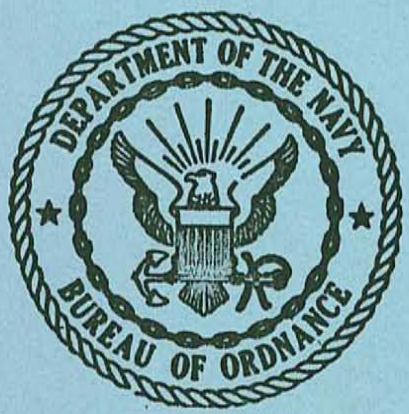
NORC SERVICE ROUTINE AND SUBROUTINE MANUAL



U. S. NAVAL PROVING GROUND
DAHLGREN, VIRGINIA

NORC SERVICE ROUTINE AND SUBROUTINE MANUAL

INTRODUCTION

This catalog contains descriptions of the routines available for use on the NORC. These are divided into two categories: Service routines, which are designed for assistance in manufacturing and debugging programs, correcting tapes, etc.; and routines which are intended for incorporation into a program itself. The pages of this catalog are numbered in a manner intended to make possible the insertion of new routines in the various sections as they become available. Each page is identified by the appropriate sectional designation and the date on which it was entered in the catalog. The "Table of Contents" will continually be revised so as to be a list of currently effective pages.

Programming Research Section
Computation & Exterior Ballistics Lab.
U. S. Naval Proving Ground
Dahlgren, Virginia

SERVICE ROUTINES

Service routines are intended as an aid to the NORC programmer in the manufacturing, debugging and running of programs, and in the handling of data. They are contained on the service tape, which is normally set on tape code 10; the operating staff is familiar with the procedures for their use.

It is desirable, whenever possible, that the programmer prepare a setup sheet (Form No. PRNC-NPG-491) when a service routine requiring extensive tape or switch settings is to be employed. When a need arises, during debugging, for one of the routines such as the memory dump, oral instructions may be given the operator.

SERVICE ROUTINES

TABLE OF CONTENTS

TRACER (Monitor Printer)

Service Routine #1

DESCRIPTION:

This routine prints the location, the instruction, the operands, the result, the modified locations and the modifiers for all or any selected locations. It prints the first iteration only, of a 58 loop, except in special cases, or when otherwise instructed.

OPERATING INSTRUCTIONS:

a. Put machine on 3600 word storage.

b. Put special tracer board in the printer.

c. Put 10 94 0002 0007 0001 into CRT 0001 and give full start. The machine will stop after completing line 3214.

d. For program with standard start and for a complete tracing of every location:

   1. Give another full start.

e. For program with non-standard start and for a complete tracing of every location:

   1. Put location of first instruction to be executed in the T field of 3532.

   2. Put the program start instruction into 3215.

   3. Start in 3215.

f. To run problem up to a certain point under normal programming, and then start tracing:

   1. When desired stopping point is reached, dump the CRT memory on a convenient tape. This should not be a tape used in the problem as it will be positioned wrong. Make a note of modifiers and do <u>not</u> <u>reset</u>.

   2. Read in tracer and give full start. (If modifiers have been reset since the memory block was written, give two oprn starts in place of full start. Put 00 57 4000+m4 2000+m6 m8 in location 3204. Full start in 3202.)

3. At the 3214 stop put the read instruction for the memory block in 3215.

4. Put location of first instruction to be executed in the T field of 3532.

5. Start at 3215.

g. To print only those instructions lying within certain areas, whenever they are operated:

1. Put location of first instruction to be printed in the T field of 3238.

2. Put location of last instruction to be printed in the T field of 3239.

3. Put location of first instruction to be printed of a second section in the T field of 3240.

4. Put location of last instruction to be printed of the second section in the T field of 3241.

h. To print when a specified location is referred to in the R, S. or T field of an instruction in the program. This location can be the results of address modification:

1. Put this location in the T field of 3577.

i. To stop on any desired line of a program before the instruction is operated:

1. Put location of this line in the T field of 3242. (This will give a 71 instruction stop on line 3251.)

j. To repeat an instruction or transfer manually to another instruction:

1. Put the location of this instruction in the T field of 3532.

2. If dependent upon register storage, put what should be in register storage in 3250.

3. Start at 3370.

k. To print all iterations of a 58 loop:

1. Put 00 60 0000 0000 3158 in 3350.

1. If any unforeseen stop occurs, the location of the current instruction being traced is in the T field of 3532.

AUTHOR: Ruth W. Montville

PRINTER FORMAT

Loc #	Instruction					R'	S'	Result	Modified loc. used in: R	S	T	Contents of: M_4	M_6	M_8
xxxx	xx	xx	xxxx	xxxx	xxxx	xxxxxxxxxxxxxxx	xxxxxxxxxxxxxxx	xxxxxxxxxxxxxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx
EXAMPLE														
0313	00	60	1234	0262	0000	9902500000000000	9902500000000000	0000000000000000	0000	0000	0000	0000	0000	0000
0314	50	26	0000	0757	0264	9902500000000000	0003000000000000	9808333333333333	0000	0000	0000	0000	0000	0000
0315	00	70	1644	0757	0319	0001000000000000	0003000000000000	9998000000000000	0000	0000	0000	0000	0000	0000
0316	50	26	1234	0755	0262	9902500000000000	0002000000000000	9901250000000000	0000	0000	0000	0000	0000	0000
0317	50	24	0264	1644	0000	9808333333333333	0001000000000000	9808333333333333	0000	0000	0000	0000	0000	0000
0318	50	26	0000	0755	0264	9803333333333333	0002000000000000	9801666666666666	0000	0000	0000	0000	0000	0000
0319	50	20	5878	0262	1238	0001250000000000	9901250000000000	0001375000000000	1933	0000	0000	0000	0000	0000
0320	00	54	0000	0001	0010	0000000000000000	0000000000000000	0000000000000000	0000	0000	0000	0055	0041	0010

2000 WORD STORAGE TRACER

Service Routine #11

DESCRIPTION:

This routine prints the address, instruction, operands, result, and modifiers of each line of coding whose location number is less than but not equal to some number R. It is possible to enter the routine at some point in the main program (E).

The routine occupies memory locations 1900-2000.

OPERATING INSTRUCTIONS:

a. Use 2000 word storage.

b. Use standard board.

c. Read in program to be traced by using op. starts, or operate it to a desired point.

d. Read in the tracing routine: 1094 1900 0000 0011.

e. Enter: contents of E into 1901

0060 0000 0000 1901 into E

0000 R L 0000 into 1913

where E is the line preceding the first line to be traced, R is the line to which control is to be returned (i.e., the line following the last line to be traced), and L is the first line to be traced (either E+1 or if E is a 60 transfer, the location specified by the T field of E). If E is a 60 transfer make the T field blank when putting it into 1901. The contents of E cannot be a made-up instruction, a conditional transfer, nor can it use register storage.

f. Start on next program line.

PRINTER FORMAT:

See next page.

AUTHOR: Ruth W. Montville

PRINTER FORMAT

Loc. #	Instruction	R'	S'	Result	Not Used	Contents of M_4 M_6 M_8
000000000000xxxx	xxxxxxxxxxxxxxx	xxxxxxxxxxxxxxx	xxxxxxxxxxxxxxx	xxxxxxxxxxxxxxx	0000000000000000	0000xxxxxxxxxxx
EXAMPLE:						
0000000000000313	0060123402620000	9902500000000000	9902500000000000	0000000000000000	0000000000000000	0000005500400021
0000000000000314	5026000007570264	9902500000000000	0003000000000000	9808333333333333	0000000000000000	0000005500400021
0000000000000315	0070164407570219	0001000000000000	0003000000000000	0000000000000000	0000000000000000	0000005500400021
0000000000000316	5020587802621238	0001250000000000	9901250000000000	0001750000000000	0000000000000000	0000005500400021
0000000000000317	0054000000010010	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000005500410010
0000000000000318	0063000000000327	0001750000000000	0001750000000000	0000000000000000	0000000000000000	0000005500410010
0000000000000319	0072164407570313	0001000000000000	9998000000000000	0000000000000000	0000000000000000	0000005500410010
0000000000000313	0060123402620000	9902500000000000	8902500000000000	0000000000000000	0000000000000000	0000005500410010

NOTE: If a 70 through 74 code is used and a transfer occurs S' field will contain the difference and not true S'.

If a transfer does not occur the S' field contains true S'.

5 WORD/LINE MEMORY DUMP

Service Routine #2

DESCRIPTION:

This routine prints the contents of the memory on printer 1.

The first line of the print-out contains the contents of the first seven memory locations. This line is manually printed from the console.

The fourth line contains the contents of the modifiers in the first column (0000, M4, M6, M8) and the contents of memory locations 0018 and 0019 in the third and fourth columns. There is a double space before and after this line of printing.

The remaining lines contain the contents of the memory printed five memory locations per line. Columns two through six contain these five words. Columns one and seven contain the location numbers of the words in column two and six respectively.

The routine omits printing a line in which columns two through six would be zero.

The location numbers in column one are multiples of five from location 0020 on.

OPERATING INSTRUCTIONS:

a. Do not re-set modifiers.

b. Print out first seven words manually.

c. Place service tape on tape code 10.

d. If the machine is operating on "3600 word storage" the routine will automatically print out the contents of all of the 3600 locations.

e. Put 1094 0002 0007 0002 into location 0001 and start in location 0001.

f. Use the standard board.

AUTHORS: Paul M. Botting and William J. Graves

6 WORD/LINE MEMORY DUMP

Service Routine #3

DESCRIPTION:

This routine prints the contents of the memory and modifiers on printer 1. It tests for 3600 word storage and adjusts itself accordingly.

The first line contains the contents of the first 7 memory locations. The first word of the second line shows the contents of M4, M6, and M8 in the R, S, and T fields respectively. The rest of this line has no meaning. The third and fourth lines of printing consist of the contents of locations 0008 through 0021.

The routine prints the contents of locations from 0022 on with the location number in the first word and the contents of that location and of the following five locations in the next six words. Any location containing zero is omitted when it would otherwise be printed at the beginning of a line. In such a case, the next non-zero word is used to begin the printed line and is followed by the five words immediately following it in the memory.

OPERATING INSTRUCTIONS:

a. Do not reset modifiers.

b. Print the first 7 words manually.

c. Place the service tape on t.c. 10.

d. Key 1094 0002 0007 0003 into location 0001 and give a full start.

e. Use the standard board.

NOTE: If the machine is on 3600 word storage but only the first 2000 words are of interest, stop the machine when the contents of M4 as shown on the display panel exceeds 2000.

AUTHOR: M. L. Hagemeyer

CROSS REFERENCE ROUTINE

Service Routine #6

DESCRIPTION:

The cross reference routine sets up a program for listing with the address to the left of each word, and to the right the locations of any instructions which use the address. References to all locations from 0001 to 2000 are included in the list. Output is on tape, from which cards are made for listing on the CCP. Sample output is as follows (program block extends from 0008 to 0010):

ADDRESS							
0004						0008	
0008	00	40	0004	0012	0012	0009	0010
0009	00	60	0009	0008	0010	0009	
0010	00	70	0008	0012	0101	0009	
0012						0008	0010
0101						0010	

OPERATING INSTRUCTIONS:

a. Use 3600 word storage.

b. Put program to be cross referenced on t.c. 03.

c. Put output tape on t.c. 04.

d. Put service tape on t.c. 10.

e. Put "T" tapes on t.c. 01, 02, 11, 12.

f. Set all condition switches to proceed.

g. Option switches are not used.

h. Key 1094 0002 0007 0006 into 0001 and give full start. When program stops in line 0142, key 0394 R S T into 0001, where R and S are the first and last locations the input block is to occupy, and T is the block number.

If only part of the block is to be referenced, key the first and last locations of interest into the R and S fields of 0002.

If the program has been compiled and still has the block containing the correspondence table following the last program block, the cross reference routine can convert the "Address" and "Reference" columns of its output into symbolic form. If this is desired, key the first symbolic address of the segment into the PQ and R fields of 0004. The first digit of the address goes into the 16^{th} digit position of the word.

Start at U+1.

The machine stops in location 0142 after completing a block. If more than one block is to be referenced, enter the read instruction for the next block into location 0001, enter the appropriate information into 0002 and 0004 (optional), and continue at U+1.

i. List output on CCP board. The output tape consists of consecutively numbered blocks terminated by an EOF.

NOTE: This routine places unusual stress on tapes and memory. It is advised that programs consisting of more than 400 lines of instructions be cross-referenced in sections.

STOPS:

0006 - TCF from reading t.c. 10
0041 - TCF from reading t.c. 03
0042 - read check routine
0142 - program stop for entering information
0263 - symbolic address in location 0004 is not in correspondence table
0344 - EOF label found on output tape
0372, 0393, 0430 - record check routine
0330, 0334, 0342, 0346, 0516, 0520, 0537 - EOF not written properly
0364 - EOF sensed while verifying written block

AUTHOR: John H. Walker

REVIEWED BY: M. L. Hagemeyer

MEMORY COMPARER

Service Routine #5

DESCRIPTION:

This routine makes a word-by-word comparison of the data currently in the memory and the data loaded into the memory by the original program and prints information concerning those addresses where a difference is found.

OPERATING INSTRUCTIONS:

a. DO NOT RESET MEMORY

b. Log displays, indicators, etc. on "Console Information" form.

c. Set switch 68 (TCF) to "stop".

d. Place service tape on tape code 10.

e. Place "temporary" tape on tape code 01.

f. Use standard board on printer #1.

g. Print first seven words manually.

h. Type 1094 0002 0007 0005 into location 0001 and start.

i. The routine stops in location 0012 after printing several lines and writing the memory on the T tape. If the program block corresponding to the contents of the memory can be read in by a standard start, continue at U+1 (0013). Otherwise, read the program into the proper locations and start at location 0003. If reading in the program changes location 0003, type

1094 0004 0028 0000

into location 0003 before starting there.

DESCRIPTION OF PRINT-OUTS:

The routine prints the contents of locations 0001 through 0028 in the first four lines of printing without making any comparison. It then makes a word-by-word comparison of the remainder of the

memory with the original program, starting with location 0029. When the memory word and the program word differ, the address and the differing words are printed. Each line of printing contains two such sets, arranged as follows:

1	2	3	4	5	6	7
Address	Program Word	Memory Word	-	Address	Program Word	Memory Word

REMARKS:

This routine may be used after the regular memory print-out has been performed. In this case the contents of locations 0001 through 0028 printed by this routine will be meaningless. This routine cannot be restarted. This routine can be used with 2000 word storage only.

AUTHOR: M. L. Hagemeyer

TAPE COPYING ROUTINE

Service Routine #9

DESCRIPTION:

This routine copies blocks from one tape to another. This routine cannot be used unless S-R+1 of the EOB word will give the block length.

OPERATING INSTRUCTIONS:

a. Use 3600 word storage.

b. Put tape to be copied on t.c. 01.

c. Put output tape on t.c. 02.

d. Put service tape on t.c. 10.

e. Use 10 94 0002 0007 0009 to start.

SWITCHES:

Set all condition switches to proceed.

Option Switches:

74__ "stop" to stop after each block is read in, otherwise on transfer or off.

75__ "stop" to stop after each block has been written, otherwise on transfer.

DO NOT LEAVE 75 ON OFF

76 and 77 Not used.

78__ "off" to use original block numbers.

"transfer" to renumber the blocks in consecutive order starting with 0001.

79__ "off" to copy with a 91 code.

"transfer" to copy with a 90 code.

STOPS:

0001: 61 0000 0000 0006

An EOF has been reached and copied.

0005: 71 0001 2001 0000

The machine is not on 3600 word storage.

0005: 61 8000 0000 0000

A TCF has occurred on reading in the copying routine.

0046: 61 0000 0000 0040

A tape check failure has occurred in reading a block. The "read" instruction is in location 0016 and register storage, and the block number is in location 0034. To try reading this block again push start button. To omit copying this block start in location 0007. This will not disturb the consecutive numbering of copied blocks because the counter (location 0004) is not incremented until after the TCF has been sensed for.

REMARKS:

a. Location 0003 contains the block length minus one, of the block being copied, in the S field.

b. Location 0004 contains the block counter in the T field.

c. Location 0011 contains the increment for the block counter (0000 0000 0000 0001 normally).

d. Location 0016 contains the read instruction.

e. Location 0051 contains the first word of the block being copied.

f. If 74 is on stop, the program will stop in location 0033 with the write instruction in register storage and location 0034.

g. If 75 is on stop, the program will stop in location 0039 with the EOB word, as it appears in the copied block, in register storage.

h. Regardless of the setting of 78, the EOF's will be numbered as they appear on the tape to be copied.

i. If 78 is on trasfer, one may start numbering the copied blocks at some number (n) other than 0001 by op-starting through location 0006 and keying n-1 into the T field of location 0004. One may increment the block number by some number other than 0001 by keying this new increment into the T field of location 0011. One may skip r block numbers after copying block p by leaving 75 on stop until block p has been copied and then keying p+r into the T field of location 0004 and continuing. If one copies n blocks and then copies through an EOF with 78 on transfer, the block numbering will continue as follows: . . . . n-2, n-1, n, EOF, n+1 . . . . To start numbering the blocks at 0001 after the EOF, clear location 0004 before starting again.

AUTHOR: Paul M. Botting

TAPE COPYING AND CORRECTION ROUTINE

Service Routine #10

DESCRIPTION:

This routine can be used for simple copying of one tape to another or for copying selected blocks from one or two tapes and making changes in them. This routine cannot be used unless S-R+1 of the EOB word gives the block length.

1. Simple Copying:

   To copy from t.c. 01 to t.c. 02 without making changes, follow the "Operating Instructions" below, omitting c. and d. Set all option switches to "off". The routine stops in location 0001 after copying an EOF. To continue, push the start button.

2. Copying with Changes:

   Given input of the proper form, the routine can copy all or selected blocks from one or two tapes and change words within the blocks. Input can be on tape, or it can be keyed in at the console. It is composed of the desired combination of "Changes and Corrections", as listed below.

   When the changes and corrections are supplied to the routine on tape, they must be in fifty-word blocks, whether or not this ends in the middle of a correction. The last block may be shorter if necessary. The blocks must be numbered in ascending order starting with 0001 and must be followed by an EOF.

   If very few lines of changes and corrections are needed, they may be entered at the console. In this case, set option switch 77 to "stop". When the 77 stop occurs, enter the words of changes in consecutive locations beginning at 3550. Start at V (0109).

OPERATING INSTRUCTIONS:

a. Use 3600 word storage.

b. Put tape to be copied first on t.c. 01.

c. Put tape to be copied second, if used, on t.c. 04.

d. Put correction tape, if used, on t.c. 03.

e. Put output tape on t.c. 02.

f. Put service tape on t.c. 10.

g. Use 1094 0002 0007 0010 to start.

SWITCHES:

Set all condition switches to "proceed".

Option switches:

74__ "off" for copying without changes.

"transfer" for making changes, either from tape or from the console.

75__ "off" to write with 91 code.

"transfer" to write with 90 code.

76__ "off" to stop after TCF from reading input tape.

"transfer" to delete 4000 word space on output tape after TCF from reading (this leaves space to insert block later).

77__ "off" when correction tape is to be used or when 74 is "off".

"stop" when corrections are to be keyed in.

78__ "off" for uninterrupted copying.

"stop" to stop after each block has been written.

79__ Not used.

CHANGES AND CORRECTIONS:

If correction 1000 is used, it should appear first, followed by any of the corrections 1--8 in order. These indicate to the routine which blocks are to be copied and what changes are to be made in <u>all</u> blocks copied. 9--12, which make changes in individual blocks, should be ordered the same as the <u>output</u> blocks to which they apply. No more than one of the corrections 2, 3, 4, 5 may be used in one run. 6 and 7 cannot be used together. 4, 5, 7-12 may be any desired length.

1000	____	a	b	Copy through block c of t.c. 01; then copy blocks a through b of t.c. 04;
____	____	c	d	then read 01 forward to block d and continue copying from there. If b is zero, tape 04 is copied from a through the next EOF. If d is zero, copying stops after the required portion of tape 04 has been copied. CAUTION: This does not work if there is a block b on tape 01 before block c. The routine will only take one group of blocks from t.c. 04. It will not go back and forth between tapes.
0001	____	____	n	Copy n EOF's. The routine numbers EOF's consecutively. If one EOF is to be copied, this line may be omitted.
0002	____	a	n	Copy every n^{th} block, starting with the a^{th} block. If a is zero, the n^{th} block is the first copied. If several EOF's are to be copied, they are copied but are not counted as blocks.
0003	____	a	n	Omit every n^{th} block, starting with the a^{th} block. Remarks for preceding correction apply here.
0004	a	b	c	Copy blocks a, b, c ... i in this order. Blocks need not be listed in the order
____	d	e	f	in which they appear on input tapes, and the same blocks are copied as often
____	g	h	i	as their numbers are listed. The list may include any number of blocks, but there can be only one EOF.
0005	a	b	c	Omit blocks a, b, ... f. Blocks must be listed in the order in which they
____	d	e	f	appear on <u>input</u> tapes. Any number of EOF's may appear.
0006	____	____	____	Number blocks consecutively, beginning with 0001.
0007	____	a	b	Change block numbers from a to b, c to d, etc.
____	____	c	d	

		loc.	loc.	
0008	base	a	b	Change the contents of locations a, b, c, d in every block, substituting words (a), (b), (c), (d). "Base" is starting address of blocks; if zero, then "location a" should read "a^{th} word", etc.
(a)				
(b)				
____	base	loc. c	loc. d	
(c)				
(d)				

In the following corrections, n is the block's original number.

0009	base	____	n	Change the contents of locations a, b, ... e in block n to words listed. Base has same meaning as above.
____	loc. a	loc. b	loc. c	
(a)				
(b)				
(c)				
____	loc. d	loc. e	____	
(d)				
(e)				

0010	base	____	n	Change m consecutive words in block n, starting with word a, substituting (1), (2), ... (m).
____	____	a	m	
(1)				
(2)				
.				
.				
.				
(m)				

0011 base ____ n Shorten block n by removing words a, b, ... d.

	loc. a	loc. b	loc. c
____			

	loc. d		
____		____	____

0012 base ____ n Lengthen block n by inserting u in location a, v in location b, etc. Locations should be arranged in ascending order and should be figured as the locations in the expanded block.

	loc. a	loc. b	loc. c
____			

(u)
(v)
(w)

	loc. d	loc. e	
____			____

(x)
(y)

STOPS:

0001: Standard stop. Start at V to copy through next EOF.

0005: TCF from reading program block.

0051: Machine not on 3600 word storage.

0077: TCF from writing on 02.

0084: TCF from reading 01 or 04.

0098: TCF from reading 03. Do not continue from here.

0162: TCF from reading 01 or 04.

0048: An EOF label has been detected during writing on 02.

0092: EOF could not be written on 02.

PRINTING:

Write instruction in 0001: TCF from writing on tape designated in 0001.

End of block word in 0001 and 1111 in PQ of 0002: Read TCF which did not occur upon re-reading.

End of block word in 0001, delete instruction in 0002: TCF from reading, followed by deleting long block on output tape.

AUTHOR: M. L. Hagemeyer

TAPE COMPARISON ROUTINE

Service Routine #4

DESCRIPTION:

This routine compares the information contained on two tapes and prints appropriate indications whenever corresponding block numbers, block lengths, or words, are not identical. Blocks to be compared must have EOB words such that S'-R'+1 gives the correct block lengths.

OPERATING INSTRUCTIONS:

a. Use 3600 word storage.

b. Set all condition switches to "proceed".

c. Set switch 74 to "off" for continuous comparison; to "stop" in order to stop after comparing each pair of blocks. No other option switches are used.

d. Place the two tapes to be compared on tape codes 01 and 02; the service tape on tape code 10.

e. Place a "temporary" tape on tape code 03. (This is necessary only if any of the blocks contain more than 1600 words.)

f. Place standard board on printer #1.

g. Begin by typing 1094 0002 0007 0004 into 0001 and "start".

STOPS:

a. In 0001 - after reading EOF mark on one of the tapes (see explanation below). Usually indicates end of comparison. May be restarted.

b. In 0047 - after comparing each pair of blocks, if switch 74 is set to "stop". May be restarted at U+1 (0048).

c. In 0181 - if memory has not been set to 3600 words. Set memory properly and start over.

DESCRIPTION OF OPERATION AND PRINTING:

A block count, which starts at 0001 and is increased by 0001 each time a block is read from tape 01, appears in the first word of each seven-word line of printing. The block number appears in

the second word. If the corresponding block on tape 02 has a different number, it is printed in the third word (example 1), and no further comparison of this pair of blocks is made.

```
          1                   2                   3          4  5  6  7

1)  xxxx000000000000  xxxx000000000000  xxxx000000000000  -  -  -  -

    block             block             block
    count             number            number
                      (01)              (02)
```

If a block on tape 01 differs in length from the corresponding block on tape 02, the two block lengths appear in the sixth and seventh words (example 2), and the length of the block on tape 01 determines the number of words to be compared.

```
          1                   2                3    4    5

2)  xxxx000000000000  xxxx000000000000  -    -    -

    block             block
    count             number


    xxxx000000000000  xxxx000000000000

    length            length
    (01)              (02)
```

The words of each pair of blocks are compared in order, and any two words which are not identical are printed in the sixth and seventh words. A word count is kept for each pair of blocks. This word count, which starts at 0001 and is increased by 0001 as each pair of words is compared, appears in the fourth word. The block count and block number are in the first two words (example 3).

```
          1                   2                3    4

3)  xxxx000000000000  xxxx000000000000  -    xxxx000000000000

    block             block                  word
    count             number                 count

    5         6                    7

    - xxxxxxxxxxxxxxxx  xxxxxxxxxxxxxxxx

      word              word
      (01)              (02)
```

If a tape check failure occurs (and switch 68 is set to "proceed") a line is printed with the tape code (01, 02, 03) in the P field of the fifth word (example 4). The routine then proceeds to the next pair of blocks. If switch 68 is set to "stop", the calculator stops immediately when reading the block which fails.

	1	2	3	4	5	6	7
4)	xxxx000000000000	-	-	-	0x00000000000000	-	-
	block count				tape code		

When an end of file is sensed on either tape, the tape code on which it occurred is printed in the Q field of the 5th word (example 5). If an EOF is sensed in the same position on each tape, two lines are printed. In any of these cases the machine stops after printing and may be restarted immediately if desired.

	1	2	3	4	5	6	7
5)	xxxx000000000000	-	-	-	000x000000000000	-	-
	block count				tape code		

REMARKS:

a. This routine will compare blocks containing up to 2200 words.

b. The two tapes to be compared need not be rewound before operating this routine. They may be positioned just before the first block to be compared on each tape.

AUTHOR: M. L. Hagemeyer

TAPE PRINT-OUT
(7 words to a line)

Service Routine #7

DESCRIPTION:

This routine prints the words contained on a tape, including the EOB words, on the monitor printer. There is a space separating each block printed.

In printing, words of sevens follow the end of block word when necessary to fill out a line of printing.

On an end-of-file indication, the end of file word is printed in print locations 0001 and 0007 with words of sevens in print locations 0002 through 0006.

OPERATING INSTRUCTIONS:

a. Place the tape which is to be printed on tape code 01 and the service tape on tape code 10. Position the tape to be printed in front of the first block to be printed.

b. Set the machine for 3600 word storage. (If 2000 word storage is used there will be a stop in location 0005.)

c. Set switches 64, 65, 66 and 67 to proceed and 68 to stop.

d. To print blocks without stopping set switch 74 to transfer. To stop after printing each block set switch 74 to stop. DO NOT SET SWITCH 74 TO OFF.

e. Set switches 75, 76, 77, 78 and 79 to stop. (They are not used.)

f. Use standard board on Monitor Printer 1.

g. Type 1094 0002 0007 0007 into location 0001 and start.

h. Standard stop is on an end-of-file indication.

i. The routine may be restarted in location 0008 at any time.

NOTE: This routine cannot be used unless S-R+1 of the EOB word will give the block length.

AUTHOR: Orbie L. Jones

TAPE PRINT-OUT
(4 words to a line)

Service Routine #8

DESCRIPTION:

This routine prints the words contained on a tape, including the EOB words, on the monitor printer four words to a line. The four words are printed in locations 0002 through 0005. There is a space separating each block printed.

The PQ field of print location 0001 contains the card count. The T field of print location 0001 contains the address of the first word of each line printed beginning with the number in the R field of the beginning-of-block word. Zeros are in print locations 0006 and 0007.

In printing, words of sevens follow the end of block word when necessary to fill out a line.

On an end-of-file indication, the end of file word is printed in print locations 0002 and 0005 with words of sevens in print locations 0003 and 0004.

OPERATING INSTRUCTIONS:

a. Place the tape which is to be printed on tape code 01 and the service tape on tape code 10. Position the tape to be printed in front of the first block to be printed.

b. Set the machine for 3600 word storage. (If 2000 word storage is used there will be a stop in location 0005.)

c. Set switches 64, 65, 66 and 67 to proceed and 68 to stop.

d. To print blocks without interruption set switch 74 to transfer. To stop after printing each block set switch 74 to stop. DO NOT SET SWITCH 74 TO OFF.

e. Set switches 75, 76, 77, 78 and 79 to stop. (They are not used.)

f. Use the standard board on Monitor Printer 1.

g. Type 1094 0002 0007 0008 into location 0001 and start.

h. Standard stop is on an end-of-file indication.

i. The routine may be restarted in location 0008 at any time.

NOTE: This routine cannot be used unless S-R+1 of the EOB word will give the block length.

AUTHOR: Orbie L. Jones

THE COMPILER AND COMPILER CODING CORRECTOR

Service Routine #9000

DESCRIPTION:

*The Compiler is an automatic coding technique designed to minimize the time spent on coding of problems for solution on NORC. It converts unordered symbolic or relative codes to ordered machine (absolute) codes, generates and assembles subroutines from a library file, provides a variety of other services for the coder, and writes the final coding on tape ready to be run on the calculator.

The Compiler Coding Corrector, a part of the Compiler routine, revises the Compiler input tape according to information which is put on the Compiler output tape. It is called during compilation if switch 75 is on transfer.

OPERATING INSTRUCTIONS:

a. Put the symbolic coding (Compiler input) on t.c. 04.

b. Tape code 02 contains the corrections for the correction routine, if there are any, and after compilation will contain the absolute coding of the problem (Compiler output).

c. Put the Service Tape on t.c. 09.

d. Put T tapes on t.c.'s 01, 03, 11 and 12.

e. Put switch 75 on transfer if corrections are to be made.

f. Put switch 76 on transfer to obtain a print-out of the correspondence table.

g. To be safe put all other switches on off.

h. Key 0994 0002 0007 9000 into location 0001 and start.

RESTARTING INSTRUCTIONS:

a. If one line has been printed switch 75 must be on off, then a standard restart can be taken.

b. If no printing has been done a standard restart can be taken.

*For complete information see NPG Report No. 1374.

FORM OF CORRECTIONS: (Use compiler coding sheets)

Control Word:

R - absolute field contains the block number of the correction

S - absolute field contains the card number of the correction (the card number of the first correction if more than one correction is made by the control word)

T - absolute field contains:

(a) 0000 if a single correction

(b) # of additions in sequence

(c) 2000 + # of changes in sequence

(d) 4000 + # of cards to be removed

Each control word is followed immediately by the corrections or additions it is to make. All corrections are made in the order in which they appear on the 02 tape.

Example:

	C	location	P	Q	R	S	T
a. Control word					0003	0027	
Change word	01	621.0	08	68	621.0		10.0020
b. Control word					0002	0005	4003
c. Control word					0002	0005	2002
Change word					0001	0020	
			04	58			600.2
Change word		630.7	16	60		401.1	

The above example:

a. Corrects card 27 of block 3.

b. Removes cards 5, 6 and 7 of block 2.

c. Corrects cards 5 and 6 of block 2.
(Cards 5 and 6 were actually cards 8 and 9 prior to correction b)

Printed line from monitor printer:

a. 09 96 0002 0007 0000 in word seven if corrector routine has been used.

b. Zero in word seven if corrector routine has not been used.

Error stops:

a. Errors that the compiler corrector routine can detect in the compiler set-up are as follows:

Location	
0019	Switch 74 on transfer. (To restart, put 74 on off and go to U + 1.)
0020	Both switches 76 and 77 on transfer. (Check set-up sheet.)
0021	Machine not on 2000 word storage. (To restart, put machine on 2000 word storage and go to U + 1.)

b. Coding errors that the compiler corrector routine can detect in the corrections are indicated by a coded machine stop and a line of special function printing on the monitor printer. The first word of the printed line will contain one of the error codes listed below:

19: Control and list words of the corrections are improperly grouped.

20: Control word calls for adding cards other than at the end of a block. (Word 2 contains the block number.)

21. Control word calls for removing too many cards at the end of a block. (Word 2 contains the block number.)

Limitations:

a. Only one block of corrections is allowable. It must not exceed 120 cards.

b. This routine will not add or delete blocks.

c. The corrections on the 02 tape are destroyed after they have been made.

d. The maximum allowable block length for compiler input is the equivalent of 120 cards.

e. Cards may be added only to the end of a block.

f. Corrections need not be ordered according to block number.

AUTHORS: W. J. Graves and M. J. Burkhart

GRAPH PLOTTING ROUTINE

Service Routine #12

DESCRIPTION:

This routine converts tape output to a form which can be used by the Benson-Lehner Electro-plotter. It selects pairs of x's and y's from the input tape and scales them so that the maximum value of x and maximum value of y on the input tape will equal respectively a prescribed maximum scaled value of x and maximum scaled value of y. It records the output on tape which can be punched on cards. All of the cards except the BOB cards and EOB cards will be punched as:

Word I	Word II	Word III	Word IV
00X0 0XXX 00Y0 0YYY	----------	----------	----------

sign for x's

sign for y's

The card with the beginning of block word on it will contain three zero words and this card may be discarded.

The EOB word will be the second word of the last card of a block, and since the first word of this card is to be used, this card must be saved.

The input block can be any size up to 2275 words.

There may be two output blocks per input block, depending on the size of the input block. The output blocks will be numbered consecutively, starting with block number 0001. If the input block is approximately 1650 words long it will be broken into two output blocks.

OPERATING INSTRUCTIONS:

a. Put machine on 3600 word storage.

b. Put service tape on tape code 10, input tape on tape code 01, and output tape on tape code 02.

c. Put the instruction 1094 0002 0007 0012 into location 0001.

d. Set switch 74 on stop.

e. Start at location 0001.

f. The machine will stop on 74 code at location 0014. Then put the following information into locations 0008 through 0013, and location 0150:

loc 0008 min x if known or
0020 ---- ---- ---- if values for locations 8, 9, 10 and 11 are not known, and locations 9, 10 and 11 will not be used. Min x is the smallest x on the input tape.

loc 0009 max x if known. Max x is the largest x on the input tape.

loc 0010 min y if known. Min y is the smallest y on the input tape.

loc 0011 max y if known. Max y is the largest y on the input tape.

loc 0012 (no. of prs. x's and y's if known) (no. x_o) (no. y_o) (loc x_{i+1} - loc. x_i)

If the number of pairs of x's and y's are known, put switch 75 on transfer. If not known, put 75 on off.

No. x_o and no. y_o are the no. of words before and including x_o and y_o respectively on the input tape.

The number of locations between the consecutive x's must be the same as the number of locations between the consecutive y's.

loc 0013 (min scaled x) (max scaled x) (min scaled y) (max scaled y)

In loc 0013 the first digit in each parenthesis is the sign as either 0 or 1 and the remaining 3 digits make up the number itself. (i.e., the values in loc 0013 range from -999 to +999.)

loc 0150 Graph paper conversion factor, exponential style. This routine contains the conversion factor of 0.7 in loc. 0150. This is the ratio of the length of the y axis divided by the length of the x axis of the graph paper used. This ratio is used in order to graph circles as circles, and with it the only portion of the graph paper used is the largest square possible with its center as the center of the paper.

Dietzgen graph papers #340R-10 and #340R-20 use this factor. Dietzgen graph paper #340D-34 uses the factor two-thirds in which case 0000 6666 6666 6667 is put into loc 0150 after the program is read in (by setting sw 74 on "stop" and then putting the factor into loc. 0150).

g. If desired to change loc 0008 through 0013 and/or loc 0150 for use with the next block, leave switch 74 on "stop". If not, put switch 74 on "off".

h. Start at U+1. If desired to write EOF on output tape and end problem immediately, start at 0131 instead of U+1.

i. The routine scales x's and y's with the formulas:

$$x_s = a_x x + b_x$$

$$y_s = a_y y + b_y$$

Where x_s = scaled x, and y_s = scaled y. The scaling factors will be printed on the monitor printer from the first four print locations as (a_x) (b_x) (a_y) (b_y).

NON-STANDARD STOPS:

Stop	Reason
loc 0006	TCF for tape on tape code 10.
loc 0141	TCF for tape on tape code 02 after tapes are rewound. Program will start again at loc 0014 on pushing "start" button.
loc 0134	No EOF verified on tape code 02.
loc 0142	loc x_o = loc y_o .

AUTHOR: Charles H. Juergensmeyer

SUBROUTINES

The routines described in this section are generated by the compiler to perform one or more specific functions, and are incorporated by it into the problem-solving program. The specific functions to be performed are indicated by a set of subroutine parameters. Parameters which determine the structure of the generated subroutine are called preset parameters; those parameters which do not affect the structure of the generated subroutine, but are used by it at the time of execution of the problem-solving program, are called program parameters. Use of preset parameters allows the compiler to generate an efficient routine specialized to the needs of the particular problem.

The instruction which causes the compiler to generate the routine and adapt it for use in the particular program must be indicated by a C code of 01 on the Compiler Coding Sheet (this instruction corresponds to that indicated for "location M" in the individual descriptions which follow). The number of the subroutine desired is always placed in the symbolic portion of the T field of this instruction. (The subroutine number will be converted by the compiler to the address at which the routine is located.) The absolute portion of the T field is called the identification tag (described below), and is usually left blank. If any preset parameters are required, they are indicated on the succeeding lines with C codes of 02. These preset parameter words do not appear in the final compiled program, therefore, they are not subject to programmed modification. Following the preset parameters (if any), the program parameters are indicated; the C code is the normal 00(blank). These words will appear in the compiled program. The specific form for indicating the input parameters for each routine is given in its description.

Within a segment, each subroutine is compiled into the program only once, even though multiple references may be made to it, as long as the identification tag (the absolute portion of the T field of the reference instruction) is always left blank. However, the compiler will insert one version of the subroutine for each different identification tag. This is sometimes useful when it is necessary, in a segment, to generate more than one version of a routine requiring preset parameters. Each different version (identified by its tag) may be referred to several times but the preset parameters required need only be indicated once for each version following the reference with the lowest symbolic address for that version. As subroutines are generated for each segment of a program independently, it is necessary to indicate preset parameters for each version of a routine once in each segment requiring the routine.

If a routine requires program parameters, they must be indicated following all references to the routine.

SUBROUTINES

TABLE OF CONTENTS

SIN x COS x I

Routine No. 10.0001

This routine is not used in new codings. It is left in the library for use in codings which used it in the past and which may have to be recompiled in the future.

SIN x COS x II

Routine No. 10.0013

TRANSFER TO ROUTINE:

Location M:	0060	M	0000	routine
Location M+1:	B_s	B_c	0000	A

DESCRIPTION:

The argument x (in radians) is in location A. Sin x and Cos x are computed and stored in locations B_s and B_c respectively. Control is returned to location M+2.

MEMORY LOCATIONS: 49

AVERAGE TIME: 12 milliseconds

ACCURACY:

Sin (0) and Cos ($\pi/2$) are exactly zero; sin ($\pi/2$) and cos (0) are exactly one; and sin ($\pi/6$) is exactly one half. For $10^{-42} \leq x \leq 10^{-8}$ sin x = x and cos x = 1.

For $10^{-8} \leq x \leq 1$, sin x is either accurate to 13 digits or is one too large in the 13^{th} place (a mean error of + .3 in the 13^{th} digit). Cos x has a mean error of + 1.2 in the 13^{th} digit in this range.

METHOD:

$x - n\pi = q$ where $n = \frac{x}{|x|}$ $\left[\text{integral part of } \left(\frac{|x|}{\pi} + \frac{1}{2}\right)\right]$

q will lie between $-\pi/2$ and $+\pi/2$.

$(-1)^n \sin x = q - \frac{q^3}{3!} + \frac{q^5}{5!} + \cdots\cdots$

since $\cos x \equiv \sin (\pi/2 - |x|)$

the cycle is repeated with x replaced by $(\pi/2 - |x|)$

AUTHOR: A. V. Hershey

ARCTAN x

Routine No. 10.0003

TRANSFER TO ROUTINE:

Location M:	0060	M+1	0000	routine
Location M+1:	B	C	0000	A

DESCRIPTION:

The argument x is in location A. Arctan x is computed and stored in location B. Control is returned to location C.

Arctan x is in radians with

$$-\frac{\pi}{2} < \arctan x < \frac{\pi}{2} .$$

MEMORY LOCATIONS: 44

AVERAGE TIME: 6 milliseconds

ACCURACY:

The result is accurate to within two units in the last digit position.

METHOD:

let $|x| = w$

then $\arctan w = u + v + s$

where $s = -\frac{u^3}{3} + \frac{u^5}{5} - \frac{u^7}{7} + \cdots\cdots$

and $u = w$, $v = 0$ when $0 \leq w < 1/2$

$u = \frac{w-1}{w+1}$, $v = \frac{\pi}{4}$ when $1/2 \leq w \leq 2$

$u = -\frac{1}{w}$, $v = \frac{\pi}{2}$ when $w > 2$.

If $x < 0$, the relation $\arctan x = -\arctan w$ is used.

REMARKS:

The average number of terms required in the sum S is twelve.

AUTHOR: John H. Walker

ARCTAN $\frac{y}{x}$

Routine No. 10.0008

TRANSFER TO ROUTINE:

Location M:	0060	M+1	0000	routine
Location M+1:	B	C	A_y	A_x

DESCRIPTION:

The argument y is in location A_y, argument x in location A_x. Arctan $\frac{y}{x}$ is computed and stored in location B. Control is returned to location C.

Arctan $\frac{y}{x}$ is in radians with

$$0 \leq \arctan \frac{y}{x} < 2\pi .$$

Arctan $\frac{y}{x}$ will lie in the quadrant of a vector whose components are x and y.

MEMORY LOCATIONS: 63

AVERAGE TIME: 6 milliseconds

ACCURACY:

The result is accurate to within one unit in the last digit position.

METHOD:

For $\frac{y}{x} > 0$; $\arctan \frac{y}{x} = T + u + v + s$

For $\frac{y}{x} < 0$; $\arctan \frac{y}{x} = T - (u + v + s)$

$T = (n - 1) \frac{\pi}{2}$, where n is the quadrant of the geometric representation of y/x .

let $\left|\frac{y}{x}\right| = W$; then $s = -\frac{u^3}{3} + \frac{u^5}{5} - \frac{u^7}{7} + \ldots$

where $u = W \quad v = 0 \quad$ when $0 \leq W < \frac{1}{2}$

$$u = \frac{W-1}{W+1} \quad v = \frac{\pi}{4} \quad \text{when } \frac{1}{2} \leq W \leq 2$$

$$u = \frac{-1}{W} \quad v = \frac{\pi}{2} \quad \text{when } W > 2$$

When $x = 0$, $\arctan \frac{y}{x} = 0$ or π depending on the sign of y.

REMARKS:

The average number of terms required in S is twelve.

A stop occurs when both x and y are zero.

AUTHORS: William J. Graves and David W. Ross

10^x

Routine No. 10.0002

TRANSFER TO ROUTINE:

Location M:	0060	M	0000	routine
Location M+1:	A	0000	0000	B

DESCRIPTION:

The argument x is in location A. 10^x is computed and stored in location B. Control is returned to location M+2.

x is restricted to the range $-30 \leq x \leq 30$

MEMORY LOCATIONS: 40

AVERAGE TIME: 10 milliseconds

ACCURACY: The result is accurate to 5 units in the twelfth digit.

METHOD:

let i = integral part of x

f = fractional part of x

$$10^x = 10^{i+f} = 10^i \cdot 10^f = 10^i \cdot e^y$$

where $y = f \log_e 10$

$$e^y = 1 + y + \frac{y^2}{2!} + \frac{y^3}{3!} + \cdots\cdots$$

REMARKS:

A program stop occurs on the 30^{th} line of the routine if $|x| > 30$.

AUTHOR: David F. Eliezer

10^x , e^x

Routine No. 10.0009

TRANSFER TO ROUTINE:

Location M:	0060	M+1	0000	routine
Location M+1:	B	C	D	A

DESCRIPTION:

The argument x is in location A. The desired exponential is computed and stored in location B. If D = 0000, e^x will be computed and if D = 0001, 10^x will be computed. Control is returned to location C.

MEMORY LOCATIONS: 51

AVERAGE TIME: 9 milliseconds

ACCURACY:

The results are accurate to within 5 units in the last digit position.

METHOD:

$e^x = 10^p$ where $p = x \log_{10} e$

$10^x = 10^p$ where $p = x$

$10^p = 10^{i+f}$; $f < 0$

where $i = [p] + 1$

let $S = 10^{f/2} = e^u = 1 + u + \frac{u^2}{2!} + \frac{u^3}{3!} + \cdots + \frac{u^n}{n!}$

where $u = \frac{f}{2} \log_e 10$

and $n = 10 - [10u]$

then $10^p = 10^i S^2$

REMARKS:

If $p < -42$, zero will be stored in location B of call line.

If $p > 54$, there will be a stop in the return line of the routine which is 36 locations below the first location of the routine. If you have a value of $10^x (e^x)$ that you would like to use, put it into register storage, do the 61 instruction over again and then do a full start. The computed value is no good.

AUTHOR: John H. Walker

LOG_e x, LOG_{10} x

Routine No. 10.0005

TRANSFER TO ROUTINE:

Location M:	0060	M+1	0000	routine
Location M+1:	A	B	C	D

DESCRIPTION:

The argument x is in location A. If D equals 0000, $\log_e$ x will be calculated and stored in location B. If D equals 0001, $\log_{10}$ x will be calculated and stored in location B. Control is returned to location C.

x is restricted to the range $10^{-42} \leq x \leq 10^{30}$. Any x not in this range will give a wrong answer.

MEMORY LOCATIONS: 68

AVERAGE TIME: 13 milliseconds

ACCURACY:

Log_e x is accurate within 3 units in the 13th digit.

METHOD:

let $x = W \times 10^P \qquad (1 \leq W < 10)$

then $\log_{10} x = p + \log_{10} W$

and $\log_e x = (p + \log_{10} W) \log_e 10$

where $\log_{10} W = \log_{10} a + 2 \log_{10} e \left[\frac{w-a}{w+a} + \frac{1}{3}\left(\frac{w-a}{w+a}\right)^3 + \frac{1}{5}\left(\frac{w-a}{w+a}\right)^5 + \cdots\right]$

and $a = 1$ if $W \leq \sqrt{10}$

$a = \sqrt{10}$ if $W > \sqrt{10}$ and $p \neq -1$

$a = 10$ if $W > \sqrt{10}$ and $p = -1$

REMARKS:

No more than 25 terms in the expansion of $\log_{10} W$ are required.

A program stop occurs if $x \leq 0$.

AUTHOR: John H. Walker

SQUARE ROOT I

Routine No. 10.0004

TRANSFER TO ROUTINE:

Location M:	0060	M+1	0000	routine
Location M+1:	B	C	0000	A

DESCRIPTION:

The argument x is in location A. The square root of x is computed and stored in location B. Control is returned to location C.

MEMORY LOCATIONS: 26

AVERAGE TIME: 4 milliseconds

ACCURACY:

The result is accurate to within one unit in the last digit position.

METHOD: Newton-Raphson

Let $\sqrt{x} = u$

then u is obtained from the iteration equation

$$u_{i+1} = u_i + \frac{1}{2}\left(\frac{x}{u_i} - u_i\right)$$

$u_o = 1 \times 10^{\left[\frac{p+1}{2}\right]}$ where p is the exponent of 10 in the normalized x.

REMARKS:

The average number of iterations performed is five. No iterations are required if $x = 0$.

A program stop occurs on the 21st line of the routine if $x < 0$. If one wishes to assume a value for $\sqrt{x}$ put this value into register storage (R_1 and R_2) and start on the 20th line.

AUTHOR: John H. Walker

SQUARE ROOT II

Routine No. 10.0027

TRANSFER TO ROUTINE:

Location M:	0060	M+1	0000	routine
Location M+1:	B	C	D	A

DESCRIPTION:

For the square root of x:

D must be 0000. A is the location of the argument x. The square root of x is computed and stored in location B. Control is returned to location C.

For the square root of $(1 - x^2)$:

D must be 0001. A is the location of the argument x. The square root of $(1 - x^2)$ is computed and stored in location B. Control is returned to location C.

For the square root of $(x_1^2 + x_2^2 + x_3^2 + \ldots + x_n^2)$:

D must be (n), where (n) is the total number of argument values. The consecutive locations A, A+1, A+2, ..., A+n-1 contain the argument values x_1, x_2, x_3 ... x_n. The square root of $(x_1^2 + x_2^2 + \ldots x_n^2)$ is computed and stored in location B. Control is returned to location C.

MEMORY LOCATIONS OCCUPIED: 48

AVERAGE TIME: 4 milliseconds

METHOD: Newton-Raphson Iterative Process

$$\text{Let } w = \begin{cases} x \\ 1 - x^2 \\ x_1^2 + x_2^2 \ldots\ldots x_n^2 \end{cases} \quad \text{and} \sqrt{w} = u$$

u is obtained from the iteration equation

$$u_{i+1} = u_i + \frac{1}{2}\left(\frac{w}{u_i} - u_i\right)$$

$u_o = 1 \times 10^{\frac{p+1}{2}}$ where p is the exponent of 10 in the normalized w.

REMARKS:

a. The average number of iterations performed is five. No iterations are required if the argument w = 0.

b. The result is accurate to within one unit in the last digit position.

c. A program stop occurs on the 35^{th} line of the routine if $w < 0$.

AUTHOR: Sally A. Duval

CUBE ROOT

Routine No. 10.0025

TRANSFER TO ROUTINE:

Location M:	0060	M+1	0000	routine
Location M+1:	B	C	0000	A

DESCRIPTION:

The argument x is in location A. The cube root of x is computed and stored in location B. Control is returned to location C.

MEMORY LOCATIONS: 33

AVERAGE TIME: 5 milliseconds

ACCURACY:

The result is accurate to within one unit in the last digit position.

METHOD: Newton-Raphson

let $\sqrt[3]{x} = u$

then u is obtained from the iteration equation

$$u_{i+1} = u_i + \frac{1}{3}\left(\frac{x}{u_i^2} - u_i\right)$$

$$u_o = \left(\frac{x}{|x|}\right) 8.4 \cdot 10^{\lceil p/3 \rceil}$$ where p is the exponent of 10 in the normalized x.

this iteration continues until

$$\left|\frac{1}{3}\left(\frac{x}{u_i^2} - u_i\right)\right| \leq 5 \cdot 10^{\lceil p/3 \rceil} - 12$$

AUTHORS: Patricia M. Smith and John H. Walker

SINH x , COSH x

Routine No. 10.0014

TRANSFER TO ROUTINE:

Location M:	0060	M	0000	routine
Location M+1:	B_s	B_c	0000	A

DESCRIPTION:

The argument x is in location A. Sinh x and Cosh x are computed and stored in B_s and B_c respectively. Control is returned to location M+2.

x is restricted to the range $-69 < x < 69$.

MEMORY LOCATIONS: 58

AVERAGE TIME: 17 milliseconds

ACCURACY:

If $|x| \leq \frac{1}{2} \ln 10$, cosh x is accurate to 13 digits and sinh x has an error of ± .5 in the 13th digit. For $|x| > \frac{1}{2} \ln 10$, sinh x and cosh x are slightly small because of the truncation of ln 10.

METHOD:

$q = |x| - n \ln 10$, where n is an integer such that $-\frac{1}{2} \ln 10 \leq q \leq \frac{1}{2} \ln 10$.

e^q and e^{-q} are calculated from

$$e^{\pm q} = 1 \pm q + \frac{q^2}{2!} \pm \frac{q^3}{3!} + \cdots\cdots$$

Then:

$$\sinh x = \frac{e^q \, 10^n - e^{-q} \, 10^{-n}}{2}$$

and

$$\cosh x = \frac{e^q \, 10^n + e^{-q} \, 10^{-n}}{2}$$

AUTHOR: A. V. Hershey

RUNGE-KUTTA I

Routine No. 10.00061

There is no write-up of this routine available. Routine No. 10.00071 is an improved version of this routine.

RUNGE-KUTTA METHOD

Routine No. 10.00071

TRANSFER TO ROUTINE:

	C	PQ	R	S	T
Start R.K.:					
Location M	01	1060	M	0000	10.00071
Continue R.K.:					
Location M	01	1660	M	0000	10.00071
Preset Parameters:	02	N	D	E	H
	02	J	A	B	C

The preset parameters must follow the call line M which has the lowest symbolic location.

EXPLANATION OF CALL LINES:

A - First of N consecutive locations of input to differential equations.

B - First of N consecutive locations of output from differential equations.

C - Location to which control is returned after the integration over the full increment has been completed.

D - Location of entrance to computation of differential equations (referred to as D.E. entrance).

E - Location of a blank word which must be put in by the coder following the last instruction in the computation of the differential equations. It is the exit line from the computation of the differential equations and is set up and completely controlled by the R.K. routine. This line is referred to as D.E. exit.

H - Location of interval of integration h.

J - These four digits are not used and need not be zero.

N - Number of differential equations.

MEMORY LOCATIONS REQUIRED: 41+2N

METHOD:

The method used is a fourth order Runge-Kutta process for first order differential equations. If the routine is to be used for higher order systems, the differential equations must be expressed as sets of first order differential equations.

The routine integrates the differential equations between two successive values of the independent variable. The interval of integration (h) remains fixed within an integration cycle, but may be changed if desired before integrating to the next value of the independent variable.

In order to start the process, it is necessary to evaluate the differential equations (DE) at the initial arguments. This evaluation may be done by the "start" portion of the routine. This procedure may also be used whenever it is necessary to recompute the functional values because of a discontinuity in the arguments. If this evaluation has already been done without the aid of the routine, the "start" portion of the routine need not be called.

The integration cycle itself is referred to as the "continue" portion of the routine. In integrating from one value of the independent variable to the next (t_i to t_{i+1}) it is assumed that both the argument values (DE input, u_i) and the functional values (DE output, $\dot{u}_i$) have been computed at t_i. The cycle consists of:

Step (1) - compute DE inputs (u_a) at $t_i + \frac{h}{2}$:

$$u_a = u_i + \frac{h}{2} \dot{u}_i$$

compute DE outputs ($\dot{u}_a$) by evaluating the differential equations

Step (2) - compute DE inputs (u_b) at $t_i + \frac{h}{2}$:

$$u_b = u_i + \frac{h}{2} \dot{u}_a$$

compute DE outputs ($\dot{u}_b$) by evaluating the differential equations

Step (3) - compute DE inputs (u_c) at t_i+h :

$$u_c = u_i + h\dot{u}_b$$

compute DE outputs ($\dot{u}_c$) by evaluating the differential equations

Step (4) - compute the argument values at t_{i+1} :

$$u_{i+1} = u_i + \frac{h}{6}\left(\dot{u}_i + 2\dot{u}_a + 2\dot{u}_b + \dot{u}_c\right)$$

compute the functional values ($\dot{u}_{i+1}$) by evaluating the differential equations.

EXAMPLE:

To integrate the set of second order differential equations

$$\ddot{x} = f\ (t, x, y, \dot{x}, \dot{y})$$

$$\ddot{y} = g\ (t, x, y, \dot{x}, \dot{y})$$

with respect to time express them as a set of first order differential equations

(1) $\frac{d}{dt}\ (x) = \dot{x}$ (2) $\frac{d}{dt}\ (y) = \dot{y}$

(3) $\frac{d}{dt}\ (\dot{x}) = \ddot{x}$ (4) $\frac{d}{dt}\ (\dot{y}) = \ddot{y}$

together with a fifth differential equation

(5) $\frac{d}{dt}\ (t) = 1$

so that time dependent variables in the differential equations may be properly evaluated. Equations (1) and (2) would be written simply as transfers of $\dot{x}$, $\dot{y}$ (DE input) to $\dot{x}$, $\dot{y}$ (DE output).

To start the integration, the initial values t_o, x_o, y_o, $\dot{x}_o$, $\dot{y}_o$ must be transferred to the DE input locations A, A+1, ..., A+4. By initiating the RK Start line, the differential equations are evaluated and the DE output ($\dot{t}$, $\dot{x}$, $\dot{y}$, $\ddot{x}$, $\ddot{y}$) stored in locations B, B+1, ..., B+4. The program continues at location C.

To continue the integration, the RK continue line is initiated. Following the evaluation of the differential equations in step (4), control is returned to location C.

REMARKS:

a. One must have all necessary information available not only at the points in which he is interested, but also at the points halfway between them. This information may be given in the form of interpolation tables, polynomial fits, etc.

b. The routine integrates from t to $t+\Delta t$ exactly.

c. The original contents of the modifiers are restored on exit from the routine to either the main program or the D.E. entrance.

AUTHOR: Paul M. Botting

To start the integration, the initial values t_o, x_o, y_o, $\dot{x}_o$, $\dot{y}_o$ must be transferred to the DE input locations A, A+1, ..., A+4. By initiating the RK Start line, the differential equations are evaluated and the DE output ($\dot{t}$, $\dot{x}$, $\dot{y}$, $\ddot{x}$, $\ddot{y}$) stored in locations B, B+1, ..., B+4. The program continues at location C.

To continue the integration, the RK continue line is initiated. Following the evaluation of the differential equations in step (4), control is returned to location C.

REMARKS:

a. One must have all necessary information available not only at the points in which he is interested, but also at the points halfway between them. This information may be given in the form of interpolation tables, polynomial fits, etc.

b. The routine integrates from t to $t+\Delta t$ exactly.

c. The original contents of the modifiers are restored on exit from the routine to either the main program or the D.E. entrance.

AUTHOR: Paul M. Botting

FOUR POINT INVERSE INTERPOLATION

Routine No. 10.0011

TRANSFER TO ROUTINE:

Location M:	0060	M	0000	routine
Location M+1:	No. of direct interpolations (exp. style)			
Location M+2:	Tolerance on interpolation (exp. style)			
Location M+3:	0000	0000	0000	α
Location M+4:	0000	0000	0000	C
Location M+5:	0000	0000	K_1	0000
Location M+6:	0000	0000	Loc. $\overline{X}$	0000
Location M+7:	0000	0000	Loc. X_{n-2}	0000
Location M+8:	0000	0000	Loc. A_{n-2}	0000
Location M+9:	0000	0000	Loc. B_{n-2}	0000
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

DESCRIPTION:

X, A, B, C, ..., N, are functions of a common independent variable t which is equally spaced. $A(t_n) = A_n$, $X(t_n) = X_n$, $A(\overline{t}) = \overline{A}$, $X(\overline{t}) = \overline{X}$, etc. $\overline{X}$ is a given value of X which generally lies between X_{n-1} and X_n. This routine will compute $\overline{A}$, $\overline{B}$, $\overline{C}$, etc. If $\overline{t}$ is not to be computed one need never know any t values. The functional values are required at the four points t_{n-2}, t_{n-1}, t_n and t_{n+1}.

The accuracy of $\overline{A}$, $\overline{B}$, $\overline{C}$, etc. is determined by the tolerance on the phase iterations (usually 10^{-8}). $\overline{A}$, $\overline{B}$, $\overline{C}$, etc. are stored in locations α, $\alpha + 1$, $\alpha + 2$, etc. respectively. C is the return address and K_1 = Loc. A_{n-1} - Loc. A_{n-2} .

It is necessary that:

K_1 = Loc A_{n+1} - Loc A_n = Loc A_n - Loc A_{n-1} = Loc A_{n-1} - Loc A_{n-2}
K_1 = Loc X_{n+1} - Loc X_n = Loc X_n - Loc X_{n-1} = Loc X_{n-1} - Loc X_{n-2}
K_1 = etc.

MEMORY LOCATIONS: 90

AVERAGE TIME:

$\frac{11n + 25}{4}$ milliseconds where n is the number of interpolations and the tolerance is 10^{-8} .

METHOD: (Using Horizontal Difference Notation)

$$\bar{N} = N_{n-1} + u \Delta N_n + \frac{1}{2} u (u-1) \Delta_2 N_n + \frac{1}{6} u(u-1)(u+1) \Delta_3 N_{n+1}$$

where the phase, u, is obtained by the iteration

$$u_o = \frac{\bar{X} - X_{n-1}}{\Delta X_n}$$

$$u_{i+1} = u_o - \frac{u_i (u_i-1)}{2} \frac{\Delta_2 X_n}{\Delta X_n} + \frac{(u_i+1)(u_i)(u_i-1)}{6} \frac{\Delta_3 X_{n+1}}{\Delta X_n}$$

AUTHOR: David F. Eliezer

2 POINT - 2 SLOPE INVERSE INTERPOLATION

Routine No. 10.0010

TRANSFER TO ROUTINE:

Location M:	0060	M	0000	routine
Location M+1:	0000	Loc. Y_2	Loc. $\overline{Y}$	Loc. Y_1
Location M+2:	Loc. t_1	0000	0000	Loc. t_2
Location M+3:	Loc. $\dot{Y}_1$	0000	0000	Loc. $\dot{Y}_2$
Location M+4:	0060	0000	B	C

DESCRIPTION:

This routine fits a parabola through the points $(t_1, y_1, \dot{y}_1)$ and $(t_2, y_2, \dot{y}_2)$ and determines t, the independent variable, for a given value $(\overline{y})$ of y, the dependent variable. The larger value of t is stored in B and the smaller in B+1. Control is returned to C if the values of t are real and to C-1 if they are complex.

MEMORY LOCATIONS: 106

ACCURACY:

The results are accurate to 12 significant digits.

METHOD:

$$\Delta y = y_2 - y_1 \quad , \quad \Delta\overline{y}_1 = y_1 - \overline{y}$$

$$\Delta t = t_2 - t_1 \quad , \quad \Delta\overline{y}_2 = y_2 - \overline{y}$$

$$k' = \frac{(\dot{y}_1 - \dot{y}_2)^2}{4(\dot{y}_1 \Delta t - \Delta y)(\dot{y}_2 \Delta t - \Delta y)}$$

$$a = \dot{y}_1 \dot{y}_2 + k' \Delta y^2$$

$$b = \dot{y}_1 (\Delta\overline{y}_2 - \dot{y}_2 t_2) + \dot{y}_2 (\Delta\overline{y}_1 - \dot{y}_1 t_1) - 2k' \Delta y (t_1 \Delta\overline{y}_2 - t_2 \Delta\overline{y}_1)$$

$$c = (\Delta\bar{y}_2 - \dot{y}_2\, t_2)\ (\Delta\bar{y}_1 - \dot{y}_1\, t_1) + k'\ (t_2\, \Delta\bar{y}_1 - t_1\, \Delta\bar{y}_2)^2$$

$$t = \frac{-\,b \pm \sqrt{b^2 - 4ac}}{2a}$$

REMARKS:

Do not use this routine if no parabola exists that will fit the data points and their slopes.

If the results are complex, meaningless material will be stored in B and B+1.

One must sense on the 2 t's to see which one lies in the range of the desired result. If both lie in the range and only one is wanted, some other means of deciding which is wanted must be used.

AUTHOR: David F. Eliezer

2 POINT, 2 SLOPE DIRECT INTERPOLATION

Routine No. 10.0018

TRANSFER TO ROUTINE:

Location M:	0060	M	0000	routine
Location M+1:	0060	0000	B	C
Location M+2:	0000	Loc. $\dot{S}_2$	Loc. h	Loc. $\dot{S}_1$
Location M+3:	Loc. S_2	Loc. S_1	0000	Loc. u

DESCRIPTION:

This routine fits a third degree polynomial in t through two points $(t_1, S_1, \dot{S}_1)$ and $(t_2, S_2, \dot{S}_2)$ and determines $\overline{S}$, the value of the dependent variable for a given $\overline{t}$. $\overline{S}$ is stored in B.

h and u should be computed according to the following formulas and stored in the specified input locations.

$$h = t_1 - t_o$$

$$u = \frac{\overline{t} - t_o}{t_1 - t_o} = \frac{\overline{t} - t_o}{h}$$

Control is returned to location C.

MEMORY LOCATIONS: 43

METHOD:

$$\Delta S_1 = S_2 - S_1$$

$$\overline{S} = S_1 + h\,\dot{S}_1\,(u^3 - 2u^2 + u) + h\,\dot{S}_2\,(u^3 - u^2) - \Delta S_1\,(2u^3 - 3u^2)$$

AUTHOR: Paul M. Botting

C - 3.0 13 March 1958

N-POINT LAGRANGIAN INVERSE INTERPOLATION

Routine No. 10.0017

TRANSFER TO ROUTINE:

Location M:	0060	M	0000	routine
Location M+1:	No. of points in exp. style			
Location M+2:	0000	0000	Loc x_{i+1} - Loc x_i	0000
Location M+3:	0000	0000	Loc y_1	0000
Location M+4:	0000	0000	Loc x_1	0000
Location M+5:	0000	0000	Loc $\overline{y}$	0000
Location M+6:	0000	0000	B	C

DESCRIPTION:

Given n points (x_i, y_i for i = 1,2,3, ... n) and argument value ($\overline{y}$), the routine will find an $\overline{x}$ corresponding to $\overline{y}$ such that the point ($\overline{x}$, $\overline{y}$) lies on the $(n - 1)^{th}$ degree polynomial in y determined by the points (x_i, y_i). $\overline{x}$ is stored in location B. For the routine to work y_i must not equal y_k for $i \neq k$ and loc x_i - loc x_{i-1} must equal loc y_i - loc y_{i-1}. Control is returned to C.

MEMORY LOCATIONS: 53

METHOD:

This routine uses Lagrange's interpolation formula:

$$\overline{x} = \sum_{j=1}^{n} \left[\prod_{\substack{i=1 \\ i \neq j}}^{n} \frac{\overline{y} - y_i}{y_j - y_i} \; x_j \right]$$

AUTHOR: David F. Eliezer

RANDOM NUMBER GENERATOR

Routine No. 10.00221

TRANSFER TO ROUTINE:

	C	Location	PQ	R	S	T
Location M:	01	M	0860	M	----	10.00221
	02	----	B	L	Loc (max)	Loc (min)

DESCRIPTION:

A random number, scaled to a uniform distribution between, but not including, the numbers in loc (min) and loc (max), is placed in B. L is the location in which the routine stores each new unscaled random number; the coder must set the initial value in L and it should be either 1 (in floated form) or the contents of L resulting from some previous run. Control is returned to M+1.

MEMORY LOCATIONS REQUIRED: 11

TIME: 448 microseconds

METHOD:

13 low-order digits of the product $L' \times 3^{27}$ are stored in L with a 12 exponent.

then $L' \times 10^{-13} \times (max'-min') + min'$ is stored in B

AUTHORS: Lottie J. Griffin and Mary L. Hagemeyer

$$\int_0^t e^{-\frac{x^2}{2}}\,dx$$

Routine No. 10.0023

TRANSFER TO ROUTINE:

Location M:	0060	M+1	0000	routine
Location M+1:	B	C	0000	A

DESCRIPTION:

The argument t is in location A. The integral is computed and stored in location B. Control is returned to location C.

MEMORY LOCATIONS OCCUPIES: 35

AVERAGE TIME: 10 milliseconds

METHOD:

The series for $e^{-\frac{x^2}{2}}$ is integrated term by term.

The integral is obtained by summing the series

$$S = t - \frac{t^3}{3} + \frac{t^5}{5\cdot 2^2\cdot 2!} - \frac{t^7}{7\cdot 2^3\cdot 3!} + \cdots$$

until the addition of the n^{th} term contributes nothing to the sum.

REMARKS:

a. The routine accepts ± values of t (for -t, the integral is negative).

b. The results are accurate to ± 5 units in the seventh significant digit. The accuracy according to the absolute value of the argument is

$t \leq 2$	5 in 13^{th} significant digit
$2 < t \leq 3$	5 in 12^{th} " "
$3 < t \leq 4$	5 in 10^{th} " "
$4 < t \leq 6$	5 in 7^{th} " "
$6 < t \leq 7$	5 in 9^{th} " "
$7 < t \leq 8$	5 in 12^{th} " "
$8 < t$	5 in 13^{th} " "

c. If $|t| \geq 5$, the limit of the integral, which is $\pm\sqrt{\frac{\pi}{2}}$, is taken for the value of the series. This has caused some difficulty in evaluating the integral

$$p = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{x^2}{2}} \, dx$$

(when $t \leq -5$) in the expression

$$p = \tfrac{1}{2} + \frac{1}{\sqrt{2\pi}} \left(\sqrt{\frac{\pi}{2}}\right)$$

$$= \tfrac{1}{2} + (.3989422804014)\ (-1.253314137315)$$

$$= \tfrac{1}{2} - .4999999999997 = 3\text{x}10^{-13} \neq 0.$$

This can be avoided by taking $\frac{1}{\sqrt{2\pi}} = .3989422804016$ and using a 34 code for the multiplication of $\left(\frac{1}{\sqrt{2\pi}}\right)\left(\sqrt{\frac{\pi}{2}}\right)$.

AUTHOR: John H. Walker

MATRIX ARITHMETIC

Routine No. 10.00241

TRANSFER TO ROUTINE:

	C	PQ	R	S	T
Location M:	01	XX60	M	0000	10.00241
	02	0000	0000	0000	K

Location M+1: (see below)

Location M+2: (used for inversion only - see below)

DESCRIPTION:

Location A, B, C and H contain the matrix specification words of matrices $[A]$, $[B]$, $[C]$, and $[H]$. Location K is the first of N consecutive locations reserved by the programmer for use by the routine as temporary storage. $N \geq$ the largest row length of any matrix resulting from the operations described below in a).

Matrix specification words are words of the following form:

a # rows # cols. Δ row

where

a = the location of element a_{11} of matrix $[A]$

rows = the number of rows in matrix $[A]$

cols. = the number of columns in matrix $[A]$

Δ row = location of $a_{i+1,\,j}$ - location of $a_{i,j}$

(location of $a_{i,j+1}$ - location of $a_{i,j}$ = 1)

The routine performs the following operations:

a.) Addition, subtraction, and multiplication as follows:

Loc. M+1 = 5020 A B C $[A] + [B] + \sigma [C] \rightarrow [C]$

Loc. M+1 = 5021 A B C $-([A] + [B]) + \sigma [C] \rightarrow [C]$

Loc. M+1 = 5022 A B C $[A] - [B] + \sigma [C] \rightarrow [C]$

Loc. M+1 = 5023 A B C $-([A] - [B]) + \sigma [C] \rightarrow [C]$

Loc. M+1 = 5024 A B C $[A] \times [B] + \sigma [C] \rightarrow [C]$

Loc. M+1 = 5025 A B C $-([A] \times [B]) + \sigma [C] \rightarrow [C]$

σ = 0 if xx = 00, σ = 1 if xx = 98

b.) Transference and transposition as follows:

Loc. M+1 = 0060 0001 A B $[A] \rightarrow [B]$

Loc. M+1 = 0060 0011 A B $[A]' \rightarrow [B]$

xx must = 00

In the case of operations a.) and b.) control is returned to location M+2.

c.) Inversion as follows:

Loc. M+1 = 5026 H A B $[A]^{-1} \rightarrow [B]$

Loc. M+2 = Δ E 0000 F $[A][B] - [I] \rightarrow [H]$

Loc. M+1 5027 H A B $-[A]^{-1} \rightarrow [B]$

Loc. M+2 Δ E 0000 F $-[A][B] - [I] \rightarrow [H]$

Δ = location of tolerance δ

E = location of tolerance ϵ

F = print control = 007X

x = 0 print unconditionally on printer 1.

x = 2 do not print

x = 4, 5, 6, 7, 8, 9 print on printer 1 if appropriate option switch is on "transfer".

Printer 1 prints the sentinel line:

5027
5026 - - h , - - - - , c k k l , a k k m , b k k n, - - - - , q in which h is the order of the matrix being inverted; k is the order of the current iteration; c, a, and b, are the locations of the first elements of matrices [H], [A], and [B] ; l, m, and n are the row increments of [H], [A], and [B] ; and $1/q$ is the pivot number. The printer then prints the elements of matrix [H] , row by row, with spacing between rows.

Control is returned to location M+3.

MEMORY LOCATIONS REQUIRED: 344

METHOD OF INVERSION:

The inversion is done by the partitioning method with Hotelling's refinement. The routine proceeds to Hotelling's refinement after every iteration in which the largest element of the error matrix ($h_{max.}$) exceeds the tolerance δ , or in any event after the last iteration. The refinement continues until $h_{max.}$ increases. At this point the routine stops if $h_{max.} > \epsilon$. If $h_{max.} \leq \epsilon$, the routine continues with the next step of the partitioning method. Thus, δ is a loose tolerance used to determine when to refine and ϵ is the tolerance which must be met by the final inverse.

REMARKS:

a. If inversion is being performed:

   (1) the routine stops if the pivot number is less than 10^{-31} (0061 0000 0000 0000)

   (2) the routine stops if $h_{max.}$ has increased as a result of an iteration of the Hotelling refinement but does not meet the tolerance ϵ (0061 Y 0000 Z). $h_{max.}$ is in register storage at this time. One may push through this stop if one is willing to accept an inverse which doesn't meet the tolerance ϵ .

b. The operation $\pm$ ([A] [B]) + σ [A] $\rightarrow$ [A] is permissible but $\pm$ ([A] [B]) + σ [B] $\rightarrow$ [B] is not.

c. The operations [A] $'\rightarrow$ [A] and $\pm$[A] $^{-1}\rightarrow$[A] are not permissible.

d. The routine uses all the modifiers and does not reset them.

AUTHOR: Dr. A. V. Hershey

MATRIX INVERSION

Routine No. 10.0030

TRANSFER TO ROUTINE:

Location M:	0060	M	0000	routine
Location M+1:	σ	-	B	A

DESCRIPTION:

Locations A and B contain the matrix specification words of matrices [A] and [B] . The routine inverts matrix [A] and stores the result as matrix [B] . If σ = 0000 no refinement takes place. If σ = 0001 the inverse will be refined until the error measure ceases to decrease. The routine returns control to location M+2. The error measure is in register storage at this time.

A matrix specification word is a word of this form:

a # rows # cols Δ row

a = location of element a_{11} of matrix [A]

rows = # rows of matrix [A]

cols = # columns of matrix [A]

Δ row = location of $a_{i+1,j}$ - location of $a_{i,j}$

(location of $a_{i,j+1}$ - location of $a_{i,j}$ = 1)

MEMORY LOCATIONS OCCUPIED: 152

METHOD:

Bi-orthoganalization with automatic pivot selection using the identity matrix as the initial approximation to the inverse.

REMARKS:

The accuracy of the inverse depends on the condition number of the matrix to be inverted. The error measure is the maximum $\sum_{k=0}^{n} e_{jk}$ where (e_{jk}) is the error matrix. It is a rough measure of the accuracy of the inverse: i.e., an error measure of 7×10^{-6} indicates approximately 6 digit accuracy.

AUTHOR: Dr. R. J. Arms and Paul M. Botting

MATRIX MULTIPLY

Routine No. 10.00311

TRANSFER TO ROUTINE:

	C field	P Q	R	S	T
Location M:	01	0060	M	0000	10.00311
	02	0000	0000	0000	R
Location M+1:	00	0X0Y	A	B	C

DESCRIPTION:

Locations A, B, and C contain the matrix specification words* of matrices [A], [B], and [C]. The routine performs the operation $[A][B] + y[C] \rightarrow [C]$ $(y = 0,1)$ in the following manner. If $x = 0$, the routine accumulates a single row at a time of the product $[A][B]$ in consecutive locations starting with location R. This row is then either added to or transferred to the corresponding row of [C] before another row is calculated. Similarly, if $x = 1$, the routine accumulates the product a column at a time in consecutive locations starting with location R. This column is then either added to or transferred to the corresponding column of [C].

MEMORY LOCATIONS REQUIRED: 90

REMARKS:

To perform $[A][B] + y[B] \rightarrow [B]$ $\quad y = 0,1$
x must be 1.

To perform $[A][B] + y[A] \rightarrow [A]$ $\quad y = 0,1$
x must be 0.

The programmer must reserve n locations

(n = row length of $[A][B]$ if $x = 0$
n = column length of $[A][B]$ if $x = 1$)

in his program starting with location R.

*For a description of matrix specification words, see the write up of routine 10.0030.

AUTHOR: Paul M. Botting

MATRIX ADD AND SUBTRACT

Routine No. 10.0032

TRANSFER TO ROUTINE:

Location M:	0060	M	0000	routine
Location M+1:	5Y2X	A	B	C

DESCRIPTION:

Locations A, B, and C contain the matrix specification words* of matrices [A], [B], and [C]. This routine performs the following operations:

P Q	R	S	T		
5Y20	A	B	C	$[A] + [B] + y\,[C] \rightarrow [C]$	$(y = 0,1)$
5Y21	A	B	C	$-([A] + [B]) + y\,[C] \rightarrow [C]$	$(y = 0,1)$
5Y22	A	B	C	$[A] - [B] + y\,[C] \rightarrow [C]$	$(y = 0,1)$
5Y23	A	B	C	$-([A] - [B]) + y\,[C] \rightarrow [C]$	$(y = 0,1)$
5Y28	A	B	C	$\lvert[A]\rvert - \lvert[B]\rvert + y[C] \rightarrow [C]$	$(y = 0,1)$

$$\lvert[A]\rvert \equiv (\lvert a_{ij}\rvert)$$

Corresponding Q codes of the thirty series may also be used. Control is returned to location M+2.

MEMORY LOCATIONS REQUIRED: 59

*For description of matrix specification words, see write up of routine 10.0030.

AUTHOR: Paul M. Botting

MATRIX TRANSFER-TRANSPOSE

Routine No. 10.0033

TRANSFER TO ROUTINE:

Location M:	0060	M	0000	routine
Location M+1:	σ	0000	A	B

DESCRIPTION:

Locations A and B contain the matrix specification words* of matrices [A] and [B] respectively. The following operations may be performed:

σ = 0000 $\quad [A] \rightarrow [B]$

σ = 0001 $\quad [A]' \rightarrow [B]$

Control is returned to location M+2.

MEMORY LOCATIONS REQUIRED: 47

*For description of matrix specification words, see write up of routine 10.0030.

AUTHOR: Paul M. Botting

LOAD IDENTITY MATRIX

Routine No. 10.0034

TRANSFER TO ROUTINE:

Location M:	0060	M	0000	routine
Location M+1:	0000	A	0000	0000

DESCRIPTION:

Location A contains the matrix specification word of matrix [A]. The routine loads the identity matrix into matrix [A] and returns control to location M+2.

MEMORY LOCATIONS REQUIRED: 36

*For description of matrix specification words, see write up of routine 10.0030.

AUTHOR: Paul M. Botting

INPUT GENERATOR

Routine No. 10.00151

DESCRIPTION:

This routine contains provisions for manufacturing various types of parameter combinations. New sets of parameters may also be produced by making changes in previously formed sets. Provision for "coupling" allows use of related parameters.

For complete information on this routine see NPG REPORT No. 1487.

OUTPUT EDITOR

Routine No. 10.00161

This routine is not used in new codings. It is left in the Library for use in codings which used it in the past and which may have to be recompiled in the future.

PACKING ROUTINE FOR THE CCP

Routine No. 10.0019

TRANSFER TO ROUTINE:

Location M:	0060	M	0000	routine
Location M+1:	Z	0000	0000	0000
Location M+2:	0000	0000	Loc. X_1	0000
Location M+3:	0000	0000	Loc. X_n	0000
Location M+4:	0000	0000	Y	0000
Location M+5:	Number of words per line in exponential style			
Location M+6:	0000	0000	W	0000
Location M+7:	0000	0000	0000	C

DESCRIPTION:

This routine takes the numbers X_1, X_2, --- X_n from consecutive memory locations, rounds them, packs them for printing on one line by the Card Controlled Printer and then stores them with the appropriate control words in consecutive memory locations starting at location Y.

Z = 0000 if the printing is to be done on the whole page.

Z = 0001 if the printing is to be done on the left side of the page.

Each column requires 3 control words to define its style, number of decimal places and decimal point position. These words are stored in consecutive memory locations starting at W.

Location W:	J	0000	0000	0000
Location W+1:	0000	0000	0000	K
Location W+2:	0000	0000	0000	L

J = 0000 if fixed style is to be used.

J = 0001 if exponential style is desired.

K is the number of digits to be retained to the right of the decimal point.

L gives the print position of the decimal point. The range of L is from 6 to 125. Print position 1 corresponds to 6.

Control is returned to line C. When this routine transfers back to the main program the last output register plus one is in register storage.

MEMORY LOCATIONS: 390

REMARKS:

To delete a word, put a "2" in the sign column and zero elsewhere.

Words must be in the order in which they are to be printed (row-wise).

Printed words must not overlap.

The decimal point position, for a number which will have no decimal point printed, must be all zeros.

The number of digits to the right of the decimal point must not exceed 12.

For fixed style it must be true that, $|\text{exponent}| < 12$.

When exponential style is used six digit positions must be left to the right of the decimal point to put in the exponent. These are in addition to K which is the number of significant digits to the right of the decimal point.

The total number of digits specified for each word cannot exceed 13.

This routine requires a maximum of 16 output storage locations per line.

AUTHOR: David F. Eliezer

MATRIX MONITOR PRINTING ROUTINE

Routine No. 10.0035

TRANSFER TO ROUTINE:

Location L:	00XX	L	0000	routine
Location L+1:	8X60	C	____	D
Location L+2:	E	F	G	H

DESCRIPTION:

This routine prints an m by n matrix A on the standard board.

00XX of location L may be 0060 or 007X, where x = 4 through 9. 0060 indicates that the routine is always to be used, whereas 007X indicates that the routine may be used by putting condition switch 7X on transfer, or that the routine may be by-passed by putting condition switch 7X on off, or that the machine may be stopped by putting condition switch 7X on stop.

The P field of line L+1 contains the printing style as 8X where x = 1 through 4. If x = 1 or 2, the entire printed output will be single spaced on printer one or two respectively; if x = 3 or 4, the matrix will be single spaced, with double spacing after the last line of the matrix, on printer one or two respectively.

C = loc. $a_{i,\ j+1}$ - loc. $a_{i,j}$ (adjacent columns of the matrix)

D is the return line.

E is the location of a_{11} (first element of matrix A).

F is the number of rows of the matrix.

G is the number of columns of the matrix.

H = loc. $a_{i+1,\ j}$ - loc. $a_{i,j}$ (consecutive rows of the matrix).

PRINTING:

A row of the matrix is printed across the page and if necessary is continued on successive lines of the printer sheet. Each row is started on a new line. If any one row of the matrix does not fill out a line, words of 16 sevens are used to complete it.

MEMORY LOCATIONS REQUIRED: 46

TIME REQUIRED IN MINUTES:

$$\left[\text{Integral part of } \left(\frac{\#\text{ of columns } +6}{7}\right)\right] \ (\#\text{ of rows}) \ \frac{1}{150}$$

REMARKS: Modifiers may be used in the call lines.

AUTHOR: Charles H. Juergensmeyer

CRT PRINTING ROUTINE

Routine No. 10.00431

TRANSFER TO ROUTINE:

C	Location	P	Q	R	S	T
01	M	0Z	60	M	0000	10.00431
02		0X	K	P	0X K	P
.		.				
.		.				
.		.				
.		.				
02		0X	K	P	0X K	P
	M+1		A_1	A_n	0000	Y
	M+2	J0	00	BE	D	C

The preset parameter lines must follow the call line M which has the lowest symbolic location.

DESCRIPTION OF ROUTINE:

The routine prints the words in locations A_1 through A_n on the CRT printer and returns control to C, unless the "end of page" condition causes control to be returned to D. This printing may be performed on either or both cameras and may be done in groups of lines with specified intergroup spacing or without grouping.

EXPLANATION OF CALL LINES:

Z - A "start print" call is indicated by Z = 0. The routine clears the group control counter, sets the digit counter equal to zero, and sets the line counter equal to Y before starting to print. A "continue print" call is indicated by Z = 1. In this case the routine starts printing without first altering the digit, line, and group control counters. The group counter is a counter, internal to the routine, which indicates which line of a group is currently being printed.

A_1 - The first of n consecutive locations of data words to be printed.

A_n - The last of n consecutive locations of data words to be printed.

Y - The coordinate needed only with the "start print" call. It sets the line counter initially. ($0000 \leq Y \leq 1199$; Y = 0042 indicates line 2.1)

J - A single digit for camera selection.

J = 1 use camera A
J = 2 use camera B
J = 3 use both camera A and camera B

B - Two digits indicating the number of lines to be printed as a single group. If B = 00 or 01 a single line will be printed per group.

E - Two digits indicating the line spacing between groups. E = 00 or 01 indicates single spacing. For example, E = 04 indicates quadruple spacing (i.e., 3 lines between groups).

D - Location to which control is returned when the end of page condition exists. The film is not advanced. The line and group control counters are advanced and the digit counter is set to zero. However, if D = 0000 and the end of page condition exists, then the line and digit counters are reset to zero, the film is advanced once, the group control counter is advanced, and printing will continue on the first line of the next page.

C - Location to which control is returned after the word in A_n has been printed, provided control has not been returned to D. The line and group control counters are advanced and the digit counter is set to zero.

X - A single digit which controls the print style for a column.

X = 0 indicates fixed style
X = 1 indicates exponential style

The form of printing with the fixed and exponential styles is illustrated below.

fixed style:	-0.01234	56.78
exponential style:	-1.234 -02	5.678 01

K - Two digits indicating the number of digits to the right of the decimal point for a column.

P - The digit position of the decimal point as in Form NPG-564. ($3 \leq P \leq 120$) P is greater than 2 because allowance must be made for the sign and a single digit preceding the decimal point.

MEMORY LOCATIONS REQUIRED:

$(125 + \frac{5}{4} c) \leq N < 182$ if no columns are printed in exponential style.

$(151 + \frac{5}{4} c) \leq N < 209$ if any column is printed in exponential style.

c = number of columns

N = number of memory locations required

N is generally nearer the lower bound than the upper bound.

REMARKS:

a. To delete a word, replace it with 0020 0000 0000 0000.

b. Words must be in the order in which they are to be printed.

c. Printed words must not overlap. If overlap occurs a program stop occurs in the routine.

d. The total number of digits to be printed as one number must not exceed 13 (not counting the exponent in exponential style).

e. When exponential style is used, four digit positions must be left to the right of the decimal point for the exponent. These are in addition to K, which is the number of digits to the right of the decimal point.

f. When control is returned to either C or D the routine transfers into the PQ field of register storage the location of the next word that would have been used as a data word.

g. The maximum number of 02 codes which may be used in this routine is 16.

AUTHOR: Raymond V. Borchers

IDENTIFY CRT OUTPUT
Routine No. 10.00441

TRANSFER TO ROUTINE:

	C field	PQ	R	S	T
Location M:	01	0060	M+1	0000	10.00441
	02	X000	00$\alpha\beta$	$\alpha\beta\alpha\beta$	$\alpha\beta\alpha\beta$
Location M+1:	00	0060	0000	0000	R

DESCRIPTION:

This routine prints five large block characters specified by the $\alpha\beta$ digits in the 02 code. Camera selection is indicated by the digit X. Control is returned to location R.

X = 1 print on camera A

X = 2 print on camera B

X = 3 print on both camera A and B

The $\alpha\beta$ digits of this routine specify the same characters that would be selected by these $\alpha\beta$ digits in the alpha-numeric mode of CRT printing.

The block characters are 19 lines high and 20 digits wide. They extend vertically from line 5 through line 23 and cover the frame horizontally.

There is no film advance in this routine.

MEMORY LOCATIONS: 31 + N

N = the total number of line segments making up the five characters to be printed (maximum N = 35).

EXAMPLE:

The following 02 code:
10 00 0041 4260 4746
will preset the routine to print on camera A the block characters 12-76.

AUTHOR: Paul M. Botting and Lawrence C. Brown

TRACER

Routine No. 10.0026

TRANSFER TO ROUTINE:

Location M:	9260	M+1	0000	routine
Location M+1:	0000	0000	C	L

OPTIONAL TRANSFER TO ROUTINE:

Location M:	927X	M+1	0000	routine
Location M+1:	0060	C	C	L

Any of the switches may be used here. Tracing is done only when the switch is on "transfer".

DESCRIPTION:

L is the location of the first instruction to be traced, the routine will immediately operate the instruction in this location. Control is returned to C, which is the location following the last line to be traced.

This routine prints the address, instruction, operands, result and modifiers of each line of coding, whose location number is less than but not equal to C until the return line is operated. Tracing is printed on printer 1, using the standard board.

MEMORY LOCATIONS REQUIRED: 100

PRINTER FORMAT: See next page.

AUTHOR: Ruth W. Montville

PRINTER FORMAT

Loc. #	Instruction	R'	S'	Result	Not Used	Contents of M_4 M_6 M_8
000000000000xxxx	xxxxxxxxxxxxxxx	xxxxxxxxxxxxxxx	xxxxxxxxxxxxxxx	xxxxxxxxxxxxxxx	0000000000000000	0000xxxxxxxxxxx
EXAMPLE:						
0000000000000313	0060123402620000	9902500000000000	9902500000000000	0000000000000000	0000000000000000	0000005500400021
0000000000000314	5026000007570264	9902500000000000	0003000000000000	9808333333333333	0000000000000000	0000005500400021
0000000000000315	0070164407570219	0001000000000000	0003000000000000	0000000000000000	0000000000000000	0000005500400021
0000000000000316	5020587802621238	0001250000000000	9901250000000000	0001750000000000	0000000000000000	0000005500400021
0000000000000317	0054000000010010	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000005500410010
0000000000000318	0063000000000327	0001750000000000	0001750000000000	0000000000000000	0000000000000000	0000005500410010
0000000000000319	0072164407570313	0001000000000000	9998000000000000	0000000000000000	0000000000000000	0000005500410010
0000000000000313	0060123402620000	9902500000000000	8902500000000000	0000000000000000	0000000000000000	0000005500410010

NOTE: If a 70 through 74 code is used and a transfer occurs S' field will contain the difference and not true S'.

If a transfer does not occur the S' field contains true S'.

STANDARD RESTART

Routine No. 10.00291

TRANSFER TO ROUTINE:

Location	C	P	Q	R	S	T
Start Call						
M	01	01	Transfer Code	M	0000	10.00291
Control Call						
M	01	08	60	M	0000	10.00291
Preset Parameters						
	02	XX_0	00 [93] [94]	0000	C	Y_0
	*02	XX_1	[93] [94]	0000	0000	Y_1
	02	XX_2	[93] [94]	0000	0000	Y_2
	.	.	.	.	.	.
	.	.	.	.	.	.
	02	XX_n	[93] [94]	0000	0000	Y_n

The 02 codes must follow the 01 code having the lowest symbolic address.

*The following 02 codes need to be put in only those segments in which the control call is used.

EXPLANATION OF CALL LINES:

The Q field of the Start Call can be any transfer instruction.

C - Return line for the Start Call. The Control Call will return to M+1.

XX_i - Tape code being used in the program.

XX_0 - Tape code of the control tape.

Y_i - Location containing the block number corresponding to tape code XX_i, in the T field. Location Y_0 need be put in only if XX_0 is to be used between calls to the Control Call line.

[93] - The Q field of an 02 code will be a 94 if that tape
[94] is to be an input tape and a 93 if it is to be an output tape. The routine will position after the block number specified for input tapes. It will subtract one from the block number, position after the resulting block number and then delete for output tapes.

DESCRIPTION:

Tape XX_0 may be either an input tape with 2 EOF's after the data or a tape with nothing but 2 EOF's on it. The routine will delete the 2nd EOF and record the contents of memory.

After a standard start is taken the Start Call line should be operated. This will read XX_0 through the 1st EOF and then read the next block. If this is an EOF it will rewind XX_0 and return to C.

If the block read is not an EOF the routine will read the block into memory, position the tapes, delete on the output tapes, set up the contents of the modifiers, reposition XX_0 and return to the same line that control was returned to when the Control Call was last operated.

In the course of the computation a sensing should be done so that the Control Call will not be operated too often (approximately every 15 minutes). This sensing can be on the completion of a specified number of items. When the proper condition exists and the Control Call is operated the routine will store the contents of the modifiers, position past the 1st EOF on XX_0, delete, and record the contents of memory on the control tape XX_0. Tape XX_0 is then repositioned and control is returned to line M+1.

MEMORY LOCATIONS: 75

REMARKS:

a. If the block number specified is 0000 the routine will not move the tape at this time.

b. It is suggested that the control tape, XX_0 be an input tape which changes for each set-up.

c. This routine can be used if more than one EOF is on any tape other than XX_0, by specifying the tape code more than once. In this event the EOF indicator will be on upon returning from the restarting section of the routine to line M+1 of the Control Call.

d. To insure that upon demand the program will start back at the beginning of the input, the following can be done:

Location M-1	007X	0000	0000	C
M	0160	M	0000	10.00291

where C has the same meaning as before.

When switch 7X is on transfer the program will start at the beginning and the restart controls will be set up accordingly.

e. Stops:

34^{th} line of the routine - TCF trying to verify the writing of controls on XX_0. A start at V will try to write again.

75^{th} line of the routine - TCF trying to read in controls or position tapes. The read instruction will be in register storage.

f. There can be no more than 9-02 codes in one segment.

g. The output blocks should be numbered uniquely.

AUTHOR: W. P. Warner

RE-READ BLOCK
(If TCF Occurs)

Routine No. 10.0020

TRANSFER TO ROUTINE:

Location M-1: "Read" or "Verify" instruction

Location M: 0868 M 0000 routine

It is assumed that the TCF indicator is in the "off" position before operating the "read" or "verify" instruction in location M-1. If a tape check failure occurs as a result of this tape operation, control will be transferred to the "Re-read Block" routine.

DESCRIPTION:

The tape is stepped one block in the direction opposite to that of the original instruction (without storing). The original tape operation is then repeated. If a tape check failure results from either of these operations a program stop will occur on the 7th line of the routine. The end of block word of the block which was read will be in register storage, the read instruction operated will be in the 6th line of the routine, and the blue lights on the indicator panel will tell exactly what is wrong with the block. If a tape check failure does not occur on the second try, control will be returned to line M+1 with the end of block word in register storage and the TCF indicator reset.

If one desires to continue, and assume that the block was read in satisfactorily, the TCF indicator should be reset and the machine started on the 8th line of the routine, this will return control to M+1. If one wishes to try to read the block again, push the start button.

MEMORY LOCATIONS REQUIRED: 17

AUTHOR: W. P. Warner

CHECK WRITTEN BLOCK AND
REWRITE (IF NECESSARY) I

Routine No. 10.0021

TRANSFER TO ROUTINE:

Location M-1: "Write" or "Write output" instruction

Location M: 0860 M 0000 routine

DESCRIPTION:

The block written as the result of operating the instruction in location M-1 is verified backwards and forwards (without storing), thus repositioning the tape. If either reading is successful, control is returned to location M+1 with the TCF indicator reset. If both readings fail to verify the block is deleted, rewritten and verified backwards and forwards again. If both of these readings fail a program stop occurs. Otherwise control is returned to M+1.

MEMORY LOCATIONS: 30

REMARKS:

Routine #10.0028 is preferable since it requires both readings to be correct.

If an EOF label is passed while a block is being written, the EOF indicator will remain "on" at the conclusion of the routine.

If the block written by the instruction in location M-1 is an EOF mark, the EOF indicator will be turned on by this routine and will remain on at the conclusion of the routine.

AUTHORS: Karl Kozarsky and Gene Gleissner

CHECK WRITTEN BLOCK AND
REWRITE (IF NECESSARY) II

Routine No. 10.0028

TRANSFER TO ROUTINE:

Location M-1: "Write" or "Write output" instruction

Location M: 0860 M 0000 routine

DESCRIPTION:

The block written as the result of operating the instruction in location M-1 is verified backward and forward without storing thus repositioning the tape. If both readings are successful, control is returned to location M+1 with the TCF indicator reset. If either reading fails to verify, the block is deleted, rewritten and verified backward and forward again. If either of these readings fails a program stop occurs. Otherwise control is returned to M+1.

MEMORY LOCATIONS: 24

REMARKS:

If an EOF label is passed while a block is being written, the EOF indicator will remain on at the conclusion of the routine.

If the block written by the instruction in location M-1 is an EOF mark, the EOF indicator will be turned on by this routine and will remain on at the conclusion of the routine.

AUTHOR: Mary L. Hagemeyer

FORMAT						MEANING TO COMPILER	RESULT IN COMPILED PROGRAM
Group Tag	Location	Command	R Operand	S Operand	T Operand		
		CSIX	TABLE*OF*SINES*AND*COSINES			Similar to CONE, but make six words. Asterisks indicate spaces.	7351526355006656 0072596555720051 6554005366725965 5572960000000000 9600000000000000 9600000000000000
		CLUE				Print this line when listing the symbolic coding, but do not put it into the compiled program.	

CORRECTION CODES

FORMAT						MEANING TO CORRECTION ROUTINE
Group Tag	Location	Command	R Operand	S Operand	T Operand	
			REPLACE	Y + 2	3	Replace the contents of 3 symbolic lines, starting at Y + 2, with the 3 lines following this one.
			NSERT	X + 4	1	Insert one line of coding after X + 4.
			REMOVE	WA + 50	2	Remove 2 lines, beginning with WA + 50.

APPENDIX C

Sample Setup Sheet and Coding Example

The next pages contain a sample setup sheet and an example of a program in the alphanumeric compiler language, followed by the machine-language coding which the compiler would produce. The program reads a block from tape, finds the block length, computes some values from the input, finds the largest of these values, prints the one computed just before the largest, and stops. Before converting the program, the compiler must order the coding and make up a subroutine from part of it.

NORC SETUP AND CONSOLE INFORMATION
PRNC-NPG-10462'5 (Rev. 8-58)

SETUP NUMBER

NOTE: In the event of a non-scheduled stop, complete the reverse of this form.

PROGRAMMER *(Name)*	*(Number)*	DATE	EST. COMPUTING TIME	WAIT FOR TAPES
PROBLEM		CHARGE TO CODE NO.	☐ CHECK OUT ☐ PRODUCTION	

TAPE SETUP

PROGRAMMER NUMBER								
REEL NUMBER(S)	Input	Output	T	T	Correction tape (optional)	Auxiliary subroutine tape (optional)	Service tape	
TAPE CODE	01	02	03	04	05	06	09	
MECHANISM								
CHECK OUTPUT REELS TO BE PRINTED								

CONDITION AND OPTION SWITCHES

	64	65	66	67	68	74	75	76	77	78	79	
STOP	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	STOP
						*	* *					TRANSFER
PROCEED	X	X	X	X	X	*	* *	X	X	X	X	OFF

MONITOR PRINTERS

	STANDARD BOARD	OTHER
1.	☐	☐
2.	☐	☐

START: ☐ STD ☒ NON-STD

STORAGE: ☐ 2000 ☒ 3600 ☒ 20 K

CRT PRINTER CAMERA SELECTION

	A		B
TRAID	X	*(Standard)*	☐ K-H
K-H	☐		☐ TRAID
BOTH	☐		☐ BOTH

NON-STANDARD START: 0994 0002 0007 9500

STOP: ☒ STD ☐ NON-STD

☐ STANDARD RE-START

SPECIAL INSTRUCTIONS

* "Transfer" if updating 06 tape or using subroutines from it. Otherwise "off".

* "Transfer" if making corrections. Otherwise "off".

PRNC-NWL-10462/56T (12/59)

NORC CODING SHEET NO. 3

REMARKS	GROUP TAG	LOCATION	COMMAND	R	S	T
			STAR	1	READ	
			CLUE	READ*AND*INITIALIZE		
Read and	DA	READ	0194	INPUT		
check			ROUT			TCF
			EOF			STOP
Find block	EA	LENGTH	0460		T19	
length - 1			0460			
—> PQ of T19			8841	T19		T19
T19 - 1 —> S field	FA		0841		ONE	
		ONE	CONS	1.0		
58 code + length			MADD	DUMMY58		LOOP
Set modifiers			57	2	OUTPUT	
			CLUE	COMPUTE*OUTPUT*AND*PRINT*VALUE*BEFORE*MAXIMUM		
$X_i - X_{i-1}$		COMPUTE	SUB	INPUT/4	INPUT/4-2	
Max - result	FM		SUB	MAXVAL		
			BRAN	MOD	MOD	
Result—> max			T	/6	MAXVAL	
			59			
M6' —> loc. max.			9942		S	LOCMAX
		S	CONS		1111	

C3

PRNC-NWL-10462/56T (12/59)

NORC CODING SHEET NO. 3

REMARKS	GROUP TAG	LOCATION	COMMAND	R	S	T
		MOD	50		1	
		LOOP	HALT			
			MADD	LOCMAX	DUMMY	PRINT
		PRINT	HALT			
		STOP	FINI			
	FA		MUL	INPUT + 1/4		/6
	GA		SUBR	LGTH + 99	25	XEAEA
		DUMMY	PRTA	△ 39	-1	1
		DUMMY58	0458	2		COMPUTE
		INPUT	SAVE	200		
		OUTPUT	SAVE	100		
		MAXVAL				
		LOCMAX				

Read	100	01	94	0150		
	101	05	68	2000	1010	0127
	102		67	2000	0000	0121
Length	103	04	60	4000	0000	0039
	104	04	60			
	105	88	41	2000	0390	0039
	106	08	41	4000		0147
	107		40	2000	1240	0117
	108		57	0002	0350	
Compute	109	50	22	4040	1504	0148
	110	50	24	2040	1516	0000
	111	50	22	4000	1250	0000
	112		63	4000	1160	0116
	113		60	4060	0000	0125
	114		59			
	115	99	42	0000	1480	0126
Mod	116		50		0001	
Loop	117		61	4000		
	118		60	2000	1230	
	119		40	2000	1260	0120
Print	120		61	4000		
Stop	121		80	6000	1220	0001
	122		61	0000	0000	0001
Dummy	123		81	3999	9990	0001
Dummy 58	124	04	58	0002		0109

1

Maxval	125					
Locmax	126					
TCF Routine	127					
	⋮					
	146					
One	147	00	01			
S	148				1111	
Phony used by compiler	149	99	99	9999	9989	9999
Input	150					
Output	350					

APPENDIX D

Notes About Instruction Arrangement

When the compiler starts to convert a line of symbolic coding, it first decides what format the line should have. If the line is a pseudo-code, it is treated as discussed in the section on pseudo-codes. If the R field begins with △ , the line becomes one instruction in the PQHJK format. If the line contains only digits, no more than four per field, in the Command, R, S, and T fields, it becomes one sixteen-digit word of four digits from each field.

If the line contains some symbolic or five-digit numbers in the R, S, or T fields, but it is not an instruction, i.e., the Q portion of its command field is less than 20, its conversion is as follows: the operand fields become five digits each and are placed in digit positions 15-11, 10-6, 5-1 for R, S, and T; then the command field is added into positions 16-13. Such a line might be input to a subroutine, and its command field would most likely be blank.

If the line is none of the above types, it must be an instruction of either the PQHJK or PQRST format.

The Q field is tested, and if it is in the ranges 50-59 or 90-99, the instruction is the PQRST type. Each operand field is converted and truncated to the low-order four digits. For the T field of a 58 code or the R field of a tape instruction the fifth digit is tested. If it is odd, eight is added to digit position 16.

When the compiler arranges an instruction in the PQHJK format, it examines each operand field to see if it can be placed in the H field. The T field is tested first, followed by S and R. If the T field is blank or contains nothing but a number between 21 and 39, the compiler

converts this and places it in the H field. If T does not fit this range, the compiler tests S similarly, then R. It must be emphasized that the T field is tested first. Thus the instruction

```
                                        Comm  R     S    T
"ADD ---- BETA 35"
```

```
P Q  H   J     K
```

is converted to "5020 55 00000 (BETA)". If no value can be found for the H field from these tests, the S and T fields are compared. If they are identical, 60 is placed in the H field. If not, each field is tested for an address of the type T1...T19, again starting with the T field. If one of these is found, it is converted and its absolute field is added to it. If the sum is still in the range of 21...39, it is changed as required by its field and placed in the H field. If none of the fields have this type of address, the symbolic portion of each one is compared with the symbolic locations defined by DEFT pseudo-codes. If nothing is found in this search, the instruction must be converted into two or three absolute instructions. The following table shows how this is done:

Q Field of PQRST	Compiled Results
20-49, 70-73, 85-89	006020 S 00000 P Q 20 R T
60-62, 80-84	006020 R 00000 P Q 00 S T
63, 74-79	P Q 40 R S 00Q 00 00000 T
64-69	L: P Q 20 R L + 2 006000 00000 L + 3 006000 S T