

RECOMP II USERS' PROGRAM NO. 1137

PROGRAM TITLE: CARD NUMERIC INPUT/OUTPUT ROUTINE

PROGRAM CLASSIFICATION: Service

AUTHOR: W. Wellman
Servomechanisms, Inc.

PURPOSE: To read Hollerith coded digits from card columns and convert that information to fixed point binary form for use by the RECOMP II. Also does the converse, punching card columns from binary integers.

DATE: July 1962

Published by

RECOMP Users' Library

at

AUTONETICS INDUSTRIAL PRODUCTS
A DIVISION OF NORTH AMERICAN AVIATION, INC.
3400 East 70th Street, Long Beach 5, California

DISCLAIMER

Although it is assumed that all the precautions have been taken to check out this program thoroughly, no responsibility is taken by the originator of this program for any erroneous results, misconceptions, or misrepresentations that may appear in this program. Furthermore, no responsibility is taken by Autonetics Industrial Products for the correct reproductions of this program. No warranty, express or implied, is extended by the use or application of the program.

CARD NUMERIC INPUT/OUTPUT ROUTINE

PURPOSE:

Reads Hollerith coded digits from card columns and converts that information to fixed point binary form for use by the Recomp II. Also does the converse, punching card columns from binary integers.

PUNCHING CARDS:

Calling Sequence

A) Using SM/I's trapping mode interpretive routine, 108R:
OCD 000N.0 (Output to Card)

B) For use without the trapping mode macro-instruction feature (the transfer instruction must be in the right half-word):

TRA (Location of this routine)
PZE 000N.0

Enter with a number in the accumulator at a "b" of 39.

Exit with a zero in the accumulator and garble in the R register, and the correct numeric digits punched on the card. In case of an error (number $\geq 10^N$ or $N > 13$ octal), the computer will exit to the "return" and halt with garble in the registers without punching anything.

Time: approximately $(80 + 90 N)$ milliseconds, or 8 seconds per 80 column card.

Code: N is the number of digits to be punched, in octal, in the sector position, and must be in the range $1 \leq N \leq 13$ octal.

READING CARDS:

Calling sequence:

A) Using SM/I's trapping mode interpretive routine, 108R:
ICD 000N.0 (Input from Card)

B) For use without the trapping mode macro-instruction feature (the transfer instruction must be in the right half-word):

TRA (Location of this routine + 2)
PZE 000N.0

Enter with anything in the registers.

Exit with the number in the accumulator, in binary, at a "b" of 39. In the event of an error ($N \geq 2^{39}$), the computer will exit to the "return" with garble in the registers, halt, and display the word in memory containing the code number N. In case of such an error, not all N card columns may be read, which will upset the card format. No error is possible if N is in the range $1 \leq N \leq 13$ octal.

Time: approximately $(90 + 120 N)$ milliseconds or 11 seconds per 80-column card.

Code: N is the number of digits to be read, in octal, in the sector position, and should be in the range $1 \leq N \leq 13$ octal.

Remarks: The Hollerith space (blank column) and " - " (11 column) will both be interpreted as the digit zero. Leading spaces (if any) are ignored and create a positive number; an initial " - " generates a negative number. A "space" or " - " at the end of a string of digits is treated as a zero (multiplies the number by 10) and makes the number negative. A space in the midst of a series of non-zero digits will produce a negative number which will probably be garbled.

The letters A - I and J - R are interpreted as the numbers 1 - 9, in sequence, which may be of use in some special application.

DESCRIPTORS:

Card, input, output, data, number, IBM, keypunch, punch, read, translate, binary, Hollerith, code.

Program No. _____ Title CARD NUMERIC INPUT / OUTPUT

Programmed by: W Wellman Date April 16, 1962

Loc'n	Cm'd	Addr.	Contents	Accumulator	b	Remarks
.00	C.T.L	.020				← input
. . .	SAX	7.77.1.0	junk			
. . . 1	ADD	7.7.6.50				
. . . .	T.RA	7.7.6.40				
. . . 2	C.T.L	.200				← output
. . . .	C.T.V	.400				
. . . 3	S.AX	7.7.7.7.0				
. . . .	T.RA	7.7.6.00	→ 20			
. . . 4	S.T.O	7.7.6.5.0				
. . . .	C.T.V	.300				
. . . 5	STA	7.7.7.2.1	exit			
. . . .	C.LA	00.0.1.0	(code)	code		
. . . 6	EXT	7.7.7.6.0		m		
. . . .	S.T.O	7.7.7.6.0	m			
. . . 7	C.T.L	.100				
. . . .	T.RA	7.7.6.00				
. . . 10	A.L.S	00.0.2.0				V0 = 30
. . . .	R.DY	7.7.7.1.0		hollerith pattern		
. . . 1	STA	7.7.7.4.1	temp	"		
. . . .	F.CA	7.7.7.4.0	-temp	" / 37039		
. . . 2	T.ZE	.7.1.0	→ space			
. . . .	A.RS	00.0.2.0		drop H 11, H 12		
. . . 3	T.ZE	.7.0.0	→ "-", "&			
. . . .	F.NM	0000.0				shift & count
. . . 4	X.AR	00.00.0		binary digit @ 39		
. . . .	S.T.A	7.7.7.4.1	-temp			
. . . 5	C.LA	7.7.7.3.0	N	N dc		
. . . .	D.S.R	7.7.7.7.0	0.1 @ 0	10. old N		
. . . 6	T.O.V	.7.2.0	→ error			
. . . .	A.DD	7.7.7.4.0	digit	new N		
. . . 7	T.O.V	.7.2.0	→ error			
. . . .	S.T.O	7.7.7.3.0		N		→ 30
. . . 20	A.DD	7.7.6.1.0	1 @ 38	code loc; exit		← 03
. . . .	S.T.O	7.7.6.1.0				
. . . 1	STA	7.7.7.2.1	exit			
. . . .	C.LA	000.1.0	(code)	code		
. . . 2	EXT	7.7.7.1.0		m		
. . . .	S.T.O	7.7.7.1.0	m			
. . . 3	A.DD	7.7.6.6.0	locn 10°			
. . . .	S.T.O	7.7.6.6.0				
. . . 4	C.LA	7.7.7.1.0	m	m		
. . . .	S.U.B	7.7.7.2.0	11 decimal	m-11		
. . . 5	T.PL	.7.2.0	→ error: m 11			
. . . .	C.L.S	7.7.7.7.1				
. . . 6	D.S.L	.4.4.0	(10 ^m)	(N ÷ 10 ^m) @ 0		
. . . .	T.O.V	.7.2.0	→ error			
. . . 7	C.T.L	.6.0.0				
. . . .	T.RA	7.7.6.00				→ 60

Program No. _____ Title _____

Programmed by: _____ Date _____

Loc'n	Cm'd	Addr.	Contents	Accumulator	b	Remarks
30	CLA	77.760	n	digits to go	..	← 17
...	SUB	77.750	1	" -1	..	
1	STD	77.760	n		..	
...	TPL	77.600	!	+ , 0 : repeat	..	
2	CLA	77.730	N		..	
...	TRA	~~~~~	EXIT	N @ 39	..	
3	+00	00.000) N in loops		..	
...	-0.0	00.000			..	
4	+0.0	00.000) digit " "		..	
...	-0.0	00.000			..	
5	+0.0	000.10	1 @ 18		..	
...	-0.0	00.230	3 @ 39		..	
6	+0.0	00.77.0) n in loops		..	
...	-0.0	00.000			..	
7	+06	31.46.0	0.1 @ 0		..	
...	+46	31.46.0			..	
40	TZE	77.62.0	→ 9		..	→ change sign bit
...	TRA	77.62.1	→ any other		..	→ punch as is
1	+00	00.77.0) n in loop		..	
...	-0.0	0000.0			..	
2	+0.0	00.13.0	11 decimal		..	
...	TRA	~~~~~	EXIT		..	
3	+0.0	1.000.0	1 @ 9		..	
...	-0.0	0000.0			..	
4	+00	00.0.10	1 @ 18		..	
...	-0.0	00.000			..	
5	+0.0	00.00.0	10 @ 39		..	
...	-0.0	00.0.5.0			..	
6	+0.0	00.00.00	10 ²		..	
...	-0.0	00.6.2.0			..	
7	+0.0	00.00.00	10 ³		..	
...	-0.0	0.764.0			..	
50	+0.0	0000.00	10 ⁴		..	
...	-0.1	1.6.1.00			..	
1	+0.0	000.000	10 ⁵		..	
...	-1.4	1.52.00			..	
2	+0.0	00.000.00	10 ⁶		..	
...	+7.2	044.00			..	
3	+0.0	00.0.4.1	10 ⁷		..	
...	+0.4	5.5.00.00			..	
4	+0.0	005.7.1	10 ⁸		..	
...	-5.7	02.0.00			..	
5	+0.0	07.3.4.1	10 ⁹		..	
...	+2.6	24.0.0.0			..	
6	+0.1	1.2.4.00	10 ¹⁰		..	
...	+3.7	1.0.00.0			..	
7	+1.3	5.1.03.1	10 ¹¹ @ 39		..	
...	-6.7	2.0.0.00			..	

Program No _____ Title _____

Programmed by: _____ Date _____

Loc'n	Cm'd	Addr.	Contents	Accumulator	b	Remarks
60	SUB	760	$8 \times 10^{-12} @ 0$			decimal round
	XAR	00.0.00				N → R
1	CLA	77.7.10	m			
	TRA	77.6.4.0				
2	ADD	77.7.4.0	1 @ 18	make sign 1		
	PNC	77.7.1.0	12 bits	Hollerith pattern		
3	CLA	77.7.1.0	m			
	SUB	77.7.4.0	1 @ 18			
4	TZE	77.7.2.1	→ exit			
	STO	77.7.1.0	m			
5	XAR	00.0.00		N @ 0		
	MPY	77.7.5.0	10 @ 39	digit @ 39		
6	ALS	00.0.1.0		" @ 38 = d		
	STA	77.6.7.1		d		
7	CLS	77.7.3.0	1 @ 9			
	ALS	~~~~~	d	1 @ 9 - d		
70	CLS	3.4.1	digit	- 0		" = " 12
	STO	77.7.4.0	digit			
1	CLS	77.7.3.1		- N		" = " 13
	TRA	77.6.5.1	→			→ change sign of N → 15
2	CLA	77.7.2.0				← error
	STA	7.3.1				
3	STA	7.5.1				set error exit
	CLA	~~~~~				
4	STO	77.7.7.0				
	DIS	77.7.7.0				display code word
5	ARS	5.0.0				
	HTR	~~~~~	ERROR : EXIT			
6	+00	0000.0				
	-00	0.0.0.2.0	2 @ 38 ~ 8×10^{-11}			
		X/				
		X/				