UTILITY
USERS MANUAL

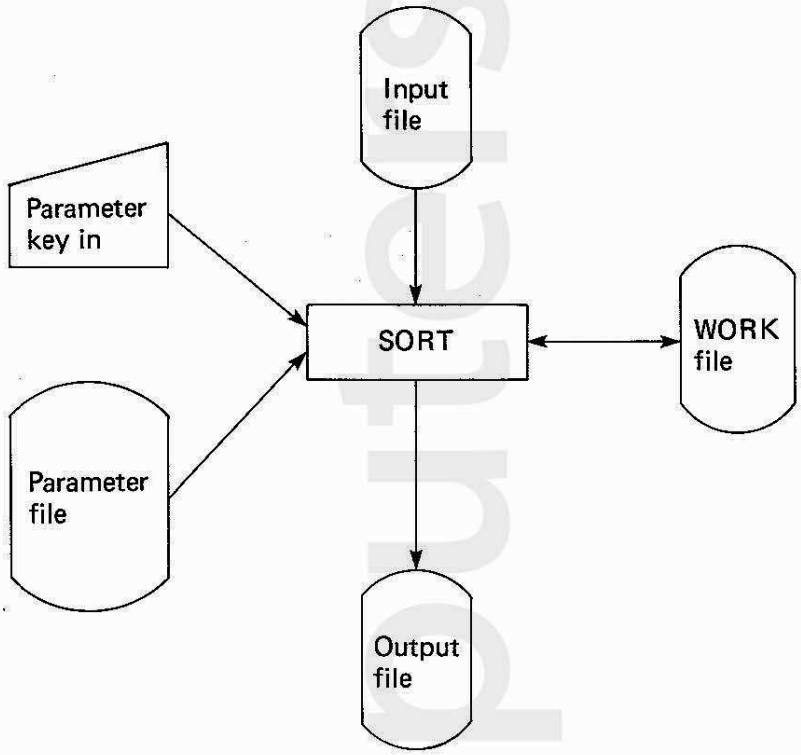
SANYO

Table of Contents

1. SORT

1.1 Functions of SORT

SORT is a utility that compares each of the records of a file containing fixed length records against the specified key items and arranges these records in the specified order (ascending or descending).



The SORT utility first extracts only the key items specified as the sort key from the records, and orders them in the work area in the main memory. If the number or records is too large to make the available work area insufficient, it uses the WORK file. In the latter case, the following vacant space must be available:

WORK file area = (Length of key + 2) x Number of records + Input file size + INT (m x number of records ÷ n)

where

n = INT (128 – (length of key + 2)), and
m = 128 – n x (length of key + 2).

When the number of records so selected is large, many work files with the same name as that of the input file but with different extended identifiers ranging from AOA to ZOZ are made. All these work files will be purged at the end of sorting and hence care should be taken before sorting that any file in the disk with the same name as that of these work files is assigned a different name if that file has to be retained even after sorting is completed. The SORT utility provides the functions of all records sort and sampling sort.

1.2 Limitations of SORT

The following are the restrictions on using the SORT utility.

1. The maximum length of one record is 384 bytes.
2. The maximum number of items in any one record is 20.
3. A maximum of five sort key items are permitted. The total length of the key items is limited to 124 bytes.
4. The maximum number of records in one file is restricted to 65,534.
5. The EOF symbol (1AH) should not be present at the beginning of any record.
6. In a random organization file no record should be without a serial number.
7. The end of a file should be indicated by the EOF symbol (1AH).

In SBASIC, although the EOF symbol is added automatically at the end of a file with sequential organization, the user program will have to write the EOF symbol at the end of a randomly organized file. The following is an example of how the EOF symbol is written at the end of a randomly organized file by a user program.

```
1. OPEN "R", #1, "DATA", 10          :' RANDOM FILE OPEN
2. FIELD #1, 5 AS A$, 5AS B$
3. ----------------
4. ----------------
   ----------------
   ----------------
   ----------------
10. LSET A$ = CHR$ (&H1A)            :' EOF MARK SET
11. PUT#1, N                         :' EOF MARK WRITE
12. CLOSE                            Note: N is equal to the last record
                                           number plus 1.
```

In the case of a sequentially organized file, make the data records have a fixed length by using the clauses such as PRINT USING, etc.

1.3 Method of Use

The method of calling the SORT Utility differs depending on whether the required parameters are input directly by typing in or by using an already prepared parameter file as shown below.

1. When typing in the parameters ..... A>SORT
2. When using a parameter file ..... A>SORT parameter file name

1.3.1 Parameter Input Format

(1) INPUT FNAME ..... Specifies the name of the input file to be sorted.

(2) FILE MODE ..... Specifies the mode of the input file (S or R).

S: Sequential organization file (A file with the separator codes CR and LF at the end of each record).

R: Random organization file.

(3) RECORD FORMAT ..... Specifies the format (number of bytes) of each record in the input file. In the case of sequential organization files the number of bytes specified here should exclude the CR code and the LF code.

n .... Character, n bytes.

B_n .. B2 – integer, 2 bytes (corresponds to MKI$)

B4 – single precision real, 4 bytes (MKS$)

B8 – double precision real, 8 bytes (MKD$)

(4) SAMPLING DATA ..... Specifies the records of the input file that are to be sorted.

* – All records of the input file are to be sorted.

Conditional expression – sampling sort is carried out according to the following rules.

1. ["xxx"<(=)]In[<(=)"yyy"] [x<(=)]In[<(=)y]
2. In="xxx"[,"yyy"] In=x[,y]
3. In< >"xxx"[,"yyy"] In=< >x[,y]

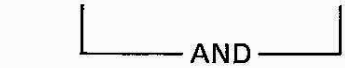
i) Upto a maximum of three conditional expressions are allowed.

ii) The different conditions should be separated from each other by a semicolon (;) when their logical product (AND) is to be obtained, and by a colon (:) if their logical sum (OR) is to be obtained.

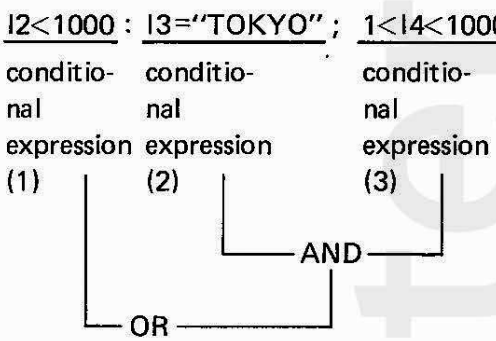
iii) When both AND and OR are present, then AND will be processed first.

Example:

1 "ABC"<=I1<"xyz";I2="AA", "BB"



2 I2<1000 : I3="TOKYO" ; 1<I4<1000



(5) SORT SEQUENCE ..... Specifies the sort key (within 124 bytes)

U/In ..... Ascending order sort (U/ can be omitted).
D/In ..... Descending order sort.

(6) OUTPUT FNAME ..... Specifies the name of the output file (for storing the result of sorting). If this specification is defaulted, the output file name will be assumed to be the same as the input file name. If another file with the same name as that specified here is already present on the disk, then that file will be destroyed.

(7) WORKDISK ..... Specifies the disk drive unit containing the work file. If defaulted, this will be assumed to be the B side drive unit.

(8) PARAMETER FILE MAKE ? ..... Specifies the generation of a parameter file

Y: The parameter file is prepared. The name of the file should be input following 'Y'.
N: No parameter file is prepared.

(9) SORT START OK ? ..... Y: Sorting is started
N: Processing is halted and returned to the OS mode.
P: The parameters will have to be input again. Operation returns to (1).

NOTE: The processing returns to the OS mode if CTRL and C are pressed simultaneously when the system is waiting for parameter inputs.

1.3.2 Examples of Use

1. Keying in the parameters

The underlined portions below are keyed in.

```
A>SORT
  SORT Ver X.X
  INPUT FNAME ..... B:DATA.NUM          Input file name
  FILE MODE ..... R                     R (Random), S (Sequential)
  RECORD FORMAT ..... 20, 5, B2, B4, 69 Structure of each record
    RECORD-SIZE ..... 100 BYTES ITEM ... 5
  SAMPLING DATA ..... *
  SORT SEQUENCE ..... I2, D/I3, D/I4    Specification of key item
    SORTKEY ITEM ..... 3
  OUTPUT FNAME ..... B:DATA.OUT         Output file name
  WORK DISK ..... B                     Work disk drive specification
  PARAMETER FILE MAKE? Y                Parameter file is generated
  PARAMETER FNAME .... B:PFILE          Parameter file name
  SORT START OK? Y
  INPUT-DATA 200 RECORDS
  END OF SORT RESULT IS B:DATA.OUT
A>
```

2. Parameters are input from a file

```
A>SORT B:PFILE
  SORT Ver X.X
  INPUT FNAME ..... B:DATA.NUM
  FILE MODE ..... R
  RECORD FORMAT ..... 20, 5, B2, B4, 69
    RECORD-SIZE ..... 100 BYTES ITEM ... 5
  SAMPLING DATA ..... *
  SORT SEQUENCE ..... I2, D/I3, DI4
    SORTKEY ITEM ..... 3
  OUTPUT FNAME ..... B:DATA.OUT
  WORK DISK ..... B
    INPUT-DATA ..... 200 RECORDS
    END OF SORT RESULT IS .....B:DATA.OUT
A>
```

1.4 Notes on Parameter Files

1. A Parameter file has the same form as the comment statements in SBASIC. Hence, the parameter file B: PFILE explained in Section 1.3.2 takes the form shown below.

```
1 △' B:DATA.NUM [CR] . [LF]
2 △' R [CR] . [LF]
3 △' 20, 5, B2, B4, 69 [CR] . [LF]
4 △' * [CR] . [LF]
5 △' I2, D/I3, D/I4 [CR] . [LF]
6 △' B:DATA.OUT [CR] . [LF]
7 △' B [CR] . [LF]
```

2. If the extended file name NON is used for the parameter file name, the contents of the parameters will not be displayed during execution.

3. The parameter file so prepared must be within 115 bytes in length.

1.5 Error Messages

1. During parameter input

MISS FORMAT	The input was made in an unspecified format.
RECORD FORMAT LENGTH OVER	A record with more than 384 bytes was specified.
OVER KEY	More than six key items were specified.
KEY FORMAT LENGTH OVER	The total length of the key items exceeds 124 bytes.

2. Other error messages

INPUT FILE NOT PRESENT	The specified input file is not present on the disk.
PARAMETER FILE NOT PRESENT	The specified parameter file is not present on disk.
INPUT-FNAME-ERROR	There is an error in the input file entry of the specified parameter file.
FILE-MODE-ERROR	There is an error in the specified parameter file mode entry.
RECORD-FORMAT-ERROR	There is an error in the record format entry of the specified parameter file.

SAMPLING-DATA-ERROR ..... There is an error in the sampling data entry of the specified parameter file.

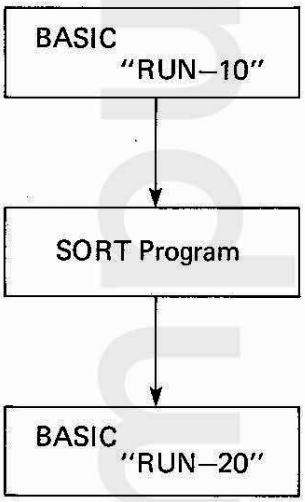
SORT-SEQUENCE-ERROR ..... There is an error in the sort sequence entry of the specified parameter file.

OUTPUT FNAME-ERROR ..... There is an error in the output file entry of the specified parameter file.

1.6 Combining BASIC and SORT

1. Purpose

The BASIC program "RUN-10" is executed, then the SORT program is executed, after which BASIC is loaded, and then the BASIC Program "RUN-20" is executed.



2. Method

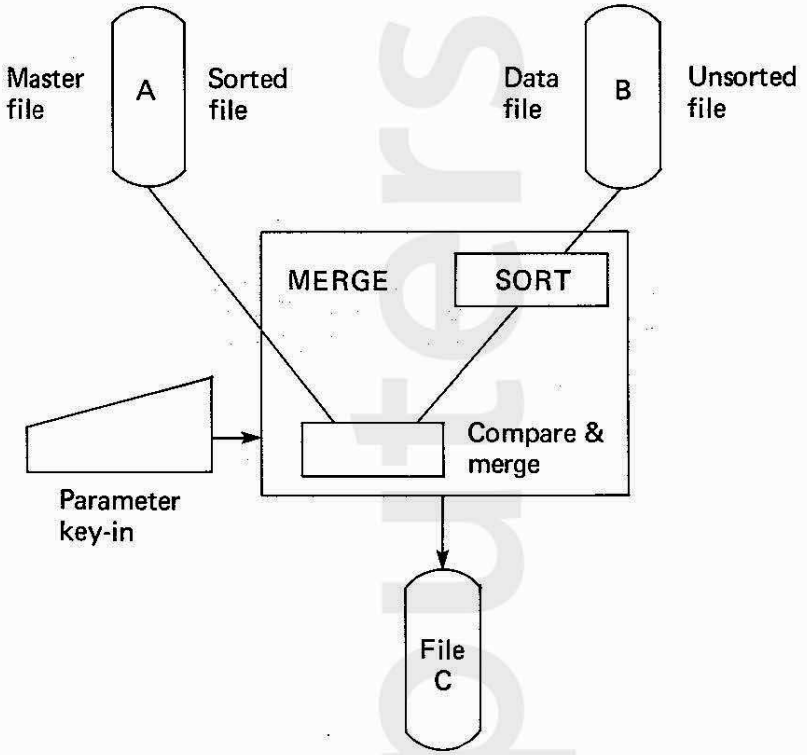
The following instructions are included in the BASIC program "RUN-10" and then that program is executed.

```
50000  OPEN "0", #1, "$$$.SUB"
50010  ZA$="SORT WORKSORT.NON"
50020  ZB$="SBASIC"+CHR$(&H22)+"RUN-20"+CHR$(&H22)
50030  PRINT#1, CHR$(LEN(ZB$));ZB$;CHR$(0);SPC(124-LEN(ZB$))
50040  PRINT#1, CHR$(LEN(ZA$));ZA$;CHR$(0)
50050  CLOSE
50060  SYSTEM
```

2. MERGE

2.1 Function of Merge

After sorting a data file (B) according to the designated key items, the MERGE utility combines the data file with a master file (A), and makes a new file (C).



2.2 Limitations of MERGE

The MERGE utility has the following limitations:

a) Data file
   1) (length of all key items + 2) x (number of record B) < 30 Kbytes
   2) Number of records $\leq$ 65,534

b) The other limitations are the same as those of item 1.2, Limitations of SORT
   1) The maximum length of one record is 384 bytes.
   2) The maximum item number of one record is 20.
   3) The maximum number of the SORT key item is 5.
   4) File end (EOF): 1AH

2.3 Operating Procedures

MERGE can be invoked by the following two ways:

1) By keying-in parameters: A> MERGE
2) By using a parameter file: A> MERGE parameter file name

2.3.1 Parameter input form

a) MASTER FNAME ..... Designates the master file name.

b) INDATA FNAME ..... Designates the input file name.

c) FILE MODE ..... Designates the form for the master and input files.

"S" ..... Sequential file
"R" ..... Random file

d) RECORD FORMAT ..... Designates the format (byte number of the master and input files).

n ..... Character item, byte number
B_n ..... Integers (B2), single precision real numbers (B4), double precision real numbers (B8)

e) SORT SEQUENCE ..... Designates the sort key selection. (within 124 bytes)

D/In ..... Ascending (U/ can be omitted)
D/In ..... Descending
n ..... Item number

f) OUTPUT FNAME ..... Designates the output file name. (can be omitted)

When omitted: Merged data is added up onto the master file.

When inputted: If the same file exists on the same disk, the file will be destroyed. When an output file name is the same as the master file name and the input data file name, the merged data is added up onto the master file.

g) UTILITY START OK?

"Y" ..... Starts the MERGE process.
"N" ..... Stops the MERGE process and returns to the OS mode.
"P" ..... Becomes the parameter re-input mode.

h) PARAMETER FILE MAKE? ..... Asks for registration of the parameter file.

"Y" ..... Inputs the registration file name.
"N" ..... No registration.

Caution:
If [CTRL] · [C] keys are pressed simultaneously in the parameter input mode, the computer will return to the OS mode.

2.3.2 Execution examples

1) When keying-in parameters:

Input the underline (_____).

```
A > MERGE
    MERGE Ver X.X
    MASTER FNAME ..... B:MASTER.FL
    IN-DATA FNAME ..... B:DATA.999
    FILE MODE ..... R
    RECORD FORMAT ..... 20, 5, B2, B4, 69
    RECORD SIZE ..... 100 BYTES ITEM ... 5
    SORT SEQUENCE ..... I12, D/I3, D/I4
    SORTKEY ITEM ..... 3
    OUTPUT FNAME .....
    PARAMETER FILE MAKE? Y
    PARAMETER FNAME ..... B: PFILE
    MERGE START OK? Y
    INPUT-DATA 200 RECORDS
    END OF MERGE RESULT IS B: MASTER.FL
A >
```

2) When using a parameter file

```
A MERGE B: PFILE
    MERGE Ver. X.X
    MASTER FNAME ..... B: MASTER.FL
    IN-DATA FNAME ..... B:DATA.999
    FILE MODE ..... R
    RECORD FORMAT ..... 20, 5, B2, B4, 69
    SORT SEQUENCE ..... I2, D/I3, D/I4
    OUTPUT FNAME .....
    RECORD SIZE ..... 100 BYTES ITEM ... 5 SORTKEY. ITEM ... 1
    INPUT-DATA 200 RECORDS
    END OF MERGE RESULT IS B:MASTER.FL
A >
```

2.4 About the Parameter File

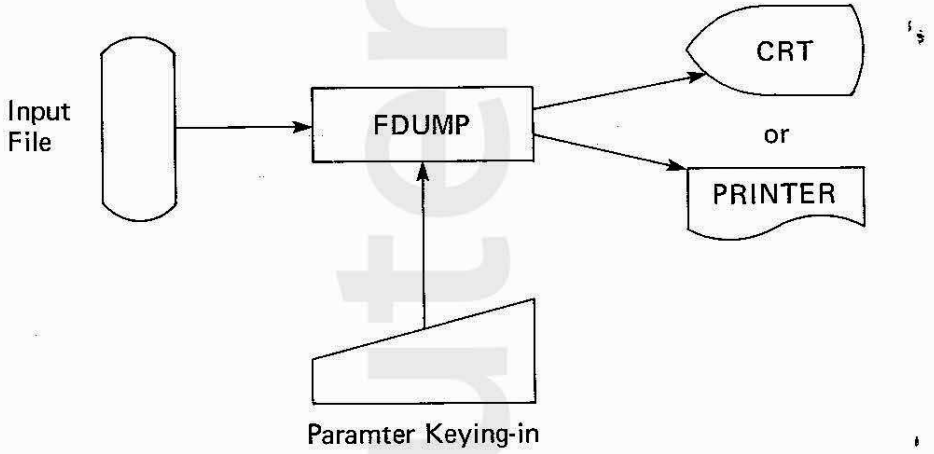
Items described in 1.4, about the Parameter File are also applied to the parameter file for MERGE.

3. FDUMP

The FDUMP is a utility which outputs the file contents either onto the CRT or to the printer according to the keyed-in parameter. There are two output forms.

(1) Character display: The file contents are displayed with characters determined by ASCII. If the corresponding character does not exist in ASCII, "? " will be displayed.

(2) Hexadecimal: The output is made in hexadecimal.



3.2 Operating Procedures

3.2.1 Parameter input form

1) FNAME ..... Designates the input file name.

2) MODE ..... Designates the output form.

A ..... Displays character
H ..... Hexadecimal output

3) DEVICE ..... Assigns the output equipment

C ..... CRT output (can be omitted)
P ..... Printer output

P X X/ X X

- Column number in one row (can be omitted; default 80 columns, XX=1 – 255)
- Line number of one page (can be omitted; default 66 lines, XX=1 – 99)

4) R-SIZE ..... Designates the record length. (can be omitted)

Designation is unnecessary with the sequential file.

5) END ..... E ..... Ends the DUMP with the EOF flag (1AH) of the file.

When omitted ..... Outputs to the final byte of the sector when EOF.

Caution:

1. If [CTRL] · [C] keys are pressed simultaneously in the parameter input mode, the computer will return to the OS mode.

2. If [CTRL] · [C] keys are pressed simultaneously during dumping, the dump execution stops. After dumping has been stopped, the second [CTRL] · [C] starts dumping again.

3.2.2 Execution Example

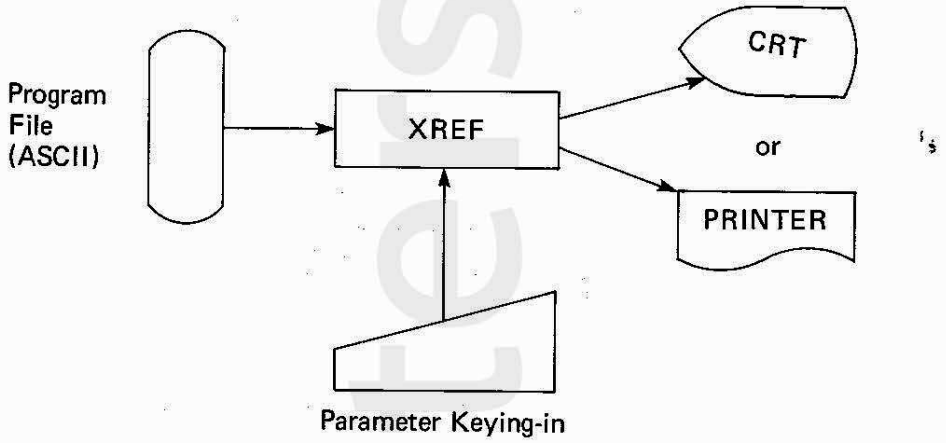
```
A FDUMP
  FDUMP Ver X.X
  FNAME, MODE, DEVICE, R-SIZE, END ....TEST. BAS, A, C
  10 PRINT "TEST"
  20 FOR N=1 TO 100
  30 A=SQR(N)
  40 B=A*N
  50 PRINT N, A, B
  60 NEXT
```

. Displayed on CRT
.
.
.
.

4. XREF

4.1 Function of XREF

The XREF has a function to make a reference of parameters used in a program made by SBASIC and saved in the ASCII form. In the list, parameters are shown in what line they are defined and are employed. (allowable parameters: 240)



4.2 Operating Procedures

4.2.1 Parameter input form

1) FNAME ..... Designates input file name.

2) MODE ..... Designates output form

A ..... Source or cross reference file form
S ..... Source file form
X ..... Cross reference file form

3) DEVICE ..... Assigns the output equipment

C ..... CRT output (Can be omitted)
P ..... Printer output
P99
Number of line in one page (can be omitted; default 66 lines)

4) DATE ..... Output date (can be omitted) YY-MM-DD (eight digits)

Cautions:

1. If [CTRL] · [C] keys are pressed simultaneously in the parameter input mode, the computer will return to the OS mode.

2. [CTRL] · [S] stops a display on CRT. After the CRT display has been stopped, the second [CTRL] · [S] continues the display again.

4.2.2 Execution Examples

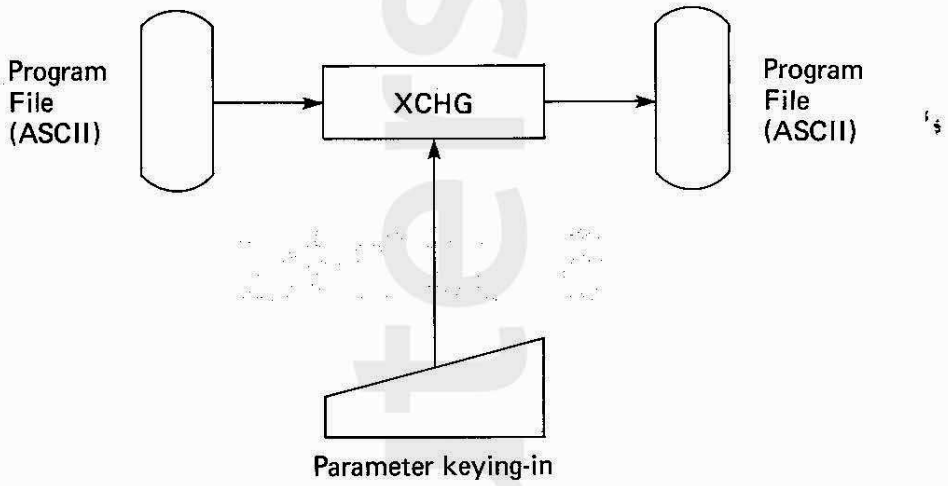
```
A XREF
  XREF Ver. X.X
  FNAME, MODE, DEVICE, DATE ..... TEST. BAS, A, C
  10 PRINT "TEST"
  20 FOR N=1 TO 100
  30 A=SQR(N)
  40 B=A*N
  50 PRINT N, A, B

A! DEFINE      30
   REFERENCE   40   50
B! DEFINE      40
   REFERENCE   50
N! DEFINE      20
   REFERENCE   30   40   50
```

In a cross reference, a type (!, #, %, $, etc.) is added.

5. XCHG

5.1 Function of XCHG

The XCHG has a function to replace characters and parameters in a program made by SBASIC and saved in the ASCII form.



5.2 Operating Procedures

5.2.1 Parameter input form

1) INPUT FNAME ..... Designates an input file name.

2) OUTPUT FNAME ..... Designates an output file name.
(When omitted, it is the same as the output file name.)

3) CHANGE SYMBOL ..... Designates a new parameter name.

9 – 9 – 9 – 9/S/"X – X" = "Y – Y", ...

- "Y – Y": Old parameter name
- "X – X": New parameter name
- S: When input S: Not only parameters but all equivalent characters are replaced.
- When defaulted: Only parameters are replaced.
- 9 (second): End line number. When omitted, to the final line.
- 9 (first): Start line number. When omitted, from the first line.

Up to 5 parameter names can be input.
Parameter names should be within 20 columns.

Caution:

If CTRL · C keys are pressed simultaneously in the parameter input mode, the computer returns to the OS mode.

5.2.2 Execution examples

```
A XCHG
  XCHG Ver. X.X
  INPUT FNAME ..... TEST.BAS
  OUTPUT FNAME ..... TEST.NEW
  CHANGE SYMBOL ..... "XX"="A", "YY"="B", "MM"="N"
```

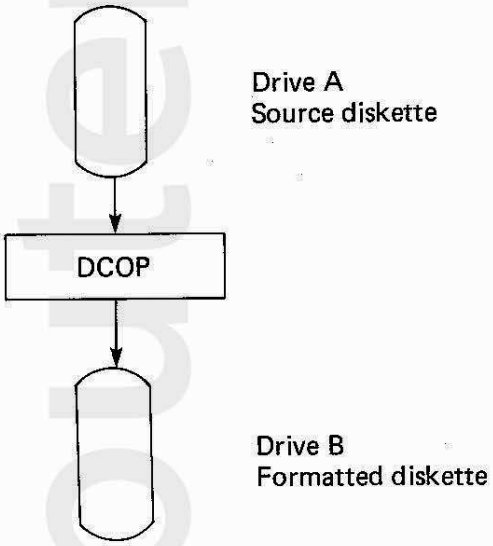
TEST.BAS	CHANGED TO	TEST.NEW
10 PRINT "TEST"		10 PRINT "TEST"
20 FOR N=1 TO 100		20 FOR MM=1 TO 100
30 A=SQR(N)		30 XX=SQR(MM)
40 B=A*N		40 YY=XX*MM
50 PRINT N, A, B		50 PRINT MM, XX, YY
60 NEXT		60 NEXT

6. DCOP

6.1 Function of DCOP

DCOP copies all the content of a diskette. It copies the content of a diskette set in Drive A to another diskette set in Drive B. Two types of copy are provided: STANDARD MODE and VERIFY MODE.

STANDARD MODE: Read from Drive A and write to drive B.
VERIFY MODE: Read from Drive A and write to drive B. Check if the written data to B can be read.



6.2 Condition of DCOP

The diskette in Drive B should be of the same type as the one in Drive A and should be formatted.

6.3 Operation

A > DCOP

DCOP – DISK TO DISK COPY – VER X.X DOUBLE

Set formatted disk to unit B, source disk to unit A.
Type S (STANDARD MODE) or V (VERIFIED MODE) followed by carriage return, or carriage return to exit.

(1) If the message shown above is displayed, set a source diskette in Drive A and a formatted diskette in Drive B. And specify one of the copy types as follows:

S (CR)	Copies in a standard manner.
V (CR)	Copies with verification.
CR	Returns to OS mode.

(2) When copy is completed, "Function completed" will be displayed.

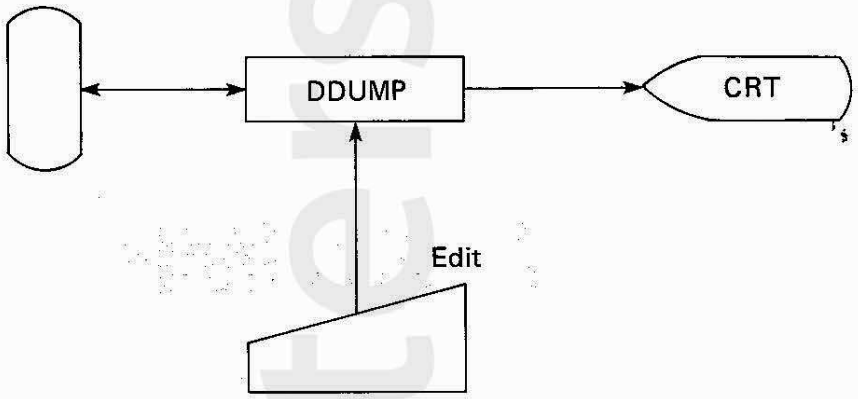
6.4 Error Messages

DISK READ ERROR	The disk in Drive A cannot be read.
DISK WRITE ERROR	The disk in Drive B cannot be written.
INVALID SYNTAX (use S or V)	Other than S or V is input for copy type specification.
SYSTEM TYPE MISMATCH ERROR	OS or Drive A and Drive B are mismatched.
NOT SUPPORTED SYSTEM	The system is not supported.

7. DDUMP

7.1 Function of DDUMP

DDUMP displays on the CRT or converts the data content of the diskette in unit of track or sector.



7.2 Condition of DDUMP

The number of tracks or sectors varies depending on each system:

Model	Used diskette	Drive	Track	Logical No. of sectors	Physical No. of sectors
MBC-3000D (1024)	Double side, Double density	A – P	0 – 153	1 – 64	32
3000S	Single side, Single density	A – P	0 – 76	1 – 26	26
2000	Single side, Double density, Double track	A – P	0 – 79	1 – 32	16
3000D (256)	Double side, Double density	A – P	1 – 153	1 – 52	26

- Track: For double side type, the right side and the reverse side are alternately numbered as 0, 1, 2, ... , n.
- Logical No. of Sectors: Number of sectors in one track where 128 bytes are counted as one sector.
- Physical No. of Sectors: Actual number of sectors in one track

The logical sector is to be used for DDUMP execution thus "sector" hereafter denotes the logical sector.

7.3 Operation

When DDUMP is executed, Command Mode and Edit Mode are provided:

(1) Function of Command Mode

- * appears on CRT as a prompter.

Input format of parameters

*[[Drive Name:],]Track No.[,Sector No.[,EDIT]] (CR)

Note: Although "," (comma) is used as a separator, "." (period) or " " (space) can also be used.

1) CHANGE OF CURRENT DISK . . . . . Changes the drive currently selected.

2) DISPLAY OF TRACK OR SECTOR . . Displays in unit of track or sector.

3) CHANGE TO EDIT MODE . . . . . Changes to Edit Mode.

4) RETURN TO OS . . . . . Terminates DDUMP program and returns to OS.

(2) Function of Edit Mode

- – appears on CRT as a prompter.

1) A data of one byte within the data of one sector shown on CRT can be modified.

i) Specify an address (00H–7FH).
The specified address and its content will be displayed.

ii) Input the data to replace (00H–FFH).
The next address and its content will be displayed.

iii) If modification is completed, input a "." (period).
The modified content of one sector is displayed on CRT and returns to the Edit Mode.

2) Change to Command Mode.

i) W (CR) :
Write the modified one sector in the disk and returns to the Command Mode.

ii) Q (CR) :
Cancel the modified part and returns to the Command Mode.

7.4 Example of Command Execution

*0, 1 (CR)	1 sector in 0 track is displayed.
*B:, 2, 3 (CR)	The current disk is changed to Drive B and 3 sector in 2 track of Drive B is displayed.
*A: (CR)	The current disk is changed to Drive A.
*1, 50, EDIT (CR)	50 sector in 1 track is displayed and changes to the Edit Mode.
*E (CR)	Returns to OS Mode.

7.5 Example of Edit Execution

A > DDUMP (CR)

DDUMP – DISK DUMP – VER X.X DOUBLE

*1, 50, EDIT (CR) ← Track and sector are specified. Changed to the Edit Mode.

```
                DRIVE A – TRACK    1    SECTOR     50
0000   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0050   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0060   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0070   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

–6F (CR) ← Specifies the address of the data to be modified (00H–7FH).

006F 00 41 (CR) ← Modification data (00H–FFH)

0070 00 . (CR) ← Modification completed

```
                DRIVE A – TRACK    1    SECTOR     50
0000   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0050   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0060   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 41  ...............A
0070   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

–W (CR) ← Writes in the disk and returns to the Command Mode.

*E (CR) ← Returns to OS.

7.6 Error Message

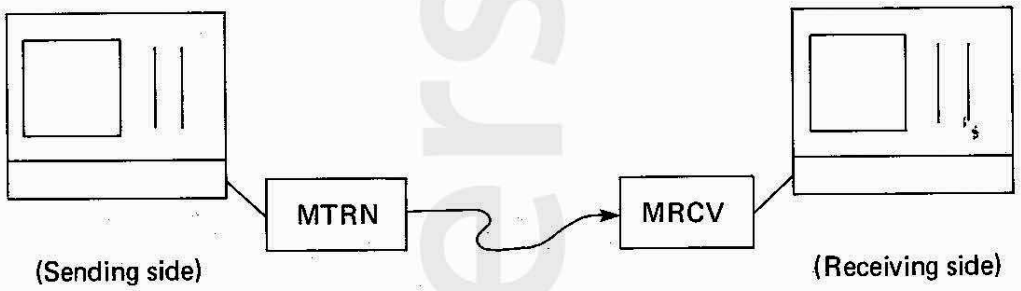
DISK READ ERROR	Cannot be read from the disk.
DISK WRITE ERROR	Cannot be written in the disk.
ADDRESS ERROR	Address specification error at edit execution
ERROR HEX INPUT ONLY	Modification data error at edit execution
NOT SUPPORTED SYSTEM	A drive which OS does not allow is specified.
INVALID SYNTAX	Input error
INVALID SYNTAX Q (ABORT)	Input error at return from Edit to Command
INVALID SYNTAX W (WRITE)	Input error at return from Edit to Command

8. MTRN, MRCV (FTRN, FRCV)

8.1 Function of MTRN, MRCV

File transfer is performed between two devices using a communication line.

MTRN is used at sending side and MRCV at receiving side.



Transfer speed: 4,800 baud, 8 bits, no parity

8.2 Condition of MTRN and MRCV

Connection terminal of communication line varies depending on each device and see the notes.

8.3 Operation

(1) Connect the sending side and the receiving side with a communication line.

(Sending side) (Receiving side)

(2) A > MRCV (CR)

(3) A > MTRN file-name (CR)

(4) SELECT DRIVE (A, B, C, or D) will be displayed.
Specify one of these drives.

Transfer

(5) At the completion of transfer, the buzzer sounds and "SUCCESS" is displayed.

- Transfer of more than one file can be specified (*, ?).
- If the receiving side has a file whose name is the same as that of the transfered file, "FILE EXISTS, DELETE (Y/N)? " is displayed and file transfer is paused. If this file is to be transfered, input [Y] , and if not, input [N] , respectively.
- There may be a case where "LINE ERR" is displayed. If it occurs, repeat the same procedure from the beginning.

8.4 Example

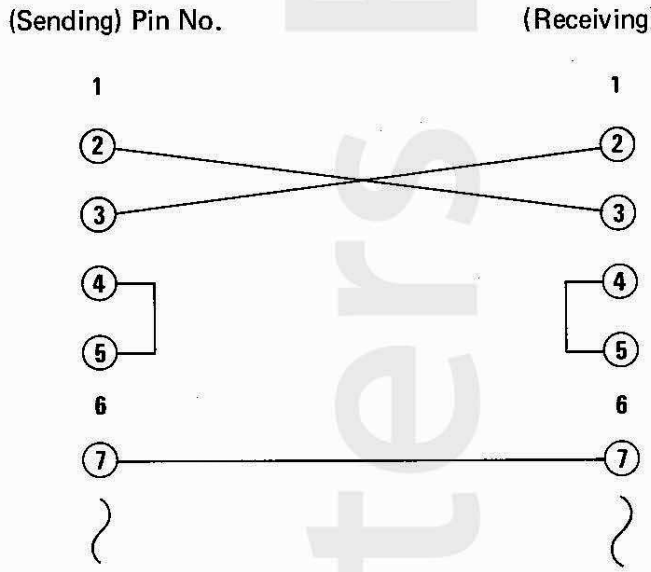
(1) Connection Example on MBC-3000/2000 Series

- Use the line 1

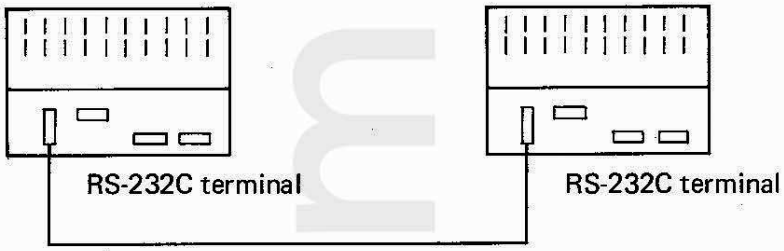
- Signal Names and Flows

Code	Signal name	Signal direction Device⇄External	Connector Pin	Remarks
FG	Protective Ground		1	
SD	Sending Data	→	2	Data that the device outputs
RD	Receiving Data	←	3	Data that the device inputs
RD	Send Request	→	4	
SC	Send Ready	←	5	
DR	Data Set Ready	←	6	
SG	Signal Ground		7	

○ Connection of Connector Pins of Sending and Receiving Sides



(2) Connection Example on MBC-1000



Change the switch of the RS-232C interface board as follows:

[JP1] IC8251A mode setting
Set No. 1 ON.

[JP2] Baud rate setting
Set No. 6 ON (4,800 bauds).

[JP3] Line clock setting
Set No. 2, No. 4 ON.

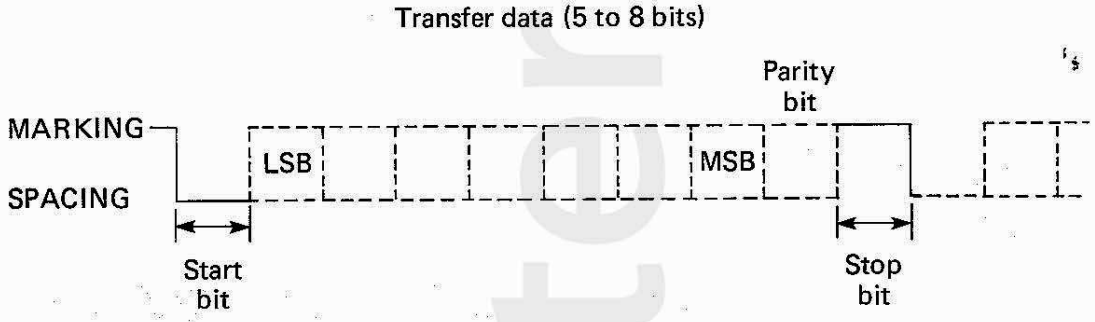
Note: For MBC-1000, use FTRN, FRCV in place of MTRN, MRCV.
FTRN, FRCV and MTRN, MRCV are the same except for the above jamper setting.

9. LINE
(only for MBC-2000/3000)

9.1 Function of LINE

LINE sets the mode of connection terminals of a communication line.

9.2 Signal Specification



* Start bit is fixed to 1.

9.3 Operation

A > LINE (CR)

Select conditions for each item displayed on CRT

(1) Item 1 SELECT LINE ..... Specifies the connection terminal to be used.

(2) Item 2 STOP BIT ..... Specifies the length of the stop bit.

(3) Item 3 PARITY ..... Specifies if with parity or without parity.

(4) Item 4 PARITY CHECK ..... Specifies Odd parity or Even parity if with parity.

(5) Item 5 CHARACTER LENGTH .... Specifies the length of one character.

(6) Item 6 BAUD RATE ..... Specifies the transfer speed.

9.4 Example of LINE Execution

Set the LINE-1 of MBC-3000 to the following conditions (conditions at power-on).

STOP BIT 1
PARITY ..... With Parity
PARITY CHECK ..... Even Parity
CHARACTER ..... 7 bits
BAUD RATE ..... 9600 baud

A LINE (CR)

SELECT LINE

LINE-1 ——— 1
LINE-2 ——— 2

Key-in 1

STOP BIT

1 bit ——— 1
1.5 bit ——— 2
2 bit ——— 3

Key-in 1

PARITY

enable ——— 1
disable ——— 2

Key-in 1

PARITY CHECK

odd ——— 1
even ——— 2

Key-in 2

CHARACTER LENGTH

5 bit ——— 1
6 bit ——— 2
7 bit ——— 3
8 bit ——— 4

Key-in 3

BAUD RATE

19200 ——— 1
9600 ——— 2
4800 ——— 3
2400 ——— 4
1200 ——— 5
600 ——— 6
300 ——— 7

Key-in 2

Note: LINE.COM is valid only for MBC-2000/3000

10. VERIFY

10.1 Function of VERIFY

VERIFY is a program which compares two files byte by byte.

File 'F1' File 'F2'

10.2 Condition of VERIFY

Files to be compared should be of sequential files.

10.3 Operation

Input as the following in the OS mode:

A > VERIFY file-1 file-2 (CR)

Specify the file-1 and the file-2 to be compared. The name of drive can be specified right before the file name.

VERIFY DIR.COM B:DIR.COM (CR)

If the contents of the two files are completely identical, "VERIFY PASS" will be displayed.

10.4 Error Messages

SYNTAX ERROR	File name input error.
OPEN ERROR	File name not found in the disk.
READ ERROR	The number of sectors of the file-1 is greater than that of the file-2.
SECTOR NO.	Comparison until the displayed sector number has been completed.
LENGTH ERROR	The number of sectors of the file-1 is less than that of the file-2.
VERIFY ERROR	The file-1 and the file-2 are not identical each other.

10.4.1 Display of Error Data

SECTOR XXXX POSITION XXX DATA XX,XX

SECTOR	Indicates the sector number. (1—9999) Decimal
POSITION	Indicates the position in a sector. (0—127) Decimal
DATA	Indicates data of the file-1 and the file-2. Hexadecimal

SANYO

Specifications and information are subject to change without prior notice. ABC 820001

Printed in Japan Jun./'82