HuCAL

for the
SC-3000 SERIES
+
SF-7000

INSTRUCTION MANUAL

SEGA®

HuCAL INSTRUCTION MANUAL

PROLOGUE

It is a plain fact that a computer language is needed in order to operate a computer. For personal computers a language called "BASIC" is mainly used. No matter how easy a language "BASIC" is, anyone can see that a short-term study does not help to formulate programs. Busy people, although they acknowledge the availability of their personal computers, cannot make full use of them. Also, the kind of software on the market is limited, and it is expensive to have a collection of those products.

For the above reasons, a number of non-programming software or simplified languages have been developed lately.

One of them is HuCAL, which is a simplified language of the spread-sheet type. It also has a programming device that uses 12 macro-instructions. It is non-programming software, which exceeds the traditional boundaries of a simplified language and has as wider versatility as BASIC.

Moreover, it's optimum use can be mastered in only a few hours of study since its operations are not as complicated as those of BASIC, and this also makes it possible to formulate a program only in a few hours while it takes a few days to do so in BASIC.

Operations are very easy! You just regard a large

chart of HuCAL as a reporting pad and fill in numbers and letter by using the keyboard instead of a pencil. Of course, you can correct any errors without using an eraser and draw chart lines without a ruler.

HuCAL can be considered as an "almighty calculating machine" equipped with all required elements such as reporting pads, pencil, eraser and ruler.

Well, let's use this HuCAL and try to create diverse applicable software.

CONTENTS

PART I

INTRODUCTION

PART I. INTRODUCTION

1. What to prepare

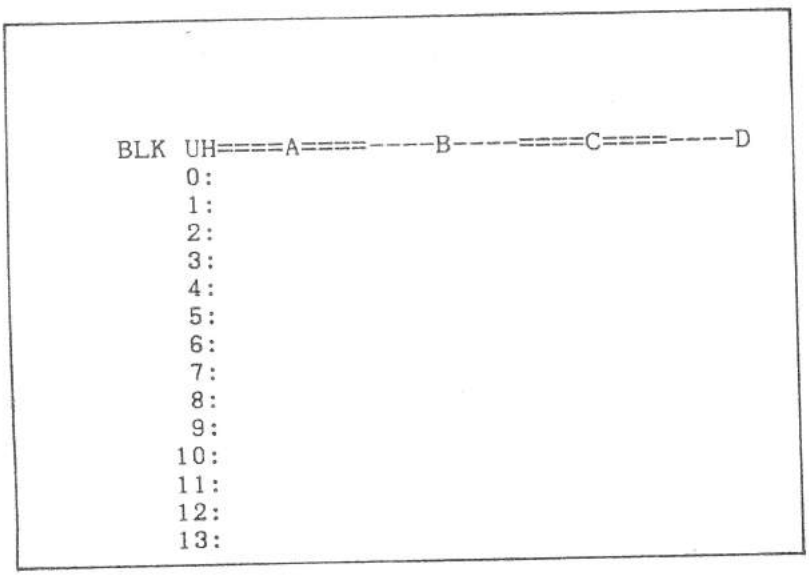
The folowing are needed to operate HuCAL:

- SEGA SC-3000 Main Unit
- SEGA Single Floppy Disk Drive SF-7000
- Monitor TV
- SEGA SP-400 or EPSON RX-80 Printer
  (HuCAL will work without this.)

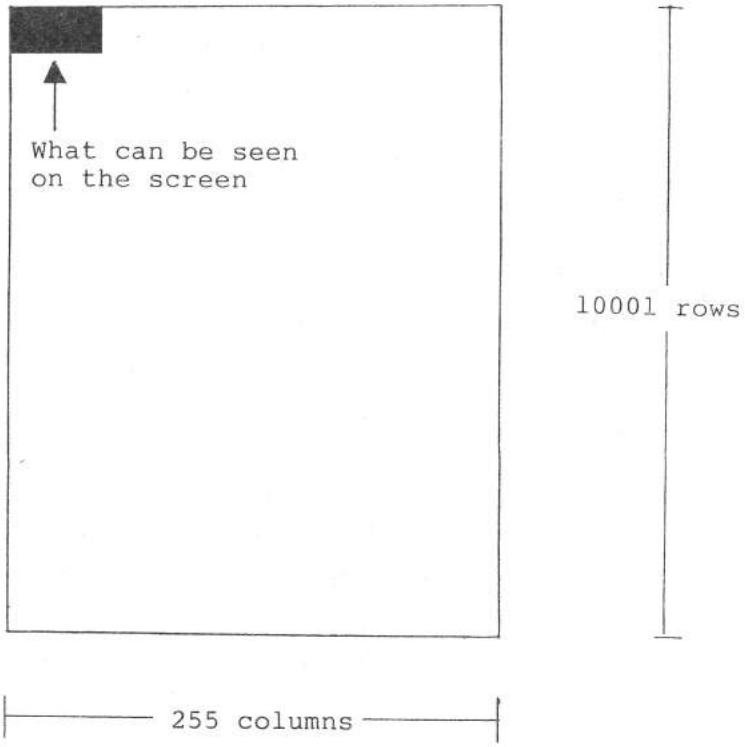
Refer to the appropriate Manuals attached to each equipment for how to connect and operate them.

2. Lets get started

First confirm the appropriate connection of connecting cords or power cords for each equipment. Insert the master diskette of HuCAL into the disk drive unit and turn the power ON. When "HuCAL" is displayed and the cursor begins to flash, press the RETURN key. Then, the following screen will be displayed:



Now a chart has appeared on the display screen, hasn't it? However, HuCAL is supposed to be a much larger chart. Therefore, what you are seeing now is only a small part of it.


What can be seen
on the screen
10001 rows
255 columns

(1) Size of a Chart

The whole chart of HuCAL comprises 255 columns horizontally and 10001 rows vertically and thus a total of 2,550,255 sections. (The size of a chart, however, is limited down to a certain capacity of memory, so its range available for use is about 19 KB.) What can be seen now on the screen is the top left part of the whole chart, and in the first trial the first three columns and the first 14 rows will be covered.

> Note: Capacity of memory is ordinarily measured in bytes. One letter can be input into 1 byte, however, 1024 letters can be input into 1 kbyte since 1kbyte equals 1024 byte.
>
> Therefore, in the case of HuCAL, 20 items of data can be input at 1 kbyte with every 10 letters being input 5 columns (A through E). Of course, as one may realize, the amount of data will increase inversely as the number of columns and thus letters is decreased.

Though it is not visible, a HuCAL Chart is divided by both vertical and horizontal lines, and you can fill in data by moving the cursor to where you want.

However, you may not be able to input to a section which is so far from the section (A, 0). This is due to lack of capacity of memory as mentioned earlier and it is not a malfunction.

(2) Row Number and Column Number

What are shown in a horizontal line as ==A==--B--==C== are column numbers and those shown vertically 0, 1, 2, 3, ... are row numbers. And these horizontal and vertical borders are called frames.

As shown horizontally as A, B, C, ... IT, IU, we have a total of 255 columns and the column numbers are displayed using alphabetic character from A through IU, or using numbers from 0 through 254. You can easily understand each number between 0 and 254 corresponds to each alphabetic character between A and IU, can't you?

> Note: The counting of number always starts with '0'. So, if counted from '0' to '10', that means eleven data and if counted from '0' to '100', that means 101 data.

Vertically, we have a total of 10001 rows as displayed by numbers from 0 through 10,000.

In this way, each section can be designated by calling both a column number and a row number, such as ! (A, 5), !(C, 20), !(0, 100), and so on. Needless to say, !(B, 100) and !(1, 100) are used to specify the same frame. (The exclamation mark (!) means field.)

(3) Field

Each section designated with a combinaion of a column number and a row number is called a field. A user can move a cursor to an intended field and input necessary data. Every field allows input of figures, letters, expressions and macro instructions, however, under certain rules. So please keep them in mind beforehand.

The length of a field (namely, digit number) can be altered freely from 3 to 32 digits, and the number of letters to be input is limited to the number the digit number specifies.

Similarly, figures are also limited to the digit number and if a figure exceeds the number of specified digits, a percentage symbol (%) will appear to indicate this. (But it will be shown correctly on the drum.)

In this case, the proper indication will be offered as the length of a field is adjusted.

Incidentally, expressions and macro instructions are not to be inscribed directly on a field but on a drum as explained in the following paragraph.

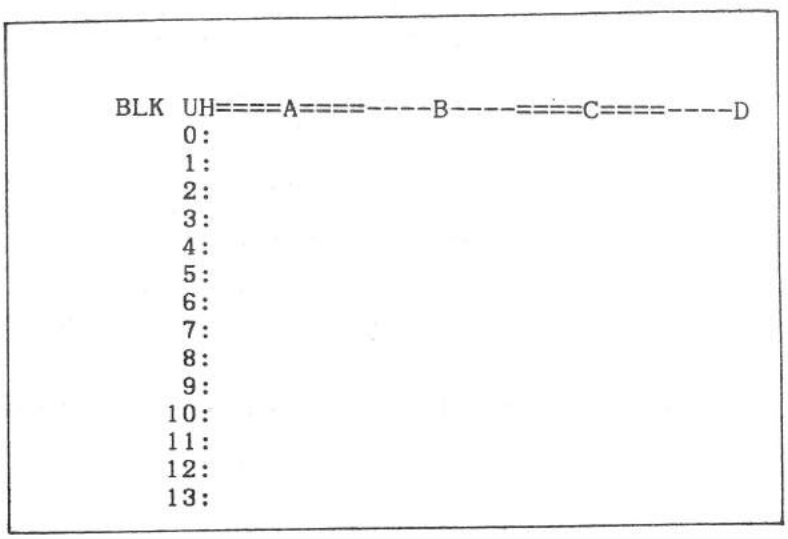
Note: When moving a cursor to input data, the number of digits indicated at the time will be exactly the number of letters allowed to be input. If you want to input up to 32 letters, please extend your field beforehand and then shorten it

after the data has been input.

For letter, the indication will be shortened in the situation as it is, however, if certain figures are input, the whole part will be filled with percentage symbols (%) unless the indication is more than 6 letters, although it is shown as 100000 which is literally 6 letters.

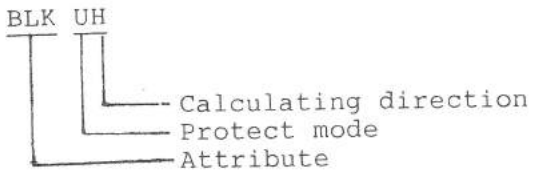
(4) Drum

The top margin comprising 6 rows above the set of row numbers is called a drum. This area is a place where certain information, such as the indication or edition of data in a field or expressions, functions, or macro instructions, should be input. Expressions and macro instructions should not be input directly into a field but into a drum first. The drum's maximum capacity is 233 letters.



(5) Attribute, Protect Mode and Calculating Direction

Now please take a look at the top left part of the frame. It is written "BLK UH", isn't? What does that mean? The part written as "BLK" is for attribute, "U" for protect mode, and "H" for calculating direction.



The list below shows what each single sign means, concerning attribute, protect mode, and calculating direction.

ATTRIBUTE	BLK	Nothing is input in a field where the cursor is located.
	NUM	Certain figures are input in a field where the cursor is located.
	ASC	Certain letters are input in a field where the cursor is located.
	EXP	Certain expressions or functions are input in a field where the cursor is located.
	MAC	Certain macro-instructions are input in a field where the cursor is located.
	FRM	The cursor is located on a frame.

PROTECT MODE	U	A field is in the state of being unprotected. However, a field which has an asterisk (*) between the indication of attribute and protect mode is protected.
	P	A field is being protected. In this mode, all the field into which data has been input will be protected and have an asterisk (*) between the indication of attribute and protect mode.

CALCULATIING DIRECTION	H	To be calculated horizontally.
	V	To be calculated vertically. Vertical calculation takes 100 times as much time as horizontal one. Macro instructions are not affected by calculating directions.

Even if protect mode is at U, a field to which some data has been input will be protected when both [CTRL] and [I] keys are pressed simultaneously.

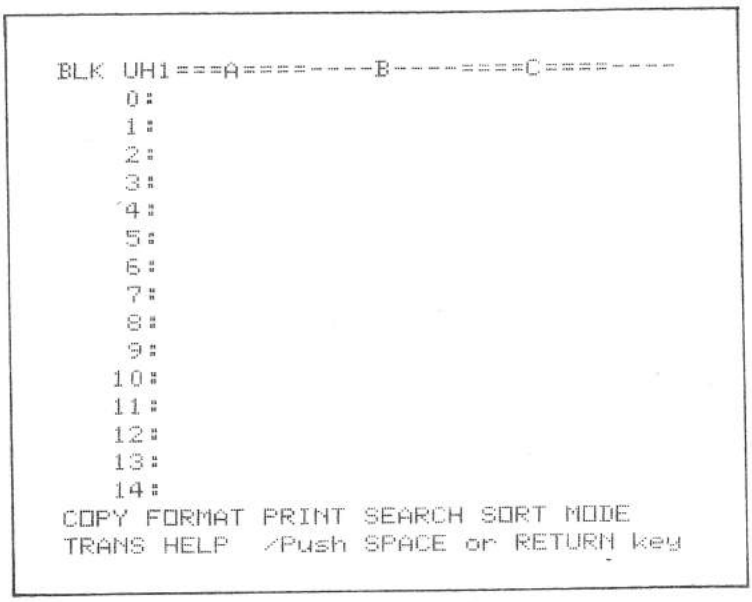
A field will be automatically protected if data is input when its protect mode is at P. Then the protection will not be released even if protect mode is changed from P to U.

If you want to release the protection, press both [CTRL] and [I] keys simultaneously once again at the field you want to clear, and the protection will be thus be released.

Note: Once a field is protected, nothing will be accepted no matter what keys you may press. Protection is most helpful if you do not want your data, such as macro instructions or other important data, to be deleted.

3. Let's Move the Cursor

Now let's look at the screen carefully once again. As you can see, the cursor is now placed at (A, 0). Now let's get moving and try to press the cursor key to input your letters or figures into a field at will.



For correction, each time the DEL key is pressed one letter is deleted if done before moving the cursor. If this is done after having moved the cursor, you must return the cursor to the field to be corrected and re-input new data. The RETURN key can also be used after a field has been filled. For the RETURN key, the cursor

will move in the same direction as before.

Now, please keep on pressing the cursor key [⇨] . You can see that the Chart being scrolled to the left and frames D, E, or so appear, can't you? How about the other one, [⇩] ?

It will be scrolled upward by pressing [SHIFT] + [⇩] or rightward by pressing [SHIFT] + [⇨]. Now, how do we return the cursor to (A, 0), which is now far away from the present position? Push [CLR] key. Then it will return to (A, 0) although it cannot be seen on the screen, and the position the chart will also be retrieved to the original one.

Push [HOME] key if you want to return the cursor not to (A, 0) but to the top left part of the screen. Did it turn out all right?

"I want to delete all the letters and the figures from the field." Then please push [&] key and input [Y] in case that the message comes out. It is all cleared now, isn't it. If deletion is not necessary, input any other keys and the original chart will be retrieved.

4. Selecting a Job

The field has been explained so far, but why don't you look at the row of letters now? There are eight words and a message there. They are read as "COPY", "FORMAT", "PRINT", "SEARCH", "SORT", "MODE", "TRANS" and "HELP", and these are eight kinds of job that will be explained in this section.

At present, a semi-cursor of one character is flashing at the letter "C" in "COPY", isn't it? To move this semi-cursor, use the SPACE key. Choose a job you want.

First of all, here are brief description of each job. This is to introduce you the outine of HuCAL's jobs.

1. COPY=RIGHT, DOWN, MOVE, BLOCK

   ① RIGHT = Copies a data horizontally to the right.

   ② DOWN = Copies a data downward.

   ③ MOVE = Copies a data in a desired field.

   ④ BLOCK = Copies a data by a block as a unit.

2. FORMAT=EXPR, DELETE, BAR, LEFT, RIGHT, CENTER, COMMA, LENGTH, DEC. POS.

   ① EXPR = Enables entering of expressions. The field will be filled with "E".

② DELETE = Deletes all the characters or numbers in a field.

③ BAR = Draws a horizontal line in a field. The cursor moves by one field (to the same direction as the previous movement).

④ LEFT = Shifts the characters in a field to the left.

⑤ RIGHT = Shifts the characters in a field to the right.

⑥ CENTER = Shifts the characters in a field to the center.

⑦ COMMA = Puts commas on numbers every three digits (applicable by a column).

⑧ LENGTH = Determines the length of fields.

⑨ DEC. POS.= Determines the position of a decimal point for number in a field (applicable by a column).

3. PRINT=HCOPY, CODE, MODE, SET, ROW, COLUMN, GO

① HCOPY = Prints out a hard copy of the screen.

② CODE = Defines the control codes for outputting to a printer.

③ MODE = Specifies either pressed lines (PRESS) or enlarged characters (LARGE).

④ SET = Determines "LINE" (number of characters in a line), "PAGE" (number of lines in a page), "LEFT MARGIN" (number of blank characters at the left of the form paper) and "UP MARGIN" (number of blank lines at the top of the form paper).

⑤ ROW = Determines the number of lines in the table to be output to a printer.

⑥ COLMN = Determines the order of columns on printing out.

⑦ GO = Executes printing.

4. SEARCH=Performs searching.

5. SORT=Performs sorting.

6. MODE=AUTOCAL, AUTOADD, PROTECT, FRAMSW, COLOR, FIELD, CLEAR

① AUTOCAL = Sets to auto-calculation mode.

② AUTOADD = Sets to auto-addition mode.

③ PROTECT = Protects data.

④ FRAMSW = Enables moving onto frames from any fields.

⑤ COLOR = Determines background color of the display.

⑥ FIELD = Protects a data in a field.

⑦ CLEAR = Clears out all the data in the whole table.

7. TRANS=SAVE, LOAD, UTILITY, WINDOW

   ① SAVE = Saves the data in a disk.

   ② LOAD = Loads the data from a disk.

   ③ UTILITY = Starts the utility function.

   ④ WINDOW = Starts the window function.

8. HELP = Displays the description of special use of keys.

From now on, let me explain the details and the operations of each job along with examples.

Note: Since examples are used, the explanations follow the order below:

(1) FORMAT
(2) MODE
(3) SORT
(4) PRINT
(5) TRANS
(6) COPY
(7) SEARCH
(8) HELP

It is recommended to remember the order above because it is often used in the actual work for making a table.

(1) FORMAT

[SPACE] (Set to FORMAT) [RETURN]

```
COPY FORMAT PRINT SEARCH SORT MODE
TRANS HELP  /Push SPACE or RETURN key

FORMAT: EXPR DELETE BAR LEFT RIGHT
        CENTER COMMA LENGTH DEC.POS
```

The meaning of each item is as follows:

*EXPR enables entering of expressions. The field will be filled with "E".

[/] functions the same.

*DELETE detetes a data in a field.

[CTRL] + [E] functions the same.

*BAR draws a horizontal line in a field. The direction to which the cursor moves is the same as that of the previous movement.

["] functions the same.

*LEFT shifts the characters in a field to the left.

['] functions the same.

*RIGHT shifts the characters in a field to the right.

[)] functions the same.

*CENTER shifts the characters in a field to the center.

[(] functions the same.

*COMMA puts commas on numbers every three digits. It is usually "ON", so commas will be put on all the numbers over three digits. They will not be put on if turned "OFF".

[CTRL] + [U] functions the same.

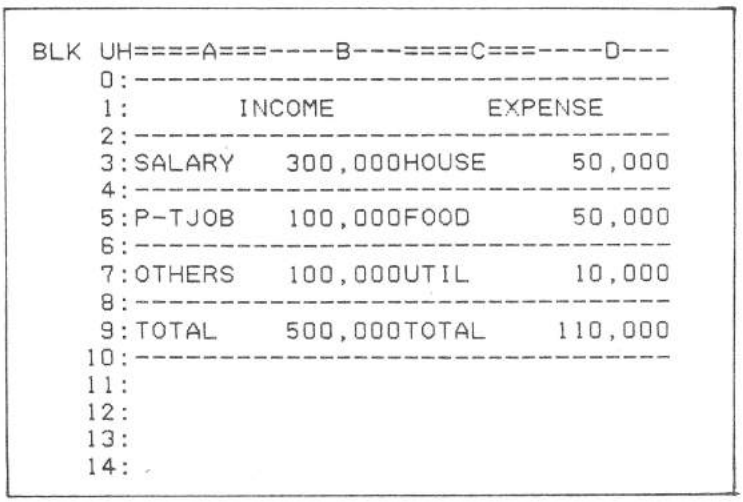
*LENGTH enables changing of the length of fields. All the fields on the right of the field where the cursor stays are changed as specified while those on the left will not be affected. The minimum length of a field is 3 and the maximum one is 31.

[!] functions the same.

*DEC. POS. enables changing the number of digits below a decimal point. When the digit of a number exceeds the length of a field, the field will be filled with "%". (Same with COMMA.)

[CTRL] + [F] functions the same.

In the following pages, the method of making a simplified housekeeping book by using the functions of FORMAT is explained. Move the semi-cursor with SPACE key and press RETURN key where you want.



① LENGTH

Changes the length of fields from column A to column D.

Set the cursor to column A.

[ ] (Set to LENGTH)
[RETURN] FORMAT FIELD LENGTH: 9

Set the cursor to column B.

[RETURN] FORMAT FIELD LENGTH: 9

[1] [2] [RETURN] FORMAT FIELD LENGTH: 12

Set the cursor to column C.

[RETURN] [RETURN] FORMAT FIELD LENGTH: 12

[9] [RETURN] FORMAT FIELD LENGTH: 9

Set the cursor to column D.

[RETURN] [RETURN] FORMAT FIELD LENGTH: 9

[1] [2] [RETURN] FORMAT FIELD LENGTH: 12

② BAR

Draws the horizontal fame lines.

Set the cursor to !(A,0) [RETURN]

[ ] (Set to BAR.) [RETURN]

Set the cursor to !(B,0) [RETURN]

Set the cursor to !(C,0) [RETURN]

Set the cursor to !(D,0) [RETURN]

Continue the similar operation until the end (!(D,10)).

```
BLK UH====A===----B---
    0:----------------
    1:
    2:----------------
    3:
    4:----------------
    5:
    6:----------------
    7:
    8:----------------
    9:
   10:----------------
   11:
```

*Enter the characters next. See Page 122 (FUNC + [E]).

Set the cursor to !(A,1). [I] [N] [C] [RETURN]

Set the cursor to !(B,1). [O] [M] [E] [RETURN]

Set the cursor to !(C,1). [E] [X] [P] [RETURN]

Set the cursor to !(D,1). [E] [N] [S] [E] [RETURN]

```
BLK UH====A===----B---      BLK UH====C===----D---
    0:----------------          0:----------------
    1:INC      OME              1:EXP      ENSE
    2:----------------          2:----------------
    3:                          3:
    4:----------------          4:----------------
    5:                          5:
    6:----------------          6:----------------
```

③ RIGHT

*Since the letters "INC" in !(A,1) and "EXP" in !(C,1) are separated from the letters to follow, shift them to the right in each field.

Set the cursor to !(A,1).

[RETURN] [ ] (Set to RIGHT.) [RETURN]

```
BLK UH====A===----B---====C===----D---
    0:--------------------------------
    1:       INCOME
    2:--------------------------------
    3:
    4:--------------------------------
    5:
    6:--------------------------------
    7:
```

Set the cursor to !(C,1). [RETURN]

```
BLK UH====C===----D---====E===----F--
    0:----------------
    1:      EXPENSE
    2:----------------
    3:
    4:----------------
    5:
    6:----------------
    7:
    8:----------------
    9:
   10:----------------
   11:
```

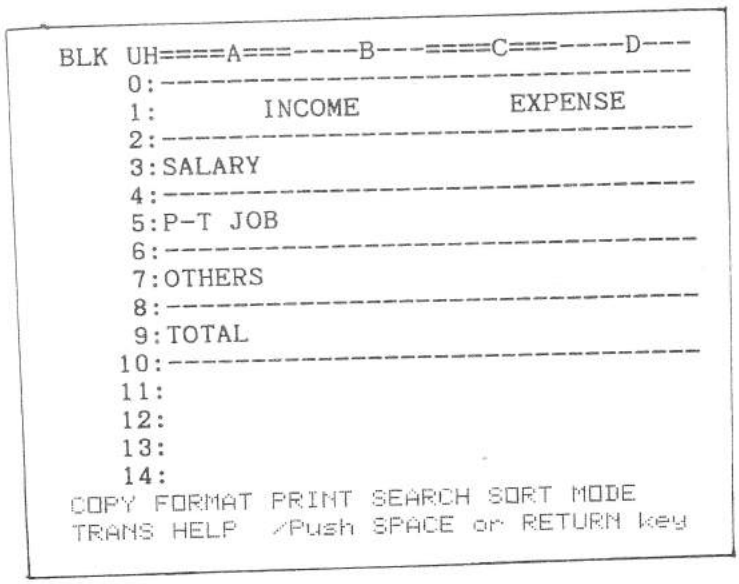
*Enter the rest of the characters.

Set the cursor to !(A,3). [S] [A] [L] [A] [R] [Y]
[RETURN]

Set the cursor to !(A,5). [P] [-] [T] [ ] [J] [O]
[B] [RETURN]

Set the cursor to !(A,7). [O] [T] [H] [E] [R] [S]
[RETURN]

Set the cursor to !(A,9]. [T] [O] [T] [A] [L] [RETURN]



④ CENTER

*Since the item names in column A are placed too left, shift them to the center.

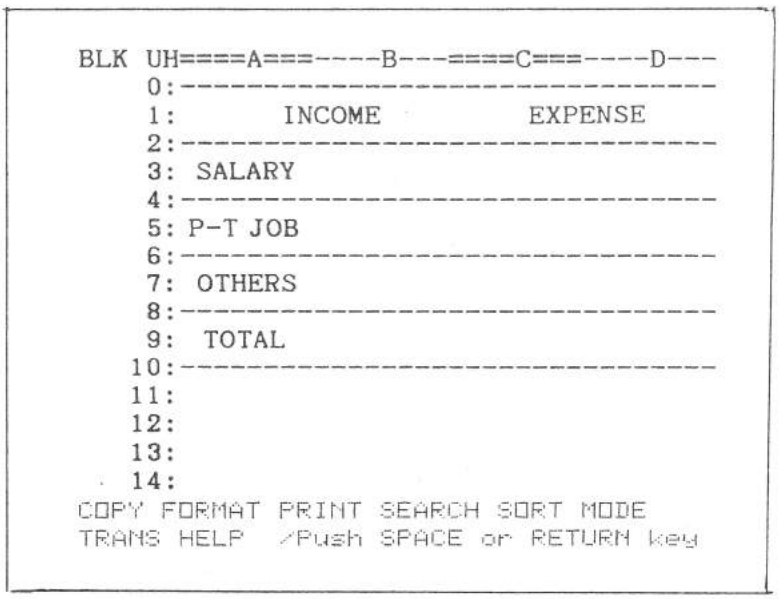
Set the cursor to !(A,3). [RETURN]

[ ] (Set to CENTER.) [RETURN]

Set the cursor to !(A,5). [RETURN] [RETURN]

Set the cursor to !(A,7). [RETURN] [RETURN]

Set the cursor to !(A,9). [RETURN] [RETURN]



⑤ COMMA (You can enter numbers without doing anything because this function is turned "ON" at power-on.)

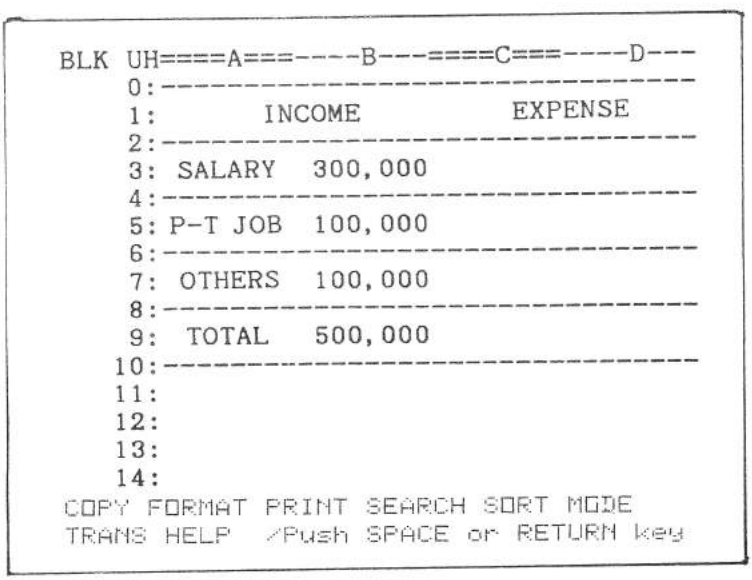
*Enter the numbers (amount of money).

Set the cursor to !(B,3). [3] [0] [0] [0] [0] [0] [RETURN]

Set the cursor to !(B,5). [1] [0] [0] [0] [0] [0] [RETURN]

Set the cursor to !(B,7). [1] [0] [0] [0] [0] [0] [RETURN]

Set the cursor to !(B,9). [5] [0] [0] [0] [0] [0] [RETURN]



*After all the numbers are entered, try turning "OFF" the "COMMA" function by putting the cursor on !(B,3) and selecting "COMMA". Can you see the commas in each of the four fields disappear? Just as you see, COMMA and DEC. POS. for NUM fields are valid for the vertical column.

⑥ DELETE

As a practice and preparation for explanation of the next item, "EXPR", delete the number in !(B,9).

Set the cursor to !(B,9). [ ] (Set to DELETE.)

[RETURN]

How about that?

Note: This function is used to delete the content of the field where the cursor stays. To clear out the whole table, "CLEAR" in the "MODE" function shall be used. (See "(2) MODE".)

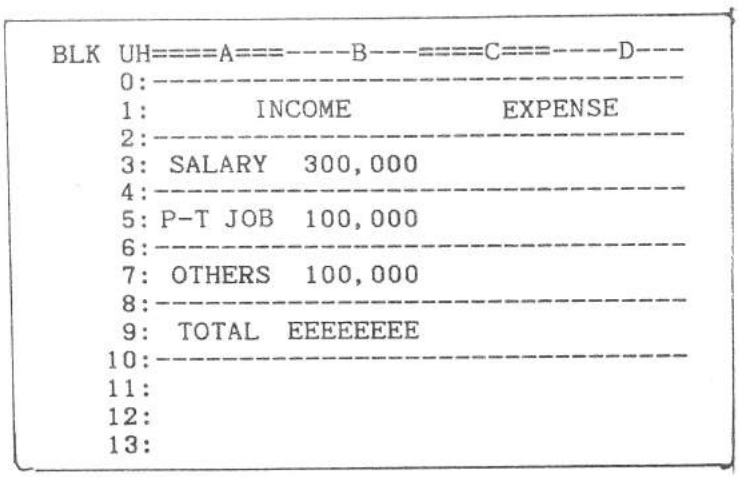
⑦ EXPR

This is to enter the total into the field in which the data is deleted in the previous 6 by using an expression.

Firstly, set the cursor to !(B,9) just as entry of normal data.

Set the semi-cursor to "EXPR". [RETURN]

FORMAT: EXPR DELETE BAR LEFT RIGHT
CENTER COMMA LENGTH DEC. POS



Input the expression onto the drum:

"300000+100000+100000"

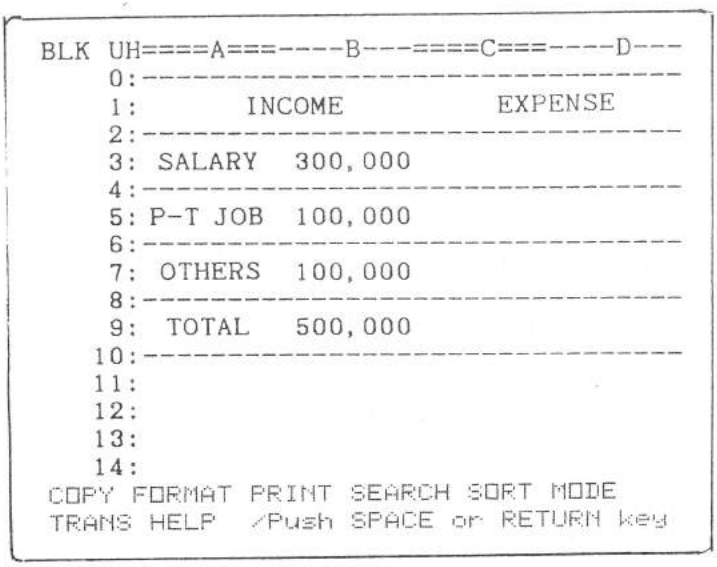
[3] [0] [0] [0] [0] [0] [+] [1] [0] [0] [0] [0] [0]

[+] [1] [0] [0] [0] [0] [0] [RETURN]

You can see the result of the calculation, "500,000", is entered in !(B,9). The drum shows the exact expression you entered and the symbol "EXP" on the upper left of the frame shows that, too.

Note: Functions of HuCAL can be used as expressions. Delete !(B,9) again and enter the following line which includes the function of total. The same result may come out.

SUM (B,3, B,7)



When you finish entering these, I bet you can do the rest of the table, can't you.

(2) MODE

Select "MODE" and press [RETURN]. Then, the following seven messages will be displayed.

MODE: AUTOCAL AUTOADD PROTECT FRAMSW
COLOR FIELD CLEAR

*AUTOCAL is the switch for auto-calculation mode. When expressions are in the fields, re-calculation will be automatically done at re-entry of new data.

*AUTOADD is the switch for auto-addition mode. When a new number is entered into the field which have another number, the sum of the former and the latter numbers will be displayed in the field.

*PROTECT protects data. When turned "ON", the data entered after turning "ON" cannot be modified.

*FRAMSW enables the cursor move to the frame from any field. However, the table cannot be scrolled leftward or upward when turned "ON".

*COLOR enables selection of color for background of the screen (either blue or black).

FIELD can protect the field where the cursor is. The symbole "" will appear on the upper left corner of the frame when turned "ON".

*CLEAR deletes all the data of the whole table.

Before making a new table, an achievement table, let's clear out the housekeeping book.

When "CLEAR" is selected, the following message will appear.

Are you sure? press [Y]

Press [Y] and delete the table.

After the screen is cleared, begin making the following table. (Leave the sections for total blank for future use.)

```
BLK UH==A===B--==C===D--==E==F-==G==
    0:
    1:NAME ENGL MATH FREN SCI SOC TOTAL
    2:--------------------------------
    3:BLACK  70   75   78   80  90   393
    4:WHITE  82   90   86   88  90   436
    5:BROWN  88   96   90   84  92   450
    6:GREEN  80   82   84   80  82   408
    7:BLUE   82   80   82   80  80   404
    8:--------------------------------
    9:TOTAL 402  423  420 492 434  2091
   10:
   11:
   12:
   13:
   14:
COPY FORMAT PRINT SEARCH SORT MODE
TRANS HELP  /Push SPACE or RETURN key
```

Well, then, let's put the total into each blank field. You can utilize a function "SUM" for this purpose.

*To sum up = Enter as follows while moving the cursor to (B,9), (C,9), (D,9), (E,9), (F,9) and (G,9) in this order.

(Press SPACE key to set to FORMAT and then press RETURN. Then, move the semi-cursor to EXPR and press RETURN again. Theis procedure should be done before each of the following expression is input.)

(B,9) = [FUNC] + [7] (B,3,B,7) [RETURN]

(C,9) = [FUNC] + [7] (C,3,C,7) [RETURN]

(D,9) = [FUNC] + [7] (D,3,D,7) [RETURN]

(E,9) = [FUNC] + [7] (E,3,E,7) [RETURN]

(F,9) = [FUNC] + [7] (F,3,F,7) [RETURN]

*Move the cursor to (G,3), (G,4), (G,5), (G,6) and (G,7) in this order.

(Press SPACE key to set to FORMAT and then press RETURN. Then, move the semi-cursor to EXPR and press RETURN again. This procedure should be done before each of the following expression is input.)

(G,3) = [FUNC] + [7] (B,3,F,3) [RETURN]

(G,4) = [FUNC] + [7] (B,4,F,4) [RETURN]

(G,5) = [FUNC] + [7] (B,5,F,5) [RETURN]

(G,6) = [FUNC] + [7] (B,6,F,6) [RETURN]

(G,7) = [FUNC] + [7] (B,7,F,7) [RETURN]

*Set the sursor to !(G,9).

(Press SPACE key to set to FORMAT and then press RETURN. Then, move the semi-cursor to EXPR and press RETURN again.)

FUNC + [7] (G,3,G,7) [RETURN]

Here, you made the same table as you see on the previous page.

① AUTOCAL

By the way, suppose that the points of ENGL for Mr. BLACK should be changed to 80 due to a mistake in counting. If you correct it just in an ordinary way, you have to take trouble calculating the fields (B,9) (the total of "ENGL") and (G,3) (the total of Mr. BLACK's points) again.

In this case, you can use a very convenient function, the Auto-calculation mode, incorporated into HuCAL. Let's try.

[SPACE] (Set to MODE.) [RETURN]

[ ] (Set to AUTOCAL.) [RETURN] "AUTO CALCULATE:ON/OFF" (Set to ON.) [RETURN]

*Changes the points of ENGL for Mr. BLACK to "80":

Set the cursor to !(B,3) and [CTRL] + [E] [8] [0] [RETURN]

The numbers in !(B,9) and !(G,3) were changed to the correct ones in a moment. Did you notice that?

② AUTOADD

The other useful function is the Auto-addition mode. If a number is newly entered into a field in which a certain number has already been entered, the number in the field will become that of the sum of the two numbers when this mode is effective.

For example, press the keys as follows after turning "AUTOCAL" off.

[ ] (Set to AUTOADD.) [RETURN] "AUTO ADDITION: ON/OFF"

(Set to ON.) [RETURN]

*Adds "5" to SCI for Mr. Blue:

Set the cursor to !(E,7) and [5] [RETURN]

Can you see that became "85"?

Note: Of course, you can use the above two functions in combination. That is,

a. Turn ON both AUTOCAL and AUTOADD.
b. Add "5" to SCI for Mr. Blue:
   Set the cursor to !(E,7) and [5]
   → The number will turn to "85".
c. [CTRL] + [Z] (Re-calculation ON)
   → Totals of "SCI" and "BLUE" will be corrected.

(3) SORT

This function is to re-order the data. When [RETURN] is pressed, the messages which ask you the last location of X (a column number or letter) and that of Y (a row number) will be on display. So, enter the appropriate answers. Then, the sorting will be started after entry of the column number (or letter) for sorting key and the sorting type (either Ascending or Descending).

For instance, let's sort the achievement table in the descending order of the total points.

*Sorts the data in the area whose diagonal is !(A,3) to !(G,7) in the descending order of the total points.

[SPACE] (Set to SORT.)

[RETURN] Input Colmn name of end corner (A-IU)

(Enter the last column number (or letter) to be sorted.)

[G] [RETURN] Input Row name of end corner (1-10000)

(Enter the last row number to be sorted.)

[7]	[RETURN]	Input Sorting-key number (1-3) (Specify the number of keys within the area.)
[1]	[RETURN]	Input Colmn name of No.1 key (A-IU) (Enter "G" for the total points.)
[G]	[RETURN]	Type (A) for Ascending or (B) for Descending (Enter "A" for ascending order and "B" for descending order.)
[B]	[RETURN]	Sorting!

Note: In the above example, when the same total points exist in the area, HuCAL will maintain the order in the original table. Suppose you decide that the points of ENGL are important and desire to put this principle into effective in the case as above. Then, enter "2" (the points of ENGL and TOTAL) for "sorting-key number". In this case, the similar message on the second key will appear after entry of the sorting type for the No.1 key. Enter "B" (column of ENGL) and "B" (the descending order), and then the result will be as desired.

(4) PRINT

When sorting is over, let's print the table. Connect a printer and turn it on.

Select "PRINT" and press [RETURN]. Then, the following seven message will be displayed.

```
PRINT: HCOPY CODE MODE SET ROW COLMN
       GO
```

The meaning of each item is as follows:

*HCOPY is an instruction to print out a hard copy (the screen copy).

[SHIFT] + [4] functions the same.

*CODE defines control codes for outputting to a printer. 40 letters can be maximally set as control codes. In execution of "PRINT GO", output is done to the printer in the set order. The codes should be decimal numbers within the range of 1 - 255. When entering the next code, press SPACE key. When making correction, just move the cursor to the field to be corrected and enter the correct code.

[CTRL] + [E] will clear a code where the cursor stays. By pressing [DEL], the code where the cursor stays will be deleted and the series of codes on the right of the cursor will be shifted leftward. By pressing [INS], on the other hand, the series of codes on the right of the cursor will be shifted rightward, thus

making room for a new code at the position where the cursor stays.

[CLR] will clear all the codes.

*MODE is to specify either pressed lines (PRESS), enlarged characters (LARGE) or small sized characters (SMALL).

•SP-400 works for normal size characters only.

*SET shows the current conditions for SET.

LINE: Determines the number of characters in a line.

PAGE: Determines the number of lines in a page.

LEFT MARGIN: Determines the number of blank characters at the left of the form paper.

UP MARGIN: Determines the number of blank lines at the top of the form paper.

*ROW asks you the number of lines in the table to be output to a printer.

*COLMN determines the order of fields on the screen when printing out a table.

*GO executes printing.

An example of print-out is shown on the next page.

NAME	ENGL	MATH	FREN	SCI	SOC	TOTAL
BLACK	70	75	78	80	90	393
WHITE	82	90	86	88	90	436
BROWN	88	96	90	84	92	450
GREEN	80	82	84	80	82	408
BLUE	82	80	82	80	80	404
TOTAL	402	423	420	492	434	2091

(5) TRANS

No matter how long it took to make a table, it will be erased when you turn off the computer. This can be used to save a table or to call it back to the computer by using a disk drive.

Select "TRANS" and press [RETURN]. Then, the following two messages will be displayed.

TRANS: SAVE LOAD UTILITY WINDOW

*SAVE saves the data in a disk.

*LOAD loads the data from a disk.

Prepare for recording and save the achievement table.

*UTILITY -- See Chapter 6 "Utilities."

*WINDOW --- See Chapter 7 "Window Function."

(6) COPY

This function is useful when you want to enter the same data into more than one fields.

*First of all, determine the location of the data to be copied, the starting field and copying times.

Enter "ABC" to !(A,0).

[SPACE] (Set to COPY.) [RETURN]

COPY:RIGHT DOWN MOVE BLOCK

① RIGHT (copies to the right of the specified field)

[ ] (Set to RIGHT.) [RETURN] COPY RIGHT TIMES: FROM:
TO:

*TIMES: copying times (determines how many times to be copied) --- ("5" for a time) [5]

FROM : starting point (enters the location of the data to be copied)

--- [A] [RETURN] [0] [RETURN]

TO : end point (enters the field to start copying)

--- [B] [RETURN] [0] [RETURN]

```
BLK UH----B----====C====----D----====E
    0:ABC       ABC       ABC       ABC
    1:
    2:
    3:
    4:
    5:
```

② DOWN (copies below the specified field)

[ ] (Set to DOWN.) [RETURN] COPY DOWN TIMES: FROM:
TO:

*Copy "ABC" entered in !(A,0) ten times downward from !(B,0).

Enter as TIMES: 10 FROM: A,0 TO: B,0 [RETURN].

```
BLK UH====A========B---====C====----D
    0:ABC       ABC
    1:          ABC
    2:          ABC
    3:          ABC
    4:          ABC
    5:          ABC
    6:          ABC
    7:          ABC
    8:          ABC
    9:          ABC
   10:
            (10 times)
```

③ MOVE (copies in a specific field)

[ ] (Set to MOVE.) [RETURN]

COPY MOVE FROM: TO:

*Copy "ABC" entered in !(A,0) to !(C,3).

Enter as FROM: A,0 TO: C,3 [RETURN].

```
BLK UH====A========B--------C====----D
    0:ABC
    1:
    2:
    3:                    ABC
    4:
    5:
    6:
    7:
```

④ BLOCK (copies as a block)

[ ] (Set to BLOCK.) [RETURN]

```
COPY BLOCK START CORNER:
     END CORNER:     TO:
```

*Copy "ABC" and "DEF" entered in the area from !(B,0) to !(C,5) to the area from !(E,0) to !(F,5).

*START CORNER: Enter the location of the upper left corner of the data to be copied

--- [B] [RETURN]

[0] [RETURN]

END CORNER: Enter the location of the lower right corner of the data to be copied

--- [C] [RETURN]

[5] [RETURN]

TO: Enter the location of the upper left corner of the block to copy the data

--- [E] [RETURN]

[0] [RETURN]

```
BLK UH==A=--B-==C=--D-==E=--F-==G=--H-
    0:      ABCDEF       ABCDEF
    1:      ABCDEF       ABCDEF
    2:      ABCDEF       ABCDEF
    3:      ABCDEF       ABCDEF
    4:      ABCDEF       ABCDEF
    5:      ABCDEF       ABCDEF
    6:
```

(7) SEARCH

(Same as [CTRL] + [V]. See page 121.)

Even if you make a table, it may be inconvenient unless you can look up a data you want to have immediately. In that case, this SEARCH function is useful.

By using this function, a specified data is searched and displayed among all the data in the table except for BLK (blank) fields. The effective range is in the entire table.

Try inputting the following data.

```
ASC UH======A=======B======C======
    0:SMITH, MARY    21NEW YORK
    1:LEE, MARK      19CHICAGO
    2:HARPER, BOB    22DALLAS
    3:SIMON, RON     20MIAMI
    4:ADAMS, GARY    18DENVER
    5:
    6:
```

Suppose you want to know how old Mr. SIMON is and where he lives, choose this function and you will see the following message in the drum.

Searching condition ▨

Input "SIMON" with double-quotation marks and press [RETURN] key. Then, the display will turn as follows.

```
SIMON, RON


ASC UH------A-------B------C------
    3:SIMON, RON     20MIAMI
    4:ADAMS, GARY    18DENVER
    5:
```

In the example above, the name (character field) was the condition. When searching, the following two types can be specified.

a) Numerical --- NUM or EXP field

b) character --- ASC, EXP or MAC field

Note: As for the EXP field, the result of the expression (shown in the display) will be searched when the numerical field is specified and the expression proper (shown in the drum) will be searched when the character field is specified.

The format of input is as follows:

$$\left\{\begin{array}{l}\underline{\text{Numerical}}\ [\underline{\text{Comparative}}]\\ \text{"}\underline{\text{Character}}\text{"}\end{array}\right\}\left[\left\{\begin{array}{l},\\ :\end{array}\right\}\left\{\begin{array}{l}\underline{\text{Numerical}}\ [\underline{\text{Comparative}}]\\ \text{"}\underline{\text{Character}}\text{"}\end{array}\right\}\left\{\ldots\right\}\right]$$

•Choose either one of the two items for those enclosed by { }.

•You can omit the item(s) enclosed by [ ].

•"..." stands for more places to input data.

•Be sure to enclose the character with double quotation marks ["].

In case of the numerical field, comparatives can be specified optionally. The following four ways are available.

① 123 --- 123 only

② 123> --- 123 or less

③ 123< --- 123 or more

④ 123# --- not equal to 123

On the other hand, there is no option for the character field. A data will be searched if it contains the specified character(s).

Use [,] and [;] as explained below.

1 [,] (Comma): AND

All the specified data should be matched for searching.

2 [;] (Semicolon): OR

Any one of the specified data should be matched for searching.

Ex.: (a) 10>;"Television"; "Radio" [RETURN]

Any fields containing numbers of 10 or less, character strings either "Television" or "Radio" will be searched.

(b) 10>;50< [RETURN]

Any fields containing numbers of 10 or less, or 50 or more will be searched.

*When a searching condition includes both AND or OR, OR has a priority. Therefore, in the following condition, "ABC" will not be searched.

10>;50<,"ABC"

*The order of input decides the entire priority in searching.

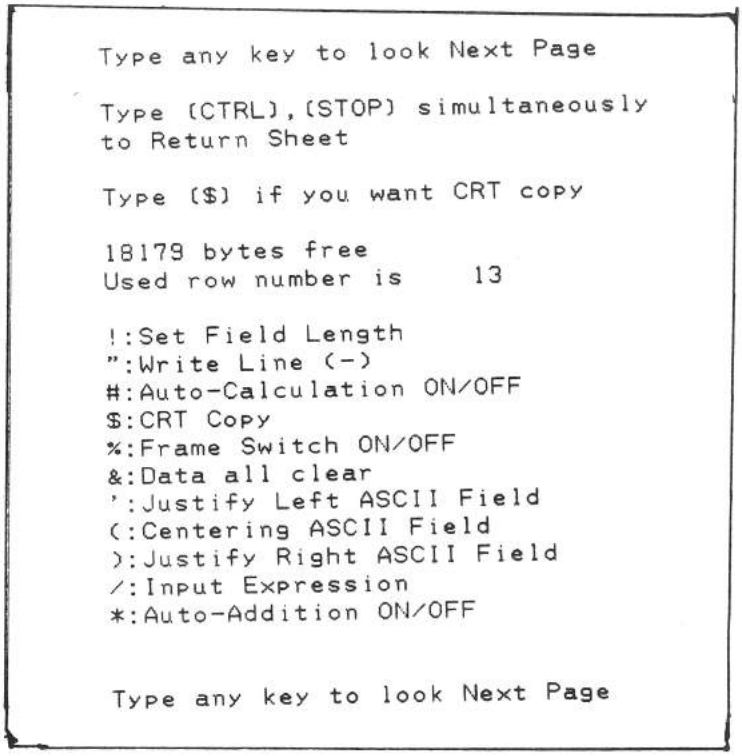
Be sure to press [RETURN] key after inputting. Then, searching will begin. And if a desired field is found, the field will be on display as a home position and a beeping sound will be made. Press any key to continue and [BREAK] key to stop. If [$] key is pressed during stopping temporarily, the screen at that time can be output to a printer.

When the entire table has been searched, the screen will return to the one which was displayed before entiring this function. Moreover, to stop searching halfway, press [BREAK] key.

(8) HELP

HuCAL has several other functions but you have to use various keys or key combinations to make them operate.

Select "HELP" and press [RETURN]. Then, the screens which describes special use of keys of will be displayed.



(A part of the screens of "HELP".)

(9) Absolute Location specification and Relative Location Specification

By the way, a field in which an expression is input may sometimes have a totally unexpected value when it is sorted.

For instance, take a look at a simple case as shown below.

First, input the following table (You may regard it as an extremely simplified version of the achievement table.)

```
BLK UH====A=======B=======C=======D===
    0:Cathy          10        8        8
    1:Ann             7        9        8
    2:Doris           9        6        7
    3:Beth            8        8        9
    4:Ema             9        8        6
    5:
    6:
    7:
    8:
    9:
   10:
   11:
   12:
   13:
   14:
COPY FORMAT PRINT SEARCH SORT MODE
TRANS HELP  /Push SPACE or RETURN key
```

Input each total point into column E. Do you remember that you can use the function SUM as in (2).

```
(E,0) --- SUM(B,0,D,0)
(E,1) --- SUM(B,1,D,1)
(E,2) --- SUM(B,2,D,2)
(E,3) --- SUM(B,3,D,3)
(E,4) --- SUM(B,4,D,4)
```

Now you got the totals of each person.

Then, let's sort the names in alphabetical order. Input appropriate answers to the prompts on display just like you did in (3).
(The last column is "E" while the last row is "4". And the sorting key is one, the column A and sorting type is "A" <Ascending>.)

BLK UH	A	B	C	D	E
0:Ann		7	9	8	24
1:Beth		8	8	9	25
2:Cathy		10	8	8	26
3:Doris		9	6	7	22
4:Ema		9	8	6	23
5:					

Did you get the above table?

Here, for example, suppose there is an input error and Ann's "7" should be changed to "10". In this case, you can use AUTOADD. Set AUTOADD to ON (refer to (3)), put the cursor on! (B,0) and then input "3" ... here you go! Press [CTRL] + [Z] to re-calculate the totals.

Oops! All the totals except for Ema's are changed. Can you guess why?

Put the cursor on !(E,0) and look at what is input in the drum ... you'll find "SUM (B,1,D,1)". This means the total for Beth's points instead of Ann's. If you look at the other fields of total you'll see that each of them shows the total of others except for Ema's total which did not move on sorting.

This unexpected situation occurred because of the fact that the sorting function moves up/down a row as a unit and that the row and the column in the function SUM are specified by constants which indicate specific range in the table (this is called "Absolute Location Specification").

In this method of specification, calculation of data can be done wherever the expression is in the table, therefore, it is convienient. However,

if the location of data change due to sorting, etc., inconvenience as seen in the above may occur.

So, we introduce you a suitable device to this kind of situation, obtaining correct totals after sorting, "Relative Location Specification" of expressions.

When you examine the contents of SUM, each has the same form, namely, "(B, 'row number', D, 'row number')". Substitute the 'row number' for '?' (a question mark) which is used as a variable and specify the meaning of it as "the column or the row number of the field in which the expression with '?' is input".

Then,

SUM (B,?,D,?)

can be read as "calculate the total of the columns B through D on the same row number as the one of the field in which the expression with '?'"

Well, let's see what will happen by inputting this expression in the fields of the column E. Put the cursor on !(E,0) and input it... Just a moment.

Here's another good point on the relative location specification. Since the "?" mark stands for a relative location, you can use COPY function

to formulate the expression for total. After inputting the expression with "?" in !(E,0), copy it four times downward with DOWN function and re-calculate. Now you got the correct totals, didn't you?

Note: The specification in the form of expression can be done by "?". That is, you can input "?-3" instead of "B" ("B" is three columns left of "E".) and "?-1" instead of "C" just as follows:

SUM (?-3,?,?-1,?)

This will bring the same result as that explained and input before.

5. Macro Instructions

HuCAL has the unique feature of macro instructions, which describe the data processing procedures.

Because of these instructions, you don't have to input expressions to each field when calculation in the vertical or horizontal direction is made, thus work efficiency is greatly improved.

Macro instructions themselves are quite similar to BASIC and there are only twelve of them. So, you'll find it easier to remember them.

Before explaining the instructions, you have to know something about the label. Though it is not an instruction, it plays an important role when it comes to execution of the macro instruction.

All commands of the macro instruction can be input lowercase letters.

(1) Labels

In BASIC, branching in a program is based on a line number.

The macro instructions for HuCAL have a branching instruction, too. But a macro sentence (a group of macro instructions) does not have a line number as in BASIC. So, a branching destination should be determined, or the program will not know where to branch.

Therefore, the label can be written in a HuCAL's macro sentence. In branching by macro instructions,

this label is used as a destination.

There are two kinds of labels: one is used to designate a whole macro sentence and the other is used inside a macro instruction. The former is used for branching from a certain macro instruction to another and the latter is for branching within a macro sentence.

@	Label name	Macro sentence

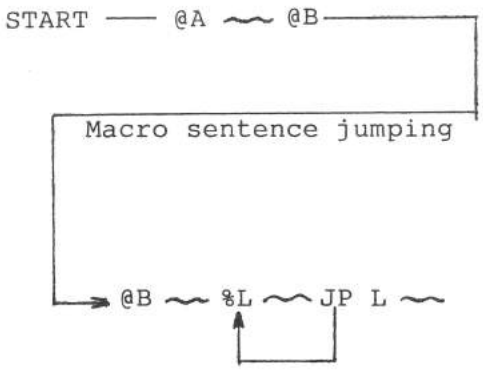
The label for branching within the sentence.

Macro sentence	%	Label Name	Macro sentence

This macro sentence may not exist.

As many characters as desired can be used for a label name so long as it is within the allowable input range. However, it is meaningless to make the label too long. Two characters may be enough.

See the following example:



Jumping within a macro sentence

The arrow shows the execution flow.

Primarily, a macro sentence with label @A begins to be executed. When the flow meets @B in the course of execution, it branches to a macro sentence with label @B. During executing the macro sentence with @B, when the flow meets "JP␣L" (Jump L), it jumps to label "L" in the sentence currently being executed.

Can you get the idea of a label?

Now, return to the example above. You'll find @A at the beginning but you don't have to put a label unless the macro sentence is to be branched from others. Therefore, the label can be dispensed with.

In general, the ordinary usage of a label might be as follows:

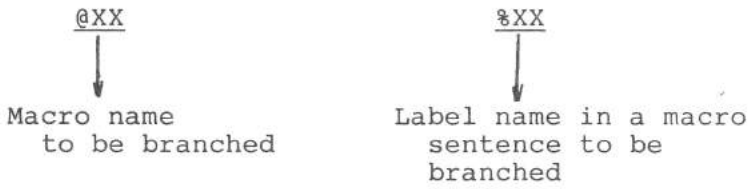
- o @XX is for sequential connection of macro sentences (where one flow is described over more than one macro sentence) or for branching to greater extent.
- o %XX or JP XX is for so-called loop control.

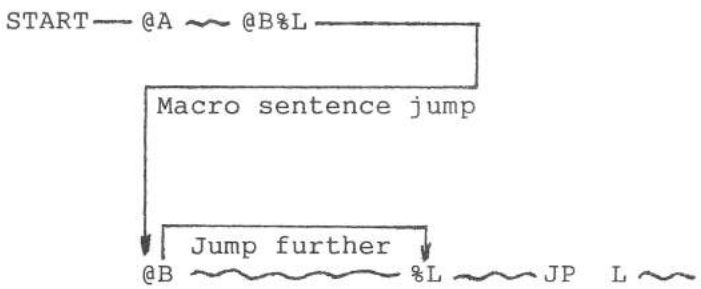
Moreover, JP (Jump) instruction is used for branching within when branching from one sentence to another.

*@XX at the beginning of the macro sentence is not regarded as a branching mark.

(2) Special Types of Macro Jumping

@XX is used for branching to another macro sentence. But when written as follows, the flow can get into any part of a macro sentence to be branched.





(3) Macro Insturction Sentences

The structure of macro instruction sentences is as follows:

Instruction: Instruction: Instruction .....

Several instructions can be put sequentially by placing a colon(:) between each of them. The maximum number of characters in a line is 233.

If a space key is pressed during execution of a macro instruction, its execution will be stopped temporarily. To print out a screen copy at that time, press [$] key. Press a space key again to resume execution.

When a complete stop is desired, press [CTRL] + [STOP].

(4) Macro Instruction Errors

When a macro instruction is to be executed, grammar is checked once inside the system.

If an error is detected at that time, execution of the instruction will be held up, the macro sentence in which the error is found will be displayed on the drum, and the cursor will move to and flash at the position where an error is detected.

After confirming the reason for the error, press any key to return to the sheet. Correct the macro sentence with the error and resume execution.

*In some cases, the cursor will be displayed a few characters away from the location of the error.

(5) Macro Instruction Grammar

The grammar of macro instruction is explained below. There are 12 instructions and they are roughly divided into the following five categories:

(1) Judgement of condition
(2) Branching
(3) Input and output
(4) Trasferring and substitution of data
(5) Loop

(1) Judgement of condition

There are the following four instructions of this type:

a) IF (if)
b) THEN (then)
c) ELSE (else)
d) STOP (stop)

These four instructions use the following grammar:

IF Condition THEN Process 1
[ELSE Process 2] STOP

IF at the beginning indicates the judgement of condition. The actual condition is described in the next Condition.

The next [THEN] means that Process 1 will be executed when the Condition is satisfied (or is true).

When the Condition is not satisfied (or is false), if followed by [ELSE], Process 2 will be executed and if not, the next macro instruction will be executed without doing anything.

* [ELSE] Process 2 can be abbreviated.

[STOP] at the end indicates the end of judgement of condition.

In Process 1 and Process 2, any macro instructions except for [IF], [THEN], [ELSE] and [STOP] can be described.

```
Ex.: IF␣ZA=1␣THEN␣HDCP␣!(A, 0)
     ELSE␣@B␣STOP
     IF␣$="#"THEN␣@END␣STOP
```

In Condition, AND or OR can be specified. Specify AND with [,] and OR with [;].

```
Ex.: IF␣ZA=0;ZA=1␣THEN..... ⟶ OR specifi-
     cation
     IF␣$="#",ZA=0␣THEN.....⟶ AND specifi-
     cation
```

(␣ means a space.)

In the conditional formula, =, >, <, =>, <=, =<, <>, >< are used. A character variables can be compared by these operators, too, and comparison is done by the size of ASCII code specified to each character. For example A<B<C<E....<Z is the order. The result of an application of the equal sign "=" to the character variable is true if the leftmost letter in the right side of the equation begins with the same leftmost letter of the left side of it. For example "ABC" = "AB" is true. Also "=?" is applicable to a character variable. If the left side character strings include the right side character strings, the result of the formula is true.

Example: If $ = "ABCDEF"

```
$ =? "ABC"      TRUE
$ =? "CDEF"     TRUE
$ =? "ABD"      FALSE
$ =  "ABC"      TRUE
$ =  "CDEF"     FALSE
$ <> "ABC"      TRUE
$ >= "ABC"      TRUE
```

Instructions IF, THEN, ELSE, STOP can be also used.

Example:

```
IF ZA = 1 THEN HDCP! (A, 0) ELSE @B STOP
IF $ = "#" THEN @END STOP
```

(2) Branching

There is only one branching instruction:

JP (jp)

A label name must follow JP. In addition, this label name should be the one included in a macro instruction currently being executed.

EX.: %L:INPUT␣ZA:IF␣ZA<0␣

THEN␣JP␣L␣ELSE ···

EX.: %l:input␣za:if␣za<0␣

then␣jp␣l␣else ···

(3) Input and output

There are the following four input and output instruction:

a) INPUT (input)

b) OUT (out)

c) POUT (pout)

d) HDCP (hdcp)

a) INPUT Variable or Expression

System variable ZA through ZZ, a field: (X, Y) or a character-type variable $ can be used as Variable.

Execute this instruction and input to it. After that, confirmation will be requested. If you want to make a correction, press [N] of uppercase letter. Press another key to continue.

Note: Confirmation is requested only when a question mark (?) is put at the end of the variable name as INPUT ZA?. If a question mark (?) is not put there, no confirmation will be made. More than one variable can be input by dividing each with commas as INPUT ZA, ZB. If confirmation is desired in this case, put a question mark (?) immediately after the last variable.

b) OUT Variable or Character string or
Control character string

c) POUT Variable or Character string or
Control character string

These two functions are the same except for control character strings and the difference is that OUT outputs to the screen while POUT outputs to the printer.

System variables ZA through ZZ, a field!(X, Y) or a character-type variable $ can be used as Variable.

Character strings should be described by enclosing in quotation marks ("").

The control character strings for the screen are different from those for the printer. Refer to the following table:

Control character string	Screen	Printer
h	Cursor home	
c	Clear screen	Home feed (Page feed)
s	——	Small letter size
n	——	Standard letter size
e	——	Enlarged letter size
r	——	Standard line feed
p	——	Interlinear pressed line feed
/ (Slash)	Line feed	Line feed

. All the letters should be lowercase.

. Enclose with ["] except for [/].

. Those strings cannot function unless they are at the beginning of the letters enclosed with ["] after OUT (or POUT). (The letters will be output as they are if they aren't initially.)

. The output width per line can be specified by the value of system variable ZZ. Change that as you wish.

. When line-feeding is not preferable, add [,] after the data.

```
Ex.:  OUT␣ZA, ZB, "HuCAL", !(B,4), $
      OUT"c", "ABC", /// ZA
          ↑              ↑
      Clear screen   Feeds three
      for OUT        lines
```

. TAB specification is possible. Put the number of spaces in the horizontal direction before the data.

```
Ex.:  OUT␣15ZA, "TAB"
          ↑
      Specifies 15 TAB letters
```

. The format will be:

[p]out␣< Array of ASCII codes in Decimal >

Ex.: out␣< 65,66 > ⟶ AB

```
FOR␣ZA=65,68:OUT<ZA>, :NEXT:OUT
```

⟶ ABCD

By using this function, you can sent an escape sequence of the printer.

d) HDCP Field position (, Number of lines)

A chart is output as the field specified by field position as home position to the printer. Output is made as number of lines is specified. (Up to 250 lines maximum) When not specified, the same chart as displayed on the screen will be output.

The width of output is 32 columns/line.

Ex.: ① HDCP␣!(A, 0)

② HDCP␣!(A, 0), 50: HDCP␣!(I, 0), 50

HDCP␣! (B, 1)

```
BLK UH--B-==C=--D-==E=--F-==G=--H-==I=
    0:
    1:   1   2   3   4   5   6   7   8
    2:   2   4   6   8  10  12  14  16
    3:   3   6   9  12  15  18  21  24
    4:   4   8  12  16  20  24  28  32
    5:   5  10  15  20  25  30  35  40
    6:   6  12  18  24  30  36  42  48
    7:   7  14  21  28  35  42  49  56
    8:   8  16  24  32  40  48  56  64
    9:   9  18  27  36  45  54  63  72
   10:  10  20  30  40  50  60  70  80
   11:  11  22  33  44  55  66  77  88
   12:  12  24  36  48  60  72  84  96
   13:  13  26  39  52  65  78  91 104
   14:  14  28  42  56  70  84  98 112
COPY FORMAT PRINT SEARCH SORT MODE
TRANS HELP  /Push SPACE or RETURN key
```

HDCP␣!(B, 1), 9

```
   1   2   3   4   5   6   7   8
   2   4   6   8  10  12  14  16
   3   6   9  12  15  18  21  24
   4   8  12  16  20  24  28  32
   5  10  15  20  25  30  35  40
   6  12  18  24  30  36  42  48
   7  14  21  28  35  42  49  56
   8  16  24  32  40  48  56  64
   9  18  27  36  45  54  63  72
  10  20  30  40  50  60  70  80
```

④ Tranferring and substituting data

There are the following two instructions for tranferring and substituting data:

a) SWAP (swap)

b) = (this is not actually treated as an instruction.)

a) SWAP Variable 1, Variable 2

It means "swap the data in Variable 1 and that in Variable 2.

By using this command, the contents of the two variables are exchanged. Since it is exchanging the contents, Variable 2 can come before Variable 1 and vice versa. The form of the variables should be as follows:

① SWAP ZA, ZA: Both are system variables.

② SWAP ZA,!(X, Y): System variable and field variable; the latter should be either NUM, BLK or EXP.

③ SWAP $,!(X, Y): Character variable and field variable; the latter should be either ASC, BLK or MAC.

④ SWAP !(X, Y),!(X', Y'): Both are field variables; any content will do.

⑤ SWAP $, ZA: Invalid

b) Variable 1 = Variable 2 or Constant or Expression

This is an instruction for substitution.

This type of the variables or constants should match.

Ex.: ZA=ZB

```
ZC=1-4 ... in this expression; ZC=-3
!(A, 0)=2
ZD=ZC+!(A, 0) ... in this expression; ZD=-1
!(B, 0)="ABC"
$="XYZ"
```

Note: Character strings should be enclosed with ["].

```
$="ABC"+!(B, 0)+$
```

The output of $ is "ABCABCXYZ".

This is the same as $="ABCABCXYZ".

Let's take a look at the regulation for addition of character strings. The following two kinds are available:

1) $=Expression of addition of character string

2) !(X, Y)=Expression of addition of character string

In case of 1), an error occurs when any numeral appears within a character string.

In case of 2), if a numeral appears within a character string, the string will be neglected and numeric operation will be done. A character string which is not enclosed by ["] will be treated as a

numeral to which the first two characters are converted. Therefore, A → 0, B → 1, AA → 26, BA → 52, etc. will be the result. As to the attribute of the field in this case, it will be NUM for the field in which a numeral appears and numeric operation is done, while it will be ASC, after substitution, for the field in which the first item in the addition of character string is either a character string enclosed by ["] or $ or ASCII field. However, if the first item is MAC field, that will be MAC after substitution.

When adding character strings, BLK field will be neglected.

If space is desired, enclose it with ["] like "␣".

⑤ Loop

The following the instructions are for looping:

a) FOR (for)

b) NEXT (next)

These two instructions should be used in the following manner:

[FOR] System variable = Initial value,
Final value Process [NEXT]

. System variables are from ZA to ZZ.

. Initial value should be equal to or less than the final value.

. Don't put any variables after NEXT . It corresponds to the nearest FOR timewise.

. The nesting of FOR and NEXT can be up to triple.

. Step is always 1.

. If the step needs to be changed, operate the system variable within Process .

. If the flow branches to another macro instruction, the set of FOR - NEXT until that branching will be ineffective.

. When branching within the same macro instruction, use the method mentioned in (2) if the FOR - NEXT set is not completely finished. Otherwise, nesting will not be cleared.

Note: FOR - NEXT is useful when the same process is executed a certain number of times. Here, the examples show two kinds of macro instructions: one is to substitute the input letters and the other is to input the answers of B*C into field D.

(Substitution)

```
@A:FOR␣ZA=0,10:INPUT␣$:!(A,ZA)=$:NEXT
```

(calculation)

```
@A:FOR␣ZA=0,10:!(D,ZA)=!(B,ZA)*!(C,ZA):NEXT
```

```
%L:FOR␣ZA=1,100:~~JP␣L ~~NEXT →X
```

```
@F:%L:FOR␣ZA=1,100: ~~ @F%L ~~NEXT →○
```

```
FOR ZA=1,9:FOR ZB=1,9:!(ZA,ZB)=ZB*ZA:NE
XT:NEXT



MAC UH=A=-B-=C=-D-=E=-F-=G=-H-=I=-J-=K
    0:FOR
     1:       1   2   3   4   5   6   7   8   9
     2:       2   4   6   8  10  12  14  16  18
     3:       3   6   9  12  15  18  21  24  27
     4:       4   8  12  16  20  24  28  32  36
     5:       5  10  15  20  25  30  35  40  45
     6:       6  12  18  24  30  36  42  48  54
     7:       7  14  21  28  35  42  49  56  63
     8:       8  16  24  32  40  48  56  64  72
     9:       9  18  27  36  45  54  63  72  81
    10:
    11:
    12:
    13:
    14:
COPY FORMAT PRINT SEARCH SORT MODE
TRANS HELP  /Push SPACE or RETURN key
```

(6) Let's Use Macro Instructions

How are the various rules on macro instructions which have been explained so far? Do they seem to be too difficult for you? No, they aren't at all. The HuCAL macro insturctions are easy to use and anyone can learn them without knowledge of BASIC.

Once you have learned them, you may feel relieved to know that they are rather easy. But it is apparent that you will feel thankful to the surprising programming function of the macro instructions.

For example, calculations of sum and mathematical means of columns and rows, searching of data, sorting of data, etc. as well as displaying in a graph can be done easily and quickly by using the macro instructions.

HuCAL is useful enough with what you've learned so far, but by taking advantage of the macro instructions which enable programming even in a simplified language, we promise that its utility will be remarkably improved.

If you learn these macro instructions, most of the spread-sheet operations can be performed perfectly. And the functions are fully applicable to daily work at school, routine information processing as well as advanced business use.

To be frank, HuCAL is more practical than BASIC. Work which requires great effort in programming in BASIC, can be done pretty easily with HuCAL.

In other words, HuCAL is designed to facilitate routine information processing or that for business immediately after purchasing. Therefore, it is quite natural that it is practical.

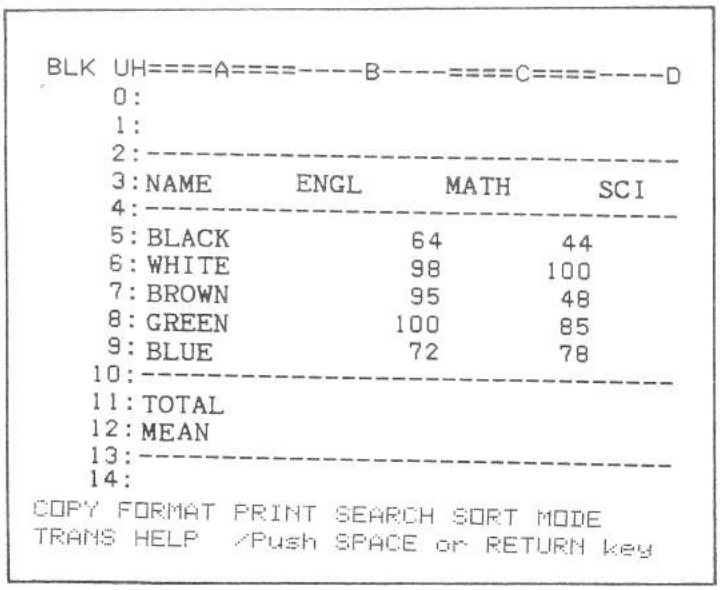
In addition, by learning the HuCAL macro instructions, you may find it somewhat easier to understand BASIC.

Well, let's use the macro instructions.

a: Calculating Sum and Mean

First of all, let's calculate the sum and mean for the table of achievements by using macro instructions.

Input data and set the screen as shown in the figure below.



The areas for inputting macro instructions are fixed at 441 fields from (A,0) to (U,20).

Note: Although the macro instructions can be input only to the area from (A,0) to (U,20), expressions can be input to any field. However, it is troublesome to input expressions one by one and they may occupy a large memory area. So, it is recommend to use macro instructions wherever possible.

Here's a simple macro instruction example. This is to input the result of Column B x Column C into Column D. The number of items is 100.

```
@A:FOR␣ZY=0,100:!(D,ZY)=!(B,ZY)*!(C,ZY):NEXT
```

Let's input a macro instruction into the field !(A,0). Move the cursor to !(A,0) and press [CTRL] and [O] simultaneously.

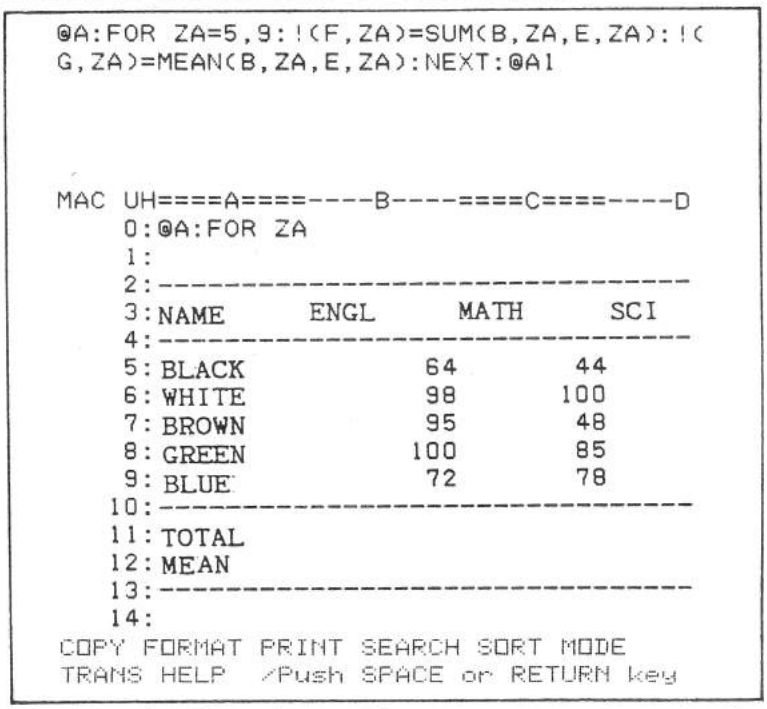
The field where the cursor was is filled with letter Ms and the cursor begins to flash on the drum. Like expressions, macro instructions should be input on the drum. The maximum input characters is also 233 characters.

Note: When inputting a macro instruction, move the cursor to the field to which the instruction will be input and start inputting by pressing [CTRL] + [O]. To end input, press [RETURN]. Press [CTRL] + [P] in the field where a macro instruction is input in order to execute the instruction.

Input the following macro instruction:

```
@A:FOR ZA=5,9:!(F,ZA)=SUM(B,ZA,E,ZA):!(
G,ZA)=MEAN(B,ZA,E,ZA):NEXT:@A1
```

In pressing [RETURN] after inputting all the instructions, the following screen will be displayed:



As seen in the above figure, you can see only a part of the instruction input previously (as much as the length of the field) in the field. This is not good for checking the input. Well, I'll tell you what to do if you want to see the whole instruction.

To see the whole macro instruction, move the cursor to

the field to which the macro instruction is input and it will be displayed on the drum.

[On labels]

Let's go back to the explanation of the previous instruction.

Suppose that "@A" is a name of the field. So, @A (@ followed by a label name) declares the label name of the field !(A,0) as A. Be sure to put that whenever a macro instruction is input.

If a name is given to the field with macro instructions, it may be convenient because the macro instruction in the other field can be designated as @ + label name when transferring execution to the macro instruction in another field. (Label name here is the one attached to the instruction to be transferred.) In other words, it functions in almost the same way as LABEL label name or GOTO label name (line number) in BASIC.

As mentioned later, please note that this @ label name is for within a field unit and is different from % label name.

Here is how you name it: put @ first and an alphabetic character for the first letter to be followed by a numeral for the second, such as @A, @AB, @BI, @C65, and so forth.

A colon (:) is to connect instruction sentences. You can freely make up a program of 233 letters at the most connecting instruction sentences with a colon.

[Column Numbers and Numeric Variables]

"FOR␣ZA=5,9.....NEXT" works much the same way as a space inbetween.

ZA is a numeric variable. Since HuCAL uses A-IU as column numbers (constant), twenty-six of them from ZA-ZZ are handled as numeric variables.

> Note: Variables must be explained before numeric variables. A variable can be assumed as a box with a name of no specific size.
>
> Namely, when data 123 has been input, simple inputting will only lead to a mix-up of where it is input or how to call it back. So, how about naming the data A, so all you have to do will be just to specify name A to call back data 123.
>
> Since HuCAL uses the alphabetic characters A-B horizontally, only twenty-six of them (ZA-ZZ) can be used for this scheme, however, they are only available for numerical values, so the $ sign must be used for character data.
>
> Please also note that even numerical data cannot be calculated since it is handled as characters when input with $.

Here, ZA is regarded as a variable to specify the column number.

Note: When numerical values are substituted into ZZ, the number of horizontal digits to print out at printer-output will be specified. In other words, it is to determine whether to print out only characters first then feed a line, or to print out up to 100 characters first and then feed line. This, however, requires careful attention since a new line will be started halfway when the indication exceeds the capacity of a printer (only 80 characters can be printed out).

Namely, ZA=5,9 shows that it is to cover from Mr. BLACK in Row 5 to Mr. BLUE in Row 9.

Note: Variable ZW substitutes the number of the character showing on which character of the macro instruction the error has occurred in case of a grammatical error occurs with the macro instruction being executed. When you attempt to execute a macro instruction with a grammatical error, the cursor will flash at the place where the error is, indicating the macro instruction, and it will not be executed. Since the cursor will return to the field when any key is pressed, press [CTRL] + [O] to correct it on the drum and then press the [RETURN] key. However, it is suggested that this number of characters be regarded as an approximate standard.

Variable ZZ is used here as a system variable to specify the horizontal width. A numerical value

substituted into ZZ is the number of digits in one line.

It will all be output in 80 digits unless the numerical values from 1 - 223 are substituted into ZZ.

The form of the sentence FOR-NEXT will be such as FOR variable name = initial value, final value:instruction sentence:NEXT.... Variable names will be such variables from ZA-ZZ, and initial and final values can be either numerical constants, expressions or variables.

"!(F,ZA)=SUM(B, ZA, E, ZA)" is an instruction to calculate the sum of from column B to column E of row ZA and to put the result into column F. An equal sign (=) used here means to substitute what is on the right side to the left side.

Therefore, you will know offhand that the individual sums are being calculated as the row shifts from one to another with the column remaining the same, won't you? First is ZA=5, so calculate Mr. BLACK's sum and substitute it to !(F, 5). Next is ZA=6, so Mr. WHITE's sum into !(F, 6).... This is how this program works and calculates automatically up to ZA=9.

"!(G, ZA)=MEAN(B, ZA, E, ZA)" also calculates similarly the individual averages and puts them into column G.

"NEXT:@A1" instructs to calculate the sums and the averages from Mr. BLACK in row 5 to Mr. BLUE in row 9 and transfer the execution to the macro instruction of @A1 escaping from the loop of FOR-NEXT (repetition). Did you understand it?

[Acquiring the sums and Averages both Horizontally and Vertically]

Next, let's try and input to !(B, 0) the following macro instruction:

```
@A1:FOR ZA=B,F:!(ZA,11)=SUM(ZA,5,ZA,9):
!(ZA,12)=MEAN(ZA,5,ZA,9):NEXT:@END
```

Move the cursor to !(B,0) and press [O] key while pressing [CTRL] key. Input the above macro instruction onto the drum and press [RETURN] key. Has something like the following figure come out by now?

```
@A1:FOR ZA=B,F:!(ZA,11)=SUM(ZA,5,ZA,9):
!(ZA,12)=MEAN(ZA,5,ZA,9):NEXT:@END


MAC UH====A=========B---====C====----D
    0:@A:FOR ZA@A1:FOR Z
    1:
    2:-----------------------------
    3:NAME      ENGL      MATH      SCI
    4:-----------------------------
    5:BLACK       64        44
    6:WHITE       98       100
    7:BROWN       95        48
    8:GREEN      100        85
    9:BLUE        72        78
   10:-----------------------------
   11:TOTAL      429       355
   12:MEAN        86        71
   13:-----------------------------
   14:
COPY FORMAT PRINT SEARCH SORT MODE
TRANS HELP  /Push SPACE or RETURN key
```

"@A1" is a name given to this field. And ZA in "FOR␣ZA=B, F.....NEXT" is used as a variable to specify the row number.

> Note: A name like @A1 is needed for macro instructions. It is alright to regard this like a row number for BASIC. However, the only difference is that each macro instruction is executed independently here.

Namely, the row numbers are all set from B to F.

"!(ZA, 11)=SUM (ZA, 5, ZA, 9)" is a program that calculates the sum of from row 5 to row 9 in column A and substitute it to row 11. This also calculates the sum for each subject as the column shifts with the row remaining the same.

"!(ZA, 12)=MEAN(ZA, 5, ZA, 9)" calculates the average for each subject and to put it into row 12.

"NEXT:@END" tells you to finish execution of the instruction after calculating the sums and the averages of from column B to F. Although END is used here as a sign to end a macro instruction, it will finish its macro instruction without @END and return to the chart if you transfer the execution to an unregistered @ label name.

Now, we can have the sums and the averages both horizontally and vertically at the same time, can't we? Move the cursor to !(A, 0).

When carrying out a macro instruction, press the [P] key while pressing [CTRL]. Then the sums and the averages will be shown as in the following figure:

```
@A:FOR ZA=5,9:!(F,ZA)=SUM(B,ZA,E,ZA):!(
G,ZA)=MEAN(B,ZA,E,ZA):NEXT:@A1



MAC UH====A====----B----====C====----D
    0:@A:FOR ZA@A1:FOR Z
    1:
    2:---------------------------------
    3:NAME      ENGL      MATH      SCI
    4:---------------------------------
    5:BLACK       64        44
    6:WHITE       98       100
    7:BROWN       95        48
    8:GREEN      100        85
    9:BLUE        72        78
   10:---------------------------------
   11:TOTAL      429       355
   12:MEAN        86        71
   13:---------------------------------
   14:
COPY FORMAT PRINT SEARCH SORT MODE
TRANS HELP  /Push SPACE or RETURN key
```

[Taking screen-copies of the chart only]

Now, let's output the chart in the figure above to the printer. Previously, we learned to take screen-copies by pressing the [$] key. However, it will also output the frames of the screen in this case. So, what we do now to avoid this is to take advantage of a macro instruction.

Why don't we try it actually? Move the cursor to !(C, 0), press the [O] key while pressing [CTRL], and input the following macro instruction:

```
@P:HDCP !(A,1),13
```

After inputting, be sure to press [RETURN].

We have named the label name of the field as "@P" since it is going to be printed out.

We have two styles of HDCP sentence, such as:

HDCP␣!(X,Y)

HDCP␣!(X,Y), the number of output rows from Row Y

The former outputs to the printer a screen-copy of the field with !(X,Y) at the top left, and the latter outputs to the printer a range as wide as the number of rows specified from the field of !(X,Y) as the starting point. The maximum horizontal width that can be specified is 31 digits and the maximum for the vertical is 250 rows.

Here, we output covering 13 rows vertically from !(A,1) and 31 digits horizontally.

It is not possible, however, to stop outputting in the middle with the [BREAK] key.

> Note: Since the horizontal width is limited to 31 digits when outputting with the HDCP sentence, it is recommended to output with the POUT sentence when 31 digits are not enough.

By the way, is the power switch on for the printer? Press the [P] key while pressing the [CTRL] key, and the chart will be printed out to the printer.

NAME	ENGL	MATH	SCI
BLACK	64	44	
WHITE	98	100	
BROWN	95	48	
GREEN	100	85	
BLUE	72	78	
TOTAL	429	355	
MEAN	86	71	

b: Word Book

Make your plan before starting:

Guidelines

1. Write the words into Column A and the meanings into Column B.
2. To enhance the efficiency of memorizing words:
   a. The input of the words
   b. The input of the meanings
   c. The test on memory
   d. The printer-output of the words

   Form a program so as to follow the above procedure.
3. Make possible improvements so that you can select optionally from the menu indication your favorite program formed respectively in 2 above.

   Note: It is a method of putting each independent program (a, b, c, and d) together as one menu.

Preparations

Let's try and change the size of the field to meet its purpose before forming a program.

1. Set the length of the field in Column A as wide as 25 characters.

   Note: See page 19 for how to change the length of a field.

2. Set the length of the field in Column B as wide as 31 characters.
3. Set the length of the field in Column C as wide as 3 characters.
4. Set the length of the fields of the remaining columns as wide 9 characters.
5. In Column C-Row 0, numerical values concerning the number of words registered is to be entered. Input 1 as an initial value.
6. In Column C-Row 1, the number of words when inputting the meanings is to be entered. Input 1 as an initial value.

   Note: Set the initial value as 1 and it will input from the 1st of the chart.

Framework

Form a program to input the words.

Procedures

1. Move the cursor to !(D,0) and press [CTRL] and [O].

   Input the next program when the field is filled with Ms.

   Note: See page 66 for how to input a macro instruction.

```
[D,O] @A:ZY=!(C,O):%L:OUT "c"////,ZY,6"   WORD: PLEASE INPUT NOW.. "//,"# FOR TE
RMINATION OF INPUT....":INPUT $:IF $="#" THEN @END STOP:!(A,ZY)=$:ZY=ZY+1:!(C,O)
=ZY:JP L
```

Note: [D,O] is a coordinate you get when printing out a macro list. (Input from @A1 in practice)

Macro Instruction	Meaning
[D,0] @A:	The name of a program
ZY=!(C,0):	Reads in the number of words written in !(C,0). [!] A field mark [:] Division of programs
%L:	A label to transfer the execution here with "JP␣L". [%] Used when jumping within a field
OUT"c"//////,	Clears the screen and feeds six lines. [/] Line feed Note) The "C" should be lowercase one.
ZY,6"..."//,	Indicates the number of words and shows a message to prompt the input. And it feeds two lines. [ZY] Variable
"# IF ...":... ... STOP:	If the word you have input is #, then the input will be completed.
INPUT $:	Inputs a word into a variable $. [INPUT] Input to -
ZY=ZY+1:	Starts with another row to write in a word.
!(C,0)=ZY	Increases the number of words by 1.
JP␣L	Returns to Label %L and repeats the input of a word. [JP␣L] Jumps to "%L"

Thus, the first program is completed.

Operation Program execution

1. Move the cursor to !(D, 0) and press [CTRL] and [P].
2. Input a word and press [RETURN] as it will be indicated 1 WORD: PLEASE INPUT NOW..

   Note: Be sure to input as many as 20 words here. They will be used for checking when you input the program for meanings.

3. Input another word in the same way as it will be indicated 2WORD: PLEASE INPUT NOW..

This how you input the words one by one. Input # after all the words are registered.

Framework b. Inputting Meanings of Words

Now, we form a program to input the meanings of the words.

(1) Move the cursor to !(D,1) and press both [CTRL] and [O] at one time to input the next program.

```
[D,1] @B:ZY=!(C,1):%L:IF ZY=!(C,0) THEN @END STOP:OUT "⊏"////,!(A,ZY),":PLEASE I
NPUT MEANING"//,"# FOR TERMINATION OF INPUT...";INPUT $:IF $="#" THEN @END STOP:
!(B,ZY)=$:ZY=ZY+1:!(C,1)=ZY:JP L
```

Macro Instruction	Meaning
[D,1] @B:	Represents the name of a program
ZY=!(C,1):	Reads in the number of the word whose meaning will now be input into variable ZY.
%L:	A label to transfer the execution here with JP␣L. Note: See page 48 for labels
IFZY=!(C,0)... ... STOP:	All the meanings will have been input when ZY becomes equal to the content of !(C,0). Then the program is completed.
OUT"c"//////,	Clears out the screen and feeds six lines.
!(A,ZY),"..."//,	Prompts the imput of a meaning of indicating the word and feeds two lines.
"# IF...":	Shows that # is to complete the execution of the input.
INPUT $:	Inputs a meaning into the variable $.
IF $="#"...STOP:	Shows that # is to complete the execution.
!(B,ZY)=$:	Writes in the meaning of the word just input into the specified field.
ZY=ZY+1:	Starts a new row to write in a meaning.
!(C,1)=ZY:	Advances the number of the word to write in a meaning.
JP␣L	Returns to label %L.

Operation Execution

1. Move the cursor to !(D,1) and press both [CTRL] and [P] at one time.
2. Input a meaning as the word will be displayed.
   If you want to stop its execution in the middle, just input # .

Note: As mentioned in the previous NOTE, a program doesn't work unless the words have been input when forming this program.

Application

Now let's try and make full use of HuCAL's functions.

*Rearranging the contents on the word book in alphabetical order.

We now use HuCAL's sorting function.

Note: See page 33 for sorting function.

- The procedures for Sorting -

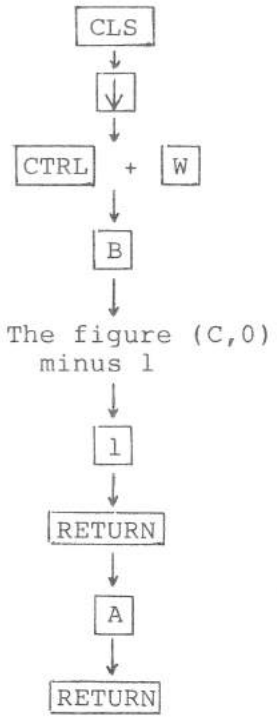
1. Press [CLS] and move the cursor to !(A,0).
2. Move the cursor to !(A,1).
3. Press [W] at the same time with [CTRL] .
4. Press [B] and then [RETURN] as the indication Input Colmn name of end corner (A-IU) will appear.
5. When the indication Input Row name of end corner (1-10000) appears,

next, input the figure equal to "[the number written on the coordinate (C,0)] minus one."

6. Input [1] and press [RETURN] when the indication Input Sorting-key number (1-3) appears.
7. When the indication Input Colmn name of No.1 key (A-IU) appears, press [A] and [RETURN] since the sorting is supposed to be handled in alphabetical order.
8. As it will be requesting you Type (A) for Ascending or (B) for Descending press [A] (meaning "of ascending order"), and the sorting will be completed. It takes no longer than just a few seconds for as many as 5 or 6 words. It's pretty handy like an ordinary word book, isn't it?

Note: It can also be sorted by using the menu-screen. adjust it to the [SPACE] key and then press [RETURN]. See page 33 for further details.

The Pressing Order for the Keys



Framework　　c. Programming a memory test for the input words

1. Move the cursor to !(D,2) and press both [CTRL] and [O] at the same time to input the next program.

Macro List

```
[D,2] @C:ZM=!(C,0)-1:ZF=!(E,0)*275+1:ZF=ZF-INT(ZF/32768)*32768:!(E,0)=ZF:ZY=INT(
ZM*ZF/32768)+1:OUT "c"////,!(A,ZY),"INPUT MEANING NOW..":INPUT $:IF !(B,ZY)=$,$=
!(B,ZY) THEN @C1 ELSE @C2 STOP
```

Macro List: Apply the following program for !(D,3).

```
[D,3] @C1:OUT "CORRECT!          IF CONTINUE PRESS [Y]    ":INPUT $:IF $="Y";$="y";
      THEN @C ELSE @END STOP
```

Macro List: Apply the following program for !(D,4).

```
[D,4] @C2:OUT "ANSWER:  ( ",!(B,ZY)," )     "//,"PRESS [Y] TO CONTINUE    ":INPUT
$:IF $="Y";$="y";       THEN @C ELSE @END STOP
```

Note: The underlined part can be used for other Programs with necessity of making random numbers.

Macro Instruction	Meaning
[D,2] @C:	Represents the name of a program.
ZM=!(C,0)-1:	Reads in the number of words into ZM. ZM is the number of words.
ZF=!(E,0)...+1:	Determines the words to test using random numbers. ZF is the number of words.
OUT"c"//////,	Clears out the screen and feeds six line.
!(A,ZY),"INPUT...":	Prompts the input of a meaning by displaying each word.
INPUT $:	Inputs the meaning of a word into $.
IF !(B,ZY)=$, ...@C2 STOP	If the word just input is the same as the preregistered one, it will be trasferred to the program named @C1. If different, it will be transferred to the one named @C2. Note: @C1 is a program to input into !(D,3), and @C2 into (D,4).
[D,3] @C1:	Represents the name of a program.
OUT"CORRECT..."	Informs that the meaning has proved correct. Also indicates that you can continue with the test by pressing [Y].
INPUT $:	Inputs [Y] or some other key.

Macro Instructions	Meaning
IF$="Y";$="y"... ... @END STOP	If the letter input into $ is either Y or y, it will return to program @C assuming you are continuing. If not, it will end. Note: No matter whether it's Y or y, pressing that particular key will be judged as Yes.
[D,4] @C2:	Represents the name of a program.
OUT"ANSWER:... //,"...":	Displays a correct answer and to press [Y] if carrying on with the test.
INPUT $:	Tells you to input either [Y] or some other keys.
IF$="Y";$="y"... ... @END STOP	Returns to program @C if with [Y], and finishes the program in case of another key.

Please be sure to input the above mentioned programs [D,2], [D,3], and [D,4] simultaneously since they are related to each other as mentioned so far.
When inputting is done, move the cursor to !(D,2) and press both [CTRL] and [P] to start the program.
Give it a try!

Framework d. Program for the printer-outputting of the words.

Let's output the words onto the printer and make what is like a word book. A program, of course, is needed for this process.

1. Move the cursor to !(D,5) and press both [CTRL] and [O] to input the following program:

Macro List

```
[D,5] @D:OUT "c"////,"PRINT WORDS TO PRINTER :"//," PRESS [RETURN] ":INPUT $:POU
T "c":ZS=1:%L:FOR ZY=ZS,ZS+59:IF ZY=!(C,0) THEN @END STOP:POUT !(A,ZY),!(B,ZY):N
EXT:POUT "c":ZS=ZS+60:JP L
```

Macro Instruction	Meanings
[D,5] @D:	Represents the name of a program.
OUT"c"//////,	Clears out the screen and feeds six lines. Note: If you started the program by mistake, just press [STOP].
"PRINT WORDS ...":	Indicates to output the words onto the printer. Prompts to press [RETURN] and feeds six lines.
INPUT $:	Tells you to press [RETURN].
POUT"c":	Feeds form paper by one page.
ZS=1:	Sets ZS as a number for the first word

Macro Instructions	Meanings
%L:	A label to return here with JP L.
FORZY=ZS,ZS+59:	It is setting a loop in order to output 60 lines of words per page.
IFZY=... STOP	Ends here since all the words will have been output when ZY becomes equal to !(C,0).
POUT!(A,ZY), !(B,ZY):	Outputs a word and its meaning in one row.
NEXT:	Carries forward a loop increasing ZY by one.
POUT"c":	Feeds form paper by one page. Note: This makes the line feed twice.
ZS=ZS+60:	Carries forward a word number to next.
JP_L	Returns to label %L.

It is now possible to output onto the printer. Are you ready with the printer? Move the cursor to !(D,5) and press [CTRL] and [P] to start the program. Words are now being printed out one after another, aren't they?

CTRL and P to start the program. Words are now being printed out one after another, aren't they?

Summary

We hope you have more or less gotten the idea about macro instructions by actually inputting some of the programs by yourselves. If you are not very sure with any of the words or the expressions, refer to the appropriate pages or NOTEs to check them out.

So far we've learned to form programs on the basis of Guideline 2 (page 77).

This may be found a good practice for learners to acquire higher aptitudes since they have to read and write the words many times due to the fact that a number of programs are to be carried out thoughout the whole process. This method, however, is little too bothersome for telephone directories or address books, isn't it? Then how about binding up the programs so as to select each of them as a set from a menu?

e. Adding some more menu-indications of the programs.

1. Move the cursor to !(A,0) and press [CTRL] and [O] to input the following program:

Macro List

[A,0] @X:@M	It's a program to jump to main program @M.

*Point: You can prevent a program from being deleted by pressing [CTRL] and [I] to apply protection after it has been input.

2. Move the cursor to !(F,0) so as to make a message for the menu. Change the length of this field !(F,0) into 30 characters by pressing [INS].
Input letters as follows into the fields from (F,0) to (F,6):

```
        [HUCAL WORDBOOK]
1. INPUT OF NEW WORDS
2. INPUT OF MEANING OF NEW WORDS
3. TESTING (WORD TO MEANING)
4. PRINTOUT OF WORD BOOK TO PRINTER
INPUT THE NUMBER YOU LIKE.
PROGRAM ENDS BY INPUT OF "0".
```

3. Next, input programs as follows into the fields from (G,0) to (G,3):

Macro List

[G,0] @Z:@A	Jump to @A
[G,1] @Z:@B	" @B
[G,2] @Z:@C	" @C
[G,3] @Z:@D	" @D

Note: This program lets the programs in !(H,0) jump to those formed according to Guideline 2 and to carry out its menu.

5. Input the following program into !(H,0):

Macro List

```
[H,0] @M:OUT "c"///:FOR ZY=0,6:OUT /20!(F,ZY):NEXT:INPUT ZJ:IF ZJ=0 THEN @EXIT S
TOP:IF ZJ<1;ZJ>4 THEN @M STOP:!(B,0)=!(G,ZJ-1):@Z
```

[We will explain further especially about the programs in !(H,0), since they are applicable in many ways.]

Macro Instruction	Meanings
[H,0] @M	Represents the name of a program. It is usually more convenient to start the program when the cursor is placed at !(A,0). On such occasions, write in jump instructions such @X:@M into !(A,0). The screen will be much clearer if a macro insturction to jump to is written away from !(A,0). So, that is why other macro instructions are written away from !(A,0).

*Point: An instruction to jump from one field to another is accompanied by @ like @X:@M, as instructing to jump from @X to @M. The combination of %L and JP␣L are used for jumping within a field (within a macro instruction).

OUT"c"///:	Clears the screen and feeds three lines.
FORZY=0,6:	Sets a loop condition to display the range of the row which has the message for a menu.
OUT/20!(F,ZY):	Feeds a line and displays the message written from column 20 to column F.
NEXT:	Carries forward a loop.
INPUT ZJ:	Inputs a work number into ZJ.
IFZJ=0... STOP:	Inputs a work number into ZJ. It means the work has been completed if ZJ is o.
IFZJ 1;...STOP:	Returns to @M if it is either smaller than 1 or larger than 4.
!(B,0)=!(G,ZJ-1):	Transfers a jump instruction specified by ZJ-1 to the field (B,0). If you have input 1 while trying to input a word, it will be ZJ-1=0. Then the programs @Z:@A will be written in !(G,0), as you can see, and this will be copied into !(B,0) as it is. Note: [B,0] will be an intermediate point when jumping to program @Z.

Macro Instructions	Meanings
@Z	Executes the program @Z. In this example, there is a total of 5 programs, one in !(B,0) and others in the field !(B,0) to !(B,3). Since it executes the program, it finds first, the program in !(B,0) will be the destination for @Z to jump to. As previously mentioned with !(B,0)=!(G,ZJ-1), for example, if program @Z:@A is copied in !(B,0), it will further jump to @A and then be executed. *Point: When looking for the name of a program, do so in the order such as (A,0), (B,0).....(u,0), then (A,D), (B,D).....(u,1).

6. Input the following program into the field (H,1):

   [H,1] @END:@M

   This program is designed for one to return to the menu when the work is done.

   * Now, instruction @END is found in each of program @A, @B, @C, and @D, and this program helps return to main program @M when it comes across it. Here, we have finally made what was initially planned.

   Move the cursor to !(A,0) and press CTRL and P to start the program.

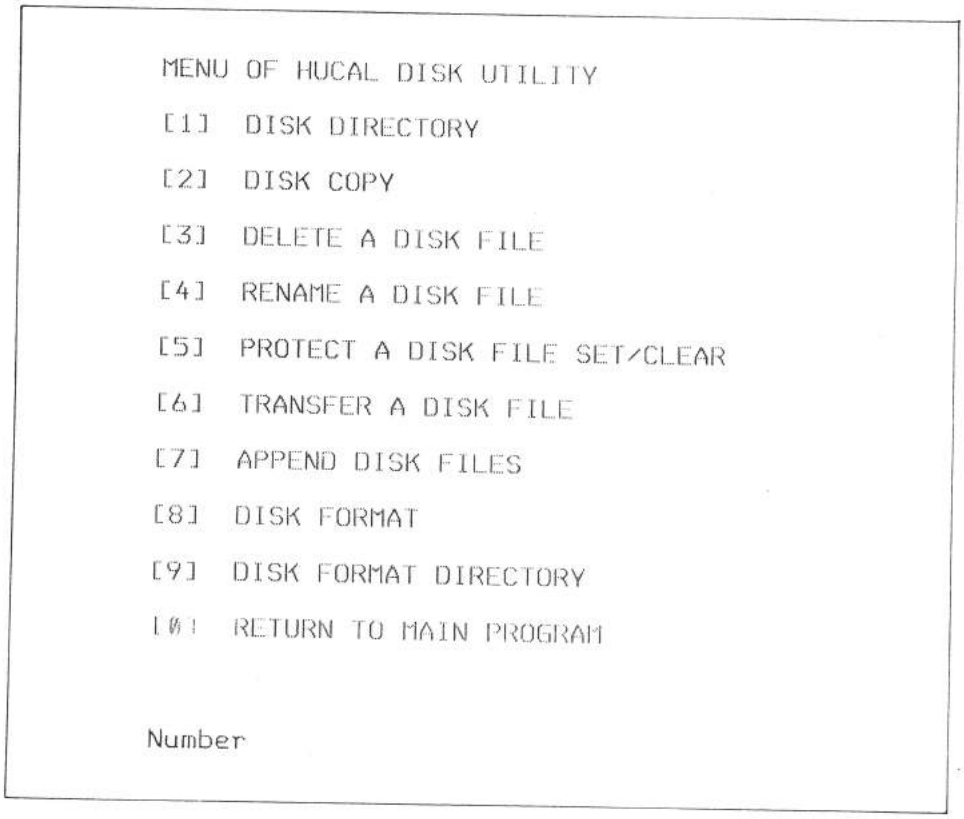
   Let's select a menu number and get started!

6. Utilities

(1) What is a Utility?

HuCAL has a powerful tool called "utility" for maintenance of data disk. Roughly speaking, it is a tool whose functions are to make the disk format for HuCAL's data disk, to transfer, to copy, or to combine files between disks, to provide write-protect on a disk or to rename a file, etc.

Then, let's start explaining each function one by one as you actually operate. Press [CTRL] + [Y] keys first. The screen will be turned to the following one.



(2) List of Functions of Utility

[1] DISK DIRECTORY	To check the contents of the disk.
[2] DISK COPY	To make a copy of all the contents in a disk.
[3] DELETE A DISK FILE	To delete a file in a disk.
[4] RENAME A DISK FILE	To rename a file in a disk.
[5] PROTECT A DISK FILE SET/CLEAR	To write-protect a file to prevent writing into the file by mistake.
[6] TRANSFER A DISK FILE	To move or transfer a file in a disk to another disk.
[7] APPEND DISK FILES	To append files in a disk.
[8] DISK FORMAT	To initialize a disk.
[9] DISK DIRECTORY FORMAT	To cancel all the registers of the file names in the formatted HuCAL's data disk.

For details, read the following pages.

(3) Making of a Data Disk

DISK FORMAT

Disk formatting (or initialization) in HuCAL means the function of reforming a blank disk or a used disk to that exclusively used for HuCAL's data. No matter what kind of disk you may use, please make it format once without fail.

Note: You don't have to worry about initializing a HuCAL's master disk by mistake because a message "THIS IS THE HuCAL MASTER DISK" will be on display.

[Procedure]

1 [CTRL] + [Y]

2 MENU on display

3 Select [8] and press [RETURN] key.

4 "INSERT A DISK FOR FORMAT AND HIT [SPACE] KEY"

5 Press [SPACE] key.

6 "OPERATION COMPLETED"

"HIT ANY KEY"

This completes formatting. After this, finish the operation by following the message.

7 [SPACE] key (Any key will do.)

8 "NUMBER"

9 [0]

10 [REUTRN] key

(4) Cancellation of a Disk File

DIRECTORY FORMAT

Directory format will be used to delete all the contents (files) in the formatted data disk by means of HuCAL's master disk.

Procedure

1 CTRL + Y

2 MENU on display

3 Select and press [9].

4 RETURN key

5 "INSERT A DISK FOR FORMAT AND HIT SPACE KEY"

6 Press SPACE key.

7 "OPERATION COMPLETED"
"HIT ANY KEY"

This completes directory format. Try selecting [1] of the disk directory and take a look at the contents of the file. You will find the files 1 through 8 empty.

* Be sure to execute formatting. If you desire to delete all the files in a disk which is one formatted, please execute directory format.

(5) List of File Names in a Disk

DISK DIRECTORY

To check what is filed in a data disk, there are two methods. The one is; pressing of [CTRL] + [B] or executing "DISKLOAD" of the "TRANS" in the menu located on the lower left corner of the screen. By doing so, the file names of the data to be loaded will be on display.

The other is; execution of this DISK DIRECTORY. This is used just to know what is in files in a disk or in case there is no need to load a disk for particular reason.

[Procedure]

1 [CTRL] + [Y] keys

2 MENU on display

3 Select and press [1].

4 File menu in the disk on display

5 "HIT ANY KEY ▨ "

When this is displayed, return to MENU by pressing any key you want.

(6) Copying of All the Contents in a Disk

If you want one more disk which is the same as that of an important disk of your own, you can do it by using the copying function provided with HuCAL. However, the master disk cannot be copied with this function. If you try, it may end up with broken master disk, so please don't copy it.

The data files in the master disk can, however, be transferred or deleted. If you want to copy all the data in the files in HuCAL's master disk, transfer each file to another data disk first, and then copy that data disk to another one.

Procedure

1 CTRL + Y keys

2 MENU on display

3 2 , RETURN key

4 "INSERT THE MASTER DISK AND HIT SPACE KEY"

5 Press SPACE key

Procedure

6 "DO YOU WNAT TO FORMAT A DESTINATION DISK? [Y/N] "

7 [N] key (or [Y] in case of formatting at this time)

8 "INSERT A TRANSFER SOURCE DISK AND HIT [SPACE] KEY"

9 Press [SPACE] key

10 "INSERT A TRANSFER DESTINATION DISK TO A DRIVE AND HIT [SPACE] KEY"

11 Press [SPACE] key

12 Repeat 2 or 4 times (Process 8 to 11)

13 "OPERATION COMPLETED"
"HIT ANY KEY"

This completes copying of all the contents.

To return to the original screen, input [0], [RETURN] key, insert HuCAL's master disk and press [SPACE] key.

* A message "INSERT THE MASTER DISK" will appear before and after the operation. This is to prevent the table in the making now from being broken by executing this. The computer once saves the file and automatically loads it at the same time when the execution ends.

(7) Deletion of a Disk File

To delete an unnecessary data file, follow the Procedure below:

[Procedure]

1 [CTRL] + [Y] keys

2 MENU on display

3 Select "[3] DELETE A DISK FILE".

4 [3], [RETURN] key

5 On pressing the [RETURN] key, a list of file names in the disk to be deleted will be on display. Input a number of a file you want to delete and press the [RETURN] key again.

6 "OPERATION COMPLETED"
"HIT ANY KEY ▨ "

This completes deletion. To continue to delete, repeat this procedure from the beginning.

(8) Renaming a Disk File

You can rename a file in a disk as much as you want. When saving data, the file name is to be asked without fail. If a file which has already been made is to be saved again, just pressing of [RETURN] key will do.

Therefore, be sure to name a file.

[Procedure]

1 [CTRL] + [Y]

2 MENU on display

3 [4], [RETURN] key

4 A list of the files in the disk.
Input the file number to be renamed.

5 Input a new name as a message "INPUT A NEW FILE NAME ▨ " is on display

6 "OPERATION COMPLETED"
Renaming will be completed as a message "HIT ANY KEY ▨ " is on display.

(9) Setting/Clearing of Write-protect on a Disk

Set write-protect on important files which contain important data. Clearing is as easy as a protective sticker which can be stuck to or detached from a disk and you can tell such a file with write-protect from ordinary ones by an indication of "*" (asterisk). If you protect a file, a message "THIS FILE IS WRITE-PROTECTED" will be on display when such a protected file is about to be deleted by mistake and that prevents important files from being deleted.

Procedure

1 CTRL + Y

2 MENU on display

3 5 , RETURN key

4 A list of file name in the disk will be on display. Input the file number to be set/cleared.

5 "PRESS [1] FOR SETTING, [0] FOR CLEARING ▨ "

6 Here, setting of write-protect is done by pressing [1] key and clearing of it is done by pressing [0] key.

(10) Transfer of a Disk File

You can transfer a file from the master disk to a data disk or from a data disk to another data disk. Unlike copying all the contents in a disk, each one of the files can be transferred. Therefore, it is indispensable when copying files in the master disk.

[Procedure]

1 [CTRL] + [Y]

2 MENU on display

3 [6], [RETURN] key

4 "INSERT TRANSFER SOURCE DISK AND HIT [SPACE] KEY"

5 Press [SPACE] key.

6 A list of file names in the transfer source disk will be on display. Input the file number to be transferred.

7 [RETURN] key

8 "INSERT A TRANSFER DESTINATION DISK AND HIT [SPACE] KEY"

9 Press [SPACE] key.

10 This time, a list of file names in the trnasfer destination disk will be on display. Input the file number to be directed.

11 "OPERATION COMPLETED"

Transfer is completed as a message "HIT ANY KEY ▨ " is on display.

To return to the original screen, input [0] [RETURN] key, insert HuCAL's master disk and then press [SPACE] key.

(11) Appending of Disks

Appending of disks is a function which can append the contents of a file to another file without changing them. This function is widely applicable because you can move and append file data only. For example, suppose you have a File 1, which has Macro instructions only, and Files 2 through 8, which have tables as data. Then, if you append the file which has Macro instructions only (File 1) to that which has a monthly data (one of Files 2 through 8), you don't have to take time inputting a long program to each data file. Moreover, you can append two data files to output the data for January and February only. Therefore, a file with data for the whole year can be prepared for stock. Other than these, application of this function is very wide. Devise usage according to your needs.

[Procedure]

1 [CTRL] + [Y]

2 MENU on display

3 [7], [RETURN]

4 "INSERT A DISK TO BE APPENDED FIRST AND HIT [SPACE] KEY"

5 On pressing [SPACE] key, a list of file names in the disk is on display and a file number to be appended is asked. Input the number and press [RETURN] key.

Procedure

6 "INSERT A DISK TO BE APPENDED NEXT AND HIT [SPACE] KEY."
[RETURN] THEN ENDING

7 Press [SPACE] or [RETURN] key.
On pressing [SPACE] key, a list of the second file names will be on display.
Input the file number and press [RETURN].

8 Repeat Procedures 6 to 7 if you need.

9 "INSERT A TRANSFER DESTINATION DISK AND HIT [SPACE] KEY"

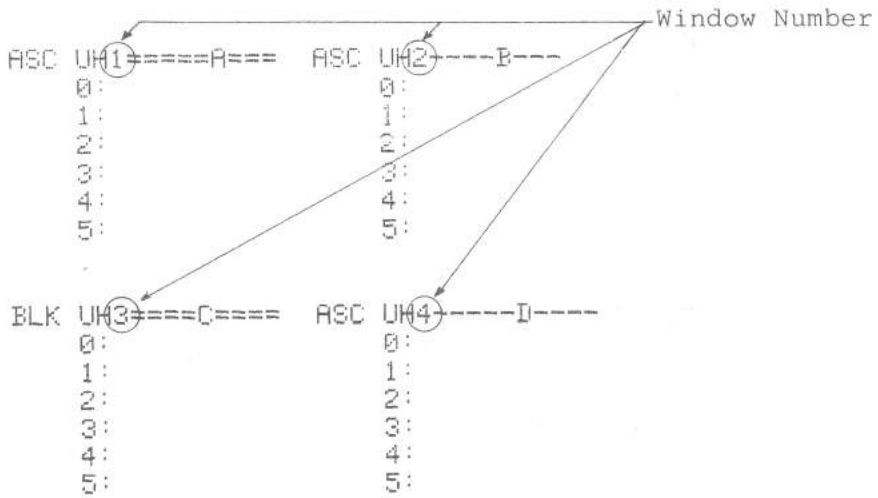
10 On pressing [SPACE] key, a list of file names of a disk to which an appended file is to be written will be on display.
Input the file number and press [RETURN] key.

11 A new file name is asked. Input the name of the appended file.[Note)] Pressing of [RETURN] key will complete the execution.
To return to the original screen, input [0], [RETURN] key, insert HuCAL's master disk and press [SPACE] key.

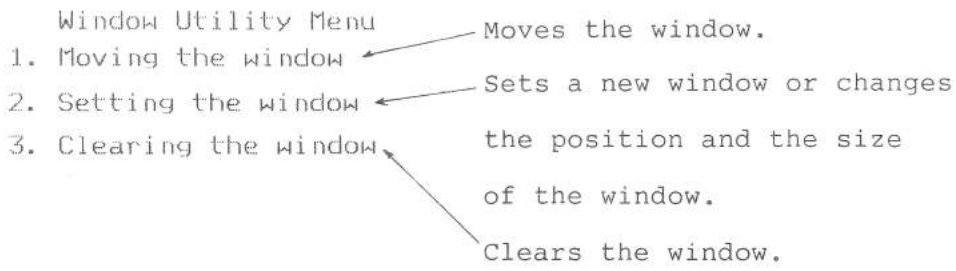
Note: When saving, if the file name remains the same, inputting of a file name is not necessary. However, in case of appending, input the same name again without fail.

7. Window Function

In HuCAL, the screen can be divided into up to four sub-screens by using a window function.



By pressing [CTRL] + [N], a menu of the window utility will be on display as follows:



1. Moving the window: This is to be selected when you move the cursor to other window.

Select a Number (1-3) ▨ - [1] [RETURN]

Window Number of the Destination ? (1-4) - Input the number desired.

2. Setting the window: This is to be selected when you add a new window or change the position or the size of the window already displayed.

Select a Number (1-3) ▨ - [2] [RETURN]

Window Number ? (2-4) - Input the number desired.

* To determine the position and the size, take the following procedure:

(1) Put the cursor on the upper left corner of the window and press [RETURN].

(2) Put the cursor on the lower right corner of it and press [RETURN].

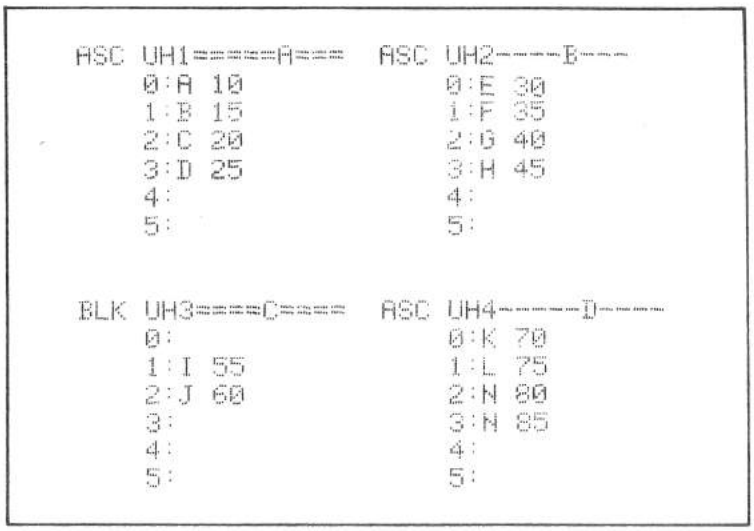
3. Clearing the window: This is to be selected when you want to erase the window(s) on display. Note that Window No.1 cannot be erased.

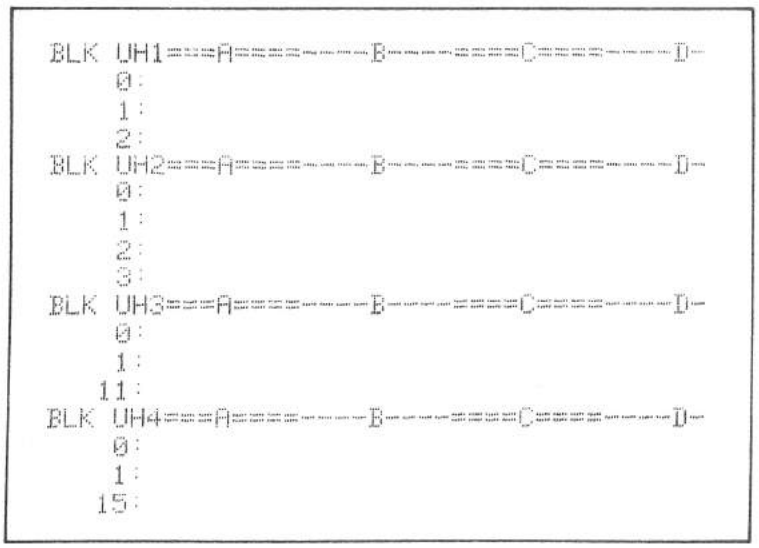
Select a Number (1-3) ▨ - [3] [RETURN]

Window Number ? (2-4) - Input the number desired.

Note: If you input the data into a field in a window, that data will not appear in the same field of other window (even if the screen displays the fields at a time). To make it appear, press [CLS] key.

Example of the windows





8. Quick Reference on Keys, Functions and Macro Instructions

(1) Function keys

Since they are set as follows, you'll find them helpful when used for inputting expressions or macro-instructions.

```
FUNC + 1  INPUT
FUNC + 2  OUT
FUNC + 3  SWAP
FUNC + 4  THEN
FUNC + 5  STOP
FUNC + 6  MEAN(
FUNC + 7  SUM(
FUNC + 8  MAX(
FUNC + 9  MIN(
FUNC + 0  COUNT(
```

(2) CONTROL keys

The [CTRL] key, similar to the [GRAPH] key, does not do anything when used by itself.

It only works when used together with other keys simultaneously.

HuCAL gives the following functions to the CONTROL key:

CTRL + D	To change the calculating direction.
CTRL + E	To delete the data input in a field where the cursor is. [A field marked with an asterisk (*), however, has to be released from protection with the CTRL and I keys together first.]
CTRL + F	To specify the number of decimal digits (0-14) [Effective in the vertical direction.]
CTRL + G	To switch ON/OFF the protection mode of the field-input. P will be indicated for the protection mode, and the data input at this moment will be protected. U will be indicated when it is not protected. An asterisk (*) will be indicated for any field that is protected.
CTRL + I	To switch ON/OFF the protection of a field. (This will only be conducted for a field that has already been input with certain data.)
CTRL + J	To make the cursor jump to a specified field.

CTRL + O	To start the inputting of a macro-instruction. Inputting will be done on the drum and completed with the RETURN key.
CTRL + P	To execute a macro-instruction.
CTRL + Q	To display the remaining capacity of the memory and the explanation on the CONTROL key.
CTRL + S	To save data on a casette tape.
CTRL + T	To load data from a casette tape.
CTRL + U	To divide numerals every 3 digits with commas or undo (FORMAT specification) [Effective in the vertical direction]
CTRL + V	To search for data in a field.
CTRL + W	To sort out data in a field.
CTRL + X	To output a macro-field to the printer.
CTRL + Z	To recalculate all the expression fields.

(3) Other Keys

!	To change the length of a field (3-32).
#	To switch ON/OFF the auto-calculation mode. It will recalculate all the expression fields when the [RETURN] is pushed with this mode ON. It will not recalculate, however, if the data has been input with the cursor key. Calculation in the vertical direction takes 100 times as long as those in the horizontal direction.
$	To take a hard copy of the screen on the printer.
%	Although a cursor cannot be moved onto a frame unless the top row is 0 or the very left column is A, it will be moved, even if not under such restrictions, when this key is pressed. It will also release the mode.
&	To delete all the data.
"	To draw a horizontal line on a field.
'	To put the content of a field to the left.
(	To put the content of a field to the center.
)	To put the content of a field to the right.
*	To switch ON/OFF the auto-addition mode when numerical data is input into a field.
[FUNC] + [E]	Input ASC Field on the DRUM.

/ To start inputting an expression.
Imputting will be performed on the drum and completed with the [RETURN] key.
To make the editing of a character field possible.
Editing will be performed on the drum and completed with the [RETURN] key.

INST
1) Lengthens a field —— when the field is blank (BLK)
2) Inserts a field ——— when the cursor is on a frame
3) Inserts a letter ——— in the edit mode

DEL
1) Shortens a field ——— when the field is blank (BLK)
2) Deletes a field ——— when the cursor is on a frame
3) Deletes a letter ——— in the edit mode

HOME
1) Moves the cursor to the upper left corner of the current screen.
2) Moves the cursor to the field !(A,0) when pressed together with [SHIFT].

BREAK
Stops the execution of macro instruction or searching when pressed together with [CTRL].

SPACE KEY Stops the execution of macro instruction temporarily and resumes it (when stopping)

(4) Functions

The inputting of functions is to be done on the drum with the [/] (SLASH) key pressed and be completed with the [RETURN] key. The result of calculations will be output immediately.

FUNCTION	EXPRESSION	EXPLANATION
SUM	SUM(a,x,b,y)	To calculate the total of numerical data embedded in the area whose diagonals extend from row a column x to row b column y.
MEAN	MEAN(a,x,b,y)	To calculate the mathematical mean of numerical data embedded in the area whose diagonals extend from row a column x to row b column y.
MAX	MAX(a,x,b,y)	To ask for the maximum value of numerical data embedded in the area whose diagonals extend from row a column x to row b column y.

FUNCTION	EXPRESSION	EXPLANATION
MIN	MIN(a,x,b,y)	To ask for the minimum value of numerical data embedded in the area whose diagonals extend from row a column x to row b column y.
COUNT	COUNT(a,x,b,y,m,n)	To count the number of numerals larger than m and smaller than n from among data embedded in the area whose diagonals extend from row a column x to row b column y.
INT	INT(x)	To ask for the integral part of numerical data x. [Example] INT(3.87)=3 INT(0.6)=0 INT(-1.5)=-1
SQR	SQR(x)	To ask for the square root of numerical data x as $\sqrt{x}$. [Example] SQR(2)=1.4142 SQR(-1)=0
ABS	ABS(x)	To ask for the absolute value of numerical data x as $\lvert x \rvert$. [Example] ABS(3)=3 ABS(-3)=3 ABS(-1.5)=1.5

FUNCTION	EXPRESSING	EXPLANATION
SIN	SIN(x)	To ask for the sine of numerical data x as sin(x) (x should be radian.) [Example] SIN(90*3.141592/180)=1
COS	COS(x)	To ask for the cosine of numerical data x as cos(x) (x should be radian.) [Example] COS(60*3.141592/180)=0.5
ATN	ATN(x)	To ask for the arctangent of numerical data x as arctan (x) (x should be radian.) [Example] ATN(1)*180/(4*ATN(1))=45
FAC	FAC(x)	To ask for the x! of numerical data x. However, if x 0<then FAC(x)=0 Displays maximum value for the system when x is more than 49.

FUNCTION	EXPRESSING	EXPLANATION
LN	LN(x)	To ask for the logarithm of numerical data x as log x (ln x). However, $LOG_{10}(x)=LN(x)/LN(10)$ [Example] LN(2)=0.6931 LN(2*3)=1.7918 LN(2)+LN(3)=1.7918
EXP	EXP(x)	To ask for the expotential function of numerical data as e^x However, if x -145 then EXP(x)= 0 and if x +145 then EXP(x)=
DEG	DEG(x)	To convert the degree of numerical data x into the radian. $(DEG(x)=x*\frac{\pi}{180})$ [Example] DEG(60)=1.0472 SIN(DEG(90))=1 COS(DEG(60))=0.5
!	!(X,Y)	To display the value of the designated NUM or EXP field.

(5) Macro Instructions

Macro Instructions can be input onto the drum by pressing [CTRL] + [O] at the same time and can be registered by pressing the [RETURN] key. To execute macro instructions, move the cursor onto the field in which an instruction to be executed is registered and press [CTRL] + [P] simultaneously.

Instruction (Format)	Meaning
If condition THEN process 1	If condition is satisfied, process 1 is executed and if not, (process 2 is executed and) the judgement of condition is ended.
JP label Ex.: %L ... JP␣L	Branches to a macro instruction with label.
INPUT variable or expression	Used when inputting to a variable with the name variable or expression.
OUT variable or character string or control character string	Outputs the contents of variable or character string onto the screen according the function of the specified control character string.

Instruction (Format)	Meaning
POUT variable or character string or control character string	Outputs the contents of variable or character string onto the printer according to the function of the specified control character string.

[Meaning of Control Character String]

Control character string	OUT	POUT
h	Cursor home	
c	Clear screen	Home feed (Page feed)
s	——	Small letter size
n	——	Normal character
e	——	Enlarged character
r	——	Standard line feed
p	——	Interlinear pressed line feed
/ (Slash)	Line feed	Line feed

Ex.: OUT"c"//,"ABC"

Clears the screen, feeds two lines and outputs ABC on the screen.

POUT"c","e","ABC"	Performs home feed (page feed) and outputs ABC with enlarged characters to the printer.
HDCP field positon (, number of lines) Ex.: HDCP!(B,1), 20	Outputs the chart as the field specified by field positon as home position (and number of lines as specified).
SWAP variable 1, variable 2	Swaps the contents of variable 1 and variable 2.
Variable 1=variable 2 or constant or expression	Substitutes either variable 2, constant or expression for variable 1. (The left and right sides should be of the same type.)
FOR system variable =initial value, final value process NEXT	Repeats the process up to (final value - initial value) + 1 times of system variable.
Ex.: FOR ZA=1,8:!(A, ZA:=ZA:NEXT	Figures 1 through 8 are input to the field from !(A,1) to !(A,8) sequentially one for each field.

[MEMO]

PART II

APPLICATIONS

From now on we will be working on some applied programs. Firstly, we will start with those relevant to statistics.

Each one is supplemented with an explanation of how to use the chart and sample formulas and programs for your reference.

In case of SC-3000, we have to put so many tables for explanation just to show the whole table due to the columns per line. Therefore, a screen of a personal computer which can display 80 columns/line is used here. Read the following explanation as they are a virtual screen of SC-3000.

1. Frequency Distribution (Statistics)

```
BLK UH====A=======----B----=C=----D----====E====---F---=============G=============
    0:@M:ZZ=79:
    1:@A1:ZB=!(              [ H I S T O G R A M ] THEME...AGE OF STAFF
    2:@A2:ZA=7:    MAX.           MIN.     AVE.   DATA                    DEVIATION
    3:@F:ZZ=80:      54             20       31     50                            8
    4:
    5:            WIDTH   OF    LEVEL     FREQ.    (%)    20    40    60    80   100
    6:                                                 +-----+-----+-----+-----+-----+-
    7:        1.     20   --       24        10     20 *****
    8:        2.     25   -        29        13     26 *******
    9:        3.     30   -        34        15     30 ********
   10:        4.     35   --       39         3      6 **
   11:        5.     40   -        44         5     10 ***
   12:        6.     45   -        49         1      2 *
   13:        7.     50   -        54         3      6 **
   14:#                   -
   15:        9.          -
   16:       10.          -
   17:       11.          -
   18:       12.          -
   19:       13.          -
```

Calculate the maximum and minimum and average values by inputting the data after deciding the theme. Secondly, indicate the number of data and the standard deviation, frequency, relative frequency, and histogram by inputting the width of the level.

Input the theme into !(G, 1) and the data into rows after 21 of column A. Data can be input into rows from the 21st to the 10000th on column A.

When the [CTRL] and [Z] keys are pressed simultaneously after the data has been input, the maximum, minimum and average values will be calculated, so input the width of the level into columns B and D. Input '#' into the field where the row number on column A after the last width the level is. ([A] [DEL] [#]).

Calculation will be carried out when the cursor is moved to !(A,0) and keys [CTRL] and [P] are pressed at the same time. When wishing to output the frequency distribution list to the printer, execute @P in !(A,3).

```
BLK UH====A=====----B----=C=----D----====E====---F---============G============
    0:@M:ZZ=79:
    1:@A1:ZB=!(                      [ H I S T O G R A M ] THEME...AGE OF STAFF
    2:@A2:ZA=7:       MAX.          MIN.        AVE.     DATA                      DEVIATION
    3:@P:ZZ=80:        54             20          31       50                              8
    4:            ----------------------------------------------------------------------------
    5:              WIDTH   OF     LEVEL     FREQ.      (%)     20     40     60     80    100
    6:            -------------------------------------------+------+------+------+------+------+--
    7:          1.          -
    8:          2.          -
    9:          3.          -
   10:          4.          -
   11:          5.          -
   12:          6.          -
   13:          7.          -
   14:#                     -
   15:          9.          -
   16:         10.          -
   17:         11.          -
   18:         12.          -
   19:         13.          -
```

[Input Example] When wishing to survey the age constitution of 50 employees.

```
BLK UH====A=====----B----=C=----D----====E====---F---============G============
   20:
   21:        20
   22:        36
   23:        44
   24:        42
   25:        23
   26:        27
   27:        29
   28:        28
   29:        21
   30:        33
   31:        30
   32:        31
   33:        32
   34:        33
   35:        34
   36:        40
   37:        25
   38:        24
   39:        26
```

Input the data of the 50 cases into column A.

```
BLK UH====A=====----B----=C=----D-----====E====---F---============G============
   0:@M:ZZ=79:
   1:@A1:ZB=!(                [ H I S T O G R A M ] THEME...AGE OF STAFF
   2:@A2:ZA=7:      MAX.          MIN.      AVE.    DATA                DEVIATION
   3:@F:ZZ=80:        54            20        31      50                        8
   4:             -----------------------------------------------------------------
   5:              WIDTH   OF     LEVEL     FREQ.    (%)    20   40   60   80  100
   6:             ---------------------------------------+----+----+----+----+----+-
   7:        1.       20    --       24        10     20 *****
   8:        2.       25    --       29        13     26 *******
   9:        3.       30    --       34        15     30 ********
  10:        4.       35    --       39         3      6 **
  11:        5.       40    --       44         5     10 ***
  12:        6.       45    --       49         1      2 *
  13:        7.       50    --       54         3      6 **
  14:#                      --
  15:        9.             --
  16:       10.             --
  17:       11.             --
  18:       12.             --
  19:       13.             --
```

When keys [CTRL] and [Z] are pressed after the data has been input, the maximum, minimum and average values will be calculated.

```
BLK UH====A=======----B----=C=----D-----====E====---F---============G============
   0:@M:ZZ=79:
   1:@A1:ZB=!(                [ H I S T O G R A M ] THEME...AGE OF STAFF
   2:@A2:ZA=7:      MAX.          MIN.      AVE.    DATA                DEVIATION
   3:@F:ZZ=80:        54            20        31      50                        8
   4:             -----------------------------------------------------------------
   5:              WIDTH   OF     LEVEL     FREQ.    (%)    20   40   60   80  100
   6:             ---------------------------------------+----+----+----+----+----+-
   7:        1.       20    --       24
   8:        2.       25    --       29
   9:        3.       30    --       34
  10:        4.       35    --       39
  11:        5.       40    --       44
  12:        6.       45    --       49
  13:        7.       50    --       54
  14:#                      --
  15:        9.             --
  16:       10.             --
  17:       11.             --
  18:       12.             --
  19:       13.             --
```

Input the width of the level and then input # into the row after the last width of the level on column A.

[Calculating Formulas]

Location of the cursor	
!(B, 3)	MAX(A,21,A,10000)
!(D, 3)	MIN(A,21,A,10000)
!(E, 3)	MEAN(A,21,A,10000)

They are to ask for the maximum, minimum and average values of the input data.

[Explanation of the Macro-instruction]

```
[A0] @M:ZZ=79:OUT "c",//////" WAIT A MOMENT....":ZA=!(B,3):ZB=!(D,3):!(F,3)=COU
NT(A,21,A,10000,ZB,ZA):@A1
```

It counts the amount of data input.

```
[A1] @A1:ZB=!(E,3):ZC=!(F,3):ZD=0:FOR ZA=21,ZC+20:ZE=!(A,ZA)-ZB:ZD=ZD+ZE*ZE:NEX
T:!(G,3)=SQR(ZD/ZC):@A2
```

Obtains the standard deviation.

```
[A2] @A2:ZA=7:%L:ZB=!(B,ZA):ZD=!(D,ZA):!(E,ZA)=COUNT(A,21,A,10000,ZB,ZD):!(F,ZA
)=INT(!(E,ZA)/ZC*100+.5):ZE=INT(!(F,ZA)/4+.5):$=!(H,ZE):!(G,ZA)=$:ZA=ZA+1:IF !(
A,ZA)="#" THEN @END STOP:JP L
```

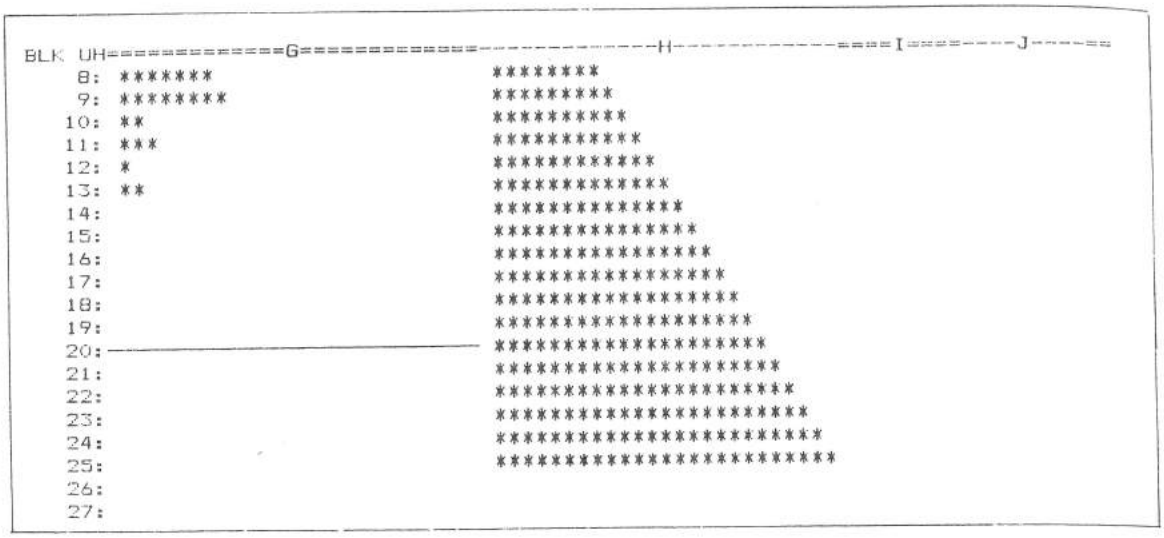
Calculates the frequency and relative frequency of each level and writes in the histogram.

```
[A3] @P:ZZ=80:ZA=1:%L:POUT !(B,ZA),!(C,ZA),!(D,ZA),!(E,ZA),!(F,ZA),!(G,ZA):ZA=Z
A+1:IF !(A,ZA)="#" THEN JP L1 STOP:JP L:%L1:POUT !(B,20),!(C,20),!(D,20),!(E,20
),!(F,20),!(G,20)
```

Prints out the frequency distribution list.

[Explanation of other fields]

```
BLK UH==========G==========----------H----------====I=======J----==
    0:
    1:THEME...AGE OF STAFF          *
    2:                    DEVIATION **
    3:                            8 ***
    4:-------------------------------****
    5:    20   40   60   80  100    *****
    6:+----+----+----+----+----+-   ******
    7: *****                        *******
    8: *******                      ********
    9: ********                     *********
   10: **                           **********
   11: ***                          ***********
   12: *                            ************
   13: **                           *************
   14:                              **************
   15:                              ***************
   16:                              ****************
   17:                              *****************
   18:                              ******************
   19:                              *******************
```



The graph pattern is input into column H. Refer to the explanation on other fields of the sales accomplish list, which will be mentioned later, to know how to perform inputting.

[Printing Example]

[H I S T O G R A M] THEME...AGE OF STAFF

MAX.	MIN.	AVE.	DATA	DEVIATION
54	20	31	50	8

WIDTH	OF	LEVEL	FREQ.	(%)	20 40 60 80 100
20	–	24	10	20	*****
25	–	29	13	26	*******
30	–	34	15	30	********
35	–	39	3	6	**
40	–	44	5	10	***
45	–	49	1	2	*
50	–	54	3	6	**

2. Home

How was the statistics? Could you enjoy it with all the necessary data input? It's really handy and easy to use, so don't give up until the last column and carry on with the input of data.

Now we are up to its use for home. We have a housekeeping book, directory, loan calculations, and so on, to meet your need and pleasure.

(1) Housekeeping Book

```
BLK UH=====A=====-----B-----====C=====----D----====E====-----F-----====G=====-----H----
    0:@A:ZZ=79:Z@A1:FOR Z@A2:FOR Z@P:ZZ=80:
    1:                            [ H O U S E K E E P I N G   B O O K ]
    2:-------------------------------------------------------------------------------
    3:   DATE          INCOME       FOOD  HOUSING  CLOTHES  LEISURE   OTHERS  BALANCE
    4:-------------------------------------------------------------------------------
    5:  TRANSFER                                                                  500
    6:1984.  3.  1                 14      200                  4        2        280
    7:1984.  3.  2                  5                11         2        4        258
    8:1984.  3.  3                  3                                   7        248
    9:1984.  3.  4                 23                 7         6                 212
   10:1984.  3.  5     125         10                          15        6        306
   11:1984.  3.  6                  2                           2        5        297
   12:1984.  3.  7                 15        8                           2        272
   13:1984.  3.  8                  4                20         9        1        238
   14:1984.  3.  9                 10                           3       10        215
   15:1984.  3.10                  18       15                           5        177
   16:#                125        104      223       38        41       42
   17:
   18:
   19:
```

It is divided into date, income, and expenditure is subdivided into food expense, housing expense, clothing expense, leisure expense, and others. We have 5 subdivisions for expenditure with reference to a multiple-column account book. The total will be shown in the row with '#'.

Firstly, input the amount of carry-over into !(H,5). Start it from row 6. Input the dates into column A ([A] [DEL]), incomes into B, food expenses into C, housing expenses into D, clothing expenses into E, leisure expenses into F, and others into column G.

Be sure to enter # into the row after the last date ([A] [DEL] [#]).

The calculation will be carried out when the cursor is moved to !(A,0) and keys [CTRL] and [P] are pressed at the same time. Execute @P of !(D,0) when wishing to output the housekeeping book to the printer.

```
BLK UH=====A=====-----B----====C====----D----====E====----F----====G====----H----
    0:@A:ZZ=79:Z@A1:FOR Z@A2:FOR Z@P:ZZ=80:
    1:                           [ H O U S E K E E P I N G   B O O K ]
    2:----------------------------------------------------------------------------
    3:    DATE          INCOME        FOOD   HOUSING  CLOTHES  LEISURE    OTHERS  BALANCE
    4:----------------------------------------------------------------------------
    5:
    6:
    7:
    8:
    9:
   10:
   11:
   12:
   13:
   14:
   15:
   16:
   17:
   18:
   19:
```

[Input Example]

```
BLK UH=====A=====-----B----====C====----D----====E====----F----====G====----H----
    0:@A:ZZ=79:Z@A1:FOR Z@A2:FOR Z@P:ZZ=80:
    1:                           [ H O U S E K E E P I N G   B O O K ]
    2:----------------------------------------------------------------------------
    3:    DATE          INCOME        FOOD   HOUSING  CLOTHES  LEISURE    OTHERS  BALANCE
    4:----------------------------------------------------------------------------
    5:   TRANSFER                                                                    500
    6:1984. 3. 1                       14       200                 4         2      280
    7:1984. 3. 2                        5                  11       2         4      258
    8:1984. 3. 3                        3                                     7      248
    9:1984. 3. 4                       23                   7       6                212
   10:1984. 3. 5          125          10                          15         6      306
   11:1984. 3. 6                        2                           2         5      297
   12:1984. 3. 7                       15         8                           2      272
   13:1984. 3. 8                        4                  20       9         1      238
   14:1984. 3. 9                       10                           3        10      215
   15:1984. 3.10                       18        15                           5      177
   16:#                   125         104       223        38      41        42
   17:
   18:
   19:
```

[Explanation of the Macro-instruction]

```
[A0] @A:ZZ=79:ZA=6:%L:IF !(A,ZA)="#" THEN ZA=ZA-1:OUT "c",//////" WAIT A MOMENT
.......":@A1 STOP:ZA=ZA+1:JP L
```

It counts up to which row the data has been input.

```
[B0] @A1:FOR ZB=6,ZA:!(H,ZB)=!(H,ZB-1)+!(B,ZB)-SUM(C,ZB,G,ZB):NEXT:@A2
```

Calculates daily remainders.

```
[C0] @A2:FOR ZC=B,G:!(ZC,ZA+1)=SUM(ZC,5,ZC,ZA):NEXT:@END
```

Prints out the housekeeping book.

[Printing Example]

[H O U S E K E E P I N G B O O K]

DATE	INCOME	FOOD	HOUSING	CLOTHES	LEISURE	OTHERS	BALANCE
TRANSFER							500
1984. 3. 1		14	200		4	2	280
1984. 3. 2		5		11	2	4	258
1984. 3. 3		3				7	248
1984. 3. 4		23		7	6		212
1984. 3. 5	125	10			15	6	306
1984. 3. 6		2			2	5	297
1984. 3. 7		15	8			2	272
1984. 3. 8		4		20	9	1	238
1984. 3. 9		10			3	10	215
1984. 3.10		18	15			5	177

[Applications]

Since it is designed on the basis of a multi-column account book, it can also be used as a small-sum cash account book or journal.

(2) Installment calculations

[REDEMPTION]...EQUAL-PRINCIPAL REDEMPTION

TIMES	LOANED	INTEREST	INSTALLMENT	REMAINED
1	83,333	85,000	168,333	916,667
2	83,333	77,917	161,250	833,334
3	83,333	70,833	154,166	750,001
4	83,333	63,750	147,083	666,668
5	83,333	56,667	140,000	583,335
6	83,333	49,583	132,916	500,002
7	83,333	42,500	125,833	416,669
8	83,333	35,417	118,750	333,336
9	83,333	28,334	111,667	250,003
10	83,333	21,250	104,583	166,670
11	83,333	14,167	97,500	83,337
12	83,337	7,084	90,421	0
TOTAL	1,000,000	552,502	1,552,502	

[The Input Data]

Amount of loan	¥1,000,000
Number of installments	12
Number of installments a year	1
Annual interest rate	8.5 %

If Amount of Loan, Number of Installments, Numer of Installments a Year, and Annual Interest Rate are input, it will calculate, on the basis of the equal-principal redemption system, the amount of principal, the interest, the sum of the principal and interest, and remaining amount of principal and it will output the redemption list on to the screen.

Answer the questions as it will be carried out when the cursor is moved to!(A,0) and [CTRL] and [P] are pressed.

Execute @P in !(A,4) when printing out simultaneously the redemption list to the printer.

```
BLK UH=A-----B--=======C=======------D----------=======E=======------F----------====G===
    0:@M:
    1:@A:                [ REDEMPTION ]...EQUAL-PRINCIPAL REDEMPTION
    2:@A1
    3:@A2-------------------------------------------------------------------------
    4:@P:   TIMES          LOANED      INTEREST  INSTALLMENT      REMAINED
    5:@D:-------------------------------------------------------------------------
    6:
    7:
    8:
    9:
   10:
   11:
   12:
   13:
   14:
   15:
   16:
   17:
   18:
   19:
```

[Explanation of the Macro-instruction]

```
[A0] @M:ZZ=79:IF ZB>0 THEN @D STOP:@A
```

If some data is already written on the front, execute @D to delete the data.

```
[A1] @A:OUT "c",////"AMOUNT LOANED":INPUT ZA:OUT ZA," DOLLAR",//"TIMES OF INSTA
LLMENTS":INPUT ZB:OUT ZB," TIMES",//"TIMES OF INSTALLMENTS A YEAR":INPUT ZC:OUT
 ZC," TIMES",//"ANNUAL INTEREST RATE":INPUT ZD;OUT "c",//////"WAIT A MOMENT....
.":@A1
```

Input the amount of loan, the total number of installments, the number of installments a year and the annual interest rate.

```
[A2] @A1:ZR=ZD/ZC/100:ZE=ZA:FOR ZF=6,ZB+4:!(B,ZF)=ZF-5:!(C,ZF)=INT(ZA/ZB+.5):!(
D,ZF)=INT(ZE*ZR+.5):!(E,ZF)=!(C,ZF)+!(D,ZF):!(F,ZF)=ZE-!(C,ZF):ZE=ZE-!(C,ZF):NE
XT:@A2
```

Calculate the amount of principal to be paid back, the interest to be paid back, the sum of principal and interest to be paid back, and the remaining amount of principal of each time.

```
[A3] @A2:!(B,ZF)=ZB:!(C,ZF)=!(F,ZF-1):!(D,ZF)=INT(!(F,ZF-1)*ZR+.5):!(E,ZF)=!(C,
ZF)+!(D,ZF):!(F,ZF)=0:ZF=ZF+2:$=!(J,0):!(B,ZF)=$:!(C,ZF)=ZA:!(D,ZF)=SUM(D,6,D,Z
F-2):!(E,ZF)=!(C,ZF)+!(D,ZF)
```

Calculate the amount of principal to be paid back, the interest to be paid back, and the sum of principal and interest to be paid back, and the total of principal paid back, the total of interest paid, and the total of principal and interest paid back for the last time.

Note: The error by rounding is adjusted at the last installment.

```
[A4] @P:ZZ=80:FOR ZA=1,ZF:FOR ZB=B,F:POUT !(ZB,ZA),:NEXT:POUT:NEXT:@END
```

Prints out the redemption list.

```
[A5] @D:OUT "c",//////" WAIT A MOMENT ......":FOR ZA=6,ZB+7:FOR ZY=B,F:!(ZY,ZA)
=":NEXT:NEXT:@A
```

Deletes the data written on the front.

[Explanation of Other Fields]

```
BLK UH======E======------F------====G====----H----====I====----J----====K====--
   0:                                                        TOTAL
   1:L-PRINCIPAL REDEMPTION
   2:
   3:---------------------------------
   4:  INSTALLMENT      REMAINED
   5:---------------------------------
   6:
   7:
   8:
   9:
  10:
  11:
  12:
  13:
  14:
  15:
  16:
  17:
  18:
  19:
```

The total is input into !(J,0).

[Printing Example]

[REDEMPTION]...EQUAL-PRINCIPAL REDEMPTION

TIMES	LOANED	INTEREST	INSTALLMENT	REMAINED
1	83,333	85,000	168,333	916,667
2	83,333	77,917	161,250	833,334
3	83,333	70,833	154,166	750,001
4	83,333	63,750	147,083	666,668
5	83,333	56,667	140,000	583,335
6	83,333	49,583	132,916	500,002
7	83,333	42,500	125,933	416,669
8	83,333	35,417	118,750	333,336
9	83,333	28,334	111,667	250,003
10	83,333	21,250	104,583	166,670
11	83,333	14,167	97,500	83,337
12	83,337	7,084	90,421	0
TOTAL	1,000,000	552,502	1,552,502	

[Explanation of Formulas]

$$\frac{\text{The sum of principal and interest of the nth time}}{\text{The total amount of loan}} = \frac{A}{T} \times [\ 1 + (T-n+1) \times R\ ]$$

A ..... The original amount of loan

Tl ..... The number of installments a year

T ..... The total number of installments

Rl ..... The annual interest rate

R ..... The interest rate between installment (R=Rl-Tl)

n ..... The nth installment

(3) Directory

If you register names, zip codes, addresses, and telephone numbers, it will ① search by name, ② search by telephone number, and ③ prints out the address for direct mail.

Input the names into column B, zip codes into C, addresses (1) into D, address (2) into E and the telephone numbers into column F.

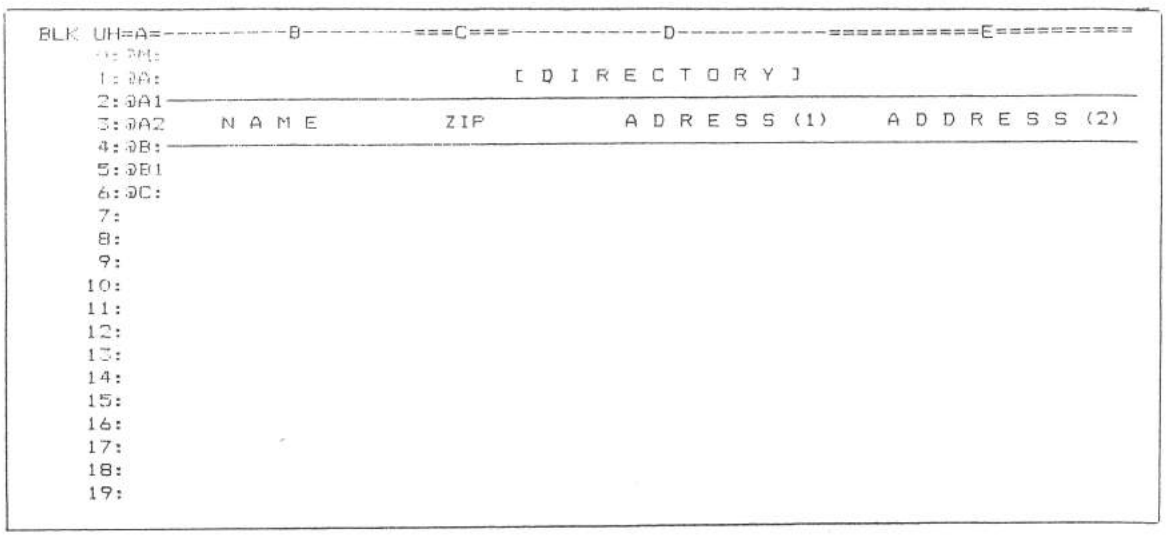
Input the zip codes and telephone numbers as character strings as well ([A] [DEL]).

Be sure to enter '#' into the row after the last section for names ([A] [DEL] [#]).

Move the cursor to !(A,0) and press keys [CTRL] and [P] at the same time. Seelct and execute one of them as the menu will be shown.

```
BLK UH=A=-----------B-----------===C===------------D-----------===========E==========
     0:@M:
     1:@A:                          [ D I R E C T O R Y ]
     2:@A1-------------------------------------------------------------------------------
     3:@A2   N A M E         ZIP            A D R E S S (1)    A D D R E S S (2)
     4:@B:-------------------------------------------------------------------------------
     5:@B1Mr.John Smith      14201 2568 Delaware Ave.         Buffalo,New York
     6:@C:Herr Johan Schmidt    20 Hauptstrabe 21             2000 Hamburg,
     7:   M. Jean Montand    65210 50,rue Anterrieur           Lyon
     8:   Mr.Taro Yamada       335 1-8-5 Midori-cho,          Urawa,Saitama
     9:   #
    10:
    11:
    12:
    13:
    14:
    15:
    16:
    17:
    18:
    19:
```

```
BLK UH=A=-----------B-----------===C===------------D-----------===========E==========
     0:
     1:                          [ D I R E C T O R Y ]
     2:-------------------------------------------------------------------------------
     3:  ZIP          A D R E S S (1)    A D D R E S S (2)          TELEPHONE
     4:-------------------------------------------------------------------------------
     5: 14201 2568 Delaware Ave.       Buffalo,New York            714-754-6809
     6:    20 Hauptstrabe 21           2000 Hamburg,                  089-777395
     7: 65210 50,rue Anterrieur         Lyon                          1-274.21.30
     8:   335 1-8-5 Midori-cho,        Urawa,Saitama               011-281-1251
     9:
    10:
    11:
    12:
    13:
    14:
    15:
    16:
    17:
    18:
    19:
```



[Explanation of the Macro-instruction]

```
[A0] @M:ZZ=79:ZY=5:%L:IF !(B,ZY)="#" THEN JP L1 STOP:ZY=ZY+1:JP L:%L1:OUT "c",/
///!(J,0),//!(J,1),/!(J,2),/!(J,3),/!(J,8),//!(J,9):INPUT ZA:IF ZA=1 THEN @A ST
OP:IF ZA=2 THEN @B STOP:IF ZA=3 THEN @C STOP:IF ZA=0 THEN @END STOP:JP L1
```

Counts to see how many rows of the data have been input. Select one and transfer the execution to the corresponding as the menu will be shown.

```
[A1] @A:OUT "c","///" INPUT NAME.....":INPUT $:ZS=S:%L:FOR ZA=ZS,ZY-1:IF !(B,ZA
)=?$ THEN @A2 STOP:NEXT:@A1
```

Searches an input name. Transfer the execution to @A2 if it is found but to @A1 if it cannot be found.

```
[A2] @A1:OUT "c",////$," UNREGISTERED NAME INPUTTED PLEASE PUSH RETURN KEY!":IN
PUT ZA:@M
```

It tells there is no such registration of a person as the one who has just been entered.

```
[A3] @A2:OUT "c",////!(J,10),///!(J,11),!(B,ZA),//!(J,12),!(C,ZA),//!(J,13),!(D
,ZA),//!(J,14),!(E,ZA),//!(J,15),!(F,ZA),///5"PUSH 1 TO CONTINUE,AND 0 TO QUIT"
:INPUT ZJ:IF ZJ=0 THEN @M STOP:ZS=ZA+1:IF ZA>ZY THEN @A1 ELSE @A%L STOP
```

Input the name, zip code, address, and telephone number of the person just searched to the screen.

```
[A4] @B:OUT "c",////" INPUT PHONE NUMBER.....":INPUT $:FOR ZA=5,ZY-1:IF !(F,ZA)
=$ THEN @A2 STOP:NEXT:@B1
```

Enter the telephone number to be searched.

Searches the input phone number. Transfer the execution to @A2 and to @B1 if it cannot be found.

```
[A5] @B1:OUT "c",////$," UNREGISTERED NUMBER INPUTTED PLEASE PUSH RETURN KEY":I
NPUT ZA:@M
```

It tells there is no such registration of a telephone number which has been entered.

```
[A6] @C:FOR ZA=5,ZY-1:$=!(E,ZA):POUT ///!(B,ZA),//!(D,ZA),/$,!(C,ZA),//:NEXT:@M
```

Print-out will be done on the address in the order of registration.

They can be sorted out with keys CTRL + W and be printed out in alphabetical order.

[Explanation on the other fields]

```
BLK UH-------F-------====G========----H----====I====-----------------J-----------------===
   0:                                                          [ M E N U ]
   1:                                                    1. SEARCH FROM NAME
   2:-----------------                                   2. SEARCH FROM PHONE NUMBER
   3:       TELEPHONE                                    3. DIRECT MAIL PRINTING
   4:-----------------
   5:   714-754-6809
   6:     089-777395
   7:    1-274.21.30
   8:   011-281-1251                                     0. END
   9:                                                       PUSH     0-3
  10:                                                       [ CUSTOMERS LIST ]
  11:                                                    1. NAME ...........
  12:                                                    2. ZIP CODE .......
  13:                                                    3. ADDRESS (1) ....
  14:                                                    4. ADDRESS (2) ....
  15:                                                    5. PHONE NUMBER ...
  16:
  17:
  18:
  19:
```

The menu and other data are input in column J.

[Printing Example]

- Direct Mail -

Mr.John Smith

2568 Delaware Ave.
Buffalo,New York 14201

Herr Johan Schmidt

Hauptstrabe 21
2000 Hamburg, 20

M. Jean Montand

50,rue Anterrieur
Lyon 65210

Mr.Taro Yamada

1-8-5 Midori-cho,
Urawa,Saitama 335

[Applications]

This application can be expanded to a customer-control program.

3. Business

(1) Analyses of Profit and Loss

The last of the applications discussed here is for business. This problem is a little more difficult since it also works on a table of B/E (break-even point) and a sales achievement chart, however, you will fully understand that your SC-3000 can also be used for business use when used together with HuCAL. Bring out some new ideas and let's make the most of them.

```
BLK UH==========A==========----B---====C====----D---====E===----F---====G====
    0:
    1:                              [ PROFIT  & LOSS  ITEM ]
    2:-------------------------------------------------------------------------
    3: ITEM NAME                  AAAA     BBBB     CCCC     DDDD     EEEE     TOTAL
    4:-------------------------------------------------------------------------
    5:SALES                     75,000  120,000  100,000   56,000   75,000   426,000
    6:UNIT PRICE                   150      150      200    1,000      750
    7:AMOUNT                       500      800      500       56      100
    8:VARIABLES                 33,750   60,000   55,000   33,600   26,250   208,600
    9:VARIABLE RATE                 45       50       55       60       35        49
   10:LIMITATIVE PROFIT RATE        55       50       45       40       65        51
   11:LIMITATIVE PROFIT         41,250   60,000   45,000   22,400   48,750   217,400
   12:FIXED EXPENSE                                                         100,000
   13:-------------------------------------------------------------------------
   14:PROFIT                                                                117,400
   15:-------------------------------------------------------------------------
   16:
   17:SALES CAPACITY          = 900,000 EXPECTED PROFIT= 200,000
   18:PROFIT AND LOSS BORDER= 195,952 WORKING  RATE  =      22
   19:SAFTY RATE              =      54 EXPECTED SALES = 587,856
```

When item names, unit prices, quantity, variable costs rate, fixed costs, sales capacity, target profit are input, and then the B/E is calculated and a profit and loss statement by section is prepared.

Input item names to row 3, unit prices to 6, quantity to 7, variable costs rate to 9, fixed costs to !(G,12), sales capacity to !(B,17) and target profit to the field (E,17).

It will be calculated when keys [CTRL] and [Z] are pressed simultaneously.

Try and simulate your profit plans by inputting various data.

```
BLK UH==========A==========a----B---====C====----D---====E====----F---====G====
    0:
    1:                              [ PROFIT  & LOSS  ITEM ]
    2:-------------------------------------------------------------------------
    3: ITEM NAME                  AAAA     BBBB     CCCC     DDDD     EEEE     TOTAL
    4:-------------------------------------------------------------------------
    5:SALES                          0        0        0        0        0         0
    6:UNIT PRICE
    7:AMOUNT
    8:VARIABLES                      0        0        0        0        0         0
    9:VARIABLE RATE                                                        %%%%%%%%%
   10:LIMITATIVE PROFIT RATE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   11:LIMITATIVE PROFIT             0        0        0        0        0         0
   12:FIXED EXPENSE
   13:-------------------------------------------------------------------------
   14:PROFIT                                                                     0
   15:-------------------------------------------------------------------------
   16:
   17:SALES CAPACITY          =          EXPECTED PROFIT=
   18:PROFIT AND LOSS BORDER=          0 WORKING  RATE  =%%%%%%%%%
   19:SAFTY RATE             =%%%%%%%%% EXPECTED SALES =          0
```

[Input Example]

```
BLK UH===========A============----B---====C====----D---====E===----F---====G====
    0:
    1:                                  [ PROFIT  & LOSS  ITEM ]
    2:--------------------------------------------------------------------------
    3: ITEM NAME                  AAAA    BBBB    CCCC    DDDD    EEEE    TOTAL
    4:--------------------------------------------------------------------------
    5:SALES                          0       0       0       0       0        0
    6:UNIT PRICE                   150     150     200   1,000     750
    7:AMOUNT                       500     800     500      56     100
    8:VARIABLES                      0       0       0       0       0        0
    9:VARIABLE RATE                 45      50      55      60     35%%%%%%%%%
   10:LIMITATIVE PROFIT RATE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   11:LIMITATIVE PROFIT              0       0       0       0       0        0
   12:FIXED EXPENSE                                                     100,000
   13:--------------------------------------------------------------------------
   14:PROFIT                                                                  0
   15:--------------------------------------------------------------------------
   16:
   17:SALES CAPACITY        = 900,000 EXPECTED PROFIT= 200,000
   18:PROFIT AND LOSS BORDER=       0 WORKING  RATE  =%%%%%%%%
   19:SAFTY RATE            =%%%%%%%% EXPECTED SALES =        0
```

[Formulas]

Location of the cursor

! (B, 5)	! (B, 6) *! (B, 7)
! (C, 5)	! (C, 6) *! (C, 7)
! (D, 5)	! (D, 6) *! (D, 7)
! (E, 5)	! (E, 6) *! (E, 7)
! (F, 5)	! (F, 6) *! (F, 7)
! (B, 8)	! (B, 5) *! (B, 9) /100
! (C, 8)	! (C, 5) *! (C, 9) /100
! (D, 8)	! (D, 5) *! (D, 9) /100
! (E, 8)	! (E, 5) *! (E, 9) /100
! (F, 8)	! (F, 5) *! (F, 9) /100
! (B, 10)	(! (B, 5) -! (B, 8)) /! (B, 5) *100
! (C, 10)	(! (C, 5) -! (C, 8)) /! (C, 5) *100
! (D, 10)	(! (D, 5) -! (D, 8)) /! (D, 5) *100
! (E, 10)	(! (E, 5) -! (E, 8)) /! (E, 5) *100
! (F, 10)	(! (F, 5) -! (F, 8)) /! (F, 5) *100
! (B, 11)	! (B, 5) -! (B, 8)
! (C, 11)	! (C, 5) -! (C, 8)
! (D, 11)	! (D, 5) -! (D, 8)
! (E, 11)	! (E, 5) -! (E, 8)
! (F, 11)	! (F, 5) -! (F, 8)
! (G, 5)	SUM (B, 5, F, 5)
! (G, 8)	SUM (B, 8, F, 8)
! (G, 9)	! (G, 8) /! (G, 5) *100
! (G, 10)	(! (G, 5) -! (G, 8)) /! (G, 5) *100
! (G, 11)	SUM (B, 11, F, 11)
! (G, 14)	! (G, 11) -! (G, 12)
! (B, 18)	! (G, 12) /! (G, 10) *100
! (B, 19)	(! (G, 5) -! (B, 18)) /! (G, 5) *100
! (E, 18)	! (B, 18) /! (B, 17) *100
! (E, 19)	(! (G, 12) +! (E, 17)) /! (G, 10) *100

[Explanation of Formulas]

$$\text{Sales amount at break-even point} = \frac{\text{Fixed costs}}{\text{Marginal profits rate}}$$

$$\text{Capacity factor at break-even point} = \frac{\text{Sales amount at break-even point}}{\text{Sales amount}} \times 100$$

$$\text{Safety rate} = \frac{(\text{Sales amount} - \text{Sales amount at break-even point})}{\text{Sales amount}} \times 100$$

$$\text{Target sales amount} = \frac{(\text{fixed costs} + \text{target profit})}{\text{Marginal profits rate}}$$

[Explanation of terms]

Break-even point ..... It is a sales level at which the amount of revenue and that of expense are equal for a company.

Variable costs rate ..... The rate of variable costs to sales amount.

Marginal profits ..... The amount which is calculated by subtracting variable costs from sales amount.

Sales capacity ..... Sales amount at 100% of capacity factor.

Safety rate ..... It is calculated as [(sales amount) -(sales amount at break-even point)] /(sales amount)(%)
The greater the safety rate is, the higher the power of revenue in the company may be and the greater the profit may be.

Target sales amount ..... The sales amount in order to achieve the target profit.

(2) Sales Achievement Chart

```
BLK UH=A=------B-----=======C=========------D------==E==----------F----------
    0:@M:
    1:@A:                              [ ACHIVEMENT ]
    2:@B:-------------------------------------------------------------------------
    3:@P:NAME                    EXPECTED      SHOWINGS RATE    30   60   90  120  150
    4:   ---------------------------------------------------------+----+----+----+----+
    5:   LONDON    BR.        1,004,000       897,500   89 ***************
    6:   PARIS     BR.        1,437,000     1,664,732  116 ********************
    7:   MUNICH    BR.        1,296,500       965,430   74 ************
    8:   CHICAGO   BR.          919,000       780,650   85 **************
    9:   TOKYO     BR.          873,000     1,234,500  141 *************************
   10:   SANTIAGO  BR.          962,000       765,000   80 *************
   11:   NAIROBI   BR.        1,041,000     1,400,000  134 ***********************
   12:   SYDNEY    BR.          998,000       850,000   85 **************
   13:   #                    8,530,500     8,557,812
   14:
   15:
   16:
   17:
   18:
   19:
```

When item names, target sales amount, and actual sales amuont are input, the achievement rates will be calculated and shown in a graph. The totals will be shown in the row with the # symbol.

Input the item names into column B, target sales amount into C, and the actual sales amount into column D and be sure to enter "#" into the row after the last item name. ([A] [DEL] [#]). This program will be executed when the cursor is moved to !(A,0) and keys [CTRL] and [P] are pressed simultaneously.

[Explanation of the Macro-instruction]

```
[A0] @M:ZZ=79:OUT"c",//////" WAIT A MOMENT .....":ZA=5:%L:IF !(B,ZA)="#" THEN @
A STOP :ZA=ZA+1:JP L
```

Counts how many rows of the data are input.

```
[A1] @A:!(C,ZA)=SUM(C,5,C,ZA-1):!(D,ZA)=SUM(D,5,D,ZA-1):@B
```

Obtains the total of target sales amount and actual sales amount, respectively.

```
[A2] @B:FOR ZB=5,ZA-1:!(E,ZB)=INT(!(D,ZB)/!(C,ZB)*100+.5):ZC=INT(!(E,ZB)/6+.5):
$=!(G,ZC):!(F,ZB)=$:NEXT:@END
```

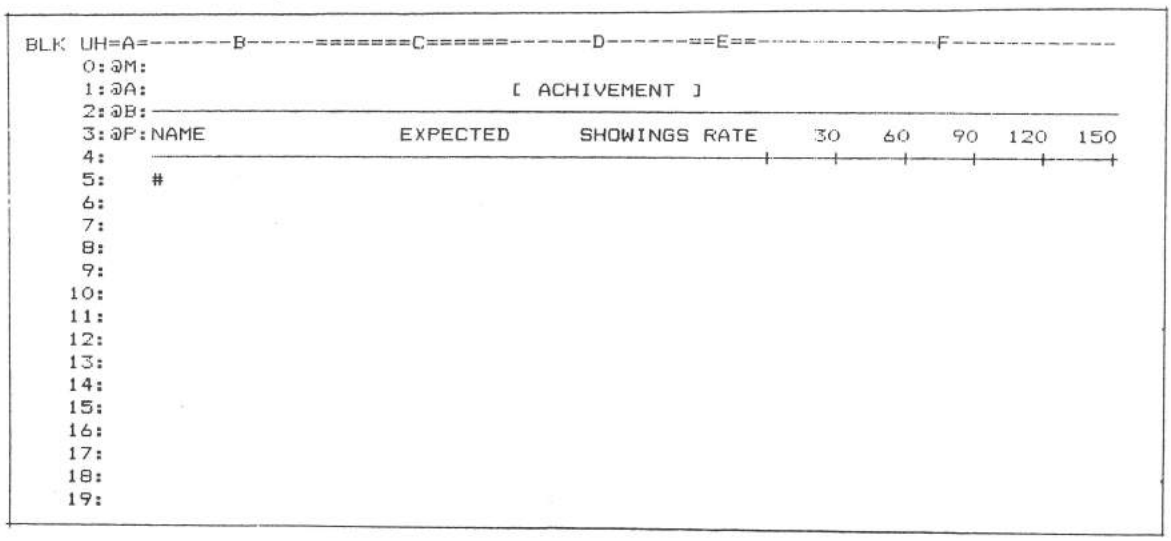
Writes in the achievement rates and draws graph.

```
[A3] @P:ZZ=80:FOR ZB=1,ZA:FOR ZA=B,F:POUT !(ZA,ZB),:NEXT:POUT:NEXT:@END
```

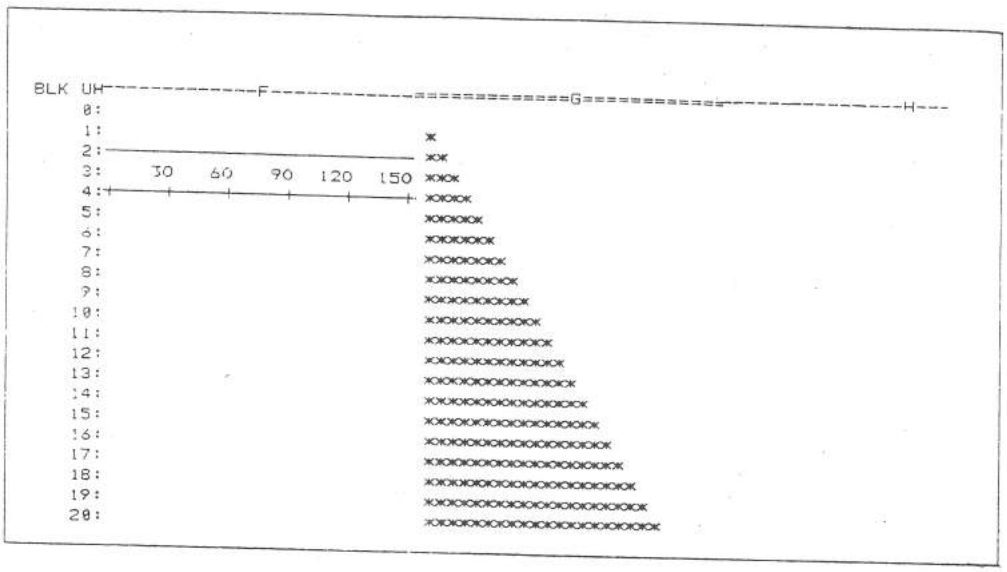
Prints out the achievement chart.

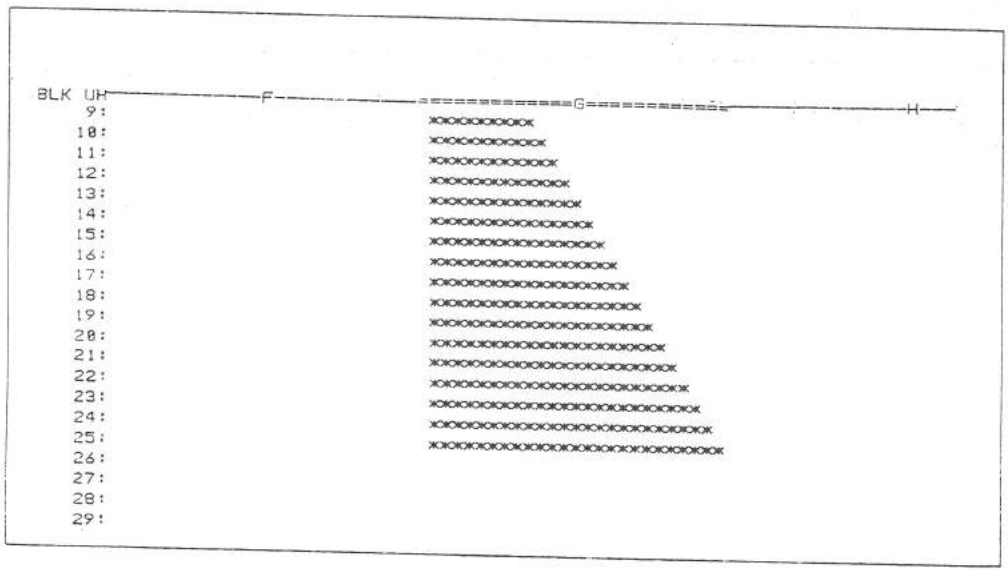
Note: The inputting of the macro-instruction will start when [O] is pressed while [CTRL] is being pressed together and be completed with the [RETURN] key. That instruction will be executed when [P] is pressed while [CTRL] is being pressed together on the field where the macro-instruction is input.

Execute @P of !(A,3) when wishing to output the sales achievement chart to the printer.



[Explanation of other fields]





Graph patterns are input in column G, [A] [DEL] [SPACE] is entered at !(G,0) to make it into a character string, so it is not a blank.

The number of graphic characters from !(G,1) through !(G,25) corresponds to each row number.

Each field starts a graph bar followed by a blank by one character, so press [A] [DEL] [SPACE] once before inputting any graphic characters.

[Printing Example]

[ACHIVEMENT]

NAME		EXPECTED	SHOWINGS	RATE	30 60 90 120 150
LONDON	BR.	1,004,000	897,500	89	**********
PARIS	BR.	1,437,000	1,664,732	116	**********
MUNICH	BR.	1,296,500	965,430	74	**********
CHICAGO	BR.	919,000	780,650	85	**********
TOKYO	BR.	873,000	1,234,500	141	**********
SANTIAGO	BR.	962,000	765,000	80	**********
NAIROBI	BR.	1,041,000	1,400,000	134	**********
SYDNEY	BR.	998,000	850,000	85	**********
#		8,530,500	8,557,812		

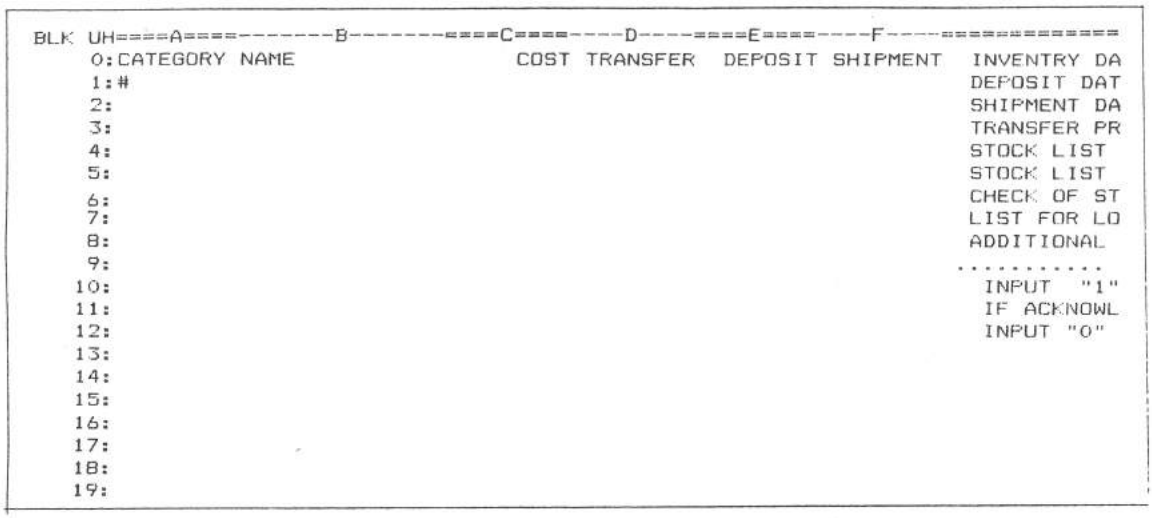
(3) Stock Management

```
BLK UH====A====-------B-------====C====----D----====E====----F----============
    0:CATEGORY NAME                  COST  TRANSFER   DEPOSIT  SHIPMENT   INVENTRY DA
    1:COM      MZ-1500            196,000         3         5         5   DEPOSIT DAT
    2:COM      CZ-800D            113,000         6         2         3   SHIPMENT DA
    3:COM      MZ-80B             268,000         7         4         6   TRANSFER PR
    4:COM      MZ-2000            218,000         4         2         1   STOCK LIST
    5:COM      MZ-700             128,000        12         3         9   STOCK LIST
    6:COM      MZ-1200            148,000        10         3         8   CHECK OF ST
    7:COM      MW-6100            400,000        10         3        10   LIST FOR LO
    8:#                                                                   ADDITIONAL
    9:                                                                    ...........
   10:                                                                     INPUT  "1"
   11:                                                                     IF ACKNOWL
   12:                                                                     INPUT "0"
   13:
   14:
   15:
   16:
   17:
   18:
   19:
```

Inputs each item's category, product name, prime cost, quantity of transfers, quantity of deposits, and quantity of shipments.

It can print out a list of inventory, a list of inventory by category and a list of lower limit inventory, outputting quantities of inventory, deposit and shipment for each product to the screen and renewal of inventory.

The "menu" will be displayed when the cursor is moved to !(H,0) and keys CTRL and P are pressed. Select and execute one of them.



[Explanation of the Macro-instruction]

```
[H0] @M:%L:ZZ=79:OUT "c",////4"[ STOCK MANAGEMENT ]"//,!(G,9)//,!(G,0)/,!(G,1)/
,!(G,2)/,!(G,3)/,!(G,4)/,!(G,5)/,!(G,6)/,!(G,7)/,!(G,8)//,!(G,11)//,!(G,12):INP
UT ZA:IF ZA=0 THEN @END STOP:IF ZA<1;ZA>9 THEN JP L STOP:ZX=A,ZY=1:@M1
```

Select one as the menu will be displayed.

```
[I0] @M1:IF ZA=1 THEN @A STOP:IF ZA=2 THEN @B STOP:IF ZA=3 THEN @C STOP:IF ZA=4
 THEN @D STOP:IF ZA=5 THEN @E STOP:IF ZA=6 THEN @F STOP:IF ZA=7 THEN @G STOP:IF
 ZA=8 THEN @H ELSE @I STOP
```

Moves the execution to the corresponding field.

```
[H1] @A:%L1:OUT "c",////!(ZX,0)," ",ZY//,"# TO RETURN TO MENU ":INPUT $:!(ZX,ZY
)=$:IF !(ZX,ZY)="#",ZX=A THEN @M STOP:%L2:ZX=ZX+1:IF ZX<C THEN JP L1 STOP:OUT "
c",////!(ZX,0)," ",ZY:INPUT ZA:!(ZX,ZY)=ZA:IF ZX<D THEN JP L2 STOP:ZX=A,ZY=ZY+1
:JP L1
```

Inputs the categories, product names, prime costs, and the quantity of transfers.

```
[H2] @B:ZY=1,ZZ=79:OUT "c",////"DEPOSIT PRODUCT NAME?",//"# TO RETURN TO MENU":
INPUT $: IF $="#" THEN @M ELSE @B0 STOP
```

Inputs the names of the deposits.

```
[H3] @B0:%L1:IF !(B,ZY)=$ THEN @B1 STOP:IF !(A,ZY)="#" THEN @B STOP:ZY=ZY+1:JP
L1
```

Searches the input product names.

```
[H4] @B1:ZZ=79:OUT "c",////!(A,0),!(B,0),!(C,0)/,!(A,ZY),!(B,ZY),!(C,ZY)//,!(G,
10):INPUT ZA:IF ZA=1 THEN @B2 STOP:ZY=ZY+1:@B0
```

Displays the data of the input items and confirms whether any of the input items should be corrected.

```
[H5] @B2:ZH=!(D,ZY)+!(E,ZY)-!(F,ZY):!(L,0)=ZH:OUT "c",////"INPUT NUMBER OF STOC
KING"//,!(A,0),!(B,0)/,!(A,ZY),!(B,ZY)//,!(C,0),!(D,0)/,!(C,ZY),!(L,0):INPUT ZA
:ZH=!(E,ZY)+ZA:!(E,ZY)=ZH:@B
```

Inputs the quantity of deposits.

```
[H6] @C:ZY=1:OUT "c",////"SHIPPING PRODUCT NAME?"//,"# TO RETURN TO MENU":INPUT
 $:IF $="#" THEN @M ELSE @C0 STOP
```

Inputs the product names of shipment.

```
[H7] @C0:%L1:IF !(B,ZY)=$ THEN @C1 STOP:IF !(A,ZY)="#" THEN @C STOP:ZY=ZY+1:JP
L1
```

Searches the input product names.

```
[I1] @C1:ZZ=79:OUT "c",////!(A,0),!(B,0),!(C,0)/,!(A,ZY),!(B,ZY),!(C,ZY)//,!(G,
10):INPUT ZA:IF ZA=1 THEN @C2 STOP:ZY=ZY+1:@C0
```

Displays the data of the input items and confirms whether any of the input items should be corrected.

```
[I2] @C2:ZK=!(D,ZY)+!(E,ZY)-!(F,ZY):!(L,0)=ZK:OUT "c",////"INPUT NUMBER OF STOC
KING"//,!(A,0),!(B,0)/,!(A,ZY),!(B,ZY)//,!(C,0),!(D,0)/,!(C,ZY),!(L,0):INPUT ZA
:ZH=!(F,ZY)+ZA:!(F,ZY)=ZH:@C
```

Inputs the quantity of shipment.

```
[I3] @D:ZY=1:OUT "c",//////" WAIT A MOMENT PLEASE ...":%L:IF !(A,ZY)="#" THEN @
M STOP:ZH=!(D,ZY)+!(E,ZY)-!(F,ZY),!(D,ZY)=ZH,!(E,ZY)=":!(F,ZY)=":ZY=ZY+1:JP L
```

Treats the transfers.

```
[I4] @E:ZS=0,ZY=1:%L:ZM=ZY-INT(ZY/50)*50:IF !(A,ZY)="#" THEN @E1 STOP:IF ZM=1 T
HEN POUT "c",!(A,0),!(B,0),!(C,0),!(D,0),!(E,0),!(F,0),!(K,1),!(L,1) STOP:@E0
```

```
[I5] @E0:ZK=!(D,ZY)+!(E,ZY)-!(F,ZY):!(K,0)=ZK:ZK=ZK*!(C,ZY):!(L,0)=ZK:POUT !(A,
ZY),!(B,ZY),!(C,ZY),!(D,ZY),!(E,ZY),!(F,ZY),!(K,0),!(L,0):ZS=ZS+ZK,ZY=ZY+1:@E%L
```

Prints out the inventory list of all the items. 50 items can be printed out on a page.

```
[I6] @E1:POUT //33"TOTAL AMOUNT IN STOCK"/,33ZS:@M
```

Prints out the total worth of all the products in stock.

```
[I7] @F:ZS=0,ZC=0,ZY=1:OUT "c",////"INPUT CATEGORY NAME"//,"# TO RETURN TO MENU
":INPUT $:IF $="#" THEN @M ELSE POUT "c","CATEGORY NAME = ",$/:@F0 STOP
```

```
[J0] @F0:ZM=ZC-INT(ZC/50)*50:%L:IF !(A,ZY)="#" THEN @E1 STOP:IF !(A,ZY)=$ THEN
ZK=!(D,ZY)+!(E,ZY)-!(F,ZY):!(K,0)=ZK:ZK=ZK*!(C,ZY):!(L,0)=ZK:@F1 STOP:ZY=ZY+1:J
P L
```

```
[J1] @F1:IF ZM=0 THEN POUT !(B,0),!(C,0),!(D,0),!(E,0),!(F,0),!(K,1),!(L,1) STO
P:POUT !(B,ZY),!(C,ZY),!(D,ZY),!(E,ZY),!(F,ZY),!(K,0),!(L,0):ZC=ZC+1,ZS=ZS+ZK,Z
Y=ZY+1:@F0
```

Prints out the inventory list by category which is input.

```
[J2] @G:ZY=1:OUT "c",////"NAME OF PRODUCT TO BE CHECKED?"//,"# TO RETURN TO MEN
U":INPUT $:IF $="#" THEN @M ELSE @G0 STOP
```

Inputs the product names to be searched.

```
[J3] @G0:%L:IF !(B,ZY)=$ THEN @G1 STOP:IF !(A,ZY)="#" THEN @G STOP:ZY=ZY+1:JP L
```

Searches the input product names.

```
[J4] @G1:ZZ=79:ZS=!(D,ZY)+!(E,ZY)-!(F,ZY):OUT "c",////!(A,0),!(B,0),!(C,0)/,!(A
,ZY),!(B,ZY),!(C,ZY)//,!(D,0),ZS//,"INPUT '1' IF ACKNOWLEDGED":INPUT ZA:IF ZA=1
 THEN @G ELSE ZY=ZY+1:@G0 STOP
```

Outputs the searched stock data onto the screen.

```
[J5] @H:ZC=0,ZY=1:OUT "c",////"INPUT LOWER LIMIT OF STOCK "/,"INPUT '-1' TO RET
URN TO MENU":INPUT ZL:IF ZL=-1 THEN @M ELSE @HO STOP
```

```
[J6] @HO:ZM=ZC-INT(ZC/50)*50:%L:IF !(A,ZY)="#" THEN @M STOP:ZS=!(D,ZY)+!(E,ZY)-
!(F,ZY):IF ZS<=ZL THEN @H1 STOP:ZY=ZY+1:JP L
```

```
[J7] @H1:!(K,0)=ZS:IF ZM=0 THEN POUT "c",!(A,0),!(B,0),!(C,0),!(D,0),!(E,0),!(F
,0),!(K,1) STOP:POUT !(A,ZY),!(B,ZY),!(C,ZY),!(D,ZY),!(E,ZY),!(F,ZY),!(K,0):ZY=
ZY+1,ZC=ZC+1:@HO
```

Prints out the items whose quantity is less than the input quantity of inventory.

50 items can be printed out on a page.

```
[J8] @I:%L:IF !(ZX,ZY)="#" THEN @A STOP:ZY=ZY+1:JP L
```

Determines the row numbers on registering additional items.

[Explanation of other fields]

```
BLK UH===============G================----H---====I=======---J---===K===-----L----
    0: INVENTRY DATA INPUT..........1@M:%L:ZZ@M1:IF ZA@F0:ZM=Z      3 1,200,000
    1: DEPOSIT DATA INPUT...........2@A:%L1:0@C1:ZZ=79@F1:IF Z   REST      TOTAL
    2: SHIPMENT DATA INPUT..........3@B:ZY=1,@C2:ZK=!(@G:ZY=1:
    3: TRANSFER PROCESS OF DATA.....4@B0:%L1:@D:ZY=1:0@G0:%L:I
    4: STOCK LIST FOR ALL PRODUCTS..5@B1:ZZ=7@E:ZS=0,Z@G1:ZZ=7
    5: STOCK LIST BY CATEGORY.......6@B2:ZK=!@E0:ZK=!(@H:ZC=0,
    6: CHECK OF STOCK...............7@C:ZY=1:@E1:POUT @H0:ZM=Z
    7: LIST FOR LOWER LIMIT PRODUCT.8@C0:%L1:@F:ZS=0,Z@H1:!(K,
    8: ADDITIONAL ENTRY OF PRODUCT..9                  @I:%L:IF
    9:........... M E N U ...........
   10:  INPUT '1'  IF ACKNOWLEDGED
   11:     INPUT NUMBER (0...9)
   12:       INPUT '0' TO END
   13:
   14:
   15:
   16:
   17:
   18:
   19:
```

The menu and others are input in column G.

[Printing Example]

Inventory list of all the products

CATEGORY	NAME	COST	TRANSFER	DEPOSIT	SHIPMENT	REST	TOTAL
COM	MZ-1500	196,000	3	5	5	3	588,000
COM	CZ-800D	113,000	6	2	3	5	565,000
COM	MZ-80B	268,000	7	4	6	5	1,340,000
COM	MZ-2000	218,000	4	2	1	5	1,090,000
COM	MZ-700	128,000	12	3	9	6	768,000
COM	MZ-1200	148,000	10	3	8	5	740,000
COM	MW-6100	400,000	10	3	10	3	1,200,000

TOTAL AMOUNT IN STOCK
6,291,000

Inventory list by category

CATEGORY	NAME	COST	TRANSFER	DEPOSIT	SHIPMENT	REST	TOTAL
COM	MZ-1500	196,000	3	5	5	3	588,000
COM	CZ-800D	113,000	6	2	3	5	565,000
COM	MZ-80B	268,000	7	4	6	5	1,340,000
COM	MZ-2000	218,000	4	2	1	5	1,090,000
COM	MZ-700	128,000	12	3	9	6	768,000
COM	MZ-1200	148,000	10	3	8	5	740,000
COM	MW-6100	400,000	10	3	10	3	1,200,000

TOTAL AMOUNT IN STOCK
6,291,000

Inventory list of lower limit products

(fewer than 5 in stock)

CATEGORY	NAME	COST	TRANSFER	DEPOSIT	SHIPMENT	REST	TOTAL
COM	MZ-1500	196,000	3	5	5	3	588,000
COM	CZ-800D	113,000	6	2	3	5	565,000
COM	MZ-80B	268,000	7	4	6	5	1,340,000
COM	MZ-2000	218,000	4	2	1	5	1,090,000
COM	MZ-700	128,000	12	3	9	6	768,000
COM	MZ-1200	148,000	10	3	8	5	740,000
COM	MW-6100	400,000	10	3	10	3	1,200,000

Epilogue

Do you thoroughly understand how to use HuCAL?

Apart from Part I "INTRODUCTION", we are afraid there may be someone who got stuck in the latter half since Part II "APPLICATIONS" was a little too difficult.

That is to say, the use of HuCAL itself should be really simple. It should be so simple that average people can get to know everything in a week and much smarter ones take only a few days, however, when it comes to formulating a program, it's not as easy to understand the flow of the program as you may think.

So the crucial point in this, as it may seem, is to put the flowchart of any process securely in your mind first, before moving on to the actual programming.

In order to do that, it may be helpful to learn how to make a flowchart.

With HuCAL, programming itself is easy enough for anyone to follow.

The point is to clarify what you want HuCAL to do for you, so jot down on a paper what job and by what flow you want HuCAL to work, before moving onto the process of programming.

For example, the following procedure is one of those that was obtained in this way:

1) What kind of chart to be output ultimately?
   And how many kinds?
2) What data is needed for outputting the chart mentioned in 1)?
3) Outlook of collecting the basic data for it?
4) How to settle the format of the basic data in order to output the aimed chart most efficiently.
5) Collection of the basic data to be input.
6) Setting of the input items for the basic data to be input to the chart of HuCAL → Inputting the basic data → Structuralizing the master data.
7) Programming
8) Performance
9) Output
10) Check on input or calculation errors
11) Recalculation
12) Output

The reason why we do it this way is because simplified languages in general, including HuCAL, have the procedures starting with constituting data base by inputting certain data through keys and continuing with obtaining the target

chart (result of performance) by carrying out sorting, searching, and calculations on the basis of the data base (master data).

Therefore, you will be running into a roundabout way where you face a lot of trouble until you can get the performance result to meet your aim, unless you carefully figure out what may happen while setting up a format, when producing the primary data base.

Not only the HuCAL but all other simplified languages are deeply dependent upon the situation of how primary data bases are made, so be extraordinarily careful of its composition when using the keyboard. If you take it too easy at the beginning, it will be 2 or 3 times as hard.

Well, it's needless to say that to those people who have just studied this text through the very end, isn't it?

At any rate, you must have thoroughly understood the use of HuCAL, so it's all up to you from now on to bring out your own ideas and make original programs.

We believe any member of the family can make use of it, such as studies and hobbies of the children, housekeeping work of the mother, or business of the father. And we bet it will bring success each time.

If there is anything that is not still clear, just go on with your further study to reach the stage of perfection. It will bring you a more comfortable life if you can become an expert on the simplified languages.

Published by Sega Enterprises, Ltd., Japan

Printed in Japan

If you have any questions concerning this product, please contact the store that you bought it from or SEGA's distributor in your area.

HuCAL

utilizing "no-programing" computer language for the most advanced business use

Main characteristics:

- Equipped with a window function.
- 14 digit highly accurate calculations are possible.
- The number of digits after the decimal point can be specified for each column.
- Commas for numerals can be specified for each column.
- Editing functions by the "full screen editor" are provided.
- Data search function is provided.
- Data sort function is available.
- Given a calculation formula, repeated processing by FOR-NEXT is possible.
- Hard copy can be produced by such COMMANDs as HDCP! (A, 10) or HDCP! (X, 100), 230, etc.
- Data protection for each horizontal row is possible.
- Macrocosm COMMAND syntax error points can be located by the CURSOR.
- Sums (totals) and mean values that are used especially in businesses, etc., can be calculated easily by utilizing the internal mathematical functions.
- The spread sheet can be as large as 255 cells (horizontal) x 1,000 cells (vertical), i.e., up to the memory limit.
- The macrocosm COMMANDs can be printed out by a one-touch operation.
- When using macrocosm COMMANDs such as INPUT ZA, etc., a formula which includes mathematical functions can be entered.

SEGA ENTERPRISES, LTD.