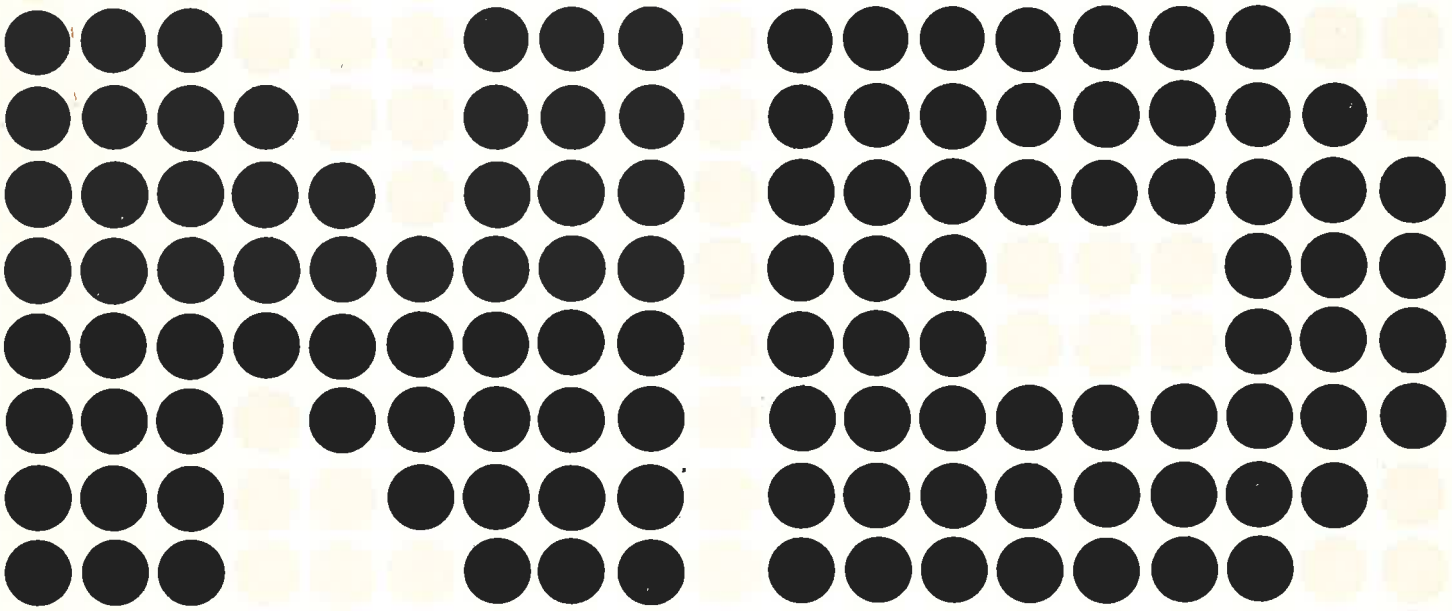
MSD-SORT SYSTEM

NORSK DATA A.S



MSD-SORT SYSTEM

NOTICE

The information in this document is subject to change without notice. Norsk Data A.S assumes no responsibility for any errors that may appear in this document. Norsk Data A.S assumes no responsibility for the use or reliability of its software on equipment that is not furnished or supported by Norsk Data A.S.

PRINTING RECORD

Printing	Notes
06/79	VERSION 01 – ORIGINAL PRINTING
06/80	VERSION 02

MSD-Sort System
Publication No. ND–123.02

NORSK DATA A.S
P.O. Box 4, Lindeberg gård
Oslo 10, Norway

Manuals can be updated in two ways, new versions and revisions. New versions consist of a complete new manual which replaces the old manual. New versions incorporate all revisions since the previous version. Revisions consist of one or more single pages to be merged into the manual by the user, each revised page being listed on the new printing record sent out with the revision. The old printing record should be replaced by the new one.

New versions and revisions are announced in the ND Bulletin and can be ordered as described below.

The reader's comments form at the back of this manual can be used both to report errors in the manual and to give an evaluation of the manual. Both detailed and general comments are welcome.

These forms, together with all types of inquiry and requests for documentation should be sent to the local ND office or (in Norway) to:

Documentation Department
Norsk Data A.S
P.O. Box 4, Lindeberg gård
Oslo 10

TABLE OF CONTENTS

+ + +

Section:		*Page:*
1	INTRODUCTION	1—1
2	CAPACITY OF MSD-SORT SYSTEM	2—1
3	USING THE SYSTEM	3—1
3.1	Description of Parameters, Used as SINTRAN III Subsystem	3—1
3.2	Description of Parameters, Used as Subroutine	3—4
3.3	Examples of Using The System	3—6
3.4	Messages From The System	3—8
3.5	Error Messages	3—8
3.6	Hints and Restrictions For This Sort System	3—9
4	METHOD USED	4—1
4.1	Sorting	4—1
4.2	Merging	4—2

Appendix:		*Page:*
A	ASCII CHARACTER SET	A—1
B	FORTRAN CHARACTER STRINGS	B—1

1 INTRODUCTION

The MSD-Sort System is a program package enabling the user to sort mass storage file (magnetic tapes included) containing fixed length records (but not variable length). The package is available in the following two forms:

- SINTRAN III interactive subsystem
- Subroutine callable from a user program

The program is written in NORD PL and occupies 6k of the main memory as SINTRAN III subsystem and 5k as subroutine. In addition, a sort buffer area is used. To minimize the sort time, the buffer area is set as large as possible. For the SINTRAN III subsystem, the buffer area is set to 58k; for subroutine, the user must specify his own buffer area.

The MSD-Sort System uses a scratch file. The size of the scratch file depends on the size of the input file and will be either the same size or twice the size of the file to be sorted (depending on the buffer size). The user may specify his own scratch file, or he may use the default scratch file.

In this version it is possible to sort on alphanumeric keys, and on numeric-display keys (stored as ASCII characters) with or without sign. Ascending or descending sort sequence may be specified. It is also possible to build up an alternative sort sequence.

2 CAPACITY OF MSD-SORT SYSTEM

The file to be sorted is divided into partitions, which are sorted independently of each other. The sorted partitions are stored temporarily on the scratch file. When all partitions are sorted, they are merged and written into the output file. If the file is small, the entire file will be sorted as one partition.

If not all sorted partitions can be merged in one pass (due to lack of available memory buffer area), they will be merged into greater partitions and stored temporarily back on the scratch file. The process will be repeated until the number of partitions is less than the maximum number that can be merged in one pass.

The maximum input file size the MSD-Sort System is able to sort is approximately:

((30.000 * A) – 60.000.000) bytes

where A is the buffer area size in bytes.

This gives as the maximum size of the input file:

Buffer area size in k words	Maximum input file size in mega-bytes *
2	60 (30)
4	180 (90)
8	420 (210)
16	900 (450)
32	1860 (930)
58	3400 (1700)

TABLE 2.1

*If the record length is an odd number of bytes, the maximum size is half (See Section 4.2)

3 USING THE SYSTEM

3.1 *DESCRIPTION OF PARAMETERS, USED AS SINTRAN III SUBSYSTEM*

The MSD-Sort System is implemented as a subsystem of SINTRAN III, and may be called from the terminal as follows:

@ MSD-SORT

and the following commands will be available:

HELP
lists the available commands

EXIT
Exits to the operating system

SCRATCH-FILE <file-name >

<file-name > = A file name or, if previously opened for random read/write, an octal file number. Default file type is DATA.

This command may be omitted and the scratch file 100 will be used.

RECORD-DESCRIPTION <record-len > <key-pos > <len > <seq >
[<key-pos > <len > <seq >] . . .

<record-len > = Length, in bytes, of the fixed length records on the file. (Remember to count carriage return and line feed).

<key-pos > = The position in the record of the first byte of the key. If the key is located at the start of the record, this parameter should be specified to 1.

<len > = The length, in bytes, of the key.

<seq > = The sequence the key is to be sorted in. It may be specified as:

ASCENDING DESCENDING	the key will be sorted according to the ASCII value of the character
ALT-ASCENDING ALT-DESCENDING	the key will be sorted according to the alternative collating sequence specified (See ALTERNATIVE-COLLATING-SEQUENCE).

NUMERIC-ASCENDING
NUMERIC-DESCENDING

the key is supposed to be a numeric-display key (stored as ASCII characters). If the first byte in a key is the character –(minus), the key is taken as a negative number. Any character different from the decimal digits, letters, and –, will be taken as 0.

BLOCK-FACTOR-INPUT <number>

<number> = number of records in each block for input file.

This gives the block factor for input file (only for magnetic-tape files).

BLOCK-FACTOR-OUTPUT <number>

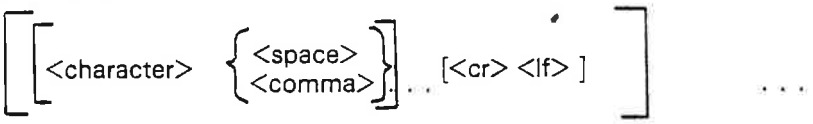
<number> = Number of records in each block for output file.

This gives the block factor for output file.

ALTERNATIVE-COLLATING-SEQUENCE <file-name>

<file-name> = Name or octal number of file where the ascending sequence of the alternative sort sequence is specified.

The format of the contents of the file is:



The characters not specified, will be appended to the ascending sequence, according to the ASCII value. Default file type is DATA.

Example of a file where the alternative collating sequence is specified. The ascending sequence is to be:

- space
- characters A–Z
- figures 0–9
- the remaining part of the ASCII character set.

,A,B,C,D,E,F,G,H,I,J,K,L,M,N
O,P,Q,R,S,T,U,V,W,X,Y,Z,0,1,2
3,4,5,6,7,8,9

SORT <input-file> <output-file>

<input-file> = Name or octal number of input file. If number, the file must have been opened for random-read.

<output-file> = Name or octal number of output file. If number, the file must have been opened for random-write.

Input file and output file may be the same file. Default file type is DATA.

The command SORT must be the last command. The Sort System will ask for all missing parameters, so if you do not remember which parameters to give, just terminate each parameter by <CR > . In RECORD-DESCRIPTION the System will only ask for one key – list (<key-pos > <len > <seq >). If you have more keys, you have to type the first parameter of the new key-list on the same line as the end of the last key-list.

The parameters may be separated either by comma(s) or by one or more spaces.

3.2 *DESCRIPTION OF PARAMETERS, USED AS SUBROUTINE*

The MSD-Sort System is implemented as a subroutine callable from user programs. It may be called as follows:

CALL SORT (Input, Output, Scratch, Recl, N-key, Key, Buff-size, Buff, BI-inp, BI-outp, Coll-File, Status)

Input = Name of input file (FORTRAN Character Format) For the specification of FORTRAN Character Format see Appendix B. Default file type is DATA.

Output = Name of output file (FORTRAN Character Format). If no output file 0 (integer) is specified. Default file type is DATA.

Scratch = Name of scratch file (FORTRAN Character Format). If default scratch file is to be used this parameter is set to 0 (integer). Defaulat file type is DATA.

Recl = Length, in bytes, of the fixed length records on the file. (Remember to count carriage return and line feed).

N-key = Number of keys (Integer).

Key = An integer array containing as many <key-lists > as specified in N-Key.

A <key-list > consists of <key-pos >,<len >,<seq>

<key-pos > = The position in the record of the first byte of the key. If the key is located at the start of the record, this parameter should be specified to 1.

<len > = The length, in bytes, of the key.

<seq > = The sequence the key is to be sorted in. It may be specifed to:

0 = normal ascending sort sequence.
1 = normal descending sort sequence.
2 = alternative ascending sort sequence.
3 = alternative descending sort sequence.
4 = numeric ascending sort sequence.
5 = numeric descending sort sequence.

Buff-size = Sort buffer area size in words (Integer). The buffer size must be greater than 1k words (1024).

Buff = The sort buffer area.

Bl-inp = Number of records in each block for input file (Integer). (Only for magnetic tape files). If specified to 0, default block size is used.

Bl-outp = Number of records in each block for output file (Integer). If specified to 0, default block size is used.

Coll-File = Name of file where the ascending sequence of the alternative sort sequence is specified (FORTRAN Character Format). If no alternative sort sequence is used, this parameter should be specified to 0 (Integer). For the format of the contents of the file, See ALTERNATIVE-COLLATING-SEQUENCE used as SINTRAN III subsystem. Default file type is DATA.

Status = Status return from the MSD-Sort System.

STATUS = 0 The sorting has finished and no error has occurred

0 <STATUS <400B I/O system error, and STATUS contains the SINTRAN III File System Error Code. The SINTRAN III manual should be consulted.

STATUS = 402B NO SUCH COLLATING SEQUENCE
= 403B ERROR IN OCTAL NUMBER
= 404B SORT FILE TOO BIG FOR SPECIFIED BUFFER SIZE
= 406B TOO LONG TOTAL KEY
= 407B ERROR IN SPECIFYING ALTERNATIVE COLLATING SEQUENCE
= 410B ILL. RECORD LENGTH

These messages are explained in Section 3.5 (Error Messages).

3.3 *EXAMPLES OF USING THE SYSTEM*

EXAMPLE 1:

```
@MSD-SORT
*RECORD-DESCRIPTION  80,1,12,ASCENDING,25,5,DESCENDING
*SORT INN-DATA, OUT-DATA
*EXIT
```

EXAMPLE 2:

```
@MSD-SORT
*REC-DE
RECORD-LEN: 80
KEY-POS:      1
LEN:         12
SEQ:         ASC,25
LEN:          5
SEQ:         DES
*SORT
INPUT-FILE: INN-DATA
OUTPUT-FILE: OUT-DATA
*EX
```

EXAMPLE 3:

```
@MSD-SORT
*RECORD  80
KEY-POS: 1,12,ASC,25,5,DESC
*SORT  INN-DATA
OUTPUT-FILE: OUT-DATA
*EX
```

All three of the above examples have the following meaning:

- Sort the file INN-DATA and place the result on file OUT-DATA.
- The data on INN-DATA is considered as records of length 80 bytes (characters).
- Two sort keys
  - Most significant key starts at the first byte, with a length of 12 bytes, collating sequence is ascending.
  - Second key starts at byte 25 with a length of 5 bytes, collating sequence is descending.

EXAMPLE 4:

```
@MSD-SORT
*SCRATCH  SSSSS
*ALT-COLL-SEQ  COLL-FILE
*REC-DE 80,1,12,ALT-AS,25,5,ALT-DES
*SORT  INN-DATA  OUT-DATA
*EX
```

Example 4 shows, in addition to the three examples above, the following:

— Use the scratch file SSSSS

— The keys are to be sorted according to an alternative collating sequence, specified in file COLL-FILE.

Example of using the MSD-Sort package as a subroutine called from a FORTRAN program.

Example 5 has the same meaning as Example 1 to Example 3 and Example 6 has the same meaning as Example 4. Buffer size used in these examples is 8k words (20000 octal words).

EXAMPLE 5:

```
10      FORMAT (' **** ERROR *** ',Z3)
        INTEGER KEY(6)
        INTEGER IBUFF(20000B)
        CHARACTER FINAM*16
        DATA KEY/1,12,0,25,5,1/
        FINAM = 'INN-DATA'
        CALL SORT (FINAM,'OUT-DATA',0,80,2,KEY,20000B,IBUFF,0,0,0,IST)
        IF (IST .NE. 0) THEN
              WRITE (1,10) IST
        ENDIF
        END
```

EXAMPLE 6:

```
10      FORMAT (' **** ERROR *** ',Z3)
        INTEGER RECL
        INTEGER KEY (6)
        INTEGER IBUFF (20000B)
        CHARACTER FINAM*16
        DATA KEY/1,12,2,25,5,3/
        RECL = 80
        FINAM = 'OUT-DATA'
        CALL SORT ('INN-DATA',FINAM,'SSSSS',RECL,2,KEY,20000B,IBUFF,
      * 0,0,'COLL-FILE',IST)
        IF (IST .NE. 0) THEN
              WRITE (1,10) IST
        ENDIF
        END
```

3.4 *MESSAGES FROM THE SYSTEM*

While the sorting is being done, and after it is finished, the system will print out some information, such as:

MERGE STARTED
<number > RECORDS SORTED

These messages are all self-explanatory.

3.5 *ERROR MESSAGES*

The following error messages may be printed by the Sort program.

SORT FILE TOO BIG FOR SPECIFIED BUFFER SIZE
This message is printed if the input file size is greater than what is possible to sort with the present buffer area (See Table 2.1).

TOO MANY KEYS
Maximum 7 keys permitted, only for SINTRAN III subsystem.

TOO LONG TOTAL KEY
Maximum length (sum of individual key lengths) of total key is 255 bytes.

ERROR IN DECIMAL NUMBER
Input not decimal number.

ERROR IN OCTAL NUMBER
Input not octal number

NO SUCH COLLATING SEQUENCE
Specified sequence does not exist.

ERROR IN SPECIFYING ALTERNATIVE COLLATING SEQUENCE
The collating sequence file is not specified correctly, and the Sort System can not make an alternative sort sequence.

ILL. RECORD LENGTH
The record length does not match with maximum byte pointer of the file to be sorted.

3.6 *HINTS AND RESTRICTIONS FOR THIS SORT SYSTEM*

This documentation covers a preliminary version of the MSD-Sort System, and therefore some hints will be given for how to use it, and also some restrictions which exist in the preliminary version.

- It is possible to sort on alphanumeric keys, and on numeric integer keys.
- Restrictions on input file size (See Table 2.1).
- Maximum number of keys is 7, only for SINTRAN III subsystem. Used as a subroutine there is no limit on number of keys.
- Total key length is maximum 255.
- Magnetic tape output is not permitted.
- For magnetic tape input, always specify block factor input. The number must be equal to the block factor on the tape.
- Using alternative sort sequence is as fast as normal sort sequence. Alternative and normal sort sequence can be used in the same run.
- Specify your own scratch file and use a continuous file. Using continuous file is faster than using indexed file.

4 METHOD USED

4.1 *SORTING*

A most-significant-digit-first radix sort algorithm is used to sort the partitions (MSD-radix).

The number of records sorted in each partition is determined as the integral number buffer size/record length. The sorting is performed from the most significant byte towards the least significant byte. Records with identical k first bytes in their keys are chained together. The sorting of their k + 1'th key position will generally split the chain into several sub-chains. When a chain contains a single record, its position can be determined and this record is not involved in any further processing. The sequence of sorted records is built up in an array and each record will be moved once (at most). The terminal sort condition is reached when:

$$\frac{n}{C^k} = 1, k <= k_{max}$$

where:

n	number of records in the partition
C	number of different characters used in the key alphabet
k	average number of key characters to be processed
k_{max}	total number of key characters in a record

This means that:

$$k = \ln n / \ln C$$

If we roughly assume the sorting time (exclusive I/O, which is proportional to the record length) to be proportional to the number of characters processed (all records in main memory) the algorithm is always better than normal radix sort where all key positions are processed (in reversed order) ($k = k_{max}$). When either the key-alphabet-set or the key-length are reduced, the improvements of MSD-radix are rather poor. However, in practical cases the improvements are significant. With a record length of 80 characters (all key characters randomly distributed), key length of 20, C = 26 (all letters) and n = 1000, the MSD-radix is 9 times faster. If the key is extended to cover all 80 characters, the difference will increase to about 36 times faster because it is independent of key length.

4.2 *MERGING*

The merging system simply compares the keys of the first records in each partition and outputs the least (if ascending sequence is specified) of them to the output file. This is repeated until all partitions are empty. The merging system uses a variable length buffer for each input file partition and one (1 k buffer) for the output file.

If the numbers of partitions sorted is greater than the number of partitions the system is capable of merging in one pass, than the maximum number of partitions will be merged and stored temporarily back on the scratch file. This will be repeated until all sorted partitions are merged and stored back on the scratch file. The scratch file will now contain sorted partitions with greater partition size and a smaller number of partitions. A new pass of merging will be started and the process will be repeated until all partitions can be merged and written to the output file.

The number of passes the merge process will require is:

$$n = \lceil \log a / \log b \rceil + 1$$

where a is the number of partitions sorted from the sort phase and
b is the maximum number of partitions that can be merged.

$$a = \lceil F/A \rceil + 1 \text{ and}$$

$$b = \left\lceil \frac{A - U}{L + 16} \right\rceil$$

F is the size of the input file in bytes.
A is available memory buffer size in bytes.
U is output buffer size, default 2048 bytes. Can be changed by using BLOCK-FACTOR-OUTPUT (BI-outp).
L is the record length in bytes, or if the record length is an odd number of bytes, L is 2* record length. This is due to even byte block transfer.

If $a = 1$ then the entire input file is sorted directly into the output file and no scratch file will be used. If $a > 1$ and $n = 1$ then the scratch file will be of the same size as the input file, and if $n > 1$, the scratch file needed is twice the size of the input file.

APPENDIX A

ASCII CHARACTER SET

Graphic:	Octal Value:	Decimal Value:	ASCII Abbreviation:	Comments:
	0	0	NUL	Null
	1	1	SOH	Start of heading
	2	2	STX	Start of text
	3	3	ETX	End of text
	4	4	EOT	End of transmission
	5	5	ENQ	Enquiry
	6	6	ACK	Acknowledge
	7	7	BEL	Bell
	10	8	BS	Backspace
	11	9	HT	Horizontal tabulation
	12	10	LF	Line feed
	13	11	VT	Vertical tabulation
	14	12	FF	Form feed
	15	13	CR	Carriage return
	16	14	SO	Shift out
	17	15	SI	Shift in
	20	16	DLE	Data link escape
	21	17	DC1	Device control 1
	22	18	DC2	Device control 2
	23	19	DC3	Device control 3
	24	20	DC4	Device control 4
	25	21	NAK	Negative acknowledge
	26	22	SYN	Synchronous idle
	27	23	ETB	End of transmission block
	30	24	CAN	Cancel
	31	25	EM	End of medium
	32	26	SUB	Substitute
	33	27	ESC	Escape
	34	28	FS	File separator
	35	29	GS	Group separator
	36	30	RS	Record separator
	37	31	US	Unit separator
	40	32	SP	Space
!	41	33	!	Exclamation marks
"	42	34	"	Quotation marks
#	43	35	#	Number sign
$	44	36	$	Dollar sign
%	45	37	%	Percent sign
&	46	38	&	Ampersand
'	47	39	'	Apostophe
(	50	40	(	Opening parenthesis

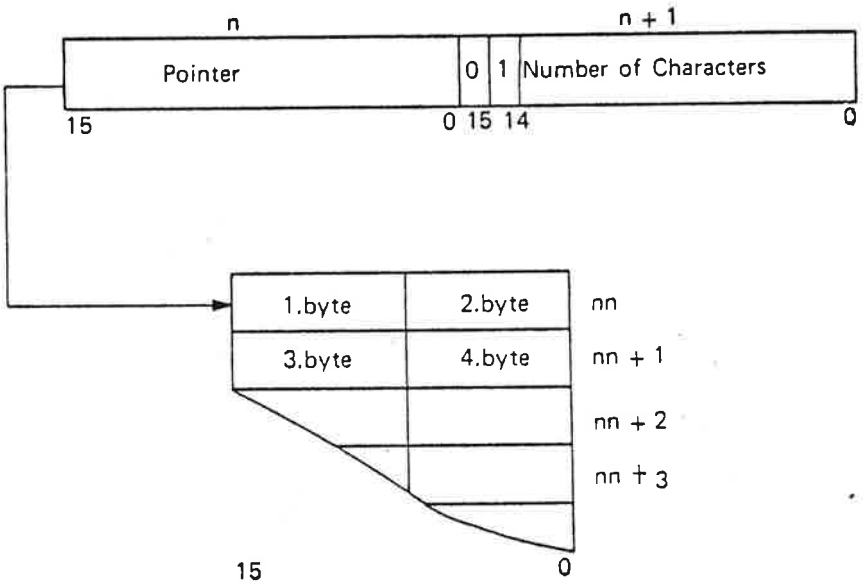
Graphic:	Octal Value:	Decimal Value:	ASCII Abbreviation:	Comments:
)	51	41	)	Closing parenthesis
*	52	42	*	Asterisk
+	53	43	+	Plus
,	54	44	,	Comma
—	55	45	—	Hyphen (Minus)
.	56	46	.	Period (Decimal)
/	57	47	/	Slant
0	60	48	0	Zero
1	61	49	1	One
2	62	50	2	Two
3	63	51	3	Three
4	64	52	4	Four
5	65	53	5	Five
6	66	54	6	Six
7	67	55	7	Seven
8	70	56	8	Eight
9	71	57	9	Nine
:	72	58	:	Colon
;	73	59	;	Semi-colon
<	74	60	<	Less than
=	75	61	=	Equals
>	76	62	>	Greater than
?	77	63	?	Question mark
@	100	64	@	Commercial at
A	101	65	A	Uppercase A
B	102	66	B	Uppercase B
C	103	67	C	Uppercase D
D	104	68	D	Uppercase E
E	105	69	E	Uppercase E
F	106	70	F	Uppercase F
G	107	71	G	Uppercase G
H	110	72	H	Uppercase H
I	111	73	I	Uppercase I
J	112	74	J	Uppercase J
K	113	75	K	Uppercase K
L	114	76	L	Uppercase L
M	115	77	M	Uppercase M
N	116	78	N	Uppercase N
O	117	79	O	Uppercase O
P	120	80	P	Uppercase P
Q	121	81	Q	Uppercase Q
R	122	82	R	Uppercase R
S	123	83	T	Uppercase S
T	124	84	T	Uppercase T
U	125	85	U	Uppercase U
V	126	86	V	Uppercase V
W	127	87	W	Uppercase W
X	130	88	X	Uppercase X
Y	131	89	Y	Uppercase Y
Z	132	90	Z	Uppercase Z

Graphic:	Octal Value:	Decimal Value	ASCII Abbreviation:	Comments:
[	133	91	[	Opening bracket
\	134	92	\	Reversing slant
]	135	93	]	Closing bracket
or ↑	136	94	↑	Circumflex, up-arrow
or ←	137	95	, UND, BKR	Underscore, back arrow
	140	96	, GRA	Grave accent
a	141	97	a, LCA	Lowercase a
b	142	98	b, LCB	Lowercase b
c	143	99	c, LCC	Lowercase c
d	144	100	d, LCD	Lowercase d
e	145	101	e, LCE	Lowercase e
f	146	102	f, LCF	Lowercase f
g	147	103	g, LCG	Lowercase g
h	150	104	h, LCH	Lowercase h
i	151	105	i, LCI	Lowercase i
j	152	106	j, LCJ	Lowercase j
k	153	107	k, LCK	Lowercase k
l	154	108	l, LCL	Lowercase l
m	155	109	m, LCM	Lowercase m
n	156	110	n, LCN	Lowercase n
o	157	111	o, LCO	Lowercase o
p	160	112	p, LCP	Lowercase p
q	161	113	q, LCQ	Lowercase q
r	162	114	r, LCR	Lowercase r
s	163	115	s, LCS	Lowercase s
t	164	116	t, LCT	Lowercase t
u	165	117	u, LCU	Lowercase u
v	166	118	v, LCV	Lowercase v
w	167	119	w, LCW	Lowercase w
x	170	120	x, LCX	Lowercase x
y	171	121	y, LCY	Lowercase y
z	172	122	z, LCZ	Lowercase z
{	173	123	{, LBR	Opening (left) brace
|	174	124	|, VLN	Vertical line
}	175	125	}, RBR	Closing (right) brace
~	176	126	~, TIL	Tilde
	177	127	DEL	Delete, rubout

APPENDIX B

FORTRAN CHARACTER STRINGS

The data format of strings consists of a two-word object which contains a pointer to the memory location of the string and the number of characters in the string. Bit 15 of the second word indicates odd (right) 1'th byte. The string itself consists of the ASCII values packed two by two into one word. The words are stored in consecutive order. The parity bit (bit 7) is always set to zero.



NORSK DATA A.S
P.O. Box 4, Lindeberg gård
Oslo 10, Norway

COMMENT AND EVALUATION SHEET

MSD-Sort System ND–60.123.02
June 1980

In order for this manual to develop to the point where it best suits your needs, we must have your comments, corrections, suggestions for additions, etc. Please write down your comments on this preaddressed form and mail it. Please be specific wherever possible.

FROM .....
.....
.....