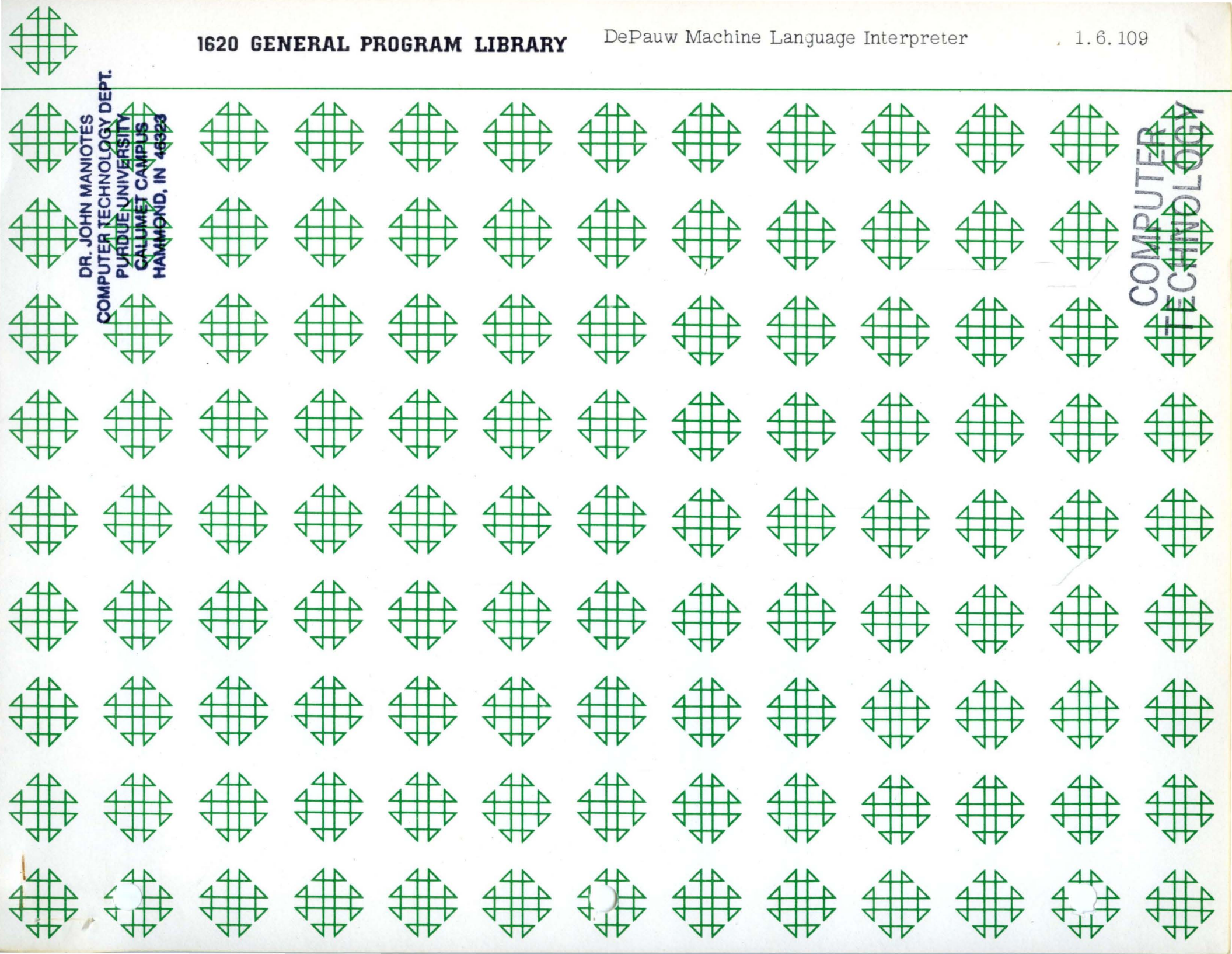


DR. JOHN MANIOTES  
COMPUTER TECHNOLOGY DEPT.  
PURDUE UNIVERSITY  
CALLUMET CAMPUS  
HAMMOND, IN 46323

COMPUTER  
TECHNOLOGY



HARRISON IN 4823  
CALLUMT CARPUS  
RUBENS UNIVERSITY  
COMPUTER TECHNOLOGY DEPT  
DR JOHN HARRISON

0

0

0

DISCLAIMER

Although each program has been tested by its contributor, no warranty, express or implied, is made by the contributor or 1620 USERS Group, as to the accuracy and functioning of the program and related program material, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the contributor or 1620 USERS Group, in connection therewith.

0

0

0

1620 USERS GROUP PROGRAM REVIEW AND EVALUATION

Program No. \_\_\_\_\_

Date \_\_\_\_\_

Program Name: \_\_\_\_\_

1. Does the abstract adequately describe what the program is and what it does? Yes \_\_\_ No \_\_\_  
Comment \_\_\_\_\_
2. Does the program do what the abstract says? Yes \_\_\_ No \_\_\_  
Comment \_\_\_\_\_
3. Is the Description clear, understandable, and adequate? Yes \_\_\_ No \_\_\_  
Comment \_\_\_\_\_
4. Are the Operating Instructions understandable and in sufficient detail? Yes \_\_\_ No \_\_\_  
Comment \_\_\_\_\_  
Are the Sense Switch options adequately described (if applicable)? Yes \_\_\_ No \_\_\_  
Are the mnemonic labels identified or sufficiently understandable? Yes \_\_\_ No \_\_\_  
Comment \_\_\_\_\_
5. Does the source program compile satisfactorily (if applicable)? Yes \_\_\_ No \_\_\_  
Comment \_\_\_\_\_
6. Does the object program run satisfactorily? Yes \_\_\_ No \_\_\_  
Comment \_\_\_\_\_
7. Number of test cases run \_\_\_\_\_  
Are any restrictions as to data, size, range, etc. covered adequately in description? Yes \_\_\_ No \_\_\_  
Comment \_\_\_\_\_
8. Does the Program meet the minimal standards of the 1620 Users Group? Yes \_\_\_ No \_\_\_  
Comment \_\_\_\_\_
9. Please list any suggestions to improve the usefulness of the program. These will be passed on to the author for his consideration.  
Comment \_\_\_\_\_

Please return to:

Mr. Robert J. Robinson (PREP)  
Marquette University  
Computing Center  
1515 W. Wisconsin Avenue  
Milwaukee 3, Wisconsin

Your Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
User Group Code \_\_\_\_\_

THIS REVIEW FORM IS PART OF THE 1620 USER GROUP ORGANIZATION'S PROGRAM REVIEW AND EVALUATION PROCEDURE. NONMEMBERS ARE CORDIALLY INVITED TO PARTICIPATE IN THIS EVALUATION.



DEPAUW MACHINE LANGUAGE INTERPRETER

Roger B. Nelsen  
Computer Center  
DePauw University  
Greencastle, Indiana 46135

Users Group Code 3225

April 23, 1964





ABSTRACT

TITLE DePauw Machine Language Interpreter

SUBJECT CLASSIFICATION Utility - Miscellaneous General Purpose

CODE 1.6

AUTHOR Roger B. Nelsen  
Computer Center (Users Group Code 3225)  
DePauw University  
Greencastle, Indiana 46135

DIRECT INQUIRIES TO: the Author

DESCRIPTION The program is designed to interpret a deck of machine language cards, each card containing six machine language orders in card columns 1-72. The program separates the numeric operation code and the P and Q addresses, and also supplies the mnemonic operation code and the address of each order. The output is designed to resemble the coding sheet form X26-5591-0. The purpose of this program is to aid the programmer in debugging machine language routines and also provides a convenient method for listing student machine language programs.

METHOD N/A

RESTRICTIONS N/A

SPECIFICATIONS IBM 1620 Card System, 20K Memory, Indirect Addressing

LANGUAGE Basic Machine Language

DECK KEY

1. Object Deck

## PROGRAM WRITEUP

1. DePauw Machine Language Interpreter
2. April 23, 1964
3. Roger B. Nelsen  
Computer Center (Users Group Code 3225)  
DePauw University  
Greencastle, Indiana 46135
4. The purpose of this program is to provide a routine to facilitate the debugging and subsequent correcting of a machine language program. The author has found it quite inconvenient to try to debug such a program from either the cards or a numeric dump output - this program is designed to type out a program in a form similar to a coding sheet. Cards are read into the computer numerically (see I/O Formats and Restrictions) and then each 12 digit order is separated into its numeric operation code and P and Q addresses. Each numeric operation code is then interpreted into a mnemonic code by use of indirect addressing.

Alphameric data representing the mnemonic symbol for operations 00-99 is stored in locations 11001 through 11991. Each is stored as 1-4 letters and a record mark, or 5 alphameric characters (10 numeric). Then each numeric operation code that is found on a card is preceded by  $\bar{1}$  and followed by a 1, so that a numeric code of xx will appear as  $\bar{1}xx1$ . The address of the  $\bar{1}xx1$  is 14102. To type out the alphameric operation code, the program types out from the indirect address  $1410\bar{2}$ , which then uses  $\bar{1}xx1$  as an address, which is the desired alphameric operation code symbol.

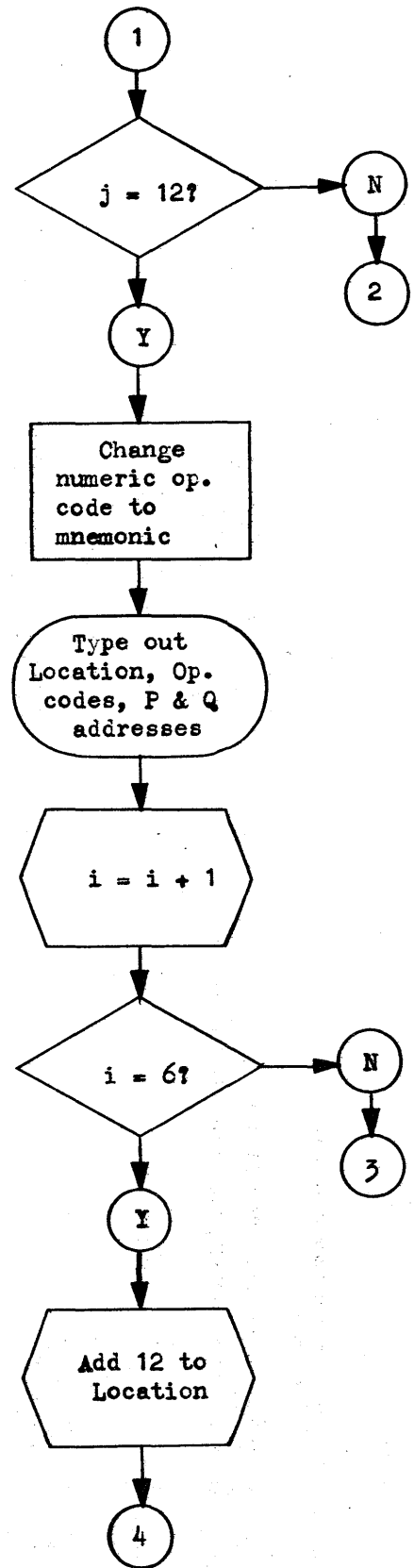
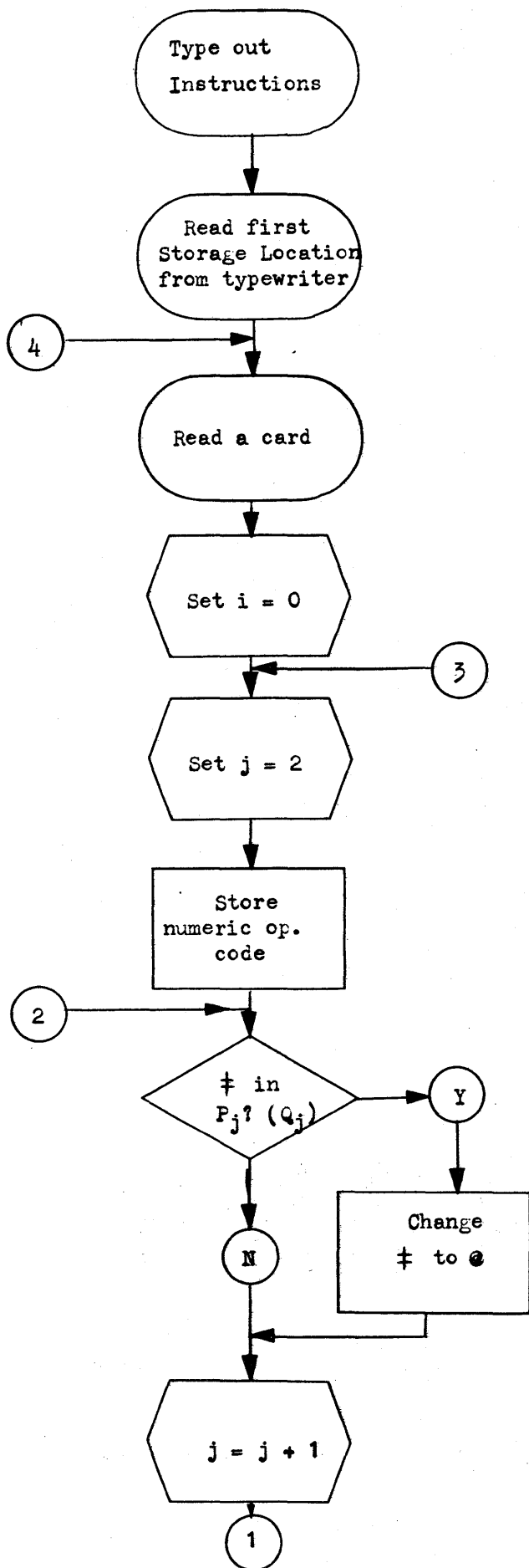
Occasionally, an instruction will contain a record mark in either the P or Q address. Since a record mark terminates transmission of data on a Write Numeric order, this record mark must be changed to some symbol which will be typed out. The program scans each P and Q address, and changes record marks to @, which does not terminate transmission.

Since this program uses the cards of the program to be interpreted as data, the interpreted program is not stored in the computer. To actually execute the program, this interpreter must be cleared and the program run in the usual manner.

5. I/O Formats. The input for this program is a deck of machine language cards. Each card may contain up to six (6) machine language instructions, in card columns 1-72. The last eight columns (73-80) are not read and will not be interpreted. It might be added here that the author has found it expedient to write machine language programs in this manner; orders do not begin on one card and end the next, and these last eight columns can be used for numbering the cards sequentially. The output is via typewriter (see Sample Output).

6. Restrictions. Since the program determines the operation code and the P and Q addresses according to location on the input data card and subsequent location in memory, the program to be interpreted must be in the form given under I/O format, and six orders is the maximum per card. Also, since record marks are changed to @, both record marks and @ will be typed out as @.
7. Stops. No stops have been observed by the author after many runs of this program. However, the operator is cautioned to use only order cards; other types of numeric and alphameric data input will yield unusable output, and could quite conceivably cause stops.
8. Detailed Operating Instructions.
  1. Clear Core Storage (enter 260000800009RS, Instant Stop, Reset)
  2. Load Object Deck. After loading, the computer will type out three lines of information, and halt for the entry of the first desired storage location.
  3. Enter this location, for example, 00402RS
  4. The computer will type out column headings and halt.
  5. Load deck to be interpreted and push Start.
  6. Machine will interpret instructions as long as there are cards in the read hopper.
  7. To enter a new deck to be interpreted:
    - a. Push Reset and Insert.
    - b. Type in 4900606RS
    - c. Type in new initial storage location and proceed as in 3. above.
9. Equipment Required. IBM 1620 Card System, 20K Memory, Indirect Addressing.
10. Written in Basic Machine Language.
11. Additional Remarks.
  1. The Load Routine. The load routine for this program (6 cards, including add and multiply tables) can be easily modified for use for any machine language program in the 6 orders per card format. The last five columns (76-80) of the first card contain the address of the first instruction in the program; here it is 00402. Card columns 44-48 on the sixth card contain the address of the first order on the last program card plus 72. Here it is 01410. The operator need only change these addresses to suit his own use. Both must contain flags on the high-order digit. By use of a compare statement, loading is automatically terminated and the program is executed.
  2. Sample Output. The sample output included in this documentation is derived from the instruction portion of this program deck. The program has in effect then, interpreted itself.

FLOW DIAGRAM



i counts 6 instructions per card  
 j counts 10 digits per P and Q addresses

SAMPLE OUTPUT: Interpretation of Instruction Cards  
of this Program.

DEPAUW MACHINE LANGUAGE INTERPRETER FOR 6-ORDER CARDS.  
@ INDICATES RECORD MARK.  
ENTER FIRST DESIRED STORAGE LOCATION, E.G. 00402 RELEASE-START.

00402RS

LOC.	OPERATION	P	Q	COMMENTS.
00402	RA	37	10001	00500
00414	RA	37	10161	00500
00426	RA	37	10321	00500
00438	RA	37	11001	00500
00450	RA	37	11161	00500
00462	RA	37	11321	00500
00474	RA	37	11481	00500
00486	RA	37	11641	00500
00498	TFM	16	14099	00011
00510	TDM	15	14102	00001
00522	SF	32	12999	00000
00534	B	49	01338	00000
00546	WA	39	10161	00100
00558	K	34	00000	00102
00570	WA	39	10321	00100
00582	K	34	00000	00102
00594	K	34	00000	00102
00606	RN	36	13000	00100
00618	K	34	00000	00102
00630	K	34	00000	00102
00642	WA	39	10221	00100
00654	K	34	00000	00102
00666	TD	25	13005	00400
00678	RN	36	14000	00500
00690	TD	25	14207	00400
00702	TD	25	14312	00400
00714	TDM	15	14102	00001
00726	SF	32	14000	00000
00738	TF	26	14101	14001
00750	CF	33	14100	00000
00762	BNR	45	00786	14002
00774	TDM	15	14002	0000@
00786	TD	25	14202	14002
00798	AM	11	00773	00001
00810	AM	11	00780	00001
00822	AM	11	00792	00001
00834	AM	11	00797	00001
00846	CM	14	00773	14006
00858	BNR	47	00762	01100
00870	CM	14	00773	14011
00882	BI	46	00930	01100
00894	AM	11	00792	00100

00906	AM	11	00857	00005
00918	B	49	00762	00000
00930	WN	38	13000	00100
00942	K	34	00000	00101
00954	AM	11	00857	00007
00966	AM	11	00881	00012
00978	K	34	00000	00101
00990	WA	39	14102	00101
01002	K	34	00000	00101
01014	TD	25	14102	00400
01026	WN	38	14100	00100
01038	NOP	41	00000	00000
01050	K	34	00000	00101
01062	K	34	00000	00101
01074	WN	38	14202	00100
01086	K	34	00000	00101
01098	WN	38	14307	00100
01110	K	34	00000	00102
01122	AM	11	00749	00012
01134	AM	11	00773	00002
01146	AM	11	00780	00002
01158	AM	11	00797	00002
01170	SM	12	00792	00110
01182	AM	11	00732	00012
01194	AM	11	13004	00012
01206	CM	14	00749	14061
01218	BNR	47	00714	01100
01230	K	34	00000	00102
01242	TFM	16	00732	14000
01254	TFM	16	00749	14001
01266	TFM	16	00773	14002
01278	TFM	16	00780	14002
01290	TFM	16	00797	14002
01302	TFM	16	00857	14006
01314	TFM	16	00881	14011
01326	B	49	00678	00000
01338	RA	37	11801	00500
01350	RA	37	11961	00500
01362	K	34	00000	00102
01374	WA	39	10001	00100
01386	K	34	00000	00102
01398	B	49	00546	00000

9.

PROGRAM DECK LISTING

1. Load Routine (includes Add and Multiply Tables

```

36001000050036001800050036002600050036003400050026000840007936000000050000000402
0000000000010203040002040608000306090210040802161005001510200602181420070411282
0080014223009081726300000000050607080900121416181518112427202428223635203530454
03632484455324946536048465462754453627180123456789123456789023456789013456789012
456789012356789012346789012345789012345689012345679012345678#000000000000000000
2600018000843600000005001100018000721400018014104700012012002600078000844900000#
    
```

2. Instructions

```

37100010050037101610050037103210050037110010050037111610050037113210050000000000
37114810050037116410050016140990001151410200001321299900000490133800000000000000
39101610010034000000010239103210010034000000010234000000010236130000010000000000
34000000010234000000010239102210010034000000010225130050040036140000050000000000
2514207004002514312004001514102000013214000000002614101140013314100000000000000
4500786140021514002000025142021400211007730000111007800000111007920000100000000
1100797000011400773140064700762011001400773140114600930011001100792001000000000
11008570000549007620000038130000010034000000010111008570000711008810001200000000
340000000101391410700101340000000101251410200400381410000100410000000000000000000
34000000010134000000010138142020010034000000010138143070010034000000010200000000
11007490001211007730000211007800000211007970000212007920011011007320001200000000
11130040001214007491406147007140110034000000010216007321400016007491400100000000
16007731400216007801400216007971400216008571400616008811401149006780000000000000
37118010050037119610050034000000010239100010010034000000010249005460000000000000
    
```

3. Alphameric Data Cards

```

DEPAUW MACHINE LANGUAGE INTERPRETER FOR 6-ORDER CARDS.#
@ INDICATES RECORD MARK.# LOC. OPERATION P Q COMMENTS.#
ENTER FIRST DESIRED STORAGE LOCATION, E.G. 00402 RELEASE-START.#
TFM #BTM #LDM #DM # # #A #S #M #C #TD #TF #BT #LD #D # #TR #
SF #CF #K #DN #RN #RA #WN #WA # # #NOP #BB #BD #BNF #BNR #BI #BNI #
H #B # # # # # # #BNG # # # # # # # #
# # # # # # # #MF #TNS #TNF # # # # # # # #
# # # # # # # # # # # # # # # # # #
    
```

# COMPUTER TECHNOLOGY

THE COMPUTER MUSEUM HISTORY CENTER  
  
1 026 2031 3