Catalog Number 26-3820

PORTABLE COMPUTING with the MODEL 100

Radio Shack
TRS-80
COMPUTER PRODUCTS

TERMS AND CONDITIONS OF SALE AND LICENSE OF RADIO SHACK COMPUTER EQUIPMENT AND SOFTWARE PURCHASED FROM A RADIO SHACK COMPANY-OWNED COMPUTER CENTER, RETAIL STORE OR FROM A RADIO SHACK FRANCHISEE OR DEALER AT ITS AUTHORIZED LOCATION

LIMITED WARRANTY

I. CUSTOMER OBLIGATIONS

A. CUSTOMER assumes full responsibility that this Radio Shack computer hardware purchased (the "Equipment"), and any copies of Radio Shack software included with the Equipment or licensed separately (the "Software") meets the specifications, capacity, capabilities, versatility, and other requirements of CUSTOMER.

B. CUSTOMER assumes full responsibility for the condition and effectiveness of the operating environment in which the Equipment and Software are to function, and for its installation.

II. RADIO SHACK LIMITED WARRANTIES AND CONDITIONS OF SALE

A. For a period of ninety (90) calendar days from the date of the Radio Shack sales document received upon purchase of the Equipment, RADIO SHACK warrants to the original CUSTOMER that the Equipment and the medium upon which the Software is stored is free from manufacturing defects. THIS WARRANTY IS ONLY APPLICABLE TO PURCHASES OF RADIO SHACK EQUIPMENT BY THE ORIGINAL CUSTOMER FROM RADIO SHACK COMPANY-OWNED COMPUTER CENTERS, RETAIL STORES AND FROM RADIO SHACK FRANCHISEES AND DEALERS AT ITS AUTHORIZED LOCATION. The warranty is void if the Equipment's case or cabinet has been opened, or if the Equipment or Software has been subjected to improper or abnormal use. If a manufacturing defect is discovered during the stated warranty period, the defective Equipment must be returned to a Radio Shack Computer Center, a Radio Shack retail store, participating Radio Shack franchisee or Radio Shack dealer for repair, along with a copy of the sales document or lease agreement. The original CUSTOMER'S sole and exclusive remedy in the event of a defect is limited to the correction of the defect by repair, replacement, or refund of the purchase price, at RADIO SHACK'S election and sole expense. RADIO SHACK has no obligation to replace or repair expendable items.

B. RADIO SHACK makes no warranty as to the design, capability, capacity, or suitability for use of the Software, except as provided in this paragraph. Software is licensed on an "AS IS" basis, without warranty. The original CUSTOMER'S exclusive remedy, in the event of a Software manufacturing defect, is its repair or replacement within thirty (30) calendar days of the date of the Radio Shack sales document received upon license of the Software. The defective Software shall be returned to a Radio Shack Computer Center, a Radio Shack retail store, participating Radio Shack franchisee or Radio Shack dealer along with the sales document.

C. Except as provided herein no employee, agent, franchisee, dealer or other person is authorized to give any warranties of any nature on behalf of RADIO SHACK.

D. EXCEPT AS PROVIDED HEREIN, RADIO SHACK MAKES NO EXPRESS WARRANTIES, AND ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE IS LIMITED IN ITS DURATION TO THE DURATION OF THE WRITTEN LIMITED WARRANTIES SET FORTH HEREIN.

E. Some states do not allow limitations on how long an implied warranty lasts, so the above limitation(s) may not apply to CUSTOMER.

III. LIMITATION OF LIABILITY

A. EXCEPT AS PROVIDED HEREIN, RADIO SHACK SHALL HAVE NO LIABILITY OR RESPONSIBILITY TO CUSTOMER OR ANY OTHER PERSON OR ENTITY WITH RESPECT TO ANY LIABILITY, LOSS OR DAMAGE CAUSED OR ALLEGED TO BE CAUSED DIRECTLY OR INDIRECTLY BY "EQUIPMENT" OR "SOFTWARE" SOLD, LEASED, LICENSED OR FURNISHED BY RADIO SHACK, INCLUDING, BUT NOT LIMITED TO, ANY INTERRUPTION OF SERVICE, LOSS OF BUSINESS OR ANTICIPATORY PROFITS OR CONSEQUENTIAL DAMAGES RESULTING FROM THE USE OR OPERATION OF THE "EQUIPMENT" OR "SOFTWARE". IN NO EVENT SHALL RADIO SHACK BE LIABLE FOR LOSS OF PROFITS, OR ANY INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY BREACH OF THIS WARRANTY OR IN ANY MANNER ARISING OUT OF OR CONNECTED WITH THE SALE, LEASE, LICENSE, USE OR ANTICIPATED USE OF THE "EQUIPMENT" OR "SOFTWARE". NOTWITHSTANDING THE ABOVE LIMITATIONS AND WARRANTIES, RADIO SHACK'S LIABILITY HEREUNDER FOR DAMAGES INCURRED BY CUSTOMER OR OTHERS SHALL NOT EXCEED THE AMOUNT PAID BY CUSTOMER FOR THE PARTICULAR "EQUIPMENT" OR "SOFTWARE" INVOLVED.

B. RADIO SHACK shall not be liable for any damages caused by delay in delivering or furnishing Equipment and/or Software.

C. No action arising out of any claimed breach of this Warranty or transactions under this Warranty may be brought more than two (2) years after the cause of action has accrued or more than four (4) years after the date of the Radio Shack sales document for the Equipment or Software, whichever first occurs.

D. Some states do not allow the limitation or exclusion of incidental or consequential damages, so the above limitation(s) or exclusion(s) may not apply to CUSTOMER.

IV. RADIO SHACK SOFTWARE LICENSE

RADIO SHACK grants to CUSTOMER a non-exclusive, paid-up license to use the RADIO SHACK Software on **one** computer, subject to the following provisions:

A. Except as otherwise provided in this Software License, applicable copyright laws shall apply to the Software.

B. Title to the medium on which the Software is recorded (cassette and or diskette) or stored (ROM) is transferred to CUSTOMER, but not title to the Software.

C. CUSTOMER may use Software on one host computer and access that Software through one or more terminals if the Software permits this function.

D. CUSTOMER shall not use, make, manufacture, or reproduce copies of Software except for use on **one** computer and as is specifically provided in this Software License. Customer is expressly prohibited from disassembling the Software.

E. CUSTOMER is permitted to make additional copies of the Software **only** for backup or archival purposes or if additional copies are required in the operation of **one** computer with the Software, but only to the extent the Software allows a backup copy to be made. However, for TRSDOS Software, CUSTOMER is permitted to make a limited number of additional copies for CUSTOMER'S own use.

F. CUSTOMER may resell or distribute unmodified copies of the Software provided CUSTOMER has purchased one copy of the Software for each one sold or distributed. The provisions of this Software License shall also be applicable to third parties receiving copies of the Software from CUSTOMER.

G. All copyright notices shall be retained on all copies of the Software.

V. APPLICABILITY OF WARRANTY

A. The terms and conditions of this Warranty are applicable as between RADIO SHACK and CUSTOMER to either a sale of the Equipment and/or Software License to CUSTOMER or to a transaction whereby RADIO SHACK sells or conveys such Equipment to a third party for lease to CUSTOMER.

B. The limitations of liability and Warranty provisions herein shall inure to the benefit of RADIO SHACK, the author, owner and/or licensor of the Software and any manufacturer of the Equipment sold by RADIO SHACK.

VI. STATE LAW RIGHTS

The warranties granted herein give the **original** CUSTOMER specific legal rights, and the **original** CUSTOMER may have other rights which vary from state to state.

Portable Computing with the Model 100

10 9 8 7 6 5 4 3 2 1

Portable Computing with the Model 100

Techniques and Applications in:

- Word Processing
- Telecommunication
- Databases
- Calculations

by George Stewart

Radio Shack®
A DIVISION OF TANDY CORPORATION

PART 1: TECHNIQUES

Part 1: Techniques

Introduction

What Is Portable Computing?

"Sure, I've seen the ads for portable computers. But I never could see the need...." Frank Kelly would be darned if he'd be caught lugging a 25-pound box all the way to Minneapolis.

Kelly stood attentively in front of Robert M. Fowler's desk. His boss had been telling him about problems at the new district office. The sales crew was having trouble cracking some of the big accounts. Fowler wanted to know why. Kelly was to drop everything and jump on a plane that same afternoon.

And as if that wasn't enough, the president wanted him to take along one of those so-called portable computers....

Fowler opened a desk drawer and slid out a keyboard about the size of a thick notebook. "Kelly, this one's different. Word processing, telecommunication, computing — it's all built-in. Carry it with you. Use it on the airplane, at the hotel, and at clients' offices. When you get back, then you can tell me if you see a need for a portable computer. Okay?"

Kelly held the thing in his hands: a TRS-80 Model 100. It couldn't weigh more than four pounds. The keyboard looked every bit as good as his desktop computer's. The liquid-crystal display was largest he had seen. Looking at the rear panel, he read the connector labels with interest: cassette, phone, printer, RS-232C. More input/output capability than his desktop unit. It would take a while to absorb it all...

By the end of his trip, Frank Kelly had changed his mind about portable computing. After all, with just a little study and practice, he was able to do all of the following things with ease:

While on the airplane, he drafted an introduction to the annual market forecast he'd been preparing at the office, and laid out a daily agenda for the trip. From the hotel that evening, he transmitted the memo to the home office computer so Fowler could check it over. The next morning, he called the office computer again and downloaded the current price and delivery schedules for his company's product.

He accompanied the district sales manager on calls to major accounts around the city. Kelly impressed prospective clients by transmitting a job spec to headquarters and receiving his company's detailed bid within the hour. Quite often, he was able to connect the Model 100 to a printer in the client's office and produce a hard copy of the calculations.

Keeping track of expenses was easier than ever. Kelly entered them into the Model 100, where they could be sorted for easier entry into an expense report when he returned to the home office.

Kelly spent a few evenings at the hotel entertaining himself with the Model 100. He called up an on-line information service to see stock prices and chat with a few cross-country users. He used a BASIC program (his boss must have typed it in) that computed golf handicaps.

That's portable computing!

About This Book

This book emphasizes realistic and practical applications of portable computing for your Model 100 — in an airplane, a hotel, or a client's office. But the techniques we present are just as useful when you get back to the office and plunk your portable computer on the corner of your desk for quick access to word processing, telecommunication, data management, and computations.

In Part 1, you'll learn how to use Model 100's five built-in applications programs, and, just as important, how to make the different programs interact (so you can use calculations from a BASIC program in a letter, for example).

The full potential of the Model 100 will be yours when you learn to customize BASIC programs and write your own. The application programs described in Part 2 and listed in the Appendix will get you started. You don't have to know BASIC programming to benefit from these sample programs, but whenever you're ready to try your hand at it, they will help you understand exactly how it's done.

The Software Cassette included with this book contains all the application programs. To load and run them on your Model 100, all you need is a standard audio cassette recorder and computer cable.

To use the programs and procedures in this book effectively, you should be familiar with the basic operation of your Model 100. This includes using a cassette recorder, printer, and the built-in modem. The owner's manual provides all of this information.

Notation and Other Details

This book contains two kinds of text: material for you to read and material for you to type into the computer or find on the screen. Computer-related material is indicated by a special typeface.

`Dot matrix typeface` highlights text that you enter into the computer or find on the screen.

Key symbols such as [F1] or [ENTER] indicate Model 100 keys that you press to obtain special functions.

Whenever [CTRL], [SHIFT], [GRPH], or [CODE] appears in combination with another key, as in [CTRL]-[U] or [SHIFT]-[PRINT], press both keys down at once (press the first key and hold it down while you press the second key; release both at once).

When other keys appear together, as in [F1] [ENTER], press the keys one at a time.

The symbol ▢ represents the spacebar.

The Model 100 can display labels for the function keys [F1] through [F8]. To make the function key labels appear on the screen at any time, press the [LABEL] command key. Unless otherwise noted in the text, you should have the labels showing on the display.

1 / TRS-80, Take a Letter

Writing letters, notes, and memos on the Model 100 is as simple as typing on the keyboard. In this chapter, we present an overview of Model 100 word-processing functions and then demonstrate the following applications:

- creating a document
- combining text from two documents
- editing efficiently

Overview of Model 100 Word Processing

TEXT is the application program built into the Model 100 that lets you create or modify documents.

To create a new word-processing document, you select the TEXT program from the main menu (figure 1.1) and type in the name of a file that doesn't already exist on the menu. File names must start with a letter and be no more than six characters long.

To edit an existing document, simply select that document on the main menu; the Model 100 starts TEXT and loads the document automatically.

A *document* is any file marked .DO on the main menu. The two other kinds of Model 100 files are BASIC programs marked .BA and machine-language programs marked .CO. TEXT cannot use either of these kinds of files. Documents may contain ordinary word-processing materials such as letters or reports; data such as tables or lists; and BASIC programs that are stored in ASCII form (see Chapter 5).

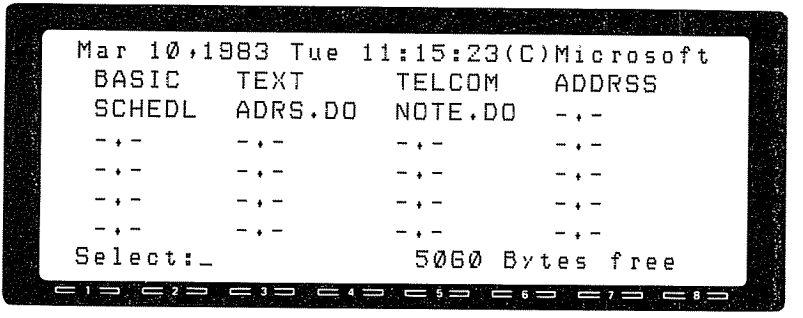


Figure 1.1. The main menu.

Text Storage. How much text can you store in the Model 100? That depends on how much RAM (random-access memory) you purchased with your unit and how much RAM your own files use. To find out the character capacity remaining at any time, look at the main menu's lower right corner: `XXXX Bytes free`.

For word processing, a byte holds one character of text. Table 1.1 shows the maximum character capacity for 8K-, 16K-, 24K-, and 32K-byte units with no RAM files present (and after having executed the BASIC commands `CLEAR 256,MAXRAM` and `MAXFILES=0`). The table also gives the rough equivalent in double-spaced, typewritten pages (assuming one double-spaced page contains 27 lines of 60 characters each).

	8K	16K	24K	32K
Characters	5329	13521	21713	29905
Pages	3.3	8.3	13.4	18.5

Table 1.1. Maximum text capacity of the Model 100.

For many applications you must augment the Model 100's internal storage, usually by means of cassette tapes. However, while traveling, you may not have ready access to a cassette recorder or other means of storing documents outside the Model 100. At these times, smart memory-management techniques can help you get the most out of the computer's built-in memory, forestalling the `Memory full` message. We'll stress these techniques throughout this chapter and, indeed, throughout this book.

Now let's get on to the specifics of word processing with the Model 100.

Creating a Document

From the Model 100's main menu, press the spacebar until the word `TEXT` is highlighted. Press (ENTER). When the Model 100 asks you what file to edit, type in a name you haven't used and press (ENTER). We use the name `MEM419` because the memo date is April 19. Type in the memo that's shown in figure 1.2.

After you type in the document's name, the Model 100 shows you a blank screen. Type in the sample memo, duplicating the format as much as possible within the limits of a 40-column display. Figure 1.3 illustrates how the document appears on the Model 100 display.

```
                    Memo


To: Larry Thompson
cc: Robert M. Fowler
From: Frank Kelly
Re: New sales prospects
Date: 4/19/83

-----------------------------------------

During my recent trip to the Minneapolis
office, I visited with three companies
and found a definite interest in our
products. Here are the company names and
personal contacts:

(Insert list here)

Please have your sales people make
follow-up calls immediately. I'd also
like to receive a progress report in a
few weeks.

Larry, let me know if I can be of further
assistance.
```

Figure 1.2. Type this document into a TEXT file.

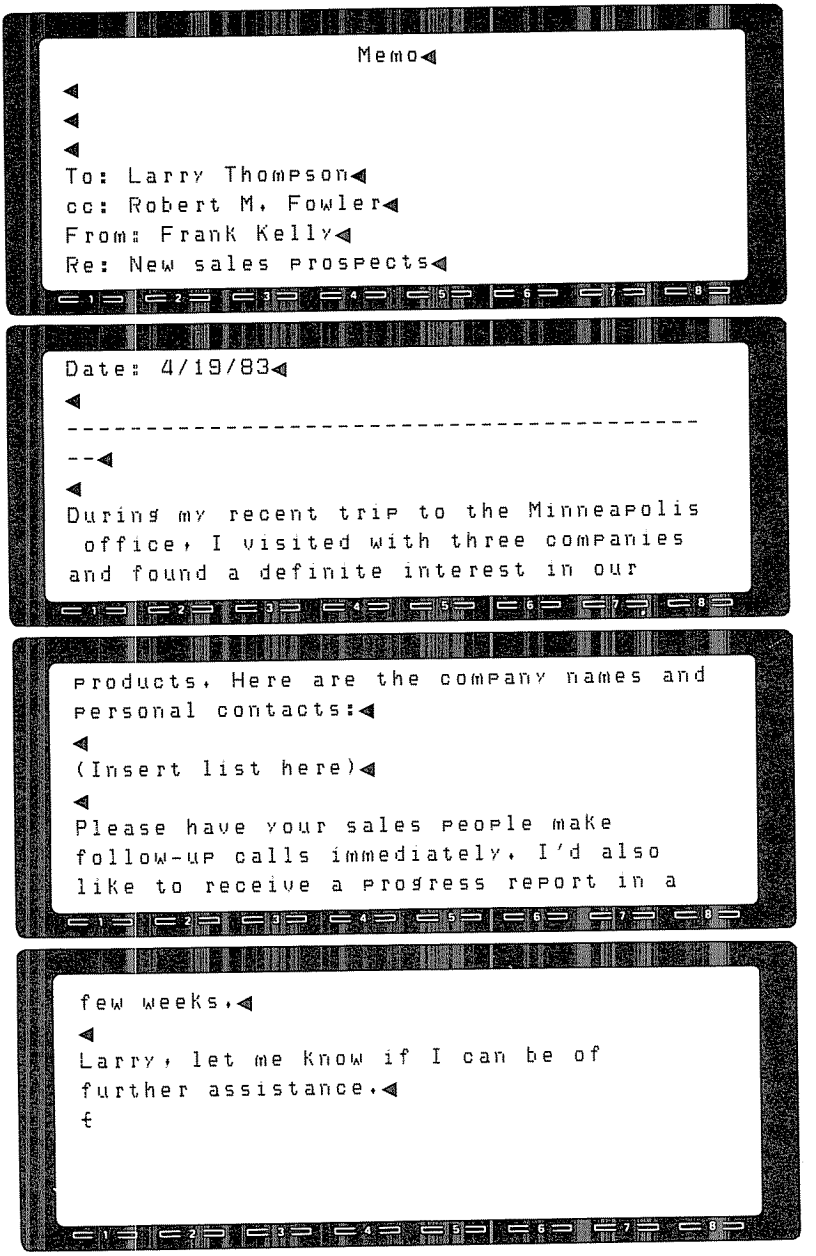


Figure 1.3. The memo as it appears on the Model 100 screen.

Let's look at your handiwork. In the first line, did you indent the word MEMO by pressing the spacebar ▭ repeatedly? That's fine, but using the TAB key is better because it uses up less memory. It takes 33 bytes to space over to column 33 on the display, but only four bytes using TAB. The tab stops are permanently set at columns 1, 9, 17, 25, and 33. To indent to any other column position, press ▭ to add the necessary spaces *after* the tabs. For example, to indent to column 20, tab twice then space three times.

Look at the body of the memo. Did you end your lines by pressing ENTER at the end of each one? Hopefully not — that uses up two bytes of memory needlessly (one for a carriage-return code, one for a line-feed code).

Let the Model 100 end the lines automatically with its *auto-wrap* facility, which starts a new line when your typing approaches the right margin; this only uses a single byte of memory. Besides conserving memory, the auto-wrap facility gives you more flexibility when printing the letter or transmitting it to another computer. The only time you should press ENTER is when you must force a line ending, create a blank line, or end a paragraph. You can always tell where you have pressed ENTER. The Model 100 marks those places with a sideways triangle like this: ◀.

Look closely at the left margin of your memo, using the cursor keys (arrows) to scroll through it. Is the margin a little ragged? That happens when the auto-wrap facility breaks a line *before* a space instead of after it. But don't worry — the raggedness disappears when you print or transmit the document.

Look at this line from figure 1.3:

```
and found a definite interest in our
```

There appear to be four spaces after our. How can you be sure there is only one? If you typed only one, only one should be there, but occasionally during editing you might accidentally insert one or more extras. These extra spaces can upset the format of the letter when you print it. To be sure no blanks are hidden at the end of a line, press CTRL-→. The cursor should move onto or just past the last letter in the last word. If it goes further, extra blanks are hidden. Press BKSP until the cursor is on or just past the last letter on the line.

After making sure that the text is free of extra spaces and otherwise matches figure 1.3, close the document by pressing F8. The memo now shows up on the main menu as a new document.

To summarize the key points about creating documents:

- Use TAB for indenting, not the spacebar.

- Let auto-wrap end lines; press (ENTER) only to end short lines, insert blank lines, and end paragraphs.
- Watch out for hidden spaces at the ends of auto-wrapped lines; these spaces upset the format when the document is printed.

Getting Data from Another File

When traveling with the Model 100, you'll find it convenient to record names, addresses, and other data in the computer rather than using bulky notebooks or scraps of paper. You can retrieve the information from the Model 100 quickly and accurately later on. But just as important, you'll be able to take data from one document and copy it into another, saving you the trouble and possible errors of extra typing.

Let's illustrate the point using Frank Kelly's memo of April 19. Frank hasn't yet inserted the list of new sales prospects in the memo. But he doesn't need to look them up in a telephone directory and type them in: He has already entered them in the Model 100's address file, ADRS.DO, so he can simply copy them into the memo.

To continue with this example, you need to store Frank's address information in your computer. From the main menu, select TEXT again and name `ADRS.DO` as the document to edit. Delete anything you've put there already (save it on tape first). Type in the information from figure 1.4 and then close the document by pressing (F8).

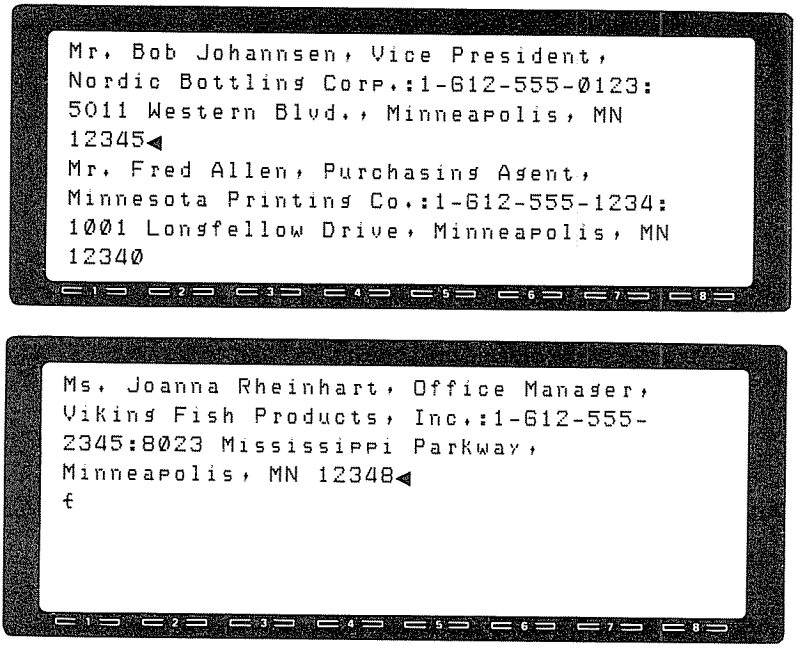


Figure 1.4. Frank Kelly's address list as it appears on the Model 100 screen.

Now you're ready to copy the prospect list into the memo.

Select ADRS.DO. For your convenience, press (LABEL) so the function-key definitions appear on the screen. Now press the Select key, (F7). Press (CTRL)-(Y). The entire list should appear highlighted. Press the Copy key, (F5). The highlighting disappears, and the document has now been copied into a temporary holding area called the paste buffer.

Close the address document ADRS.DO and open Frank Kelly's memo document. Now follow these steps to insert the information into the memo:

1. Press the Find key, (F1).
2. Type a short string like `(Insert`.
3. Press (ENTER). The cursor should move to the line that says `(Insert list here,)`.
4. Use (DEL) repeatedly to delete the entire line.
5. Press (PASTE) to transfer the list into the memo.

You now have three copies of the sales prospect list: in ADRS.DO, in MEM419.DO, and in the paste buffer. To conserve memory, eliminate the third copy. Do so by pressing the Select key (F7) and then pressing the Copy key (F5).

Adjusting the Format

Having copied the raw data into the memo, we're going to use the editing keys to put it into a neat format (as shown in figure 1.5).

```
interest in our products. Here are the
company names and personal contacts:

Mr. Bob Johannsen, Vice President, Nordic
Bottling Corp. Phone 1-612-555-0123. 5011
Western Blvd., Minneapolis, MN 12345

Mr. Fred Allen, Purchasing Agent,
Minnesota Printing Co. Phone
1-612-555-2345. 1001 Longfellow Drive,
Minneapolis, MN 12340

Ms. Joanna Rheinhart, Office Manager,
Viking Fish Products, Inc. Phone
1-612-555-2345. 8023 Mississippi Parkway,
Minneapolis, MN 12348

Please have your sales people make
follow-up calls.
```

Figure 1.5. A portion of the memo with the addresses inserted and edited for a neat appearance.

First put a blank line before each prospect name by moving the cursor to the start of each name and pressing (ENTER).

Next, replace the colons before each telephone number with the word "Phone." The following keystroke sequence shows how you can speed this by using Find (F1) and (PASTE). You may have to modify the Find commands a little if your prospect list differs from the one shown in figure 1.4:

1. Move to the top of the memo: (CTRL)-(↑)
2. Find the first telephone number: (F1) : 1 (ENTER)
3. Replace the colon with a space: (DEL) (SPACE)
4. Find next telephone number: (F1) (ENTER) (F1) (ENTER)

Repeat steps 3 and 4 until no additional : 1 occurrences are found. Then use this sequence to insert the word "Phone" followed by a space before each telephone number:

5. Move to the top of the memo: (CTRL)-(↑)
6. Find the first telephone number: (F1) 1 - (ENTER)
7. Insert the word "Phone" followed by a space: Phone (SPACE)

Move the cursor to the first letter in Phone. Then continue with this key sequence:

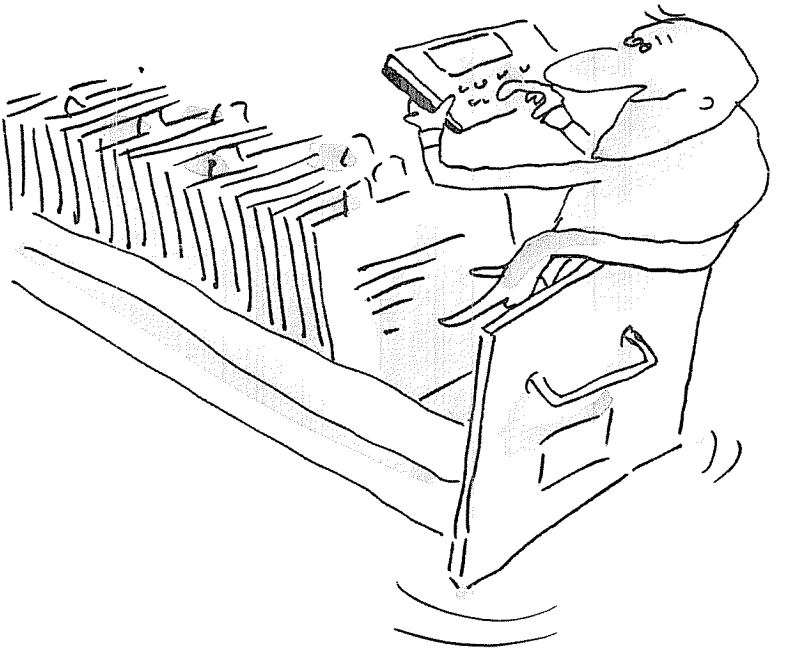
8. Select the word: (F7) (SHIFT) (SPACE)
9. Copy it into the paste buffer: (F5)
10. Find the next telephone number: (F1) (ENTER)
11. Insert the word "Phone": Phone

Repeat steps 10 and 11 to get the third telephone number.

Last, delete the colon that follows each telephone number and replace it with a period and a space. Try to come up with an efficient keystroke sequence for doing this.

To recap the techniques for efficient editing:

- Press (LABEL) to display the function-key definitions.
- After a Select-and-Copy operation, empty the paste buffer with (F7) (F5). This frees up useful memory.
- Use (F1) and (PASTE) to speed repetitive operations.



2 / Storing and Printing Files

In the preceding chapter, you learned how to create and edit a document file by using TEXT. What can you do with a file? In this chapter, we describe some very important file operations:

- saving documents on cassette tape
- copying documents into memory
- renaming documents
- printing documents

Saving Documents on Tape

Cassette tape storage serves two purposes: First, it gives you a safe copy of the original, in case you ever want to return to it. Second, it lets you create long texts that exceed your Model 100's storage capacity.

We'll cover these uses one at a time.

Safe Copies. Before doing extensive modifications to an important document, always make a cassette copy of the original (if your Model 100 has sufficient memory, a RAM copy does just as well). You should copy the original before modifying it because your changes take effect instantly. In general, changes cannot be undone, short of retyping the original material.

The exception to this rule is that the most recent contents of the paste buffer can be recovered at any time by pressing the (PASTE) key. However, this only gives you one level of safety since previous cuts cannot be recovered. Furthermore, the technique of emptying the paste buffer to conserve memory eliminates even this safeguard. Having a cassette copy is good insurance in any case.

Here's the procedure for copying a document onto cassette tape using the Frank Kelly memo from the preceding chapter as an example:

1. Connect the recorder, insert a *blank* cassette tape, and put the recorder in Record mode.
2. Open the file that contains Frank Kelly's memo: From the main menu, press the spacebar until the memo name is highlighted, then press (ENTER).
3. Press the Save key (F3). In response to the `Save to:` prompt, type `CAS:`.
4. If you plan to record multiple documents on the same side of the tape, include a name as well, for example: `CAS:MEM419`.
5. Press (ENTER). The Model 100 turns on the recorder motor and copies the text onto the cassette tape.

6. As an added precaution, you may want to make a second copy on the same or a different cassette.

The steps for loading the tape text into the computer are equally simple:

1. Rewind the tape and put the recorder in Play mode.
2. Create a *new* file using TEXT.
3. Press the Load key (F2), and in response to the `Load from:` prompt, type `CAS:`. If you have more than one document stored on the tape, specify the file name, too, for example, `CAS:MEM419`.
4. Press (ENTER). The Model 100 turns on the recorder and loads the first document that is stored on the tape (or the one you named).

Lengthy Texts. Besides giving you a safe copy of your document, cassettes are useful when you have to write reports or other texts that exceed your Model 100's storage capacity. In such cases, you can break the text up into several documents, as follows:

1. Select TEXT, create `PART1.DO`, and type in the first part of your text. When memory is almost full, save the text to `CAS:PART1`.
2. Close the document, select BASIC, and erase the document with the command `KILL "PART1.DO"`. This releases the Model 100's memory so you can continue typing in more text.
3. Exit from BASIC to the menu, select TEXT, and create `PART2.DO`.

Continue as with steps 1, 2, and 3 until you have finished writing and saving the entire text in `PART2.DO`, `PART3.DO`, and so forth.

Copying Documents into Memory

For small documents, you can make extra copies in memory rather than on cassette. If you have a 24K or 32K RAM unit, RAM copies are feasible even with multi-page documents. This method is much faster and more convenient than using cassettes.

Suppose you composed a form letter that you want to customize for several individuals. You want the original to remain unchanged, so you use a *copy* of it for each customized letter.

To make a RAM copy, start by opening the document to be copied. Then follow this key sequence:

1. Move to the top of the file: (CTRL)(↑)
2. Select the entire document: (F7) (CTRL)(↓)
3. Copy it to the paste buffer: (F5)
4. Close the document: (F8)

Now open a new document and press (PASTE). The text is copied into the new document — but also remains in the paste buffer in case you need

it again. When you are through making copies of the form letter, empty the paste buffer by pressing (F7) then (F5) to free up memory for other uses.

To summarize the techniques of copying documents:

- Copy important documents before you begin changing them.
- If a text exceeds memory capacity, break it into smaller documents which can be stored separately on cassette.
- Put form letters into the paste buffer for quick recall while composing customized versions.

Renaming a Document

It is impossible for two documents to share the same name. Depending on the application program you are using, if you try to create a duplicate name, the original document is replaced by the new one or the attempt to create it fails. Hence the need for renaming documents.

The procedure is quite simple. Select BASIC from the main menu, then type NAME "current file name" AS "new file name" (ENTER), substituting the appropriate names for *new file name* and *current file name*. Press (F8) to return to the main menu, and you should see the renamed file.

For example, to change ABC.DO to DEF.DO, select BASIC and type NAME "ABC.DO" AS "DEF.DO"(ENTER).

Printing A Document

If you have a printer connected, you can see what the Frank Kelly memo from the preceding chapter looks like on paper.

Open the memo document, then press (SHIFT)-(PRINT). The Model 100 prompts you with Width:. Enter the line length measured in columns you want on the printout (use 51 for the memo). The Model 100 then prints out the entire document, auto-wrapping any lines that exceed the length you entered.

The Model 100 starts each printed line at the leftmost print position of your printer. If this is undesirable, adjust the paper to the left a half-inch or so before printing.

Special effects. Suppose you want to underline the word "immediately" in the memo. The Model 100 display doesn't provide underlining, but most printers do. You simply tell the printer where to start and stop underlining. With most printers, a pair of printer control codes turn underlining on and off. For example, Radio Shack printers interpret control-O as "underlining on" and control-N as "underlining off." Check your printer operation manual to find out which codes your printer recognizes.

TEXT lets you embed these codes in a document right along with the text. To try out the technique, open the Frank Kelly memo and position the cursor over the first letter of `immediately`.

Press (CTRL)(P). This gets the Model 100 ready to accept a printer control code, but has no effect on the display. Press (CTRL)(O). `^O` appears on the screen to tell you that a control-O code is stored there.

Now move the cursor to just after `immediately` and type (CTRL)(P) again, then (CTRL)(N). `^N` appears on the display (see figure 3.1).

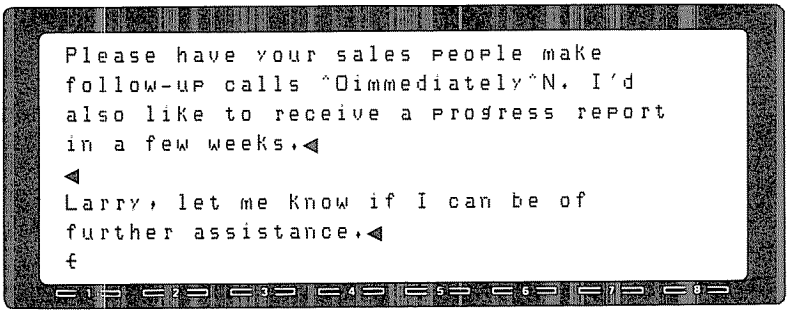


Figure 3.1. A portion of the Frank Kelly memo with printer control codes added around the word *immediately*.

That was easy enough. Now you'd probably like to see if it works. Unfortunately, if you try to print out the document using (SHIFT)(PRINT), you'll find that the Model 100 does not send the needed printer codes to the printer. Instead, it sends the same "dummy characters," `^O` AND `^N`, that you see on the screen.

To force the Model 100 to send the actual printer codes to the printer, you can use the Save key, (F3). When the Model 100 prompts you with `Save to:` type `LPT:`, which means line printer. Now you'll see the underlining. But there's a new problem — the auto-wrap facility doesn't work when you use the TEXT Save function, so the text lines may be too wide for the paper.

An application program described in Part 2 remedies both of these problems (see PRINT.BA). In addition, it adds page numbers, headings, and margins to the printout, so you don't need to adjust the paper manually before printing.

Here's a summary of printing techniques:

- Use TEXT's (SHIFT)(PRINT) function to get a rough copy of your document.

- The (SHIFT)-(PRINT) function doesn't provide margins. Adjust the paper manually before beginning.
- To enter a printer control code for special effects, type (CTRL)-(P) followed by the needed control character.
- Use the application program PRINT.BA to print documents that require margins or contain printer control codes.


Niculae Asciu

3 / Telecommunication

In this chapter, we present a crash course in the principles and terminology you need for practical telecommunication. The more you use telecommunication, the more you'll keep running into the jargon explained in this chapter. Don't try to absorb all the information right away; you can always refer back to the explanations and definitions later.

We also go through the steps needed to start communication (go online) with a "host" computer and troubleshoot some of the common problems that arise.

Before beginning, we assume you know how to connect your computer to the telephone lines and call up another computer. If you haven't done so already, you should read chapter 11 ("Computer-to-Computer Communications") in the Model 100 Owner's Manual, and practice using the procedures outlined there. You'll benefit much more from this chapter if you have some hands-on experience.

You should also have access to one of the information services such as CompuServe, Dow Jones, or the Source. You could also use one of the many free computer bulletin board services around the country. Whatever you choose, your local Radio Shack salesperson can help you get connected.

Model 100 Data Telecommunication

Data telecommunication is the transfer of data between computers using the telephone lines. Computers "talk" and "listen" over the telephone lines using devices called *modems*. The Model 100 includes a built-in modem with the added capability of dialing the "host" computer to establish the communication link. Instead of using the built-in modem, you can connect an external modem using the Model 100's RS-232C interface. If you have a high-speed modem, you might find this worthwhile. However, using the Model 100's built-in modem is by far the easiest way.

Model 100 software provides two ways of using telecommunication: first, through TELCOM, which allows direct keyboard communication as well as file transfers to and from your computer; and second, through BASIC programs and commands (for instance, TERM.BA in Part 2).

Before communicating with another computer, you need to find out several technical details about the host system to ensure that the Model 100 is compatible:

- What is the information rate, measured as baud or in bits per second?
- How many bits are used per data-word?

- Is there parity checking? If so, is it even or odd?
- How many stop bits are used?
- Does the host use the XON/XOFF protocol?

Having determined the technical requirements of the host computer, you must configure the Model 100 to match exactly. The Model 100 is quite flexible, letting you select from a wide variety of communication options to suit almost any host computer. When using TELCOM, you specify the desired options with the STAT command. When using BASIC programs, you specify the desired options as part of the device specification. You'll see examples of both methods later in this chapter and in the TERM.BA application program described in Part 2. But first, let's explain what each communication option means.

Information rate. How fast do computers communicate over telephone lines? The rate is measured in bits per second or *bps*. Bits are the on-off signals on which digital computers are based. The term *baud*, named after a French telegraph officer, J.M.E. Baudot, is technically different from bps, but the two are practically interchangeable for 300 and 1200 bps communication.

The most frequently used information rate for personal computers is 300 bps, although 1200 bps is becoming common. CompuServe, for example, offers both 300 bps and 1200 bps telephone access numbers. At 300 bps, it takes about one minute to send a single, double-spaced page of text. At 1200 bps, it takes about 15 seconds to send the same text.

The Model 100's built-in modem communicates at 300 bits per second; to communicate at higher rates requires that you connect the Model 100 to an external 1200 bps modem using the computer's RS-232C interface.

A communication option in the Model 100 lets you specify the information rate and the modem type (internal or external).

Word length. In telecommunication, computers send one bit at a time. These bits are usually grouped into meaningful units called *data-words* or *words*; we use the term data-words to prevent confusion with ordinary words. A data-word usually represents one character of text.

The length of a data-word depends on which character coding system is being used. Most personal computers use the ASCII coding system, in which data-words are seven bits long.

The seven-bit ASCII system allows transmission of 128 different characters including all the standard text, punctuation, and control characters. However, it does not allow sending of graphic characters such as those generated from the Model 100 keyboard when you press GRAPH together with certain other keys.

If your document contains graphic characters, your computer and the host computer must both be set up to handle eight-bit data-words. Since the Model 100's graphic characters are non-standard, the only time you are likely to use this capability is when communicating with another Model 100.

A Model 100 communication option sets the data-word length from six to eight bits.

Parity checking. This is a simple and fairly reliable means of detecting transmission errors (characters in which one or more bits are garbled). An extra bit, called a *parity bit*, is added to each data-word. The value of this bit (1 or 0) depends on whether there is an even or odd number of 1s in the data-word. With so-called *even parity* the added parity bit always produces an even number of 1s. With *odd parity*, the added parity bit always produces an odd number of 1s.

Figure 3.1 illustrates the three parity options: no parity, odd parity added, and even parity added.

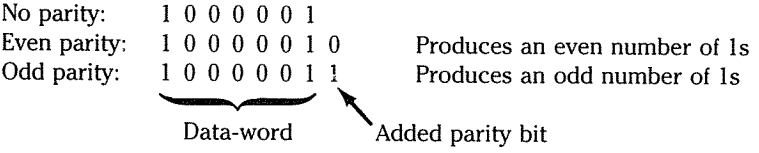


Figure 3.1. The letter "A" represented as a seven-bit data-word with various parity options.

If the host computer uses a parity bit, you have two choices: configure your Model 100 to recognize the same parity (even or odd), or ignore parity completely. When parity checking is enabled, the Model 100 prints a solid block like this ■ for each character received with a parity error. If the host computer does not use parity, then set your Model 100 for no parity as well.

A Model 100 communication option sets the parity.

Start and stop bits. The Model 100 and most other personal computers use *asynchronous communication*, in which information is organized into data-words that are sent at varying time intervals. Because of this unpredictable timing, computers need a way of indicating the beginning and end of each character. Start and stop bits serve this purpose. (In contrast, some microcomputer-to- mainframe communication involves *synchronous communication*, in which the data is sent in larger blocks of precisely spaced bits. With synchronous communication, start and stop bits are not needed.)

With asynchronous communication, one start bit is always sent before each data-word. Because start bits are a fixed requirement, the Model 100 has no start-bit option. However, either one or two stop bits can be added after the data word plus an optional parity bit. The Model 100 has a communication option which controls this parameter.

If in doubt as to the number of stop bits a host computer requires, use one stop bit. CompuServe and most other host computers use one stop bit.

In summary, each character transmitted requires:

- one start bit
- six to eight data bits
- one optional parity bit
- one or two stop bits

In the most common setting, data transmission requires a total of 10 bits for each character. Under these conditions, 300 bps produces a transmission rate of 30 characters per second.

XON/XOFF protocol. Suppose computers A and B are in the process of telecommunication. What happens if computer A sends characters faster than computer B can handle them? The XON/XOFF (x-on, x-off) protocol takes care of this problem. Computer B sends a special character called XOFF, which tells computer A to stop sending. When computer B is ready for more data, it sends an XON code, which tells computer A to resume sending.

Many host computers — CompuServe and the Source, for example — use this protocol; so does the Model 100, if you select it as a communication option. This feature is often necessary. For example, during an upload operation (in which you send a document to another computer), the remote computer transmits XOFF to your Model 100 whenever it requires a pause. The Model 100 then stops sending until the remote computer sends the XON.

Similarly, suppose you are receiving information and printing it at the same time (TELCOM's "echo" option). Because of the printer's slower speed, the Model 100 usually has difficulty keeping up with the incoming data. When it falls too far behind, it sends an XOFF to the remote computer, pausing the data flow. As soon as the Model 100 catches up, it sends an XON so communication can resume. This pause-and-resume dialogue takes place automatically without your being involved.

If the host computer uses the XON/XOFF protocol, always configure the Model 100 to do the same. If the host computer does not recognize the XON/XOFF protocol, you should disable this option. Otherwise, your com-

puter might mistakenly pause when not requested to by the host.

You can always send an XON and XOFF manually by typing on the keyboard. CTRL-S sends XOFF, and CTRL-Q sends XON. This method is convenient, for example, if you want to force a pause in communication to stop the display from scrolling while you read it.

Table 3.1 summarizes the important communication parameters you must check before trying to establish communication with another computer. The status-switch settings in the right-hand column are explained in the following pages.

Preliminaries

Setting the status switches. After determining the technical requirements for communication with a host computer, you must set the Model 100's communication status switches accordingly. The status switches are

Communication Parameter*	Typical Values	Model 100 status-switch
1. Information Rate†	300 bps	M — internal modem
		3 — external modem
	1200 bps	5 — external modem only
2. Data-word length	7 bits	7
	8 bits	8
3. Parity	even	E
	odd	O
	none	N
	ignore	I¶
4. Stop bits	1	1
	2	2
5. XON/OFF protocol	enable	E
	disable	D

Table 3.1. Checklist of Model 100 communication options.

*Numbers refer to the status-switch position.

†Higher rates are possible using a high-speed modem or direct connection to a local computer.

¶Use the ignore-parity option when you are not sure if the host computer uses even or odd parity.

a sequence of five characters, one for each option, as numbered in table 3.1. For example, to select "internal modem, seven-bit data-words, ignore parity, one stop bit, enable the XON/XOFF protocol," use the switch setting: M7I1E.

Let's select this setting. From the main menu, press the spacebar until TELCOM is highlighted, then press (ENTER). The current switch settings are displayed on the top line of the display. To change them, type STAT M7I1E and press (ENTER). (Or press (F3), which produces the word STAT automatically, then complete the command.) Type STAT (ENTER) to check the switch settings.

Following the communication switch setting is one more item, either 10 pps or 20 pps. This indicates the setting of the Model 100's built-in telephone dialer. 10 pulses per second (pps) is standard, but if your telephone exchange can handle 20 pps, you can change this setting. To do so, add ,20 after the five-character communication switch setting, for example, STAT M7I1E,20. To return to 10 pps, add ,10 after the five-character switch setting. (To change the pulse rate without changing the other settings, type STAT ,10 or STAT ,20.)

Setting the Mechanical Switches. To use the built-in modem for communication, you must also set the two mechanical switches on the left side-panel of the Model 100. The switches do not affect communication using the RS-232C connection to an external modem.

The answer-originate switch (labeled ANS-ORIG) selects the audio frequencies your Model 100's built-in modem uses to communicate with another computer. If you are originating the call, set the switch to ORIG. If someone else is originating the call and you are answering it, set the switch to ANS before putting the Model 100 on-line.

The direct-acoustic switch (labeled DIR-ACP) tells the Model 100 whether you are using a direct connection to the telephone line or using acoustic cups with the telephone handset.

Telephone access. In addition to knowing what communication options to use, you have to know the host's *telephone access number* — the number you dial to get the other computer on-line. If you call one of these numbers manually, you hear the host answer the telephone with the high-pitched "carrier" tone of its modem.

When calling commercial information services from a city, there's often a local telephone access number, regardless of where the host computer is located. Sometimes, you will call the host directly. For example, the CompuServe Information Service offers direct call-in numbers in many cities around the country. Alternatively, you may have to call a carrier

network which then routes your call to the host computer. Tymnet and Telenet are examples of carrier networks.

In any case, contact the service representative for the host system for a list of its telephone access numbers.

Log-on sequences. Every information service or computer bulletin board has a unique sequential system of qualifying incoming callers. The sequence serves two purposes:

- To alert the host that another computer is on the line. This usually requires you to press a "wake-up" control key such as ENTER or CTRL C.
- To establish that you are authorized to use the service.

These required inputs are called a log-on sequence. For bulletin-board type services, the sequence may consist of simply pressing ENTER a few times until the host prints a welcoming message. With services like CompuServe, you must press CTRL C and, upon prompting from the host, enter your user ID and password.

When calling a carrier (such as Tymnet or Telenet, mentioned previously), you must complete the carrier's log-on sequence *before* beginning the host's log-on sequence.

This network log-on sequence usually consists of a "wake-up" control key followed by a code for the desired information service. Once you complete the log-on, you will be prompted to begin the host's log-on sequence.

To avoid typing in the log-on sequence each time you call a host, you can store the sequence along with the host's telephone number in the Model 100's ADRS.DO file. This method is available only when using the built-in modem and auto-dialer. See chapter 11 of the Model 100 Owner's Manual for details on using this auto log-on technique.

Going On-line

Let's assume that you have set the Model 100 communication options for use with a host system and that your computer is linked to the telephone lines (using the built-in or an external modem).

The procedures that follow outline the steps for calling a host: first, using the Model 100's built-in dialer; then, using a telephone to dial the number manually.

To call a host system using the Model 100's built-in dialer:

1. Select TELCOM. The `Telcom:` prompt is displayed.
2. Use the STAT command to set the communication switches.

3. Press the Call function key, (F2).
4. Type in the number of the host, followed by <>. For example, type in `Call 1-555-1234 <>`. You can include the auto log-on sequence inside the angle-braces.
5. Press (ENTER).
   Note: If you have pre-stored the telephone number in the Model 100's ADRS.DO file, you may skip steps 4 and 5 and use the Find function key (F1) to retrieve the number.
6. Wait while the Model 100 displays `Calling` followed by each of the digits in the number as it dials them. Within a few seconds after the last digit appears, you should hear some noise from the Model 100's built-in speaker; this sound is the telephone ringing at the host's end. Next you should hear a brief tone, which is the carrier signal indicating the host is on-line.

   The Model 100 now goes into the Terminal mode, displaying the following labels on the bottom line:

   `Prev Down Up Full Bye`

   (If the Model 100 does not go into the terminal mode within 10 or 15 seconds of dialing the last digit, press (BREAK) (press (SHIFT) and (PAUSE) together) until the `Telcom:` prompt reappears. Recheck the telephone number. Make sure the Model 100's answer-originate and direct-acoustic mechanical switches are set correctly. Try the call again.)
7. Complete the log-on dialogue, if necessary. If the information on your screen is garbled or otherwise unexpected, see Testing and Troubleshooting below.
8. You should now be at the menu or command level of the host.

To call a host system using manual dialing:

1. Start TELCOM. The `Telcom:` prompt is displayed.
2. Use the STAT command to set the communication switches appropriately.
3. Call the host manually using your telephone.
4. Wait until the host answers. You will hear a high-pitched tone.
5. Press the Term function key (F4). The Model 100 should go into the Terminal mode, with the following labels displayed on the bottom line of the screen:

   `Prev Down Up Full Bye`

   (If the Model 100 does not go into the Terminal mode, press (BREAK) until the `Telcom:` prompt reappears. Be sure the Model 100's ANS-ORIG and DIR-ACP mechanical switches are set correctly. Try the call again.)

6. Typing on the keyboard, complete the log-on dialogue that is required to begin communication with the host. If the information on your screen is garbled or otherwise unexpected, see Testing and Troubleshooting below.
7. You should now be at the menu or command level of the host.

Testing and troubleshooting.

Now you can begin using the host's facilities. Following the instructions provided by the information service, type in some commands that tell the host to send several pages of information to you: option menus, explanations of commands, wire-service news reports — anything to fill up your screen so you can test the communication link for quality and compatibility.

Most host systems work with a variety of personal computers, each of which may have a different display size and may respond differently to various control characters. Even after establishing a good link to the host, you may still have some difficulties until the host knows exactly what your Model 100's characteristics are. Here are some typical problems that might come up, with suggestions for their solution.

Meaningless characters or solid blocks like ■ are displayed on the screen. If the displayed material is completely unreadable, it usually indicates a mismatch in data-word length or parity. Terminate the call with (F8), check the communication switch settings, and try again. If much of the incoming material is readable and it corresponds to some degree with what you are typing, then the problem is with the telephone lines. Hang up by pressing (F8) and try the call again, hoping for a better connection next time.

Keys you type are not shown on your screen, but text originated from the host is displayed. Press (LABEL) to display the function key definitions. You'll probably see the word Full over [4], indicating that the Model 100 expects the host to use the *full-duplex* convention of repeating or "echoing" characters you type. Apparently, the host is using the half-duplex convention (not repeating the characters you type).

(In full-duplex communication, the host and terminal can send and receive simultaneously. In half-duplex communication, the two computers must take turns sending and receiving. By convention, the host echoes data sent over a full-duplex channel, but does not echo data sent over a half-duplex channel.)

In this case, you can make your Model 100 do its own "local echo" by pressing [F4]. Now the word `Half` should be displayed over the [4], and whatever you type will be displayed as it is sent to the host.

Keys you type are displayed in duplicate on your screen, but text originated from the host is displayed properly. This is the opposite situation from the one just described. Turn on the function-key labels and you'll probably see the word `Half` over [4], indicating that the Model 100 is doing a local echo of data you transmit. However, the host is also echoing your transmissions, creating the double-character effect.

To eliminate the problem, press [F4] so that `Full` is displayed on the bottom line of the screen.

Lines are too long for the display, so that words break in the middle and continue on the next line. The host doesn't know what screen size you have, and probably assumes a width of 64 or 80 characters per line. If you save the text in memory for later use (download), you can display it with proper line breaks at that time using TEXT. Or if the host lets you change your terminal characteristics, you'll be able to solve the problem immediately by specifying a line length of 40 characters.

Too many lines are sent before the host stops — making it difficult to read incoming text. The host doesn't realize your display holds only eight lines, and probably assumes 16 or 24 lines to a video "page." It transmits what it considers to be a full page, and only then asks you to "Press Enter for More."

There are several remedies for this problem. Start by turning the function-key labels off using [LABEL] to give yourself eight usable lines on the display. Your Model 100 stores the most previous eight lines in addition to what is stored on the display. To see the preceding lines, press the Prev function key, [F1]. Press any key to go back to the newest lines. This feature gives you a virtual 16-line display — much more respectable as a communication terminal display.

The best solution, however, is to tell the host to send only eight or 16 lines to a page, if the host has this option.

Lines are displayed on top of one another. This problem occurs when the host does not send a line feed after each carriage return. (The line feed advances the cursor to the next line, and the carriage return moves the cursor to the leftmost column of the display.) If the host sends carriage returns without line feeds, text is almost impossible to read, because each new line is superimposed on the previous one, except for those lines that exceed your display width — the Model 100 will break these onto a new line properly.

One less-than-ideal solution is to save the incoming text in a RAM file (download). Then, after ending the telephone call, examine the file using TEXT. Presto! The line endings are fixed. This is a hidden benefit of the Download function.

The best solution, by far, is to tell the host to send line feeds after carriage returns.

The host erases your display at the end of each page. This occurs when the host sends a form feed (control-L) at the end of each video page, which the Model 100 interprets to mean "clear the screen." If you find this undesirable, you may be able to tell the host to simulate form feeds with blank lines instead of using the form feed code.

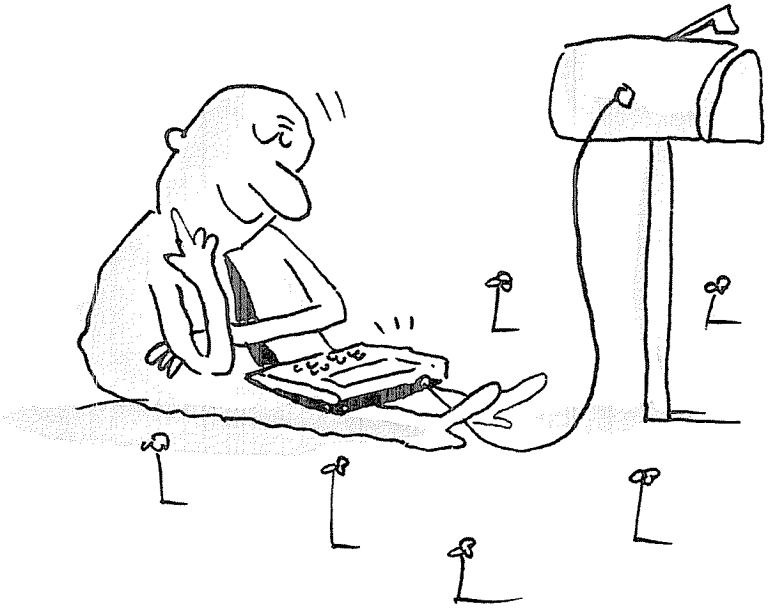
As you've probably gathered, the best answer to all of these types of problems is to inform the host of your terminal characteristics. Most information services have a special menu or command to accomplish this. Table 3.2 summarizes some key information about the Model 100. Refer to it when telling a host your terminal's characteristics.

Characteristic	**Recommended setting for the Model 100**
Terminal width	40 characters
Page size	8 lines (16 if you use the previous-screen recall function (F1))
Form feed handling	A form feed code clears the screen. If undesirable, request the host to simulate form feeds using blank lines
Tab handling	Tab stops are preset at columns 1,9,17,25,33 (no need to simulate them)
Carriage return and line feed handling	The Model 100 requires a line feed after each carriage return
Erase character	(BKSP) character moves the cursor back without erasing; (DEL) moves the cursor back and erases. Tell host to send decimal code 127 ("DEL" or "RUBOUT") to erase a character
XON/XOFF protocol	The Model 100 recognizes this protocol if it is enabled with the STAT command

Table 3.2. Terminal characteristics of the Model 100.

Signing Off. When ready to end a telecommunication session, type the recognized sign-off command. *After the host computer has gone off-line*, press (F8) to disconnect the telephone. If you disconnect without first logging off, you may incur some extra connect-time charges from the host. Furthermore, if you have transmitted any text to the host during the session, it may be lost unless you log off properly.

In the next chapter, we put the Model 100 to practical use as an electronic mail handler, sending and receiving text files.



4 / Electronic Mail

Electronic mail or E-mail is the transmission and reception of letters, memos, reports, and other documents using telephone connections to a central host. The host keeps a "letter box" for each person and informs the user when mail has arrived.

The Model 100 makes electronic mail operations smooth and efficient, enabling you to send documents and numeric results across the country with speed and accuracy. This chapter covers the four main E-mail functions:

- Typing in a letter while you are on-line ("interactive communication")
- Sending prepared text from a document file to a remote computer (uploading)
- Storing incoming data in a document file (downloading)
- Printing received documents

Interactive Communication

For sending brief letters or messages, it's simple and practical to type the letter while you are on-line with the host.

For sending a longer letter, you'll probably want to use the upload function described later in this chapter.

To send a letter via E-mail you first log-on to the host and reach the command or menu level of the system. Then you request the E-mail program. (With CompuServe, type `GO EMA` (ENTER). With the Source, type `MAIL` (ENTER).)

At this point the host requests three items of information:

1. The message itself
2. The address of the recipient, usually given as an ID number
3. A message summary

The recipient's address and the message summary are simple inputs. Providing the message itself can be a little more complicated.

Some E-mail services, such as CompuServe's, require that you store your letter in a file created with one of the system's built-in text editors. Other systems — especially computer bulletin-boards — simply prompt you for the text of the message, one line at a time. The file-type system requires that you learn the details of yet another text editor program, but also provides more flexibility: you can edit or copy parts of a file for use in another letter or report before telling the host to "deliver" it.

In just about every E-mail system, you must type in the document in lines, pressing (ENTER) at the end of each line. Often, hosts don't have an auto-wrap function. For this reason, you must be sure that the lines you type aren't too long for the host, otherwise the host may lose part of your message. The maximum line length varies with the host system, but 80 characters is a safe number to use. Since your Model 100 screen shows 40 columns per line, you may find it convenient to use 40 as a maximum, pressing (ENTER) before the cursor on your LCD advances to the next line.

When you finish typing the text of your letter, you must also indicate this to the host. Usually you type a special character sequence to "escape" from the text entry mode and return to the command mode. (With CompuServe, you type /EX (ENTER) on a line by itself. With the Source, you type .S (ENTER) on a line by itself.)

Depending on the system, your letter may be sent automatically or after entering explicit commands. In either case, the host usually provides the necessary prompting.

To end a telecommunication session, be sure to type the recognized log-off command to the host (usually OFF or BYE). After the host goes off-line, press (F8) to disconnect the telephone line. If you disconnect without first logging off, you may incur some extra connect-time charges from the host. Furthermore, if you have transmitted any text to the host during the session, it might be lost unless you log off properly.

A sample dialog for sending a letter using CompuServe appears on the following pages. Model 100 inputs are underlined, and explanatory remarks appear to the right.

```
Enter your selection
number, or H for more
information.
```

This prompt appears at the end of every menu.

```
!GO EMA (ENTER)
```

Go to the E-mail menu.

```
CompuServe          Page EMA-1
1    Read mail
2    Compose and send mail
Last menu page. Key digit
or M for previous menu.
!2 (ENTER)
```

You select the option for creating and sending mail.

```
CompuServe          Page EMA-4
CREATE a new message in
your temporary workspace
using:
     1  FILGE editor
     2  ICS editor
     3  File from disk space
EDIT message in workspace
using:
     4  FILGE editor
     5  ICS editor
     6  SEND message from
        workspace
     7  Information on
        FILGE
     8  Information on ICS
Last menu page. Key digit
or M for previous menu.
!1 (ENTER)
```

Before sending mail, you must create it, using one of the CompuServe text editors.

Select the FILGE editor for creating your mail.

```
New file Z99EMA.TMP
created - ready
May 30, 1983 (ENTER)
Frank Kelly: (ENTER)
I received a copy of your
memo (ENTER)
regarding new prospects in
Minneapolis.(ENTER)
Congratulations on the
good work! (ENTER)
Bob Fowler (ENTER)
/EX (ENTER)
```

The host is ready to accept the text of the letter now

This tells CompuServe you're ready to exit from text entry mode.

```
CompuServe          Page EMA-4
CREATE a new message in
your
temporary workspace using:
     1  FILGE editor
     2  ICS editor
     3  File from disk
        space

EDIT message in workspace
using:
     4  FILGE editor
     5  ICS editor
     6  SEND message from
        workspace
     7  Information on
        FILGE
     8  Information on ICS
Last menu page. Key digit
or M for previous menu.
!6 (ENTER)

CompuServe          Page EMA-8
Key <ENTER> to leave; ?
for help
Send to user ID
:70000,9999 (ENTER)
Subject (32 characters
max)
:Acknowledgement (ENTER)
Your name (32 characters
max)
:Bob Fowler (ENTER)
Is this correct? (Y or
N):Y (ENTER)
Message awaiting delivery.
Key <Enter> to continue
!OFF (ENTER)
```

The message has been created in your workspace now and is ready to be sent.

Send the message.

Frank's user ID

To help Frank when he's scanning through his mail

End of E-mail session. Logoff

Uploading

Uploading is sending a prepared file to a remote computer. The file may be a word processing document, a table of numeric data, or a previously downloaded file. While traveling, you may use uploading for a variety of reasons: to send electronic mail to one of the information services or computer bulletin boards; to send memos, reports, and letters back to your home office computer for further word-processing and printout; to submit completed sales order forms to an order processing department at the home office; and so forth.

The advantages of uploading over interactive communication (typing in the information while you are on-line with the remote computer) are:

- speed — the data is sent at the fastest possible rate, without the delay of typing
- accuracy — you prepare the document ahead of time and can check it carefully before sending
- ability to send computed results — in a matter of minutes you can send detailed results generated by BASIC programs (such as electronic spreadsheets).

The first step in uploading is to create a document off-line using the Model 100's TEXT word processing program. The text must be saved in a RAM file; the TELCOM Upload function will not load and transmit a file from cassette. (To upload cassette files, use the application program called TERM.BA, described in Part 2.)

Type in the letter as usual, letting TEXT auto-wrap the lines except for blank or short lines and paragraph endings — press (ENTER) to get these results.

Include all the information you would normally put in the letter — date, greeting, body, and your name.

Now call up the host and request the E-mail menu, exactly as described in the previous section. Proceed up to the point where the host asks for the body of the message (`New file Z99EMA.TMP - ready` in the sample CompuServe dialog).

Now press the Upload function key, (F3). The Model 100 will ask you: `File to Upload?`. Type in the name you assigned to the Model 100 RAM file.

The Model 100 will ask you: `Width:`. Enter the maximum line width allowed by the host; this may be as many as 132 characters, but 80 is a safe number to use. The Model 100 will insert line endings in your document to ensure that none of the lines exceed the specified width. This

"auto wrap" function breaks lines at word boundaries just as in the TEXT program.

If your document includes any formatted lines, that is, lines that must not be broken by the auto-wrap function, be sure the specified width is great enough to accommodate these.

For example, suppose your document contains these lines:

```
1. Four bags of adhesive paste
2. Ten precut box forms
3. Two rolls strapping tape
```

If you specify a width of 25, the text is transmitted as:

```
1. Four bags of adhesive
paste
2. Ten precut box forms
3. Two rolls strapping
tape
```

In this example, since the longest formatted line is 30, the width should be specified as 30 or greater.

If your document is *entirely* pre-formatted, with forced line endings exactly where you want them, you may choose to send it as-is. This might be true of a table of numbers, for example. In such cases, simply press (ENTER) in response to the `Width:` question. The Model 100 will not insert any line endings in the text. Do this only if you are sure that your lines aren't too wide for the host.

As soon as you have specified the width, the Model 100 will begin transmitting the file. During uploading, the function-key label `Up` will be highlighted on the display, and the text will appear just as if you were typing it — only much faster than that!

After file transmission is complete, the label `Up` will no longer be highlighted, and you will again be in the Terminal mode. You will now need to tell the host you have finished text entry. At this point, the procedure is the same as that described in the previous section. For example, with CompuServe, you type `/EX` (ENTER) to close the file (refer to the sample dialog).

Control characters. Any control character in the document to be uploaded will be transmitted as a dummy sequence — just as it appears on the display when you view the document with TEXT (`^G`, `^C`, and so forth). The only exceptions to this are tabs and carriage returns, which are sent as actual control codes.

Carriage return-line feed sequences are sent as single carriage returns. For this reason, the lines may be overlaid on your screen as they are sent.

The recipient of the letter will not be bothered by this, however, since the host software (or his own software) will be able to re-insert line feeds after carriage returns.

If you want to include control codes in a file to be uploaded, try the application program, TERM.BA, described in Part 2.

Logging-off. Be sure to log-off from the host before pressing (F8) to disconnect the telephone line. Otherwise, the letter you uploaded might be erased from the host.

Downloading

Ordinarily, only the the last 16 lines of received text are accessible; eight on the display and eight in the "previous page" buffer.

It is often convenient to store incoming data in a RAM file; this process is known as downloading. If the information you receive is a report or another kind of word processing document, you can edit it later using TEXT. If the downloaded file consists of numeric data, you can process it later with a BASIC program. Last, you can retransmit the same data with or without modifications using the Model 100's upload function; this is called a *store-and-forward operation* (very popular in portable computing circles).

To begin downloading while you are on-line in the terminal mode, press (F2). The Model 100 will ask, `File to Download?`. Enter the name of the file where you want the document stored. *If you name an existing file, its previous contents will be lost.* If the function-key labels are on, the word `Down` will be highlighted on the display.

From this point on, everything received will also be saved in RAM. To stop downloading, press (F2) again.

(The TELCOM file downloads to RAM files only; to download to a cassette file, try the application program, TERM.BA, described in Part 2.)

Often you'll want to download for a while, then turn downloading off while unimportant information, such as a menu, is displayed. The (F2) key lets you turn the function on and off quickly. Just remember to use a new file name each time you begin downloading, so as not to erase previously stored data.

Suppose you realize that you want to save a page that is already being transmitted. Many hosts have a retransmit-page command that can help you in this situation. Just wait until the current page-transmission is complete and the end-of-page prompt appears. Press the (F2) to start downloading, name the file, and type the retransmit-page command for the host (with CompuServe, type `R` (ENTER) at the end-of-page prompt).

If you cannot make the host retransmit a page, you can still begin downloading "on the fly": while the information is being received, press (F2). The Model 100 will stop displaying the incoming data and will ask you to name a file. Incoming text will continue to be saved in a temporary storage buffer. As soon as you name the file, all the information received since you pressed (F2) will be put into the download file.

The Model 100 might use up all available memory during a download operation. If this ever happens, the download file is closed automatically and the computer returns to the Terminal mode. The downloaded file retains the text that was received before memory filled up. The host, unaware that the Model 100 memory is full, will continue to send text. In this case, you may need to send some kind of interrupt character to the host, such as (CTRL)(C) for CompuServe or (CTRL)(P) for the Source.

To download a file larger than your Model 100's memory capacity, use the application program, TERM.BA, described in Part 2. The program lets you download directly to a cassette file.

Logging-off. Be sure to log off from the host before pressing (F8) to disconnect the telephone line. Otherwise you might incur some extra connect charges and any text you sent may be erased from the host.

A sample dialogue for uploading a letter using CompuServe appears on the following pages. Model 100 inputs are underlined, and explanatory remarks appear to the right.

```
You have EMAIL waiting.
```

After you log-in, the host tells you that mail is waiting.

```
CompuServe                Page CIS-1
CompuServe Information Service
1 Home Services
2 Business & Financial
3 Personal Computing
4 Services for Professionals
5 User Information
6 Index
Enter your selection number,
or H for more information.
!GO EMA (ENTER)
```

Go directly to the E-mail menu.

```
CompuServe                Page EMA-1
Electronic Mail Main Menu
1  Read mail
2  Compose and send mail
Last menu page, Key digit
or M for previous menu.
!1 (ENTER)
```

Select the read-mail option.

```
CompuServe                Page EMA-3
1 Bob Fowler/Acknowledgement

2 Larry Thompson/Progress
Report
Last menu page, Key
digit or M for previous menu.

11 (ENTER)
```

Two letters are waiting

Press (F2) and name a download file now before selecting an option.

```
CompuServe                Page EMA-5
3-Jun-83 11:29 Fr [70000,9999]
May 30, 1983
Frank Kelly
I received a copy of your memo
regarding new prospects in
Minneapolis. Congratulations
on the good work!.
Bob Fowler
Key <ENTER> to continue
10FF (ENTER)
Delete this message? (Y or
N):Y (ENTER)
off at 15:09 EDT  28-Jul-83
Connect time =    0:03
```

Here's the first letter.

Press (F2) to stop downloading here.

You don't want to read the second one just now. Hang up.

Host asks what to do with the message you read. Delete it.

Working with a downloaded file. To examine a downloaded file, press (F8) in response to the `Telcom:` prompt. This returns you to the main menu. You'll see the downloaded file listed as a document. Select it with the cursor and press (ENTER). You can now edit the file just as if you had typed it in!

You might notice a few anomalies. For example, if any parity errors occurred during the transmission, they will appear as solid blocks like this: ■. If the host transmitted any unusual control codes, they will be indicated by ^ followed by a letter or other symbol, for example, ^P. You'll probably want to delete these kinds of extraneous characters from the file.

The downloaded document might already be formatted for printing, that is, it might have a forced line ending at the end of every line. This thwarts the Model 100's auto-wrap capability and uses up extra memory. Furthermore, these "logical" lines probably do not correspond to your display's physical lines, making it difficult to read the text. Delete any unnecessary forced line endings, leaving only those that mark blank lines or paragraph endings.

Printing while On-line

Printing while on-line is useful for electronic mail when you need a hard copy and don't need to do further word processing on the received messages. You might also want to print on-line when the document you are receiving is too large to fit in memory.

One disadvantage of printing while on-line is that it tends to slow down communication, increasing your connection time and cost. Remember, you can always print a downloaded document after you go off-line, using TEXT's Print function.

To start printing while on-line, press (F5). If the function-key labels are turned on, you'll see `Echo` added to the bottom line. The Model 100 immediately begins outputting the received data to the printer. If your printer's average speed is slower than the rate of communication with the host, the Model 100 will use the XON/XOFF protocol to pause communication periodically and let the printer catch up. For this reason, the printer Echo function usually requires that you have the XON/XOFF protocol enabled.

To stop printing, press (F5) again; the `Echo` message will be erased and printing will stop.

If your printer hangs up for any reason (out of paper, off-line, disconnected, turned off, etc.) while the Echo function is on the Model 100 will pause until the problem is corrected. At such times, you may want to cancel the Echo function. Press (BREAK) until the Model 100 beeps and the `Echo` message disappears. Communication can then continue.

Nicolae Asaiy

5 / Plugging into Databases

Far more than an electronic mail machine that sends and receives letters, the Model 100 can give you access to databases filled with useful information on almost any subject imaginable, from stocks to sports. Furthermore, you can perform a variety of computations on the received data.

In this chapter, we discuss several kinds of databases you may find useful, and explain three ways your Model 100 can use received data:

- Off-line storage of frequently needed information for instant retrieval, reducing your need to call up the information service
- Sorting and rearranging received data to suit your needs
- Computations on numeric data

Some databases include BASIC programs written by other Model 100 users; this is an excellent way to get interesting software for free (except for the cost of the connect time to a host computer system).

For each application, we give a practical example using the CompuServe Information Service. If you use the Source, a computer bulletin board, or any other service, you can adapt the procedures to that system.

Types of Data Available

It would require a good-sized book just to catalog all the types of data you can retrieve from the many databases now online. We'll look at a few examples that may be of most interest to you.

In the area of business and finance, you can retrieve up-to-the minute (or quarter-hour) trading reports or historical market information on stocks, bonds, mutual funds, government issues, and options. Rather than calling your broker, you simply call up an information service and retrieve the figures you need. You can view the data at your leisure and use your computer to compute statistics, evaluate a portfolio, or graphically chart a stock's progress.

When traveling or planning a trip, you can obtain the latest information about plane schedules, or even make your own reservations. Again, with your Model 100's computing power, you can re-sort a table of scheduled flights, for example, according to any criterion you choose — carrier, airplane type, number of stops, flying time, and so on.

Many information services provide detailed sports coverage, especially in the form of box scores. If you like to keep track of all the numbers, using an information service is an excellent way to get the data into your computer for further processing.

Whatever your profession, there's probably a specialized database that can be extremely useful to you. *The Directory of On-line Databases* (Cuadra Associates, Inc., 2001 Wilshire Blvd., Suite 305, Santa Monica, CA 90403) lists over 1600 databases, ranging from chemical engineering data to oil-well reports.

Getting the Data

It takes a little planning to get the data you need in the most convenient format for use after you go off-line.

First preview the data you want to download. Learn the commands that let you "interrogate" the database. Use TELCOM's Echo function, (F5), to get a printout of the various menus you'll encounter. That way you can locate the needed commands on paper, saving the time and expense of having them transmitted repeatedly.

The host often sends lengthy menus after each block of data, which is undesirable, because you don't want to store the menus. The host might offer an "expert user" or "brief" mode in which the menus are eliminated or abbreviated. Learn to use this mode; it will reduce your connect time and keep superfluous text out of the downloaded file.

Find out if you can cause the host to send you the data continuously, without stopping after each video "page" or after each individual record. This, too, minimizes the extraneous information that goes into your downloaded file.

Once you know how to get the most data out of the host with the least interaction, you're ready to download. While the host is on-line and waiting, press (F2), specify a distinctive file name, and then give the host the command to retrieve the information you want.

After you have the information you need, press (F2) to stop downloading, sign off from the host, and terminate the call by pressing (F8).

Now you can examine the downloaded file using TEXT. Eliminate any extraneous messages that were included.

If the data is a simple text file, your job is done.

However, if it is a table or a series of formatted records, you will probably need to work on it a little bit more to allow sorting or computation.

Examine the file closely, to be sure that you understand the format. In a table, each column entry appears at a fixed column position. In a formatted record, by contrast, there are no fixed column positions, but the items are separated by one or more spaces, commas, or carriage returns.

In summary, here are some hints and tips for getting data from a remote database:

- Learn the commands you'll need for efficient use of the database.
- Plan your command sequence carefully while off-line.
- Use the expert or brief mode.
- Set the host to send a stream of data uninterrupted by end-of-page prompts.
- Start downloading just before you enter the command that request the data. Stop downloading as soon as you've received the data you need, then log off.
- Use TEXT to edit extraneous material from the file.
- If the file is formatted, note the format and be sure that it is consistent.

Text Storage and Retrieval

For text retrieval, you simply download the information into one or more Model 100 RAM files, as explained previously.

Suppose you are interested in wire-service stories on a certain topic for a report you're writing. Using the information service's scanning facilities, locate the relevant stories, make a note of their reference numbers, and begin to read them.

Whenever you come to a section you want to quote in your report, turn on the Model 100's download function by pressing (F2) and specifying a distinctive name for the download file. Turn off the download function by pressing (F2) again when you reach the end of the material you want to quote.

Later, using TEXT to write your report, you can copy the quote into your report with just a few keystrokes. (Be sure to respect any copyright restrictions that apply to the text you download.)

For a detailed example, let's suppose you are planning a trip to Maryland and want to bring along your Model 100 to use the CompuServe Information Service. What are the telephone access numbers for use in that state?

Before leaving your office, you log-on to CompuServe and request the telephone access numbers for area codes in Maryland. ("Telephone Access Numbers" is one of the options listed under the User Information menu.)

The dialogue for requesting this information from CompuServe appears on the following pages. (The information is for example only and may not be accurate when you read this.)

```
! GO PCS-177 (ENTER)

CompuServe                Page CIS-177
1  Instructions & Helpful Info.
2  Ph. # Changes/Future Info.
3  300 Baud - Ph. # Search
4  1200 Baud - Ph. # Search
5  300 Baud - List All Ph. #s
6  1200 Baud - List All Ph. #s
7  2-Ltr Province Codes: Canada
8  2-Ltr State Codes: U.S.
0  Exit Phone Program
Enter option number: 3 (ENTER)

Enter AREA CODE, two-letter
STATE abbrev. or two letter
PROVINCE abbrev. MD (ENTER)

PLEASE NOTE: There are
surcharges associated with the
use of all network access
telephone numbers with the
letters T, G or D following the
number. See main menu item 5,
User Information, to find the
current surcharge rates.
Annapolis        MD 301 224-8550G
Baltimore        MD 301 254-7113C
Baltimore        MD 301 547-8100T
Baltimore        MD 301 962-5010G
Bethesda         MD 202 429-7896G
Key <ENTER> for next page  !
(ENTER)

Dundalk          MD 301 962-5010G
Frederick        MD 301 293-1072T
Hagerstown       MD 301 293-1072T
Rockville        MD 301 770-1680T
Rockville        MD 202 429-7896G
Silver Springs MD 202 429-7896G
```

Begin downloading by pressing (F2) before pressing (ENTER) on this line.

```
Towson              MD 301 962-5010G
Last page Key M for menu !
OFF (ENTER)
```

Stop downloading by pressing (F2) before pressing (ENTER).

After downloading, you can go off-line and edit the file, deleting everything except for the lines that contain telephone access numbers.

Since you'll be calling these numbers with the Model 100, it's convenient to copy this information into the ADRS.DO file so that TELCOM's Find Command (F1) can search through it. Before copying the text to ADRS.DO, put colons around the telephone access numbers, move the network identifier letters, and delete extra spaces. After editing, the first entry should appear like this:

```
Annapolis MD(G): 301 224-8550< >:◀
```

Once you have formatted the file, add it to ADRS.DO with this key sequence:

1. Move to the top of file: (CTRL)(↑)
2. Select entire file: (F7) (CTRL)(↓)
3. Delete but save in the paste buffer: (F6)
4. Close the file: (F8)
5. Open the ADRS.DO file: ADRS.DO (ENTER)
6. Move to the bottom of the file: (CTRL)(↓)
7. Copy the data from the paste buffer: (PASTE)
8. Empty the paste buffer: (F7) (F5)
9. Close the file: (F8)
10. Start BASIC: BASIC (ENTER)
11. Erase the original file KILL "file" (ENTER)

Now that you have added these numbers to the ADRS.DO file, you'll be able to dial them using TELCOM.

Sorting

Much of the data you'll receive will be structured in a table or some other consistent format. Tables 5.1 and 5.2 give examples of tabular and fixed-format data, respectively. Either kind of data is valuable because it can be sorted or used in computations.

```
Williams, J.     Food          (214)555-1202
Jefferson,R.     Petroleum     (214)555-1223
Alcorn,L.        Plastics      (214)555-3000
Davis,B.         Medicine      (303)123-4500
Peabody,A.       Food          (303)123-5100
Lennox,R.        Agriculture   (714)555-0001
```

Table 5.1. Tabular data.

```
Williams,J.,Food,(214)555-1202
Jefferson,R.,Petroleum,(214)555-1223
Alcorn,L.,Plastics,(214)555-3000
Davis,B.,Medicine,(303)123-4500
Peabody,A.,Food,(303)123-5100
Lennox,R.,Agriculture,(714)555-0001
```

Table 5.2. Consistently formatted data.

After downloading a file, you may wish to sort it to suit your needs. First use TEXT to put the information into a table or other consistent format. Then use a sort program such as SORT.BA, described in Part 2.

For a detailed example, let's take a list of Model 100 users obtained from the Compuserve Information Service. An abbreviated sample dialogue to get the information appears on the following pages. Model 100 inputs are underlined, and comments are given in the margin.

```
!GO PCS-8 (ENTER)

CompuServe                Page PCS-8
1 Find user info
2 Enter your info
3 Change your info
4 Delete your info
0 Exit DIRECTORY
Last menu page. Key digit
or M for previous menu.:
1 (ENTER)
```

```
Find by:
1 User ID
2 Name
3 City
4 Equipment type
5 Interests
0 Exit DIRECTORY
Last menu page, Key digit
or M for previous menu.:
4 (ENTER)
Key <ENTER> at the find
prompt to end search.
Find what type of equipment
:MODEL 100 (ENTER)
70 Users found
98700,1057 RICH
YONKERS, NY
RADIO SHACK          MODEL 100
GAMES
90903,151 ROBERT
PITTSFORD, NY
RADIO SHACK          MODEL 100
BUSINESS
62741,1312 CHRISTOPHER
CHARLOTTE, NC
RADIO SHACK          MODEL 100
PROGRAMMING
Key <ENTER> for next page.:
(ENTER)
```

Start downloading by pressing (F2) before pressing (ENTER)

Continue until the listing is complete. Then stop by pressing (F2). Log-off.

After logging off, terminate the call by pressing (F8) and edit the downloaded file using TEXT. First, delete all the extraneous lines. Then, since all of the users have Radio Shack Model 100s, you can delete the third line of each entry. That leaves you with a series of records like this:

```
98700,1057 RICH
YONKERS, NY
GAMES
```

The list follows no apparent order. Suppose you'd like to order it by interests, so that, for example, all the business users are listed together.

The first step is to take a close look at the data to determine its structure. Each record consists of five "fields":

[ID] [name]
[city] [state]
[interests]

Except for the two-letter state abbreviation, the fields are of variable length. Fields 1 and 3 are terminated by a space. Fields 2, 4, and 5 are terminated by a carriage return.

If the data had been arranged in a table as in table 5.1, we would need to note the starting and ending position of each column, rather than the character used to mark the end of each field.

Knowing the format of each record, you can now sort the data using any of the fields as a key. Of course, you'll need a general-purpose sort program as well, such as the SORT.BA, described in Part 2.

Computation

Doing computations is similar to sorting: the data must be uniformly structured and you must know the structure. Furthermore, you must know something about the kinds of numbers that are contained in the table — magnitude, range, format, and so forth. For the actual computations, you need a program such as TCALC.BA, described in Part 2.

Table 5.3 shows some financial data obtained from Compuserve's Microquote database.

```
TANDYCORP

87538210  TAN        Exch: N
  Date          Volume
  High/Ask      Low/Bid     Close/Avg
--------------------------------
16-May-83         198,600
  61 3/4        60 1/2        61 3/4
17-May-83         359,100
  63 1/4        61            63 1/8
18-May-83         307,600
  63 3/4        61 5/8        61 7/8
19-May-83         237,900
  62 5/8        60 5/8        60 7/8
20-May-83         343,000
  60 1/2        59 1/2        60
```

Table 5.3. Stock trading records.

The table shows a five-day trading history for Tandy Corporation stock. A number of calculations could be performed using these figures. Suppose, for example, you wanted to know the average spread between the stock's high and low trading price and the average number of shares traded daily over the time period. Although the calculations are simple, there are several minor complications related to the format of the table and of the numbers themselves.

We'll look at these problems and highlight a few general principles about computations on downloaded information.

Notice that the table contains four fields of information for each trading day: Date, Volume, High/Ask, Low/Bid, and Close/Avg. The fields won't fit on a single line. The table would be much easier to work with if it were presented on a wider display or printout. Then the lines would appear like this:

```
  Date       Volume   High/Ask   Low/Bid   Close/Avg
-----------------------------------------------------
16-May-83   198,600   61 3/4     60 1/2      61 3/4
```

This would facilitate reading and calculations on the information.

Whenever possible, set the host to download the information as a simple table, with one line per record. If that is not possible, you may want to reformat the data manually so that it makes such a table.

The format of the numbers themselves also presents a few problems. The volume figures, for example, include commas to separate thousands from hundreds. This is fine for your reading, but most computers like their numbers without commas. Furthermore, the high and low figures are shown as proper fractions, but for computer use, they should be given in decimal form, for example, 63.875 instead of 63 7/8.

In summary, try to get the data in simple tabular format, with all numbers in decimal form without commas. Many financial information services offer this kind of formatting as a special option for use by downloaders.

Downloading BASIC Programs

Downloading BASIC programs is a good way to learn more about programming. It's also one of the cheapest sources of programs though there's no guarantee of quality. Unless you are already experienced in BASIC programming, stick with programs written specifically for the Model 100.

In this section, we go through a typical program downloading session, and outline the steps to getting the downloaded program ready to run. We show how to format the program and how to recognize extraneous characters. However, we don't go into program debugging, a subject that is beyond the scope of this book.

In terms of Model 100 commands and procedures, downloading a BASIC program is just like downloading any other text. However, the information service often has a different set of commands for this function.

For example, on Compuserve you'll find a Model 100 interest group (SIG) allowing users to share questions, comments, news, and favorite programs. Here's an abbreviated sample dialogue for downloading a BASIC program from the Compuserve Model 100 SIG. Model 100 inputs are underlined, and comments appear in the margin.

```
!GO PCS-154 (ENTER)

CompuServe              Page PCS-154
Welcome to Model 100 SIG
Function: XA (ENTER) ———————————— Select SIG/Access.
```

```
Database for which Section:
0 1 2 3 5   0 (ENTER)

SIG Database Access System
Use ? for help
SIG/Access: CAT /DES
/KEY DEMO (ENTER)

------------
[60000,111]
DEMO.100
        Accesses: 5 1-Jun-83
------------
```

0 is the general-interest section.

Request a description of programs labeled as "DEMO."

Program author's user ID.

Program name.

Number of accesses and date of last access.

(Other entries omitted)

```
SIG/Access: DOW DEMO.100
[60000,111] (ENTER)

^O^[ICapture Buffer Transfer


No error detection/
correction
Opening capture buffer...
^R100 PRINT "DEMO PROGRAM"
110 PRINT "THIS PROGRAM LINE
IS LONGER THAN 40 CHARACTERS
AND SO IT IS SPLIT"
120 FOR I=32 TO 127:PRINT
CHR$(I);:NEXT
130 PRINT "END OF DEMO"
^TCapture buffer closed.
SIG/Access: EXI (ENTER)

Function: OFF (ENTER)
```

Press (F2) to start downloading before entering this command, which tells the host to transmit the program.

Press (F2) to stop downloading before pressing (ENTER)

After completing such a session, you have a downloaded file containing the program. Before running the program, you must edit it to delete any extraneous characters.

If you had completed the dialogue as shown, you would have a file starting with:

```
^O^[ICapture Buffer Transfer◀
```

and ending with:

```
SIG/Access: EXI◀
```

Start by deleting all the text that precedes and follows the actual program lines. Delete control characters such as ^R and ^T, which are part of a download protocol that Compuserve uses but TELCOM does not.

Figures 5.1 and 5.2 show the BASIC program as it appears on the display before and after editing.

Notice that line 110 requires two lines on the display. Often the host will insert a carriage return to break up such a long line. When you edit the file, check for these extra carriage returns and delete them. The general rule for BASIC programs is: Every line must begin with a line number and end with a carriage return. A program line must not be broken in half by a carriage return.

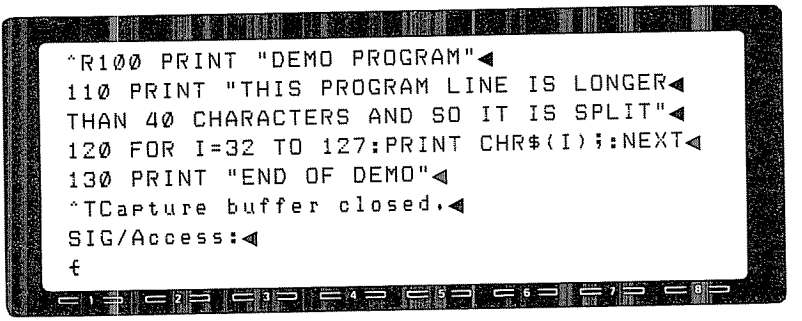


Figure 5.1. Downloaded BASIC program before editing.

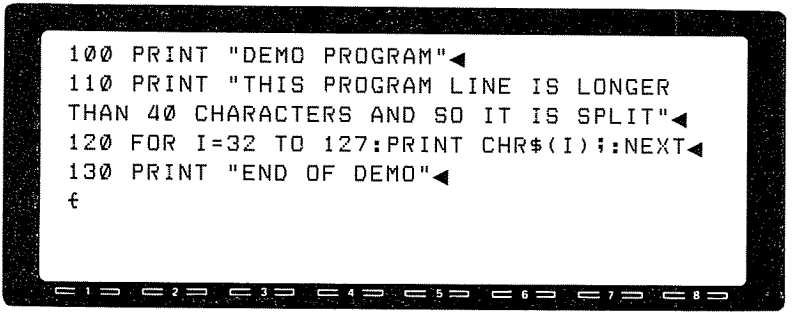


Figure 5.2. Downloaded BASIC program after editing.

After editing the text until it resembles figure 5.2, close the file. The program will appear as a document on the main menu. To run the program, follow these steps:

1. Select BASIC. When the OK prompt appears, type LOAD "file.DO" (ENTER), where *file* is the name you assigned to the downloaded program.
2. Type RUN (ENTER). To run the program. Assuming the program was received without transmission errors, it should perform perfectly. If not, dig out a BASIC programming manual.
3. After verifying that the program works, you will probably want to save it in program format (.BA) rather than in a document format (.DO). Program format uses much less memory for storage. Simply type SAVE "file" (ENTER), omitting the .DO suffix.
4. .Delete the .DO version by typing KILL "file.DO" (ENTER).
5. Press (F8) to return to the main menu, and you will see the program listed as file.BA.
6. To run a .BA file, select the file from the main menu and press (ENTER).

6 / Creating Your Own Database

In the last chapter you learned ways to use databases offered by commercial information utilities. But the Model 100 is an ideal tool for creating your own databases as well.

By now, you may have discovered that on-line data can be expensive. Yet, much of that information is available in printed form. Spending some time typing in printed data can pay off handsomely with tables and graphs based on the information.

You can use the Model 100 as an electronic entry form combining the portability of a notepad with the intelligence of a computer. A warehouse clerk, for instance, can use the Model 100 to do a physical inventory check. Or suppose you want to catalog a book or record collection. Rather than writing the information on paper and retyping it later, you can enter it directly into the Model 100.

Do you ever have to prepare reports based on research notes? The Model 100 can serve the function of a set of index cards, allowing you to enter notes in any order and to retrieve them later in a more organized sequence.

Last, you can use the Model 100 for basic data-entry in which you store a stream of numbers to be processed later on. A lab technician might use the Model 100 to enter temperature measurements. A realtor might use it to record the dimensions and other features of a new property for sale.

In this chapter we explain the basic principles you need to plan, create, and use your own databases. Whether you plan to upload the data to another computer or to handle it with your own data management program, having this background information will make the process go much smoother.

You can use the Model 100's built-in progams to accomplish many database operations. TEXT lets you enter data, update it, and store it in RAM or cassette files. SCHEDL performs search-and-retrieval functions. TELCOM uploads your database files to another computer. We go over some practical methods for creating and using a database with TEXT and SCHEDL. Information for uploading data is contained in chapters 4 and 5.

Additionally, you'll find several database-related programs described in Part 2. After reading the instructions, you can load and run the programs from the Software Cassette included with this book.

What Is a Database?

Before getting into specific database uses for the Model 100, let's take a quick look at a few concepts and terms.

By a *database*, we mean organized information that is stored in a computer. A mailing list, a set of note cards, or a table of numbers, can be stored as a database. Table 6.1 shows samples from each type of database.

Databases are made up of *records*, which are collections of related information. Every record in a database conforms to the same structure (some databases allow for multiple record types, however). Consistent structure enables the computer to perform powerful database functions

Mailing list

NAME: Robert Cotter
ADDRESS: 3000 Waverly Way
CITY: New York
STATE: NY
ZIP: 10001

Research notes

REFERENCE NO.: 3.71
KEYWORDS: Finance, Clinch River
SOURCE: Wall Street Journal, 6/28/83, page 4
SUBJECT: A task force of utilities and Wall Street investment firms devised a new plan to help pay for the controversial Clinch River breeder-reactor project.

Sales figures

	Sales by Region (000's)			
Region	Jan-Mar	Apr-Jun	Jul-Sep	Oct-Dec
I	77.1	44.3	55.1	97.6
II	33.7	30.6	34.8	45.5
III	81.0	65.5	50.5	65.2
IV	60.0	59.0	50.0	60.3

Table 6.1. Sample records from three kinds of databases.

such as sorting, searching, updating, report generation, and so on — tasks that are difficult or impossible to do with an ordinary word-processing file.

The related pieces of information that make up a record are called *fields*. The mailing address record in table 6.1 consists of five fields: name, street, city, state, zip code. The table of sales figures consists of 16 fields: one at the intersection of each row and column.

The *data type* of each field is usually predetermined. Alphanumeric and numeric are the two most common types. Alphanumeric fields store textual information (names, addresses, and descriptions) as well as numeric sequences (zip codes, part numbers, and telephone numbers). Numeric fields store information that will be used in arithmetic operations, such as the sales numbers of table 6.1. Some databases have special data types for dates, times, and other kinds of items.

The *length of a field* can be fixed or variable. In a typical mailing list, for example, 25 characters are allocated for the name, 15 for the city, two for the state abbreviation, and five for the zip code. When a data item exceeds the fixed length, extra characters are discarded; when a data item is too short, blanks are added at the end to fill the field. With variable-length fields, special characters separate or *delimit* the fields. Carriage returns and commas are commonly used.

Database Operations

The four primary database operations are data entry, updating, sorting, and report-printing. We'll look at each of these in turn.

Data entry. This simply refers to typing in data records. The order in which you type the records is usually unimportant. Ideally, a data-entry program prompts you to enter each field, and ensures that your entries match the data type of the corresponding field. However, you can also use a word processing program such as TEXT for data entry. In this case, it's up to you to ensure that the fields are entered in the correct sequence and format.

Updating. This operation involves changing records that have already been entered. Updating may also include inserting new records into the middle of a database or deleting records.

The procedure used for updating depends greatly on the type of file access. With true direct access, you can retrieve or modify any record in the database by reference to a record number (without having to read through preceding records). With sequential access, you must read through the database from the beginning up to the record you want. Up-

dating with sequential access involves reading from an input file and writing the updated records to an output file.

Model 100 RAM and cassette files permit only sequential access, so BASIC database programs typically input from one file and output to another. TEXT gives you pseudo-direct access because it lets you make changes directly in the original file.

One useful tool in updating is the ability to search for a record that contains a specified value in one of its fields. For example, suppose you want to change an address. It's helpful if the program can find the record automatically, given the person's name, rather than your having to page through the records until you find it.

Sorting. This operation lets you put database records in a sequence that suits your interest at that moment. At different times, you might want to see a database sorted according to different criteria. For example, you can sort a mailing list by zip code before printing mailing labels, or by name before printing a membership list.

The criterion used for sorting depends on the type of data in the sort field. Alphanumeric information is sorted according to the ASCII character sequence, an extension of the common alphabetic sequence. Dates are sorted in chronological order. Numbers are sorted in order of magnitude. If the sorting criterion doesn't match the data type, the results will not be useful. Table 6.2 shows what happens when the same data is sorted using various criteria.

Generating reports. There are several occasions for generating reports from a database. The most obvious is to get a printed copy of the data. A copy of the screen format might be sufficient; to do this with the

Data Type	Sorting Sequence	
	ASCII	**Chronology**
Dates	03/12/82	06/01/80
	06/01/80	03/12/82
	08/01/83	08/01/83
	ASCII	**Magnitude**
Numbers	100.81	3.88
	3.88	44.08
	44.08	100.81

Table 6.2. Data sorted according to different criteria.

Model 100, you simply press (PRINT) while the data is on the screen. But more often, you want the data in a format that takes advantage of the printer paper dimensions.

Report generator programs often allow you to select which fields to print, rearrange fields, and add labels and column headings. Some programs can even add column totals or other computations to the report. A monthly bank statement is an example of a report to which column totals are added.

Reports also provide a way of preparing your database for uploading to another computer. Instead of outputting the formatted report to a printer, you can easily send it to a RAM file or even to a modem if you've already established a telephone hookup to the other computer.

Suppose your database uses variable-length fields separated by commas, and you want to upload the data to another computer that requires fixed-length fields. A versatile report program will take care of this conversion for you.

Now let's look at ways to use the Model 100's built-in software for simple database management.

Using TEXT for Data Entry

The first step is to define the format of the data you want to store, down to the level of each field within a record. Are the fields fixed-length or variable-length? If variable-length, what are the delimiters?

If you plan to upload the data to another computer for use by a database management program, find out what format the other program needs. For example, you can separate each field with a comma and end each record with a carriage return. Also, consider the maximum record length; it could be as small as 80 to 132 characters because of the size of the host's input buffer (comparable to an in-basket).

When using TEXT for data entry, the most convenient format is variable-length fields separated by commas, with carriage returns at the end of each record. Variable-length fields let you enter data without worrying about column positions. Using a carriage return at the end of each record also makes the data compatible with SCHEDL. Figure 6.1 shows a few records in this format.

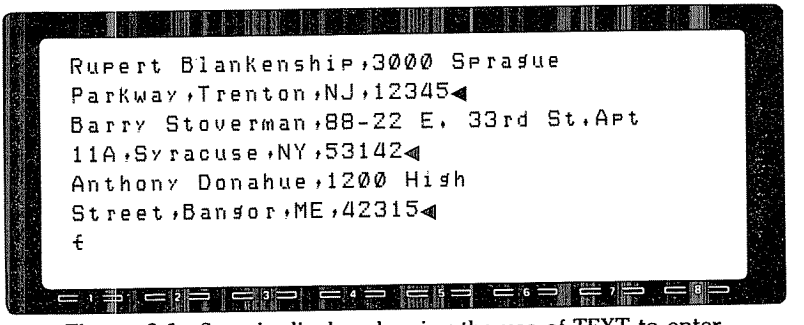


Figure 6.1. Sample display showing the use of TEXT to enter variable-length fields.

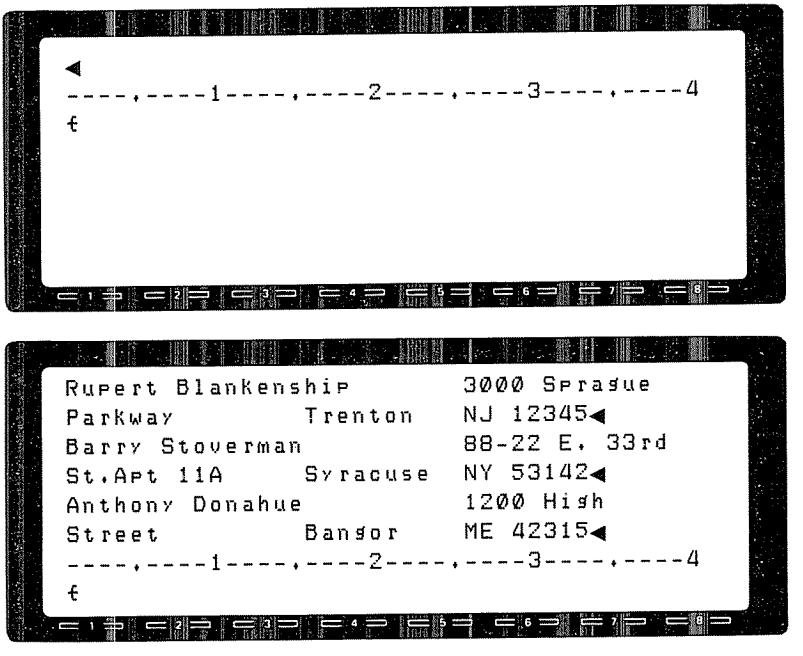


Figure 6.2. To enter fixed-length items, start with a column guideline as shown in the upper display. Note the carriage return (◀) above the column guide; put the cursor on top of this carriage return and start typing in the data, as shown in the lower display.

If you must enter fixed-length fields, you may find it convenient to put a moving "column guideline" in your text. Figure 6.2 illustrates this handy trick. As you type in the data, the guideline stays one line below the line you're typing, making it easy to find the column positions you need for fixed-length records.

Using SCHEDL for Data Retrieval

To retrieve data using SCHEDL, you must store the data in the file called NOTE.DO. Either type the data directly into that file with TEXT, or if the data is already in another file, copy it into NOTE.DO with TEXT.

To use SCHEDL effectively, you must end each record with a carriage return. Records can fill more than one line on the display because of the word-wrap facility. Fields can be fixed or variable length, but variable-length fields are easier to enter and usually require less memory for storage.

SCHEDL has two data-retrieval commands: Find, (F1), and Lfnd, (F5). Both commands prompt you to enter a search string and then retrieve all records containing that text *in any field.* The Find command displays the matching records on the LCD six lines at a time, and the Lfnd command sends all the data to the printer.

Suppose your NOTE.DO file includes many records like those in figure 6.1, and you want a list of all the people living in New York.

Select SCHEDL and type (F1) NY (ENTER) after the `Schd:` prompt. The program then displays two records as shown in figure 6.3.

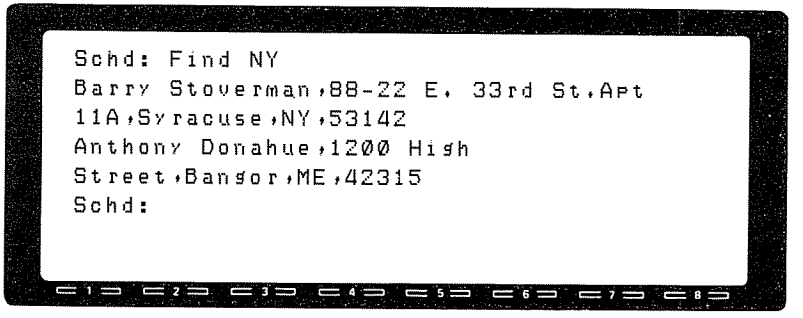


Figure 6.3. Use of SCHEDL's Find command.

Stoverman is a valid find since he lives in New York. But Donahue is not — he lives in Maine. SCHEDL retrieves his record because *ny* is included in *Anthony*. Remember, the program does not search through specific fields but rather searches the entire record. Furthermore, the program ignores capitalization, so *ny*, *NY*, *nY*, and *Ny* are equivalent.

To get around this problem, include a character sequence that is unique to the state field. Usually you can include the field delimiters in the search string. For example, if you tell SCHEDL to find ,NY, chances are good that you will retrieve only those records with NY in the state field. Regardless of the field you are searching through, you can usually come up with a combination that minimizes invalid finds.

So far, we've assumed that all records contain the same type of information. However, you don't need to limit the NOTE.DO file to a single type of data. The great convenience of the SCHEDL/NOTE.DO system is that you can store a grab bag of information in it in random sequence and sort it all out later.

To do this you include a special character in each record, usually as the first character in a record. The Model 100 graphic characters are ideal for this. Suppose you want to include the addresses in figure 6.1 as well as records of long-distance phone calls. Put a telephone symbol (GRPH)(p) at the start of each telephone record and a stick figure (GRPH)(q) at the start of each address record.

Other useful graphic symbols include an airplane (GRPH)(a) for flight reservations, a house (GRPH)(h) for personal records, and an automobile (GRPH)(c) for auto mileage figures.

Figure 6.4 shows four different types of records, each type marked with its own graphic symbol.

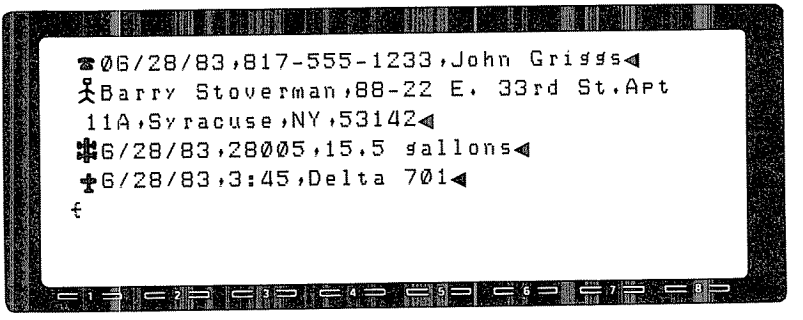


Figure 6.4. Records in NOTE.DO with appropriate graphic symbols.

No matter how much varying information you store in NOTE.DO, you can select a group of related records simply by specifying the unique symbol.

You can use any character. However, if you plan to transmit a file containing graphic characters, both computers must use an eight-bit data-word, since the characters cannot be coded with the seven-bit convention.

Sorting the NOTE.DO file. There are two methods of sorting. The simplest method alphabetizes the records by comparing the characters from left to right. Applying this technique to the mixed NOTE.DO file just described, the outcome is a file sorted as follows:

phone calls
addresses
auto mileage figures
airplane reservations

Here the order is determined by the sorting precedence of the graphic characters used. (See the ASCII table in Appendix D of the Model 100 Owner's Manual for a listing of all Model 100 characters in order of precedence.) precedence.)

Using this method has some limitations. Records with the same starting character are sorted according to the next character to the right, and so forth. This can produce incorrect orderings. For example, if the characters being examined represent dates, then the dates will be sorted in alphabetical rather than chronological order.

A more sophisticated sorting method allows you to specify any field as a sort key, and then sorts using a criterion that is appropriate for the data type of that field. For example, you might want to sort a telephone directory in order of individuals' first names. See SORT.BA in Part 2.

Such a program requires that all the fields be of the same format, however, which is not usually the case with our mixed-bag NOTE.DO file. Therefore we need a way to extract records from the file and save them in another file. See XTRACT.BA in Part 2.

Table 6.3 summarizes the database operations you can perform using TEXT, SCHEDL, and NOTE.DO.

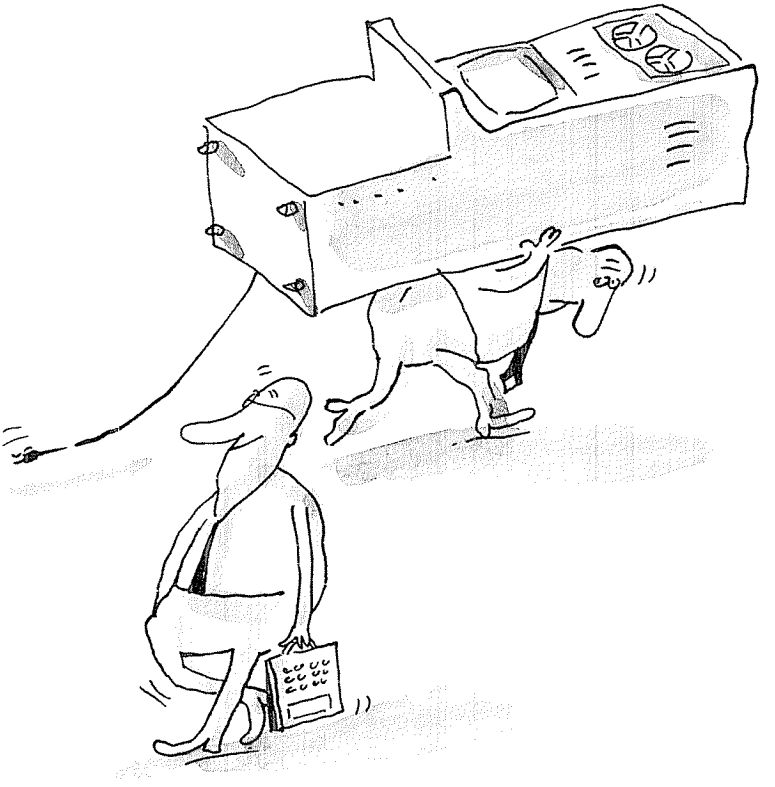
Operation	Method	Sample Record(s)
1. Find a string in any field in the document.	Use SCHEDL Find command and enter the exact string.	No restrictions
2. Find a string in one particular field.	Use SCHEDL Find command and the string with its delimiters, for example: `Find ,Nashua,,`	`Mill House Inn,Nashua,` `$50.00,Sam Roberts◀`
3. Distinguish between multiple record types in NOTE.DO.	Prefix a special character to each record. Graphic characters are useful for this. Use SCHEDL Find command and enter the appropriate graphic character.	✝ `6/28/83,3:45,Delta 701◀` ✱ `6/28/83,28005,15.5 gallons◀` ☎ `817-555-1212,$3.75◀`

4. Sort a file in simple alpha-numeric order from left to right.	File can have one record type or multiple record types with identifying prefixes. Use SORT.BA.	`Albert Duncan,RYCO, 215-555-1010◀`
		`Bruce Raskin,Ventura Systems,801-999-1010◀`
		`Carol Avery,The Brodkin Co.,503-888-0001◀`

5. Find and copy records of the same type into another file.	NOTE.DO contains multiple record types with identifying prefixes. Use XTRACT.BA.	Same as 3 above.

6. Sort a file using a specific field as the sort key.	File must have a single record type. Use SORT.BA.	`Carol Avery,The Brodkin Co., 503-888-0001◀`
		`Albert Duncan,RYCO, 215-555-1010◀`
		`Bruce Raskin,Ventura Systems,801-999-1010◀`

Table 6.3. Database operations you can perform with TEXT, SCHEDL, and two application programs from Part 2.



7 / A Computer in Your Briefcase

In the preceding chapters, you learned how to use the Model 100 for word processing, telecommunication, and database management. But what about arithmetic? Any computer worth its weight in silicon chips *must* be able to compute.

The Model 100 is indeed a capable number-cruncher, far more powerful than a calculator and in many ways more convenient than one. Here are a few of its strong points:

- It performs all computations with 14-digit precision
- It has hundreds of *variables* available for storing intermediate results
- It lets you evaluate long formulas in a single step

In this chapter we survey the Model 100's most important math tools and show how to use them for calculations — no programming knowledge required.

We use the BASIC language for all computations, so if you're following along with the Model 100, select BASIC from the main menu. You'll see a display similar to that shown in figure 7.1.

Figure 7.1. Display after selecting BASIC from the main menu.

An Easy Way to Get Answers

Two BASIC commands store data and print the results of calculations: PRINT and LET.

PRINT displays a result on the screen. Simply type `PRINT` followed by the formula or *expression* that you want evaluated, then press (ENTER). For example, `PRINT 33+7` (ENTER) displays 40 on the screen.

LET stores a result in memory for later recall; this type of command is often called *assignment* because it assigns a name to a piece of data. First make up a name for the data that is to be stored. This *variable name* must start with a letter and can have an optional number or letter added. A few two-letter combinations are not permitted because they have a special meaning in BASIC. They are: IF, ON, OR, and TO.

Type LET followed by variable name = data to be stored. Press (ENTER). For example, LET X=33+7 (ENTER) stores 40 in the variable X. PRINT X (ENTER) displays the contents of X.

To put a new value in X, type in a new assignment command, such as LET X=-100 (ENTER). To see the new contents of X, type PRINT X (ENTER).

The contents of variables are erased when you run a program or exit from BASIC and after certain other BASIC commands not used in this chapter.

You can type ? instead of PRINT and omit the word LET from the assignment command. We take advantage of these two shortcuts in the following examples. Furthermore, to make the text easier to read, we omit the (ENTER) symbol after commands in the following pages. *Remember that you must press (ENTER) at the end of each command.*

Precision and Accuracy

Before using the Model 100 (or any computer) for crucial calculations such as your golf handicap or the number of calories in a serving of spinach souffle, take a few minutes to understand the machine's limitations in accuracy and precision.

The Model 100 can handle numbers as large as 10^{62} and as small as 10^{-64}, but it can't store every number in that range with full precision. It stores the number's magnitude and up to 14 digits.

To demonstrate, type ? 12345678901234567 8 (remember to press (ENTER)). In response to this command, the Model 100 prints its own version of the number on the next line: 1.23456789012345E+17. (Refer to figure 7.2 for screen displays of examples in this section.)

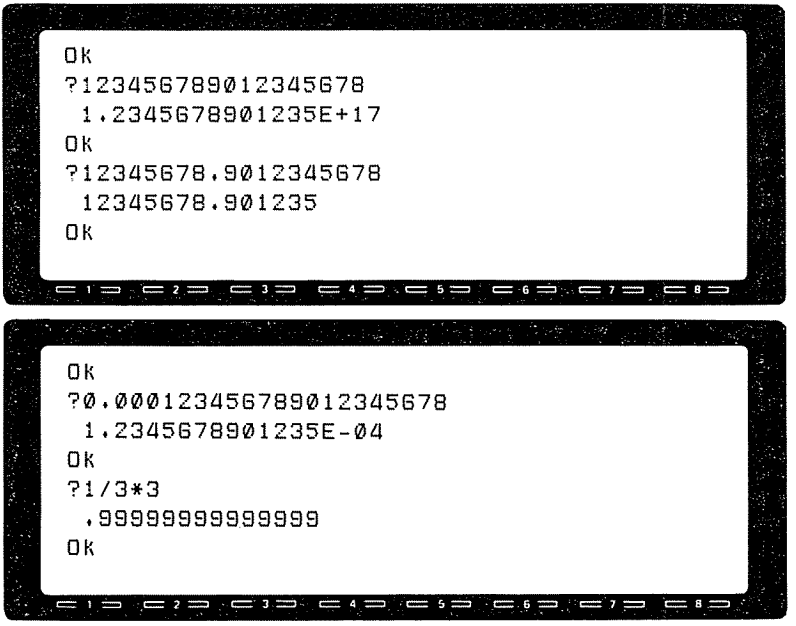


Figure 7.2. BASIC commands demonstrating Model 100 precision and accuracy in handling numbers.

The computer rounds the number to 14 digits and, because the result is larger than 10^{14}, prints it in *exponential form*: `E+17` means "times 10^{17}."

Try `? 12345678.9012345678`. The Model 100 displays `12345678.9012345`, again rounding the number to 14 digits. This time, the computer does not convert the number to exponential form since it is smaller than 10^{14}.

Now try `? 0.000123456789012345678`. The computer prints `1.23456789012345E-04` (the `E-04` means "times 10^{-4}"). In terms of precision, the four leading zeros are not significant; they simply set the magnitude of the number.

For technical reasons, the computer cannot store all fractions exactly. For some fractions, the computer stores an *approximation* that is accurate to 14 digits. This small error can produce unexpected results.

To demonstrate, type `? 1/3*3`. The computer displays `.99999999999999`. The answer is off by 10^{-14}, which is nothing to worry about if you're counting calories or computing a golf handicap. But if the slight inaccuracy bothers you, there are ways (discussed below) to make the computer round-off its results to hide the inaccuracy.

Evaluating Long Formulas

So far we've used simple formulas consisting of two numbers. Even calculators can do that! The Model 100 lets you evaluate far more complicated math expressions.

When entering a math expression that involves two or more operations, you must consider the order in which you want the operations performed. The computer performs the operations according to a preset priority. `? 2+3*4` gives `14`, not `20`, because multiplication has a higher priority than addition. Table 7.1 lists all the math operators and their priorities. (Refer to figure 7.3 for screen displays of commands given in this section.)

When two operators in an expression have the same priority, the computer executes them from left to right. So `? 2*3/2` gives `3`, and `? 2/3*2` gives `1.3333333333333`.

To override the priorities, use parentheses, as in `? (2+3)*4`. The computer gives the result of `20`, completing the operations inside the parentheses first. If in doubt about the order of operations, add parentheses to force the correct sequence.

Operator	Description	Example	Priority
^	Exponentiation	`2 ^ 3`	1
+	Positive sign	`+ 3`	2
-	Negative sign	`- 3`	2
*	Multiplication	`7 * 3`	3
/	Division	`2 / 5`	3
MOD	Modulo	`365 MOD 7`	4
\	Integer division	`13\4`	4
+	Addition	`9 + 15`	5
-	Subtraction	`33 - 11`	5

Table 7.1. Math operators.

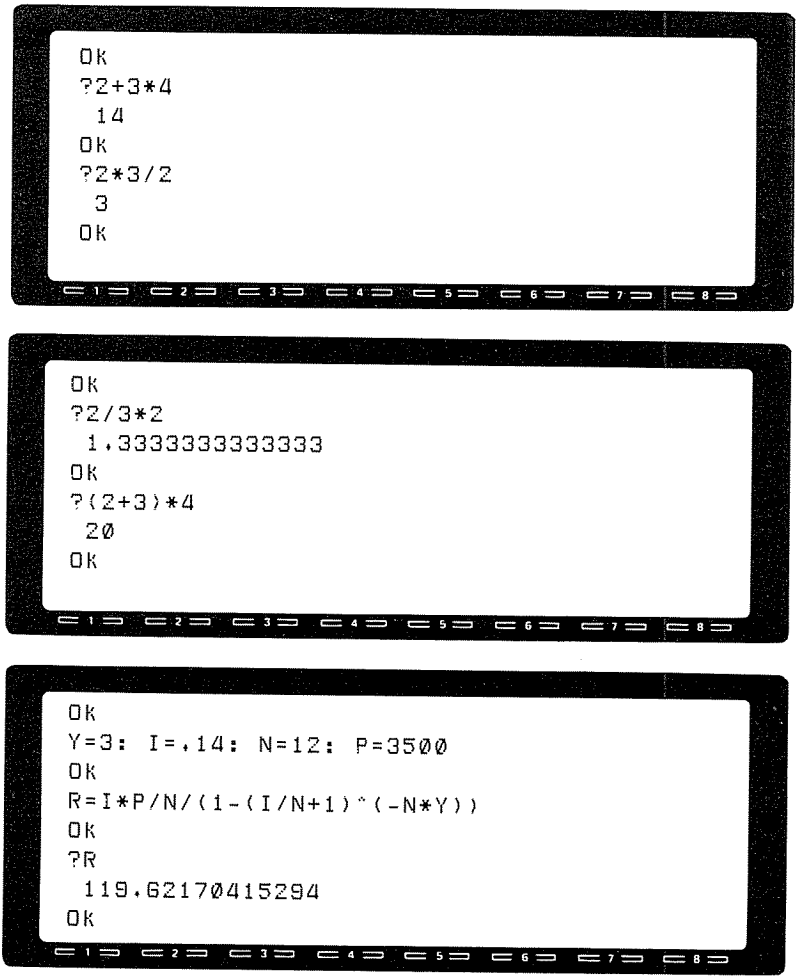


Figure 7.3. Sample commands showing how the Model 100 evaluates longer formulas.

Turning textbook formulas into BASIC expressions is a logical (but sometimes still tricky) operation. As an illustration, take the formula for computing the regular payment on a loan:

$$R = \frac{I(\frac{P}{N})}{1 - (\frac{I}{N} + 1)^{-NY}}$$

The variable I is the annual interest rate, P is the principal, N is the number of payments per year, Y is the number of years, and R is the calculated regular payment.

Translating this formula into BASIC, we have:

```
R = I * P / N / (1 - (I / N + 1) ^ (-N * Y))
```

The parentheses are to preserve the correct order of operations.

Before using this formula, you must assign appropriate values to the variables I, P, N, and Y, using assignment statements. As an example, consider a three-year loan at 14 percent interest with monthly payments. The principal is $3500.

For convenience, you can type several commands on the same line and use a colon to separate each command. Type `Y=3: I=.14: N=12: P=3500`. Now type `R=I*P/N/(1-(I/N+1)^(-N*Y))` to calculate and store the result. Type `? R` to display it.

To see the effect of changing any of the variables, enter new assignment commands and then reenter the formula. Later in this chapter we present a method for storing the formula so you don't have to reenter it each time you want a new calculation.

First let's take a closer look at the Model 100's math tools.

Useful Operators and Functions

Model 100 math tools fall into two categories: operators and functions. Table 7.1 lists the math operators. For a complete listing of math functions, refer to the Model 100 Owner's Manual. You're familiar with many of these operators and functions, so we don't discuss each of them. However, we do pay special attention to a few from each group.

To keep things simple, we use short BASIC commands in the following examples. (Refer to figure 7.4 for screen displays of the BASIC examples in this section.) The real power of many of the operators and functions becomes obvious only later on, when you learn to store formulas and variables in function keys for instant recall.

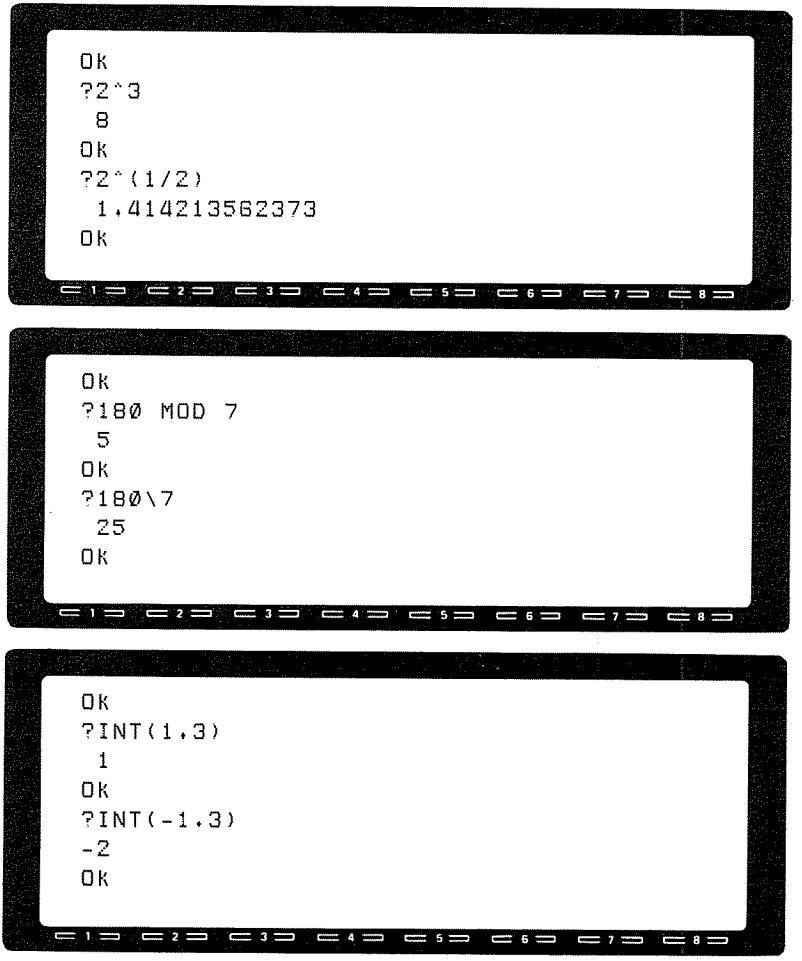
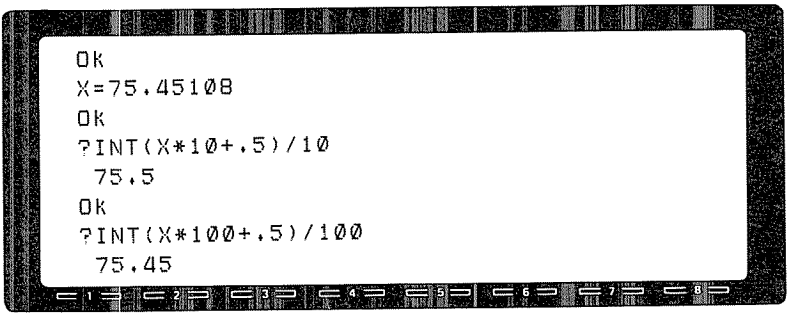
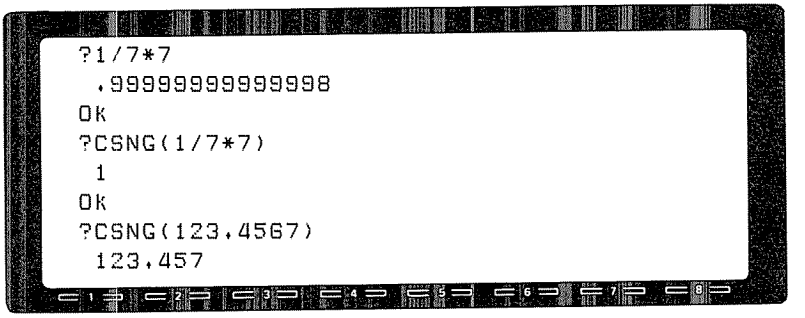


Figure 7.4. Sample screens demonstrating Model 100 math tools (continued on next page).





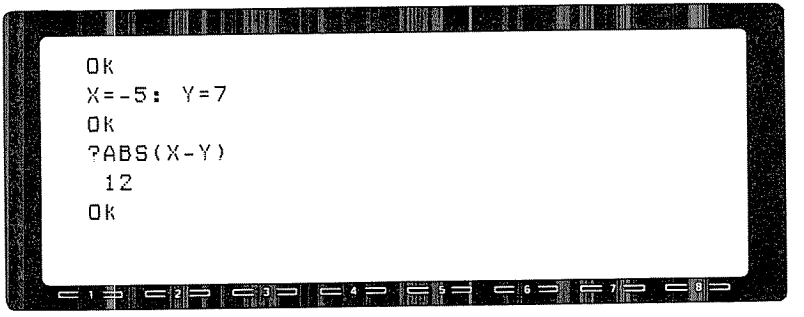


Figure 7.4, continued.

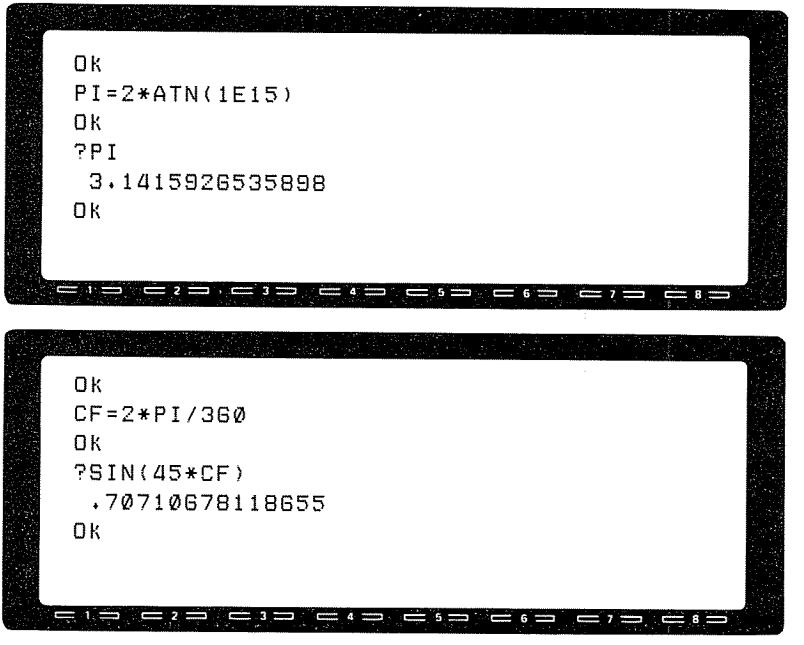


Figure 7.4, continued.

Exponentiation raises a number to a power. For example, to compute 2^3, type `? 2^3`. The same operator gives you fractional roots as well. For example, the square root of 2 is the same as $2^{1/2}$, so type `? 2^(1/2)`. The parentheses force the computer to evaluate ½ before doing the exponentiation.

The **modulo operator** is useful for evaluating cyclic quantities such as dates and times. It calculates the remainder after dividing one value by another. Suppose you want to know what day of the week falls 180 days from today. Type `? 180 MOD 7`. The computer displays `5`, indicating that the future date will fall five days later in the week than today does.

Integer division complements the modulo operation because it gives you the whole number quotient after dividing one number by another, ignoring the remainder. To enter `\`, type (GRPH)(−).

For example, to find the number of weeks in 180 days, type `? 180 \ 7`. The computer displays `25`. There are 25 complete weeks in 180 days (with 5 days left over).

Now for a few of the more useful *functions.*

The **INT (integer) function** computes the largest integer that is not greater than the input value. `? INT(1.3)` gives `1`, and `? INT(-1.3)` gives `-2`. Surprised? −2 is the largest integer not greater than −1.3.

One of the many interesting uses for INT is rounding off numbers. It's a two-step process. Suppose you want to round a number to the nearest tenth. Here's the manual procedure:

1. Check the digit in the hundredth's place; if it's greater than 4, you add one tenth to the number.
2. Discard any digits to the right of the tenth's place.

The BASIC equivalent of this procedure is concise: `? INT(X*10+.5)/10`. Try it, substituting various numbers for X. Rounding to other decimal places is equally simple. For example, to round X to the nearest hundredth, use `? INT(X*100+.5)/100`.

The **CSNG (convert-to-single-precision) function** is helpful when you don't want 14 digits of precision. CSNG rounds a number to six digits. This reduced precision is sufficient to "hide" the inaccuracy with which the computer stores certain fractions. For example, compare the result of `?1/7*7` and `?CSNG(1/7*7)`. Also try `?CSNG(123.4567)`.

The **ABS (absolute value) function** gives the magnitude of a number. Used in conjunction with subtraction, ABS tells you how far apart two values are, without regard to which is the larger. For example, if a package was shipped five days ago and the shipper says it will arrive in seven more days, then let `X=-5` and `Y=7` and type `? ABS(X-Y)` to get the total shipping time. The formula works regardless of which value is larger or which is positive or negative.

The **ATN (arctangent) function** has many applications in engineering and science. One handy trick is to use it to generate an approximation of π.

As a number n gets very large, ATN(n) approaches π. 1E15 represents 10^{15} — large enough. Accordingly, the command `PI=2*ATN(1E15)` stores 3.1415926535898 in the variable $\pi/2$. PI displays the number, accurate to 14 digits.

The Model 100 **trigonometric functions SIN, COS, and TAN** require angles to be input in radians. However, most of us measure angles in degrees. We can use our derived variable PI to convert degrees to radians.

Type `CF=2*PI/360` to store the conversion factor in CF. To convert any angle from degrees to radians, multiply the angle by CF. For example, `? SIN(45*CF)` prints the sine of 45 degrees.

Using Function Keys for Shortcuts

Often you'll want to use the same formula with several different sets of data. It's tedious to retype the calculation command `? formula` each time you change the variables. To avoid this, you can store commonly used formulas for instant recall with the function keys. (Refer to figure 7.5 for screen displays of the BASIC examples in this section.)

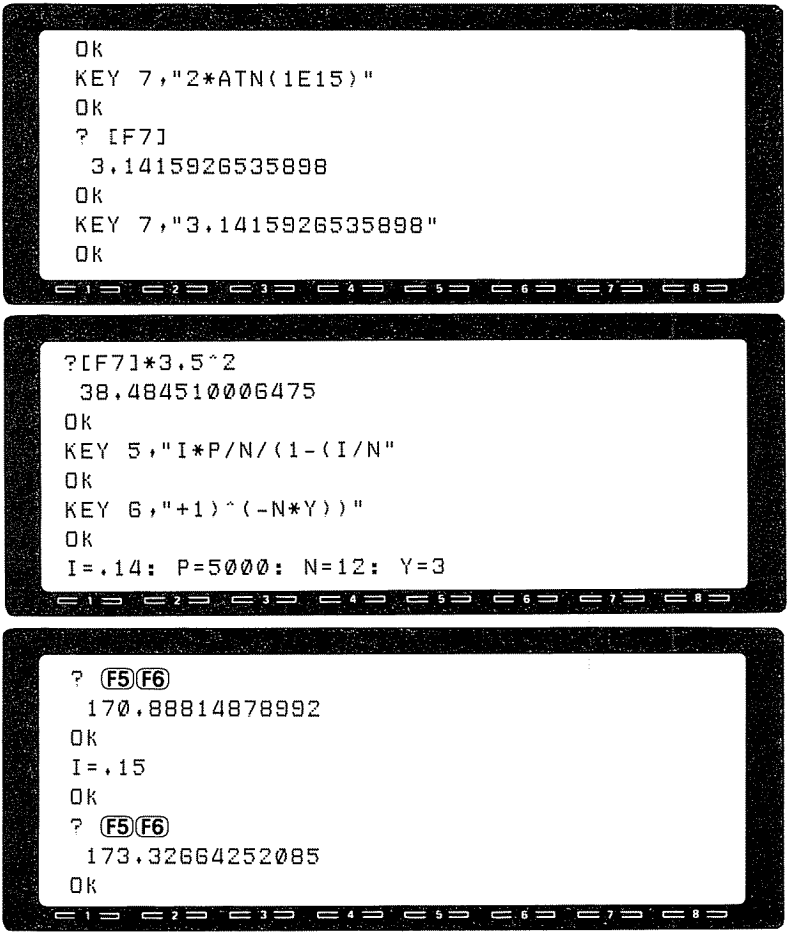


Figure 7.5. Sample screens demonstrating Model 100 function-key shortcuts.

KEY is the BASIC command that defines the contents of the eight function keys. Its format is:

```
KEY Key-number,"text"
```

Suppose you need the constant π quite often. Then assign the calculation 2*ATN(1E15) to a function key such as (F7). Simply type: `KEY 7,"2*ATN(1E15)"`. Remember to press (ENTER). Alternatively, you can store the constant itself by typing `KEY 7,"3.1415926535898"`. Then, whenever you need the value π in a formula, press (F7).

For example, type `?` (F7)`*3.5^2` to calculate the area of a circle with radius 3.5. Pressing (F7) is equivalent to typing in the stored data.

This method of storing formulas limits you to 15 characters per function key. For example, the loan repayment formula, `R=I*P/N/(1-(I/N+1)^(-N*Y))`, is too long. One solution is to assign half the formula to one function key, and the second half to another: `KEY 5,"I*P/N/(1-(I/N"` and `KEY 6,"+1)^(-N*Y))"`. After entering these commands, you can quickly recall the formula by typing (F5) (F6).

After making these function-key assignments, it's much easier to do a series of calculations with the formula. First assign values to all variables used by the formula. For instance, type `I=.14: P=5000: N=12: Y=3`. Then type `?` (F5) (F6) to see the result. To recalculate for a different interest rate, say, 15 percent, type `I=.15` and then `?` (F5) (F6).

To view all the current function-key assignments, press (LABEL). The first four characters assigned to each key appear on the bottom row of the display. Press (LABEL) again to make the labels disappear. Alternatively, type `KEY LIST` to see the the full contents of each function key assignment.

Function-key assignments remain in effect until you change them by entering a new KEY command. To erase a function key assignment, reenter the KEY command with nothing inside the quotes: `KEY 5,"": KEY 6,""`. To restore all eight original function key definitions provided by Radio Shack, type `CALL 23164,0,23366` (ENTER) `CALL 27795` (ENTER).

Notes


RAM
BASIC

8 / Using BASIC with RAM Files

One of the great strengths of the Model 100 is the ease with which its programs and files interact. In previous chapters, you've seen numerous examples of the interplay between TEXT, TELCOM, SCHEDL, and ADDRSS.

BASIC's role has been limited to killing and renaming files. In this final chapter of Part 1, we'll demonstrate a couple of additional ways in which you can make BASIC take a more creative part:

- Storing BASIC commands in a text file and having them performed in a single step when you start BASIC.
- Saving the results of BASIC calculations in RAM files which can then be included in reports, letters, and telecommunication files.

No programming is required, but the procedures you learn will be helpful if you take up programming later on.

Command Files

As you use BASIC more and more for doing calculations and running application programs, you'll have to repeat certain command sequences quite often.

The process of defining function keys (illustrated in the preceding chapter) is a good example. Suppose you have redefined all eight function keys to suit your needs perfectly. Then you run an application program that changes them all. Now you have to recall the function key assignments (from your own memory, not the computer's RAM!) and painstakingly type them in again.

What you need is a *command file* which lets you enter a whole slew of commands at the push of a button. A command file is simply a document containing a logical sequence of BASIC commands.

Table 8.1 lists a set of handy function-key definitions. (For information on unfamiliar commands, see the Model 100 owner's manual.)

KEY Command	Explanation	Sample Use
`KEY 1,"*,45359"`	Pounds-to-kilograms	`?3,5` (F1)(ENTER)
`KEY 2,"*2,54001"`	Inches-to-centimeters	`?12` (F2)(ENTER)
`KEY 3,"*3,785"`	U.S. gallons-to-liters	`?4` (F3)(ENTER)
`KEY 4,"*,06"`	Sales tax	`?28,73` (F4)(ENTER)
`KEY 5,"?TIME$+CHR$(13)"`	Print the time	(F5)
`KEY 6,"?DATE$+CHR$(13)"`	Print the date	(F6)
`KEY 7,"FILES+CHR$(13)"`	Display file names	(F7)
`KEY 8,"MENU+CHR$(13)"`	Go to main menu	(F8)

Table 8.1. Some useful function-key definitions.

That's quite a lot to type in each time the function-key definitions are inadvertently changed. Let's store the commands in a text file instead.

From the main menu, select TEXT and create a file called KEYDEF-.DO. Then type in the eight lines shown in figure 8.1, carefully matching the spacing and format.

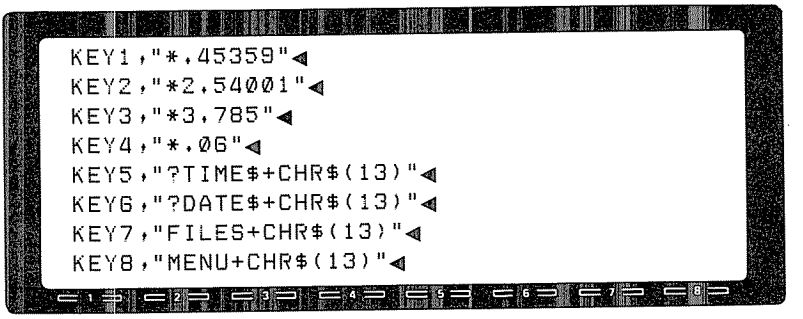


Figure 8.1. Contents of the KEYDEF.DO command file.

To "execute" the command file, follow these steps.

1. Using TEXT, copy the command file into the paste buffer: CTRL ↑ F7 CTRL ↓ F5
2. Close the command file: F8
3. Select BASIC: BASIC ENTER
4. Recall the contents of the paste buffer: PASTE

The commands appear just as if you were typing them (only faster!).

Press LABEL to see the first part of each key definition. Test each function key, using the examples given in the right-hand column of table 8.1. Remember to press ENTER to complete functions F1 through F4; this is not necessary with functions F5 through F8, since the ENTER key is included in those key definitions as CHR$(13).

If the computer prints an error message like ?SN Error or ?FC Error, go back and check your command file. The error is in the command that is printed just above the error message.

Setting up constants. The same technique comes in handy for initializing variables used in calculations.

For example, you could include the assignment:

```
PI=3.1415926535898◀
```

in a command file. Then, after you complete step 5 in the procedure just

given, the variable PI is set to the correct value, and can be used in a formula wherever you need the constant π. (Remember, however, that variable assignments are erased when you run a program, exit from BASIC or execute certain BASIC commands.)

Saving Data in a RAM File

If you haven't done so already, select BASIC from the main menu. The Ok prompt should appear on the display. (Refer to figure 8.2 for displays corresponding to the examples in this section.)

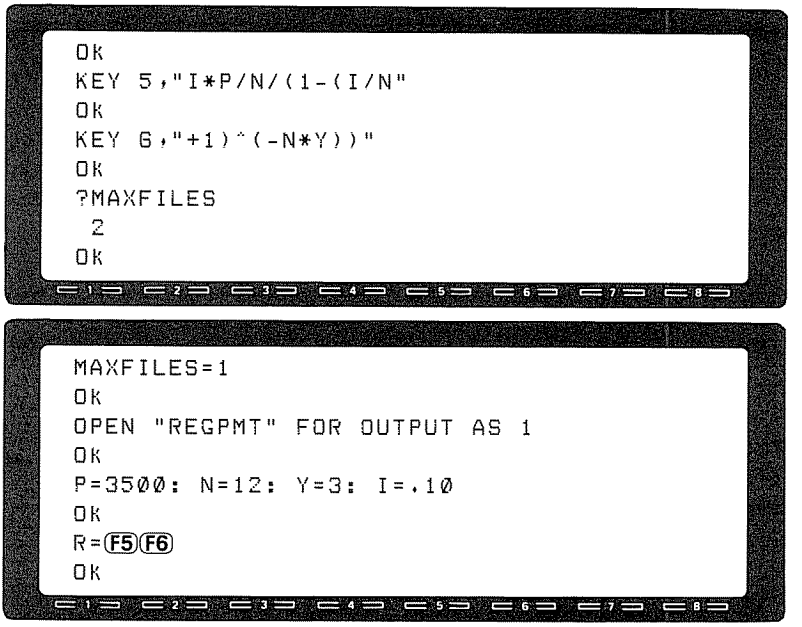


Figure 8.2. Saving calculations in a RAM file.

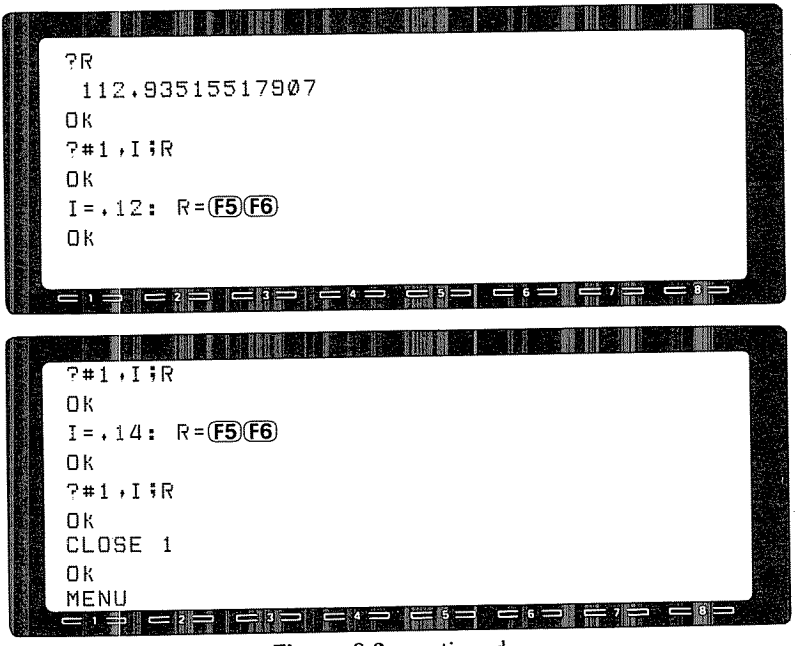


Figure 8.2, continued.

For a practical example, suppose you are exploring the effects of changing interest rates on a three-year loan agreement. We presented the regular-payment formula in the preceding chapter. Store it in F5 and F6 by entering the following commands:

```
KEY 5,"I*P/N/(1-(I/N"
KEY 6,"+1)^(-N*Y))"
```

In this example, we plan to compute the regular payment R for several values of I, and to save the values of I and R in a file. Before outputting data to a RAM file, we must make a few technical preparations.

Setting MAXFILES. The MAXFILES variable determines how many data files you may have in use at once.

To check the current MAXFILES setting, type ?MAXFILES. Remember to press ENTER. The computer prints the current maximum.

To change the MAXFILES setting, type MAXFILES=n, substituting a number for n. Entering this command erases all variables, so do it before

you assign any values. The MAXFILES setting remains in effect until it is changed with another MAXFILES command.

In this chapter, we use only one file at a time, so you should type MAXFILES=1.

Opening a file. Next we open the data file using the OPEN command, which has this general form:

```
OPEN "file" FOR OUTPUT AS 1
```

Replace file with an appropriate file name. For our example, enter the following command:

```
OPEN "REGPMT" FOR OUTPUT AS 1
```

The file is now open and ready to accept data.

Outputting the data. From here on, the procedure parallels that for printing data on the display. Simply use PRINT #1 in place of PRINT (or ? #1 in place of ?).

Using the formula stored in (F5) and (F6), we'll compare the monthly payments for a $3500, three-year loan at three different interest rates: 10, 12, and 14 percent.

Enter the first set of values: P=3500: N=12: Y=3: I=.10. Remember to press (ENTER). Compute the regular payment R by typing R=(F5)(F6). To check the result before storing it in RAM, type ?R.

We want to store I and R in the RAM file, so type: ?#1, I; R. (Notice the semi-colon after I. It serves as a delimiter between the two variables.)

Set I to the next value, recalculate, and store the result: I=.12 then R=[F5] [F6] and ?#1, I;R.

Set I to the final value, recalculate, and store the result: I=.14 then R=[F5] [F6] and ?#1, I;R.

Closing the file. As soon as you finish outputting to the file, close it by typing CLOSE 1 (ENTER). This command ensures that your information is stored properly.

Now let's find out what the RAM file looks like. Type MENU (ENTER) to return to the main menu, then select the file REGPMT.DO for editing with TEXT. It should resemble the display in figure 8.3.

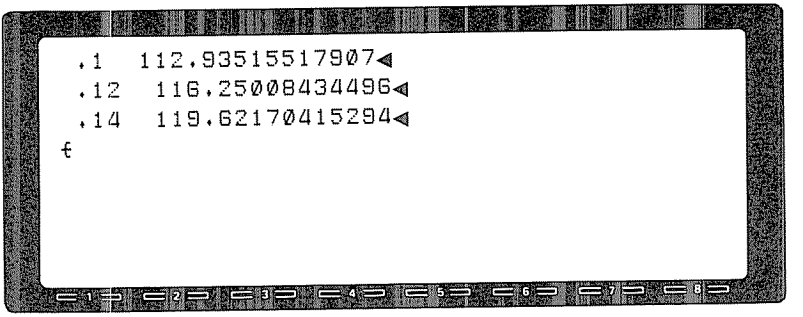


Figure 8.3. Contents of the RAM file REGPMT.DO.

Using Calculations in Another Document.

The whole point of this exercise is to use the calculations in another document. Figure 8.4 shows a sample letter; only the numbers are missing. To continue with this example, type in the letter now.

Copying the information is quick and easy. Select the REGPMT.DO file, and copy it into the paste buffer: (CTRL)(↑) (F7) (CTRL)(↓) (F5).

Now select the letter, position the cursor where the first number goes, and press (PASTE). Presto! The data appears. Use (TAB) and ▭ to line up the columns.

In summary, the steps for saving data in a RAM file are:

1. Select BASIC.
2. Set MAXFILES = 1
3. Open the RAM file.
4. Perform the calculations and output the data using `PRINT #1, data`.
5. Close the RAM file.
6. Exit from BASIC and edit the RAM file as needed using TEXT.

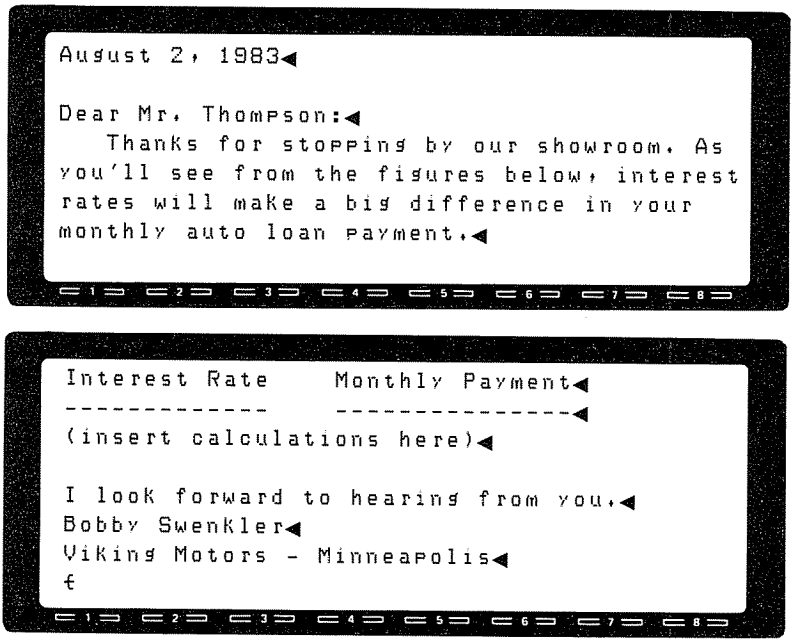


Figure 8.4. Copy the interest-rate calculations into this letter.

Part 2: Application Programs

Introduction

An application is a program that accomplishes a specific, useful function. The 12 programs described in Part 2 provide applications in word processing, telecommunication, database management, and computation.

All the programs are written in BASIC and are stored on the Software Cassette that was packaged with this book. To run an application program, follow these steps.

1. Read the detailed program description in the following pages to get specific instructions on running the program. Make sure you have enough memory (RAM) to run the program; memory requirements are listed at the beginning of the program description.
2. Connect a cassette recorder to the Model 100.
3. Adjust the cassette recorder Volume control to slightly above the middle level (it may take a little fiddling to find the correct setting).
4. Insert the Software Cassette into your recorder. The cassette may be recorded on both sides; find out which side contains the program you want.
5. Rewind the cassette and put the recorder into the Play mode.
6. On your Model 100, select BASIC from the main menu.
7. The Model 100 should be displaying the prompt, OK. Type CLOAD "program" (ENTER) replacing program with the exact name of the program (except that you may omit the suffix .BA). For example, to load the program SR.BA, you would type CLOAD "SR" (ENTER).
8. The Model 100 will find and load the program.
9. Type RUN (ENTER) to run the program.

After trying the program and before you exit from BASIC to the main menu, you can save the program in memory. Type SAVE "program" (ENTER). The next time you want to run the program, you can simply select it from the main menu.

The programs are meant to be educational as well as practical. To assist you in studying or modifying them, we've included complete listings for each program (see Appendix).

Hints and Tips

If you get an OM error (out of memory) while running a program, you will usually need to delete some documents and start over with the program. Use the KILL command described in your Model 100 Owner's Manual.

If you get an OS error (out of string space) while running a program, you may need to reserve more space for the storage of strings (text). Find the CLEAR *xxx* command in the program listing. Increase the value of *xxx* and try the program again. (To change a program line, retype the entire line, including the line number, or use the edit mode. See your owner's manual for details.)

Whenever a program asks you for the name of an input or output file, you may specify a RAM file or a cassette file. To specify a cassette file, prefix `CAS:` to the file name. Be sure the cassette recorder is ready before you do this. (Use the Play mode for input files and the Record mode for output files.)

If you accidentally press (BREAK) or (CTRL)-(C) while running a program, the computer will stop and print a message like `Break in line number`. You can usually continue with the program by typing `CONT` (ENTER).

A / Word Processing

Global Search and Replace Program

Program Name: SR.BA

Purpose: To search for and replace or delete any sequence of characters throughout a document.

Size (.BA format): 804 bytes

Minimum Memory Requirements: 8K

Operation: The program reads the original text from one file (the "input file") and copies the corrected text to a different file (the "output file"). If your Model 100 has enough memory, both the input and output files can be in RAM. Otherwise, one of the files can be on cassette.

First the program asks for the names of the input and output files.

The program next asks you to specify the string to search for. Type it in and press CTRL Z to signify the end of the string. If you make a mistake while typing in the string, press BKSP to erase it.

If you press ENTER or any other control character, the corresponding control character will be included as part of the search string. The familiar dummy characters (e.g., ^M for ENTER) will appear on the display. The ability to find and replace control characters comes in handy in many applications involving documents that are sent or received via telecommunication.

After you end the search string by pressing CTRL Z, the program asks you for the replacement string. Here again, whatever you type will be included in the replacement string — including control characters. Type BKSP to erase errors, and CTRL Z to end the string.

The program finds all occurrences of the string, replaces them with the new string, and puts the results in the output file.

To delete a string rather than replace it, type CTRL Z without any preceding characters for the replacement string. This says, "Replace the search string with an empty string."

To count the number of times a string occurs without replacing it, use an identical value for the search string and the replacement string.

Sample Dialogue To change a misspelled word in a RAM file:

```
CLOAD "SR" (ENTER)

Ok
RUN (ENTER)
Input file? MEM419 (ENTER)

Output file?
(ENTER for LCD:) MEMO (ENTER)
Search for?
(^Z when done, ^C to cancel)
Paulson (CTRL)-(Z)
Replace with?
(^Z when done, ^C to cancel)
Paulsen (CTRL)-(Z)
Found 5
Ok
```

Load the program from cassette.

MEM419 is an existing file in RAM.

MEMO will be a new file in RAM.

Search for the name Paulson.

Change it to Paulsen. The computer found and replaced five occurrences. The file named MEMO contains the changed text.

To display a cassette file with all tabs deleted (the changes are not saved since the LCD is used for output):

```
RUN (ENTER)
Input file? CAS:AUG15 (ENTER)


Output file?
(ENTER for LCD:) (ENTER)
Search for?
(^Z when done, ^C to cancel)
(TAB)(CTRL)-(Z)
Replace with?
(^Z when done, ^C to cancel)
(CTRL)-(Z)
(file contents)
Found 7
Ok
```

Name a cassette file. Get cassette ready to play before continuing.
Output to the LCD.

Search for tab (^I).

Delete (replace with nothing).
Found and deleted seven tabs.

Character and Word Count Program

Program Name: SIZER.BA

Purpose: To count the number of characters and words in a document.

Size (.BA format): 483 bytes

Minimum Memory Requirements: 8K

Operation: This program counts the number of characters and words in a document file. The character count lets you know how much room the file takes up in memory. The word-count gives you a more intuitive idea of the length of the text.

In counting the number of words, the program treats the following characters as word-delimiters: tab, line feed, carriage return, space, hyphen, period, and slash. The end of file is also considered to be a delimiter.

The program first asks you to name a document. Enter the name of a RAM or cassette file. (If you are using a cassette file, be sure to have the recorder ready to play before running the program.) After a brief pause while it counts the characters and words, the program prints the totals on the screen. The program takes about one minute to process a 3000-character document.

Sample Dialogue

```
CLOAD "SIZER" (ENTER)
OK
RUN (ENTER)
Document name? SAMPLE (ENTER)
SAMPLE contains 2976
characters and 528 words.
OK
```

Load the program from cassette.

SAMPLE is the name of an existing document.

Document Printer

Program Name: PRINT.BA

Purpose: To print text files in a neat format with page headings, page numbers, and margins.

Size (.BA format): 2624 bytes

Minimum Memory Requirements: 8K

Operation: The program is designed to print documents created with the TEXT word processing program. It formats the text according to your specifications and prints it on the line printer or stores it in another file for later use.

After you name the document and the output device, the program displays the current format settings. You can change any of these by selecting an option from the parameter menu and then typing in a new setting. The parameters are:

1 - Page width: The width (in pica-size characters) of the paper including margins. Use 80 for standard 8½ by 11 inch paper.

2 - Page length: The length of the paper (measured in lines) including margins. Use 66 for standard 8½ by 11 inch paper.

3 - Left margin: The number of spaces to print in the left-hand margin.

4 - Right margin: The number of spaces to print in the right-hand margin. Ordinarily, you use the same value as that for the left margin.

5 - Top margin: The number of blank lines at the top of the page.

6 - Bottom margin: The number of blank lines at the bottom of the page.

7 - Line space: Use 1 to print single-spaced text, 2 for double-spaced text, etc.

8 - Number pages: Select YES to print page numbers as part of the page heading, NO to omit them.

9 - Title pages: Select YES to print a title as part of the page heading, NO to omit a title.

10 - Page pause: Select YES for a pause between pages for use with sheet-fed printers, NO for continuous printing for use with continuous-form-fed printers.

11 - Skip headings on first page: For letters and other documents, you may want to omit the page number and title from the first page only. Select YES to omit these on page 1, NO to include them.

To change one of these settings, enter the option number. The program will prompt you for a new value, and will then re-display the parameter menu. Enter 0 when all the parameters are set correctly.

If you have requested a title, the program will ask you to enter a one-line title.

Now the program will print the document.

The program takes quite a while to format a document, so you won't want to use it for rough copies; use the TEXT program's (SHIFT)(PRINT) function for that. Use PRINT.BA for your final copy.

Sample Dialogue

CLOAD "PRINT" (ENTER)	Load the program from cassette.
Ok	
RUN (ENTER)	
Print which file?	Name a RAM file.
LETTER (ENTER)	
Output device (press ENTER for printer)? (ENTER)	Select the printer for output.
Enter a one-line title maximum 60 characters.	The program asks this question only if you have selected the titles option.
Letter to Ralph Jones, September 14, 1983(ENTER)	Your title.
Align paper then press ENTER? (ENTER)	Press (ENTER) to begin printing.

B / Telecommunication

Telecommunication Cost Estimator

Program Name: XTIME.BA

Purpose: To estimate the cost of sending or receiving a document over the telephone lines.

Size (.BA format): 941 bytes.

Minimum Memory Requirements: 8K

Operation: The program figures the total cost of sending or receiving a file, as determined by the file size, communication parameters, and costs charged by the host computer, telephone company, and network system. (For an explanation of communication parameters, see Chapter 3).

How do you measure the file size? If you are sending a file, use the application program, SIZER.BA. If you are planning to receive a file, you can probably find the file size in advance using one of the host computer's commands such as CAT (catalog) or DIR (directory).

The program computes the cost based on the shortest possible call; cost may be higher due to pauses in communication.

Sample Dialogue

```
CLOAD "XTIME" (ENTER)                    Load the program from
OK                                       cassette.
RUN (ENTER)
Telecomm time and cost
estimator

Document length (1-29905)?               Enter the number of
11000 (ENTER)                            characters.
Info rate in bps
(300,600,1200)? 300 (ENTER)              Bits per second.
Data word length (7 or 8)?
7 (ENTER)                                Bits per word.
Parity (E/I/O/N)? I (ENTER)              Ignore parity.
Stop bits (1 or 2)? 1 (ENTER)            One stop bit.
Host computer's hourly rate
(0-99.99)? 25 (ENTER)                    Cost per hour of
                                         connect time.
```

```
Network surcharge per minute
(0-9.99)? .10 (ENTER)

Phone system's minute rate
(0-9.99)? 0 (ENTER)

The file transfer will take
at least 7 minutes and cost
at least    $3.62.
Ok
```

Cost per minute for network service (Tymnet, Telenet)

Long distance or other charge.

Terminal Program

Program Name: TERM.BA

Purpose: To allow communication with a host computer, including uploading and downloading cassette files.

Size (.BA format): 2609 bytes

Minimum Memory Requirements: 8K

Operation: The program works like the built-in Model 100 program TERM, but adds the following extra functions:

- Download text from the host to a cassette file
- Upload to the host from a cassette file
- Download to an existing RAM file without erasing its previous contents (append new text onto old)
- List the RAM file directory while on-line
- Get the current time while on-line
- Cancel the printer-echo function without interrupting download or upload operations.
- Get the amount of free memory while you are downloading.

If you have already read Chapter 3 and used the Model 100's built-in TERM program, you will have no difficulty learning to use TERM.BA.

First the program lets you view and change the communication parameters; press (ENTER) to leave a setting unchanged, or enter a new value.

The program assumes you are using the Model 100's built-in modem and autodialer. You can store the host computer's telephone number in the program (line 110) or simply type in the number when the program asks for it. Include the angle-brackets after the number with the auto log-on se-

quence inside, if desired. (If you omit the angle brackets, the program will add them.)

After you enter the host's telephone number (or press ENTER to use the pre-set number), the program dials the host and executes the auto log-on sequence, if provided. Next it beeps, displays a new set of function key labels, and puts you into the interactive terminal mode.

If the computer doesn't beep within 15 seconds after the last digit is dialed, press BREAK to interrupt the program. Check the phone number and try again.

Interactive terminal mode. In the interactive terminal mode, the eight function keys have the following definitions:

1. `File:` List all user RAM files.
2. `Down:` Prepare to download.
3. `Up:` Prepare to upload.
4. `Full/Half:` Switch between full and half duplex.
5. `Echo:` Switch the printer echo function on or off.
6. `^C:` Send a control-C ("Break character") to the host computer. This is the only way to send a control-C or break character. You cannot use BREAK or CTRL C, because these keys will interrupt the BASIC program. If you accidentally type CTRL C or BREAK while running the program, you will see the message `Break in 260` or some other line number. Type `CONT` ENTER to continue.
7. `Time:` Display the current time.
8. `Bye:` End the call. Before using this key, log-off from the host.

Downloading. After you select the Download function, the program asks you to name a download file. To specify a RAM file, simply enter the file name. The program will then ask whether you want to append data or create a new file. If you specify append, the data will be added to the end of the file; otherwise, any previous data in the file will be erased.

To specify a cassette file, prefix `CAS:` to the file name, as in `CAS:AUG15`. With cassette files, you will not have the append/new file option.

Before entering a cassette file name, be sure the cassette recorder is ready to record.

Next the program returns you to an interactive terminal mode, except that now all incoming data is stored in the file.

During downloading, the function keys have the following definitions:

1. `File.`
3. `Mem:` Prints the amount of memory remaining. This function is available only when downloading to a RAM file.

(2) X-Dn: Stops downloading.
(4) Full/Half: Switch between full and half duplex.
(5) Echo: Switch printer echo on and off.
(6) ^C: Send a control-C or "break" character.
(7) Time.
(8) Bye.

To stop downloading, press (F3) until the function-key labels change.

Uploading. After you select the upload function, the program will ask for the file name. Specify a RAM or cassette file as explained previously.

Before entering a cassette file name, be sure the cassette recorder is ready to play.

During uploading, the function keys have the following functions:

(1) File.
(2) No function.
(3) X-Up: Stop uploading.
(4) Full/Half: Switch between full and half duplex.
(5) Echo: Switch printer echo on and off.
(6) No function.
(7) Time.
(8) Bye.

To stop uploading, press (F3) until the function-key labels change.

Note: If you stop the program by pressing (BREAK), the function labels will not be restored to their usual definition. Type GOTO 860 to restore the labels.

Sample Dialogue. The following dialogue shows a typical use of the program to call the Compuserve Information Service.

```
CLOAD "TERM" (ENTER)                    Load the program from
                                        cassette.
OK
RUN (ENTER)
Word length=7
Press ENTER to leave as-is
or ENTER a new value (7/8)?
(ENTER)

Parity=I
Press ENTER to leave as-is
or ENTER a new value
(E/O/I/N)? (ENTER)
```

```
Stop bits=1
Press ENTER to leave as-is
or ENTER a new value (1/2)?
(ENTER)

Phone number to call is
555-1111<>
Press ENTER to leave as-is or
ENTER a new number?
883-0884<=^C> (ENTER)


Calling host computer...
8830884
^C
User ID:
```

The pre-set number is stored in line 110; you can change it. Dial the approprate number, wait two seconds, send a control-C character.

After completing the auto log-on, the program beeps and enters the interactive terminal mode. You can now communicate with the host by typing on the keyboard.

Automatic Data Retrieval Program

Program Name: QUOTE.BA

Purpose: To call a host computer at a preset hour, request data and store it in a file, then hang up automatically.

Size (.BA format): 2483 bytes

Minimum Memory Requirements: 8K

Operation: Before adapting this program to your own needs, you must know the exact key sequence required to retrieve data from the host computer. You then store that information in the program.

The program interacts with the host computer according to a series of data-triplets which you store at the end of the program (see lines 670-830 in the listing).

Each triplet consists of the following:

1. A command to send to the host computer.
2. A prompt to wait for before continuing.

3. A number indicating whether to store the information that is received prior to the reception of the prompt (0 = don't store, 1 = store)

To indicate a control character, use the familiar dummy character notation, for example, ^C for control-C. Use quotation marks whenever an item includes punctuation.

When calling Compuserve, the first data triplet might be: `^C,U,0`. This means send a control-C, wait for the letter U, and don't save any characters received ahead of the letter U. (U is the first letter of User ID:, sent by Compuserve.)

The next triplet would be something like this: `"70000,999^M",P,0`. This means: send 70000,999 (the user ID) followed by a carriage return; wait for the letter P (the first letter of Password:); and don't save anything received ahead of P.

Next would come something like this: `PASSWORD^M,!,0`, meaning: send PASSWORD (your secret password) followed by a carriage return; wait for a "!"; and don't save the intervening data.

After the preceding exchange, you would be at the command level of Compuserve, and your next data triplet should request the specific database you need. Here is a sequence of triplets that requests stock information:

`G FIN-6^M, ":", 0`	Request page 6 of Microquote; wait for ":".
`QUOTES^M,":", 0`	Specify the Quotes program; wait for ":".
`^M,^JI,0`	Carriage return; wait for a line feed followed by I.
`TAN^M,^JI,0`	Stock symbol for Tandy Corp.; wait for a line feed followed by I.
`T^M,^JI,0`	Stock symbol for AT&T; wait for line feed followed by I.
`^M,"!",0`	Carriage return; wait for "!".
`"1,2,3,4,5^M",PROGRAM,1`	Request five items and store the incoming data up to receipt of PROGRAM.
`/OFF^M,Off,0`	Sign off; wait for Off.

To use the program as a demonstration, store the correct telephone number in line 660 (be sure to include the angle-brackets). Store your user ID and password in place of those contained in the program (lines 710-720).

QUOTE.BA is set up to call Compuserve and request stock information for five companies. It stores the information in a file called QUOTE.DO. To change the list of companies for which data is requested, change lines 730-800.

If you do change the company list or any of the data triplets, be sure that the number in line 680 is the same as the number of data triplets.

When you run the program, it asks you when you want the call made. To make the call immediately, simply press (ENTER). Otherwise, type in a call-alarm time in 24-hour format.

Next the program will ask for a time limit for the call; this is a safeguard in case the dialogue between the Model 100 and the host doesn't go as expected; after the specified time has elapsed, the Model 100 will automatically terminate the call. A safe value to use would be five or ten minutes, depending on the amount of information you have requested.

After you specify the time limit, the computer will display the current time and the alarm time. At the specified time, the computer will make the call.

Sample Dialogue. In this dialogue, text that is generated by the program is underlined. The dialogue worked at the time this book was written, but might require modification by the time you read this.

```
Call at what time
(hh:mm:ss)? 23:00:00 (ENTER)          Call at 11 PM.
Time limit for call (1-60
minutes)? 5 (ENTER)                   Five-minute time limit.
```

The program next clears the screen and displays the call-alarm time and the current time.

After completing the call and hanging up, the computer displays a message like this:

```
On at 23:00:00. Off at 23:04:00
The call was completed normally.
```

If the call was terminated by an error or exceeded the time limit, the computer will print that information.

The received data is stored in QUOTES.DO. Use TEXT to examine it.

C / Databases

Database Program

Program Name: DB.BA

Purpose: To enter, retrieve, and update information using a personalized data entry form.

Size (.BA format): 4487 bytes

Minimum Memory Requirements: 16K

Operation: DB.BA is a simple but flexible program for entering and maintaining information that is ordinarily entered onto paper forms. Before using this program, you should review Chapter 6, which defines many of the terms we'll be using.

The first step in preparing a database for DB.BA is to design the screen-entry form, showing where the labels and blanks appear. The form must be narrow enough to fit on the 40-column screen. However, it can be as long as you want; the program will automatically divide it into pages that fit onto the eight-line LCD.

To show you exactly how to use DB.BA, we'll create a database from the form shown below. After you've used it, you can begin designing your own forms.

Our form contains 10 lines and 11 fields:

```
LAST NAME:_____
FIRST NAME:_____
STREET ADDRESS:_____
APT. #:_____
CITY:_____
STATE:_____ ZIP:_____
SOCIAL SECURITY #:_____
BIRTHDAY (MM/DD/YY):_____
ANNUAL PLEDGE: $_____
LAST CONTRIBUTION (MM/DD/YY):_____
```

To use this form with DB.BA, we first specify the exact data type and length of each of the blanks or "fields" (to use the terminology of Chapter 6).

Here's the breakdown of our sample form:

Field Identifier	Data Type	Length (alpha) or format (numeric)
LAST NAME:	Alpha	25
FIRST NAME:	Alpha	15
STREET ADDRESS:	Alpha	25
APT. #:	Alpha	15
CITY:	Alpha	35
STATE:	Alpha	2
ZIP:	Alpha	5
SOCIAL SECURITY #:	Alpha	11
BIRTHDAY (MM/DD/YY):	Alpha	8
ANNUAL PLEDGE: $	Numeric	######.##
LAST CONTRIBUTION (MM/DD/YY):	Alpha	8

Notice that alpha fields require just a length specification, but numeric fields require more information: the placement of the decimal point (if any) must also be specified. We do this by using the # symbol to indicate a digit position, and by placing the decimal point where it will appear. A person's age can be represented as ## (two digits without a decimal point). A monetary figure ranging from 0.00 to 99999.99 is represented as #####.##.

Any number of # symbols may be used on either side of the decimal point, and the decimal point may be omitted for integers. The important thing is to provide enough # symbols to the left of the . for all the digits in the largest number that will be stored.

After you have planned the database in this manner, you're ready to create a "format file" in the Model 100 using the TEXT program. The format file tells DB.BA how many fields each record contains, what type of data goes in each field, and how to display the information on the screen.

The format file presents all that information in the following sequence:

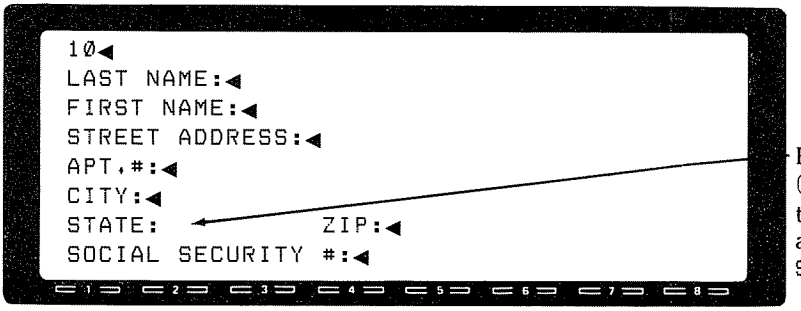
1. The number of lines in the form (NL).
2. The actual lines of the form; there must be NL of them, and each must be terminated by a carriage return (ENTER).
3. The number of fields in the form (NF).
4. Three descriptive items for each field.

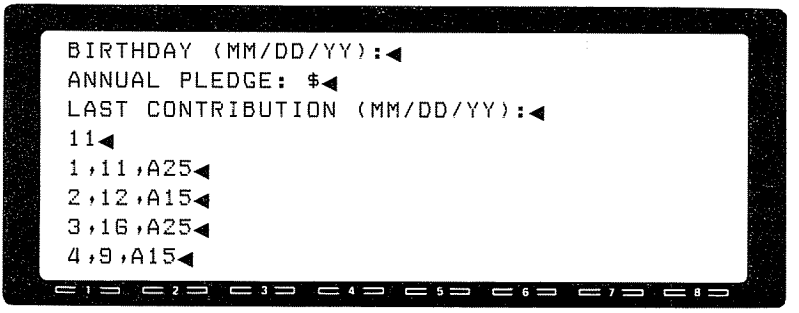
The first and second descriptors are numbers giving the line and column at which the field starts.

The third descriptor is the display format. For alphanumeric information, use the letter A followed by the length, e.g., A8. For numeric fields, use the letter N followed by a format specification such as #####.##, explained previously.

The first field in our sample form (LAST NAME) is defined by: 1,11,A25. It starts on line 1, column 11, and is an alphanumeric field of length 25. The tenth field (ANNUAL PLEDGE) is defined by: 9,15,N#####.##

With that explanation, you're ready to type in the format file. Use the TEXT program to enter the information exactly as it appears on the following sample display screens. Name the file FORM.DO.





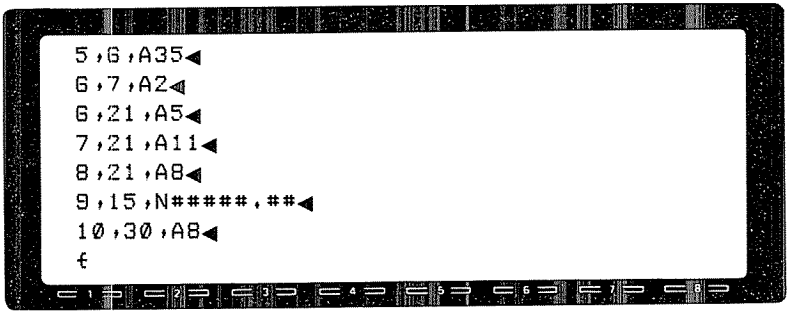


Figure: Before trying DB.BA, store this information in a file called FORM.DO. DB.BA will use FORM.DO as a format file.

When you run DB.BA, the program first will list all the user RAM files. Your format file should be included among them. The program asks if you want to update or add records. For this initial run, specify "Add."

The program asks you to name the format file. Enter FORM, which is the name of our sample format file.

Next the program will ask for the name of the output file — the file that will contain the new records. Type DATA.DO. (In general, you may specify any RAM or cassette file.) The program asks whether to add to or destroy the previous contents of DATA.DO. Specify "Destroy."

The program will then display the first page of the entry form with the first field highlighted. Type in some typical data, pressing ENTER after each field. As you are typing in the data, it will appear on the seventh line of the display; when you press ENTER, it will be placed into the highlighted field of the form.

After entering the first six fields, and just before entering the seventh, your display will resemble the one shown below.

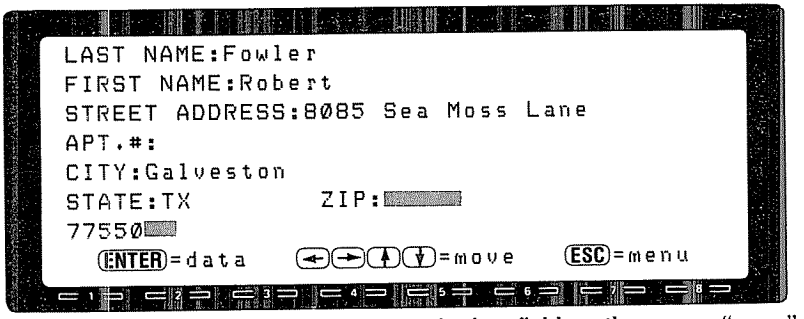
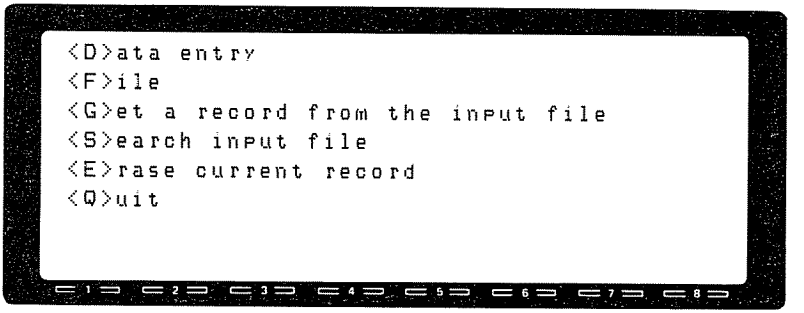


Figure: Sample display before entering the last field on the screen "page."

You can move around from field to field using ← and →. Use ↑ and ↓ to move backwards and forwards one "page" at a time (this only applies if the form exceeds six lines). When you complete the last field on the screen, the next screen, if there is one, will automatically be displayed.

Once you have filled in all the fields of a record, you are ready to file the record. Press ESC for the menu, which gives you these options:



Press F to file the record. To return to the data entry mode, press D. Continue with this sequence until you have entered all the data. After filing your last record, press Q to exit from the program.

Later, when you want to update an existing file such as DATA, run the program and specify the "Update" option. The program will ask you to name the input file. Enter DATA.DO (or whatever file contains the data). The program now asks for the name of the output file. Enter a different name, such as DATA1.DO.

The program asks if you want to append or destroy existing data in DATA1.DO. Ordinarily, you will select "Destroy," since DATA1.DO is a new file.

Now the computer will show you a blank entry form. Press ESC G to get the first record from the input file. Make any changes you need, then press ESC F to file the updated record. Get the next record and continue this process (enter changes, file the record, get the next one) until you reach the end of the input file.

When updating an existing database, you can locate a specific record quickly with the Search function. Before selecting this function, use the data entry mode to highlight the field you want to search through. Then press ESC S.

The program will remind you that the search will use the highlighted field, and will then prompt you to enter the search string. You may search

starting at the next record or at the first record in the input file. If the program finds a matching record, it will put you into the data entry mode with the matching record displayed.

Sample Dialogue. The following dialogue shows how you would update the DATA.DO file.

```
CLOAD"DB" (ENTER)                          Load program from
RUN (ENTER)                                cassette
<U>pdate or <A>dd records? (U)
ADRS .DO   ENTER .BA SORT .BA              The program displays
DB   .BA*  FORM  .DO DATA .DO              the file directory
Name of format file? FORM (ENTER)
Name of input file? DATA (ENTER)
Name of output file? DATB (ENTER)
<D>estroy old output file
(if any) or <A>ppend? (D)
```

The program displays an empty form. You can type in a new record now, or get one from the input file. To get one from the input file, press (ESC) (G). You will then see the first record displayed. Make your changes, if any, and press (ESC) (F) to file the record in the output file DATB.DO.

Suppose you want to search for persons who live in Texas. In the data entry mode, press the arrows until the STATE field is highlighted. Now press (ESC) (S). The dialogue continues like this:

```
Search hi-lited field only, OK? (Y/N) (Y)
Enter the Search value: ?TX (ENTER)
Start at 1st or next record? (F/N) (F)
```

If the computer locates a matching record, it will return to the data entry mode and display that record.

Sorting Program

Program Name: SORT.BA

Purpose: To sort data files using any data field as the sort-key. The data records can consist of either fixed-length or variable-length fields.

Size (.BA format): 3307 bytes

Minimum Memory Requirements: 8K

Operation: This program lets you change the sequence of information in a document. The document must consist of uniformly arranged records, each record terminated with a carriage return. However, the fields within the record may be either variable length separated by commas, or fixed length separated by column boundaries. (For definitions and illustrations of terms like variable-length and fixed length records, review Chapters 5 and 6.)

Before trying SORT.BA, you need a sample data file to be sorted.

We'll use two sample data files, named DATA.DO (variable-length fields) and QUOTES.DO (fixed-length fields). The sample screens below show how each document appears when you select it from the main menu. Rather than running the BASIC programs DB.BA and QUOTES.BA to create the files, you may simply type in the sample data as it appears here:

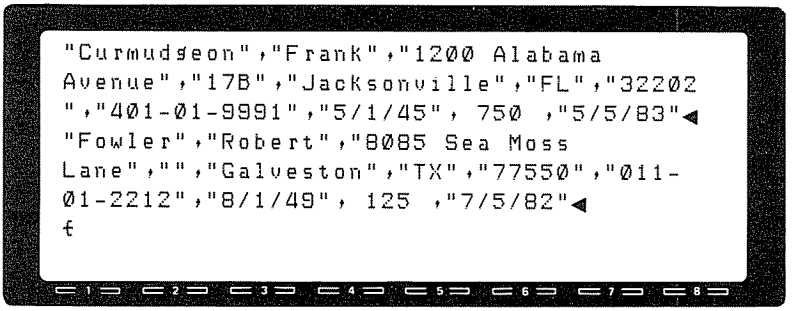


Figure: A a variable-length-field database called DATA.DO. You may add your own data to the file, so long as the record format remains the same.

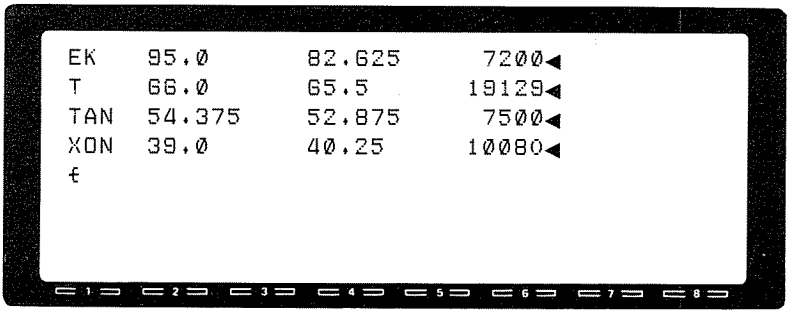


Figure: A fixed-length-field database called QUOTES.DO.

When you run SORT.BA, the program lists all user RAM files and asks you to name the input file — that is, the file that you want to sort. To use a RAM file, type in the name. To use a cassette file, prefix `CAS:` to the file name.

Next the program asks you to name the output file to contain the sorted data. If you simply press (ENTER), the sorted data will be displayed on the screen but not saved in a RAM file.

After you have named the input and output files, the program will show you a sample record from the file, and ask you if the file contains fixed-length or variable-length fields. You'll be able to tell from the sample record (review Chapter 6 if you're not sure). Files created with DB.BA always consist of variable-length fields. Files downloaded from an information service in tabular form usually consist of fixed-length fields.

Variable-length fields. If you indicate that the fields are of variable length, the program analyzes the sample record to locate each field. The analysis will work without fail only if all fields are separated by commas and all alphanumeric fields are enclosed in double-quotes, as shown in the figure above.

The program next shows you the fields it has located. If you agree with its analysis, the program asks you which field to use as a sort key.

Finally, the program asks what type of data the field contains: alphanumeric, numeric, or date (mm/dd/yy). Your answer determines which sorting criterion will be used.

Note that dates need not include leading zeroes; for example, 1/3/81 and 01/03/81 are equivalent.

The program stores the sorted information in the specified output file.

Fixed-length fields. If you indicate that the file contains fixed-length fields, the program displays the sample record along with a column guideline, and asks you to enter the starting and ending column of the sort field. Be sure to allow for leading or trailing spaces that may be part of the sort field. For example, the sample record might have the number 123 in columns 3-5; however, columns 1-2 might also be part of the field to accommodate larger numbers. The field should be specified as 1-5 if that is the case.

The program displays the field you selected, asks for your confirmation, and then asks what kind of data it contains (alphanumeric, numeric, or date).

The sorted data is stored in the specified output file.

Sample Dialogue. The sample dialogue shows the use of the program with the sample data in DATA.DO.

```
CLOAD"SORT.BA" (ENTER)
Ok
RUN (ENTER)
DATA .DO SORT .BA* QUOTES.DO
```

The program lists all your RAM files.

```
Name of input file?
DATA (ENTER)
Name of output file? (ENTER)
```

Output the result to the LCD. If a cassette file is to be used, get the recorder ready to record first!

```
Here is a sample record:
"Curmudgeon","Frank","1200
Alabama Avenue","17B",
"Jacksonville","FL",
"32202","40 1-01-9991",
"5/1/45", 750 ,"5/5/83"
Fixed- or variable-length
fields (F/V)? (V)
```

File contains variable-length fields separated by commas.

```
Analyzing sample record
Record contains 11 fields.
Fields are
1 - "Curmudgeon"
2 - "Frank"
3 - "1200 Alabama Avenue"
4 - "17B"
5 - "Jacksonville"
6 - "FL"
7 - "32202"
8 - "401-01-9991"
9 - "5/1/45"
10 - "750 "
11 - "5/5/83"
Agreed (Y/N)? (Y)
Sort on which field (0 to
review)? 10 (ENTER)
```

```
What type of data is in
that field
<A>lpha   <N>umeric   <D>ate mm/dd/yy
Select one: A/N/D (N)
Sorting the fields now
Sort completed. Writing
data to the output file.
"Fowler","Robert","8085 Sea
Moss Lane"," ","Galveston,
"TX","77550","011-01-2212"
,"8/1/49", 125 ,"7/5/82"
"Curmudgeon","Frank","1200
Alabama Avenue","17B",
"Jacksonville","FL","32202",
"40
1-01-9991","5/1/45", 750,
"5/5/83",
Sort completed
Ok
```

Program may pause here, depending on the number of records to be sorted.

Fowler's record is first since 125 precedes 750.

Sorted records only show when the LCD is used for the output file.

Data Record Extraction Program

Program Name: XTRACT.BA

Purpose: To extract selected records from a database and store them in another file.

Size (.BA format): 745 bytes

Minimum Memory Requirements: 8K

Operation: The program is intended for use with SCHEDL and NOTE.DO, as explained in Chapter 6. As explained in that chapter, you can store various record types in the NOTE.DO file, identifying each type with a special character at the beginning of each record. For example, use the telephone symbol (GRPH)(P) to identify telephone records, and the car symbol (GRPH)(C) for car rental records.

The figure below shows this use of the NOTE.DO file.

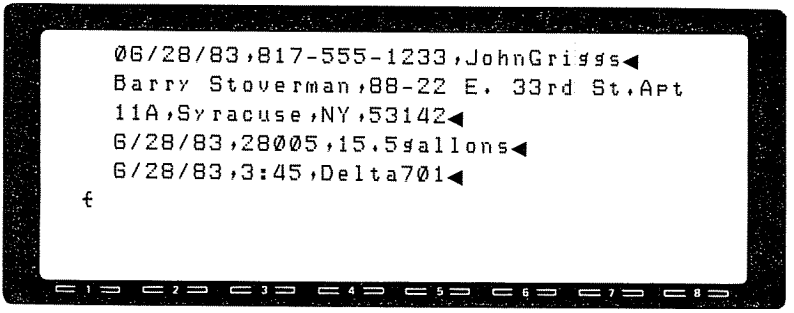


Figure. Records in NOTE.DO using graphic symbols to identify four different record types.

First the program lists the user RAM file directory and asks you to specify an input and an ouptut file. The input file can be NOTE.DO or any other data file in which each record is terminated with a carriage return.

To specify a cassette file, prefix `CAS:` to the file name (get the recorder ready to play or record before running the program). To use the LCD instead of an output file, simply press (ENTER) when the program asks for the output file name.

Next the program asks you to enter the character or characters to search for. Type in the graphic symbol or other identifying information and press (ENTER).

The program will ask you which columns to search through. In the case of fixed-length fields (tables), this specification allows you to search through a single column of the table. In the case of variable-length fields, it is not meaningful. Enter the starting and ending column numbers, or press (ENTER) to search the entire record.

The program will output all matching records to the file you specified.

Sample Dialogue

```
CLOAD"XTRACT" (ENTER)
Ok
RUN (ENTER)
DATA  .DO    XTRACT.BA*  NOTE.DO
Name of input file? NOTE (ENTER)
Name of output file? (ENTER)
```

Press (ENTER) to send ouptut to LCD.

```
Enter character(s) to
search for (GRPH)(C)                    Search for car records.
Enter starting and ending
columns for search
(0=entire record)? 1,1 (ENTER)          Search column 1 only.
   6/28/83,28005,15.5                   Found one matching
gallons                                 record.
End of file.
Ok
```

Data Entry Program

Program Name: ENTER.BA

Purpose: To allow quick entry and storage of data, checking for probable entry errors.

Size (.BA format): 1847 bytes

Minimum Memory Requirements: 8K

Operation: ENTER.BA is for those times when you must type in large amounts of data, numeric or alphanumeric, into a tabular format. The program lets you specify the number of columns (from 1 to 8) and the minimum and maximum for numeric items.

The program stores the data as variable-length fields separated by commas, with a carriage return at the end of each record (or line, if you are thinking of the data as a table). Therefore the data file can be read in by DB.BA (with a suitable format file), SORT.BA, or SCHEDL.

ENTER.BA starts by asking you to name the output file. For a RAM file, simply type in the name. For a cassette file, prefix `CAS:` to the file name. Have the cassette recorder ready to record before running the program.

After you name the file, the program asks whether to append to or destroy the file's previous contents, if any. Select A or D. The program asks how many items per record. (This also means how many columns, if you're thinking of a table.)

The program asks you for the name and data type of each field. For numeric fields, the programs asks for the minimum and maximum acceptable values.

Once all this initial information has been supplied, the program beeps and requests data for the first item, second item, etc.

Whenever you are to enter a numeric field, the program displays the acceptable minimum and maximum. If you enter an out-of-range number, the program gives a warning beep and asks if you want to accept or cancel the entry.

After you complete a record (enter all the fields in a record), the program plays a special sequence of notes and asks whether to store or redo the current record.

To stop the program at any time, type an empty line when the program asks for a data item.

Sample Dialogue

```
"CLOAD"ENTER.BA" (ENTER)
Ok
RUN (ENTER)
TEMPS .DO    ENTER .BA*  DATA   .DO
Name of output file?
TEMPS (ENTER)
<A>ppend to or <D>estroy
existing file (if any) (A)          Add to existing data.
How many items per record
(1-8)? 2 (ENTER)

Name of item 1 (10
characters max)
?Time hh:mm (ENTER)                 The program will use
                                    this as a prompt for
                                    item 1.
Data type of 'Time hh:mm'
<A>lpha or <N>umeric (A)            Time is alphanumeric.
Name of item 2 (10
characters max)
?Temp (F) (ENTER)
Data type of 'Temp (F)'
<A>lpha or <N>umeric (N)
Enter minimum and maximum           For each numeric item,
values                              the program requests a
for 'Temp (F)' (ENTER=no            minimum and
limit).                             maximum acceptable
Minimum value? -100 (ENTER)         value. Values must
```

```
Maximum value? 150 (ENTER)

Enter data for 'Time hh:mm'
ENTER=quit
? 11:30 (ENTER)
Enter data for 'Temp (F)'
ENTER=quit
Min=-100          Max= 150
? 58 (ENTER)
<S>tore  <R>edo (S)
```

range from -100 to 150 degrees F.
Actual data entry begins here.

The program files the record and asks for another. It repeats this sequence until you press ENTER at the `Enter data` prompt.

D / Computation

Calculator Program

Program Name: CALC.BA

Purpose: To evaluate complicated formulas without the need to use BASIC PRINT and LET commands.

Size (.BA format): 2885 bytes

Minimum Memory Requirements: 8K

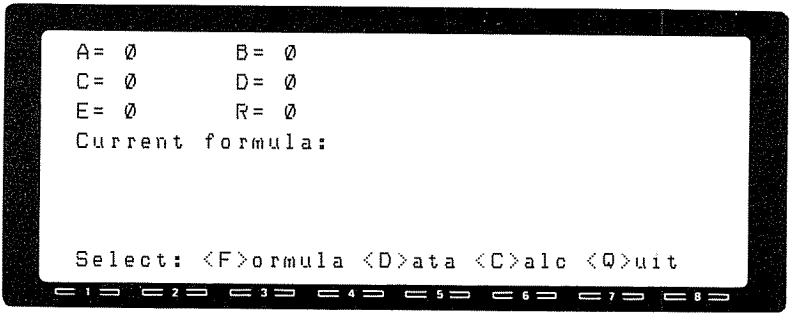
Operation: CALC.BA works like a handheld calculator but has these extra features:

- Entry of complicated formulas
- Ability to store and recall up to six data items
- Double-precision results

In Chapter 7, you learned how to use BASIC commands to do computations. CALC.BA eliminates the need to use BASIC commands such as PRINT and LET. You simply type in a formula such as 33*12/71 and tell the computer to calculate. You can prestore data such as A=356 and then use A in formulas: A*25+32.

The program evaluates expressions using the same hierarchy of operations that BASIC follows; to override the normal hierarchy, you can use parentheses. (See chapter 7 for an introduction to computation with the Model 100 and an explanation of certain technical terms.)

When you start the program, you'll see the following screen:



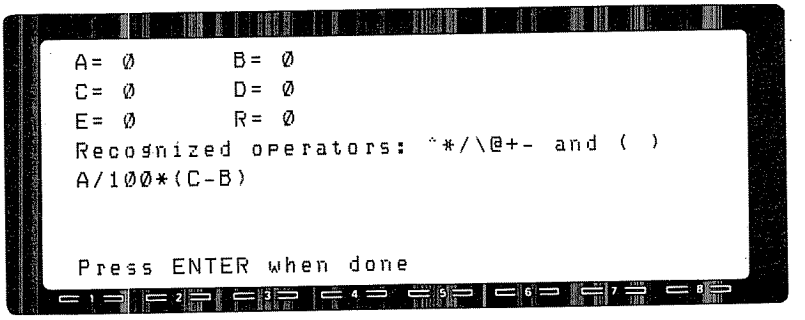
The main menu prompt, shown on the bottom line of the display, is `Select: <F>ormula <D>ata <C>alc <Q>uit`. We'll cover the first three options (F), (D), and (C), one at a time.

Setting up a formula. Press (F) from the main menu. The computer will list the math operation that you can use in the formula. See Chapter 7 for an explanation of these operations.

^	Exponentiation
*	Multiplication
/	Division
\	Integer division
@	Modulo
+	Addition or positive sign
–	Subtraction or negative sign
( )	Parentheses to override the normal hierarchy

Type in the formula, including variables if desired. Press (ENTER) when you are finished. The main prompt will return to the bottom line.

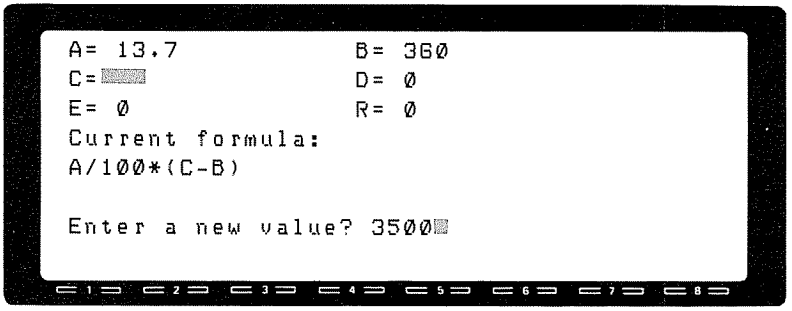
Here's a sample screen during formula entry:



Storing data. The six variables or data-storage registers are: A, B, C, D, E, and R. R is a special register because it always retains the result of the most recent calculation.

To store a number in any of the registers, press (D) for data. The prompt on the bottom line will change to: `Change which variable (ABCDER)?`. Specify the variable you want to change and the corresponding label on the screen will become highlighted. Now enter the data you wish to store. As soon as you press (ENTER), the new value will be shown beside the register, and the computer will repeat the prompt, `Change which variable (ABCDER)?`.

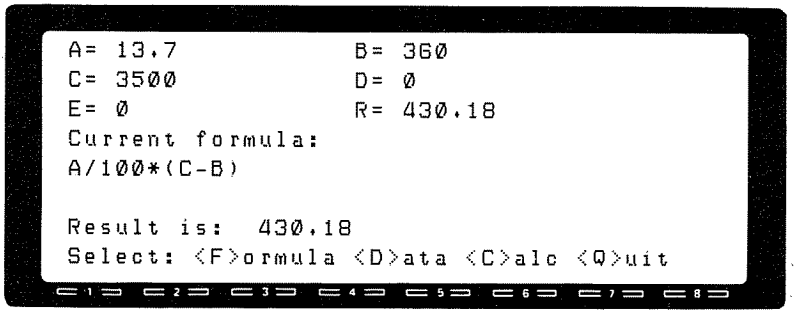
Here's a sample screen during data entry (registers A and B have been set; register C is being set):



When you have finished setting the variable registers, press (ENTER) in response to the `new value` prompt. The main prompt will return to the bottom line.

Calculating. When the formula and data registers are set, press (C). The computer will display the message `Calculating . . .` while it is analyzing the formula and calculating a result. Then it will display the result in full double-precision on line seven, and in single-precision beside the R register. However, the R register (and all the other registers, too) store numbers in double-precision.

Here's a sample screen after a calculation:



You can change the data and recalculate without changing the formula, or vice versa, or you can start over with new data and a new formula.

If you enter an invalid formula, the program will warn you and ask you to check the formula. However, if you enter an improper operation (such as

dividing by zero), the program will print an error message and stop. Other than division by zero, the most common error will be an overflow, in which a result is larger than the computer can handle ($+/-$ 10^{62}). In that case, you must re-run the program.

The sample screen listed previously takes the place of a sample dialogue.

Data File Calculator

Program Name: TCALC.BA

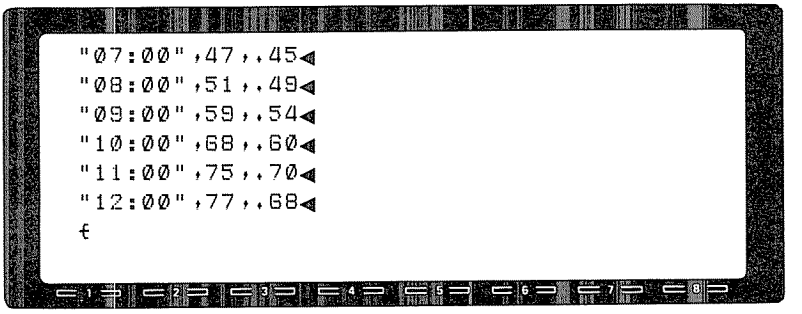
Purpose: To perform row- or column-calculations on a table stored in a data file.

Size (.BA format): 1612 bytes

Minimum Memory Requirements: 8K

Operation: This program computes sums and averages of tabular data. The data must be stored as a series of variable-length fields, each record terminated by a carriage return. This is the same format used by DB.BA, SORT-.BA, ENTER.BA and SCHEDL. Typically, you would create the data file by downloading it from an information service, or by typing it into DB.BA or ENTER.BA.

The screen reproduced below shows some typical data (time, temperature, humidity) in an appropriate format, as seen using the TEXT editor.



To test the TCALC.BA program, enter the sample data into a file called TEMPS.DO.

The program lets you select row or column calculations. We'll demonstrate each in the sample dialogue.

Sample Dialogue

```
CLOAD"TCALC" (ENTER)
Ok
RUN (ENTER)
NOTE  .DO  TCALC .BA*  TEMPS.DO
Name of input file?
TEMPS (ENTER)
Analyzing a sample row.
The sample row contains 3
columns.
The sample column entries
are
1 - "07:00"
2 - "47"
3 - ".45"
Agreed (Y/N)? (Y)
Compute <R>ow or <C>olumn
totals? (C)
Compute totals for which
column (0 to review)? 2 (ENTER)
Total of column 2 is 377
Average of 6 items is
62.833333333333
Ok
RUN (ENTER)
NOTE  .DO  TCALC .BA*  TEMPS.DO
Name of input file?
TEMPS (ENTER)
Analyzing a sample row.
The sample row contains 3
columns.
The sample column entries
are
1 - "07:00"
2 - "47"
3 - ".45"
Agreed (Y/N)? (Y)
```

Total and average temperature.

```
Compute <R>ow or <C>olumn
totals? R
```

Row totals aren't meaningful in this case, but are shown for example only.

```
Include column 1 in the
total? N ENTER
Include column 2 in the
total? Y ENTER
Include column 3 in the
total? Y ENTER
Total of row 1 is 47.45
Average of 2 items is
15.816666666667
Total of row 2 is 51.49
Average of 2 items is
17.163333333333
```

Column 1 contains the time of day, not a number.
Column 2 contains the temperature reading.
Column 3 contains the humidity reading.

The program prints totals and averages for every row in the table.

SR.BA

```
10 'SR.BA: Global Search and Replace
     Program
20 'Copyright 1983 by Tandy Corp. All
     rights reserved.
30 MAXFILES=2: CLEAR 1000
40 DEFINT A-P: DEFSTR Q-Z
50 FILES
60 LINEINPUT"Input file? ";WI
70 LINEINPUT"Output file? (ENTER for LCD:)
     ";WO: IF WO=""THEN WO="LCD:"
80 R="": S=""
90 PRINT"Search for?"
100 GOSUB 230: S=T
110 PRINT"Replace with?"
120 GOSUB 230: R=T
130 LS=LEN(S)
140 K=0
150 OPEN WI FOR INPUT AS 1
160 OPEN WO FOR OUTPUT AS 2
170 PS=1: T=""
180 IF EOF(1)THENPRINT#2,T;: PRINT"Found"K:
     CLOSE: END
190 Q=INPUT$(1,1)
200 IF Q=MID$(S,PS,1)THEN T=T+Q: IF PS=LS
     THEN220ELSE PS=PS+1: GOTO 180
210 PRINT#2,T;Q;: GOTO 170
220 K=K+1: PRINT#2,R;: GOTO 170
230 T=""
240 PRINT"(^Z when done, ^C to cancel)"
250 U=INPUT$(1): IF U=CHR$(26)THEN330
260 IF U<>CHR$(8)THEN300
270 IF LEN(T)=0THEN250
```

```
280 PRINT CHR$(127);: IF RIGHT$(T,1)<"
     "THENPRINTCHR$(127);
290 T=LEFT$(T,LEN(T)-1): GOTO250
300 T=T+U
310 IF U>=CHR$(32)THENPRINTU;: GOTO250
320 PRINT "^"CHR$(ASC(U)+64);: GOTO250
330 PRINT: RETURN
```

SIZER.BA

```
10 'SIZER.BA: Character and Word Count
     Program
20 'Copyright 1983 by Tandy Corp. All
     rights reserved.
30 MAXFILES=1
40 DEFINT A-Z
50 FILES
60 LINE INPUT "Document name? ";FI$
70 NC=0: NW=0
80 W=1: D=0
90 S2=0
100 WD$=CHR$(9)+CHR$(10)+CHR$(13)+
     CHR$(32)+"-"+".,"+"/"
110 OPEN FI$ FOR INPUT AS 1
120 S1=S2
130 IF EOF(1) THEN GOSUB 210: GOTO 180
140 C$=INPUT$(1,1)
150 NC=NC+1
160 IF INSTR(1,WD$,C$)=0 THEN S2=W: GOTO
     120
170 GOSUB 210: GOTO 120
180 CLOSE
190 PRINT FI$ " contains" NC "characters":
     PRINT "and" NW "words."
```

```
200 END
210 S2=D: IF S1=W THEN NW=NW+1
220 RETURN
```

PRINT.BA

```
10 'PRINT.BA: Document Printer
20 'Copyright 1983 by Tandy Corp. All
     rights reserved.
30 MAXFILES=2: CLEAR 500: DEFINT A-Z
40 DIM TF$(1),PM$(11),PM(11)
50 PW=1: PL=2: LM=3: RM=4: TM=5: BM=6:
     LS=7: NF=8: TF=9: PZ=10: FF=11
60 VW=40: VH$=CHR$(27)+"p":
     VN$=CHR$(27)+"q": EF$=CHR$(27)+"J"
70 TF$(0)="NO": TF$(1)="YES": PN$="Page"
80 FOR I=1 TO 11: READ PM$(I): NEXT I
90 DATA Page width,Page length,Left
     margin,Right mgn.,Top margin,Bottom
     mgn.,Line space,Number pgs,Title
     pgs,Pg-pause,Skip headings 1st pg
100 FOR I=1 TO 11: READ PM(I): NEXT I
110 DATA 80,66,10,10,6,6,1,1,1,0,0
120 FF$=CHR$(12): CR$=CHR$(13):
     LF$=CHR$(10): TB$=CHR$(9):
     BS$=CHR$(8): SP$=" ": HY$="-":
     CC$=FF$+CR$+LF$+TB$+BS$+SP$
130 CLS
140 FILES
150 INPUT"Print which file"; FI$
160 INPUT"Output device (press ENTER for
     printer)"; FO$
170 IF FO$="" THEN FO$="LPT:"
```

```
180 CLS: FOR I=1 TO 7: PRINT @
    VW/2*(I-1),VH$;I;VN$;PM$(I)" ="PM(I);:
    NEXT I
190 FOR I=8 TO 11: PRINT
    @VW/2*(I-1),VH$;I;VN$;PM$(I)" =
    "TF$(PM(I));: NEXT I
200 S=0: PRINT: INPUT"Change which setting
    (0=no change)";S
210 IF S=0 THEN 250
220 IF S>=1 AND S<=7 THEN PRINT"Enter a new
    value for '"PM$(S)"'";: INPUT PM(S):
    GOTO 180
230 IF S>=8 AND S<=11 THEN PRINT
    "'"PM$(S)"' (YES/NO)";: INPUT S1$: IF
    S1$="YES"OR S1$="yes" THEN PM(S)=1
    ELSE IF S1$="NO" OR S1$="no" THEN
    PM(S)=0
240 GOTO 180
250 LW=PM(PW)-PM(LM)-PM(RM):
    TL=PM(PL)-PM(BM):
    LM$=STRING$(PM(LM),SP$)
260 CLS: IF PM(TF)=1 THEN PRINT "Enter a
    one-line title
    maximum"LW"characters.": LINE INPUT
    TI$: TI$=LEFT$(TI$,LW)
270 IF FO$="LPT:" OR FO$="lpt:" THEN INPUT
    "Align paper then press ENTER";RD$
280 OPEN FI$ FOR INPUT AS 1
290 OPEN FO$ FOR OUTPUT AS 2
300 CLS
310 C$="": PN=0
320 IF EOF(1) THEN CLOSE: PRINT "Done":
    GOTO 140
330 IF PM(PZ)=1 THEN INPUT"Press ENTER to
    print next page"; RD$
```

```
340 PN=PN+1: LN=0
350 IF LN<PM(TM) THEN GOSUB 660: GOTO 350
360 IF PN=1 AND PM(FF)=1 THEN 400
370 IF PM(TF)=1 THEN BH$=TI$: GOSUB 670
380 IF PM(NF)=1 THEN BH$=PN$+STR$(PN):
     GOSUB 670
390 IF PM(TF)=1 OR PM(NF)=1 THEN GOSUB 660:
     LK=1: GOSUB 700
400 IF EOF(1) THEN 530
410 IF LN>=TL THEN 630
420 C1$=C$: C$=INPUT$(1,1)
430 ON INSTR(1,CC$,C$) GOTO
     530,550,570,590,620,650
440 B$=B$+C$: IF C$>=SP$ THEN CP=CP+1
450 IF CP<LW THEN WF=0: GOTO 400
460 WF=1: LB=LEN(B$): SP=1: HP=0: S=1
470 HQ=HP: HP=INSTR(S,B$,HY$): IF HP>0 THEN
     S=HP+1: GOTO 470
480 HP=HQ: S=1: SP=0
490 SQ=SP: SP=INSTR(S,B$,SP$): IF SP>0 THEN
     S=SP+1: GOTO 490
500 SP=SQ: BP=-(SP>HP)*SP-(HP>=SP)*HP
510 IF BP=0 THEN GOSUB 680: GOTO 400
520 B1$=RIGHT$(B$,LB-BP): B$=LEFT$(B$,BP):
     GOSUB 680: B$=B1$: CP=LEN(B$): GOTO
     400
530 AF=0:WF=0:IF B$>""THEN GOSUB 680
540 GOTO 630
550 IF WF=0 THEN GOSUB 680ELSE WF=0
560 GOTO 400
570 IF C1$=CR$ OR AF=1 OR WF=1 THEN 400
580 GOSUB 680: GOTO 400
590 I1=8-CP MOD 8: I2=LW-CP
600 TB=-(I1<I2)*I1-(I2<=I1)*I2
```

```
610 CP=CP+TB: B$=B$+STRING$(TB,SP$): GOTO
    450
620 B$=B$+C$: CP=CP-1: WF=0: GOTO 400
630 IF LN<PM(PL) THEN GOSUB 660: GOTO 630
640 GOTO 320
650 IF CP=0 THEN 400ELSE 440
660 PRINT #2,"": LN=LN+1: RETURN
670 PRINT #2,LM$BH$: LN=LN+1: RETURN
680 IF B$<>"" THEN PRINT#2,LM$;
690 PRINT#2,B$: LN=LN+1: B$="": CP=0: LK=1
700 IF LK<PM(LS) AND LN<TL THEN GOSUB 660:
    LK=LK+1: GOTO 700
710 RETURN
```

XTIME.BA

```
10 'XTIME.BA: Telecommunication Cost
    Estimator
20 'Copyright 1983 by Tandy Corp. All
    rights reserved.
30  PRINT "Telecomm time and cost
    estimator": PRINT
40 INPUT "Document length (1-29905)"; L
50 IF L<1 OR L>29905 THEN 40
60 INPUT "Info rate in bps
    (300,600,1200)";IR
70 IF IR<>300 AND IR <>600 AND IR<>1200
    THEN 60
80 IM=IR*60
90 INPUT "Data word length (7 OR 8)";DW
100 IF DW<>7 AND DW<>8 THEN 90
110 INPUT "Parity (E/O/I/N)"; P$
120 IF P$="E" OR P$="O" OR P$="I" THEN
    PB=1: GOTO 150
```

```
130 IF P$="N" THEN PB=0: GOTO 150
140 GOTO 110
150 INPUT "Stop bits (1,2)"; SB
160 IF SB<>1 AND SB<>2 THEN 150
170 PRINT "Host computer's hourly rate":
     INPUT "(0-99.99)";HR
180 PRINT"Network surcharge per minute":
     INPUT"(0-9.99)";NM
190 IF NM<0 OR NM>9.99 THEN 180
200 IF HR<0 OR HR>99.99 THEN 170
210 HM=HR/60
220 PRINT "Phone system's minute rate":
     INPUT "(0-9.99)";TM
230 IF TM<0 OR TM>9.99 THEN 220
240 T=-INT(-L*(1+DW+PB+SB)/IM)
250 C=T*(HM+TM+NM)
260 PRINT "The file transfer will take
     at least"
270 PRINT USING "## minutes and cost at
     least $$###.##."; T;C
```

TERM.BA

```
10 'TERM.BA: Terminal Program
20 'Copyright 1983 by Tandy Corp. All
     rights reserved.
30 MAXFILES=3: CLEAR 256,HIMEM-14: DEFINT
     A-Z
40 DIM KD$(3,8),DX$(1),PX$(1),QI
50 AD!=HIMEM: FOR I=1 TO 13: READ B: POKE
     AD!+I,B: NEXT I: RD!=HIMEM+1
60 DATA 235,205,109,109,62,255,18,
     200,205,126,109,18,201
```

```
70 FOR S=1 TO 3: FOR K=1 TO 8: READ
     KD$(S,K): NEXT K,S
80 DATA File,Down," Up",,, " ^C",Time,Bye
90 DATA File,X-Dn,Mem,,," ^C",Time,Bye
100 DATA File,,X-Up,,,,Time,Bye
110 DW$="7": PY$="I": SB$="1":
     PH$="555-1111<>"
120 DX$(0)="Full": DX$(1)="Half":
     PX$(0)="": PX$(1)="Echo": BR$=CHR$(3)
130 DX=0: PX=0
140 PR$="Word length": RG$="7/8": V$=DW$:
     GOSUB 180: DW$=V$
150 PR$="Parity": RG$="E/O/I/N": V$=PY$:
     GOSUB 180: PY$=V$
160 PR$="Stop bits": RG$="1/2": V$=SB$:
     GOSUB 180: SB$=V$
170 CW$="MDM:"+DW$+PY$+SB$+"E": GOTO 210
180 CLS: PRINT@80,PR$"="V$: PRINT"Press
     ENTER to leave as-is or": PRINT"ENTER
     a new value ("RG$")";
190 V1$=V$: INPUT V1$: IF INSTR(1,RG$,V1$)
     MOD 2=0 THEN BEEP: GOTO180
200 V$=V1$: RETURN
210 CLS: PRINT@80,"Phone number to call is
     "PH$: PRINT"Press ENTER to leave as-is
     or": PRINT"Enter a new number? ";:
     LINE INPUT P1$: IF P1$>""THEN PH$=P1$
220 IF INSTR(1,PH$,"<")=0 THEN PH$=PH$+"<>"
230 M!=VARPTR(PH$):
     CD!=PEEK(M!+1)+PEEK(M!+2)*256
240 PRINT"Calling host computer..."
250 CALL 21200: CALL 21293,0,CD!
260 OPEN CW$ FOR OUTPUT AS 1: OPEN CW$ FOR
     INPUT AS 2
270 BEEP: S=1: GOSUB 660
```

```
280 CALL RD!,0,VARPTR(QI): IF QI=255 THEN
     320
290 CI$=CHR$(QI): PRINTCI$;
300 IF S=2 THEN PRINT#3,CI$;
310 IF PX=1 THEN ON ERROR GOTO 410: LPRINT
     CI$;: ON ERROR GOTO 0
320 IF S=3 THEN 350
330 IF BR=1 THEN CO$=BR$: BR=0: GOTO 360
340 CO$=INKEY$: IF CO$=""THEN 280ELSE 360
350 IF EOF(3) THEN GOSUB 650: GOTO 280ELSE
     CO$=INPUT$(1,3)
360 PRINT#1,CO$;
370 IF DX=1 AND PX=1 THEN ON ERROR GOTO
     410: LPRINT CO$;: ON ERROR GOTO 0
380 IF DX=0 THEN 280
390 PRINTCO$;: IF S=2 THEN PRINT#3,CO$;
400 GOTO 280
410 BEEP: PRINT: PRINT"Echo function
     aborted.": GOSUB 580: RESUME NEXT
420 PRINT: FILES: RETURN
430 PRINT: LINE INPUT"Download file name?
     ";FO$
440 IF INSTR(1,FO$,"CAS:")>0 THEN A$="N"
     ELSE INPUT "Append or New file
     (A/N)";A$
450 ON ERROR GOTO 510
460 IF A$="A" OR A$="a" THEN 480
470 OPEN FO$ FOR OUTPUT AS 3: GOTO 490
480 OPEN FO$ FOR APPEND AS 3
490 ON ERROR GOTO 0: S=2: GOSUB 660
500 RETURN
510 BEEP: PRINT"Aborted": RESUME 500
520 PRINT: LINE INPUT"Upload file name?
     ";FI$
```

```
530 ON ERROR GOTO 560: OPEN FI$ FOR INPUT
     AS 3: ON ERROR GOTO 0
540 S=3: GOSUB 660
550 RETURN
560 BEEP: PRINT"Aborted": RESUME 550
570 DX=1 XOR DX: KEY 4,DX$(DX): SCREEN 0,1:
     RETURN
580 PX=1 XOR PX: KEY 5,PX$(PX): SCREEN 0,1:
     RETURN
590 BR=1: RETURN
600 PRINT: PRINTTIME$: RETURN
610 PRINT: PRINT"Disconnect (Y/N)?";:
     A$=INPUT$(1): PRINT: IF A$<>"Y"AND
     A$<>"y" THEN RETURN
620 GOTO 820
630 PRINT: IF INSTR(1,FO$,"CAS:")>0 THEN
     PRINT"Function not available for CAS:
     files": RETURN
640 CLOSE 3: PRINT"Free memory="FRE(0):
     OPEN FO$ FOR APPEND AS 3: RETURN
650 CLOSE 3: S=1: GOSUB 660: RETURN
660 SCREEN 0,0
670 FOR K=1 TO 8: KEY(K) OFF
680 IF K=4 THEN KEY K,DX$(DX): GOTO 710
690 IF K=5 THEN KEY K,PX$(PX): GOTO 710
700 KEY K,KD$(S,K)
710 NEXT K
720 ON S GOSUB 760,780,800
730 FOR K1=1 TO 8: KEY(K1) ON: NEXT K1
740 SCREEN 0,1
750 RETURN
760 ON KEY GOSUB
     420,430,520,570,580,590,600,610
770 RETURN
```

```
780 ON KEY GOSUB
     420,650,630,570,580,590,600,610
790 RETURN
800 ON KEY GOSUB 420,,650,570,580,,600,610
810 RETURN
820 CLOSE: CALL 23164,0,23366: CALL 27795:
     CLEAR 256,HIMEM+14: END
```

QUOTE.BA

```
10 'QUOTE.BA: Automatic Data Retrieval
     Program
20 'Copyright 1983 by Tandy Corp. All
     rights reserved.
30 MAXFILES=3: CLEAR 256,HIMEM-14: DEFINT
     A-Z: DIM QQ
40 AD!=HIMEM: FOR I=1 TO 13: READ B: POKE
     AD!+I,B: NEXT I: RC!=HIMEM+1
50 READ PH$
60 M!=VARPTR(PH$):
     CD!=PEEK(M!+1)+PEEK(M!+2)*256
70 RP$(0)="The call did not go through."
80 RP$(1)="Unable to log-on."
90 RP$(2)="The call was completed
     normally."
100 RP$(3)="Call stopped after reaching
     time limit."
110 RP$(4)="The call was terminated by "
120 CLS
130 TF=0: INPUT "Call at what time
     (hh:mm:ss)";T1$: IF T1$=""THEN 200
140 ON ERROR GOTO 180: ON TIME$=T1$ GOSUB
     190: TIME$ ON: ON ERROR GOTO 0
```

```
150 INPUT"Time limit for call (1-60
    minutes)";TL: IF TL<1 OR TL>60 THEN
    150
160 CLS: PRINT@80,"Call-alarm is set for
    "T1$
170 IF TF=1 THEN 200ELSE PRINT@160,"The
    time is "TIME$:  GOTO 170
180 BEEP: PRINT"Invalid time format":
    RESUME 130
190 TF=1: TIME$ OFF: RETURN
200 A$=TIME$: AH!=VAL(MID$(A$,1,2))*3600:
    AM=(TL+VAL(MID$(A$,4,2)))*60:
    AS=VAL(MID$(A$,7,2)): AT!=AH!+AM+AS
210 AH=INT(AT!/3600): A1=AT!-AH*3600:
    AM=A1\60: AS=A1-AM*60
220 A=AH MOD 24: GOSUB 270: AH$=A$
230 A=AM: GOSUB 270: AM$=A$
240 A=AS: GOSUB 270: AS$=A$
250 T2$=AH$+":"+AM$+":"+AS$
260 GOTO 280
270 A1$=STR$(A):
    A$=RIGHT$("0"+RIGHT$(A1$,LEN(A1$)-1),2)
    : RETURN
280 CLS: CC=0: ON ERROR GOTO 610
290 ON TIME$=T2$ GOSUB 620: TIME$ ON
300 T1$=TIME$: PRINT: PRINT"Calling host
    now.": CALL 21200
310 CALL 21293,0,CD!
320 CC=1: PRINT: PRINT"Host computer is
    on-line.": PRINT"Logging-on and
    requesting data..."
330 OPEN "quotes" FOR OUTPUT AS 3
340 PRINT #3,DATE$" "TIME$
350 OPEN "MDM:7I1E" FOR INPUT AS 1: OPEN
    "MDM:7I1E" FOR OUTPUT AS 2
```

```
360 CC=2
370 READ NL: L=1
380 READ CM$,ST$,SC: GOSUB 490: IF CC=3
     THEN 400
390 L=L+1: IF L<=NL THEN 380
400 CLOSE: CALL 21179
410 PRINT: PRINT"On at "T1$". Off at "TIME$
420 PRINTRP$(CC)
430 IF CC=4 THEN PRINT "error number"ER"in
     line"EL
440 CLEAR 0,HIMEM+14: END
450 SP=1
460 P=INSTR(SP,TX$,"^"): IF P=0 THEN RETURN
470 SP=P+1: C$=MID$(TX$,SP,1): IF
     C$>="@"AND C$<"\" THEN
     TX$=LEFT$(TX$,P-1)+CHR$(ASC(C$)-64)+RIG
     HT$(TX$,LEN(TX$)-P-1)
480 GOTO 460
490 TX$=CM$: GOSUB 450: CM$=TX$
500 TX$=ST$: GOSUB 450: ST$=TX$
510 PRINT#2,CM$;
520 P=1
530 SC$=MID$(ST$,P,1)
540 IF CC=3 THEN RETURN ELSE CALL
     RC!,0,VARPTR(QQ): IF QQ=255 THEN 540
550 CC$=CHR$(QQ): IF SC=1 AND CC=2 THEN
     PRINT#3,CC$;
560 PRINT CC$;
570 IF SC$<>CC$THEN 520
580 P=P+1: IF P<=LEN(ST$)THEN 530
590 CALL RC!,0,VARPTR(QQ): IF QQ<255 THEN
     PRINTCHR$(QQ);: GOTO 590
600 RETURN
610 CC=4: ER=ERR: EL=ERL: RESUME 400
620 CC=3: RETURN
```

```
630 REM -- machine code subroutine
640 DATA 235,205,109,109,62,255,18,200,
     205,126,109,18,201
650 REM -- phone number
660 DATA 555-1111<>
670 REM -- number of dialog triplets
680 DATA 14
690 REM -- dialog triplets including
     log-on/log-off
700 DATA ^C,U,0
710 DATA "70000,999^M",P,0
720 DATA PASSWORD^M,"!",0
730 DATA G FIN-6^M,":",0
740 DATA QUOTES^M,":",0
750 DATA ^M,"^JI",0
760 DATA XON^M,^JI,0
770 DATA T^M,^JI,0
780 DATA BUD^M,^JI,0
790 DATA TAN^M,^JI,0
800 DATA IBM^M,^JI,0
810 DATA ^M,!,0
820 DATA "1,2,3,4,5^M",PROGRAM,1
830 DATA /OFF^M,Off,0
```

DB.BA

```
10 'DB.BA: Database Program
20 'Copyright 1983 by Tandy Corp. All
     rights reserved.
30 PRINT"<U>pdate or <A>dd records?":
     R$=INPUT$(1):
     ON INSTR(1,"AaUu",R$)/2+.5
      GOTO 50, 60
40 GOTO 30
```

```
50 MAXFILES=1: GOTO 70
60 MAXFILES=2
70 CLEAR 1000: DEFSNG A-C,E-Z: DEFDBL D
80 ON ERROR GOTO 1390
90 VL=8: VW=40
100 VU=VL-2: LZ=VU*VW
110 BK$=CHR$(8): CR$=CHR$(13):
     ES$=CHR$(27): LF$=CHR$(29):
     RT$=CHR$(28): UP$=CHR$(30):
     DN$=CHR$(31): SP$=CHR$(32):
     DE$=CHR$(127): QT$=CHR$(34)
120 TB$=BK$+CR$+LF$+RT$+UP$+DN$+ES$
130 LB$=ES$+"Y  <D>ata entry"+ES$+"Y!
     <F>ile"+ES$+"Y"+CHR$(34)+" <G>et a
     record from the input file"+ES$+"Y#
     <S>earch input file"+ES$+"Y$ <E>rase
     current record"+ES$+"Y% <Q>uit"
140 CS$=ES$+"P"
150 SK$=ES$+"W": SL$=ES$+"V"
160 MT$="DdFfGgSsEeQq": YN$="YyNn"
170 DI$=STRING$(VW,SP$)
180 VH$=ES$+"p":  VN$=ES$+"q"
190 KY$=VH$+"ENTER"+VN$+"=data
     "+VH$+CHR$(155)+VN$+SP$+VH$+CHR$(154)+V
     N$+SP$+VH$+CHR$(153)+VN$+SP$+VH$+CHR$(1
     52)+VN$+"=move
     "+VH$+"ESC"+VN$+"=menu"
200 EL$=ES$+"K"
210 FILES: LINE INPUT "Name of format file?
     "; FF$
220 OPEN FF$ FOR INPUT AS 1
230 INPUT #1, NL
240 DIM L$(NL)
250 FOR J=1 TO NL: LINE INPUT #1, L$(J):
     NEXT J
```

```
260 INPUT #1, NF
270 DIM LL(NF),LC(NF),T$(NF),PC$(NF),
    D(NF),D$(NF),DV(NF)
280 FOR F=1 TO NF
290 INPUT #1, LL(F),LC(F),DS$
300 T$(F)=LEFT$(DS$,1)
310 IF T$(F)="A" THEN 320ELSE 350
320 LF=VAL(RIGHT$(DS$,LEN(DS$)-1))
330 IF LF=1 THEN PC$(F)="!": GOTO 360
340 IF LF>1 AND LF<=VW THEN
    PC$(F)="\"+STRING$(LF-2,32)+"\": GOTO
    360: ELSE BEEP: PRINT"Field exceeds
    display width": GOTO 970
350 PC$(F)=RIGHT$(DS$,LEN(DS$)-1)
360 NEXT F
370 CLOSE #1: ON ERROR GOTO 1400
380 IF MAXFILES=2 THEN LINE INPUT"Name of
    input file? ";GF$: OPEN GF$ FOR INPUT
    AS 2
390 LINE INPUT "Name of output file? ";OF$:
    IF INSTR(1,OF$,"CAS:")>0 THEN 420
400 PRINT"<D>estroy old output file (if
    any) or": PRINT"<A>ppend?":
    DA$=INPUT$(1): PRINT: ON
    INSTR(1,"DdAa",DA$)/2+.5 GOTO 420,430
410 GOTO 400
420 OPEN OF$ FOR OUTPUT AS 1: GOTO 440
430 OPEN OF$ FOR APPEND AS 1
440 SCREEN 0,0: FOR I=1 TO 8: KEY I,"":
    NEXT I
450 PRINTSL$;
460 PG=(NL-1)\VU+1: PL=(NL-1)MOD VU+1
470 GOSUB 1370
480 FC=1
490 GOSUB 1360
```

```
500 CLS: PRINT CS$;
510 IF PC=PG THEN PS=PL ELSE PS=VU
520 FOR I=1 TO PS: PRINTL$((PC-1)*VU+I):
     NEXT I
530 TT=(PC-1)*VU+1: TB=TT+PS-1
540 FOR I=1 TO NF
550 IF LL(I)>=TT AND LL(I)<=TB THEN FD=I:
     GOSUB 1280
560 NEXT I
570 GOSUB 1340: FD=FC: GOSUB 1280: GOSUB
     1350
580 SCREEN 0,0: PRINT@LZ+VW+2,KY$;:
     PRINT@LZ,EL$;
590 DP=0
600 SOUND 2400,1
610 K$=INKEY$: IF K$=""OR K$=QT$THEN610
620 IF K$>=SP$ AND K$<DE$ THEN 650
630 ON INSTR(1,TB$,K$) GOTO
     660,670,700,720,760,780,850
640 GOTO 610
650 IF DP=VW THEN 600ELSE PRINTK$;:
     DP=DP+1: MID$(DI$,DP)=K$: GOTO 610
660 IF DP=0 THEN 600ELSE PRINT DE$;:
     DP=DP-1: GOTO 610
670 DV(FC)=1: IF T$(FC)="N" THEN 690
680 D$(FC)=LEFT$(DI$,DP): GOTO 720
690 D(FC)=VAL(LEFT$(DI$,DP)): GOTO 720
700 FO=FC: FC=FC-1: IF FC<1 THEN FC=1: GOTO
     600
710 GOTO 730
720 FO=FC: FC=FC+1: IF FC>NF THEN FC=NF:
     GOTO 570
730 PO=PC: GOSUB 1360
740 IF PC<>PO THEN 500
750 FD=FO: GOSUB 1280: GOTO 570
```

```
760 PO=PC: PC=PC-1: IF PC<1 THEN PC=1: GOTO
    600
770 GOTO 790
780 PO=PC: PC=PC+1: IF PC>PG THEN PC=PG:
    GOTO 600
790 IF PC=PG THEN PS=PC+PL ELSE PS=VU
800 TT=(PC-1)*VU+1: TB=TT+PS-1: FF=0
810 FOR I=1 TO NF
820 IF LL(I)>=TT AND LL(I)<=TB THEN FF=I:
    I=NF
830 NEXT I
840 IF FF=0 THEN STOP ELSE FC=FF: GOTO 500
850 CLS: PRINTLB$
860 PRINTEL$;
870 R$=INPUT$(1): ON INSTR(1,MT$,R$)/2+.5
    GOTO 500,890,980,1040,960,970
880 GOTO 870
890 FOR I=1 TO NF
900 IF T$(I)="N" THEN 920
910 PRINT#1,QT$D$(I)QT$;: GOTO 930
920 PRINT #1,D(I);
930 IF I<NF THENPRINT#1,",";
940 NEXT I: PRINT#1,
950 CLS: PRINT@LZ,"Record filed.": GOSUB
    1380: GOSUB 1370: GOTO 850
960 GOSUB 1370: GOTO 480
970 CLOSE: PRINTSK$;: CALL 23164,0,23366:
    CALL 27795: END
980 IF MAXFILES=1 THEN 1270
990 EF=0: FOR I=1 TO NF: IF EOF(2)THEN
    EF=1: CLS: PRINT@LZ,"End of file."EL$:
    I=NF: GOSUB 1380: GOTO 1030
1000 IF T$(I)="N" THEN 1010ELSE INPUT
    #2,D$(I): GOTO 1020
1010 INPUT#2,DY$: D(I)=VAL(DY$)
```

```
1020 DV(I)=1
1030 NEXT I: IF EF=1 THEN 850ELSE 480
1040 IF MAXFILES=1 THEN 1270
1050 PRINT@LZ+VW,"Search hi-lited field
     only. OK? (Y/N)"EL$;: R$=INPUT$(1)
1060 ON INSTR(1,YN$,R$)/2+.5 GOTO 1080,500
1070 GOTO 1050
1080 MF=0: EF=0: PRINT@LZ+VW,"Enter the
     search value: "EL$;
1090 IF T$(FC)="N" THEN 1100ELSE INPUT SV$:
     GOTO 1110
1100 INPUT SV
1110 PRINT@LZ+VW,"Start at 1st or next
     record? (F/N)"EL$;: R$=INPUT$(1)
1120 ON INSTR(1,"FfNn",R$)/2+.5GOTO
     1140,1150
1130 GOTO 1110
1140 CLOSE #2: OPEN GF$ FOR INPUT AS 2
1150 PRINT@LZ+VW,"Searching"EL$;: FOR I=1
     TO NF: IF EOF(2) THEN EF=1: CLS:
     PRINT@LZ,"Stopped at end of file."EL$:
     I=NF: GOSUB 1380: GOSUB 1370: GOTO
     1240
1160 D$(I)="": D(I)=0
1170 IF T$(I)="N" THEN 1180ELSE INPUT
     #2,D$(I): GOTO 1190
1180 INPUT #2,DY$: D(I)=VAL(DY$)
1190 DV(I)=1: IF FC<>I THEN 1240
1200 IF T$(FC)="N" THEN 1230
1210 IF LEFT$(D$(FC),LEN(SV$))=SV$ THEN
     MF=1
1220 GOTO 1240
1230 IF D(I)=SV THEN MF=1
1240 NEXT I
1250 IF EF=1 THEN850ELSE IF MF=1 THEN 500
```

```
1260 GOTO 1150
1270 CLS: PRINT@LZ,"Input file not
     requested at startup."EL$: GOSUB 1380:
     GOTO 850
1280 PZ=((LL(FD)-1)MOD VU)*VW+LC(FD)-1
1290 IF DV(FD)<>0 THEN IF T$(FD)="A" THEN
     1310ELSE 1320
1300 PRINT@ PZ, STRING$(LEN(PC$(FD)),32);:
     GOTO 1330
1310 PRINT@ PZ, USING PC$(FD);D$(FD);: GOTO
     1330
1320 PRINT@ PZ, USING PC$(FD);D(FD);
1330 RETURN
1340 PRINT VH$;: RETURN
1350 PRINT VN$;: RETURN
1360 PC=(LL(FC)-1)\VU+1: RETURN
1370 FOR I=1 TO NF: D$(I)="": D(I)=0:
     DV(I)=0: NEXT I: RETURN
1380 BEEP: INPUT "Press ENTER to
     continue";EN$: RETURN
1390 BEEP: PRINT"Problem in format file."
1400 PRINT"Program stopped due to
     error"ERR: PRINT"at line "ERL;
1410 RESUME 970
```

SORT.BA

```
10 'SORT.BA: Sorting Program
20 'Copyright 1983 by Tandy Corp. All
     rights reserved.
30 MAXFILES=1: CLEAR 0: CLEAR FRE(0)-1000:
     DEFINT A-Z: CM$=",": QT$=CHR$(34):
     VW=40
```

```
40 FILES: LINEINPUT"Name of input file?
     ";FI$: LINEINPUT"Name of output file?
     ";FO$: IF FI$=""THEN END ELSE IF
     FO$=""THEN FO$="LCD:"
50 ONERROR GOTO 60: OPEN FI$ FOR INPUT AS
     1: ONERROR GOTO 0 :GOTO 70
60 BEEP: PRINT"Can't open "FI$" due to
     error"ERR: RESUME 40
70 IF EOF(1) THEN BEEP: PRINT"Empty file.":
     GOTO 40
80 LINEINPUT#1,SL$: IF LEN(SL$)=255THEN940
90 PRINT"Here is a sample record:"
100 PRINTSL$
110 NL=1
120 IF NOT EOF(1) THEN LINEINPUT#1,D$: IF
     LEN(D$)=255 THEN 940ELSE NL=NL+1: GOTO
     120
130 CLOSE 1: IF NL=1 THEN BEEP: PRINT
     "Aborted.": PRINT"Input file contains
     only 1 record.": END
140 DIM DB$(NL),SQ(NL): FOR I=1 TO NL:
     SQ(I)=I: NEXT I
150 PRINT"Fixed- or variable-length fields
     (F/V)?";: F$=INPUT$(1): PRINT
160 FT=INSTR(1,"FfVv",F$)/2+.5: ON FT GOTO
     360,180
170 GOTO 150
180 PRINT"Analyzing sample record": NF=1:
     QF=0: I=0
190 I=I+1: IF I>LEN(SL$) THEN260
200 C$=MID$(SL$,I,1): ON
     INSTR(1,CM$+QT$,C$) GOTO 220,240
210 GOTO 190
220 IF QF=0 THEN NF=NF+1
230 GOTO 190
```

```
240 QF=QF XOR 1
250 GOTO 190
260 PRINT"Record contains"NF"fields."
270 IF NF=1 THEN SF=1: GOTO 530
280 PRINT"Fields are"
290 OPEN FI$ FOR INPUT AS 1
300 FOR I=1 TO NF: INPUT#1,F$: PRINT I"-
    "QT$F$QT$: NEXT I: CLOSE
310 PRINT"Agreed (Y/N)?";: YN$=INPUT$(1):
    PRINT: ON INSTR(1,"YyNn",YN$)/2+.5
    GOTO 340,330
320 GOTO 310
330 PRINT"Can't sort file. Check record
    format.": END
340 PRINT"Sort on which field (0 to
    review)";: INPUT SF: IF SF<1 OR SF>NF
    THEN 280
350 GOTO 530
360 CLS: PRINT"Find starting and ending
    column": PRINT"of sort field:"
370 NS=-INT(-LEN(SL$)/VW):
    LS=LEN(SL$)-(NS-1)*VW
380 S=1: LC=VW
390 IF S=NS THEN LC=LS
400 FOR I=1 TO LC: TC=I+(S-1)*VW
410 IF I MOD 10=0 THEN
    PRINTCHR$(TC/10+48);: GOTO 440
420 IF I MOD 5=0 THEN PRINT".";: GOTO 440
430 PRINT"-";
440 NEXT I
450 IF S=NS AND SL<VW THEN PRINT
460 FOR I=1 TO LC: TC=I+(S-1)*VW:
    PRINTMID$(SL$,TC,1);: NEXT I
```

```
470 IF S<NS THEN S=S+1: PRINT: INPUT"Press
    ENTER to view next line";DY$: CLS:
    PRINT: GOTO 390
480 IF LS<VW THEN PRINT
490 PRINT: PRINT"Enter starting and ending
    columns of": PRINT"sort field (0,0 to
    review sample data)": INPUT SC,EC
500 IF SC<1 OR SC>EC OR EC>LEN(SL$) THEN360
510 PRINT"Here is the field you selected:":
    PRINTMID$(SL$,SC,EC-SC+1)
520 PRINT"Ok (Y/N)?";: YN$=INPUT$(1):
    PRINT: IF YN$<>"Y"AND YN$<>"y"THEN 360
530 PRINT"What type of data is in that
    field": PRINT"<A>lpha   <N>umeric
    <D>ate mm/dd/yy": PRINT"Select one:
    A/N/D";
540 DY$=INPUT$(1): PRINT:
    DT=INSTR(1,"AaNnDd",DY$)/2+.5: IF DT=0
    THEN BEEP: GOTO 530
550 IF DT=2 THEN DIM DB!(NL)
560 IF FT=2 THEN630
570 OPEN FI$ FOR INPUT AS 1
580 FOR L=1 TO NL
590 IF EOF(1) THEN DY$="" ELSE LINEINPUT
    #1,DY$: DY$=MID$(DY$,SC,EC-SC+1)
600 ON DT GOSUB 820,830,840
610 NEXT L: CLOSE
620 GOTO 680
630 OPEN FI$ FOR INPUT AS 1
640 FOR L=1 TO NL: FOR F=1 TO NF
650 IF EOF(1) THEN DY$="" ELSE INPUT #1,DY$
660 IF F=SF THEN ON DT GOSUB 820,830,840
670 NEXT F,L: CLOSE
680 PRINT "Sorting the fields now"
690 SW=0: FOR I=1 TO NL-1
```

```
700 IF DT=2 THEN 720
710 IF DB$(SQ(I))>DB$(SQ(I+1)) THEN 730
    ELSE 740
720 IF DB!(SQ(I))<=DB!(SQ(I+1))THEN740
730 Q=SQ(I): SQ(I)=SQ(I+1): SQ(I+1)=Q: SW=1
740 NEXT I: IF SW>0 THEN 690
750 PRINT"Sort completed. Writing data to":
    PRINT"the output file."
760 OPEN FI$ FOR INPUT AS 1
770 FOR I=1 TO NL: LINEINPUT#1,DB$(I): NEXT
    I
780 CLOSE: ONERROR GOTO 790: OPEN FO$ FOR
    OUTPUT AS 1: GOTO 800
790 BEEP: PRINT"Can't open "FO$" due to
    error"ERR: ON ERROR GOTO 0
800 FOR I=1 TO NL: PRINT#1, DB$(SQ(I)):
    NEXT I
810 CLOSE: PRINT"Sort completed.": END
820 DB$(L)=DY$: RETURN
830 DB!(L)=VAL(DY$): RETURN
840 T1$="000000"
850 T1=INSTR(1,DY$,"/"): IF T1<2 OR
    T1>3THEN 920
860 T2=INSTR(T1+1,DY$,"/"): IF T2<4 OR T2>6
    THEN 920
870 TM=VAL(DY$):
    TD=VAL(MID$(DY$,T1+1,T2-T1)):
    TY=VAL(RIGHT$(DY$,LEN(DY$)-T2))
880 IF TM<1 OR TM>12 OR TD<1 OR TD>31 OR
    TY<0 OR TY>99 THEN 920
890 T5$=STR$(TY): GOSUB 930: T1$=T5$
900 T5$=STR$(TM): GOSUB 930: T1$=T1$+T5$
910 T5$=STR$(TD): GOSUB 930: T1$=T1$+T5$
920 DB$(L)=T1$: RETURN
```

```
930 T5$= RIGHT$("0"+RIGHT$(T5$,
    LEN(T5$)-1),2): RETURN
940 BEEP: PRINT"Aborted. One of the records
    in "FI$: PRINT"exceeds 254 characters
    in length": END
```

XTRACT.BA

```
10 'XTRACT.BA: Data Extraction Program
20 'Copyright 1983 by Tandy Corp. All
    rights reserved.
30 MAXFILES=2: CLEAR 500
40 FILES: LINEINPUT"Name of input file?
    ";FI$: LINEINPUT"Name of output file?
    ";FO$: IF FI$=""THEN END ELSE IF
    FO$=""THEN FO$="LCD:"
50 ONERROR GOTO 60: OPEN FI$ FOR INPUT AS
    1: OPEN FO$ FOR OUTPUT AS 2: ONERROR
    GOTO 0     : GOTO 70
60 BEEP: PRINT"Can't open file due to
    error"ERR: RESUME 40
70 LINE INPUT "Enter character(s) to search
    for ";C$: IF C$=""THEN CLOSE: END
80 SC=0: EC=0: PRINT"Enter starting and
    ending columns": PRINT"for search
    (0=entire record)";: INPUT SC,EC
90 IF SC=0 AND EC=0 THEN SC=1: EC=255: GOTO
    110
100 IF SC<1 OR SC>EC OR EC>255 THEN BEEP:
    PRINT"Error in column specification":
    GOTO 80
110 IF EOF(1) THEN PRINT"End of file.":
    CLOSE: END
120 LINE INPUT #1,TX$
```

```
130 IF TX$="" THEN 110
140 ST$=MID$(TX$,SC,EC-SC+1)
150 IF INSTR(1,ST$,C$)=0 THEN 110
160 PRINT#2,TX$;: IF LEN(TX$)<255 THEN
      PRINT#2,
170 GOTO 110
```

ENTRY.BA

```
10 'ENTRY.BA: Data Entry Program
20 'Copyright 1983 by Tandy Corp. All
      rights reserved.
30 MAXFILES=1: CLEAR 256: DEFINT A-Z
40 ES$=CHR$(27): EF$=ES$+"J": CR$=CHR$(13):
      NL$=CR$+CHR$(10): VH$=ES$+"p":
      VN$=ES$+"q": HI=4697: LO=5272:
      QT$=CHR$(34): CM$=","
50 DIM N(12)
60 FOR I=1 TO 12: READ N(I): NEXT I
70 DATA 3134,2959,2793,2636,2484,
      2348,2216,2092,1975,1864,1758,1660
80 CLS: FILES: LINE INPUT "Name of output
      file? ";FO$
90 PRINT"<A>ppend to or <D>estroy
      existing": PRINT"file (if any) ";
100 AD$=INPUT$(1): ON
      INSTR(1,"AaDd",AD$)/2+.5 GOTO 120,130
110 BEEP: GOTO 100
120 OPEN FO$ FOR APPEND AS 1: GOTO 140
130 OPEN FO$ FOR OUTPUT AS 1
140 PRINT: INPUT "How many items per record
      (1-8)";NI: IF NI<1 OR NI>8 THEN 140
150 DIM DT(NI),DP$(NI),RG#(NI,2),D$(NI)
160 FOR I=1 TO NI
```

```
170 PRINT: DP$(I)="Item"+STR$(I):
     PRINT"Name of item"I"(10 characters
     max)": INPUT DP$(I):
     DP$(I)=LEFT$(DP$(I),10)
180 PRINT"Data type of '"DP$(I)"'";
     PRINT"<A>lpha or <N>umeric ";
190 AN$=INPUT$(1):
     DT=INSTR(1,"AaNn",AN$)/2+.5: IF DT=0
     THEN BEEP: GOTO 190ELSE DT(I)=DT
200 IF DT(I)=1 THEN 260
210 PRINT: PRINT"Enter minimum and maximum
     values"NL$"for '"DP$(I)"' (ENTER=no
     limit)."
220 MI$="-1E62": INPUT "Minimum value";MI$
230 RG#(I,1)=VAL(MI$)
240 PRINT: MA$="1E62": INPUT "Maximum
     value";MA$
250 RG#(I,2)=VAL(MA$)
260 NEXT I
270 CLS: FOR I=1 TO NI
280 PRINT@160,EF$"Enter data for '"DP$(I)"'
     "VH$"ENTER"VN$"=quit"
290 IF DT(I)=1 THEN 310
300 PRINT"Min="RG#(I,1)
     TAB(20)"Max="RG#(I,2)
310 SOUND 2500,1: D$="": INPUT D$: IF D$=""
     THEN BEEP: CLOSE: END
320 IF DT(I)=1 THEN D$(I)=D$: GOTO 390
330 D#=VAL(D$): IF D#>=RG#(I,1) AND
     D#<=RG#(I,2) THEN 380
340 SOUND HI,8: SOUND LO,8
350 PRINT"Out of range. <A>ccept or
     <C>ancel? ";
360 AC$=INPUT$(1): ON
     INSTR(1,"AaCc",AC$)/2+.5 GOTO 380,280
```

```
370 GOTO 360
380 D$(I)=STR$(D#)
390 FOR J=1 TO I: JC=(J-1)\4:
     JR=(J-1) MOD 4
400 PRINT@JR*40+JC*20, VH$ DP$(J) VN$D$(J);
410 NEXT J
420 NEXT I
430 FOR N=1 TO 5: SOUND N(N),1: NEXT N
440 PRINT@160,EF$"<S>tore <R>edo "
450 SR$=INPUT$(1):
     ON INSTR(1,CR$+CR$+"SsRr",SR$)/2+.5
     GOTO 470,470,270
460 GOTO 450
470 FOR I=1 TO NI
480 IF DT(I)=1 THEN PRINT#1,QT$;
490 PRINT#1,D$(I);
500 IF DT(I)=1 THEN PRINT#1,QT$;
510 IF I<NI THEN PRINT#1,CM$;
520 NEXT I
530 PRINT#1,
540 GOTO 270
550 CLOSE: END
```

CALC.BA

```
10 'CALC.BA: Calculator Program
20 'Copyright 1983 by Tandy Corp. All
     rights reserved.
30 MAXFILES=0: CLEAR 256: DEFINT A-U,X-Z:
     DEFDBL V
40 T=30: DIM VL(T),V(6)
50 FOR I=1 TO 6: V(I)=0: NEXT I
60 SG$="+-": DG$="0123456789": RN$=DG$+".":
     VA$="ABCDER": VB$="AaBbCcDdEeRr"
```

```
70 OP$="^*/\@+-": PR$="1223344": LP$="(":
     RP$=")": BF$=STRING$(16,32)
80 WV=40: Z4=WV*3: Z5=WV*4: Z7=WV*6:
     Z8=WV*7
90 ES$=CHR$(27): EL$=ES$+"K": EF$=ES$+"J":
     VH$=ES$+"p": VN$=ES$+"q"
100 CLS
110 FOR I=1 TO 6: PRINT@
     WV/2*(I-1),MID$(VA$,I,1)"="CSNG(V(I))
     EL$;: NEXT I
120 PRINT@Z4,"Current formula:"; EL$: PRINT
     E1$
130 PRINT@Z8,"Select: <F>ormula <D>ata
     <C>alc <Q>uit ";
140 A$=INPUT$(1): ON
     INSTR(1,"FfDdCcQq",A$)/2+.5 GOTO
     160,190,250,830
150 GOTO 140
160 PRINT@Z4,"Recognized operators: "OP$; "
     and ( )";EF$;
170 PRINT@Z8,"Press Enter when done";EF$;:
     PRINT @Z5,"";
180 LINE INPUT E1$: E2$=E1$: GOTO 120
190 PRINT@Z7,EF$: PRINT "Change which
     variable ("VA$")? ";
200 A$=INPUT$(1): NV=INSTR(1,VB$,A$)/2+.5:
     IF NV=0 THEN 130
210 PV=WV/2*(NV-1)+2: PRINT@PV,VH$
     CSNG(V(NV)) VN$;
220 NV$=STR$(V(NV)): PRINT @Z7,EF$;: INPUT
     "Enter a new value"; NV$
230 V(NV)=VAL(NV$)
240 PRINT@PV,BF$;: PRINT @PV,CSNG(V(NV));:
     GOTO 190
```

```
250 PRINT@Z7,"Calculating . . .";EL$:
     LE=LEN(E1$)
260 LR=INSTR(1,E1$,RP$): IF LR=0 THEN
     E$=E1$: GOTO 340
270 IF LR<2 THEN 810
280 LL=0
290 FOR SP=LR-1 TO 1 STEP -1
300 C$=MID$(E1$,SP,1): IF C$=LP$ THEN
     LL=SP: SP=1
310 NEXT SP
320 IF LL=0 THEN 810
330 E$=MID$(E1$,LL+1,LR-LL-1)
340 L=LEN(E$)
350 P=1: EOT=0: TC=0: OC=0: PF=0
360 FOR I=1 TO T: VL(I)=0: NEXT I: O$="":
     TL$=""
370 TF=0: S$="+": I$="": F$="": GOSUB 840:
     IF EOT THEN 490
380 IF INSTR(1,SG$,C$)=0 THEN 400
390 S$=S$+C$: GOSUB 840: IF EOT THEN
     490ELSE 380
400 IF INSTR(1,RN$,C$)>0 THEN 440
410 IF INSTR(1,VB$,C$)>0 THEN 430
420 GOTO 490
430 I=INSTR(1,VB$,C$)/2+.5: VL=V(I): TF=-1:
     GOSUB 840: GOTO 500
440 IF INSTR(1,DG$,C$)>0 THEN 460
450 IF C$="." THEN 470ELSE 490
460 I$=I$+C$: TF=-1: GOSUB 840: IF EOT THEN
     490ELSE 440
470 GOSUB 840: IF EOT THEN 490
480 IF INSTR(1,DG$,C$)>0 THEN F$=F$+C$:
     TF=-1: GOTO 470
490 VL=VAL(I$+"."+F$)
500 SN=1: SF=1
```

```
510 SF=INSTR(SF,S$,"-"): IF SF>0 THEN
     SF=SF+1: SN=-SN: GOTO 510
520 VL=VL*SN
530 IF NOT TF THEN 810
540 TC=TC+1: IF TC>T THEN 820ELSE VL(TC)=VL
550 IF EOT THEN 580
560 IF INSTR(1,OP$,C$)=0 THEN 810
570 PF=-1: OC=OC+1: O$=O$+C$:
     TL$=TL$+MID$(PR$,INSTR(1,OP$,C$),1):
     GOTO 370
580 FOR TN=1 TO 4
590 SP=1
600 TF=INSTR(SP,TL$,CHR$(TN+48)): IF TF=0
     THEN 710
610 BP$=MID$(O$,TF,1): BF=INSTR(1,OP$,BP$)
620 ON BF GOSUB 740,750,760,770,780,790,800
630 IF TF+1=TC THEN 670
640 FOR TM=TF+1 TO TC-1
650 VL(TM)=VL(TM+1)
660 NEXT TM
670 TC=TC-1
680 TL$=LEFT$(TL$,TF-1)+ RIGHT$(TL$,OC-TF)
690 O$=LEFT$(O$,TF-1)+RIGHT$(O$,OC-TF)
700 OC=OC-1: SP=TF: IF SP<=OC THEN 600
710 NEXT TN
720 IF LR=0 THEN PRINT @Z7,"Result is:
     ";VL(1) EL$;:V(6)=VL(1):E1$=E2$:
     GOTO 110
730 E1$=LEFT$(E1$,LL-1)+
     STR$(VL(1))+RIGHT$(E1$,LE-LR):
     GOTO 250
740 VL(TF)=VL(TF)^VL(TF+1): RETURN
750 VL(TF)=VL(TF)*VL(TF+1): RETURN
760 VL(TF)=VL(TF)/VL(TF+1): RETURN
770 VL(TF)=VL(TF)\VL(TF+1): RETURN
```

```
780 VL(TF)=VL(TF)MOD VL(TF+1): RETURN
790 VL(TF)=VL(TF)+VL(TF+1): RETURN
800 VL(TF)=VL(TF)-VL(TF+1): RETURN
810 BEEP: PRINT @Z7,"Error: Check
    formula"EL$;: E1$=E2$: GOTO 110
820 BEEP:PRINT@Z7, "Error: Formula contains
    >"T"terms."EL$;: E1$=E2$: GOTO 110
830 MENU
840 IF P>L THEN EOT=-1: RETURN
850 C$=MID$(E$,P,1): P=P+1
860 IF C$=" " THEN 840ELSE RETURN
```

TCALC.BA

```
10 'TCALC.BA: Data File Calculator Program
20 'Copyright 1983 by Tandy Corp. All
    rights reserved.
30 MAXFILES=1:CLEAR 1000:DEFINT
    A-Z:CR$=CHR$(13):CM$=",":QT$=CHR$(34):N
    L$=CR$+CHR$(10)
40 FILES:LINE INPUT"Name of input file?
    ";FI$
50 OPEN FI$ FOR INPUT AS 1
60 IF EOF(1) THEN BEEP:PRINT"Empty
    file":CLOSE:END
70 LINE INPUT #1,SL$:CLOSE:IF LEN(SL$)=255
    THEN BEEP:PRINT"Record exceeds maximum
    length of 254."NL$"Check file
    format.":END
80 PRINT"Analyzing a sample
    row.":NF=1:QF=0:I=0
90 I=I+1:IF I>LEN(SL$) THEN160
100 C$=MID$(SL$,I,1):ON INSTR(1,CM$+QT$,C$)
    GOTO 120,140
```

```
110 GOTO 90
120 IF QF=0 THEN NF=NF+1
130 GOTO 90
140 QF=QF XOR 1
150 GOTO 90
160 DIM CF(NF):PRINT"The sample row
    contains"NF"columns."
170 PRINT"The sample column entries are"
180 OPEN FI$ FOR INPUT AS 1
190 FOR I=1 TO NF:INPUT#1,F$:PRINT I"-
    "QT$F$QT$:NEXT I:CLOSE
200 PRINT"Agreed
    (Y/N)?";:YN$=INPUT$(1):PRINT:ON
    INSTR(1,"YyNn",YN$)/2+.5 GOTO
    230,220
210 GOTO 200
220 PRINT"Can't continue. Check record
    format.":END
230 PRINT"Compute <R>ow or <C>olumn
    totals? ";
240 RC$=INPUT$(1):ON
    INSTR(1,"RrCc",RC$)/2+.5 GOTO
    280,260
250 BEEP:GOTO 240
260 PRINT:PRINT"Compute totals for which
    column"NL$"(0 to review)";:INPUT CN:IF
    CN<1 OR CN>NF THEN 170
270 GOTO 410
280 PRINT:FOR I=1 TO NF:PRINT"Include
    column"I"in the total";:INPUT
    YN$:CF(I)=INSTR(1,"Yy",YN$)/2+.5:NEXT
    I
290 PRINT:OPEN FI$ FOR INPUT AS 1
300 RN=0
310 T#=0:NI=0
320 IF EOF(1) THEN CLOSE: END
330 FOR I=1 TO NF
```

```
340 IF EOF(1) THEN 510
350 INPUT #1, D$:IF CF(I)=0 THEN 370
360 T#=T#+VAL(D$):NI=NI+1
370 NEXT I
380 RN=RN+1
390 PRINT"Total of row"RN"is"T#;NL$"Average
     of"NI"items is"T#/NF
400 GOTO 310
410 OPEN FI$ FOR INPUT AS 1
420 T#=0:NI=0
430 IF EOF(1) THEN 490
440 FOR I=1 TO NF:IF EOF(1) THEN 510
450 INPUT #1,D$
460 IF I=CN THEN T#=T#+VAL(D$):NI=NI+1
470 NEXT I
480 GOTO 430
490 PRINT"Total of
     column"CN"is"T#;NL$"Average
     of"NI"items is"T#/NI
500 CLOSE:END
510 BEEP:PRINT"Reached end of file in the
     middle"NL$"of a record.":CLOSE:END
```

Index

RADIO SHACK, A DIVISION OF TANDY CORPORATION

U.S.A.: FORT WORTH, TEXAS 76102
CANADA: BARRIE, ONTARIO L4M 4W5

TANDY CORPORATION

AUSTRALIA	BELGIUM	U. K.
91 KURRAJONG ROAD MOUNT DRUITT, N.S.W. 2770	PARC INDUSTRIEL DE NANINNE 5140 NANINNE	BILSTON ROAD WEDNESBURY WEST MIDLANDS WS10 7JN

Printed in U.S.A.