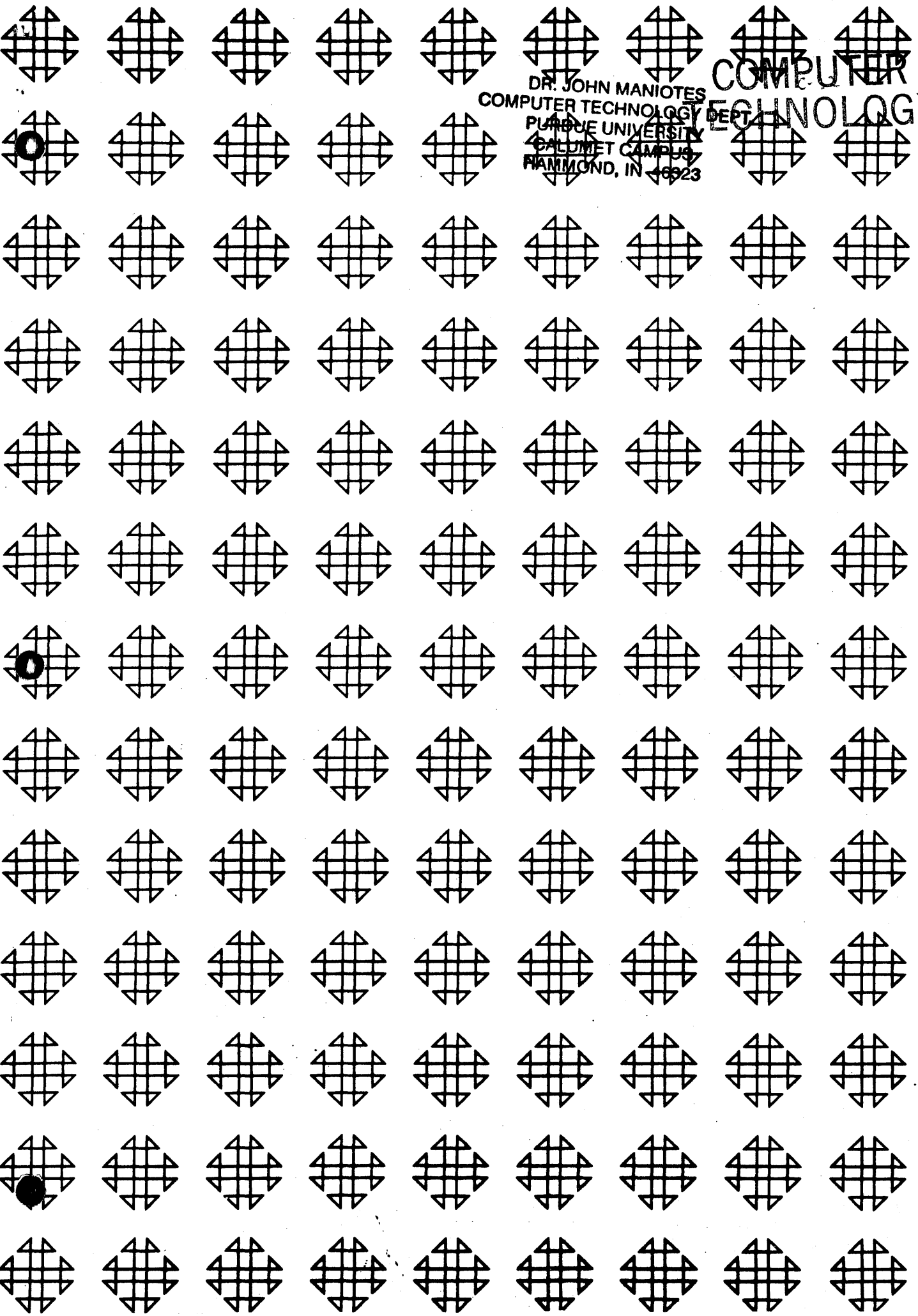


COMPUTER  
TECHNOLOGY

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



1620 GENERAL PROGRAM LIBRARY

Critical Path Node and Dummy Assignment  
Technique (CPNAT)

10.3.030

DR. JOHN MAMONTE  
COMPUTER TECHNOLOGY DEPT  
Purdue University  
CAMPUS  
HAMMOND, IN 46323

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.

1620 USERS GROUP PROGRAM REVIEW AND EVALUATION

(fill out in typewriter or pencil, do not use ink)

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. Were all necessary parts of the program received? Yes \_\_\_ No \_\_\_  
Comment \_\_\_\_\_

10. Please list on the back any suggestions to improve the usefulness of the program. These will be passed onto the author for his consideration.

Please return to:

Mr. Richard L. Pratt  
Data Corporation  
7500 Old Xenia Pike  
Dayton, Ohio 45432

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.

A3

CRITICAL PATH NODE  
AND  
DUMMY ASSIGNMENT TECHNIQUE (CPNAT)

RONALD J. SRODAWA  
  
Computer Center  
The University of Detroit  
Detroit 21, Michigan

Modifications or revisions to this program, as they occur, will be announced in the appropriate Catalog of Programs for IBM Data Processing Systems. When such an announcement occurs, users should order a complete new program from the Program Information Department.

A4

DECK KEY

1. Source Deck - 660 Cards
2. Object Deck - 157 Cards
3. Sample Problem Deck - 50 Cards  
Sample Problem I - Cards 01001 to 01020  
Sample Problem II - Cards 02001 to 02030

1620 USERS GROUP LIBRARY  
PROGRAM ABSTRACT

1. TITLE (If subroutine, state in Title): Critical Path Node And Dummy Assignment  
Technique Program ( CPNAT ) 10.3.030
2. Author; Organization: Ronald J. Srodawa Computer Center  
University of Detroit
- Date: August 8, 1963 Users Group Membership Code: 3006
3. Direct Inquiries to Name: Ronald J. Srodawa Computer Center  
University of Detroit Detroit, Michigan, 482 Phone: 342-1000 Ext 325
4. Description/Purpose: (5. Method; 6. Restriction/Range; When Applicable) This program  
reads a list of jobs along with the prerequisites ( a list of the  
jobs which must be performed before the job in question ), time  
duration and costs for each job. It then assigns node numbers,  
generates any needed dummies, and punches out an output deck which  
can be used as input for 10.3.011 . The restrictions are that  
3A+4B+4C+2D must be less than 3022, where A is the number of jobs,  
B is the number of prerequisites, C is the number of jobs with no  
prerequisites, and D is the number of jobs not used as a prerequisite.  
Add 5000 to the upper limit for every additional 20K of memory.
7. Specifications (Check or fill in appropriate spaces):
- a. Storage used by program: 20K-100K Automatically adjusts to memory size.
- b. Equipment required by program:  
Card System X ; Magnetic Tape System      ; No. of Tapes      ;  
Paper Tape System      ; Disk File System      ; No. of Packs      ;  
TNS, TNF, MF      ; Auto divide      ; Indirect addressing      ; Floating point hardware      ;  
Other (specify)
- Can program be used on lesser Machine? Yes . Specify which requirements can be easily removed I/O devices can be changed to fit machine configuration.
- c. Programming type (Check appropriate spaces):  
Fortran without Format      ; Fortran with Format      ;  
Fortran II      ; Mainline, Complete      ; Subroutine or function subprogram(S or F)      ;  
Is the program a library (ie, SPS) function to the Fortran system checked?      ;  
SPS      ; SPS - 1620/1710 X ;  
Mainline, Complete X ; Macro      ; Subroutine      ;  
Other programming language:      ; Give details
- d. Language used in the writeup: SPS listing with machine code. Writeup in Eng.
8. Additional Remarks: This program generates input to program 10.3.011  
( Miss Less) by Ray N. Sauer . The sample II job is processed  
by CPNAT in about 35 seconds. The program was developed through  
part time work over a period of about six months.

TABLE OF CONTENTS

	Page
Description . . . . .	1
Purposes and Uses . . . . .	1
Inquiries . . . . .	2
Restrictions . . . . .	2
Method of Computation . . . . .	4
Input Card Description . . . . .	7
Output Card Descriptions . . . . .	9
Recommended Job Procedure . . . . .	11
Sample Problem I . . . . .	12
Sample Problem II . . . . .	19
Operating Instructions . . . . .	26
Expected Stops and Action to be Taken . . . . .	27
Error Messages . . . . .	28
Errors While Running Miss Less . . . . .	29
Core Layout . . . . .	30
Likely Changes . . . . .	30
Flow Chart . . . . .	32
Program Listing . . . . .	51

Description:

The Critical Path Node and Dummy Assignment Technique (CPNAT) is a program which makes the manual assignment of nodes and dummy jobs, and the drawing of an arrow diagram before the running of a least cost estimating and scheduling program unnecessary. All that is required is to write a list containing each real job, its estimated time and cost, and the other real jobs which must be done immediately preceding to this job. This is punched into cards and used as input to CPNAT. This program then assigns nodes to each real job and generates dummy jobs to satisfy the prerequisites. It then punches out an output deck which can be used directly as input to the MISS-LESS program.

Purposes and Uses:

By reducing the amount of labor necessary to prepare a run for MISS-LESS, this program should reduce the overall time required to set up and obtain information from a scheduling program. Also, it reduces the opportunities for time consuming human errors. Also, runs which have been very difficult to draw arrow diagrams for are easily coped with. For instance, Dr. Kenney, at the University of Detroit has been experimenting with the breaking down of Chemistry courses into the various topics which compose them, and rearranging these topics using MISS-LESS. The networks which

resulted were very complicated and could not easily be drawn by hand. CPNAT was developed to cope with this situation and has done so very nicely. To conclude then, CPNAT was developed to reduce the time necessary for setting up a run for MISS-LESS. It is especially useful with runs which require very complicated arrow diagrams. Because it assigns the nodes and generates dummy jobs automatically, it reduces the room for human error and results in very efficient arrow diagrams.

Inquiries:

Any inquiries should be addressed to:

Ronald J. Srodawa  
Computer Center  
University of Detroit  
Detroit 21, Michigan

Restrictions:

$3A + 4B + 4C + 2D$  should be less than or equal to 3022 where A is the number of jobs, B is the number of prerequisites, C is the number of jobs with no prerequisites, and D is the number of jobs not used as prerequisites. Add 5000 to the upper limit for every additional 20K of memory. Under certain conditions, another temporary table may be built by the program. It would be

wise to subtract 15 or 20 from the upper limit to be certain that the program will not run out of storage if it generates this short table. Also, A itself must be less than 5000, otherwise arithmetic overflows will result.

#### METHOD OF COMPUTATION

The program reads in the job cards and builds a table of jobs along with the node numbers which it assigns consecutively. It also builds a table consisting of each job and its prerequisites. After this another table is built of all dummies needed to satisfy all the prerequisites. A series of algorithms is performed to weed out the unnecessary dummies and change the numbering of the corresponding nodes. The cards are then read in again, the necessary information looked up in the tables and pulled off the input cards, and the output cards punched. All the tables are floating, that is, the next table begins where the last table ended, thus using all available storage.

The following is a more complete description of the algorithms employed.

1. Initialize all storage and indicators.
2. Assign 0001 as the start.
3. Read each job card, assign an I and J to each job and store along with its number in a table.  
    ex: First Job (2,3)  
        Second Job (4,5)
4. Store the job number along with its prerequisites in a prerequisite table.
5. Define the end to be the J of the last job defined plus 1.

6. Assign dummies to correspond to the prerequisites.

ex: If (10,11) has prerequisite (2,3)  
assign the dummy (3,10).

7. If a job has no prerequisites, assign a dummy  
(1,I) where I is the I of the job.

8. If a job is not used as a prerequisite, assign a  
dummy connecting its J with the end found in 5.

Perform the following tests using each dummy in turn as  
the test dummy. (I<sub>0</sub>,J<sub>0</sub>) are the nodes of the test  
dummy, (I<sub>i</sub>,J<sub>i</sub>) the nodes of the other jobs or dummies.

9. If no other dummy or job has the same J value as  
the test dummy, find the job whose I value equals  
the J of the dummy, set this I value equal to  
the I of the test dummy and remove the dummy.

10. If there exists a dummy or job such that J<sub>i</sub> equals  
J<sub>0</sub>; then:

If there exists a dummy or job such that I<sub>i</sub> equals  
I<sub>0</sub>, do not change or remove anything.

If no other job or dummy exists such that I<sub>i</sub> equals  
I<sub>0</sub>, find all jobs or dummies for which J<sub>i</sub> equals  
I<sub>0</sub>, and for these jobs and dummies set J<sub>i</sub> equal  
to J<sub>0</sub> and remove the dummy.

After repeating the above algorithm for all dummies,  
proceed.

11. Find a job having only dummies going into it and  
save the I's of all these dummies.

12. Scan for other jobs having only dummies going  
into them, if each dummy of the second has an  
I corresponding to an I of a dummy of the  
first, save the job number. This must be a  
1 to 1 correspondence.

13. If another job or dummy has an I which is  
equal to the I of any of the saved jobs,  
change it to the I of the first job.

14. Remove any dummy which has a J equal to the I  
of any of the saved jobs.

15. Change any J which is the same as the I of  
the saved dummies (any one) to the I of the  
first job.

16. Change any I which is the same as the I of a  
saved dummy to the I of the first job.

After the preceding algorithm is completed for all jobs:

17. If there exist any dummies which are identical,  
remove all but one of them.

18. Reread the input and punch the output.







#### RECOMMENDED JOB PROCEDURE

1. Make a list of all the jobs to be performed, and assign a unique number to each.
2. From the list in 1, compile a list of each job, its time duration, estimated cost, and the numbers of the jobs which must be completed immediately before it is started.
3. From the list in 2, punch up the input for CPNAT.
4. Run the input data using CPNAT.
5. If desired, draw an arrow diagram using the nodes and dummies assigned by CPNAT. This step may be omitted but will make it easier to find errors and incorporate later changes.
6. Correct any errors found by CPNAT either directly in the output deck or in the input deck which would then have to be rerun.
7. After a correct output deck is obtained, and any other desired information has been punched into it, run it with MISS LESS to obtain a critical path scheduling.
8. If any later changes are necessary, they can be made directly in the output of CPNAT or in the input in which case they would have to be rerun with CPNAT.

#### SAMPLE PROBLEM I

The following sample problem is one of the trial runs used to test the application of least cost estimating and scheduling to the assignment of course topics by Dr. Kenney. Each job number refers to one chemistry course topic, such as Avogadro's Law, Equation Balancing, etc. As can be seen, the job results in a very complicated arrow diagram. The output from MISS LESS has been sorted down by earliest start date to make the results easier to use. This problem took 1 1/2 minutes to run on CPNAT.

A LIST OF REAL JOBS WITH TIME DURATION, COST, AND PREREQUISITES

Job	Time	Cost	Prerequisites
100	5		-
110	10		100
120	8		110
130	4		140
140	15		120
150	5		110
160	3		200
170	1		110
180	1		-
190	2		180
200	25		140,150
210	15		200,220
220	15		150,180
315	10		314,306
303	10		109,102
305	20		112
306	10		305,303
308	10		105
309	3		105
310	30		308
311	20		310
312	10		309,311,305
313	15		312

A LIST OF REAL JOBS WITH TIME DURATION, COST, AND PREREQUISITES  
(Continued)

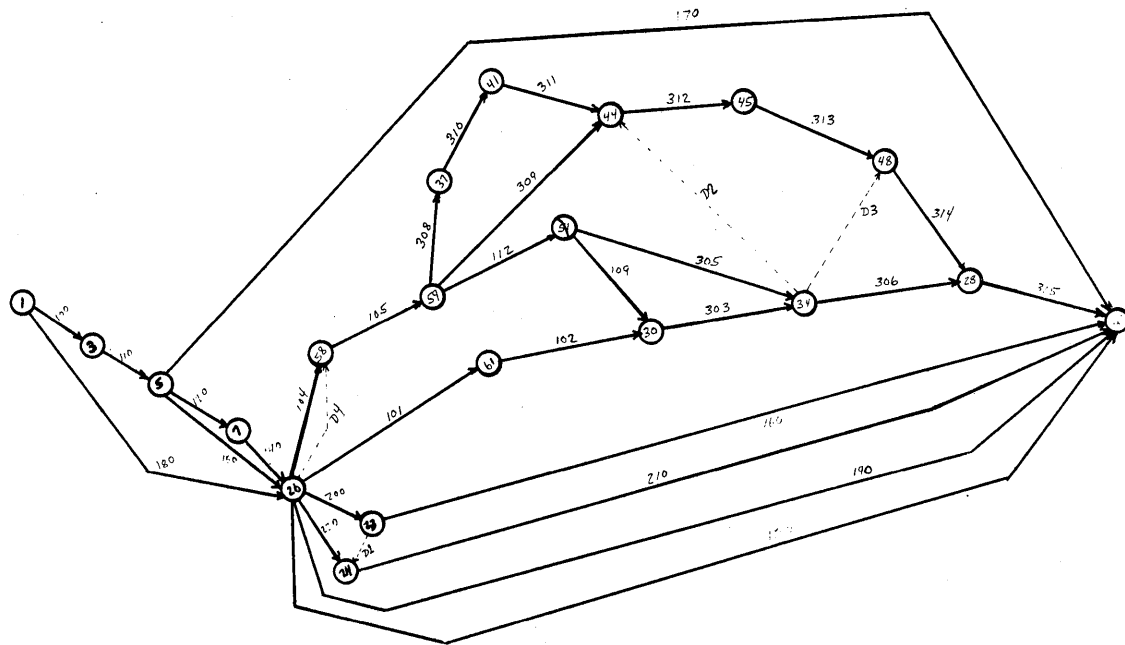
Job	Time	Cost	Prerequisites
314	30		313,303
112	10		105
102	3		101
109	20		112
104	20		180
105	5		180,104
101	2		180

CPNAT Input:

0005	0100
0010	01100100
0008	01200110
0004	01300140
0015	01400120
0005	01500110
0003	01600200
0001	01700110
0001	0180
0002	01900180
0025	020001400150
0015	021002000220
0015	022001500180
0010	031503140306
0010	030301090102
0020	03050112
0010	030603050303
0010	03080105
0003	03090105
0030	03100308
0020	03110310
0010	0312030903110305
0015	03130312
0030	031403130303
0010	01120105
0003	01020101
0020	01090112
0020	01040180
0005	010501800104
0002	01010180

CPNAT Output:

000100030005	0100
000300050010	0110
000500070008	0120
002600620004	0130
000700260015	0140
000500260005	0150
002300620003	0160
000500620001	0170
000100260001	0180
002600620002	0190
002600230025	0200
002400620015	0210
002600240015	0220
002800620010	0315
003000340010	0303
005100340020	0305
003400280010	0306
005900370010	0308
005900440003	0309
003700410030	0310
004100440020	0311
004400450010	0312
004500480015	0313
004800280030	0314
005900510010	0112
006100300003	0102
005100300020	0109
002600580020	0104
005800590005	0105
002600610002	0101
00230024 00	0000001
00340044 00	0000002
00340048 00	0000003
00260058 00	0000004



Arrow Diagram drawn from CPNAT Output

MISS LESS Output:

000100030005	0100	5	5	*	
000100026001	0180	1	37	38	37
000300050010	0110	15	23	15	*
000500070008	0120	15	20	33	38
000500260005	0150	15	16	187	188
000500620001	0170	23	38	23	38
000700260015	0140	38	42	184	188
002600620004	0130	38	40	186	188
002600620002	0190	38	63	148	173
002600230025	0200	38	53	158	173
002600240015	0220	38	58	38	58
002600580020	0104	38	40	108	110
002600610002	0101	38	38	58	58
00260058 00	0000004	38	38	58	58
006100300003	0102	40	43	110	113
005800590005	0105	58	63	58	63
002300620003	0160	63	66	185	188
002400620015	0210	63	78	173	188
005900370010	0308	63	73	63	73
005900440003	0309	63	66	120	123
005900510010	0112	63	73	83	93
00230024 00	0000001	63	63	173	173
005100340020	0305	73	93	103	123
003700410030	0310	73	103	73	103
005100300020	0109	93	93	93	113
003000340010	0309	93	103	113	123
003400280010	0306	103	113	168	178
004100440020	0311	103	123	103	123
00340044 00	0000002	103	103	123	123
00340048 00	0000003	103	103	148	148
004400450010	0312	123	133	123	133
004500480015	0313	133	148	133	148
004800280030	0314	148	178	148	178
002800620010	0315	178	188	178	188
- PROJECT COST		PROJECT COMPLETION	188		

## SAMPLE PROBLEM II

The following sample problem was run using CPNAT and MISS LESS according to the recommended procedure already outlined. The problem is the same one which appears in the MISS LESS abstract, except that the nodes and dummies were assigned by CPNAT, instead of first making an arrow diagram of the job. The following items are contained on the following pages:

1. A list of the real jobs with unique numbers assigned to them.
2. A list of the real jobs by number with the time duration, estimated cost, and prerequisites.
3. A listing of the input cards for CPNAT.
4. A listing of the output cards from CPNAT.
5. An arrow diagram as drawn from the listing of the output cards from CPNAT using its node numbers and generated dummies.
6. A final listing as obtained from MISS LESS by running the output from CPNAT as input.

This sample job is the one which was used in the debugging of CPNAT. It takes about 35 seconds to process this job, not including I/O time. Input and output are both read and punch bound.

## A LIST OF REAL JOBS WITH ASSIGNED UNIQUE NUMBERS:

JOB	#
Lead Time	1
Assemble Crews	2
Measure and Sketch	3
Develop Material List	4
Time Available	5
Deactivate Line	6
Erect Scaffold	7
Procure Pipe	8
Procure Valves	9
Prefab Sections	10
Remove Old Pipe	11
Place New Pipe	12
Weld Pipe	13
Place Valves	14
Insulate	15
Fit up	16
Pressure Test	17
Remove Scaffold	18
Clean Up	19
Desired Completion	20

A LIST OF REAL JOBS WITH TIME DURATION, COST, AND PREREQUISITES

CPNAT Input:

Job	Time	Cost	Prerequisites
1	10		
2	1	25	1
3	2	300	1
4	1	100	2,3
5	28		
6	1	100	5,4
7	2	300	4
8	30	850	4
9	45	300	4
10	5	1200	8
11	6	400	6,7
12	6	800	11,10
13	2	100	12
14	1	100	9,6,7
15	4	300	13,14
16	1	100	13,14
17	1	50	16
18	1	100	15,16
19	1	100	18,17
20	60		

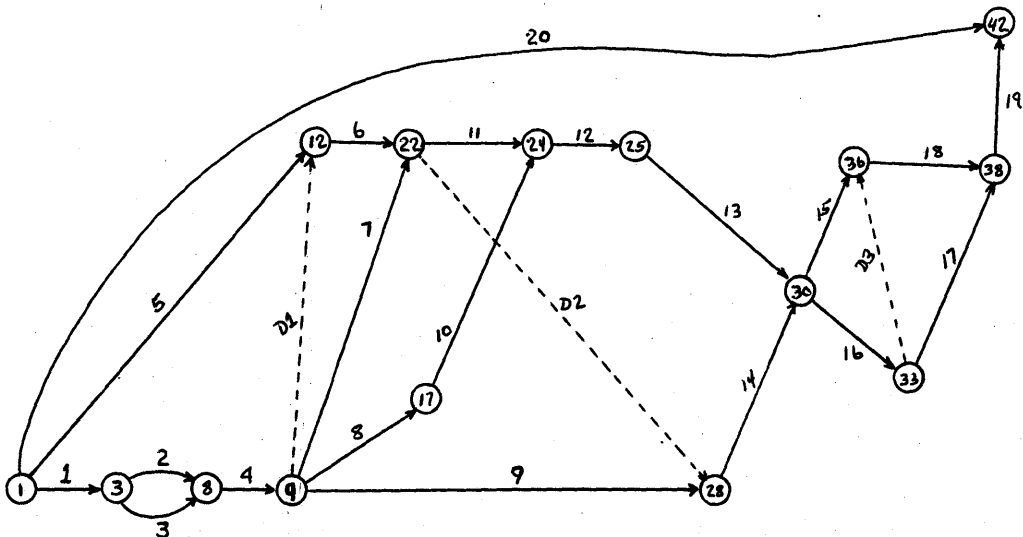
```

0010      0001
00010002500020001
00020030000030001
000100100000400020003
0028      0005
000100100000600050004
00020030000070004
00300085000080004
00450030000090004
00050120000100008
000600400001100060007
000600800001200110010
00020010000130012
0001001000014000900060007
000400300001500130014
000100100001600130014
00010005000170016
000100100001800150016
000100100001900180017
0060      0020
    
```



```

000100030010 0001
000300080001000250002
00030008000200300003
00080009000100100004
000100120028 0005
00120022000100100006
00090022000200300007
00090017003000850008
00090028004500300009
00170024000501200010
00220024000600400011
00240025000600800012
00250030000200100013
00280030000100100014
00300036000400300015
00300033000100100016
00330038000100050017
00360038000100100018
00380042000100100019
000100420060 0020
00090012 00 0000001
00220028 00 0000002
00330036 00 0000003
    
```



An Arrow Diagram from the Output of CPNAT

MISS LESS Output:

```

000100030010 0001
000300080001000250002
00030008000200300003
00080009000100100004
000100120028 0005
00120022000100100006
00090022000200300007
000900170030008500008
00090028004500300009
00170024000501200010
00220024000600400011
00240025000600800012
00250030000200100013
00280030000100100014
00300036000400300015
00300033000100100016
00330038000100050017
00360038000100100018
00380042000100100019
000100420060 0020
00090012 00 0000001
00220028 00 0000002
00330036 00 0000003
- PROJECT COST 5225

```

```

10 10 *
10 11 11 12 1 * 1
10 12 10 12 *
12 13 12 13 *
28 28 16 44 16
28 29 44 45 16
13 15 43 45 30 14
13 43 16 46 3
13 58 13 58 *
43 48 46 51 3
29 35 45 51 16 13
48 54 51 57 3
54 56 57 59 3 3
58 59 58 59 *
59 63 59 63 *
59 60 62 63 3
60 61 63 64 3 3
63 64 63 64 *
64 65 64 65 *
60 5 65 5 5
13 13 44 44 31 15
29 29 58 58 29 29
60 60 63 63 3 3
PROJECT COMPLETION 65

```

OPERATING INSTRUCTIONS

A. Console Settings

PARITY	Switch	STOP
OFLOW	Switch	STOP
I/O	Switch	STOP
All other Switches		Not Interrogated

B. Operating Procedure

Load Program Deck - Zero core. Depress RESET, place program deck in read hopper, press LOAD. To read final cards, press READER START. Computer halts when the program is loaded.

Data Pass I - Place data deck in the read hopper, press READER START and computer START. To read the final data card press READER START. The program will then assign the necessary dummies and arrow nodes. If an error message is typed out and the computer continues, the most reasonable decision has been made and work continues, but should be checked for a possible error if the action taken does not fit this job. If an error message is typed and the computer halts, a serious error exists and must be corrected before the data deck can be handled

successfully. If pass I is successfully completed, END PASS I is typed and the computer halts.

Data Pass II - Place the data deck in the read hopper and blank cards in the punch hopper. Press READER START, PUNCH START, and computer START. When this pass is completed, PROCESSING COMPLETE. is typed out and the computer halts. Pressing START then begins PASS I again for a new job.

C. Expected Stops and Action to be Taken

1. When END PASS I is typed out and the computer halts, press START to begin Pass II.
2. When "PROCESSING COMPLETE." is typed out and the computer halts, press START to begin another job and Pass I.
3. When an error message is typed out and the computer halts, press START to begin another job on Pass I. This job cannot be run until the condition in it which caused the error is corrected. See Error Messages for a list of all error messages and the action taken with each.

ERROR MESSAGES

The following is a list of each error message and the action taken.

- ERROR 1 The same job appears more than once. The job number is typed out, the nodes of the job are changed, and processing continues. The final effect is equivalent to using this card as input with a punch in card column 1.
- ERROR 2 The tables are full. The computer halts. Pressing START will cause a return to Pass I. This job must either be shortened or run on a machine with more available memory.
- ERROR 3 A job used as a prerequisite was not defined. The job number is typed out. Processing continues by ignoring this prerequisite.
- ERROR 4 A job read in to be punched in Pass II was not defined in Pass I. The job number is typed out. The job is ignored and processing continues.

## ERRORS WHILE RUNNING MISS LESS

It is possible to successfully run a job on CPNAT, and then find errors while running on MISS LESS. These are not caused by CPNAT, but rather were present in the input used for CPNAT. For instance, it is possible to assign prerequisites in such a way that a loop will be found by MISS LESS. Remember, CPNAT generates the nodes and dummies to fit the prerequisites exactly. If your prerequisites describe a loop, a loop will be generated.

A common mistake made when assigning prerequisites is to specify for a prerequisite of a job, a job which is a prerequisite of another prerequisite of that job. For instance: Suppose job 10 has job 1 as a prerequisite. Suppose job 11 has job 10 as a prerequisite. It would be wrong to call both jobs 1 and 10 prerequisites of job 11. In this case, job 10 would have the same node assigned as both its I and J. This would then be caught by MISS LESS.

The main rule to follow is name only jobs which must be performed immediately before the job in question as prerequisites.

Errors of these types are easily found by examining a listing of the output from CPNAT.

## Core Layout:

1. Console area is used for temporary storage.
2. 00401 up to 01090 is used for storage, 01090 is the beginning of the instruction area.
3. START (01690) is the first instruction to be executed by the program.
4. INIT (01742) is the first instruction of Pass I.
5. The tables are built in the area from TABLES - 3 (07891) to the top of the core.

## Likely Changes:

1. Although the program was written to take the least storage possible, the programming time available did not allow full optimization. Many routines could probably be made shorter and some of the loops could be combined. Also, it probably would be possible to build new tables over old, partially used ones if great care was taken not to destroy needed information.
2. Incorporating Strip, Fill, or Indirect Addressing would shorten the program considerably.
3. Some of the routines which remove unnecessary dummies could be removed. This would result in a program which would run in a shorter period of time, but would have a longer number of dummies

in the output.

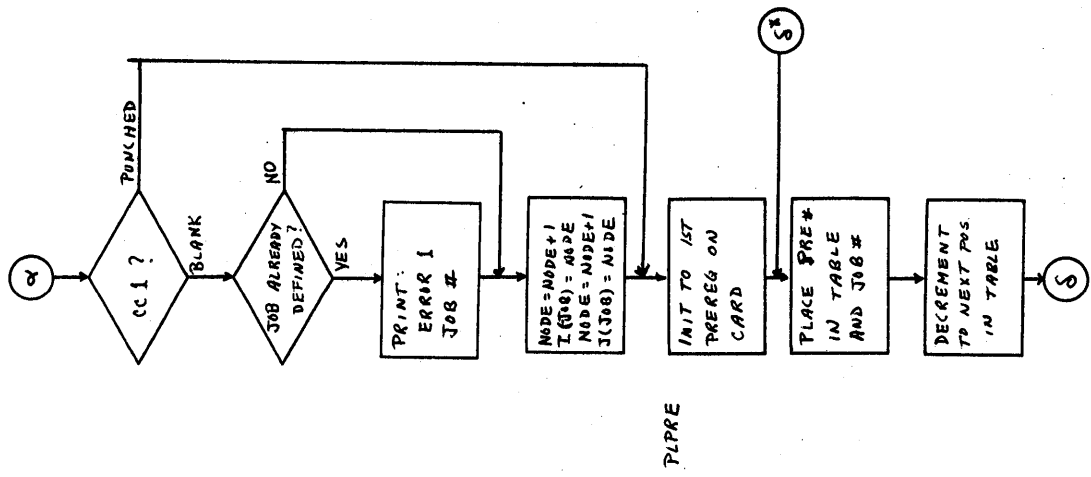
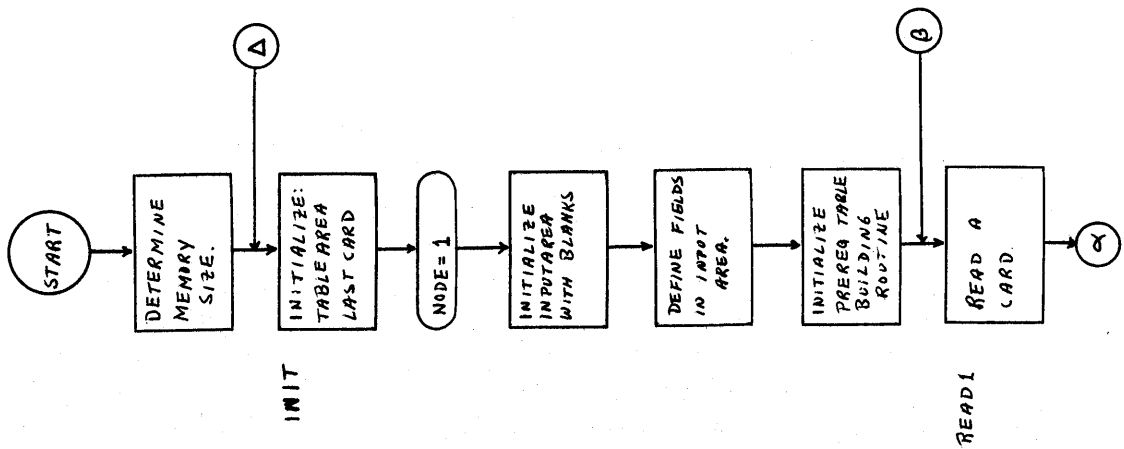
4. Any changes can best be incorporated by altering the source deck and reassembling.
5. Information and answers to specific questions can be answered either by consulting the listing and flow chart, or through correspondence with the author.

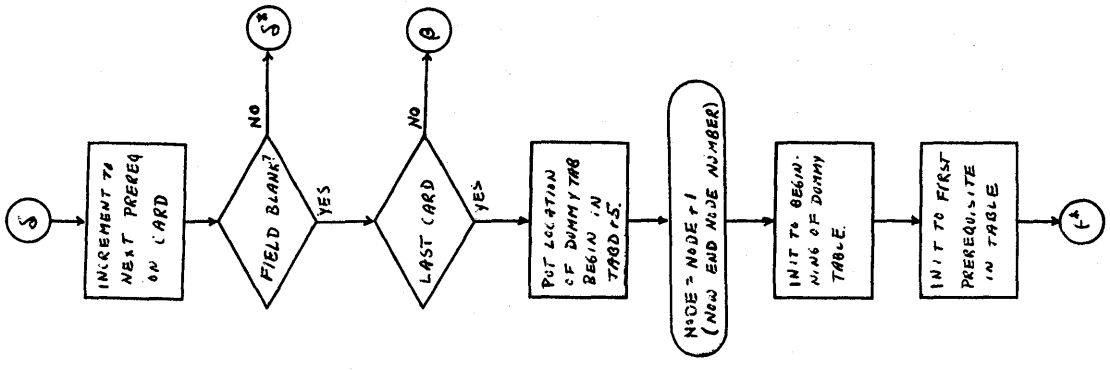
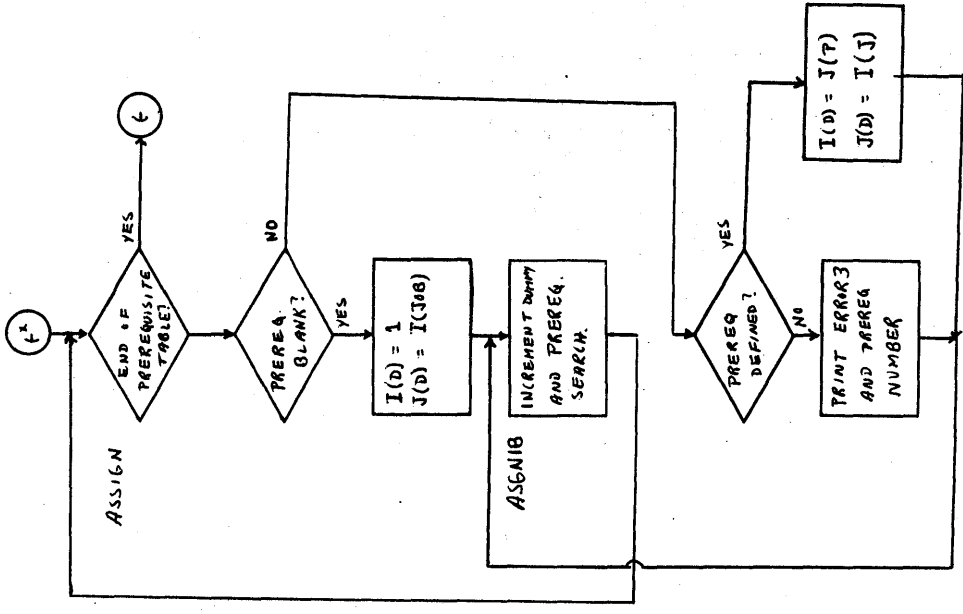
#### FLOW-CHART

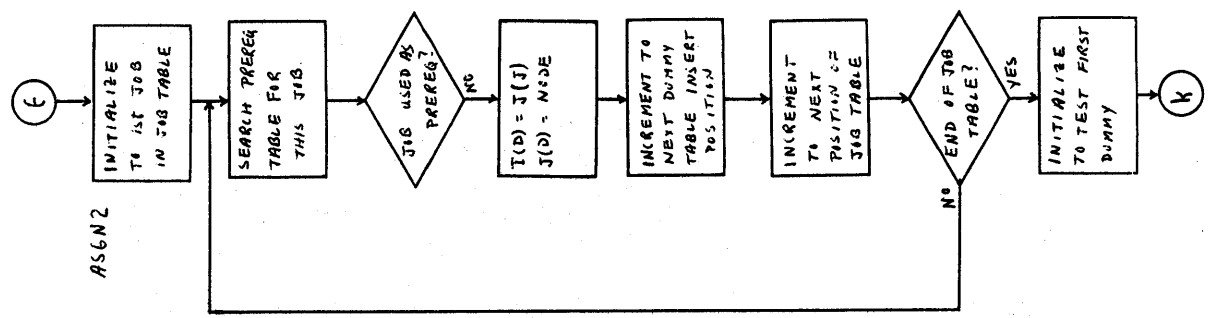
Following is the flow chart for CPNAT. Although subscripts are used quite extensively, they do not actually appear as such in the program. This could at times be confusing, but is necessary since the various tables can contain the actual information, the first address of the table containing the information, or even the address of the address containing the information. By referring to the flow-chart, the program listing, and the method of computation it should be possible to completely understand the program.

The backbone of the program is a powerful Table Lookup routine which is used for locating information, depositing information, finding ends of tables, and recognizing table overlaps.

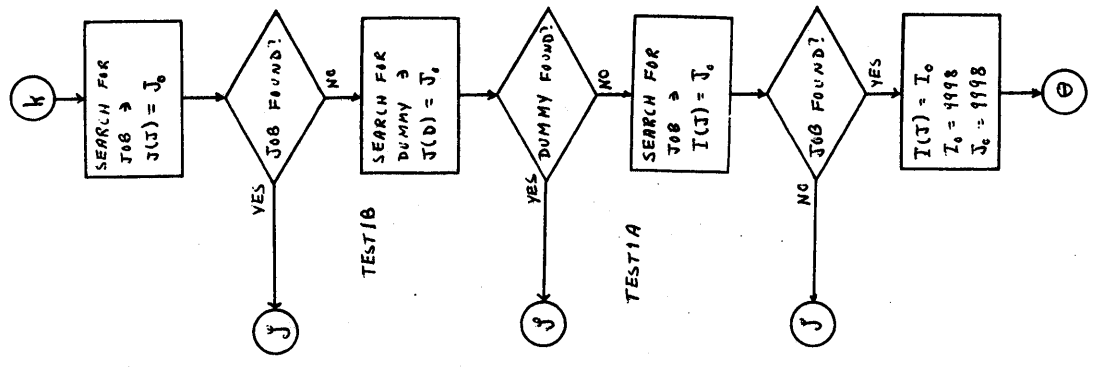
The symbols used in the flow chart pretty much follow the ones used in the program. There are discrepancies where the program has used an area at different times for different things to conserve storage. In these cases the flow chart used meaningful names to make itself clearer.



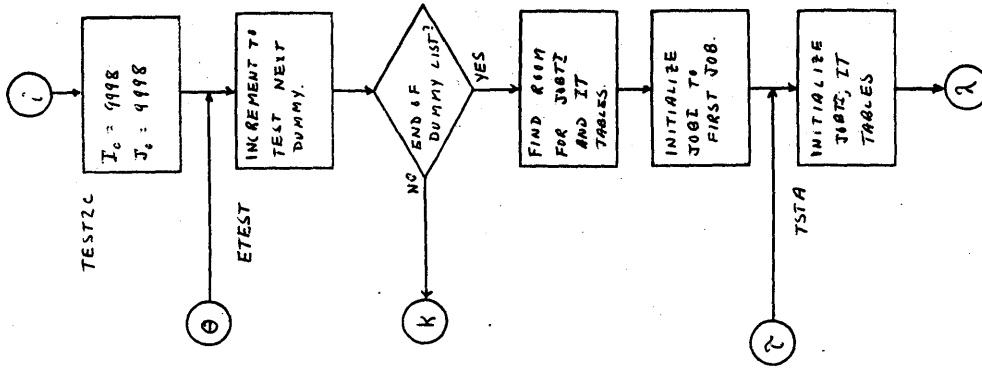
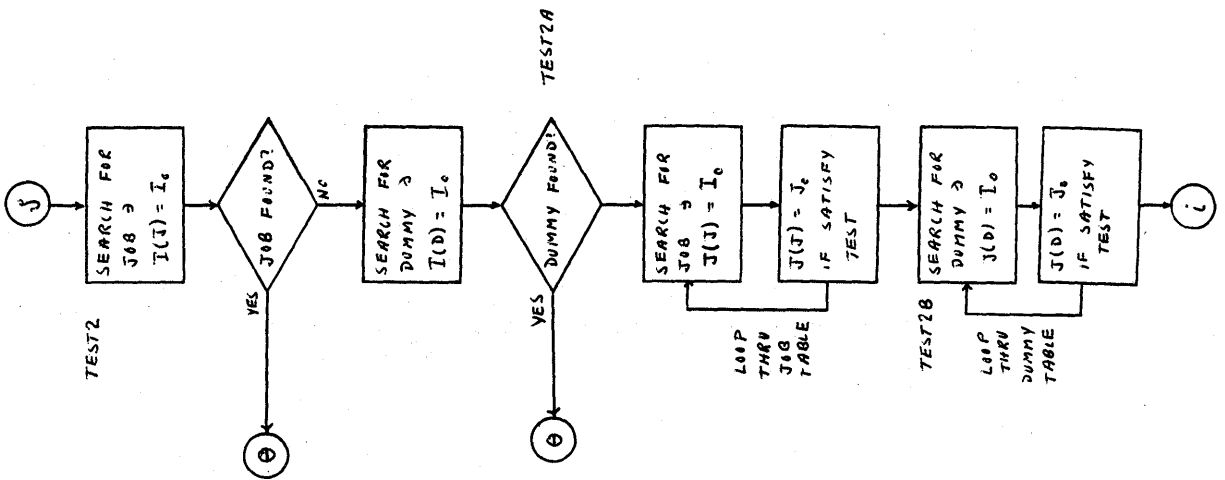


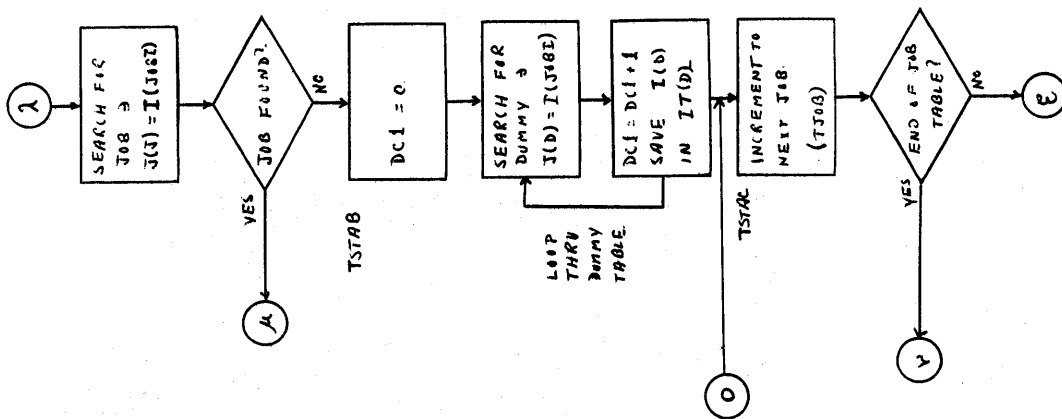
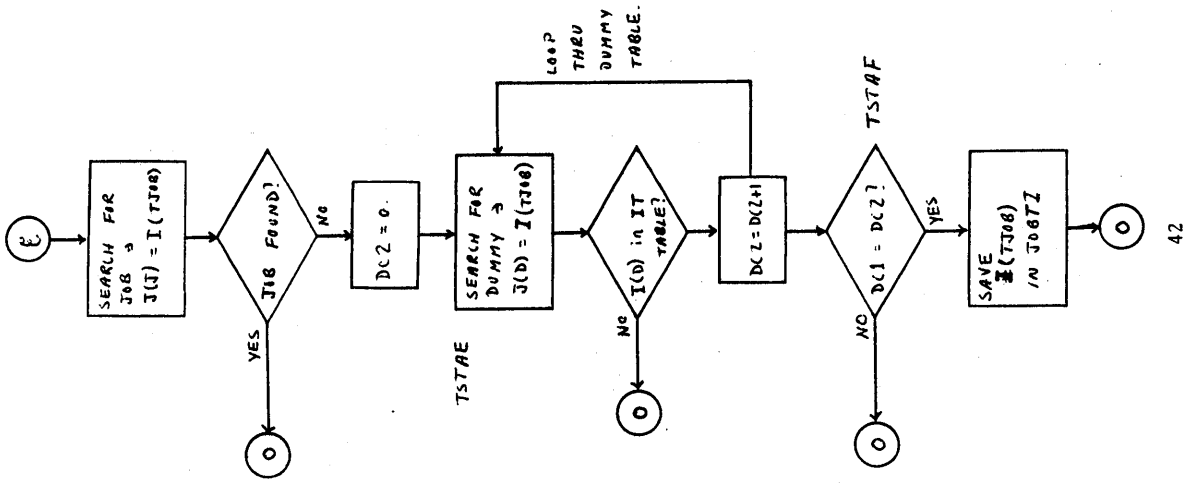


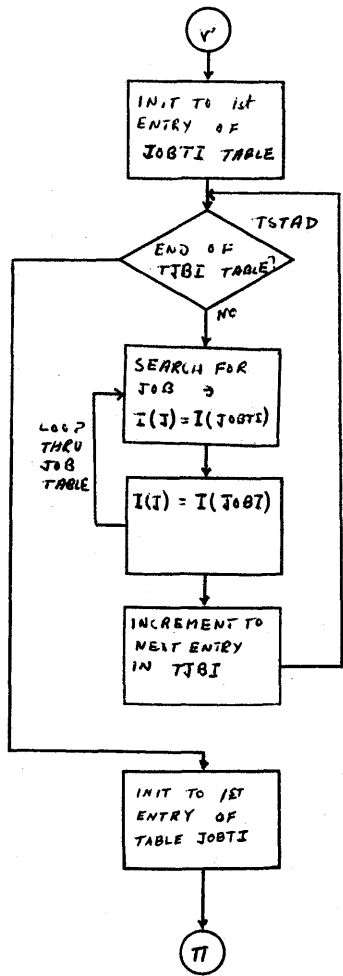
$I_0 = I$  of test dummy  
 $J_0 = J$  of test dummy  
 $J(I) = J$  of a job.  
 $J(D) = J$  if a dummy.  
 SEARCH (TLU) routine actually loops through entire table.





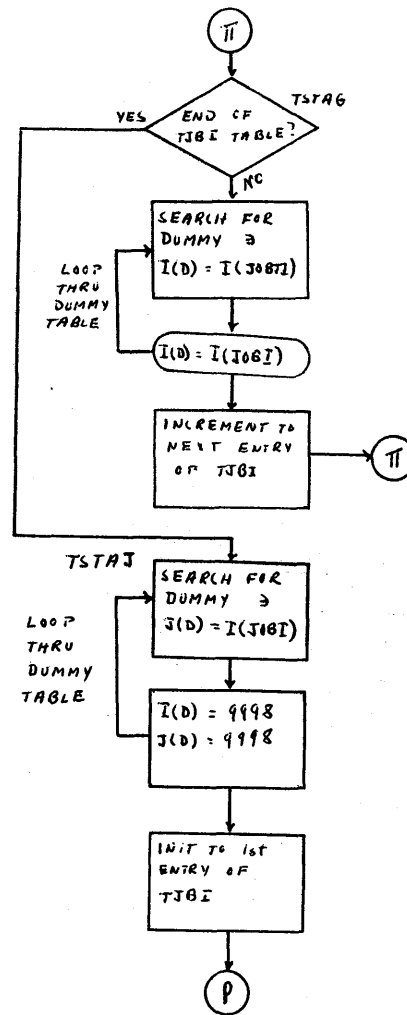


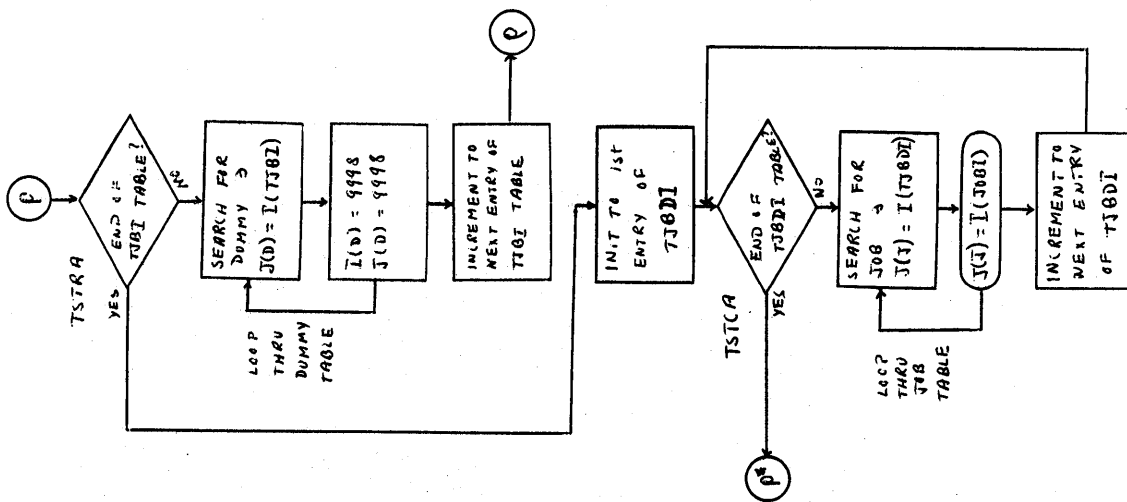




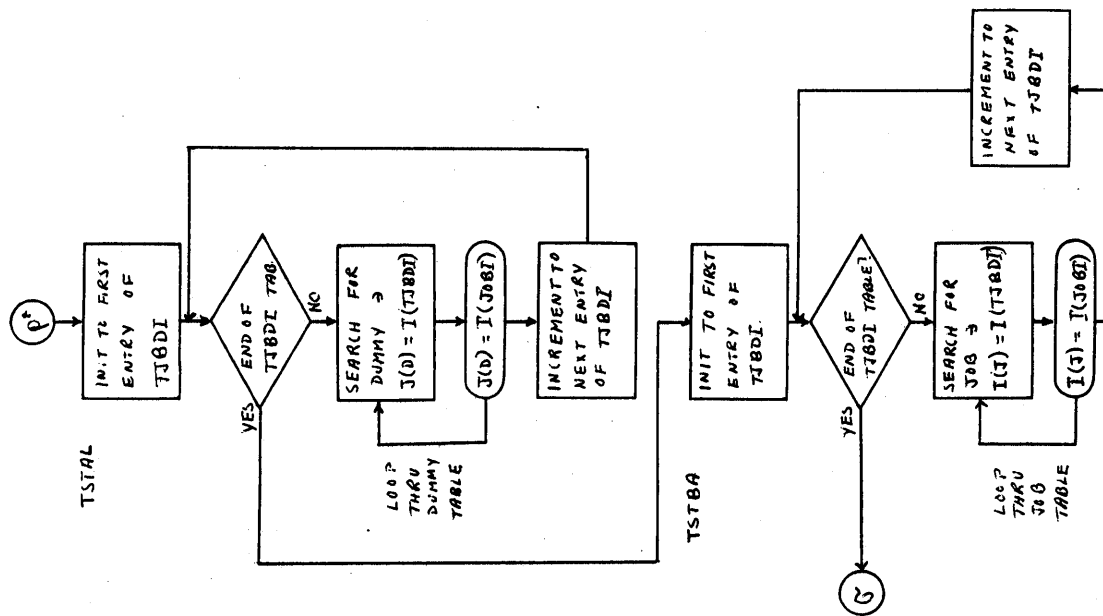
TJBI is actual name of JOBTI table. It contains the I's of the saved jobs.

TJBDI is actual name of IT table. It contains the I's of the dummies going into these jobs.

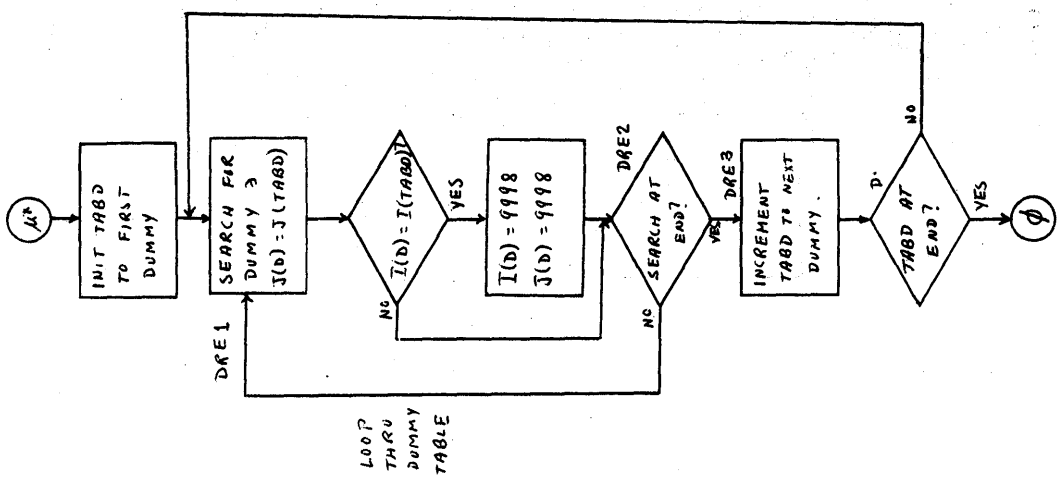
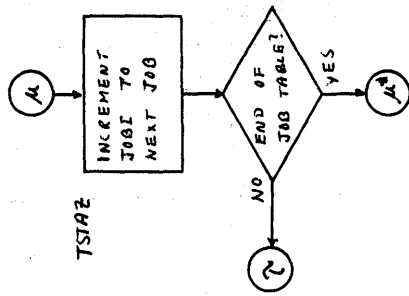
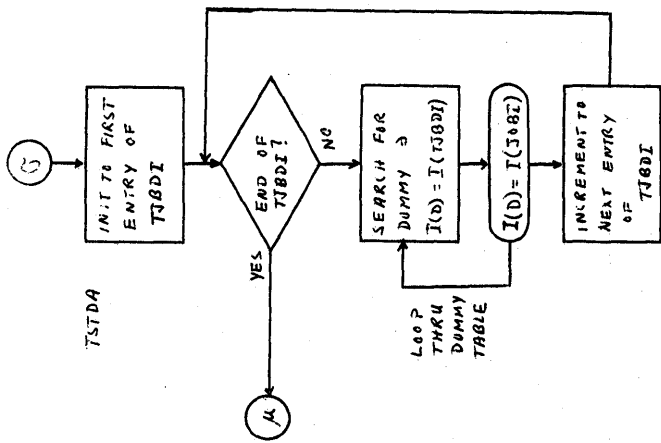


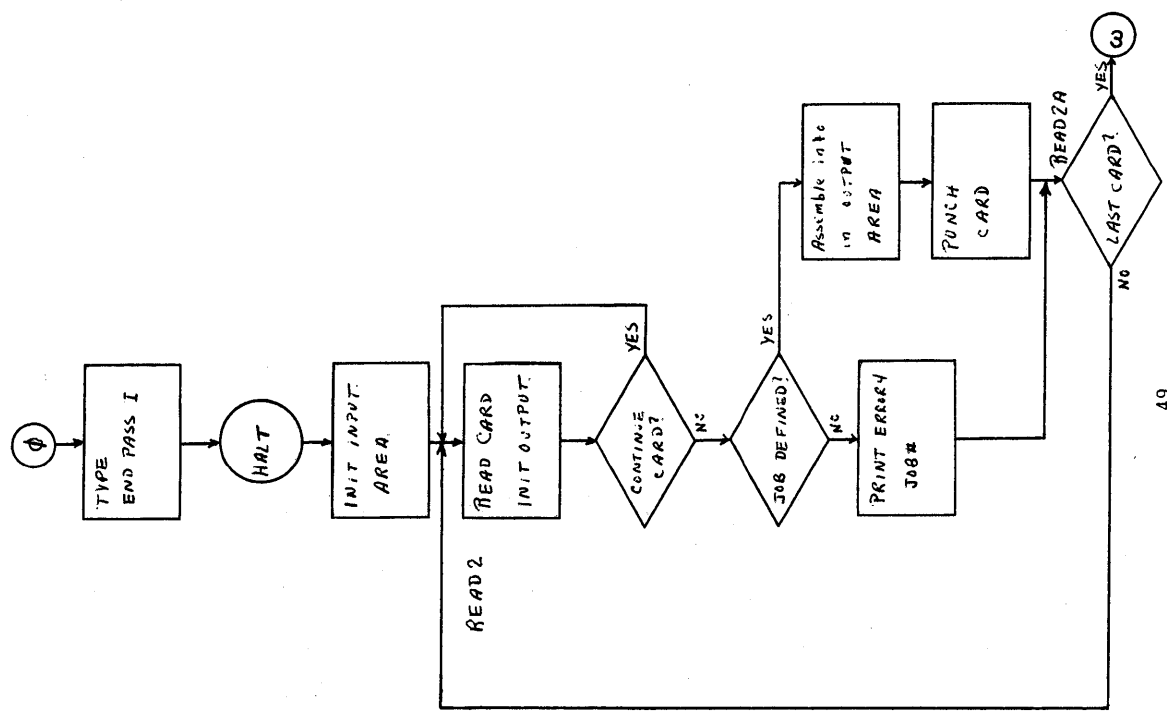
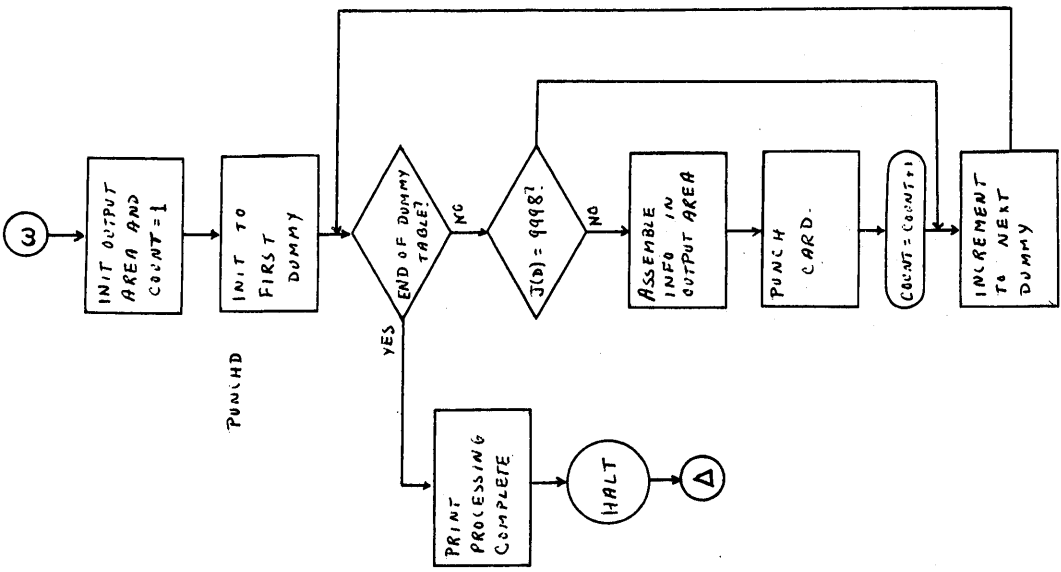


45



46





\* C R I T I C A L P A T H N O D E A S S I G N M E N T T E C H N I Q U E . Z

00405	00004	NODE	DS	4Z	
00407	00080	INPUT	DAS	80Z	
00566	00001		DC	1,@Z	
00567	00160	ZERO	DSS	160Z	
00727	00001		DC	1,@Z	
00729	00080	OUTPUT	DAS	80Z	
00888	00001		DC	1,@Z	
00891	00008	ERROR	DAC	8,ERROR 0@Z	
00910	00005 01301	TABJ	DSA	SUB1-1, TABLES+8,12Z	
00915	00005 07902				
00920	00005 00012				
00925	00005 R9999	TABD	DSA	99999,99999,8Z	
00930	00005 R9999				
00935	00005 00008				
00940	00005	DUMMY	DS	5Z	
00945	00005 07902	TABP	DSA	TABLES+8,99999,-8Z	
00950	00005 R9999				
00955	00005 0000Q				
00960	00005 R9999	TABJJ	DSA	99999, TABLES+4,12Z	
00965	00005 07898				
00970	00005 00012				
00975	00005	DUMAD	DS	5Z	
00980	00005 R9999	TABJI	DSA	99999, TABLES,12Z	
00985	00005 07894				
00990	00005 00012				
00995	00005 R9999	TABJT	DSA	99999, TABLES+4,12Z	
01000	00005 07898				
01005	00005 00012				
01007	00011	EPI	DAC	11,END PASS I@Z	
01029	00021	PC	DAC	21,PROCESSING COMPLETE.@Z	
01089	00020		DS	20Z	
01090	16 01132 01089	SUB2	TFM	**+42,SUB2-1Z	
01102	16 01137 01080		TFM	**+35,SUB2-10Z	
01114	26 01080 01089		TF	SUB2-10,SUB2-1Z	
01126	25 00000 00000		TD	Z	
01138	26 01168 01132		TF	**+30,*-6Z	
01150	12 01168 00001		SM	**+18,1,10Z	
01162	15 00000 00007		TDM	,7Z	
01174	26 01197 01132		TF	**+23,*-42Z	
01186	44 01260 00000		BNF	**+74Z	
01198	26 01216 01197		TF	**+18,*-1Z	
01210	33 00000 00000		CF	Z	
01222	26 01252 01216		TF	**+30,*-6Z	
01234	12 01252 00001		SM	**+18,1,10Z	
01246	32 00000 00000		SF	Z	
01258	42 00000 00000		BB	Z	
01260		DORG	**+9Z		
01260	12 01132 00002		SM	SUB2+42,2,10Z	
01272	12 01137 00001		SM	SUB2+47,1,10Z	
01284	49 01126 00000		B	SUB2+36Z	
01291		DORG	**+4Z		
01300	00010		DS	10Z	
01302	16 01337 01299	SUB1	TFM	**+35,SUB1-3Z	
01314	16 01332 01300		TFM	**+18,SUB1-2Z	
01326	25 00000 00000		TD	Z	

01338	26	01361	01332	
01350	44	01364	00000	
01362	42	00000	00000	
01364				
01364	12	01332	00001	
01376	12	01337	00002	
01388	49	01326	00000	
01395				
01397		00003		
01398	34	00000	00102	
01410	25	00903	01397	
01422	39	00891	00100	
01434	44	01466	01396	
01446	48	00000	00000	
01458	49	01742	00000	
01466				
01466	42	00000	00000	
01468				
01472		00005		
01474	26	01497	01473	
01486	26	01569	00000	
01498	11	01497	00005	
01510	26	01533	01497	
01522	26	01564	00000	
01534	11	01497	00005	
01546	26	01629	01497	
01558	24	00000	00000	
01570	46	01638	01200	
01582	26	01600	01564	
01594	14	00000	0R999	
01606	46	01638	01200	
01618	21	01564	00000	
01630	49	01558	00000	
01638				
01638	26	00099	01564	
01650	24	00099	02252	
01662	47	01686	01300	
01674	17	01398	00002	
01686	42	00000	00000	
01688				
01689		00002		
01690	31	J9999	01688	
01702	45	01722	00000	
01714	49	01742	00000	
01722				
01722	11	01693	000K0	
01734	49	01690	00000	
01742				
01742	16	01760	07894	
01754	16	00000	0R999	
01766	11	01760	00004	
01778	24	01760	01696	
01790	47	01754	01100	
01802	12	01760	00004	
01814	46	01826	00900	

TF	**+23,*-6Z
BNF	**+14Z
BB	Z
DORG	**+9Z
SM	SUB1+30,1,10Z
SM	SUB1+35,2,10Z
B	SUB1+24Z
DORG	**+4Z
DS	3Z
RCTY	Z
TD	ERROR+2*6,ERMES-1Z
WATY	ERRORZ
BNF	**+32,ERMES-2Z
H	Z
B	INITZ
DORG	**+3Z
BB	Z
DORG	**+9Z
DS	5Z
TF	**+23,*-1Z
TF	**+83Z
AM	**+1,5,10Z
TF	**+23,*-13Z
TF	**+42Z
AM	**+37,5,10Z
TF	**+83,*-49Z
C	Z
BZ	**+68Z
TF	**+18,*-18Z
CM	,9999,8Z
BZ	**+32Z
A	**+54Z
B	**+72Z
DORG	**+3Z
TF	99,TLU+90Z
C	99,PLPRE+30Z
BN	**+24Z
BTM	ERMES,2,10Z
BB	Z
DORG	**+9Z
* F I N D T H E S I Z E O F T H E M E M O R Y . Z	
RECMK	DC 2,@Z
START	TR 19999,RECMK-1,2Z
BNR	**+20Z
B	INITZ
DORG	**+3Z
AM	START+3,20,10Z
B	STARTZ
DORG	**+3Z
* I N I T I A L I Z E T H E T A B L E A R E A A N D L C I A N D A L L C O U N T E R S	
INIT	TFM **+18, TABLESZ
TFM	,9999,8Z
AM	**+6,4,10Z
C	INIT+18,START+6Z
BNP	INIT+12Z
SM	INIT+18,4,10Z
BLC	**+12Z

01826	15	01663	00007	TDM	TLU+189,7Z
01838	16	00405	00001	TFM	NODE,1,8Z
01850	31	00406	00567	TR	INPUT-1,ZEROZ
01862	16	01880	00427	TFM	**18,INPUT+2*11-2Z
01874	32	00000	00000	SF	Z
01886	11	01880	00008	AM	*-6,8,10Z
01898	14	01880	00547	CM	*-18,INPUT+2*71-2Z
01910	47	01874	01100	BNP	*-36Z
01922	26	02252	01760	TF	PLPRE+30,INIT+18Z
01934	32	00406	00000	SF	INPUT-1Z
01946	37	00407	00500	READ1 RACD	INPUTZ
01958	27	01302	00433	BT	SUB1,INPUT+2*14-2Z
01970	26	00089	01301	TF	89,SUB1-1Z
01982	14	00407	00000	CM	INPUT,,10Z
01994	47	02222	01200	BNZ	PLPREZ
02006	17	01474	00910	BTM	TLU,TABJZ
02018	26	02036	00099	TF	**18,99Z
02030	14	00000	0R999	CM	,9999,8Z
02042	46	02102	01200	BZ	**60Z
02054	17	01398	00001	BTM	ERMES,1,9Z
02066	34	00000	00102	RCTY	Z
02073		00001		DC	1,@,*-4Z
02078	26	02072	01301	TF	*-6,SUB1-1Z
02090	38	02069	00100	WNTY	*-21Z
02102	26	02120	00099	TF	**18,99Z
02114	26	00000	01301	TF	,SUB1-1Z
02126	11	00405	00001	AM	NODE,1,10Z
02138	12	02120	00008	SM	*-18,8,10Z
02150	26	02168	02120	TF	**18,*-30Z
02162	26	00000	00405	TF	,NODEZ
02174	11	02168	00004	AM	*-6,4,10Z
02186	11	00405	00001	AM	NODE,1,10Z
02198	26	02216	02168	TF	**18,*-30Z
02210	26	00000	00405	TF	,NODEZ
02222	16	02245	00441	PLPRE TFM	**23,INPUT+2*18-2Z
02234	27	01302	00000	BT	SUB1Z
02246	26	00000	00089	TF	,89Z
02258	12	02252	00004	SM	*-6,4,10Z
02270	26	02288	02252	TF	**18,*-18Z
02282	26	00000	01301	TF	,SUB1-1Z
02294	12	02252	00004	SM	*-42,4,10Z
02306	11	02245	00008	AM	PLPRE+23,8,10Z
02318	26	02336	02245	TF	**18,PLPRE+23Z
02330	14	00000	00000	CM	,,10Z
02342	46	02234	01100	BP	PLPRE+12Z
02354	47	01946	00900	BNLC	READ1Z
				* FIN D FI	RST AVAILABLE ADDRESS FOR DUMMY VARIABLE TA
02366	16	01301	00000	TFM	SUB1-1,,8Z
02378	17	01474	00910	BTM	TLU,TABJZ
02390	11	00099	00008	AM	99,8,10Z
02402	26	00930	00099	TF	TABD+5,99Z
02414	11	00405	00001	AM	NODE,1,10Z
				* ASS IGN	DUMMIES TO SATISFY ALL PREREQUISITES.Z
02426	26	00940	00930	TF	DUMMY,TABD+5Z
02438	26	02456	01760	TF	ASSIGN+6,INIT+18Z
02450	14	00000	0R999	ASSIGN CM	,9999,8Z
02462	46	02954	01200	BZ	ASGN2Z

02474	26	02504	02456	TF	**30,ASSIGN+6Z
02486	12	02504	00004	SM	**18,4,10Z
02498	14	00000	00000	CM	,,8Z
02510	47	02674	01200	BNZ	ASGN1AZ
02522	26	02552	00940	TF	**30,DUMMYZ
02534	12	02552	00004	SM	**18,4,10Z
02546	16	00000	00001	TFM	,1,8Z
02558	26	02581	02456	TF	**23,ASSIGN+6Z
02570	26	01301	00000	TF	SUB1-1Z
02582	17	01474	00910	BTM	TLU,TABJZ
02594	12	00099	00008	SM	99,8,10Z
02606	26	02641	00099	TF	**35,99Z
02618	26	02636	00940	TF	**18,DUMMYZ
02630	26	00000	00000	TF	Z
02642	11	00940	00008	ASGN1b AM	DUMMY,8,10Z
02654	12	02456	00008	SM	ASSIGN+6,8,10Z
02666	49	02450	00000	B	ASSIGNZ
02674				DORG	*-3Z
02674	26	02697	02504	ASGN1A TF	**23,ASSIGN+54Z
02686	26	01301	00000	TF	SUB1-1Z
02698	17	01474	00910	BTM	TLU,TABJZ
02710	26	02728	00099	TF	**18,99Z
02722	14	00000	0R999	CM	,9999,8Z
02734	47	02802	01200	BNZ	**68Z
02746	17	01398	00003	BTM	ERMES,3,9Z
02758	26	02776	01301	TF	**18,SUB1-1Z
02770	34	00000	00102	RCTY	Z
02777		00001		DC	1,@,*-4Z
02782	38	02773	00100	WNTY	*-9Z
02794	49	02654	00000	B	ASGN1B+12Z
02802				DORG	*-3Z
02802	12	00099	00004	SM	99,4,10Z
02814	26	02861	00099	TF	**47,99Z
02826	26	02856	00940	TF	**30,DUMMYZ
02838	12	02856	00004	SM	**18,4,10Z
02850	26	00000	00000	TF	Z
02862	26	02885	02456	TF	**23,ASSIGN+6Z
02874	26	01301	00000	TF	SUB1-1Z
02886	17	01474	00910	BTM	TLU,TABJZ
02898	12	00099	00008	SM	99,8,10Z
02910	26	02945	00099	TF	**35,99Z
02922	26	02940	00940	TF	**18,DUMMYZ
02934	26	00000	00000	TF	Z
02946	49	02642	00000	B	ASGN1BZ
02954				DORG	*-3Z
				* ASS IGN	DUMMIES FOR JOBS NOT USED AS A PREREQUISITE
02954	16	00945	07902	ASGN2 TFM	TABP, TABLES+8Z
02966	15	01663	00009	TDM	TLU+189,7Z
02978	26	00950	01760	TF	TABP+5,INIT+18Z
02990	12	00950	00004	SM	TABP+5,4,10Z
03002	17	01474	00945	BTM	TLU,TABPZ
03014	26	03032	00099	TF	**18,99Z
03026	14	00000	0R999	CM	,9999,8Z
03038	47	03146	01200	BNZ	**108Z
03050	26	03068	00940	TF	**18,DUMMYZ
03062	26	00000	00405	TF	,NODEZ
03074	26	03133	00945	TF	**59,TABPZ



03086	12	03133	00004	SM	**47,4,10Z
03098	26	03128	00940	TF	**30,DUMMYZ
03110	12	03128	00004	SM	**18,4,10Z
03122	26	00000	00000	TF	Z
03134	11	00940	00008	AM	DUMMY,8,10Z
03146	11	00945	000J2	AM	TABP,12,10Z
03158	26	03176	00945	TF	**18,TABPZ
03170	14	00000	CR999	CM	,9999,8Z
03182	47	03002	01200	BNZ	ASGN2+48Z
* ROU TINE FOR DISCARDING THE UNNECESSARY DUMMIES.Z					
03194	26	00975	00930	TF	DUMAD,TABD+5Z
03206	26	00940	00930	TF	DUMMY,TABD+5Z
03218	26	00960	00940	TEST	TF TABJJ,DUMMYZ
03230	17	01474	00960	BTM	TLU,TABJJZ
03242	26	03260	00099	TF	**18,99Z
03254	14	00000	OR999	CM	,9999,8Z
03266	47	03570	01200	BNZ	TEST2Z
03278	26	00925	00940	TF	TABD,DUMMYZ
03290	26	00930	00975	TF	TABD+5,DUMADZ
03302	17	01474	00925	TEST1B	BTM TLU,TABDZ
03314	26	03332	00099	TF	**18,99Z
03326	14	00000	OR999	CM	,9999,8Z
03338	46	03406	01200	BZ	TEST1AZ
03350	24	00099	00925	C	99,TABDZ
03362	47	03570	01200	BNZ	TEST2Z
03374	11	00099	00008	AM	99,8,10Z
03386	26	00930	00099	TF	TABD+5,99Z
03398	49	03302	00000	B	TEST1BZ
03406				DORG	*-3Z
03406	26	00980	00960	TEST1A	TF TABJI,TABJJZ
03418	17	01474	00980	BTM	TLU,TABJIZ
03430	26	03448	00099	TF	**18,99Z
03442	14	00000	OR999	CM	,9999,8Z
03454	46	03570	01200	BZ	TEST2Z
03466	26	03513	00980	TF	**47,TABJIZ
03478	12	03513	00004	SM	**35,4,10Z
03490	26	03508	00099	TF	**18,99Z
03502	26	00000	00000	TF	Z
03514	26	03532	03513	TF	**18,*-1Z
03526	16	00000	OR998	TFM	,9998,8Z
03538	26	03556	00940	TF	**18,DUMMYZ
03550	16	00000	OR998	TFM	,9998,8Z
03562	49	04122	00000	B	ETESTZ
03570				DORG	*-3Z
03570	26	00980	00940	TEST2	TF TABJI,DUMMYZ
03582	12	00980	00004	SM	TABJI,4,10Z
03594	17	01474	00980	BTM	TLU,TABJIZ
03606	26	03624	00099	TF	**18,99Z
03618	14	00000	OR999	CM	,9999,8Z
03630	47	04122	01200	BNZ	ETESTZ
03642	26	00930	00975	TF	TABD+5,DUMADZ
03654	12	00930	00004	SM	TABD+5,4,10Z
03666	26	00925	00980	TF	TABD,TABJIZ
03678	17	01474	00925	BTM	TLU,TABDZ
03690	24	00099	00925	C	99,TABDZ
03702	47	03734	01200	BNZ	TEST2AZ
03714	11	00930	00008	AM	TABD+5,8,10Z

03726	49	03678	00000	B	*-48Z
03734				DORG	*-3Z
03734	26	03752	00099	TEST2A	TF **18,99Z
03746	14	00000	OR999	CM	,9999,8Z
03758	47	04122	01200	BNZ	ETESTZ
03770	26	00995	00940	TF	TABJT,DUMMYZ
03782	12	00995	00004	SM	TABJT,4,10Z
03794	16	01000	07898	TFM	TABJT+5, TABLES+4Z
03806	17	01474	00995	BTM	TLU,TABJTZ
03818	26	03836	00099	TF	**18,99Z
03830	14	00000	OR999	CM	,9999,8Z
03842	46	03922	01200	BZ	TEST2BZ
03854	26	03884	00099	TF	**30,99Z
03866	26	03889	00940	TF	**23,DUMMYZ
03878	26	00000	00000	TF	Z
03890	11	00099	000J2	AM	99,12,10Z
03902	26	01000	00099	TF	TABJT+5,99Z
03914	49	03806	00000	B	*-108Z
03922				DORG	*-3Z
03922	26	00930	00975	TEST2B	TF TABD+5,DUMADZ
03934	26	00925	00995	TF	TABD,TABJTZ
03946	17	01474	00925	BTM	TLU,TABDZ
03958	26	03976	00099	TF	**18,99Z
03970	14	00000	OR999	CM	,9999,8Z
03982	46	04062	01200	BZ	TEST2CZ
03994	26	04024	00099	TF	**30,99Z
04006	26	04029	00940	TF	**23,DUMMYZ
04018	26	00000	00000	TF	Z
04030	11	00099	00008	AM	99,8,10Z
04042	26	00930	00099	TF	TABD+5,99Z
04054	49	03946	00000	B	*-108Z
04062				DORG	*-3Z
04062	26	04080	00940	TEST2C	TF **18,DUMMYZ
04074	16	00000	OR998	TFM	,9998,8Z
04086	26	04116	04080	TF	**30,*-6Z
04098	12	04116	00004	SM	**18,4,10Z
04110	16	00000	OR998	TFM	,9998,8Z
04122	11	00940	00008	ETEST	AM DUMMY,8,10Z
04134	26	04152	00940	TF	**18,DUMMYZ
04146	14	00000	OR999	CM	,9999,8Z
04158	47	03218	01200	BNZ	TESTZ
* ROU TINE TO ELIMINATE DUMMIES FROM JOBS WITH IDENTI					
04170	15	01663	00007	TFM	TLU+189,7Z
04182	26	04703	02252	TF	TJBI+5,PLPRE+30Z
04194	11	02252	00004	AM	PLPRE+30,4,10Z
04206	26	00930	00975	TF	TABD+5,DUMADZ
04218	16	00925	01765	TFM	TABD,INIT+23Z
04230	17	01474	00925	BTM	TLU,TABDZ
04242	11	00099	00008	AM	99,8,10Z
04254	26	04718	00099	TF	TJBDI+5,99Z
04266	16	04728	07894	TFM	JOB1, TABLESZ
04278	26	00960	04728	TSTA	TF TABJJ,JOBIZ
04290	26	04308	04718	TF	**18,TJBDI+5Z
04302	16	00000	OR999	TFM	,9999,8Z
04314	11	04308	00004	AM	*-6,4,10Z
04326	26	04344	04308	TF	**18,*-18Z
04338	14	00000	OR999	CM	,9999,8Z

04350	47	04302	01200	BNZ	*-48Z
04362	26	04380	04703	TF	**+18,TJBI+5Z
04374	16	00000	0R999	TFM	,9999,8Z
04386	12	04380	00004	SM	*-6,4,10Z
04398	26	04416	04380	TF	**+18,*-18Z
04410	14	00000	0R999	CM	,9999,8Z
04422	47	04374	01200	BNZ	*-48Z
04434	17	01474	00960	BTM	TLU,TABJJZ
04446	26	04464	00099	TF	**+18,99Z
04458	14	00000	0R999	CM	,9999,8Z
04470	47	06780	01200	BNZ	TSTAZZ
04482	16	04731	00000	TFM	DC1,,9Z
04494	26	00925	00960	TF	TABD,TABJJZ
04506	26	00930	00975	TF	TABD+5,DUMADZ
04518	17	01474	00925	TSTAB	BTM
04530	26	04548	00099	TF	TLU,TABDZ
04542	14	00000	0R999	TF	**+18,99Z
04554	46	04736	01200	CM	,9999,8Z
04566	26	00089	00099	BZ	TSTACZ
04578	26	04649	00099	TF	89,99Z
04590	12	04649	00004	TF	**+71,99Z
04602	16	04713	01765	SM	**+59,4,10Z
04614	17	01474	04713	TFM	TJBDI,INIT+23Z
04626	26	04644	00099	BTM	TLU,TJBDIZ
04638	26	00000	00000	TF	**+18,99Z
04650	11	00089	00008	TF	Z
04662	26	00930	00089	AM	89,8,10Z
04674	11	04731	00001	TF	TABD+5,89Z
04686	49	04518	00000	AM	DC1,1,10Z
04694				B	TSTABZ
04698	00005	R9999		DORG	*-3Z
04703	00005	R9999		TJBI	DSA
04708	00005	0000M			99999,99999,-4Z
04713	00005	R9999		TJBDI	DSA
04718	00005	R9999			99999,99999,4Z
04723	00005	00004			
04728	00005			JOB	DS
04731	00003			DC1	DS
04734	00003			DC2	DS
04736	11	00960	000J2	TSTAC	AM
04748	26	04766	00960	TF	TABJJ,12,10Z
04760	14	00000	0R999	TF	**+18,TABJJZ
04772	46	05196	01200	CM	,9999,8Z
04784	17	01474	00960	BZ	TSTADZ
04796	26	04814	00099	BTM	TLU,TABJJZ
04808	14	00000	0R999	TF	**+18,99Z
04820	47	04736	01200	CM	,9999,8Z
04832	26	00925	00960	BNZ	TSTACZ
04844	26	00930	00975	TF	TABD,TABJJZ
04856	16	04734	00000	TF	TABD+5,DUMADZ
04868	17	01474	00925	TFM	DC2,,9Z
04880	26	04898	00099	TSTAE	BTM
04892	14	00000	0R999	TF	TLU,TABDZ
04904	46	05044	01200	TF	**+18,99Z
04916	26	04713	00099	CM	,9999,8Z
04928	26	00089	00099	BZ	TSTAFZ
04940	12	04713	00004	TF	TJBDI,99Z
				TF	89,99Z
				SM	TJBDI,4,10Z

57

04952	17	01474	04713	BTM	TLU,TJBDIZ
04964	26	04982	00099	TF	**+18,99Z
04976	14	00000	0R999	CM	,9999,8Z
04988	46	04736	01200	BZ	TSTACZ
05000	11	04734	00001	AM	DC2,1,10Z
05012	11	00089	00008	AM	89,8,10Z
05024	26	00930	00089	TF	TABD+5,89Z
05036	49	04868	00000	B	TSTAEZ
05044				DORG	*-3Z
05044	24	04731	04734	TSTAF	C
05056	47	04736	01200		DC1,DC2Z
05068	16	04698	01765	BNZ	TSTACZ
05080	17	01474	04698	TFM	TJBI,INIT+23Z
05092	26	05122	00099	BTM	TLU,TJBIZ
05104	12	05122	00004	TF	**+30,99Z
05116	14	00000	0R999	SM	**+18,4,10Z
05128	46	05152	01200	CM	,9999,8Z
05140	17	01398	00002	BZ	**+24Z
05152	26	05187	00925	BTM	ERMES,2,10Z
05164	26	05182	00099	TF	**+35,TABDZ
05176	26	00000	00000	TF	**+18,99Z
05188	49	04736	00000	TF	Z
05196				B	TSTACZ
05196	26	05214	04703	DORG	*-3Z
05208	14	00000	0R999	TSTAD	TF
05220	46	05392	01200		**+18,TJBI+5Z
05232	16	01000	07894	CM	,9999,8Z
05244	26	00995	05214	BZ	TSTAGZ
05256	17	01474	00995	TFM	TABJT+5,TABLESZ
05268	26	05286	00099	TF	TABJT,TSTAD+18Z
05280	14	00000	0R999	BTM	TLU,TABJTZ
05292	46	05372	01200	TF	**+18,99Z
05304	26	05339	04728	CM	,9999,8Z
05316	26	05334	00099	BZ	TSTAHZ
05328	26	00000	00000	TF	**+35,JOBIZ
05340	11	00099	000J2	TF	**+18,99Z
05352	26	01000	00099	AM	99,12,10Z
05364	49	05256	00000	TF	TABJT+5,99Z
05372				B	TSTAD+60Z
05372	12	05214	00004	DORG	*-3Z
05384	49	05208	00000	TSTAH	SM
05392					TSTAD+18,4,10Z
05392	26	05410	04703	B	TSTAD+12Z
05404	14	00000	0R999	DORG	*-3Z
05416	46	05600	01200	TSTAG	TF
05428	26	00930	00975		**+18,TJBI+5Z
05440	12	00930	00004	CM	,9999,8Z
05452	26	00925	05410	BZ	TSTAJZ
05464	17	01474	00925	TF	TABD+5,DUMADZ
05476	26	05494	00099	SM	TABD+5,4,10Z
05488	14	00000	0R999	TF	TABD,TSTAG+18Z
05500	46	05580	01200	BTM	TLU,TABDZ
05512	26	05547	04728	TF	**+18,99Z
05524	26	05542	00099	CM	,9999,8Z
05536	26	00000	00000	BZ	TSTAKZ
05548	11	00099	00008	TF	**+35,JOBIZ
05560	26	00930	00099	TF	**+18,99Z
				TF	Z
				AM	99,8,10Z
				TF	TABD+5,99Z

58

05572	49	05464	00000	B	TSTAG+72Z
05580				DORG	*-3Z
05580	12	05410	00004	TSTAK	SM TSTAG+18,4,10Z
05592	49	05404	00000	B	TSTAG+12Z
05600				DORG	*-3Z
05600	26	00925	04728	TSTAJ	TF TABD,JOBIZ
05612	26	00930	00975	TF	TABD+5,DUMADZ
05624	17	01474	00925	BTM	TLU,TABDZ
05636	26	05654	00099	TF	**18,99Z
05648	14	00000	0R999	CM	,9999,8Z
05660	46	05764	01200	BZ	TSTRAZ
05672	26	05690	00099	TF	**18,99Z
05684	16	00000	0R998	TFM	,9998,8Z
05696	12	05690	00004	SM	*-6,4,10Z
05708	26	05726	05690	TF	**18,*-18Z
05720	16	00000	0R998	TFM	,9998,8Z
05732	11	00099	00008	AM	99,8,10Z
05744	26	00930	00099	TF	TABD+5,99Z
05756	49	05624	00000	B	TSTAJ+24Z
05764				DORG	*-3Z
05764	26	05782	04703	TSTRA	TF **18,TJBDI+5Z
05776	14	00000	0R999	CM	,9999,8Z
05788	46	05984	01200	BZ	TSTCAZ
05800	26	00925	05782	TF	TABD,TSTRA+18Z
05812	26	00930	00975	TF	TABD+5,DUMADZ
05824	17	01474	00925	BTM	TLU,TABDZ
05836	26	05854	00099	TF	**18,99Z
05848	14	00000	0R999	CM	,9999,8Z
05860	46	05964	01200	BZ	TSTRBZ
05872	26	05890	00099	TF	**18,99Z
05884	16	00000	0R998	TFM	,9998,8Z
05896	12	05890	00004	SM	*-6,4,10Z
05908	26	05926	05890	TF	**18,*-18Z
05920	16	00000	0R998	TFM	,9998,8Z
05932	11	00099	00008	AM	99,8,10Z
05944	26	00930	00099	TF	TABD+5,99Z
05956	49	05824	00000	B	TSTRA+60Z
05964				DORG	*-3Z
05964	12	05782	00004	TSTRB	SM TSTRA+18,4,10Z
05976	49	05776	00000	B	TSTRA+12Z
05984				DORG	*-3Z
05984	26	06002	04718	TSTCA	TF **18,TJBDI+5Z
05996	14	00000	0R999	CM	,9999,8Z
06008	46	06180	01200	BZ	TSTALZ
06020	16	01000	07898	TFM	TABJT+5,TABLES+4Z
06032	26	00995	06002	TF	TABJT,TSTCA+18Z
06044	17	01474	00995	BTM	TLU,TABJTZ
06056	26	06074	00099	TF	**18,99Z
06068	14	00000	0R999	CM	,9999,8Z
06080	46	06160	01200	BZ	TSTAMZ
06092	26	06127	04728	TF	**35,JOBIZ
06104	26	06122	00099	TF	**18,99Z
06116	26	00000	00000	TF	Z
06128	11	00099	000J2	AM	99,12,10Z
06140	26	01000	00099	TF	TABJT+5,99Z
06152	49	06044	00000	B	TSTCA+60Z
06160				DORG	*-3Z

06160	11	06002	00004	TSTAM	AM TSTCA+18,4,10Z
06172	49	05996	00000	B	TSTCA+12Z
06180				DORG	*-3Z
06180	26	06198	04718	TSTAL	TF **18,TJBDI+5Z
06192	14	00000	0R999	CM	,9999,8Z
06204	46	06376	01200	BZ	TSTBAZ
06216	26	00930	00975	TF	TABD+5,DUMADZ
06228	26	00925	06198	TF	TABD,TSTAL+18Z
06240	17	01474	00925	BTM	TLU,TABDZ
06252	26	06270	00099	TF	**18,99Z
06264	14	00000	0R999	CM	,9999,8Z
06276	46	06356	01200	BZ	TSTVVZ
06288	26	06323	04728	TF	**35,JOBIZ
06300	26	06318	00099	TF	**18,99Z
06312	26	00000	00000	TF	Z
06324	11	00099	00008	AM	99,8,10Z
06336	26	00930	00099	TF	TABD+5,99Z
06348	49	06240	00000	B	TSTAL+60Z
06356				DORG	*-3Z
06356	11	06198	00004	TSTVV	AM TSTAL+18,4,10Z
06368	49	06192	00000	B	TSTAL+12Z
06376				DORG	*-3Z
06376	26	06394	04718	TSTBA	TF **18,TJBDI+5Z
06388	14	00000	0R999	CM	,9999,8Z
06400	46	06572	01200	BZ	TSTDAZ
06412	16	01000	07894	TFM	TABJT+5,TABLESZ
06424	26	00995	06394	TF	TABJT,TSTBA+18Z
06436	17	01474	00995	BTM	TLU,TABJTZ
06448	26	06466	00099	TF	**18,99Z
06460	14	00000	0R999	CM	,9999,8Z
06472	46	06552	01200	BZ	TSTDBZ
06484	26	06519	04728	TF	**35,JOBIZ
06496	26	06514	00099	TF	**18,99Z
06508	26	00000	00000	TF	Z
06520	11	00099	000J2	AM	99,12,10Z
06532	26	01000	00099	TF	TABJT+5,99Z
06544	49	06436	00000	B	TSTBA+60Z
06552				DORG	*-3Z
06552	11	06394	00004	TSTDB	AM TSTBA+18,4,10Z
06564	49	06388	00000	B	TSTBA+12Z
06572				DORG	*-3Z
06572	26	06590	04718	TSTDA	TF **18,TJBDI+5Z
06584	14	00000	0R999	CM	,9999,8Z
06596	46	06780	01200	BZ	TSTAZZ
06608	26	00930	00975	TF	TABD+5,DUMADZ
06620	12	00930	00004	SM	TABD+5,4,10Z
06632	26	00925	06590	TF	TABD,TSTDA+18Z
06644	17	01474	00925	BTM	TLU,TABDZ
06656	26	06674	00099	TF	**18,99Z
06668	14	00000	0R999	CM	,9999,8Z
06680	46	06760	01200	BZ	TSTDDZ
06692	26	06727	04728	TF	**35,JOBIZ
06704	26	06722	00099	TF	**18,99Z
06716	26	00000	00000	TF	Z
06728	11	00099	00008	AM	99,8,10Z
06740	26	00930	00099	TF	TABD+5,99Z
06752	49	06644	00000	B	TSTDA+72Z

06760			DORG	*-3Z
06760	11	06590	00004	TSTDD AM TSTDA+18,4,10Z
06772	49	06584	00000	B TSTDA+12Z
06780			DORG	*-3Z
06780	11	04728	000J2	TSTAZ AM JOBI,12,10Z
06792	26	06810	04728	TF **18,JOBIZ
06804	14	00000	0R999	CM ,9999,8Z
06816	47	04278	01200	BNZ TSTAZ
* THI S RO UTINE DROPS IDENTICAL DUMMIES.Z				
06828	26	00925	00975	TF TABD,DUMADZ
06840	26	00930	00975	DRE1 TF TABD+5,DUMADZ
06852	17	01474	00925	BTM TLU,TABDZ
06864	26	06882	00099	TF **18,99Z
06876	14	00000	0R999	CM ,9999,8Z
06888	46	07088	01200	BZ DRE3Z
06900	24	00099	00925	C 99,TABDZ
06912	46	07056	01200	BZ DRE2Z
06924	26	06983	00925	TF **59,TABDZ
06936	26	06978	00099	TF **42,99Z
06948	12	06978	00304	SM **30,4,10Z
06960	12	06983	00004	SM **23,4,10Z
06972	24	00000	00000	C Z
06984	47	07056	01200	BNZ DRE2Z
06996	26	07014	00099	TF **18,99Z
07008	16	03000	0R998	TFM ,9998,8Z
07020	12	07014	00004	SM *-6,4,10Z
07032	26	07050	07014	TF **18,*-18Z
07044	16	00000	0R998	TFM ,9998,8Z
07056	11	00099	00008	DRE2 AM 99,8,10Z
07068	26	00930	00099	TF TABD+5,99Z
07080	49	06852	00000	B DRE1+12Z
07088			DORG	*-3Z
07088	11	00925	00008	DRE3 AM TABD,8,10Z
07100	26	07118	00925	TF **18,TABDZ
07112	14	00000	0R999	CM ,9999,8Z
07124	47	06840	01200	BNZ DRE1Z
07136	34	00000	00102	RCTY Z
07148	39	01007	00100	WATY EPIZ
07160	48	00000	00000	H Z
* ROU TINE FOR PUNCHING THE JOB CARDS.Z				
07172	31	00406	00567	TR INPUT-1,ZEROZ
07184	32	00406	00000	SF INPUT-1Z
07196	32	00408	00000	SF INPUT+2*2-3Z
07208	32	00416	00000	SF INPUT+2*6-3Z
07220	32	00427	00000	SF INPUT+2*11-2Z
07232	37	00407	00500	READ2 RACD INPUTZ
07244	31	00728	00567	TR OUTPUT-1,ZEROZ
07256	14	00407	00000	CM INPUT,,10Z
07268	47	07552	01200	BNZ READ2AZ
07280	27	01302	00433	BT SUB1,INPUT+2*14-2Z
07292	17	01474	00910	BTM TLU,TABJZ
07304	26	07322	00099	TF **18,99Z
07316	14	00000	0R999	CM ,9999,8Z
07328	47	07396	01200	BNZ **68Z
07340	17	01398	00004	BTM ERMES,4,9Z
07352	26	07370	01301	TF **18,SUB1-1Z
07364	34	00000	00102	RCTY Z

07371		00001	
07376	38	07367	00100
07388	49	07552	00000
07396			
07396	12	00099	00004
07408	26	07431	00099
07420	27	01090	00000
07432	26	00743	01089
07444	12	00099	00004
07456	26	07479	00099
07468	27	01090	00000
07480	26	00735	01089
07492	26	00751	00415
07504	26	00761	00425
07516	27	01090	01301
07528	26	00769	01089
07540	39	00729	00400
07552	47	07232	00900

07564	31	00728	00567
07576	16	00763	00004
07588	16	00761	00670
07600	16	00751	00670
07612	16	00940	00001
07624	26	07642	00975
07636	14	00000	0R999
07648	47	07704	01200
07660	34	00000	00102
07672	39	01029	00100
07684	48	00000	00000
07696	49	01742	00000
07704			
07704	26	07722	00975
07716	14	00000	0R998
07728	46	07872	01200
07740	26	07775	00975
07752	12	07775	00004
07764	27	01090	00000
07776	26	00735	01089
07788	26	07811	00975
07800	27	01090	00000
07812	26	00743	01089
07824	27	01090	00940
07836	26	00771	01089
07848	39	00729	00400
07860	11	00940	00001
07872	11	00975	00008
07884	49	07624	00000
07894		00000	
01690			

DC	1,@,*-4Z
WNTY	*-9Z
B	READ2AZ
DORG	*-3Z
SM	99,4,10Z
TF	**23,99Z
BT	SUB2Z
TF	OUTPUT+2*8-2,SUB2-1Z
SM	99,4,10Z
TF	**23,99Z
BT	SUB2Z
TF	OUTPUT+2*4-2,SUB2-1Z
TF	OUTPUT+2*12-2,INPUT+2*5-2Z
TF	OUTPUT+2*17-2,INPUT+2*10-2Z
BT	SUB2,SUB1-1Z
TF	OUTPUT+2*21-2,SUB2-1Z
WACD	OUTPUTZ
BNLC	READ2Z
* ROU	TINE FOR PUNCHING THE DUMMY CARDS.Z
TR	OUTPUT-1,ZEROZ
TFM	OUTPUT+2*18-2,44,10Z
TFM	OUTPUT+2*17-2,7070,8Z
TFM	OUTPUT+2*12-2,7070,8Z
TFM	DUMMY,1,8Z
PUNCHD	TF **18,DUMADZ
CM	,9999,8Z
BNZ	**56Z
RCTY	Z
WATY	PCZ
H	Z
B	INITZ
DORG	*-3Z
TF	**18,DUMADZ
CM	,9998,8Z
BZ	PCHENDZ
TF	**35,DUMADZ
SM	**23,4,10Z
BT	SUB2Z
TF	OUTPUT+2*4-2,SUB2-1Z
TF	**23,DUMADZ
BT	SUB2Z
TF	OUTPUT+2*8-2,SUB2-1Z
BT	SUB2,DUMMYZ
TF	OUTPUT+2*22-2,SUB2-1Z
WACD	OUTPUTZ
AM	DUMMY,1,10Z
PCHEND	AM DUMAD,8,10Z
B	PUNCHDZ
TABLES	DS **1Z
DEND	STARTZ