

MODEL 20 CASSETTE MEMORY SYSTEM (11223A INSTALLED IN MODEL 20)

HEWLETT-PACKARD CALCULATOR PRODUCTS DIVISION
P.O. Box 301, Loveland, Colorado 80537, Tel. (303) 667-5000
Rue du Bois-du-Lan 7, CH-1217 Meyrin 2, Geneva, Tel. (022) 41 54 00

CHAPTERS

CHAPTER 1. GENERAL INFORMATION

INTRODUCTION	1-1
Special Programs	1-1
Storage Capacity	1-1
Storing Information on Tape	1-1
Data Error Detection	1-1
EQUIPMENT SUPPLIED	1-1
INITIAL INSPECTION	1-2
INSTALLATION PROCEDURE	1-2
ROM Block Memory Usage	1-2

CHAPTER 2. CASSETTE CONTROL

INTRODUCTION	2-1
Key Mnemonics	2-1
Diagnostic Notes	2-1
Specifying Parameters	2-1
Select Code Specification	2-1
MARKING TAPE	2-1
The MARK TAPE Syntax	2-1
File Marking Format	2-2
Successive Mark-Tape Operations	2-2
RECORD AND LOAD STATEMENTS	2-2
Record Program or Data	2-2
Load Program or Data	2-2
Examples	2-3
POSITIONING THE TAPE	2-5
FIND FILE	2-5
Example	2-6
IDENTIFY FILE	2-6
Example	2-7
BACKSPACE	2-8
REWIND	2-8
SECURE PROGRAMS	2-8
HALTING CASSETTE OPERATION	2-8
Opening the Cassette Memory Door	2-8
Halting 'Local' Rewind	2-8

CHAPTER 3. OPERATING HINTS

TAPE STORAGE CAPACITY	3-1
Determining Program Size	3-1
RE-MARKING THE TAPE	3-2
MISCELLANEA	3-2
Master Recordings	3-2
Recording on a Protected Cassette	3-2
Clearing a File	3-2

APPENDIX

DIAGNOSTIC NOTES
KEY MNEMONICS

Chapter 1

GENERAL INFORMATION

INTRODUCTION

The -hp- Model 11223A Cassette Memory/Special Programs Block (the cassette control block) consists of a read-only-memory (ROM block) and a keyboard overlay. The cassette control block enables the Model 20 Calculator to control the Model 9865A Cassette Memory. The cassette control operations are described in Chapter 2.

SPECIAL PROGRAMS

The cassette control block also provides two control keys which are used only with special programs available from -hp-. One such program is supplied with the cassette control block and permits the operator to specify any file (on tape) by name, rather than a file number. Instructions on how to load and use this special program are contained in the 'Special Program Booklet', which is supplied with the program.

STORAGE CAPACITY

The cassette memory expands the Model 20 storage capacity by more than 6000 data registers or 48,000 program steps*. However, since most applications require both data and program storage on the same tape, the exact cassette memory storage capacity depends upon each individual application. Techniques for determining exact tape storage capacity are described in Chapter 3.

*The expanded storage capacity depends upon the tape length; -hp- tape cassettes contain 300' of tape.

STORING INFORMATION ON TAPE

The 'file' is the basic unit of storage used with the cassette memory; each file can contain either data or program information. The user specifies the quantity and storage capacity of all files established on each tape. Once files are established, the user can store information into, or recall information from, any file.

The storage capacity (size) of each file is expressed in 'registers'. Each register can hold the contents of one calculator data-register or approximately eight programmed keystrokes.

DATA ERROR DETECTION

The cassette memory routinely checks to ensure that information being loaded into the calculator corresponds exactly to the information originally recorded. If any error† is detected during a LOAD FILE operation, the file is automatically re-loaded into the calculator. If the file cannot be successfully loaded after four attempts, the program is halted, and NOTE 30 is displayed. This feature ensures that calculator operation will not continue after erroneous data or program lines are loaded into the calculator.

† Typical causes for errors are badly worn, wrinkled, or dirty tapes.

EQUIPMENT SUPPLIED

The items supplied with the cassette control block are listed below.

Table 1-1. Equipment Supplied

DESCRIPTION	QTY.	-hp- PART NO.
Key Overlay	1	7120-3311
Operating Manual	2	09820-90023
Cassette Memory System User's Guide	2	09820-90010
Supplement F for the Model 20 Electrical Inspection Booklet	1	09820-90059
Special Program 'File-by-Name' (instruction booklet and a pre-recorded magnetic card)	1	09820-90011

INITIAL INSPECTION

The cassette control block and the equipment listed in Table 1-1 were carefully inspected before they were shipped to you. Please verify that all the equipment listed is present, and inspect the ROM block for physical damage.

To check operation of the cassette control block, see the Model 20 Electrical Inspection Booklet, which is supplied with your calculator. To check operation of your Model 20/Cassette Memory

System, see the Cassette Memory Exerciser, which is supplied with the cassette control block and should be added to the electrical inspection booklet.

If any damage or electrical malfunction is found, contact the nearest -hp- Sales and Service Office; office locations are listed at the back of this manual.

INSTALLATION PROCEDURE

As with other ROM blocks, the cassette control block may be installed in any one of the ROM Slots on top of the Model 20. Also, the cassette control block defines the keyblock directly in front of the slot in which it is installed.

Even though a ROM block can be installed in any of the slots, programs recorded on magnetic cards or tape cassettes dictate that any required block be in a specific slot — namely, the slot that the block was in when the program was recorded. Before loading any recorded program, always check the instructions which accompany the program to determine which ROM's should be installed in which slots.

It is recommended that the cassette control block be installed in ROM Slot 1, since most programs published by -hp- will specify that the cassette control block be in that slot.

To install the cassette control block:

1. Switch the calculator OFF — if you leave the calculator ON, the installed block will not be 'accepted' until ERASE is pressed.
2. Position the block over the desired slot (ROM Slot 1 is recommended), such that the 'CASSETTE MEMORY' label is readable from the front of the calculator (see Figure 1-1). Push the block straight down until it is firmly seated.

3. Install the key overlay by inserting the tab at the tip of the overlay into the locking slot at the top of the keyblock; then press the overlay down over the keys.
4. Switch the calculator ON.

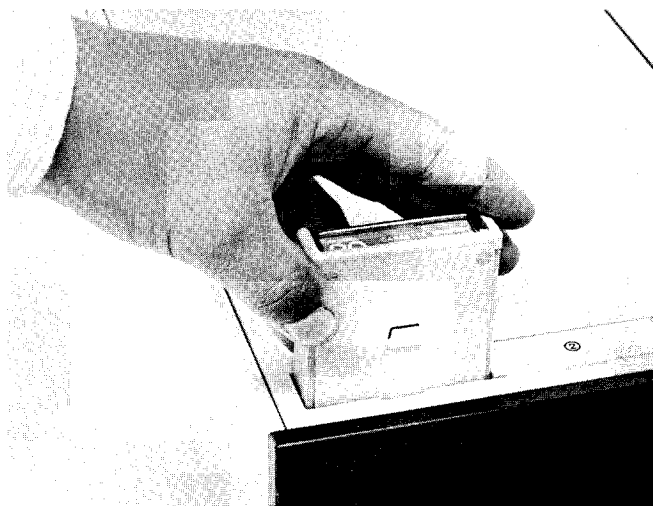


Figure 1-1. Cassette Control Block Installation

ROM BLOCK MEMORY USAGE

When the cassette control block is installed, it requires the use of 22 words of User Read-Write Memory; this use is indicated by the loss of 6 R-registers, as indicated after pressing ERASE LIST once the block is installed.

Chapter 2

CASSETTE CONTROL

INTRODUCTION

This chapter describes the cassette control operations available for the Model 20/Cassette Memory System. You should be familiar with the general operating instructions in the Cassette Memory Peripheral Manual before reading this chapter.

KEY MNEMONICS

The mnemonics which appear in the display for cassette control statements are shown in the APPENDIX at the back of this manual.

DIAGNOSTIC NOTES

The cassette control block adds eight diagnostic notes to the Model 20 System. Typical cassette control operations which cause diagnostic notes to appear are listed in the APPENDIX.

SPECIFYING PARAMETERS

Most cassette control statements use one or more parameters, such as register names, file numbers, etc. In each cassette control statement described in this chapter, any parameter can be expressed as a positive integer number, or as a variable (specified by a register name), or as an expression. For example, executing any of the SET SELECT

CODE statements shown below cause select code 6 to be set.

```
SSC 6+
SSC X+ (when X=6)
SSC 2R10+ (when R10=3)
```

SELECT CODE SPECIFICATION

SSC <select code>

Each cassette control statement which follows the SET SELECT CODE statement will be addressed to the device which is set to the select code specified*. If the cassette memory is set to respond to select code 5, the SET SELECT CODE statement may be omitted, since select code 5 is automatically set whenever the calculator is switched ON or whenever ERASE is pressed.

If the select code specified is not within the range of from 0 through 15, NOTE 20 appears in the display.

*The select code is indicated by a label on the front of the instrument. If you do not know the cassette memory select code, refer to the Cassette Memory Peripheral Manual.

MARKING TAPE

Prior to using a new (blank) tape or a scratch-pad tape** in the cassette memory, the tape must be marked with blank files. The user need only mark the number of files required for his immediate needs, as additional files may be marked at any time.

The MARK TAPE Syntax:

MRK <number of files> <file size>

The <file size> parameter is specified in registers; thus, a ten-register file can store the contents of

ten data registers or a program containing just less than 80 programmed keystrokes†.

If the tape is positioned on clear-leader, the first file is marked at the beginning of the tape and is identified as file 0; however, if the tape is not on clear-leader‡, the tape must be positioned by using a FIND FILE statement (see Page 2-5) prior to executing a MARK TAPE operation. In this latter case, the first file marked is assigned the file number of the file at which the MARK TAPE operation is begun.

**A 'scratch-pad' tape is an ordinary tape which is not used for permanent data or program storage. Such a cassette might be used when editing new programs or for temporary information storage during program operation.

† See 'Determining Program Size' in Chapter 3.

‡ A MARK TAPE operation will not be completed if it is attempted on a blank tape and the tape is not positioned on clear-leader.

MARKING TAPE

The MARK TAPE Syntax: (cont'd)

After the specified number of files are marked, an extra file is marked and the tape is then stopped at the beginning of the extra file. This extra file, which is included to facilitate the marking of additional files, should not be used for information storage, as it will be erased when additional files are marked.

FILE MARKING FORMAT

Files are marked in the format shown in Figure 2-1. Each file consists of a file-header, a file-body, and a length of slack tape. The length of tape used for each file is determined by the file-size parameter.

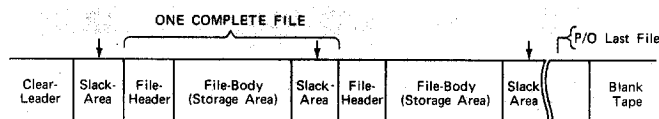


Figure 2-1. The File Marking Format

The file-header contains information such as the file number, type (data, program, or unused), absolute size (maximum file size, in registers) and current size (number of registers currently in use). The file-body contains space for storing a speci-

fied number of data (registers). The slack-tape which is at the end of each file provides an area within which the tape stops after each cassette control operation (see arrows in Figure 2-1). When the tape is stopped within the slack-tape area of one file, the cassette memory is ready to record into, load from, or identify, the next file on the tape.

SUCCESSIVE MARK TAPE OPERATIONS

Before additional files can be marked, the tape must be positioned in front of a previously marked file; usually, this should be the last file on the tape. For example, if a tape which is originally marked with ten files (the files are numbered 0 — 9) is to be marked with an additional twenty files, the tape must be positioned at file 10 (the extra file) prior to marking the new files. The following diagram shows the resulting file sequence.

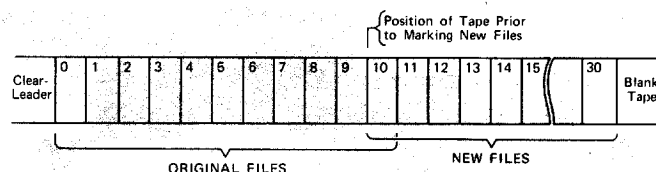


Figure 2-2. Marking Additional Files

RECORD AND LOAD STATEMENTS

RECORD PROGRAM OR DATA

```
RCF<file number>[ , R<number1>
[ , R<number2>]]
```

A program is recorded by omitting both R<number> parameters; the program lines beginning at the current line number and ending with the END statement are recorded into the specified file.

When only R<number₁> is specified, the data from R0 to the specified R-register are recorded into the specified file. When both register parameters are specified, the data from R<number₁> to R<number₂> are stored in the specified file. In general*, if R<number₁> is greater than R<number₂>, the operation is terminated before data is recorded and NOTE 02 appears in the display.

*See 'Clearing a File' on Page 3-2 for an exception.

If the specified file is too small to contain the program lines or data, the information is not recorded; however, the tape is stopped at the file which is too small, the program is halted, and NOTE 29 appears in the display.

For information on recording 'secure' programs on tape, see Page 2-8.

LOAD PROGRAM OR DATA

```
LDF<file number>[ , R<number>]
```

When the specified file contains data, the entire file contents is loaded, beginning at the calculator register specified. If the register parameter is omitted, the data is loaded beginning at R0.

When the file contains a program, the program lines are loaded beginning at the current line

RECORD AND LOAD STATEMENTS

number. In this case a register parameter is not required. If present, the contents of the register specifies the line number at which the calculator will begin executing the new program.

If the specified block of memory is too small to contain the file contents (when loading either data or program lines), the operation is halted before information is loaded and NOTE 12 appears.

EXAMPLE A

This example shows how to mark a tape with various-sized files and record programs into files.

1. Before operating the cassette memory system, verify that the cassette control block is plugged in correctly. Then verify that the cassette memory is properly installed, switched ON, and loaded with a scratch-pad cassette. Press REWIND to ensure that the tape is positioned on the beginning clear-leader.
2. Press ERASE. If the cassette memory is not set to respond to select code 5, specify the correct select code. For example, to specify select code 6, execute the following line:
3. To mark one twenty-register file (file 0) and twenty ten-register files (files 1 – 20), execute the following line:

SSC 6F

MRK 1,20;MRK 20,10F

After the mark operation is completed (it will take about 20 seconds), the tape is positioned in front of file 21 (the extra file).

4. Load the following program into the calculator. This program lists the type and size information on each file, beginning at file 0.

```
0:
FXD 0:REW F
1:
SPC 2:PRT "----TA
PE bLIST----"SPC
2F
2:
PRT "FILE bNO. ---
->(N)"F
```

(The b symbol
indicates the SPACE
key.)

```
3:
PRT "FILE bTYPE---
->(N)"F
4:
PRT "CUR.SIZE---
->(N)"F
5:
PRT "ABS.SIZE---
->(N)"F
6:
SPC 2:IDF A,B,C,
ZF
7:
PRT A,C,B,Z:GTO
6F
8:
END F
```

(The b symbol
indicates the SPACE
key.)

5. Press END RUN-PROGRAM. A portion of the printout is shown below.

---TAPE LIST---

```
FILE NO.---->(N)
FILE TYPE---->(N)
CUR.SIZE---->(N)
ABS.SIZE---->(N)
```

0
0
0
20

1
0
0
10

The file 'type' datum indicates whether the file is unused or blank (type 0), contains data (type 2) or contains program lines (type 20). The 'current size' datum indicates the number of registers now in use. The 'absolute size' datum indicates the maximum size, in registers, of the file.

RECORD AND LOAD STATEMENTS

EXAMPLE A (cont'd)

6. Now record the Tape List program into file 0 by executing the following lines from the keyboard.

```
ENDF
RCF 0F
```

7. You can verify the contents of file 0 by running the Tape List program. A portion of the printout is shown below.

```

---TAPE LIST---

FILE NO.---->(N)
FILE TYPE--->(N)
CUR.SIZE---->(N)
ABS.SIZE---->(N)

0
20
18
20

1
0
0
10
```

EXAMPLE B

This example shows how to record data into, and load data from, selected files.

1. Execute the following lines to store data in registers R1 through R5.

```
1→R1F
2→R2F
3→R3F
4→R4F
5→R5F
```

2. Enter and run the following program. This program records the contents of registers R1 through R5 into each file from file 5 to file 15.

```
0:
5→AF
1:
RCF A,R1,R5;A+1→
A;JMP A=16F
2:
END F
```

3. Now load the Tape List program (this program was recorded in the previous example) by executing the following lines from the keyboard.

```
ENDF
LDF 0F
```

4. Press END RUN-PROGRAM. After the program has listed file 15, press STOP. Verify that files 5 through 15 are now data files (file type 2) and each file contains five registers of data.

5. To load the contents of file 5 back into the calculator, beginning at R30, execute the following line.

```
LDF 5,R30F
```

Recall the contents of registers R30, R31, R32, etc. Notice that data is loaded from a file in the same order as it is recorded.

6. Now clear registers R0 through R4. Then load the contents of file 5, beginning at R0 by executing the following line.

```
LDF 5F
```

Recall the contents of R0, R1, etc. and verify their contents. Also, notice that omitting the register parameter from the LOAD FILE statement causes the file to be loaded beginning at register 0.

LINKING PROGRAMS

The technique of recording programs in separate files and using a mainline program to call and run each program as needed offers tremendous programming flexibility. This method enables one program to call literally hundreds of programs or subroutines, each of which can be loaded and executed in the same area of memory.

RECORD AND LOAD STATEMENTS

The general syntax used to link programs is shown below.

```
...;GTO<line number>;LDF<file number>+
                                end of line↑
```

When the above syntax is encountered, the program branches to the specified line number. Then the specified file is loaded (beginning at the line number) and program operation is resumed at the first line of the new program.

EXAMPLE C

This example demonstrates how a mainline program calls and runs any specified program which is recorded in a file. It's assumed that the tape contains files 0 through 5, as marked in Example A.

1. Load the following program.

```
0:
DSP "FILE 1 LOAD
ED!"
1:
DSP ;DSP ;DSP ;
GTO 0+
2:
END +
```

(The '!' character is obtained by pressing the STOP key.)

Now record the program into file 1 by executing the following lines from the keyboard.

```
END+
RCF 1+
```

2. Modify line 0 of the program to display 'FILE 2 LOADED' and record the new program into file 2.
3. In the same manner, modify the program to display 'FILE 3 LOADED' and load the program into file 3; then modify the program to display 'FILE 4 LOADED' and load the program into file 4.
4. Load the following 'mainline' program.

```
0:
FXD 0:0+AF
1:
ENT "LOAD FILE N
0.?" ;AF
2:
GTO 3;LDF AF
3:
END +
```

5. Press END RUN-PROGRAM, then enter any file number (1 through 4) and press RUN-PROGRAM again. Notice that the mainline program loads and runs the specified 'sub-program'. Now specify another file number and press RUN-PROGRAM (if the Tape List program is still in file 0, specify it).
6. Press STOP to exit the ENTER statement. Now list the current program lines. Notice that the specified file is loaded at the line number specified by the GO TO statement in line 2.

POSITIONING THE TAPE

FIND FILE

FDF<file number>

This statement causes the tape to search for, and stop in front of, the file number specified. Once the cassette memory begins executing a FIND FILE operation, keyboard control is returned or program execution is resumed with the next statement. If another cassette control statement is executed or encountered while the cassette memory is searching, calculator operation will wait at that statement until the file search is

completed — then the other cassette memory command is executed.

NOTE

Once a FIND FILE operation has begun, it cannot be immediately halted by pressing STOP. If you wish to quickly halt a FIND FILE operation, open the door on the cassette memory.

POSITIONING THE TAPE

EXAMPLE D

Re-mark the data files on the tape used in the previous examples with blank files (the tape should contain data in files 5 through 15).

1. Prior to re-marking the tape, the tape must be positioned at file 5 by executing the following FIND FILE statement.

```
FDF 5F
```

2. Now mark fifteen ten-register files by executing the following line from the keyboard.

```
MRK 15,10F
```

If the Tape List program is still in file 0, load it and verify that files 5 through 15 are now blank.

3. To show some problems which may occur if the tape is not re-marked correctly, position the tape at file 5 and mark a one-hundred-register file by executing the following line.

```
FDF 5:MRK 1,100F
```

Now run the Tape List program (stop the program after file 15 is listed). A portion of the printout is shown below.

```

4
0
0
10

5
0
0
100

6 (extra file)
0
0
100

file numbers missing

11
0
0
10

12
0
```

Notice that the new files replaced some of the original ten-register files and files 7, 8, 9, and 10 are now missing*.

4. Now execute the following line:

```
FDF 8:IDF A:AF
```

Notice that since four files are missing*, the "FIND FILE 8" statement caused the tape to be positioned at file 4 instead of file 8 (if only three files were missing, the tape would now be at file 5). This is because the cassette memory does not check to see if it has 'found' the correct file during a FIND FILE operation. Instead, it merely checks to see where the tape is positioned and then moves the tape the correct number of files in the right direction to position the tape at the specified file. Thus, if file numbers are missing, the tape will be positioned at the wrong file.

In this case, file 5 can be found by rewinding the tape prior to executing the FIND FILE statement.

5. The cassette memory **does** verify the file number during each Load File or Record File operation. To show this feature, attempt to load the contents of file 8 by executing a 'LOAD FILE 8' statement... the cassette memory will not stop searching (between files 6 and 10) for the missing file until STOP is pressed.

The operator should be aware of these, and other, situations which can result from using a tape which is incorrectly re-marked.

IDENTIFY FILE

```
IDF [<register name1>[:<register name2>
[:<register name3>[:<register name4>]]]
```

This statement causes the following information contained in the current file header to be transferred to the specified registers. When the IDENTIFY FILE operation is completed, the tape is positioned between the file-header and the file-body of the current file.

*The actual number of missing files may vary.

POSITIONING THE TAPE

- Register₁ contains the 'File Number' datum.
- Register₂ contains the 'Current Size' (no. of registers in use) datum.
- Register₃ contains the 'File Type' datum.
- Register₄ contains the 'Absolute File Size' (in registers) datum.

As indicated by the syntax, any or all of the register parameters may be omitted. If all parameters are omitted, the IDENTIFY FILE statement may be used as a 'Forward Space' command, this merely causes the tape to be advanced one file per command.

File Type Indicators:

- '0' indicates an unused (empty) file.
- '2' indicates a data file.
- '20' indicates a 'Model 20' program file.
- '28' indicates a file which contains a 'Model 20' Special Program, such as the 'File-by-Name' program supplied with the cassette control block.

NOTE

Any other file types may indicate the presence of a program for another 9800-Series calculator; however, only Model 20 programs can be loaded and run in the Model 20.

EXAMPLE E

The following program could be titled "Record in the First Available File", since after the operator specifies the first and last R-registers of the block of data to be recorded, the program searches for the first empty file which can contain the data. The programmed file-search is accomplished by continually executing an IDENTIFY FILE statement (in line 2) until an empty file is found. Then the absolute size of the file is compared with the size of the data block to be recorded; if the file size is adequate, the data is recorded. However, if the file is too small, the file search is continued.

```

0:
REW IFXD 010→X→Y
1:
ENT "FIRSTbR-REG
  =" ,X,"LASTbR-RE
G =" ,YF
2:
IDF A,B,C,Z;JMP
C=0F
3:
IF Y-X>Z;GTO 2F
4:
RCF A,RX,RYF
5:
SPC 2;PRT "DATAb
FROMbREG:" ,XF
6:
PRT "TObREG:" ,YF
7:
PRT "STOREDbINbF
ILE:" ,A;SPC 1F
8:
END F

```

The ':' character is obtained by pressing the R() key.

To run the program:

1. Load the program into the calculator. The cassette memory should be loaded with a tape which contains some blank files (as in the previous example).
2. Press END RUN-PROGRAM. After entering each register number (enter only the integer value of the register name; do not press the R() key), press RUN-PROGRAM. Once a suitable file is found and the data block is loaded, the register limits of the data block and the file number are printed. A sample printout is shown below.

```

DATA FROM REG: 0
TO REG: 9
STORED IN FILE: 5

```

POSITIONING THE TAPE

BACKSPACE

```
...JBKSI...
```

This statement causes the tape to backspace one file.

REWIND

```
...JREW...
```

This statement causes the tape to rewind to the beginning clear-leader. Once the cassette memory begins rewinding, keyboard control or program execution is resumed. Any non-cassette control operation can be performed while the tape is rewinding; however, once a cassette control statement is executed from the keyboard, or encountered in a program, the calculator will wait at that

statement until the REWIND operation is completed — then the cassette control statement is executed.

NOTE

Once a REWIND operation has begun, it cannot be immediately halted by pressing STOP. If you wish to quickly halt a REWIND operation, open the door on the cassette memory.

Sample uses of the REWIND operation are shown in the Tape List program (see Page 2-3) and the program listed in the previous example.

SECURE PROGRAMS

The same capability of recording and loading 'secure' programs when using the calculator card reader* is also available using the cassette memory.

To record a program into a file as a 'secure' program, utilize the RECORD FILE statement as previously described, but include the characters "SE" in the statement. For example, a statement to record a program as secure into file 10 must appear as shown below.

```
RCF 10,"SE"R
```

A secure program can be loaded into the calculator by executing a normal LOAD FILE statement. Once a secure program is loaded into the calculator, any other programs which are loaded with, or after, the secure program will also be treated as secure programming. This condition can be cleared by pressing ERASE or LIST or by switching the calculator OFF.

*See the description of secure programs in Chapter 5 of the Model 20 Operating and Programming Manual.

HALTING CASSETTE OPERATION

In general, any cassette control operations can be halted by pressing and holding down the STOP key until the operation is halted. In some cases, however, pressing STOP to halt a REWIND or FIND FILE operation will only temporarily halt the operation — when the key is released, the cassette memory will resume the operation. In these cases, the display also flashes NOTE 01 while the STOP key is pressed. The best way to halt a REWIND or FIND FILE operation is to open the door on the cassette memory. Also, executing an END statement will halt either of these operations.

cassette control operation is being executed immediately terminates the operation and halts the program. However, the calculator display and keyboard control may not return until STOP is pressed.

HALTING 'LOCAL' REWIND

A REWIND operation which is initiated by pressing the REWIND button on the cassette memory will be immediately 'overridden' by executing any cassette control operation from the calculator. Also, switching the calculator OFF will halt a 'local' REWIND operation.

OPENING THE CASSETTE MEMORY DOOR

Opening the door on the cassette memory while a

Chapter 3

OPERATING HINTS

TAPE STORAGE CAPACITY

The following table lists an approximate maximum number of files of a given size which can be marked on one 300' tape.

Table 3-1. Tape Storage Capacities

FILE SIZE (in registers)	MAX NO. OF FILES	FILE SIZE (in registers)	MAX NO. OF FILES	FILE SIZE (in registers)	MAX NO. OF FILES	FILE SIZE (in registers)	MAX NO. OF FILES
1	866	108	78	220	38	332	25
2	787	112	75	224	38	336	25
4	666	116	73	228	37	340	25
8	509	120	70	232	36	344	25
12	412	124	68	236	36	348	24
16	346	128	66	240	35	352	24
20	298	132	64	244	35	356	24
24	262	136	62	248	34	360	23
28	234	140	60	252	34	364	23
32	211	144	59	256	33	368	23
36	226	148	57	260	33	372	23
40	204	152	56	264	32	376	22
44	187	156	54	268	32	380	22
48	172	160	53	272	31	384	22
52	159	164	52	276	31	388	22
56	148	168	50	280	30	392	21
60	139	172	49	284	30	396	21
64	130	176	48	288	29	400	21
68	123	180	47	292	29	404	21
72	116	184	46	296	29	408	21
76	110	188	45	300	28	412	20
80	105	192	44	304	28	416	20
84	100	196	43	308	27	420	20
88	95	200	42	312	27	424	20
92	91	204	42	316	27	428	20
96	88	208	41	320	26	429	19
100	84	212	40	324	26		
104	81	216	39	328	26		

Use this program to find the amount of tape needed for a given number of equal-sized files.

```

0:
ENT "NO. OF FILES
?" ; A: ENT "FILE S
IZE?" ; B: FXD 1F
1:
IF B < 35: (54+6B) /
520 > C: JMP 2F
2:
(14+6B) / 520 > C
3:
PRT "PERCENT OF
TAPE" ; PRT "NEEDE
D:" ; AC: SPC 2F
4:
END F

```

To run the program, press END RUN-PROGRAM; then press RUN-PROGRAM again after making each requested number entry. A sample printout is shown below.

```

PERCENT OF TAPE
NEEDED:
99.6

```

DETERMINING PROGRAM SIZE

The file size required for any given program can be easily found by comparing the number of R-registers available in memory before the program is entered with the number available after the program is entered. The difference between these two numbers indicates the file size required to contain the program.

3-2 OPERATING HINTS

RE-MARKING THE TAPE

A tape which is not protected (the protect tabs on the cassette are not removed) may be re-marked by rewinding the tape and performing the required MARK TAPE operation(s). It is recommended that, if possible, the tape be completely re-marked (i.e., mark over all the files marked originally), since, if only a portion of the tape is re-marked, the original files which remain may cause problems. Figure 3-1 shows two situations that may occur when a tape is not completely re-marked.

Figure 3-1a shows the file sequence resulting after a tape, which originally contained twenty five-register files, is re-marked with five ten-register files. Notice that files 6 through 8 are now absent* and some of the original files (and their contents) remain on the tape.

Figure 3-1b shows the file sequence resulting after a tape which originally contained ten twenty-register files is re-marked with ten five-register files. Notice that some file numbers are duplicated. Under certain conditions**, this situation will cause the tape to go to the original file, rather than the re-marked file of the same number.

*The exact number of missing file numbers may vary. The example on Page 2-6 shows one problem that this situation can cause.

ORIGINAL SEQUENCE

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20*
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	-----

NEW SEQUENCE

0	1	2	3	4	5*
---	---	---	---	---	----

RESULTING FILE SEQUENCE (Missing File Numbers)

0	1	2	3	4	5		9	10	11	12	13	14	15	16	17	18	19	20*
---	---	---	---	---	---	--	---	----	----	----	----	----	----	----	----	----	----	-----

Figure 3-1a.

ORIGINAL SEQUENCE

0	1	2	3	4	5	6	7	8	9	10*
---	---	---	---	---	---	---	---	---	---	-----

NEW SEQUENCE

0	1	2	3	4	5	6	7	8	9	10*
---	---	---	---	---	---	---	---	---	---	-----

RESULTING FILE SEQUENCE (Duplicate File Numbers)

0	1	2	3	4	5	6	7	8	9	10	5	6	7	8	9	10*
---	---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	-----

* Extra File

Figure 3-1b. Re-Marking Tape

**For example, if the tape is positioned within the area originally marked, a 'FIND FILE 5' command will cause the tape to go to the original file 5, and not the new file 5.

MISCELLANEA

MASTER RECORDINGS

Since a vast amount of information can be stored on a single tape cassette, loss of the cassette's contents (e.g., normal tape wear, physical damage, exposure to a strong magnetic field, or instrument failure) could be extremely expensive to the user ... both in time and resources lost. One method of preventing such a loss is to make a duplicate recording of each often-used or valuable tape, and then storing these 'master tapes' in a safe place. the practice of making master tapes (or making master recordings on magnetic cards) should be considered mandatory when the application requires that the tape be used on a daily basis.

RECORDING ON A PROTECTED TAPE

A protected tape cassette (i.e., the protect tabs are removed) may be converted to an unprotected cassette by placing a piece of adhesive tape over

each of the former protect-tab locations. The cassette will again be protected when the adhesive tapes are removed.

CLEARING A FILE

The contents of a file will be cleared by executing a RECORD FILE statement of the following form:

RCF <file number>, R n, R n-1

For example, if file 0 contains a program or data, it would be changed to an empty file (file type is 0 and current size is 0) by executing the following line:

RCF 0, R1, R0F

This operation does not affect the absolute size of the file.

DIAGNOSTIC NOTES









Most of the execution and syntax errors associated with the cassette control block are similar in form to those of the basic calculator and cause the same 'notes' to appear. The cassette control block adds NOTE 20 and NOTES 25 through 31. The most likely errors (not all possible errors) are listed below.

- NOTE 02 Parameter Error
- a. Missing parameter in a RECORD FILE, LOAD FILE, FIND FILE, or MARK TAPE statement.
 - b. Parameter₁, of a RECORD FILE statement is greater than parameter₂ (see 'Clearing a File' on Page for an exception).
 - c. The parameter is missing in a SET SELECT CODE statement.
- NOTE 12 The available calculator memory is too small to store the specified file contents.
- NOTE 20 The select code specified is not within the range of from 0 to 15.
- NOTE 25 The specified Special Program is not stored in memory.
- NOTE 26 There are no Special Programs stored in memory.
- NOTE 27 The cassette memory door is open or ajar.
- NOTE 28 The tape cassette is protected (i.e., the protect tabs are removed).
- NOTE 29 The specified file is too small to contain the data or program lines.
- NOTE 30 The cassette memory detected an error when loading data or program lines from the specified file (see 'Data Error Detection' on Page 1-1).
- NOTE 31 The file is empty or contains information that cannot be loaded into the Model 20 (e.g., a Model 10 program).



APPENDIX

KEY MNEMONICS

CASSETTE CONTROL

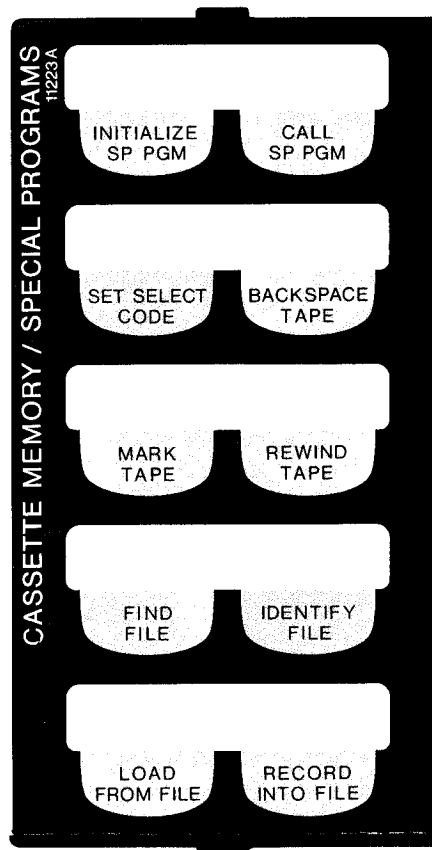
KEY	MNEMONIC*	KEY	MNEMONIC*
 SET SELECT CODE	SSC <i>b</i>	 FIND FILE	FDF <i>b</i>
 MARK TAPE	MRK <i>b</i>	 IDENTIFY FILE	IDF <i>b</i>
 RECORD INTO FILE	RCF <i>b</i>	 BACKSPACE TAPE	BKS
 LOAD FROM FILE	LDF <i>b</i>	 REWIND TAPE	REV

SPECIAL PROGRAMS**

KEY	MNEMONIC*	KEY	MNEMONIC*
 INITIALIZE SP PGM	ISP <i>b</i>	 CALL SP PGM	CSP <i>b</i>

* *b* indicates a blank space.

**These keys are used with the Special Programs supplied by -hp-. Instructions on use of these keys accompany each Special Program.





MANUAL CHANGES

MODEL 11223A CASSETTE MEMORY/SPECIAL PROGRAMS BLOCK OPERATING MANUAL

Manual Part No. 09820-90023

Page 1-2.

Replace the second paragraph in the 'INITIAL INSPECTION' section with the text given below:

To check operation of the cassette control block and the Model 20/Cassette Memory System, see the Cassette Memory System Test, which is supplied with the cassette control block and should be added to the Model 20 Electrical Inspection Booklet. The ROM TEST program, which is used to check other Model 20 ROM Blocks, cannot be used to check the cassette control block.

Page 2-1.

Add this information to the 'SELECT CODE SPECIFICATION' section:

Calculator control may be lost if a programmed cassette control statement is executed after a select code other than from 1 through 9 has been specified. This is because the cassette memory can be set to respond only to a select code from 1 through 9. Control can be regained by switching the calculator OFF and then ON.

Page 2-2.

Replace the first paragraph of the 'RECORD PROGRAM OR DATA' section with this information:

A program is recorded on tape by omitting both R(number) parameters from the above syntax. The program lines recorded start with the current line number and end with the last line of the program memory (this is usually the last END statement).

Page 2-5 and 2-8.

Add the following information to the 'FIND FILE' and 'REWIND' sections:

Calculator control may be lost if a READ statement (available with the Peripheral Control I Block) is executed while the tape is in motion in response to a FIND FILE or REWIND statement. Calculator control can be regained only by switching it OFF and then ON. This problem should not occur when using the Peripheral Control II Block.

APPENDIX

Add the following comment concerning NOTE 27:

In some cases, NOTE 27 will not appear when the cassette memory door is open.

AUGUST 24, 1972

Supplement A for 09820-90023

Scan Copyright ©
The Museum of HP Calculators
www.hpmuseum.org

Original content used with permission.

Thank you for supporting the Museum of HP
Calculators by purchasing this Scan!

Please to not make copies of this scan or
make it available on file sharing services.