PUTCOM LD HL,STORAG
706A CDCB71 01020 COMOUT CALL RSIN
706D FE13 01030 CP XOFF
706F 2007 01040 JR NZ,COMO
7071 CDCB71 01050 XXON CALL RSIN
7074 FE11 01060 CP XON
7076 20F9 01070 JR NZ,XXON
7078 7E 01080 COMO LD A,(HL)
7079 CDBE71 01090 CALL RSOUT
707C 23 01100 INC HL
707D 0B 01110 DEC BC
707E 79 01120 LD A,C
707F FE00 01130 CP 0
7081 20E7 01140 JR NZ,COMOUT
7083 78 01150 LD A,B
7084 FE01 01160 CP 1
7086 20E2 01170 JR NZ,COMOUT
7088 3E1A 01180 LD A,01AH
708A CDBE71 01190 CALL RSOUT
708D C32D40 01200 JP QUIT
01210 ;----- GET FILE FROM RS232 COM PORT
7090 CDF470 01220 GETCOM CALL READY
7093 010000 01230 LD BC,0000H
7096 210773 01240 LD HL,STORAG
7099 CDCB71 01250 COMIN CALL RSIN
709C B7 01260 OR A
709D 2002 01270 JR NZ,DATIN
709F 10F0 01280 JR COMIN
70A1 FE0A 01290 DATIN CP 0AH
70A3 28F4 01300 JR Z,COMIN
70A5 FE1A 01310 CP 01AH
70A7 2805 01320 JR Z,EOP
70A9 77 01330 LD (HL),A
70AA 23 01340 INC HL
70AB 03 01350 INC BC
70AC 18EB 01360 JR COMIN
70AE C5 01370 EOP PUSH BC
01380 ;----- PUT THE FILE ON THE DISK
70AF 11E771 01390 WDISK LD DE,DCB
70B2 210772 01400 LD HL,BUFFER
70B5 010000 01410 LD BC,0000H
70B8 CD2844 01420 CALL DINIT
    
```

Listing 1 continued

# MODEL 100

## MARKET YOUR OWN SOFTWARE!

ALPHAWARE proudly announces two new distribution programs designed to publish and market **YOUR** software: the **Complete Concept Marketing Program** distributes your software on a risk-free royalty basis. The **Independent Distribution Publishing Program** is perfect for those who want to mass produce their works and independently market them. Write for complete details, free samples, and enter the Model 100 software market today!

**MicroEditor II**, pagination, right, left & alternate margins, line and page numbering, headings, multiple data form letters, etc. **\$49.95**

**Mail Master**, sorts, alphabetizes, prints mailing labels, form letter compatibility with MicroEditor II **\$34.95**

**M100 Tape Log**, reference log designed specifically for the M100, columns for begin and end positions, file name, description, etc. Pad of 50 sheets **\$3.50**



2100 South Boulevard • Edmond, OK 73034

FREE  
Gift with your  
order See Below

"... your Word Machine is the best bargain I have found in software in my last two years of personal computing"

J. B. Tuscon, AZ

## PEL-TEK'S word machine 2.0

only \$20.00

A unique Word Processor for the TRS-80 Mod I/III  
32/48K Disk System — different in design, concept,  
function and price.

Your SECOND Word Processor shouldn't be a "me too" product but should be useful, economical, and truly different with features like these:

- Easy to learn Commands — great for kids, wives, and nervous secretaries
- Graphics mode — mix graphics with text on the same lines (printer support needed)
- On screen formatting — So what you type is what you print — including centering, word wrapping and tabbing
- Lowercase support — for unmodified Mod I's
- Create and Edit Communications files — like those sent to/from CompuServe
- Exit to D.O.S. and re-enter without losing text (even after re-booting!)
- Embed any ASCII character 0-255 in text. Supports any printer's special features
- Software forms control. Eject lines, pages, print multiple copies, print time entries
- Written in machine language — printer independent
- And much more

Send self-addressed, stamped envelope for FREE copy of Manual (\$37 postage, please)

\*\*\*\*\* SPECIAL — Free JUMPY™ Arcade game on disk

PEL-TEK • P.O. Box 1026 • Southampton, PA 18966  
Toll free order line 1-800-523-2445, Ext. 19  
in Pennsylvania call 1-800-346-7511, Ext. 19

Visa, Master Charge, check or money order  
Add \$2.00 Postage and Handling ■ PA. residents add 6% sales tax ✓ 324

Listing 1 continued

```

70BB C21571 01430 JP NZ,ERWRIT
70BE 210773 01440 LD HL,STORAG
70C1 C1 01450 POP BC
70C2 C5 01460 WRITLP PUSH BC
70C3 110772 01470 LD DE,BUFFER
70C6 010001 01480 LD BC.256
70C9 EDB0 01490 LDIR
70CB C1 01500 POP BC
70CC C5 01510 PUSH BC
70CD E5 01520 PUSH HL
70CE 210772 01530 LD HL,BUFFER
70D1 11E771 01540 LD DE,DCB
70D4 CD3944 01550 CALL DWRITE
70D7 B7 01560 OR A
70D8 C21571 01570 JP NZ,ERWRIT
70DB E1 01580 POP HL
70DC C1 01590 POP BC
70DD 05 01600 DEC B
70DE 78 01610 LD A,B
70DF FEFF 01620 CP 255
70E1 C2C270 01630 JP NZ,WRITLP
70E4 11E771 01640 WCLOSE LD DE,DCB
70E7 79 01650 LD A,C
70E8 32EF71 01660 LD (DCB+8),A
70EB CD2844 01670 CALL DCLOSE
70EE C21571 01680 JP NZ,ERWRIT
70F1 C32D40 01690 JP QUIT
01700 ;----- DISPLAY MESSAGES
70F4 21AF71 01710 READY LD HL,RDY
70F7 7E 01720 DSPLY LD A,(HL)
70F8 B7 01730 OR A
70F9 C8 01740 RET Z
70FA CD3300 01750 CALL VIDEO
70FD 23 01760 INC HL
70FE 18F7 01770 JR DSPLY
01780 ;
7100 213471 01790 ERCMD LD HL,ERR0
7103 1824 01800 JR DISPLY
7105 215071 01810 EROPEN LD HL,ERR1
7108 181F 01820 JR DISPLY
710A 11E771 01830 ERREAD LD DE,DCB
710D CD2844 01840 CALL DCLOSE
7110 216071 01850 LD HL,ERR2
7113 1814 01860 JR DISPLY
7115 11E771 01870 ERWRIT LD DE,DCB
    
```

```

7118 CD2844 01880 CALL DCLOSE
711B 217E71 01890 LD HT,ERR3
711E 1809 01900 JR DISPLY
7120 11E771 01910 TOOBIG LD DE,DCB
7123 CD2844 01920 CALL DCLOSE
7126 219971 01930 LD HL,TOBIG
7129 7E 01940 DISPLY LD A,(HL)
712A B7 01950 OR A
712B CA2D40 01960 JP Z,QUIT
712E CD3300 01970 CALL VIDEO
7131 23 01980 INC HL
7132 18F5 01990 JR DISPLY
02000 ;
7134 20 02010 ERR0 DEFM ' '
20 20 20 20
7139 2A 02020 DEFM '* COMMAND LINE ERROR *'
20 43 4F 4D 4D 41 4E 44
20 4C 49 4E 45 20 45 52
52 4F 52 20 2A
714F 00 02030 DEFB 0
7150 20 02040 ERR1 DEFM ' '
20 20 20 20
7155 2A 02050 DEFM '* FILE NOT FOUND *'
20 46 49 4C 45 20 4E 4F
54 20 46 4F 55 4E 44 20
2A
7167 00 02060 DEFB 0
7168 20 02070 ERR2 DEFM ' '
20 20 20 20
716D 2A 02080 DEFM '* SOURCE ERROR *'
20 53 4F 55 52 43 45 20
45 52 52 4F 52 20 2A
717D 00 02090 DEFB 0
717E 20 02100 ERR3 DEFM ' '
20 20 20 20
7183 2A 02110 DEFM '* DESTINATION ERROR *'
20 44 45 53 54 49 4E 41
54 49 4F 4E 20 45 52 52
4F 52 20 2A
7198 00 02120 DEFB 0
7199 20 02130 TOBIG DEFM ' '
20 20 20 20
719E 2A 02140 DEFM '* FILE TOO BIG *'
20 46 49 4C 45 20 54 4F
4F 20 42 49 47 20 2A
    
```

Listing 1 continued

# SUPERTAX™

## PROFESSIONAL QUALITY PERSONAL INCOME TAX PROGRAMS NOW IN OUR SECOND GREAT YEAR AND STILL UNDER \$30

- Don't wait until April 15. Use SUPERTAX to develop your tax strategy NOW.
- An ABSOLUTE MUST for every personal software library.
- Invaluable for YEAR-END tax planning or for tax return preparation.
- The friendliness and efficiency of these programs are setting standards by which others are judged.
- Highly acclaimed by tax professionals and laymen alike.
- Written by CPA.

- Available for IBM PC, Radio Shack Models I, II, III, 12 and 16, Apple II+, and Kaypro II (all require at least 48K).
- Also available on standard 8" CP/M using Microsoft BASIC.
- Data can be saved on disk.
- Changing any data item instantly changes entire return.
- Subsequent year versions of SUPERTAX available at 40% off.
- Programs are fully prompted and include manual loaded with valuable tax information and guidance.

**SUPERTAX I:** Using either screen or printer output. SUPERTAX I generates clear and concise summaries of Page 1 and 2 and Schedule A of FORM 1040, allowing you to see at a glance and quickly comprehend your tax situation. This program also prints an OVERALL SUMMARY of the return showing Adjusted Gross Income, Itemized Deductions, Taxable Income, Regular Tax, Income Averaging Tax, Minimum Tax and Payment Due or Refund — all of which are calculated by the program. SUPERTAX I also calculates the moving expense deduction, investment credit, taxable capital gains, political and child care credits, medical limitations, and much more. Input is fast and easy and changes can be made in seconds. This program actually makes tax planning a breeze.

**SUPERTAX III:** This package includes both the SUPERTAX I and SUPERTAX II programs PLUS a program to calculate and print Schedule C of the FORM 1040. Also included is a stand alone depreciation program which calculates and prints your depreciation schedule using both the old rules and the new ACRS rules. Output from the depreciation program is designed to serve as a supplement to IRS FORM 4562.

**SUPERTAX II:** Includes the efficient SUPERTAX I program as well as the more detailed SUPERTAX II program which makes all of the SUPERTAX I calculations, but which also PRINTS THE INCOME TAX RETURN. This program prints Page 1, page 2, Schedules A, B, and G (income averaging) of the FORM 1040 as well as FORM 3468 (investment tax credit) on standard government forms or on blank computer paper for use with transparencies. Any input item can be changed in seconds and the entire return recalculated almost instantly.

|                                                                                                             |
|-------------------------------------------------------------------------------------------------------------|
| <p><b>SUPERTAX I</b> — \$29.50</p> <p><b>SUPERTAX II</b> — \$49.95</p> <p><b>SUPERTAX III</b> — \$59.95</p> |
|-------------------------------------------------------------------------------------------------------------|

Add \$2.00 for postage & handling Add \$2.00 if ordering on 8" diskettes. Radio Shack models require 2 disk drives.

**ROCKWARE DATA CORPORATION**  
1635 Dorchester • PLANO, TX 75075 • (214) 596-0588

CP/M, Radio Shack & Apple II+ are trademarks of Digital Research, Tandy Corp. and Apple Computers, Inc.

Listing 1 continued

```

71AE 00      02150      DEFB  0
71AF 20      02160      RDY   DEFM  ' '
          20 20 20 20
71B4 2A      02170      DEFM  '* READY *'
          20 52 45 41 44 59 20 2A

71BD 00      02180      DEFB  0
          02190 ;
          02200 ;----- THE RS232 COM PORT
00E8        02210      RESET EQU  0E8H
00E9        02220      CONFIG EQU  0E9H
00EA        02230      CONTRL EQU  0EAH
00EB        02240      DATA  EQU  0EBH
          02250 ;
71BE C5      02260      RSOUT PUSH BC
71BF 4F      02270      LD     LD  C,A
71C0 DBEA    02280      SENT  IN  A,(CONTRL)
71C2 CB77    02290      BIT   BIT  6,A
71C4 20FA    02300      JR    JR  2,SENT
71C6 79      02310      LD     LD  A,C
71C7 D3EB    02320      OUT   OUT (DATA),A
71C9 C1      02330      POP   POP  BC
71CA C9      02340      RET
          02350 ;
71CB DBEA    02360      RSIN  IN  A,(CONTRL)
71CD CB7F    02370      BIT   BIT  7,A
71CF 2808    02380      JR    JR  2,NODATA
71D1 E638    02390      AND   AND  3BH
71D3 C20A71  02400      JP    JP  NZ,ERREAD
71D6 DBEB    02410      IN    IN  A,(DATA)
71D8 C9      02420      RET
71D9 AF      02430      NODATA XOR  A
71DA C9      02440      RET
          02450 ;
71DB D3E8    02460      RSRST OUT (RESET),A
71DD 3EE5    02470      LD     LD  A,CONFG
71DF D3EA    02480      OUT   OUT (CONTRL),A
71E1 3EE6    02490      LD     LD  A,BAUDS
71E3 D3E9    02500      OUT   OUT (CONFIG),A
71E5 AF      02510      XOR   XOR  A
71E6 C9      02520      RET
          02530 ;
0020        02540      DCB   DEFS  32
0100        02550      BUFFER DEFS  256
7307        02560      STORAG EQU  $
7000        02570      END   END   START
00000 TOTAL ERRORS
    
```

**Changes**

| Line | Label | Source | Stmt     |
|------|-------|--------|----------|
| 400  | HIMEM | EQU    | 4412 hex |
| 410  | CMD   | EQU    | 4225 hex |

**Additions**

| Line | Label | Source | Stmt         |
|------|-------|--------|--------------|
| 945  |       | INC    | A            |
| 1642 |       | LD     | A,(DCB + 12) |
| 1644 |       | DEC    | A            |
| 1646 |       | LD     | (DCB + 12),A |

Table 2. Model III modifications for disk storage program (COM/CD).

**The Great Escape**

by Scott Jones

Among the data omitted in the Model 100 user's manual are the escape codes, which control screen functions and cursor positioning. Though designed for use in the text editor, you can use the escape codes in Basic programs and with telecommunications (TELCOM) when transmitted from a host computer.

**Using Escape Codes**

To use an escape code in Basic, type

```
PRINT CHR$(27);"***";
```

replacing the asterisk with the letter in Table 3 after ESC, noting whether the letter is in upper- or lowercase.

To use an escape code in TELCOM, the host computer must first transmit an escape code (ASCII value of 27 decimal or 1B hexadecimal), then transmit the letter after ESC (see Table 3). The host computer should transmit the two characters together with no intervening space.

**Special Instructions**

Some of the codes in Table 3 require an explanation.

Use ESC T to set up the label line the Model 100 displays when you press the label key. ESC T protects the bottom line from scrolling and prevents the cursor from entering the bottom line. If the cursor is on the bottom line when you issue ESC T, it stays there until it's moved up or past the lower right position on the display.

ESC U unlocks the label line by allowing the cursor to move onto the last line; it does not clear the last line. If you press the label key and issue ESC U, you can print over the label line.

ESC V keeps the display from scrolling. When the cursor is in the lower right position and you've invoked ESC V, characters print over each other. ESC W allows scrolling to reoccur.

Use ESC Y to position the cursor. You must issue two characters after ESC Y: The first one determines the row; the second one, the column where the cursor goes. The cursor is positioned at the ASCII value of the characters minus 32. To move the cursor to row R and column C in Basic, type this command:

```
PRINT CHR$(27);"Y";CHR$(R + 32);CHR$(C + 32);
```

Note that column numbers begin with zero instead of 1. The upper left corner is Row 0, Column 0. This method may be more convenient than using the one value in a PRINT@ statement. ■

Contact Scott Jones at 3908 Stoney Ridge Trail, Charlotte, NC 28210.

Table 3. Escape codes.

| Escape Code | Description                                                                   |
|-------------|-------------------------------------------------------------------------------|
| ESC A       | Moves cursor up                                                               |
| ESC B       | Moves cursor down                                                             |
| ESC C       | Moves cursor to the right                                                     |
| ESC D       | Moves cursor to the left                                                      |
| ESC E       | Clears display and homes cursor                                               |
| ESC H       | Homes cursor (does not clear display)                                         |
| ESC J       | Clears from current position to end of display                                |
| ESC K       | Clears from current position to end of line                                   |
| ESC L       | Inserts a blank line at the current line; moves text below cursor down a line |
| ESC M       | Deletes the current line; moves text below cursor up a line                   |
| ESC P       | Turns on flashing cursor                                                      |
| ESC Q       | Turns off flashing cursor                                                     |

Table 3 continued



Table 3 continued

|           |                                                                       |
|-----------|-----------------------------------------------------------------------|
| ESC T     | Protects bottom line from scrolling and overprint (used by label key) |
| ESC U     | "Breaks through" label line                                           |
| ESC V     | Stops scrolling                                                       |
| ESC W     | Starts scrolling                                                      |
| ESC Y X Z | Moves cursor to row X and column Z (ASCII character values minus 32)  |
| ESC j     | Clears display and homes cursor                                       |
| ESC l     | Clears entire current line                                            |
| ESC p     | Displays characters as white-on-black (inverse)                       |
| ESC q     | Displays characters as black-on-white (normal)                        |

## Rooting Out the Problem

by William R. Harlow

Get to the roots of your math problems.

Two short programs for the Model 100, Plot (Program Listing 2) and Bisection (Program Listing 3), help you determine the root(s) of any given function.

Suppose you have an interval I(a,b) and a variable  $V_{old}$  associated with it. The equation needed to transfer points to an interval I'(c,d) with associated variable  $V_{new}$  is:

$$V_{new} = \frac{(d - c)V_{old} + (b*c - a*d)}{(b - a)}$$

The PSET command of the Model 100 allows 15,360 pixels. I decided to use a horizontal range of five to 235 units and a vertical range of three to 60 units. To establish the horizontal range, let XL and XR be the old values of X-to-the-left;X-to-the-right. Then XP equals  $(230*X + 5*XR - 235*XL)/(XR - XL)$ . Use a similar expression for the YP. The program tests whether it can plot the axes; then it plots the curve and PRINT@ puts the original ranges at the top of the plot. The program puts the function at line 140 in the form  $Y = a$  function of X.

The program includes two examples. When you run the first,  $Y = 4*\text{SIN}(3*X) - X^2 + 3$ , it asks for XL and XR. I keyed in -4,4. The suggested increment becomes .08. I keyed in .1. Too large an increment spreads pixels out; too small an increment slows the plotting speed. Line 30 gives a reasonable value.

The program then asks for Y below and Y above. I keyed in -15,10. Now plotting takes place reasonably quickly. Line 200 is a loop; press the break key to get control back. A subroutine to select the smallest and largest Y value would add about five lines to the program. If the plot is poor, you can always try again with different values. You can also estimate roots by expanding the plot range.

### The Bisection Program

The bisection method, one of the oldest known algorithms to determine a real root, isn't fast but it's fairly foolproof. If a function changes sign in an interval, there is at least one root in that interval and the bisection method will get it. Here is where the plot comes in handy: you can see places where roots occur or are close together and can use the appropriate values.

Line 300 is in the form  $F =$  a function of X. Here I used  $F = \text{LOG}(X) - X^2 + 4$ . X must be greater than zero. The plot revealed two roots: one close to zero and the other between

two and three. The program asks you to key in X left, X right, and epsilon. I keyed in .01, .2,  $1E-8$ . The program iterates, showing successive values. When ABS(F) is less than epsilon, the program displays the root and functional values. You can then run the program for other possible roots.

To check your program, print out the two roots of the log function. The six roots of the trigonometry function displayed show how the plot helps you see close roots and how the bisection method determines them. ■

*William R. Harlow has taught at the University of Cincinnati (the Department of Mechanical and Industrial Engineering, 836 Rhodes Hall, Cincinnati, OH 45221) since 1943.*

```

10 CLS
20 INPUT"KEY IN X-LEFT, X-RIGHT ";XL,XR
30 PRINT"A SUGGESTED INCREMENT IS ";(XR-
XL)/100
40 INPUT"KEY IN YOUR INCREMENT ";DX
50 INPUT"KEY IN YOUR Y-BELOW, Y-ABOVE
";YB,YA
60 CLS
70 YV=(60*YA-3*YB)/(YA-YB)
80 IF YV<3 OR YV>60 THEN 100
90 FOR H=5 TO 235:PSET(H,YV):NEXT H
100 XH=(5*XR-235*XL)/(XR-XL)
110 IF XH<5 OR XH>235 THEN 130
120 FOR V=3 TO 60:PSET(XH,V):NEXT V
130 FOR X=XL TO XR STEP DX
140 Y=4*SIN(3*X)-X^2+3
145 XP=(230*X+5*XR-235*XL)/(XR-XL)
150 YP=(60*YA-3*YB-57*Y)/(YA-YB)
160 IF YP<3 OR YP>60 THEN 180
170 PSET(XP,YP)
180 NEXT X
190 PRINTXL;"<x<";XR;TAB(27);YB;"<y>";YA

200 GOTO 200
    
```

Program Listing 2. Plot.

```

10CLS
20PRINT"      SOLUTION OF Y=F(X) BY
BISECTION"
30INPUT"KEY IN X-LEFT,X-RIGHT,EPSILON
";L,R,E
40X=L:GOSUB 300:FL=F
50X=R:GOSUB 300:FR=F
60M=(L+R)/2:X=M:GOSUB 300:FM=F
70CLS:PRINT:PRINT:PRINT
80PRINTTAB(10)USING"ITERATE IS +
#.###^";M
95IF ABS(FM)<E THEN 200
100IF FL*FM<0 THEN 120
110L=M:FL=FM:GOTO 60
120R=M:FR=FM:GOTO 60
200PRINT:PRINTUSING"      F(+
#.#####^)=+#.##^";M,FM
210PRINT:END
300F=LOG(X)-X^2+4
310RETURN
    
```

Program Listing 3. Bisection.

# UNITED SOFTWARE ASSOCIATES

PRESENTS

## ULTRA TERM

### A FULL FEATURED TERMINAL PROGRAM

The Ultra Term communications package is one of the easiest to use and most versatile communications programs available for the TRS-80. It includes a full featured intelligent terminal program, with all the popular features of competing programs costing two to three times as much, and some new features that can't be found anywhere else at any price. Ultra Term also includes a self relocating host program, and hex conversion utilities for bulletin board downloading. Some of Ultra Term's unique features are:

- Supports both manual and auto dial modems
  - Exclusive Ultra Term direct to disk file transfer mode, allows unattended operation at the receiving computer.
  - Exclusive split screen feature allows simultaneous two way communications without confusion.
  - Line printer support with a 1K print buffer.
  - Half and full duplex support.
  - Universal ASCII format file transfer with a 33K Buffer.
  - A full featured host program.
- |                   |         |                 |          |
|-------------------|---------|-----------------|----------|
| • ULTRA TERM      | \$59.95 | • LYNX MODEM    | \$249.00 |
| • INFOEX 80       | \$99.95 | • ANCHOR 1-300  | \$89.95  |
| • M-TERM          | \$69.95 | • HAYES 300     | \$249.00 |
| • COMMBAT         | \$44.95 | • HAYES 1200    | \$619.00 |
| • PIANO SOFTWARE  | \$34.95 | • U.S. ROBOTICS |          |
| • ORCHESTRA 85/90 | \$89.00 | 1200            | \$399.95 |

ORDERS 305/965-3496 BBS 305/842-2687  
TECHNICAL INFORMATION 305/842-8805 (5:00-9:00 E.S.T.)

734 Flamingo Way, North Palm Beach, FL 33408 ☐332

## C-Notes

### Autoliner

by Richard Ramella

Autoliner (Program Listing 4) gives the Model 100 an automatic line-numbering capability similar to what Level II Basic provides.

The program lets you specify the starting line number of a program and the increment for each succeeding line. It is most useful when keying in published programs with regular line increments—100, 110, 120, and so on. Autoliner can also handle irregular line numbers—for instance, a line number of 237 in a program otherwise marked by lines in increments of 10.

Autoliner creates a text (.DO) file that becomes the program you key in.

#### Program Operation

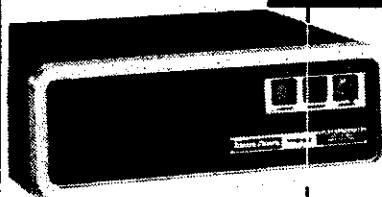
First the prompt Name of Program ? appears. Name the program as you wish. If the name has more than six characters, the text file name becomes the first six characters in the name. This is required because a text file must be named in six or fewer characters.

The next prompt is Starting line number ? and may be any number from zero to 65,529, the highest line number the Model 100 accepts. Type the number and tap the enter key.

The final parameter-setting prompt is Line increment ? and

ALWAYS AT SALE PRICES **TRS-80** ALWAYS AT SALE PRICES

OUR TRS-80s Have the Radio Shack warranty

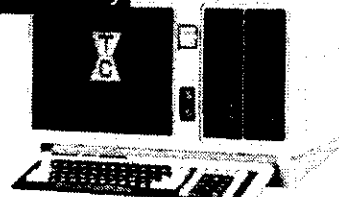


HARD DISK DRIVES

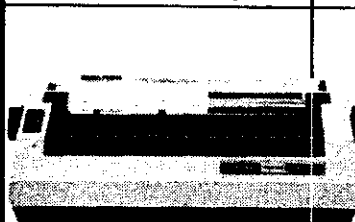


**WE SHIP FAST!**

DFW COMPUTER CENTER  
326 MAIN ST., GRAPEVINE  
TEXAS 76051 — (817) 481-7283



MODEL 16-B

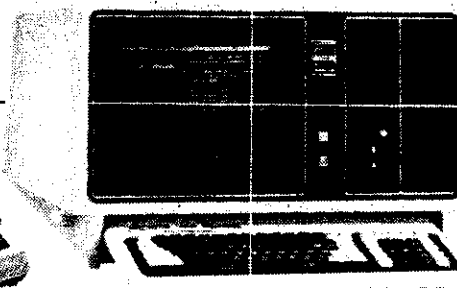


DAISY WHEEL PRINTERS

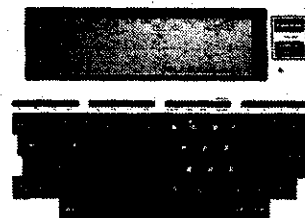
To get YOUR SALE price CALL:  
**1-800-433-SALE**



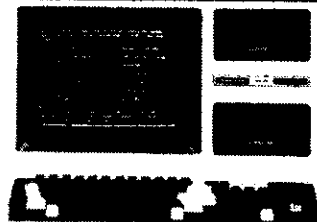
DOT MATRIX PRINTERS



MODEL 12



MODEL 100



MODEL 4

☐282

may be any number from one up. Type the number and tap the enter key.

Now you're ready to key in the program. The prompts appear starting with the first line number and continuing in the increments you set.

**An Example**

Run the program Autoliner. Answer the first prompt by typing TEST and tapping the enter key. Answer the second prompt by typing 100 and hitting the enter key. Answer the third prompt by typing 10 and pressing the enter key. Line 100 appears with a prompt; answer it REM TEST and tap the enter key. Answer the line 110 prompt by typing PRINT "IT WORKS!" and hitting the enter key. Answer the line 120 prompt by typing ! and pressing the enter key. Answering any line prompt with an ! ends the program.

As it stops, the program announces See TEST.DO for program. TEST is the name of the program you have keyed in.

Then type RUN "TEST" and the program creates and runs a Basic version of the text file. Break into this new program and type SAVE "TEST" and tap the enter key. This puts the program in a Basic (.BA) file where it will be safe. Then type KILL "TEST.DO" and tap enter. This kills the unneeded text (.DO) version of the program.

Don't attempt to run your new program until you have completely keyed it in. It doesn't exist as a Basic program until

From the makers of MAP:

# MAP-MATE™

## Relational Database Management System

- Complements MAP's free-text information retrieval system
- Ideal for numeric data and structured text (information that fits into rows and columns)
- Menu driven
- Versatile report generator
- Rapid multiple-field searches

Prices for MAP and MAP-MATE range from \$35.00 to \$295.00. Programs for TRSDOS and CP/M operating systems.

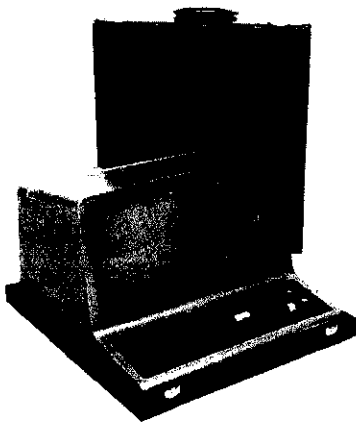
Send for information on MAP and MAP-MATE

SOFTSHELL Corporation  
The Small Computer Specialists for Professionals  
P.O. Box 18522, Baltimore, MD 21237

✓ 57

**Computer Case Company**

# WE MAKE A GREAT CASE FOR YOUR RADIO SHACK.



RS204

One size does not fit all. Our cases are designed for specific hardware configurations. When you put your computer in our case, it fits hand-in-glove. Once your equipment is safely inside the attache-style carrying case, it never has to be taken out again. To operate, simply remove the lid and connect the power. To store your computer, disconnect the power, enclose your disks, working papers, and manuals in the compartments provided, and attach the lid. It's as easy as that.

CALL TOLL FREE: (800) 848-7548

|         |                                                             |       |
|---------|-------------------------------------------------------------|-------|
| • RS201 | Model I with expansion unit and drives                      | \$109 |
| • RS204 | Model III or IV                                             | 129   |
| • AP106 | Amdek Color I, II or III Monitor                            | 119   |
| • P401  | Paper Tiger Printer (400/445/460)                           | 99    |
| • P402  | Centronics 730/737 & Radio Shack Printer                    | 89    |
| • P403  | Epson MX70 or MX80, Microline 82A Printer or Color Computer | 89    |
| • P404  | Epson MX100 Printer                                         | 99    |
| • P405  | IDS 560 or Prism 132 Printer                                | 109   |
| • P406  | C. Itoh Starwriter/Printmaster F-10 Printer                 | 119   |
| • P407  | Okidata Microline 83A or 84 Printer                         | 99    |
| • P408  | C. Itoh Prowriter 2 Printer                                 | 99    |
| • P409  | C. Itoh Prowriter (Apple Dot Matrix) or NEC PC8023 Printer  | 89    |
| • P410  | Epson FX80 Printer                                          | 109   |
| • IB501 | IBM Personal Computer with Keyboard                         | 129   |
| • IB502 | IBM Monochrome Monitor                                      | 99    |
| • CC50  | Case Cart                                                   | 79    |
| • CC80  | Matching Attache Case (5")                                  | 85    |
| • CC90  | Matching Attache Case (3")                                  | 75    |
| • CC91  | Matching Accessories Case (5 1/4" Diskettes, Paper, etc.)   | 95    |
| • CC92  | 5.25" Diskette Case (Holds 75 Diskettes)                    | 49    |



you end things by typing ! and tapping the enter key.

If you find an irregular number in the listing, answer the line prompt with an asterisk and the program responds New line number and data ?. Answer this by typing the irregular line number and the program information that accompanies it. Then tap the enter key, and the program returns to the regular line increments you specified.

To correct a line, answer a line prompt with the asterisk and then type in the line number you want corrected and the correct version of the line. While useful for short lines, the program doesn't allow the full-screen text editing available when you've turned your program into a Basic file. You can put the same line number twice into a text file, but when you run it as a Basic program, it eliminates the old line and uses the new one.

A few lines at the start serve no purpose other than goof-proofing the program. If your starting line number is 65,000 or more, the program tells you the number of lines with which you have to work. If the increment you choose allows 150 or fewer lines for the program, Autoliner tells you that number. ■

*Richard Ramella can be reached at 1493 Mt. View Ave., Chico, CA 95926.*

THE ULTIMATE COMPUTER. THE NEW  
**LNW80 2 +**

NOW:

MODEL 1, 3, and 4 COMPATIBLE!!

CP/M 2.2 and 3.0 COMPATIBLE!!

SOON:

IBM MS-DOS and CP/M-86 COMPATIBILITY!!

PLUS add up to 1 Megabyte of RAM!!

High Speed RAMDISK is coming!

**SYSTEMS START AT \$1999**

Includes: LNW80-2+, Hi-Res Green Monitor, One Disk Drive, CP/M 2.2, DOSPLUS, Microterm, Electric Pencil, Electric Spreadsheet, Chartex, and the LNW Small Business and Professional Accounting Software Series.

CALL US FOR THE BEST PRICES AVAILABLE  
IN STOCK NOW! Ready for immediate delivery!

**EXCELLONIX**

(714) 973-1939 \*\* (213) 650-5754

```

100 REM * Autoliner * TRS-80 Model 100
8K / Richard Ramella
110 CLS
120 MAXFILES=1
130 CLEAR 500
140 DEFSTR A
150 INPUT "Name of Program";L$
160 C=ASC(LEFT$(L$,1))
170 IF C>96 THEN C=C-32: L$=CHR$(C)+
RIGHT$(L$,LEN(L$)-1)
180 IF L$="" THEN GOSUB 420: GOTO 150
190 M$=LEFT$(L$,1)
200 IF ASC(M$)<65 OR ASC(M$)>90 THEN
GOSUB 420: GOTO 150
210 IF LEN(L$)>6 THEN Z$=LEFT$(L$,6)
ELSE Z$=L$
220 OPEN Z$ FOR OUTPUT AS 1
230 CLS
240 INPUT "Starting line number";B
250 B=INT(B)
260 IF B<0 THEN CLS: GOSUB 420: GOTO 240

270 IF B>65000 THEN PRINT "The Model
100 allows line numbers up to 65529. You
have"65529-B"lines with which to": PRINT
"work.": GOSUB 570
280 CLS
290 INPUT "Line increment";D
300 F=INT((65529-B)/D)
310 IF F<150 THEN PRINT "You
have"F"lines with which to work.":
GOSUB570
320 CLS
330 PRINT B;
340 LINEINPUT A
350 IF A="1" THEN 530
360 IF A="*" THEN PRINT "New line number
and data": INPUT A: PRINT #1,A: GOTO 330

370 A=STR$(B)+A
380 PRINT #1,A
390 A=""
400 B=B+D
410 GOTO 330
420 PRINT "Impossible. Please try again"

430 GOSUB 500
440 FOR T=1 TO 5
450 BEEP
460 NEXT
470 GOSUB 500
480 CLS
490 RETURN
500 FOR T=1 TO 500
510 NEXT
520 RETURN
530 CLS
540 PRINT "See "Z$".DO for program"
550 PRINT L$
560 END
570 PRINT
580 INPUT "Tap a key to continue";G
590 CLS
600 RETURN
610 END
    
```

*Program Listing 4. Autoliner.*



# RAM FILES

## Disassembler Debug

Several errors appeared in my Disassembler program (September 1983, p. 283) that make it run improperly. The following is a list of corrections:

RCLA in line 20 should be RLCA.

Lines 160 and 170 are the same in the listing. Line 160 should read:

```
160 DATA "LD L,D","LD L,E","LD
L,H","LD L,I","LD
L,(HL)","LD L,A","LD (HL),B"
```

Don't forget that tabs, not spaces, separate the mnemonics and operands.

Line 170 is missing a right parenthesis. "LD (HL,H" should read "LD(HL),H".

Lines 1760 and 1770 should end in RETURN:ELSE RETURN.

Finally, the two spaces in line 10530 in Fig. 1 should be carets for exponentiation (press the shift/6 keys).

I also have a few changes in the mnemonics. When I wrote the program I used Z80 code, but no relative jumps or alternate register sets exist in the 8085. You should make EX AF,AF' and EXX undefined. Leave EX DE,HL, and EX (SP),HL as they are.

All but two JR...@ (including DJNZ @) should be changed to undefined. JR NZ,@ should be RIM and JR NC,@ should be SIM. RIM is the read interrupt mask, and SIM is the set interrupt mask.

I put the equivalent restart commands for the Z80 and the 8085 in Table 1. Their functions are identical.

*David A. Cloutier  
Bullard Road  
North Brookfield, MA 01535  
CIS #75705,730*

## CR Patch

Most printers have a switch labeled "Auto LF, after CR." Most computers (e.g., the IBM PC) require this switch to be off to distinguish between a line feed (LF) and a carriage return (CR).

TRS-80 computers require this switch be on so when

| Z80     | 8085  |
|---------|-------|
| RST 00H | RST 0 |
| RST 08H | RST 1 |
| RST 10H | RST 2 |
| RST 18H | RST 3 |
| RST 20H | RST 4 |
| RST 28H | RST 5 |
| RST 30H | RST 6 |
| RST 38H | RST 7 |

Table 1. Restart commands for the Z80 and the 8085.

the computer sends a carriage return, the printer supplies the implied line feed.

If you use two different computers with the same printer, this protocol can be a hassle. We wrote the patch in Program Listing 1 to eliminate this problem.

Model 100 document files have CR/LF pairs embedded in them (try using control/P, control/M, control/P, control/J in Text), but the printer driver strips out any line feed immediately following a carriage return. This patch intercepts the printer driver between each character so it never knows that it just printed a carriage return.

We offer this patch with some trepidation, since it redirects a system jump vector into user memory. It presents no problems under most circumstances, but if you leave it switched on and load a program that clears memory above 62700 decimal, the results are unpredictable, and probably disastrous if you try to print anything.

The programs from the Portable Computer Support Group (PCSG) are an exception to the above caveat. They are fully compatible with this patch, even though they reserve memory above 62700.

*Michael Stanford &  
Robi Robinson  
Portable Computer Support Group  
11035 Harry Hines #207  
Dallas, TX 75229  
Support Hotline: 214-351-0564*

## Built-in Criticism

Publicity for the Model 100 always emphasizes the five built-in programs, including the editing program, Text; the scheduler program, Schedl; and the address organizer, Addrss.

Schedl and Addrss consist of nothing but the search and find parts of Text, with the rest of the text modification capabilities turned off so you can't accidentally clobber any of your address or schedule information. This is OK, but it's more like one and a half programs than three.

In addition, Addrss only operates on the ADRS.DO file, and Schedl can only operate on the NOTE.DO file. This is quite inconsistent.

Next time around, Tandy should combine Schedl and Ad-

```
0 ' Program CRLFIX V1.0 830827
10 ON KEY GOSUB 20,30:KEY ON
12 CLS:PRINT@130,"PCSG Line Feed Patch";:P
RINT@281,"On Off";
14 GOTO 14
15 DATA 245,58,172,250,254,13,194,48,245,6
2,10.50,172,250,241,201
20 CLEAR 256,62700:FOR I!=62754 TO 62769:
READ A:POKE I!,A:NEXT
25 POKE 64228,34 :POKE 64229,245: MENU
30 POKE 64228,243:POKE 64229,127: MENU
```

Program Listing 1. Patch to insert a line feed after a carriage return.

drss into a single program that operates on any Model 100 file. You can then keep and search multiple lists of data, addresses, jobs, and so on.

A.E. Siegman  
Edward L. Ginzton Laboratory  
Stanford University  
Stanford, CA 94315

**Bar Graph Program**

I am looking for a bar graph program for the Model 100 that either produces printable data from the program from "The Year in Review" (August 1983, p. 207) or from another program containing a hard copy print routine.

The program must work on my A.M.P. 120 printer.

T. Hardy McCoy  
7718 D.S. Victor  
Tulsa, OK 74136

**Cram Notes**

Brad Dixon's Cram 100 program (October 1983, p. 283) is a barrel of fun.

He mentions sound capabilities, so I changed lines 100 and 130 to values other than 2216 for variety. In line 200 I put SOUND0,2: between BEEP: and NEXT.

The best change is to put Avg in addition to Turns and High at the top of the screen. Add G=G+1 to line 40, TT=TT+N to line 200 (before CLS), and add AV=TT/G. I'm looking forward to the expansion of C\*Notes.

David Dunn Thomas  
2308 Chetwood Circle #103  
Timonium, MD 21093

**Screen RAM Location**

While stepping through the Model 100's RAM (24K), I found that the screen RAM is located at 65024 to 65343, inclusive, for a total of 320 bytes.

While I can't POKE characters on the screen, I can PEEK them. Even stranger, the information POKEd to the screen is in RAM, but is not on the screen! I have displayed the keyboard matrix setup in Table 2.

I use Program Listing 2 to PEEK about in RAM. The commands are:

- CTRL to enter a new address.
- SHIFT to add one to the address.
- GRPH to subtract one from the address.

The program displays both the decimal value and the ASCII value if it is printable. The best feature of this program is that it constantly updates information on the screen.

I hope the readers of 80 Micro can make use of this information.

Joseph Gaudreau  
314 E. Main St.  
Malone, NY 12953

**"Worksheet 100" — A Spreadsheet Program for the TRS-80™\* Model 100 Portable Computer**

The "Worksheet 100" works with spreadsheets of up to 16 columns by 40 rows with 24K of RAM or up to 16 columns by 60 rows with 32K. Eight main functions using the programmable function keys plus three entry modes provide large spreadsheet convenience. The "Goto" function, doubling as a "search function", will speedily move the marker to any cell given coordinates, column and row labels or cell contents. Full arithmetic operators plus summation, averaging, maximum and minimum functions are provided.

Worksheets can be saved or loaded using RAM or cassette. A very useful group of worksheet templates (with formulas) is provided ready to load. These are "Expense Report", "Sales Report", "Service Report", "Weekly Schedule", "Personal Tax Worksheet", and "Personal Finances I and II".

The "Worksheet 100" and the seven ready to use templates are supplied on a single cassette and require 24K RAM.

**The "Worksheet 100" and seven useful templates— for only \$89.95**

Mail this Order Slip to **Diaz Enterprises**  
P.O. Box 4609  
Mountain View, CA 94040

Please send me the "Worksheet 100" and 7 useful templates for \$89.95

Payment Enclosed \_\_\_\_\_ MasterCard \_\_\_\_\_ Visa \_\_\_\_\_ Amex \_\_\_\_\_

Card # \_\_\_\_\_ Signature \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Postage and handling included within the USA and Canada. California residents should add 6.5% sales tax.

\* TRS-80 is a trademark of Tandy Corporation. —183

| Address | Value found when PEEKing matrix: |      |       |      |       |       |       |       |
|---------|----------------------------------|------|-------|------|-------|-------|-------|-------|
|         | 1                                | 2    | 4     | 8    | 16    | 32    | 64    | 128   |
| 65425   | Z                                | X    | C     | V    | B     | N     | M     | L     |
| 65426   | A                                | S    | D     | F    | G     | H     | J     | K     |
| 65427   | Q                                | W    | E     | R    | T     | Y     | U     | I     |
| 65428   | O                                | P    | BRAKT | ::   | " "   | <     | >     | /?    |
| 65429   | 1                                | 2    | 3     | 4    | 5     | 6     | 7     | 8     |
| 65430   | 9                                | 0    | —     | + =  | LA    | RA    | UA    | DA    |
| 65431   | SPC                              | DEL  | TAB   | ESC  | PASTE | LABEL | PRINT | ENTER |
| 65432   | F1                               | F2   | F3    | F4   | F5    | F6    | F7    | F8    |
| 65433   | SHIFT                            | CTRL | GRPH  | CODE | NUM   | CAPS  | ????  | BREAK |

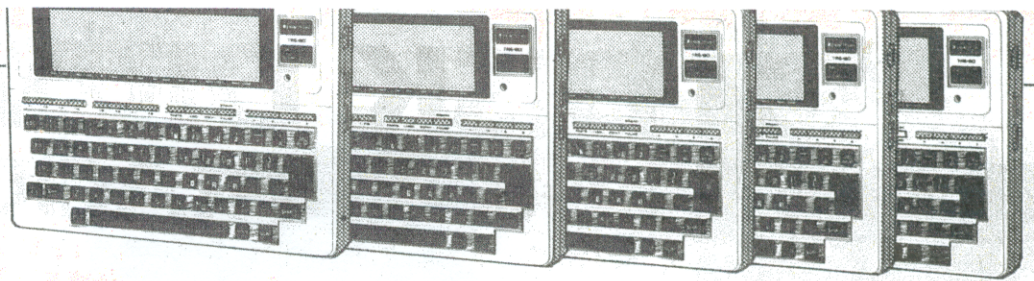
Table 2. Keyboard matrix setup for the Model 100.

```
100 ' "RAMER.BA" - A RAM SCANNER
110 PRINT:PRINT:INPUT"Start @:A:CLS
120 PRINT@,A;:FORP=@TO6
130 PRINT@+4*P,PEEK(A+P);" ";
150 IFPEEK(A+P)>32THEN PRINT @+4*P,CHR$(PEEK(A+P));
ELSE PRINT @+4*P," ";
160 NEXT
170 C=PEEK(65433):IFC=@THEN120
180 IFC=1THENA=A+1
190 IFC=2THEN110
200 IFC=4THENA=A-1
500 GOTO 120
```

Program Listing 2. RAM screen POKE for the Model 100.

C-NOTE FOR FEBRUARY 1984





## Seven 100 Percent Solutions

by Rolf A. Deininger

Once upon a time, there was darkness in Tandyland. The apples and oranges and other fruits blossomed and the Big Blue grapes grew bigger and bluer each day, but the models numbered one through four would not sell.

And John, prophet of Tandyland, looked throughout the warehouses at the other items, the works of the mighty Archer and others, and chose batteries, saying, "If we cannot sell computers, then let us sell batteries."

"But how do we do this?" asked the sages and shopkeepers.

And John said, "Realistically. Let us find a toy which everyone on this earth needs, and one which devours batteries. Let it consume two hundred AA batteries in a year."

Now the shopkeepers were conservative, and were abashed by the boldness of John. They thought they could sell but 100 batteries per year, and thus the name Model 100 was born.

The wise men searched high and low for such a product, until Jon, aide of John, prophesied, "Let us look to the land of the rising sun, and the software bedlam of Bellevue."

So they looked to Japan and Washington, and men there created a portable computer. And John, Jon, and the sages and shopkeepers rejoiced, because they saw that the toy was good and that it would sell many batteries.

And later, Jon left for Bellevue himself, to be received by open Gates. But here endeth the story of John and Jon. And our story begins.

### The Problem

In my first two weeks with the Model 100, I went through three sets of alkaline AA batteries at a cost of roughly \$10. Scared by the prospect of a \$200 annual battery bill, I bought

an ac adapter (\$5.95)—which freed me from alkalines, but kept me within a 75-inch radius of the nearest wall outlet.

A 50- or 100-foot extension cord would give me some leeway, but not enough to write a story at an airport. The gang ways are long, the doors of the plane crimp the cord, and when the plane taxis from the gate, I lose both the power and the cord.

What I needed was an auxiliary power supply capable of giving my 32K RAM Model 100 the 60 milliamperes (mA) it needs for longer than the 20 hours available on alkaline batteries (20 hours  $\times$  60 milliamps = 1,200 mAhrs or milliamperhours).

### The Solutions

I came up with seven possible solutions. Solution 1 was to replace the Model 100's four alkaline batteries with rechargeable nickel-cadmium batteries—about \$10 for four NiCads, plus \$7 for a battery charger. Unfortunately, this doesn't work very well.

The NiCads produce 5 volts when fresh and then drop to 4.8 volts, triggering the Model 100's low-power light. At this voltage, the Model 100 draws 75 mA, and up to 120 mA while dialing the telephone.

Five NiCads would provide the necessary 6 volts, but there's no room for a fifth cell in the battery compartment.

Solution 2, by contrast, works like a charm: Put only three NiCads into the 100 and tape a two-battery holder (Radio Shack part number 270-382, 79 cents) to the top of the case to hold a few more NiCads (see Photo 1).

With the outside battery pack properly connected to the partially empty battery compartment, the NiCads produce about 450 mAhrs, powering the computer for seven to eight hours between recharges.

Fitting five batteries into a recharge holder for four is a problem, but so is this solution's ugly appearance.

Solution 3 is a similarly simple, clumsy kludge. A 6-volt



Photo 1. Model 100 with two AA batteries taped next to the display.



Photo 2. Gel cells—from a 9.5-Ahr Sears cell to a 1-Ahr cell.



lantern battery (\$3 to \$13) and a homemade cable using Radio Shack's coaxial adapter plugs (Radio Shack part number 274-1551, \$1.69 for two) provide from three to 40 ampere-hours (Ahrs) of power. A 20-Ahrs Duracell MN908 battery (\$6), for instance, should power your Model 100 for over 300 hours; a 40-Ahrs MN918 battery (\$13) for over 650.

Solution 4 is a compromise: Attach the plug described in Solution 3 to a six-cell battery holder for C cells. Put five rechargeable NiCads (about \$16, plus \$12 for a charger) in the holder. For the sixth cell space, use any dead C battery. Solder a wire from the bottom to the tip of the dead battery so it serves as a space filler only.

A fully charged set should last between 15 and 20 hours. If you don't want to use a dead cell to fill the holder, you can use six NiCads; 7.2 volts won't hurt the 100 and the current will go down to about 50 mA.

### Gel Cells and Solar Power

Solutions 5 and 6 are more exotic. Gel cells are normal electrolyte batteries (like your car battery), except that their electrolyte solution is gelled; they are excellent power supplies, a bit heavy, but with ample reserve for remote computing.

Photo 2 shows a 9.5-Ahrs Sears cell (toy catalog part number 49 N 86522, \$15) and three Globe Industries cells (7.5, 1.8, and 1 Ahrs). You'll have to make the appropriate cables and buy a charger (\$6 to \$10).

Returning to Radio Shack, Tandy's solar panel (part number 277-1250, \$25) delivers 80 mA at 6 volts in bright sunlight (see Photo 3). If you can keep the panel in the sun and yourself and the 100 in the shade, it's okay. However, it doesn't work at night, and it makes it hard to take notes in a lecture hall.

Solar power may not be practical, but my other solutions are bargains compared to alkaline AA cells. With 20 hours' computing per week, you'd need 52 sets (208 batteries) per year—at about \$3 per set, a \$156 total. By contrast, just three Duracell MN908 lantern batteries will provide a year's worth of use for \$18.

Rechargeables are even cheaper. Solution 4's five NiCad C cells will survive several hundred weekly recharges, or at least three years' service. If the cells, charger, and cable cost \$30,

that's a modest \$10 per year. And the Sears gel cell mentioned in Solution 5, about \$21 with charger and cable, will run your 100 for \$7 a year.

### The Ideal Solution

The best bargain of all? Solution 7: Enroll 18 friends or relatives in Radio Shack's Battery-of-the-Month Club (see p. 150 of the 1984 Radio Shack catalog). Each month, have your friends pick up the give-away AA battery. Except for the gas they use trotting to participating Radio Shack dealers, it's free. ■

*Rolf A. Deininger is a professor of environmental health at the University of Michigan's School of Public Health, Ann Arbor, MI 48109.*

## The Shadow Knows

by Richard Ramella

Lamont Cranston, or The Shadow as old-time radio listeners will remember, had the ability to cloud men's minds. I've named this short program in his honor. LAMONT.BA protects sensitive text material in the Model 100 by encoding it (see Program Listing 1).

The encrypted files thwart the casual sneakpeek as well as the most determined snooper. In addition, LAMONT.BA befuddles many line printers' efforts to print the coded version. And the same program both encodes and decodes text files.

LAMONT works by increasing the ASCII value of each of your old file's characters by 100, then sending them to a new .DO file. The resulting program code looks like gibberish because it comprises both pictographic and non-English characters instead of conventional ASCII character values.

### Using the Program

First, create a short .DO file to test the system. Call it Test.DO.

Then run LAMONT. Answer the first prompt, "Will you be 1-Coding? 2-Decoding?", by typing 1 and pressing the enter key. Answer the next prompt, "File to be coded?", by typing TEST and hitting the enter key. Answer the last prompt, "Name of new file?", with TEST2 and tap the enter key.

Once the program writes the new encoded or decoded file, it displays "File transferred," and beeps 20 times. It also reminds you to destroy the plaintext version by stating: To kill old file, type "KILL TEST.DO" and tap enter.

To see the coded version of your text, press the F8 key to enter the menu mode. Put the cursor over the file named TEST2 and press the enter key.



Photo 3. A solar power panel for the Model 100.

### The Key Box

The programs in "The Shadow Knows," "Backing Up the 100," and "Write Away" run in 8K RAM.



## Decoding Material

To decode your text, run LAMONT.BA again. This time answer the first prompt by pressing 2 and then the enter key. The old file is now TEST2, the new file, TEST3, and the program has restored the Text file to plaintext.

To store a Basic program file in plaintext, first run the program, then stop it by pressing the shift and the break keys simultaneously. Note that all subsequent references to the word program in commands denote the name you gave the program, not the word itself. Type SAVE"PROGRAM.DO" and press the enter key.

To reclaim the program in Basic, decode the coded file, then type SAVE"PROGRAM" and wait until it begins to run. Stop the run by pressing the shift and break keys together, then type SAVE"PROGRAM" and the program returns to a Basic file.

## Further Security

You can, of course, enhance this method in order to provide more security. Line 280 is a likely place to customize LAMONT.BA. Note that 100 appears twice in that line. You can substitute any single number from one to 133 to yield a different set of encryption symbols. Be sure to use the same number in both places.

If you're really paranoid, you can store an encoded text file or program in two different files, sending every other charac-

ter to alternate files, then have the same program reassemble the file in plaintext. ■

*You can reach Richard Ramella at 1493 Mt. View Ave., Chico, CA 95926.*

## Backing Up the 100

by Bryan R. Leipper

In spite of the Model 100's constant memory feature, you still need to back up your files. If you maintain several document files, it's inconvenient to save them all to tape by going in and out of Text or by typing the file names. Here's a way to automate that process.

BKUPDO (Program Listing 2) PEEKs into memory to find the file names for all of the document files in the Model 100 menu. The program asks you whether you want to store each file to tape. It notes the files you select for tape back-up by typing a Y, stores them in a cassette file, and then reads the files from memory and stores them to tape.

To ensure that the 100 properly stores the files on cassette, BKUPDO has a verify function that compares the tape files with the RAM files. It tells you if you changed a line in the file, changed the length of a file, or if there is an input/output error in the file. One error that doesn't show up directly is when the 100 cannot find a file (probably due to a recording problem). In this case the tape machine continues to run, and you have to hit the break key or control/C to interrupt the program.

When you want to reload the programs, BKUPDO first reads the list of file names from cassette and then reads the files and places them in memory. Be careful to avoid overwriting updated files that you haven't backed up.

## How It Works

The 100 stores its directory of user files starting at memory location 63930 (F98A hexadecimal (hex)). Each entry is 11 bytes long. The first byte identifies the type of file, with 80 hex indicating a Basic program file and a C0 hex (192 decimal) indicating a document file. This is followed by a 2-byte starting address for the file. Then the Model 100 stores the file name as 6 bytes, padded on the right with blanks. The last 2 bytes are the ASCII codes for the extension, which should be .DO for document files.

Knowing this, you can write a Basic program to PEEK at the addresses looking for the proper codes in order to extract a file name. The computer then reads and stores this file on tape.

When you run the program, you must first decide whether you want to read files from the tape, save files to tape, or verify what you've previously saved. If you select R to read the tape, make sure you have a proper back-up tape in the cassette machine ready to play. The program looks for cassette file FINM to read a list of names and then will load each file just as it was recorded. All you have to do is wait a few minutes until the program loads all of your document files to RAM.

Typing S makes the program search memory for document file names. As it finds them, the program asks if you want the file saved to cassette. An upper- or lowercase Y sets the file name in an array for later processing. Any other response bypasses back-up of the file named. After the program ex-

```

100 REM * Lamont * TRS-80 Model 100 8K *
Richard Ramella
110 MAXFILES=2
120 CLS
130 PRINT "Will you be...
140 PRINT "1 - Coding
150 PRINT "2 - Decoding
160 INPUT Z
170 IF Z<>1 AND Z<>2 THEN 120
180 PRINT "File to be ";
190 IF Z=2 THEN PRINT "de";
200 INPUT "coded";A$
210 INPUT "Name of new file";B$
220 IF LEN(B$)>6 THEN CLS: PRINT "New
file name must be in 6 or fewer
characters.": GOTO 210
230 BEEP
240 PRINT "Translation begun. Patience,
please."
250 OPEN A$ FOR INPUT AS 1
260 OPEN B$ FOR OUTPUT AS 2
270 C$=INPUT$(1,1)
280 IF Z=1 THEN C$=CHR$(ASC(C$)+100)
ELSE C$=CHR$(ASC(C$)-100)
290 PRINT #2,C$;
300 IF EOF(1) THEN 320
310 GOTO 270
320 PRINT "File transferred."
330 FOR T=1 TO 20
340 BEEP
350 NEXT
360 PRINT "To kill old file "A$",
370 PRINT "Type " CHR$(34)"KILL
"A$".DO"CHR$(34)" and tap Enter
380 END

```

Program Listing 1. LAMONT.BA.




# YORK 10™ CASSETTES



**COMPUTER GRADE  
DATA TRAC  
BLANK CASSETTES**

**C-05, C-06, C-10, C-12, C-20, C-24, C-30**

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

-  **BASF-LHD (DPS) world standard tape.**  
Premium 5 screw shell with leader.
-  **Internationally acclaimed.**  
Thousands of repeat users.
-  **Error Free • Money back guarantee.**

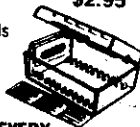
**\* 500 C-12's or C-10's — 38¢ each**  
(w/labels, add 4¢ • Shipping \$17./500  
500 Boxes 13¢ ea. • Shipping \$10./500

**CASSETTE STORAGE CADDY**  
Holds 12 cassettes  
w/o boxes \$2.95



**TRACTOR FEED • DIE-CUT  
BLANK CASSETTE LABELS**  
WHITE: \$3.00/100 \$20.00/1000  
**COLORED LABELS • Pastels -**  
Red, Blue, Green, Yellow, Lavender  
\$4.00/100 \$30.00/1000

**FREE**  
1 CADDY WITH EVERY  
4 DOZ. CASSETTES PURCHASED



Call: 213/700-0330 for IMMEDIATE SHIPMENT  
on Credit Card Orders



**ORDER NOW** MAIL TO **YORK 10™** 9525 Vassar Ave. #80  
Chatsworth, CA 91311

ORDER FORM

| ITEM                                                                                     | 1 DOZEN                           | 2 DOZEN                             | TOTAL |
|------------------------------------------------------------------------------------------|-----------------------------------|-------------------------------------|-------|
| C-05                                                                                     | <input type="checkbox"/> 7.00     | <input type="checkbox"/> 13.00      |       |
| C-06                                                                                     | <input type="checkbox"/> 7.00     | <input type="checkbox"/> 13.00      |       |
| C-10                                                                                     | <input type="checkbox"/> 7.50     | <input type="checkbox"/> 14.00      |       |
| C-12                                                                                     | <input type="checkbox"/> 7.50     | <input type="checkbox"/> 14.00      |       |
| C-20                                                                                     | <input type="checkbox"/> 9.00     | <input type="checkbox"/> 17.00      |       |
| C-24                                                                                     | <input type="checkbox"/> 9.00     | <input type="checkbox"/> 17.00      |       |
| C-32                                                                                     | <input type="checkbox"/> 11.00    | <input type="checkbox"/> 21.00      |       |
| Hard Box                                                                                 | <input type="checkbox"/> 2.50     | <input type="checkbox"/> 4.00       |       |
| White Labels                                                                             | <input type="checkbox"/> 3.00/100 | <input type="checkbox"/> 20.00/1000 |       |
| Colored Labels<br>Color                                                                  | <input type="checkbox"/> 4.00/100 | <input type="checkbox"/> 30.00/1000 |       |
| DESCRIPTION                                                                              | PRICE                             | QUANTITY                            |       |
| Storage Caddy                                                                            | 2.95                              |                                     |       |
| SUB TOTAL                                                                                |                                   |                                     |       |
| Calif. residents add sales tax                                                           |                                   |                                     |       |
| Shipping/handling (any quantity — using prices above)                                    |                                   |                                     | 3.50  |
| Outside 48 Continental States — Additional \$1 per<br>caddy; per doz. cassettes or boxes |                                   |                                     |       |
| TOTAL                                                                                    |                                   |                                     |       |

Each cassette includes two YORK 10 labels only. Boxes are sold separately. We prefer to ship by UPS as being the fastest and safest. If you need shipment by Parcel Post, check here

NOTE Additional charges outside 48 Continental States. Shipments to AK, HI, and USA possessions go by Priority Mail. Canada & Mexico—Airmail. All others—Sea Mail

**Ask about our  
DUPLICATING  
SERVICE**

CHECK OR M.O. Charge to  
ENCLOSED  Credit Card: VISA  MASTERCARD   
 PLEASE SEND QUANTITY DISCOUNTS

Card No. \_\_\_\_\_ Exp. \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State/Zip \_\_\_\_\_

Signature \_\_\_\_\_ Phone \_\_\_\_\_

Computer make & model \_\_\_\_\_ Disk? (y/n) \_\_\_\_\_

Program Listing 2. BKUPDO.

```

10 CLEAR2E3 :DIMF$(19) :MAXFILES=2
15 INPUT"(R)ead, (S)ave, or (V)erify";B$
:IF B$="R" THEN 700 ELSE IF B$="S" THEN
500 ELSE IF B$="V" THEN 300 ELSE 15
20 OPEN"RAM:"+F$(J%)+".DO" FOR INPUT AS
1 :RETURN
30 OPEN"CAS:"+F$(J%) FOR INPUT AS 2
:RETURN
60 OPEN"CAS:FINM" FOR INPUT AS 2 :X%=0
:PRINT"Reading File Names from cassette"

70 IF EOF(2) THEN 90 ELSE X%=X%+1
:LINEINPUT#2,F$(X%) :IF F$(X%)="" THEN
X%=X%-1
75 PRINT F$(X%);"- ";
80 GOTO70
90 PRINT:CLOSE:RETURN
100 I=I+1:A=PEEK(I)+256*PEEK(I+1)
120 N$=""
130 FOR I=I+2 TO I+5
140 J%=PEEK(I) :IF J%>32 THEN N$=N$+CHR$(
J%)
150 NEXT
160 EX$=CHR$(PEEK(I))+CHR$(PEEK(I+1))
170 IFEX$<>"DO"THENPRINT"Bad
extension":RETURN
180 PRINT"File: ";N$;".DO at
address: ";A:PRINT"store on cassette (y/n
)? ";
190 B$=INKEY$:IFB$=""THEN190
200 IFB$="Y"ORB$="y"THENX%=X%+1:F$(X%)
=N$:PRINT"save ";N$ELSEPRINT"skip ";N$
210 RETURN
300 PRINT"Failure to find file will
cause": PRINT"cassette to run until tape
out.":PRINT"Looking for FINM":ON ERROR.
GOTO 450
310 GOSUB60
320 FOR J%=1 TO X%
330 PRINTUSING"Attempting to verify: \
\ ";F$(J%);
340 GOSUB30:PRINT" Found ";
350 GOSUB20
360 IFEOF(1)ANDEOF(2)THEN PRINT" ok"
:GOTO410
370 IFEOF(1)OREOF(2)THENPRINT" length
mismatched":GOTO410
380 A$=INPUT$(1,1):B$=INPUT$(1,2)
390 IFAS<>B$ THEN PRINT"char mismatch"
400 GOTO360
410 CLOSE:NEXT
420 PRINT:PRINT"Verify Completed":END
450 MOTOR OFF:PRINT:PRINT"Machine error
at file: ";F$(J%)
460 RESUME 410
500 I=63930:X%=0
510 IF PEEK(I)=192THENGOSUB100:GOTO510
520 I=I+1:IF I<64140 THEN 510
530 IF X%<1 THEN END ELSE INPUT"prepare
cassette and hit <ENTER>";B$
540
OPEN"CAS:FINM"FOROUTPUTAS2:PRINT"Saving
File NaMes":FORJ%=1TOX%:PRINT#2,F$(J%)
:NEXT:CLOSE
550 FOR J%=1TOX%
560 PRINT"Saving: ";F$(J%)
570 GOSUB20
    
```

Listing 2 continued

Listing 2 continued

```

580 OPEN"CAS:"+F$(J%)FOROUTPUTAS2
590 IFEOF(1)THEN610ELSEB$=INPUT$(1,1)
600 PRINT#2,B$;:GOTO590
610 CLOSE:NEXT
620 CSAVE"BKUPDO":END
700 GOSUB 60
710 FOR J%=1 TO X%
720 PRINT"Reading: ";F$(J%)
730 OPEN"RAM:"+F$(J%)+".DO"FOROUTPUTAS1
740 GOSUB 30
750 IFEOF(2)THEN770ELSEB$=INPUT$(1,2)
760 PRINT#1,B$;:GOTO750
770 CLOSE:NEXT
780 CLOSE:END

```

amines all file names, it writes file FINM to cassette, followed by each file in the back-up list. Finally, a CSAVE"BKUPDO" command saves a copy of the program as a follower to the data files. This command stops program execution.

Because the 100 requires a .DO extension for files opened in Basic, it cannot save Basic and command program files by reading them from memory. This isn't usually a nuisance as working programs are not often updated. You can make backups by the usual loading and saving routine with little inconvenience. ■

Contact Bryan R. Leipper at 714 Terra Court, Reno, NV 89506.

## Write Now

by Ronald F. Balonis

While not a full-blown word processor with fancy editing features, Writer.BA is a print processor that complements the Model 100's text editor (see Program Listing 3).

Writer.BA provides the most basic formatting features (see Table 1): page length (PL), left margin (LM), right margin (RM), top margin (TM), bottom margin (BM), line spacing (LS), and page numbering (PG). The program left-justifies the printed text and determines line length using the last space in the text before the right margin value. The only in-text editing command is the down-arrow key used to start a new page.

### Program Operation

I chose variable names that relate closely to the variables' functions. Program lines 0-30 initialize the variables and the screen. In lines 35-75, the program reads the directory with the text files printed on the screen. Lines 80-90 prompt for the name of the file you want printed. Then lines 100-180 open the file, read the format line, and set the print format parameters. Lines 190-450 read and print the text file to your format specifications. Lines 500-600 error-trap for file or format errors; all other errors cause an abort to the menu.

### Using the Program

To save memory and attain an acceptable printing speed (about 30 characters per second), I limited the program's functions to those I usually need. You must start a Writer text file with a format line: begin with a greater-than sign and end with an end-of-text marker (by pressing the enter key).

# BASIC

## FOR MODELS I AND III

# JUST GOT BETTER

### INTRODUCING BASRUM. THE ONLY LINKING LOADER & LIBRARY MANAGER FOR BASIC.

BASRUM is designed to manage the subroutines you write and use in your Basic programs. Like an efficient filing system, BASRUM saves you time, energy, and space. Best of all, it makes your program writing FASTER and EASIER.

#### HOW DOES IT WORK?

BASRUM lets you store your subroutines in "library" files, separate from the programs that use them. ANY Basic program can then use any routine in any file. You can even load up to 256 routines to a special, protected memory area and use them AFTER you load other Basic programs! Naturally, you can replace or delete any routine from memory at any time.

#### SUPER BENEFITS!

NOW you can write a subroutine ONCE, store it on disk ONCE ... but use it in ANY program at ANY time, AUTOMATICALLY! Load them once, use them all day. You can CHAIN from program to program, keeping the existing subroutines, variables, and file buffers. Your Basic programs, being shorter, will LOAD FASTER. NEED MORE MEMORY? You can break long programs into shorter modules and "overlay" the modules. BASRUM gives you complete control of WHAT is loaded, WHEN it is loaded, and WHERE it is loaded.

#### THE NITTY GRITTY.

BASRUM was designed specifically for the models I and III for use with any DOS that is compatible with TRSDOS®. It's overlay structure requires only the top 1600 bytes of memory. There is NO LIMIT on the number of working copies.

**SIMPLE ENOUGH FOR THE BEGINNER, VERSATILE ENOUGH FOR THE PRO: IF YOU WRITE PROGRAMS IN BASIC YOU NEED BASRUM.**

**TO ORDER:** Money order, check, VISA, MASTERCARD: \$149.50  
Specify memory size: 16K 32K 48K

Select disk format:

Model I: 35 track, single density

Model III: 40 track, double density

Guaranteed to perform as described or your money back. Orders are shipped via air, most within 24 hours. A complete manual is included. Information only? Send self-addressed, stamped envelope.



by **Wiley, Inc.**

841 Bishop St., Ste. B-2 Honolulu, HI 96813  
(808) 531-4314

TRSDOS is a registered trademark of the Tandy Corp.  
BASRUM is a trademark of Wiley, Inc.



| Format         | Default    | Low | High   |
|----------------|------------|-----|--------|
| Page Length    | PL-66      | 1   | 90     |
| Left Margin    | LM-12      | 0   | 131    |
| Right Margin   | RM-72      | 1   | 132    |
| Top Margin     | TM-6       | 1   | 89     |
| Bottom Margin  | BM-60      | 2   | 90     |
| Line Spacing   | LS-1       | 1   | 90     |
| Page Numbering | PG-0 (off) | 0   | (on) 1 |

Table 1. Text format values.

The program tests the format values to ensure that the given range of values exist; the rest is left up to you. The format line need only contain those format values that differ from the default values listed in Table 1.

A typical format line might look like this:

>PL=66 LM=12 RM=72 TM=6 BM=60 LS=1 PG=0

The program consists of two screens: The first, or initialization, screen lists your text files and prompts you to type in the name of a file you want printed or press the enter key to exit the program. The second screen notifies you of format or file errors or that the program is printing a file.

After it prints a file, the program redisplay the first screen so you can either select and print another text file or exit the program.

Initially, it might seem difficult to visualize how the text on the Model 100 screen would appear once printed, but with experience you'll become proficient at it. ■

You can reach Ronald F. Balonis at 118 Rice St., Trucksville, PA 18708.

Program Listing 3. Writer.BA.

```

0 ' WRITER.BA MEMO SCRIBE
2 ' A TEXT PRINT PROCESSOR
5 'BY RON BALONIS
10 CLS: CLEAR500:DEFINT A-Z:N=0:NX=0
12 ON ERROR GOTO 500
15 NPAGES$="":CHAR$="":LINES$=""
20 TITLES$="----- Memo Scribe -----"
25 LF$=CHR$(10):CR$=CHR$(13):SPAC$=" "
30 CLS:PRINT@10,TITLES$:PRINT:FNMES$=""
32 '
35 '---READ MENU FOR .DO FILES ONLY---
40 FOR II=63930 TO 64139 STEP 11
45 B$="":IF PEEK(II)=0 THEN 75
50 FOR II=3 TO 10
55 B$=B$+CHR$(PEEK(II+II))
60 NEXT II
65 IF INSTR(B$,"DO")<1 THEN 75
70 PRINTLEFT$(B$,6) " ";
75 NEXT II:PRINT
80 PRINT@240,"<FILE NAME> OR <> TO
EXIT";:INPUT FNME$
85 IF FNME$="" THEN BEEP: MENU
90 FNME$=LEFT$(FNME$,6)
95 '
98 '---GET FILE AND PRINTING FORMAT---
100 OPEN FNME$ FOR INPUT AS 1
    
```

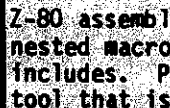
Listing 3 continued

# PRO-fessional Software

## for LDOS/TRSDOS 6.0



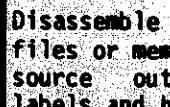
A maintenance tool for "CMD" files. Allows you to append 2 or more files, reorganize, and offset. Extract LIB members. \$40



Z-80 assembler/editor supporting nested macros, conditionals, and includes. PRO-CREATES a powerful tool that is easy to use. \$100



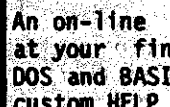
Transfer files directly to DOS 6.0 from selected CP/M media. PRO-CURE supports Omikron, IBM, Kaypro, and Osborn formats. \$50



Disassemble directly from disk files or memory. The disk file source output generates 100% labels and handles data. \$40



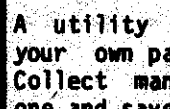
A 4-function utility package that is loaded with power: DOCONFIG; MEMDIR; PARMDIR; and SWAP. A must for JCL users. \$40



An on-line quick reference card at your fingertips. Screens for DOS and BASIC. Create your own custom HELP files, too. \$25



This is the LC C-language compiler now compatible with DOS 6.0. LC includes the PRO-CREATE macro-assembler package. \$150



A utility to build and maintain your own partitioned data sets. Collect many small files into one and save disk space. \$40



A block-graphics screen editor which is used to create graphic images for BASIC, assembler programs, or printing. \$50

U.S. Shipping: PRO-LC, \$5; PRO-CREATE, \$4; All others \$2. COD add \$1.50. VISA/MC/CHOICE.

MISOSYS  
P.O. Box 4848 - Dept M  
Alexandria, Virginia 22303-0848  
703-960-2998

107

LDOS is a trademark of Logical Systems, Inc.  
TRSDOS is a trademark of Tandy Corp.



Listing 3 continued

```

105 INPUT#1,F0RMT$:PAGE=1
110 CLS:PRINT@10,TITLE$:RESTORE:ERX=0
115 IF LEFT$(F0RMT$,1)<>">" THEN 600
120 FOR I=1 TO 7
125 READ T$,L,H,DFLT
130 II=INSTR(F0RMT$,T$)
135 IF II=0 THEN 150
140 DFLT=VAL(MID$(F0RMT$,II+3,2))
145 IF DFLT<L OR DFLT>H THEN ERX=-1
150 X0(I)=DFLT
155 NEXT I
160 PL=X0(1):LM=X0(2):RM=X0(3)
165 TM=X0(4):BM=X0(5):LS=X0(6)
170 PG=X0(7):LCNT=0:LGTH=RM-LM:PN=0
175 IF LM>RM OR TM>BM OR BM>PL THEN
ERX=-1
180 IF ERX=-1 THEN 600
185 '
190 PRINT@120," -- PRINTING TEXT
FILE < "FNME$" > --"
200 LINE$="":N=0:LCNT=0:PN=PN+1
210 IF TM=1 THEN 300
220 FOR I=1 TO TM
230 FOR II=1 TO LS
240 LPRINT"":LCNT=LCNT+1
250 NEXT II
260 NEXT I
290 '
300 CHAR$=INPUT$(1,1):IF EOF(1) THEN
350
305 IF CHAR$=NPAGES THEN 410
310 IF CHAR$=CR$ THEN 350
315 IF CHAR$=LF$ THEN 300
320 N=N+1:LINE$=LINE$+CHAR$
325 IF CHAR$=SPAC$ THEN NX=N
330 '
335 IF N<LGTH THEN 300 ELSE 360
340 '
350 LPRINTTAB(LM)LINE$;
355 LINE$="":N=0:NX=0:GOTO370
360 LPRINTTAB(LM)LEFT$(LINE$,NX);
365 LINE$=RIGHT$(LINE$,N-NX):N=N-NX
370 FOR I=1 TO LS
375 LPRINT"":LCNT=LCNT+1
380 NEXT I
390 '
400 IF LCNT<BM AND NOT EOF(1) THEN 300
410 FOR I=LCNT TO PL-1
415 IF I<>PL-3 THEN 425
420 IF PG=1 THEN LPRINTTAB(((RM-LM)
/2)-7+LM)USING"Page ##";PN;
425 LPRINT""
430 NEXT I
440 IF EOF(1) THEN CLOSE 1 ELSE 200
450 GOTO 30
490 '
500 CLOSE 1
510 IF ERL<>100 OR ERL<>105 OR ERL<>300
THEN MENU
520 PRINT@250,"---- FILE ERROR ----"
530 FOR I=1 TO 500:NEXT I:RESUME 30
540 '
600 CLOSE 1
610 PRINT@250,"--- FORMAT ERROR ---"
650 FOR I=1 TO 500:NEXT I:GOTO 30
800 END
890 'FORMAT,LOW,HIGH,DEFAULT--VALUES
900 DATA PL,1,90,66,LM,0,131,12
905 DATA RM,1,132,72,TM,1,89,6
910 DATA BM,2,90,60,LS,1,90,1
915 DATA PG,0,1,0

```

## Pin Pals

by Ronald F. Balonis

You can transfer Basic and Text files between two Model 100s with a homemade cable. I'll describe the construction of such a cable and asynchronous file transfers using the Model 100 RS-232 ports for file input/output (I/O). See Table 2 for a list of parts necessary for this project.

### Constructing the Cable

Since the Model 100's RS-232 port is flush with its case, you first have to trim the plastic lips that cover the pin flange on the cable covers (see Photo 4). Lay the covers on a flat surface

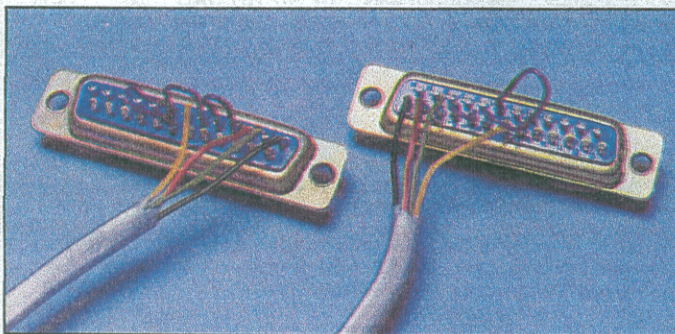


Photo 4. Pin connections for null modem RS-232 connectors.

and cut through the molded groove with a hacksaw blade.

Next, cut six 1-inch jumper wires from some hook-up wire; trim 1/16 inch of insulation from the ends of each and tin the exposed ends with solder.

Then, on both ends of the cable, cut the outside insulation back 1 inch and trim 1/16 inch of insulation from each cable wire and tin the ends with solder.

On each 25-pin connector, solder jumper wires as follows (see the Figure): one from pin 4 to 5, one from pin 6 to 8, and one from pin 8 to 20.

Push the cable ends through the covers.

To one 25-pin connector and cable end, solder the wires as follows: black to pin 1, green to pin 2, red to pin 3, and yellow to pin 7.

To the other connector and cable end, solder the wires as follows: black to pin 1, red to pin 2, green to pin 3, and yellow to pin 7.

| Quantity | Part Description            | Manufacturer | Part Number |
|----------|-----------------------------|--------------|-------------|
| 5 feet   | conductor cable             | Radio Shack  | 278-365     |
| 2        | 25-pin D submini connectors | Radio Shack  | 276-1547    |
| 2        | 25-pin D submini hoods      | Radio Shack  | 276-1549    |

Table 2. Parts list.



Fasten the connectors to the covers with the screws provided and tighten the strain relief screws. Your null modem cable is ready for use.

**File Transfer Procedure**

On the Model 100, communications files require an RS-232 transmission configuration (refer to p. 125 of the reference manual for more detailed information).

The configuration I use, 88E1E, is 9,600 baud, 8-bit word length, even parity, 1 stop bit, and XON/XOFF enabled.

**Transferring Basic Programs**

To transfer a Basic program file, connect the null modem cable to both machines. Load Basic on the destination machine, type LOAD "COM:88E1E" and press the enter key.

Then load Basic on the source machine, load the file to transfer, then type SAVE"COM:88E1E" and press the enter key. When the cursors on both machines reappear, indicating the transfer is complete, type SAVE"PROGRAM NAME" on the destination machine, and press the enter key.

**Transferring Text Files**

To transfer a text file first interconnect the machines with the null modem cable. Load the 100's built-in Text file on the destination machine, create a text file with the same file name, and press the F2 key (Load). Then type COM:88E1E and press the enter key.

Load Text on the source machine, load the text file to transfer, press the F3 key (Save); then type COM:88E1E and press the enter key. When the cursors reappear, the transfer is complete.

Both cable and procedure have worked well for me. With the configuration of 88E1E, small programs of 1 to 2 kilobytes require just a few seconds. ■

*Contact Ronald F. Balonis at 118 Rice St., Trucksville, PA 18708.*

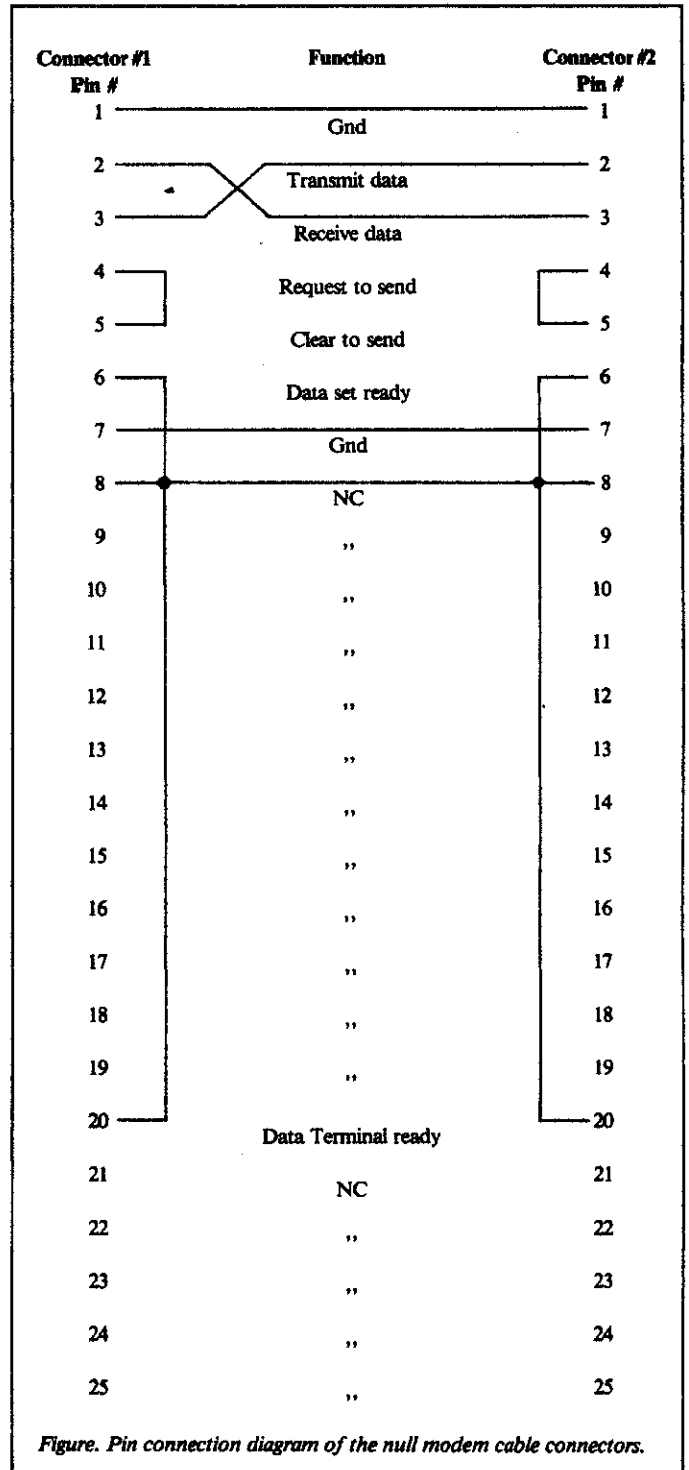


Figure. Pin connection diagram of the null modem cable connectors.

**"Worksheet 100" — A Spreadsheet Program for the TRS-80™ Model 100 Portable Computer**

The "Worksheet 100" works with spreadsheets of up to 16 columns by 40 rows with 24K of RAM or up to 16 columns by 60 rows with 32K. Eight main functions using the programmable function keys plus three entry modes provide large spreadsheet convenience. The "Goto" function, doubling as a "search function", will speedily move the marker to any cell given coordinates, column and row labels or cell contents. Full arithmetic operators plus summation, averaging, maximum and minimum functions are provided.

Worksheets can be saved or loaded using RAM or cassette. A very useful group of worksheet templates (with formulas) is provided ready to load. These are "Expense Report", "Sales Report", "Service Report", "Weekly Schedule", "Personal Tax Worksheet", and "Personal Finances I and II".

The "Worksheet 100" and the seven ready to use templates are supplied on a single cassette and require 24K RAM.

**The "Worksheet 100" and seven useful templates—for only \$89.95**

Mail this Order Slip to **Diaz Enterprises  
P.O. Box 4609  
Mountain View, CA 94040**

Please send me the "Worksheet 100" and 7 useful templates for \$89.95

Payment Enclosed \_\_\_\_\_ MasterCard \_\_\_\_\_ Visa \_\_\_\_\_ Amex \_\_\_\_\_

Card # \_\_\_\_\_ Signature \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Postage and handling included within the USA and Canada. California residents should add 6.5% sales tax.

\*TRS-80 is a trademark of Tandy Corporation. ✓183

# RAM FILES

## Peripheral Problems

While on my way through a Model 100-to-peripheral routine for the serial port, I tried without success to use STR\$ for the conversion of values to string form as the instruction manual description suggests.

I traced the peripheral's resulting fits and starts to the computer's unexpected insertion of a space character just before the numerals in each output string.

If your unit has the same characteristics as mine (serial number 301005984), you may see the problem by typing and running the following routine, based on the manual's example:

```
10 BAL = 133
20 BS = "$" + STR$(BAL) + ".00"
30 PRINT BS
```

Note that the unwanted space occurs between the dollar sign and the numeral 1 on the screen display.

One way to handle this is to substitute a synthetic command for STR\$, automatically removing the leading space:

```
RIGHT$(STR$(n.e.),LEN(STR$(n.e.))-1)
```

where n.e. is the numeric expression you want to convert.

This command is long enough to qualify for subroutine

status when used frequently in a program. You can see its effectiveness by running a revised version of the above program that includes the solution as part of two additional lines:

```
40 BS = "$" + RIGHT$(STR$(BAL),LEN(STR$(BAL))-1) + ".00"
50 PRINT BS
```

The screen printout provides results both before and after running this program for quick performance confirmation.

John M. Hicks  
117 Presidio Court  
Verona, PA 15417

## Mighty Write Debug



In the PRNXT.BA program in "Mighty Write" (C\*Notes, October 1983, p. 266) the N option to print the next page of text only works the first time you use it. This creates a problem when text is longer than two pages.

To correct the problem, replace lines 200, 230, and 340 with:

```
200 LC = LC + 1:IF LC = 57 THEN 320
230 IF LC < 57 THEN 140 ELSE GOTO 320
340 FORQ = 1 TO 50:NEXT Q:CLS:LC = 1
```

—Eds.

**We Will BEAT Any Advertised Price or You Receive a FREE Gift!!**

|                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>REG. PRICE 5 1/4"</b><br/><b>28.95</b> 86/00<br/>SOFT SECTOR</p> <p><b>5 1/4"</b><br/><b>39.50</b> 06/00<br/>SOFT SECTOR</p>                                                                                                                                                                                                                                                             | <p><b>1 Buy 2 to 4 Boxes</b><br/>Get a Library Case With Ea. Box <i>Free</i></p> <p>OR</p> <p><b>10% OFF</b> REG. PRICE <i>\$4.50 VALUE</i></p>                                     |
| <p><b>SPECIAL</b></p> <p>5.95/00 VERBATIM SOFT SECT 23.95</p> <p>5.95/00 MEMOREX SOFT SECT 23.95</p> <p>5.95/00 3M - SCOTCH SOFT SECT 22.95</p>                                                                                                                                                                                                                                                | <p><b>2 maxell Buy 5 to 9 Boxes</b><br/>Get a Plastic Storage Flip n File Box <i>Free</i> (Holds 75 Mini-Disks)</p> <p>OR</p> <p><b>15% OFF</b> REG. PRICE <i>\$35.00 VALUE</i></p> |
| <p>Add \$2.00 Handling in USA</p> <p><b>VISA</b> <b>C.O.D.</b> <b>MasterCard</b></p> <p>5% Tax in Fla. Only<br/>APO/Box No's &amp; Canada<br/>Add 3.50 Shipping</p> <p><b>MEGA o BYTE</b></p> <p>4699 S.W. 45th Street<br/>Ft. Lauderdale, Florida 33314<br/>305-587-1130</p> <p><b>1-800-327-1013</b></p> <p><small>Allow one week delivery. This offer may be retracted anytime.</small></p> | <p><b>3 maxell Buy 10 Boxes</b><br/>Get a Complete Computer Maintenance Kit <i>Free</i></p> <p>OR</p> <p><b>25% OFF</b> REG. PRICE <i>\$45.00 VALUE</i></p> <p>199</p>              |

ONLY

# \$199

**HIGH SPEED  
CENTRONICS 101A  
DOT MATRIX LINEPRINTERS**

Fully Reconditioned  
Heavy Duty Steel Construction  
1980 List Price \$4800 Our Price \$199

- 165 Characters per second (1920 words per minute)
- 2" to 15" Adjustable Tractor Feed
- 132 Columns Expanded Print & Multi-strike Highlighting
- 9 x 7 Matrix with "End of Line" Seeking Carriage Return
- Optional Lower Case Character Sets with Graphics Available
- Compatible with Centronics Parallel Outputs: (TRS-80, Apple, IBM-PC, Franklin Ace, Osborne, Atari, TI-99, Commodore 64, Vic-20, and More)

**BRAND NAME COMPUTERS AND PRINTERS**  
New in stock at competitive prices.  
Send for price list.

**A.C.E. SYSTEMS**  
106 E. BROAD ST. BETHLEHEM PA 18018  
215-867-5066



✓ 38



C-NOTE FOR MARCH 1984

## Fast Backwards

by Ray Cadmus

CALC (Program Listing 1) provides an easy-to-use four-function calculator that incorporates a form of reverse Polish notation like that found in Hewlett-Packard calculators.

Reverse Polish notation is a way of writing mathematical equations that takes advantage of the way computers store and manipulate numbers to streamline the calculating process. Unlike algebraic equations, where mathematical operators appear in the conventional computational sequence, equations that use reverse Polish notation have the numbers before the operator. For instance, adding five and seven algebraically appears as  $5 + 7$ ; in reverse Polish notation, this same operation is  $5 7 +$ .

### The Program

In addition to the four basic math functions, CALC provides two special-function command keys. Function key 6 (labeled Dec on the display) lets you specify the number of decimal positions displayed in the result. Function key 5 (labeled \$ on the display) automatically formats the display to read out values in units of dollars and cents. When you press the \$ key, you toggle this function on and off.

To perform a mathematical calculation, key in a number

(two numbers for an initial calculation) followed by the desired function. As you enter the number, it appears on the left side of the screen. When you press the operation key, the entry clears and the result appears on the right side of the display. See Table 1 for additional active keys. The Figure depicts a series of possible entries and their results.

Note that the \$ function is engaged in the course of the calculation. Be careful in this mode. The program divides all entries by 100, so 1.00 followed by  $20 \times$  results in .20—not the 20.00 you may expect.

The enter key acts like the + key and is an easier target to hit. Pressing any other key clears only the current entry; the total remains undisturbed.

You can expand the program to include your favorite functions: assign a key, test for it in the 50-series of program lines, and set up a routine to handle it as the 4000-series routines do.

The 3000-series of instructions define and activate function keys F1-F8. The screen statement in the third line in the program turns on the label display for the function keys. The Menu routine in line 4080 runs BSET (Program Listing 2), restoring the standard basic functions.

The 5000-series lines convert the current entry to a usable form and divide the value by 100 if you've turned on the \$ function.

The 6000-series lines display the total, after first clearing the screen. The decimal function in lines 4060 and 4062 modifies

Program Listing 1. CALC.

```

2 CLS
3 SCREEN0,1
5 GOSUB3000
6 MBASE$="###,###,###"
7 M$=MBASE$
8 LINE(10,6)-(73,17),1,B
9 LINE(126,6)-(239,17),1,B
10 PRINT@125,"Entry";
11 PRINT@145,"Result";
20 PRINT@42,STRING$(9," ");
25 V$=""
30 C$=INKEY$:IFC$=""THEN30
40 IFC$=>"0"ANDC$<="9"ORC$="."THENV$=V$+C$
C$:PRINT@42,V$;:GOTO30
50 IFC$="+ORC$="-ORC$=CHR$(13)
THENGOSUB4010
51 IF C$="-"THEN GOSUB 4020
52 IFC$="*"THEN GOSUB 4030
53 IF C$="/"THEN GOSUB 4040
60 PRINT@62,USINGM$;T;
70 GOTO 20
3000 'SET KEYS
3001 KEY1,"+"
3002 KEY2,"-"
3003 KEY3,"X"
3004 KEY4,"/"
3005 KEY5,"$off":D%=0
3006 KEY6,"Dec"
3007 KEY7,"Clr"
3008 KEY8,"Menu"
3010 KEY ON
3020 ON KEY GOSUB
4010,4020,4030,4040,4050,4060,4070,4080
3030 RETURN
4010 GOSUB 5000:T=T+V:GOSUB 6000:RETURN
4020 GOSUB 5000:T=T-V:GOSUB 6000:RETURN
4030 GOSUB 5000:T=T*V:GOSUB 6000:RETURN
4040 GOSUB 5000:IF V<>0 THEN T=T/V:
GOSUB 6000:RETURN ELSE BEEP:RETURN
4050 IF D%THEN D%=0:KEY5,"$off"ELSE
V$="2": GOSUB 4060:KEY5,"$on":D%=1
4055 RETURN
4060 GOSUB 5000:IF V>6 THEN V=6
4062 M$=MBASE$+"."+STRING$(V,"#"):
GOSUB6000:RETURN
4070 T=0:PRINT@62,USINGM$;T;:RETURN
4080 RUN"bset"
5000 V=VAL(V$):V$="":PRINT@42,STRING$(
9," ")
5010 IF D%THENV=V/100
5020 RETURN
6000 PRINT@62,STRING$(20," ")
6010 PRINT@62,USINGM$;T;:RETURN

```

End

M\$, the print-using mask, as necessary. I've restricted the number of decimal places to no more than six in order to prevent line wrap. ■

Contact Ray Cadmus at 600 W. Lee, Moberly, MO 65270.

## Thanks for the Memory

by Bradford N. Dixon  
80 Micro Technical Editor

By now Model 100 owners are used to the machine's limited memory for storing text and programs. You can usually overcome this limitation, but a better solution is to find an inexpensive way to add memory to your machine.

Holmes Engineering (5175 Green Pine Drive, Salt Lake City, UT 84107; 801-261-5652) has the answer: their \$75 IM-100 8K RAM module is just what the doctor ordered for Model 100 users tired of out-of-memory errors (see Photo 1).

Each 8K memory module comprises a piggy-back unit of two printed circuit boards, one mounted atop the other, and plugs directly into the memory upgrade sockets inside the 100. Installing an IM-100 is so easy you can do it in the time it takes to read this article.

### Getting It Together

First, collect the materials required for the installation: a Phillips-head screwdriver, a flat surface twice the size of the Model 100, and a soft towel to serve as a pad to protect the computer when you turn it over.

Then save all resident programs to tape because you'll switch off the RAM memory before removing the bottom of the computer.

Spread the soft cloth over the flat surface and turn the 100 over onto it. Turn off the RAM memory power switch and

remove the Model 100's four corner screws. Carefully turn the computer face up, letting the screws fall onto your work surface. Put the screws aside; you'll need them when the installation is complete.

Turn the computer face down again. If you try to separate the two halves of the computer at this point, you'll notice they're still locked together. You must first disengage locking tabs on each side and the top of the unit. When you separate the two halves of the computer, you'll see the RAM module sockets on the lower left corner of the CPU board (see Photo 2). If you have an 8K Model 100 you'll notice three empty sockets labeled M8, M7, and M6 from right to left. In a 24K unit, the M8 and M7 sockets are filled with 8K chips.

### Installing the Modules

You install the IM-100 modules from right to left, so put your first IM-100 over the M8 socket. Make sure that the small notch at the end of the chip faces you. The IM-100 notch faces

### The Key Box

The programs in "Fast Backwards," "Wipe Out," and "A New Pair of Shoes" run in 8K RAM. The program in "Account for Yourself" requires 24K RAM; a printer is optional.

## CONVERT MOD III BASIC TO MOD 4

**Good News!** The Model 4 is twice as fast as the Model III.  
**Bad News!** Model III disk BASIC programs only run at half speed on the Model 4.  
**Best News!** "CONVSTO4" will automate many of the changes required to convert Model III BASIC programs to Model 4 BASIC. Your Model III BASIC programs will run twice as fast as they do now, and your manual conversion efforts will be vastly reduced.

In minutes, this powerful utility can insert all required spaces, recalculate PRINT# addresses, adjust TAB() addresses, insert correct exponentiation symbols, and tap special function key numbers.

Other options enable your programs to run even faster: remove REM's, comments, line numbers, and unnecessary spaces.

Or, format your programs and make them easier to read and debug: insert down arrows and indent between multiple instructions, IF, THEN, and ELSE statements.

Write the author, Educational Micro Systems, Inc., at the address below. Send the \$29.95 CONVERSION Utility Enclosed (also check whether order for \$49.95 with 320-line wordwrap utility) to the address on page 181.

NAME \_\_\_\_\_  
STREET \_\_\_\_\_  
CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP \_\_\_\_\_

EDUCATIONAL MICRO SYSTEMS, INC.  
P.O. Box 471, Chester, New Jersey 07930  
201-679-0643

| Entry   | Result |
|---------|--------|
| 12+     | 12     |
| 12+     | 24     |
| 2;      | 48     |
| 3/      | 16     |
| 2dec    | 16.00  |
| 0dec    | 16.    |
| chr     | 0.0    |
| \$(off) | 0.00   |
| 123+    | 1.23   |

| Key             | Function                            |
|-----------------|-------------------------------------|
| + =<br>or enter | Adds the current entry to the total |
| -               | Subtracts the current entry         |
| /               | Divides by the current entry        |
| *               | Multiplies by the current entry     |

Table 1 (above). CALC's active keys.

Figure (left). A series of calculations done in reverse Polish notation.

```
10 KEY1, "Files" +CHR$(13)
20 KEY2, "Load" +CHR$(34)
30 KEY3, "Save" +CHR$(34)
40 KEY4, "Run" +CHR$(13)
50 KEY5, "List" +CHR$(13)
60 KEY6, "?tre(0)" +CHR$(13)
70 KEY7, "Edit"
80 KEY8, "Menu" +CHR$(13)
90 MENU
```

Program Listing 2. BSET.



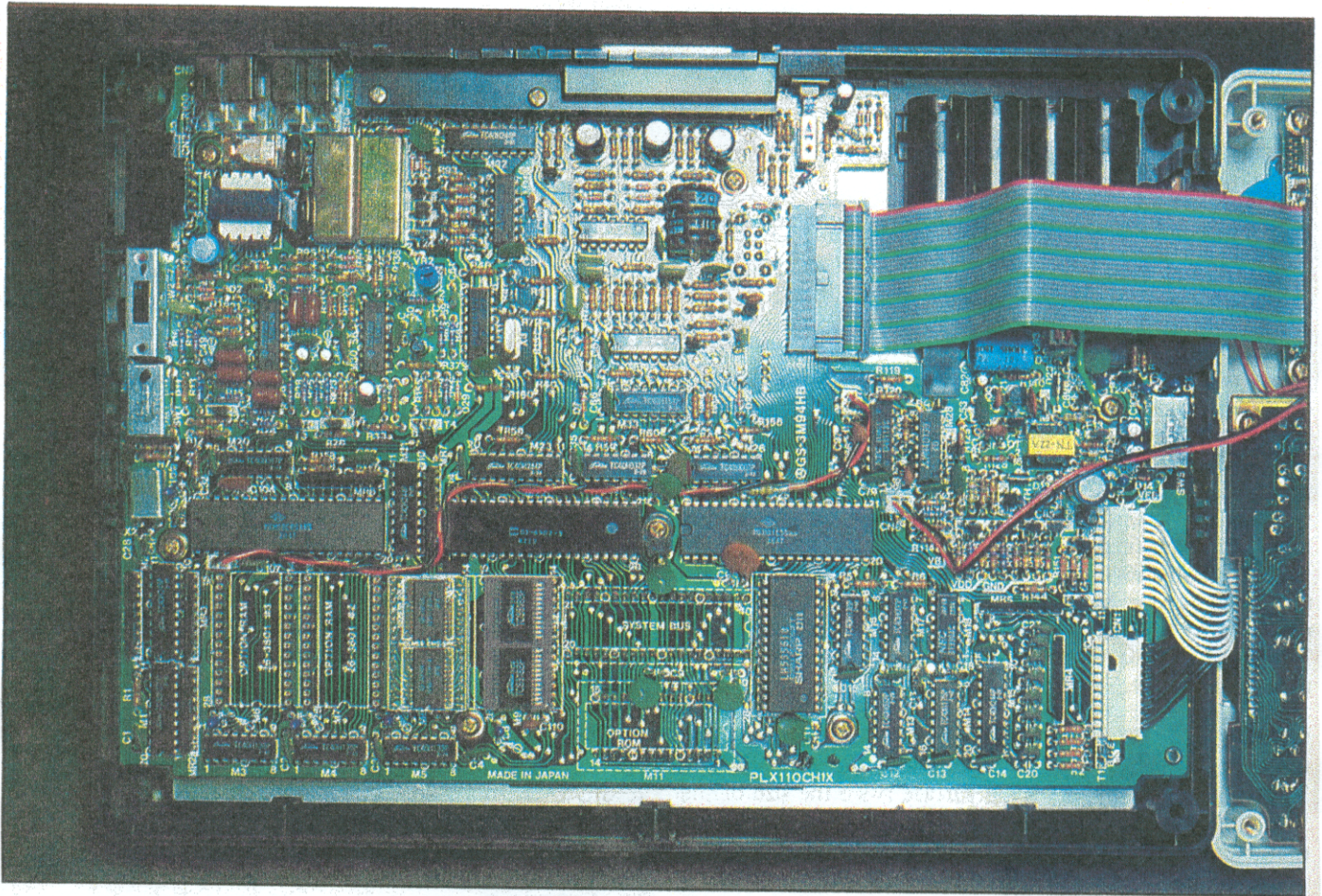


Photo 2. CPU board with RAM expansion sockets in lower left corner.

in the same direction as the notch in the 8K module already installed in the M9 socket (see Photo 3).

Be sure to align the pins on the IM-100 with the socket holes. Gently push the module into the socket until it's firmly in place. If you're installing more than one IM-100, repeat this procedure.

After you've installed all the modules, put the bottom of the Model 100 on top of the inverted keyboard and make sure that the side and top tabs lock.

Before you put the corner screws back in, push the RAM memory switch back on and look at the display. At the bottom of the display on an 8K machine, the total number of free bytes should now read 13,638. An upgrade to 24K reads 21,638, and a 32K total reads 29,638 bytes free. These amounts are lower than the upgrade itself because the operating system uses approximately 2,362 bytes of the RAM memory. If the RAM total is correct, turn the computer over once again and replace the four corner screws.

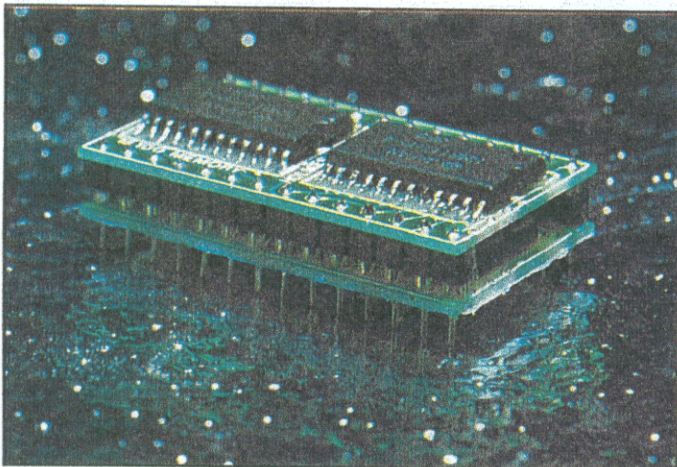


Photo 1. IM-100 8K RAM module.

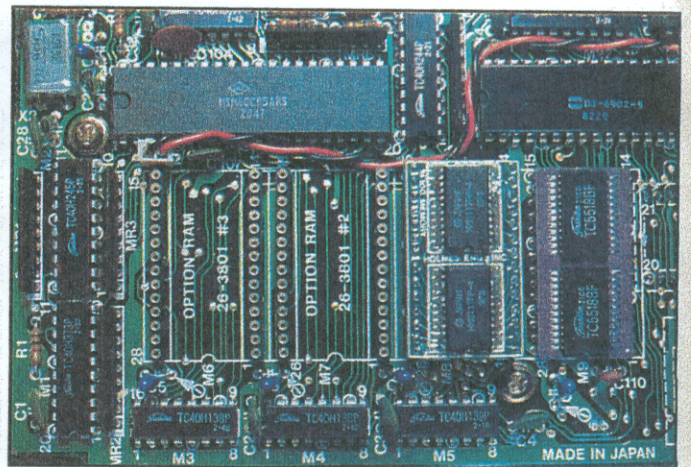


Photo 3. 8K module installed in M8 socket.



**Solving Problems**

I encountered only two problems in upgrading my 8K Model 100. First, the rows of pins on my IM-100 were too close together to slide neatly into the M8 socket. I had to bend one row out slightly to make the module fit.

The second problem appeared when I looked at the display after installing the module: I had only 13,254 bytes free. The new RAM module seemed to work fine otherwise. I downloaded the memory check program available free from Holmes's BBS (801-263-1103) and found that the original RAM and the IM-100 worked perfectly.

According to Walt Bascom, marketing director at Holmes, different models of the 100 allocate RAM differently for the operating system. As long as the upgrade is close to that noted in the instructions, everything is working properly. ■

**Account for Yourself**

by Mark Hickenbottom

Draw up a business expense report on the road or keep family budgets in order with Data Manager (see Program Listing 3). Data Manager maintains a spreadsheet and prints a columnar summary report.

**The Program**

On boot-up, Data Manager first asks if your printer has graphics capability. If so, answer yes. It then asks for the printer width, or the number of columns the printer can print. Enter the width or press the enter key if it is 80 columns wide.

Next, Data Manager displays the current file along with the six main menu options (create a new file, manage a file, add data to a file, print a report, display account balances, or quit the program) and repeats them in abbreviated form at the bottom of the screen near the Model 100 function keys (see Table 2). To select a particular option, press the function key aligned with that option.

**Creating a File**

Select this option to create a new data file. Data Manager first asks you to name the new data file and then specify a number of fields per record (the default is one).

For each field, the program asks a series of questions. When it asks for the field name, enter a name of no more than six characters. It then asks whether the field is numeric. If it is, answer yes. The program first prompts you for the width of the integer, then for the fractional portion of the field.

If the field isn't numeric, Data Manager asks for the field's width. In either case, if this isn't the last field, the program

asks whether you want the next field printed on the same line.

Data Manager then prompts you for the name of the amount field, the numeric field used to calculate account balances as well as the total when printing reports. Enter the name of the amount field you set up earlier, or press the enter key if you didn't include an amount field. The program then asks for the number of accounts (savings, checking, and so on). Enter the desired number of accounts or press the enter key to continue.

For each account, Data Manager asks a series of questions. It first asks for the account name, which must be fewer than 16 characters long, then the names of the check field and the check string. In the first check field case, a record only affects an account if the check field contains the check string. For example, a savings account is affected only when the category field contains Savings. If this is so, enter the names of the check field (Category) and the check string (Savings).

For the second check field case, a record only affects an account if the check field is not empty. For example, a checking account is affected only when there is some data in the check number field. If this is the case, enter the name of the check field and press the enter key to continue to the check string question. In either case, Data Manager then prompts you to enter the current account balance.

**Managing a File**

Select this option to add to, print a report of, or display account balances for a particular data file. When Data Manager prompts you, enter the name of the file you want managed.

| Option | Description              |
|--------|--------------------------|
| new    | Create a new file        |
| man    | Manage a file            |
| add    | Add data to a file       |
| rep    | Print a report           |
| accs   | Display account balances |
| end    | Quit the program         |

Table 2. Data Manager's main menu options.

**"Worksheet 100" — A Spreadsheet Program for the TRS-80™\* Model 100 Portable Computer**

The "Worksheet 100" works with spreadsheets of up to 16 columns by 40 rows with 24K of RAM or up to 16 columns by 60 rows with 32K. Eight main functions using the programmable function keys plus three entry modes provide large spreadsheet convenience. The "Goto" function, doubling as a "search function", will speedily move the marker to any cell given coordinates, column and row labels or cell contents. Full arithmetic operators plus summation, averaging, maximum and minimum functions are provided.

Worksheets can be saved or loaded using RAM or cassette. A very useful group of worksheet templates (with formulas) is provided ready to load. These are "Expense Report", "Sales Report", "Service Report", "Weekly Schedule", "Personal Tax Worksheet", and "Personal Finances I and II".

The "Worksheet 100" and the seven ready-to-use templates are supplied on a single cassette and require 24K RAM.

**The "Worksheet 100" and seven useful templates—for only \$89.95**

Mail this Order Slip to **Diaz Enterprises**  
P.O. Box 4609  
Mountain View, CA 94040

Please send me the "Worksheet 100" and 7 useful templates for \$89.95

Payment Enclosed  MC  Visa  Amex  Exp. Date \_\_\_\_\_

Card # \_\_\_\_\_ Signature \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Postage and handling included within the USA and Canada. California residents should add 6.5% sales tax.

\*TRS-80 is a trademark of Tandy Corporation

**Adding Data to a File**

Choose this option to add records to the end of the file. Data Manager displays brief instructions about the function keys. Once you've read the display, press any key. For each field name Data Manager displays, enter the necessary data. If you make a mistake, press the F1 key and then the enter key before pressing the enter key on the last field. Data Manager displays the field names again, letting you reenter the record. When you're done entering data, press the F2 key, then the enter key after pressing the enter key on the last field.

**Printing a Report**

Choose this option to print a columnar report of the data


file. The program displays a menu that lets you print all or selected records and a total of the amount field. To print only records that have a certain field containing a certain string, select one of the record-selection options. For a total of the amount field printed, select one of the print total options. Data Manager then asks you for the names of the check field and the check string.

The program asks for the header, which it prints at the top of every page. Once you've entered the header, Data Manager prints the report.

**Displaying Account Balances**

When you select this option, Data Manager displays all the account balances.

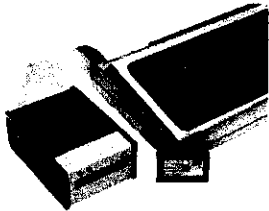
A&J Micro Drive exclusive distributor for Exatron.



Authorized distributor for Holmes Engineering and IJG.

---

### EXATRON STRINGY FLOPPY



**Model I ESF STARTER KIT** ..... \$149.50  
Add \$8.50 Shipping. Includes: Drive 0, Manual, Operating System, Power Supply, Cable, Data I/O Tape, 5 Microwafers

**Model I ESF DRIVE 0** ..... \$124.50  
Add \$8.50 Shipping

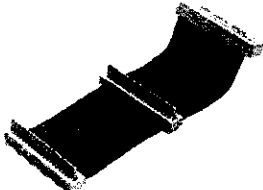
**Model I ESF DRIVE 1 thru 7** ..... \$99.50  
Add \$8.50 Shipping

**Model I OWNER'S MANUAL** ..... \$4.95

1200 Baud Transfer Rate  
No Volume Controls or Rewind  
No Expansion I/F Required  
Automatic Verify with Save  
Uses only 4 Bytes User Memory  
Key Debounce Routine in ESF

**BUS-EXTENDERS**


|         |       |                 |
|---------|-------|-----------------|
| 2 for 1 | ..... | \$25.00         |
| 3 for 1 | ..... | \$30.00         |
| 4 for 1 | ..... | \$35.00         |
| 5 for 1 | ..... | \$40.00         |
| Custom  | ..... | call for prices |



**MICRO WAFER**

|       |       |        |
|-------|-------|--------|
| 5 ft  | ..... | \$3.50 |
| 10 ft | ..... | \$3.50 |
| 20 ft | ..... | \$4.00 |
| 35 ft | ..... | \$4.00 |
| 50 ft | ..... | \$4.50 |

### HOLMES PORTABLE MICRO DRIVE



**MODEL 100 PMD-100** ..... \$339.50  
Add \$8.50 Shipping. Includes: PMD-100 Drive, 3 Wafers, AC Adaptor/Battery Charger, Power Cable, Manual, PMD User Group Membership

**MODEL 100 IM-100 8K RAM Modules** ..... \$69.50

**PMD-100 Owners Manual** ..... \$4.95

High Speed Data Transfer  
Battery Powered with Charger  
No Modification to Model 100  
Saves Text Files and Programs  
Uses Built-in Telecom Mode

**WAFERS**  
Wafers are the same as used in Model I ESF.  
Also compatible with OLIVETTI MTU-231, Royal and ADLER Tape Drives, and Structure Design SD-2024 PAL Programmer, Stringy Floppy for TIMEX 1000 and 1500.

**MORE FOR MODEL I**

Holmes IM-2 48K in Keyboard Memory .. \$119.50  
Holmes SP-It Computer Speed-Up ..... \$ 89.50  
Upper/Lower Case Kit ..... \$ 14.95

Installation available for above items  
The following are registered trademarks:  
TRS-80, MicroWafer, Stringy Floppy, Scriptit, PMD-100  
Prices subject to change without notice.

---

### MODEL I ESF SOFTWARE

|                                        |       |         |
|----------------------------------------|-------|---------|
| Patch R/S ED/ASSEM Version 1.1 and 1.2 | ..... | \$11.95 |
| IJG Electric Pencil 2.0                | ..... | \$59.95 |
| Advanced Programmers Guide             | ..... | \$ 9.95 |
| File Management System I               | ..... | \$19.95 |
| Patchword (Patch Scriptit 1.0)         | ..... | \$ 9.95 |
| ESF Forth                              | ..... | \$45.00 |
| Electric Spreadsheet 16K               | ..... | \$44.95 |
| ESF-80 Monitor                         | ..... | \$ 9.95 |
| The Basic Toolkit                      | ..... | \$19.95 |

More Software Available

### IJG PUBLICATIONS

#### THE OTHER MYSTERIES SERIES

|                                |       |         |
|--------------------------------|-------|---------|
| TSR-80 Disk                    | ..... | \$20.25 |
| Microsoft Basic Decoded        | ..... | \$26.95 |
| The Custom TRS-80              | ..... | \$26.95 |
| Basic Faster & Better          | ..... | \$26.95 |
| Machine Language Disk I/O      | ..... | \$26.95 |
| TRSDOS 2.3 Decoded             | ..... | \$26.95 |
| BASIC Disk I/O Faster & Better | ..... | \$26.95 |

"How To do it On The TRS-80"  
by William Barden, Jr. .... \$25.00  
Application guide to the TRS-80  
Models I, II, III, 100, and color computer

**A&J Micro Drive**  
1050 E. Duane Ave., Suite I  
Sunnyvale, CA 94086  
(408) 732-9292      ✓ 166

Send \$4.95 for Owners Manual and more information

Check or Money Order Enclosed  
 C.O.D. add \$1.65  
 U.P.S. shipping cost, add \$3.00, unless other specified  
 CA residents add 6.5% sales tax  
 Charge to my  VISA  MASTERCARD

Card # \_\_\_\_\_ Exp. date \_\_\_\_\_  
Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

188 • 80 Micro, March 1984

**Expense Log Example**

First determine how to set up the data file. An expense log requires the following information: month, day, category, name, description, check number, and amount. For this example, assume you want all the fields printed on the same line. The log also needs a checking account.

Run Data Manager and select the Creation option from the main menu. Type in EXPLOG.DO for the file name and 7 for the number of fields.

The month field allows a maximum width of two numbers. In this example, enter Month for the field name, no to the numeric question, 2 for the width, and yes for the same line question. The maximum width for the day field is two, so respond to the prompts in this field with Day, no, 2, and yes in sequence.

Expenses (for example, savings and food) belong in the category field, whose maximum width is eight characters. Enter Cat, no, 8, and yes in this field. The name field, with a maximum width of 18 characters, contains vendors' names—Radio Shack, for example. Enter Name, no, 18, and yes in this field.

Descriptions of merchandise delivered or services rendered ("Installed 8K RAM," for example) comprise the 20-character-wide description field. Enter Desc, no, 20, and yes to the prompts. The check number field has a maximum width of three characters (for a maximum of 999). Enter Check, no, 3, and yes. The amount field contains the amount of the transaction; its maximum integer width is five (for -9999.99), while its maximum fractional width is two characters. Enter Amount, yes, 5, and 2.

For this example, enter Amount for the name of the

amount field. Enter a 1 for the number of accounts and Checking for the account name. Records affect the checking account whenever the check number field is not empty. Enter Check for the name of the check field and press the enter key for the check string. Enter 200.00 for the balance.

Now select the Manage option by pressing the F1 key. Enter EXPLOG.DO as the file name.

Press the F2 key to select the Addition option. If you make a mistake entering any of the records, press the F1 key, the enter key, and then reenter the record. For the first record, enter 02 for the month, 12 for the day, xfer for the category, First Interstate for the name, deposit for the description, 000 for the check number, and 1000 for the amount. For the next record, enter 02, 12, food, Safeway, groceries, 237, and -23.89. For the next record, enter 02, 13, trans, Shell, gas, press the enter key, and then enter -12. At this point, the program displays the name of the first field of the next record. Press F2 and then the enter key.

Select the Report option and then the All-Records-With-Total option. After you enter Expense Log for the header, the program prints the report.

Now select the Accounts option. Data Manager displays the checking account balance, 1176.11. Press any key when you're ready to go back to the main menu.

Choose the End option to end the program and return to the Model 100 menu. ■

Contact Mark Hickenbottom at 28 Wrangler Court, Chico, CA 95926.

*Program Listing 3. Data Manager.*

```

0 CLEAR 1000:ON KEY GOSUB
1,2,3,4,5,6,7,8:KEY ON:GOTO 100
1 K=1:RETURN
2 K=2:RETURN
3 K=3:RETURN
4 K=4:RETURN
5 K=5:RETURN
6 K=6:RETURN
7 K=7:RETURN
8 K=8:RETURN
100 DIM FN$(30),FW(30),FF$(30),FSS$(30),
,RC$(30)
110 MS$="Configuring":GOSUB 10000
120 QU$="Can your printer print
graphics":GOSUB 11000
130 IF NO THEN 160
140 P1$=CHR$(188):P2$=CHR$(172):P3$=CHR$(
204):P4$=CHR$(181)
150 P5$=CHR$(202):P6$=CHR$(173):P7$=CHR$(
174):GOTO 170
160
P1$="":P2$="":P3$="":P4$="":P5$="":
P6$="":P7$=""
170 PW=80:INPUT "Printer width";PW
180 IF FL$="" THEN MS$="Data manager"
ELSE MS$="Managing "+FL$
190 GOSUB 10000:K=0
200 PRINT "man   Manage a file"
210 PRINT "add   Add data to file"
220 PRINT "rep   Print report"

```

```

230 PRINT "accs  Display account
balances"
240 PRINT "new   Create a new file"
250 PRINT "end   End data manager"
260 PRINT "man   add rep accs new
end";
270 IF K=6 THEN MENU ELSE IF K=0 OR K>5
THEN 270
280 ON K GOSUB
1000,2000,3000,4000,5000:GOTO 180
1000 AB=0:RL=0
1010 MS$="File selection":GOSUB 10000
1020 INPUT "File to manage";FL$
1030 OPEN FL$ FOR INPUT AS 1
1040 INPUT #1,FQ
1050 FOR F=0 TO FQ-1
1060   INPUT #1,FN$(F):INPUT #1,FW(F)
:INPUT #1,FF$(F):INPUT #1,FSS$
1070   IF FSS$="yes" THEN FF$(F)=FF$(F)+
":GOTO 1090
1080   FF$(F)=FF$(F)+CHR$(13)+CHR$(10)
:RL=RL+1
1090   NEXT
1100 INPUT #1,AF:INPUT #1,AQ:IF AQ=0
THEN 1140
1110 FOR A=0 TO AQ-1
1120   INPUT #1,AN$(A):INPUT #1,AF(A)
:INPUT #1,AC$(A)
1130   INPUT #1,OB(A):AB(A)=OB(A):NEXT
1140 INPUT #1,TM$:CLOSE:RETURN
2000 IF FL$="" THEN 2180
2010 MS$="Data addition":GOSUB 10000

```

## C•NOTES

*Listing 3 continued*

```

2020 PRINT "During data addition,"
2030 PRINT "use the following function
keys:"
2040 PRINT "F1: Re-enter record."
2050 PRINT "F2: End adding data."
2060 GOSUB 12000
2070 OPEN FL$ FOR APPEND AS 1
2080 GOSUB 10000
2090 FOR F=0 TO FQ-1:K=0
2100   PRINT FN$(F);TAB(6);": ";
2110   LINE INPUT RC$(F)
2120   IF K=1 THEN 2080
2130   IF K=2 THEN 2190
2140   IF LEN(RC$(F))>FW(F) THEN 2100
2150   NEXT
2160 FOR F=0 TO FQ-1
2170   PRINT #1,RC$(F):NEXT
2180 GOTO 2080
2190 CLOSE:AB=0
2200 RETURN
3000 IF FL$="" THEN 3180
3010 AT=0:K=0:MS$="Report
generation":GOSUB 10000
3020 PRINT "all   Print all records"
3030 PRINT "allt  Print all records with
total"
3040 PRINT "sel   Print selected
records"
3050 PRINT "selt  Print selected records
with total"

```

```

3060 PRINT:PRINT
3070 PRINT "all  allt sel  selt";
3080 IF K=0 OR K>4 THEN 3080 ELSE IF (
K=2 OR K=4) AND AF=-1 THEN 3080
3090 CLS:IF K<3 THEN 3130
3100 INPUT "Field to check";FD$:GOSUB
13000
3110 IF FF=-1 THEN 3100
3120 INPUT "String to check for";FC$
3130 LINE INPUT "Header? ";HD$
3140 IF AB THEN 3170
3150 FOR A=0 TO AQ-1
3160   AB(A)=OB(A):NEXT
3170 PR=-1:GOSUB 14000
3180 RETURN
4000 IF FL$="" OR AQ=0 THEN 4080
4010 IF AB THEN 4040
4020 CLS:PR=0:FOR A=0 TO AQ-1
4030   AB(A)=OB(A):NEXT:GOSUB 14000
4040 MS$="Account balances":GOSUB 10000
4050 FOR A=0 TO AQ-1
4060   PRINT AN$(A);TAB(15);": ";AB(A)
:NEXT
4070 GOSUB 12000
4080 RETURN
5000 MS$="File creation":GOSUB 10000
5010 INPUT "File to create";FL$
5020 CLS:INPUT "Number of fields";FQ
5030 FOR F=0 TO FQ-1
5040   CLS:PRINT "Field";F

```

*Listing 3 continued*

# MODEL-16 FORTRAN-77

TriSoft is pleased to announce the availability of SVS FORTRAN-77 under TriSoft CP/M-68K for the Radio Shack Model-16 computer. This FORTRAN implements the full ANSI-77 standard and is not the subset FORTRAN-77 available on most microcomputers.

Full ANSI-77 FORTRAN on a 68000 based computer gives the user the power and flexibility of a mainframe computer at a fraction of the cost. FORTRAN has been the primary language for scientific, mathematical and statistical applications for many years. A vast wealth of programs already exist for a variety of applications but require a mainframe environment. FORTRAN-77 running under TriSoft CP/M-68K provides a suitable environment to allow the user to take advantage of these programs.

In addition to supporting the full ANSI-77 standard, the FORTRAN package allows you to link in routines written in DRI 'C' and 68000 Assembly languages.

SVS FORTRAN-77 is a native code compiler. This combined with the efficiency with which it was designed provides very fast execution speeds.

**TriSoft**  
**4102 Avenue G**  
**Austin, Texas 78751**  
**(512) 453-2233**  
**(800) 531-5170**

|                            |                 |
|----------------------------|-----------------|
| <b>FORTRAN-77</b>          | <b>\$495.00</b> |
| <b>CP/M-68K</b>            | <b>\$395.00</b> |
| <b>68K-BASIC</b>           | <b>\$299.00</b> |
| <b>Pickles &amp; Trout</b> | <b>CP/M 2.2</b> |

CP M-68K™ Digital Research  
Model-16™ Tandy



-227



# C•NOTES

Listing 3 continued

```

5050 INPUT "Name";FN$(F)
5060 IF LEN(FN$(F))>6 THEN 5050
5070 QU$="Is field numeric":GOSUB
11000
5080 IF YES THEN 5120
5090 INPUT "Width of field";FW(F)
5100 IF FW(F)=1 THEN FF$(F)="!":GOTO
5160
5110 FF$(F)=" "+SPACES$(FW(F)-2)+
"\":GOTO 5160
5120 INPUT "Width of integer
portion";WI
5130 INPUT "Width of fractional
portion";WF
5140 FW(F)=WI:FF$(F)=STRING$(WI,"#")
5150 IF WF>0 THEN FW(F)=FW(F)+1+WF:FF$(
F)=FF$(F)+". "+STRING$(WF,"#")
5160 IF F=FQ-1 THEN YES=0:GOTO 5180
5170 QU$="Is the next field on the
same line":GOSUB 11000
5180 IF YES THEN FS$(F)="yes" ELSE FS$(
F)="no"
5190 NEXT
5200 CLS:INPUT "Name of amount
field";FD$
5210 GOSUB 13000:AF=FF
5220 CLS:AQ=0:INPUT "Number of
accounts";AQ
5230 IF AQ=0 THEN 5330
5240 FOR A=0 TO AQ-1

```

```

5250 CLS:PRINT "Account";A
5260 INPUT "Name of account";AN$(A)
5270 IF LEN(AN$(A))>15 THEN 5260
5280 INPUT "Name of check
field";FD$:GOSUB 13000:AF(A)=FF
5290 IF AF(A)=-1 THEN 5280
5300 LINE INPUT "Check string? ";AC$(A
)
5310 INPUT "Account balance";AB(A)
5320 NEXT
5330 OPEN FL$ FOR OUTPUT AS 1
5340 PRINT #1,FQ:FOR F=0 TO FQ-1
5350 PRINT #1,FN$(F):PRINT #1,FW(F)
:PRINT #1,FF$(F):PRINT #1,FS$(F):NEXT
5360 PRINT #1,AF:PRINT #1,AQ:IF AQ=0
THEN 5390
5370 FOR A=0 TO AQ-1
5380 PRINT #1,AN$(A):PRINT #1,AF(A)
:PRINT #1,AC$(A):PRINT #1,AB(A):NEXT
5390 PRINT #1,"":CLOSE
5400 FL$="":RETURN
10000 CLS:ZL=INT((40-LEN(MS$))/2):ZR=40-
LEN(MS$)-ZL
10010 PRINT CHR$(27);"p";SPACES$(ZL)
;MS$;SPACES$(ZR);CHR$(27);"q";
10020 RETURN
11000 AN$="":PRINT QU$;:INPUT
AN$:AN$=LEFT$(AN$,1)
11010 IF AN$<>"y" AND AN$<>"Y" AND AN$<>
"n" AND AN$<>"N" THEN 11000

```

Listing 3 continued

## Get more use from your Model 100 Let the Elves help!

### ELF-WRITER

Expand your text processor! Features include:  
— Formatted viewing mode with horizontal scrolling — Formatting uses Wordstar™ dot commands: Variable margins; Pagination; Headings & Footings; Right justification of text; And Much More!!

### ELF-COP

Manage your file space problems!  
Backup all of your files at one time. A backup report documents your backup. ELF-COP will also Copy, Rename, Delete and Report file sizes.

TM - MicroPro International Corp.

**\$39.99 EACH**  
**•SPECIAL OFFER•**  
All 3 for \$90.00  
Price includes shipping.



### ELF-PAK = ELF 1 + 1 CALCULATOR + ELF-BANKER

ELF 1 + 1 turns your computer into an easy to use calculator. Features: Entry, Total & Memory displays; %, Exponent & Square Root keys; + More!

ELF-BANKER Evaluates your money for the most return. 20 calculations including: Compound interest, Present value, Loan & Mortgage payments, Annuities & Active Savings accounts.



Send the coupon below to:  
**Ceres Software Inc.**  
4303 S.W. Chesapeake Ave.  
Portland, OR 97201  
(503) 245-9011

125

Please send me on cassette:

- ELF-WRITER     ELF-COP  
 ELF-PAK  
 Special Offer \$90.00  
 Check for \_\_\_\_\_ enclosed  
 Send C.O.D.

NAME: \_\_\_\_\_

STREET: \_\_\_\_\_

CITY: \_\_\_\_\_ STATE: \_\_\_\_\_ ZIP: \_\_\_\_\_

TELEPHONE: \_\_\_\_\_

Listing 3 continued

```

11020 IF AN$="y" OR AN$="Y" THEN YES=-
1:NO=0
11030 IF AN$="n" OR AN$="N" THEN
YES=0:NO=-1
11040 RETURN
12000 IF INKEY$="" THEN 12000 ELSE
RETURN
13000 FOR F=0 TO FQ-1
13010 IF FN$(F)=FD$ THEN FF=F:GOTO
13030
13020 NEXT:FF=-1
13030 RETURN
14000 CLS:OPEN FL$ FOR INPUT AS 1
14010 FOR F=1 TO 4+4*FQ+4*AQ:LINE INPUT
#1, TM$:NEXT
14020 IF PR THEN GOSUB 15000
14030 IF EOF(1) THEN 14210
14040 FOR F=0 TO FQ-1:LINE INPUT #1, RC$(
F):NEXT
14050 IF AB OR AQ=0 THEN 14120
14060 FOR A=0 TO AQ-1
14070 IF AC$(A)="" THEN 14090
14080 IF RC$(AF(A))<>AC$(A) THEN 14110
ELSE 14100
14090 IF RC$(AF(A))="" THEN 14110
14100 AB(A)=AB(A)+VAL(RC$(AF))
14110 NEXT
14120 IF NOT PR THEN 14030
14130 IF K<3 THEN 14170
14140 IF FC$="" THEN 14160
14150 IF RC$(FF)<>FC$ THEN 14030 ELSE
14170
14160 IF RC$(FF)="" THEN 14030
14170 IF LS>=58 THEN GOSUB 16000:GOSUB
15000
14180 IF K=2 OR K=4 THEN AT=AT+VAL(RC$(
AF))
14190 FOR F=0 TO FQ-1
14200 FD$=RC$(F):GOSUB
17000:NEXT:LS=LS+RL:GOTO 14030
14210 IF NOT PR THEN 14300
14220 IF K=1 OR K=3 THEN 14290
14230 FOR F=0 TO FQ-1
14240 FD$=STRING$(FW(F), "-"):GOSUB
17000:NEXT:LS=LS+RL
14250 FOR F=0 TO FQ-1
14260 IF F=AF THEN FD$=STR$(AT):GOSUB
17000:GOTO 14280
14270 FD$=SPACE$(FW(F)):GOSUB 17000
14280 NEXT:LS=LS+RL
14290 GOSUB 16000
14300 CLOSE:AB=-1:RETURN
15000 FOR ZZ=1 TO 6:LPRINT "":NEXT
15010 LPRINT P1$;STRING$(PW-2,P2$);P3$
15020 ZL=INT((PW-2-LEN(HD$))/2):ZR=PW-2-
LEN(HD$)-ZL
15030 LPRINT P4$;SPACE$(ZL);HD$;SPACE$(
ZR);P5$
15040 LPRINT P6$;STRING$(PW-2,P2$);P7$
15050 LPRINT "":LS=10:RETURN
16000 LPRINT CHR$(12);:LS=0:RETURN
17000 IF INSTR(FF$(F), "#")=0 THEN LPRINT
USING FF$(F);FD$;:GOTO 17030
17010 IF VAL(FD$)<>0 OR FD$=" 0" THEN
LPRINT USING FF$(F);VAL(FD$);:GOTO 17030

17020 LPRINT SPACE$(FW(F)-LEN(FD$))
;FD$;RIGHT$(FF$(F), 2);
17030 RETURN
    
```

End

## Wipe Out

by Thomas Robb

The conventional way to kill a file on the Model 100 is to load Basic and type in KILL followed by the file name and extension. I wrote a program that simplifies this process; Delete lets you erase unwanted directory files and programs with a single keystroke (see Program Listing 4).

I also wrote a related program, DIRTBL, that PEEKs into the Model 100's file directory to examine the directory table (see Program Listing 5).

### Deleting Files

In the Delete program (see Listing 4), line 200 and the subroutine in lines 1100-1125 clear the screen and display active RAM files through the Files command, with a delete prompt at the bottom of the screen. Line 220 forms the For...Next loop to examine the directory table entries.

Variable F% in line 1205 picks up each entry's directory flag. The program bypasses the file (that is, it doesn't display it with the delete prompt) if it isn't a valid entry (bit 7 doesn't equal 1), if it's one of the five ROM based programs (bit 4 equals 1), or if it's an "invisible" entry (bit 3 equals 1).

For all remaining files, line 1215 builds the ASCII file name in string variable FN\$ from this entry's directory table in bytes 4-11. Line 1217 bypasses Delete, then displays the file name beside the delete prompt. Typing an upper- or lowercase Y kills the file; any other response leaves the file intact.

### The Directory Table

The directory table is located at RAM address F962 hexadecimal (63842 decimal), regardless of your unit's memory size. To see the table, in Basic type PRINT PEEK(63842). You should get back 176. An 11-byte directory entry manages each potential file in the system. There are 27 possible file entries in the directory table, 24 of which you can display on the menu and three "invisible" entries (i.e., they don't appear on the menu). The first byte, byte 1 of the 11-byte file entry, or the directory flag, contains the information listed in Table 3.

The directory flag for Basic reads B0 hexadecimal (260 octal) because bits 7, 5, and 4 are set. The flag is located at the first byte of the directory table, F962 hexadecimal (hex). The directory flags for the other four ROM-based programs (Text, TELCOM, ADDRSS, and SCHEDL) are the same; you'll

| Directory flag bits | Description                                          |
|---------------------|------------------------------------------------------|
| Bit 7*              | 1 = valid entry                                      |
| Bit 6               | 1 = ASCII text file (extension is .DO)               |
| Bit 5               | 1 = machine-language file (implied extension of .CO) |
| Bit 4               | 1 = ROM file                                         |
| Bit 3               | 1 = "invisible" file                                 |
| Bit 2               | reserved                                             |
| Bit 1               | reserved                                             |
| Bit 0†              | internal use only                                    |

\*MSB (most significant bit)  
†LSB (least significant bit)

Table 3. The first byte of the directory file entry.

## C•NOTES

find them at F96D hex, F978 hex, F983 hex, and F98E hex, respectively. The directory flag for an ASCII file, for example, is C0 hex (300 octal) because bits 7 (valid entry) and 6 (ASCII file) are set.

Bytes 2-3 of each valid entry in the directory table give the first, or starting, address for that file. Byte 2 is the low-order byte; byte 3 is the high-order byte. (This may seem backward but it's consistent with the Model 100's 80C85 chip addressing scheme.) For Basic, this starting address works out to be 6C49 hex (27721 decimal). Bytes 4-11 of each valid entry contain the 8-byte ASCII file name.

Assuming a maximum of 24 files in the Model 100, I was surprised to find 27 entries in my directory table. As I'd expected, the first five entries are the ROM-based programs. The

```

190 DIRTBL=63842
200 FOR J%=1 TO 27
210 F%=PEEK(DIRTBL+(J%-1)*11)
220 HB%=PEEK(DIRTBL+(J%-1)*11+2)
225 LB%=PEEK(DIRTBL+(J%-1)*11+1)
230 PRINT "File" J% " " F% HB%*256+LB%

240 FOR I%=3 TO 10
250 PRINT CHR$(PEEK(DIRTBL+(J%-1)*11+
I%));
260 NEXT I%
265 PRINT
270 NEXT J%
    
```

*Program Listing 5. DIRTBL.*

```

100 ' Model 100 DIRECTORY file delete
      with QUERY
110 ' (C) 1983 by Thomas L. Robb,
      WBL, MN
120 '
200 CLS: FILES: GOSUB 1100: ' Display
      directory and prompt screen
210 DIR=63842 ' RAM Location of
      directory table (F962h)
220 FOR J%=1 TO 27: GOSUB 1200: NEXT '
      Consider all directory entries
230 CLS: MENU ' Exit
1100 '
1101 ' Display prompt screen
1102 XY%=282 ' PRINT @ position
1103 X1%=57: Y1%=53 ' Pixel coordinates
      for FILENAME hilite
1105 PRINT @XY%,"Delete";
1110 LINE (X1%,Y1%)-(X1%,Y1%+10): LINE
      (X1%,Y1%)-(X1%+58,Y1%): LINE (X1%+58,Y1%
      )-(X1%+58,Y1%+10) ' Hilite FILENAME
1115 PRINT @XY%+14,".";
1120 PRINT @XY%+19,"(Y/N)?";
    
```

```

1125 RETURN
1200 '
1205 F%=PEEK(DIR+(J%-1)*11) ' Pickup
      FILE Directory flag
1210 IF (F%<128) OR ((F% AND 16)>0) OR
      ((F% AND 8)>0) THEN RETURN ' Bypass this
      FILE if NOT VALID entry, a ROM entry, or
      an INVISIBLE entry
1215 FN$="": FOR I%=3 TO 10: FN$=FN$+
      CHR$(PEEK(DIR+(J%-1)*11+I%)): NEXT '
      Build this entry's FILENAME
1217 IF FN$="DELETEBA" THEN RETURN ' DO
      NOT allow this program to be deleted
1220 PRINT @XY%+8,LEFT$(FN$,6);: PRINT
      @XY%+15,RIGHT$(FN$,2); ' Display this
      FILENAME
1225 PRINT @XY%+26," "":: PRINT
      @XY%+26,: K$=INPUT$(1) ' Wait for user
      response
1230 IF K$="Y" OR K$="y" THEN PRINT @XY%
      +26,"Deleted";: KILL FN$ ELSE PRINT @XY%
      +26,"No";
1235 RETURN
    
```

*Program Listing 4. Delete.*

### "WHERE TO FIND FREE PROGRAMS FOR YOUR TRS-80, APPLE, OR IBM MICROCOMPUTER"

**Vol. 1. Index to 4000 BASIC Programs**  
Contained in 160 Popular Computer Books  
176 pages. \$14.95

**Vol. 2. Index to 4000 BASIC Programs**  
Contained in 500 Computer Magazines 1979-1983  
196 pages. \$14.95

All 8000 programs described by title and indexed by  
subject — Astronomy to Bookkeeping to Games to  
Graphics to Utilities to Zoology, etc.

For programmers, computer users, teachers, stu-  
dents — Learn where to find program listings and  
algorithms. Saves hours of programming time.



Attn. Dept. EM  
PASADENA TECHNOLOGY PRESS  
P. O. Box 3836  
South El Monte, CA 91733

✓ 20

Include \$2.00 Postage and Handling per order.  
California residents add 6.5% sales tax.

## VITAMIN and MINERAL DEFICIENCY CHART

Your computer will analyze your answers to 192 questions  
concerning your body and then print a chart showing probable  
deficiencies of 33 vitamins and minerals plus protein and fat.  
(Such questions as, Do your eyes hurt when you go out into  
the sun? Is your hair beginning to grey? Do you sleep well?  
etc etc.)

Covers complete with 50 questionnaires, 50 blank Nutritional  
program forms, instructions on completing the Nutritional  
forms from your computer generated chart, and an excellent  
instruction manual and a 5 1/4 inch disk. This program can put  
you into a business of your own selling nutritional programs.

**ALLIED SYSTEMS INC**

P. O. BOX 245 TRONA, CALIF. 93562

FOR MODELS I/III/4 48K 619 372-5355  
C. O. D. AND MONEY ORDERS \$89.50  
RRE RUSH ORDERS ✓ 140

\*\*\*\*\*  
SuperSCRIPT PRINTER DRIVERS  
\*\*\*\*\*

With an ALPS Printer Driver (software), you can  
attach your printer to SuperSCRIPT Word Process-  
ing for automatic interfacing. Over 70 printers now  
supported. Call or write for latest Product Review  
Sheet. Retailing features supported.  
\$49 each w/CD. Proportional. \$39 with Proportional

\*\*\*\*\*  
MEMBER FOR MODEL 3 IN MODEL 10 MODE  
\*\*\*\*\*

Easy to use (LDOS only). Use MEMBER as your  
System Drive or load other files into memory for  
Instant Access. Use 50K of 128K memory. Too  
many features to describe here. Call or write for  
more details. \$99.

\*\*\*\*\*  
MODEL 3 UTILITIES (FOR MODEL 10 MODE) \$19  
\*\*\*\*\*

Run Model 3 at High CPU speed (4MHz)  
Test Memory (to 256K Automatic & Clear Memory  
Early Speed (Beep) or JCL file or BASIC

\*\*\*\*\*  
MEMORY DIAGNOSTIC (MAIN Screen) \$24  
LOADING UTILITY \$19  
\*\*\*\*\*

MEM/Phone Orders Accepted

**ALPS**  
34 Apple Road  
Wanamansett, New York 11740  
201-647-7236

✓ 74

## Program Listing 6. SHOOTR.

```

0 ' SHOOTR.BA CRAPS ON THE 100
5 ' BY RON BALONIS 11/24/3
10 CLEAR 200:DIM KUR(12),ODDS(12)
15 RVID$=CHR$(27)+"p":NVID$=CHR$(27)+"q"

20 ODD$=RVID$+"##"+NVID$
25 '---RANDOMIZE
30 T=VAL(RIGHT$(TIMES,2))
32 FOR I=1 TO T*3+7
34 X=RND(1)
36 NEXT I
38 FOR I=1 TO 12:'--- POINT CURSOR
40 READ KUR(I)
42 NEXT I
44 FOR I=1 TO 12:'--- POINT ODDS
46 READ ODD$(I)
48 NEXT I
50 CLS:'---MAKE THE GAME SCREEN
52 PRINT@203,"***** CASINO CRAPS
SIMULATION *****";
54 PRINT@289,"=== PRESS E TO EXIT ===";
56 PRINT@41,"CRAPS POINT WIN";
58 PRINT@67,"POINT CRAPS";
60 PRINT@81," 2 3 4 5 6 7 11";
62 PRINT@106," 8 9 10 12";
65 LINE(2,3)-(236,27),1,B
70 FOR X=2 TO 74 STEP 18
72 FOR Y=16 TO 26
74 PSET(X,Y):PSET(X+144,Y)
76 NEXT Y
78 NEXT X
80 FOR Y=4 TO 26
82 PSET(38,Y):PSET(92,Y):PSET(98,Y)
84 PSET(140,Y):PSET(146,Y):PSET(200,Y)
86 NEXT Y
95 '---COME OUT ROLL
100 PRINT@200,"CHANCES ARE: 22.2%";

105 PRINT" 67.7% 11.1%";
110 PRINT@286,"== PRESS SPACEBAR TO ROLL
==";
120 GOSUB 400:'---GET A ROLL
130 P0INT=DT0T:DT$=STR$(DT0T)
140 DT$=RIGHT$(DT$,LEN(DT$)-1)
150 IF INSTR("4568910",DT$)<>0 THEN 200
160 IF DT0T=7 OR DT0T=11 THEN 360
170 PRINT@286,"*** CRAPS! SHOOTER
LOSES ***";
175 RUN
180 '
190 '---POINT ESTABLISHED
200 PRINT@41,"ROLL ";:PRINT@74,"ROLL ";
210 PRINT@58,"LOSE";:PRINT@98," 7 ";
220 PRINT@114,"11";
230 PRINT@KUR(DT0T),USINGODD$:DT0T;
240 PRINT@280," % CHANCE";
250 PRINT" <PRESS SPACEBAR TO ROLL>";
260 PRINT@280,USING"##.##";ODDS(P0INT);
290 '
300 GOSUB 400:'---GET A ROLL
320 IF P0INT=DT0T THEN 360
330 IF DT0T<>7 THEN 300
340 PRINT@280,SPACE$(39);
350 PRINT@286,"***** SHOOTER LOSES! ***
***";
355 RUN
360 PRINT@280,SPACE$(39);
365 PRINT@286,"***** SHOOTER WINS! ***
***";

```

sixth entry is an "invisible" file (directory flag is 88 hex) located at C602 hex in my machine with a file name of Suzuki. The seventh entry, also an invisible ASCII file, is named Hayashki. The eighth entry is always an invalid entry, but the only one with a directory flag of 00 hex. Files 9-27 are valid files.

ASCII file Hayashki is an empty file (that is, its first byte is the end-of-file sentinel, control-Z). The Suzuki file contains two 00 hex bytes. Obviously, these are not user files—perhaps they represent the "fingerprint" of someone associated with the design of the machine. David Sumner's memory map (80 Micro, December 1983, pp. 232 and ff.) shows Suzuki to be any Basic program not yet saved to RAM and Hayashki to be the end address of documents.

To display the directory in your machine, run the short program in Listing 5. You'll see the directory flag, starting address, and name of every file in the DIRTBL. ■

You can reach Thomas Robb at 2632 Sumac Ridge, White Bear, MN 55110.

## A New Pair of Shoes

by Ronald F. Balonis

SHOOTR simulates a casino craps game with a success probability readout at each roll to help you develop a gambler's most important skill—knowing when to quit (see Program Listing 6). Although the program doesn't let you make bets, it'll give you an idea of the odds you're up against in craps.

### The Game

The outcome of craps depends on the throw of the dice. There are 36 ways a pair of dice can make number totals of from 2 to 12; the odds range from 1 in 36 for a 2 or a 12, to 1 in 6 for a 7. In casino craps when your first, or come-out, roll results in a 7 or an 11, you win immediately. A roll of 2, 3, or 12 is an immediate loss (craps), while a roll of 4, 5, 6, 8, 9, or 10 means you have established a point and can roll again. Your point value appears on the display in inverse video.

From then on, you continue to roll the dice until you either

```

370 RUN
390 '---ROLL THE DICE
400 POKE 65450,0:'---CLEAR KEYBOARD QUEUE
410 X=RND(1):KB$=INKEY$
420 IF KB$="E" THEN MENU
430 IF KB$<>" " THEN 410
450 '
490 '
500 PRINT@160,SPACE$(120);
510 D1=INT(RND(1)*6+1)
520 D2=INT(RND(1)*6+1)
530 X=65:Y=31:DIE!=D1:GOSUB 1000
540 X=150:Y=31:DIE!=D2:GOSUB 1000
550 DT0T=D1+D2:RETURN
900 '
1000 '---DIE GRAPHICS
1010 LINE (X+1,Y+1)-(X+21,Y+21),1,B
1020 ON DIE! GOSUB

```

Listing 6 continued



## C•NOTES

lose with a roll of 7, or win when you roll the point value you set in your first roll. All other number combinations represent points or rolls, which let you continue shooting the dice. Success in this game is determined by the betting strategy at each throw of the dice.

### The Program

SHOOTR's algorithm is almost a direct coding of the game's rules and play sequence. To roll the dice, press the space bar. To exit the game, press the E key instead of the space bar.

The lines of code before line 100 define the come-out roll screen and initialize variables. The program processes the first roll between lines 100 and 200. It then displays an immediate win or loss, and, after a short delay, the first screen appears.

Then the program continues at lines 200-400; you roll until you get a 7 to lose or your point to win.

Lines 400-550 form the dice-rolling routine and lines 1000-1700, the dice graphics routine. The program uses the Model 100's pseudorandom function to roll the dice, and to make them natural, randomizes them in lines 30-36 and 400.

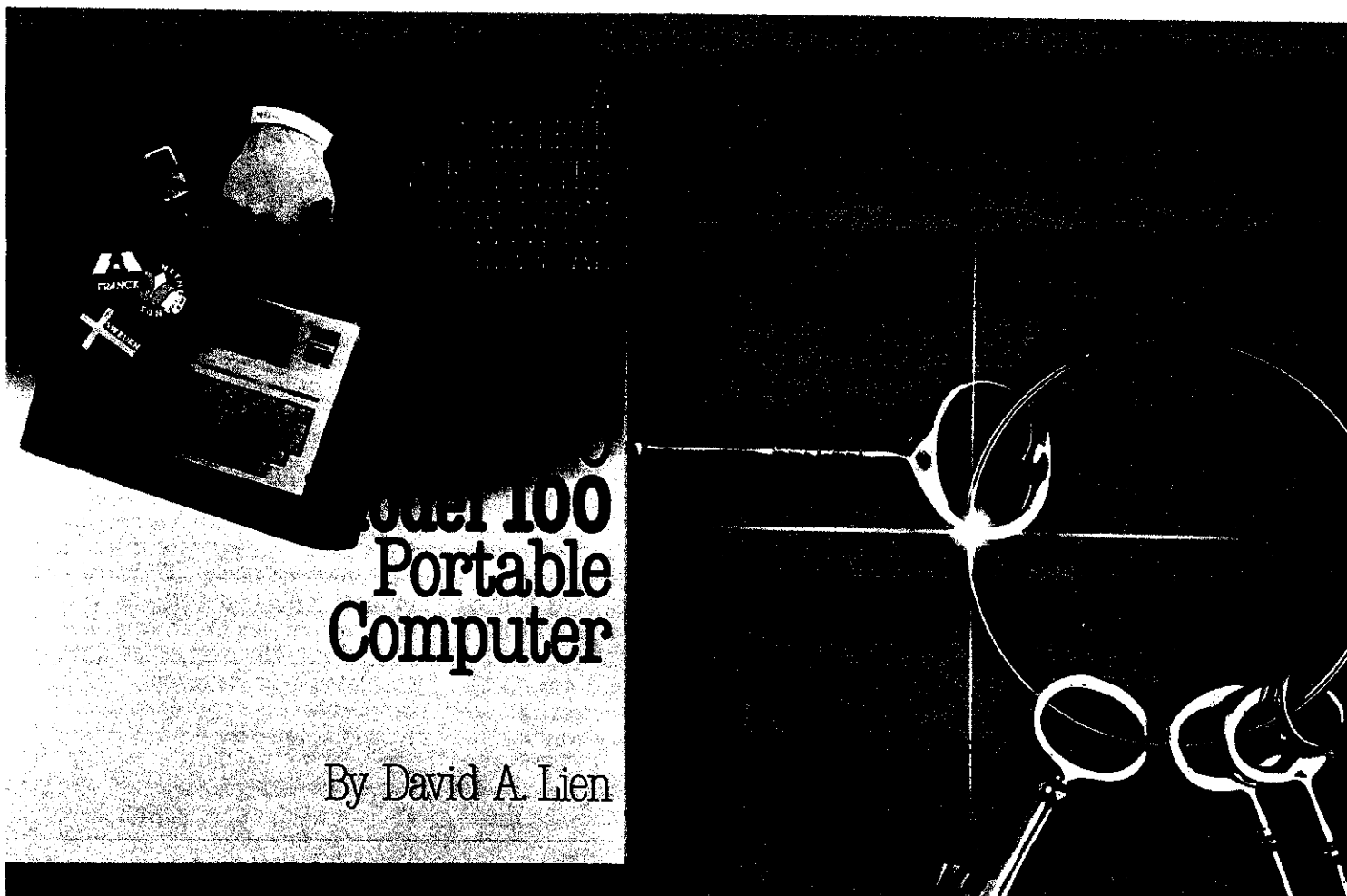
Lines 100 and 260 display the odds of making a number at each roll.

With practice, you'll become a hot shooter who knows when to bet big. ■

*You can reach Ronald F. Balonis at 118 Rice St., Trucksville, PA 18708.*

*Listing 6 continued*

```
1100,1200,1300,1400,1500,1600
1030 RETURN
1090 '---ONE DOT
1100 LINE(X+10,Y+10)-(X+12,Y+12),1,BF
1120 RETURN
1190 '---TWO DOTS
1200 LINE(X+4,Y+4)-(X+6,Y+6),1,BF
1220 LINE(X+16,Y+16)-(X+18,Y+18),1,BF
1230 RETURN
1290 '---THREE DOTS ONE + TWO
1300 GOSUB 1100:GOTO 1200
1390 '---FOUR DOTS TWO + TWO
1400 LINE(X+4,Y+16)-(X+6,Y+18),1,BF
1450 LINE(X+16,Y+4)-(X+18,Y+6),1,BF
1480 GOTO 1200
1490 '---FIVE DOTS FOUR + ONE
1500 GOSUB 1400:GOTO 1100
1590 '---SIX DOTS TWO + FOUR
1600 LINE(X+4,Y+10)-(X+6,Y+12),1,BF
1650 LINE(X+16,Y+10)-(X+18,Y+12),1,BF
1700 GOTO 1400
4900 '
5000 '---POINT CURSOR TABLE
5010 DATA 0, 81, 84, 87, 90, 93
5020 DATA 0,105,108,111,114,117
5030 '---CHANCE OF MAKING POINT TABLE
5040 DATA 0.0, 0.0, 0.0,33.3,40.0,45.4
5050 DATA 0.0,45.5,40.0,33.3, 0.0, 0.0
6000 '-----END OF PROGRAM
```



**Model 100  
Portable  
Computer**

By David A. Lien

C-NOTE FOR APRIL 1984

## Calling Wall Street

by Thomas L. Robb

While stock market trading prices fluctuate daily, the Model 100's display remains the same—making it difficult to cram stock quotations from the Dow Jones News/Retrieval Service into the 8-line by 40-column screen.

Few information utilities are set up to work with such a small display. Dow Jones in particular doesn't support the XON/XOFF protocol needed to stop the flow of data from the host. But with DJN/R.BA (Program Listing 1) you won't have to speed-read scrolling stock statistics: The program displays one quotation on the screen at a time.

### Making the Connection

You can access the Dow Jones via the Model 100's TELCOM program in two ways. If Tymnet is your packet switcher, you're in luck: Tymnet supports the XON/XOFF protocol. In response to the log-in prompt, simply type control-R before entering DOW1;. Control-S (or the pause key) stops the scroll; control-Q restarts it—even with a TELCOM line status parameter of D (disable XOFF). In fact, I always use the parameter M7I1D, 10 since I don't find M7I1E, 10 useful in programmed access or the TELCOM mode.

Another way to get a readable columnar display of a single stock quotation (see Fig. 1) when using the Dow Jones stock quote reporter is to enter the command //SIZE32. (//SIZE40, which represents the default screen format, would display 80 characters in two word-wrapped lines, each of up to five stock quotes in columns, one per line.)

Before using DJN/R.BA, insert the 9-pin direct-connect modem adapter cable plug into the phone connection at the back of the Model 100. Plug the RJ-11 telephone jack into the beige adapter on the modem cable, then, to keep the telephone connected, plug the RJ-11 jack on the silver cable into the phone (optional).

Remember to put your local Tymnet telephone number in line 1402 of the program and your password in line 1730.

When you run the program, it auto-dials and signs onto the Dow Jones service via the Tymnet packet switch network. It then displays a screen (see Fig. 2) to prompt you for the symbol of a particular stock. If, for example, you type in IBM in response to the prompt, the program displays the screen depicted in Fig. 3.

|                |         |
|----------------|---------|
| STOCK          | IBM     |
| BID/CLOSE      | 120 1/2 |
| ASKED/OPEN     | 120 3/4 |
| HIGH           | 121 1/4 |
| LOW            | 120 3/8 |
| LAST           | 121     |
| VOLUME (100'S) | 1665    |

Figure 1. Columnar stock quotation obtained using TELCOM with format //SIZE 32.

To end the program, depress the enter key without a stock symbol (see line 620). My remark statements should help you analyze the program.

### Problems

The program can hang up. Line 635 requests input from the modem port. If it receives none, control never passes to the next statement. In Model 100 Basic, even with a hardware clock, this program doesn't time-out communications input.

If nothing appears on the screen for 30 to 40 seconds when you're using DJN/R.BA, reset the computer and run the program again. You won't lose your data since I set up the Quote.DO file (see line 636) to capture all the two-way communications between the program and the host. Using the text editor, you can re-create the data exchanges up to the hung condition.

### Graphing Stock Statistics

My related program, Graph.BA (see Program Listing 2), also accesses Dow Jones, then gathers historical stock data and graphs the monthly volume and closing price for a selected stock on the Model 100's screen.

You can obtain monthly historical stock data for common and preferred stocks and warrants from 1979 to the present. Available markets include the New York, American, Pacific, Midwest, and national Over the Counter (OTC) stock exchanges. Since the national OTC doesn't carry a monthly closing price, I set up the query (line 627) to exclude a national OTC stock request and to request New York stock exchange data (the 1 following the semicolon). A 2 identifies the American, 3 the Pacific, and 4 the Midwest stock exchanges.

On the graph, double vertical bars represent volume data (the right vertical axis); horizontal dashes represent closing prices (left vertical axis). I omitted units of thousands (000) on the volume scale. The program uses the ON MDM GOSUB command, rather than INPUT or LINE INPUT commands, to capture modem input. Through this logic, you can detect time-out conditions and request retransmissions, thereby avoiding hung programs.

I chose to alternate between two interrupt routines—using one that, during stock transmission, writes the received characters to a RAM file, Quote.DO (see line 390), and the other that, during sign-on and sign-off, doesn't write to the RAM file (see line 380). Even at the slow transmission rate of 300 baud, the combined number of statements you can execute in the program's wait loop (lines 630–635) and the interrupt loop (line 390) is limited. If you require more logic in either

### The Key Box

The programs in "Calling Wall Street" require 24K RAM. The programs in "North by Northwest" and "Graphic Results" run in 8K RAM.

## C•NOTES

loop, you can begin to drop input characters from the RAM file (file 3).

As in DJN/R.BA (Program Listing 1), set up your local Tymnet telephone number (line 1402) and Dow Jones password (line 1730) in Graph (Program Listing 2) before you run the program.

Once you've selected Graph from the main menu, it makes the Dow Jones connection and then prompts you for a stock symbol and a year. The program goes into a timing loop, and if the computer receives no modem input before the loop is over, the program detects a time-out condition.

During normal operation, there will be about a 30-second wait while the program collects data in the temporary file Quote.DO. (You need an available slot for this file in your directory.)

The program then generates and displays the graph. If you hear a beep during the sign-on or data collection phase, the program has timed-out or received garbled data and will request a retransmission. The program terminates after three unsuccessful retransmissions.

The program is in an INKEY\$ loop (lines 728-730) when it displays the graph. Depress any key to return to the stock symbol prompt, and then the enter key to terminate the program. The program signs-off, disconnects, displays the connect time and charges, deletes the Quote.DO file, and then returns to the main menu.

Dow Jones News/Retrieval returns a footnote to indicate a special situation associated with the returned data. For example, in the graph of IBM for 1979, a stock split occurred. If the stock symbol in the lower left of the screen is flashing, depress any key to display the footnote.

### The Bottom Line

Accessing Dow Jones during prime time (between 5 a.m. and 6 p.m.) costs 90 cents per minute; at other times, the rates drop to 15 cents per minute. ■

*Contact Thomas L. Robb at 2632 Sumac Ridge, White Bear, MN 55110.*

```

Dow Jones NEWS/RETRIEVAL Service (C)
      08/16/83 10:10:05
      SYMBOL:
CLOSE(BID) OPEN(ASK)  HIGH      LOW

LAST          VOLUME(100'S)
```

*Figure 2. Screen prompt for individual stock symbol.*

```

Dow Jones NEWS/RETRIEVAL Service (C)
      08/16/83 10:15:21
      SYMBOL: IBM
CLOSE(BID) OPEN(ASK)  HIGH      LOW
120 1/2    120 3/4    121 1/4    120 3/8

LAST 121          VOLUME(100'S) 1665
```

*Figure 3. Display of IBM stock statistics.*

```

      Program Listing 1. DJN/R.BA.
300 ' Dow Jones News/Retrieval
      Programmed Access
302 '
305 '      Change History
310 ' 7-20-83 Initial release
400 '
410 CLEAR 512:MAXFILES=3: ' Setup string
      space and maximum number of files
415 ON ERROR GOTO 9000: ' General error
      paragraph
500 '
501 '      Control paragraph
510 GOSUB 1300: 'Setup system variables
525 GOSUB 1400: ' Dial local TYMNET
      number
535 GOSUB 1500: ' Open MODEM files
540 GOSUB 1600: ' Open RAM file
550 GOSUB 1700: ' Dow Jones signon
575 GOSUB 1800: ' Receive signon message
      from Dow Jones
585 PRINT @163,"Connect on ";DATE$;" at
      ";TIME$: ' Screen display
590 FOR I%=1 TO 3000: NEXT: ' Delay
      while CONNECT message being displayed
600 '
610 GOSUB 3000: ' Display STOCK
      background
615 GOSUB 3100: ' Solicit STOCK symbol
620 IF SYMBOL$="" THEN 900: ' Terminate
      if no user input entered
626 GOSUB 3300: ' Clear output fields
```

```

627 PRINT #2,",";SYMBOL$: ' Comma (,) is
      the DJN/R prefix for Common and
      Preferred stocks
629 ' Read data until input stock symbol
      found, then format screen display
630 FOR I%=1 TO 25
635 LINE INPUT #1,Z$
636 PRINT #3,Z$: ' Save communications
      data to RAM file, QUOTE.DO
637 IF LEFT$(Z$,LEN(SYMBOL$))=SYMBOL$
      THEN QUOTE$=Z$: GOSUB 3200: I%=25
640 NEXT
695 GOTO 615: ' Re-loop to solicit next
      stock quote
900 'Exit paragraph
905 GOSUB 1900: ' Send disconnect
      command
910 GOSUB 2000: ' Capture disconnect
      message
915 GOSUB 2100: ' Display elapsed time
      and cost (@ $.15/min NON-prime time)
925 CLOSE: ' Close all files
935 CALL 21179: ' Hangup telephone
945 GOSUB 2200: ' Delay before exit to
      allow screen display to be read
947 CALL 23164,0,23366: CALL 27795: ' Re
      -establish BASIC'S FUNCTION keys
955 MENU: ' Return to system display
995 '
1200 ' Subroutines
1300 '
```

*Listing 1 continued*



# C•NOTES

Listing 1 continued

```

1301 ' Initialize system variables
1305 KEY OFF: ' Disable FUNCTION keys
1310 SCREEN 0,0: ' Turn labels OFF
1315 SOUND OFF: ' Turn MODEM sound OFF
1320 CLS: ' Clear screen
1325 RETURN
1400 '
1401 ' Dial TYMNET telephone number
1402 PH$="9,339-5200": M=VARPTR(PH$):
AD=PEEK(M+1)+PEEK(M+2)*256
1403 ' AD points to telephone number
string, PH$.
1405 PRINT @92,"Dialing ";
1410 LINE (116,14) - (170,25),1,B: '
Draw box around dial digits area
1415 CALL 21200: ' Take telephone OFF~
HOOK
1420 CALL 21293,0,AD: ' Dial number
1425 RETURN
1500 '
1501 ' Open MODEM files
1510 OPEN "MDM:7i1d" FOR INPUT AS 1: '
Communications input path
1520 OPEN "MDM:7i1d" FOR OUTPUT AS 2: '
Communications output path
1525 RETURN
1600 '
1601 ' Open scratch pad RAM file
1602 ' This file serves as a record of
the two-way communication dialog between
the program and Dow Jones News/
Retrieval. It can be examined with the
TEXT editor after a telephone call.
1605 OPEN "QUOTE.DO" FOR OUTPUT AS 3
1610 RETURN
1700 '
1701 ' Dow Jones signon
1705 PRINT @173,"Signing on .": '
Screen display
1710 PRINT #2,"A":; ' Response to
"please type your terminal identifier"
1715 PRINT #2,CHR$(18);"DOWL";;: PRINT
".":; ' Response to "please log in: "
1720 PRINT #2,"DJNS": PRINT ".":; '
Response to "WHAT SERVICE PLEASE???"
1730 PRINT #2,"mypassword": PRINT ".":;
' Response to "ENTER PASSWORD"
1735 RETURN
1800 '
1805 ' Look for ENTER QUERY (end of
signon banner)
1810 LINE INPUT #1,Z$
1815 PRINT #3,Z$
1820 IF Z$<>"ENTER QUERY" THEN 1810
1830 RETURN
1900 '
1901 ' Disconnect
1910 PRINT #2,"DISC"
1920 RETURN
2000 '
2001 ' Capture sign-off message
2010 FOR I%=1 TO 25
2015 LINE INPUT #1,Z$:
2020 IF LEFT$(Z$,7)="LOG ON:" THEN
I%=25
2022 PRINT #3,Z$
2025 NEXT
2040 RETURN

```

```

2100 '
2101 ' Display elapsed time and cost
2110 Z%=INSTR(1,Z$,"ON: "): F1%=VAL(MID$(
Z$,Z%+4,2)): F2%=VAL(MID$(Z$,Z%+7,2))
2120 Z%=INSTR(1,Z$,"OFF: "): F3%=VAL(
MID$(Z$,Z%+5,2)): F4%=VAL(MID$(Z$,Z%+8,2
))
2125 IF F3%<F1% THEN F3%=F3%+24
2135 F5%=(F3%*60+F4%)-(F1%*60+F2%)
2140 IF F5%=0 THEN F5%=1
2145 CLS: ' Clear screen
2150 PRINT @201,"Elapsed MINUTES: "
USING "###";F5%;
2160 PRINT "; COST:" USING
"$$.###.##";F5%*.15
2180 RETURN
2200 '
2201 ' Delay for 15 secs to display sign
-off message on screen
2210 FOR I%=1 TO 15000: NEXT
2220 RETURN
3000 '
3001 ' Display STOCK background
3002 CLS: ' Clear screen
3005 PRINT " Dow Jones NEWS/RETRIEVAL
Service (C)"
3010 PRINT @52,DATE$;" ";TIME$
3015 PRINT @95,"SYMBOL:"
3020 PRINT "CLOSE(BID) OPEN(ASK) HIGH
LOW"
3025 PRINT @241,"LAST";: PRINT
@257,"VOLUME(100'S)"
3030 RETURN
3100 '
3101 ' Solicit STOCK symbol
3105 PRINT @103,"":; LINE INPUT
";SYMBOL$
3110 FOR I%=1 TO LEN(SYMBOL$)
3112 IF SYMBOL$="" THEN RETURN: ' If no
data entered, terminate program
3115 IF ASC(MID$(SYMBOL$,I%,1))=>97 AND
ASC(MID$(SYMBOL$,I%,1))<=122 THEN MID$(
SYMBOL$,I%)=CHR$(ASC(MID$(SYMBOL$,I%))-
32): ' Convert alpha lower case to upper
case to match returning stock symbol
3120 NEXT
3125 RETURN
3200 '
3201 ' Display STOCK quote
3205 PRINT @103,LEFT$(QUOTE$,10);
3210 PRINT @161,MID$(QUOTE$,11,40);
3215 PRINT @246,MID$(QUOTE$,51,10);
3220 PRINT @271,MID$(QUOTE$,61,10);
3230 RETURN
3300 '
3305 ' Clear STOCK background
3310 PRINT @103,SPACE$(10);: PRINT
@161,SPACE$(40);
3315 PRINT @246,SPACE$(10);: PRINT
@271,SPACE$(10);
3330 RETURN
9000 '
9001 ' General ERROR paragraph
9050 CLS: ' Clear screen
9060 PRINT @41,"An application program
error, CODE ";ERR;";";
9065 PRINT " has occurred at BASIC
program"
9070 PRINT " line NUMBER ";ERL;". "
9075 GOTO 900: ' Exit

```

End

## C•NOTES

### Program Listing 2. Graph.BA.

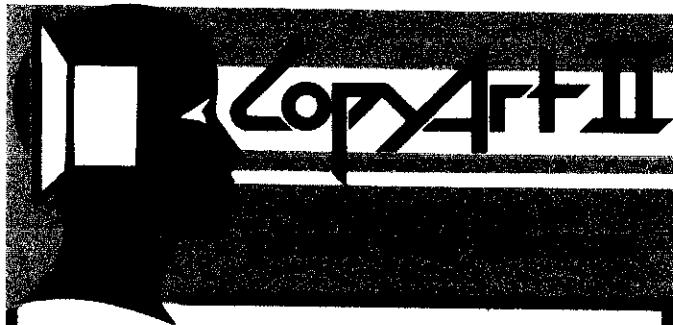
```
300 ' Dow Jones HISTORICAL QUOTES
      Programmed Access
301 ' (C) 1983 by Thomas L. Robb WBL,MN
302 '
305 '      Change History
310 '      9-20-83 Initial release
350 '
351 '
355 GOTO 400
375 '
376 ' MODEM interrupt routine
      w/o file 3 print
377 ' (entry at 380)
380 C$=INPUT$(1,1):RX$="Y":RETURN
385 '
386 ' MODEM interrupt routine
      w/ file 3 print
387 ' (entry at 390)
390 C$=INPUT$(1,1):RX$="Y":PRINT
#3,C$;:RETURN
400 '
410 CLEAR 2048: MAXFILES=3: ' Setup
string space and maximum number of files

415 ON ERROR GOTO 9000: ' General error
paragraph
500 '
501 ' Control paragraph
510 GOSUB 1300: ' Setup system variables

525 GOSUB 1400: ' Dial local TYMNET
number
535 GOSUB 1500: ' Open MODEM files
550 GOSUB 1700: ' Dow Jones signon
575 GOSUB 1800: ' Receive signon message
from Dow Jones
580 IF T%=>3 THEN 7000: ' Input time-out

585 PRINT @165,"Connect on ";DATE$;" at
";TIME$: ' Screen display
590 FOR I%=1 TO 3000: NEXT: ' Delay
while CONNECT message being displayed
600 '
610 GOSUB 3000: ' Display STOCK
background display
620 IF SYMBOL$="" THEN 900: ' Terminate
if no user input entered
622 GOSUB 1600: ON MDM GOSUB 390: ' Open
RAM file with interrupt print
625 T%=0: ' Time-out count
627 PRINT #2,";1";SYMBOL$;" ";YEAR$;"
M": ' Semi-colon (;) is the DJN/R prefix
for historical stock quotes
629 ' 1 char received in C$ and written
to file 3 when RX$="Y". If I-count
reaches 500 (time-out) and T-count (
number of time-outs) <3, then sound tone
and resend original request else abort.
630 I%=1: RX$="N"
635 IF RX$="Y" THEN IF C$<>CHR$(30) THEN
630 ELSE 700
640 I%=1%+1: IF I%<500 THEN 635
645 IF T%<3 THEN T%=T%+1: BEEP: GOTO 627
ELSE 7100
698 '
700 '
705 ON MDM GOSUB 380: CLOSE 3: ' Close
RAM file
```

Listing 2 continued



**THE MOST POWERFUL WORD PROCESSOR AND ALL PURPOSE COMPUTER PROGRAM AVAILABLE FOR THE TRS-80.**

### LOOK AT ALL THESE FEATURES

1. INSERT characters, words, lines, paragraphs or other files.
2. DELETE characters, words, lines, paragraphs.
3. COLUMNS. CopyArt II can be instructed to print your text from one to six columns. Super easy to use! No complicated commands. Great for doing newsletters, magazine layouts etc. NO MORE CUT AND PASTE!
4. SORTING. Sort lines of text by any field. Sorts up to 650 items in less than 7 seconds. Sort indices, table of contents, names, words or whatever in descending or ascending order. Used with CopyArt's math function it is great for small inventories, Receivables, Payables etc.
5. Screen widths from 32-255 characters wide. Screen widths can be changed to allow formatting your text as you want.
6. MATH. Built in MATH function for doing calculations on columns or rows. Used with the SORT command, CopyArt II can do a small inventory of 200-300 items, or keep track of small receivables or payables, general ledgers or home financial reports. Super floating point precision up to 32 digits!
7. \*GRAPHICS. CopyArt has a built in graphics program that allows inserting graphics within your text. Drawings, graphs, illustrations, cartoons etc. may be used within newsletters or company reports. Graphics commands include: Plot between points, Circles, Squares, Fill, Erase, Draw, Move, Pixel cursor controls and more.
8. \*GRAPHIC CHARACTERS. CopyArt has a built in graphics character generator. Used for typesetting large letters from 3 to 25 times normal size! Yes, you can even print characters down the page as well as across. Black on white or white on black.
9. JUSTIFICATION is fully supported. \*Proportional spaced justify is supported.
10. \*SUPER or SUB-SCRIPT.
11. UNDERLINING.
12. BOLDFACTING.
13. \*CHANGE CHARACTER SIZE or PITCH within your document. Character size changes for dot matrix printers with capability. Pitch change for daisy wheel printers with capability.
14. HELP. Help is available for all the commands at the touch of a key while using the word processor. Super for training inexperienced secretaries. Great reminder for experienced people as well. MENU DRIVEN Help for over 45 commands.

**CopyArt II** Continues on  
the next page . . .



COO and Credit Cards  
CALL TOLL FREE to order:  
1-800-528-1149

# SIMUTEK

Computer Products Inc.

163  
SIMUTEK COMPUTER PRODUCTS INC., 4007 E. SPEEDWAY BLVD., TUCSON, AZ 85712, (602) 323-9301  
DEALER, DISTRIBUTOR, & PRINTER/MANUFACTURER INQUIRIES INVITED  
TRS-80 and Scriptart are TM of Radio Shack a Tandy corp.  
\*Indicates printer must have capability to do function.



### MORE REASONS YOU SHOULD CONSIDER MOVING UP TO OUR COMPREHENSIVE PROGRAM...

15. SUPER EASY TO LEARN editing features. Logical key choices.
16. Hyphenation
17. SPELLING checkers like the 74,000 word Scripsit Dictionary work great with CopyArt II.
18. CHAINING. Chain files together to make books or manuals hundreds of pages long.
19. CENTERING.
20. HEADERS and FOOTERS. You can even put graphics within headers for super page layouts.
21. PAGE NUMBERING. Page numbers can appear at the top or bottom of the page.
22. DOS COMMANDS from within the editor. Kill files, check free space or get directories easily.
23. CUSTOMIZED PRINTER driver. Since your printer has features that other printers don't, CopyArt II will be supplied with the printer driver of your choice below. Each printer driver is custom made to provide you with commands for each of your printer's fine capabilities. If you have more than one printer, order other printer drivers for only \$19.95 each. Printer drivers are available for:
  - Radio Shack LP, V, VI, VII, VIII and Daisy Wheel II
  - Epson MX-80, MX-80/FT, MX-100 with or without graftrax.
  - Okidata Microline 80, 82a and 83a
  - NEC 8023
  - Smith Corona Daisy Wheel TP-1
  - Brother Daisy Wheel
  - C-toh Starwriters and Pro-writers all 85 10A, 1550
  - PMC Printer
  - Diablo 620

- OTHERS COMING SOON. Call if you don't see your printer!
24. Unprotected diskette. Unlimited backups can be made.
  25. MAILIST/MAILMERGE INCLUDED. CopyArt II comes with a mailist program that stores over 2,000 names on a MOD III diskette. These names can be sorted by any field and have a special field for your code. You can make PERSONALIZED FORM LETTERS that will take the following codes from the mailist and insert them in your text. FIELDS INCLUDE: Mr. or Ms., Last name, First name, Business name, City, State, up to 9 digit ZIP code and your own special 2 character code. ANY OF THESE fields can be inserted within your form letter wherever you want. You can print form letters or mailing labels to all the people on your list or to specific codes only. CopyArt makes it easy.

**CopyArt II** Continues on  
the next page...

COD and Credit Cards  
CALL TOLL FREE to order:  
1-800-528-1149

# SIMUTEK

Computer Products Inc.

SIMUTEK COMPUTER PRODUCTS INC., 4897 E. SPEEDWAY BLVD., TUCSON, AZ 85712, (602) 323-9391  
DEALER, DISTRIBUTOR, & PRINTER/MANUFACTURER INQUIRIES INVITED  
TRS-80 and Scripsit are TM of Radio Shack a Tandy corp.  
\*Indicates printer must have capability to do function.

Listing 2 continued

```

710 OPEN "QUOTE.DO" FOR INPUT AS 3
725 GOSUB 4000: ' Generate graph
727 CLOSE 3
728 IF FOOTER$<>" THEN IF (VAL(RIGHTS(
TIMES,1)) AND 1)=0 THEN PRINT
@281,SYMBOL$;: ELSE PRINT @281,"
";: ' Flash stock symbol if a footnote
was received from DJN/R.
730 A$=INKEY$: IF A$="" THEN 728 ELSE IF
FOOTER$<>" THEN BEEP: CLS: PRINT
@80,FOOTER$;: GOSUB 2200: GOTO 610 ELSE
610: ' Wait for any key to continue;
then display footnote, if received.
900 'Exit paragraph
905,GOSUB 1900: ' Send disconnect
command
910 GOSUB 2000: ' Capture disconnect
message
915 GOSUB.2100: ' Display elapsed time
and cost (@ $.15/min NON-prime time)
925 '
935 CALL 21179: ' Hangup telephone
945 GOSUB 2200: ' Delay before exit to
allow screen display to be read
947 CALL 23164,0,23366: CALL 27795: ' Re
-establish BASIC'S FUNCTION keys
950 CLOSE: ' Hangup telephone, close all
files
952 KILL "QUOTE.DO": ' Delete scratch
pad RAM file
955 MENU: ' Return to system display
1200 ' Subroutines
1300 '
1301 ' Initialize system variables
1305 KEY OFF: ' Disable FUNCTION keys
1310 SCREEN 0,0: ' Turn labels OFF
1315 SOUND OFF: ' Turn MODEM sound OFF
1320 CLS: ' Clear screen
1322 DIM VOLUME#(12), CLOS(12): '
Dimension STOCK VOLUME and CLOSING PRICE
arrays
1325 RETURN
1400 '
1401 ' Dial TYMNET telephone number
1402 PH$="9,339-5200": M=VARPTR(PH$):
AD=PEEK(M+1)+PEEK(M+2)*256
1403 ' AD points to telephone number
string, PH$.
1405 PRINT @92,"Dialing ";
1410 LINE (116,14) - (170,25),1,B: '
Draw box around dial digits area
1415 CALL 21200: ' Take telephone OFF-
HOOK
1420 CALL 21293,0,AD: ' Dial number
1425 RETURN
1500 '
1501 ' Open MODEM files
1510 OPEN "MDM:7I1D" FOR INPUT AS 1: '
Communications input path
1520 OPEN "MDM:7I1D" FOR OUTPUT AS 2: '
Communications output path
1522 ON MDM GOSUB 380: MDM ON: ' Setup
MODEM interrupt routine
1525 RETURN
1600 '
1601 ' Open scratch pad RAM file
1602 ' This file serves as a record of

```

Listing 2 continued

## C•NOTES

Listing 2 continued

```

the two-way communication dialog between
the program and Dow Jones News/
Retrieval. It is deleted upon program
termination.
1605 OPEN "QUOTE.DO" FOR OUTPUT AS 3
1610 RETURN
1700 '
1701 ' Dow Jones signon
1705 PRINT @173,"Signing on .";: '
Screen display
1710 PRINT #2,"A";: PRINT ".";: '
Response to "please type your terminal
identifier"
1715 PRINT #2,CHR$(18);"DOW1;";: PRINT
".";: ' Response to "please log in: "
1720 PRINT #2,"DJNS": PRINT ".";: '
Response to "WHAT SERVICE PLEASE???"
1730 PRINT #2,"mypassword": PRINT ".";: '
Response to "ENTER PASSWORD"
1735 RETURN
1800 '
1805 ' Look for RECORD SEPARATOR (end of
signon banner)
1810 T%=0: ' Time-out count
1815 I%=1: RX$="N"
1820 IF RX$="Y" THEN IF C$<>CHR$(30)
THEN 1815 ELSE RETURN
1830 I%=I%+1: IF I%<500 THEN 1820
1840 IF T%<3 THEN T%=T%+1: BEEP: PRINT
#2,"//COPYRT": GOTO 1815 ELSE RETURN
1900 '
1901 ' Disconnect
1910 PRINT #2,"DISC"
1920 RETURN
2000 '
2001 ' Capture sign-off message
2010 FOR I%=1 TO 25
2015 LINE INPUT #1,Z$:
2020 IF LEFT$(Z$,7)="LOG ON:" THEN
I%=25
2025 NEXT
2040 RETURN
2100 '
2101 ' Display elapsed time and cost
2105 RATE=.15: ' Per minute rate for
QUOTES during NON-prime time hours;
prime time rate is $.90/minute.
2110 Z%=INSTR(1,Z$,"ON: "): F1%=VAL(MID$(
Z$,Z%+4,2)): F2%=VAL(MID$(Z$,Z%+7,2))
2120 Z%=INSTR(1,Z$,"OFF: "): F3%=VAL(
MID$(Z$,Z%+5,2)): F4%=VAL(MID$(Z$,Z%+8,2
))
2125 IF F3%<F1% THEN F3%=F3%+24
2135 F5%=(F3%*60+F4%)-(F1%*60+F2%)
2140 IF F5%=0 THEN F5%=1
2145 CLS: ' Clear screen
2150 PRINT @201,"Elapsed MINUTES: "
USING "###";F5%;
2160 PRINT "; COST:" USING
"$###.##";F5%*RATE
2180 RETURN
2201 ' Delay for 15 secs to display sign
-off message on screen
2210 FOR I%=1 TO 15000: NEXT
2220 RETURN
3000 '
3001 ' Display screen background

```

Listing 2 continued

# NOW ONLY \$99.00



# CopyArt II

**WILL CHANGE THE WAY  
YOU THINK ABOUT  
YOUR TRS-80!**

26. **SIMPLE CURSOR** commands. Simply use the arrow keys to move your cursor around the text. The screen will scroll both vertically and horizontally. Shift arrows take you to the beginning or end instantly.
  27. **HI-Resolution graphics** supported.
  28. **COMPLETE MARGINS CONTROL.** You tell CopyArt II what margins you desire. You can even change margins within the same text. You may also have parts of your text with 2 columns, some with one etc. It's super easy to use.
  29. **BASIC PROGRAMS** can be edited easily. CopyArt is really useful for inserting graphics within quoted strings to give your programs super animation without the hassle of calculating the CHR\$( of the graphics!
  30. **VISICALC** files can be loaded into CopyArt II to be manipulated easily. Great when you want to accompany your Visicalc reports with written reports, **GRAPHS** and **BOLDFACING** etc. Visicalc reports up to 255 wide can be loaded.
  31. **SPECIAL SCRIPSIT FILE LOADER.** Allows you to load your old Scripsit files without having to save them in ASCII. Copyart will also load Pencil files and other normal ASCII files.
  32. Similar to Scripsit. If you have used Scripsit, you can use Copy Art in minutes.
  33. **CONTROL CODES.** Lets you insert special printer control codes in your text. **CODES** between 0 and 255.
  34. **BLOCK MOVE.** Simple and powerful block move. Lets you move paragraphs or lines of text around easily. No complicated marker settings required.
  35. **FIND/REPLACE/REPEAT.** Lets you find a string of characters and replace them with any other string of characters up to 20,000 times! **WILDCARD** search also supported.
  36. Professional Manual in easy to understand English.
- Copyart II requires a TRS-80 Model I or III, (or PMC-80 or LNW), 48K and 2 disk drives, now works with 4 and 4P.

# CopyArt II THIS PROGRAM DOES IT ALL!

PLEASE SPECIFY which COMPUTER and PRINTER you have when ordering.

Copyart 11 with one printer driver ..... 99.00  
Additional printer drivers ..... 19.95 each

Copyart I owners. Updates available. To registered owners for \$15.00.



**COD and Credit Cards  
CALL TOLL FREE to order:  
1-800-528-1149  
Or send check or money  
order to:**

# SIMUTEK

Computer Products Inc.

SIMUTEK COMPUTER PRODUCTS INC., 4007 E. SPEEDWAY BLVD., TUCSON, AZ 85712, (602) 323-9301  
DEALER, DISTRIBUTOR, & PRINTER/MANUFACTURER INQUIRIES INVITED  
TRS-80 and Scripsit are TM of Radio Shack a Tandy corp.  
\*Indicates printer must have capability to do function.

80 Micro April 1984 - 100



## C•NOTES

Listing 2 continued

```

3002 CLS: ' Clear screen
3015 PRINT @130,"";: LINE INPUT "market
symbol? ";SYMBOL$
3017 IF SYMBOL$="" THEN RETURN: '
Terminate if no user input entered
3020 PRINT @169,"";: LINE INPUT "graph
for year? 19";YEAR$
3022 IF (1900+VAL(YEAR$))<1979 OR VAL(
YEAR$)>VAL(RIGHT$(DATE$,2)) THEN 7200
3025 CLS: LINE (42,0) - (42,55): LINE (
42,55) - (190,55): LINE (190,55) - (
190,0)
3028 FOR I%=1 TO 7: LINE (39,54-8*(I%-1)
) - (39+2,54-8*(I%-1)): LINE (191,54-8*(
I%-1)) - (191+2,54-8*(I%-1)): NEXT
3030 PRINT @281,SYMBOL$;: PRINT
@315,"19";YEAR$;:
3032 PRINT @209,"Retrieving stock data";

3035 RETURN
4000 '
4001 ' GRAPH ROUTINE
4005 ' Convert SYMBOL$ alpha lower case
to upper case
4015 FOR I%=1 TO LEN(SYMBOL$): IF ASC(
MID$(SYMBOL$,I%,1))=>97 AND ASC(MID$(
SYMBOL$,I%,1))<=122 THEN MID$(SYMBOL$,I%
)=CHR$(ASC(MID$(SYMBOL$,I%))-32)
4020 NEXT
4022 ' Position to STOCK line

```

```

4025 IF EOF(3) THEN 7300 ELSE LINE INPUT
#3,Z$
4027 IF LEFT$(Z$,7+LEN(SYMBOL$))<>"STOCK
1"+SYMBOL$ THEN 4025
4050 ' Position to first month line
4055 IF EOF(3) THEN 7300 ELSE LINE INPUT
#3,Z$
4060 IF MID$(Z$,3,3)<>"/"+YEAR$ THEN
4055
4065 FOR I%=1 TO 12: VOLUME#(I%)=0: CLOS
(I%)=0: NEXT
4070 M%=VAL(LEFT$(Z$,2)): M1%=M%
4072 IF LEFT$(Z$,2)=LEFT$(DATE$,2) AND
MID$(Z$,4,2)=RIGHT$(DATE$,2) THEN 4090:
' Don't include current month's data.
4075 GOSUB 4900: ' Extract VOLUME
4077 GOSUB 4950: ' Extract CLOSING PRICE

4078 M%=M%+1
4080 IF EOF(3) THEN 7300 ELSE LINE INPUT
#3,Z$
4085 IF MID$(Z$,3,3)="/"+YEAR$ THEN 4072

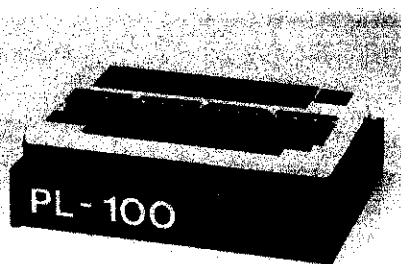
4090 '
4092 FOOTER$=""
4094 IF EOF(3) THEN 4100 ELSE LINE INPUT
#3,Z$
4096 IF LEFT$(Z$,1)="*" THEN FOOTER$=Z$:
GOTO 4100

```

Listing 2 continued

## EXPANSION TO GO

Battery powered interface for the Model 100. Gives Analog and Digital I/O capabilities to your Radio Shack or NEC portable. Whether in the lab or in the field, Data Acquisition and Process Control are now more affordable than ever. Easy to program using BASIC input and output statements.



**Complete PL-100 System \$498 includes:**

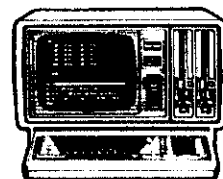
- Expansion Chassis (holds 3 boards)
- Combination I/O board with
  - 16 Channel 8 Bit Analog Input
  - 4 Channel 8 Bit Analog Output
  - 4 Analog Level Sense Inputs
  - 16 Bits Digital Output
  - 16 Bits Digital Input
- Battery Charger / AC Adapter
- 2 Empty expansion slots
- Interface Cables

- Uses rechargeable batteries
- Provides power to the Model 100
- Expansion boards available:
  - High Accuracy 12 Bit Analog I/O
  - 16 Bit Digital I/O
  - Combination Analog & Digital I/O

**ELEXOR ASSOCIATES**

P.O. Box 246  
Morris Plains, N.J. 07950 -209  
(201) 299-1615

## TRS-80 DISCOUNT



**Computers  
at Guaranteed  
Low Prices\***

ATON CP/M FOR 11, 12, 16

EPSON & NEC PRINTERS

DYSAN DISKETTES

HAYES MODEMS

**Desert Sound, Inc.  
of California**

**1-800-835-5247**

Factory Authorized Dealer

TRS-80 is a Reg. Trademark of Tandy Corp.

\*Call for FREE CATALOG  
and Price Guarantee -539  
Calif. Res. Call 619-244-6883

# C•NOTES

Listing 2 continued

```

4098 GOTO 4094
4100 '
4105 PRINT @209,SPACE$(21): ' Clear
Retrieving ... line
4110 ' Find SMALLEST and LARGEST volumes

4115 SVOLUME#=VOLUME#(M1%):
LVOLUME#=VOLUME#(M1%)
4117 FOR I%=M1% TO M%-1
4120 IF SVOLUME#>VOLUME#(I%) THEN
SVOLUME#=VOLUME#(I%)
4125 IF LVOLUME#<VOLUME#(I%) THEN
LVOLUME#=VOLUME#(I%)
4130 NEXT
4135 ' Calculate VOLUME base
4140 G1#=INT(SVOLUME#/1000)*100
4145 ' Calculate VOLUME increment
4150 DELTA=INT((((INT((LVOLUME#+500)/
1000)*100-G1#)/7)+99.5)/100)*100
4200 ' Graph VOLUME labels
4205 GP%=273: FOR I%=1 TO 7
4210 PRINT @GP%,,: PRINT USING
"#####";G1#+DELTA*(I%-1);: PRINT "K";:
GP%=GP%-40
4215 NEXT
4217 G7#=G1#+DELTA*7
4220 PRINT @288,"J F M A M J J A S O N
D";
4250 ' Graph VOLUME
4255 GDELTA=G7#-G1#: PXELS%=INT(GDELTA/

```

```

56+.5)
4265 FOR I%=M1% TO M%-1
4270 Y2%=INT((VOLUME#(I%)/10-G1#)/
PXELS%+.5)
4272 IF Y2%>54 THEN Y2%=54 ELSE IF Y2%<
1 THEN Y2%=1
4275 LINE (50+12*(I%-1),55) - (50+12*(
I%-1),54-Y2%): LINE (51+12*(I%-1),55) -
(51+12*(I%-1),54-Y2%)
4280 NEXT
4300 '
4310 ' Find LOWEST and HIGHEST closing
prices
4315 SCLOS=CLOS(M1%): LCLOS=CLOS(M1%)
4317 FOR I%=M1% TO M%-1
4320 IF SCLOS>CLOS(I%) THEN SCLOS=CLOS(
I%)
4325 IF LCLOS<CLOS(I%) THEN LCLOS=CLOS(
I%)
4330 NEXT
4335 ' Calculate CLOSING PRICE base
4340 G1=INT(SCLOS)
4345 ' Calculate CLOSING PRICE increment

4350 DELTA=INT((INT(LCLOS+.5-G1))/7+.995
)
4400 ' Graph CLOSING PRICE labels
4405 GP%=241: FOR I%=1 TO 7
4410 PRINT @GP%,,: PRINT USING
"$$$###";G1+DELTA*(I%-1);: GP%=GP%-40

```

Listing 2 continued

## Get more use from your Model 100 Let the Elves help!

### ELF-WRITER

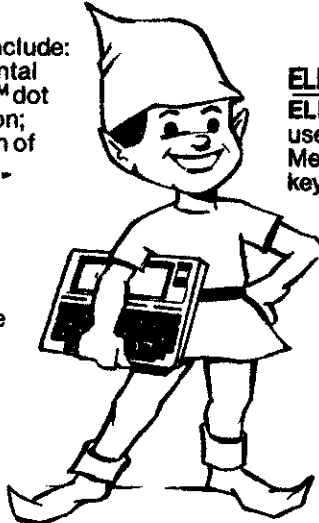
Expand your text processor! Features include:  
— Formatted viewing mode with horizontal scrolling — Formatting uses Wordstar™ dot commands; Variable margins; Pagination; Headings & Footings; Right justification of text; And Much More!!

### ELF-COP

Manage your file space problems!  
Backup all of your files at one time. A backup report documents your backup. ELF-COP will also Copy, Rename, Delete and Report file sizes.

TM - MicroPro International Corp.

**\$39.99 EACH**  
**•SPECIAL OFFER•**  
All 3 for \$90.00  
Price includes shipping.



### ELFPAK = ELF 1 + 1 CALCULATOR + ELFBANKER

ELF 1 + 1 turns your computer into an easy to use calculator. Features: Entry, Total & Memory displays; %, Exponent & Square Root keys; + More!

ELF-BANKER Evaluates your money for the most return. 20 calculations including: Compound interest, Present value, Loan & Mortgage payments, Annuities & Active Savings accounts.



Send the coupon below to:

**Ceres Software Inc.**  
4303 S.W. Chesapeake Ave.  
Portland, OR 97201  
(503) 245-9011

125

Please send me on cassette:

- ELF-WRITER     ELF-COP  
 ELF-PAK  
 Special Offer \$90.00  
 Check for \_\_\_\_\_ enclosed  
 Send C.O.D.

NAME: \_\_\_\_\_

STREET: \_\_\_\_\_

CITY: \_\_\_\_\_ STATE: \_\_\_\_\_ ZIP: \_\_\_\_\_

TELEPHONE: \_\_\_\_\_

Listing 2 continued

```

4415 NEXT
4417 G7=G1+DELTA*7
4450 ' Graph CLOSING PRICE
4455 GDELTA=G7-G1: PXELS=GDELTA/56
4465 FOR I%=M1% TO M%-1
4470 Y2%=INT((CLOS(I%)-G1)/PXELS+.5)
4472 IF Y2%>54 THEN Y2%=54 ELSE IF Y2%<
1 THEN Y2%=1
4475 LINE (48+12*(I%-1),54-Y2%) - (48+
12*(I%-1)+5,54-Y2%)
4480 NEXT
4485 RETURN
4900 '
4905 ' Extract VOLUME data from Z$
string
4910 VOLUME#(M%)=VAL(RIGHT$(Z$,9))
4920 RETURN
4950 '
4955 ' Extract CLOSING PRICE from Z$
string
4960 X$=MID$(Z$,31,10)
4962 IF LEFT$(X$,1)=" " THEN X$=RIGHT$(
X$,LEN(X$)-1): GOTO 4962
4963 IF INSTR(1,X$,"/")=0 THEN CLOS(M%)
=VAL(X$) ELSE SP%=INSTR(1,X$,"/"): IF
SP%<=3 THEN GOSUB 4980 ELSE GOSUB 4985
4965 RETURN
4980 ' Simple fraction
4982 IF SP%=2 THEN CLOS(M%)=VAL(LEFT$(
X$,1))/VAL(MID$(X$,3,2)) ELSE CLOS(M%)
=VAL(LEFT$(X$,2))/VAL(MID$(X$,4,2))
4983 RETURN
4985 ' Mixed number
4987 SP%=INSTR(1,X$," "): CLOS(0)=VAL(
LEFT$(X$,SP%-1)): X$=MID$(X$,SP%+1,10-
SP%): SP%=INSTR(1,X$,"/"): GOSUB 4980:
CLOS(M%)=CLOS(M%)+CLOS(0)
4988 RETURN
7000 '
7005 BEEP: CLS: PRINT @122,"Input
time-out receiving DJN/R Signon"
7010 GOTO 925
7100 '
7105 BEEP: CLS: PRINT @123,"Input
time-out receiving DJN/R data"
7110 GOTO 925
7200 '
7205 BEEP: CLS: PRINT @127,"Historical q
uotes exist for";: PRINT @168," years 19
79 to 19"; RIGHT$(DATE$,2);" only"
7210 GOSUB 2200: GOTO 610
7300 '
7305 BEEP: CLS: PRINT @123,"No historica
l stock data available";: PRINT @172,"on
";SYMBOL$;" for 19";YEAR$
7310 GOSUB 2200: CLOSE 3: GOTO 610
9000 '
9001 ' General ERROR paragraph
9025 IF ERR=52 AND ERL=952 THEN RESUME
NEXT
9050 P%=0: BEEP: CLS: ' Clear screen
9060 PRINT @41,"An application program
error, CODE ";ERR;",";
9065 PRINT " has occurred at BASIC
program"
9070 PRINT " line NUMBER ";ERL;". "
9075 GOTO 925: ' Exit

```

End

## North by Northwest

by Smith Harris

If you're a traveler, you may often wonder how far your current position is from your destination, or in what direction your destination lies.

If you're working with distances of less than a few hundred miles, you can answer these questions fairly accurately with a large-scale map. However, when the distance between your destination and current position is more than a few hundred miles, the distortion caused by the curvature of the earth on the flat surface of the map becomes significant.

I wrote an 8K RAM program, *Navigate.BA*, that accurately determines the distance between any two locations on the surface of the earth, as well as the compass bearing from the first to the second location (see Program Listing 3).

It does so by solving the spherical triangle whose apexes are at your current location, the north pole, and your destination. Spherical triangles, as the name implies, are triangles drawn on the surface of a sphere. They have three angles, like the more familiar plane triangles, but the resemblance stops there. Their sides are segments of great circles and their geometry is more complex than that for plane triangles.

## The Program

To use *Navigate*, enter the names of the two locations and their respective latitudes and longitudes in degrees and decimal fractions of a degree, prefixing south longitudes and east

Program Listing 3. *Navigate.BA*.

```

10 ' NAVIGATE
20 ' BY
30 ' SMITH HARRIS
40 ' ROUTE 4, BOX 59
50 ' GRAY, GEORGIA 31032
60 '
70 CLS
75 PRINT:PRINT:PRINT
80 INPUT "STARTING LOCATION";A$
90 INPUT "DESTINATION";B$
100 CLS:PRINT "LATITUDE OF ";A$
105 PRINT"(XX.XX DEGREES; - IF SOUTH)
";:INPUT N
110 PRINT "LONGITUDE OF ";A$
115 PRINT"(XX.XX DEGREES; - IF EAST)
";:INPUT O
120 PRINT "LATITUDE OF ";B$
125 PRINT"(XX.XX DEGREES; - IF SOUTH)
";:INPUT P
130 PRINT "LONGITUDE OF ";B$
135 PRINT"(XX.XX DEGREES; - IF EAST)
";:INPUT Q
140 RD=57.2958:DR=.0174533 'RADIANS TO
DEGREES - DEGREES TO RADIANS
150 N=DR*N:O=DR*O:P=DR*P:Q=DR*
Q:X$="#####.#":Y$="#####.#"
160 AA=1.5708-P
170 BB=1.5708-N
180 C=O-Q
190 IF C=0 THEN 410 'IF NORTH-SOUTH OR
EAST-WEST
200 IF N=P THEN 430 'SKIP SPHERICAL
TRIANGLE.
210 E=(BB-AA)/2:F=(BB+AA)/2:G=C/2

```

Listing 3 continued

latitudes with a minus sign.

The computer then solves the triangles and displays the distance between the two locations in international nautical (airline) miles, statute miles, and kilometers. Navigate also indicates the compass bearing from the first to the second location on a 360-degree basis (zero degrees is north, 90 degrees east, 180 degrees south, and 270 degrees west).

The program takes the mean circumference of the earth as 24,860.51 miles, making one degree equivalent to 69.05698 statute miles. It converts statute miles to international nautical (airline) miles and kilometers using conversion factors given in NASA publication SP-7021, *The International System of Units—Physical Constants and Conversion Factors* (1 nautical mile equals 1.150778 statute miles and 1 kilometer equals 0.621371 statute miles).

I chose to express latitudes and longitudes in degrees and decimal fractions of a degree because I find it easier to estimate locations on a map in this manner rather than using minutes and seconds. If you prefer to use minutes and seconds, convert these quantities to decimal fractions of a degree by dividing minutes by 60 and seconds by 3,600. The computer does the conversion for you if you change line 100 to:

```
100 PRINT "LATITUDE OF ";AS;"(DD,MM,SS)(-IF SOUTH)";
INPUT N,M,S
```

and insert line 105 as follows:

```
105 N=N+M/60+S/3600
```

You should similarly modify lines 110 through 130.

A word of warning is in order. Don't try to fly from New York to San Francisco using the compass bearing obtained in

the program. Although it's correct, you must constantly make course corrections to follow a great circle. Otherwise, you'll never get to your destination; instead, you'll follow a path called a loxodrome around the globe in ever-decreasing spirals, approaching but never reaching one of the poles. (You can find out more about great circles, loxodromes, and navigation in general in an encyclopedia.)

Navigators involved in great-circle sailing can easily adapt the program for a pocket computer. It saves a lot of tedious calculations with log tables. ■

*Smith Harris can be reached at Route 4, Box 59, Gray, GA 31032.*

## Graphic Results

by Emmett Carmody

It's often useful to see a graphic representation of how a change in quantitative variables produces different results. I've written two Basic programs that produce illustrative graphs using results from problem-solving routines.

Program Listing 4, *Pie.BA*, displays a pie chart relating three variables to an available total quantity. Program Listing 5, *Bar.BA*, lets you solve an equation for X in terms of four variables or constants, A, B, C, and D, and presents a bar chart relating X to a desired maximum. You can change any of the variables to try and get closer to the maximum value.

### The Pie Chart

Lines 20-60 in Listing 4 draw a circle on the right side of the screen using the PSET command. Quantity A in these lines is the angle in radians, whose value increases in steps of 0.1 radians from zero to 6.28 (or two times pi). Quantity R is the radius, set at 30 pixels.

Line 70 asks you for the total quantity of whatever your concern may be (budget dollars, man-hours, supplies, people surveyed, or so on). Input items A, B, and C in line 80. If the sum of A, B, and C exceeds the total, line 90 alerts you to the fact and gives you another chance to input values. If you take up too many lines entering values, the circle is pushed off the screen.

Statements 100-250 draw the pie divisions. Line 110 makes the initial cut, and line 130 converts each item to degrees and percentage. Line 140 computes the angle to place the label of the pie section; line 150 computes the angle of the cut.

Lines 160 and 170 find the point X2Y2 on the circumference of the circle corresponding to the angle figured in 150. Lines 180 and 190 do the same for the point X3Y3—the point where you place the sector label.

A Print@ statement in 240 puts the identifying letter on the screen as close as possible to X3Y3. Lines 200 through 220 find the appropriate character block number (each character block consists of a 6- by 8-pixel rectangle).

Line 230 draws each pie cut; line 240 prints each label. Lines 260 and 270 print out the percentages figured in line 130 and what is left over (if anything). Line 280 keeps everything on the screen. To solve another problem, break and run again.

Listing 3 continued

```
'LINES 210 - 310
220 X=cos(E)/(cos(F)*tan(G)) 'SOLVE
230 Y=sin(E)/(sin(F)*tan(G)) 'SPHERICAL
240 XX=ATN(X)*2 'TRIANGLE
250 YY=ATN(Y)*2
260 B=(XX+YY)/2
270 A=XX-B
280 L=(B+A)/2:M=(B-A)/2
290 ZZ=(tan(E)*sin(L))/sin(M)
300 CC=2*ATN(ZZ)
310 A=A*RD:IF A<0 AND C>0 OR A>0 AND C<0
THEN A=180+A ELSE IF A<0 AND C<0 THEN
A=360+A
320 D=CC*60.009*RD:IF D<0 THEN D=-D
330 CLS:PRINT "DISTANCE FROM ";AS;" TO
";B$;":"
340 PRINT USING X$;D;:PRINT " NAUTICAL
MILES"
350 PRINT USING X$;D*1.15078;:PRINT "
STATUTE MILES"
360 PRINT USING X$;D*1.852;:PRINT "
KILOMETERS"
370 PRINT "COMPASS BEARING FROM "
375 PRINTAS;" TO ";B$;"::PRINT USING
Y$;A;:PRINT " DEGREES"
380 INPUT "DO YOU WANT TO CONTINUE(Y/N)
";C$
390 IF LEFT$(C$,1)="Y" THEN 70
400 END
410 CC=AA-BB:IF N>P THEN A=180 ELSE A=0
420 GOTO 320
430 CC=C:IF O<Q THEN A=270 ELSE A=90
440 GOTO 320
```

End



## C•NOTES

If you need to change the number of pieces in the pie, change lines 80, 90, 120, 260, and 270. If you make too many slices, the labels will print over each other. Also be careful not to enter small quantities adjacent to one another when inputting data. Quantities representing 5 percent or more of the whole should print adjacently without problems.

If you want to get really fancy, figure out how to change the label radius (the constant, 21, in lines 180 and 190) so that labels won't over-print when adjacent in small sectors.

### Bar Chart Program

Listing 5 presents a visual indication to accompany the calculated result. Start with an engineering, financial, or mathematical equation. In lines 20 through 50 enter a needed or estimated maximum value, and the value of the variables and constants on which the unknown depends, and change lines 40 and 50 to contain the equation you want to solve.

The listing presents an example with X, the unknown, and a function of four other quantities—A, B, C, and D—for which you input values as the program runs.

Line 60 rounds the answer to two places. Lines 80-160 draw a linear scale at the top of the display; the maximum value entered at line 20 prints at the right, zero at the left. The loop in lines 130-150 computes and prints intermediate scale values.

#### Program Listing 4. Pie.BA.

```

1 'Pie Chart Program - PIE.BA
2 'Given a total and 3 quantities whose
sum is less than the total, this
program will graph a pie chart and print
percentages.
3 'By E.J. Carmody
10 CLS:PRINT"PIE CHART PROGRAM"
20 PSET (197,32)
30 FOR A=0 TO 6.28 STEP .1
40 R=30
50 PSET (197+R*COS(A),32+R*SIN(A))
60 NEXT A
70 INPUT"TOTAL QUANTITY";T
80 INPUT"ITEMS A,B,C";C(1),C(2),C(3)
90 IF T<C(1)+C(2)+C(3) THEN
PRINT@120,"OVER TOTAL":GOTO 80
100 RAD=0
110 LINE (197,32)-(227,32)
120 FOR N=1 TO 3
130 D=C(N)/T*360:
PC(N)=INT(C(N)/T*100+.5)
140 R1=RAD+D/360*3.14
150 RAD=RAD+D/360*6.28
160 X2=197+R*COS(RAD)
170 Y2=32-R*SIN(RAD)
180 X3=197+21*COS(R1)
190 Y3=32-21*SIN(R1)
200 X4=INT(X3/6)
210 Y4=INT(Y3/8)
220 P1=Y4*40+X4
230 LINE(197,32)-(X2,Y2)
240 PRINT@P1,CHR$(96+N)
250 NEXT N
260 PRINT@200,"A=";PC(1);"% B=";PC(2);"%
C=";PC(3);"%
270 PRINT@240,"LEFT: ";100-PC(1)-PC(2)-
PC(3);"%
280 GOTO 280

```

End

The ASCII 239 graphic, a solid block, makes up the bar. The value of X indicates the number of blocks used (170-200). When X exceeds the maximum, lines 180 and 250 substitute ASCII 234 to form a dashed bar and alert you.

All values and the equation itself appear below the bar graph in 210 and 220. The program then prompts you to press the space bar to change input. It also asks which variable you want to change. You can select A, B, C, or D and provide new input (lines 260-380). Operation then transfers back to line 40 and the program computes and graphs another solution.

You can easily adapt each of the programs for other applications using pie and bar charting. ■

*Contact Emmett Carmody at 2327 Birch Hill Drive,  
Florissant, MO 63033.*

#### Program Listing 5. Bar.BA.

```

1 'Bar chart program - BAR.BA
2 'A user supplied formula is solved
for X as a function of A,B,C, and D and
graphed as a bar chart.
3 'Values may be changed and the
process repeated.
4 'By E.J. Carmody
10 CLEAR 100:CLS
20 INPUT"ENTER MAXIMUM VALUE";M
30 INPUT"A,B,C,D=";A,B,C,D
40 X=5*A^2/SQR(B+C^3)+20*D
50 X$="X=5*A^2/SQR(B+C^3)+20*D"
60 X=INT(X*100+.5)/100
70 CLS:L=M
80 LINE(0,22)-(239,22)
90 FOR I=119 TO 87 STEP -8
100 PRINT@I,CHR$(245):NEXT I
110 PRINT@80,CHR$(245)
120 PRINT@156,L
130 FOR I=149 TO 125 STEP -8
140 L=L-M/5
150 PRINT@I,L:NEXTI
160 PRINT@120,"0"
170 P=239
180 IF X>M THEN 250
190 N=40*X/M
200 PRINT@40,STRING$(N,P)
210 PRINT@160,"X=";X;"A=";A;"B=";
";B;"C=";C;"D=";D
220 PRINT@200,X$
230 PRINT@240,"PRESS SPACEBAR TO CHANGE
INPUT
240 K$=INKEY$:IF K$=CHR$(32)
THEN 260 ELSE 240
250 N=40:P=234:GOTO 200
260 INPUT"WHICH VARIABLE";Y$
270 IF Y$="A" THEN 320
280 IF Y$="B" THEN 340
290 IF Y$="C" THEN 360
300 IF Y$="D" THEN 380
310 PRINT"VARIABLE INCORRECT":GOTO 260
320 INPUT"A=";A
330 GOTO 40
340 INPUT"B=";B
350 GOTO 40
360 INPUT"C=";C
370 GOTO 40
380 INPUT"D=";D:GOTO 40

```

End

## File Expansion

Line 10 of my BKUPDO program ("Backing Up the 100," February 1984, p. 190) has a problem. As it appears in the listing, the MAXFILES=2 statement nullifies the preceding DIM statement. To allow more than 10 files in the backup list, retype line 10 as:

```
10 CLEAR2E3 :MAXFILES=2 :DIMF$(19)
```

*Bryan R. Leipper  
714 Terre Court  
Reno, NV 89506*

You can overcome this problem if DJNS lets you ask for an unlimited number of quotes at a time; you could forget the M and the series of equals signs after each batch and type one AM: at the end.

I'm waiting to hear from DJNS on whether this is a technical limitation. I suspect, however, that the problem is directly related to how much can fit on the Model 100 screen.

*Alf L. Erickson  
509 Flamingo Drive  
Fort Lauderdale, FL 33301*

## Log-On Sequences

Here's a list of auto log-on sequences for the Model 100 (see the Table). It covers the Dow Jones News Service (DJNS), The Source, CompuServe, and the Official Airline Guide (OAG). I developed them for Tymnet, Telenet, Uninet, and for direct connection to CompuServe. When you use the sequences, you should substitute your local access numbers in place of the Fort Lauderdale numbers in the Table.

Since the sequences take up a small amount of memory, I built all 11 into my ADRS.DO file. Now if one line is busy, I can quickly dial up another number.

Also, you can incorporate an unlimited number of stock quote requests into the auto log-on sequence by typing in your list of five stock symbols followed by AM (see the Table). Then, before you type in the >: characters, type a comma, six or seven equals signs, five more stock symbols separated by spaces, and an AM.

When you have finished your queries, press the right-arrow and colon keys. For example, to get a quote for General Motors, type ?A,GMAM before pressing the last right-arrow key.

This procedure gives you a lot of automatic stock quote requests using a minimum of memory. However, since you only type <: after the last batch of requests, only the last batch comes to your computer in the terminal mode. The earlier batches hit the Model 100 while it is in the entry mode, so you can view them as they cross the screen, but you can't send them to the printer or download them.

## To Err Is Confusing

Most TRS-80 computers identify errors in Basic programs in plain English terminology. The Model 100, however, displays only an error code, along with the line number in which the error occurs. To find out what the error is, you have to look up the code's definition.

I wrote a program called Codes that solves this problem (see the Program Listing). Type in your own programs using lines 1-5998. Line 0 traps errors by switching the program flow to line 5999 and defining the error.

Codes catches all coded errors, but not undefined errors. You might want to include owner's manual page-number references to the errors. After you have the program running, you can delete lines 0 and 5999-7000 before printing or saving your program.

*Robert L. Green  
P.O. Box 419  
Clarkston, GA 30021*

## Super Text

I recently discovered how to set a left margin when printing on my Epson FX-80 with the Model 100's text processor. To generate the escape code for all Epson printers, type CHR\$(155). In Basic type CHR\$(27).

On the Epson, you generate a left margin by typing CHR\$(27);"1";CHR\$(n), where n denotes the number of spaces you want to indent. When using the ASCII character

|                                    |                                                                                   |
|------------------------------------|-----------------------------------------------------------------------------------|
| CompuServe local direct connection | :7723240 < = AC?U(user ID number)AM?P(password)AM > :                             |
| Dow Jones Tymnet                   | :4673807 < = = A?pDOW1; ;?WDJNSAM?P(password)AM > :                               |
| CompuServe Tymnet                  | :4673807 < = = A?pCPSAM?U(user ID number)AM?P(password)AM > :                     |
| OAG Tymnet                         | :4673807 < = = A?pOAGAM?(account#);(password)AM > :                               |
| Source Tymnet                      | :4673807 < = = A?pSOURCE10;PRIM;AM? + AMID(user ID number) (password)AM > :       |
| Dow Jones Telenet                  | :7644505 < = AM = AM?TD1AM? @C 60942AM?WDJNSAM?D(password)AM > :                  |
| Source Telenet                     | :7644505 < = AM = AM?TAM = AM? @C 30124AM = = ID(user ID number) (password)AM > : |
| CompuServe Telenet                 | :7644505 < = AM = AM?TD1AM? @C 202202AM?U(user ID number)AM?P(password)AM > :     |
| Source Uninet                      | :4676504 < = AM.AMS10AM?UID(user ID number) (password)AM > :                      |
| Dow Jones Uninet                   | :4676504 < = AM.AMPROF3AMDOWAM?*DJNSAM?E(password)AM > :                          |
| OAG Uninet                         | :4676504 < = AM.AMOAGAM?P(account number);(password)AM > :                        |

*Table. Auto log-on sequences for the Model 100.*

# BIG SAVINGS ON...

## TRS-80 FORMS & SUPPLIES

- ✓ Invoices
- ✓ Checks
- ✓ Statements
- ✓ Stationery
- ✓ Diskettes
- ✓ Ribbons

## COLWELL COMPUTER FORMS & SUPPLY CATALOG

We're so sure you'll save money, we'll give you a \$5.00 check just to prove it. That's right! Colwell has been providing top quality forms and supplies at substantial savings since 1927 and every order comes with a 100% "No Questions Asked" Guarantee...you must be satisfied or your money back.

To get your \$5.00 check good towards your first order and free catalog, fill out and return this coupon or call the toll-free number shown.

Call Toll-Free  
**800-248-7000**  
(In Illinois 800-233-7777)

Send for  
**\$5.00 check**  
and **FREE**  
catalog  
today!



202

Colwell Systems, Inc.  
201 Kenyon Road  
Champaign, Illinois 61820



294

Colwell Systems, Inc.  
201 Kenyon Road  
Champaign, Illinois 61820

|                                          |                                     |                                             |
|------------------------------------------|-------------------------------------|---------------------------------------------|
| NAME/TITLE                               |                                     | PHONE                                       |
| COMPANY                                  |                                     |                                             |
| STREET                                   |                                     |                                             |
| CITY, STATE, ZIP                         |                                     |                                             |
| SOFTWARE BRAND                           | PACKAGE #                           | <input type="checkbox"/> DO OWN PROGRAMMING |
| COMPUTER BRAND                           | MODEL                               | PLAN TO PURCHASE WITHIN                     |
| YOUR LINE OF BUSINESS                    |                                     | NUMBER OF EMPLOYEES                         |
| I MOST OFTEN USE MY COMPUTER FOR:        |                                     |                                             |
| <input type="checkbox"/> WORD PROCESSING | <input type="checkbox"/> ACCOUNTING | <input type="checkbox"/> OTHER              |

## RAM FILES

code tables from page 211 of the Model 100 manual, CHR\$(155) is (GRPH)k. This gives you a left-arrow control character.

Next, type a lowercase L, and press control-M to create CHR\$(n). You then have a 13-space left margin, the minimum you can set since the control characters A-L leave no recordable character on the screen.

For a margin of 13, Type (GRPH)k 1 and press control-M at the top of your text. The computer displays a left arrow, a lowercase L and a carriage return symbol. These codes will not appear on your printout or interfere with your text formatter code.

You can set elite spacing on the Epson RS-80 and FX-80 by pressing escape-M. Generate the escape function by typing (GRPH)k followed by an uppercase M. The printer now prints in the elite mode. You can also use this technique to print double-width, condensed, emphasized, italicized, superscripted, and subscripted type.

Lorne Nicolson  
RR#1  
Nelson, B.C.  
V1L 5P4 Canada

*Program Listing. Codes program of error definitions.*

```

0 ON ERROR GOTO 5999
5999 BEEP:CLS:ON ERR GOTO
6001,6002,6003,6004,6005,6006,6007,
6008,6009,6010,6011,6012,6013,6014,
6015,6016,6017,6018,6019,6020,6022,
6051,6052,6053,6054,6055,6056,6057,6058
6000 PRINT"UNDEFINED ERROR":END
6001 PRINT"NEXT WITHOUT FOR":END
6002 PRINT"SYNTAX ERROR":END
6003 PRINT"RETURN WITHOUT GOSUB":END
6004 PRINT"OUR OF DATA":END
6005 PRINT"ILLEGAL FUNCTION CALL":END
6006 PRINT"OVERFLOW":END
6007 PRINT"OUT OF MEMORY":END
6008 PRINT"UNDEFINED LINE":END
6009 PRINT"BAD SUBSCRIPT":END
6010 PRINT"DOUBLY DIMENSIONED
ARRAY":END
6011 PRINT"DIVISION BY ZERO":END
6012 PRINT"ILLEGAL DIRECT":END
6013 PRINT"TYPE MISMATCH":END
6014 PRINT"OUT OF STRING SPACE":END
6015 PRINT"STRING TOO LONG":END
6016 PRINT"STRING FORMULA TOO
COMPLEX":END
6017 PRINT"CAN'T CONTINUE":END
6018 PRINT"I/O ERROR":END
6019 PRINT"NO RESUME":END
6020 PRINT"RESUME WITHOUT ERROR":END
6022 PRINT"MISSING OPERAND":END
6051 PRINT"BAD FILE NUMBER":END
6052 PRINT"FILE NOT FOUND":END
6053 PRINT"ALREADY OPEN":END
6054 PRINT"INPUT PAST END OF FILE":END
6055 PRINT"BAD FILE NAME":END
6056 PRINT"DIRECT STATEMENT IN
FILE":END
6057 PRINT"UNDEFINED ERROR":END
6058 PRINT"FILE NOT OPEN":END
7000 RESUME
    
```

End

C-NOTE FOR MAY 1984



## A Real Sketch

by John and Aileen Corman

Sketchpad, a 24K RAM Basic program (with changes for 16K), can turn your Model 100 into a deluxe etch-a-sketch that lets you create designs on the screen and then print them out.

The program overcomes two deficiencies of the 100's graphics capabilities. Since it has no Point command (as in Model III Basic), you can't tell if a particular screen pixel is set or not. Because of this, you can't transfer a design from the screen to a printer—even if it has dot-addressable graphics capability.

Program Listing 1, Sketchpad, lets you draw designs with lines, boxes, circles, or individual dots, and repeat parts of the sketch to create interesting effects. It lets you include text in your designs as well as save and recall sketches from RAM or tape files.

We use machine-language subroutines to provide near instant response in most sketching functions. The heart of the system is the 1,920-byte bit-map area that we use to record the on/off status of each of the 15,360 dots in the display. Each bit in this area corresponds to one dot on the screen. By relaying each dot's status, the bit map makes it possible for you to transfer an entire screen to a printer, bit by bit.

### Some Ground Rules

When you run Sketchpad, the title screen gives you the option of going directly to its help menu or into the Sketch routine. If you want to draw, you must instruct the program to either set up for a new sketch or continue with the previous sketch in the bit map.

Pressing the H key at any time while sketching produces a reference listing of the functions and the command key for each (see Table 1). If this table is of sufficient help, press

|          |          |            |            |             |
|----------|----------|------------|------------|-------------|
| (A)xes   | (G)rid   | (L)ine     | (Q)uit     | (W)here     |
| (B)ox    | (H)elp   | (M)ove     | (R)epet    | (X)Y-set    |
| (C)ircle | (I)ntput | (N)egative | (S)napshot | (1)black    |
| (D)raw   | (J)ump   | (O)utput   | (T)ext     | (0)white    |
| (F)ill   | (K)ill   | (P)oint    | (V)iew     | ↑ - ↓cursor |

Table 1. Sketchpad function keys.

### The Key Box

The programs in "A Real Sketch" and "Getting Personal" run in 24K and 8K RAM, respectively. Both require printers.

Program Listing 1. Sketchpad.

```

10 '*** SKETCHPAD+ ***
1 CLEAR100,MAXRAM-2961:LOADM"SKSUBR":GOS
UB3000:GOSUB6000:GOTO20
2 SC=CH:IFCH>DETHEN9ELSEIFCH=DETHENCH=N4
:GOTO8
3 IFCH<CZTHEN9ELSEIFCH=CZTHENCH=N3:GOTO8
4 IFCH>CRTHEN9ELSEIFCH=CRTHENCH=N5:GOTO8
5 IFCH>N2THEN9ELSEIFCH=N0THENCH=N1:GOTO9
8 PRINT#1,C2$;
9 PRINT#1,CHR$(CH);:CH=SC:RETURN
20 GOSUB50
30 K$="":IFK>96THENK=K-32
32 K=K-64:IFK<>6THENB=0:C=0
33 IFK=-15THENK=25
34 IFK=-16THENK=26
35 IFK<1THEN20
40 ONKGOSUB1900,200,300,400,94,600,2600,
5000,700,1000,1400,1200,94,1800,100,1600
,1700,2400,900,2000,94,2200,2300,2100,50
0,1100:GOTO20
50 IFK$<>" "THENRETURNELSEPSET(X,Y)
52 FORI=1TO45:K$=INKEY$:IFK$<>" "THENI=45
53 NEXTI
54 PRESET(X,Y):IFK$<>" "THEN60
56 FORI=1TO45:K$=INKEY$:IFK$<>" "THENI=45
57 NEXTI
58 IFK$=" "THEN50
60 K=ASC(K$):GOSUB99:IFPEEK(S!+HL)ANDATH
ENPSET(X,Y)
62 IFK=1THENX=X-JX:GOTO90
64 IFK=2THENY=Y+JY:GOTO90
66 IFK=6THENX=X+JX:GOTO90
68 IFK=17THENX=0:GOTO90
70 IFK=18THENX=239:GOTO90
72 IFK=20THENY=Y-JY:GOTO90
74 IFK=23THENY=0:GOTO90
76 IFK=26THENY=63:GOTO90
78 IFK=28THENX=X+1:GOTO90
80 IFK=29THENX=X-1:GOTO90
82 IFK=30THENY=Y-1:GOTO90
84 IFK=31THENY=Y+1
90 IFX<0THENX=0ELSEIFX>239THENX=239
92 IFY<0THENY=0ELSEIFY>63THENY=63
94 RETURN
95 K$=INKEY$:IFK$=" "THEN95ELSEK=ASC(K$):
RETURN
99 A=X*64+Y:HL=A\8:A=2^(AMOD8):RETURN
100 CLS:PRINT:PRINT"Output to (R)AM, (C)
AS, (P)rinter,":PRINTTAB(7)"or (N)o outp
ut?"
102 GOSUB95
104 IFK$<>"N"ANDK$<>"n"THEN108
106 K$="":GOSUB2500:RETURN
108 IFK$<>"R"ANDK$<>"r"THEN150
110 GOSUB180:GOSUB112:GOTO118
    
```

Listing 1 continued

## C•NOTES

Listing 1 continued

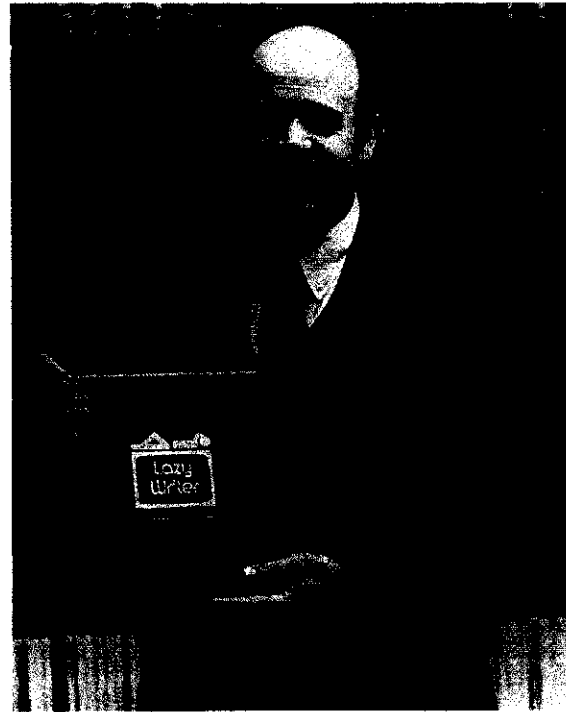
```

112 OPENFI$FOROUTPUTAS1:PRINT:PRINT"Doin
g sketch output, please wait..."
114 M!=S!:L!=V!-N!::RETURN
118 IFM!>L! THEN148ELSECH=PEEK(M!):CN=N1
120 M!=M!+N!::IFPEEK(M!)=CHANDM!<V!THENC
N=CN+N1:GOTO120
122 IFCN>N5THEN130
124 FORI=N1TOCN
126 GOSUB2:NEXTI
128 GOTO118
130 IFCN>FETHENPRINT#1,C2$;CF$;:GOSUB2:C
N=CN-FF:GOTO130
132 IFCN>DETHENPRINT#1,C2$;CHR$(CN)::GOS
UB2:GOTO118
134 IFCN=DETHENGOSUB2:PRINT#1,C2$;CD$;:G
OSUB2:GOTO118
135 IFCN=CZTHENGOSUB2:PRINT#1,C2$;CZ$;:G
OSUB2:GOTO118
136 IFCN=CRTHENGOSUB2:PRINT#1,C2$;CR$;:G
OSUB2:GOTO118
137 IFCN>N5THENPRINT#1,C2$;CHR$(CN)::GOS
UB2:GOTO118ELSEIFCN=N0THEN118ELSE124
148 CLOSE:CALLV!:K$="":RETURN
150 IFK$<>"C"ANDK$<>"c"THEN170
152 GOSUB180:GOSUB154:GOTO158
154 FI$="CAS:"+FI$:PRINT:PRINT"Press ENT
ER when recorder is ready..."
156 GOSUB95:RETURN
158 GOSUB112:FORMI=S!TOLL!
160 PRINT#1,PEEK(M!):NEXTM!
162 GOTO148
170 IFK$<>"P"ANDK$<>"p"THEN102ELSE800
180 PRINT:LINEINPUT"Enter file name: ";F
I$
182 RETURN
200 IFX=PXANDY=PYTHENRETURN
202 LINE(PX,PY)-(X,Y),CO,B
204 HL!=PX*256+PY
206 POKEY2!,PX:POKEY2!,Y:CALLL!,CO,HL!
210 POKEY2!,X:POKEY2!,PY:CALLL!,CO,HL!
212 HL!=X*256+Y
214 POKEY2!,PX:POKEY2!,Y:CALLL!,CO,HL!
218 POKEY2!,X:POKEY2!,PY:CALLL!,CO,HL!
220 B=1:RETURN
300 R2!=(PX-X)^2+(PY-Y)^2:LC=FIX(SQR(R2!
/2)+.5):XX=X:YY=Y
302 FORCY=PYTOPY-LCSTEP-1
304 CX!=PX-SQR(R2!-(PY-CY)^2):CX=FIX(CX!
+.5*SGN(CX!))
306 GOSUB350
308 NEXTCY
310 FORCX=CXTOPX
312 CY!=PY-SQR(R2!-(PX-CX)^2):CY=FIX(CY!
+.5*SGN(CY!))
314 GOSUB350
316 NEXTCX
318 X=XX:Y=YY:IFP=1THENP=0:GOSUB1650
320 C=1:RETURN
350 IFCY<0AND(PY+PY-CY)>63THENCX=PX:CY=P
Y-LC:RETURN
351 X=CX:Y=CY:IFX>0ANDX<240ANDY>=0ANDY<
64THENIFF=0THENGOSUB380ELSEGOSUB390ELSEI
FF=1THENGOSUB390
352 LY=PY+PY-CY:Y=LY:X=CX:IFY<64ANDX>=0T
HENIFF=0THENGOSUB380ELSEGOSUB390ELSEIFF=
1THENGOSUB390
354 RX=PX+PX-CX:X=RX:Y=CY:IFX<240ANDY>=0
THENIFF=0THENGOSUB380ELSEGOSUB390ELSEIFF
=1THENGOSUB390

```

Listing 1 continued

## The Johnson Law Office Depends on Lazy Writer



"When I opened my law office I needed word processing, but my resources were limited. A local computer store recommended Lazy Writer. I purchased a Radio Shack Model III and Lazy Writer, took them home, and within a day felt very comfortable with them. Lazy Writer was easier to use than the dedicated word processors at my old law firm. Now my law practice has grown and we have four Model III's and a Model IV. We recently bought the new Model IV upgrade for Lazy Writer so we can have the 80 x 24 screen display. Every attorney in the office as well as the law clerks and all of the secretaries are capable of using Lazy Writer. We spend 95 percent of our computer time using Lazy Writer.

"My practice involves a lot of estate work. The average set of estate planning documents is about 50 pages. With the text blocks I've developed, using Lazy Writer and Lazy Doc, it takes three hours for a secretary to complete the forms for a client. At my old law firm, it took six hours on a mag card word processor and five hours on a Wang to do the same thing."

Robert T. Johnson  
Attorney at Law

Lazy Writer for TRS-80 Model I/III/IV ..... \$175.00  
Lazy Doc Document Maker ..... \$ 59.95

AlphaBit Communications, Inc.  
13349 Michigan Ave.  
Dearborn, Michigan 48126  
(313) 581-2896  
MasterCard/Visa Accepted



476

Listing 1 continued

```

355 Y=LY:X=RX:IFY<64ANDX<240THENIFF=0THE
NGOSUB380ELSEGOSUB390ELSEIFF=1THENGOSUB3
90
356 RETURN
380 GOSUB99:IFCO=1THENPSET(X,Y):CALLO!,A
,HLSEPRESET(X,Y):CALLF!,A,HL
382 RETURN
390 IFY<0ORY>63THENRETURN
391 IFX<0THENX=0ELSEIFX>239THENX=239
392 LINE(X,Y)-(PX,Y),CO
393 HL!=X*256+Y
394 POKEY2!,PX:POKEY2!,Y:CALLL!,CO,HL!
396 RETURN
400 OX=X:OY=Y
402 K$="":GOSUB50
404 IFK>31THENRETURN
406 IFX=OXANDY=OYTHEN402
408 XX=X:YY=Y
410 IFOX=XTHEN412ELSEIFX>OXTHENX=X-1ELSE
X=X+1
412 IFOY=YTHEN414ELSEIFY>OYTHENY=Y-1ELSE
Y=Y+1
414 LINE(OX,OY)-(X,Y),CO
416 HL!=OX*256+OY
418 POKEY2!,X:POKEY2!,Y:CALLL!,CO,HL!
420 X=XX:Y=YY:GOTO400
500 CO=1:RETURN
600 IFB=0THEN620ELSEXX=X:YY=Y
602 IFABS(PX-X)=1ORABS(PY-Y)=1THENB=0:X=
XX:Y=YY:RETURN
604 IFABS(PX-X)>2THENIFPX>XTHENX=X+1ELSE
X=X-1
606 IFPX>XTHENPX=PX-1ELSEPX=PX+1
608 IFABS(PY-Y)>2THENIFPY>YTHENY=Y+1ELSE
Y=Y-1
610 IFPY>YTHENPY=PY-1ELSEPY=PY+1
612 GOSUB202:GOTO602
620 IFC=0THENRETURNELSEF=1
622 GOSUB300:F=0:C=0:RETURN
700 CLS:PRINT:PRINT"Input from (R)AM, (C
)AS, or (N)o input?"
702 GOSUB95
704 IFK$<>"N"ANDK$<>"n"THEN708
706 K$="":GOSUB2500:RETURN
708 IFK$<>"R"ANDK$<>"r"THEN760
710 GOSUB180:GOSUB712:GOTO716
712 OPENFI$FORINPUTAS1:PRINT:PRINT"Getti
ng sketch input, please wait...":M1=S1:R
ETURN
714 IFEOF(1)THEN715ELSE716
715 CLOSE:CALLV!:K$="":RETURN
716 CH=ASC(INPUT$(1,1))
718 IFCH=N2THEN730
720 IFCH=N1THENCH=N0
722 POKEM!,CH:M1=M1+N1!:IFM1=V!THEN715EL
SE714
730 CH=ASC(INPUT$(1,1))
732 IFCH<N6THEN750
734 CN=CH:CH=ASC(INPUT$(1,1))
736 IFCH=N1THENCH=N0:GOTO744
738 IFCH=N2THENCH=ASC(INPUT$(1,1))ELSE74
4
740 IFCH=N1ORCH=N2THEN744
742 IFCH=N3THENCH=CZELSEIFCH=N5THENCH=CR
ELSECH=DE
744 FORI=N1TOCN:POKEM!,CH:M1=M1+N1!:IFM1
=V!THENI=CN
746 NEXT:IFM1=V!THEN715ELSE714

```

Listing 1 continued

the enter key to return to sketching. If you need more information about a particular function, press the first letter of that function's name, and a screen of related information appears. Press any key to return to the help menu.

You can move the cursor around the screen in a number of ways. Each of the four arrow keys moves the cursor one dot at a time in the indicated direction. Pressing the shift key and an arrow key together moves the cursor five dots at a time up or down or 10 dots at a time left or right. Pressing the control key and an arrow key together moves the cursor to the extreme edge of the screen in one leap.

You can control the number of dots that the cursor jumps when you press the shift key and an arrow key together. Type J and the Jump command asks you two questions. Your first answer sets the left/right jump between one and 239 pixels; your second answer sets the up/down jump to between one and 63 pixels. Press the enter key after each answer and the sketchpad reappears.

The X key lets you move the cursor to a specific point on the screen by specifying its X and Y coordinates. As with the Jump function, you do this by answering two questions.

At times you can lose track of the cursor's location. The Where function helps you find it. Pressing the W key first displays the cursor's current X and Y coordinates, and then when you press any key, it restores your sketch to the screen and marks the cursor with a temporary cross.

Another option to keep in mind is your ability to work in black on white, or white on black. When you first run the program, it makes black dots on the white background. You can reverse this at any time by pressing the zero key (for drawing with white dots) or the 1 key (for drawing with black dots). Use this method to erase—simply draw over an unwanted image in the background color.

You can turn a sketch into a "negative" with the Negative function. Pressing the N key turns the background black, and anything drawn in black becomes white. Pressing the N key again returns your sketch to its original "positive" impression. Remember that the Negative function doesn't change the drawing color. If you turn your sketch into a negative, you must press the zero key to begin drawing white dots on the black background.

### Drawing the Lines

Once in the Sketch mode, the program displays a blank screen with a blinking cursor in the center. The D key puts you in the Draw mode. The cursor leaves a trail of dots as you move it around the screen. Pressing any non-arrow key cancels the Draw mode. You can, however, use the shift and control keys in combination with the arrow keys, or the M key (the Move function which lets you move the cursor without leaving a trail) without breaking out of the Draw function.

The Draw mode is useful for drawing single dots and horizontal and vertical lines, but you can best draw diagonal lines with the Line function, which lets you draw a straight line between two points. First use the P key to set a reference point at one end of the line. Position the cursor at the desired point and press the P key. Move the cursor to the other end of the line you want to draw and press the L key (for the Line function). The program draws a line from point P to the point where you press the L key.

Use the Kill function (K key) to kill the current sketch in memory and start all over again with a blank screen. The

## C•NOTES

program asks you to confirm the kill (Y or N) in case you hit the key by mistake.

The Quit command (Q key) similarly asks you to verify your request (Y or N) and upon execution it returns you to the Model 100 main menu.

Although the Draw and Line commands are sufficient to create any imaginable sketch, easier ways to draw boxes and circles and fill them in with a solid color exist.

To draw a box, place the cursor at one corner and press P to set it as a reference point. Then move the cursor to the opposite diagonal and press B (the Box command). Fill the box (if desired) by pressing the F key (the Fill command) immediately after pressing B.

You can draw a circle by indicating its center point with the cursor and pressing P. Move the cursor along the radius of your circle to the desired length and press the C key—and

*Listing 1 continued*

```

750 IFCH=N1ORCH=N2THEN722
752 IFCH=N3THENCH=CZ:GOTO722
754 IFCH=N5THENCH=CR:GOTO722
755 CH=DE:GOTO722
760 IFK$<>"C"ANDK$<>"c"THEN702
762 GOSUB180:GOSUB154:GOSUB712
768 FORM!=S!TOV!-1
770 INPUT#1,CH:POKEM!,CH:NEXTM!
772 GOTO715
800 CLS:PRINT:PRINT"Enter desired left m
argin as a number":PRINT"between 0 and 3
9,":PRINT"or enter 'N' to exit without p
rinting."
802 PRINT:LINEINPUT"Enter margin: ";M$
804 IFM$<>"N"ANDM$<>"n"THEN808
806 GOSUB2500:K$="":RETURN
808 M=VAL(M$):IFM<0ORM>39THEN800
810 PRINT:PRINT"Press any key when print
er is ready..."
812 GOSUB95:GOSUB850:GOTO806
850 LPRINTCHR$(18);:FORH=0TO7:GOSUB880
852 FORM!=S!+HTOS!+1912+HSTEP8
854 CL=(PEEK(M!)AND(2^(7-H)-1))*2^H
856 CH=PEEK(M!-1)\2^(8-H)
858 LPRINTCHR$(CHORCL)OR128);
860 NEXTM!:LPRINTCHR$(13);:NEXTH
862 GOSUB880:FORM!=S!+7TOS!+1919STEP8
864 LPRINTCHR$(PEEK(M!)OR128);
866 NEXTM!:LPRINTCHR$(13);
868 GOSUB880:FORM!=S!+7TOS!+1919STEP8
870 LPRINTCHR$(PEEK(M!)\128)OR128);
872 NEXTM!:LPRINTCHR$(13);
874 LPRINTCHR$(30):RETURN
880 IFM<>0THENLPRINTCHR$(28);CHR$(M*6);C
HR$(128);:RETURNELSERETURN
900 IFPX=XANDPY=PTHENRETURN
902 IFPX<XTHENIX=PX:XI=XELSEIX=X:XI=PX
904 IFPY<YTHENIY=PY:YI=YELSEIY=Y:YI=PY
906 IFP=1THENP=0:GOSUB1650
908 LINE(IX,IY)-(XI,YI),CO,B
910 IB=1:RETURN
1000 CLS:PRINTCHR$(155)CHR$(154);:INPUT"
jump: how many dots (1-239)";JX!:IFJX!<
10RORJX!>239THEN1000
1002 PRINTCHR$(152)CHR$(153);:INPUT" jum
p: how many dots (1-63)";JY!:IFJY!<10RORJ
Y!>63THEN1002
1004 GOSUB2500:JX=JX!:JY=JY!:RETURN
1100 CO=0:RETURN
1200 IFX=PXANDY=PYTHENRETURN
1202 LINE(PX,PY)-(X,Y),CO
1204 IFPY>YTHEN1210
1206 HL!=PX*256+PY
1208 POKE2!,X:POKEY2!,Y:CALL!,CO,HL!:R
ETURN
1210 HL!=X*256+Y
1212 POKE2!,PX:POKEY2!,PY:CALL!,CO,HL!
:RETURN

```

```

1400 PRINT@280,"Kill old sketch? (press
Y or N)";
1401 GOSUB95
1402 IFK$="Y"ORK$="y"THEN1404
1403 GOSUB2500:K$="":RETURN
1404 CLS:X=120:Y=32:CALL!,0:AX=0:G=0:IB
=0:XI=0:IX=0:IY=0:YI=0:P=0:CO=1:R=0:GOTO
1403
1600 PX=X:PY=Y:P=1
1602 GOSUB99:IF(PEEK(S!+HL)ANDA)<>0THENP
RESET(X,Y):CALLF!,A,HLELSEPSET(X,Y):CALL
O!,A,HL
1604 RETURN
1650 XX=X:YY=Y:X=PX:Y=PY:GOSUB1602
1652 X=XX:Y=YY:RETURN
1700 PRINT@280,"Quit sketching? (press Y
or N)";
1701 GOSUB95:IFK$="Y"ORK$="y"THENMENUELS
EGOSUB2500:K$="":RETURN
1800 R=NOTR:CALLN!:GOSUB2500:RETURN
1900 IFR=0THENAC=1ELSEAC=0
1901 LINE(0,32)-(239,32),AC:LINE(120,0)-
(120,63),AC
1902 FORI=0TO235STEP5:IFIMOD10=0THENLINE
(I,30)-(I,34),ACELSELINE(I,31)-(I,33),AC
1904 NEXT
1906 FORI=2TO62STEP5:IFIMOD10=0THENLINE(
118,I)-(122,I),ACELSELINE(119,I)-(121,I)
,AC
1908 NEXT:AX=1:RETURN
2000 K$="":GOSUB50
2002 IFK=27THENK$="":RETURN
2003 IF(K=127ORK=8)ANDX>5THENX=X-6:K=32:
D=1ELSESD=0
2004 IFX>234ORY>62ORY<6ORK<32ORK>126THEN
2000
2006 POKE2!,Y-6:POKEY2!,X
2008 CALLC!,CO,(K-32)*5
2010 XX=X:YY=Y:XP=PX:YP=PY:X=X+5:PX=X:PY
=Y+1:Y=Y-6:TC=CO
2012 IFCO=1THENCO=0ELSECO=1
2014 GOSUB1200
2016 X=XX:Y=YY:PX=XP:PY=YP:CO=TC
2018 IFX<229ANDD=0THENX=X+6
2020 GOTO2000
2100 CLS:PRINT:INPUT"Enter X coordinate
(0-239)";JX!:IFJX!<0ORJX!>239THEN2100
2102 PRINT:INPUT"Enter Y coordinate (0-6
3)";JY!:IFJY!<0ORJY!>63THEN2102
2104 GOSUB2500:X=JX!:Y=JY!:RETURN
2200 AX=0:G=0:GOSUB2500:RETURN
2300 PRINT@280,"X =";X;"; Y =";Y;";(Press
any key)";
2302 GOSUB95:GOSUB2500
2304 XX=X:YY=Y:FORX=XX-3TOXX+3:GOSUB2308
:NEXTX
2306 X=XX:FORY=YY-3TOYY+3:GOSUB2308:NEXT
Y:Y=YY:K$="":RETURN

```

*Listing 1 continued*



the program draws the circle. Again you may fill the circle by immediately pressing the F key.

You can add text or characters (any of the 95 characters from ASCII 32 through 126) to the screen sketch with the Text command. Press T to enter the mode, and enter text in the usual fashion. The cursor indicates where the lower-left corner of the character will appear.

After drawing a character the program advances the cursor six dots (normal text spacing). You can control exactly

where each character appears by manually positioning the cursor. The only requirement is that you must be able to fit the whole character in at the selected location. In the Text mode, use the delete/backspace key to erase and the escape key to terminate the Text mode.

### Getting Fancy

You can create nice effects by repeating a simple pattern. Once you create a design you like, you can take a snapshot of

Listing 1 continued

```

2308 IFX>=0ANDX<240ANDY>=0ANDY<64THENGOS
UB99:IF(PEEK(S1+HL)ANDA)<>0THENPRESET(X,
Y)ELSEPSET(X,Y)
2310 RETURN
2400 IFIY=0ANDIX=0ANDYI=0ANDXI=0THENRETU
RN
2402 DX=XI-IX:DY=YI-IY:IFX+DX>239THENDX=
239-X
2404 IFY+DY>63THENDY=63-Y
2406 DX=DX+1:DY=DY+1
2408 XX=X:YY=Y:X=IX:Y=IY:GOSUB99
2410 POKE X21,XX:POKEY21,YY
2412 POKENX1,DX:POKENY1,DY
2414 CALLR!,A,HL
2416 Y=YY:X=XX:IFIB=1THENIB=0:GOSUB2500
2418 RETURN
2500 CALLVI:IFG=1THENGOSUB2606
2502 IFAX=1THENGOSUB1900
2504 IFIB=1THENGOSUB908
2506 RETURN
2600 G=1:PRINT@280,"Grid size: (F)ive or
(T)en units?";
2602 GOSUB95:IFK$<>"F"ANDK$<>"f"THEN2610
ELSEGS=5
2604 GOSUB2500:K$="":RETURN
2606 IFR=0THENAC=1ELSEAC=0
2607 FORG=2TO62STEPGS:LINE(0,G)-(239,G),
AC:NEXTG
2608 FORG=0TO235STEPGS:LINE(G,0)-(G,63),
AC:NEXTG:G=1:RETURN
2610 IFK$<>"T"ANDK$<>"t"THEN2602ELSEGS=1
0:GOTO2604
3000 CLS:DEFINTA-Z:X=120:Y=32:CO=1:JX=10
:JY=5
3010 MRI=MAXRAM:S1=MRI-2960:V1=MRI-1040:
O1=MRI-991:A1=MRI-984:F1=MRI-967:N1=MRI-
959:L1=MRI-934:C1=MRI-658:R1=MRI-106
3020 X21=MRI-942:Y21=MRI-941:NX1=MRI-111
:NY1=MRI-110
3030 C2$=CHR$(2):N0=0:N1=1:N2=2:N3=3:N4=
4:N5=5:N6=6:CR=13:CZ=26:DE=127:N11=N1:CR
$=CHR$(12):CZ$=CHR$(25):CD$=CHR$(126):CF
$=CHR$(255):FE=254:FF=255:RETURN
5000 CLS:PRINT"Key is first letter of a
function name:":LINE(0,7)-(239,7):PRINT"
Axes Grid Line Quit Where?
Box Help Move Repeat XY-set
Circle Input Negative Snapshot l:black"
5001 PRINT"Draw Jump Output Text
0:white Fill Kill Point View
"CHR$(152)":cursor";:LINE(0,47)-(239,
47):PRINT"For help on a function, type i
ts key, or press enter to return to sk
etchpad.";
5003 GOSUB95:IFK=13THENGOSUB2500:K$="":R
ETURNELSEIFK>96THENK=K-32
5005 IFK=85ORK=69ORK>88THEN5003

```

```

5006 IFK=49THENK=69ELSEIFK=48THENK=85ELS
EIFK>27ANDK<32THENK=89
5007 K=K-64:IFK<1THEN5003
5008 CLS:ONKOSUB5010,5020,5030,5040,505
0,5060,5070,5080,5090,5100,5110,5120,513
0,5140,5150,5160,5170,5180,5190,5200,521
0,5220,5230,5240,5250:GOTO5000
5010 PRINT:PRINT"(A) draws temporary X a
nd Y axes throughthe center of the scree
n. The scales aremarked in 5 and 10 dot
increments."
5011 PRINT"It can be removed with (V)iew
."
5012 PRINT@280,"Press any key...";
5014 IFINKEY$=" "THEN5014ELSERETURN
5020 PRINT:PRINT"(B) draws a box with on
e corner whereveryou put a point with (P
), and the":PRINT"opposite corner at the
cursor.":GOTO5012
5030 PRINT:PRINT"(C) draws a circle with
its center wheremyou put a point with (P
), and its radiusextending to the cursor
.":GOTO5012
5040 PRINT:PRINT"(D) puts you in draw mo
de. Wherever you move the cursor, you dr
aw a trail of"
5042 PRINT"dots. Exit from draw mode by
pressing any non-arrow key (usually M
for Move).":GOTO5012
5050 PRINT"(l) means you want everything
you sketchto be done in black dots. Thi
s remains in effect until you type a (0
) to changeto white dots. When you first
run the"
5052 PRINT"program you can sketch with b
lack dots without first typing a (l).":
GOTO5012
5060 PRINT:PRINT"(F) fills the box or ci
rcle you just drew with dots of the s
ame color as the border.":GOTO5012
5070 PRINT:PRINT"(G) adds a temporary gr
id of 5 or 10 unit squares to the scr
een.":GOTO5011
5080 PRINT:PRINT"(H) shows the help menu
you just saw. The functions are used
by typing the first letter of their n
ame.":GOTO5012
5090 PRINT:PRINT"(I) allows you to input
a sketch that has been stored in RAM
or on tape.":GOTO5012
5100 PRINT:PRINT"(J) lets you set the nu
mber of dots the cursor jumps when you p
ress a shift and arrow key together. Shi
ft"CHR$(152)" and shift"CHR$(153)" ini
tially jump 5 dots, while shift"CHR$(155
)" and shift"CHR$(154)" initially jump 1
0 dots."

```

Listing 1 continued

## C•NOTES

it and repeat it elsewhere on the screen. To do this, imagine you are drawing a box around the original pattern. Mark one corner of the imaginary box by placing the cursor at that spot and pressing P. Move the cursor to the opposite corner, and press S to take a snapshot of the image between the points. A temporary frame appears around the snapshot as confirmation of what you will reproduce.

Now, to repeat the snapshot pattern, move the cursor to the point on the screen where you want the upper-left corner

Listing 1 continued

```

5105 GOTO 5012
5110 PRINT:PRINT"(K) allows you to kill
the entire sketch and start over with an
empty screen. You must confirm that y
ou want to kill the old sketch.":GOTO50
12
5120 PRINT:PRINT"(L) draws a line from w
herever you put a point with (P) to the c
ursor.":GOTO5012
5130 PRINT:PRINT"(M) lets you move the c
ursor without drawing anything. It is
used to exit from (D)raw mode.":GOTO
5012
5140 PRINT:PRINT"(N) makes a negative of
your sketch by changing all white dots
to black and all black dots to white. Pr
essing (N) a second time will restor
e the sketch to its original condition.
":GOTO5012
5150 PRINT:PRINT"(O) allows you to outpu
t your sketch. You may save it in a RA
M or tape file, or print it on a printe
r that can handle dot-addressable graphic
codes.":GOTO5012
5160 PRINT:PRINT"(P) sets a reference po
int at the cursor for use with other func
tions.":GOTO5012
5170 PRINT"(Q)uit allows you to end the
program after asking for confirmation
. If you re-run the program, your sket
ch will still be there, unless you ra
n another program that destroyed the sk
etch memory area.":GOTO5012
5180 PRINT"(R) is used to repeat part of
the sketch that you took a snapshot of w
ith (S). The snapshot will be repeated
at the"
5182 PRINT"cursor with the cursor being
the upper left corner of the repeated s
napshot. You may repeat the same snaps
hot any number of times.":GOTO5012
5190 PRINT"(S) takes a snapshot of a par
t of the sketch you want to repeat els
ewhere with (R). Put a point at one corne
r with (P), then move the cursor to the o
pposite"
5192 PRINT"corner and press (S). A tempo
rary frame appears around the snapshot t
hat will disappear when you repeat it
with (R).":GOTO5012
5200 PRINT"(T) puts you in text mode so
you can add text characters to your sket
ch.":PRINT:PRINT"BKSP erases the previous
character.":PRINT:PRINT"ESC escapes fro
m text mode.":GOTO5012

```

Listing 1 continued

SAVE AT ELEK-TEK ON CALCULATORS

TEXAS INSTRUMENTS

|          |    |             |     |
|----------|----|-------------|-----|
| BA-2     | 33 | TI-86       | 52  |
| BA-35    | 22 | PC-200      | 52  |
| TI-35-SP | 17 | TI-5040-2   | 48  |
| TI-MBA   | 30 | TI-5130     | 60  |
| BA-55    | 43 | TI-5142-3   | 75  |
| TI-55-2  | 33 | TI-5219     | 130 |
| TI-57    | 29 | TI-LCD-Prog | 50  |

HEWLETT-PACKARD  
SLIMLINE PROGRAMMABLE  
LCD PROBLEM SOLVERS

|                   |    |                   |    |
|-------------------|----|-------------------|----|
| HP-10C Scientific | 54 | HP-12C Financial  | 90 |
| HP-11C Scientific | 70 | HP-16C Programmer | 90 |
| HP-15C Scientific | 90 |                   |    |

NEW! HP-71B Expandable Computer. . . . 399

NEW! HP-41CX \$245

The great features of the 41CV

PLUS Built-in:

- Time Module Function
- Extended Functions Module
- Enhanced Text (ASCII)
- FILE EDITOR

HP-41 CALCULATOR AND ENHANCEMENTS

|                  |     |                     |     |
|------------------|-----|---------------------|-----|
| HP-41C           | 145 | HP-1L Accessories   |     |
| HP-41CV          | 200 | HP-1L Module        | 95  |
| Optical Wand     | 95  | Dig. Cassette Drive | 335 |
| Card Reader      | 145 | HP-1L Printer       | 335 |
| Printer [92143A] | 283 | Acoustic Modem      | 395 |

|                                      |    |                  |    |
|--------------------------------------|----|------------------|----|
| Memory Expansion Modules (for HP41C) |    |                  |    |
| Timer Mod.                           | 60 | Ext. Funct. Mod. | 60 |
| Quad Mod.                            | 60 | Ext. Mem. Mod.   | 60 |

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

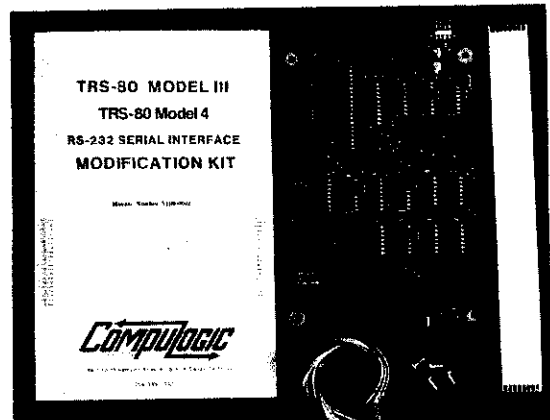
Corp. Accts. Invited. Min. Ord. \$15.00 Mastercard or Visa by mail or phone. Mail Cashier's Check. Mon. Ord. Pers. Check (2 wks to cl). Add \$4.00 tax item (AK, HI, PA, Canada add \$10.00 first item) \$1.00 ea. add'l ship. & hand. Shipments to IL address add 6% tax. Prices subj. to change. WRITE for free catalog. 30 day return policy for defective replacements only. ALL ELEK-TEK MODELS BRAND NEW, 1ST QUALITY AND COMPLETE.

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

## THE RS-232C

MODEL III

MODEL 4



State of the art technology in board design, our direct replacement of Radio Shack's internal RS-232C board, mounts inside the Model III or 4 on the existing brackets. All cables, screws and complete mounting instructions are included. Non-technical people will find that installation is quick, straight forward and simple requiring less than 15 minutes to complete.

Total compatibility with Radio Shack's and all existing software is maintained. Software programmable baud rates from 50 to 19,200 baud are supported along with programmable word length, stop bits, and parity. May be utilized in either half or full duplex operation.

Outstanding Value  
At  
Only **\$69.95**  
Guaranteed One Full Year  
Dealer Inquiries invited

Compulogic

4831 SOUTH HAMPTON RD. LB3  
DALLAS, TEXAS 75210  
214/330-1332

Please forward payment by a cashier's check or money order.  
Visa or Mastercharge also accepted.  
Add \$3.00 shipping & handling  
(Foreign orders quoted on request)

of the imaginary box to appear and press R (Repeat command). The program inserts the snapshot in the selected position, and you can repeat this process as many times as you want.

Understand that the copy always appears with the present cursor position at its top-left corner. Depending on the cursor position, the copy may not all fit on the screen or it may overlap other images. Remember that you always repeat the original snapshot, and if by chance a Repeat command overlaps the original, all Repeat commands thereafter will reproduce the overlapped original. This may sometimes be desirable.

To help you create symmetry, the program includes two functions, Axes (A key) and Grid (G key), that act as temporary overlays. The Axes function draws an X and a Y axis through the center of the screen scaled in five and 10 dot units. The Grid function lets you set the grid size to five or 10

dot squares. It's useful when copying a design from another source to your sketchpad.

As you move across a grid or axis line, the cursor erases the line at that point. If after much moving about there is little left of the original overlay, you can restore it by pressing the A or G key again. To remove the grid or axes, use the V key (the View function) to see how your sketch appears in memory. This command also removes the temporary cross that the Where function places on the cursor.

As you gain experience with the sketchpad, you'll discover many ways to accomplish the same results. For example, there are several ways to erase part of a sketch. As mentioned earlier, you can redraw what you want to erase in the background color. A faster way to do bulk erasing is to draw a box and fill it with the background color. Even in the Text mode you can white out or black out an area of 6 by 8 dots with each press of the space bar.

**Storing and Printing Sketches**

Once a sketch is complete you can save it to a RAM file, a cassette file, or send it to a Radio Shack Line Printer VII for a printout by pressing the O key (Output function). The program prompts you through the process, and if you decide to go back to the sketchpad without saving the drawing, it gives you that option as well. If you choose to print your sketch, the program lets you set the left margin between zero and 39 spaces.

In order to use the print feature, your parallel dot-matrix printer must handle the 7-bit graphic codes used by the Line Printer VII. If you have another printer, make sure it understands the decimal codes listed in Table 2.

The Input command (press I) lets you load a sketch that you have saved as a .DO file in memory or on tape. It gives you three options: to input a sketch from RAM, input a sketch from cassette, or cancel the input request and return to sketching.

To input a file enter the file name without the .DO extension. Again the program prompts you through the process. A message displays while the program inputs or outputs a program, and when done the sketch reappears.

If you get an error message from Basic such as "?FF Error in 712," simply rerun the program and try a new file name. Your sketch will not be lost.

**Installing the Programs**

Since machine-language subroutines aren't relocatable, placing the Sketchpad subroutines in machines of differing memory capacity is complicated. The subroutines must be placed in the highest part of the available memory. Here we present the steps to create a workable version of Sketchpad on a 16K or 24K computer.

*Listing 1 continued*

```
5210 PRINT:PRINT"(0) means you want ever
ything you sketch to be done in white dot
s. This remains in effect until you typ
e a (1) to change to black dots.":GOTO501
2
5220 PRINT:PRINT"(V) lets you view the s
ketch as it looks in memory. It removes a
ll grid and axes lines as well as the te
mporary cross that (W) puts around the c
ursor.":GOTO5012
5230 PRINT:PRINT"(W) tells you where the
cursor is by displaying its X and Y
coordinates and also highlighting the c
ursor by placing a temporary cross over
it.":GOTO5012
5240 PRINT:PRINT"(X) lets you move the c
ursor to a specific location by se
tting its X and Y coordinates.":GOTO5012
5250 PRINT("CHR$(152) ("CHR$(153) ("
CHR$(155) ("CHR$(154) move the curs
or one dot in the indicated direction. S
hift/arrow move it 5 dots "CHR$(152)CHR$
(153) and 10 dots "CHR$(155)CHR$(154)".
These amounts can be changed with (J)um
p."
5252 PRINT"Ctrl/arrows move the cursor t
o the edges of the screen.":GOTO5012
6000 PRINTTAB(14)"SKETCHPAD +"
6002 PRINT"designed by structured softwa
re services"
6004 PRINT"Create sketches by drawing li
nes and"
6006 PRINT"shapes with white or black do
ts. Imagine throwing your eraser away: ju
st draw"
6008 PRINT"with dots of the opposite col
or! Press"
6010 PRINT"H for help or enter to start
sketching.";
6012 GOSUB95:IFK=72ORK=104THENRETURN
6016 CLS:PRINT:PRINT"Press K to kill the
old sketch and start a new one."
:PRINT"or":PRINT:PRINT"Press any other k
ey to continue with the old sketc
h in memory.";
6018 GOSUB95:IFK=75ORK=107THENRETURNELSE
GOSUB2500:K$="":RETURN
```

End

| Code       | Function                                                            |
|------------|---------------------------------------------------------------------|
| 13         | Initiates carriage return and line feed                             |
| 18         | Designates graphics mode                                            |
| 28,xx      | Repeats next graphic pattern xx times                               |
| 30         | Designates character print mode                                     |
| 128 and up | Contain dot pattern that prints in low order seven bits of the byte |

Table 2. Printer codes.

Program Listing 2. Hex.DO.

```

CD3142215F*A110000237E06080FD2FC*
1F5C5D5E5CD4C74E1D1C1F11C05C2ED*
17BFE40DAE9*1141E003EEFBAD8C3E9*11160*
A19B677C9018007215F*A23770B5778B17AC21E*
2C91160*A192FA677C9018007215F*
A237E2F770B78B1C237*
2C9000000000000000E54F3A42*2BCD25B*
22F3C8426FFC35E*2942601573A43*
295B72E01F26D*22EFF2F3C5F7ABBDACF*
2B7C27C*2E1CD21*3C9477B87E56F26002244*
27A2F3CC54F06FF92246*2092248*
2C1E1545DE1CD21*37C8267E52A46*27CB7FAC0*
2D5545D2A48*2192246*2D17BE1856F05C29A*
2CD21*3C9D5545D2A44*2192246*2D1E1C3B8*
243E56F2600292244*27B2F3CC54F06FF092246*
2092248*2C1E1545DE1CD21*37D836FE52A46*
27CB7FA12*3D5545D2A48*2192246*
2D17AE1846705C2EC*2CD21*3C9D5545D2A44*
2192246*2D1E1C30A*
3E5D5C57DE607477D1F1FE61F6F7C070707F5E
60757F1E6F8855F2160*A1978B73E01CA4C*
38705C247*34779B778C25A*
32FA677C1D1E1C9B6C355*3545D21A6*
34F19E52A42*
2545D0605E1D5C506087E0FF5C5D5E5DA81*
30DCA81*30E01626BCD21*379B7CA91*
3CD4C74C394*3CD4D74E1D1C1F11C05C273*
3C1D1142305C26E*
3C90000000000000004F00000007000700147F147
F14242A7F2A1223130864623A45A30280004020
100001C2241000041221C0022147F142208083E0
80800806000000808080800606000004020100
8043E5149453E44427F404062515149462241494
9361814127F1047454529113C4A4949300301790
5033649494936064949291E00002400000080640
000081C36634114141414144163361C080201510
906324979413E7C1211127C417F4949361C22414
122417F41221C7F494949417F090909013E41494
93A7F0808087F000417F41003040413F017F08142
2417F404040407F020C027F7F0608307F3E41414
13E7F090909063E4151215E7F091929462649494
93201017F01013F4040403F0F3040300F7F20182
07F631408146307087808076151494543007F414
10004081020400041417F0004020102044040404
040000102040020545454787F284444383844444
428384444287F385454541808087E090A18A4A9
87C7F0404047800447D40004080847D00007F102
84400417F40007C047804787C080404783844444
438FC1824241818242418FC7C080404085854545
424043F4444203C40403C401C2040201C3C40384
03C44281028441CA0A0907C4464544C440008364
1410000770000414136080002010204020000000
000545D3283*53A81*5472160*A3A42*2192284*
5573242*23A82*54F3A43*25F3A83*
5E5D5C5F5A6626BCADB*50E01CD21*
3CD4C74F1C1D1E10DCACB*5071CD2A8*523C3A8*
505C82A84*51E0816003A42*23CC395*
50E00CD21*3CD4D74C3BA*5
    
```

End

1. Delete line 5000 through end of program.
2. Add: 5000 RETURN
3. Add: 6000 RETURN

Table 3. Changes for a 16K machine.

First, save your current files and programs on tape, and delete them from the computer's memory until you have 17,000 bytes free for a 24K machine, or 12,000 bytes free for a 16K machine.

Then use the Model 100's Text program to create a text file named HEX.DO. Type Program Listing 2, Hex, into this file exactly as it appears. Type in all numbers, letters and asterisks continuously without pressing the space bar or the enter key. Don't be concerned when the Model 100 seems to insert spaces between groups of characters; if you don't press the space bar, there are no actual spaces in the file. After you check your work, press the Function 8 key—return to menu key.

Go into Basic and type in and run Program Listing 3, Loader. This program reads the contents of Hex, calculates the correct addresses for the amount of RAM in your machine, and loads the machine-language instructions into memory.

Listing 3 also checks your Hex file for accuracy. If it finds a non-hex character (other than the asterisks), it displays the incorrect character pair and tells you how many pairs into the file the error exists. If you omit or mistype a character(s), the program gives a checksum error and asks you to recheck your Hex file.

The asterisks in the Hex file represent the high order bits of addresses you must relocate for a particular memory size. Listing 3 adjusts these addresses and loads the instruction into memory locations based on the number returned by the MAXRAM function in your machine. The Checksum routine ensures that the total number of bytes (not counting the asterisks) and the number of bytes loaded are correct.

When everything checks out correctly, you have created SKSUBR.CO, the Sketchpad subroutines. At this point, make a back-up of SKSUBR.CO which contains the machine-language subroutines tailored for your machine. Making back-ups of Hex and Loader is a good idea if you ever add memory to your computer and need to move SKSUBR.CO to a higher location in memory.

SKSUBR.CO is protected by a return instruction at its entry point address, to prevent harm from accidentally running the program directly from the main machine menu. If you should ever hit the enter key while the cursor is on SKSUBR.CO, the screen goes blank for a second, and the main menu reappears.

Once SKSUBR is successfully intact, kill the Loader and Hex files. Type in Listing 1, Sketchpad, and save it as Sketch.BA.

For a 16K machine, you must omit the help feature as well as the title screen. When you run the 16K version, you see only a blank screen and a blinking cursor. Press V (the View function) to see the old sketch in memory, or press K (the Kill function) to start a new sketch. See Table 3 for program changes for a 16K version.

Any time you feel the urge to sketch, run Sketch.BA and it automatically loads SKSUBR.CO. The subroutines and sketch bit map remain in protected memory unless you run another program that again makes that area available to Basic, such as a Clear 100, MAXRAM statement. ■

Contact John and Aileen Cornman c/o Structured Software Services, 9233 N.E. 269th St., Battle Ground, WA 98604.



Program Listing 3. Loader.BA.

```

1  '*** LOADER ***
10 CLEAR100,MAXRAM-2961
12 CLS:DEFINT A-Z:I=0:CS#=0
14 M1=MAXRAM-1040
16 IFM1>60000THENAD=15:GOTO20
17 IFM1>52000THENAD=13:GOTO20
18 IFM1>44000THENAD=11:GOTO20
19 AD=9
20 OPEN"HEX"FORINPUTAS1
30 PRINT:PRINT"LOADING 'HEX.DO' INTO
MEMORY..."
40 HX$=INPUT$(2,1)
50 D$=LEFT$(HX$,1)
55 IFD$="*"THEN300
60 GOSUB500
65 IFD=-1THEN200
70 DC=D*16
80 D$=RIGHT$(HX$,1)
90 GOSUB500
95 IFD=-1THEN200
100 DC=DC+D:CS#=CS#+DC
110 POKE(M1+I),DC:I=I+1
120 IFEOF(1)THEN130ELSE40
130 CLOSE:IFCS#=77855ANDI=1030THEN140
132 PRINT:PRINT"CHECKSUM ERROR. PLEASE
RECHECK CONTENTS"

```

```

134 PRINT"OF 'HEX.DO' FILE AND RERUN
LOADER WITH"
136 PRINT"CORRECTED FILE."
138 END
140 PRINT:PRINT"SUCCESSFUL LOAD.
'SKSUBR.CO' WILL BE"
142 PRINT"CREATED FOR USE WITH SKETCHPAD
+."
150 SAVEM"SKSUBR",M1,M1+1029,M1+155
160 END
200 CLOSE:PRINT:PRINT"NON-HEX CHARACTER
IN 'HEX.DO':"
210 PRINT:PRINT"PAIR #";I+1;"READS:
";HX$
220 PRINT:PRINT"PLEASE CORRECT AND RE-
RUN LOADER":END
300 D$=RIGHT$(HX$,1)
310 GOSUB500
320 IFD=-1THEN200
330 CS#=CS#+D
340 IFD=10THENDC=(AD-1)*16+DELSEDC=AD*16
+D
350 GOTO110
500 D=ASC(D$)
502 IFD>47ANDD<58THEND=D-48:RETURN
504 IFD>64ANDD<71THEND=D-55:RETURN
506 D=-1:RETURN

```

End

## From Soup to Nuts

by Ben Firschein

You can create a professional-looking command menu for your Model 100 programs. Menu.BA displays all of your program's options and lets you select one by hitting a few arrow keys (see Program Listing 4).

Besides adding elegance to your programs, Menu demonstrates how to use reverse video, move a cursor around the screen with the PRINT@ command, and format data with the Print Using command.

When you run Menu, sample commands appear on the screen just as programs and files appear on the Model 100's menu. Use the arrow keys to move the reverse-video cursor up, down, left, and right. This menu wraps around like the Model 100's own menu: go past the last command, and the cursor returns to the first command; hit the left-arrow key at the first command, and the cursor goes to the last command.

If you hold down the arrow key for more than a few seconds, the cursor continues to move in that direction. If you press the up- or down-arrow key for more than a few seconds, the cursor stops when it comes to the top or bottom of the screen, and if you hold down the left- or right-arrow key for more than a few seconds, the cursor continues moving and wrapping around when it hits a side.

To make a selection, position the cursor over the command you want to invoke and hit the return key. The program tells you the item you picked, as well as its number. When you select exit, the program ends.

You don't have to test to ensure that you've entered a valid command since you can choose only those commands displayed in the menu.

|       |         |         |         |        |
|-------|---------|---------|---------|--------|
| help  | exit    | assets  | profit  | intrst |
| inven | sales   | margin  | portfol | file   |
| print | retail  | utility | billing | rent   |
| amort | request | calc    |         |        |

SELECT: USE ARROW KEYS AND ENTER KEY

Figure. Sample menu selection.

### Program Description

Menu is a sample program that illustrates the idea behind the command menu. In using this menu subroutine within your programs, substitute your own options for the sample selections shown here (see the Figure).

Lines 100-995 make up the main program. Line 110 sets the maximum number of selections. I've chosen 20 as a maximum (see the Figure), but you can configure the screen to accommodate more.

Lines 350-400 call the subroutines that load the names of the selections, display, and let you choose among them. Line 500 prints your choice and its number, determined by its position in the data (lines 1008-1020). Your program could use an ON GOTO or ON GOSUB statement to branch to the part of the program dealing with a given command.

For example, if line 8000 gives help, line 8100 lets you exit, and line 8200 tells you your assets, you can add the following line:

```
505 ON CS GOSUB 8000, 8100, 8200
```

Lines 1000-1050 load the choices and store them in the array

# THE ENTREPRENEUR'S SPREADSHEET OFFERS MODEL 100 AND \*NEC PC-8201 USER MORE POWER.

- Easy to use
- Sales Projections
- Budgeting
- Financial Forecasting
- Many Other Business Uses

The Entrepreneur's Spreadsheet is a powerful 16-column by 50-row financial management tool. This flexible software has full computation capabilities including summation, exponents, and algebraic equations. Other features are expanded command set, creation of printed reports and ease-of-use. It is completely documented with sample session. This allows you to be up and running right away. Full telephone support for this versatile package is offered.

**124.95**

Also available:  
The Entrepreneur's General Ledger **\$119.95**

When you order specify computer make & model. Phone Visa or Master Charge number or send cash or money order. Your order will be shipped within 24 hours

**JAMES W. SECOR**

Data Processing Consultants

2050 N. Collins, #101, Richardson, Tx 75080  
214/899-3649



\*NEC Corporation, Tokyo, Japan

254

## Introducing I/III to 4 CONVERTER TRS 80 MODEL 4

I/III to 4 CONVERTER IS THE **EASY AND INEXPENSIVE** WAY TO CONVERT YOUR MODEL I/III **BASIC** PROGRAMS TO MODEL 4.

- Completely Menu Driven
- Runs on 1, 2, 3, or 4 Drive Systems
- Inserts necessary spaces between key words
- Prints list of key words and line numbers where exceptions occur for Model 4
- Step-by-step instruction manual included
- Converts "Print @" and "Print Tab ( )" addresses for 80 Column screen.

**NEXT DAY SHIPMENT ON TELEPHONE/CREDIT CARD ORDERS**



**\$39.95**



TO ORDER CALL

**(803) 787-7256**

TELEX 466528

**Ehlen Enterprises**

6319 Briarwood Road  
Columbia, South Carolina 29206

322

## C•NOTES

Program Listing 4. Menu.BA.

```

10 ' Menu Program
20 ' Ben Firschein
50 '
100 '---MAIN PROGRAM---'
102 '
110 MSL=20:REM maximum number of
selections
200 CLS:PRINT
210 DIM SL$(MSL):REM stores names of
selections
350 GOSUB 1000:REM load names of
selections
360 GOSUB 2000:REM display selections
400 GOSUB 3000:REM make choice
500 CLS:PRINT:PRINT"you selected:
";SEL$;" (choice";CS;")"
510 FOR ZZ=1 TO 1000:NEXT
520 IF SEL$="exit"THEN CLS:END ELSE 360
990 '
995 END
998 '
1000 '---LOAD SELECTIONS---
1002 '
1008 DATA
help,exit,assets,profit,intrst,inven,sal
es,margin,portfol,file,print
1010 DATA
retail,utility,billing,rent,amort,reques
t,calc
1020 DATA *
1030 SL=0:REM no selections
1035 RESTORE
1040 READ SL$:IF SL$<>"*" AND SL<MSL
THEN SL=SL+1:SL$(SL)=SL$:GOTO 1040
1042 IF SL$<>"*" THEN BEEP:PRINT"too
many selections in data":END
1050 RETURN
1060 '
2000 '---DISPLAY SELECTIONS---
2002 '
2010 CLS
2012 PRINT @241," ";CHR$(27)
;"p";"SELECT:USE ARROW KEYS AND ENTER
KEY";CHR$(27);"q";
2013 ' print at screen position 241
instructions in reverse video
2014 ' chr$(27);"p" turns on rev
video. chr$(27);"q" turns off rev video
2015 PRINT @40,"";:REM place cursor at
print position 40
2017 FOR K=1 TO SL
2020 PRINT USING "\ \";SL$(K);
2021 ' print selection with 8 column
width.
2022 'there are six spaces between
2023 'the slashes. For a slash hit
2024 'GRPH -
2025 '
2035 NEXT K
2040 CS=1:REM current selection
2045 PS=0:REM used by selection
subroutine
2050 RETURN
2060 '
3000 REM -----SELECT AN ENTRY -----
3002 '
3010 GOSUB 4000:REM show first choice
3020 K$=INKEY$:IF LEN(K$)=0 THEN
3020:REM TEST FOR KEY

```

Listing 4 continued

Listing 4 continued

```

3050 A=ASC(K$)
3055 IF A=13 THEN SEL$=SL$(CS)
:RETURN:REM user has selected SEL$
3056 ' 13 is ascii code for carriage
return
3057 '
3060 IF A<>28 THEN 3070
3065 CS=CS+1: IF CS <=SL THEN GOSUB 4000
ELSE CS=1:GOSUB 4000:REM right arrow
3070 IF A<>29 THEN 3080
3075 CS=CS-1:IF CS>=1 THEN GOSUB 4000
ELSE CS=SL:GOSUB 4000:REM left arrow
3080 IF A=31 AND CS+5 <=SL THEN CS=CS+
5:GOSUB 4000:REM down arrow
3090 IF A=30 AND CS-5 >0 THEN CS=CS-
5:GOSUB 4000:REM up arrow
3095 GOTO 3020:REM another key
3100 '
4000 REM ---DISPLAY CURRENT SELECTION---
4002 '
4005 IF PS <>0 THEN PRINT @PS,"";:PRINT
USING "\ \";S2$;
4010 ' print over reverse vidio. (5
spaces inside \)
4020 PS=40+(CS-1)*8:REM print position
on screen
4022 PRINT @PS,CHR$(27);"p";:REM reverse
video on
4023 PRINT USING "\ \";SL$(CS);:REM
print selection (5 spaces in \)
4024 PRINT CHR$(27);"q";:REM reverse
video off
4025 S2$=SL$(CS):REM store prev choice
4030 RETURN

```

End

SL\$. Substitute your own choices for those in my program.

Lines 2000-2050 display the selections. This section of the program features the commands PRINT@ and Print Using, and also demonstrates how to print in reverse video. PRINT@ 241 in line 2012 tells the computer to start printing at position 241, the first column of the seventh line. CHR\$(27);"p" creates reverse video. CHR\$(27);"q" reactivates normal video display.

Line 2015 starts printing at print position 40. The program prints the selections stored in the array SL\$. The PRINT USING "\ \";SL\$(K); command prints the entry in a field of eight columns. It pads entries of fewer than eight letters with spaces, and truncates longer entries. Print Using makes a field of two plus the number of spaces between the inverted slashes (six in my program). You get the inverted slash character ( \ ) by pressing the graphics (grph) and then the hyphen key.

Lines 3000-3095 select an entry. The program initially positions the cursor on the first entry. Line 3010 places the cursor on the menu. The program tests for a key using INKEY\$, and when it detects one, converts it to its ASCII code. ASCII code 13 is a carriage return; once you press a carriage return, you've made a selection. Lines 3060-3090 test for arrow keys. If you've pressed one, then the program must modify the pointer to the array SL\$, indicating what selection you're pointing to. It then calls subroutine 4000 to change the cursor's screen location.

Lines 4000-4030 remove the old cursor, compute the position of the new one, and display it. The program again uses a PRINT@ command on line 4005 to move the cursor to its old

location. The PRINT USING "\ \";S2\$ command removes the reverse-video cursor and replaces the normal one.

I use a field of seven, rather than eight, columns for the cursor so that it doesn't touch the entry to its right—a choice I made for aesthetic reasons. (A seven-column field requires five spaces inside the dashes.) Line 4020 computes the print position on the screen. Since the program displays five entries per line, each in a field eight characters wide, each line uses 40 characters, the width of the Model 100's screen.

Line 4022 moves the cursor to the new position and turns on the reverse video. Line 4023 prints the entry pointed to in reverse video in a seven-character field; this is the cursor. Line 4024 turns off the reverse video.

Menu never looks at the screen; by maintaining a pointer, CS, to the current entry, the program always knows over which entry the cursor is positioned. When you hit the return key, the program returns SEL\$, the last command indicated. ■

You can reach Ben Firschein at 29 Stowe Lane, Menlo Park, CA 94025.

## Tidbit #5

Here's a way to hide program lines so they don't appear on screen when someone lists the program. This is handy for teachers who don't want to disable the break key but also don't want students listing a program to find the answers.

End each line that you want to make disappear with a REM statement and insert one or two characters. Using the Edit mode, change those characters to a shift/up-arrow. This makes the next program line list on top of the one protected with the up arrow.

For an example, type in my sample program and run it. Then, type EDIT 30 <enter> SQ C <shift><up arrow> C <shift><up arrow> <enter>. Make sure you press the shift and up-arrow keys at the same time.

Run the program again. List it, and notice that lines 20 and 30 do not appear.

To reinsert the lines to the listing, enter the Edit mode and delete the REM statements.

David Dickey  
Millinocket, ME

```

10 CLS
20 FOR X=1TO5:READ A$:PRINT A$:NEXT X
30 DATA THIS, LINE, WILL, DISAPPEAR, SOON: END
40 'THIS WORKS FOR ANYTHING YOU WANT AND FOR MORE
PROGRAM LINES

```

## It's a Chimer

My BIGBEN program in Program Listing 1 makes the Model 100 sound Westminster chimes every quarter hour and count the time on the hour. I hope you enjoy running it.

*James M. Stubchaer  
869 North Kellogg Ave.  
Santa Barbara, CA 93111*

## A Better Solution

John Hicks' letter, "Peripheral Problems" (RAM Files, February 1984, p. 197), gives an unnecessarily complex remedy to stop the STR\$ instruction from inserting a space character to the right of the numerals.

It is simpler to use the MID\$ instruction with the position set at 2 and the length omitted as follows:

```
MID$(STR$(BAL),2)
```

Program Listing 2 is a test program that demonstrates this remedy. This idiosyncrasy also occurs on my Color Computer.

*Palmer O. Hanson Jr.  
Box 1421  
Largo, FL 34294-1421*

## Mysterious Disappearance

I redefined function key 7 as "Kill" to delete files, but the function key definition disappears every time I use my parallel printer.

Is there anything I can do to protect the function key definitions when running my printer?

*J. Hoke Peacock II  
Box 1751  
Beaumont, TX 77004*

*You've got us; can anyone else help?—Eds.*

*Program Listing 2. Test program using the MID\$ instruction.*

```
10 BAL=133
20 B$="$"+STR$(BAL)+".00"
25 LPRINT
30 LPRINTB$
40 B$="$"+RIGHT$(STR$(BAL),LEN(STR$(BAL)
)-1+ ".00"
45 LPRINT
50 LPRINTB$
60 B$="$"+MID$(STR$(BAL),2)+".00"
65 LPRINT
70 LPRINTB$
80 END
```

*End*

*Program Listing 1. Westminster chimes program.*

```
10 PRINT"pgm 'BIGBEN', time & chimes"
20 REM for TRS-80 Model 100
30 REM by James M. Stubchaer
40 REM 869 N. Kellogg Ave.
50 REM Santa Barbara, Ca. 93111
60 REM
70 READ B,A,G,D,C
80 DATA 2484,2793,3134,4184,4697
90 S=15: L=25
100 IF MID$(TIME$,4,2)="15" THEN GOSUB
190 ELSE 120
110 GOSUB 290
120 IF MID$(TIME$,4,2)="30" THEN GOSUB
190: ELSE 140
130 GOSUB 290
140 IF MID$(TIME$,4,2)="45" THEN GOSUB
190: GOSUB 210: GOSUB 230 ELSE 160
150 GOSUB 290
160 IF MID$(TIME$,4,2)="00" THEN GOSUB
190: GOSUB 210: GOSUB 230: GOSUB 250
ELSE 180
170 GOSUB 290
180 GOTO 100
```

```
190 SOUND B,S: GOSUB 300: SOUND A,S:
GOSUB 300
200 SOUND G,S: GOSUB 300: SOUND D,L:
GOSUB 300: RETURN
210 SOUND D,S: GOSUB 300: SOUND A,S:
GOSUB 300
220 SOUND B,S: GOSUB 300: SOUND G,L:
GOSUB 300: RETURN
230 SOUND B,S: GOSUB 300: SOUND G,S:
GOSUB 300
240 SOUND A,S: GOSUB 300: SOUND D,L:
GOSUB 300: RETURN
250 SOUND D,S: GOSUB 300: SOUND A,S:
GOSUB 300
260 SOUND B,S: GOSUB 300: SOUND G,L:
GOSUB 300
270 N=VAL(LEFT$(TIME$,2)): IF N>12 THEN
N=N-12
280 FOR J=1 TO N: SOUND C,L: FOR K=1 TO
300: NEXT: NEXT
290 FOR J=1 TO 35000: NEXT: RETURN
300 FOR J=1 TO 225: NEXT: RETURN
```

*End*



## RAM FILES

### Autoliner Appended

Since I don't always finish typing in a listing at one sitting, I made some changes to Richard Ramella's "Autoliner" (January 1984, p. 178). Program Listing 3 lets you continue where you left off on a listing or change the number sequence in the middle of a listing.

*Jacqueline Davis  
Box J  
Lovelock, NV 89419*

*Program Listing 3. Changes for Richard Ramella's "Autoliner."*

```

140 DEFINTB,D,F,W:DEFSTR A,L,Z,M
211 INPUT"New Program or <E>xisting
Program";M
212 IF M="E" GOTO 610
213 IF M="e" GOTO 610
365 IFA="#"THEN INPUT"NEW Line
Sequence";B:GOTO 330
610 OPEN ZFOR INPUT AS 1
612 N=0
620 IF EOF(1) THEN 650
630 N=N+1:LINEINPUT #1,A
640 PRINTA:GOTO 620
650 CLOSE 1:A=""
660 OPEN ZFOR APPEND AS 1
670 GOSUB 580:GOTO 240
710 END
    
```

*End*

**OMNITEK COMPUTERS INTERNATIONAL, INC.**  
1300 MAIN STREET TEWKSBURY, MASS 01876  
617-851-4580 ✓195

|                                                         |                        |
|---------------------------------------------------------|------------------------|
| Okidata Microline 82A.....                              | 319.00                 |
| Okidata Microline 83A.....                              | 539.00                 |
| Okidata Microline 92 (160 C.P.S.) corresponds mode..... | 419.00                 |
| Okidata Microline 93.....                               | 669.00                 |
| New EX-80 Printer.....                                  | 239.00                 |
| 13. Green Monitor.....                                  | 89.00                  |
| B.M.C. 13. Color Monitor.....                           | 229.00                 |
| Epson FX80 FT.....                                      | 479.00                 |
| Juki 6100 Parallel Daisy Wheel.....                     | 459.00                 |
| Columbia (IBM compatible).....                          | CALL                   |
| Call for Acrtix Portable, Epson, QX10                   |                        |
| 40 track economy drive Power Supply with case.....      | 159.00                 |
| Tandon drives with Power Supply and case                |                        |
| 40 track singlehead.....                                | 229.00                 |
| dual head.....                                          | 279.00                 |
| 80 track singlehead.....                                | 279.00                 |
| dual head.....                                          | 349.00                 |
| 5.25. Power Supply and case.....                        | 45.00                  |
| *BASF 40 track D.D. 5 1/4. new disk drive, as is,       |                        |
| no return.....                                          | 89.00                  |
| 8" Power Supply and case.....                           | 75.00 or 10 for 600.00 |
| Call for popular D.W. Printer Prices                    |                        |
| Full Commodore Line.....                                | CALL                   |

### OMNITEK COMPUTERS INTERNATIONAL, INC.

TRS-80 is a reg. trademark of Tandy Corp. Prices are for mail order only TERMS: Check, money order, Mastercard and Visa accepted. F.O.B. Tewksbury-freight extra. Minimum \$5.00 S & H. Mass residents add 5% sales tax. Write for FREE CATALOG.

# PortaPac™ / 100



**Priced at \$395.00  
for 64K of memory**

**DEALER INQUIRIES INVITED**

*Registered Trademarks:*

IBM - International Business Machines Corp.  
TRS-80 - Tandy Corp.  
CP/M - Digital Research

- A portable, external RAM Disk expanding the memory of your TRS-80 model 100.
- Transfers data from your portable computer to an IBM PC or CP/M based systems.
- Built-in DOS

*Please Send For Descriptive Literature*



## Cryptronics, Inc.

11711 Coley River Circle, Suite 7  
Fountain Valley, CA 92708  
Phone: (714) 540-1174 ✓268

C-NOTE FOR JUNE 1984

## Form-matters

by John and Aileen Cornman

Text, the Model 100's word processing program, is handy for portable word processing, but it lacks many of the features necessary to prepare custom-formatted documents like letters.

If you have a Model III and Scripsit, you can use our Basic program, Forms, to design final documents by embedding Scripsit formatting codes in your Model 100 text files (see Program Listing 1).

Forms converts Model 100 Text file tapes to Model III Scripsit tapes. It also converts certain Model 100 characters to Scripsit text boundary markers and strips off the extra line feeds the Model 100 invokes after a carriage return.

### Installing the Program

In addition to the object code for Forms, Listing 1 contains a Model 100-to-Scripsit loader that installs Forms in your Model III. This Basic program automatically adjusts the machine-language instructions in Forms for the amount of memory in your machine.

As a result, Forms runs on 16K, 32K, and 48K machines without modification. The loader helps you find mistyped hexadecimal (hex) codes in the data statements and computes a checksum to ensure accuracy.

### Running the Loader

To load Forms, turn on your Model III and answer the memory size prompt with 31953 for a 16K machine, 48337 for a 32K machine, and 64721 for a 48K machine. Then run the program. If you made an error in specifying the memory size, the program displays acceptable responses and asks you to try again.

#### Program Listing 1. Basic loader and object code for Forms.

```

10 CLS:DEFINT A-Z
20 PRINTTAB(19)"Model 100 to SCRIPSIT Loader"
30 PRINTTAB(17)"by Structured Software Services":PRINT
40 PRINT:PRINT"Loading FORMS into memory, please wait..."
50 GOSUB 550 "CHECK MEMORY SIZE AND SET UP ADDRESSES
60 FOR I=10014
70 READ HXS
80 DS = LEFT$(HXS,1) 'GET A PAIR OF HEX DIGITS
90 IF DS = "*** THEN 210 'LOOK AT THE LEFT DIGIT
100 GOSUB 500 '* HEANS ADDRESS ADJUSTMENT
110 IF D = -1 THEN 400 'CONVERT HEX DIGIT TO DECIMAL
120 DC = D * 16 '-1 IS FLAG FOR BAD HEX CHAR.
130 DS = RIGHT$(HXS,1) 'DC GETS VALUE OF LEFT DIGIT
140 GOSUB 500 'LOOK AT THE RIGHT DIGIT
150 IF D = -1 THEN 400 'CONVERT RIGHT DIGIT
160 DC = DC + D 'DC = THE DECIMAL CONVERSION
170 CS1 = CS1 + DC 'ADD IT TO THE CHECKSUM
180 POKE PA,DC 'LOAD IT INTO MEMORY
190 IF PA < 32767 THEN PA = PA + 1 'ADVANCE POKE ADDRESS
200 GOTO 300 'READY FOR NEXT HEX PAIR
210 DS = RIGHT$(HXS,1) 'DS = RIGHT DIGIT OF *-PAIR
220 GOSUB 500
230 IF D = -1 THEN 400
240 CS1 = CS1 + D 'ADD TO CHECKSUM
250 DC = BA + D 'BASE ADDRESS + DISPLACEMENT
260 GOTO 180 'GO POKE ADJUSTED ADDRESS
300 NEXT
310 IF CS1 <> 75355 THEN 350 'VERIFY CHECKSUM TOTAL
320 PRINT:PRINT"FORMS has been successfully loaded."
330 PRINT:PRINT"Entry address for SYSTEM command is: ";PAI
    
```

Listing 1 continued

#### Listing 1 continued

```

340 PRINT:PRINT"You may now convert Model 100 TEXT tapes to SCRIPSIT
IT format." : PRINT : END
350 PRINT:PRINT"Checksum error."
360 PRINT:PRINT"Please correct contents of DATA lines 800 - 890 and
rerun." : PRINT : END
400 LN = INT((I-1) / 18) 'RELATIVE LINE # OF ERROR
410 CH = I - (LN * 18) 'BAD CHARACTER PAIR #
420 LN = 800 + LN * 2 'ACTUAL DATA LINE #
430 PRINT:PRINT"Invalid hex character: ";HXS
440 PRINT:PRINT"Please correct pair #";CH;"in line";LN;"and rerun."
450 PRINT : END
500 D = ASC(D$)
510 IF D > 47 AND D < 58 THEN D = D - 48 : RETURN '0-9 OK
520 IF D > 64 AND D < 71 THEN D = D - 55 : RETURN 'A-F OK
530 D = -1 : RETURN 'BAD HEX CHARACTER
550 BA = PEEK(16562) 'GET BASE ADDRESS
560 IF BA = 252 OR BA = 188 OR BA = 124 THEN 570 ELSE 620
570 IF PEEK(16561) <> 207 THEN 620
580 PAI = BA * 256 + 210 'PAI = START ADDRESS FOR POKING
590 IF PAI > 32767 THEN PAI = -1 * (65536 - PAI) ELSE PAI = PAI
600 RETURN 'PA ADJUSTED IF PAI > 32767
620 PRINT:PRINT"Before running FORMS Loader, you must answer the Me
mory Size?"
630 PRINT"question according to the following table:" : PRINT
640 PRINT"Machine Memory Size?"
650 PRINT "16K 31953"
660 PRINT "32K 48337"
670 PRINT "48K 64721"
680 PRINT:PRINT"Please press the orange reset button and answer th
e"
690 PRINT"Memory Size? question again before rerunning this progra
m."
700 GOTO 700
800 DATA CD,C9,01,21,15,3C,22,20,40,21,69,*2,CD,1B,02,21,50,3C
802 DATA 22,20,40,21,FF,*2,CD,89,*2,3E,03,32,14,42,CD,91,*2,DA
804 DATA 00,80,3E,48,32,11,42,21,E,*3,CD,1B,02,21,91,*3,CD,1B
806 DATA 02,CD,89,*2,CD,7F,*2,F3,21,00,00,CD,96,02,CD,35,02,FE
808 DATA 9C,20,F3,21,F8,*3,01,00,06,CD,77,*2,77,23,10,F9,06,0A
810 DATA CD,77,*2,18,FB,79,ED,44,4F,CD,35,02,B9,C2,59,*2,CD,F8
812 DATA 01,3A,F0,*3,FE,06,28,22,61,06,00,21,F1,*3,11,F8,*3,1A
814 DATA ED,A1,13,20,05,EA,4F,*1,18,0E,21,ET,*3,CD,1B,02,21,F8
816 DATA *3,CD,89,*2,18,A8,21,50,44,E5,21,00,00,CD,96,02,EL,CD
818 DATA 35,02,FE,8D,C2,59,*2,01,00,00,CD,77,*2,FE,0A,2B,02,77
820 DATA 23,10,F5,79,ED,44,4F,CD,35,02,B9,C2,59,*2,2B,7E,23,FE
822 DATA 1A,20,00,CD,F8,01,21,50,44,C1,00,00,3E,1A,ED,B1,2B,36
824 DATA 00,23,11,50,44,37,3F,ED,52,44,4D,C5,EB,3E,5B,ED,B1,20
826 DATA 20,F5,2B,36,97,23,16,03,7E,P6,80,77,FE,BE,23,0B,28,03
828 DATA 15,20,F3,F1,E2,DD,*1,78,B7,FA,DD,*1,B1,20,DA,C1,C5,16
830 DATA 5D,1E,9B,CD,68,*2,C1,C5,16,0D,1E,8D,CD,68,*2,C1,C5,16
832 DATA AF,1E,8E,CD,68,*2,C1,C5,16,0C,1E,8C,CD,68,*2,C1,ED,43
834 DATA EE,*3,16,5E,1E,8B,CD,68,*2,CD,C9,01,21,91,*3,CD,1B,02
836 DATA 21,87,*3,CD,89,*2,AF,32,11,42,CD,7F,*2,ED,5B,EE,*3,CD
838 DATA BB,*2,CD,F8,01,21,5A,*3,CD,89,*2,CD,49,00,FE,59,20,E6
840 DATA FE,79,28,E2,21,77,*3,CD,89,*2,CD,49,00,FE,59,01,F1
842 DATA FE,59,CA,F2,*0,FE,79,CA,E2,*0,C3,50,44,7A,ED,B1,CD,FS
844 DATA 81,21,D7,*3,CD,89,*2,C3,F2,*0,21,50,44,7A,ED,B1,CD,FS
846 DATA 2B,73,23,F1,ED,18,F5,CD,35,02,F5,81,4F,F1,C9,21,36,*3
848 DATA CD,89,*2,CD,49,00,C9,CD,1B,02,2B,CD,1B,02,C9,21,1F,*3
850 DATA CD,1B,02,21,F1,*3,06,06,CD,40,00,00,48,06,00,89,3E,06
852 DATA 91,32,F8,*3,C8,47,3E,20,77,23,10,FC,C9,F5,81,4F,F1,CD
854 DATA 64,02,C9,21,00,00,CD,07,02,21,50,44,01,00,00,03,1B,7A
856 DATA B3,28,05,78,FE,01,20,F5,41,0E,60,78,CD,64,02,7E,CD,03
858 DATA *2,23,10,F9,79,CD,64,02,7A,B3,28,DC,C9,AD,6F,64,65,6C
860 DATA 20,31,30,30,20,74,6F,20,53,43,52,49,50,53,49,54,0D,62
862 DATA 79,20,53,74,72,75,63,74,75,72,65,64,20,53,6F,64,74,77
864 DATA 61,72,65,20,53,65,72,76,69,63,65,73,8D,45,6E,74,65,72
866 DATA 20,54,45,58,54,20,66,69,66,20,6E,61,6D,65,3A,20,03
868 DATA 50,72,65,73,73,20,45,4E,54,45,52,20,77,68,65,6E,20,72
870 DATA 65,63,6F,72,64,65,72,20,69,73,20,72,65,61,64,79,2E,0D
872 DATA 57,72,69,74,65,20,61,6E,6F,74,68,65,72,20,63,6F,70,79
874 DATA 3F,20,28,59,20,6F,72,20,4E,29,0D,52,65,61,64,20,61,20
876 DATA 6E,65,77,20,74,61,70,65,3F,20,28,59,20,6F,72,20,4E,29
878 DATA 0D,50,72,65,70,61,72,65,20,74,6F,20,03,70,6C,61,79,20
880 DATA 4D,6F,64,65,6C,20,31,30,30,20,54,45,58,54,20,74,61,70
882 DATA 65,2E,0D,72,65,63,6F,72,64,20,4D,6F,64,65,6C,20,49,49
884 DATA 49,20,53,43,52,49,50,53,49,54,20,74,61,70,65,2E,0D,43
886 DATA 68,65,63,6B,73,75,6D,20,65,72,72,6F,72,2E,0D,55,6B,69
888 DATA 70,3A,20,03,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
890 DATA 00,00,0D,00
    
```

End

### The Key Box

The program in "Form-matters" requires 8K RAM to run on the Model 100; it also runs on a Model III with at least 16K RAM and Cassette Basic. The programs in "The Searcher" and "Getting Personal" each require 8K RAM; "Getting Personal" also calls for a printer.

The loader then asks you to wait while it reads the hex codes in the data statements and loads them into memory. If the loader encounters an invalid hex character in the data statement, it displays the bad character pair, indicating the line number and pair number that need correction before you re-run the loader. Each data line contains 18 pairs of characters, except the last line where there are four.

Even if all the hex codes are acceptable, they can still be incorrect, so the loader tells you if the hex codes in the data statements add up to something other than the predetermined checksum total.

With an error of this type, the loader can provide no further information to help you find the incorrect code. You must re-check your data statements against Listing 1 to find the error before rerunning the loader.

If the loader finds no errors, it tells you the entry address to use with the System command when you run Forms. You're now ready to convert Model 100 Text tapes into the Model III Scripsit format.

### Using Forms

To use Forms, enter the System command, then enter the appropriate entry address at the asterisk prompt. Remember to type a slash in front of the number and press the enter key.

The Forms title screen appears and asks you to specify the name of the Text file in up to six characters. Type in the name you used when you created the Model 100 Text tape.

If you press the break key, the computer returns to the Cass? prompt. If you do not care what the name of the tape is, press the enter key and Forms reads the first Text file that it finds on the tape.

Then Forms prompts you to prepare to play your Model 100 Text tape and to press the enter key when the recorder is ready.

If the first name on the tape does not match what you request, Forms displays Skip: followed by the file name from the tape. It continues to search for the file with the name you entered.

When it finds the right file, it reads the tape into memory. If the program finds a checksum error, you must respecify the file name.

After reading the Model 100 Text tape, Forms prompts you to prepare to record a Model III Scripsit tape. Place a blank cassette into your recorder. Forms then prompts you to press the enter key when the recorder is ready.

After it writes to tape, Forms asks if you would like to create another copy. Press the Y key to record another copy or press the N key to go on.

Finally, Forms asks you if you would like to read another Model 100 Text tape for conversion. Press the Y key to read another tape. Pressing any other key ends the program and returns the computer to the Cass? prompt.

### Saving Forms

You can load a working copy of Forms at any time by running the loader program. However, if you use the program often, it saves time to have a machine-language system tape that loads directly into your Model III.

If you have a debug or monitor program that lets you write system tapes, you need the start, end, and entry point addresses for your machine. This information is in Table 1.

If you add more memory to your computer, you need to use

| MACHINE | START ADDRESS |         | END ADDRESS |         | ENTRY POINT |         |
|---------|---------------|---------|-------------|---------|-------------|---------|
|         | Hex           | Decimal | Hex         | Decimal | Hex         | Decimal |
| 16K     | 7CD2          | 31954   | 7FFF        | 32767   | 7CD2        | 31954   |
| 32K     | BCD2          | 48338   | BFFF        | 49151   | BCD2        | 48338   |
| 48K     | FCD2          | 64722   | FFFF        | 65535   | FCD2        | 64722   |

Table 1. The start, end, and entry point addresses for saving Forms on 16K, 32K, and 48K Model IIIs.

| Model III<br>Scripsit symbol | Model 100<br>Text symbol | Meaning                |
|------------------------------|--------------------------|------------------------|
| ([) left bracket             | ([) left bracket         | block start            |
| block end                    |                          |                        |
| (>) greater than             | (>) greater than         | block name end; format |
| (■) square block             | ENTER key                | forced end of line     |
| (¶) paragraph                | CODE/0                   | begin new paragraph    |
| (\) backslash                | GRAPH/-                  | forced end of page     |
| (^) caret                    | SHIFT/6                  | copy marker            |

Table 2. Reference chart of Scripsit symbol equivalents and meanings.

the loader program to tailor Forms to the new high memory addresses.

### Guidelines for Use

To take full advantage of Forms, take your Scripsit reference card and the following explanations of equivalents along with your Model 100. Table 2 contains a chart of the symbol conversion for easy reference.

Surround footers and other blocks of text with brackets on the Model III screen. To include these symbols in your text, use the Model 100's left- and right-bracket keys.

Be sure to include the one- or two-character block description code and include a greater-than symbol after the left bracket to terminate the block description or begin a format line.

Create end-of-line symbols on the Model 100 by using the enter key. The black triangle symbol on the Model 100 screen becomes a square block symbol in Scripsit.

You can include the paragraph symbol in your Model 100 text by pressing the code and zero keys together. The same symbol appears in the Scripsit version.

Scripsit uses the backslash character to indicate a forced end-of-page. Enter this symbol on your Model 100 in text by pressing the graph and hyphen keys together.

Copy markers appear as the caret symbol in Scripsit. Enter these symbols into your Model 100 text by typing the shift/6 keys simultaneously. ■

---

*John and Aileen Cornman are the owners of Structured Software Services. You can reach them at 9233 N.E. 269th St., Battle Ground, WA 98604.*

---

## The Searcher

by Carl Oppedahl

The Model 100's built-in Address and Schedule programs scan ADRS and Note files for strings of data you specify, but



they don't work on other .DO files. I wrote a program called Search that redirects the 100's ROM routine so you can search any .DO file using the Address program and display or print records containing a particular word or phrase (see Program Listing 2).

Although you can't modify the machine-language instructions in the ROM, you can modify the action of the routines. The method I use involves jumping to a point in the ROM routine other than the usual entry point and setting the register contents to point to a user-defined file.

**Creating Search**

To develop the Search program, first locate the machine-language entry points for Address and Schedule. You can find the entry addresses for these programs in the system file directory at F962-FA74 hexadecimal (hex). To see this, enter and run this program:

```
5 OPEN "jmp" FOR OUTPUT AS 1
10 FOR A = 63875 TO 63896:C = PEEK(A):PRINT I,A;C;
20 IF C<91 AND C>31 THEN PRINT 1 CHR$(C) ELSE PRINT
30 NEXT A
40 CLOSE
```

| Address | Contents | Table 4. Description of file types. |                     |
|---------|----------|-------------------------------------|---------------------|
| 63875   | 176      | <b>Bit</b>                          | <b>Meaning</b>      |
| 63876   | 104      | 128                                 | 1 if a valid file   |
| 63877   | 91       | 64                                  | 1 if a .DO file     |
| 63878   | 65 A     | 32                                  | 1 if a .CO file     |
| 63879   | 68 D     | 16                                  | 1 if located in ROM |
| 63880   | 68 D     | 8                                   | 1 if invisible      |
| 63881   | 82 R     |                                     |                     |
| 63882   | 83 S     |                                     |                     |
| 63883   | 83 S     |                                     |                     |
| 63884   | 32       |                                     |                     |
| 63885   | 0        | 63891                               | 72 H                |
| 63886   | 176      | 63892                               | 69 E                |
| 63887   | 111      | 63893                               | 68 D                |
| 63888   | 91       | 63894                               | 76 L                |
| 63889   | 83 S     | 63895                               | 32                  |
| 63890   | 67 C     | 63896                               | 0                   |

Table 3. File directory for the Address and Schedule programs.

| Decimal | Hex      | 8085 mnemonic | Z80 mnemonic |
|---------|----------|---------------|--------------|
| 209     | D1       | POP D         | POP DE       |
| 277     | E3       | XTHL          | EX (SP),HL   |
| 195     | C3 FF FF | JMP FFFF      | JP FFFF      |

Table 5. Z80 to 8085 opcode conversion table.

```
20 P$=CHR$(209)+CHR$(235)+CHR$(195)+CHR$(116)+CHR$(91):PRINT"Input file:
";:LINEINPUTF$:F$=LEFT$(F$,6)+".DO"+CHR$(0):FOR I=1 TO LEN(F$):C=ASC(MID$(F$,I,1)):IF C>96 AND C<123 THEN C=C-32
120 POKE 64984+I,C:NEXT I:A=VARPTR(P$):CALL PEEK(A+1)+256*PEEK(A+2),0,64985

Program Listing 2. Search.BA finds specified strings in any .DO file.
```

Table 3 contains a list of the information in the directory. The 3 bytes preceding the file name specify the file type and the load address. You can interpret file types using the information in Table 4.

Address and Schedule have file type 176, which comprises a valid .CO file located in ROM (128 + 32 + 16, see Table 4). The 2 bytes following the file type contain the entry addresses of the Address and Schedule programs. Combine the two numbers as follows to make up a 16-bit binary number.

```
Address: 104 + 256*91 = 23400 (decimal)
Schedule: 111 + 256*91 = 23407 (decimal)
```

In hex code, the entry addresses are 5B68 and 5B6F hex. The code at 5B68-5B72 hex shows that each entry point leads to 5B74 hex, the scanning program that underlies the Address and Schedule programs. The code from ROM is:

```
5B68 LD DE,5CCE
5B6B SUB A,A
5B6C JP 5B74
5B6F LD DE,5D02
5B72 LD A,FF
5B74 start of program
```

Prior to reaching 5B74 hex, each entry point sets the DE register to point to the ASCII string of the file name you want the program to scan. The program sets the accumulator with a flag value to indicate whether the command prompt is Adrs (A = zero) or Schd (A is a nonzero number).

The code at 5CCE hex is ADRS.DO, and the code at 5D02 hex is Note.DO. In each case, a zero (null) follows the file name.

The scanning program starts at 5B74 hex and opens the file name that DE points to, usually ADRS.DO or Note.DO. However, if you put a different value in DE, the program opens that file instead.

You can't use the Basic Call command to call 5B74 because Call doesn't let you set the DE register. Furthermore, to keep the stack in order, you should jump, not call, to 5B74 hex.

To harness the Address and Schedule programs for your own use, set HL to point to the file name you want to scan and execute the following machine code:

```
D1 POP DE
EB EX DE,HL
C3 74 5B JMP 5B74
```

The POP instruction eliminates the return address from the stack because the program never uses it. The EX DE,HL opcode makes the CPU exchange the contents of the DE and HL registers, and is an easy way to get the HL value into the DE register. The program then jumps to the ROM program, which scans the opened file for specified strings.

After writing the Assembly-language program, you can assemble it manually by using the opcode values in Table 5. The decimal values appear in the CHR\$ functions in line 20 of the finished program, Search (see Program Listing 2).

Then the program gets your input, converts it to uppercase, and POKES it into FDD9-FDE1 hex, the part of high memory that contains the file names when the program opens a file.

After setting up the file name, the program provides the location of the executable machine code for a Call command. In Basic, the Call command transfers program control to a

# BASIC

FOR MODELS I AND III

# JUST GOT BETTER

## INTRODUCING BASRUM. THE ONLY LINKING LOADER & LIBRARY MANAGER FOR BASIC.

BASRUM is designed to manage the subroutines you write and use in your Basic programs. Like an efficient filing system, BASRUM saves you time, energy, and space. Best of all, it makes your program writing **FASTER** and **EASIER**.

### HOW DOES IT WORK?

BASRUM lets you store your subroutines in "library" files, separate from the programs that use them. ANY Basic program can then use any routine in any file. You can even load up to 256 routines to a special, protected memory area and use them **AFTER** you load other Basic programs! Naturally, you can replace or delete any routine from memory at any time.

### SUPER BENEFITS!

NOW you can write a subroutine **ONCE**, store it on disk **ONCE** ...but use it in ANY program at ANY time, **AUTOMATICALLY!** Load them once, use them all day. You can **CHAIN** from program to program, keeping the existing subroutines, variables, and file buffers. Your Basic programs, being shorter, will **LOAD FASTER**. **NEED MORE MEMORY?** You can break long programs into shorter modules and "overlay" the modules. BASRUM gives you complete control of **WHAT** is loaded, **WHEN** it is loaded, and **WHERE** it is loaded.

### THE NITTY GRITTY.

BASRUM was designed specifically for the models I and III for use with any DOS that is compatible with TRSDOS<sup>®</sup>. It's overlay structure requires only the top 1600 bytes of memory. There is **NO LIMIT** on the number of working copies.

**SIMPLE ENOUGH FOR THE BEGINNER, VERSATILE ENOUGH FOR THE PRO: IF YOU WRITE PROGRAMS IN BASIC YOU NEED BASRUM.**

**TO ORDER:** Money order, check, VISA, MASTERCARD: \$149.50  
Specify memory size: 16K 32K 48K  
Select disk format:  
Model I: 35 track, single density  
Model III: 40 track, double density

Guaranteed to perform as described or your money back. Orders are shipped via air, most within 24 hours. A complete manual is included. Information only? Send self-addressed, stamped envelope.



by **Wiley, Inc.**

841 Bishop St., Ste. B-2 Honolulu, HI 96813  
(808) 531-4314

TRSDOS is a registered trademark of the Tandy Corp.  
BASRUM is a trademark of Wiley, Inc.

machine-language routine at a specified starting address.

The CPU treats the values stored in memory beginning with the starting address as executable machine-language instructions, whether or not they are in fact executable.

Usually the **LOADM**, **CLOADM**, or **POKE** command puts the machine-code numerical values into RAM, but it doesn't matter to the CPU how the values get there. When using any of these methods, you should protect that portion of RAM by using the **Clear** command so that Basic doesn't overwrite it.

Search demonstrates another means of loading executable machine-language instructions into RAM, specifically the loading of **CHR\$** values into a string variable such as **P\$**. You don't have to use the **Clear** command because Basic protects **P\$** as it protects any other variable. When using the **Call** command you must provide the starting address of the string.

The **VARPTR** function returns the numerical variable addresses, but the information that **VARPTR** gives for string variables is more complicated. Instead of returning the string address itself, **VARPTR** returns the address of the string control table. For example, if **A = VARPTR(P\$)**, the first table entry is **PEEK(A)**, the second is **PEEK(A + 1)**, and so on.

The second and third table entries make up the actual address of the string contents; the first entry in the table is the length of the string. To see this, run this Basic program:

```
10 INPUT FS: A = VARPTR(FS):PRINT PEEK(A),A
20 B = PEEK(A + 1) + 256*PEEK(A + 2)
30 FOR AD = B TO B + PEEK(A) - 1
40 PRINT CHR$(PEEK(AD)):NEXT:PRINT:END
```

Here, **A** is the address of the string table and **B** is the address of the first character in the string.

### Running the Program

Program Listing 2 contains Search, the finished program. This program lets you use the searching capabilities of the Address program to find strings in your **.DO** files.

You can run the program from Basic or from the main menu. The program requests an input file. Respond by typing the file name without its extension (the program automatically supplies the **.DO** extension).

If the program doesn't find the file, it displays an error message similar to the one you see if you select the Address program without having created an **ADRS.DO** file.

When the program finds the file, it opens it and displays key labels and a prompt. The available function keys are the same as for Address and Schedule: **Find (F1)**, **Lfnd (F5)**, and **Menu (F8)**.

Pressing the **F8** key returns control to the main menu. If you press the **F1** key and type in a search string, the program selects lines that contain the search string from the input file. Press the **F1** and enter keys to display all lines of the input file.

When the screen isn't large enough to display all lines selected, press the **F3** key to display the next screen or press the **Quit** key (**F4**) to return the computer to the **Find Lfnd Menu** prompt. The **Lfnd** key works like the **Find** key, except that output goes to the printer instead of the screen.

*This article is adapted from Carl Oppedahl's Advanced Programming for the Model 100, to be published soon by Wayne Green Books. You can reach Oppedahl at 99 Park Ave., New York, NY 10016.*

Getting Personal

by Richard Ramella

Gone are the days when a computer-written letter with inserted "personal" information might have impressed you:

YES, YOU—\*\*\*MS. MANDY BRANDY\*\*\*—MAY ALREADY BE A WINNER OF PRIZES TO ENJOY AT 323 S. BIG SANDY...

Recognizing the cynicism inherent in some letters of this ilk, I offer HIPAL.BA (see Program Listing 3) with cautions and suggestions useful to both sender and receiver.

HIPAL is an 8K RAM Model 100 program that prints a standardized message to many addressees. It requires 11½-inch fanfold paper. I've stored the addresses in a text (.DO) file (see Fig. 1), and put the letter in several strings within the program.

When you run HIPAL, it takes the facts relevant to each addressee in turn and plugs them into the template letter, including the address and a proper salutation line. The program spaces to the succeeding page and repeats the process as long as it finds a new addressee.

Simple Changes

I've thrown in a few mirror tricks to illustrate how to personalize a letter. If you need to send an identical message to all addressees, with individualized addresses and salutations, change line 890 to LPRINT SP and delete lines 800, 960-980, and 990. They are peculiar to the program as presented; I'll explain their function later.

Each address has eight fields whose numbers come into play in the program (see Table 6). Set up the address list in the .DO file just as I've done in Fig. 1: seven lines of information and a space.

The amounts in the sixth and seventh lines of each address

Figure 1. Addresses saved to .DO file.

```
Mr.                Westchester, AL 45678
Harlow             125
Barlow            125
1234 Marlowe Ct #3
Marlborough, SD 34567 Ms.
125               Alice
10.72            Chalice
                  4890 Callous Way
Ms.               Dallas, OH 21893
Mandy R.          30
Brandy            5
323 S. Big Sandy
The Andes, SA 23456 end
500
200
```

| Field | Description                |
|-------|----------------------------|
| 1     | Title                      |
| 2     | First name                 |
| 3     | Last name                  |
| 4     | Address                    |
| 5     | City-state-zip             |
| 6     | Amount                     |
| 7     | Amount                     |
| 8     | Space created by line feed |

Table 6. Address fields.

(see Fig. 1) represent money pledged and the amount received to date, respectively. You can leave these lines blank in your applications (line feeds), or use them to designate something else—the amount of a bill and the money paid in the past month, for example.

My .DO file containing addresses (Fig. 1) ends with two line feeds and the word "End", though two line feeds suffice. The B(1) to B(15) string array running from lines 370-510 (see Program Listing 3) contains the elements of the letter sent to each addressee. Lines 450-510 have null values to remind you that your message can continue through B(15).

When you store HIPAL for later use, assign all the B(X) arrays a null value, as in 370 B(1)=" ". This makes them

Program Listing 3. HIPAL.BA.

```
100 REM * HI PAL: Personalized Mass Mail
*TRS-80 Model 100 8K/ Richard Ramella *
110 REM * REQUIRES PRINTER *
120 CLS
130 CLEAR 1000
140 PRINT @ 95, "HI PAL"
150 LINE(86,12)-(127,26),1,B
160 PRINT @ 165, "Personalized Mass
Mailing"
170 PRINT@ 245, "Tap enter to continue";
180 INPUT X
190 CLS
200 DEFSTR A-D,S
210 DIM B(15)
220 INPUT "Left margin";Q
230 INPUT "Right margin";R
240 SP=SPACE$(Q)
250 S=SPACE$(1)
260 PRINT "Type date and year separated
by comma"
270 PRINT
280 PRINT SPACE$(13);CHR$(153)
290 PRINT "as in April 1,1985 and press
enter."
300 PRINT SPACE$(6);STRING$(7,"-")
;SPACE$(1);STRING$(4,"-")
310 PRINT SPACE$(13);CHR$(152)
320 INPUT DT,DY
330 CLS
340 PRINT "Type file name for letter
list and press enter"
350 INPUT F$
360 OPEN F$ FOR INPUT AS 1
370 B(1)="The work of the Fraudulent
Foundation for Frivolous Grants to
Computer Article Writers progressed
significantly in the past year due in no
small measure to contributions such as
your own. Your particular pledge was a
special joy to us at 3F."
380 B(2)="Your pledge was deeply
appreciated. You are among our most
valued contributors."
390 B(3)="The need is now greater than
ever. Perhaps now is the time to send a
check for the remainder of your pledge.
"
400 B(4)="Your pledge to this worthy
cause was"
410 B(5)="and we have received"
420 B(6)="May prosperity and a
```

Listing 3 continued

ready to accept your own material.

When you want to load a letter into the program, run HIPAL.BA, press the shift-break keys simultaneously, type in EDIT 370-510 and tap the enter key. In the edit mode, typing the letter as succeeding values of B(1) to B(15) is simple. When the values are loaded, press the F8 key. If you put more than 255 characters in any string, you'll see this error message:

Text ill-formed  
Press space bar for TEXT

Tapping the space bar takes you back to edit mode to correct the mistake.

### Line by Line

Lines 220 and 230 let you choose left and right margin widths. Experiment to get the correct margins for your printer. Later, you'll use the values chosen here to determine a line's default character. If you get strange results, you may have made strange margin choices.

An appropriate choice would be the following: If your printer offers 12 characters per inch, printing 85 characters in 8½ inches, choose left and right margins of 10 to yield a line of about 65 characters, centered on the page.

Line 240 assigns Q the value of the left margin, so that SP spaces that far to the right with each line feed.

Line 340 asks you to type the name of the .DO file containing the addressee. You'll get an FC error if you request a non-existent file and strange results if you ask for a file that doesn't have addresses formatted as in Fig. 1.

Line 360 opens the address list .DO file, and lines 370-510 contain the elements of the letter.

Lines 530-570 input each of the address lines defined earlier and put them in the array C(1) to C(8). Line 580 displays the name of the current addressee.

### Printing the Letter

After you turn on your printer, position the top perforation of the first page at the top of the print head. Printers vary, so experiment with positions if necessary. Lines 590-610 produce 12 line feeds, taking my print head down about two inches from the top of the page—enough room to clear most printed letterheads.

Line 620 prints the date you specified and positions its final character at the right margin. Lines 620-650 produce five line feeds to space down to the address area. Lines 660-680 print the current address. Strings C(1) to C(5) are, in turn: title, first name, last name, address, and city-state-zip code. Lines 690-710 space down five more lines.

Line 720 assigns NMS\$ the addressee's title, a space, and a last name, so line 730 can print the salutation "Dear Mr. Barlow" before spacing twice to begin the message (see Fig. 2).

Line 760 assigns PG a value of 29, the line the letter has reached on the page; the program counts the lines so it can space correctly to the next page when the current letter is complete.

Lines 770 and 780 assign numeric values to the sixth and seventh lines of the address. This particular application doesn't require these amounts, but I'll suggest ways you can manipulate them for your own purposes.

Lines 790-1040 make up a For...Next loop that travels through all the array strings holding the message being printed. Lines 800 and 960-990 are peculiar to my letter. In line 810 the program skips to the next value of B(G) if the current value is zero.

Listing 3 continued

```
charitable mien remain with you in the
months ahead -- months that well could
be difficult for many clients of the
Foundation."
430 B(7)="Thank you,"
440 B(8)="qsign Richard Ramella"
450 B(9)=""
460 B(10)=""
470 B(11)=""
480 B(12)=""
490 B(13)=""
500 B(14)=""
510 B(15)=""
520 CLS
530 FOR E=1 TO 8
540 IF EOF(1) THEN 1100
550 LINE INPUT #1,CA
560 C(E)=CA
570 NEXT E
580 PRINT C(1);S;C(2);S;C(3)
590 FOR T=1 TO 12
600 LPRINT
610 NEXT T
620 LPRINT SPACE$(85-(Q)-2-(LEN(DT)+LEN(
DY)));DT;" ";S;DY
630 FOR T=1 TO 5
640 LPRINT
650 NEXT
660 LPRINT SP;C(1);S;C(2);S;C(3)
670 LPRINT SP;C(4)
680 LPRINT SP;C(5)
690 FOR T=1 TO 5
700 LPRINT
710 NEXT T
720 NMS=C(1)+S+C(3)
730 LPRINT SP;"Dear";S;NMS": "
740 LPRINT
750 LPRINT
760 PG=29
770 PD=VAL(C(6))
780 GI=VAL(C(7))
790 FOR G=1 TO 15
800 IF G=3 AND GI=>PD THEN G=6:PRINT S;
810 IF B(G)="" THEN 1040
820 IF INSTR(B(G),"qsign")=0 THEN 890
830 PG=PG+4
840 FOR T=1 TO 3
850 LPRINT
860 NEXT T
870 LPRINT SP;RIGHT$(B(G),LEN(B(G))-6)
880 GOTO 1050
890 IF G<>5 THEN LPRINT SP;
900 FOR H=1 TO LEN(B(G))
910 D=MID$(B(G),H,1)
920 LPRINT D;
930 LL=LL+1
940 IF LL>85-(Q+R) AND D=CHR$(32) THEN
SP=SPACE$(Q): LPRINT: LPRINT SP;; PG=PG+
1: LL=0
950 NEXT H
960 IF G=2 THEN LPRINT S;NMS". ";
970 IF G=4 THEN LPRINT S"$C(6);S;;
GOTO1040
980 IF G=5 THEN LPRINT S"$C(7);". ";
990 IFG=1 THEN LPRINT S;;SP="":GOTO1040
1000 LL=0
1010 LPRINT
1020 LPRINT
1030 PG=PG+2
```

Listing 3 continued



Line 820 uses an instring test to see if the configuration `qsign(cq)` appears in the current string, and, if not, jumps to line 890. If lines 830-880 execute, the program has found `qsign`. I gave `B(8)` in line 440 the string value `qsign Richard Ramella`. The signature of each letter you write should comprise the following format: `qsign`, a space, and the writer's name. When it finds `qsign`, the program prints the name only and jumps to line 1050 for the sequence that spaces to the next page.

Lines 900-950 are a `For...Next` loop within the larger loop. Line 900 starts the loop, with `H` equal to the first to last character positions of the array string being printed. Line 910 assigns `D` string the single character encountered within the `B(G)` string; line 920 prints that character. Line 930 counts the number of characters per line and always adds one for the character just printed.

Line 940 determines if the letter needs a line feed and, if so, provides it. Its logic is: If current line length is over 85 characters long (the number of characters possible on the page) minus values of left plus right margin (`Q + R`), and the current value of the `D` string is a space, then restore the value of `SP` to `SPACE$` (left margin), line feed once, print the left margin value, increment `PG` by one to keep the page number current, and set the line length value at zero. With this, the `For...Next` loop ends.

Once the program prints the current `B(G)` array string, lines 1000-1040 set the line length at zero, invoke two line feeds, increment `PG` twice to keep track of the number of lines on the current page, and end the loop.

Lines 1050-1070 space to the next page. On the printer I used to test this program, 66 lines fit on one 11½-inch page. Spacing forward 66, minus the number of lines on the current page, takes you to the top of the next page. This method ensures correct spacing forward, no matter what the length of your letter, so long as it fits on one page. The program also spaces correctly in situations where one run produces varying lengths of letters to different addressees. If either the 85 or 66 value does not work with your printer, see the end of the article for remedies.

Lines 1100-1140 produce a beep, denoting the end of the program.

### Personalizing Mail

The lines I've ignored so far produce personalization effects within the program. To get an idea of what can happen, scan the message of `B(1)` to `B(5)` in lines 370-510.

Line 800 changes the letter's format. It first tests if the third paragraph, `B(3)`, is due to be printed. If so, and if `GI` (the amount given) equals or exceeds `PD` (the amount pledged), paragraphs `B(3)`, `B(4)`, and `B(5)` aren't needed in the letter (see Fig. 3). The program assigns `G` a value of 6, so it skips these paragraphs; the letter resumes without mentioning the balance of the pledge.

Consider another situation: The letter is a bill. The sixth and seventh lines in the `.DO` file address (see Fig. 1) represent the amount owed at the start of the month and the payment made at the end of the month, respectively. At the point in the letter where you want to print this information, give a `B(X)` array the value of `QP`; for example, `B(3) = "We received your April payment of '$PD' and applied it to your balance of '$PG'". Your new balance is '$PG - PD'."`

Line 960 adds the addressee's title and name to the paragraph. If the program printed `B(2)`, ending "...You are

Listing 3 continued

```
1040 NEXT G
1050 FOR J=1 TO 66-PG
1060 LPRINT
1070 NEXT J
1080 LL=0
1090 GOTO 530
1100 PRINT "Run complete."
1110 PRINT "Tap a key to end."
1120 BEEP
1130 IF INKEY$="" THEN 1120 ELSE END
1140 END
```

End

April 15, 1984

Mr. Barlow Barlow  
1234 Marlowe Ct #3  
Marlborough, SD 34567

Dear Mr. Barlow:

The work of the Fraudulent Foundation for Frivolous Grants to Computer Article Writers progressed significantly in the past year due in no small measure to contributions such as your own. Your particular pledge was a special joy to us at 3P. Your pledge was deeply appreciated. You are among our most valued contributors. Mr. Barlow.

The need is now greater than ever. Perhaps now is the time to send a check for the remainder of your pledge.

Your pledge to this worthy cause was \$125 and we have received \$18.72.

May prosperity and a charitable mien remain with you in the months ahead -- months that well could be difficult for many clients of the Foundation.

Thank you,

Richard Ramella

Figure 2. Sample letter as formatted by `HIPAL.BA`.

April 15, 1984

The Rev. Chester P. Lester  
79 Veeblefester Ave.  
Westchester, AL 45678

Dear The Rev. Lester:

The work of the Fraudulent Foundation for Frivolous Grants to Computer Article Writers progressed significantly in the past year due in no small measure to contributions such as your own. Your particular pledge was a special joy to us at 3P. Your pledge was deeply appreciated. You are among our most valued contributors. The Rev. Lester.

May prosperity and a charitable mien remain with you in the months ahead -- months that well could be difficult for many clients of the Foundation.

Thank you.

Richard Ramella

Figure 3. Modification in sample letter.

among our most valued contributors, ...", it adds the personalized material "Mr. Barlow."

Some titles in Fig. 1 are inappropriate for salutations: otherwise chatty letters would refer to "Dr. Gunnoy III" and "The Rev. Lester" (see Fig. 3). Edit your address list so as not to reveal your clever letter as a computer trick. For example, rewrite "The Hon. James Mixworth" as "Judge Mixworth" and drop generational references such as "Jr." and "III" whenever possible.

Lines 970 and 980 work in tandem. They produce the sentence combining `B(4)`, `B(5)`, and address fields 6 and 7: "Your

## C•NOTES

pledge to this worthy cause was \$100 and we have received \$10."

Line 990 adds the material of one string array to that of the previous one within the same paragraph. Its logic goes: If  $G = 1$ , then the first string array has printed; to combine the second paragraph, print a space, then make the SP string null so no margin will be incorrectly printed, and go to line 1040 to continue that next array.

### Final Hints

The paper spacing will be thrown if your letter is longer than one page.

If a two-inch top margin is insufficient, start the printing by positioning the print head below the top of the page.

If your printer won't print 12 characters per inch, giving you 85 characters across  $8\frac{1}{2}$  inches, run a test to find out how many characters it does yield. To assure correct margins, use that number in place of 85 in lines 620 and 940.

Follow the same procedure if the number of lines per page your printer can print differs from my line count; substitute that number for 66 in line 1050 to assure correct spacing to the next page. ■

*If you can't get HIPAL to run correctly, send a self-addressed, stamped envelope, a printed listing of the program as it exists in your system, and a run example to Richard Ramella, 1493 Mt. View Ave., Chico, CA 95926.*

# THE ENTREPRENEUR'S SPREADSHEET OFFERS MODEL 100 AND \*NEC PC-8201 USER MORE POWER.

- Easy to use
- Budgeting
- Sales Projections
- Financial Forecasting
- Many Other Business Uses

The Entrepreneur's Spreadsheet is a powerful 16-column by 50-row financial management tool. This flexible software has full computation capabilities including summation, exponents, and algebraic equations. Other features are expanded command set, creation of printed reports and ease-of-use. It is completely documented with sample session. This allows you to be up and running right away. Full telephone support for this versatile package is offered.

**124.<sup>95</sup>**

Also available:  
The Entrepreneur's General Ledger **\$119.<sup>95</sup>**

When you order specify computer make & model. Phone, Visa or Master Charge number or send cash or money order. Your order will be shipped within 24 hours.

## JAMES W. SECOR

Data Processing Consultants

2050 N. Collins, #101, Richardson, Tx 75080  
214/699-3649



\*NEC Corporation Tokyo Japan

254

# ALWAYS AT SALE PRICES TRS-80<sup>®</sup> ALWAYS AT SALE PRICES

OUR TRS-80s Have the Radio Shack warranty

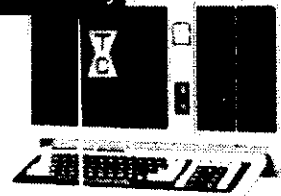


HARD DISK DRIVES

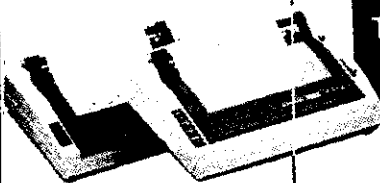


**WE SHIP FAST!** —282

DFW COMPUTER CENTER  
326 MAIN ST., GRAPEVINE  
TEXAS 76051 — (817) 481-7283

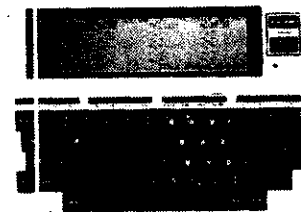


MODEL 12

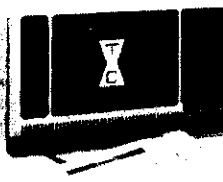


DOT MATRIX PRINTERS  
DAISY WHEEL PRINTERS

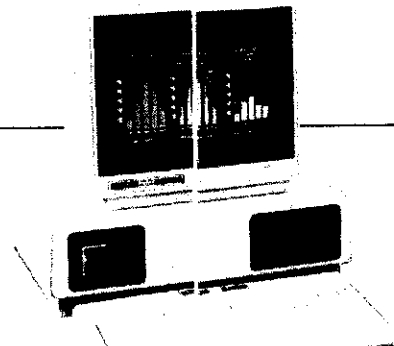
To get YOUR SALE price CALL:  
**1-800-433-SALE**



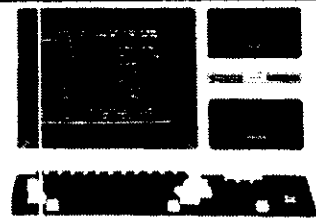
MODEL 100



MODEL 4-P



MODEL 2000



MODEL 4

## ReWrite

I have discovered two problems in line 15 of my program *Writer.BA* ("Write Now," February 1984, p. 191). First, my printer doesn't print the Model 100's down-arrow graphics character. To solve the problem, set the variable *NPAGES* equal to the down-arrow character (*CHR\$(153)*) instead of the null string character.

Also, substitute *I* for the *l* in the variable *LIne\$*. *Line* is a Model 100 function, so using the normal spelling gives you a syntax error.

After you make these changes, line 15 should appear as follows:

```
15 NPAGES=CHR$(153);CHAR$="";LINE$=""
```

*Ronald F. Balonis*  
118 Rice St.  
Trucksville, PA 18708

In Ronald Balonis's "Write Now," *Writer.BA* loses part of a word when starting a new page. To correct the problem, change line 440 to:

```
IFEOF(1)THENCLOSEIELSELCNT=0:PN=PN+1:GOTO210
```

I also added lines 301-304 to recognize an embedded command line and adjust the left margin while printing the Text file (see the Program Listing). Note that the printer expressed the caret character as a left bracket in line 305.

To make these changes, first terminate the present line by pressing the enter key, then add a command line by typing *>nn* and pressing the enter key.

The *nn* following the right arrow can be any one- or two-digit number from zero to *RM-1*. You can adjust the left margin as often as you like. The last setting holds until you change it.

*Jody Nelis*  
132 Autumn Drive  
Trafford, PA 15085

```
301 IFCHAR$<>">"THEN305
302 CHAR$=INPUT$(1,1):IFCHAR$=CR$THEN304
303 T2$=T2$+CHAR$:GOTO302
304 LM=VAL(T2$):LGTH=RM-LM:T2$="":GOTO300
305 IFCHAR$<>"["THEN310
306 IF LCNT<>TM THEN410
307 GOTO300
```

*Program Listing. Enhancement for Writer.BA.*

## Getting Through

I'd like to send Model 100 text files to my Model III disk to edit them with *SuperScript*. I can get the Model 100 to send the files, but I can't make the Model III receive them. Can anyone give me information on how to set this up?

*Jerel Peterson*  
Box 182  
Cando, ND 58324

*There's a skeleton terminal program on p. 63 of the Model III disk system manual—anyone with other ideas?—Eds.*

## COM/CMD Fix

My compliments to Ronald F. Balonis for his fine communications program for data transfer between the Model 100 and the Model I/III ("100 Disks," January 1984, p. 171).

However, in Model I's with the Radio Shack double-density board and *TRSDOS 2.7DD*, *COM/CMD* doesn't work. This is because *TRSDOS 2.7DD* places the command line in a buffer at 44DA hexadecimal (hex) rather than the standard 4318 hex. As a result, *COM/CMD* always prints the message: \* COMMAND LINE ERROR \*.

To fix the problem, change line 410 to read:

```
410 CMD EQU 44DAH
```

If you've already assembled the program, patch it as follows:

```
PATCH COM/CMD (R=1,B=20,F=1E43,C=E044)
PATCH COM/CMD (R=1,B=28,F=1C43,C=DE44)
```

*Carl Oppedahl*  
99 Park Ave.  
New York, NY 10016

## Data Manager Debug

In my article, "Account for Yourself" (March 1984, p. 187), my *Data Manager* program contains a small error. Change line 2000 to read:

```
2000 IF FL$= "" THEN 2200
```

*Mark Hickenbottom*  
28 Wrangler Court  
Chico, CA 95926

## Shooter Debug

Line 105 of my *Shooter.BA* program ("A New Pair of Shoes," March 1984, p. 194) contains a typographical error. Change 67.7% to 66.7%.

*Ronald F. Balonis*  
118 Rice St.  
Trucksville, PA 18708

C-NOTE FOR JULY 1984

## Portable Scripsit

by John and Aileen Cornman

You use Model III Scripsit regularly and you wish your documents were as portable as your Model 100. With the help of CScrip they can be (see Program Listing 1). CScrip translates Model III Scripsit tapes into ones the Model 100 can read, converts Scripsit's text boundary markers to equivalent Model 100 characters, and adds the extra line feeds that the Model 100 expects after each carriage return.

Later, you can use Forms (June 1984, p. 156) to load your document back into Model III Scripsit with all the formatting characters intact.

### Installing CScrip

Lines 10-700 are CScrip's loader section. The loader automatically adjusts CScrip's machine-language instructions in accordance with the amount of memory you have. Consequently, you can use the program on 16K, 32K, and 48K machines.

The loader also helps you find mistyped hexadecimal (hex) codes in the data statements and computes a checksum to ensure their accuracy.

To use the loader, first answer the memory size question correctly: 32100 for a 16K machine, 48484 for a 32K machine, and 64868 for a 48K machine.

Now run the program. If you answered the memory size prompt incorrectly, the program reminds you of the correct responses and asks you to try again.

The loader asks you to wait while it reads the hex codes in the data statements and loads them into memory. If the program finds an unacceptable hex character in a data statement, it shows you the bad character pair and tells you the line number and hex pair you need to correct before rerunning the loader. Each data line has 18 pairs of characters except the last one, which has 12.

Even if all the hex codes are acceptable, they might be incorrect. The loader informs you if the hex codes in the data statements don't add up to the predetermined checksum total. With this kind of error, the loader can't give you further information to help find the incorrect code. You have to recheck the data statements against the listing to find the error before rerunning the loader.

If it doesn't detect any errors, the loader tells you which entry address to use with the System command when you run CScrip. Now you're ready to convert Model III Scripsit tapes to Model 100 format.

### Using CScrip

To use CScrip, begin by entering the System command. At the asterisk prompt, enter /32101 for a 16K machine, /48485 for a 32K machine, and /64869 for a 48K machine. Be sure to type the slash before and press the enter key after the number.

Next, the CScrip title screen appears and asks you to specify the one- to six-character name of the Text file. This is the name

you give to the Model 100 Text tape. If you press the break key at this point, you return to the "Cass?" prompt.

CScrip prompts you to prepare to play a Model III Scripsit tape and to press the enter key when the recorder is ready. Then the program reads the Scripsit tape into memory. If it detects a checksum error, you return to the title screen.

After CScrip successfully reads your tape, it prompts you to prepare to record a Model 100 Text tape. Place a blank cassette in your recorder; CScrip prompts you to press the enter key when the recorder is ready.

After the program writes the tape, CScrip asks if you would like to make another copy. If you answer Y, it again asks you to press the enter key when the recorder is ready.

If you press N, CScrip asks if you want to read another Scripsit tape for conversion. If you press Y, you return to the title screen; otherwise the program ends and you return to the "Cass?" prompt.

### Saving CScrip

You can load CScrip at any time by running Listing 1, but if you use the program often, it saves time to load a machine-language System tape directly into your Model III. If you have a Debug or Monitor program that lets you write System tapes, you need the information in Table 1.

If you ever add more memory to your computer, you'll need to use the loader program to tailor CScrip to the new high-memory addresses.

### Guidelines

To take full advantage of CScrip, you should take your Scripsit reference card and the following explanation of equivalents along with your Model 100 (see Table 2).

The Model III surrounds headers, footers, and other text blocks with on-screen brackets. To include these symbols in your text, use the left ([]) and right (]) bracket keys on the Model 100 keyboard. Be sure to include the one- or two-character block description code and the greater-than (>) symbol

| Machine | Start Address |         | End Address |         | Entry Point |         |
|---------|---------------|---------|-------------|---------|-------------|---------|
|         | Hex           | Decimal | Hex         | Decimal | Hex         | Decimal |
| 16K     | 7D65          | 32101   | 7FFF        | 32767   | 7D65        | 32101   |
| 32K     | BD65          | 48485   | BFFF        | 49151   | BD65        | 48485   |
| 48K     | FD65          | 64869   | FFFF        | 65535   | FD65        | 64869   |

Table 1. System tape information.

### The Key Box

All this month's programs run in 8K RAM. A dot-matrix printer is optional for "Etch-A-Screen." "Portable Scripsit" requires a 16K RAM Model III, a cassette recorder, Cassette Basic, and Scripsit.



after the left bracket to terminate the block description or begin a format line.

Create end-of-line symbols on the Model 100 by using the enter key as usual. The triangle symbol will appear on the Model 100 screen, but when you load the file into Scripsit, it will look like the normal block symbol (■).

You can use the paragraph symbol (¶) in your Model 100 text by pressing the code and zero keys together. The same symbol will appear in the text's Scripsit version.

Scripsit uses the backslash character (\) to indicate a forced end-of-page. You can enter this symbol in your Model 100 text by pressing the graph and hyphen keys together.

Copy markers appear as the caret symbol (^) in Scripsit. You can enter carets into your Model 100 text by pressing shift-6. ■

Contact John and Aileen Cornman at Structured Software Services, 9233 N.E. 269th St., Battle Ground, WA 98604.

| Scripsit symbol | Model 100 keyboard | Meaning                |
|-----------------|--------------------|------------------------|
| [ Left bracket  | [ Left bracket     | Block start            |
| ] Right bracket | ] Right bracket    | Block end              |
| > Greater than  | > Greater than     | Block name end; format |
| ■ Square block  | Enter key          | Forced end-of-line     |
| ¶ Paragraph     | Code-zero          | Begin new paragraph    |
| \ Backslash     | Graph/-            | Forced end-of-page     |
| ^ Caret         | Shift-6            | Copy marker            |

Table 2. Symbol reference chart.

Listing 1 continued

```

580 PA1 = BA * 256 + 101          'PA1 = START ADDRESS FOR POKING
590 IF PA1 > 32767 THEN PA = -1 * (65536 - PA1) ELSE PA = PA1
600 RETURN 'PA ADJUSTED IF PA > 32767
620 PRINT#4256,"Before running CSCRIP loader, you must
answer Memory Size?"
630 PRINT"question according to the following table:" : PRINT
640 PRINT"Machine      Memory Size?"
650 PRINT" 16K      32100"
660 PRINT" 32K      48484"
670 PRINT" 48K      64868"
680 PRINT:PRINT"Please press the orange reset button and answer th
e"

690 PRINT"Memory Size? question again before re-running this progra
m."
700 GOTO700
800 DATA CD,C9,01,21,15,3C,22,20,40,21,FB,*1,CD,1B,02,21,50,3C
802 DATA 22,20,40,21,11,*2,CD,6F,*1,3E,03,32,14,42,CD,77,*1,DA
804 DATA 00,00,AF,32,11,42,21,30,*2,CD,1B,02,21,43,*2,CD,1B,02
806 DATA CD,6F,*1,CD,65,*1,F3,21,50,44,36,00,CD,96,02,CD,35,02
808 DATA 47,0E,00,CD,5D,*1,FE,80,20,05,36,0D,23,3E,0A,77,23,10
810 DATA F0,CD,35,02,E9,C2,4E,*1,2B,78,*2,B7,20,DD,CD,F8,01,21
812 DATA 50,44,7E,B7,28,2E,FE,97,20,06,3E,5B,77,23,13,2,FE,9B
814 DATA 20,04,3E,5D,18,F4,FE,*2,CD,1B,02,21,F9,*2,06,06,CD,40,00
816 DATA 3E,5C,18,E4,FE,8B,20,04,3E,5E,18,DC,EG,7F,18,08,CD,C9
818 DATA 01,21,A3,*2,CD,1B,02,21,CD,*2,CD,6F,*1,3E,40,32,11,42
820 DATA CD,65,*1,CD,9E,*1,CD,F8,01,21,6C,*2,CD,6F,*1,CD,49,00
822 DATA FE,59,28,EA,FE,79,28,E6,21,89,*2,CD,6F,*1,CD,49,00,F5
824 DATA CD,C9,01,F1,FE,59,CA,85,*0,FE,79,CA,85,*0,CD,00,00,CD
826 DATA F8,01,CD,C9,01,21,E9,*2,CD,6F,*1,C3,85,*0,CD,35,02,F5
828 DATA 81,4F,F1,C9,21,48,*2,CD,6F,*1,CD,49,00,C9,CD,1B,02,2E
830 DATA CD,1B,02,C9,21,31,*2,CD,1B,02,21,F9,*2,06,06,CD,40,00
832 DATA 08,48,06,00,09,3E,06,91,C8,47,3E,24,77,23,10,FC,C9,F5
834 DATA 81,4F,F1,CD,64,02,C9,21,00,00,CD,87,02,01,00,06,3E,9C
836 DATA CD,64,02,21,F9,*2,7E,CD,96,*1,23,10,79,06,0A,3E,AA,CD
838 DATA 96,*1,10,FB,79,ED,44,CD,64,02,21,50,44,01,00,28,CD,60
840 DATA 00,85,21,00,00,CD,87,02,E1,3E,8D,CD,84,02,71,87,20,0E
842 DATA 3E,1A,CD,96,*1,10,FB,79,ED,44,CD,64,02,C9,CD,96,*1,23
844 DATA 10,EB,79,ED,44,CD,64,02,1B,CD,53,43,52,49,50,53,49,54
846 DATA 20,74,6F,20,4D,6F,64,65,6C,20,51,30,30,0D,62,70,20,53
848 DATA 74,72,75,63,74,75,72,63,64,20,53,6F,66,74,77,61,72,65
850 DATA 20,53,65,72,76,69,63,65,73,0D,45,6E,74,65,72,20,54,45
852 DATA 50,54,20,66,69,6C,65,20,6E,61,6D,65,3A,20,01,50,72,65
854 DATA 73,73,20,45,4E,54,45,52,20,77,68,65,6E,20,72,65,63,6F
856 DATA 72,64,65,72,20,69,73,20,72,65,61,64,79,2E,00,57,72,69
858 DATA 74,65,20,61,6E,6F,74,64,65,72,20,63,6F,70,73,3F,20,28
860 DATA 59,20,6F,72,20,4E,29,0D,52,65,61,64,20,61,20,6E,65,77
862 DATA 20,74,61,70,65,3F,20,28,59,20,6F,72,20,3E,29,0D,50,72
864 DATA 65,70,61,72,65,20,74,6F,20,03,70,6C,61,79,20,4D,6F,64
866 DATA 65,6C,20,49,49,49,20,53,43,52,49,50,53,49,54,20,74,61
868 DATA 70,65,2E,0D,72,65,63,6F,72,64,20,4D,6F,64,65,6C,20,31
870 DATA 30,30,20,54,45,58,54,20,74,61,70,65,2E,0D,43,68,65,63
872 DATA 68,73,75,6D,20,65,72,72,6F,72,2E,0D
    
```

End

Program Listing 1. CScrip.

```

10 CLS:DEFINT A-Z
20 PRINTTAB(19)"SCRIPSIT to Model 100 Loader"
30 PRINTTAB(17)"by Structured Software Services":PRINT
40 PRINT:PRINT"Loading CSCRIP into memory; please wait..."
50 GOSUB 553 'CHECK MEMORY SIZE AND SET UP ADDRESSES
60 POR1=1T0660
70 READ HXS 'GET A PAIR OF HEX DIGIT'S
80 DS = LEFT$(HXS,1) 'LOOK AT THE LEFT DIGIT
90 IF DS = "*" THEN 210 '* HEANS ADDRESS ADJUSTMENT
100 GOSUB 500 'CONVERT HEX DIGIT TO DECIMAL
110 IF D = -1 THEN 400 '-1 IS FLAG FOR BAD HEX CHAR.
120 DC = D * 16 'DC GETS VALUE OF LEFT DIGIT
130 DS = RIGHT$(HXS,1) 'LOOK AT THE RIGHT DIGIT
140 GOSUB 500 'CONVERT RIGHT DIGIT
150 IF D = -1 THEN 400
160 DC = DC + D 'DC = THE DECIMAL CONVERSION
170 CS1 = CS1 + DC 'ADD IT TO THE CHECKSUM
180 POKE PA,DC 'LOAD IT INTO MEMORY
190 IF PA < 32767 THEN PA = PA + 1 'ADVANCE POKE ADDRESS
200 GOTO 300 'READY FOR NEXT HEX PAIR
210 DS = RIGHT$(HXS,1) 'DS = RIGHT DIGIT OF *-PAIR
220 GOSUB 500
230 IF D = -1 THEN 400
240 CS1 = CS1 + D 'ADD TO CHECKSUM
250 DC = BA + D 'BASE ADDRESS + DISPLACEMENT
260 GOTO 180 'GO POKE ADJUSTED ADDRESS
300 NEXT
310 IF CS1 <> 60622 THEN 350 '*VERIFY CHECKSUM TOTAL
320 PRINT:PRINT"CSCRIP has been successfully loaded."
330 PRINT:PRINT"Entry address for SYSTEM command is:";PA1
340 PRINT:PRINT"You may now convert SCRIPSIT tapes to Model 100 fo
rmat." : PRINT : END
350 PRINT:PRINT"Checksum error."
360 PRINT:PRINT"Please correct contents of DATA lines 800 - 872 an
d rerun." : PRINT : END
400 LN = INT((I-1) / 18) 'RELATIVE LINE # OF ERROR
410 CH = I - (LN * 18) 'BAD CHARACTER PAIR #
420 LN = 803 + LN * 2 'ACTUAL DATA LINE #
430 PRINT:PRINT"Invalid hex character: ",HXS
440 PRINT:PRINT"Please correct pair #";CH;"in line";LN;"and rerun.
"
450 PRINT : END
500 D = ASC(DS)
510 IF D > 47 AND D < 58 THEN D = D - 48 : RETURN '0-9 OK
520 IF D > 64 AND D < 71 THEN D = D - 55 : RETURN 'A-F OK
530 D = -1 : RETURN 'BAD HEX CHARACTER
550 BA = PEEK(16562) 'GET BASE ADDRESS
560 IF BA = 253 OR BA = 189 OR BA = 125 THEN 570 ELSE 620
570 IF PEEK(16561) <> 98 THEN 620
    
```

Listing 1 continued

Sort Order

by Ronald F. Balonis

Don't put up with unorganized data files any longer. Use my utility program, Sort.CO, to alphabetize files, beginning at any point you specify. This machine-language program requires only 988 bytes of memory and sorts your RAM files in a matter of seconds.

Sort.CO performs an in-memory sort that physically moves your file's records. However, there are file constraints: You can have no more than 254 records, and they must have equal lengths of fewer than 254 bytes.

Preparing to Sort

Program Listing 2 is the Basic boot of Sort.BA, which, when run, creates Sort.CO. I formatted the data lines so you can read and debug them. Type in Listing 2 and save it before running it.

Line 50 reserves memory for Sort.CO. Lines 200-300 read and POKE the data statements into memory, summing each for a checksum. (Pay heed to the checksum; don't run the program if it's in error, as errors are hazardous to the Model 100's files.) If the checksum equals 95393, then the data statements are correct; line 500 saves the POKEd data statements as Sort.CO. At the OK prompt, press F8 to see and use the program.

Using Sort

The program is easy to use. At the main menu, place the cursor on Sort.CO. If the machine beeps, you haven't re-

served enough memory. In this case, go into Basic and type in CLEAR 100, MAXRAM-100.

Once you've reserved enough memory and have gone into Sort.CO, enter the name of the file you want to sort. If the records in the file don't meet the specifications, the computer tells you what file error to look for. If it meets Sort's specifications, the program displays the file's statistics (number of records and record size). Then it prompts for sort positions and an ascending or descending sort. At all prompts, you can press the enter key to return to the initial screen and then to the menu. Sorting a file with 242 39-byte records takes less than five seconds.

Sort.CO's sort algorithm is a simple linear-selection type—it exchanges records at the end of each pass. The Model 100's text file structure consists of records terminated by a carriage return/line feed with an end-of-file (EOF) marker following the last record. Its operating system uses dynamic memory management so files move about in memory.

To sort a data file in RAM, you must find the file's memory location, measure the length of the records in the file, count the number of records, and compute their locations. Then, you have to compare the records' sort keys and physically swap the records.

The data-input and error-trapping routines form the major part of the program. The program uses single-byte math computations for length, number of records, compare, and swap; this is why the program has file constraints. Each record can consist of 255 bytes: 253 bytes of information plus a carriage return and a line feed. ■

Write to Ronald F. Balonis at 118 Rice St., Trucksville, PA 18708.

Program Listing 2. Sort.BA. Use this program to create Sort.CO.

```

0 'SORT.BA ** MODEL 100 SORT BOOT ** **
10 'BY RONALD F. BALONIS
20 'RESERVE MEMORY FOR MACHINE LANGUAGE
30 '
50 CLEAR 100, MAXRAM-1000
100 CLS
150 PRINT@8,"** MODEL 100 SORT BOOT **
200 FOR I=61960 TO 62948
250 READ Z:POKE I,Z:CKSUM=CKSUM+Z
300 NEXT I
350 IF CKSUM=95393 THEN 500
400 PRINT@8,"***** ERROR IN DATA!*****
450 STOP
490 '---SAVE IT AS A MACHINE FILE
500 SAVEM "SORT.CO",61960,62948,61968
1000 '---DATA VIA RS232 AND DODATA/CMD
10000 'START=61960 END=62948 EXEC=61986
10005 DATA 0, 0, 0, 0, 0, 0, 0, 0
10010 DATA 0, 0, 0, 0, 0, 0, 0, 0
10015 DATA 0, 0, 0, 0, 0, 0, 0, 0
10020 DATA 0, 0, 0, 0, 0, 0, 6, 26
10025 DATA 33, 8,242, 54, 0, 35, 5
10030 DATA 194, 39,242,205, 49, 66, 33
10035 DATA 118,244,205, 11,244, 33,154
10040 DATA 244,205, 11,244, 33,182,244
10045 DATA 205, 11,244,205, 68, 70, 33
10050 DATA 133,246,126,183,202,151, 87
    
```

Listing 2 continued

Listing 2 continued

```

10055 DATA 62, 9,235,205,171, 90,202
10060 DATA 102,242,126,205,227, 90,254
10065 DATA 192,202,108,242, 33,145,245
10070 DATA 195,247,243, 33,117,245,195
10075 DATA 247,243, 34, 22,242, 43, 6
10080 DATA 0, 35, 4, 62,255,184,202
10085 DATA 162,242,126,254, 26,202,168
10090 DATA 242,254, 13,194,114,242, 35
10095 DATA 4, 58, 12,242, 60,254,255
10100 DATA 202,173,242, 50, 12,242, 58
10105 DATA 14,242,183,194,158,242,120
10110 DATA 50, 14,242,184,202,112,242
10115 DATA 33,173,245,195,247,243, 58
10120 DATA 12,242,214, 4, 33,201,245
10125 DATA 218,247,243, 58, 14,242,254
10130 DATA 3,218,162,242, 61, 61, 50
10135 DATA 32,242, 33,210,244,205, 11
10140 DATA 244, 33, 3, 2,205,124, 66
10145 DATA 33,133,246,205,177, 39, 42
10150 DATA 12,242,205, 25,244, 33, 3
10155 DATA 16,205,124, 66, 33, 8,242
10160 DATA 205,177, 39, 33, 3, 31,205
10165 DATA 124, 66, 42, 32,242,205, 25
10170 DATA 244, 33, 8,242,205,177, 39
10175 DATA 33, 25,245,205, 11,244, 33
10180 DATA 5, 23,205,124, 66,205, 80
10185 DATA 244,123, 61, 50, 18,242,123
10190 DATA 183,202, 34,242, 58, 32,242
10195 DATA 187,218,246,242, 33, 61,245
10200 DATA 205, 11,244, 33, 5, 35,205
10205 DATA 124, 66,205, 80,244,123, 61
10210 DATA 50, 20,242,123,183,202, 34
10215 DATA 242, 58, 32,242,187,218, 22
10220 DATA 243, 58, 18,242,187,210, 22
10225 DATA 243, 33, 75,245,205, 11,244
10230 DATA 33, 6, 35,205,124, 66,205
10235 DATA 68, 70, 33,133,246,126,183
10240 DATA 202, 34,242,254, 65,202, 99
10245 DATA 243,254, 68,194, 61,243, 62
10250 DATA 235,195,101,243, 62, 0, 50
10255 DATA 198,243, 58, 20,242, 33, 18
10260 DATA 242,150, 60, 50, 16,242, 58
10265 DATA 12,242, 61, 50, 30,242, 50
10270 DATA 31,242, 42, 22,242, 34, 26
10275 DATA 242, 34, 28,242, 34, 24,242
10280 DATA 195,164,243, 42, 14,242,229
10285 DATA 193, 42, 28,242, 9, 34, 28
10290 DATA 242, 34, 24,242, 34, 26,242
10295 DATA 58, 31,242, 50, 30,242, 42
10300 DATA 14,242,229,193, 42, 26,242
10305 DATA 9, 34, 26,242, 42, 18,242
10310 DATA 229,193, 42, 24,242, 9,235
10315 DATA 42, 26,242, 9, 58, 16,242
10320 DATA 79, 58, 14,242, 71, 0,205
10325 DATA 109, 90,250,211,243, 42, 26
10330 DATA 242, 34, 24,242, 33, 30,242
10335 DATA 53,194,164,243, 42, 24,242
10340 DATA 235, 42, 28,242,126,245, 26
10345 DATA 119,241, 18, 35, 19, 5,194
10350 DATA 225,243, 33, 31,242, 53,202
10355 DATA 151, 87,195,140,243,205, 11
10360 DATA 244, 33,253,244,205, 11,244
10365 DATA 205, 66,114,254, 32,202, 34
10370 DATA 242,195, 0,244, 94, 35, 86
10375 DATA 229,235,205,124, 66,225, 35
10380 DATA 205,177, 39,201, 17, 10,242
    
```

Listing 2 continued

Listing 2 continued



**MIDWEST  
COMPUTER  
WHOLESALE**



AS A WHOLESALE, MCM'S SERVICE IS NOT RECOMMENDED FOR EVERYONE. A WHOLESALE TRANSACTION IS BEST SUITED TO THE INFORMED OR EXPERIENCED SHOPPER, ONE WHO KNOWS HIS NEEDS. IF YOU ARE A BEGINNER WE ASK THAT YOU DEAL WITH A RETAILER WHO'S HIGHER MARGIN ALLOWS THE "HAND HOLDING" LEVEL OF SUPPORT WHICH WE CAN'T PROVIDE AT THESE PRICES. IF HOWEVER YOU DON'T NEED "HAND HOLDING" WE CAN OFFER YOU TREMENDOUS VALUES AND SAVINGS ON ALL YOUR COMPUTER SYSTEMS NEEDS. PLEASE READ ON.

**CHECK THESE SAMPLES OF OUR PRODUCTS  
GIVE US A CALL IF YOU DON'T FIND  
EXACTLY WHAT YOU ARE LOOKING FOR**

**\*\*\* COMPUTERS \*\*\***  
 TRS-80 MOD4 64K (2) DRIVE 40TR/88/DD.....\$1449.00  
 TRS-80 COLOR COMPUTER 2 16K STD BASIC.....\$139.00  
 TRS-80 COLOR COMPUTER 2 16K EXT BASIC.....\$209.00  
 TRS-80 COLOR COMPUTER 2 64K EXT BASIC.....\$299.00  
 COLOR COMPUTER DRIVE 0 KIT, 40TR TANDON.....\$319.00  
 COLOR COMPUTER DRIVE 0+1 IN SINGLE CASE.....\$489.00  
 R/S MODEL 2000 (IBM WORK ALIKE).....\$2310.00  
 R/S MODEL 4-8 PORTABLE.....\$1311.00  
 SANYO MODEL 550 (IBM WORK ALIKE).....\$CALL

**\*\*\* RAM DRIVES (TANDON OR MPI) \*\*\***  
 40 TRACK SINGLE SIDE DRIVES FROM.....\$140.00  
 40 TRACK DOUBLE SIDE DRIVES FROM.....\$184.00  
 80 TRACK SINGLE SIDE DRIVES FROM.....\$202.00  
 80 TRACK DOUBLE SIDE DRIVES FROM.....\$272.00  
 77 TRACK SINGLE SIDE EIGHT INCH DRIVES FROM.....\$357.00  
 77 TRACK DOUBLE SIDE EIGHT INCH DRIVES FROM.....\$442.00  
 SINGLE 5" CASE AND POWER SUPPLY FROM.....\$53.00  
 DUAL 5" CASE AND POWER SUPPLY FROM.....\$84.00  
 SINGLE OR DUAL 8" CASE AND POWER SUPPLY FROM.....\$199.00

**\*\*\* WINCHESTER HARD DRIVES (TANDON) \*\*\***  
 COMPLETE WITH CASE & P/S, MOD 3 OR 4 DDS, 1 YR WARRANTY  
 6.4 MEG.....\$1399.00  
 12.8 MEG.....\$1549.00  
 19.1 MEG.....\$1699.00  
 31 MEG.....\$2199.00  
 50.6 MEG.....\$2799.00

**\*\*\* PRINTERS \*\*\***  
 EPSON/RX-80, RX-80FT, FX-80, MX-100, FX-100.....\$CALL  
 SMITH CORONA DAISY WHEEL.....\$CALL  
 NEW SMITH CORONA DOT MATRIX LINE.....\$CALL

**\*\*\* CABLES \*\*\***  
 (2) DISK DRIVES.....\$14.00  
 (4) DISK DRIVES.....\$24.00  
 PARALLEL PRINTER (STD).....\$18.00  
 RS232C.....\$15.00  
 NULL MODEM ADAPTER.....\$19.00  
 SPECIAL CABLES TO YOUR SPEC'S.....\$CALL

**\*\*\* MISCELLANEOUS \*\*\***  
 COMPUTER POWERLINE FILTERS FROM.....\$25.00  
 "J"-CAT MODER 300 BAUD AUTO ANG, D/C.....\$109.00  
 SMART CAT MODER 300 BAUD AUTO ANG/DIAL, D/C.....\$190.00  
 SMART CAT MODER 1200 BAUD AUTO ANG/DIAL, D/C.....\$445.00  
 GREEN OR AMBER CAT FILTERS .125" THICK.....\$9.50  
 5.1 MEG MOD 3 SPEED-UP KIT.....\$79.00  
 16K MEM KIT (150NB PRIME CHIPS).....\$18.00  
 64K MEM KIT (150NB PRIME CHIPS).....\$39.00  
 64K TO 128K MOD 4 MEMORY KIT W/PAL.....\$89.00  
 PRINTER SWITCH, 2 PRINTERS ON ONE COMPUTER.....\$68.00

DISKETTES, BOXES OF 10.....OPUS.....VERBATIM  
 40 TRACK SINGLE SIDE, DBL DENSITY.....\$19.....\$25  
 40 TRACK DOUBLE SIDE, DBL DENSITY.....\$29.....\$38  
 80 TRACK SINGLE SIDE, DBL DENSITY.....\$35.....\$45  
 80 TRACK DOUBLE SIDE, DBL DENSITY.....\$43.....\$53  
 40 TR DBL SIDE, DBL DENSITY (FLIPPY).....\$30.....\$38

ALL PRICES & SPEC'S SUBJECT TO CHANGE, WITHOUT NOTICE..

TRS-80 IS A TRADEMARK OF RADIO SHACK-DIV OF TANDY CORP. ALL TRS-80'S FROM MCM ARE CONSTRUCTED TO OUR SPEC'S AND ARE COVERED BY OUR WARRANTY SEE NOTE BELOW.

**\*\*\* PAYMENTS \*\*\***  
 WE ACCEPT VISA, M/C, MONEY ORDERS, AND CHECKS. NOTE: NON-CERTIFIED CHECKS REQUIRE (2) WEEKS FOR BANK CLEARANCE. NO SALES TAXES ARE COLLECTED ON SALES OUTSIDE OF MICHIGAN.

**\*\*\* DELIVERY \*\*\***  
 SHIPMENTS ARE MADE PROMPTLY FROM STOCK, VIA UPS. (BIZE PERMITTING). OTHERS, BEST WAY. COSTS UNLESS OTHERWISE NOTIFIED ARE 2.5% OF THE ORDER TOTAL BUT NOT LESS THAN \$3

**\*\*\* WARRANTY \*\*\***  
 YOUR HARDWARE PURCHASE IS COVERED BY AN MCM 90 DAY, LIMITED, PARTS AND LABOR WARRANTY OR THE WARRANTY OF THE ITEM'S MANUFACTURER. CONFIRM YOUR COVERAGE WHEN PLACING AN ORDER. COPIES OF THE WARRANTY ARE AVAILABLE ON REQUEST.

**MIDWEST COMPUTER WHOLESALE  
PO BOX 39278 DETROIT, MI. 48239  
TELEPHONE ORDER LINE (313) 525-3040**

|       |      |                                 |
|-------|------|---------------------------------|
| 10385 | DATA | 1, 3, 10, 197, 175, 14, 17      |
| 10390 | DATA | 195, 43, 244, 144, 242, 48, 244 |
| 10395 | DATA | 128, 41, 23, 195, 51, 244, 41   |
| 10400 | DATA | 23, 35, 13, 194, 38, 244, 31    |
| 10405 | DATA | 193, 198, 48, 18, 27, 13, 194   |
| 10410 | DATA | 31, 244, 33, 8, 242, 62, 48     |
| 10415 | DATA | 190, 192, 54, 32, 35, 190, 192  |
| 10420 | DATA | 54, 32, 201, 205, 68, 70, 33    |
| 10425 | DATA | 133, 246, 17, 0, 0, 126, 183    |
| 10430 | DATA | 200, 254, 58, 208, 254, 48, 216 |
| 10435 | DATA | 214, 48, 131, 95, 35, 62, 0     |
| 10440 | DATA | 190, 200, 235, 41, 229, 41, 41  |
| 10445 | DATA | 193, 9, 235, 195, 89, 244, 1    |
| 10450 | DATA | 8, 61, 61, 61, 61, 32, 32       |
| 10455 | DATA | 77, 79, 68, 69, 76, 32, 49      |
| 10460 | DATA | 48, 48, 32, 83, 79, 82, 84      |
| 10465 | DATA | 32, 32, 61, 61, 61, 61, 32      |
| 10470 | DATA | 82, 70, 66, 47, 56, 52, 0       |
| 10475 | DATA | 8, 8, 61, 61, 32, 80, 82        |
| 10480 | DATA | 69, 83, 83, 32, 69, 78, 84      |
| 10485 | DATA | 69, 82, 32, 84, 79, 32, 69      |
| 10490 | DATA | 88, 73, 84, 32, 61, 61, 0       |
| 10495 | DATA | 3, 1, 32, 32, 70, 73, 76        |
| 10500 | DATA | 69, 78, 65, 77, 69, 32, 60      |
| 10505 | DATA | 88, 88, 88, 88, 88, 88, 46      |
| 10510 | DATA | 88, 88, 62, 32, 63, 32, 0       |
| 10515 | DATA | 3, 1, 60, 32, 32, 32, 32        |
| 10520 | DATA | 32, 32, 32, 32, 32, 62, 32      |
| 10525 | DATA | 61, 32, 32, 32, 32, 32, 32      |
| 10530 | DATA | 82, 69, 67, 79, 82, 68, 83      |
| 10535 | DATA | 32, 79, 70, 32, 32, 32, 32      |
| 10540 | DATA | 32, 32, 66, 89, 84, 69, 83      |
| 10545 | DATA | 0, 8, 8, 61, 61, 61, 61         |
| 10550 | DATA | 32, 80, 82, 69, 83, 83, 32      |
| 10555 | DATA | 83, 80, 65, 67, 69, 32, 66      |
| 10560 | DATA | 65, 82, 32, 61, 61, 61, 61      |
| 10565 | DATA | 0, 5, 1, 83, 79, 82, 84         |
| 10570 | DATA | 32, 80, 79, 83, 73, 84, 73      |
| 10575 | DATA | 79, 78, 32, 32, 83, 84, 65      |
| 10580 | DATA | 82, 84, 61, 32, 32, 32, 32      |
| 10585 | DATA | 32, 32, 32, 32, 32, 32, 32      |
| 10590 | DATA | 32, 0, 5, 30, 69, 78, 68        |
| 10595 | DATA | 61, 32, 32, 32, 32, 32, 32      |
| 10600 | DATA | 32, 0, 6, 1, 83, 79, 82         |
| 10605 | DATA | 84, 32, 60, 65, 62, 83, 67      |
| 10610 | DATA | 69, 78, 68, 73, 78, 71, 32      |
| 10615 | DATA | 79, 82, 32, 60, 68, 62, 69      |
| 10620 | DATA | 83, 67, 69, 78, 68, 73, 78      |
| 10625 | DATA | 71, 32, 32, 32, 32, 32, 32      |
| 10630 | DATA | 32, 0, 3, 1, 32, 32, 42         |
| 10635 | DATA | 42, 42, 32, 67, 65, 78, 78      |
| 10640 | DATA | 79, 84, 32, 70, 73, 78, 68      |
| 10645 | DATA | 32, 42, 42, 42, 32, 32, 32      |
| 10650 | DATA | 32, 0, 3, 1, 42, 42, 42         |
| 10655 | DATA | 32, 70, 73, 76, 69, 32, 78      |
| 10660 | DATA | 79, 84, 32, 84, 69, 88, 84      |
| 10665 | DATA | 32, 42, 42, 42, 32, 32, 32      |
| 10670 | DATA | 32, 0, 3, 1, 42, 32, 82         |
| 10675 | DATA | 69, 67, 79, 82, 68, 32, 76      |
| 10680 | DATA | 69, 78, 71, 84, 72, 32, 69      |
| 10685 | DATA | 82, 82, 79, 82, 32, 42, 32      |
| 10690 | DATA | 32, 0, 3, 1, 42, 42, 32         |
| 10695 | DATA | 35, 32, 79, 70, 32, 82, 69      |
| 10700 | DATA | 67, 79, 82, 68, 83, 32, 69      |
| 10705 | DATA | 82, 82, 79, 82, 32, 42, 42      |
| 10710 | DATA | 32, 0:                          |

End

## Etch-A-Screen

by Richard J. Perry

When I was a kid, just about everybody I know had an Etch-A-Sketch—you know, the flat, red box with a gray screen and two knobs that looked like a toy television set. By turning the knobs you could draw lines every which way across the screen. Now that I've aged, I've gotten more sophisticated. I found a way to make an electronic etcher using the Model 100 and my Basic program, Etcher.

Etcher lets you create complex graphics on your Model 100's screen (see Program Listing 3). You can access all the Model 100's graphics capabilities by positioning the blinking pixel cursor and entering Etcher's single-key commands. Etcher lets you draw circles, enter text, and produce reverse-video images. You can even store the pictures you create in a binary (.CO) file. Etcher also lets you merge images from different files.

Two additional Basic utility programs, Dump and Hexech, dump screen images to a graphics printer and convert binary image files to ASCII text files (.DO files) for uploading and downloading files via Telecom, the Model 100's telecommunications program (see Program Listings 4 and 5, respectively).

## Running Etcher

You execute each Etcher command with a single keystroke. In conjunction with the line or pixel commands, the shift key determines whether the drawing is in normal or reverse video by setting or resetting the pixels.

Pressing the shift key along with a command key resets a screen pixel (turns it off). Pressing a command key without pressing the shift key sets a pixel (turns it on). This lets you move the pixel-sized cursor across the screen or draw a line.

Table 3 contains a list of program commands and their functions for quick reference.

## Pixel Commands

Pixel commands set and reset individual pixels and move the cursor around the screen. Hit the less-than sign to move the cursor left, the greater-than sign to move right, the A key to move up, and the Z key to move down.

## Line Commands

Etcher's line commands create lines, boxes, and circles. Etcher uses two pixel positions as reference points when executing a line command.

Use the Mark command (M) to mark the cursor position. Then move the cursor to the desired second pixel position and press the appropriate line command.

The C command connects lines that you mark. The B command draws a box. The marked pixel indicates one corner of the box, and the current cursor position indicates the opposite corner.

The H command draws a filled box, and the O command draws an oval. The marked pixel indicates the center of the circle, and the current cursor position determines the radius.

## File Commands

When you run Etcher, it first prompts you for a file name. If the file name you type in exists as a binary file, Etcher loads

TRS-80®

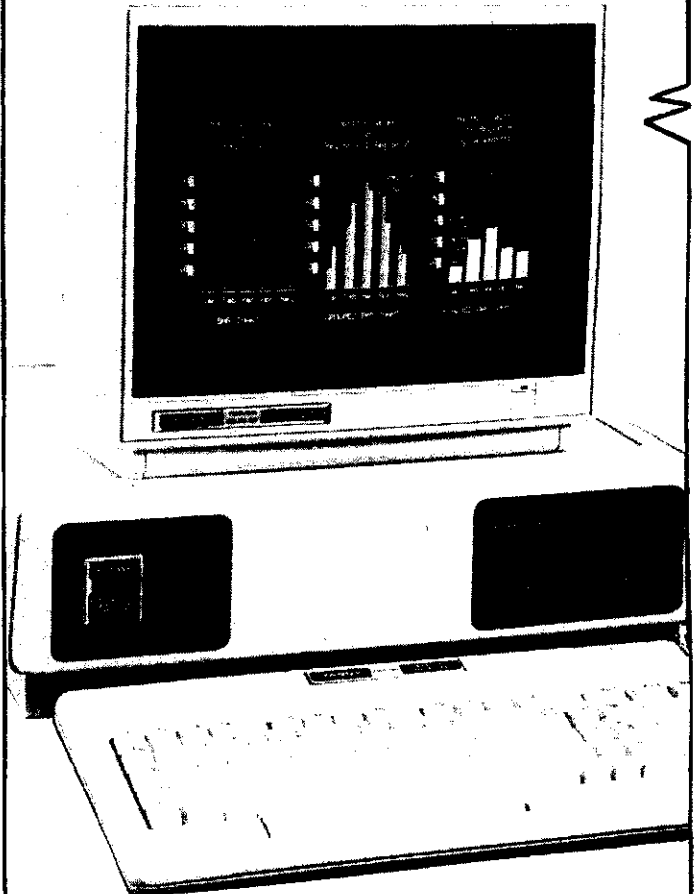
WILL YOU PAY TOO MUCH?

CAN YOU BUY DIRECT?

Ask About Our  
"30 DAYS  
Buy-Back-Policy"

## WARRANTY:

the RADIO SHACK warranty accompanies all R. S. merchandise sold by us.



MODEL 2000. 768K 2 DR.  
(10 MEG. HARD DRIVE - BUILT IN).



® TRADE MARK  
OF RADIO SHACK

SAVE SALES TAX\*  
PLUS DISCOUNT

\*TEXAS RESIDENTS ADD ONLY 4%

FORT WORTH COMPUTERS

WE ARE SERIOUS ABOUT SAVING YOU MONEY

(Located 30 miles from Fort Worth)

377 Plaza • GRANBURY • NR FORT WORTH, TEXAS 76048

TOLL FREE: 1-800-433-S-A-V-E

Monday thru Friday — 9:00 a.m. to 5:00 p.m. Texas Time  
(Order Inquiries/ Customer Service &) IN TEXAS: 817-573-4111

# BASIC

FOR MODELS I AND III

# JUST GOT BETTER

## INTRODUCING BASRUM. THE ONLY LINKING LOADER & LIBRARY MANAGER FOR BASIC.

BASRUM is designed to manage the subroutines you write and use in your Basic programs. Like an efficient filing system, BASRUM saves you time, energy, and space. Best of all, it makes your program writing FASTER and EASIER.

### HOW DOES IT WORK?

BASRUM lets you store your subroutines in "library" files, separate from the programs that use them. ANY Basic program can then use any routine in any file. You can even load up to 256 routines to a special, protected memory area and use them AFTER you load other Basic programs! Naturally, you can replace or delete any routine from memory at any time.

### SUPER BENEFITS!

NOW you can write a subroutine ONCE, store it on disk ONCE ...but use it in ANY program at ANY time, AUTOMATICALLY! Load them once, use them all day. You can CHAIN from program to program, keeping the existing subroutines, variables, and file buffers. Your Basic programs, being shorter, will LOAD FASTER. NEED MORE MEMORY? You can break long programs into shorter modules and "overlay" the modules. BASRUM gives you complete control of WHAT is loaded, WHEN it is loaded, and WHERE it is loaded.

### THE NITTY GRITTY.

BASRUM was designed specifically for the models I and III for use with any DOS that is compatible with TRSDOS®. Its overlay structure requires only the top 1600 bytes of memory. There is NO LIMIT on the number of working copies.

**SIMPLE ENOUGH FOR THE BEGINNER. VERSATILE ENOUGH FOR THE PRO: IF YOU WRITE PROGRAMS IN BASIC YOU NEED BASRUM.**

**TO ORDER:** Money order, check, VISA, MASTERCARD: \$149.50  
Specify memory size: 16K 32K 48K  
Select disk format:  
Model I: 35 track, single density  
Model III: 40 track, double density

Guaranteed to perform as described or your money back. Orders are shipped via air, most within 24 hours. A complete manual is included. Information only? Send self-addressed, stamped envelope.



by **Wiley, Inc.**

841 Bishop St., Ste. B-2 Honolulu, HI 96813  
(808) 531-4314

TRSDOS is a registered trademark of the Tandy Corp.  
BASRUM is a trademark of Wiley, Inc.

and displays it on the screen. Use the file name when saving the file with the S command.

The F command files the image under a new name. When the block cursor appears, enter the desired file name and press the enter key. To preserve the screen graphics, the name does not appear on the screen.

The L command loads an image from a binary file and merges it with the current screen. Press the L key and enter the file name as you would at the file prompt.

G is the Dump command. It sends the screen image to the printer by executing the Dump utility (see Listing 4). This utility produces a double-height, double-width image on the DMP-100 printer.

### Other Commands

T is the Text Entry command. It puts a block cursor on the screen at the current pixel position. Type in the desired text and press the enter key. To move the cursor and control reverse video in this mode, I used the escape and control codes shown in Table 3.

The Invert command (I) inverts the screen. This sets and resets pixels to create a reverse-video image.

E is the Erase command. It clears the screen and returns control to the beginning of the program.

Q, the Quit command, ends the program and returns control to the menu. It does not save the screen image.

### Program Description

Etcher's main program is a loop in lines 70-100 that controls the cursor's blinking and dispatches the appropriate subroutine to execute the commands you enter.

The C\$ variable in line 60 contains a list of command abbreviations. You can modify it to change command symbols or to add new commands. If you press the shift key when entering a command, the program sets the variable P to zero. Otherwise Etcher sets it to -1.

The On Error GOTO 410 statement in line 60 facilitates error trapping so you don't lose the screen image if anything goes wrong during program execution. If an error occurs, line 410 makes your Model 100 beep and sends the program to line 70.

The Clear 256,60829 statement at the beginning of Etcher reserves space in high memory to contain a set of machine-language utility subroutines and the binary screen image. I assembled the machine-language routines and placed them into data statements at the end of the program. These routines allow quick and efficient screen image transfers to and from the buffer in high memory.

Etcher POKES the machine code into RAM from 60830 to 61016. The calling syntax for these routines from Basic is CALL 60830,A. "A" specifies the routine the program should execute and its value ranges from zero to 2.

The routine at zero copies the screen image to the memory buffer. The routine at 1 copies the buffer to the screen, and the routine at 2 inverts the screen image.

I placed the screen buffer in RAM from 61030 to 62949 using 1 bit per pixel for a total of 1,920 bytes. Etcher saves the screen RAM image as a binary file. From there you can save it to cassette.

### The Hexech Utility

My Basic Hexech utility in Program Listing 5 lets you convert binary image files to ASCII hexadecimal (hex) representa-



tion that you can up- and download via Telecom. The binary files require 1,926 bytes for storage; the ASCII equivalent requires 3,873 bytes. Since Etcher doesn't access this utility, you must load and run it from the main menu.

**The Dump Utility**

Etcher's G command prints the screen display by executing the Dump utility in Listing 4. If you have a Radio Shack DMP-100 printer, you can use Dump.BA to create a machine-code Dumpbinary file that you can call from Etcher.

Dumpbinary loads into RAM from 61900 to 62260. You can call it from Basic by typing CALL 61900. If you don't have the Dumpbinary file in RAM, Etcher ignores the G command.

If you have a printer other than the DMP-100 and you have the necessary software to dump screen graphics, modify line 360 in Etcher to call your routine. ■

*Richard J. Perry teaches electrical engineering at Villanova University. He is the co-founder of Secure Systems Inc. (P.O. Box 30, Blue Bell, PA 19422), which provides encryption programs for the Model 100 and other computers. Write to him at Villanova University, Department of Electrical Engineering, Villanova, PA 19085.*

*Program Listing 3. Etcher.*

```

10 REM ETCHER by Richard J. Perry
20 CLEAR256,60829:SCREEN0,0:DEFINTA-
Z:READX,Y:FORK=XTOY:READZ:POKEK,Z:NEXT:O
NERRORGOTO420
30 LINEINPUT"File:";F$:IFF$=""THEN400
ELSEZ=-4516:POKEZ,LEN(F$):FORK=1TOLEN(F$
):POKEZ+K,ASC(MID$(F$,K,1))
:NEXT:LOADMF$:CLS:CALL60830,1
40 DEFINTA-Z:GOTO60
50 PRINT@280,"Box ? ";I=INSTR(
"YyNn",INPUT$(1)):IFI=0THEN50ELSECLS:IFI
<3THENLINE(0,0)-(239,63),1,B
60 ONERRORGOTO410:X=120:Y=32:M=0:P=0:
Q=0:C=0:C$=",<.>aAzZmMcCbBhHoOiItTsSff
lLgGeEqQ"
70 C=C+1:IFC>20THENC=0:IFQTHENPRESET(X,Y
):Q=0ELSEPSET(X,Y):Q=-1
80 A$=INKEY$:IFA$=""THEN70ELSEI=INSTR(
C$,A$):IFI=0THEN70ELSEQ=0:IFPTHENPSET(
X,Y)ELSEPRESET(X,Y)
90 IFI<23THENP=-1(IMOD2)
100 I=(I+1)\2:ONIGOSUB110,130,150,170
,190,200,220,240,260,290,300,320,320,370
,360,390,400:GOTO70
110 X=X-1:IFX<0THENX=239
120 RETURN
130 X=X+1:IFX>239THENX=0
140 RETURN
150 Y=Y-1:IFY<0THENY=63
160 RETURN
170 Y=Y+1:IFY>63THENY=0
180 RETURN
190 M=-1:X1=X:Y1=Y:RETURN
200 IFMTHENLINE(X1,Y1)-(X,Y),-P
210 RETURN
220 IFMTHENLINE(X1,Y1)-(X,Y),-P,B
230 RETURN
240 IFMTHENLINE(X1,Y1)-(X,Y),-P,BF
250 RETURN
260 IFNOTMTHENRETURNELSEI=SQR((X-X1)^2+
(Y-Y1)^2):Y2=Y1:X2=X1+R!+.5:IFX2>239
THENX2=239
    
```

*Listing 3 continues*

*Listing 3 continued*

```

270 FORZ!=0TO8*ATN(1)STEP.25:X3=X1+R!*
COS(Z!)+.5:Y3=Y1+R!*SIN(Z!)+.5:IFX3<
240ANDX3>=0ANDY3<64ANDY3>=0THENLINE(
X2,Y2)-(X3,Y3),-P:X2=X3:Y2=Y3
280 NEXT:RETURN
290 CALL60830,2:RETURN
300 E$=CHR$(27):PRINT@40*(Y\8)+
X\6,E$"V"E$"q";
310 A$=INPUT$(1):IFASC(A$)
=13THENPRINTES"W"E$"q"E$"Q";:X=6*POS(0)
:Y=8*CSRLIN:RETURNELSEPRINTA$;:GOTO310
320 CALL60830,0:F$="":IFI=13THEN350
330 Z=-4516:FORK=1TOPEEK(Z):F$=F$+CHR$(
PEEK(Z+K)):NEXT
340 SAVEMF$,61030,62949
350 A$=INPUT$(1):I=ASC(A$):IFI=13THENIF
F$=""THEN330ELSEF$=F$+A$:GOTO350
360 LOADM"DUMP":CALL61900:GOTO40:REM
This line can be replaced by a simple
RETURN*if you do not have a DMP-100
printer
370 F$=""
380 A$=INPUT$(1):I=ASC(A$):IFI<>
13THENF$=F$+A$:GOTO380ELSELOADMF$:
CALL60830,1:GOTO40
390 CLS:GOTO20
400 CLEAR256,MAXRAM:MENU
410 BEEP:RESUME70
420 IFERR=52THENRESUME50ELSEBEEP
:RESUME30
430 DATA-4706,-4520,254,3,208,111,38,0
,41,17,174,237,25,94,35,86,213,201,180,2
37,9,238,60,238,205,73,66,33,102,238,30,
1,22,1,205,211,237,20,62,41,186,194,190,
237,28,62,9,187,194,188
440 DATA237,205,78,66,201,213,235,205
,124,66,235,62,1,50,243,255,62,10,6,170,
5,194,226,237,61,194,224,237,62,47,50,25
5,237,58,242,255,183,194,249,237,50,255,
237,17,236,255,6,6,26,0
450 DATA19,35,19,5,194,254,237,209
,201,33,102,238,14,0,22,0,121,7,7,95,6
,8,126,229,15,210,38,238,245,197,213,205
,76,116,209,193,241,28,5,194,25,238,225,
35,20,62,240,186,194
460 DATA6,238,12,62,8,185,194,14,238
,201,205,180,237,33,102,238,30,8,22,240,
126,47,119,35,21,194,70,238,29,194,68,23
8,205,49,66,205,9,238,201
    
```

*End*

*Table 3. Command and function reference table for Etcher.*

| Command       | Function                                  |
|---------------|-------------------------------------------|
| A             | Move cursor up                            |
| Left arrow    | Move cursor left                          |
| Right arrow   | Move cursor right                         |
| Shift         | Use with other commands for reverse-video |
| Z             | Move cursor down                          |
| Line Commands |                                           |
| B             | Draw a box                                |
| C             | Connect pixels                            |
| H             | Draw a filled box                         |
| M             | Mark pixel position                       |
| O             | Draw an oval                              |

*Table continued*

## C•NOTES

Table continued

| Command                         | Function                         |
|---------------------------------|----------------------------------|
| <b>File Commands</b>            |                                  |
| F                               | Save image to new name           |
| G                               | Print screen                     |
| L                               | Load file over current screen    |
| S                               | Save screen to current file name |
| <b>Other Commands</b>           |                                  |
| E                               | Erase screen                     |
| I                               | Invert screen                    |
| Q                               | Quit                             |
| T                               | Text entry mode                  |
| <b>Escape and Control Codes</b> |                                  |
| ESC/A                           | Cursor up                        |
| ESC/B                           | Cursor down                      |
| ESC/C                           | Cursor right                     |
| ESC/D                           | Cursor left                      |
| ESC/P                           | Reverse video on                 |
| ESC/Q                           | Reverse video off                |
| CTRL/J                          | Cursor down                      |
| CTRL/K                          | Cursor home                      |
| CTRL/L                          | Clear screen                     |

### Program Listing 4. Dump utility for printing out the screen display.

```

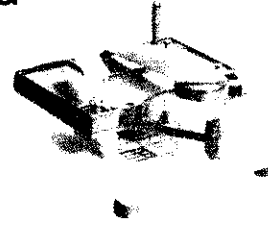
1 REM DUMP.BA by Richard J. Perry
2 CLEAR256, 61899 :DEFINTA-
Z:READX,Y:FORK=XTOY:READZ:POKEK,Z:NEXT:SA
AVEM"DUMP",X,Y,X
10 DATA-3636,-3276,33,73,242,229,42
,57,246,34,117,247,33,133,246,6,240,175,
119,35,5,194,220,241,62,18,205,63,109,21
6,205,73,66,30,1,33,123,242,54,1,35,35,5
4,6,35,54,7,33,164,242
11 DATA54,8,205,96,242,33,123,242,54
,254,35,35,54,1,35,54,15,205,96,242,33,1
23,242,126,183,31,119,35,35,35,126,254,1
5,194,37,242,54,0,195,50,242,183,194,46,
242,54,7,195,50,242
12 DATA43,126,60,119,33,164,242,126
,61,119,28,62,9,187,194,13,242,29,33,164
,242,54,1,205,96,242,201,62,30,205,63,10
9,218,73,242,62,13,205,63,109,42,117,247
,34,57,246,205,78,66,201
13 DATA62,2,50,119,247,22,1,33,133
,246,205,251,242,14,6,213,229,33,122,247
,126,35,34,120,247,225,230,0,6,0,0,5,194
,126,242,182,95,58,119,247,61,123,202,15
0,242,7,7,7,205,226
14 DATA242,195,172,242,205,226,242
,122,15,87,229,42,120,247,43,126,225,6,0
,183,31,5,194,165,242,119,122,246,128,20
5,63,109,212,63,109,210,187,242,209,225,
201,35,229,42,120,247,13,194,116,242
15 DATA225,209,20,62,41,186,194,106
,242,62,13,205,63,109,210,215,242,225,20
1,58,119,247,61,50,119,247,194,101,242,2
01,7,22,0,6,4,95,122,7,7,87,123,7,95,210
,246,242,122,246,3,87
16 DATA5,194,232,242,201,229,213,235
,205,124,66,62,1,50,243,255,62,10,6,170,
5,194,10,243,61,194,8,243,62,47,50,42,24
3,58,242,255,183,194,33,243,50,42,243,17
,236,255,33,122,247,6
17 DATA6,26,0,119,35,19,5,194,41,243
,209,225,201
    
```

End

# E-ZEE INKER

## STOP THROWING AWAY FABRIC RIBBONS

### \$39.50



Stop throwing away used fabric ribbons. Re-ink them yourself with the E-ZEE INKER. The E-ZEE INKER can re-ink almost any cartridge type fabric ribbon in only a few minutes. And you can do most reel to reel ribbons with an optional attachment. All without fuss! Without mess!

The E-ZEE INKER uses a proven capillary type inking system. This ink especially formulated to provide many pages of "like new" print from your ribbon. Ink cost is only a few cents per ribbon. Most ribbons can be re-inked three times or more.

Most ribbons are mounted on the E-ZEE INKER in seconds! The E-ZEE INKER is a precision compact machine. It is fully assembled, with complete instructions. The E-ZEE INKER is approximately 6" x 8" x 4" high. And you get two ounces of ink with every E-ZEE INKER. Additional ink may be purchased in 4 ounce bottles.

- Versatile
- Simple to Use
- No fuss... No Mess
- Economical

## CUT RIBBON COST



**BORG INDUSTRIES**  
PO Box 508  
Baltimore, Maryland

Order Toll Free 800-553-2404  
In Iowa 319-967-2976  
VISA MC accepted

✓ 256

## Beyond the Standard

**AS LOW AS**  
**\$1.85**  
*each*

Introducing the perfect partner to the TRS-80, IBM PC, APPLE, and all other PC's. Now with Storage Master, you can use the top of the Control Data line in flexible diskettes. Control Data Storage Master diskettes are known for their high quality and consistency. Each diskette is 100% certified and backed with a full 5-year warranty.

| Size            | Boxes 1-5 | Boxes 6-9 | Boxes 10 or more |
|-----------------|-----------|-----------|------------------|
| 5 1/4" S.S.D.D. | 19.45     | 18.95     | 18.50            |
| 5 1/4" D.S.D.D. | 27.95     | 27.45     | 27.00            |
| 8" S.S.S.D.     | 19.50     | 19.00     | 18.55            |
| 8" S.S.D.D.     | 25.00     | 24.50     | 24.05            |
| 8" D.S.D.D.     | 29.50     | 29.00     | 28.55            |

Add \$3.50 shipping & handling for 1-5 boxes and \$5.00 for orders of 6 or more.

— Fill in this portion, cut and mail today. —

294 **KAR Printing of Florida, Inc.**  
Computer Supplies Dept. • 7780 W. 2nd Court • Hialeah, FL 33014 • (305) 557-4782

Please ship: \_\_\_\_\_ Quantity \_\_\_\_\_ Product \_\_\_\_\_ for \$ \_\_\_\_\_

Authorized Signature \_\_\_\_\_ ( ) Phone \_\_\_\_\_

VISA Credit Card # \_\_\_\_\_

Mastercard Exp. Date \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

## C•NOTES

Program Listing 5. Hexech binary hex file conversion utility.

```
10 REM HEXECH by Richard J. Perry
20 CLEAR256,61009:DEFINTA-Z
30 CLS:PRINT@122,"Convert <1> to hex, <
2> from hex":PRINT@280,"Select:";
40 N=VAL(INPUT$(1)):IFN<LORN>
2THENBEEP:GOTO40ELSECLS:FILES:LINEINPUT"
Input:";F$:ONNGOTO110,140
50 A$="":K=X-16*(X\16):IFK>9THENK=K+7
60 A$=CHR$(K+48):K=X\16:IFK>9THENK=K+7
70 A$=CHR$(K+48)+A$:RETURN
80 K=ASC(LEFT$(A$,1))-48:IFK>9THENK=K-7
90 X=K*16:K=ASC(RIGHT$(A$,1))-48:IFK>
9THENK=K-7
100 X=X+K:RETURN
110 LOADMF$:DEFINTA-Z:LINEINPUT"Output:"
```

```
;F$:OPENF$FOROUTPUTAS1
120 Z=-4507:FORI=1TO16:FORJ=1TO120:Z=Z+
1:X=PEEK(Z):GOSUB50:PRINT#1,A$;:NEXT
:PRINT#1,"":NEXT
130 CLOSE:CLEAR256,MAXRAM:BEEP:END
140 OPENF$FORINPUTAS1:
LINEINPUT"Output:";F$:Z=-4526:POKEZ,LEN(
F$):FORK=1TOLEN(F$):POKEZ+K,ASC(MID$(
F$,K,1)):NEXT
150 Z=-4507:FORI=1TO16:FORJ=1TO120:Z=Z+
1:A$=INPUT$(2,1):GOSUB80:POKEZ,X:NEXT:
A$=INPUT$(2,1):NEXT
160 CLOSE:BEEP:F$="":Z=-4526:FORK=1
TOPEEK(Z):F$=F$+CHR$(PEEK(Z+K))
:NEXT:SAVEMF$,61030,62949
```

End

## RAM FILES

### Alternative to Uploading

The article "Uploading Text to WordStar" (August 1983, p. 210) discusses the disadvantage of using Telecom to upload files: It inserts carriage returns where none existed before.

The author's program solves the problem by transmitting the text as typed, without extra carriage returns. This slows transmission, however, so that a computer set at 19,200 baud actually transmits at 1,200 baud.

One way to solve this problem is to use Telecom for transmission and then eliminate the extra carriage returns on the receiving end. In WordStar, use the search and replace function (^QA) to search for ^P^M (a carriage return character).

Replace it with ^N, a carriage return/line feed sequence. Do a global search and replace without query (options G and N). Speed up the search and replace operation by typing ^X.

If your original text uses a double carriage return to denote a paragraph, eliminate the extra carriage returns and save the ones you type. To do this, turn your double carriage returns into double carriage return/line feed sequences by searching for ^P^M^P^M and replacing it with ^N^N.

Remove all carriage returns by searching for ^P^M. Then replace all line feed characters with carriage return/line feed sequences by searching for ^P^J and replace it with N. Use the Reformat Paragraph command (^B) to insert soft carriage returns according to the margins.

Another method is to use an undocumented feature of the Text Save command (F3). Use Text to open the file you want to save, then press the F3 key. After the Save to: prompt, type COM:parameters, where "parameters" is the parameter list for the RS-232 port.

You can use Telecom's Stat command to see what RS-232 parameters you've been using. I use 9711E to transmit at

19,200 baud by typing COM:9711E. This lets you send the file without inserting carriage return characters and without having to use a separate program.

Warren H. Buske  
Det. Wobeck Box 5683  
502d ASA Bn.  
APO, NY 09742

### The Bottom Line

Radio Shack's Personal Finance programs for the Model 100 lack many useful features.

My patches in Program Listing 1 calculate budget totals for the ACCTM.BA program so you can see how budget changes affect your total.

Pressing the F6 key displays the budget total. Pressing any other key returns control to the ACCTM menu.

Brad Alan Wright-Hulett  
500 S. 4th St., Apt. C9  
Morton, IL 61550

Program Listing 1. Patches for ACCTM.BA.

```
11 ONERROR GOTO0:INPUT#1,N:IFN=0THEN1ELSEBA=0
:FORX=1TON:FORY=1TO3:INPUT#1,A$(X,Y):
NEXTY:BA=BA+VAL(A$(X,3)):NEXTX
14 GOSUB50:KEYON:
ONKEYGOSUB5000,6000,7000,70,75,8000,16,80
30 GOSUB 31:PRINT@280,
"Add Edit Rem Wtpe Wram Tot PF ";
:RETURN
```

Listing 1 continued

## RAM FILES

Listing 1 continued

```

5200 FORF5=2TO3:B=F5:A=Q:GOSUB36:A=N:GOSUB40
:GOSUB55:B$=T$:GOSUB56:A$(N,B)=T$:NEXT F5
:BA=BA+VAL(A$(N,3)):B=1:J=0:IFN=1THENP=1:S=1
:BEEP:GOTO5505
6005 A$(A,B)=T$:GOSUB55:GOSUB20:GOSUB18
:IFB>1THENIFB=3THENBA=BA-VAL(Q1$)
:BA=BA+VAL(A$(A,3)):RETURNLSERETURN
7100 BA=BA-VAL(A$(A,3)):FORX=ATON:FORY=1TO3
:A$(X,Y)=A$(X+1,Y):NEXTY,X:IFA=NTHENA=A-1
:IFA=0THENA=1:P=1:S=1
8000 PRINT@280,CHR$(27);"K";"TOTAL BUDGET ";
:PRINTUSING"$$###.###.###":BA:
8010 K$=INKEY$:IFK$=""THEN8010ELSEGOSUB30
:RETURN
    
```

End

## Improved Gas Mileage

You can modify Gas.BA from my article, "Gas/Oil Mileage" (July 1983, p. 170), to keep a running record of your gas costs by typing in the line changes in Program Listing 2.

Beve Woodbury  
Technical Editor  
80 Micro

Program Listing 2. Modifications for Gas.BA.

```

90 DIMDD(L+1),Y(L+1,4):INPUT#1,Y(0,2)
130 FOR K=0TO4
195 PRINTB;"3. TOTAL GAS COST"
200 PRINTB;"4. PRINT OIL USAGE":PRINT
240 ONZGOTO250,500,592,600
425 Y(N,4)=P*Q
470 FORK=0TO4
592 FORK=1TON
593 TG=TG+Y(K,4)
594 NEXT
596 PRINT:PRINT"TOTAL GAS COST";:PRINT
USING"$$###.###.###":TG
598 GOTO 710
    
```

End

## Around in Circles

You can reduce the time needed to plot the circle in Emmett Carmody's pie chart program ("Graphic Results," April 1984, p. 188) by making these minor changes:

```

30 FOR A=0 TO 1.57 STEP .1
45 R1=R*COS(A):R2=R*SIN(A)
50 PSET(197+R1,32+R2):PSET(197-R1,32-R2)
55 PSET(197+R2,32-R1):PSET(197-R2,32+R1)
    
```

The revised program plots opposite points on each quarter of the circle each time through the loop, and plotting time drops from 16 to 4 seconds.

Michael ShROUT  
200 N. 35th Ave. #98  
Greeley, CO 80631

## Is BASIC holding you back? pC won't.

Allegra Software presents pC, a structured, high-level language and compiler designed to meet your TRS-80 programming needs. pC is simple, fast and efficient. And its 100+ page manual includes an excellent tutorial introduction to structured programming.

Other pC features:

- Fast, one-step operation
- pC programs run up to 50 times faster than comparable BASIC programs
- pC generates optimized Z80 native code — no need to struggle with assembly language
- programs can be compiled separately — create your own program libraries
- comes with a library of over 60 functions, supporting disk files, graphics memory management, strings, device-independent I/O and I/O redirection
- pC subroutines can be called from assembly language, and assembly-language subroutines can be called from pC
- run time library requires less than 2K of memory
- no royalties to pay on pC programs you sell
- comes with free machine-language arcade game — with source code!

Requires TRS-80 Model I/III, 2 disk drives, 48K

pC Compiler ..... \$99.95

And for after hours . . .

### Sam Steel and the Stone Girl

The Place: A city like any other — only more so

The Time: August 17, 1940

The Man: Sam Steele — a private dick who could use a little folding green, not to mention some luck!

It all starts with a call from one of the city's most prominent and influential citizens. Thereafter it's a race against time to solve the case. If you don't, someone is going to die — maybe you!

Sam Steele and the Stone Girl is a private-eye adventure game

Sam Steele and the Stone Girl ..... \$24.95

Allegra Software  
P.O. Box 6593, Station J  
Ottawa, Ontario, Canada K2A 3Y7 ✓ 369

Above prices in Canadian funds. Add \$3.00 per order for shipping and handling. Ontario residents add 7% sales tax. Visa and MasterCard accepted. No CODs, please. TRS-80 is a trademark of Tandy Corp.

## INTELLIBURNER EPROM-EEPROM-MICROCOMPUTER PROGRAMMER

### UNIVERSAL PROGRAMMING CAPABILITIES AT AN AFFORDABLE PRICE

- Ultra Fast Programming — 2716's in 16 Seconds
- Programs & Verifies 8K thru 256K Single Voltage EPROMs
- Erases, Programs & Verifies 2815/16/17 EEPROMs
- Programs & Verifies 8748 and 8751 Series MICROCOMPUTERS\*
- Programming Characteristics Selected by Convenient Personality Jumper Plug (DIP Header)
- Program, Verify, Status, & Diagnostic Display with Tricolor LED
- Serial Interface — 3, 4, or 5 wire — 1200 to 19200 Baud
- Supports XON/XOFF and READY/BUSY Protocols

NO SPECIAL SOFTWARE REQUIRED. Transfer disk files (Intel Hex Format) to EPROM with your system's line printer or modem software. Transfer EPROM contents to disk file in Intel Hex Format with your system's modem software. Or use the supplied software\*\* to transfer any binary or ASCII file to/from EPROM.

#### PROGRAMS:

|      |       |        |       |       |       |
|------|-------|--------|-------|-------|-------|
| 2758 | 2716  | 27128  | 27C16 | 8741* | 8748* |
| 2516 | 2732  | 27128A | 2815  | 8742* | 8749* |
| 2532 | 2732A | 27256  | 281c  | 8751* |       |
| 2564 | 2764  | 68764  | 2817  | 8755* |       |

\*Requires low cost personality adapter

#### SOFTWARE AVAILABLE FOR:\*\*

CP/M systems on 8" SSSD - many 5 1/4 formats  
TRS-80 Model I & III TRSDOS  
Heath IB/H89 HDOS & CP/M  
ZENITH Z90 & Z100 CP/M - Z100 ZDOS  
KAYPRO II/IV  
IBM PC-DOS

IntelliBurner Programmer with Software \$ 269.00  
RS-232 Interconnect Cable 9.00

Low Cost "DumBurner" serial programmers harness the power of your personal computer with the supplied software for full programming capabilities:

|                                                                                  |           |
|----------------------------------------------------------------------------------|-----------|
| DumBurner II Programmer for 28 Pin and 24 Pin EPROMs and EEPROMs (with Software) | \$ 189.00 |
| DumBurner II Bare PC Board, Plans & Software                                     | 39.00     |
| 16K/32K DumBurner for 24 Pin EPROMs (with Software)                              | 139.00    |
| 16K/32K DumBurner PC Board, Plans, & Software                                    | 25.00     |

Add \$2.00 Shipping & Handling. C.O.D.s accepted. Foreign Orders add required postage. Specify Environment and Media Requirements. IB, H89, Z100 are TM Heath/Zenith; CP/M is TM Digital Research; TRS-80 Model III is TM Tandy Corp; IBM-PC and PC-DOS are TM International Business Machines.



ROSS CUSTOM ELECTRONICS

1307 Darlene Way - Suite A12  
Boulder City, Nevada 89005

PHONE (702) 293-7426

✓ 370