CASIO®

Printed in Japan

HANDY PERSONAL COMPUTER

CASIO FP-200

OPERATION MANUAL

CASIO®

HANDY PERSONAL COMPUTER

CASIO FP-200

OPERATION MANUAL

PREFACE

This manual is intended as an FP-200 operation manual to provide beginning users with easy-to-understand explanations of how to operate the FP-200 and how to make full use of its extensive facilities. It also contains information required as an introduction to more advanced computer software techniques.
It is strongly recommended that you read this manual carefully before you use the FP-200.

Before Using the FP-200

The FP-200 is produced with sophisticated electronics and has been delivered to you after strict testing under comprehensive quality control.
Please keep in mind the following points in operating the computer, in order to ensure that it can serve you for a long time:

- Never disassemble the FP-200, it is made of precision electronic components. Do not store or operate it in an environment with high temperatures, high humidity or dust.
- Whenever the FP-200 is not in use, leave the power switch turned off. If it will not be used for a long period of time, leave the AC adapter plug disconnected.
- The FP-200 causes noise in a radio or television set if it is used near them. Also strong magnetic fields may affect the operation of the FP-200.
- Only CASIO peripheral devices should be connected to the FP-200. Casio takes no responsibility for any problems incurred by connecting products of other manufacturers. A cassette tape recorder should conform to the FP-200 standards.
- The FP-200 should be cleaned, as required, with a piece of soft dry cloth or a cloth slightly dampened with a neutral cleanser.

CONTENTS

CHAPTER 1
Let's begin by operating the FP-200

The purpose of this chapter is to explain the basic topics and features which the FP-200 provides, so that even people who have never handled a personal computer will easily be able to operate the FP-200.

People who have not actually used a computer are likely to feel that it requires a great deal knowledge to handle personal computers, and that they are hard to operate and do not tolerate mistakes. It is true that giant computers like those installed in the computer rooms of big corporations and universities often give this impression. On the contrary, however, the handy FP-200 computer is not at all difficult or troublesome to operate.

The FP-200 is a friendly computer, designed so that anyone can learn to make full use of it in a very short time.

After having read this chapter, you will be able to accomplish most of the basic operations.

1. Features of the FP-200

The Casio FP-200 is not only equipped with features that make it a handy personal computer, but also has an easy language called CETL (Casio Easy Table Language) built in which can be used without knowing the BASIC language. These features make the FP-200 a computer which can be used from the very first day it is purchased. The FP-200 also has the BASIC language (C_{85}-BASIC) built in so that it can serve as an introductory machine for people who want to study BASIC. Furthermore, because of its facility that allows CETL to be linked to BASIC, the FP-200 can provide powerful data processing capability.

- **CETL program**

In contrast to the tedious data preparation, editing, computations, re-arrangement, and rewriting with paper and pencil that used to be common in data processing, such efforts are eliminated with the FP-200, which computes surprisingly quickly, and which can further re-arrange and even tabulate the data and results. The programs for such individual tasks are already built into the FP-200. With the FP-200, using CETL, it is only necessary to give simple commands or to respond to questions.
CETL, which has sorting and retrieval capabilities in addition to such features as totals, data analysis, and editing, provides an interactive mode in which data is processed while you answer questions asked by CETL. This ensures that anyone can tabulate data easily, in contrast the tedious efforts required in the past, and makes the FP-200 a handy data bank. Data files generated by CETL, which may be linked to those generated by BASIC (C_{85}-BASIC), can be analyzed in detail. They can then be saved on cassette tapes or floppy disks for later use.

- **Large memory space**

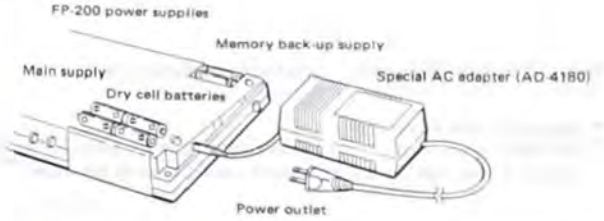
The FP-200 has a standard 32K-byte ROM. This can be expanded to 40K bytes via an optional ROM pack in an increment of 8K bytes.
Standard RAM capacity is 8K bytes which can be expanded to 32K bytes in increments of 8K bytes via optional RAM packs.

2. Power supply features

(1) Power Supplies

The FP-200 has both main and a memory back-up power supplies. The main power supply uses a dual system which can supply power to the computer from either an internal source consisting of four dry cell batteries or from a special AC adapter (AD-4180). The adapter converts the AC input power to the DC level required internally. When operating the FP-200 with batteries, the FDD and printer, serial ports cannot be used.
The memory back-up power supply uses two dry cell batteries. Two batteries must always be installed for normal operation.

■ **Power supply block diagram**



*The display disappears when the batteries are low. If the display does not come on when the **ON** key is pressed, replace the batteries.

■ **How to replace the batteries**

- **Main power supply**

1) The batteries are housed in a case located at the rear of the computer (indicated by A in the following figure). First turn the power off, then slide the case backward until it comes completely out.

2) Remove all four old batteries and replace them with new ones, being careful to observe correct polarity.
3) Slide the battery case closed.

4) Turn power on and check that the following display comes on:

<When the mode switch is set to BASIC>

```
CE5-BASIC Ver.1.0
1902 Bytes Free
Ready P0
> _
```

<When the mode switch is set to CETL>

```
CETL Ver.1.0
2923 Bytes Free
Ready
> _
```

Note: All the batteries must be installed with the correct polarity.

- Memory back-up power supply

1) The batteries are at the bottom (indicated by B in the following figure). First turn the power off, then slide the battery holder lid as indicated by the arrow.

2) Remove the two old batteries and replace them with new ones, being careful to observe correct polarity.
3) Replace the holder lid.
4) Turn the power on. The message <Memory Illegal> should appear on the display, indicating that the memory contents were lost. Push the reset button (C) located at the right of the computer with a thin rod for about 1 second with power on, turn the power off and then immediately back on again, or input the **RESET** from the keyboard.

Note: Removing the batteries causes programs and data that have been stored in the FP-200 to be lost. Any programs and data that need to be preserved should be saved on a cassette tape or floppy disk in advance.

- Supply AC power

1) When you connect the AC adapter to the FP-200, the main battery supply is automatically disconnected and external power is supplied instead to the main computer section, preventing battery waste.
2) Always turn the FP-200 power off before connecting or disconnecting the AC adapter.
3) No AC adapter other than the Casio AD-4180 model should be used.
4) This special AC adapter must be used whenever a printer or FDD (floppy disk drive) or serial interface is to be utilized.

- **Auto Power Off feature**

The FP-200 is provided with an Auto Power Off feature which automatically turns power off in order to prevent battery waste if no key entry has been made within 7 to 9 minutes, unless a program is running.
To restore power to the FP-200 after it has been turned off by this feature, press the **ON** key, or turn the power switch OFF and then turn it back ON.

- **Low Voltage Sensing feature**

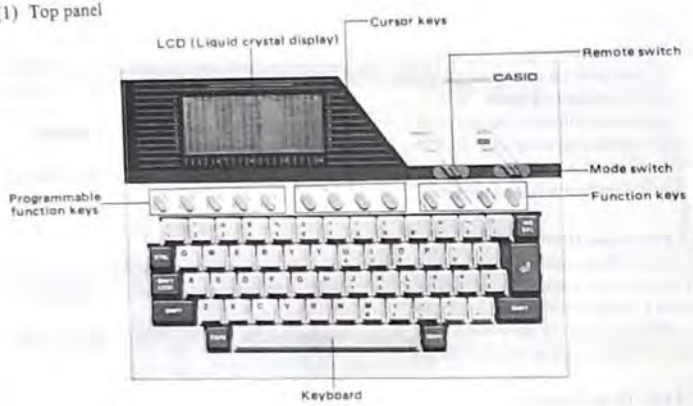
The FP-200 has a Low Voltage Sensing feature for protecting memory contents when the batteries becoming depleted and their voltage goes below a certain level.
When the batteries become low, the liquid crystal display becomes completely blank, so that the FP-200 is no longer operable. The batteries must then be replaced.

(2) External Hardware Components and Their Functions

The FP-200 has various keys and switches on its top panel. It also has several connectors for peripheral devices on the sides. All these components are listed and their functions are explained below:

■ **Components**

(1) Top panel



1. LCD (Liquid crystal display)
   A liquid crystal display device which has a display capacity of 8 lines with 20 characters per line (160 horizontal x 64 vertical dots).
2. Mode switch
   This switch selects either BASIC or CETL mode.
3. Remote switch
   With this switch ON, the remote mode on CMT is enabled. As long as no CMT is used, the switch should be set OFF to prevent waste of the batteries.
4. Keyboard
   Typewriter-like keyboard.
5. Programmable function keys
   These keys are associated with BASIC or CETL commands so that the commands can be entered simply by pressing the corresponding keys, without typing the command words.
6. Cursor keys
   Used to change character positions when entering BASIC or CETL programs and data.
7. Function keys
   (a) **CLS/HOME**
   Clears the display and returns the cursor to the home (top left) position.
   (b) **STOP/CONT**
   Stop and resumes program execution.
   (c) **BREAK**
   Breaks program execution and puts the FP-200 in command mode.
   (d) **ON**
   Restores power to the FP-200 after it is turned off by the auto power off feature.

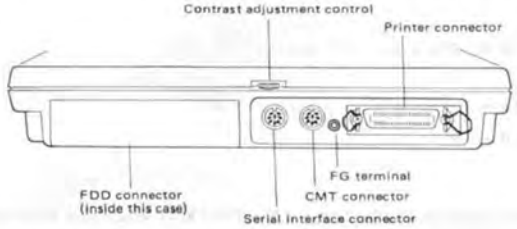
(2) Left side

Power switch

(3) Right side

1. AC adapter connector
   Whenever the printer, FDD or serial interface is used, the AC adapter (AD-4180) must be connected here.
2. Ten-key keypad connector
   Used to connect the (optional) ten-key keypad.
3. Reset button
   If the FC error persists when you turn the power back ON after replacing the back-up batteries, push this button for about 1 second. Then turn the power OFF and turn it back ON again. The FP-200 will return to its normal operating condition.

(4) Rear panel



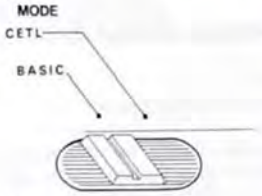
1. Contrast adjustment control
   Used to adjust LCD display contrast.
2. FDD connector
   Connects a minifloppy disk drive (FP-1021FDI).
3. Serial interface connector
   Allows connection of a peripheral device through an RS-232C standard interface (300 baud), such as an acoustic coupler or another FP-200, etc.
4. CMT connector
   Provides a means of connecting a CMT via a CMT cable (FP-1084CMC).
5. Printer connector
   Allows connection of a mini plotter printer (FP-1011PL) or graphic printer (FP-1012PR) via a printer cable (FP-1085).
6. FG terminal
   The frame ground (chassis ground) terminal for the FDD and printer.

3. Practice makes perfect

Now that we have given a brief introduction to the basic FP-200 components, let's begin with actually touching the keys to display letters and digits, and operating the FP-200.

(1) Using the Keys

1. Turn the FP-200 power switch ON.
2. Find the MODE switch that selects either CETL or BASIC at the top right of the keyboard panel, and set it to BASIC.



This will display the following:

```
C85-BASIC  Ver. 1.0

38204 Bytes Free

<<Memory Illegal>>
Ready P:
> _
```

Now look at the typewriter-like keyboard. All of the keys which have letter, digit, and special symbol key caps are used to enter characters into the computer.
The above message is transitory. To start using the FP-200, first type **RESET** slowly and correctly. This will display the following message.

```
C85-BASIC Ver. 1.0
1902 Bytes Free
Ready P0
> _
```

This is an opening message telling you that "the FP-200 is ready for your service". The greater-than symbol (>) at the beginning of the line immediately following the message is called a prompt means that you may enter a command. The underline following the prompt is the cursor, which indicates that what you enter will appear there. (If the display is not easily readable, adjust it for best readability with the contrast adjustment control.)

(2) Try to Display Your Computer's Name

First press the **F** key. The letter F will appear at the top left of the display. Then, press keys to enter **P-200**. The numeric keys 1-9, 0 are at the top of the keyboard. Press the [-] key in the top row for the hyphen (-).

(3) Shift Keys Allow a Single Key to be Used in Many Ways

Press the **CLS/HOME** key while holding the **SHIFT** key.

The display was cleared.
Press keys [* 3] [(4] [A] [S] while holding the **SHIFT** key. This **SHIFT** is provided at both the left and right ends of the keyboard so that you can use whichever is most convenient. They function completely identically, just as on a typewriter.

*To use the **SHIFT** key, you need to hold it down while another key is pressed.

Key operation: [SHIFT] [* 3] [(4] [A] [S]
This key operation displays the following:

```
# $ a s_
```

In this way, the labeled top left symbols or lower-case letters are displayed when the **SHIFT** key is used. This key provides you with a way of using a single key for more than one purpose. While the **SHIFT** key is pressed, the keyboard is in "shift mode".

(4) Capital Shift Key

At the left of the keyboard, you will find a key labeled **CAPS**. This key is used when you want to display lower case letters of alphabet.

(5) Graphic Keys

Press the following keys with the **GRAPH** key hold down:
Key operation: GRAPH J K I O

```
♦♣♠♥
```

The playing card figures are displayed by the above key operation.
That is, you can display graphic symbols by pressing character keys with the **GRAPH** key held down. The graphic symbols and their corresponding keys are listed at the end of the FP-200 Reference Manual.

(6) Cursor

It acts like a guide for you.
Let's clear the display first to begin with our practices with the cursor.

Key operation: SHIFT CLS

Everything disappeared from the screen except for an underscore, didn't it? You may have already noticed this underscore during the above your practices of just selecting and entering characters from the keyboard. It is called a "cursor". The next character you enter appears above this cursor, which then moves to the right. Actually, the cursor is a guide which tells you that "the character you are entering will appear here". When you want to enter a character somewhere on the display, first move the cursor to the desired position.
The cursor can be also used to correct or re-enter any already-entered character.
Now, let's move the cursor.
At the top of the keyboard, you will find four cursor keys ↑ ↓ ← →. Each time you press any one of these keys, the cursor moves by one line or character position in the direction indicated by the arrow.

Key operations:
Press → 9 times and then,
↓ 3 times.

```
_
```

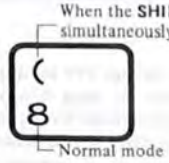

The cursor is now positioned at the center of the display. In this way, the cursor can be moved as desired on the display by using the cursor keys.
Now press the **CAPS** key. The cursor will be split into two in the line where it was positioned. This indicates that FP-200 is in the **CAPS** (lower case) mode. Press the **CAPS** key again. The split lines cursor will return to the original single cursor. Now, press the **SHIFT LOCK** key. Two lines cursor will appear at the same character position.

Cursors	– :	Single cursor	Normal mode
	= :	Two lines cursor	Shift lock mode
	⁼ :	Split lines cursor	**CAPS** (lower-case letters) mode
	≡ :	Three lines cursor	**SHIFT LOCK** and **CAPS** mode

The cursor symbol varies, depending or the mode (shift or CAPS) as follows:

Let's see what happens if we press the **CLS/HOME** key this time.
The cursor will move to the top left corner of the display, called the home position (the initial cursor position).
The basic methods of displaying characters that we have discussed so far are summarized below.

Summary



1 When you want to display on upper-case letter, or the digit or symbol appearing on the bottom left on a key cap, press the key alone (without the **SHIFT** key).
. . . . . Normal mode
2 When you want to display the lower-case letter or the symbol appearing at the top left on a key cap, press the key while holding down the **SHIFT** key.
. . . . . Shift mode
3 When you want to remain in the shift mode, press the **SHIFT LOCK** key and then the desired key. . . . . SHIFT LOCK mode
4 When you want to display only lower-case letters, press the **CAPS** key and then the desired keys. . . . . Capital shift mode
5 To clear the display, press the **SHIFT** and **CLS/HOME** keys.
6 To return the cursor to the home position, press the **CLS/HOME** key.

(7) Special Keys

- **Control Key (CTRL)**

Press the L key with the **CTRL** key held pressed. The display will disappears all at once. This functions the same as the combination of **SHIFT** and **CLS/HOME** keys.
By combining the **CTRL** key and other letters of the alphabet, you can perform various control functions. (See the Reference Manual for details.)

- **Stop/Continue key (STOP/CONT)**

If you press the **STOP/CONT** key during program execution, the program stops running. Press the key again and the program will resume execution. This key is useful for stopping the display when it moves upward (rolls up) and you have difficulty reading what is displayed.

- **Break key (BREAK)**

This key forces the program which is running to stop and returns the FP-200 to command mode (the prompt ">–" will appear).

- **Return key (↵)**

You press this key at the end of each command, line, or data item you have entered.

- **Space key**

This key enters a blank each time you press it.

Key operation: A [SPACE] B

This will display **A␣B** where "␣" indicates that there is a blank between the letters A and B.

- **Programmable function keys**

The keys which are labeled PF0 through PF9 located at the top left of the keyboard are called programmable function keys. By using these keys, you can enter frequently-used instructions with a single keystroke without having to type the instructions character-by-character. Initially, the keys are associated with the instructions shown in the tables below. These standard settings can be changed or completely revised as you wish by using an instruction called KEY (see the Reference manual for details).

- **BASIC MODE**

PF0	PF1	PF2	PF3	PF4	PF5	PF6	PF7	PF8	PF9
EDIT	PROG	SYSTEMc_R	LISTc_R	RUNc_R	S_0	FILESc_R	LOAD"	SAVE"	P.DATE$, TIME$$c_R$

- **CETL MODE**

PF0	PF1	PF2	PF3	PF4	PF5	PF6	PF7	PF8	PF9
D_3(EDIT)	FILE	SYSTEMc_R	S_H	P.FREc_R	S_0	FL(	RC(	IT(	P.DATE$, TIME$$c_R$

*S_0 : CTRL + N (NUM mode ..... ten-key like usage)
D_3 : CTRL + S (Edit mode)
S_H : CTRL + A (Command menu)

(8) Key Entry Error Correction

Here, let's study how to correct mistakes made during keyboard entry by using a simple example program which consists of only three lines.
A BASIC program line consists of a "statement number" (or "line number") and a "statement":

10	INPUT A
Statement number	Statement

Integers (while numbers) from 1 to 64999 can be used as statement numbers.

*Programs are explained in detail in Chapter 3, "BASIC Language".

First, let's clear the display:

Key operation: [SHIFT] + [CLS/HOME]

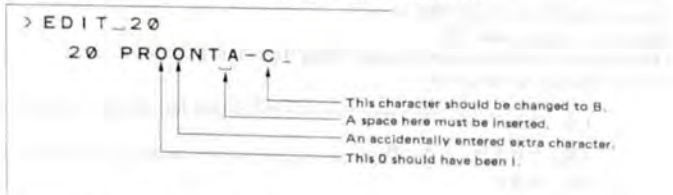
Then, let's enter the following program:

```
10␣INPUT␣A,B↵
20␣PROONTA-C↵
30␣END↵
```

The second statement intentionally contains a mistake, in order to get fractive in correcting errors; the line should have been **20 PRINT A–B** instead.
Now, let's correct the incorrect line character by character. Before you can make the correction, the line 20 must be displayed:

Key in: [PF0] **20** ↵

PF0 will display "EDIT", followed by a blank which indicates that FP-200 is ready to accept the line number 20. This key entry will display the second statement:



First, let's change the last letter C to B.
Move the cursor one character to the left by pressing the cursor key ← . It will be positioned below C. Then, press the **B** key. The display will change to the following, indicating that the correction has been made.

```
20 PROONTA-B␣
```

Now, let's insert a blank before the letter A. To do this, move the cursor underneath A by repeatedly pressing the cursor key ← , and then press the **INS DEL** key with the **SHIFT** key held down. This changes the display to the following, indicating that the blank has been inserted:

```
20 PROONT _A-B
```

There are still two errors to be corrected.
First, we will delete one "O" in the string PROONT.
Move the cursor below the left-most O. After it is positioned, press the **INS DEL** key. The display should change to:

```
20 PRONT  A-B
```

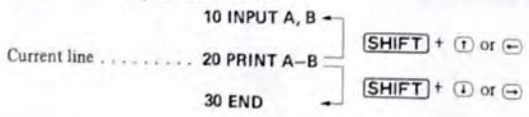
Let's change the remaining O to I next.
Press the **I** key now that the cursor is below O. The display will change to:

```
20 PRINT  A-B
```

This completes the necessary correction on the display. However, the FP-200 still has the original incorrect statement **20 PROONTA–C** in its memory. Press the ↵ key to store the corrected statement, replacing the incorrect one. This will display line 30 which contains only the END statement. Pressing the **BREAK** key here completes the correction.
Now, let's verify that the errors were corrected.
Press the **SHIFT** and **CLS/HOME** keys.
Press **PF3**, or key in **LIST** ↵ .
(This operation is used when reviewing ("listing") an entire program.)
Now the program is completed:

```
10 INPUT A, B
20 PRINT A-B
30 END
```

We can also change the command mode to the edit mode by pressing the cursor key (one of ↑ , ↓ , ← , →) with **SHIFT** key, instead of using the **EDIT** command.
On the example shown in the above, pressing the **SHIFT** + ↑ after the input of the line 30, the contents of line number 20 are displayed and come to be changeable. The example for **SHIFT** + cursor keys is shown below.



Let's abandon program line correction here, and learn more about editing programs.
To add or replace a line in the middle of creating a program, key in:

```
line-number  statement ↵
```

To delete an entire line, key in:

```
line-number ↵
```

These are the most fundamental editing operations.

In addition, there are the following various ways of displaying programs which are also frequently used:

(PF3) Display an entire program by rolling it up.

```
LIST line-number ↵
```
Displays only the specified line.

```
LIST line-number , ↵
```
Displays the specified line and all subsequent lines.

(A hyphen " - " may be used in place of the comma " , ".)

```
LIST line-number , line-number ↵
```
Displays from the first line through the second line.

```
LIST , line-number ↵
```
Displays from the beginning of a program to the specified line.

(9) Deleting Programs and Reserving Program Areas

Let's now study how to delete existing programs and reserve areas for new programs.

- **Deleting programs**

Programs in the current program area can be deleted by keying in **NEW** ↵ . This key operation cannot delete the contents of any variable or any programs in other program areas.
To delete all programs in all the areas and the contents of all their variable, key in **NEW ALL** ↵ .

• **Designating a program area**

There is no shortcut to mastering BASIC other than to get skillful at it by writing many programs. However, it would be annoying if you had to erase a program which you had previously created with much effort in order to enter a new program.

Not only in practices, but also in actual FP-200 operation, there will often be cases when you want to keep several programs in memory simultaneously or enter a program into the FP-200 for testing without destroying other existing programs.

A program partitioning facility, that allows more than one program to be simultaneously stored in memory, is useful in such situations.

The FP-200 has 10 program areas (PROG 0 through 9) which can be used to store programs arbitrarily within the memory capacity.

As you will have noticed, **Ready P0** has appeared on the display panel which indicates that the program area in current use is PROG 0.

Let's change the program area to PROG 1. Key in **PROG 1** ↵ from the keyboard. **Ready P1** will appear on the display:

```
Ready P0
PROG 1

Ready P1
```

Now we can use program area 1. Let's enter the following program in program area 1:

```
10 INPUT X ↵
20 PRINT X/2 ↵
30 END ↵
```

Press **PF3** and check that the program is correctly entered. Do you have the following on the display? If you don't, key in the correct program again or correct the errors.

```
LIST

  10 INPUT X
  20 PRINT X/2
  30 END
```

After the correct program in program area 1, let's check to see if the program which was program area 0 remains unchanged. Key in **PROG 0** ↵. This should display **Ready P0**. Then, press **PF3**. Now, check that the following appears on the display:


```
Ready P0
LIST

  10 INPUT A, B
  20 PRINT A-B
  30 END
```

We have learned that two programs can be simultaneously stored in memory. Of course, you can designate PROG 8, PROG 3 or any other program area for another program.

• **Status of the program areas**

You can examine how the program areas from PROG 0 to PROG 9 are currently used by keying in **SYSTEM** ↵ (or pressing **PF2**). This will display the following information. The numbers 26 and 28 following the values 0 and 1 (which are the program areas 0 and 1) are the sizes of (in bytes) the programs in the respective areas, (1 byte is the amount of memory needed to store one character).

```
>SYSTEM
1846 Bytes Free
0: 26
1: 28
```

• **Program protection**

We have already learned that a statement line can be deleted by keying in line-number ↵. This is a feature which can simplify program editing. If it is improperly used, however, an undesired situation can occur. All too often, the untimely entry of a number intended to be used as data causes the program statement with that line number to be deleted.

To prevent such inconvenience, the FP-200 has a feature called "passwords".

Key in **PASS "MM"** ↵ and then **LIST** ↵. No listing will appear. Do not be surprised. This is because one function of the feature is to prevent your program from being read by another person.

A password, once assigned, can be removed only by entering the same password again: keying in **PASS "MM"** ↵ again will delete the password MM.

For a password, you should select something that is easy for you to remember but hard for other people to guess.

CHAPTER 2

CETL
(CASIO Easy Table Language)

1. What is CETL?

CETL will make your FP-200 a powerful partner which can calculate totals and search and sort data from the moment you first turn it on.
Computers can carry out such tasks accurately and rapidly for us. Before they can perform such tasks, however, it is necessary to describe the task in a form called a program which can precisely instruct the computer how to perform the task. Programming a computer imposes considerable demands on us, since it is necessary to be familiar with programming languages, such as BASIC, etc., in order to make use of the computer.
Many people are said to be "allergic to computers". It is at least still generally perceived that computers are difficult to operate.
The FP-200 already has built-in programs which make the FP-200 so easy to use that it can be used from the first day you buy it, without special programming knowledge. CETL is a package of these programs.

The features provided by CETL are listed in the following:
First of all, it simplifies computer operations:

- CETL is powerful in tabulation. If you only specifies the numbers of rows and columns for each field, it prepares a table which is ready to accept data.
- CETL provides a simple interactive method for data entry in which you can enter data for all the cells in the table by answering questions asked by CETL.

As one of its most powerful features, CETL has a capability which edits interactively entered data. This capability is combination of data sorting and searching (retrieval) functions which are most frequently used in the data processing.

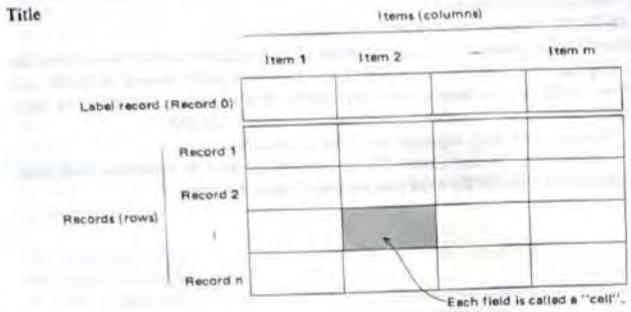
- You have probably had the experience of having to re-write a large volume of data when sorting it into alphabetical order. In addition to the alphabetical order, data is often also arranged in ascending and descending numerical order.
  CETL can rapidly arrange data in any of these three orders. For example, it has the remarkable capability to sort 50 ten digits numbers in only about eight seconds.
- Data retrieval involves looking for and picking out a specific item of data, and is one of the most common tasks.
  CETL has a conditional retrieval capability. With this capability, you can impose a certain condition on the retrieval data so that you can retrieve only data which satisfy the condition on each of the individual data items.
- If you want to perform a computation on data entered into CETL, put the necessary computational expression in the corresponding field of the table. CETL can perform function computations as well as the four arithmetic operations.
- CETL provides another capability that you can easily make corrections and changes to data in a table.
  With this capability, you can correct, delete, or insert data in a cell while displaying the data. Furthermore, you can delete or insert an entire row or column of a table simply by designating the row or column.

■ Advantages of CETL

1. CETL has ten file areas.
   When creating a new file in a computer, the previous file must be saved on a floppy disk or cassette tape. However, CETL allows you to assign up to ten file areas, F0 to F9, so that you can simultaneously create up to ten files.
   For more convenience, you can simply switch from one file area to another. Furthermore, computational operations can be easily performed on data from more than one file area by using the FL function.
2. CETL and BASIC can be used together.
   Conventional table processing programs cannot directly transfer control to and from the BASIC language processor. However, CETL can directly transfer control to BASIC and also allows BASIC to use data in any CETL table. That is, you can transfer data from CETL to BASIC and you can also transfer data from BASIC to CETL.
3. CETL commands are simply organized and easy to remember.
   CETL provides only 16 commands. The commands are easy to remember since they consist of the first letter of the word that represents their functions.

2. File organization

Files that are processed by CETL are organized in the form of table, as shown below.

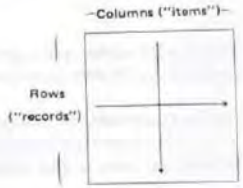


- **File naming rules**

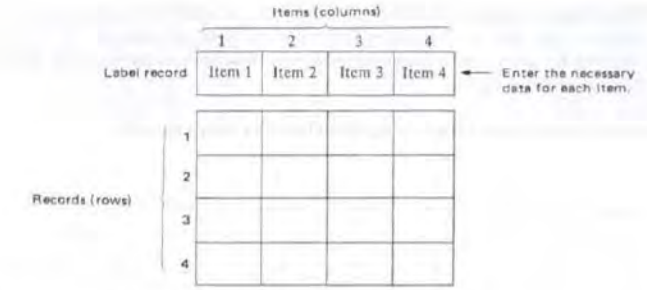
(1) A file name may have up to 8 characters.
(2) The following characters can not be used in a file name.
Comma (,), Colon (:), Period (.).

- **File creation procedure**

1. Assign a table name.
2. Determine all the rows ("records") and columns ("items") that comprise the table.



3. Enter label record items.
CETL will ask you the following four questions for each item, to which you must answer appropriately:
   - Item 1 Name?
   Reply with the name to be given to that item.
   - Type (N/S):N
   Specify whether that item is letters (S) or a number (N) by keying in S or N.
   - Expression?
   Define a computational expression.
   - Format?
   Specify the length of the item.
   (number of characters or number of digits)



The above steps create a table.

4. File all table fields with data.
Determine the direction in which you are entering individual data.
5. Compute row and column totals and sort the table, and retrieve data from the table.

- **Calculating storage capacity**

To create a large table, you must know how much data can be stored by CETL.
Here is an example of a storage capacity calculation for the standard 8K-byte memory used as file storage. In the initial state, 2,923 bytes are available for use by CETL out of the total 8K bytes.
The expression below indicates how many records (rows) can be allocated in a memory area of 2,923 bytes:

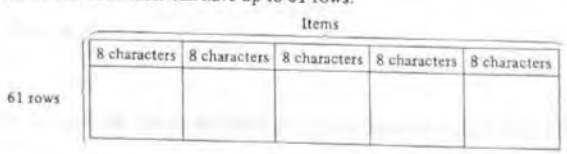
$$R = (M - 4I - 4 - F) / (2 + I + I \times C)$$

where R : Number of records
M : Available memory size (i.e., 2,923 characters)
I : Number of items (columns)
C : Number of character positions comprising a cell
F : Number of characters comprising the file name

Let's assume that the table will have five items (columns), the title (file name) is a string of 8 characters, and each cell will be able to store up to eight characters. Then, the above expression gives.

$$R = (2923 - 4 \times 5 - 4 - 8) / (2 + 5 + 5 \times 8)$$
$$= 61.510 .....$$

Thus, we find that the table can have up to 61 rows:



If you use additional RAM which provides a total memory capacity of 16K bytes and assume the following parameters:

Number of items (columns) : 10
File name length (in characters) : 6
Cell size (in characters) : 12,

the table can have up to 52 rows. It is important to ascertain beforehand how much data CETL can process by estimating with the above expression.

*The display capacity of the FP-200 display panel is limited to 8 lines with 20 character positions per line. It can display individual items of information, but a printer is required for listing entire tables. Therefore, a simple table layout should be determined before you enter data.

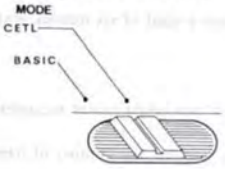
Actual operating procedures will be explained hereafter using examples:

3. Creating a table of grades

Table of Grades

1	2	3	4	5	6	7	8
SUBJECT	(Student) A	(Student) B	(Student) C	(Student) D	(Student) E	AVERAGE	DATE
MATHEMATICS	75	60	50	55	40		
ENGLISH	50	55	40	80	70		
FRENCH	45	60	70	75	65		
SCIENCE	80	65	55	40	70		
SOCIAL	50	40	65	80	30		
TOTAL							

Let's create a table like the one shown above, which contains the grades and averages of five students for five subjects, and an additional column for the date.
Now, let's study the procedure for actually creating a table with CETL.
(1) Set the mode switch to CETL.
(2) Turn the power switch ON.




```
CETL  Ver. 1.0

2923 Bytes Free

Ready F0
> _
```

The above message will appear on the display panel which indicates that CETL is ready to be used.
The "2923 Bytes Free" tells us that 2,923 bytes of memory are available for use by CETL. It depends on BASIC and character areas, and how the areas are used. It also varies when additional RAM packs are installed.
The "F0" in the third line is analogous to P0 in the corresponding BASIC message. CETL provides 10 file areas for storing files: F0 to F9.* The message means that CETL will store the data you enter in file area 0.

*A file refers to a collection of data.

(1) Creating a table ..... N Command

First, enter the N command, which stands for "New file". This instruction allows you to specify the title and size of the table you are creating.

N ↵

In response to this, CETL will ask you:

```
File Name?_
```

Reply to this:

GRADES ↵

A file name may have up to eight characters.
Then, CETL will ask you for the total number of records of the file (or rows of the table) by:

```
Number of Rec. ?_
```

Since this example uses a total of six records; Mathematics, English, French, Science, Social and Total, enter:

6 ↵

(Because the "Subject" (or label) record occupies row 0, the table will actually have a total of seven lines.)
Now, CETL will ask you for the number of items (that is, the number of columns) comprising the table by:

```
Number of Item?_
```

The items are "Subject", "Student A", ... "Student E", "Average", and "Date"; amounting to 8 items. (Please note that "Subject" is counted as item 1.) Key in:

8 ↵

Through the above operations, the title and the number of rows and columns of the table have been defined.

Rows (referred to as records).

0	Label record
1	
2	
3	
4	
5	
6	

Columns (referred to as items).

1	2	3	4	5	6	7	8

As the next step, we define the format of each item by answering questions asked by CETL for each item in sequence.
CETL first asks you:

```
Item1: Name?_
```

The first item is "Subject". Key in **SUBJECT** ↵ .
CETL will ask you next:

```
Type(N/S) : N_
```

By answering this, you determine the data type for this item: numeric or string. N stands for "Numeric", and means that the item will contain numeric data. S stands for "String" and means that the item will contain string (i.e. character) data.
Because the first item was subject (character), change the "N" after the colon to S by keying in ← **S** ↵ . (If you key in any character other than S, it is assumed that an N is entered.)

SUBJECT	
⟆	
TOTAL	

The second question from CETL is:

```
Expression ?_
```

This asks you if you want to use an expression for a calculation on the item, etc. This item is only the subject and does not need any computation. When you need to define nothing as in this case, enter only ↵ .
The third question from CETL is:

```
Format ?_
```

This asks you a cell size (in characters) you want for the item. The response to the question varies depending on whether the item is numeric or non-numeric (i.e., a string) as well as how long it will be. Since this involves string (character) data, enter a pair of ampersands "&" separately the desired number of blanks (nine in this case):
&␣␣␣␣␣␣␣␣␣& ↵
(The cell size includes a pair of ampersands, i.e., 11 characters are allocated.)

Note: If you do not specify any size and key in only ↵ , a default size of 10 characters will be automatically allocated.

This completes specifications for item 1. We will next enter the specifications for item 2.
For item 2, CETL repeats the same questions as for item 1. It first asks you:

```
Item2:Name ?_
```

The second item will contain performance data for student.
First enter A ↵. (His name).
For the second question:

```
Type( N/S) :N_
```

Enter only ↵ because the data is numeric (student A's grades in Mathematics, English, French, Science, Social, or the total of all the grades).
For the second question:

```
Expression ?_
```

Enter only ↵ because no computation is required for the item 2.
The third question is again:

```
Format ?_
```

The cells in item 2 will contain numbers. Unlike item 1, whose size was specified by blanks enclosed within ampersands, the size of item 2 is specified by as many number signs "#" as desired, where each number sign represents a digit.
Let's assume a size of four digits. Enter #### ↵.

> For a type N (numeric) item, the format is specified by combining number signs (#), a period (.), and/or a comma or commas (,). Each number sign represents a digit. Commas indicate that a comma will be inserted every three digits, and a period means the decimal point. For example, "#, ###" means that the item will contain a 4-digit number which a comma is inserted every three digits. For example, if a format #,###.## is applied to the value 3333.1, the value is represented as: 3,333.1

Repeat the same responses for items 3 through 6. Students B through E; their identifiers and attributes are defined as follows:

Item identifier and attributes for student B

Item3:Name?_	B ↵
Type(N/S) :N_	↵
Expression ?_	↵
Format ?_	####↵

Key entries for student C

Item4:Name?_	C ↵
Type(N/S) :N_	↵
Expression ?_	↵
Format ?_	#### ↵

Key entries for student D

Item6:Name?_	D ↵
Type(N/S) :N_	↵
Expression ?_	↵
Format ?_	#### ↵

Key entries for student E

Item6:Name?_	E ↵
Type(N/S) :N_	↵
Expression ?_	↵
Format ?_	#### ↵

For the seventh item, "Average", use the identifier AVERAGE, select data type N, and use an expression: SUMIT (2,6)/5. This is a function and is explained below. The format will be "#####.##". Thus, the key entries for this item are:

Item7:Name?_	AVERAGE ↵
Type(N/S) :N_	↵
Expression ?_	SUMIT
	(2,6) /5 ↵
Format ?_	######.## ↵

An expression was given in response to the question "Expression? ".
The expression used here is the function:

SUMIT	(2,6)	/	5 ↵
Function name (SUM of Items)	range of sum (Items 2 through 6)	÷	5

That is, it instructs CETL to calculate the average of items 2 through 6 by:
(Total of items 2 through 6) / 5
Functions will be explained in more detail later.

For the eighth item, "Date", use the identifier DATE, select data type S, do not use any expression, and use a 9-character string format:

Item8:Name?_	DATE↵
Type(N/S) :N_	←S↵
Expression?_	↵
Format ?_	&␣␣␣␣␣␣␣␣&↵

The data type for the dates needs to be string in order to prevent CETL from dividing 2 by 83, for example, if you enter 2/83 instead of Feb., 1983.
You can enter a date in the form MM/DD/YY with this format.
Now, the framework of the performance table is completed. Then, let's determine the direction in which we will enter data in the table before proceeding with data entry.

(2) Data Entry Direction ..... A command

The data entry direction is initially set to horizontal.
Data can be entered in a table horizontally (from left to right) or vertically (from top to bottom). The A (Auto) command is used to determine one of the two directions.
The key operations for entering the A command is:

(PF0) (or (BREAK)) A ↵

CETL will respond to this with:

```
Rec/Item(R/I) ?_
```

which asks you to enter R (Record) if you are entering data horizontally or I (Item) if you are entering data vertically.

A CETL command, including A, must be entered after you have put the FP-200 in command mode.
To accomplish this, press the **PF0** or **BREAK** key. The following message will appear on the display panel, indicating that CETL is ready to accept a command.

```
Ready F0
> _
```

Once a direction (R or I) is specified with this command, it remains effective until changed by another A command.
To find the direction of data entry, press only the ↵ key without entering data for "**1–1** ?–." In response to this, CETL will display,

1–1 ?_	↵

1–2 ?_	. . . This shows that the cell moves horizontally.
2–1 ?_	. . . This shows that the cell moves vertically.

The first number in the above displays (i.e., 1 in the first display and 2 in the second display) corresponds to the vertical direction (Records), and the second number (i.e. 2 in the first display and 1 in the second display) corresponds to the horizontal direction (Items).
Now you have found whether the entry cell will move horizontally or vertically.
To enter the A Command, press **PF0** or **BREAK** and then key in **A** ↵ .
CETL will display:

```
Rec/Item(R/I) ?_
```

Select R (horizontal) or I (Vertical). At last we can proceed with data entry.

(3) Data entry

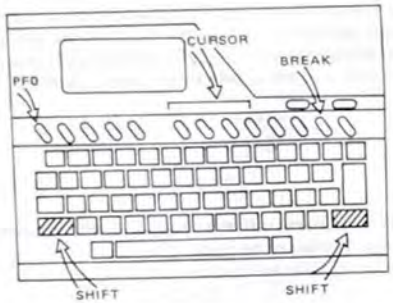
The performance data and dates are as listed in the following:

SUBJECT	A	B	C	D	E	AVERAGE	DATE
MATHEMATICS	75	60	50	55	40		4/6/83
ENGLISH	50	55	40	80	70		5/8/83
FRENCH	45	60	70	75	65		6/13/83
SCIENCE	80	65	55	40	70		7/1/83
SOCIAL	50	40	65	80	30		9/3/83
TOTAL							

If you noticed any entry error before pressing ↵ , correct it by moving cursor to the error position and entering correct character.
First we fill up to "Mathematics" row:

Display	Key entry
1–1 ?_	MATHEMATICS ↵
1–2 ?_	75↵
1–3 ?_	60↵
1–4 ?_	50↵
1–5 ?_	55↵
1–6 ?_	40↵
1–7 ?_	↵ Since we have already defined an expression (for the seventh item), skip over this entry by immediately pressing ↵.
1–8 ?_	04/06/83↵

If you notice any entry errors after you pressed ↵, you need not be worried about them. Take note of the record and item numbers (displayed at the left, such as 1 – 8, for example) and keep going. They can be corrected altogether when the data entry is completed. If you wish entry error at the time you notice it, stop the entry by pressing **PF0** (or **BREAK**) and enter the number of the record and item in which you made a mistake using the J (Jump) command. Then, CETL will display the cell which contains the error and you can then enter the correct data. You can also locate any cell location where you want to make a correction by using cell movement keys such as **SHIFT** + ↑, etc. which allow you to move to any cells within table.



We next fill the "English" row.

2 – 1 ? _	ENGLISH ↵
2 – 2 ? _	5 0 ↵
2 – 3 ? _	5 5 ↵
2 – 4 ? _	4 0 ↵
2 – 5 ? _	8 0 ↵
2 – 6 ? _	7 0 ↵
2 – 7 ? _	↵
2 – 8 ? _	0 5 / 0 8 / 8 3 ↵

Entries for FRENCH:

3 – 1 ? _	FRENCH ↵
3 – 2 ? _	4 5 ↵
3 – 3 ? _	6 0 ↵
3 – 4 ? _	7 0 ↵
3 – 5 ? _	7 5 ↵
3 – 6 ? _	6 5 ↵
3 – 7 ? _	↵
3 – 8 ? _	0 6 / 1 3 / 8 3 ↵

Entries for SCIENCE:

4 – 1 ? _	SCIENCE ↵
4 – 2 ? _	8 0 ↵
4 – 3 ? _	6 5 ↵
4 – 4 ? _	5 5 ↵
4 – 5 ? _	4 0 ↵
4 – 6 ? _	7 0 ↵
4 – 7 ? _	↵
4 – 8 ? _	0 7 / 0 1 / 8 3 ↵

Entries for SOCIAL:

5 – 1 ? _	SOCIAL ↵
5 – 2 ? _	5 0 ↵
5 – 3 ? _	4 0 ↵
5 – 4 ? _	6 5 ↵
5 – 5 ? _	8 0 ↵
5 – 6 ? _	3 0 ↵
5 – 7 ? _	↵
5 – 8 ? _	0 9 / 0 3 / 8 3 ↵

The entries for the total row are different from the rest:

6 – 1 ? _	TOTAL ↵
6 – 2 ? _	SUMRC(1 , 5)
	↵ (Total grades of student A)

The instruction SUMRC calculates the total of the records (rows) in an item while the previous SUMIT calculates the total for the items in a row or record. Thus, SUMRC (1,5) will give the total of the grades of student A from mathematics to social.

6 – 3 ? _	SUMRC(1 , 5)
	↵ (Total grades of student B)
6 – 4 ? _	SUMRC(1 , 5)
	↵ (Total grades of student C)
6 – 5 ? _	SUMRC(1 , 5)
	↵ (Total grades of student D)

```
6-6  ?_

6-7  ?_

6-8  ?_

Ready F0
>_
```

SUMRC(1, 5)

↵ (Total grades of student E)

↵ (Skip over this entry because the item has already been defined to contain an expression.)

↵ (Skip over this entry because the item has already been defined to contain a date.)

If you want to verify the data, press **PF0** or **BREAK** and enter J ↵ 1 ↵ 1 ↵ . This will return the display to the beginning of the table. Then, examine the marks one by one pressing the ↵ key. If you find an error, move the cursor back to the data item and re-enter the correct value.

This completes the table of grades.

The following is the table, printed on a printer. If no printer is provided, you can make use of the table stored in memory. Let's pick up some parts of the printed table and take a look at them.

SUBJECT	A	B	C	D	E	AVERAGE	DATE
MATHEMATICS	75	60	50	55	40	56.000	4/06/83
ENGLISH	50	55	40	80	70	59.000	5/08/83
FRENCH	45	60	70	75	65	63.000	6/13/83
SCIENCE	80	65	55	40	70	62.000	7/01/83
SOCIAL	50	40	65	80	30	53.000	9/03/83
TOTAL	300	280	280	330	275	293.00	

The following are some parts of the listing of what we have prepared in the above as printed by using the L command:

Defining items

```
File Name:GRADES

  Number of Rec.:6
  Number of Item:8


Label Record
Item1:SUBJECT
   Type(N/S)   :S
   Expression ?
   Format      :&          &

Item2:A
   Type(N/S)   :N
   Expression ?
   Format      :####
```



```
Item3:B
   Type(N/S)   :N
   Expression ?
   Format      :####

Item4:C
   Type(N/S)   :N
   Expression ?
   Format      :####

Item5:D
   Type(N/S)   :N
   Expression ?
   Format      :####

Item6:E
   Type(N/S)   :N
   Expression ?
   Format      :####

Item7:AVERAGE
   Type(N/S)   :N
   Expression :SUMIT(2,6)/5
   Format      :#####.##

Item8:DATE
   Type(N/S)   :S
   Expression ?
   Format      :&          &
```

Data entry

```
Data Area
1-1 :MATHEMATICS
1-2 :75
1-3 :60
1-4 :50
1-5 :55
1-6 :40
1-7 ?
1-8 :04/06/83
```

Mathematics grade

```
2-1 :ENGLISH
2-2 :50
2-3 :55
2-4 :40
2-5 :80
2-6 :70
2-7 ?
2-8 :05/08/83
```

English grade

```
3-1 :FRENCH
3-2 :45
3-3 :60
3-4 :70
3-5 :75
3-6 :65
3-7 ?
3-8 :06/13/83
```
French grade

```
4-1 :SCIENCE
4-2 :80
4-3 :65
4-4 :55
4-5 :40
4-6 :70
4-7 ?
4-8 :07/01/83
```
Science grade

```
5-1 :SOCIAL
5-2 :50
5-3 :40
5-4 :65
5-5 :80
5-6 :30
5-7 ?
5-8 :09/03/83
```
Social grade

```
6-1 :TOTAL
6-2 :SUMRC(1,5)
6-3 :SUMRC(1,5)
6-4 :SUMRC(1,5)
6-5 :SUMRC(1,5)
6-6 :SUMRC(1,5)
6-7 ?
6-8 ?
```
Total

(4) Let's output the results ..... T command

Let's first examine the average for mathematics. It is at the intersection of row 1 and column 7, which corresponds to the seventh item in record 1.
What should we enter now?

`>`	`T`↵
`Rec. ?`	`1`↵
`Item?`	`7`↵
`Printer(Y/N) ?`	`N`↵ (Specify whether or not to output the contents of the cell to the printer. We chose to send the output to the LCD display, so select N.)
`AVERAGE`	
`56.00`	

To output the table to the printer, make sure that it is correctly connected to the FP-200 and then enter the following:

```
>    T↵
Rec. ?  .↵
Item?  .↵
Printer(Y/N) ?    Y↵
```

The following is another example of examining a result:

`>`	`T`↵
`Rec. ?`	`6`↵
`Item?`	`2`↵
`Printer(Y/N) ?`	`N`↵
`A`	
`300`	(Total grades of student A)
`Ready F0`	
`> _`	

Item 2

	1	2	3	4	5	6	7	8
	SUBJECT	A	B	C	D	E	AVERAGE	DATE
1	MATHEMATICS						Average of Math.	
2	ENGLISH							
3	FRENCH							
4	SCIENCE							
5	SOCIAL							
⑥	TOTAL	Total grade of student A						

Rec. 6

Summary of CETL – (1)

Purpose: To create a table of grades

(1) Design the layout of the table you are creating.
(2) Determine how many rows and columns the table requires.
(3) Begin creating the actual table using the N (New file) instruction. (If a different table already exists in the area where you attempted to create the new table, an error message will be displayed.)
- Enter the File Name (title of the table).
- Enter Rec. (the number of rows comprising the table).

- Enter Item (the number of columns comprising the table).
  Enter the name of each item (i.e., item identifier).
- Specify the data type for the item (numeric or string).
- Enter an expression which will be used in the item. (If no expression needs to be defined, press only ↵ .)
- Specify the format of the item (two types of formats, numeric and string, are available).

(4) Specify whether to enter data horizontally (from left to right) or vertically (from top to bottom) by using the A command.

- **What are command mode and edit mode?**

CETL operates in either of two modes. One mode, which is called command mode is used to enter the CETL command. In this mode, CETL gives a display as follows:

```
Ready  F0
> _
```

In command mode, we enter such commands as N, which creates (or formats) a table and A, which changes the order of data entry, etc.
The other mode, which is called edit mode, is used when we enter data into a table or change data which has already been entered.
You can switch from one mode to the other at any time by pressing the **PF0** key or **BREAK** key. An example of the use of this mode will be given next.

(Example)
Assume for example, that you noticed in the course of entering data horizontally that vertical entry would be much easier.

```
1 - 1   ? _
```

Assume that the current display is as shown above (this indicates that CETL is in the edit mode for data entry).
Press the **PF0** key:

```
Ready  F0
> _
```

The display changes to what is shown above, indicating that edit mode has changed to command mode.
Now you can change the data entry direction by entering **A** ↵ :
Then enter

I ↵	to enter data vertically, or
R ↵	to enter data horizontally.

Command mode For CETL command entry.

Edit mode For data entry/correction and table format modification.

CETL Management Functions

CETL provides the following seven functions:

1. RC Indicates the current record number.
2. IT Indicates the current item number.
3. RC (<record-specification>)
   Returns the contents of the data cell in the current item from the specified record.
   The record specification may be a record name or a record number. If it is a record name, it must be one of the names entered in item 1 of some record. If a record number of 0 is specified, the item name is retrieved.
4. IT (<item-specification>)
   Returns the contents of the data cell in the current record from the specified item.
   The item specification must be an item number or item name which appears in the label record (record 0).
5. FL ([<file-specification>], [<record-specification>], [<item-specification>])
   Returns the contents of, or inserts data into, the data cell designated by the item specification and record specification in the specified file or table.
   <file-specification>: May be either the file name (identifier) or the file area number where the file resides.
   <record-specification>: May be either a record name or a record number.
   <item-specification>: May be either an item name or an item number.
   - If the file specification, record specification, or item specification is omitted, the respective current value is used. A file area, record, or item number may be replaced instead by a mathematical expression whose result will be evaluated as an integer and interpreted as the desired number.
   - If the FL command is used on the left of the equal sign in a BASIC assignment statement, the value on the right of the equal sign is assigned to the designated cell. If a label record is used in the record specification, the value on the right of the equal sign is defined and recorded as an item name.
   - When a label record is used as the record specification in an FL, 0 or blank is returned depending on the data type of the item.
     (0 ..... data type = N, blank ..... data type = S)
6. SUMRC (<record specification 1>, <record specification 2>)
   Gives the sum of the data in the cells in the current item between record specification 1 and record specification 2.
   The two record specifications may be either record name or record numbers.
   The record specified first must precede the second record in order of appearance in the table.
7. SUMIT (<item specification 1>, <item specification 2>)
   Gives the sum of the cells in the current record between the items designated by the two item specifications. The item specified first must precede the second item in order of appearance in the table.
   Some examples are given below:

Assume that you want to get the sum of A + B + C + D + E + F + G + H in cell I (represented as item 1 – 9). You can get the sum by entering SUMIT (1, 8) in cell I.

Assume next that you are entering the average of the cells A through H in cell J. Since the result is given by dividing the previous sum by 8, you may use the function IT (9) / 8 or SUMIT (1, 8) / 8.

Now, let's obtain the difference between the averages of the first four and last four cells and put the result in cell K (1 – 11). Enter the following expression in cell K:

SUMIT (1, 4) / 4–SUMIT (5, 8) / 4

You can perform similar calculations on the corresponding items in different records. Assume that you are storing the sum of cells B, L, M, and N (i.e., B + L + M + N) in cell O. This can be accomplished by entering the function SUMRC (1, 4) in O.

Since the functions can be used in any combination with arithmetic operations, they are applicable to most calculations. The CETL management functions can also be used through a link with BASIC.

4. Data retrieval ..... F command

The F command is used to retrieve data in CETL mode. This command provides a conditional data retrieval capability so as to retrieve only data which fulfills the given condition. Let's create the following table to illustrate the F command.

Title: CETL

		1	2	
The function names are strings up to 8 characters.	0	FUNCTION	COMMAND	
	1	CLEAR	K	The command names are strings up to 7 characters.
	2	SEARCH	F	
	3	INSERT	I	
	4	DELETE	D	
	5	JUMP	J	

Remember here that file area F0, already contains the table of grades. Therefore, you must decide whether to create the new table in another file area, or to delete the existing table in F0.

(1) In order to clear file areas in the FP-200 before creating a new table, use the Kill command:

To this command, CETL will respond with:

K ↵

```
>
All/Present (A/P) ? _
```

This asks you to specify whether to delete all the stored files or only the current file. The following three choices are allowed to this question:

A ↵ will delete all the files in memory.
P ↵ will delete only the current file (e.g., F0).
↵ alone will cancel the K command and not delete any files.

(2) When creating the table in another file area, enter:

(PF1) 1 ↵

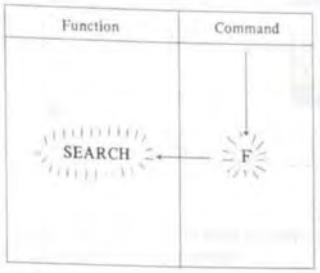
CETL will display:

```
> FILE1

Ready F1
> _
```

This indicates that F1 is now the current file area instead of F0.
Let's create the CETL command table in file area 1.

Display	Key entry
Ready F1 > _ File Name? _ Number of Rec.? _ Number of Item? _ Item 1 :Name? _ Type(N/S) :N _ Expression ? _ Format? _ Item2 :Name? _ Type(N/S) :N _ Expression ? _ Format ? _	N↵ (Prepare for creating a new table.) CETL ↵ (Enter the title of the table.) 5↵ (Specify the number of rows (records in the table).) 2↵ (Specify the number of columns (items in the table).) FUNCTION ↵ ←S↵ (Key in S because a string is to be entered in item 1 in each record.) ↵ (No expression is used for this item.) &␣␣␣␣␣␣␣&↵ (Allocate a size of 9 characters for this item.) COMMAND ↵ ←S↵ ↵ &␣␣␣␣␣&↵ End of table layout definition
	Beginning of data enter.
1 – 1 ? – 1 – 2 ? – 2 – 1 ? – 2 – 2 ? – 3 – 1 ? – 3 – 2 ? – 4 – 1 ? – 4 – 2 ? – 5 – 1 ? – 5 – 2 ? – > _	CLEAR ↵ K↵ The data is entered in the following order: SEARCH ↵ F↵ INSERT ↵ I↵ DELETE ↵ D↵ JUMP ↵ J↵

(1) Now let's actually retrieve data! ..... Use the F (Find) command for data retrieval.



Enter: F ↵

```
Rec. ? _
```

1,5 ↵ Specify the range of data search in the first item. (Record numbers as being from the first to the last record.)

1		K
2		F
3		I
4		D
5		J

```
Condition ? _
```

Enter a condition, which instructs CETL to search for the desired record name. This entry directs CETL to find "F".

IT (2) = "F" ↵

```
Printer (Y/N) ? _
```

Select N. We will not use a printer here.

N ↵

```
2: SEARCH
```

CETL displays the contents of F. 2: indicates record number 2.

(2) Let's insert more data with the I command

Use the I (Insert) command to insert more data.

	4	DELETE	D
	5	JUMP	J
(Add data.) →	6	OUTPUT	T

I ↵

```
> I
Rec/Item( R/I) ?_
```

CETL will ask you whether to add records (i.e., rows) or items (i.e., columns). Enter R because we want to add records.

R ↵

```
Rec. ?_
```

This question asks you where (after which record) we want to insert the new record or records. We will insert one record after the last record (i.e., record 5).

5 ↵

```
6-1  ?_
```

This asks you what to put in the first item of the new record (i.e., record 6). Enter the new command name.
CETL asks you what to put in the second item.

OUTPUT ↵

```
6-2  ?_
```

T ↵ Enter the code for the command which outputs the result.
This completes the insertion process.

(3) Deleting data with the D command

We learned that data can be inserted with the I command. It is also possible to remove (delete) data from the table.
Now, let's delete the data for I (i.e., record 3). To delete a record or records, we use the D (Delete command.)

D ↵

```
Rec/Item (R/I) ?_
```

CETL first asks you whether you want to delete records (rows) or items (columns).
Enter **R**, since we are deleting a record.

R ↵

```
Rec. ?_
```

This question asks you which record you want to delete.

3 ↵ You want to delete record 3.

This will delete the data for I and CETL will automatically reduce the table to five rows.

CETL

		FUNCTION	COMMAND
1	1	CLEAR	K
2	2	SEARCH	F
Removed.			
3	4	DELETE	D
4	5	JUMP	J
5	6	OUTPUT	T

Deleted record

3	Insert	I

The deleted space is removed and the record 6 disappears.

(4) You can output results with the T command

Use the T (Table) command when you want to output any results you have obtained.

T ↵

```
Rec. ?_
```

With this question, CETL asks you for records and items you want to output (or display). If you enter only a comma ",", CETL assumes that you want to output the entire table.

, ↵

```
Item  ?_
```

, ↵

```
Printer (Y/N) ? -
```

If your FP-200 has a printer, enter **Y** ↵.
When you enter **N** ↵ here, CETL will produce a listing as shown below.

```
FUNCTION    COMMAND
CLEAR       K
SEARCH      F
DELETE      D
JUMP        J
OUTPUT      T
```

(5) To skip over records or items, use the J (Jump) command

So far, we have learned the commands that operate on records (rows) or items (columns). However, CETL also provides the J (Jump) command which displays the contents of any desired data cell (i.e., jumps to a new cell). Let's learn here how to apply this new command by using the table created earlier:

	FUNCTION	COMMAND
1	CLEAR	K
2	SEARCH	F
3	DELETE	D
4	JUMP	J
5	OUTPUT	T

← CELL

J↵

```
Rec. ? -
```

Let's specify record number 2.

2↵

```
Item ? -
```


2↵ CETL displays the command name F in the selected cell (R, I) = (2, 2).

```
2 - 2 : F
```

↵

```
3 - 1 : DELETE
```

↵

```
3 - 2 : D
```

↵

```
4 - 1 : JUMP
```

In this way, the J command allows us to examine the contents of the cell whose record and item numbers are specified with this command. You can examine the subsequent items in that column one by one by successively pressing the return key.

Summary of CETL commands – 2

K command	Erases files stored in the FP-200. A erases all the file areas. P erases only the current file.
F command	Gives the data from the current file which satisfies the specified condition.
I command	Adds (inserts) records or items.
D command	Deletes the specified records (rows) or items (columns).
J command	Jumps to the specified cell and displays its contents. This command is useful for correcting data or mathematical expressions in the table.

CETL provides the following relational operations for conditional expressions >, <, =, >=, and <=. Relational expressions can be combined by logical operators such as AND, OR, XOR and NOT.

- Several examples which use relational operations and expressions are given here. Use the following sample table for your practices with the examples:

Sample Table

(Enter this table for practicing examples 1 through 6)

NAME	JAN	FEB	MAR	APR	MAY	JUN	TOTAL
A	1,230	1,350	4,450	4,470	6,680	5,580	23,760
B	1,580	3,560	8,760	5,580	2,230	1,350	23,060
C	3,860	4,430	1,280	7,890	4,450	1,790	23,700
D	2,680	2,230	3,250	1,160	2,280	5,560	17,160
E	3,560	5,670	4,450	3,360	2,460	7,300	26,800
F	2,560	2,840	2,490	4,560	7,760	9,870	30,080
G	1,450	3,390	5,670	9,760	1,890	1,450	23,610
H	2,050	4,450	3,890	2,340	3,340	6,670	22,740
I	4,200	5,340	2,240	1,470	3,560	2,560	19,370
J	2,830	1,130	7,780	5,430	1,890	3,580	22,640
TOTAL	26,000	34,390	44,260	46,020	36,540	45,710	232,920

Example 1: Select the names of people who recorded a sales result more than 5,000 in June.

```
F↵
Rec?_
1,10↵
```

Look for data for the persons A through J.

```
Condition?_
IT(7) > 5000↵
```

Item of June, greater than 5,000.

Example 2: Select the names of people who recorded a total sales less than 5000. in January and February.

```
F↵
Rec?_
1,10↵
Condition?_
SUMIT(2,3)<5000↵
```

Example 3: Compare April and May, and select the names of persons who recorded higher sales in May:

```
Condition?_
IT(5)<IT(6)↵
```

Direct CETL to determine whether the score in May (IT(6)) is greater than (>) that in April (IT(5)).

Example 4: Select the names of persons who recorded an average more than 4,000 in Jan. through June.

```
Condition ?_
SUMIT(2,7)/6>4000↵
```

Example 5: Select the names of persons who recorded a sum for Jan. through Mar. which was smaller than the sum for Apr. through June.

```
Condition ?_
SUMIT(2,4)<SUMIT(5,7)↵
```

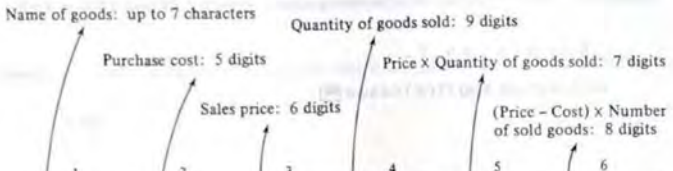
Example 6: Select the persons who recorded a sales of between 2,500 and 4,500 in April.

```
Condition ?_
IT(5)>2500 AND IT(5)<4500↵
```

5. Applying CETL to sales management

Most stores tabulate daily, weekly, and monthly sales. These tables may look simple but you will find it far more tedious than you might expect to actually make them. With CETL, you can create tables which allow automatic calculations of sales volume and profits from by only entering the quantity of goods sold.
First, let's design the sales table which we are going to create.

Title: SALES ⟶ Name the table "Sales"



	GOODS	COST	PRICE	QUANTITY	SALES	PROFIT
1	FILE	500	700	20		
2	PENCIL	50	120	30		
3	ERASER	30	80	40		
4	CLIP	15	50	23		
5	KNIFE	120	200	18		

- Assume that you are processing five kinds of goods in this example.

Now that the overall table layout is determined, let's select a new file area where we can create the sales table. How about F2?

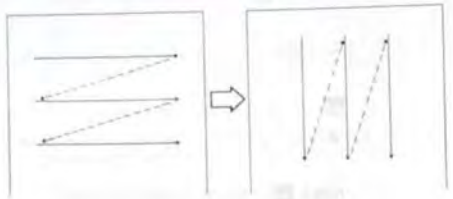
(PF 1) 2 ↵

Display	Key entry
Ready F2	This message indicates that CETL is ready to create the sales file in file area F2.
> _	N ↵ (Enter N because you are creating a new file.)
File Name ? _	SALES ↵ (Enter the title of the table.)
Number of Rec. ? _	5 ↵ (Specify the number of rows (records) comprising the table.)
Number of Item ? _	6 ↵ (Specify the number of columns items comprising the table.)

Item 1 : Name ? _	GOODS ↵ (Enter the first item name "Goods".)
Type(N/S) : N _	← S ↵ (Change N to S because the item is a string.)
Expression ? _	↵ (No expression is used for this item.)
Format ? _	&␣␣␣␣␣& (Allow the item to contain a string of up to 7 characters.)
Item 2 : Name ? _	COST ↵ (Enter the second item name "Cost".)
Type(N/S) : N _	↵ (Because costs are numeric, immediately press ↵.)
Expression ? _	↵ (No expression is used.)
Format ? _	##### ↵ (Allow the cost to have up to 5 digits.)
Item 3 : Name ? _	PRICE ↵
Type(N/S) : N _	↵
Expression ? _	↵
Format ? _	###### ↵
Item 4 : Name ? _	QUANTITY ↵
Type(N/S) : N _	↵
Expression ? _	↵
Format ? _	######### ↵
Item 5 : Name ? _	SALES ↵
Type(N/S) : N _	↵
Expression ? _	IT (3) • IT (4) ↵ (Define this expression as the product of items 3 and 4 which will give the sales amount.)
Format ? _	###,### ↵
Item 6 : Name ? _	PROFIT ↵
Type(N/S) : N _	↵
Expression ? _	(IT (3) – IT (2)) • IT (4) ↵ (Define this expression because the profit is given by multiplying the difference between item 3 (price) and item 2 (cost) by item 4 (quantity).)
Format ? _	####,### ↵

Now the table is ready to accept data.

This time we will enter data vertically instead of horizontally.



(1) Determining the data entry direction with the A command

To change the data entry direction we use the A (Auto) command. Before we can enter the A command, however, we first change to command mode:

```
1-1  ?_
```

Command mode can be entered by pressing the **PF0** key. The display will change to:

```
Ready  F1
>_
```

CETL now waits for a command from you.

Display	Key entry
>_	A↵ (Enter the A (Auto) command.)
Rec/Item(R/I)?_	I↵ (Enter data by varying items i.e., horizontally in contrast to the previous example.)
>_	J↵ (Select the top left cell in which to enter data first by using the J command.)
Rec.?_	1↵
Item?_	1↵
1-1 ?_	FILE↵
2-1 ?_	PENCIL↵ (Enter the names of goods.)
3-1 ?_	ERASER.↵
4-1 ?_	CLIP↵
5-1 ?_	KNIFE↵
Item2:Name:COST	↵ (Press only the ↵ key because the
Type(N/S):N_	↵ previous or default settings need not be
Expression?_	↵ changed and no expression is used.)
Format:#####_	↵
1-2 ?_	500↵
2-2 ?_	50↵ (Enter the costs of files, pencils,
3-2 ?_	30↵ erasers, clips and knives.)
4-2 ?_	15↵
5-2 ?_	120↵
Item3:Name:PRICE_	↵ (Press only the ↵ key because the
Type(N/S):N_	↵ previous or default settings need not be
Expression ?_	↵ changed and no expression is used.)
Format:######_	↵
1-3 ?_	700↵
2-3 ?_	120↵ (Enter the prices of the goods
3-3 ?_	80↵ in the order of files, pencils,
4-3 ?_	50↵ erasers, clips and knives.)
5-3 ?_	200↵
Item4:Name:QUANTITY	↵ (Press only the ↵ key because the
Type(N/S):N_	↵ previous settings or default settings need
Expression?_	↵ not be changed and no expression is
Format:##########_	↵ used.)

Title: SALES

	1	2	3	4	5	6
0	GOODS	COST	PRICE	QUANTITY	SALES	PROFIT
1	FILE	500	700	20		
2	PENCIL	50	120	30		
3	ERASER	30	80	40		
4	CLIP	15	50	23		
5	KNIFE	120	200	18		

Let's enter quantity data:

Display	Key entry
1-4 ?_	20↵
2-4 ?_	30↵
3-4 ?_	40↵ (Enter the quantities in the
4-4 ?_	23↵ order of files, pencils, erasers,
5-4 ?_	18↵ clips and knives.)
Item5:Name:SALES_	↵ (Press only the ↵ key because the
Type(N/S):N_	↵ previous or default settings need not
Expression:IT(3)*IT	be changed.)
(4)_	↵
Format:###,###_	↵

Leave the command mode here by pressing the **BREAK** or **PF0** key.

(● The table is never affected if you press the **BREAK** key.)

(2) Let's examine sales and profit using the T (Table) command

All of the data have been entered. Let's examine the calculated results:

Display	Key entry
`> _`	T↵ (Enter the T command, which allows you to examine the contents of any cell.)
`Rec.? _`	1↵ (Examine the sales of file.)
`Item? _`	5↵
`Printer(Y/N)? _`	N↵ (We will not use a printer.)
`SALES` `14000`	
`Ready F2` `> _`	

	1	2	3	4	5	6
	GOODS	COST			SALES	PROFIT
1	FILE					
2						
3						
4						
5	KNIFE					

Let's examine all the results, including the profits. Repeat the following sequence of key entries:

```
                    T↵
Rec.?               .↵
Item?               .↵
Printer (Y/N)?_ N↵
```

You can examine the sales and profits for all the goods on the display, one after another. If a printer is used, you can obtain the following listing.

GOODS	COST	PRICE	QUANTITY	SALES	PROFIT
FILE	500	700	20	14,000	4,000
PENCIL	50	120	30	3,600	2,100
ERASER	30	80	40	3,200	2,000
CLIP	15	50	23	1,150	805
KNIFE	120	200	18	3,600	1,440

(3) Sorting data in the table using the S command

Let's make further use of the table. As an example, we will sort the goods in the order of decreasing sales. To accomplish this, use the S (Sort) command.

Display	Key entry
`> _`	S↵ (Enter the S command.)
`Key Item? _`	4↵ (Enter 4, which designates "Quantity" (item 4.))
`Up/Down(U/D)? _`	D↵ (This question asks whether to sort in increasing (U, upward) order or decreasing (D, downward) order.)
`Rec.? _`	1.5↵ (Specify the first through last rows (records) to be sorted.)
`Mem.Move(Y/N)? _`	N↵ (Select N because the memory contents themselves are not to be changed.)
`Printer(Y/N)? _`	N↵ (Select N because we don't want to list the results on the printer.)
`3: 40` `2: 30` `4: 23` `1: 20` `5: 18`	(The goods are displayed in descending quantity: i.e., erasers, pencils, clips, files and knives.)
`Ready F2` `> _`	

- Try this with other items (cost and price, for example) as well as quantity.
- The display can hold only 20 characters per line. If the display data exceeds 20 characters, it moves (scrolls) up a line. If this continues press the **STOP/CONT** key once. The Display will stop. After you have had a chance to read it, press the **STOP/CONT** key again to allow the scrolling to continue.

	1	2	3	4	5	6
1	GOODS	COST	PRICE	QUANTITY	SALES	PROFIT
	FILE	500	700	20	14,000	4,000
2	PENCIL	50	120	30	3,600	2,100
3	ERASER	30	80	40	3,200	2,000
4	CLIP	15	50	23	1,150	805
5	KNIFE	120	200	18	3,600	1,440

The items in this column are displayed in descending (from greatest to least) order.

This table can be examined by using the T command.

Example:

T ↵

Rec? . ↵

Printer(Y/N)? N ↵

By repeating this sequence of key entries, you can display the quantities one after another. If more than eight lines are needed, the display will scroll up.
If you respond with "Y" to the question from CETL "Mem:Move (Y/N)? ", all internally stored data are rearranged, resulting in a new table, as follows. When this mode of sorting is selected, the question "Printer (Y/N)? " is not displayed.

Title: Sales

The entire table in the file area is sorted in descending quantity order and the sorted table replaces the original one.

	1 GOODS	2 COST	3 PRICE	4 QUANTITY	5 SALES	6 PROFIT
1	ERASER	30	80	40	3,200	2,000
2	PENCIL	50	120	30	3,600	2,100
3	CLIP	15	50	23	1,150	805
4	FILE	500	700	20	14,000	4,000
5	KNIFE	120	200	18	3,600	1,440

Selecting "N" to the question "Mem:Move (Y/N)? " is very useful. For example, if you want data from January to March sorted in order of decreasing by the data in February, use this command. You will be able to examine the sorted results but the table remains as it was originally entered so that you can resume entering data from April on.
If you select "Y" to the above question, the table in memory will be rearranged and it will be hard to return it to its original arrangement. This inconvenience can be eliminated by adding another column or item called "item number"—:

(Example)

Add an item "item number" to the table.

SALES	PROFIT	NO.
⋮	⋮	1
⋮	⋮	2
⋮	⋮	3
⋮	⋮	

Summary of CETL – 3

S command..... Sorts table data in either ascending or descending item number order.
You can further select whether to just display the sorted results on the display or to rearrange the table itself in memory.

The CETL commands so far we have studied are all basic commands.
CETL provides other commands as well, which will be explained in the following.

We have sorted the quantities of the goods into descending order in the sales management example. Let's sort the "Sales" item here.

Sort this item in the order of the amounts.

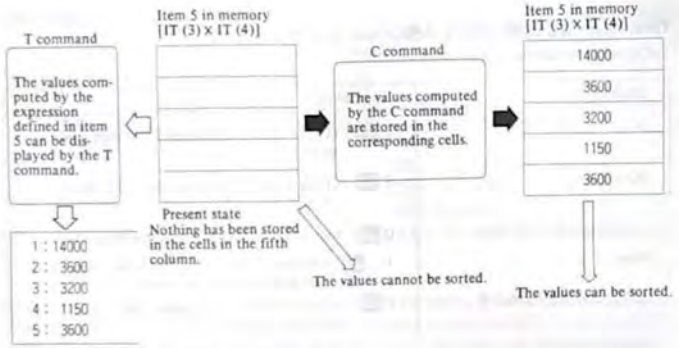
GOODS	COST	PRICE	QUANTITY	SALES	PROFIT
FILE	500	700	20	14,000	4,000
PENCIL	50	120	30	3,600	2,100
ERASER	30	80	40	3,200	2,000
CLIP	15	50	23	1,150	805
KNIFE	120	200	18	3,600	1,440

Then, enter **FILE 2** ↵ ((PF1) 2 ↵) from the keyboard.
Let's make the following key entries:

Display	Key entires
> _	S ↵ (Enter the S command to sort the table.)
Key Item ? _	5 ↵ (Direct CETL to sort the fifth item, "Sales".)
Up/Down(u/d) ? _	D ↵ (Specify sorting into descending order.)
Rec. ? _	1 . ↵ (Specify the first through last records (rows) to be sorted.)
Mem.Move(Y/N)? _	Y ↵ (Direct CETL to store the sorted results in memory.)
Ready F2	This message indicates the end of the sort.
> _	T ↵ (Enter the T command which displays the sorted results.)
Rec. ? _	. ↵ (Specify the first through last records to be sorted.)
Item ? _	5 ↵ (Direct CETL to display the fifth item, "Sales".)
Printer(Y/N)? _	N ↵
1: 14000 2: 3600 3: 3200 4: 1150 5: 3600	This is supposed to be the sorted result, but the values are not in the order of the sales amounts.
Ready F2	
> _	

This is because the S (Sort) command can only sort cells which really contains a number or string, but it cannot sort cells in which an expression is defined or in which nothing has been entered. This is why it failed to sort item 5, in which the expression IT (3) * IT (4) appears. CETL provides a command called C (Compute) to solve this problem:

Display	Key entry
> _	C↵ (Enter the C (Compute) command.)
Rec.? _	1.↵ (Direct CETL to perform a computation on the first through last records.)
Item? _	5↵ (Select the fifth item.)
> _	



Now, let's try to sort item 5 again using the S command:

Display	Key entry
Repeat the previous key entries.	
SALES 1: 14000 2: 3600 3: 3600 4: 3200 5: 1150	

GOODS	COST	PRICE	QUANTITY	SALES	PROFIT
FILE	500	700	20	14,000	4,000
PENCIL	50	120	30	3,600	2,100
KNIFE	120	200	18	3,600	1,440
ERASER	30	80	40	3,200	2,000
CLIP	15	50	23	1,150	805

The sales amounts are correctly rearranged into descending order and all the other items are also rearranged in the order of the sales amounts.
When you want to sort an item in which an expression appears, as in the above example, you need to first use the C command and store the numerical result, then sort the items.

After the execution of the C command, you cannot perform the automatic recalculation any more when you change the value of the cell which is involved in the expressions specified before. Let's actually try it:

Display	Key entry
> _	J↵ (Enter the J command.)
Rec.? _	3↵
Item? _	4↵ (Designate the column which contains the quantities of knives.)
3-4 : 18 _	←←30↵ (Correct 18 pcs to 30 pcs.)
	PF0 (Change from edit mode to command mode.)
> _	T↵
Rec.? _	.↵
Item? _	5↵
Printer(Y/N)? _	N↵
1 : 14000 2 : 3600 3 : 3600 4 : 3200 5 : 1150	Although the quantity is different, the sales amount in record (row) 3 remains 3,600. (This should be corrected to 6,000.)

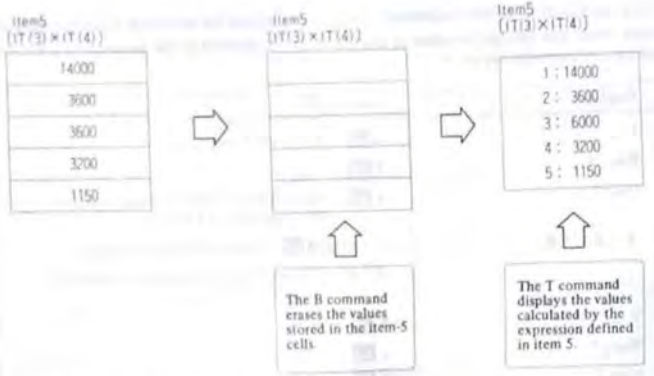
GOODS	COST	PRICE	QUANTITY	SALES	PROFIT
FILE	500	700	20	14,000	4,000
PENCIL	50	120	30	3,600	2,100
KNIFE	120	200	30	3,600	2,400
ERASER	30	80	40	3,200	2,000
CLIP	15	50	23	1,150	805

This calculation is not correct. (200 x 30 = 6,000)

This calculation is correct. (6,000 - 3,600 = 2,400)

How can we obtain the correct answer? To solve this problem, use the B (Blank) command which erases the contents of the designated cell. Let's use the B command:

Display	Key entry
> _	B↵ (Enter the B command.)
Rec.? _	1. 5↵ (Direct CETL to erase the item in the first through last records.)
Item? _	5↵ (Designate item 5.)
> _	This indicates that the B command has been executed.



Examine the results (remember how to display them).

GOODS	COST	PRICE	QUANTITY	SALES	PROFIT
FILE	500	700	20	14,000	4,000
PENCIL	50	120	30	3,600	2,100
KNIFE	120	200	30	6,000	2,400
ERASER	30	80	40	3,200	2,000
CLIP	15	50	23	1,150	805

(The correct calculation result is stored.)

When you sort an item in which an expression is defined, first store the result in that cell with the C (Compute) command, and then sort it with the S command. All the items that are contained in an expression must have been entered before the expression is evaluated. If you rewrite the contents of any item which appears in the expression after you have executed the C (Compute) command on the item where the expression is defined, it is not automatically computed again using the modified value; B command and another C command are required to accomplish this.

Exercise

Sort the sales items and produce the following table:

GOODS	COST	PRICE	QUANTITY	SALES	PROFIT
FILE	500	700	20	14,000	4,000
KNIFE	120	200	30	6,000	2,400
PENCIL	50	120	30	3,600	2,100
ERASER	30	80	40	3,200	2,000
CLIP	15	50	23	1,150	805

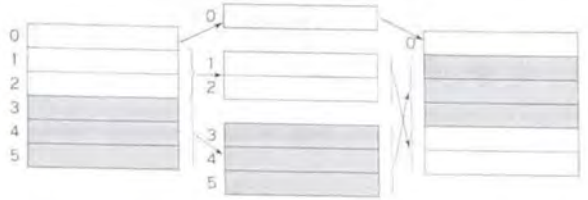
6. Rearranging records and items ..... M command

We often notice, after creating a table that we should have arranged some items differently. CETL provides a command called M (Move) which assists in such situations and allows rearrangement of rows and columns after having created a table.

The use of the M command will be explained below by using the previous sales management example. Let's move records 3 through 5 ("pencil" through "clip") so that they appear after "Goods".

GOODS	COST	PRICE	QUANTITY	SALES	PROFIT
FILE	500	700	20	14,000	4,000
KNIFE	120	200	30	6,000	2,400
PENCIL	50	120	30	3,600	2,100
ERASER	30	80	40	3,200	2,000
CLIP	15	50	23	1,150	805

Display	Key entry
> _	M↵ (Enter the M command.)
Rec/Item(R/I) ?_	R↵ (Enter R to direct CETL to rearrange records.)
Location from?_	3, 5↵ (Instruct CETL to move rows 3 through 5.)
Location to?_	0↵ (Instruct CETL to move the rows 3 through 5 to a position after row 0.)



The original records 3 through 5 are inserted after the record designated by the value supplied in answer to the question "Location to? ". If you specify a number which is greater than the largest record number in the file as either origin or destination row, an error occurs.

Let's examine (with the T command) whether the rows have been correctly rearranged (the key operations are not illustrated).

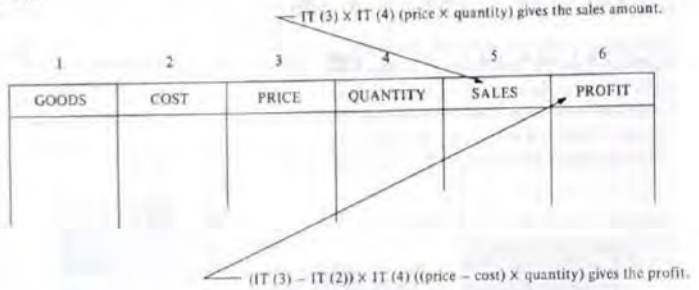
GOODS	COST	PRICE	QUANTITY	SALES	PROFIT
PENCIL	50	120	30	3,600	2,100
ERASER	30	80	40	3,200	2,000
CLIP	15	50	23	1,150	805
FILE	500	700	20	14,000	4,000
KNIFE	120	200	30	6,000	2,400

When you rearrange rows or columns in a table in which expressions have been defined, however, care must be used. Remember the previous sales management table which defines expression for item (column) 5 that computes sales amount, and an expression for item (column) 6 that calculates profits.
For example, let's see what happens if items 2 and 3 ("Cost" and "Price") are exchanged in the table.

The item-5 expression IT (3) × IT (4) will give a meaningless result because item 3 now contains costs. The item-6 expression (IT (3) - IT (2)) × IT (4) will give an undefined result because now item 2 contains prices and item 3 contains costs. If we exchange items 2 and 3, then the expressions for items 5 and 6 must also be rewritten.

Expression for item 5: IT (2) * IT (4)
Expression for item 6: (IT (2) - IT (3)) * IT (4)

After the items have been exchanged, make sure whether the expressions should be changed or not.



7. Saving data .....P and G command

Data processed by the FP-200 can be saved in either of two types of storage: main memory, which is contained in the computer, and peripheral devices which are connected to the computer and which can exchange data and programs in the computer.
Data in internal memory may be lost unintentionally during computer operations or when the batteries are exhausted. However, the external devices can save data and/or programs temporarily or semi-permanently, and they are very useful when properly used.
Peripheral devices include cassette tape units and floppy disk drives. You can make use of your home cassette recorder as a cassette tape device. Floppy disks provide much faster processing speed and greatest convenience. CETL provides the P (Put) command for saving information from FP-200 on peripheral devices.
After correctly connecting a peripheral device to the computer (see CHAPTER 5 for details), enter:

Display	Key entry
> _	P↵ (Enter the P commnad. If no data has not been entered in the table, an error occurs.)
Out(F/S/C) ?_	C↵ (When using a cassette tape unit, enter C after pushing the REC/PLAY button on the drive. Enter F when using a floppy disk. The drive will automatically start operating.)
> _	(This symbol will appear when all the data has been output to the device.)

The file name (entered with the N command) is automatically recorded on the tape or disk so that it can be easily identified when it is read at a later time.
In addition to F and C, CETL provides another peripheral device designator code "S", which designates an RS232C serial interface.
To read (load) data saved on peripheral devices into the computer, CETL provides the G (Get) command. Check that the device is correctly connected to the computer and then enter:

Display	Key entry
> _	G↵ (Enter the G command. If a file already exists in the current file area, an error will occur.)
In(F/S/C) ?_	C↵ or F↵ (Enter C or F to instruct CETL to read from a cassette tape or floppy disk, as with the P command.)
FileName?_	SALES↵ (Enter the file name.)
> _	(This symbol will appear when the data has been read.)

8. Changing the name of a file .....R command

We enter file name with the N command. File names can be changed after they have been entered for file handling flexibility. CETL provides the R (Rename) command for this purpose.

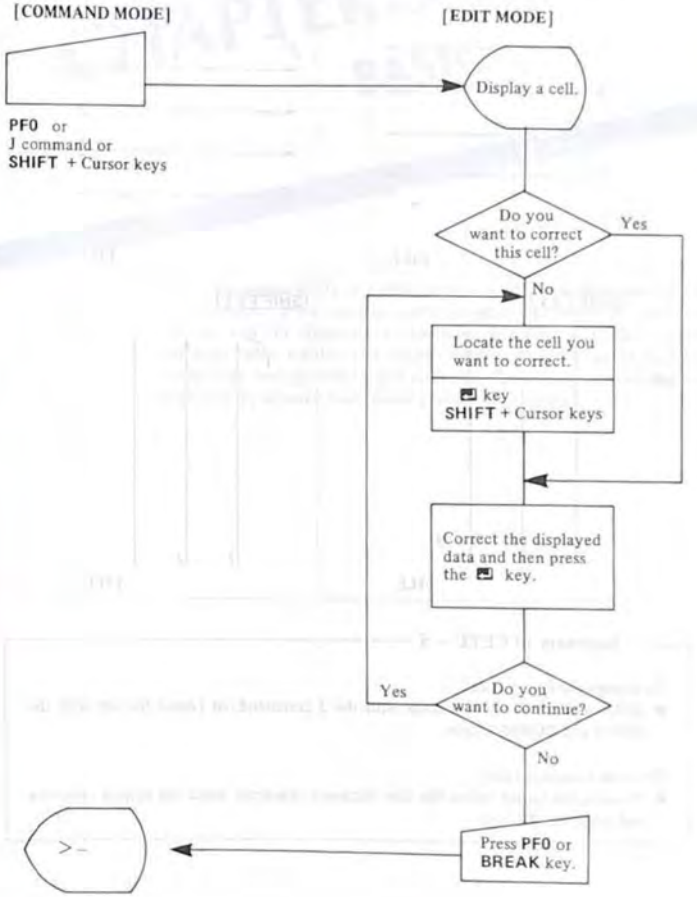
Display	Key entry
> _	R ↵ Enter the R command.
Old Name: SALES	This message informs you that the current file name is "Sales".
New Name? _	PROCEEDS ↵ Enter new name "Proceeds" in response to this question.

Summary of CETL – 4

C command	Causes the expression defined for an item to be computed, and stores the result in the cell.
B command	Erases the contents of a cell.
M command	Moves rows or columns.
P command	Saves data on a peripheral device.
G command	Reads (loads) data from a peripheral device.
R command	Changes the name of a file.

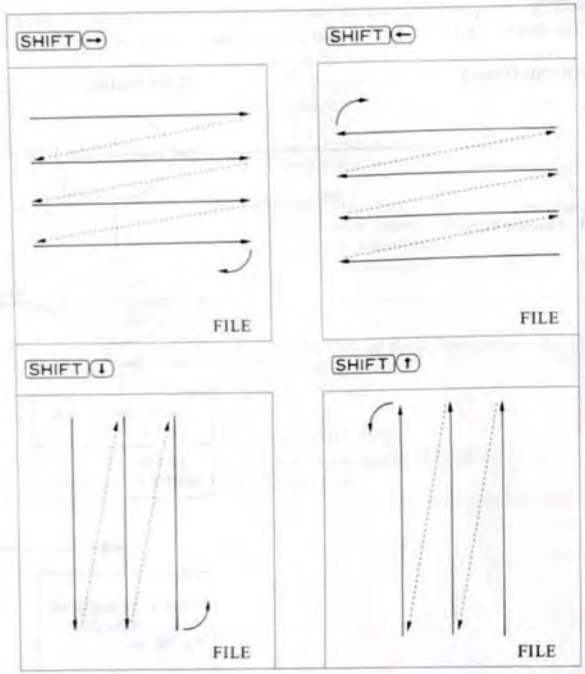
9. Correcting data within CETL

CETL has many functions which allow you to correct data stored in cells.
In CHAPTER 1 we practiced the use of the cursor keys for correcting data displayed on the panel. The data in cells in the table can also be displayed and corrected in similar ways.



However, you normally need to use only a few commands and keys. The J command and the **PF0** or **BREAK** key are probably the most frequently used keys, in addition to the **SHIFT** and cursor keys. In larger tables, it is recommended first to jump directly to, or close to, a cell you want to edit and then to edit using the **SHIFT** and **CURSOR** keys.

(Example)



Summary of CETL – 5

To display the data in a cell:

- Either specify the cell directly with the J command, or Locate the cell with the **SHIFT** and **CURSOR** keys.

To correct displayed data:

- Position the cursor below the first incorrect character, enter the correct character and press the ↵ key.

CHAPTER 3 BASIC

As we have seen, CETL is a simple program which can only perform table computations. If we want to handle graphics and games, etc. while conversing with the computer, however, we also need to be able to use a language called BASIC. This chapter is arranged to let you study BASIC commands through well-chosen examples. Please study this chapter carefully and try as many examples as possible by yourself.

1. Let's use BASIC direct commands

(1) Press the **CLS/HOME** key while simultaneously holding down the **SHIFT** key:

```
_
```

Pressing a key while the **SHIFT** key is held down will be represented by symbols such as **SHIFT CLS** or **SHIFT I** in the subsequent explanations.
If you enter in **SHIFT CLS**, the display panel will be cleared with only the symbol "_" (an underscore) remaining at the leftmost character position in line 0 (let's call this position row 0, column 1), as shown at the left. This symbol is called a cursor, and it tells you the character position where the character you are entering will appear.

(1) Write your name ..... PRINT statement (1)

Display your name on the FP-200 display panel.
Assume that your name is "CASIO FP". Display it first of all:
Enter "CASIO FP" from the keyboard. To separate F from 0 by one character position, press the space bar (a long key at the bottom of the keyboard) once. The display should be as shown below.

```
CASIO FP_
```

You displayed your name on the display panel.
Before explaining the BASIC commands, a mistake you will often make is explained here.
Press the **RETURN** key (represented by the symbol ↵):

```
CASIO FP

SN Error
Ready P0
>_
```

What do these messages mean? Is the FP-200 rebelling against you?
The **SN Error** is a message from the FP-200 which means: "You made a syntactical mistake in the above entry. Please try another entry."

What is the SN (Syntax) error?
This error is also associated with the symbol (>) displayed on the panel. This symbol is referred to as the BASIC mode prompt and tells you that FP-200 is ready to accept a command. That is, it prompts you for a command. The FP-200 is waiting for a command, not your name. What is a command, then? Remember the title of this section "Write your name." The verb "write" corresponds to a command.
You didn't enter the command to write "**CASIO FP**"; you just entered "**CASIO FP**" and the FP-200 just displayed it like an echo. That is why the FP-200 reported that you entered an illegal command (as you will see, **CASIO FP** is not a BASIC command). The next message (**Ready P0**) reminds you that your command or commands will be stored in a memory area called the program area.
The next prompt (>) again asks you to enter a command.
Let's use the PRINT command to display a message.

• **Display your name ..... PRINT "CASIO FP"**

Enter:

```
PRINT "CASIO FP"
```

Command word — What you want the FP-200 to write

Enter the character string by pressing the keys in sequence, except for the quotation marks (") which can be entered by keying in **SHIFT 2**.
Let's analyze what this line means.
PRINT is a command directing the FP-200 to display something. What are the two quotation marks " ", then? They indicate what the FP-200 should write.

• **What does ↵ mean? – 1 –**

Have you entered PRINT "**CASIO FP**"? Then, did FP-200 print (display) the name? No, it didn't. Do you know why? It is because the FP-200 will wait for other commands to be entered until it receives some indication from you which means "The command entry is complete. Execute it (or them)!"
The key labeled ↵ (called the RETURN key) is used for this purpose.
Press it now:

```
>PRINT "CASIO FP"
CASIO FP
>_
```

The display shown above will appear. That is, the FP-200 executed the PRINT command and displayed **CASIO FP** after the ↵ key was pressed.

Summary of PRINT statement 1 – message output

Purpose: Display CASIO FP.
(1) Use the command PRINT "character-string", where the string may be up to 248 characters long.
(2) Press the ↵ key.

We are now studying the command PRINT by itself, not included in any program. We are using the command "directly", as this mode of usage is referred to. Commands used in this mode are called direct commands. You need to press the ↵ key each time you enter a direct command or a series of such commands.
The ↵ key has another meaning or function which will be explained in the later explanations on programming.
Let's try several examples which use the PRINT command.

- **Let's include a comma "," and semicolon ";" in the PRINT command.**

(1) Enter:

```
>PRINT "CASIO","FP"_
```

(2) Now press the ↵ key.

```
>PRINT "CASIO","FP"
CASIO          FP
>_
```

The display shown above should appear on the FP-200 display panel.

(1) This time, enter:

```
>PRINT "CASIO";"FP"_
```

(2) Now press the ↵ key.

```
>PRINT "CASIO";"FP"
CASIOFP
>_
```

The display shown above should appear.

You learned about the PRINT statement – 1 –

(1) Spaces inside the quotation marks after PRINT are displayed as they are.
(2) One character string enclosed within quotation marks may be combined with another by a comma. When strings are combined by commas, they are displayed some distance apart from each other.
(3) One character string enclosed within quotation marks may also be combined with another by a semicolon ";". When strings are combined by semicolons, they are displayed with no intervening space between them.

Note: A string within a pair of quotation marks may be up to 248 characters long. When you use a PRINT command by combining more than one string with commas or semicolons as follows, the total length from the beginning of the command word PRINT ⑨ to the last quotation mark ⑩ must not exceed 255 characters.

```
PRINT"          ","          ";"          "
⑨①②③④⑤⑥⑦⑧⑨⑩⑪⑫⑬⑭⑮⑯⑰⑱⑲⑳㉑㉒㉓㉔㉕㉖㉗㉘㉙㉚㉛㉜㉝㉞㉟㊱㊲㊳㊴㊵㊶㊷㊸㊹㊺㊻㊼㊽㊾㊿(51)(52)(53)(54)(55)(56)(57)(58)(59)(60)(61)(62)(63)(64)(65)(66)(67)(68)(69)(70)(71)(72)(73)(74)(75)(76)(77)(78)(79)(80)(81)(82)(83)(84)(85)(86)(87)(88)(89)(90)(91)(92)(93)(94)(95)(96)(97)(98)(99)(100)(101)(102)(103)(104)(105)(106)(107)(108)(109)(110)(111)(112)(113)(114)(115)(116)(117)(118)(119)(120)(121)(122)(123)(124)(125)(126)(127)(128)(129)(130)(131)(132)(133)(134)(135)(136)(137)(138)(139)(140)(141)(142)(143)(144)(145)(146)(147)(148)(149)(150)(151)(152)(153)(154)(155)(156)(157)(158)(159)(160)(161)(162)(163)(164)(165)(166)(167)(168)(169)(170)(171)(172)(173)(174)(175)(176)(177)(178)(179)(180)(181)(182)(183)(184)(185)(186)(187)(188)(189)(190)(191)(192)(193)(194)(195)(196)(197)(198)(199)(200)(201)(202)(203)(204)(205)(206)(207)(208)(209)(210)(211)(212)(213)(214)(215)(216)(217)(218)(219)(220)(221)(222)(223)(224)(225)(226)(227)(228)(229)(230)(231)(232)(233)(234)(235)(236)(237)(238)(239)(240)(241)(242)(243)(244)(245)(246)(247)(248)⑩
```

*If you notice an error before pressing the ↵ key, move back the cursor to the incorrect character with the cursor key ← and enter the correct character again.

(2) Let's display a computational result. PRINT statement (2)

Perform several computations by using the PRINT statement:

- **Addition (+)**

```
>PRINT 1+2 ↵
 3

>_
```

The action of pressing the ↵ key will be represented by its key symbol following a command or a series of commands as shown above.

- **Subtraction (–)**

```
>PRINT 100-80 ↵
 20

>_
```

- **Multiplication (∗)**

```
>PRINT 3∗4↵
 12

>_
```

- **Division (/)**

```
>PRINT 4/8 ↵
 0.5

>_
```

• **Compound operation 1**
Expressions within parentheses take priority over others.

```
> PRINT  (1+2)*3 ↵
  9

> _
```

• **Compound operation 2**
Multiplications take priority over addition and subtruction.

```
> PRINT  1+2*3↵
  7

> _
```

• **Single precision operations**

```
> PRINT123456789*63↵
  7.77778E+09

> _
```

You will obtain the answer at the bottom of the display. The actual product of 123456789*63 is 7777777707.
Because the FP-200 cannot hold more than six digits in the normal mode single precision mode, the full ten digit result is automatically rounded up after the sixth significant digit and represented in the form 7.77778×10^9 ($\times 10^9$ is displayed as E+09 on the display).

```
> PRINT  5*3-8,6*7 ↵
  7            42

> _
```


Commas "," may also be used between expressions used in a PRINT command to separate the results from each other:

```
> PRINT "CASIO" ,5*7 ↵
CASIO          35

> _
```

A PRINT command may include messages and expressions separated by commas.
Messages in a PRINT command combined by semicolons are concatenated (combined with no intervening spaces) but values are not concatenated:

```
> PRINT "CASIO " ; "FP-"
;40*5 ↵
CASIO FP- 200
```

Summary of PRINT statement 2 – output of computational results

Purpose: Display the result of the computation.
(1) Enter an expression following the command word PRINT. (If the result exceeds 6 digits it is automatically converted to a different representation (called floating point format).
(2) Press the ↵ key.

You learned about a PRINT statement .....
(1) PRINT statement computes arithmetic expressions included in it.
(2) Arithmetic expressions may be combined by commas or semicolons within a PRINT command. A message string and expressions may also be combined.
(3) If arithmetic expressions are combined with semicolons ";", they are not concatenated (displayed next to each other).

Note: The total length of a PRINT statement from the beginning of the command word (P) to the end of its strings or expressions must not exceed 255 characters.

(3) Storing constants. Assignment statement – A = expression

When you want to perform arithmetic operations, it is often convenient to store frequently-used constants in memory. Let's practice this.

```
> PRINT 28*6:28*30:50
*(28-15) ↵
 168   840   650
```

When you use the above PRINT statement, it would be convenient if the frequently-used number 28 could be stored in memory. To accomplish this, enter:

```
> A=28 ↵

> _
```

The A = 28 is called an assignment statement and stores the number 28 in a memory location called A.
To verify that it is stored, let's display the contents of A by entering what is shown below.

```
> PRINT A ↵
 28

> _
```

28 should appear below the command.

```
> PRINT B ↵
 0

> _
```

Now let's see what happens if we try the PRINT command on another location, B.
0 should appear, indicating that nothing has yet been stored in B.
Store a number in B now:

```
> B=33 ↵

> _
```

This stores 33 in B.

Let's display the contents of both A and B this time:
Do you have the same display as shown below?

```
> PRINT A,B ↵
 28          33

> _
```

```
> PRINT "A",A ↵
A            28

> _
```

A and "A" in this example are different. The PRINT command displays "A" literally as the letter A, while it considers the letter A alone (not enclosed within quotation marks) as the name of a memory location where some number is stored. A letter or a character string used in this way (i.e., as a name or identifier) is called a variable.

```
> PRINT "A=";A ↵
A= 28

> _
```

Let's next try a computation using a variable, as shown below.

```
> PRINT A*6:A*30:50*(
A-15) ↵
 168   840   650
```

This is much easier than the previous example, in which the number 28 was used directly instead of the variable A.

```
> A=10 ↵

> PRINT A,B ↵
 10          33
```

As you can see, a variable keeps its current contents until a new value is assigned to it.

Summary – Assignment statement: A = expression

Purpose: To repeatedly use the constant 28 which is stored in memory.

(1) You can store the number 28 in memory by using a statement which has the form A = 28. A more complicated arithemetic expression may be entered on the right of the equal sign.
(2) A variable name (or identifier) may be any combination of upper- and lower-case letters and digits.
(3) The value of a variable in which no value has yet been stored is assumed to be 0.
(4) Once stored, the value of a variable is maintained until it is changed by storing a different value in the variable.
(5) Variables may be freely used in a PRINT statement.

[Examples of variables]

A variable is a name or identifier given to a place in memory which can contain a number such as 10 or 20, or a string of characters.
The following letters and digits may be used for variable names:

- Upper-case letters (A through Z)
- Lower-case letters (a through z)
- Any combination of upper-case letters and digits (called alphanumeric characters) with the exception that no variable name may begin with a digit. (Examples: A1, ABC8, etc.)
- ★ No BASIC command word must be used as a variable name; otherwise, the FP-200 would be "confused."

(Example)

○
```
> FBI=20
> A 2=10
> SCORE=20
```

×
```
IFF=30    Begins with IF, which is a BASIC command word.
TOKYO=100 Begins with the BASIC command word TO.
```

* Illegal variable naming examples are listed on page 13 of the Reference Manual.
★ A variable name can be up to 255 characters long.

You can also use string variables in BASIC.
Try the following:

```
> A$="CASIO" ↵

> PRINT A$ ↵
CASIO

> _
```


Try the next examples now:

```
> A$="I␣LOVE␣" ↵

> B$="YOU" ↵

> C$=A$+B$ ↵
PRINT C$ ↵
I LOVE YOU

> _
```

The symbol "␣" indicates each action of pressing the space bar, which puts a space between the strings.
Strings can also be "added" (i.e., concatenated).
The contents of A$ above can be changed from "I LOVE" to "YOU LOVE", if you wish.
The contents of A shown in the previous page can also be changed from 10 to 100 by changing the statement from A = 10 to A = 100. Data items represented by names, and whose contents can vary, are called variables. Variables play very important roles in BASIC and they will be explained in detail each time they appear in the following descriptions.

2. What is a program?

Compare the following two examples:

(Example 1)

```
> A=10 ↵

> B=5 ↵

> PRINT A+B ↵
  15

> _
```

(Example 2)

```
> 10 A=10 ↵
> 20 B=5 ↵
> 30 PRINT A+B ↵
> _
```

Both the examples perform the same steps of operations, but they are represented differently.
Example 1 is a sequence of direct commands, while example 2 is a BASIC program.
Now let's see what the difference between them is:

(Example 1)

The commands in example 1 are not preceded by numbers, while those in example 2 are preceded by the numbers 10, 20, and 30, respectively.

The numbers 10, 20, and 30 are called statement numbers, or line numbers.

```
10      A=10 ↵
```

Statement number Command

```
30      PRINT A+B ↵
```

The statement numbers define the sequence in which the statement are to be executed (performed).
You may wonder why the statement numbers need to be two digits 10, 20, and 30 instead of simply 1, 2, and 3 which could define their sequence just as well. They could of course be one digit numbers.
But then, how could you insert new statements between existing statements in a program after you had written it? You cannot use a statement number like 1.5, and therefore no new statement number can be inserted between statement 1 and 2. But you can insert a statement numbered 15 between statement 10 and 20, for example:

```
>15 PRINT A ↵
```

This is the reason why we usually use statement numbers separated by 5 or 10 or even 100.

(Example 2)

The number 15 is displayed after the last command, **PRINT A + B** ↵, in example 1, while no number is displayed after the corresponding statement "**30 PRINT A + B** ↵" in example 2.

This means that the action the FP-200 takes when you press ↵ varies, depending on whether the command is preceded by a statement number or not.
The ↵ after "**PRINT A + B**" in example 1 is a signal that tells the FP-200, "Command entry is complete. Execute the command!". However, the ↵ after the corresponding command in example 2 simply tells the FP-200: "Statement entry is complete." It does not direct FP-200 to execute the command.
This difference is very important. Imagine, for example, that you are to ask a child to go out buy something for you.
Assume that the child does not yet think for himself, and that you have to give him precise instructions, such as:
(1) Go to the bakery.
(2) Buy two loaves of bread.
(3) Bring them back.
If the child ran out after hearing only the first instruction (1), he would fail to hear the remaining instructions (2) and (3). To avoid this, you have to keep him there until you give him all the instructions correctly, and then say, "Now, go and do what I told you." A direct command can be likened to such a situation as the child going out after hearing only the first instruction. (A command is executed immediately after it is entered, while a program can be likened to the situation in which you let the child go after you give him all the instructions. I.e., a series of statements are executed in sequence after all of them have been entered.)

When you pressed ↵ after you entered "**A = 10**" in example 1, 10 is stored in A.

However,

When you pressed ↵ after you entered "**10 A = 10**" in example 2, 10 is not stored and A remains 0.

What is the command that tells the FP-200, "Now, begin".

It is the BASIC command:

```
RUN ↵
```

Execute the RUN command according to the following procedure:

Procedure

(1) Turn the power off.

(2) Turn the power on again.

(3) Enter:

```
> 10 A=10 ↵
```

```
> PRINT A ↵
 0

> _
```

Check whether 10 has been stored in A by the statement are entered at step (3). 0 indicates that no value has been stored yet.

(4) End the program with a statement that directs the FP-200 to terminate the execution of the program:

```
> 20 END ↵
```

(5) Tell the FP-200 to execute the program with the RUN command.

```
> RUN ↵
Ready P0
> _
```

(6) Verify that 10 has been stored in A after the RUN command has been executed.

```
> PRINT A ↵
 10

> _
```

Do you understand the function of the RUN command?
Let's proceed further with programming.

(7) Examine the program we have entered this time by:

```
> LIST ↵
```


(8) You should obtain the following display:

```
    10 A=10
    20 END
Ready P0
> _
```

(9) Let's add a statement to the program:

```
> 15 PRINT A ↵
```

(10) Then, verify that the statement has been added:

```
> LIST ↵
```

```
    10 A=10
    15 PRINT A
    20 END
```

The LIST command we used above is a command that gives a listing of an entire program, including any statements added later.

(11) Run the program:

```
> RUN ↵
 10
Ready P0
> _
```

Nothing was displayed at step (5), but 10 appears on the display panel this time. This indicates that the statement on line number 15 (or simply statement 15) has been executed.

Now, let's enter and run example-2 program by entering:

```
> 10 A=10 ↵
> 20 B=5 ↵
> 30 PRINT A+B ↵
> 40 END ↵
> 15 ↵
```

The last "15 ↵ " is a command that directs the FP-200 to delete statement 15. You can delete a statement line by entering the statement number immediately followed by a ↵ .
Run the program:

```
> RUN ↵
  15
Ready  P0
> -
```

The correct result, 15, appears below the RUN command.

Summary – What is a program?

What is a program?

(1) A program is a sequence of statements precedes by their respective statement (or line) numbers, which define the sequence in which the statements are to be executed.
(2) Within a program, a ↵ only indicates the end of a statement entered after a statement number. It does not execute the command.
(3) One program line may not always correspond to one line on the display panel, but it starts with the statement number and ends with the ↵ which terminates the command. (The total length may be up to 255 characters). Therefore, one program line may be more than one line on the display panel.
(4) END is a statement that indicates the end of program execution.
(5) LIST is a command that gives a listing of a program.

★ Programs stored in the FP-200 are not erased if power is turned off. However, it would be very laborious if you had to erase a program to make a room for a new program by deleting one line at a time in the way we learned above.
Use the following command instead:

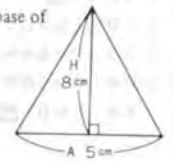
```
> NEW ↵
```

This will erase the program in the current program area.

(1) Basic Program Samples

• **Practical example of general processing**
Let's become familiar with the PRINT and assignment statements, expressions, etc.

The following program obtains the area of a triangle with a base of 5 cm and a height of 8 cm:




```
> NEW ↵
Ready  P0
> 10  A=5 ↵
> 20  H=8 ↵
> 30  C=A*H/2 ↵
> 40  PRINT  A:H,C ↵
> 50  END ↵
> _
```

The NEW command erases the existing program in order to get the FP-200 ready to accept another program. All the statements in this program, other than END in line 50, perform general processing:

Statement number	Function
10	Stores 5 in variable A.
20	Stores 8 in variable H.
30	Stores the result (area) in variable C.
40	Displays the contents of variables A, H, and C.

After this program is run, the numbers 5, 8, and 20 are displayed. If you want to vary the base length and height, replace the numbers in lines 10 and 20 with any desired values. How about trying the program with a base of 20 cm and a height of 50 cm? The values can be replaced by the following procedure:

```
> 10  A=20 ↵
> 20  H=50 ↵
> LIST ↵
```

Run the program after verifying the modification by using the LIST command.
The expression in line 30 present no problem, but it is awfully troublesome to modify the program each time we change either the base length or height. To avoid this let's use a statement that allows us to enter numeric data directly into the variables A and H.

• **Let the FP-200 request data. INPUT**
Enter:

```
> 10  INPUT  A,H ↵
```

This line is a statement that lets the computer ask you for two numbers to be stored in the variables A and H.
To respond to a request for value for both A and H, you must enter two numbers.

```
> NEW ↵
> 10  INPUT  A,H ↵
> 30  C=A*H/2 ↵
> 40  PRINT  A;H,C ↵
> 50  END ↵
> _
```

You can respond to the INPUT command in either of the following two ways:
(1) Enter two numbers separating with a comma ",":

```
Ready  P0
> RUN ↵
?20,50↵
 20    50      500

Ready  P0
> _
```

(2) It is possible to enter each number individually.

```
Ready  P0
> RUN ↵
?20↵
?50↵
 20    50      500

Ready  P0
> _
```

The same result, 500 cm^2, will be obtained with either entry.

- Looping – 1

Unconditional jump statement (GOTO) which causes repetition of specific statements.
The previous programs compute the given expressions correctly but they perform the computations only once. If you want to repeat the computation in any of the programs 10 times you have to run it 10 times by entering "RUN ↵ " each time.
To eleminate wasted effort, let's use the GOTO statement.
This command is entered in the form "GOTO statement number" where "statement number" is the statement number to which control will be transferred when the command is executed. If the program does not contain the specified statement number, an error will occur.

```
> NEW ↵
Ready  P0
> 10  INPUT  A,H ↵
> 30  C=A*H/2 ↵
> 40  PRINT  A;H,C ↵
> 50  GOTO  10 ↵
> 60  END ↵
> _
```

First, let's run the last example program by using the GOTO statement.

```
> RUN ↵
?10 ↵
?10 ↵
 10    10      50
?_
```

After you enter the first pair of numbers and the result is displayed, the program again asks you for another pair of numbers.
Answer the second pair of prompts:

```
?20,50↵
 20    50      500
?_
```

The program can be used as long as you want in this way. However, this means that the program cannot leave the same sequence of steps.

```
10 INPUT A,H      ← Start
30 C=A*H/2
40 PRINT A;H,C
50 GOTO 10
60 END
```

```
10
30
40
50
60 END
```

The GOTO command may often form an "infinite loop" that allows a program to work without stopping in this way.
To leave this "infinite loop", stop the program with the **BREAK** key. (If you stop a program with the **BREAK** key, it will never be destroyed or erased.)

- **Looping – 2**
  Repeating a specific sequence of statements a definite number of times –
  FOR – NEXT statement
  To repeat the triangle area computation three times.

How can we terminate the program after repeating the computation three times? This can be accomplished by the FOR – NEXT statements.

```
>10 FOR I=1 TO 3 ↵
>20 INPUT A,H ↵
>30 C=A*H/2 ↵
>40 PRINT A;H,C ↵
>50 NEXT I ↵
>60 END ↵
>_
```

Examine this program. Notice that only the two lines 10 and 50 have been added to the previous program:

10 FOR I = 1 TO 3

This statement counts the number of times the program loops from 1 to 3 by using the variable I.

50 NEXT I

These statements determine how many times the triangle area computation will be performed. That is, it causes statements 20 through 50 to be repeated three times.


```
>RUN ↵
?10,30↵          (First data entry)
 10   30  150    (First result output)
?20,50↵          (Second data entry)
 20   50  500    (Second result output)
?30,16↵          (Third data entry)
 30   16  240    (Third result output)
Ready P0
>_
```

Let's run the program:
The program terminates after the loop has been executed three times.
Now, let's modify the program so that we can observe visually how many times the program is repeated.
Modify the program as follows:

```
>EDIT 40↵
 40 PRINT A;H,C_
```

The "**EDIT 40**" in the first line above informs the computer that you are modifying statement 40 of the program. This causes the statement 40 to be displayed and the cursor to be positioned at the next to the end of the sentence.
Enter only:

```
:I ↵
```

The next line, 50, will then appear on the display panel, but we do not have to make any modification on it. Let's press the **BREAK** key to inform the FP-200 that our program modification has been completed. Verify that the program has been modified correctly by using the LIST statement.

```
Ready P0
>LIST ↵
10 FOR I=1 TO 3
```

The display panel scrolls up and these lines will disappear.

```
 20 INPUT A,H
 30 C=A*H/2
 40 PRINT A;H,C:I
 50 NEXT I
 60 END

Ready P0
>_
```

We inserted the variable I, which will display how many times the program has been repeated. Now let's enter **RUN** ↵ in order to run the modified program.

①A = 5, H = 8
②A = 10, H = 10
③A = 15, H = 20

Use these base lengths and heights.

```
Ready  P0
> RUN ↵
? 5 , 8 ↵
  5    8          20     1   (First loop)
? 1 0 , 1 0 ↵
  1 0    1 0      5 0    2   (Second loop)
? 1 5 , 2 0 ↵
  1 5    2 0      1 5 0  3   (Third loop)
```

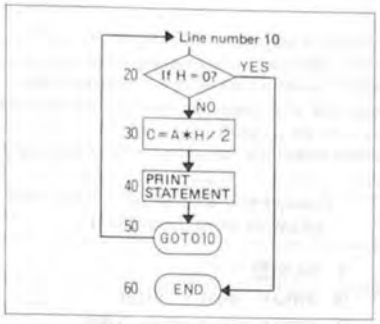
Little by little, we have made the triangle area computation program easier to use. Now let's modify the program so that it can be stopped when it should be.

- **There cannot be a triangle with a height of 0 – Stop the program when a height of 0 is encountered. – IF statement**

This time, we will modify the program so that it will stop whenever all necessary computations have been performed. To accomplish this we will let the FP-200 determine whether to continue or stop the program each time it enters a new loop, by checking to see if the height is 0.
Program the following:

```
> 1 0  I N P U T  A , H ↵
> 2 0  I F  H = 0  T H E N  6 0 ↵
> 5 0  G O T O  1 0 ↵
> L I S T ↵
    1 0  I N P U T  A , H
    2 0  I F  H = 0  T H E N  6 0
    3 0  C = A * H / 2
    4 0  P R I N T  A ; H , C ; I
    5 0  G O T O  1 0
    6 0  E N D

Ready  P0
> _
```

Statement 20 instructions the FP-200 to go to statement 60 if H is 0. The statement also tells the FP-200 to go to statement 30 if H is not 0.



Modify statement 40 as follows:

```
> E D I T  4 0 ↵
  4 0 P R I N T  A ; H , C ; I _
```

Enter ←← **SPACE SPACE** ↵. This will erase "; I", which is now unnecessary.

```
5 0  G O T O 1 0 _
```

The next line (statement 50) will appear, which we do not modify. Press the **BREAK** key to inform the FP-200 that the modification is completed.
Run the modified program now:
First loop: **A = 4, H = 8** – The computation is performed.
Second loop: **A = 4, H = 10** – The computation is performed.
Third loop: **A = 4, H = 0** – The computation is not performed but the program is terminated.

```
> R U N ↵
? 4 , 8 ↵
  4    8    1 6
? 4 , 1 0 ↵
  4    1 0  2 0
? 4 , 0 ↵

Ready  P0
> _
```

You succeed in modifying the program so that it can be terminated when you wish.

• **Polishing up programs**

Let's discuss here some ways of polishing up programs so that they can be used by people other than the person who writes them, using the previous triangle area computation as an example. The previous program displays a question mark immediately after you enter "**RUN** ↵ ". Nobody except you can understand what this means. Only you who know how to respond to the question, because you wrote the program.

Compare the following example, polished up by little "decoration" of the basic program:

(Basic program)	(Example with additions which make it easy for anyone to use the program.)
	5 CLS↵
10 INPUT A,H	10 INPUT "BASE?=" ,A↵
	15 INPUT "HEIGHT?=" ,H↵
20 IF H=0 THEN60	20 IF H= 0 THEN60
30 C=A*H/2	30 C=A*H/2
40 PRINT A:H.C	40 PRINT:PRINT "BASE =";A;HEIGHT=" ;H ↵
	45 PRINT "AREA=" ; C ↵
50 GOTO 10	50 FOR I=0 TO1000:NEXT I:GOTO 5↵
60 END	60 PRINT:END↵

(The statements followed by a ↵ have been modified.)

Let's run the "new" program, using a base length of 20 cm and a height of 80 cm as an example:

```
> RUN ↵
BASE ? = 2 0 ↵
HEIGHT ? = 8 0 ↵

BASE  =  2 0  HEIGHT =  8 0
AREA  =  8 0 0
```

- The CLS in line 5 clears the display panel.
- Lines 10 and 15 display messages which tell what you should enter.
- Line 50 provides a delay by forcing the FP-200 to count from 1 to 1000. The display remains unchanged during this time (about 5 seconds). If you change the "**GO TO 5**" to "**GO TO 10**", the program can start the next computation without erasing the display.

CLS Clears the display panel.

INPUT "message", A

The INPUT statement can display a message, indicated by enclosing it within a pair of quotation marks, before asking for keyboard input.

: You can create a multiple statement line by using a colon.

For example: **35 PRINT**
40 PRINT A
can be entered as: **35 PRINT: PRINT A** on a single line.

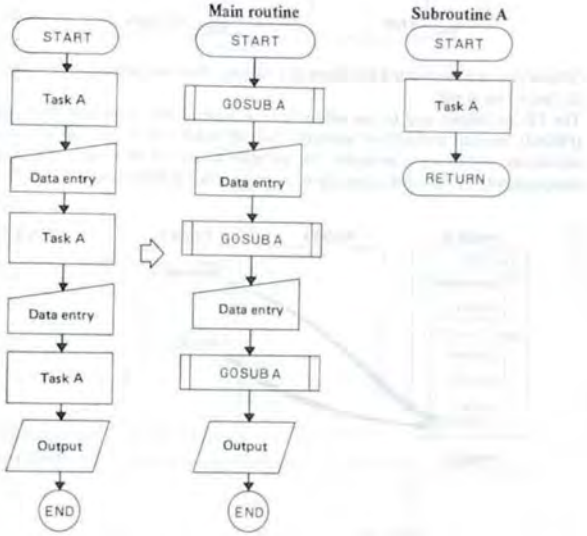
PRINT What will happen if you execute the statement **35 PRINT** which is a special form of the PRINT statement? It only changes to the next line, without any data output (i.e., it inserts a blank line). The PRINT statement in line 60 of the improved program inserts a blank line.

(2) The Subroutine Concept

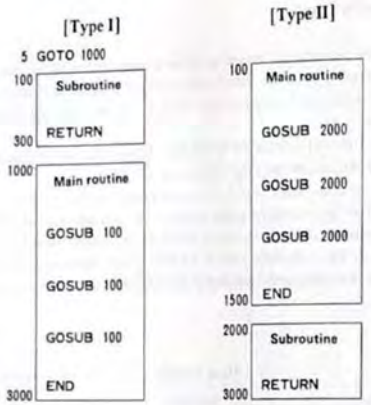
Most larger programs will consist of several sections which provide functions which are used several times throughout the program. Each such commonly used program section is written as an independent program called a subroutine, while the main body of the original program is called the main routine.

What is a subroutine, then? In order to learn this, let's examine closely how the BASIC **GOSUB** and **RETURN** statements are used in the following example.

A section which may appear many times in the program can be extracted from the main program flow and written as an independent program called a subroutine. When the subroutine needs to be executed during the course of the main program, control is transferred to it by a **GOSUB** command. The subroutine has a **RETURN** command at its end, which returns control to the statement immediately following the **GOSUB** command that caused the jump to the subroutine.

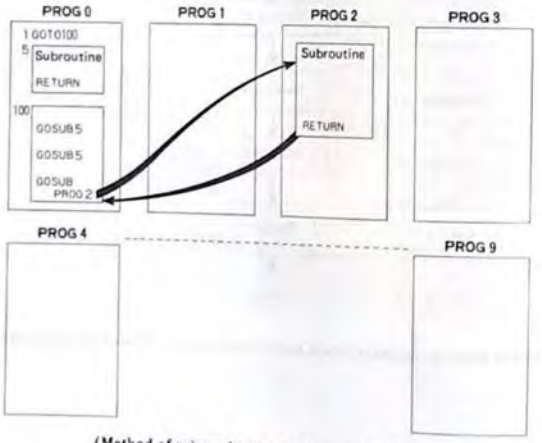


The following diagram illustrates how a main routine and one of its subroutines are associated:



Subroutines may be located anywhere in a program. However, they are normally collected at the beginning or end.

The FP-200 allows you to use subroutines in another way, also. It has ten program areas (PROG0 through PROG9) in memory, each of which can be thought of as being like a warehouse containing a program. The program in any of these areas can be either used independently or executed indirectly by a program in a different area:



(Method of using subroutines unique to the FP-200)

- **Handling data – 1**

READ, DATA, and RESTORE statements

The word "computer" will probably remind you of data processing. Let's examine here how our FP-200 can deal with data by using simple examples.

Write a program that functions as a dictionary to help you to memorize the following ten most important FP-200 BASIC statements:

(1) CLEAR	clear screen
(2) PASS	set a password
(3) PRINT	display
(4) READ	read data
(5) DATA	specify data
(6) INPUT	input data
(7) LIST	list program
(8) LOAD	load program
(9) SAVE	save program
(10) QUAD	draw a rectangle

```
 10 RESTORE
 20 FOR I= 1 TO 10
 30 READ A$,B$
 40 PRINT A$,B$
 50 INPUT N$
 60 NEXT I
 70 END
 80 DATA CLEAR,clear screen
 90 DATA PASS,set a password
100 DATA PRINT,display
110 DATA READ,read data
120 DATA DATA,specify data
130 DATA INPUT,input data
140 DATA LIST,list program
150 DATA LOAD,load program
160 DATA SAVE,save program
170 DATA QUAD,draw a rectangle
```

The program is

10 This initializes the program to read the data from the beginning.

20 Read and display ten pairs of data through 10 loops.

30 The READ statement reads the data in the DATA statements in its order of appearance.

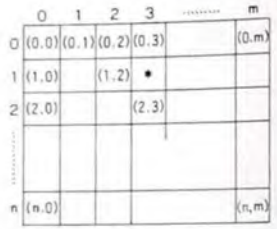
50 This INPUT statement temporarily stops the program before proceeding to the next statement, so that you can have time to watch the display. Otherwise, it displays the ten command words and their meanings so fast that you cannot even identify a single character. With this single statement, you can watch each command as long as you wish until you press [EXE].

- **Handling data – 2**

Array variables

When a larger volume of data has to be processed, however, the previous method is too laborious and time-consuming.

To eliminate this inconvenience, the FP-200 provides a method for storing and handling data in which the data is contained within structure called an array. An array can have up to 3 dimensions. The name given to an array is called the array name or identifier.



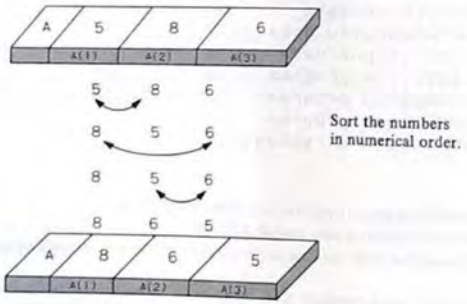
Data stored in such an array can be brought out for processing as required.

The "•" is in the partition two rows down from the top and four columns from the left end, and its location is described as (1, 3) because the rows and columns are numbered sequentially starting with 0.

Now, as an example, let's write a program that sorts numbers by using an array.

Purpose:

(1) Enter three numbers.
(2) Sort them into descending order.
(3) Display the result.



Sort the numbers in numerical order.

The program:

```
 10 DIM A(3)
 20 FOR I=1 TO 3
 30 INPUT A(I)
 40 NEXT I
 50 FOR I=1 TO 2
 60 FOR J=I+1 TO 3
 70 IF A(I)>=A(J) THEN 110
 80 M=A(I)
 90 A(I)=A(J)
100 A(J)=M
110 NEXT J
120 NEXT I
130 FOR I=1 TO 3
140 PRINT A(I),
150 NEXT I
160 END
```

The program is explained below:

10 Declares that the array will be used in the program.

20 | Inputs three numbers.
40

50 ○
60 ○
70 Compares the next number.
80 | Replaces the numbers. M → A(I), A(I) → A(J), A(J) → M
100
110 ○
120 ○

M A(I) A(J)
Stores the contents of A (I) in M. —— **80**
Stores the contents of A (J) in A (I). —— **90**
Stores the contents of M in A (J). —— **100**

130 | Displays the sorted numbers.
150

Now let's try the program by entering "**RUN** ↵ ".

We have studied several frequently-used BASIC programming techniques. Next we will study statistics, which are frequently used in business-oriented programming. Mathematical expressions that appear in statistics are too complicated for most beginners, so that people are sometimes discouraged from using computers.

The FP-200 dedicated statistical functions so that nobody needs to be bothered with such complexity. Let's study how to use them.

3. Statistics processing

We often compute totals, averages and the like in everyday life. Once the term statistics is used, such computations suddenly sound very difficult. This is probably because of our vague impression that statistics involves difficult computations on a huge volume of data. The FP-200 has many statistical functions which allow you to obtain such values as standard deviation, etc. without any mathematical knowledge whatsoever.
Aided by these functions, you can more easily than ever obtain statistical computations which used to be very troublesome.
Let's study the statistics processing facilities of the FP-200 by actually using them on simple examples.
We have the following eight numbers:

```
60.20.70.10.40.30.80.50
```

We obtain their total, mean, and standard deviation by using the statistical facilities.
Enter the following program:

```
PROG 1 ↵
NEW ↵

 10 STAT CLEAR
 20 FOR I=1 TO 8
 30 READ X
 40 STAT X
 50 NEXT I
 60 PRINT"NUMBER OF DATA";CNT
 70 PRINT"TOTAL OF DATA";SUMX
 80 PRINT"AVERAGE OF DATA";MEANX
 90 PRINT "STANDARD DEVIATION";SDX
100 END
110 DATA 60,20,70,10
120 DATA 40,30,80,50
```

Computes the statistics for the eight numbers.

The STAT CLEAR and STAT in this program are called statistics functions. If these statistics functions (and the FP-200 also provides the CNT, SUMX, MEANX and SDX functions) were not available, the computations would have to be programmed in the BASIC language as shown below.

```
10 CLEAR
20 SUM=0
30 MUL=0
40 N=0
50 FOR I=1 TO 8
60 READ X
70 N=N+1
80 SUM=SUM+X
90 MUL=MUL+X*X
100 NEXT I
110 SD=SQR((N*MUL-SUM*SUM)/(N*(N-1)))
120 AV=SUM/N
130 PRINT"NUMBER OF DATA";N
140 PRINT"TOTAL OF DATA";SUM
150 PRINT"AVERAGE OF DATA";AV
160 PRINT"STANDARD DEVIATION";SD
170 END
180 DATA 60,20,70,10
190 DATA 40,30,80,50
```

The program looks very complicated, compared with the previous one. This tells us that even complicated statistics on a large volume of data can be obtained simply with the aid of the statistical functions.
The two programs will give the following results; the functions provide a higher data precision:

Result using the conventional BASIC program.

```
NUMBER OF DATA 8
TOTAL OF DATA 360
AVERAGE OF DATA 45
STANDARD DEVIATION 24.4949
```

Result using the program which uses the statistics functions.

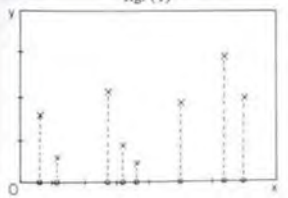
```
NUMBER OF DATA 8
TOTAL OF DATA 360
AVERAGE OF DATA 45
STANDARD DEVIATION 24.49489742783178
```

Now, let's study the statistics functions used in the sample program.

STAT <X-data> [, <Y-data>] —— Statistics Data Entry

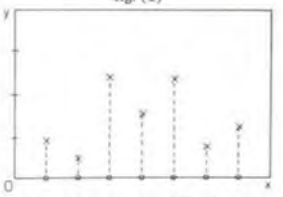
This command performs a statistical processing on the data entered as X- and Y-data. Two types of statistics are available, as shown in figures (1) and (2), (2) is used in the above example.

fig. (1)



Irregularly spaced X values

fig. (2)



Regularly spaced X values

STAT CLEAR —— Initializing Statistical Facilities

This command readies the statistical facilities for use and it must always be executed before any statistics processing is performed.

CNT —— Processed Data Counter

This function gives the number of data points processed by the STAT command. It counts them one by one when they are read by the STAT command.
CNT gave a value of 8 in the above example.

SUMX —— Summing X Data

This function gives the sum of the X data processed by the STAT command.
In the above example, SUMX has a value of 360, which is the sum of 60+20+70+10+40+30+80+50.

MEANX —— Averaging X Data

This function gives the mean value of the X data processed by the STAT command.
In the above example, it had a value of 45, which is the mean of the eight numbers: 360/8 = 45.

SDX —— X Data Standard Deviation

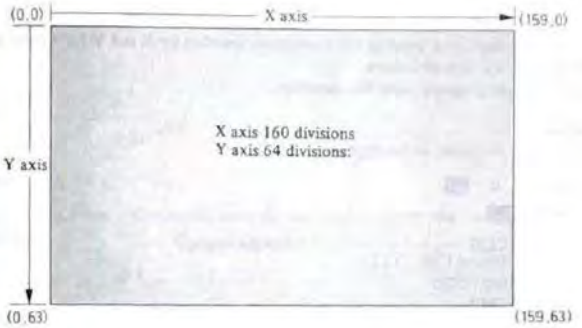
This function gives the standard deviation of the X data processed by the STAT command.
In the above example, it gave a value of 24.49489 ..... which was computed by the expression:

$$\sqrt{\frac{8 \times \Sigma x^2 - (\Sigma x)^2}{8 \times (8-1)}} = 24.49489742783178$$

A complicated computation like the standard deviation can be accomplished by a single function.
The FP-200 has other statistical functions, which are detailed in a separate reference manual.

4. Graphics

We often use a table or graph to display and compare various quantities.
The FP-200 provides a graphic display panel consisting of a large matrix of 64 x 160 dots so that it can accurately reflect even small differences among several quantities.
Before proceeding to study graphics, let's introduce the concept of "coordinates", which needs to be understood as a basis for graphics.
The FP-200 provides the following coordinate system:



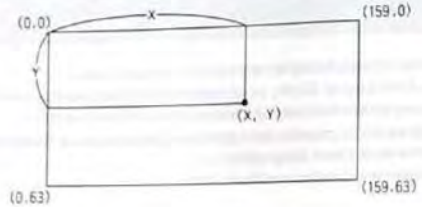
The FP-200 display panel can be looked upon as a sheet of graph paper which has 64 vertical and 160 horizontal divisions.
The FP-200 LCD display provides a coordinate plane which has 160 horizontal (x-axis direction) and 64 vertical (y-axis direction) divisions. The top left of the display is the origin (0, 0). As you move away from the origin toward the right the x coordinate increases, and moving from the origin downward the y coordinate increases. For example:

(50 , 23) —— Look upon the two numbers separated by a comma "," and enclosed within parentheses as the intersection of the 50th vertical and 23rd horizontal dividing lines; 50 is called the x coordinate and 23 is called the y coordinate.

X axis Y axis

Now, let's study the graphics statements.

DRAW (X, Y) —— Drawing a Point or Straight Line



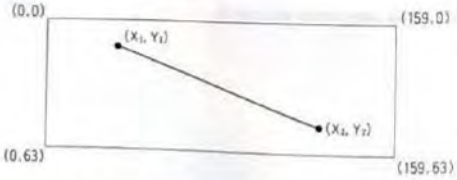
This command displays a point at the coordinates specified by X and Y (or simply at the coordinates (X, Y)) as shown above.
Let's write simple programs using this statement.

(Example)
Draw a point at the center of the display.

```
PROG ⊘ ↵
NEW ↵
 10 CLS ←———————— Clears the display
 20 DRAW(79,31)
 30 GOTO30
 40 END
```

If you press the **BREAK** or **STOP** key, the program will stop and FP-200 will wait for a new command entry.
To run the program, press the **RUN** key. A point will be drawn at the center, as shown below.

DRAW $(X_1, Y_1) - (X_2, Y_2)$

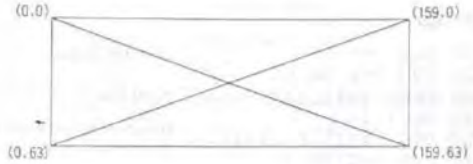


This statement draws a straight line which begins at the coordinates (X_1, Y_1) and ends at the coordinates (X_2, Y_2).

(Example)
Draw the two diagonal lines on the display.

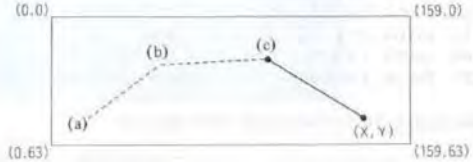
```
NEW ↵
 10 CLS ←———————— Clears the display.
 20 DRAW(0,0)-(159,63) ← Draws a diagonal line from the top left
 30 DRAW(159,0)-(0,63)   corner to the bottom right corner.
 40 GOTO40 ↖ Draws a diagonal line from the top right
 50 END        corner to the bottom left corner.
```

Run the program. It will display the diagonal lines as shown below:



DRAW – (X, Y)

This statement draws a straight line from the last point drawn to the coordinates (X, Y) as shown below.

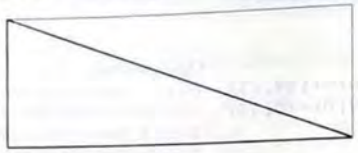


Two line segments have already been drawn from point (a) to (b) and from (b) to (c) on the display. This statement draws a new line from the last point (C) to the coordinates (X, Y).

(Example)
Draw a large triangle that fills up the display.

```
NEW ↵
 10 CLS ←———————— Clears the display.
 20 DRAW(0,0)-(0,63) ←— Draws a line from the top left corner to the
 30 DRAW(0,63)-(159,63) ↖ bottom left corner.
 40 DRAW-(0,0) ←———— Draws a line from the bottom left corner to
 50 GOTO50          the bottom right corner.
 60 END ←———————— Draws a line from the bottom right corner
                    back to the top left corner.
```

RUN the program. The following triangle will be drawn on the display.

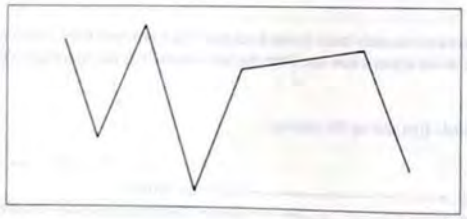


This time let's write a program that draws a line graph on the panel as an application of the DRAW command.

NEW ↵

```
 10 CLS                          ← Clears the display.
 20 FOR I=1 TO 9                 ┐
 30 READ X(I),Y(I)               ┘ Reads data.
 40 NEXT I
 50 DRAW(X(1),60-Y(1))           ← Determine the point from which the line graph begins.
 60 FOR I=2 TO 9                 ┐
 70 DRAW-(X(I),60-Y(I))          ┘ Draws the line graph.
 80 NEXT I
 90 GOTO 90                      ← Stop the program when the STOP or BREAK key.
100 END
110 DATA 20,50,35,20             ┐
120 DATA 50,55,65,5              │
130 DATA 80,45,125,53            │ Data
140 DATA 110,50,125,53           │
150 DATA 140,15                  ┘
```

RUN the program. The following graph will be displayed:



DRAWC —— Erasing a Specified Point or Straight Line

This command issued in the same way as the DRAW command except that it erases – instead of drawing – the existing specified point or straight line.

(Example)
First draw lines for the whole screen, and then perform the DRAWC command to erase diagonal lines on it.

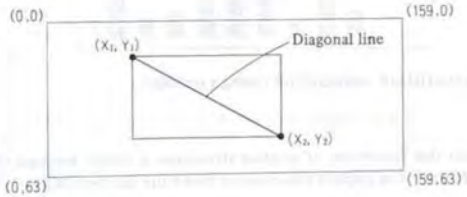
NEW ↵

```
 10 CLS                          ← Clears the display.
 20 FOR I=0 TO 63                ┐
 30 DRAW(0,I)-(159,I)            ┘ Draws lines.
 40 NEXT I
 50 DRAWC(0,0)-(159,63)          ┐
 60 DRAWC(159,0)-(0,63)          ┘ Draws the diagonal lines in reverse video.
 70 GOTO 70                      ← Stops the program when the STOP or BREAK key is pressed.
 80 END
```

RUN the program. The following display will appear.

QUAD —— Drawing a Rectangle

QUAD $(X_1, Y_1) - (X_2, Y_2)$



This statement draws a rectangle whose diagonal is the line joining the specified coordinates (X_1, Y_1) and (X_2, Y_2), as follows:

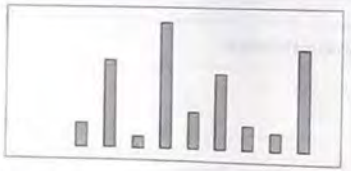
(Example)
Draw a frame which just fits in the display.

```
10 CLS ———— Clears the display.
20 QUAD (10, 10) – (149, 53) ———— Draws the frame.
30 GOTO 30 ———— Stops the program when the STOP or BREAK key is pressed.
40 END
```

Now, let's write a program using the QUAD command that draws a bar graph on the display.

```
NEW ↵
 10 CLS ———————————————— Clears the display.
 20 FOR I=1 TO 9
 30 READ X(I),Y(I)        Reads the data.
 40 NEXT I
 50 FOR I=1 TO 9
 60 QUAD(X(I),Y(I))-(X(I)+5,60)   Draws the bar graph.
 70 NEXT I
 80 DRAW(10,4)-(10,60)
 90 DRAW-(155,60)
100 GOTO100 ———————————— Stops the program when the STOP or
110 END                   BREAK key is pressed.
120 DATA 25,50,40,20
130 DATA 55,55,70,5
140 DATA 85,45,100,30     Data
150 DATA115,50,130,53
160 DATA145,15
```

RUN the program. The following bar graph should appear on the display:



(We also have the QUADC command for erasing a rectangle.)

- To conclude this description of graphics statements, a sample program which uses the statistical facilities and graphics statements is shown and discussed in detail below:

(Sample) – Student performance

This program accepts examination grades in a given subject for the students in a class, computes the average, displays the deviation values of the individual students and generates a bar graph of their grades:

```
 10 REM**READING DATA**
 20 CLEAR
 30 READ NUM,DAY$,SUB$
 40 DIM NAM$(NUM),MAR(NUM),DEV(NUM)
 50 FOR I=1 TO NUM
 60 READ NAM$(I),MAR(I)
 70 NEXT I
 80 REM **CALCULATING DEVIATION**
 90 STAT CLEAR
100 FOR I=1 TO NUM
110 STAT MAR(I)
120 NEXT I
130 FOR I=1 TO NUM
140 DEV(I)=10*(MAR(I)-MEANX)/SDXN+50
150 NEXT I
160 REM**EXPRESSING DATA**
170 CLS
180 LOCATE 0,0:PRINT"SUB:   ";SUB$
190 PRINT"DAY:   ";DAY$
200 PRINT"AVE:  ";MEANX
210 PRINT"--------------------"
220 FOR I=1 TO NUM
230 FOR J=4 TO 7
240 LOCATE 0,J
250 NEXT J
260 LOCATE 0,4:PRINT"(NUM)";I
270 PRINT"(NAM)  ";NAM$(I)
280 PRINT"(MAR)";MAR(I)
290 PRINT"(DEV)";DEV(I);
300 A$=INKEY$:IF A$="" THEN 300
310 NEXT I
320 REM**EXPRESSING GRAPH**
330 GOSUB 450
340 A=0
350 FOR I=1 TO NUM
360 A=A+1:IF A>=7 THEN A=1:GOSUB 450
370 LOCATE 0,A:PRINT I
380 X=MAR(I)
390 Y=A*8+2
400 QUAD(36,Y)-(36+X,Y+3)
410 A$=INKEY$:IF A$="" THEN 410
420 NEXT I
430 END
440 REM**ESTABLISHING SCREEN**
450 CLS
```

```
460 DRAW(36,4)-(36,60):DRAW-(136,60)
470 FOR J=46 TO 136 STEP 10
480 DRAW(J,57)-(J,60)
490 NEXT J
500 S=INT(MEANX):DRAW(36+B,4)-(36+B,60)
510 RETURN
520 REM**DATA**
530 DATA 10,4/12,MATH
540 DATA A,80
550 DATA B,45
560 DATA C,50
570 DATA D,90
580 DATA E,55
590 DATA F,78
600 DATA G,98
610 DATA H,63
620 DATA I,85
630 DATA J,77
```

Detail explanations on the program

1.	30– 70 :	Initializing the program and reading data
2.	90–150 :	Processing statistics and computing deviations
3.	170–310 :	Displaying results
4.	370–410 :	Generating bar graph
5.	450–500 :	Initializing the display
6.	530– :	Data

30 : Reads the number of students in the class, the date of the examination and the name of the subject.
40 : Defines the arrays that store the names of the students and their grades and deviation values.
50– 70 : Reads the names and grades.

90 : Initializes the statistical facilities.
100–120 : Processes the data.
130–150 : Computes and stores the deviation values for the students.

170 : Clears the display.
180 : Displays the subject name on the first line.
190 : Displays the test date.
200 : Displays the mean of the grades.
230–250 : Clears lines 5 through 8 on the display.
260 : Displays the student number on the fifth line.
270 : Displays his name.
280 : Displays his grade.
290 : Displays his deviation value.
300 : Stops the program until any key is pressed.

370 : Displays the student number.
380 : Determines the height of the bar.
390 : Determines the horizontal position of the bar.
400 : Displays the bar.
410 : Stops the program until any key is pressed.

450 : Clears the display.
460 : Displays the horizontal and vertical axes of the graph.
470–490 : Displays the divisions along the vertical axis.
500 : Displays the horizontal average line.

530 : Data

Variable list

NUM	: Number of students in the class
DAY$	: Date of the examination
SUB$	: Subject
NAM$( )	: Student names
MAR( )	: Grades
DEV( )	: Deviation values

5. How to modify and add data

- Modifying data

The data appear from line 520 in the previous program.

```
520 REM**DATA**
530 DATA 10,4/12,MATH
540 DATA A,80
550 DATA B,45
560 DATA C,50
570 DATA D,90
580 DATA E,55
```

Line 530 contains the number of students, data and subject:

```
530 DATA 10,4/12,MATH
         |   |   |
         |   |   └─Subject
         |   └─Date
         └─Number of students (i.e., number of data items)
```

Lines 540 through 580 contain the student names and grades in the following format:

```
540 DATA A,80
550 DATA B,45
560 DATA C,50
570 DATA D,90
580 DATA E,55
         |  |
         |  └─Grade
         └─Student name
```

Modifying data

Any value in a data statement can be modified by editing the data. Let's change the mark of student E from 55 to 57 for example:

PF0

```
>EDIT 580 ↵                    Specifies line 580.
580 DATA, E, 55–               ←7↵ Modified data

580 DATA, E, 57–
```

The data was correctly modified. All data can be modified in this way.

- Adding Data

If you want to add grade data, first increase the number of students in line 530, and then add the student names and grades on subsequent unused lines (from line 640 on, for example).
If the following data is added, increase the number of students in line 530 from 10 to 12.

```
530 DATA 12, 4/12, MATH
```

And then add the following two DATA statements on lines 640 and 650:
This successfully accomplished the necessary modification and data addition.

```
640 DATA K, 60 ↵
650 DATA L, 47 ↵
```

Concluding Note to Chapter 3

You have now studied the major BASIC facilities and features. You probably feel now, as your studies have progressed, that BASIC is not so difficult as to discourage you; its syntax is rather simple and very clear-cut.
To keep improving your BASIC programming, however, you will have to continue to use what we have covered in this chapter.
Now, let's see how to use CETL and BASIC together.

CHAPTER 4
CETL and BASIC

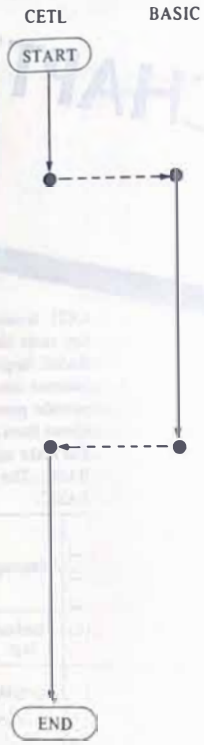
CETL is easy to use, but it provides very powerful capabilities for everyday tasks like tabulation that involve large quantities of data, while the BASIC language allows high-level scientific calculations and complicated business computations. The FP-200 has both of them built in, and they provide powerful capabilities independently. However, the FP-200 also allows them to be used in combination. This can extend their advantages and make up for their disadvantages. This chapter will explain CETL and BASIC. The following table summarizes comparisons between CETL and BASIC:

CETL	Advantages	• Can simply tabulate data. • Allows recomputations after data modification. • Requires no programming. • Can be operated with a limited number of commands.
CETL	Disadvantage	• Lacks flexibility in building systems.
BASIC	Advantage	• Allows programming of complicated expressions and complex logic and graphics.
BASIC	Disadvantages	• Requires more time for programming. • Requires practice to write programs with general usefulness. • Requires the users to be familiar with many types of statements.

This comparison may leave some people confused as to which of them they should use. When they have an FP-200 in their hands, they will probably select CETL since it has fewer commands and can be used for daily applications.
An important way to make more efficient use of CETL is to link CETL to BASIC.

1. Linkage between CETL and BASIC

Linking CETL to BASIC will make CETL even more useful. As an introduction to a detailed discussion of this subject, let's show the linkage between CETL and BASIC:



(1) Process data with CETL.

(2) When you need help from BASIC, pass the data to BASIC.

(3) A BASIC program processes the data.

(4) After being processed by BASIC, the data is returned to CETL.

(5) CETL resumes processing with the returned data.

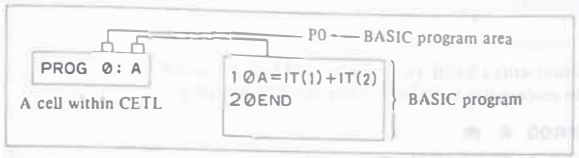
(6) End of CETL Processing.

This method of switching from CETL to BASIC and then from BASIC to CETL is called linkage.

How is program execution actually transferred from one program to another? Since CETL and BASIC are in the different area, we first need to determine how to transfer the execution sequence and how to pass data from one to the other.

- **Transfer data from CETL to BASIC**

We use the form "PROG 0; A" in the CETL program, for example, where the number 0 designates the area in which the BASIC program resides (one of such areas 0 through 9) and A is a variable which receives the result returned from the BASIC program.



- **Transfer data from BASIC to CETL**

When the BASIC program terminates, the program control returns to the CETL where the BASIC program was called. At this time, the result given by the BASIC program is placed in the variable A.

Now, let's write programs.

Layout the CETL program file we use as follows:

Title DEMO

Determine a data size of 6 digits for all the data.

Slide the mode switch to CETL.

Display	Key entry
> _	N ↵ Instruct CETL to create a new table.
File Name? _	DEMO↵ Title.
Number of Rec.? _	1 ↵
Number of Item? _	3 ↵
Item 1 Name? _	DATA 1 ↵
Type(N/S)? N_	↵
Expression? _	↵
Format? _	######

(The key entries for DATA 2 and DATA 3 are omitted.)

1–1 ?	100↵ Assigns 100 to cell 1 – 1.
1–2 ?	200↵ Assigns 200 to cell 1 – 2.
1–3 ?	PROG 0: A↵ Links this cell to the BASIC program in program area 0.

With these key entries, CETL creates the following table (or file):

DEMO

DATA 1	DATA 2	DATA 3
100	200	PROGO: A

We will next write a BASIC program which will be linked with the above CETL file.
Slide the mode switch to BASIC and enter the following program:

```
PROG 0 ↵
10 A=IT(1)+IT(2)↵
20 END
```

Let's run the program:
Change the mode switch back to CETL and enter:

Display	Key entry
> _	T↵
Rec. ? _	.↵
Item? _	.↵
Printer(Y/N)? _	N↵

You should obtain the following results:

DATA1	DATA2	DATA3
100	200	300

Check that DATA 3 is the sum of DATA 1 and DATA 2.

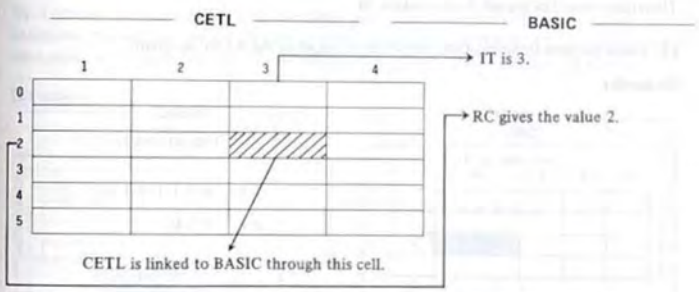
2. CETL management functions

IT (1) and IT (2) used in the sample BASIC program are called CETL management functions.
The FP-200 provides the following CETL management functions:

RC
IT
RC (n)
IT (n)
FL (l, m, n)
SUMRC (m, n)
SUMIT (m, n)
(l, m, and n are numbers or expressions.)

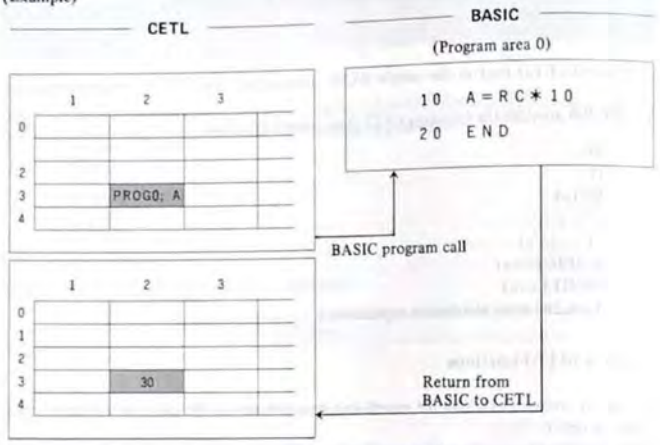
(1) RC and IT Functions

RC and IT, respectively, give the record and item numbers of the cell where is linked to a BASIC program.
These functions are mainly used as useful pointers for computing the cell position within the BASIC program:



RC: Gives the record number of the cell where is linked to the BASIC program.

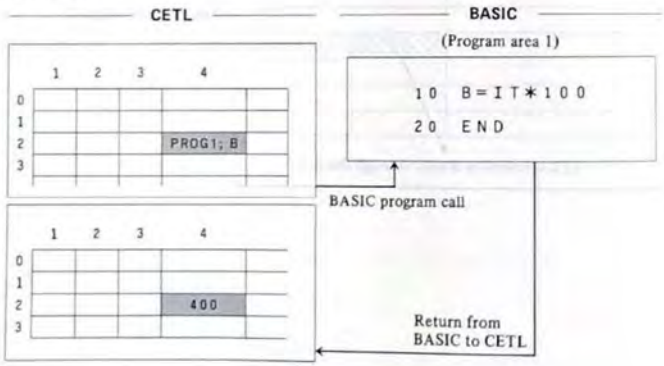
(Example)



Because the RC function gives 3, the statement A = RC x 10 means A = 3 x 10. Therefore, item 2 of record 3 will contain 30.

IT: Gives the item number of the cell where is linked to the BASIC program.

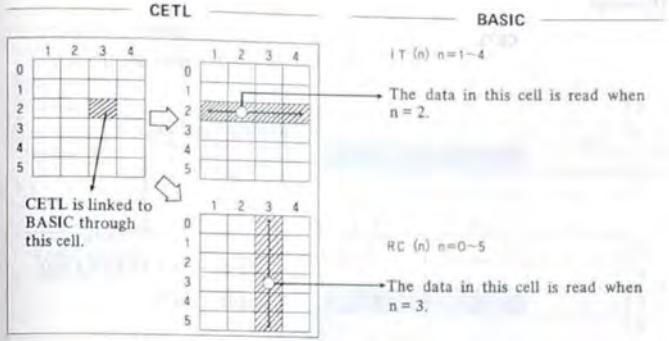
(Example)



Because the IT function gives item number 4, the statement B=IT x 100 means B=4 x 100. Therefore, item 4 of record 2 will contain 400.

(2) RC (n) and IT (n) functions

With the RC (n) and IT (n) functions, you can read the contents of any desired cell within the CETL table by varying the value of n.



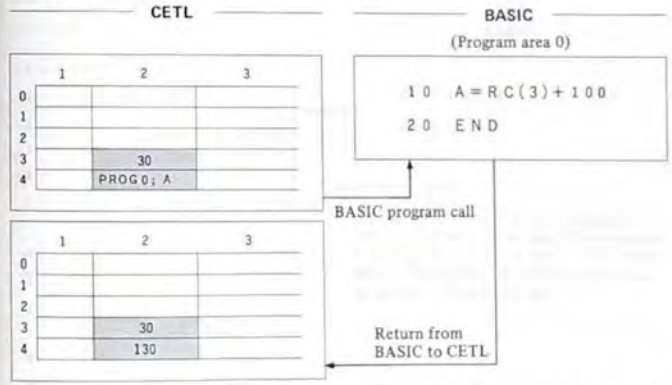
With these capabilities, complicated totals are possible depending on how you select the value of n. If you write a BASIC program for example:

```
100 A=0 : FOR N= 1 TO 5 STEP 2 : A=A+RC(N) : NEXT : END
```

The odd-numbered items in the CETL table can be totaled.

RC (<record-specification>): You can fetch the contents of the cell in the current file designated by the <record-specification> with this variation of the RC (n) function. (The item number is the one corresponding to the cell through which CETL links to BASIC.)

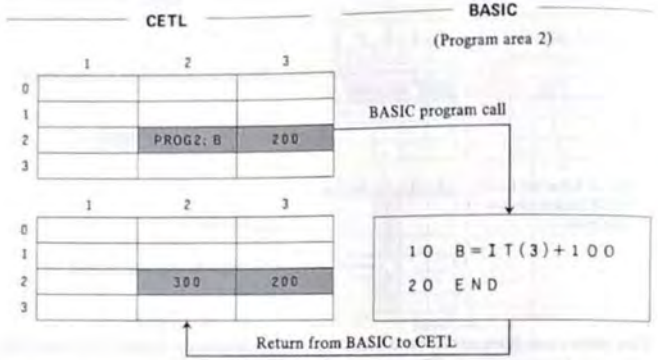
(Example)



Variable A will contain 30 + 100 = 130, because RC (3) gives the contents of the data cell which is item 2 in record 3.
Consequently, the cell which is item 2 of record 4 will contain 130.

IT (<item-specification>): You can fetch the contents of the cell in the current file designated by the <item-specification> with this variation of the IT (n) function. (The record number is the one corresponding to the cell through which CETL is linked to BASIC.)

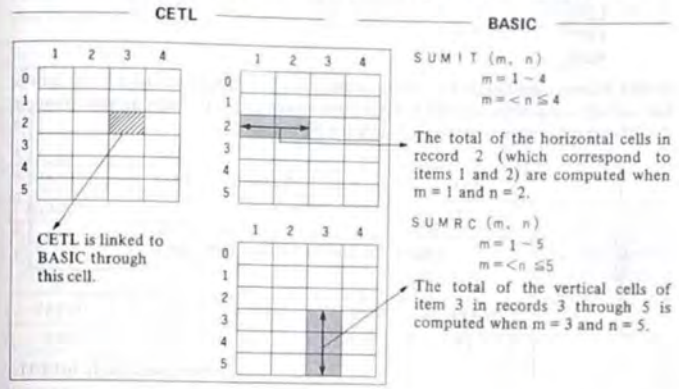
(Example)



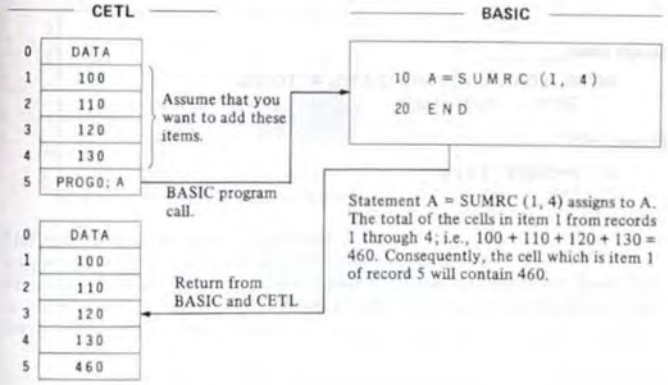
200 + 100 is assigned to variable B, because IT (3) gives the contents of the cell which is item 3 of record 2.
Consequently, the cell which is item 2 of record 2 will contain 300.

(3) SUMRC (m, n) and SUMIT (m, n) functions

By using the SUMRC (m, n) and SUMIT (m, n) functions, you can obtain the horizontal and vertical totals within a given range (of items and records) by varying the values of m and n.



These functions are most useful in exploiting the linkage between CETL and BASIC. With these functions, you can easily and rapidly obtain any horizontal and vertical totals of tables which can have long records or a large number of records. Because the computed totals are also immediately available for use in the BASIC Program, tables can be processed faster.
SUMRC (<record-specification-1>, <record-specification-2>):
Obtains the total of the cells in the current item in the records between <record-specification-1> and <record-specification-2> in the current file. The item number is the item number of the linkage cell.



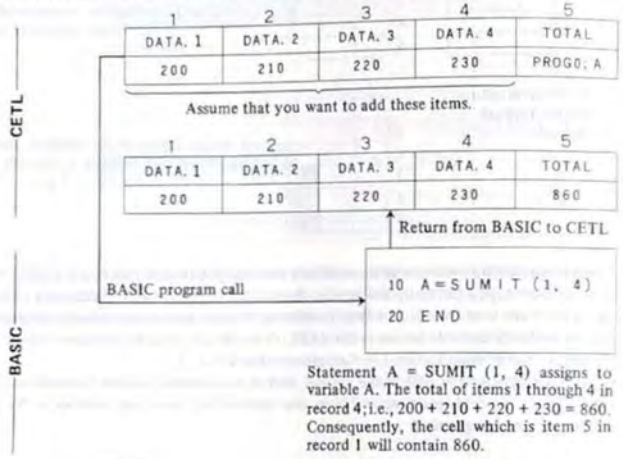
Sample result:

```
DATA
 100
 110
 120
 130
 460
```

Program in P0

```
10 A=SUMRC(1,4)
20 END
```

SUMIT (<item-specification-1>, <item-specification-2>): Obtains the total of the cells in the current record from the items between <item-specification-1> and <item-specification-2>. The record number is the record number of the linkage cell.



Statement A = SUMIT (1, 4) assigns to variable A. The total of items 1 through 4 in record 4; i.e., 200 + 210 + 220 + 230 = 860. Consequently, the cell which is item 5 in record 1 will contain 860.

Sample result:

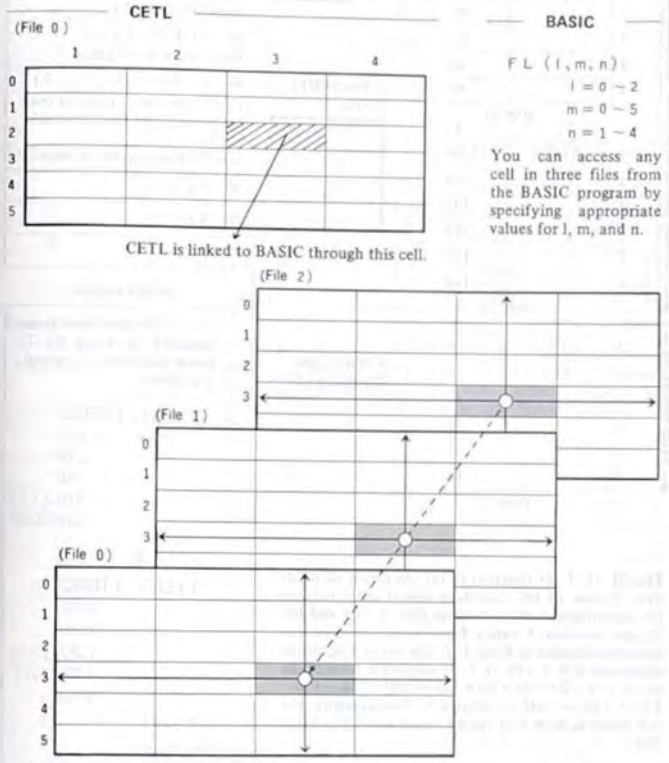
```
DATA.1DATA.2DATA.3DATA.4 TOTAL
   200    210    220    230   860
```

Program in P0

```
10 A=SUMIT(1,4)
20 END
```

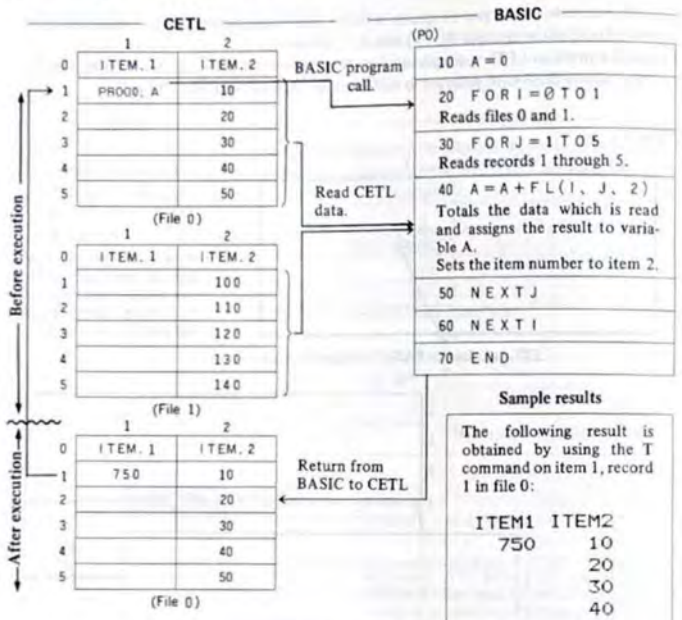

(4) FL Function

The FL function allows you to specify a file in addition to providing the capabilities that combine those of the previous RC ( ) and IT ( ) functions.
Unlike the previous CETL management functions, which can only read CETL data, the FL function allows the BASIC program to write directly into any desired CETL cell.



This function may first appear complicated to use, but it is frequently used in BASIC programs linked with CETL, and it is important to master its use.
FL ([<file-specification>], [<record-specification>], [<item-specification>]): Reads from or writes to the cell designated by <item-specification> and <record-specification> in the file designated by <file-specification>.

(A) Reading data from cells in CETL files



Sample results

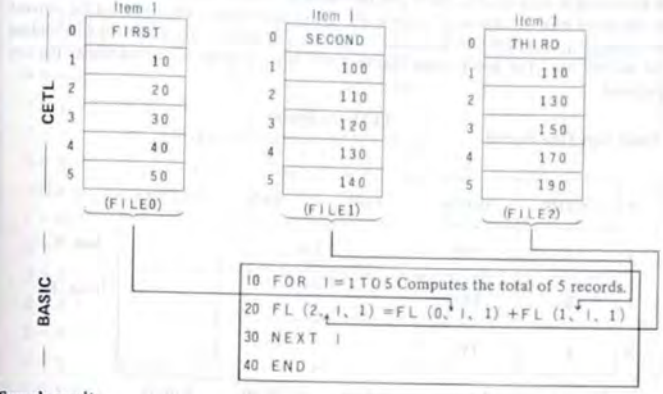
The following result is obtained by using the T command on item 1, record 1 in file 0:

```
ITEM1  ITEM2
  750     10
          20
          30
          40
          50

(FILE 0)

ITEM1  ITEM2
         100
         110
         120
         130
         140

(FILE1)
10 A=0
20 FOR I=0 TO 1
30 FOR J=1 TO 5
40 A=A+FL(I,J,2)
50 NEXT J
60 NEXT I
70 END
 (Program in P0)
```

The FL (I, J, 2) function in the statement 40 reads item 2 from all the records in files 0 and 1 because the record-specification J varies from 1 to 5 and the file-specification I varies from 0 to 1, while the item-specification is fixed at 2. The entire assignment statement A = A + FL (I, J, 2) assigns the total of the items: (10 + 20 + 30 + 40 + 50) + (100 + 110 + 120 + 130 + 140) = 750 to variable A. Consequently, the cell which is item 1 of record 1 in file 0 will contain 750.

(B) Writing data to cells in CETL files

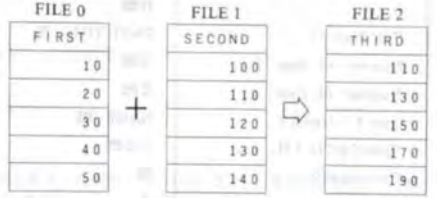
(Items are processed differently than in the previous example (1))



Sample results

```
(FILE0)
FIRST
   10
   20
   30
   40
   50
(FILE1)
SECOND
  100
  110
  120
  130
  140
(FILE2)
THIRD
  110
  130
  150
  170
  190
```

The statement 20 FL (2, I, 1) = FL (0, I, 1) + FL (1, I, 1) reads and totals items 1 of records 1 through 5 in files 0 and 1, and then assigns the result to item 1 of the corresponding records 1 through 5 in file 2 because the record-specification I varies from 1 to 5 while the item-specification is fixed at 1.



(Program in P0)

```
10 FOR I=1 TO 5
20 FL(2,I,1)=FL(0,I,1)+FL(1,I,1)
30 NEXT I
40 END
```

Note!
The FL function is permitted to be placed at the left side of the assignment statement only when using it in the BASIC program and executing it at the BASIC mode.

(5) An example of using linkage with BASIC

A supermarket store employs three part time workers. Assume that the wage of each worker is calculated based on his work hours at the end of each month. 10 percent of the amount in excess of a basic ¥70,000 deduction is subtracted from the total amount as withholding for income tax. The hourly wage rate is ¥500. Write a program that calculates the net payment.

CETL file layout

Title: **Part Time Payroll**

→IT (2)×500 →PROG 0:A →IT(3)−IT(4)

	1	2	3	4	5
0	NAME	HOUR	PAY	TAX	NET PAY
1	A	200			
2	B	190			
3	C	195			

(Rec : 3, Item : 5)

8 characters | 4 digits | 9 digits | 7 digits | 9 digits

A comma must be inserted in these numbers every 3 digits.

Slide the mode switch to CETL and enter the following:

Display	Key entry	
> _	N↵	
File Name? _	PART TIME ↵	
Number of Rec. ? _	3↵	
Number of Item? _	5↵	
Item 1 : Name? _	NAME ↵	
Type (S/N) ? N_	⊖S↵	
Expession? _	↵	
Format ? _	&␣␣␣␣␣␣&	
Item 2: Name? _	HOUR ↵	
Type(S/N) ? N _	↵	
Expression? _	↵	
Format ? _	#####↵	
Item 3: Name? _	PAY ↵	
Type(S/N) ? N _	↵	
Expression? _	IT(2)*500↵	Gross wages are given by "work hours × 500".
Format ? _	#,###,### ↵	
Item 4: Name? _	TAX ↵	
Type(S/N) ? N _	↵	
Expression? _	PROG0: A↵	Taxes are calculated in the BASIC program.
Format ? _	###,### ↵	
Item 5: Name? _	NET PAY ↵	
Type(S/N) ? N _	↵	
Expression ? _	IT(3)−IT(4)↵	Net pay amounts are given by "gross wages − tax".
Format ? _	#,###,### ↵	
1 − 1 ? _	A ↵	
1 − 2 ? _	200↵	
1 − 3 ? _	↵	
1 − 4 ? _	↵	
1 − 5 ? _	↵	
2 − 1 ? _	B ↵	
2 − 2 ? _	190↵	
2 − 3 ? _	↵	
2 − 4 ? _	↵	
2 − 5 ? _	↵	
3 − 1 ? _	C ↵	
3 − 2 ? _	195↵	
3 − 3 ? _	↵	
3 − 4 ? _	↵	
3 − 5 ? _	↵	

Enter the BASIC program next:
First slide the mode switch to BASIC and enter the following:

Entry

```
PROG  0↵
10  A=0↵
20  IF IT(3) > 70000 THEN  A= (IT(3)− 70000)*0. 1↵
30  END
```

When gross wages for a part time employee exceed the amount of the basic deduction (¥70,000), the tax can be calculated by multiplying the difference between the wage and the deductible amount by 0.1 (10%).
Now, slide the mode switch to CETL and run the program:

Display	Key entry
> _	T↵
Rec. ? _	, ↵
Item ? _	, ↵
Printer(Y/N) ? _	Y↵

Sample output

```
NAME      HOUR        PAY        TAX     NET PAY
A          200    100,000      3,000      97,000
B          190     95,000      2,500      92,500
C          195     97,500      2,750      94,750
```

Summary

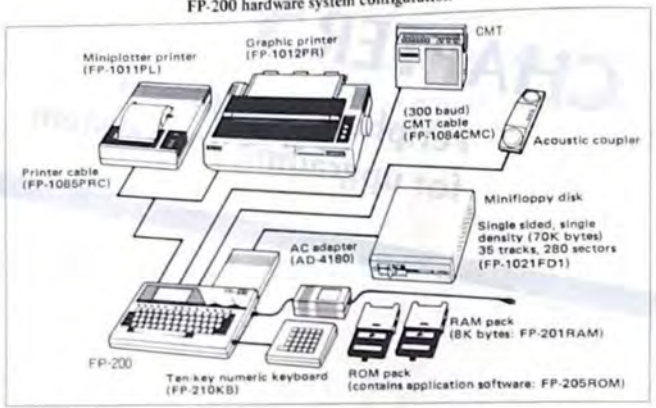
- The both CETL and BASIC can run independently. However, CETL is further enhanced when linked with BASIC.
- The CETL management functions: RC, IT, RC (n), IT (n), FL (l, m, n), SUMRC (m, n), and SUMIT (m, n) can be also used in BASIC programs.

CHAPTER 5

Peripheral equipment for upgrading your system

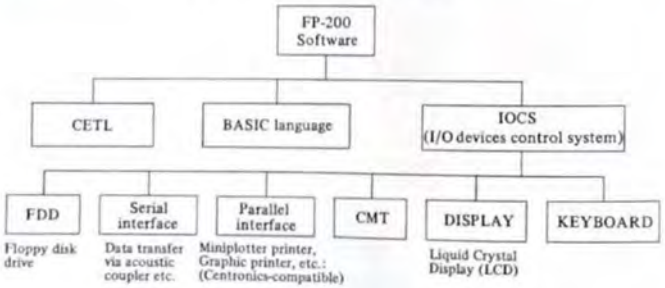
1. Hardware and software system configurations

FP-200 hardware system configuration



The FP-200 supports all of the peripheral equipment shown in the maximum system configuration above.
The software configuration is shown as follows:

FP-200 Software system configuration



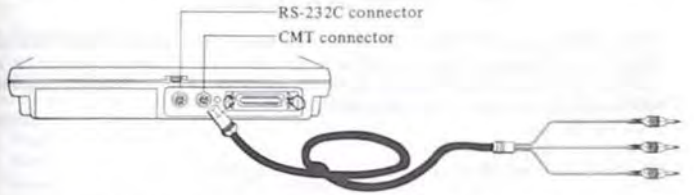
2. Cassette tape recorder

This section discusses how to save data and programs by using a cassette tape, and how to read (or load) data written on a cassette tape into the FP-200.

■ Connection

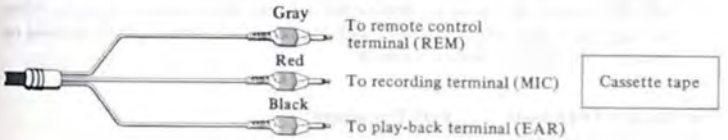
1. Turn the FP-200 off.
2. Connect the cassette tape cable (FP-1084CMC: supplied separately) to the CMT connector located at the rear of the FP-200.

(Rear view of FP-200)



The CMT and serial interface (RS-232C) connectors use the same DIN 8-pin type. Be careful not to connect the cable to the wrong connector.

3. Connect the three plugs at the end of the cassette tape cable to the recorder: the gray plug goes to the REM jack, the red plug to the MIC jack, and the black plug to the EAR jack.



■ Saving onto a Cassette Tape

Recording data and programs stored in computer memory onto a peripheral device such as cassette tape is called "saving". CETL and BASIC require different saving operations. Saving with BASIC is explained first.

- **Saving in BASIC mode**

1. Turn the remote switch located at the top right of the keyboard to ON.
2. Insert a cassette tape into the cassette recorder. (Note that previously recorded information is erased by saving new information.) When you save on any tape which already has information recorded, use the tape counter and take note of where the previous data ends so that you can begin the next section on the same tape after a few seconds.
   Be sure to bypass the tape leader before recording. (You can play the tape recorder by turning the remote switch to off.)
3. If your tape recorder has a recording level control, adjust it close to its maximum limit.
4. Turn on the record/play button on the recorder. The tape will not start running yet.
5. Set up the FP-200 for cassette tape saving now:
   Make sure that the mode switch is set to BASIC, and then enter the following from the keyboard:

```
SAVE "CAS0:[program-name]"
Save   to cassette   less than 8 characters
```

   where CAS0 tells the FP-200 to save to "cassette tape 0". The "program-name" must be string up to 8-characters long that refers to the name of a program in memory. It is recommended that you use program names that are clearly associated with the program, such as its function, as an aid to subsequent identification.
6. Press the ↵ key.
   The tape will start running here.
7. When saving is complete, the following message will appear:

```
Ready P 0
>_
```

8. If you notice that the procedure fails due to the wrong file name or because the tape runs out, don't worry; just press the **BREAK** key and start again from the beginning. When you use a new tape, take into account its leader and trailer; nothing can be recorded on them since no magnetic coating is present.

- **Saving in CETL mode ..... P (PUT) command**

1. Turn the remote switch ON.
2. Insert a tape cassette into the recorder.
3. Push the locking RECORD/PLAY button on the recorder.
4. Make sure that the mode switch is set to CETL.
5. Check to see if the FP-200 is waiting for a command (the prompt is displayed on the panel). If not, press the **BREAK** key.

6. Enter:

   P↵

   This will display:

```
Out(F/S/C)?_
```

   This is a prompt by which the FP-200 asks you which device you want to use:
   - F : Floppy disk
   - S : Serial port (acoustic coupler, etc.)
   - C : Cassette tape

   You are using the cassette tape. Enter:

   C↵

   This will start the tape running, and saving begins. Unlike saving in the BASIC mode, you need not enter the program name; the file name is automatically recorded.
7. When the saving is complete, >_ will appear on the display, requesting another command.

■ Verifying the Data

After you save any program or data, you need to verify that it has been correctly saved. Verify the saved program or data according to the following procedure (this procedure applies to both BASIC and CETL):

1. Turn the remote switch ON.
2. Rewind and then reposition the tape to the beginning of the saved file. After the tape is rewound, adjust the volume level control close to its maximum limit and put the tape recorder in RECORD/PLAY mode.
3. Make sure that the correct mode (BASIC or CETL) is selected by the mode switch. Then, enter:

```
VERIFY "CAS0:[file-name]"
```

4. Press the ↵ key.
   This will start a comparison for any discrepancy between the data which is read and the existing contents of memory.
5. If the file has been saved correctly, the following will appear on the display panel:
   CETL mode:

```
>_
```

   BASIC mode:

```
Ready P 0
>_
```

6. If the comparison detected any discrepancy (i.e., if the file was not saved correctly), the following message will appear:

```
RW Error
>_
```

Data recording is sometimes affected by the tape recorder characteristics and the tape quality. You should make it a standard practice to verify any recording after saving a program or data.
The following are common causes of cassette tape saving failures. Check them before saving or verification.

- A cable connection is not properly secured or the cable is connected to the wrong connector.
- The verification procedure is incorrect.
- The volume level control is not properly adjusted when the tape is played back.
- Problems on the tape itself, such as dirt or damage.
- Dirt on the recorder heads.

■ Loading from a Cassette Tape

Reading data and programs recorded on a peripheral device back into computer memory is called "loading". BASIC and CETL require different loading procedures. The BASIC loading procedure is explained first:

- **Loading in BASIC mode**

1. Turn the remote switch to ON.
2. Select a program area (one of P0 through P9). If an unnecessary program exists in the desired program area, erase the program by entering **NEW** ↵ .
3. Make sure that the mode switch is set at BASIC.
4. Rewind and reposition the cassette tape to a position a little ahead of the part you want to load.
5. Adjust the volume level control (similar to the verification process).
6. Push the PLAY button on the tape recorder.
7. Enter the following:

```
LOAD "CAS0:[program-name]"
```

   where the program name must agree with the one used when the program was saved, both as to characters and upper/lower case.
8. Press the ↵ key. This will start the loading process.
   If you press the ↵ key before entering the program name (i.e., if you omit the program name specification), the first program encountered will be loaded. If the tape was not positioned before the desired program, the FP-200 will read the programs on the tape in sequence, searching for that program. If the program is found, it is loaded and the tape stops running. If it is not found, the tape will run all the way to the end.
9. If loading fails, the message "**RW Error**" will be displayed.
   If loading is unsuccessful, repeat this procedure starting from step 1 after adjusting the volume level control somewhere between the middle and maximum positions. It is important to know the optimum volume level for your tape recorder because their characteristics vary considerably.

Note!

- Use the same cassette recorder for loading as the one which was used for recording. If different recorders are used for recording and play back, loading may be unsuccessful.
- Normally, all of the three CMT cable plugs should be inserted for proper operation. However, some recorders generate undesired noise when both the red and black plugs are simultaneously connected. With such a recorder, connect only the red plug when recording (tape write) and only the black plug when playing back (tape read); leave the gray plug connected for both operations.
- Some cassette recorders are not suitable for operation with the FP-200. Select a unit meets the FP-200 standards.

- **Loading in CETL mode ..... G (GET) command**

1. Turn the remote switch ON.
2. Select a file area (one of F0 through F9). If data already exists in the selected file area, the FP-200 will reject reading into that file area. If this occurs, select another file area or first erase the contents of the selected file area.
   * To select another file area, enter:

     PF1 (file-area-number) ↵

   * To erase the data in the selected file area, enter:

     K (K is the KILL command.) ↵

   The FP-200 will ask you whether you want to erase all the file areas or just the selected one by:

```
All/Presnt(A/P)? _
```

   If you want to erase only the selected file area, enter:

   P ↵
3. Rewind and reposition the tape to a position a little ahead of the file you want to load.
4. Adjust the volume level (according to the instructions given for verification) and push the locking PLAY button.
5. Enter:

   G ↵

   This will display the following message:

```
In (F/S/C)? _
```

   This message asks you from what peripheral device you want to load. Since you are using a cassette tape enter:

   C ↵

   FP-200 will then ask you for the file name by:

```
File Name? _
```

   In response to this, enter the correct file name that you used when saving the file.

6. After entering the file name, press the return key:
   ↵
   This will start the loading process.
7. When loading is completed, the FP-200 will display the following:

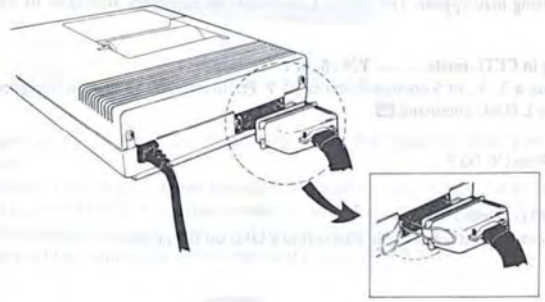
```
File name      F
Ready F Ø
> _
```


3. Printers

To produce a printed listing of programs and data, a miniplotter-printer (FP-1011PL) or graphic printer (FP-1012PR) is used. A printer cable (FP-1085PRC) is required to connect either of these printers to the FP-200.

■ Connection

1. Turn off the FP-200 and all peripheral equipment.
2. Connect the printer cable to the connector located at the rear of the FP-200. Connect the other end of the cable to the printer connector and secure the connector by putting the hook in place.



■ Operating Procedure

1. Turn on the FP-200 and the printer.
2. Select the ONLINE mode at the printer.
3. Select either the BASIC or CETL mode on the FP-200 with the mode switch. Operation varies, depending on the mode selected.

• Printing while in BASIC mode ..... LLIST or LPRINT command

1. Set the mode switch to BASIC.
2. Enter an LLIST or LPRINT command followed by a ↵ .

This will start the printing.

(1) LLIST command

This command generates a listing on the printer instead of displaying on the display panel, and is used to list programs and data. It has the following format:

LLIST [statement-number] [{ , / - }statement-number]

The statement number before the optional hyphen or comma is the first statement to be listed and the statement number after the separator is the last statement to be listed.

[Example]

LLIST↵	Lists from the beginning to the end.
LLIST 100-↵	Lists from statement 100 to the end.
LLIST 100-500↵	Lists statements 100 to 500.
LLIST -300↵	Lists from the beginning to statement 300.

(2) LPRINT command

This statement prints the results of the specified arithmetic expressions or strings. It has the following format:

LPRINT { expression / "character-expression" } [{ , / ; } { expression / " character-expression" }]*

The expressions may be any arithmetic or character (string) expressions, and may be separated by either commas or semicolons. The asterisk "*" means that any number of expressions or string may appear. The LPRINT statement is completely analogous to the PRINT statement.

- **Printing in CETL mode ..... T, F, S, or L command**

(1) Execute a T, F, or S command and enter **Y** ↵ in response to the following prompt, or enter a **L** (List) command ↵:

```
Print(Y/N)? _
```

(2) **L** (List) command
This command lists the whole contents in a table on the printer.

> **Note!**
> When the miniplotter-printer (FP-1011PL) or graphic printer (FP-1012PR) is used, be sure to supply power to the FP-200 via the special AC adapter (AD-4180) instead of by the batteries.

- **For the miniplotter printer FP-1011PL**

The following two modes can be switched by using control characters (CHR$ (28), CHR$ (37), CHR$ (46), CHR$ (27)).

Character Mode: **LPRINT CHR$(28); CHR$(46)**

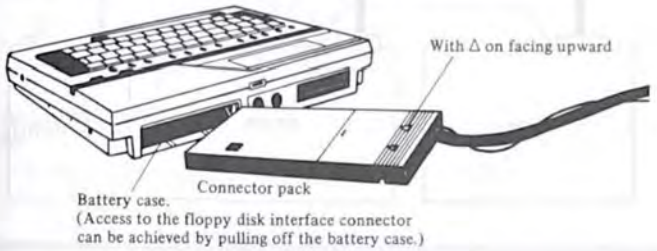
Graphic Mode: **LPRINT CHR$(28); CHR$(37)**

Temporary Graphic Mode: **LPRINT CHR$(27);**

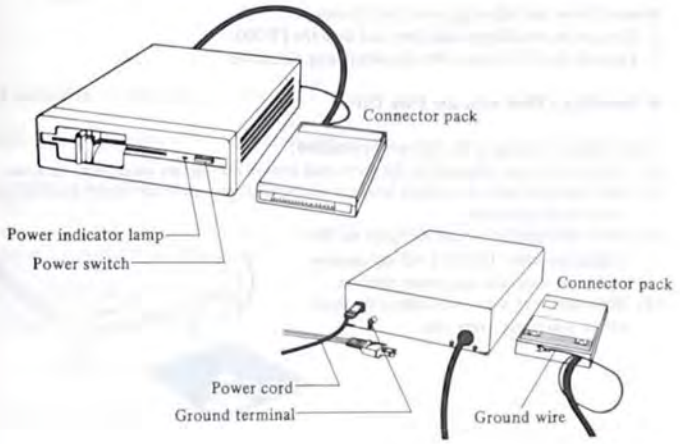
4. Minifloppy disk

The minifloppy disk can significantly reduce the time required for saving and loading.

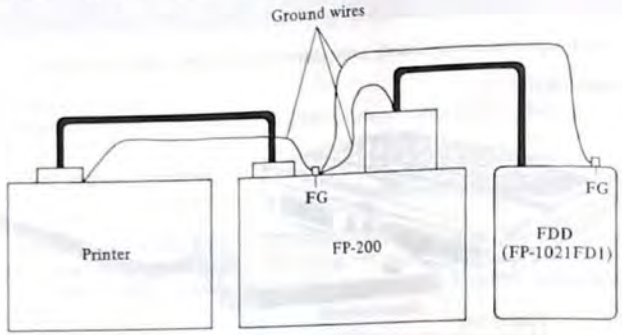
■ Connection



(1) Turn off the power to the FP-200, minifloppy disk drive and other peripheral equipment.
(2) Remove the battery case and connect the connector pack with the Δ sign facing upward to the minifloppy disk interface connector, which is located behind the battery case.
(3) Plug the minifloppy disk power cord into an AC power outlet.
(4) Connect the ground wire of the connector pack to the FP-200.



Grounding the minifloppy disk drive
When the optional FDD (FP-1021FD1) and printer (FP-1011PL or FP-1012PR) are connected, connect the ground wires to the FP-200 as illustrated below.



Note!
When the minifloppy disk drive is used, be sure to supply power to the FP-200 via the special AC adapter (AD-4180) instead of by the batteries.

■ Turning Power On and Off

Always follow the following power on/off sequence:
1. Turn on the minifloppy disk drive and then the FP-200.
2. Turn off the FP-200 and then the minifloppy disk drive.

■ Inserting a Disk into the Disk Drive

Insert a disk according to the following procedure:
(1) Turn the lever as indicated by the arrow and remove the dummy cardboard protector.
(2) Hold the disk with the CASIO label on top toward you, and then gently insert it into the drive until it stops.
(3) After inserting it, return the lever to the original position. The drive will not operate unless the lever is in the proper position.
(4) Make sure that there is no disk in the drive before you insert a new one.

■ Saving to a Minifloppy Disk

BASIC and CETL require different operating procedures:

- **Saving in BASIC mode**

Enter:

SAVE "program name" ↵

This will start the saving process. The FP-200 will display the following prompt when the saving process is complete.

```
>_
```

- **Saving in CETL mode**

Enter:

P (PUT command) ↵

```
Out(F/S/C)?_
```

The FP-200 will display the above question, which asks you what peripheral device you want to use. In response to this, enter:

F ↵

This will start the saving process. When it has been completed, the FP-200 will wait for the next command after issuing the prompt:

```
Ready F Ø
>_
```

■ Loading from a Minifloppy Disk

BASIC and CETL require different operating procedures:

- **Loading in BASIC mode**

1. Select a program area.
2. Display the disk directory by entering the following command in order to make sure that the disk contains the desired program:

FILES ↵

3. Enter:

LOAD "program name" ↵

This will start the loading process.

4. When loading has been completed, the FP-200 will display the following prompt to request the next command:

```
Ready P Ø
> _
```

- **Loading CETL mode**

1. Select a file area.
2. Examine the disk directory by entering:

   FILES ↵

3. Enter:

   G (GET command) ↵

   The FP-200 will display the following message, which asks you from what peripheral device you want to load:

```
In (F/S/C) ? _
```

   You are loading from the minifloppy disk drive, so enter:

   F ↵

   The FP-200 will ask you for the name of the file you want to load:

```
File Name ? _
```

   In response, enter the file name followed by ↵.
   This will start the loading process.
4. When loading has been completed, the FP-200 will display the following to request the next command:

```
Ready F Ø
> _
```

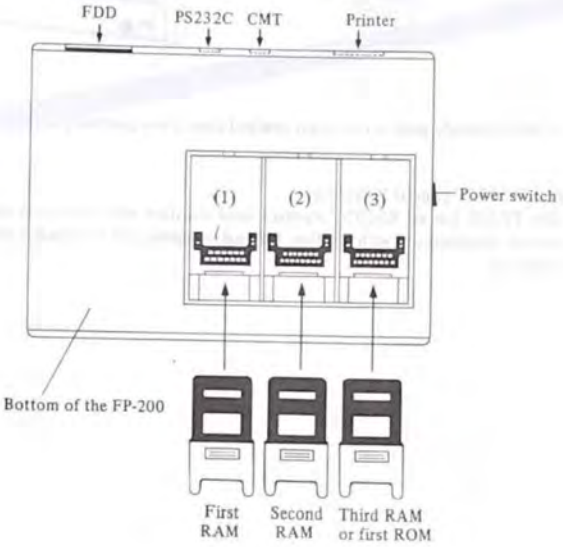
If an "NF Error" message is displayed after execution of the G command, which indicates that the FP-200 cannot find the file name on the disk, repeat the command with the correct file name.

5. RAM/ROM Pack /RS232C

(1) RAM/ROM Pack

■ How to Set the RAM/ROM Pack into the FP-200.

1. Turn the power off and remove the back-up batteries.
2. Install the RAM/ROM Packs into the RAM/ROM Pack sockets at the bottom of the unit as follows:



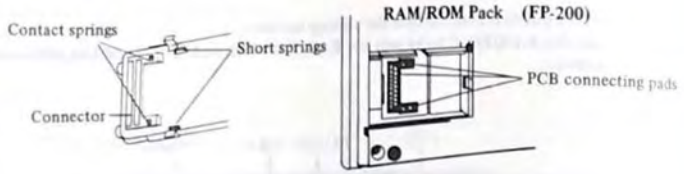
* Only one ROM pack is available.

3. After the installation, set the back-up batteries and turn the power on again, then enter the RESET command.

   > RESET ↵

■ **Precautions for Handling RAM and ROM Packs**

1. Discharge any static electricity which has accumulated in your body by touching a door knob or faucet, etc., before you handle a RAM or ROM pack; if this is not done, its internal circuitry may be destroyed.
2. Dirt or oil from fingers on the pack connector or the PCB connector area of the FP-200 can be a cause of poor connections. Never touch or handle these areas.
   Do not push in or touch the contact springs; they are easily damaged.



3. Put a removed pack in the proper case and keep it in a dust-free place.

(2) RS-232C (Serial Interface)

The FP-200 has an RS-232C standard serial interface with which you can transfer data to and communicate with another personal computer, via an acoustic coupler, over the telephone.

CHAPTER 6
Hardware descriptions

1. FP-200 Hardware Overview

Specifications

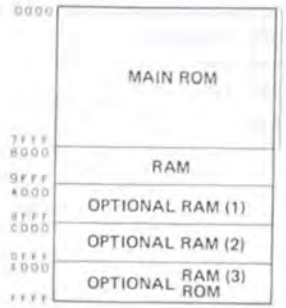
CPU	MSM80C85 AGS'
	Clock rate: 6.144 MHz
ROM	Standard: 32K bytes, Maximum: 40K bytes
RAM	Standard: 8K bytes, Maximum: 32K bytes
	Note: FP-200 can provide a combined RAM and ROM capacity of up to 64K bytes.
Display panel	64 Vertical x 160 Horizontal dot matrix LCD: 1/64 duty cycle, provided with a contrast adjustment control.
LSIs	Liquid crystal display driver LSIs
	Character ROM: 64 bits x 256 characters
	LCD driver RAM: 8 bits x 640 bytes (built in)
	I/O control LSI
Timer	Capable of maintaining and supplying system time consisting of year, month, day, hour, minute, and second.
Keyboard	ASCII (69 keys)
Interface	CMT: 300 baud
	△ Printer: Centronics standard interface
	△ RS-232C standard serial interface, 300 baud
	△ FDD (FP-1021FD1) interface
	△ denotes that the interface can be used only when the AC adapter is used.
Power	Main power: 4 dry cell (AA UM-3) batteries or AC adapter (AD-4180)
	Memory back-up power: 2 dry cell (AA UM-3) batteries
Battery life	Main batteries (With 3 RAM packs installed, REMOTE relay OFF, and program running)
	AA UM-3: Approx. 6 hours
	Alkali: Approx. 11 hours
	Back-up batteries (With 3 RAM pack installed)
	AA UM-3: 6 months

- FP-200 provides a standard user area of 8K bytes. However, approximately 2.3K bytes of the user area are reserved for a system area and the remaining approximately 5.7K bytes are available for use as BASIC, CETL, and character data area whose sizes can be independently varied by an AREA or CLEAR statement.

2. Memory Map

Optional RAM

The FP-200 has a ROM of 32K bytes and a RAM of 8K bytes installed as standard. RAM and ROM packs with a capacity of 8K bytes each are available. The RAM can be expanded up to 32K bytes by adding RAM packs. Only one ROM pack can be installed in the last slot (E000–FFFF).
In addition, the FP-200 has an LCD character ROM of 256 bytes and an LCD driver RAM of 1.2K bytes which are not included in either the memory of I/O map and which are not accessible to the user.

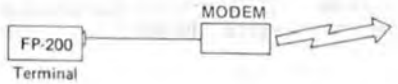


3. Connecting Peripheral Equipment

Turn off the FP-200 and all peripheral equipments before you connect or disconnect any device.
When you connect an RS-232C standard I/O device, a printer or the FDD, use the AC adapter (AD-4180).

(1) CMT
The CMT uses the same connector as the RS-232C serial I/O cable. Be careful not to connect the CMT cable to the wrong connector.

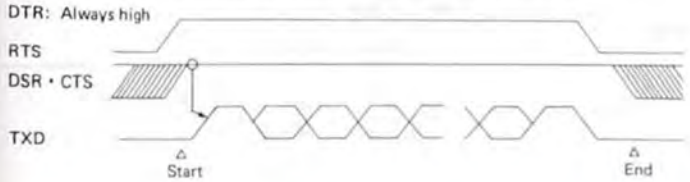
(2) RS-232C Serial I/O Interface
The RS-232C standard provides specifications for an interface used for data communications over public or private communications. In this mode of operation, the FP-200 works as a terminal, and this interface provides a simple connection with the modem:



Operation
The FP-200 uses the following RS-232C interface signals:

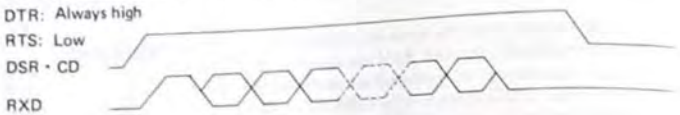
- **Output operation**

During output of each data byte, the FP-200 keeps RTS high, and outputs the data after making sure that both DSR and CTS are high. (DSR and CTS are checked for high levels at the beginning of each output byte.) RTS goes low after each byte is transmitted.

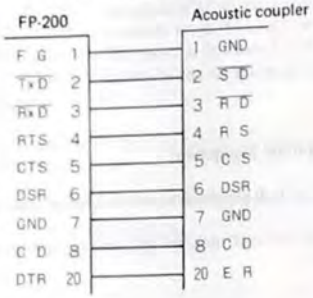


• **Input operation**

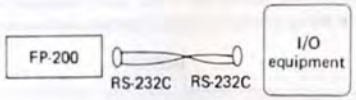
The input data is assumed to be valid only while both DSR and CD are high.



Connection examples

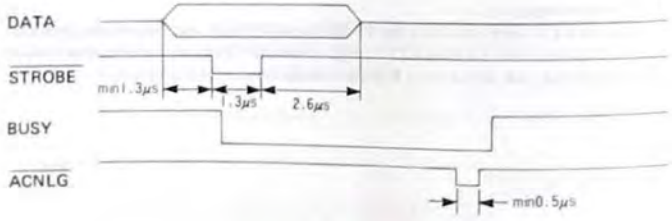


Many types of I/O equipment used through this interface operate as terminals themselves. The RS-232C interface must control the various signals such as RTS, CTS, DSR, and CD, depending on which I/O device is connected.



(3) Printer

The FP-200 can connect a Centronics-compatible printer or plotter. The following is a timing diagram of the interface signals:



(4) FDD interface

The FP-200 can connect only the FP-1021FD1 FDD. You cannot connect other devices through this interface.

4. Connector Terminals

(1) CMT

DIN 8 pin

Terminal No.	I/O	Signal name
1		
2		GND
3		
4	I	MIC
5	O	EAR
6		REM+
7		REM−
8		
Frame		F G

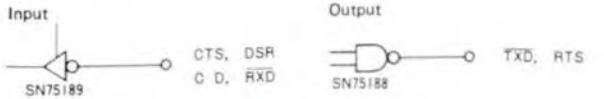
Electrical specifications

Terminal	Specifications	
MIC	Output impedance:	5 kΩ
	Output voltage:	3 mVp-p
EAR	Input impedance:	10 kΩ
	Input voltage:	3 – 10 Vp-p
REMOTE	Rated capacity:	24 V, 1 A

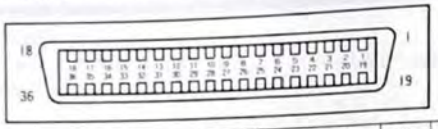
(2) RS-232C Serial I/O Interface

DIN 8 pin

Terminal No.	I/O	Signal name
1	O	DTR
2		GND
3	O	$\overline{TXD}$
4	I	$\overline{RXD}$
5	I	DSR
6	I	CTS
7	I	C D
8	O	RTS
Frame		F G



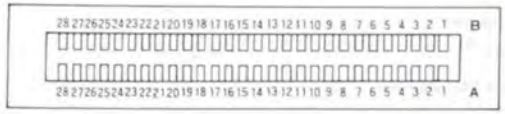
(3) Printer (Centronics Standard Interface)



Terminal No.	I/O	Signal name	Terminal No.	I/O	Signal name
1	O	$\overline{\text{STROBE}}$	19		G N D
2	O	DATA1	20		G N D
3	O	DATA2	21		G N D
4	O	DATA3	22		G N D
5	O	DATA4	23		G N D
6	O	DATA5	24		G N D
7	O	DATA6	25		G N D
8	O	DATA7	26		G N D
9	O	DATA8	27		G N D
10	I	$\overline{\text{ACNLG}}$	28		G N D
11	I	BUSY	29		G N D
12			30		G N D
13			31	O	$\overline{\text{INIT}}$
14			32	I	$\overline{\text{ERROR}}$
15			33		G N D
16		GND	34		
17		F G	35		
18			36		

Note: The FP-200 can operate the printer without ACNLG or BUSY signal. When one of them is not used, connect the other to ground.

(4) FDD connector



Terminal No.	I/O	Signal name	Terminal No.	I/O	Signal name
A 1			B 1		G N D
A 2			B 2		G N D
A 3			B 3		
A 4		GND	B 4		
A 5		GND	B 5	O	$\overline{\text{RESET}}$
A 6	O	$\overline{\text{REQ}}$	B 6		G N D
A 7	O	$\overline{\text{RD}}$	B 7	O	$\overline{\text{WR}}$
A 8	O	$\overline{\text{CS}}$	B 8		
A 9			B 9		
A10			B10		
A11			B11		
A12			B12	I	$\overline{\text{I N T}}$
A13	I/O	D0	B13	I/O	D1
A14	I/O	D2	B14	I/O	D3
A15	I/O	D4	B15	I/O	D5
A16	I/O	D6	B16	I/O	D7
A17	O	$\overline{\text{A0}}$	B17	O	$\overline{\text{A1}}$
A18	O	$\overline{\text{A2}}$	B18		G N D
A19		GND	B19		G N D
A20		GND	B20		G N D
A21		GND	B21		G N D
A22		GND	B22		G N D
A23		GND	B23		G N D
A24		GND	B24	O	$\overline{\text{A15}}$
A25			B25		
A26			B26		G N D
A27			B27		G N D
A28			B28		

5. Peripheral Equipment

- **CMT DIN connector**
  A cassette tape recorder provided with remote terminals.
- **RS-232C serial I/O connector**
  Acoustic coupler
- **Centronics standard interface connector**
  Centronics compatible printer or plotter. (FP-1012PR, FP-1011PL)
- **FDD connector**
  FP-1021FD1 (single sided/single density minifloppy disk)
- **Memory**
  C MOS RAM Pack: FP-201 (RAM)
  EP ROM Pack: FP-205 (ROM)
- **Auxiliary ten-key numeric keypad**
  FP-210KB

CHAPTER 7
Program examples

FP-200 provides a library which includes the following application programs:

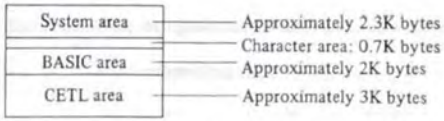
1. Sales Results (CETL)
2. Mail List (CETL and BASIC)
3. Golf Scoring (CETL and BASIC)
4. Simulation Games (BASIC)
   Bowling
   Gold Mine

NOTE:

- **Changing Initial BASIC and CETL Areas**

The FP-200 memory is allocated as follows initially, or after the RESET command is executed:

Standard 8K byte RAM



These initialized area sizes can be changed by using the AREA command which has the following format:

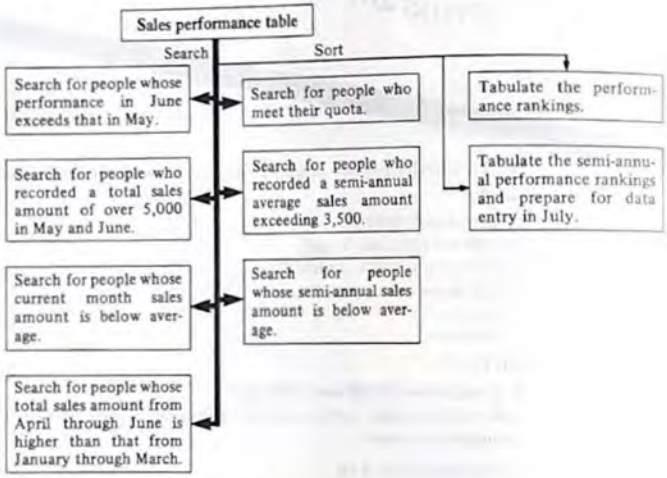
AREA	CETL area size

where "CETL area size" is the desired CETL area size in bytes.
If you decrease the CETL area size, the BASIC area increases by the same amount. The area sizes should be selected by estimating beforehand the sizes of the BASIC program and CETL table you are developing.

1. Sales results (CETL)

Once data is stored in a CETL table, the file can be processed any time you want, so that you can add data weekly and monthly and you can also utilize the hot data. We will discuss what data can be retrieved from a given table in the following sales performance example. This is only an example; you can learn how to apply the CETL facilities to various fields through the discussions.

The following nine items of information can be retrieved from a sales performance table by using the FP-200 search and sort facilities:



Let's retrieve the above data from the following sales performance table.

Sales performance table

(Units: ¥1,000)

NAME	JANUARY	FEBRUARY	MARCH	APRIL	MAY	JUNE	TOTAL
A	1,230	3,500	3,010	4,450	2,050	4,200	
B	1,350	1,580	2,890	2,050	4,450	5,340	
C	4,450	5,510	3,150	3,890	3,650	2,240	
D	4,370	2,630	4,360	2,340	3,890	1,470	
E	2,450	3,250	4,730	3,340	2,340	3,560	
F	6,680	1,160	5,200	6,670	1,890	2,560	
G	2,920	5,300	1,930	4,100	7,760	5,670	
TOTAL							

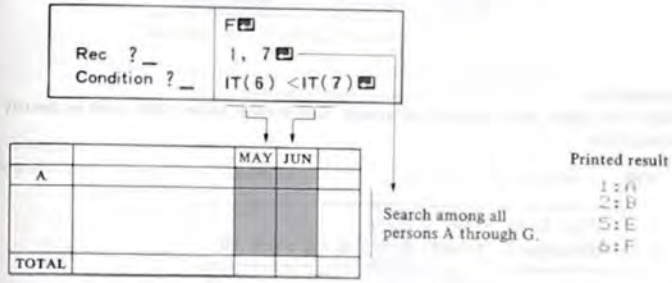
Output

```
NAME      JAN      FEB      MAR      APR      MAY      JUN    TOTAL
A       1,230    3,500    3,010    4,450    2,050    4,200   18,440
B       1,350    1,580    2,890    2,050    4,450    5,340   17,660
C       4,450    5,510    3,150    3,890    3,650    2,240   22,890
D       4,370    2,630    4,360    2,340    3,890    1,470   19,060
E       2,450    3,250    4,730    3,340    2,340    3,560   19,670
F       6,680    1,160    5,200    6,670    1,890    2,560   24,160
G       2,920    5,300    1,930    4,100    7,760    5,670   27,680
TOTAL  23,450   22,930   25,270   26,840   26,030   25,040  149,560
```

Search ..... F (Find) Command

(Example 1)

Search for people who recorded a higher performance in June than May by comparing the performances of all people in the two months.

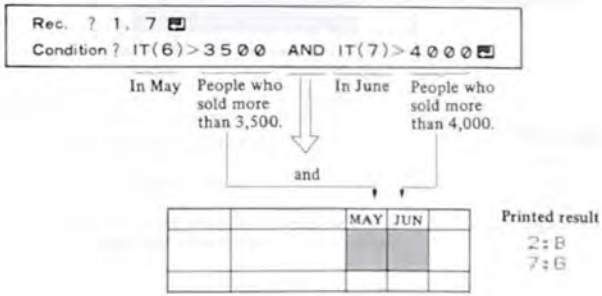


Determine who recorded a higher performance in June than May.

(Example 2)

Find the people who achieved the May and June sales goals, which were respectively 3,500 and 4,000.

F (F command for the search)



(Example 3)
Search for people who recorded a total sales amount lower than 5,000 in May and June.

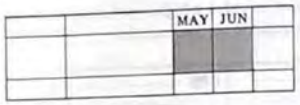
F↵

```
Rec. ? 1, 7 ↵
Condition ? SUMIT(6, 7)<5000 ↵
```

Sum — Sum (the sales amounts for) May and June.

People who sold less than 5,000.

(6, 7) represents items 6 and 7 which correspond respectively to May and June.



Printed result
6:F

(Example 4)
Search for people who recorded an average sales amount higher than 3,500 in January through June.

F↵

```
Rec. ? 1, 7 ↵
Condition ? SUMIT(2, 7)/6>3500 ↵
```

Sum of sales amounts from January through June. ÷6 — Average from January to June

People who sold more than 3,500 (average)

Printed result
3:C
6:F
7:G

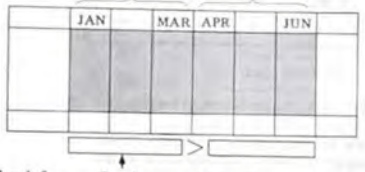
(Example 5)
Search for people whose total sales amount in January through March exceeds that in April through June.

F↵

```
Rec. ? 1, 7 ↵
Condition ? SUMIT(2, 4)>SUMIT(5, 7) ↵
```

Sum (sales amounts in January through March.)

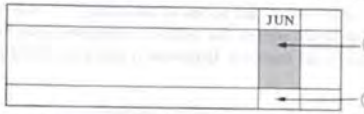
Sum (sales amounts in April through June.)



Look for people whose total from January to March is greater than that from April to June.

Printed result
3:C
4:D
5:E
6:F

(Example 6)
Search for people who recorded a sales amount in June which was below average.



Use the FL function.

(Example) FL (0, 8, 7)

File number (designates file area 0)

Rec. number

Item number

You do not necessarily need to use the FL function, but it simplifies the computations.

F↵

```
Rec. ? 1, 7 ↵
Condition ? FL(0, 8, 7)/7>IT(7) ↵
```

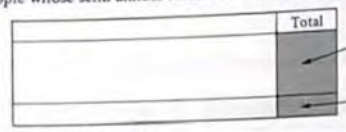
Obtain the total sales for June.

Average for June.

Look for people who sold less than the average by comparing each sales amount with the average.

Printed result
3:C
4:D
5:E
6:F

(Example 7)
Search for people whose semi-annual total sales amount is below average.



Use the FL function.

F↵

```
Rec. ?  1 , 7 ↵
Condition ?  FL(0, 8, 8)/7>IT(8) ↵
```

The total sales.

Compute the semi-annual average.

Look for the poeple whose semi-annual sales are below the semi-annual average.

Printed result

```
1:A
2:B
4:D
5:E
```

Sort ..... S (Sort) Command

It is convenient to assign a code number to each record (or row) of the above sales performance table for sorting purposes. After records are sorted by answering "Y" to the "Memory Move (Y/N)?" question, you can easily restore the original arrangement for further data addition, etc. by using the assigned code numbers. Otherwise it might be difficult to restore the original arrangement.

Sample output with a response of N to the "Memory Move (Y/N)?" question.

```
    JUN

7:  5,670
2:  5,340
1:  4,200
5:  3,560
6:  2,560
3:  2,240
4:  1,470
```


- Assigning a code number

I ↵ (Enter the I (Insert) command.)

```
Rec / Item  (R / I)  ? I↵
Item ? 8↵
Item 9 : Name ?  CODE↵
```

Instruct CETL to add a new column (item).

Instruct CETL to add the new item (column) after the existing item 8.

Specify the name of this item.

CETL will then ask you about the new item (column) 9:

```
Type  (N / S) : N ↵
Expression  ? ↵
Format  ###↵
```

Enter code numbers for all the records (rows):

```
9 - 1 ? 1↵
9 - 2 ? 2↵
  ⋮
9 - 7 ? 7↵ (BREAK)
```

Verify that the code numbers are correctly entered:

T↵

```
Rec. ? , ↵
Item ? , ↵
Printer  (Y / N)  ? Y↵
```

NAME	JAN	FEB	MAR	APR	MAY	JUN	TOTAL	CODE
A	1,230	3,500	3,010	4,450	2,050	4,200	18,440	1
B	1,350	1,580	2,890	2,050	4,450	5,340	17,660	2
C	4,450	5,510	3,150	3,890	3,650	2,240	22,890	3
D	4,370	2,630	4,360	2,340	3,890	1,470	19,060	4
E	2,450	3,250	4,730	3,340	2,340	3,560	19,670	5
F	6,680	1,160	5,200	6,670	1,890	2,560	24,160	6
G	2,920	5,300	1,930	4,100	7,760	5,670	27,680	7
TOTAL	23,450	22,930	25,270	26,840	26,030	25,040	149,560	

(Example 1)
Rank the personnel according to their performance in June.

```
S↵
Key Item ? "JUN"↵
Up/Down (U/D) ? D↵
Rec. ? 1, 7↵
Mem. Move (Y/N) ? Y↵
```

Verify the result by the T command:

```
T↵
```

NAME	JAN	FEB	MAR	APR	MAY	JUN	TOTAL	CODE
G	2,920	5,300	1,930	4,100	7,760	5,670	27,680	7
B	1,350	1,580	2,890	2,050	4,450	5,340	17,660	2
A	1,230	3,500	3,010	4,450	2,050	4,200	18,440	1
E	2,450	3,250	4,730	3,340	2,340	3,560	19,670	5
F	6,680	1,160	5,200	6,670	1,890	2,560	24,160	6
C	4,450	5,510	3,150	3,890	3,650	2,240	22,890	3
D	4,370	2,630	4,360	2,340	3,890	1,470	19,060	4
TOTAL	23,450	22,930	25,270	26,840	26,030	25,040	149,560	

(Example 2)
Examine the semi-annual total rankings and prepare for data entry in July.

The Sort command cannot work if the item (column) contains expressions because the sort operation is valid for only actual data. We use the C command when we want to sort the item which contains some expressions. (The C command computes the expression and insert the result as data in that item.)

```
C↵
Rec. ? 1, 8↵
Item ? 8↵
```

This will put the totals into the "total" item of all the records as numerical data.
Rank the totals by using the S command:

```
S↵
Key Item ? 8↵                    Or "TOTAL"↵
Up/Down (U/D) ? D↵
Rec. ? 1, 7↵
Mem. Move (Y/N) ? Y↵
```


Printed result

NAME	JAN	FEB	MAR	APR	MAY	JUN	TOTAL	CODE
G	2,920	5,300	1,930	4,100	7,760	5,670	27,680	7
F	6,680	1,160	5,200	6,670	1,890	2,560	24,160	6
C	4,450	5,510	3,150	3,890	3,650	2,240	22,890	3
E	2,450	3,250	4,730	3,340	2,340	3,560	19,670	5
D	4,370	2,630	4,360	2,340	3,890	1,470	19,060	4
A	1,230	3,500	3,010	4,450	2,050	4,200	18,440	1
B	1,350	1,580	2,890	2,050	4,450	5,340	17,660	2
TOTAL	23,450	22,930	25,270	26,840	26,030	25,040	149,560	

Next, prepare for data entry in July; restore the original arrangement by sorting the records by code number:

NAME	JAN	FEB	MAR	APR	MAY	JUN	TOTAL	CODE
A	1,230	3,500	3,010	4,450	2,050	4,200	18,440	1
B	1,350	1,580	2,890	2,050	4,450	5,340	17,660	2
C	4,450	5,510	3,150	3,890	3,650	2,240	22,890	3
D	4,370	2,630	4,360	2,340	3,890	1,470	19,060	4
E	2,450	3,250	4,730	3,340	2,340	3,560	19,670	5
F	6,680	1,160	5,200	6,670	1,890	2,560	24,160	6
G	2,920	5,300	1,930	4,100	7,760	5,670	27,680	7
TOTAL	23,450	22,930	25,270	26,840	26,030	25,040	149,560	

```
I↵ (Enter the I (Insert) command.)
Rec/Item (R/I) ? I↵   Add a column (item).
Item ? 8↵             Put the item after item 8.
Item 9:Name JUL↵      Enter "July".
Type (N/S) :N↵
Expression ?↵
Format? ###,###↵
```

This completes the framework for accepting data entered for the July period. It is good practice to save important data which is used for a long period of time on a peripheral device, so that it can be loaded as required.
You can also obtain the totals for the subsequent semi-annual term or immediately examine the annual totals. If the data cannot fit on a printer line, you can divide the records into two or three groups of items for improved readability.

Sales Performance Table Source Listing

Sample entries for defining the table

```
File Name:SALES
```
Define the file name.

```
Number of Rec.:8
Number of Item:8
```
Enter the table size.

```
Label Record
Item1:NAME
   Type(N/S)   :S
   Expression  ?
   Format      :&          &
```
Define the first item (column), named "Name", into which a person's name will be entered.

```
Item2:JAN
   Type(N/S)   :N
   Expression  ?
   Format      :###,###
```
Define the second item, named "Jan.", into which the sales data for January will be entered.

```
Item3:FEB
   Type(N/S)   :N
   Expression  ?
   Format      :###,###

Item4:MAR
   Type(N/S)   :N
   Expression  ?
   Format      :###,###

Item5:APR
   Type(N/S)   :N
   Expression  ?
   Format      :###,###

Item6:MAY
   Type(N/S)   :N
   Expression  ?
   Format      :###,###

Item7:JUN
   Type(N/S)   :N
   Expression  ?
   Format      :###,###

Item8:TOTAL
   Type(N/S)   :N
   Expression  :SUMIT(2,7)
   Format      :###,###
```
Define the eighth item, named "Total", into which the semi-annual sales total is entered.

Data entry

```
Data Area
1-1 :A
1-2 :1230
1-3 :3500
1-4 :3010
1-5 :4450
1-6 :2050
1-7 :4200
1-8 ?
```
Enter the sales amount for person A. Enter the data from January to June.

Enter only ↵ because item 1–8 (item 8 in record 1) is already defined.

```
2-1 :B
2-2 :1350
2-3 :1580
2-4 :2890
2-5 :2050
~~~~~~~~~~

~~~~~~~~~~
6-1 :F
6-2 :6680
6-3 :1160
6-4 :5200
6-5 :6670
6-6 :1890
6-7 :2560
6-8 ?

7-1 :G
7-2 :2920
7-3 :5300
7-4 :1930
7-5 :4100
7-6 :7760
7-7 :5670
7-8 ?

8-1 :TOTAL
8-2 :SUMRC(1,7)
8-3 :SUMRC(1,7)
8-4 :SUMRC(1,7)
8-5 :SUMRC(1,7)
8-6 :SUMRC(1,7)
8-7 :SUMRC(1,7)
8-8 ?
```
Enter the expression that gives the vertical total in all items in the eighth record (row).

Enter only ↵ for 8–8, as with 1–8 above.

2. Mailing lists (CETL and BASIC)

32K bytes, 150 records

Addressing letters is very tedious work; it is an enormous waste of time and effort to write the address on every piece of mail to the same addressee.
Once installed, this program lets FP-200 address mail for you. It also provides a routine that can retrieve only those people living in the city of Tokyo or in the state of California.

*Let's use the following sample addresses:

NAME	ZIP	CITY	STATE	ADDRESS	TEL
CASIO	160	SHINJUKU	TOKYO	2-6 NISHISHINJUKU	03-345-9999
BILL KING	52551	FAIRFIELD	IOWA	85 NORTH 2nd AVENUE	515-777-4444
MARK SMITH	92122	SAN DIEGO	CALIFORNIA	7000 ERLANGER STREET	714-621-8000
PAUL PERKINS	93274	SIMI VALLEY	CALIFORNIA	2807 ELIZONDO	805-652-3222
KEN FARMER	10031	NEW YORK	NEW YORK	6 EAST 43rd STREET	800-344-6434

Printed sample addresses

```
CASIO
2-6 NISHISHINJUKU
SHINJUKU,TOKYO 160
```

Preparation

This program requires a large memory size. If there are any programs or data in the FP-200 which you want to keep, save them onto a cassette tape or disk and then clear the FP-200 before entering:

RESET ↵

which will clear and ready the FP-200 for the program.

Defining an address table

- Select the CETL mode and enter:

N ↵

Now begin defining the address table:

```
File Name:ADDRESS

  Number of Rec.:5
  Number of Item:6          ← Enter the table size.

Label Record
Item1:NAME                  ← Define the first item (column), named "NAME".
   Type(N/S)  :S
   Expression ?
   Format     :&                    &

Item2:ZIP                   ← Define the second item, named "ZIP".
   Type(N/S)  :S
   Expression ?
   Format     :&          &

Item3:CITY                  ← Define the third item, "CITY".
   Type(N/S)  :S
   Expression ?
   Format     :&                &

Item4:STATE                 ← Define the fourth item, "STATE".
   Type(N/S)  :S
   Expression ?
   Format     :&                &

Item5:ADDRESS               ← Define the fifth item, "ADDRESS".
   Type(N/S)  :S
   Expression ?
   Format     :&                          &

Item6:TEL                   ← Define the sixth item, "TELephone".
   Type(N/S)  :S
   Expression ?
   Format     :&                &
```

Data entry

```
Data Area
1-1 :CASIO ──────────── NAME (13 characters)
1-2 :160 ────────────── ZIP (7 characters)
1-3 :SHINJUKU ──────── CITY (13 characters)
1-4 :TOKYO ──────────── STATE (13 characters)
1-5 :2-6 NISHISHINJUKU ─ ADDRESS (22 characters)
1-6 :03-345-9999 ────── TEL (15 characters)
```

```
5-1 :KEN FARMER
5-2 :10031
5-3 :NEW YORK
5-4 :NEW YORK
5-5 :6 EAST 43rd STREET
5-6 :800-344-6434
```

If any data exceeds the specified size, redefine the data format.

Examine the entered data ..... T command

Examine the data you have entered. (You need not enter all of the 150 items of data at one time. Entering them a few at a time will minimize errors.
Enter:

T ↵

The FP-200 will respond with the following questions:

```
Rec.? , ↵ ──── Instruct the FP-200 to output all records.
Item? , ↵ ──── Instruct the FP-200 to output all items.
Printer (Y/N)? Y↵ ── Instruct the FP-200 to use the printer for output.
```

NAME	ZIP	CITY	STATE	ADDRESS	TEL
CASIO	160	SHINJUKU	TOKYO	2-6 NISHISHINJUKU	03-345-9999
BILL KING	52551	FAIRFIELD	IOWA	85 NORTH 2nd AVENUE	515-777-4444
MARK SMITH	92122	SAN DIEGO	CALIFORNIA	7000 ERLANGER STREET	714-621-8000
PAUL PERKINS	93274	SIMI VALLEY	CALIFORNIA	2807 ELIONDO	805-652-3222
KEN FARMER	10031	NEW YORK	NEW YORK	6 EAST 43rd STREET	800-344-6434

All the data have been entered correctly.

Saving data ..... P command

With the above mailing list program, you can create various address files ranging from business files which contain companies and stores to private files which contain addresses of club members, acquaintances and friends.
Files created with so much effort must be preserved for later use. Let's save the above address file on a cassette tape, as an example. (Cassette tape saving is described in detail on page 129.)
Enter:

P ↵

```
Out(F/S/C)   Respond to this with C ↵ to select the cassette tape.
Ready F0
>_           } This indicates that the data has been saved.
```

Addressing Mail

Now address mail with a BASIC library program.
Select the BASIC mode and press:

(PF 4) (Same as RUN ↵)

```
Rec.? 5 ↵   (Number of entered data items)
```

The program will ask you for the range of data over which you want to search.

```
SEARCH Y/N ?
```

Determine whether or not to make a search by responding with either Y or N to this question.
When you want no search, enter:

N ↵

The program will ask you what you want to print with the following message:

```
1 : TEL  2 : LABEL  3 : END?
```

Determine what to print by responding with a number:
1. TEL ..... Prints phone numbers
2. LABEL ..... Prints the labels.
3. END ..... Terminates the program.

When you want a search, enter:

Y ↵

The program will ask you what you want to use as search key:

```
SEARCH MODE (1. NAME  2. ZIP  3. CITY  4. STATE)?
```

Select a search mode by responding with the corresponding number:

1. NAME ..... Searches the address file by name.
2. ZIP ..... Searches the address file by ZIP code.
3. CITY ..... Searches the address file by city.
4. STATE ..... Searches the address file by state.

The program will request a search key (or comparison data) by:

```
COMPARISON DATA ?
```

Respond with a search key, such as a city name, etc.
For example, enter:

```
CALIFORNIA
```

The program will ask you what you want to print:

```
1 : TEL   2 : LABEL   3 : END
```

Determine what to print by responding with a number:

1. TEL ..... Prints phone numbers.
2. LABEL ..... Prints the labels.
3. END ..... Terminates the program.

A search example

Address mail only to persons in California.
Press:

PF4

or enter:

```
RUN ↵
```

Respond to the questions presented by the program as follows:
Enter the number of data records.

```
Rec. ?  5 ↵          (Number of entered data items)
SEARCH   Y ↵
SEARCH MODE (1. NAME  2. ZIP  3. CITY  4. STATE) ?   3 ↵
COMPARISON DATA ?   CALIFORNIA ↵
1 : TEL   2 : LABEL   3 : END
2 ↵
```

This would address mail to persons living in California, as follows:

```
MARK SMITH
7000 ERLANGER STREET
SAN DIEGO,CALIFORNIA 92122

PAUL PERKINS
2807 ELIZONDO
SIMI VALLEY,CALIFORNIA 93274
```


Phone number listing example

```
CASIO          03-345-9999
BILL KING      515-777-4444
MARK SMITH     714-621-8000
PAUL PERKINS   805-652-3222
KEN FARMER     800-344-6434
```

BASIC programs

10 – 60	Initialization
70 – 250	Search
260 – 320	TEL listing
330 – 470	Address print
480 – 520	Searched TEL listing
530 – 640	Address printing

Variable tables

A	Number of entered data items
A2$( )	Data
A$	Search selection
B	Search mode
B$	Comparison (key) data
C$, D$, AI$	Character variables
I	Loop counter
C, D	Print-out selection

BASIC program listing

```
 10 CLS:CLEAR
 20 INPUT"Rec.";A
 30 DIMA2$(A,6)
 40 INPUT"SEARCH Y/N";A$
 50 IF A$="Y" OR A$="N" THEN 60 ELSE 40
 60 IF A$="N" THEN C=1:GOTO 190
 70 REM SEARCH
 80 INPUT"SEARCH MODE (1:NAME 2:ZIP 3:CITY 4:STATE)";B
 90 IF B=1 OR B=2 OR B=3 OR B=4 THEN 100 ELSE 80
100 INPUT"COMPARISON DATA";B$
110 FOR I=1 TO A
120 A1$=FL(0,I,B)
130 IF B$=A1$ THEN 140 ELSE 170
140 FOR D=1 TO 6
150 A2$(I,D)=FL(0,I,D)
160 NEXT D:GOTO 170
170 NEXT I
180 C=2
```

```
190 INPUT"1:TEL 2:LABEL 3:END";D
200 IF D=1 AND C=1 THEN 270
210 IF D=2 AND C=1 THEN 340
220 IF D=1 AND C=2 THEN 490
230 IF D=2 AND C=2 THEN 540
240 IF D=3 THEN 640
250 GOTO 190
260 REM TEL LIST
270 FOR I=1 TO A
280 C$=FL(0,I,1)
290 D$=FL(0,I,6)
300 IF C$="" THEN 320
310 LPRINT C$;TAB(15);D$
320 NEXT I:GOTO 640
330 REM LABEL
340 FOR I=1 TO A
350 C$=FL(0,I,1)
360 LPRINT TAB(4);C$
370 C$=FL(0,I,5)
380 IF C$="" THEN 470
400 LPRINT TAB(4);C$
410 C$=FL(0,I,3)
420 D$=FL(0,I,4)
430 C2$=FL(0,I,2)
440 LPRINT TAB(4);C$;",";D$;" ";C2$
450 LPRINT
460 LPRINT
470 NEXT I:GOTO 640
480 REM TEL LIST
490 FOR I=1 TO A
500 IF A2$(I,J)="" THEN 640
510 LPRINT A2$(I,J);TAB(15);A2$(I,6)
520 NEXT I:GOTO 640
530 REM LABEL
540 FOR I=1 TO A
550 IF A2$(I,2)="" THEN 630
560 LPRINT:LPRINT
570 LPRINT TAB(7);A2$(I,1)
580 LPRINT TAB(7);A2$(I,5)
590 LPRINT TAB(7);A2$(I,3);
600 LPRINT",";A2$(I,4);
610 LPRINT" ";A2$(I,2)
620 LPRINT
630 NEXT I
640 END
```


Notes:

a) When varying the data length, reset its CETL format.
b) When varying the addressing format, change both program sections 330 through 470 and 530 through 630 together.
c) A TEL listing can be displayed on the display panel by changing all LPRINT statements in program sections 260 through 320 and 480 through 520 to PRINT statements.
d) If you list addresses by phone number (TEL) or address mail after sorting the address file with a memory move, the addresses are listed or printed in sorted order.

(Example) Alphabetical addressing by name
T command from CETL

NAME	ZIP	CITY	STATE	ADDRESS	TEL
BILL KING	52551	FAIRFIELD	IOWA	85 NORTH 2nd AVENUE	515-777-4444
CASIO	160	SHINJUKU	TOKYO	2-6 NISHISHINJUKU	03-345-9999
KEN FARMER	10031	NEW YORK	NEW YORK	6 EAST 43rd STREET	800-344-6434
MARK SMITH	92122	SAN DIEGO	CALIFORNIA	7000 ERLANGER STREET	714-621-8000
PAUL PERKINS	93274	SIMI VALLEY	CALIFORNIA	2807 ELIZONDO	805-652-3222

Addressing Example (Alphabetically by name)

```
BILL KING
85 NORTH 2nd AVENUE
FAIRFIELD,IOWA 52551


CASIO
2-6 NISHISHINJUKU
SHINJUKU,TOKYO 160


KEN FARMER
6 EAST 43rd STREET
NEW YORK,NEW YORK 10031


MARK SMITH
7000 ERLANGER STREET
SAN DIEGO,CALIFORNIA 92122


PAUL PERKINS
2807 ELIZONDO
SIMI VALLEY,CALIFORNIA 93274
```

3. Golf tournament scoring table (for up to 90 players .... CETL and BASIC)

Golf scoring calculations are annoying. Sometimes, a preoccupation with scoring details takes our mind off the game.
This program can help. It can show you your ranking after entering only the OUT and IN course scores and members' handicaps (HD).
Now, enjoy golfing by making full use of this program!

Key entries to CETL

Save any necessary data before resetting the FP-200. Then enter:

RESET ↵

After resetting the FP-200, enter the BASIC program first and the CETL data next:
Select CETL mode and enter:

N ↵

Then, define a CETL table by responding to the following questions from CETL:

```
File Name:GOLF 1

   Number of Rec.:20 ———— Enter the number of records – members.
   Number of Item:6 ———————— Define the table size.


Label Record
Item1:NAME
   Type(N/S)   :S
   Expression ?
   Format      :&      &

Item2:OUT
   Type(N/S)   :N
   Expression ?
   Format      :####

Item3:IN
   Type(N/S)   :N
   Expression ?
   Format      :####

Item4:GROSS
   Type(N/S)   :N
   Expression :IT(2)+IT(3)
   Format      :#####
```



```
Item5:HD
   Type(N/S)   :N
   Expression ?
   Format      :###

Item6:NET
   Type(N/S)   :N
   Expression :IT(4)-IT(5)
   Format      :####
```

This completes the table definition.
SAVE the BASIC program and PUT the CETL table at this time.

Data entry

Go to cell 1-1 in CETL mode and begin data entry from that cell.

	J ↵	Enter the Jump command.
Rec. ?	1 ↵	Select a record,
Item ?	1 ↵	Select an item.

Data entry example. (Names may be up to 8 characters long)

```
Data Area
1-1 :A
1-2 :46
1-3 :47
1-4 ?
1-5 :14
1-6 ?

2-1 :B
2-2 :50
2-3 :57
2-4 ?
2-5 :18
2-6 ?

3-1 :C
3-2 :54
3-3 :53
3-4 ?
3-5 :17
3-6 ?
    |
11-1 :K
11-2 :56
11-3 :55
11-4 ?
11-5 :20
11-6 ?
```

```
12-1 :L
12-2 :72
12-3 :70
12-4 ?
12-5 :34
12-6 ?

13-1 :M
13-2 :74
13-3 :62
13-4 ?
13-5 :36
13-6 ?

14-1 :N
14-2 :52
14-3 :61
14-4 ?
14-5 :16
14-6 ?

15-1 :O
15-2 :51
15-3 :47
15-4 ?
15-5 :7
15-6 ?
```

```
16-1 :P
16-2 :52
16-3 :48
16-4 ?
16-5 :18
16-6 ?

17-1 :Q
17-2 :48
17-3 :45
17-4 ?
17-5 :10
17-6 ?

18-1 :R
18-2 :43
18-3 :45
18-4 ?
18-5 :5
18-6 ?

19-1 :S
19-2 :50
19-3 :65
19-4 ?
19-5 :18
19-6 ?
```

```
20-1 :T
20-2 :46
20-3 :44
20-4 ?
20-5 :2
20-6 ?
```

If you make any errors, correct them by using the J command.
No problem results if the number of participants suddenly increases or decreases on the day of the tournament. You can adjust the number of participants by using the I or D command immediately before the tournament begins.
After data entry has been completed, compute the gross scores and handicaps of the participants:

	C ↵	Enter the C command.
Rec.?	1 . ↵	Instruct CETL to process from record 1 to the last record.
Item?	2 . ↵	Instruct CETL to process from item 2 to the last item.

This completes the computations.
Now sort the table:

S ↵	Enter the Sort command.
Key Item? "NET" ↵	
Up/Down(U/D)? U ↵	
Rec. ? 1, 20 ↵	
Mem. move(Y/N)? Y ↵	

This will complete the sorting by score. If a printer is available, print the results:

	T ↵	Examine the table.
Rec. ?	, ↵	Instruct CETL to output all the records.
Item ?	, ↵	Instruct CETL to output all the items.
Printer(Y/N)?	Y ↵	Instruct CETL to send output to the printer.

Printed example

NAME	OUT	IN	GROSS	HD	NET
A	46	47	93	14	79
P	52	48	100	18	82
F	64	59	123	40	83
Q	48	45	93	10	83
R	43	45	88	5	83
E	59	60	119	34	85
D	50	44	94	8	86
T	46	44	90	2	88
B	50	57	107	18	89
G	50	50	100	11	89
C	54	53	107	17	90
K	56	55	111	20	91
O	51	47	98	7	91
H	62	57	119	26	93
S	50	65	115	18	97
N	52	61	113	16	97
M	74	62	136	36	100
L	72	70	142	34	108
J	61	60	121	11	110
I	71	70	141	30	111

This program can also display the results on the display panel if no printer is available. Select BASIC mode and press:

PF4

or enter:

RUN ↵

The BASIC program will ask you for the number of participants to be listed:

Rec. ?	20	Enter the number of participants.

This will begin an alphabetic display of the participants and their scores on the panel.

		NAME	NET	HD
1	:	A	79	14
2	:	P	82	18
3	:	F	83	40
⋮		⋮	⋮	⋮

When all the participants have been displayed, the program will ask you to select either special prize displays or the previous participant listing:

```
SPECIAL PRIZE (Y/N) ?
```

N ↵ will display the participant listing again.
Y ↵ will display the special prizes:

The program will first display the best gross score prize:

```
** BEST     GROSS **
   R          88
```

Then it displays the first, second and third prizes, followed by the fifth prize, ten prize, and finally the boobie prize in the format shown in the following example.

```
    : NAME      NET     HD
1   : A         79      14
2   : P         82      18
3   : F         83      40
    :           :       :
19  : J         110     11      Boobie prize
```

Here you will be asked the final question:

```
END (Y/N) ?
```

Enter:

N ↵

If you want to repeat the prize displays, enter:

Y ↵

To terminate the program.
All these data can also be saved:
Return the FP-200 to CETL mode and enter:

P ↵

CETL will respond with:

```
OUT (F/S/C) ? _ ↵
```


Select either the floppy disk or cassette tape as the saving device by responding with F (floppy disk) or C (cassette tape).
The prize winner listing may be saved for reference in a later tournament.

BASIC program

10 – 110	Reads CETL data.
120 – 150	Displays all data on the display panel.
160 – 260	Searches for and displays the best gross score.
270 – 400	Displays other prizes.
410 – 470	PRINT subroutine.

Variable table

J, I, L	Loop counters
K	Number of data items entered
A$( )	Names
K (K, 2)	Scores
B	Used for item search
D$	Used for decision
N	Prize counter

Note

This program assumes that the CETL data resides in file 0, as referenced by the FL function calls used in statements 80, 100, 180, and 200; i.e., the "file specification" (the first parameter) is 0 in all the FL function calls. If the CETL data were in file 5, for example, the function calls in statement 80 should be FL (5, J, B).

BASIC program listing

```
 10 CLS:CLEAR
 20 INPUT"RC.";K
 30 DIMA$(K):DIMK(K,2)
 40 FOR J=1 TO K
 50 FOR I=1 TO 2
 60 IF I=1 THEN B=6
 70 IF I=2 THEN B=5
 80 K(J,I)=FL(0,J,B)
 90 NEXT I
100 A$(J)=FL(0,J,1)
110 NEXT J
120 PRINT"  :NAME     NET HD"
130 FOR J=1 TO K
135 IF K(J,1)=0 THEN 150
140 GOSUB 410
150 NEXT J
160 INPUT "SPECIAL PRIZE(Y/N)";D$
170 IF D$="Y"THEN 180 ELSE 120
```

```
180 B(1)=FL(0,1,4)
190 FOR I=2 TO K
200 B(2)=FL(0,I,4)
205 IF B(2)=0 THEN 220
210 IF B(2)<B(1) THEN B(1)=B(2):A=I
220 NEXT I
230 CLS
240 PRINT" ** BEST GROSS **":PRINT:PRINT
250 PRINT"** "+A$(A);" **";TAB(12);B(1)
260 FOR I=1 TO 800:NEXT I
270 CLS
280 PRINT"  :NAME     NET HD"
290 FOR J=1 TO 3
300 GOSUB 410
310 NEXT J
320 N=K/5:N=INT(N)
330 FOR J=5 TO N*5 STEP 5
335 IF K(J,1)=0 THEN 350
340 GOSUB 410
350 NEXT J
360 J=K-1
365 IF K(J,1)=0 THEN 380
370 GOSUB 410
380 INPUT"END(Y/N)";D$
390 IF D$="Y" THEN 400 ELSE 230
400 END
410 REM SUB PRINT
420 PRINT USING"##";J;:PRINT":";
430 PRINT USING"&      &";A$(J);
440 PRINT USING"####";K(J,1);
450 PRINT USING"###";K(J,2)
460 FOR L=0 TO 300:NEXT L
470 RETURN
```

847 bytes

Notes:

Modifications for one and half rounds.

- **CETL**

```
                    I↵
Rec/Item(R/I)?      I↵
Item?               3↵

Item4:OUT           ↵
Type(N/S):N         ↵
Expression?         ↵
Format:     ####↵

Press the BREKA key.
```

```
                    J↵
Rec.?               0↵
Item?               5↵

Item5:? GROSS       ↵
Type(N/S):N         ↵
Expression: IT(2)+IT(3)+IT(4)↵
Press the BREAK key.

                    J↵
Rec. ?              0↵
Item?               7↵

Item7: NET          ↵
Type(N/S):N         ↵
Expression: IT(5)-IT(6)↵
Press the BREAK key.
```

- **BASIC**

```
 60 IF    I=1 THEN B=7
 70 IF    I=2 THEN B=5
180 B(1)=FL (0,1,5)
200 B(2)=FL (0,1,5)
```

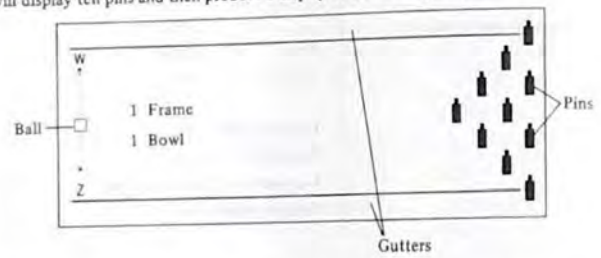
The display panel scrolling speed can be changed by changing the value of 300 in line 460.

4. Bowling

This program displays a bowling lane and ten pins filling up the display panel. You guide the "ball" but the program affects the "ball" in numerous ways, by curving it in either direction or letting it go straight depending on a calculation that uses random numbers, so that it is very hard for you to estimate the "ball's" path. The program produces an animated display of the pins falling down when hit by the ball, so you can enjoy the feeling of really having a strike. It also produces a clearly readable display of your score card.

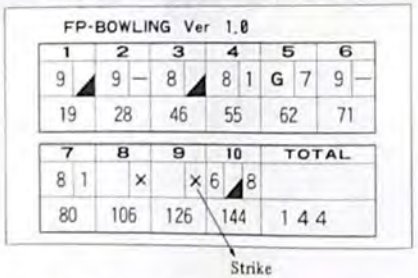
Rules and data entry

Press: PF4 or enter: RUN ↵
This will display ten pins and then produce a display of a ball and gutters:



Selecting a course

Position the ball where you like by using the **W** and **Z** keys. The **W** key moves the ball upward while the **Z** key moves it downward.
Throw the ball by pressing the ↵ key. If any pins remain standing, the program will display the word "BOWL" to request the second ball. Bowl again in the same way.
When you finish the first frame, "Push (RETURN)" is displayed. In response to this, press the ↵ key. This will clear the display and display a score table. Examine the table for correct scoring and press the ↵ key. Repeat this series of operations for subsequent frames. When you finish the tenth frame, the total score is displayed:



You can terminate the game here by pressing the ↵ key. The display will be cleared.

Variable table

A( )	Points (score)
A$( )	Points (Display characters and symbols such as strike, etc.)
P( )	Pin flag
G	Number of pins on the first ball
F	Numbers of pins on the second ball
M	Number of pins on the third ball
O$	Data used by MID$
O	Scoring control flag

- This program requires a memory of 4K bytes. Add a RAM pack or increase the BASIC memory size by using the AREA statement before entering the program.

BASIC program listing

```
1000 CLS:CLEAR:RANDOMIZE:RESTORE:INIT(0,0),1,1
1010 DIM A(10),A$(10),P(10):L=0:U=0:V=0:W=0:X=0:O=-1E4
1100 FOR K=1 TO 10:T=0:F=0:G=0:GOSUB 2000:M=0
1110 IF K=10 AND F+G>=10 THEN R=G:S=F:GOSUB 2000:GOTO 1600
1120 O$="G123456789 ":A$(K)=MID$(O$,G+1,1)
1130 O$="-123456789d":A$=MID$(O$,F+1,1)
1140 IF F+G=10 THEN A$="d":IF G=10 THEN A$="p"
1150 A$(K)=A$(K)+A$+" "
1200 A=U+V:L=L+A
1210 IF A=10 THEN L=L+W:IF U=10 THEN L=L+X:IF W=10 THEN L=L+G
1220 IF K>2 THEN A(K-2)=L
1230 A=W+X:B=L+A
1240 IF A=10 THEN B=B+G:IF W=10 THEN B=B+F:IF G=10 THEN B=B+O
1250 IF K>1 THEN A(K-1)=B
1260 B=B+F+G+M:IF F+G=10 THEN B=B+O
1270 A(K)=B:U=W:V=X:W=G:X=F
1300 CLS:LOCATE 2,0:PRINT"FP-BOWLING Ver1.0";
1310 FOR A=1 TO 6:LOCATE 3*A-2,1 :PRINT A;:NEXT
1320 FOR A=7 TO 10:LOCATE 3*A-20,5:PRINT A;:NEXT
1330 LOCATE 14,5:PRINT"TOTAL";:DUAD(8,7)-(152,31)
1340 QUAD(8,15)-(152,23):QUAD(8,39)-(152,63):QUAD(8,47)-(152,55)
1350 FOR A=32 TO 128 STEP 24:DRAW(A,7)-(A,31)
1360 IF A<90 THEN DRAW(A,39)-(A,63)
1370 NEXT:DRAW(112,39)-(112,63)
1380 FOR A=20 TO 140 STEP 24:DRAW(A,15)-(A,23):IF A<95 THEN DRAW(A,47)-(A,55)
1390 NEXT:DRAW(104,47)-(104,55)
1400 FOR A=1 TO K:B=A*3-2:C=2:IF A>6 THEN B=B-18:C=6
1410 LOCATE B,C:PRINT LEFT$(A$(A),1);
1420 LOCATE B+2,C:PRINT MID$(A$(A),2,1);
1430 IF A(A)>=0 THEN LOCATE B-(A=10),C+1:PRINT USING"###";A(A);
1440 IF A=10 THEN 1470
1450 QUAD(B*8,C*8-1)-(B*8+24,B*C+15)
1460 DRAW(B*8,C*8+7)-(B*8+24,C*8+7):GOTO 1520
1470 LOCATE 13,6:PRINT MID$(A$(A),3,1);
1480 QUAD(80,47)-(112,63):DRAW(80,55)-(112,55),(104,47)-(104,55)
1490 LOCATE 14,7:PRINT USING"####";A(A);
1500 DRAW(80,55)-(152,55),(104,47)-(104,55)
1510 QUAD(80,47)-(112,55):QUAD(112,55)-(152,63)
1520 NEXT A:GOSUB 3000
1530 A$=INKEY$:IF A$<>CHR$(13) THEN 1530
1540 NEXT K:CLS:END
1600 O$="G123456789p":IF R=10 THEN S=G:G=F
1610 IF R+S<20 THEN 1630
1620 GOSUB 2000:A$(K)="pp"+MID$(O$,G+1,1):GOTO 1670
1630 A$=MID$(O$,R+1,1)
1640 IF R<>10 THEN A$(K)=A$+"d"+MID$(O$,G+1,1):GOTO 1670
1650 A$=A$+MID$(O$,S+1,1):O$="-123456789d"
1660 A$(K)=A$+MID$(O$,G+1,1)
1670 O=0:M=G:F=S:G=R:GOTO 1200
2000 CLS:FOR A=0 TO 3:FOR B=-A TO A STEP2:C=132+B*A:D=33+B*B
2010 DRAW(C,D-7)-(C,D):QUAD(C-1,D-5)-(C+1,D):NEXT B,A
2020 FOR A=0 TO 9:P(A)=1:NEXT:T=T+1
2030 DRAW(0,5)-(151,5),(0,58)-(151,58):GOSUB 3000
2040 H=1:GOSUB 4000:H=0:LOCATE 1,2
2050 IF C=4 OR C=63 THEN PRINT"GUTTER":GOTO 2090
```

```
2060 GOSUB 5000:LOCATE 1,2:G=10-B
2070 IF G=10 THEN PRINT "STRIKE !!":GOTO 2160
2080 PRINT G;"PIN"
2090 GOSUB 3000:T=T+1:IF T>3 THEN 2160
2100 H=2:GOSUB 4000:IF C=4 OR C=63 THEN 2130
2110 A=(C-6)/8:IF P(A)+P(A+.5)+P(A-.5)>0 THEN 2140
2120 IF P(A+5)+P(A+4.7)+P(A+5.3)>0 THEN 2140
2130 LOCATE 1,2:PRINT"MISS":GOTO 2160
2140 H=.3:GOSUB 5000:F=10-G-B:IF F=0 THEN 2130
2150 LOCATE 1,2:PRINT F;"PIN"
2160 GOSUB 3000:LOCATE 1,3:PRINT"Push [RETURN]"
2170 IF INKEY$<>CHR$(13) THEN 2170 ELSE RETURN
3000 FOR A=0 TO 50:B$=INKEY$:NEXT:RETURN
4000 Z=31:C=Z:LOCATE 1,3:PRINT K;"Frame"
4010 LOCATE 1,4:PRINT T;"BOWL"
4020 QUADC(1,Z-4)-(5,Z):QUAD(1,C-4)-(5,C):Z=C
4030 A$=INKEY$:IF A$="M" THEN C=C+5*(C>11):GOTO 4020
4040 IF A$="Z" THEN C=C-5*(C<54):GOTO 4020
4050 IF A$<>CHR$(13) THEN 4030
4060 LOCATE 1,3:PRINT"                ":LOCATE 2,4:PRINT"          "
4070 LOCATE 1,2:PRINT"              "
4080 D=SGN(RND(1)*8+22-C)*(0.2+TAN(RND(1)*180-90)/15)*1.5/H
4090 FOR B=1 TO 25:A=B*5+5:C=C+D:IF C <=10 THEN C=4 ELSE IF C>=57 THEN C=63
4100 QUADC(A-9,Z-4)-(A-5,Z):QUAD(A-4,C-4)-(A,C):Z=C
4110 NEXT:QUADC(A-4,C-4)-(A,C):RETURN
5000 FOR A=0 TO 6:B=9.5+B*A
5010 Z=1.85-ABS(B-C-SGN(D)/8)/16-H+ABS(D/3)
5020 IF (Z+0.1)*(Z+0.1)+0.1>RND(1) THEN P(A)=0
5030 IF A>1 AND A<5 AND Z*Z+0.1>RND(1) THEN P(A+5)=0
5040 NEXT
5050 D$="3021120314253632233 4"
5060 FOR A=0 TO 9:IF P(A)=1 THEN 5090
5070 B$=MID$(D$,A*2+1,1):B=VAL(B$):C$=MID$(D$,A*2+2,1):C=VAL(C$)
5080 LOCATE 16+B,C:PRINT" ":
5090 NEXT:B=0
5100 FOR A=0 TO 9:B=B+P(A):NEXT:RETURN
```

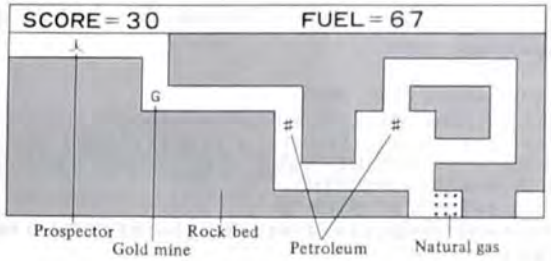
Notes:

1000 –	Initialization
1100 –	The beginning of the main routine
1120 –	Scoring characters for frames 1 through 9
1200 –	Score computation
1300 –	Scoring table display
1400 –	Result entry
1470 –	Subroutine for 10th frame score entry
1540 –	Program termination
1600 –	10th frame scoring characters
2000 –	Used for ball control
3000 –	Key buffer filler subroutine
4000 –	Ball display generation subroutine
5000 –	Pin processing subroutine

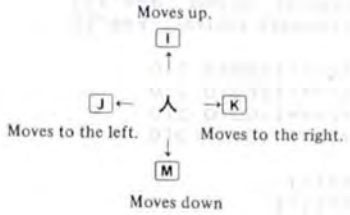
5. Gold mine

Three prospectors start searching for gold veins and petroleum deep in the earth. Dig your way to the left and right or up and down, into and through rock beds and find the resources before your fuel is exhausted. If the prospectors encounter a gas blowoff, they will disappear from the display.

Panel display



Rules and data entry



Press: PF4 or enter: RUN ↵

A title will appear and the game will begin. The symbol "人" is a prospector. Move the prospectors left and right or up and down with the **I**, **M**, **J**, and **K** keys. Gold mine "G" and petroleum "#" are waiting to be discovered. If you hit a gold mine, 10 points are added to the score. If you find petroleum, 20 units are added to the fuel count. When the entire rock bed is gone, the display is automatically cleared and a new rock bed will appear. The game will terminate when all three prospectors disappear or when the fuel decreases to 0.

Variable table

A (1, ..., 18)	Locations of gold veins, petroleum, and natural gas
B$	Used for INKEY$ entry.
X	X coordinate of a "人"
Y	Y coordinate of a "人"
X1	X coordinate of the location where a "人" is digging
Y1	Y coordinate of the location where a "人" is digging
S	Score
F	Fuel
P	Number of remaining prospectors
I	Loop counter for the FOR – NEXT statement
J	Timing loop counter

BASIC program listing

```
 10 REM ***INITIAL SET
 20 CLEAR:RANDOMIZE:DIM A(18):F=100:P=3:CLS
 30 FOR I=1 TO 140:PRINT "▒";:NEXT:GOSUB 480
 40 LOCATE 3,3:PRINT "GOLD MINE SEARCH":GOSUB 480
 50 REM ***MAIN ROUTINE
 60 FOR I=1 TO 17 STEP 2
 70 A(I)=INT(RND(1)*19)+1:A(I+1)=INT(RND(1)*6)+1
 80 NEXT
 90 CLS:LOCATE 0,1:PRINT "人";
100 FOR I=1 TO 138:PRINT "▒";:NEXT
110 X=0:X1=0:Y=1:Y1=0
120 LOCATE 0,0:IF F<=0 THEN 410
130 PRINT "SCORE=";:PRINT USING "###";S;
140 PRINT "  FUEL=";:PRINT USING "###";F
150 B$=INKEY$
160 IF B$="K" THEN X1=X+1:GOTO 210
170 IF B$="J" THEN X1=X-1:GOTO 210
180 IF B$="M" THEN Y1=Y+1:GOTO 210
190 IF B$="I" THEN Y1=Y-1:GOTO 210
200 GOTO 150
210 IF X1<0 THEN X1=X1+1
220 IF X1>19 THEN X1=X1-1
230 IF Y1<1 THEN Y1=Y1+1
240 IF Y1>7 THEN Y1=Y1-1
250 LOCATE X,Y:PRINT " ";:F=F-1:X=X1:Y=Y1
260 IF X=19 AND Y=7 THEN 320
270 LOCATE X,Y:PRINT "人";:LOCATE X,Y
280 IF X=A(1) AND Y=A(2) THEN P=P-1:GOTO 340
290 FOR I=3 TO 17 STEP 2
300 IF X=A(I) AND Y=A(I+1) THEN 450
310 NEXT:GOTO 120
320 PRINT "THIS MINE ALREADY EMPTY! GO AHEAD TO THE
    NEXT"
330 S=S+5:GOSUB 480:GOTO 60
```



```
340 FOR I=1 TO 5:PRINT "人";:FOR J=1 TO 99:NEXT
350 LOCATE X,Y:PRINT "▒";:FOR J=1 TO 99:NEXT
360 LOCATE X,Y:NEXT:CLS
370 LOCATE 0,2:PRINT "GASS ACCIDENT OCCURED!!"
380 GOSUB 480:LOCATE 0,5
390 PRINT "YOU HAVE";P;"MEMBERS NOW":GOSUB 480
400 IF P>0 THEN 60
410 CLS:LOCATE 5,2:PRINT "GAME OVER":GOSUB 480
420 LOCATE 3,5:PRINT "YOUR SCORE";S:GOSUB 480
430 END
440 REM ***SUB ROUTINE
450 A(I)=0:A(I+1)=0
460 IF I<9 THEN PRINT "#";:F=F+20:GOTO 120
470 PRINT "G";:S=S+10:GOTO 120
480 FOR I=1 TO 300:B$=INKEY$:NEXT:RETURN
```

Notes:

The key buffer clearing and timing subroutines are on line 480.
If the initial fuel amount is too low, increase the value of F in line 20. It may be fun to modify statements 330, 460 and 470, which increment the fuel and score.

■ Character Code Table

(*) 01 code → COMMAND MENU
Characters not shown in this table are output as blank.
The 7F and the FF code cannot be printed on the printer.

■ Troubleshooting

If any of the following abnormal symptoms is observed, check the items in the corresponding list.

Symptom	Cause	Item to be checked	Solution
No display appears on the LCD display panel with the power switch on.	The contrast adjustment control is set at the minimum.	Check to see if the control is set at the minimum.	Turn the contrast adjustment control to the optimum setting.
	No main batteries are installed.	Check whether the batteries are properly installed.	Install new batteries.
	The main batteries are not properly installed.	Check whether the batteries are properly installed.	Install the batteries properly.
	The main batteries are exhausted.	Check whether the batteries are too old. (Refer to "Battery Life" in the FP-200 specifications.)	Replace the old batteries with new ones.
	The AC adapter cord is not properly plugged in.	Check whether the cord is disconnected.	Insert the plug into the receptacle until it comes to a complete stop. If the plug is still loose, replace the receptacle with a new one.
	A poor power cord connection at the receptacle due to dirt on the plug.	Check whether there is dirts on the plug or receptacle. Remove and insert the plug a few times.	Remove the dirt on the plug or receptacle.
"Memory Illegal" appears with the power switch on.	No back-up batteries are installed.	Check whether back-up batteries are installed.	Install new batteries.
	The back-up batteries are improperly installed.	Check whether the back-up batteries are properly installed.	Install the batteries properly.
	The back-up batteries are exhausted.	Check whether the back-up batteries are too old. (Refer to "Battery Life" in the FP-200 Specifications.)	Replace the batteries with new ones.
	One or more RAM packs are removed.	Check whether any RAM pack or packs have been removed.	Enter the RESET command.
	The memory contents have varied.		Enter the RESET command.
"FC Error" remains displayed and no key entry is possible with the power on.	The memory contents have varied.		Press the RESET switch for about 1 second, turn off the FP-200, then turn it on again. The FP-200 will be ready to accept a key entry. Enter the RESET command.
The FP-200 malfunctions intermittently.	The FP-200 is affected by external noise.	Check whether the ground wires of peripheral devices are connected to the ground terminal on the FP-200.	Connect the ground wire of each device to the ground terminal on the FP-200.

Symptom	Cause	Item to be checked	Solution
The FDD (FP-1021FD1) does not operate.	Power is not supplied to the FDD.	Check whether the FDD power indicator lamp is off.	Supply power to the FDD.
The CMT REMOTE function does not operate.	The REMOTE switch is OFF.	Check whether the REMOTE switch is OFF.	Turn the REMOTE switch ON.
The display suddenly disappears.	The main batteries are exhausted.	Check whether batteries are old.	Replace the batteries with new ones.
	The APO (Auto Power Off) function is enabled.		Press the ON key.
The printer does not operate.	Power is not supplied to the printer.	Check whether the printer power indicator lamp is off.	Turn the printer on.
	The printer is OFF LINE.	Check whether the printer is OFF LINE.	Set the printer to ON LINE.

■ FP-200 Specifications

CPU : i8085 compatible (3MHz)
Memory : RAM: Standard 8K bytes, Max. 32K bytes
ROM: Standard 32K bytes, Max. 40K bytes
Keyboard : ASCII key arrangement
Display panel : LCD (Liquid Crystal Display)
Capacity
Text: 20 characters x 8 lines (160 characters)
Graphics: 160 x 64 dots
Cassette tape interface : 300 baud, remotely controllable
Electrical characteristics of CMT terminals

MIC terminal:	Output impedance	5 KΩ
	Output voltage	3 mVp-p
EAR terminal:	Input impedance	10 KΩ
	Max. input voltage	10 Vp-p

REMOTE terminal: DC 24V, up to 1A.
Printer interface : Centronics standard 8-bit parallel interface
Serial interface : RS-232C standard, 300 baud
Power : DC Main: 4 AA UM-3 dry cell batteries (used for computation and display)
Memory back up: 2 AA UM-3 dry cell batteries
AC Via a special AC adapter (AD-4180). (Supplied separately.)
Battery life : Main: Approx. 6 hours with AA UM-3
Approx. 11 hours with Alkaline
Memory back-up: Approx. 6 months with AA UM-3
Operating environment : Temperature: 0 – 40°C
Humidity: 20 – 85%
Dimensions : (W) 310 x (D) 220 x (H) 55.5 mm
Weight : 1.54 kg (FP-200 only)

CTX = Text verzenden
B = Ebaar plus naam
L = Laad plus naam
O =
P = Print
Q =
T = Text wissen
V = Voeg toe
∂ =

Basic = SHIFT + CRTL + PF5