Linear Programming Code for the IBM 1620 with Card Input and Output

Linear Programming Code for the IBM 1620 with Card Input and Output

R. Dietz
IBM Corporation
Queens, New York

Modifications or revisions to this program, as they occur, will be announced in the appropriate Catalog of Programs for the IBM Data Processing Systems. If such announcement indicates a change to the program decks or tapes, a complete new program, if needed, should be requested from the Program Distribution Center.

Acknowledgement

This program is a modification of one for the basic 1620 with paper tape input and output written by C. R. Nichols, IBM, Inglewood, California. The only changes made were those necessary to enable his program to be used with a card 1620. Much of the information in the writeup is taken directly from his writeup.

TABLE OF CONTENTS

I. PROGRAM DESCRIPTION

This program solves linear programming problems with output of detailed results. That is, given coefficients a_{ij}, cost coefficients c_j, and requirements b_i, determine x_j such that

$$\sum_j a_{ij}x_j = b_i \qquad \text{with } x_j \geq 0$$

and

$$\sum_j c_j x_j = \text{maximum}$$

Computations are performed by the Dual Algorithm until a feasible solution is obtained. Control is then given to the Simplex Algorithm for optimization. Cost changes and requirement changes can be made after loading original matrix or after solving original matrix.

A. Machine requirements

1. Basic 1620 with card input and output.

2. Any size storage can be used. The larger the storage, the larger the problem that can be solved. The size of the problem is restricted by the following relationship:

$$(m + 2)\ (n + 3) \leq \frac{\text{memory} - 3920}{10}$$

where: m is the number of restrictions
n is the number of non-basis independent variables
memory is 20,000, 40,000, or 60,000

3. Source language: actual machine language

4. Subroutines: floating point subroutines supplied with program. No other subroutines needed.

5. Input media: card reader and console typewriter.

6. Output media: card punch and console typewriter.

7. Running time: The precise time required per iteration depends on the size and density of the matrix. As an approximation, a problem with 30 equations and 40 non-basis variables requires about 20 seconds per iteration.

8. Decimal accuracy: All computations are performed in 2-and-8 floating point form. Matrix input can be either fixed point or floating point.

B. Error stops

1. All stops may be recognized by displaying IR-1 in MAR. MAR will contain the address of the halt instruction plus 12.

Halt instr.	Reason
01126	Floating point overflow
01288	Floating point attempted division by zero
00012	Loader - ready for next code
02458	Cost changer - Change entered for a non-existent identification number
00012	Cost changer - ready for next code
03110	RHS changer - change entered for a non-existent identification number
00012	RHS changer - ready for next code
02534	Dual algorithm - inconsistent matrix
02654	Simplex algorithm - unbounded solution
02838	Matrix punch - ready for next code

2. Error procedure

   a. Cost changer (02458). Erroneous identification number is the last one typed.

   If data are being entered through the typewriter, depress RESET and INSERT.
   Type 49 02126.
   Depress RELEASE and START.
   Corrected or new change data may now be entered normally.

   If data are being entered from cards, the erroneous datum may be ignored by following the same steps as above (for entry through typewriter).
   If it is desired to enter the change correctly through the typewriter, the following steps must be taken:
   Set console switch two on.
   Depress RESET and INSERT.
   Type 49 02126.
   Depress RELEASE and START.
   Enter correct datum through typewriter (complete with record mark).
   Depress RELEASE and START.
   When typewriter is again selected, depress RESET.
   Set console switch two off.

Depress INSERT, RELEASE, and START.
Processing will now continue normally, reading data from cards.

b. RHS Changer (03110): Erroneous identification number is the last one typed.

If data are being entered through the typewriter, depress RESET and INSERT.
Type 49 02126.
Depress RELEASE and START.
Corrected or new change data may now be entered normally.

If data are being entered from cards, the erroneous data may be ignored by following the same steps as above (for entry through typewriter).
If it is desired to enter the change correctly through the typewriter, the following steps must be taken:
Set console switch three on.
Depress RESET and INSERT.
Type 49 02126.
Depress RELEASE and START.
Enter correct identification number (4 digits plus record mark) through typewriter.
Depress RELEASE and START.
Enter requirement change (including record mark) through typewriter.
Depress RELEASE and START.
When typewriter is again selected, depress RESET.
Set console switch three off.
Depress INSERT, RELEASE, and START.
Processing will now continue normally, reading data from cards.

c. Dual algorithm (02534): To obtain solution existing at the time of discovery of inconsistency:
Run out cards from reader.
Replace cards in reader hopper starting with Final Basis Output program.
Depress READER START.
Depress RESET and INSERT.
Type 49 00024.
Depress RELEASE and START.

d. Simplex Algorithm (02654): To obtain solution existing at the time of discovery of unbounded solution:

Depress RESET and INSERT.
Type 49 00024.
Depress RELEASE and START.

e. Overflow-underflow: Overflow causes a stop at 01126 or 01288. Underflow causes the result to be set to zero, and processing continues. Either condition indicates either machine malfunction or unreasonable data.

f. Any other error: unrecoverable. Restart problem.

II. THE PROGRAM

A. Operation Instructions

1. Set typewriter margins at 12 and 92; tabs at 24, 36, 48, 60, and 69.

2. It is important to note that no more than one instruction at a time may be inserted at any time after the initiation of the Loader routine.

3. To load initial data:

   a. Ready the LOADER deck in the card reader (including matrix input data). Depress READER START.

   b. Depress RESET and INSERT.

   c. Type 16 00010 00000.

   d. Depress RELEASE and START.
   This initiates memory clear routine.

   e. After at least one second, depress INSTANT STOP, RESET, and INSERT.

   f. Type 36 03826 00500
   49 03826

   g. Depress RELEASE and START.

   h. Program deck loads, data are loaded, and a halt at location 00012 is executed. Operation may now proceed to the COST CHANGER, RHS CHANGER, or SOLUTION programs.

4. To effect cost changes:

   a. Ready the COST CHANGER deck in the card reader (including data change cards if changes will be made from cards). Depress READER START.

   b. Set console switch two off if cost changes will be entered from card reader; switch two on if data will be entered through the typewriter.

   c. Depress START.

   d. If data are entered through the typewriter, RELEASE and START must be depressed after each entry. (Don't forget the record mark to terminate each entry.)

   e. After all data are entered a halt at location 00012 is executed. Operation may now proceed to the RHS CHANGER or SOLUTION programs.

5. To effect requirement changes:

   a. Ready the RHS CHANGER deck in the card reader (including data change cards if changes will be made from cards). Depress READER START.

   b. Set console switch three off if requirement changes will be entered from card reader; switch three on if data will be entered through the typewriter.

   c. Depress START.

   d. If data are entered through the typewriter, RELEASE and START must be depressed after the entry of each identification and each change. (Don't forget the record mark to terminate each entry.)

   e. After all data are entered a halt at location 00012 is executed. Operation may now proceed to the COST CHANGER or SOLUTION programs.

6. To solve and obtain output:

   a. Ready SOLVER deck in the card reader. Depress READER START.

   b. Set console switch one on to type iterations, off to omit iteration typeout.

c. Set console switch four on to obtain punch-out of final matrix, off to suppress punching.

d. Depress START.

e. After all output is complete, a halt at location 02838 will be executed. Operation may now proceed to the LOADER, COST CHANGER, or RHS CHANGER programs.

B. Data Preparation

1. The first card to be entered is a case identification card. A five digit alphabetic identification is punched in columns 1 through 5. Columns 6 through 80 are blank.

2. The second card specifies the size of the problem being entered. It is punched as follows:

Information	Card columns
$\bar{0}$ (zero with a minus sign)	1
number of equations (3 digits)	2 - 4
0 (zero)	5
$\bar{0}$ (zero with a minus sign)	6
number of non-basis variables (3 dig.)	7 - 9
0 (zero)	10
input code	11
a. 1 for row-column, fixed point input	
b. 0 for a complete floating point matrix	

Columns 12 through 80 are blank.

3. The matrix can be entered in one of two forms: floating point (the entire matrix is entered, column by column, exactly as it is to be stored, including zero entries) or fixed point (entries can be in any order, accompanied by their respective row-column designations, and zero elements need not be entered).

4. For floating point entry, eight elements are punched on each card, with a minus sign over the first position of each ten position field, and a minus sign over the last position of negative fields (units position). There are ten positions per field, with the first two positions specifying placement of the decimal point (50 designates a decimal point to the left of the

first non-zero digit; 51 designates one non-zero digit to the left of the decimal; 49 designates that the first non-zero digit is in the hundredths position, etc.) and the last eight digits (with no leading zeroes) giving the element itself. A zero element is specified by ten zeroes, with a minus sign over the first.

The elements must be punched in column order, starting with the first basis cost. See the matrix layout included with this writeup to know how the elements are stored in the matrix.

5. Fixed point matrix elements are entered four to a card.

Two types of cards are necessary for matrix elements:

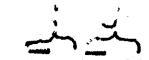
a. Row-column cards designate where the elements will be stored in the matrix. Row-column designations are punched in columns 1-6, 21-26, 41-46, and 61-66 in the form $\bar{R}RR\bar{C}CC$ (a minus sign accompanies the first digit of each row and column designation). The column designation specifies column number considering only non-basis vectors. The requirement variables have only a row designation, and $\bar{0}00$ should be entered for their column.

b. Matrix element cards contain fixed-point entries including optional leading sign, up to eight numeric digits, and mandatory decimal point. Each entry must be terminated by a record mark (0-2-8 punch). The four elements are punched beginning in columns 1, 21, 41, and 61 for each card.

Elements may be entered in any order, but corresponding row-column designations and matrix entries must be in the same positions of successive cards (row-column card preceding).

Loading of zero elements is optional

If any card contains fewer than four elements, the first blank field (column 21, 41, or 61) of the row-column card must be punched with a record mark (0-2-8 punch).

Item	Description	Row-column
a_{ij}	matrix element of the ith row and jth non-basis vector	
b_i	element in the ith row of the requirement vector	

6. Fixed point cost entries are also four per card, but the row-column designation is on the same card as the cost.

Entries consist of a six digit row-column designation followed by a ten digit ID/cost, where the ID is a four digit numerical designation to identify the cost variable, and cost is a six digit fixed point entry with the decimal after the third digit.

Item	Description	Row-column
c_i	cost per unit of basis variable for row i	
c_j	cost per unit of jth non-basis vector	

Negative cost is signified by a minus sign over the units position of the cost field.

If any card contains fewer than four elements, the first blank field (column 21, 41, or 61) of that card must be punched with a record mark (0-2-8 punch).

7. The last data entry for fixed point mode must be 000 to terminate the loading operation. This can be punched in columns 1-3 of a new card, or in 21-23, 41-43, or 61-63 or the last row-column or cost entry card.

8. Entry of cost changes:

Data for the COST CHANGER routine can be entered either from cards or from the typewriter. If entered from cards, one entry is punched in each card, in columns 1-10. The entry consists of the four digit ID and six digit cost figure, with a minus sign over the first digit of the cost (column 5) and over the last digit of negative costs (column 10). Following the last cost change card must be a card punched with four zeroes (columns 1-4) to terminate the operation.

If entered from the typewriter, the format is the same as from cards: four digit ID and six digit cost figure with flag over the first digit of all cost figures, and over the units position of negative costs. The last entry must be four zeroes to terminate the operation.

9. Entry of requirement changes:

There are two entries for each requirement change: the identification and the change in the value of that requirement element. Note that the new requirement element is not entered, but rather the difference between the old and new requirement elements. If changes are entered from cards, the identification and change are entered on separate cards, one per card.

The identification is entered first, and consists of the ID associated with the cost element for that row. Thus, to change $b_1 = b_3$, use the ID associated with $c_1 = c_3$. No flags or record marks are needed in entering the ID either from cards or typewriter. It is punched in columns 1-4 of card entries, with 5-80 blank.

After entering the ID, its associated requirement change is entered (by typewriter or on a separate card). The entry starts in column 1, if a card is being used, and consists of optional leading sign, up to eight numeric digits, and mandatory decimal point. It must be terminated by a record mark, whether it is entered from cards or from the typewriter.

The last entry must be four zeroes to terminate the operation.

C. Interpretation of Results

1. Cost changes or requirement changes are typed if the input is from cards. If input is through the typewriter, the input operation itself produces a log of the changes made.

2. Dual Algorithm and Simplex Algorithm:

Results of iterations are typed only if console switch one is on. If it is desired to monitor the course of the solution onlh occasionally, this may be done by setting switch one on and off periodically. The following points should be borne in mind:

a. Page headings for iteration output are typed only if switch one is on at the beginning of the solution.
b. The iteration count is reset when the dual algorithm is finished and control passes to the simplex algorithm.

The information which may be typed includes the iteration count, the current value of the profit function (floating point), the last variable to leave the basis, and the variable which entered the basis.

3. Final Basis Output:

The final value of the profit function is given in floating point.

For each final basis variable, the following information is supplied (in fixed point):

a. Identification/cost coefficient.

b. Activity.

c. The limits of the cost coefficient over which the current solution is optimal.

d. The variables which limit the range of the cost coefficient and will enter the basis if a limit is exceeded.

4. Final Non-Basis Output:

For each non-basis variable, the following information is given (in fixed point):

a. Identification/cost coefficient.

b. Shadow price - the penalty to the total system if a unit of this variable is forced into the final solution.

c. The limits of activity over which the shadow price applies.

d. The variables which limit the range of applicability of the shadow price. The upper limiting variable is the one which would leave the basis if the associated non-basis variable were forced into the solution.

5. A punchout of the complete final matrix is optional under control of console switch four. The cards punched also contain all necessary supplementary information and are in a format which is suitable for direct re-loading by the LOADER routine.

D. Functions of Individual Sub-programs

1. The data load routine reads and stores parameters for the problem. It then proceeds to load and store the matrix in the proper format. In the case of fixed point input, storage locations are computed from the row/column designations, and the input elements are converted to floating point notation as they are read and stored.

2. The cost change routine reads ID/cost elements, searches the matrix for a corresponding identification number, and stores the new cost coefficient in the proper location.

3. The requirement change routine reads variable identification numbers and changes for the associated requirement element. The change is converted to floating point notations, the matrix is searched for the identification number, and the requirement vector is updated.

4. The shadow price calculator converts all cost coefficients to floating point notation and evaluates $Z_j - c_j$ for each non-basis variable.

5. The dual algorithm computes a feasible solution for the problem. Its operation may be monitored allowing the typing of the value of the functional and the identifications of the variables that enter into and are removed from the basis. It should be noted in this regard that the objectives of feasibility and optimality may not be compatible at this point, and therefore the functional may actually decrease from one iteration to the next while the dual algorithm is in control.

During the reduction of the matrix of coefficients a test is made to determine whether the resulting quantity has a magnitude less than an amount called "essential zero." Any element falling into this category is set to zero in order to avoid computation with numbers which are actually not significant. "Essential zero" has been set to 10^{-8}. Its value (the corresponding floating point exponent) is located in memory addresses

3144 and 3145 and may be changed if desired. If this change is made, it is important to note that the high order digit must carry a flag.

6. The simplex algorithm starts with any feasible solution and computes the optimal feasible solution. Its operation may be monitored by allowing iteration typeout as in the case of the dual algorithm.

   The discussion of essential zero as applied to the dual algorithm in 5. above is equally valid for the simplex algorithm. In this case the test value is located in memory addresses 3264 and 3265.

   In the case of a tie in the quotient used as the criterion for choosing the variable to leave the basis on any iteration, the choice is resolved by using the largest denominator as a secondary criterion.

7. The basis output routine searches out and computes the information described in section C. above. Floating point quantities (with the exception of the functional) are converted to fixed point before being typed.

8. The non-basis output routine searches out and computes the information described in section C. above. Floating point quantities are converted to fixed point before being typed.

9. The matrix punch routine resets the permanent instructions in low memory to allow re-initiation of the input or change phases. Problem parameters and the final matrix are punched into paper tape if called for by the setting of console switch four.

E. Use of Storage

Memory locations from 00012 through 03919 are used by the program. Memory from 03920 up to the required location is used for the matrix and its manipulation. Locations 00000 through 00011 are available for the insertion of one instruction at a time through the typewriter. A halt instruction in position 00012 ensures that only this one instruction will be executed.

00012-00042 load routine
00048-00058 case identification
00060-00064 S address

00065-00069	M and 2
00070-00074	N
00075-00079	M
00080-00099	product area
00100-00400	arithmetic tables
00402-02064	floating point subroutines
variable	programs
03758-03917	input area (record mark in 03918)
03920-	matrix

F. Program Decks

1. LOADER

3 pre-load cards.
Arithmetic tables.
Floating point routines.
Data load routine.
Matrix data to be entered.

2. COST CHANGER

2 pre-load cards.
Arithmetic tables.
Floating point routines.
Cost changer routine.
Cost change data (if entered from cards).

3. RHS CHANGER

2 pre-load cards.
Arithmetic tables.
Floating point routines.
Requirement change routine.
Requirement change data (if entered from cards).

4. SOLUTION

2 pre-load cards.
Shadow price calculation routine.
Page headings.
2 pre-load cards.
Dual algorithm routine.
2 pre-load cards.
Simplex algorithm.
2 pre-load cards.
Final basis output routine.
2 pre-load cards.
Fix and print routine.
Page headings.
2 pre-load cards.
Non-basis output routine.
Page headings.
2 Pre-load cards.
Matrix punch routine.

G. Storage of matrix in memory

The matrix is stored in column sequence and uses memory from 3920 upward to the extent required by the problem size. The following diagram indicates the manner in which matrix elements are stored:

3920			S+10(2m+3) ID/C_j	S+10[(n+1)(m+2)-1]
	S	Functional	S+10(2m+4)	S+10[(n+1)(m+2)]
Working column	ID/C_i	S+10(m+3) b_i	S+10(2m+5) Increasing j → ; a_{ij}: S+10[(j+1)(m+2)+i] ; Increasing i ↓	S+10[(n+1)(m+2)+1]
3920+10(m+1)	S+10m	S+10(2m+2)	S+10(3m+4)	S+10[(n+2)(m+2)-2]

S is computed for each problem and is equal to 3920 + 10(m+3). All addresses are field addresses.

III. SAMPLE PROBLEM

A. Matrix

Basis variables		Non-basis variables			Requirement Vector	Variable ID
9901	9992	0001	0002	9902		
1		1	2		6	
	1	1	-1	-1	-4	
0	-10	1	1	0		Cost

B. Matrix in storage

	$\bar{0}000000000$	$\bar{0}001001000$	$\bar{0}002001000$	9902000000
	$\bar{0}000000000$	$\bar{0}000000000$	$\bar{0}000000000$	$\bar{0}000000000$
$\bar{9}901000000$	$\bar{5}160000000$	$\bar{5}110000000$	$\bar{5}120000000$	$\bar{0}000000000$
$\bar{9}992010000$	$\bar{5}140000000$	$\bar{5}110000000$	$\bar{5}110000000$	$\bar{5}110000000$

C. Input

1. Fixed point

```
2 X 3
0002000301
001001          001002          002001          002002
1.+             2.+             1.+             1.+
002003          001000          002000          +
+1.+            6.+             -4.+
001001-9901000000   002001-9992010000  001001-0001001000  001002-0002001000
001003-9902000000   +
000
```

2. Floating point

```
2 X 3
000 000300
R90 000000R9920100000000000000000000000000N160000000N140000000000100100000000000
N110000000N110000000000200100000000000000N120000000N110000000R902000000000000000
00000 0 N11000000000000000000000000000000000000000000000000000000000000000000000
```

D. Output

1. Final matrix punched

```
2 X 3
00020000300
R902000000000100100000000000000N160000000N210000000N160000000000200100ON110000000
N130000000N120000000R992010000N210000000N110000000000000000000R901000000N110000000
N110000000N110000000000000000000000000000000000000000000000000000000000000000000000
```

2. Typewriter output

CASE 2 X 3

ITER NO	FUNCTIONAL	VAR OUT	VAR IN
001	5140000000	9992010000	0002001000
002	5130000000	9901000000	9902000000
001	5160000000	0002001000	0001001000
FUNCTIONAL	5160000000		

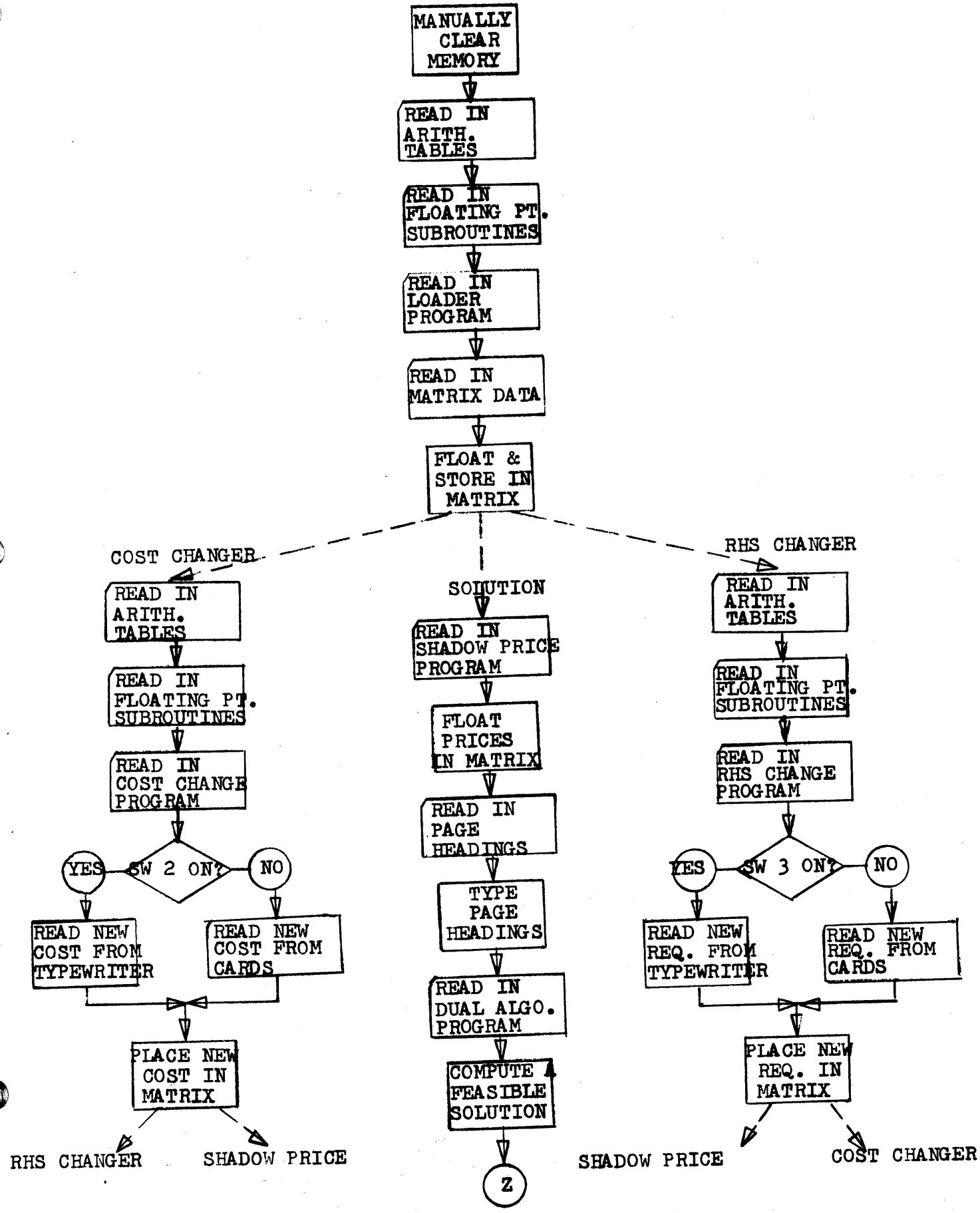
VAR/COST	ACTIVITY	LIM VAR	LOWER LIM	LIM VAR	UPPER LIM
9902000000	10.0000	0002	.3333-	9992	10.0000
0001001000	6.0000	0002	.5000	0000	9999.9000

VAR/COST	SHAD PRICE	LIM VAR	LOWER LIM	LIM VAR	UPPER LIM
0002001000	1.0000	0000	9999.9000-	0001	3.0000
9992010000	10.0000	9902	10.0000-	0000	9999.9000
9901000000	1.0000	0000	9999.9000-	0001	6.0000

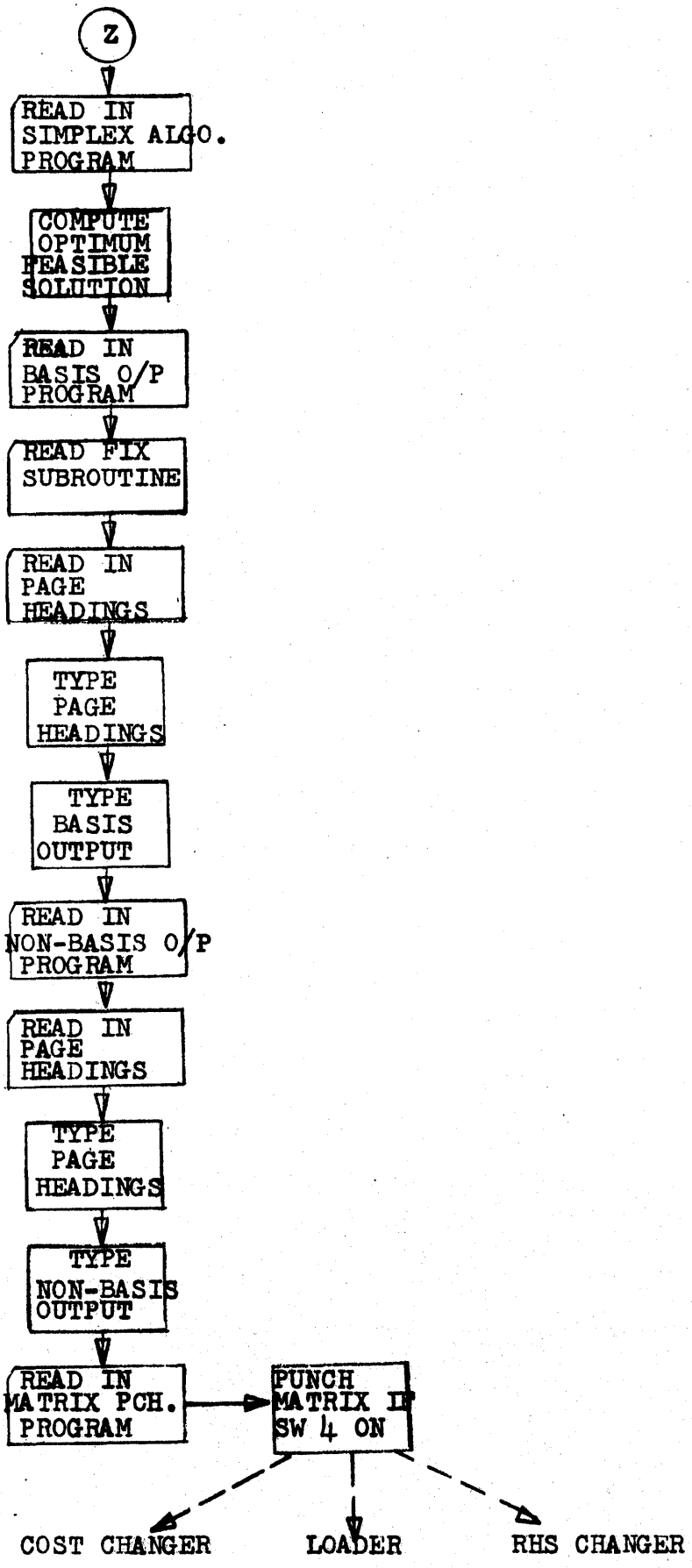
IV. METHOD OF LOADING PROGRAM

The "LOAD" key cannot be used to load programs in this routine since the first 80 positions of memory are not free. Therefore cards are read into locations 03838 through 03917, with a record mark in 03918 being utilized to transfer up to six instructions to their place in memory. The first two instructions on each card contain the "transfer record" instruction and a branch to location 03826, which contains the read instruction 36 03838 00500. The first two cards of each routine perform the function of setting the read instruction in 03826 and the record mark in 03918. The first card is read into 03826, and control is transferred there, using instructions 36 03826 00500 49 03826. This must be done by the operator at the beginning of the "LOADER" program, it is done automatically for all following routines.

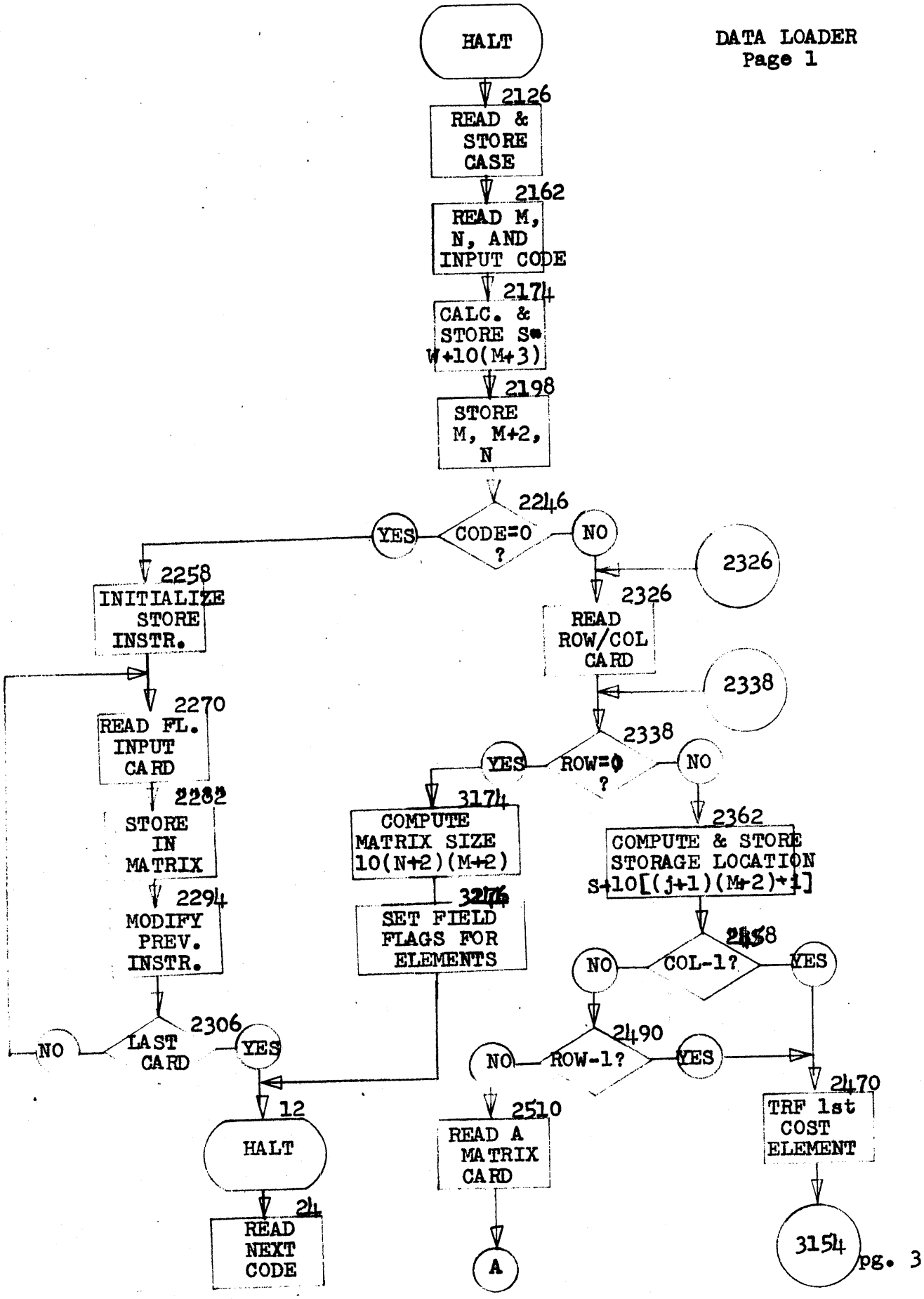
GENERAL FLOW
Page 1



GENERAL FLOW
Page 2

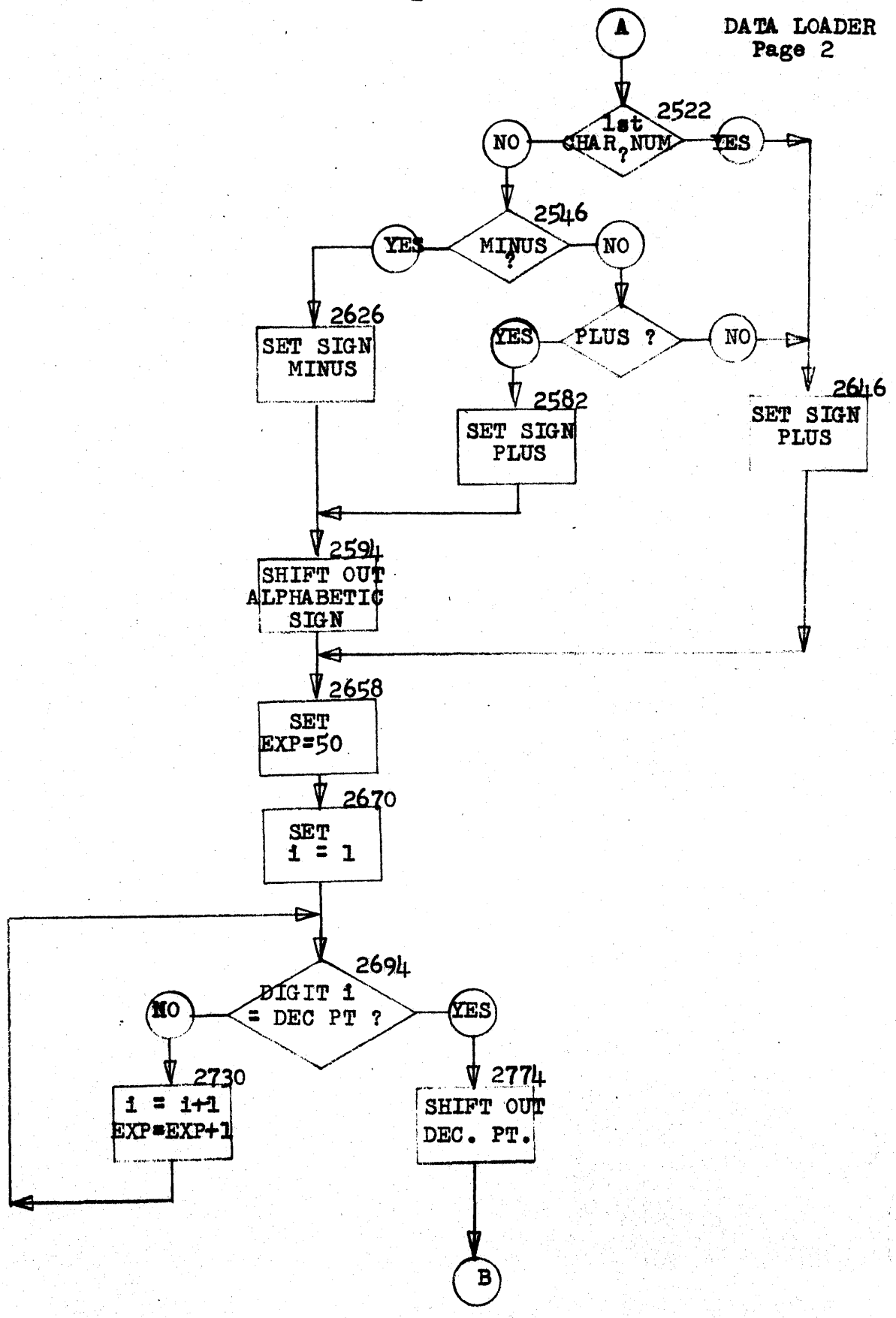


DATA LOADER
Page 1

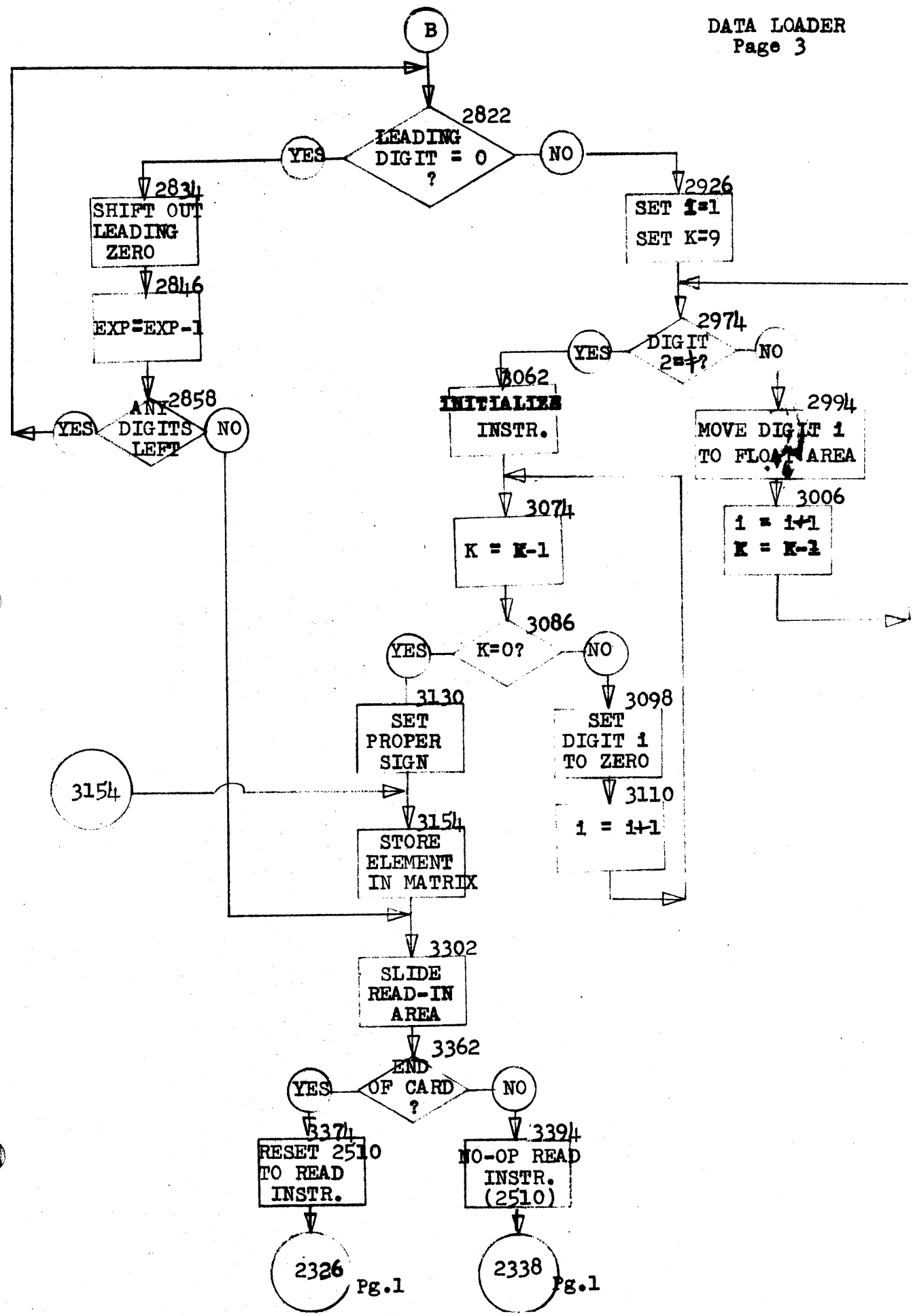


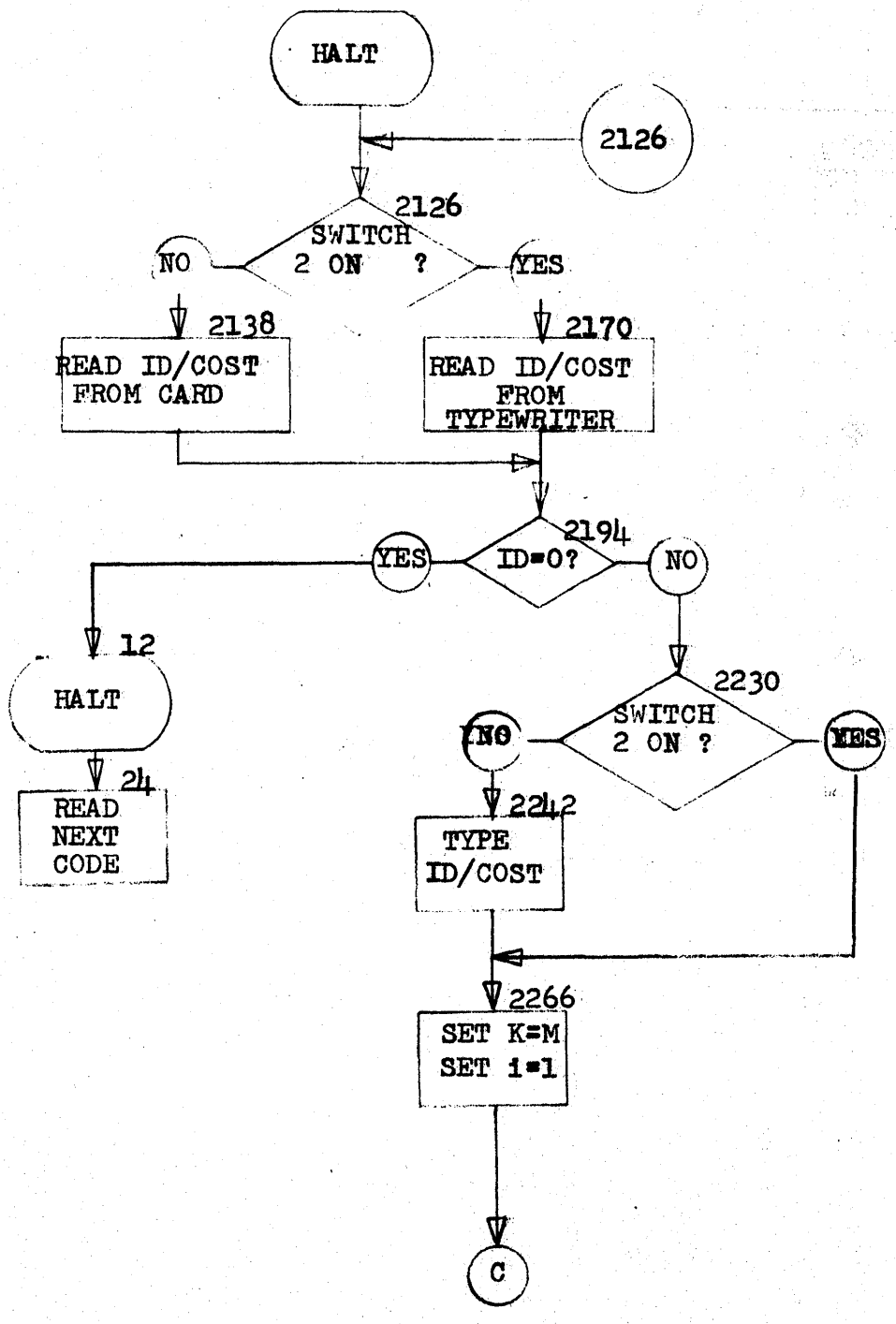
pg. 3

DATA LOADER
Page 2

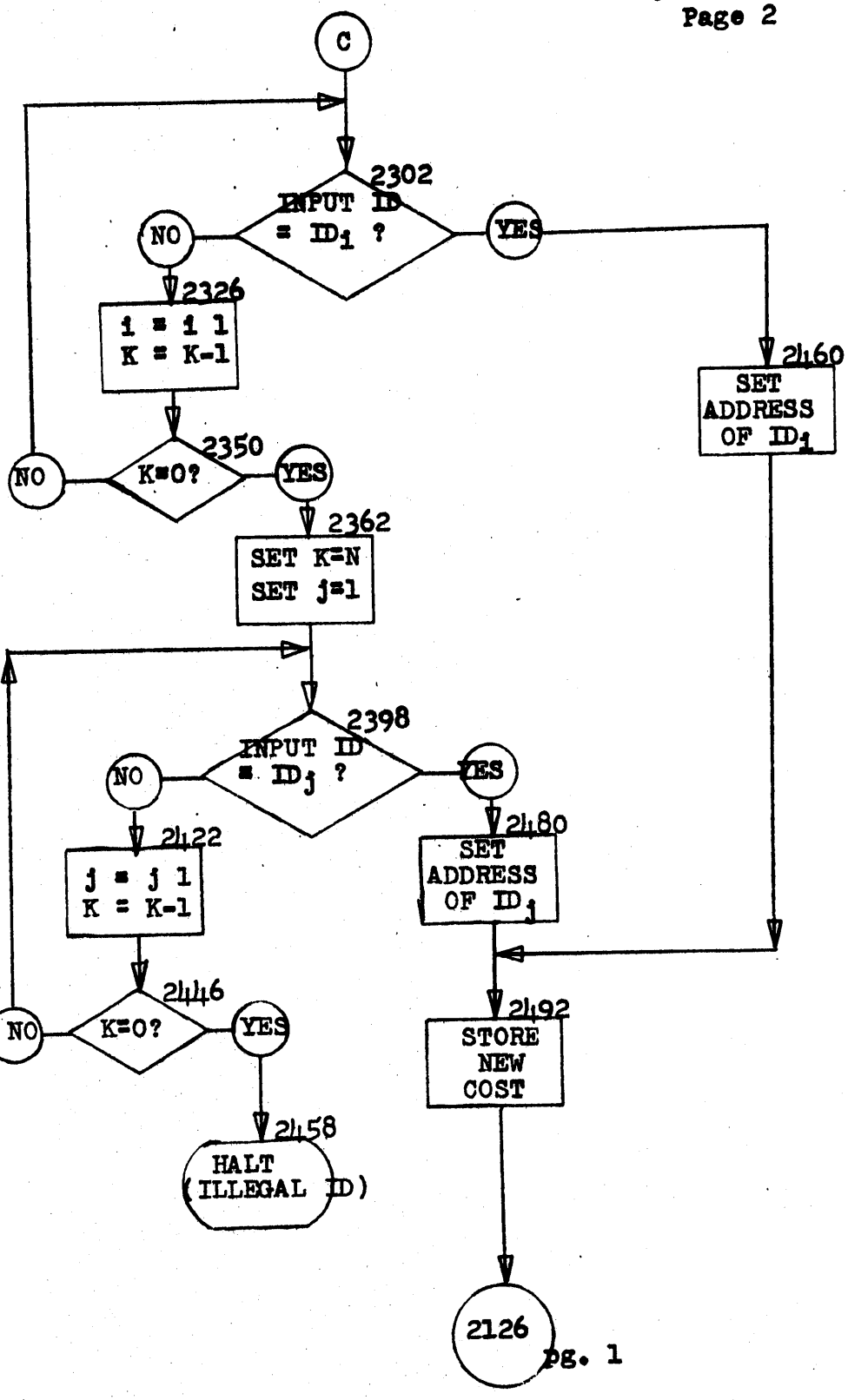


DATA LOADER
Page 3



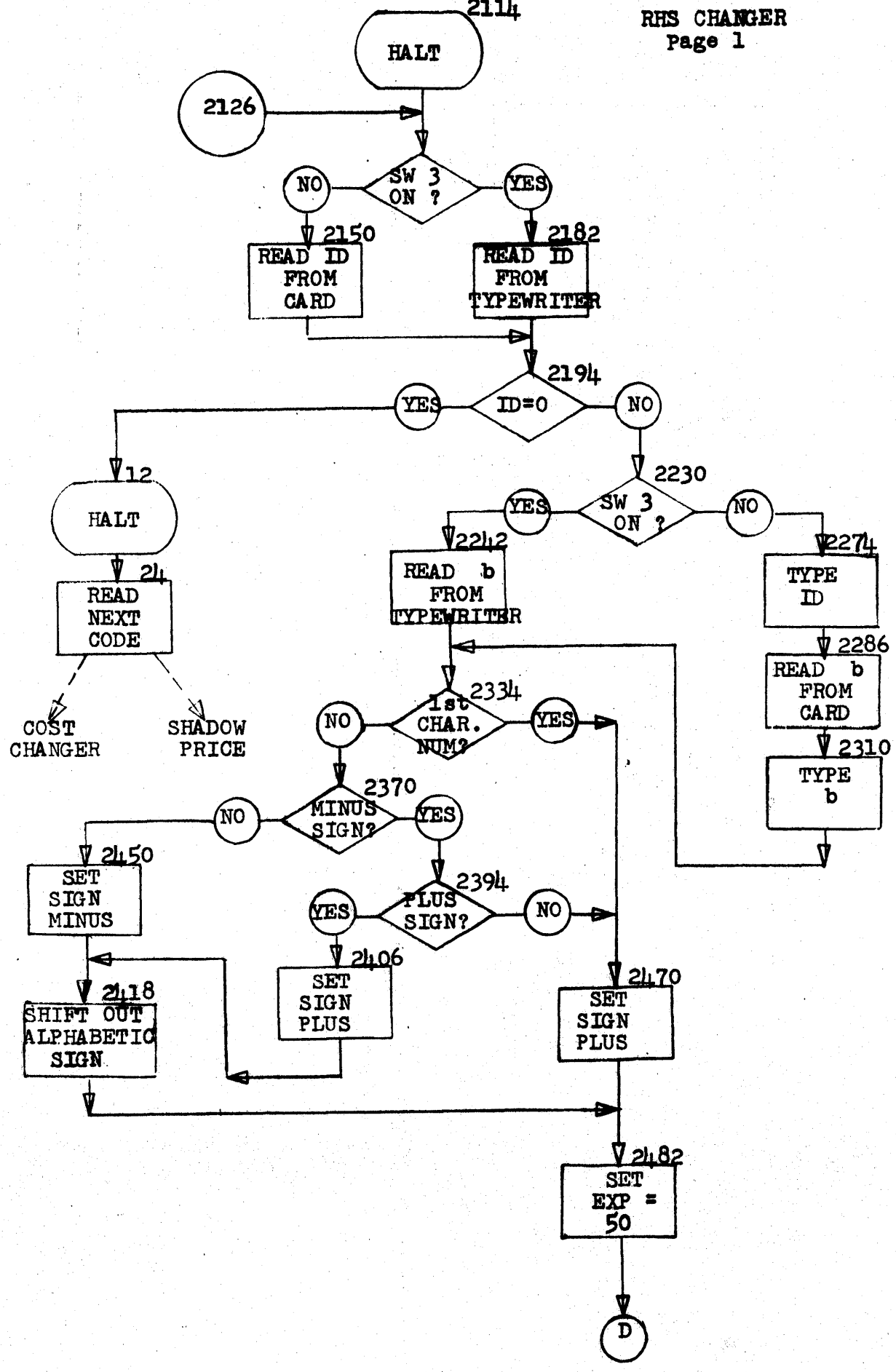

HALT
2126
2126
SWITCH
2 ON ?
NO
YES
2138
READ ID/COST
FROM CARD
2170
READ ID/COST
FROM
TYPEWRITER
2194
ID=0?
YES
NO
12
HALT
24
READ
NEXT
CODE
2230
SWITCH
2 ON ?
NO
YES
2242
TYPE
ID/COST
2266
SET K=M
SET i=1
C

COST CHANGER
Page 2

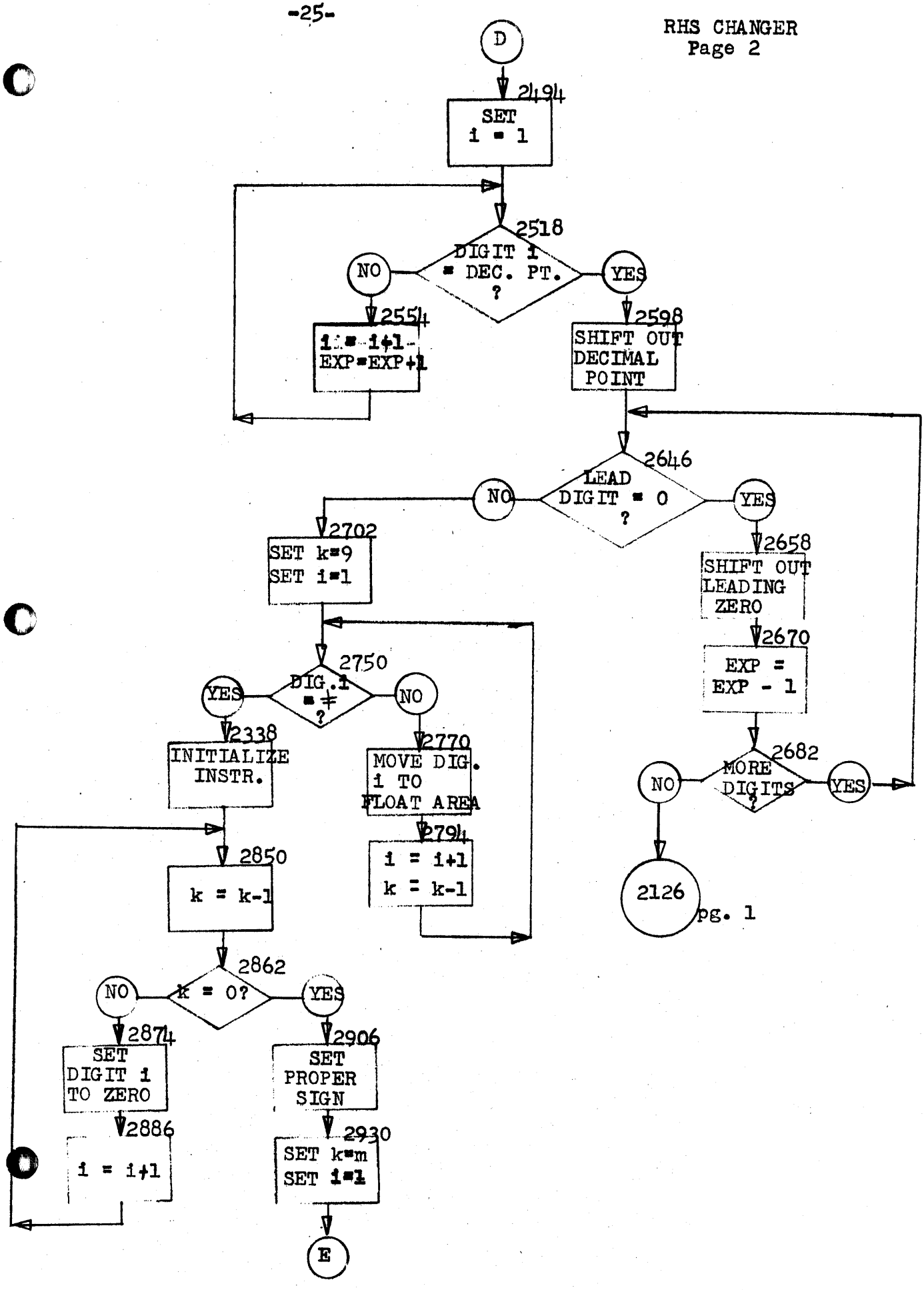


pg. 1

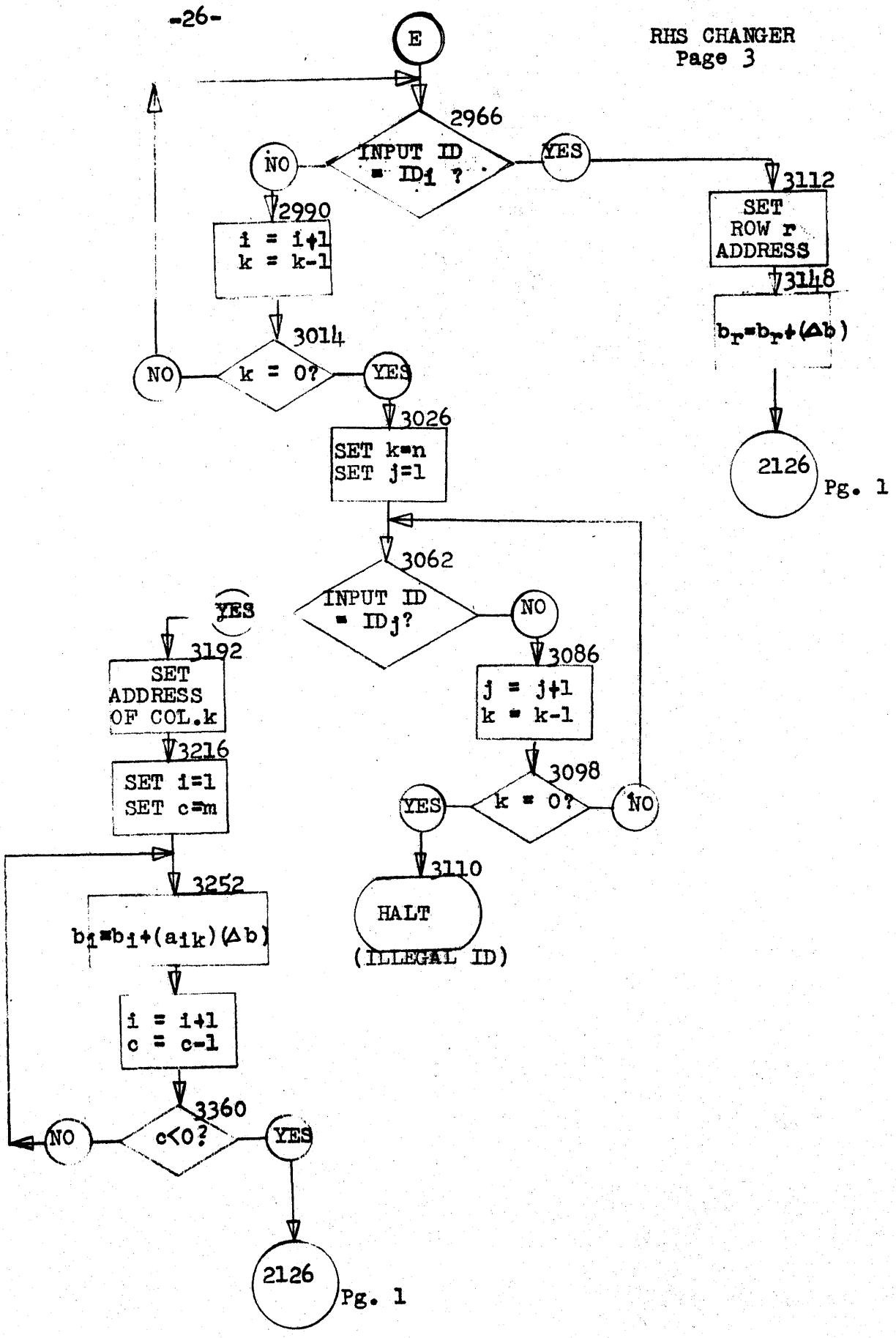
RHS CHANGER
Page 1

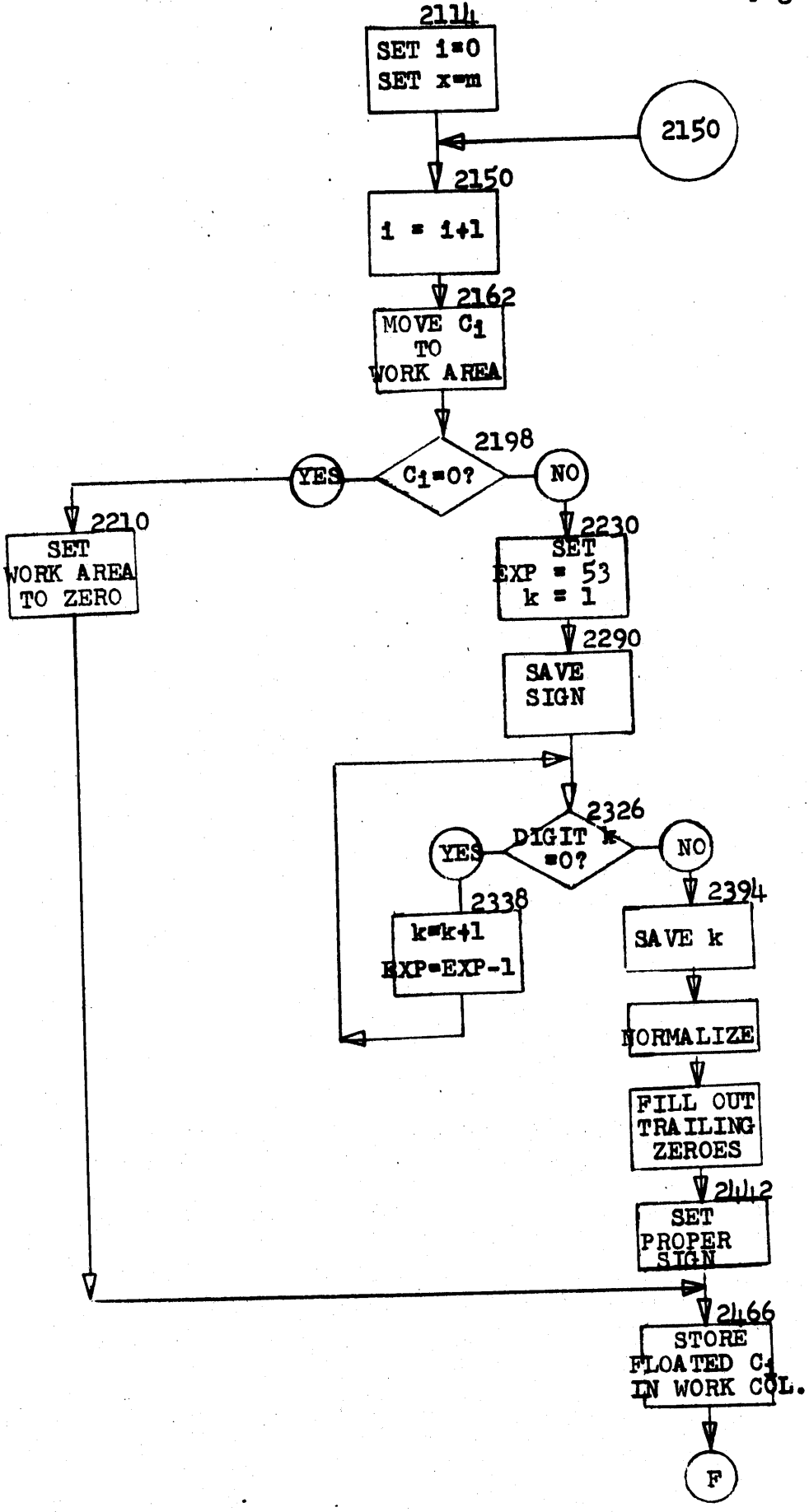


RHS CHANGER
Page 2

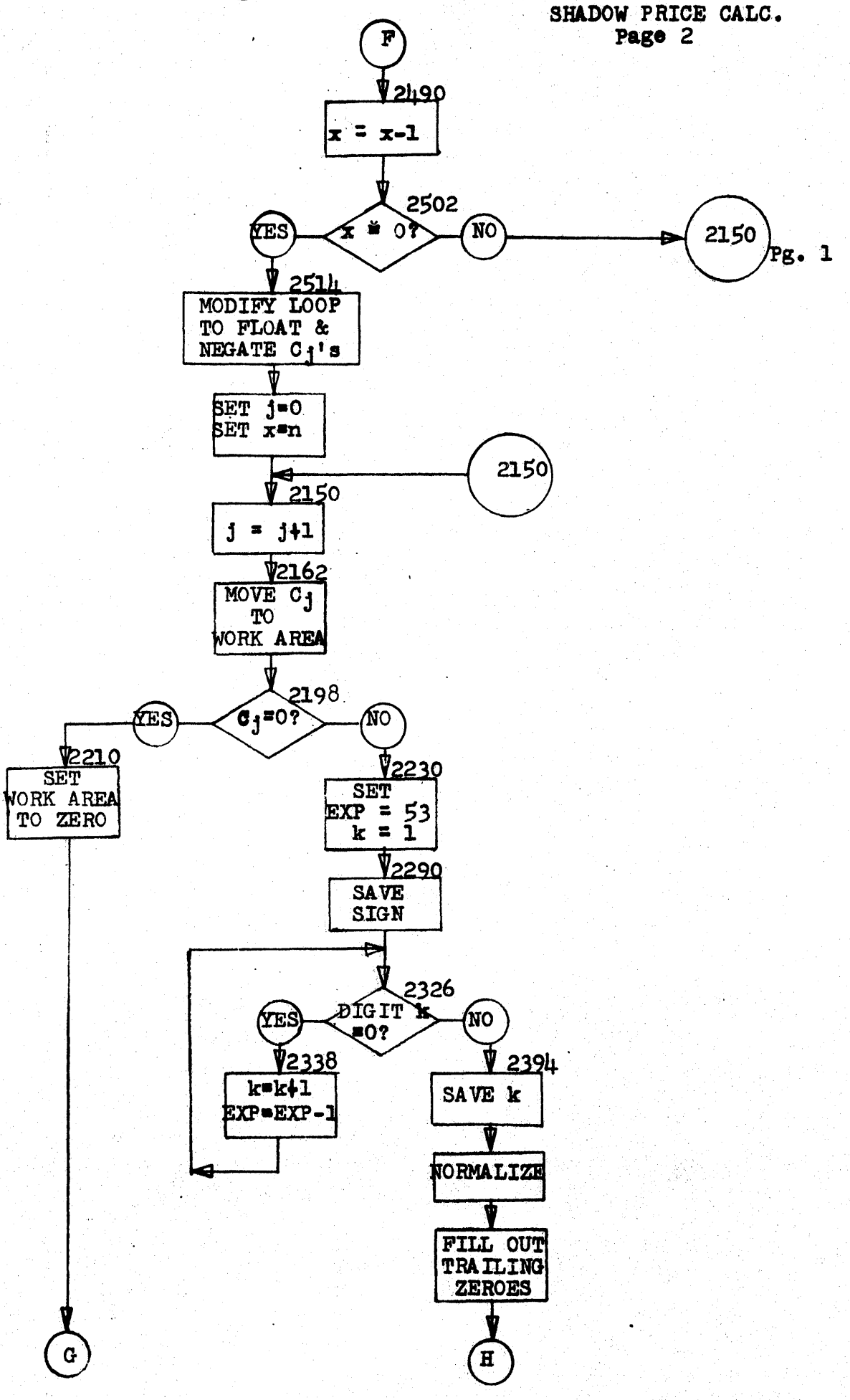


RHS CHANGER
Page 3

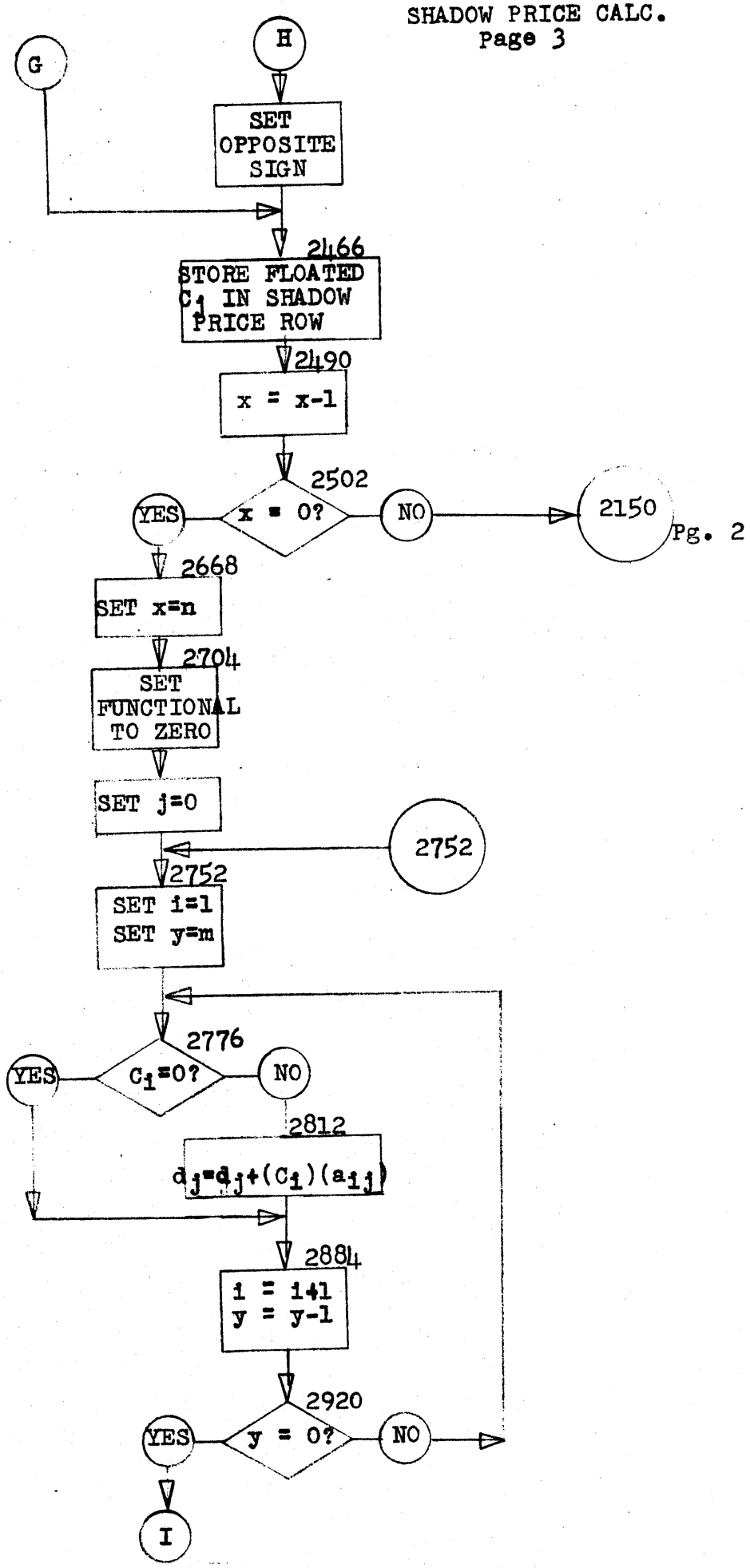



2114
SET i=0
SET x=m
2150
2150
i = i+1
2162
MOVE C1 TO WORK AREA
2198
C1=0?
YES
NO
2210
SET WORK AREA TO ZERO
2230
SET EXP = 53
k = 1
2290
SAVE SIGN
2326
DIGIT k =0?
YES
NO
2338
k=k+1
EXP=EXP-1
2394
SAVE k
NORMALIZE
FILL OUT TRAILING ZEROES
2442
SET PROPER SIGN
2466
STORE FLOATED C1 IN WORK COL.
F

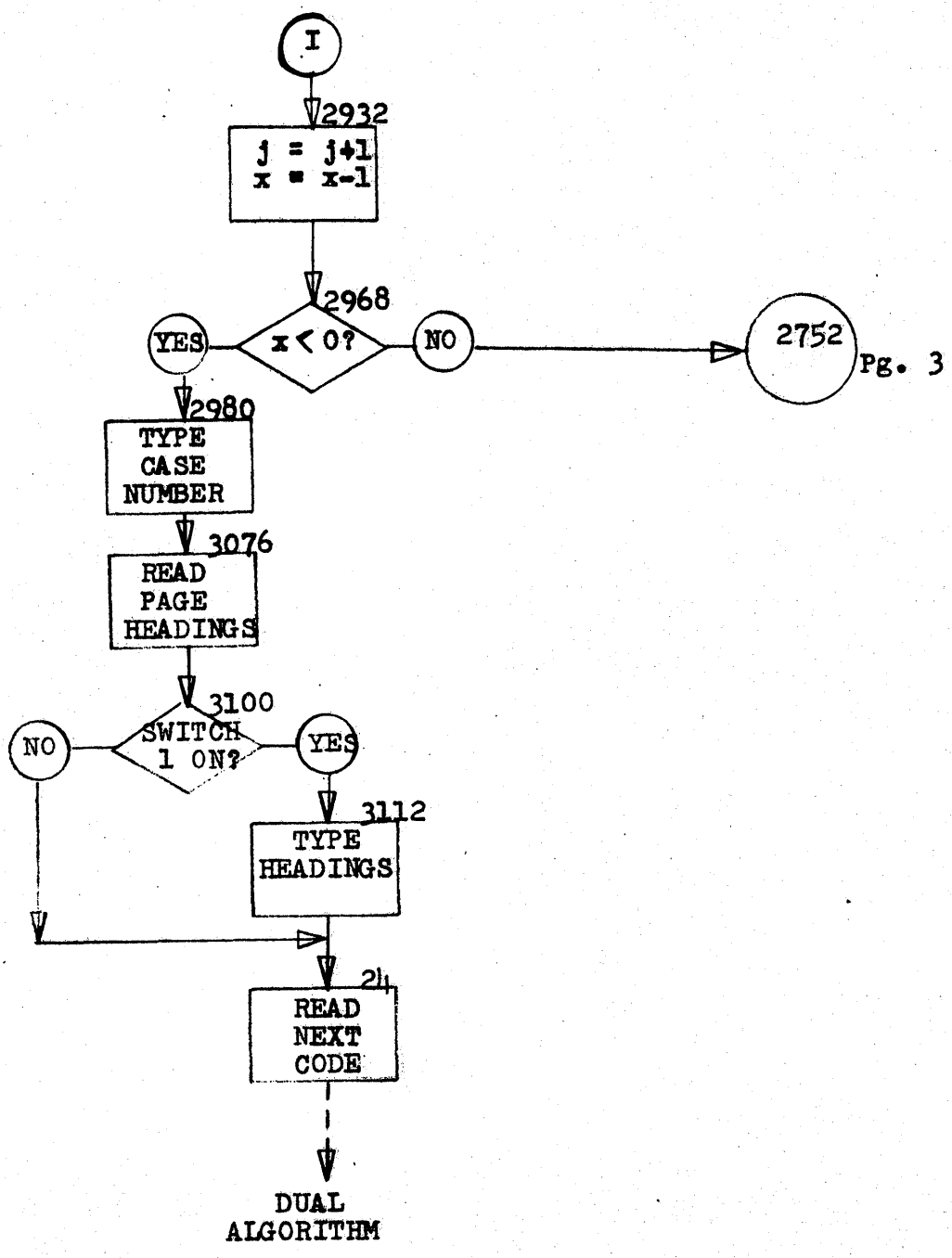
SHADOW PRICE CALC.
Page 2



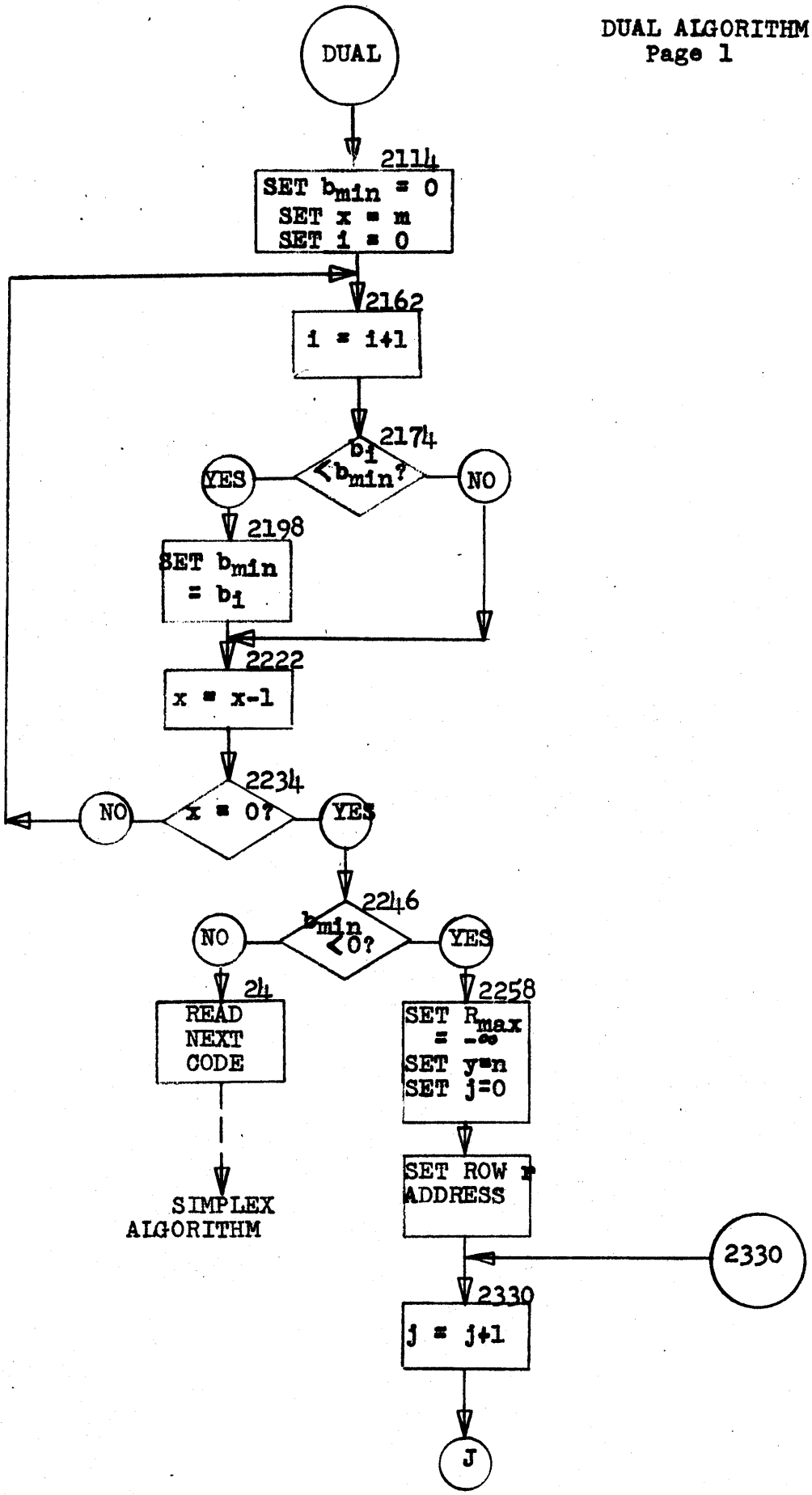
SHADOW PRICE CALC.
Page 3



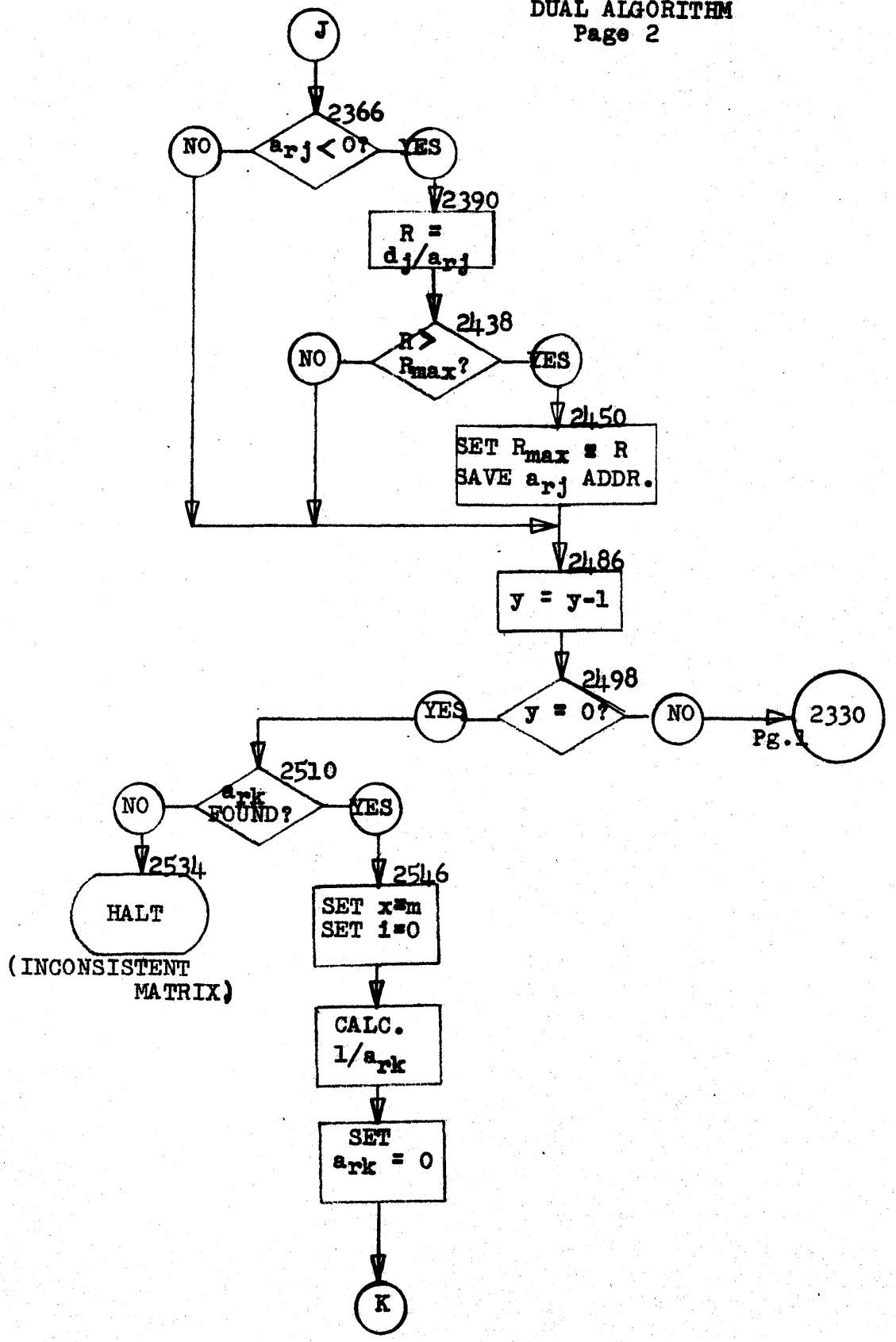
SHADOW PRICE CALC.
Page 4

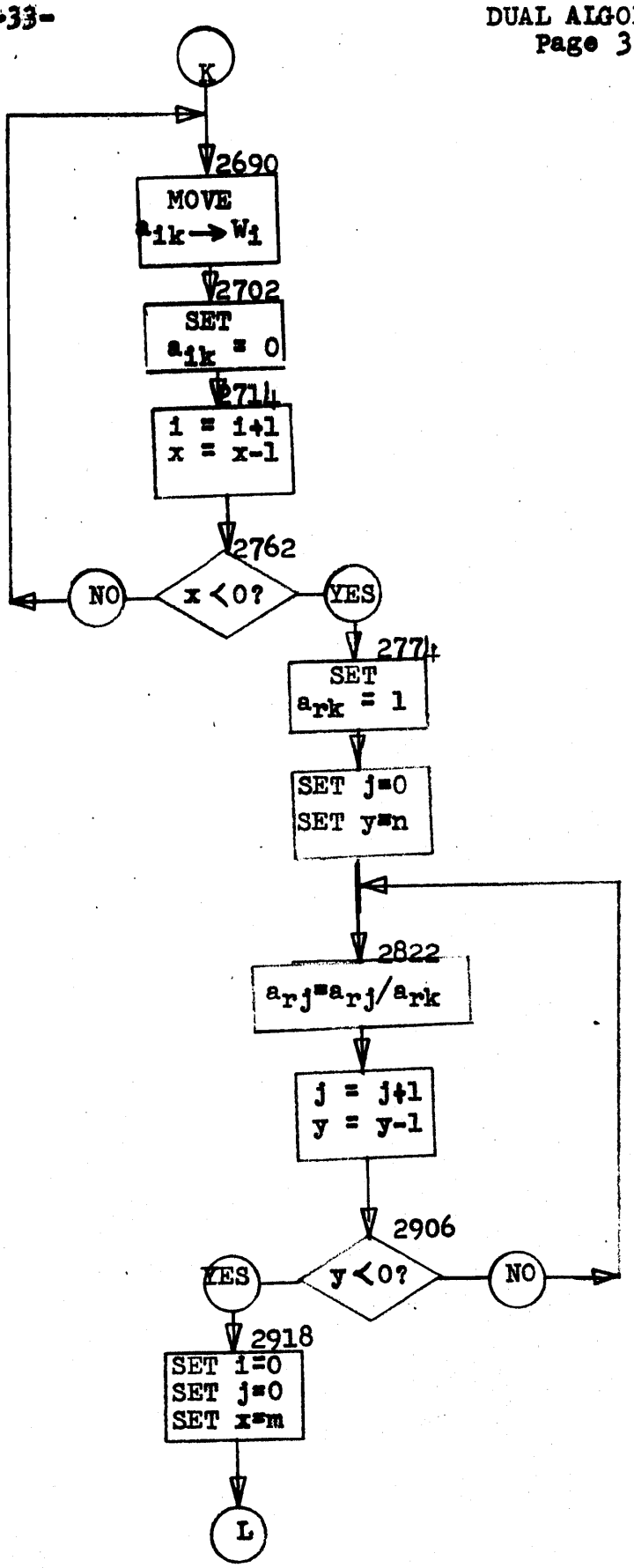


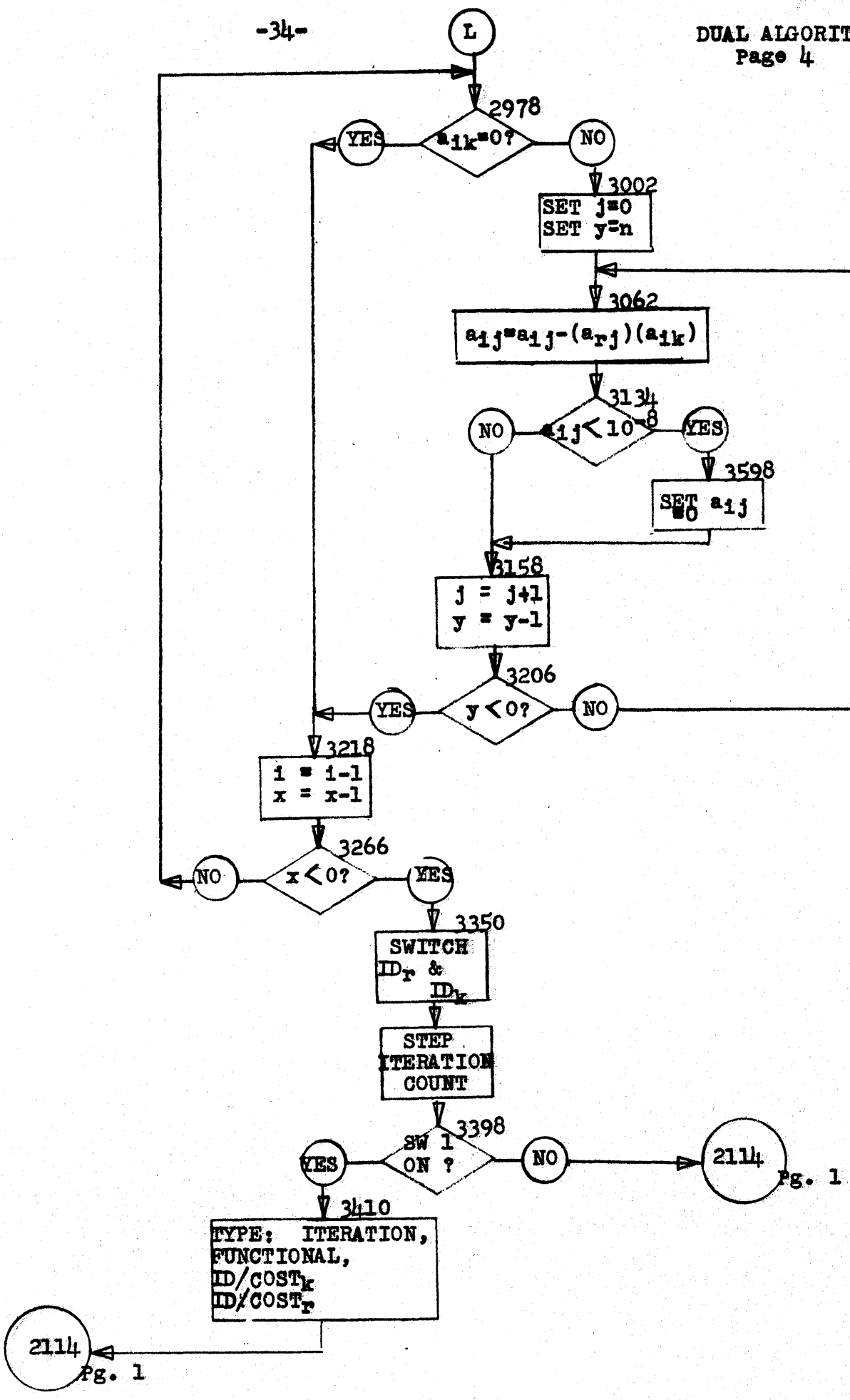
DUAL ALGORITHM
Page 1



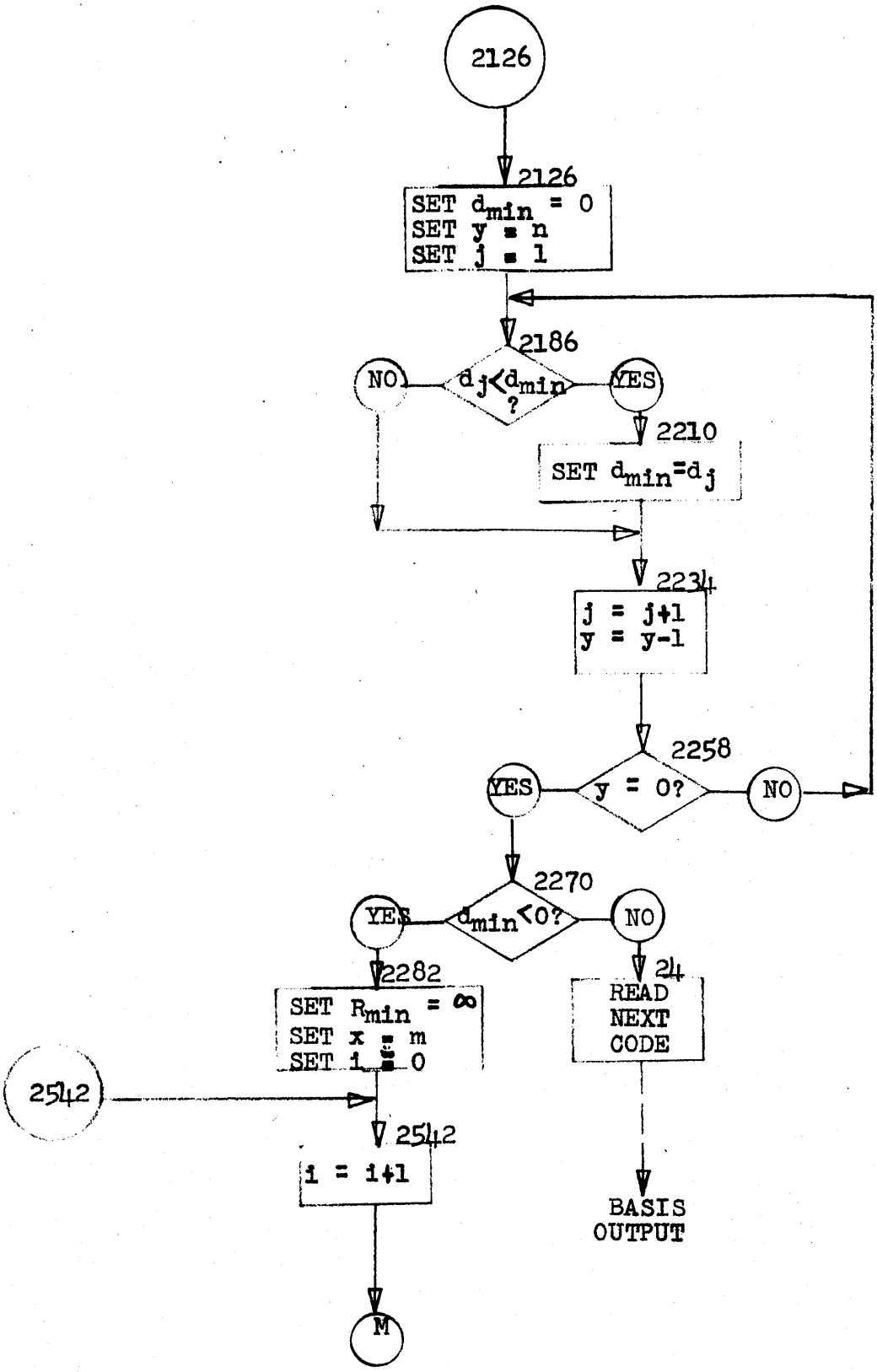
DUAL ALGORITHM
Page 2

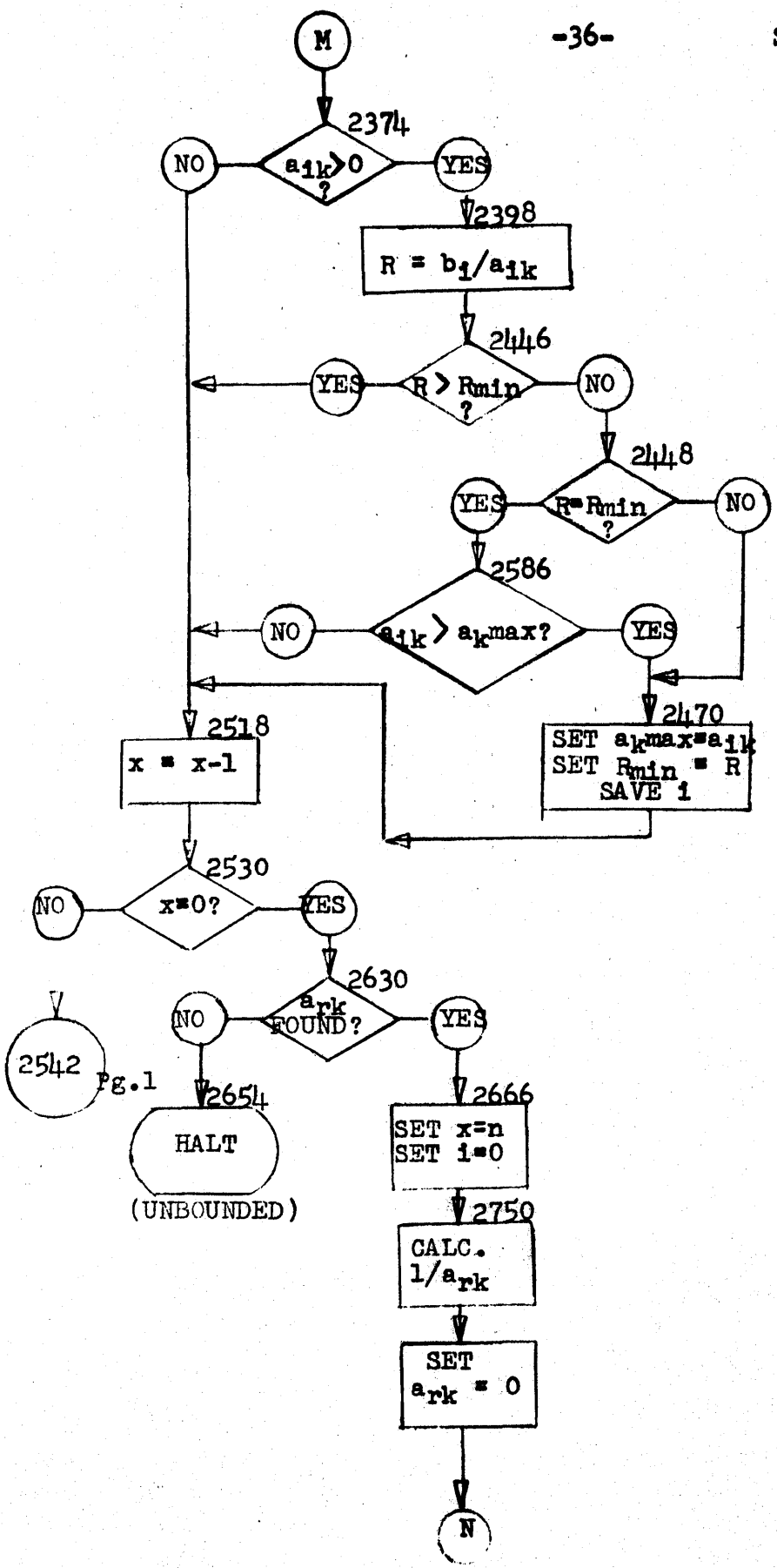


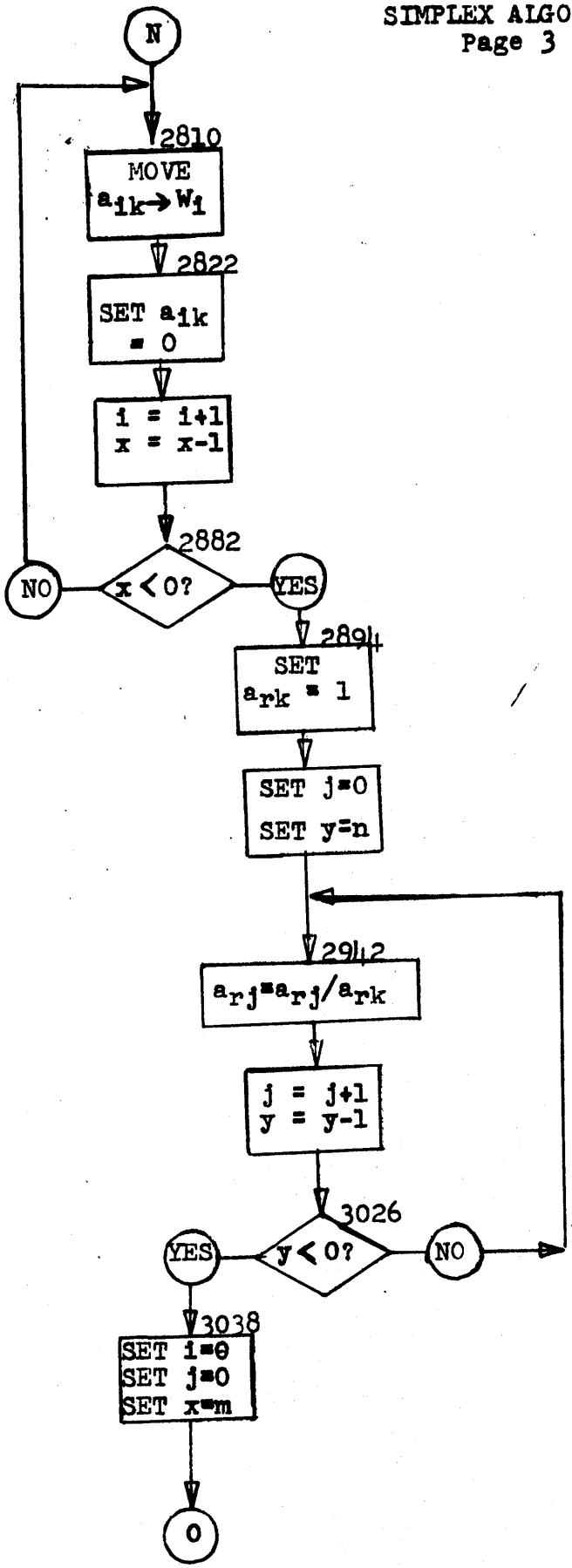

K
2690
MOVE
$a_{ik} \rightarrow W_i$
2702
SET
$a_{ik} = 0$
2714
i = i+1
x = x-1
2762
NO
x < 0?
YES
2774
SET
$a_{rk} = 1$
SET j=0
SET y=n
2822
$a_{rj}=a_{rj}/a_{rk}$
j = j+1
y = y-1
2906
YES
y < 0?
NO
2918
SET i=0
SET j=0
SET x=m
L

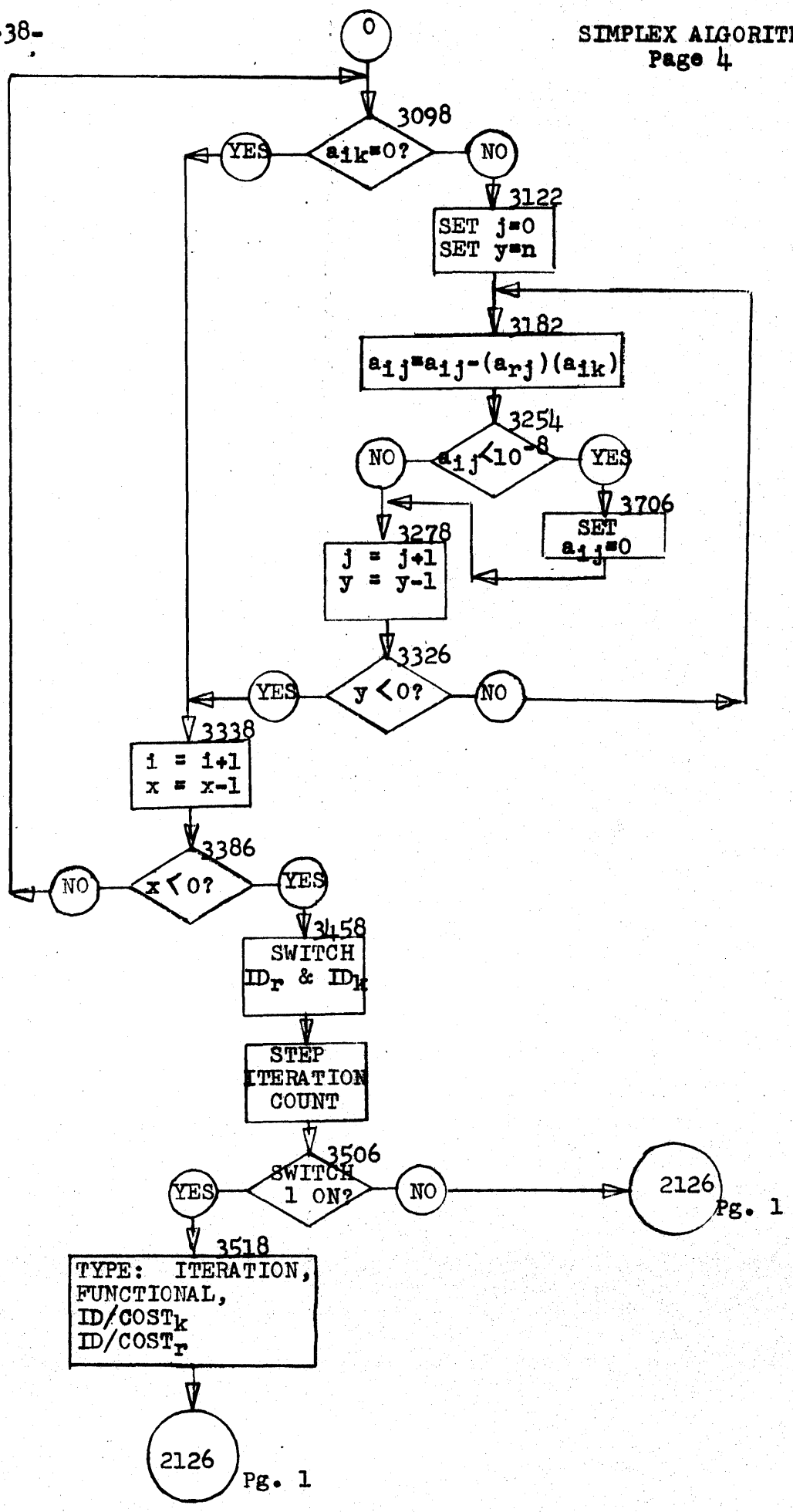

L
2978
$a_{ik}=0$?
YES
NO
3002
SET j=0
SET y=n
3062
$a_{ij}=a_{ij}-(a_{rj})(a_{ik})$
3134
$a_{ij}<10^{-8}$
NO
YES
3598
SET a_{ij}=0
3158
j = j+1
y = y-1
3206
y < 0?
YES
NO
3218
i = i-1
x = x-1
3266
x < 0?
NO
YES
3350
SWITCH ID_r & ID_k
STEP ITERATION COUNT
3398
SW 1 ON ?
YES
NO
2114 Pg. 1
3410
TYPE: ITERATION,
FUNCTIONAL,
$ID/COST_k$
$ID/COST_r$
2114 Pg. 1

SIMPLEX ALGORITHM
Page 1

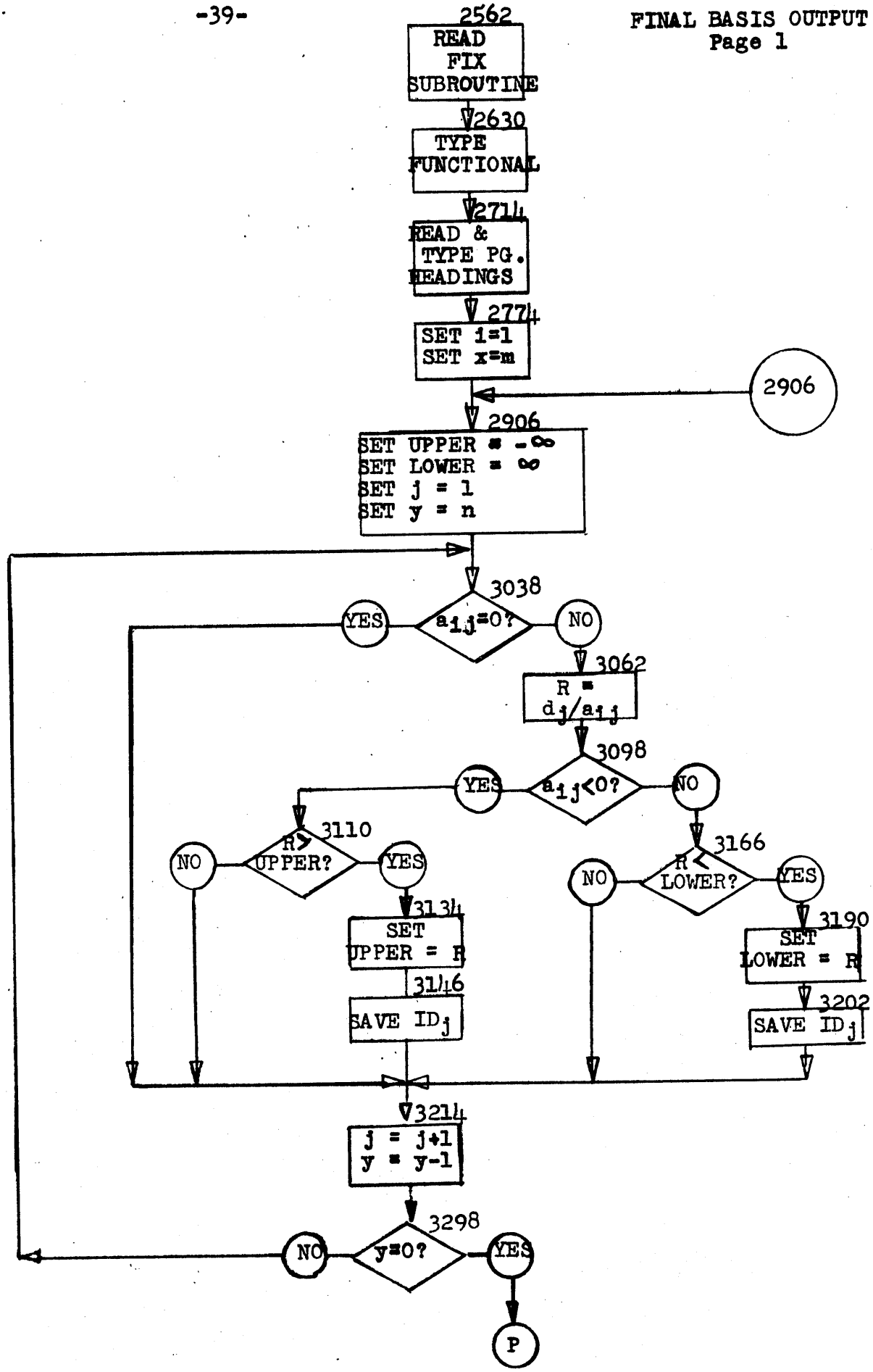



M
2374
NO
$a_{ik} > 0$?
YES
2398
$R = b_i/a_{ik}$
2446
YES
$R > R_{min}$?
NO
2448
YES
$R = R_{min}$?
NO
2586
NO
$a_{ik} > a_k max$?
YES
2518
x = x-1
2470
SET $a_k max = a_{ik}$
SET $R_{min} = R$
SAVE i
2530
NO
x=0?
YES
2630
NO
a_{rk} FOUND?
YES
2542
Pg.1
2654
HALT
(UNBOUNDED)
2666
SET x=n
SET i=0
2750
CALC. $1/a_{rk}$
SET $a_{rk} = 0$
N

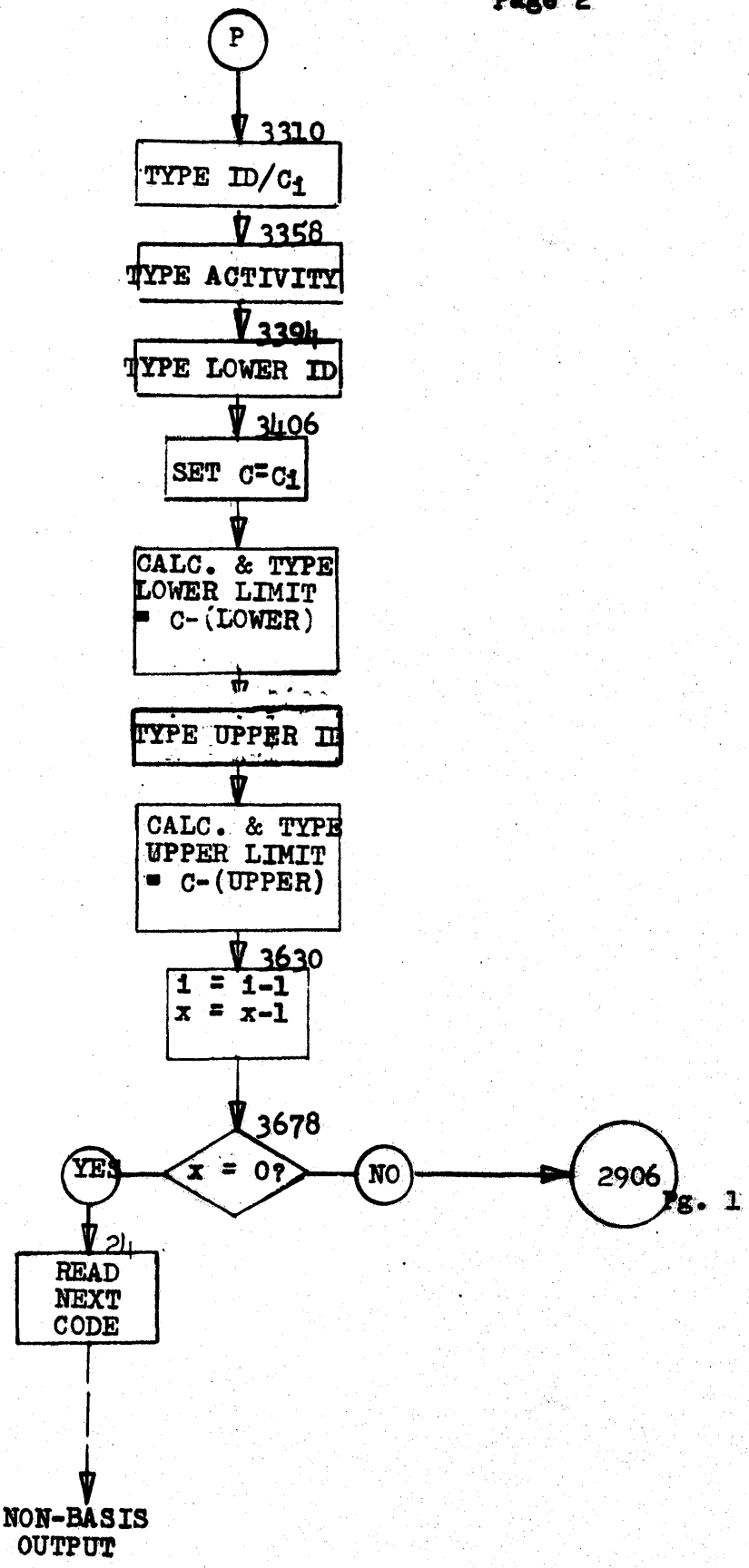

N
2810
MOVE
$a_{ik} \rightarrow W_i$
2822
SET a_{ik} = 0
i = i+1
x = x-1
2882
NO
x < 0?
YES
2894
SET
a_{rk} = 1
SET j=0
SET y=n
2942
$a_{rj}=a_{rj}/a_{rk}$
j = j+1
y = y-1
3026
YES
y < 0?
NO
3038
SET i=0
SET j=0
SET x=m
O


0
3098
YES
$a_{ik}=0$?
NO
3122
SET j=0
SET y=n
3182
$a_{ij}=a_{ij}-(a_{rj})(a_{ik})$
3254
NO
$a_{ij}<10^{-8}$
YES
3706
SET $a_{ij}=0$
3278
j = j+1
y = y-1
3326
YES
y < 0?
NO
3338
i = i+1
x = x-1
3386
NO
x < 0?
YES
3458
SWITCH ID_r & ID_k
STEP ITERATION COUNT
3506
YES
SWITCH 1 ON?
NO
2126
Pg. 1
3518
TYPE: ITERATION, FUNCTIONAL, $ID/COST_k$ $ID/COST_r$
2126
Pg. 1

FINAL BASIS OUTPUT
Page 1

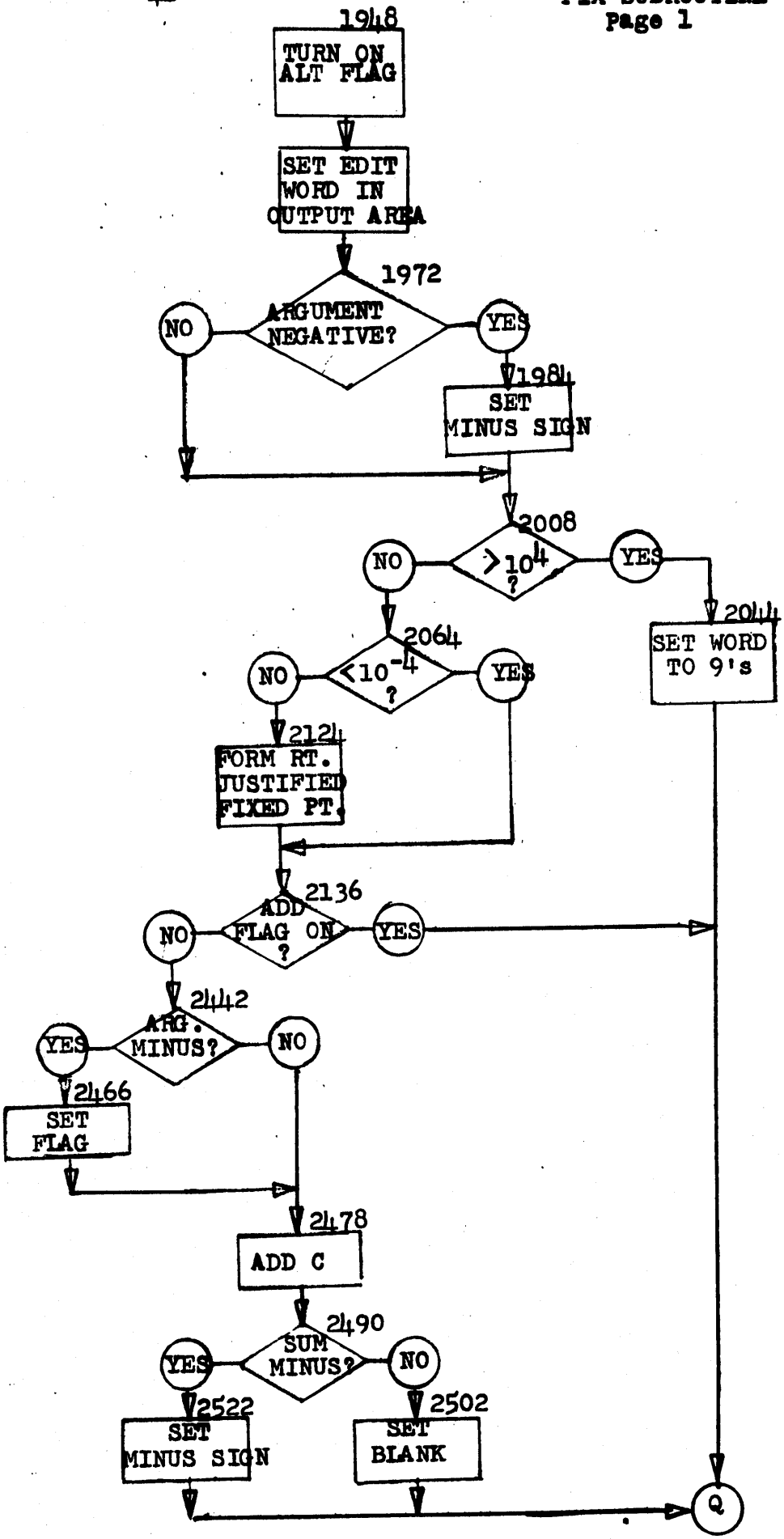


FINAL BASIS OUTPUT
Page 2

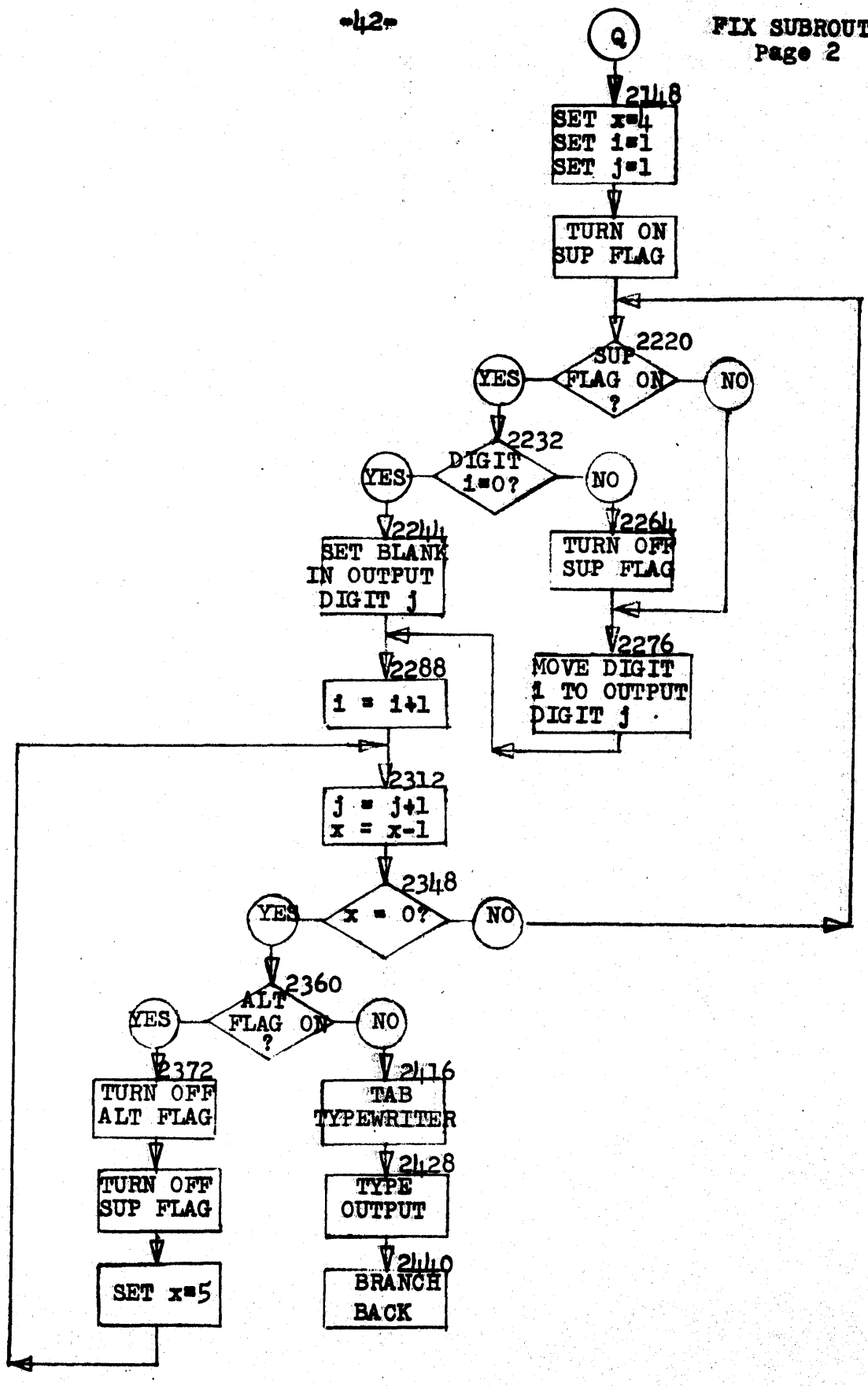


Fg. 1

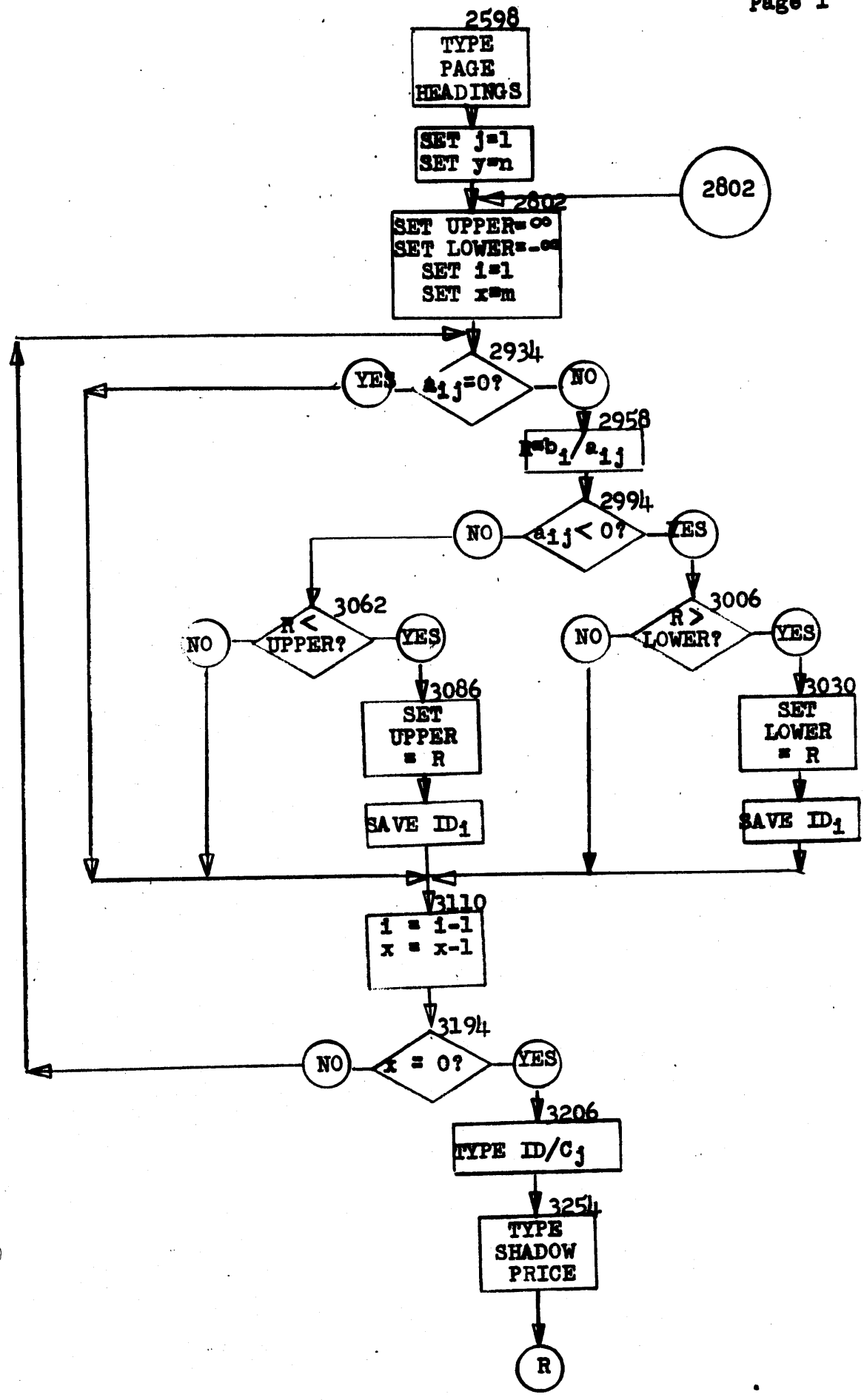
FIX SUBROUTINE
Page 1

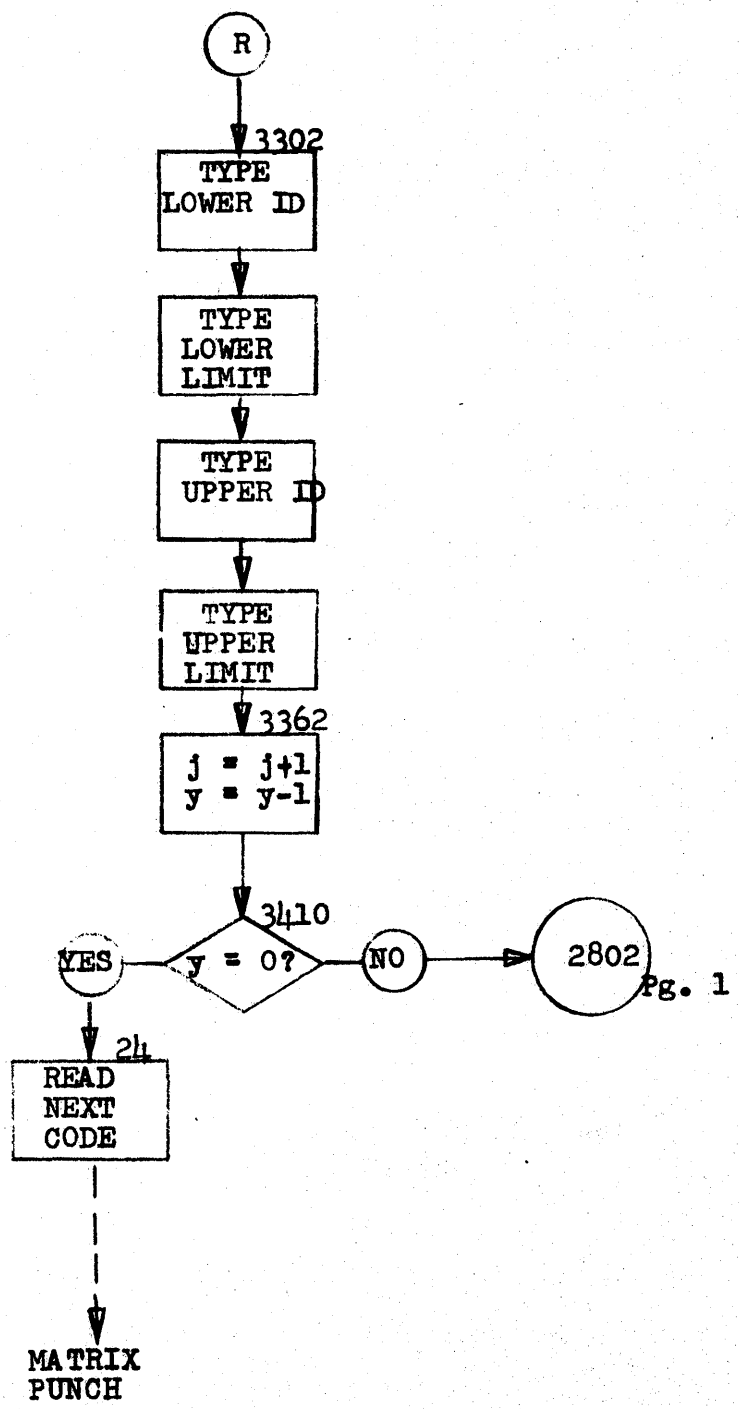


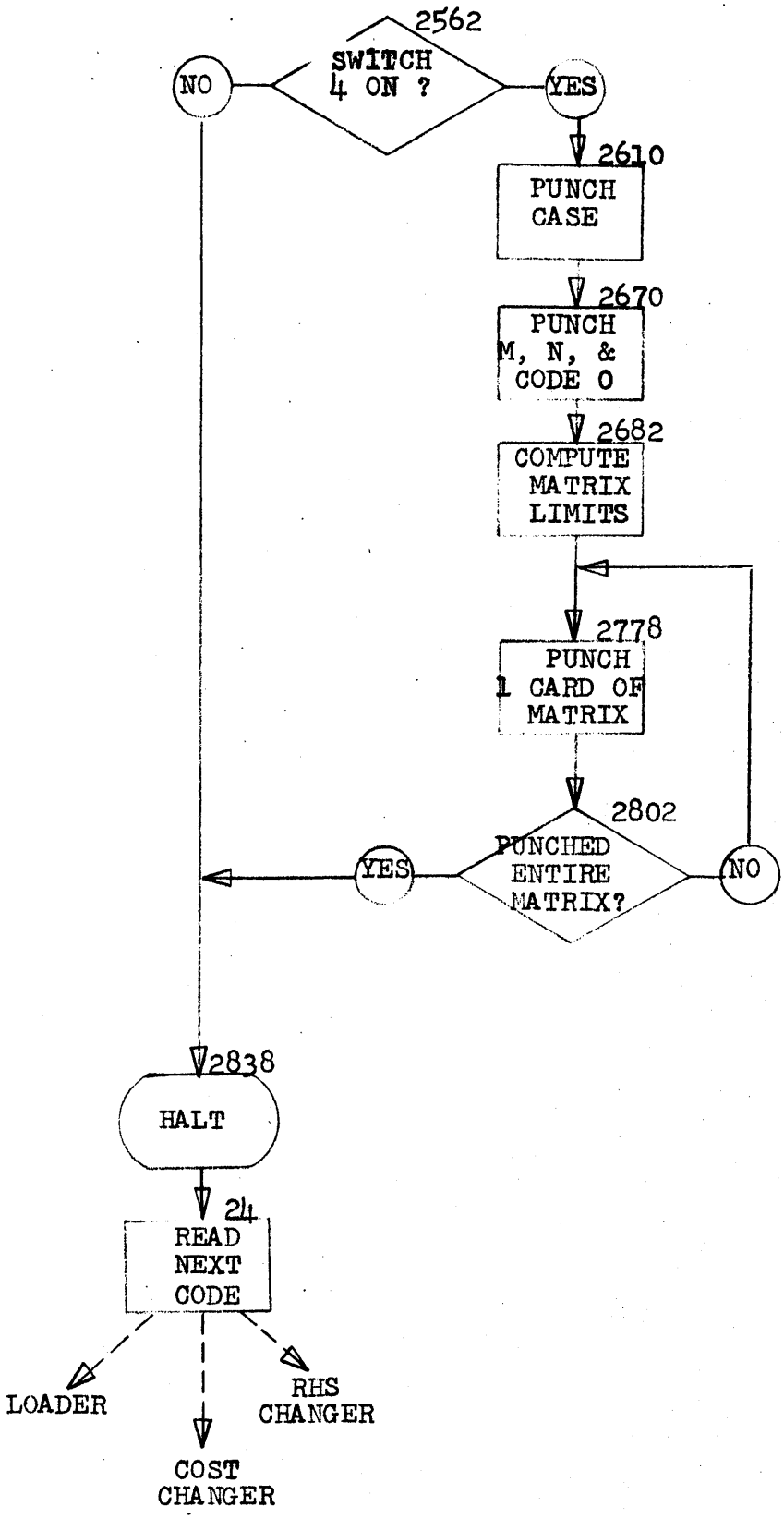
FIX SUBROUTINE
Page 2



NON-BASIS OUTPUT
Page 1




R
3302
TYPE LOWER ID
TYPE LOWER LIMIT
TYPE UPPER ID
TYPE UPPER LIMIT
3362
j = j+1
y = y-1
3410
y = 0?
YES
NO
2802
Pg. 1
24
READ NEXT CODE
MATRIX PUNCH


2562
SWITCH 4 ON ?
NO
YES
2610
PUNCH CASE
2670
PUNCH M, N, & CODE 0
2682
COMPUTE MATRIX LIMITS
2778
PUNCH 1 CARD OF MATRIX
2802
PUNCHED ENTIRE MATRIX?
YES
NO
2838
HALT
24
READ NEXT CODE
LOADER
RHS CHANGER
COST CHANGER

PRE-LOAD CARDS

ARITHMETIC TABLES

FLOATING POINT ROUTINES

```
360383800500
150391800002490382 6
3100012038874903826                    *                    4800000000003603826005004903826
3100100038584903826 0000000000001020304000204060800030609021004080216100500151 02
3100160038584903826 0060218142007041128200806142230090817263000000000050607080 90
3100220038584903826 0121416181518112427202428223635203530454036324844553249465 36
3100280038584903826 0484654627544536271801234567891234567890234567890J34567890JK
3100340038584903826 4567890JKL567890JKLM67890JKLMN7890JKLMNO890JKLMNOP90JKLMNOPQ
3100402038584903826 1600536005384900434016005360055826011970000026011120044516 01
3100462038584903826 1240000011011240000146004940140049011280046926011760000026 01
3100522038584903826 1870182749000000220118601176490057002601186011761500847000 01
3100582038584903826 1600857011872601092011892201178011894700702011002601176011 97
3100642038584903826 2601197011862601186011762101092011892601178011893201178000 00
3100702038584903826 1501198000001401178000047010940130044007740119732011980000 0
3100762038584903826 3301197000002100857011784400822011863301186000001500847000 02
3100822038584903826 1501178000001501189000002101198000000250109301198330119800000
3100882038584903826 4300986011894301034011904600966012003101190011912501198005 57
3100942038584903826 1201092000001460089401300260119701826490110602601198011973301
3101002038584903826 1900000011010920000146011260140015011890000011011980000543 00
3101062038584903826 9860118944010940109332011970000026011890109226000000119749 00
3101122038584903826 0000482601151004692600517000004900506000000000000000000000000 0
3101182039004903826                    *                                    * 000000000000000002
3101200038584903826 1600536012561501753000021601112000993201190000004900458043 01
3101260038584903826 2760119049009660430129001169482201189011682501325011763301 17
3101320038584903826 6000001501168000001601347000001101347000012101176011764301 39
3101380038584903826 4011684901350016014410183725014280116911014410 *002301175000
3101440038584903826 0026014770144112014770000426011660000002201166000933201162000
3101500038584903826 0023011760116526011860193722011860009632011780000023011860 11
3101560038584903826 8633000820000015000810000J21000900118626011860009023011860 11
3101620038584903826 6521000950009512013470000147016220120026011860009444016940 13
3101680038584903826 2532011860000023011860119711011890000N02501093000992601092011
3101740038584903826 8926011980009246017740130049009660470088201400490112600000 00
3101800038584903826 0000000000000000000000000000000J90692R080J74077P568J6012804 05J4
3101860038584903826 8249N490J38013M759J29099M164J21267L674J14332L266J08147K922J0
3101920038584903826 2597K630J000000000160053601970150175300003490122404301990011
3101980038584903826 9049009660430201001169490096603201169000001201168000N1230119
3102040038934903826                    *                    701176210118901168490171 8
```

```
3102114038584903826 480000000000370375900500250376900400310004803758360383800500
3102174038584903826 210219603841160006403950260007903842110384200OK0260006903842
3102234038584903826 260007403847430232603848210228800064360383800500310000103838
3102294038584903826 110228800OQ0470227000900490001203603643005001403645000004603
3102354038584903826 174012002303648000692100099000693300099000002100098036452100
3102414038584903826 098000633303646000003200095000002603160000994402490036482602
3102474038584903826 125036584903154044025100364549024700490341800000140375900OP0
3102534038584903826 460264601300140375900OK0460262601200430264603759330260500000
3102594038584903826 320376000000310375803760490265803202605000004902594033026050
3102654038584903826 000016021170000N01602712037591602700037603200000000001400000O
3102714038584903826 000346027740120011027120000211027000000211021170000149026940
3102774038584903826 260282102700260281602712120281600001310000000000430292603759
3102834038584903826 310375803760120211700001450282203759490330200000000000000000
3102894038584903826 000000000000000000000000000000000001602657000091602985037591603
3102954038584903826 000021181603005037594502994000004903062025000000000011029850
3103014038584903826 000211030000000111030050000212026570000149029740260310403000
3103074038584903826 120265700001460313001200150000000000110310400001490307404403
3103134038584903826 154026053202125000002600000021254903302026032520006412032520
3103194038584903826 00J923000690007332000950000021000990006921000990006932000000
3103254038704903826 000011032520000J012000990000J047032460120049000120
3103302038584903826 310375803798310379803838310383803878310387803918310364303663
3103362038584903826 450339403643160251603418490232601602516034544902338000003103
3103422038664903826 7580350031038380350037037590050032037580000049025220
3103500038584903826 ----------
3103560038974903826 *                                                    Z
2503723004004902126
```

FLOATING POINT ROUTINES

COST CHANGER ROUTINE

```
360383800500
150391800002490382 6
3100100038584903826 000000000000102030400020406080003060902100408021610050015102
3100160038584903826 006021814200704112820080614223009081726300000000005060708090
3100220038584903826 012141618151811242720242822363520353045403632484455324946536
3100280038584903826 048465462754453627180123456789123456789023456789OJ34567890JK
3100340038584903826 4567890JKL567890JKLM67890JKLMN7890JKLMNO890JKLMNOP90JKLMNOPQ
3100402038584903826 160053600538490043401600536005582601197000002601112004451601
3100462038584903826 124000001101124000014600494014004901128004692601176000002601
3100522038584903826 187018274900000022011860117649005700260118601176150084700001
3100582038584903826 160085701187260109201189220117801189470070201100260117601197
3100642038584903826 260119701186260118601176210109201189260117801189320117800000
3100702038584903826 150119800000140117800000470109401300440077401197320119800000
3100762038584903826 330119700000210085701178440082201186330118600000150084700002
3100822038584903826 150117800000150118900000210119800000025010930119833011980000O
3100882038584903826 430098601189430103401190460096601200310119001191250119800557
3100942038584903826 120109200001460089401300260119701826490110602601198011973301
3101002038584903826 190000001101092000014601126014001501189000001101198000054300
3101062038584903826 986011894401094010933201197000002601189010922600000011974000
3101122038584903826 000048260115100469260051700000490050600000000000000000000000
3101182039004903826 * * 000000000000000002
3101200038584903826 160053601256150175300002160111200099320119000000490045804301
3101260038584903826 276011904900966043012900116948220118901168250132501176330117
3101320038584903826 600000150116800000160134700000110134700001210117601176430139
3101380038584903826 401168490135001601441018372501428011691101441O *002301175000
3101440038584903826 002601477014411201477000042601166000002201166000933201162000
3101500038584903826 002301176011652601186019372201186000963201178000002301186011
3101560038584903826 863300082000001500081000OJ2100090011862601186000902301186011
3101620038584903826 652100095000951201347000014701622012002601186000944401694013
3101680038584903826 253201186000002301186011971101189000N02501093000992601092011
3101740038584903826 892601198000924601774013004900966047008820140049011260000000
3101800038584903826 000000000000000000000000000000J90692R080J74077P568J60128040SJ4
3101860038584903826 8249N490J38013M759J29099MI64J21267L674J14332L266J08147K922J0
3101920038584903826 2597K630J000000000160053601970150175300003490122404301990011
3101980038584903826 904900966043020100116949009660320116900000120116800ON1230119
3102040038934903826 * 701176210118901168490171 8
3102114038584903826 480000000000046021700020036038380050025038480040049021940340O
3102174038584903826 000001023603838001003203838000001403841000004600012012004602
3102234038584903826 266002003400000000102380383800100260220500079260230800064110 2
3102294038584903826 308000042400000038414602460012001102308000J0120220500OJ04702
3102354038584903826 302012002602205000742602404023082102404000692400000038414602
3102414038584903826 480012002102404000691202205000J04702398012004826025100230849
3102474038764903826 * 024920260251002404110251000006260000003847
3102516038744902126 *260254602510120254600005330000000004902126O
```

PRE-LOAD CARDS

ARITHMETIC TABLES

FLOATING POINT ROUTINES

```
360383800500
1503918000OZ4903826
3100100038584903826 000000000000102030400020406080003060902100408021610050015102
3100160038584903826 006021814200704112820080614223009081726300000000005060708090
3100220038584903826 012141618151811242720242822363520353045403632484455324946536
3100280038584903826 048465462754453627180123456789123456789023456789OJ34567890JK
3100340038584903826 4567890JKL567890JKLM67890JKLMN7890JKLMNO890JKLMNOP90JKLMNOPQ
3100402038584903826 160053600538490043401600536005582601197000002601112004451601
3100462038584903826 124000001101124000014600494014004901128004692601176000002601
3100522038584903826 187018274900000022011860117649005700260118601176150084700001
3100582038584903826 160085701187260109201189220117801189470070201100260117601197
3100642038584903826 260119701186260118601176210109201189260117801189320117800000
3100702038584903826 150119800000140117800000Q4701094013004400774011973201198000000
3100762038584903826 330119700000210085701178440082201186330118600000150084700002
3100822038584903826 150117800000150118900000210119800000250109301198330119800000
3100882038584903826 430098601189430103401190460096601200310119001191250119800557
3100942038584903826 120109200001460089401300260119701826490110602601198011973301
3101002038584903826 190000001101092000014601126014001501189000001101198000054300
3101062038584903826 986011894401094010933201197000002601189010922600000011974900
3101122038584903826 000048260115100469260051700000490050600000000000000000000000000
3101182039004903826                    *                              *  000000000000000000Z
3101200038584903826 160053601256150175300002160111200099320119000000490045804301
3101260038584903826 276011904900966043012900116948220118901168250132501176330117
3101320038584903826 600000150116800000160134700000011013470000121011760117643013 9
3101380038584903826 401168490135001601441018372501428011691101 4410 *002301175000
3101440038584903826 002601477014411201477000042601166000002201166000933201162000
3101500038584903826 002301176011652601186019372201186000963201178000002301186011
3101560038584903826 863300082000001500081000OJ21000900118626011860009023011860 11
3101620038584903826 652100095000951201347000014701622012002601186000944401694013
3101680038584903826 253201186000002301186011971101189000N025010930009926010920 11
3101740038584903826 892601198000924601774013004900966047008820140049011260000000
3101800038584903826 000000000000000000000000000000J90692R080J74077P568J6012804 05J4
3101860038584903826 8249N490J38013M759J29099M164J21267L674J14332L266J08147K922J0
3101920038584903826 2597K630J000000000160053601970150175300003490122404301990011
3101980038584903826 904900966043020100116949009660320116900000120116800N1230119
3102040038934903826                    *                          701176210118901168490171 8
```

REQUIREMENT CHANGE ROUTINE

```
3102114038584903826 480000000000340000000102460218200300360340000500250340400400
3102174038584903826 490219403603400001003203400000000140340300000460001201200 4702
3102234038584903826 27400300340000000108370375900100490233403803400001003703759 0
3102294038584903826 05004100000000003400000001083903759001003203758000001403759 0
3102354038584903826 00P0460247001300140375900 0K0460245001200430247003759330248 10
3102414038584903826 0000320376000000310375803760490248203202481000004902418033 02
3102474038584903826 48100000160211700 0N0160253603759160252403760320376000000140 3
3102534038584903826 75900003460259801200110253600002110252400002110211700001490 2
3102594038584903826 51802602645025242602640025361202640000013103758037604302702 0
3102654038584903826 37593103758037601202117000014502646037594902126016022050000 9
3102714038584903826 16027610375916027760211816027810375945027700375949028380250 2
3102774038584903826 11803759110276100002110277600001110278100002120220500001490 2
3102834038584903826 75002602880027761202205000014602906012001502118000001102880 0
3102894038584903826 00014902850044029300248132021250000026022050007926029720006 4
3102954038584903826 11029720000424000000340346031120120011029720 0J012022050 00J0
3103014038584903826 47029660120026022050007426030680297221030680006924000000340 3
3103074038584903826 46031920120012022050 00J0470305001200482603171029722103171000
3103134038584903826 69110317100006160046903183160044500000490042202125490212602 6
3103194038584903826 03275030681103275000 1626033110006421033110006926022050007916
3103254038584903826 00469032872601197000000490193802125160046903323160044500000049
3103314038584903826 00422000991103275000J0110331100 0J012022050 00J0460325201300 49
3103374039124902126                                                          021260
```

PRE-LOAD CARDS
SHADOW PRICE CALCULATION ROUTINE

```
360383800500
1503918000Z4903826
3102114038584903826 160247203940260217300064260225300079110217300OJ026039180000O
3102174038584903826 320391300000220391801822470223001200260392001826490246601603
3102234038584903826 912000N3330391300000160233703913160242901818160243603919250S
3102294038584903826 910039183303918000002503919004004302394000001102337000011203
3102354038584903826 912000011102429000011202436000014902326026024170233731039130
3102414038584903826 000026039200181833039190000044024660391032039200000026000000
3102474038584903826 392011024720000J012022530000J047021500120026024720006421024720
3102534038584903826 006921024720006911021730000J026021610006926024890006926022530
3102594038584903826 007416024480264816024550000M126025200264749021500M902668320392
3102654038584903826 000004902466026026590007426027100006421027100006926000000018
3102714038584903826 262602871027102602847027101102847000J016027990394026022530000
3102774038584903826 792603920018262103920000004602884012001600469028472601197039
3102834038584903826 204901938000001600469028831600445000004900422000991102799000
3102894038584903826 J01102847000J01202253000J047027760120011028470000K02102871000
3102954038584903826 691202659000J046027520130037035630050025035830040034000000001
3103014038584903826 023903563001003900049001003400000001023400000001021602659000
3103074038584903826 043703563005002503583004004703136001003903563001003400000001
3103134038734902114 08120265900001470307601200340000000102490002 4
CASE
ITER NO
FUNCTIONAL
  VAR OUT
  VAR IN
```

PAGE HEADINGS

PRE-LOAD CARDS.
DUAL ALGORITHM ROUTINE

```
360383800500
1503918000Z4903826
3102114038584903826 260002301826260344000079260218500064210218500069110218500OJ0
3102174038584903826 240002300000470222201102602221021852600023000001203440000J0
3102234038584903826 470216201200440002400023260002302205160356000000260341600074
3102294038584903826 260237202221260241300064210241300069210237200069260242502372
3102354038584903826 210241300069240000001826460248601300160046902425260119700000
3102414038584903826 490120000000240009900023470248601100260002300099260352402372
3102474038584903826 260356002413120341600OJ047023300120014035600000047025460120O
3102534038584903826 48N1100000002602684035242602708035602603440000792602665035 24
3102594038584903826 160269603930260278003524260270103560160046902665260119702545
3102654038584903826 490120000000260374000099260000000182626039300000026000000182 6
3102714038584903826 110269600OJ011027010OOJ011027080OOJ012034400OJ046026900130O
3102774038584903826 260000002545260285702221260286402221260341600074160046902857
3102834038584903826 260119703740490193800000260000000099210285700069210286400069
3102894038584903826 120341600OJ046028220130016029840393026030250006421030250006 9
```

PRE-LOAD CARDS

SIMPLEX ALGORITHM ROUTINE

```
3102954038584903826 160309703930260344000079220393001826460321801200260308502221
3103014038584903826 160312100000260341600074260314003121120314000008160046903097
3103074038584903826 260119700000490193803930160046903133160044500000490040200099
3103134038584903826 1400000000M3470359801300210308500069210314000069210312100069
3103194038584903826 1203416000J 460306201300110298400OJ0110309700OJ0110302500OJ0
3103254038584903826 1203440000J 460297801300260337302221260336101356012033610000J0
3103314038584903826 260336803361220337300069260338003373260002300000260000000000
3103374038584903826 260000000023110363100001470211400100340000000102380362900100
3103434038584903826 340000000108260357703373260354103361260350500064210350500069
3103494038584903826 260391700000380390800100340000000108260391700000380390800100
3103554038584903826 340000000108260391700000380390800100490211402603616031212600
3103614039004902114                    *                              *  000018264903158000
360383800500
150391800002490382 6
3102114038584903826 250366803747260002301826260354800074260219700064210219700069
3102174038584903826 210219700069240002300000470223401100260223302197260002300000
3102234038584903826 210219700006912035480000J0470218601200440002400023260002303522
3102294038584903826 160348100000260363200079260238002233260243302233260242100064
3102354038584903826 210242100069490254202400000001826470251801100160046902433260 1
3102414038584903826 197000004901200000002400099000234602518011004602586012002602
3102474038584903826 493023802603746000002600023000992603481024211203632000J04602
3102534038584903826 630012001102380000J0110242100OJ0110243300OJ0490237402602609O
3102594038584903826 238024037460000047024700130049025180140348100000470266601200
3102654038584903826 48N1100000002602804024932602828022332603632000792602785024 93
3102714038584903826 160281603930260290002493260282102233160046902785260119702665
3102774038584903826 490120000000260392000099260000001826260393000000260000001826
3102834038584903826 110281600OJ0110282100OJ0110282800OJ0120363200OJ0460281001300
3102894038584903826 260000002665260297703481260298403481260354800074160046902977
3102954038584903826 260119703920490193800000260000000009921029770006921029840006 9
3103014038584903826 120354800OJ0460294201300160310403930260314500064210314500069
3103074038584903826 160321703930260363200079220393001324603338012002603205034 81
3103134038584903826 160324100000260354800074260326003241120326000008160046903217
3103194038584903826 260119700000490193803930160046903253160044500000490040200099
3103254038584903826 1400000000M3470370601300210320500069210326000069210324100069
3103314038584903826 120354800OJ0460318201300110310400OJ0110321700OJ0110314500OJ0
3103374038584903826 120363200OJ0460309801300260346902233120346900OJ0260347603469
3103434038584903826 220348100069260348803481260002300000260000000000260000000023
3103494038584903826 110366700001470212600100340000000102380366500100340000000108
3103554038584903826 260368503481260364903469260361300064210361300069260374600000
3103614038584903826 380373700100340000000108260374600000380373700100340000000108
3103674038584903826 260374600000038037370010049021260260372403241260000001826490 3
3103734039044902114                    *                              *  27800000000002
```

SOLUTION

PRE-LOAD CARDS

```
360383800500
1503918000Z4903826
3102562038584903826 360382600500490382603400100001022602653000642102653000693703
3102622038584903826 759005003903759001002603917000003400010001081803908001003400
3102682038584903826 000001023400000001021603533000063703759005003903759001001203
3102742038584903826 533000014702714012003400000001022603321000641103321000J02603
3102802038584903826 369026531103369000J026030130336921030130007926030010301311103
3102862038584903826 001000J026029650301311029650O0K026035450007926007500266626O0
3102922038584903826 740025931600725000001600715000001603044000002603097029652603
3102982038584903826 109029651603085000001603157000002603213030132603533000742200
3103042038584903826 000037064603214012001600469030972601197000004901200000004403
3103102038584903826 166000002400099007504703214011002600750000992600730000004903
3103162038584903826 214024000990074046032140130026007400009926007200000021030440
3103222038584903826 006921030850006921030970006921031090006921031570006921032130
3103282038584903826 006912035330O0J047030380120026039180000034000000010238039090
3103342038584903826 010032019550000027019480000026O-9180071434000000010838039150
3103402038584903826 010026034290332126007600000003 C07550000044034860076033007600
3103462038584903826 000032035210000049034980330352100000260076900760150077000000
3103522038584903826 330195500000320074000000027019480074026039180072434000000010B
3103582038584903826 380391500100330075000000330195500000027019480075011029650O0J0
3103642038584903826 110332100OJ011033690O0J012035450O0J047029060120049000240000O
3103702038904902562 * 000007070707003707070700OZ0
```

PRE-LOAD CARDS

```
360383800500
1503918000Z4903826
3101948038584903826 320195600000310389803707440200801947330194700000160391700OK0
3102008038584903826 260076003706140193900ON4470206401100160075TR999949021480120l
3102068038584903826 939000M747021360130016021350194021021350193915019390000021O0
3102128038584903826 760019474402442019551602219000041602287007531602282038991602
3102188038584903826 243007531602250038993201957000004402276019574302264007531603
3102248038584903826 759000004902288033019570000025038990075311022870000111022430
3102308038584903826 000111022820000211022500000212022190000147022200120044024160
3102368038584903826 195633019560000033019570000016022190000549023120340000000108
3102428038584903826 390389900100421403917000K047024780120032007600000021007600O7
3102488038584903826 704702522013001603917000004902148016039170O0K049021487070707
3102548039044903826 * * 00370707070000
2503919004004902582
```

PAGE HEADINGS

```
FUNCTIONALZ
 VAR/COST    Z
ACTIVITY  Z
LIM VAR      Z
LOWER LIM  Z
LIM VAR   Z
UPPER LIM     Z
```

SOLUTION

NON-BASIS OUTPUT ROUTINE

PAGE HEADINGS

PRE-LOAD CARDS

MATRIX PUNCH ROUTINE

```
360383800500
1503918000Z4903826
3102562038584903826 340001000102L400100001021603253000063703759005003903759001 00
3102622038584903826 120325300001470259801200340000000102260321700064210321700069
3102682038584903826 210321700069260326503217120321700OJ026028610321711028610000K0
3102742038584903826 260289702861220289700069260290902897220290900069260322400074
3102802038584903826 260075002585260074002574160071500000160072500000160294000000
3102862038584903826 260299302861260300502861160298100000160305300000260310902909
3102922038584903826 260325300079220000003438460311001200160046902993260119700000
3102982038584903826 490120000000440306200000240009900740470311001100260074000099
3103042038584903826 260072000000490311002400099007504603110013002600750000992600
3103102038584903826 730000001102940000J01102981000J01102993000J01103005000J01103
3103162038584903826 053000J01103109000J01203253000J04702934012002603918000003400
3103222038584903826 000001023803909001003201955000002701948000002603918007143400
3103282038584903826 000001083803915001002701948007402603918007243400000001083803
3103342038584903826 915001002701948007502102861000692103217000692103265000691203
3103402038814902562                    *       224000J04702802012004900024000000000
 VAR/COST   Z
SHAD PRICE Z
LIM VAR      Z
LOWER LIM  Z
LIM VAR   Z
UPPER LIM      Z
360383800500
1503918000Z4903826
3102562038584903826 470282600400310356303401310356200048150357300000390356300400
3102622038584903826 310356303481260356700079260357200074150357300000380356300400
3102682038584903826 260278400064110278400001230006900073320009500000210009900069
3102742038584903826 210009800078210009902784260281300099380000000400110278400000
3102802038584903826 140278400000470277801300410000000J00488888888888490002400002
3103401038584903826 ----------------------------------------------------------------
3103461038584903826 ----------------------------------------------------------------
3103521038784903826                    *       ----------------------------------------
2503561004002503723004004902562
```

COMPUTER
TECHNOLOGY

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

Acknowledgement

This program is a modification of one for the basic 1620 with paper tape input and output written by C. R. Nichols, IBM, Inglewood, California. The only changes made were those necessary to enable his program to be used with a card 1620. Much of the information in the writeup is taken directly from his writeup.

Linear Programming Code for the IBM 1620 with Card Input and Output

C. R. Nichols
IBM Corporation
Inglewood, California

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.

TABLE OF CONTENTS

I. PROGRAM DESCRIPTION

This program solves linear programming problems with output of detailed results. That is, given coefficients a_{ij}, cost coefficients c_j, and requirements b_i, determine x_j such that

$$\sum_j a_{ij}x_j = b_i \qquad \text{with } x_j \geq 0$$

and

$$\sum_j c_j x_j = \text{maximum}$$

Computations are performed by the Dual Algorithm until a feasible solution is obtained. Control is then given to the Simplex Algorithm for optimization. Cost changes and requirement changes can be made after loading original matrix or after solving original matrix.

A. Machine requirements

1. Basic 1620 with card input and output.
2. Any size storage can be used. The larger the storage, the larger the problem that can be solved. The size of the problem is restricted by the following relationship:

$$(m + 2)(n + 3) \leq \frac{\text{memory} - 3920}{10}$$

where: m is the number of restrictions
n is the number of non-basis independent variables
memory is 20,000, 40,000, or 60,000

3. Source language: actual machine language
4. Subroutines: floating point subroutines supplied with program. No other subroutines needed.
5. Input media: card reader and console typewriter.
6. Output media: card punch and console typewriter.
7. Running time: The precise time required per iteration depends on the size and density of the matrix. As an approximation, a problem with 30 equations and 40 non-basis variables requires about 20 seconds per iteration.
8. Decimal accuracy: All computations are performed in 2-and-8 floating point form. Matrix input can be either fixed point or floating point.

B. Error stops

1. All stops may be recognized by displaying IR-1 in MAR. MAR will contain the address of the halt instruction plus 12.

Halt instr.	Reason
01126	Floating point overflow
01288	Floating point attempted division by zero
00012	Loader - ready for next code
02458	Cost changer - Change entered for a non-existent identification number
00012	Cost changer - ready for next code
03110	RHS changer - change entered for a non-existent identification number
00012	RHS changer - ready for next code
02534	Dual algorithm - inconsistent matrix
02654	Simplex algorithm - unbounded solution
02838	Matrix punch - ready for next code

2. Error procedure

a. Cost changer (02458). Erroneous identification number is the last one typed.

If data are being entered through the typewriter, depress RESET and INSERT.
Type 49 02126.
Depress RELEASE and START.
Corrected or new change data may now be entered normally.

If data are being entered from cards, the erroneous datum may be ignored by following the same steps as above (for entry through typewriter).
If it is desired to enter the change correctly through the typewriter, the following steps must be taken:
Set console switch two on.
Depress RESET and INSERT.
Type 49 02126.
Depress RELEASE and START.
Enter correct datum through typewriter (complete with record mark).
Depress RELEASE and START.
When typewriter is again selected, depress RESET.
Set console switch two off.

Depress INSERT, RELEASE, and START.
Processing will now continue normally, reading data from cards.

b. RHS Changer (03110): Erroneous identification number is the last one typed.

If data are being entered through the typewriter, depress RESET and INSERT.
Type 49 02126.
Depress RELEASE and START.
Corrected or new change data may now be entered normally.

If data are being entered from cards, the erroneous data may be ignored by following the same steps as above (for entry through typewriter).
If it is desired to enter the change correctly through the typewriter, the following steps must be taken:
Set console switch three on.
Depress RESET and INSERT.
Type 49 02126.
Depress RELEASE and START.
Enter correct identification number (4 digits plus record mark) through typewriter.
Depress RELEASE and START.
Enter requirement change (including record mark) through typewriter.
Depress RELEASE and START.
When typewriter is again selected, depress RESET.
Set console switch three off.
Depress INSERT, RELEASE, and START.
Processing will now continue normally, reading data from cards.

c. Dual algorithm (02534): To obtain solution existing at the time of discovery of inconsistency:
Run out cards from reader.
Replace cards in reader hopper starting with Final Basis Output program.
Depress READER START.
Depress RESET and INSERT.
Type 49 00024.
Depress RELEASE and START.

d. Simplex Algorithm (02654): To obtain solution existing at the time of discovery of unbounded solution:

Depress RESET and INSERT.
Type 49 00024.
Depress RELEASE and START.

e. Overflow-underflow: Overflow causes a stop at 01126 or 01288. Underflow causes the result to be set to zero, and processing continues. Either condition indicates either machine malfunction or unreasonable data.

f. Any other error: unrecoverable. Restart problem.

II. THE PROGRAM

A. Operation Instructions

1. Set typewriter margins at 12 and 92; tabs at 24, 36, 48, 60, and 69.

2. It is important to note that no more than one instruction at a time may be inserted at any time after the initiation of the Loader routine.

3. To load initial data:

   a. Ready the LOADER deck in the card reader (including matrix input data). Depress READER START.

   b. Depress RESET and INSERT.

   c. Type 16 00010 00000.

   d. Depress RELEASE and START.
   This initiates memory clear routine.

   e. After at least one second, depress INSTANT STOP, RESET, and INSERT.

   f. Type 36 03826 00500
   49 03826

   g. Depress RELEASE and START.

   h. Program deck loads, data are loaded, and a halt at location 00012 is executed. Operation may now proceed to the COST CHANGER, RHS CHANGER, or SOLUTION programs.

4. To effect cost changes:

   a. Ready the COST CHANGER deck in the card reader (including data change cards if changes will be made from cards). Depress READER START.

   b. Set console switch two off if cost changes will be entered from card reader; switch two on if data will be entered through the typewriter.

   c. Depress START.

   d. If data are entered through the typewriter, RELEASE and START must be depressed after each entry. (Don't forget the record mark to terminate each entry.)

   e. After all data are entered a halt at location 00012 is executed. Operation may now proceed to the RHS CHANGER or SOLUTION programs.

5. To effect requirement changes:

   a. Ready the RHS CHANGER deck in the card reader (including data change cards if changes will be made from cards). Depress READER START.

   b. Set console switch three off if requirement changes will be entered from card reader; switch three on if data will be entered through the typewriter.

   c. Depress START.

   d. If data are entered through the typewriter, RELEASE and START must be depressed after the entry of each identification and each change. (Don't forget the record mark to terminate each entry.)

   e. After all data are entered a halt at location 00012 is executed. Operation may now proceed to the COST CHANGER or SOLUTION programs.

6. To solve and obtain output:

   a. Ready SOLVER deck in the card reader. Depress READER START.

   b. Set console switch one on to type iterations, off to omit iteration typeout.

c. Set console switch four on to obtain punch-out of final matrix, off to suppress punching.

d. Depress START.

e. After all output is complete, a halt at location 02838 will be executed. Operation may now proceed to the LOADER, COST CHANGER, or RHS CHANGER programs.

B. Data Preparation

1. The first card to be entered is a case identification card. A five digit alphabetic identification is punched in columns 1 through 5. Columns 6 through 80 are blank.

2. The second card specifies the size of the problem being entered. It is punched as follows:

Information	Card columns
$\overline{0}$ (zero with a minus sign)	1
number of equations (3 digits)	2 - 4
0 (zero)	5
$\overline{0}$ (zero with a minus sign)	6
number of non-basis variables (3 dig.)	7 - 9
0 (zero)	10
input code	11

a. 1 for row-column, fixed point input
b. 0 for a complete floating point matrix

Columns 12 through 80 are blank.

3. The matrix can be entered in one of two forms: floating point (the entire matrix is entered, column by column, exactly as it is to be stored, including zero entries) or fixed point (entries can be in any order, accompanied by their respective row-column designations, and zero elements need not be entered).

4. For floating point entry, eight elements are punched on each card, with a minus sign over the first position of each ten position field, and a minus sign over the last position of negative fields (units position). There are ten positions per field, with the first two positions specifying placement of the decimal point (50 designates a decimal point to the left of the

first non-zero digit; 51 designates one non-zero digit to the left of the decimal; 49 designates that the first non-zero digit is in the hundredths position, etc.) and the last eight digits (with no leading zeroes) giving the element itself. A zero element is specified by ten zeroes, with a minus sign over the first.

The elements must be punched in column order, starting with the first basis cost. See the matrix layout included with this writeup to know how the elements are stored in the matrix.

5. Fixed point matrix elements are entered four to a card.

Two types of cards are necessary for matrix elements:

a. Row-column cards designate where the elements will be stored in the matrix. Row-column designations are punched in columns 1-6, 21-26, 41-46, and 61-66 in the form $\overline{R}RR\overline{C}CC$ (a minus sign accompanies the first digit of each row and column designation). The column designation specifies column number considering only non-basis vectors. The requirement variables have only a row designation, and 000 should be entered for their column.

b. Matrix element cards contain fixed-point entries including optional leading sign, up to eight numeric digits, and mandatory decimal point. Each entry must be terminated by a record mark (0-2-8 punch). The four elements are punched beginning in columns 1, 21, 41, and 61 for each card.

Elements may be entered in any order, but corresponding row-column designations and matrix entries must be in the same positions of successive cards (row-column card preceding).

Loading of zero elements is optional

If any card contains fewer than four elements, the first blank field (column 21, 41, or 61) of the row-column card must be punched with a record mark (0-2-8 punch).

Item	Description	Row-column
a_{ij}	matrix element of the ith row and jth non-basis vector	i j xxxxxx
b_i	element in the ith row of the requirement vector	i xxx000

6. Fixed point cost entries are also four per card, but the row-column designation is on the same card as the cost.

   Entries consist of a six digit row-column designation followed by a ten digit ID/cost, where the ID is a four digit numerical designation to identify the cost variable, and cost is a six digit fixed point entry with the decimal after the third digit.

Item	Description	Row-column
c_i	cost per unit of basis variable for row i	i ID cost xxx00Ixxxxxxxxx
c_j	cost per unit of jth non-basis vector	j ID cost 00Ixxxxxxxxxxxx

   Negative cost is signified by a minus sign over the units position of the cost field.

   If any card contains fewer than four elements, the first blank field (column 21, 41, or 61) of that card must be punched with a record mark (0-2-8 punch).

7. The last data entry for fixed point mode must be 000 to terminate the loading operation. This can be punched in columns 1-3 of a new card, or in 21-23, 41-43, or 61-63 or the last row-column or cost entry card.

8. Entry of cost changes:

   Data for the COST CHANGER routine can be entered either from cards or from the typewriter. If entered from cards, one entry is punched in each card, in columns 1-10. The entry consists of the four digit ID and six digit cost figure, with a minus sign over the first digit of the cost (column 5) and over the last digit of negative costs (column 10). Following the last cost change card must be a card punched with four zeroes (columns 1-4) to terminate the operation.

   If entered from the typewriter, the format is the same as from cards: four digit ID and six digit cost figure with flag over the first digit of all cost figures, and over the units position of negative costs. The last entry must be four zeroes to terminate the operation.

9. Entry of requirement changes:

   There are two entries for each requirement change: the identification and the change in the value of that requirement element. Note that the new requirement element is not entered, but rather the difference between the old and new requirement elements. If changes are entered from cards, the identification and change are entered on separate cards, one per card.

   The identification is entered first, and consists of the ID associated with the cost element for that row. Thus, to change $b_i = b_3$, use the ID associated with $c_i = c_3$. No flags or record marks are needed in entering the ID either from cards or typewriter. It is punched in columns 1-4 of card entries, with 5-80 blank.

   After entering the ID, its associated requirement change is entered (by typewriter or on a separate card). The entry starts in column 1, if a card is being used, and consists of optional leading sign, up to eight numeric digits, and mandatory decimal point. It must be terminated by a record mark, whether it is entered from cards or from the typewriter.

   The last entry must be four zeroes to terminate the operation.

4. Interpretation of Results

   1. Cost changes or requirement changes are typed if the input is from cards. If input is through the typewriter, the input operation itself produces a log of the changes made.

   2. Dual Algorithm and Simplex Algorithm:

      Results of iterations are typed only if console switch one is on. If it is desired to monitor the course of the solution onlh occasionally, this may be done by setting switch one on and off periodically. The following points should be borne in mind:

a. Page headings for iteration output are typed only if switch one is on at the beginning of the solution.
b. The iteration count is reset when the dual algorithm is finished and control passes to the simplex algorithm.

The information which may be typed includes the iteration count, the current value of the profit function (floating point), the last variable to leave the basis, and the variable which entered the basis.

3. Final Basis Output:

The final value of the profit function is given in floating point.

For each final basis variable, the following information is supplied (in fixed point):

a. Identification/cost coefficient.

b. Activity.

c. The limits of the cost coefficient over which the current solution is optimal.

d. The variables which limit the range of the cost coefficient and will enter the basis if a limit is exceeded.

4. Final Non-Basis Output:

For each non-basis variable, the following information is given (in fixed point):

a. Identification/cost coefficient.

b. Shadow price - the penalty to the total system if a unit of this variable is forced into the final solution.

c. The limits of activity over which the shadow price applies.

d. The variables which limit the range of applicability of the shadow price. The upper limiting variable is the one which would leave the basis if the associated non-basis variable were forced into the solution.

5. A punchout of the complete final matrix is optional under control of console switch four. The cards punched also contain all necessary supplementary information and are in a format which is suitable for direct re-loading by the LOADER routine.

D. Functions of Individual Sub-programs

1. The data load routine reads and stores parameters for the problem. It then proceeds to load and store the matrix in the proper format. In the case of fixed point input, storage locations are computed from the row/column designations, and the input elements are converted to floating point notation as they are read and stored.

2. The cost change routine reads ID/cost elements, searches the matrix for a corresponding identification number, and stores the new cost coefficient in the proper location.

3. The requirement change routine reads variable identification numbers and changes for the associated requirement element. The change is converted to floating point notations, the matrix is searched for the identification number, and the requirement vector is updated.

4. The shadow price calculator converts all cost coefficients to floating point notation and evaluates Zj-cj for each non-basis variable.

5. The dual algorithm computes a feasible solution for the problem. Its operation may be monitored allowing the typing of the value of the functional and the identifications of the variables that enter into and are removed from the basis. It should be noted in this regard that the objectives of feasibility and optimality may not be compatible at this point, and therefore the functional may actually decrease from one iteration to the next while the dual algorithm is in control.

During the reduction of the matrix of coefficients a test is made to determine whether the resulting quantity has a magnitude less than an amount called "essential zero." Any element falling into this category is set to zero in order to avoid computation with numbers which are actually not significant. "Essential zero" has been set to 10^{-6}. Its value (the corresponding floating point exponent) is located in memory addresses

3144 and 3145 and may be changed if desired. If this change is made, it is important to note that the high order digit must carry a flag.

6. The simplex algorithm starts with any feasible solution and computes the optimal feasible solution. Its operation may be monitored by allowing iteration typeout as in the case of the dual algorithm.

   The discussion of essential zero as applied to the dual algorithm in 5. above is equally valid for the simplex algorithm. In this case the test value is located in memory addresses 3264 and 3265.

   In the case of a tie in the quotient used as the criterion for choosing the variable to leave the basis on any iteration, the choice is resolved by using the largest denominator as a secondary criterion.

7. The basis output routine searches out and computes the information described in section C. above. Floating point quantities (with the exception of the functional) are converted to fixed point before being typed.

8. The non-basis output routine searches out and computes the information described in section C. above. Floating point quantities are converted to fixed point before being typed.

9. The matrix punch routine resets the permanent instructions in low memory to allow re-initiation of the input or change phases. Problem parameters and the final matrix are punched into paper tape if called for by the setting of console switch four.

E. Use of Storage

Memory locations from 00012 through 03919 are used by the program. Memory from 03920 up to the required location is used for the matrix and its manipulation. Locations 00000 through 00011 are available for the insertion of one instruction at a time through the typewriter. A halt instruction in position 00012 ensures that only this one instruction will be executed.

00012-00042	load routine
00048-00058	case identification
00060-00064	S address

00065-00069	M and 2
00070-00074	N
00075-00079	M
00080-00099	product area
00100-00400	arithmetic tables
00402-02064	floating point subroutines
variable	programs
03758-03917	input area (record mark in 03918)
03920-	matrix

F. Program Decks

1. LOADER

   3 pre-load cards.
   Arithmetic tables.
   Floating point routines.
   Data load routine.
   Matrix data to be entered.

2. COST CHANGER

   2 pre-load cards.
   Arithmetic tables.
   Floating point routines.
   Cost changer routine.
   Cost change data (if entered from cards).

3. RHS CHANGER

   2 pre-load cards.
   Arithmetic tables.
   Floating point routines.
   Requirement change routine.
   Requirement change data (if entered from cards).

4. SOLUTION

   2 pre-load cards.
   Shadow price calculation routine.
   Page headings.
   2 pre-load cards.
   Dual algorithm routine.
   2 pre-load cards.
   Simplex algorithm.
   2 pre-load cards.
   Final basis output routine.
   2 pre-load cards.
   Fix and print routine.
   Page headings.
   2 pre-load cards.
   Non-basis output routine.
   Page headings.
   2 Pre-load cards.
   Matrix punch routine.

G. Storage of matrix in memory

The matrix is stored in column sequence and uses memory from 3920 upward to the extent required by the problem size. The following diagram indicates the manner in which matrix elements are stored:

3920			S+10(2m+3) ID/C_j	S+10[(n+1)(m+2)-1]
	S	Functional	S+10(2m+4)	S+10[(n+1)(m+2)]
Working Column	ID/C_i	S+10(m+3) b_i	S+10(2m+5) Increasing j → ; a_{ij}: S+10[(j+1)(m+2)+i] ; Increasing i ↓	S+10[(n+1)(m+2)+1]
3920+10(m+1)	S+10m	S+10(2m+2)	S+10(3m+4)	S+10[(n+2)(m+2)-2]

S is computed for each problem and is equal to 3920 + 10(m+3). All addresses are field addresses.

III. SAMPLE PROBLEM

A. Matrix

Basis variables		Non-basis variables			Requirement Vector	
9901	9992	0001	0002	9902		Variable ID
1		1	2		6	
	1	1	-1	-1	-4	
0	-10	1	1	0		Cost

B. Matrix in storage

	$\bar{0}$000000000	$\bar{0}$001001000	$\bar{0}$002001000	9902000000
	$\bar{0}$000000000	$\bar{0}$000000000	$\bar{0}$000000000	$\bar{0}$000000000
$\bar{9}$901000000	$\bar{5}$160000000	$\bar{5}$110000000	$\bar{5}$120000000	$\bar{0}$000000000
$\bar{9}$992010000	$\bar{5}$140000000	$\bar{5}$110000000	$\bar{5}$110000000	$\bar{5}$110000000

C. Input

1. Fixed point

```
2 X 3
0002000030 1
001001                  001002                  002001                  002002
1.+                     2.+                     1.+                     1.+
002003                  001000                  002000                  +
+1.+                    6.+                     -4.+
0010019901000000        0020019992010000        0010010001001000        0010020002001000
0010039902000000        +
000
```

2. Floating point

```
2 X 3
00020000300
R901000000R9920100000000000000000000000000N160000000N140000000000100100000000000
N110000000N110000000000020010000000000000N120000000N110000000R902000000000000000
00000 .0  N11000000000000000000000000000000000000000000000000000000000000000000000
```

D. Output

1. Final matrix punched

```
2 X 3
00020000300
R9020000000001C010000000000000N160000000N210000000N160000000000200100ON110000000
N130000000N120000000R992010000N210000000N110000000000000000000R901000000N110000000
N110000000N110000000000000000000000000000000000000000000000000000000000000000000
```

2. Typewriter output

CASE 2 X 3

ITER NO	FUNCTIONAL	VAR OUT	VAR IN
001	5140000000	9992010000	0002001000
002	5130000000	9901000000	9902000000
001	5160000000	0002001000	0001001000
FUNCTIONAL	5160000000		

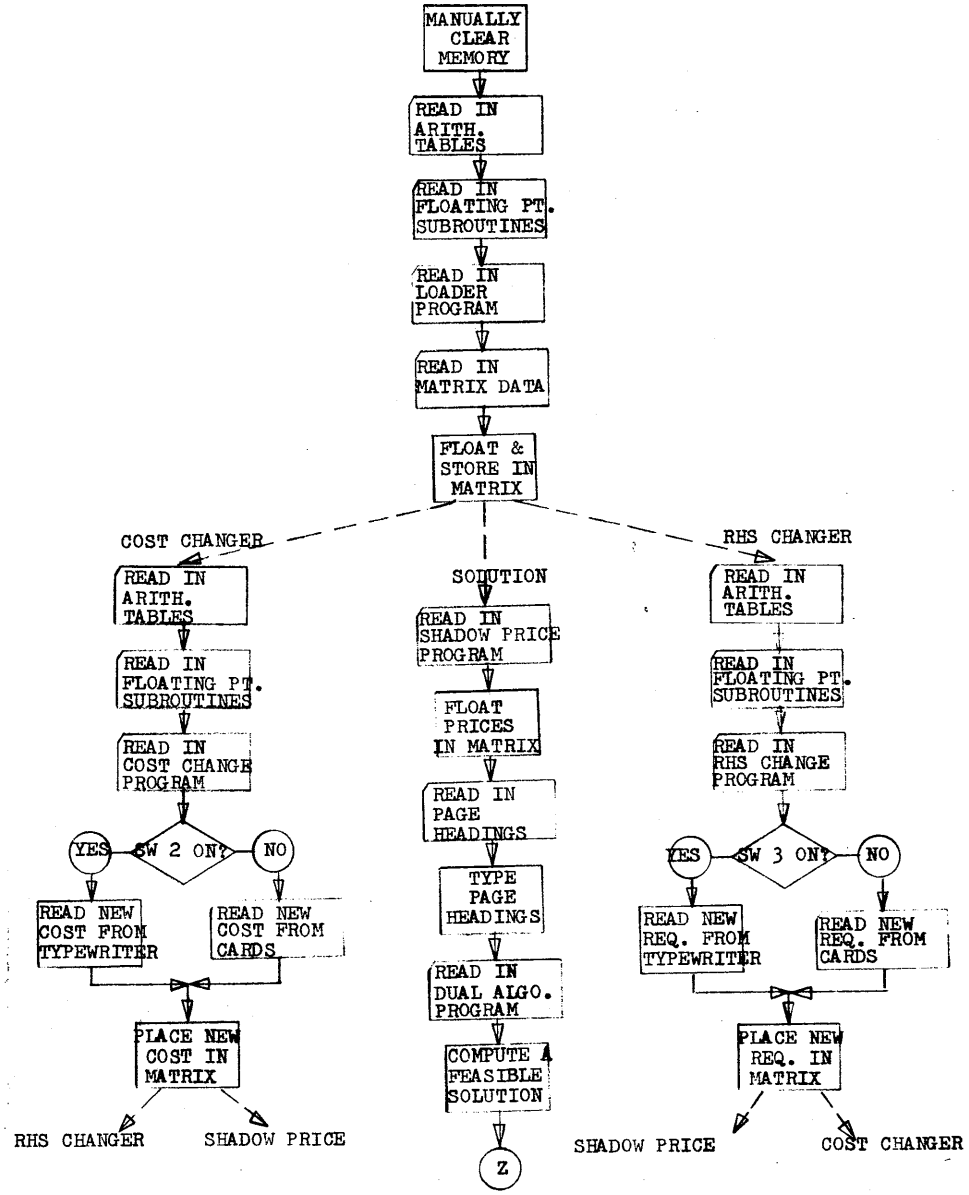
VAR/COST	ACTIVITY	LIM VAR	LOWER LIM	LIM VAR	UPPER LIM
9902000000	10.0000	0002	.3333-	9992	10.0000
0001001000	6.0000	0002	.5000	0000	9999.9000

VAR/COST	SHAD PRICE	LIM VAR	LOWER LIM	LIM VAR	UPPER LIM
0002001000	1.0000	0000	9999.9000-	0001	3.0000
9992010000	10.0000	9902	10.0000-	0000	9999.9000
9901000000	1.0000	0000	9999.9000-	0001	6.0000

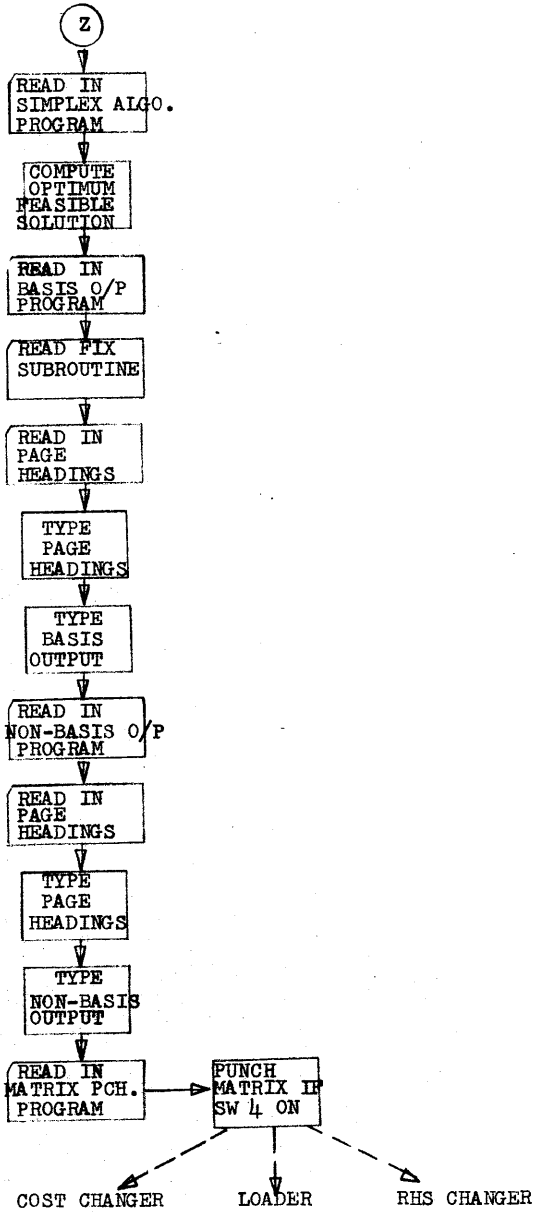
IV. METHOD OF LOADING PROGRAM

The "LOAD" key cannot be used to load programs in this routine since the first 80 positions of memory are not free. Therefore cards are read into locations 03838 through 03917, with a record mark in 03918 being utilized to transfer up to six instructions to their place in memory. The first two instructions on each card contain the "transfer record" instruction and a branch to location 03826, which contains the read instruction 36 03838 00500. The first two cards of each routine perform the function of setting the read instruction in 03826 and the record mark in 03918. The first card is read into 03826, and control is transferred there, using instructions 36 03826 00500 49 03826. This must be done by the operator at the beginning of the "LOADER" program, it is done automatically for all following routines.

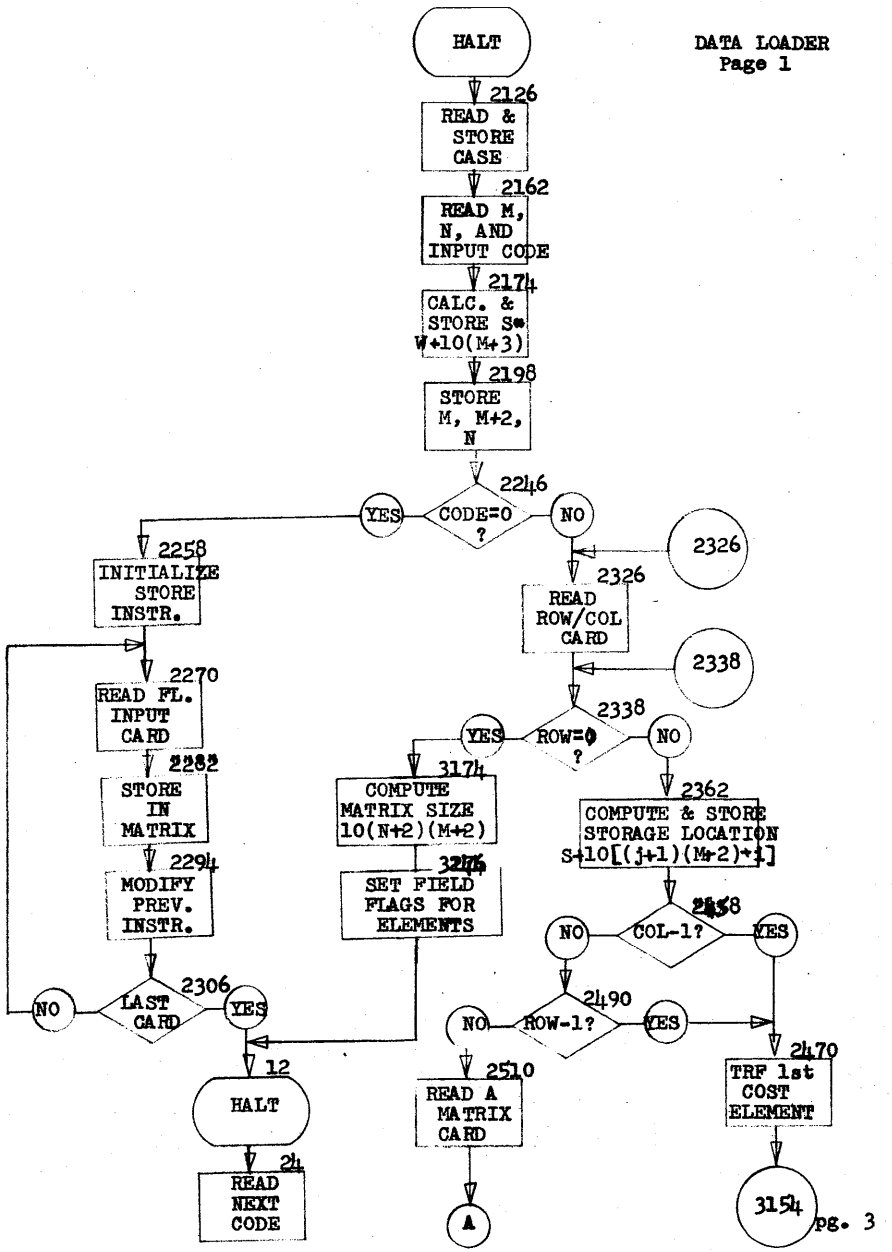
GENERAL FLOW
Page 1



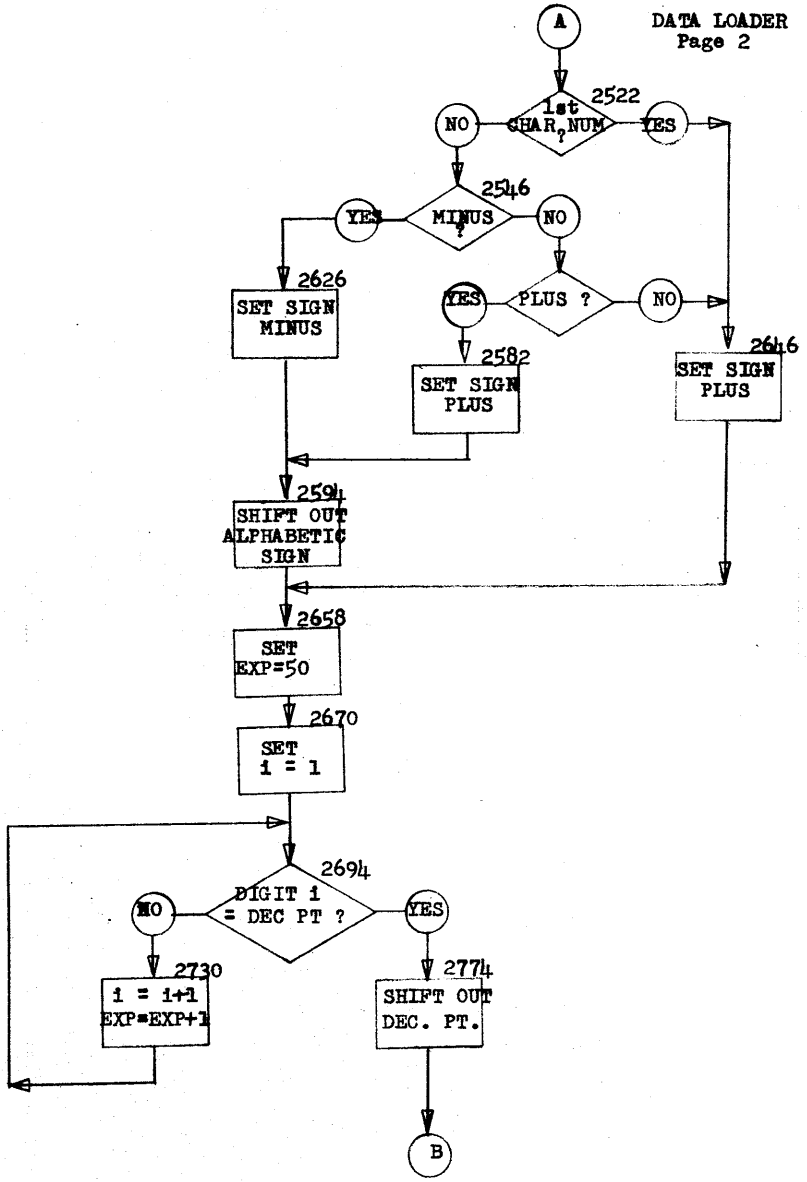
GENERAL FLOW
Page 2



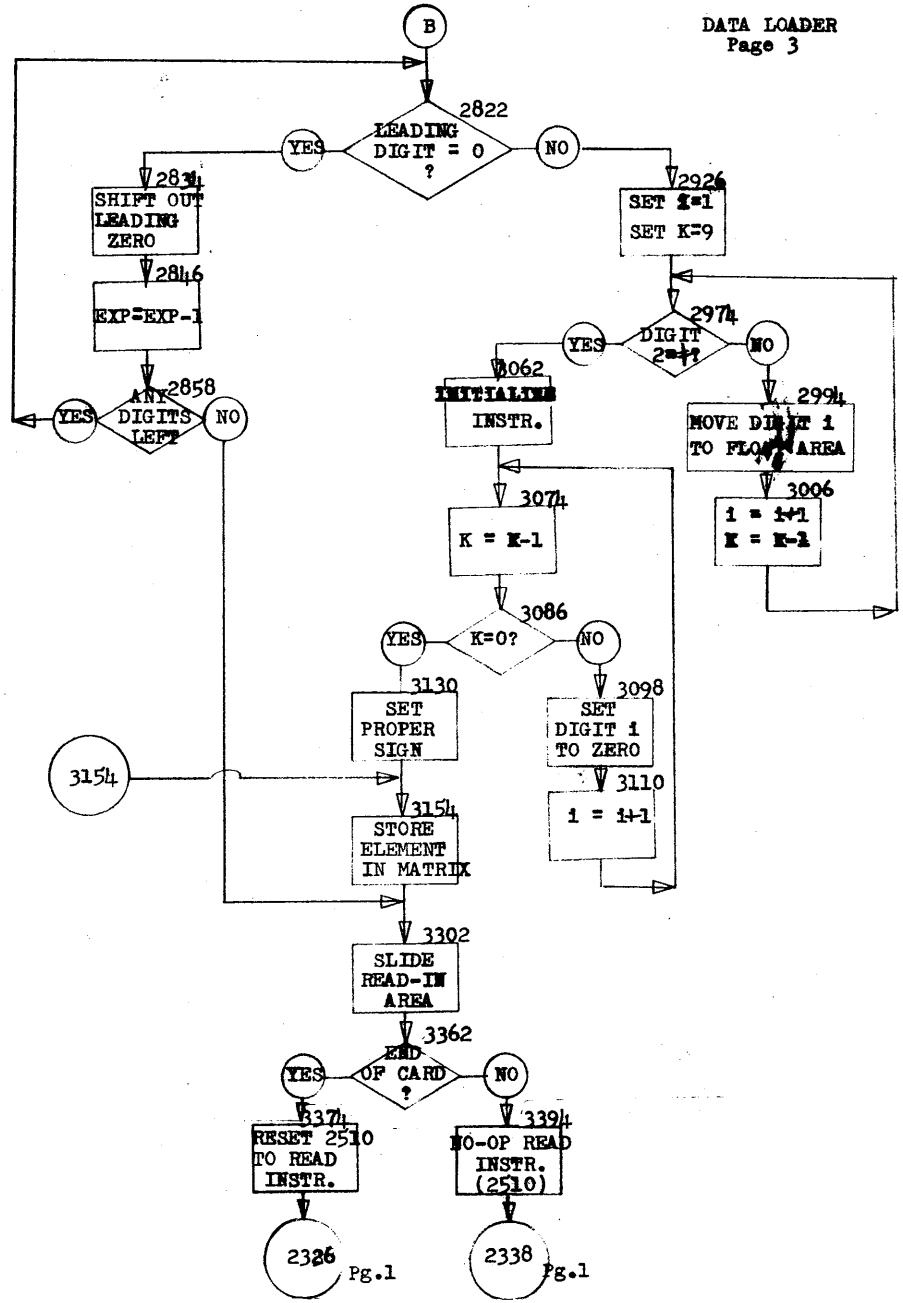
DATA LOADER
Page 1



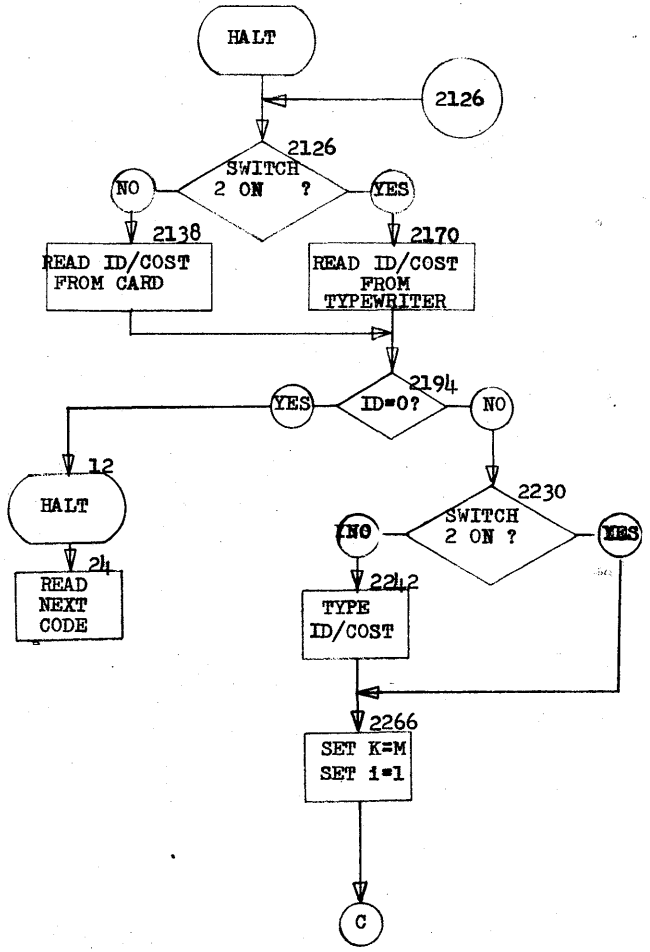
DATA LOADER
Page 2



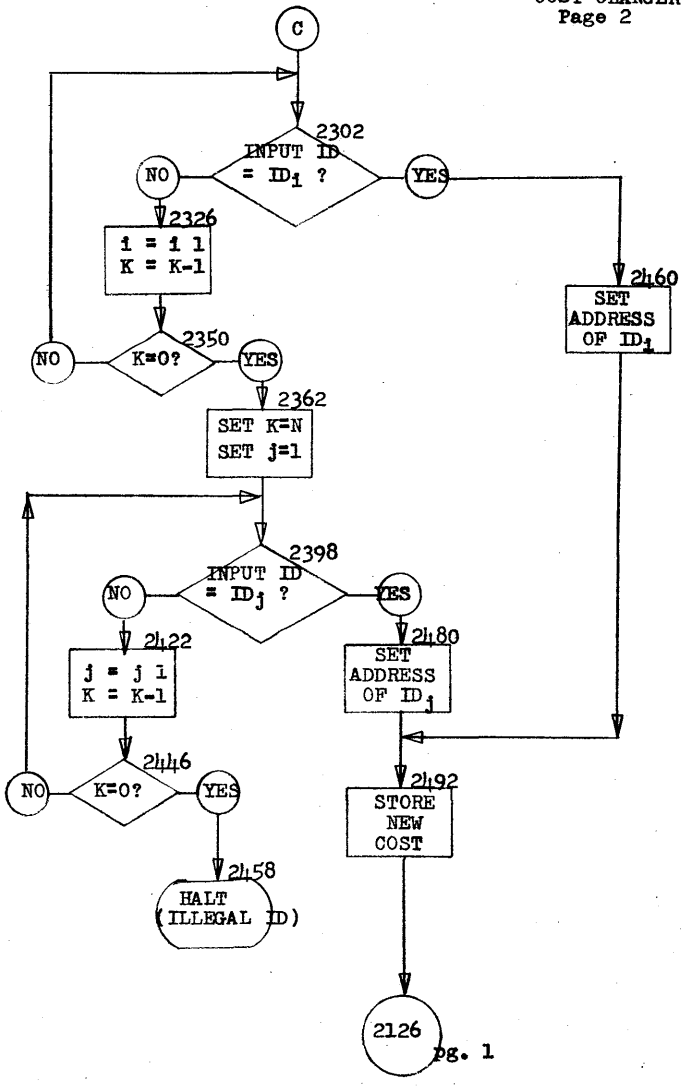
DATA LOADER
Page 3



COST CHANGER
Page 1

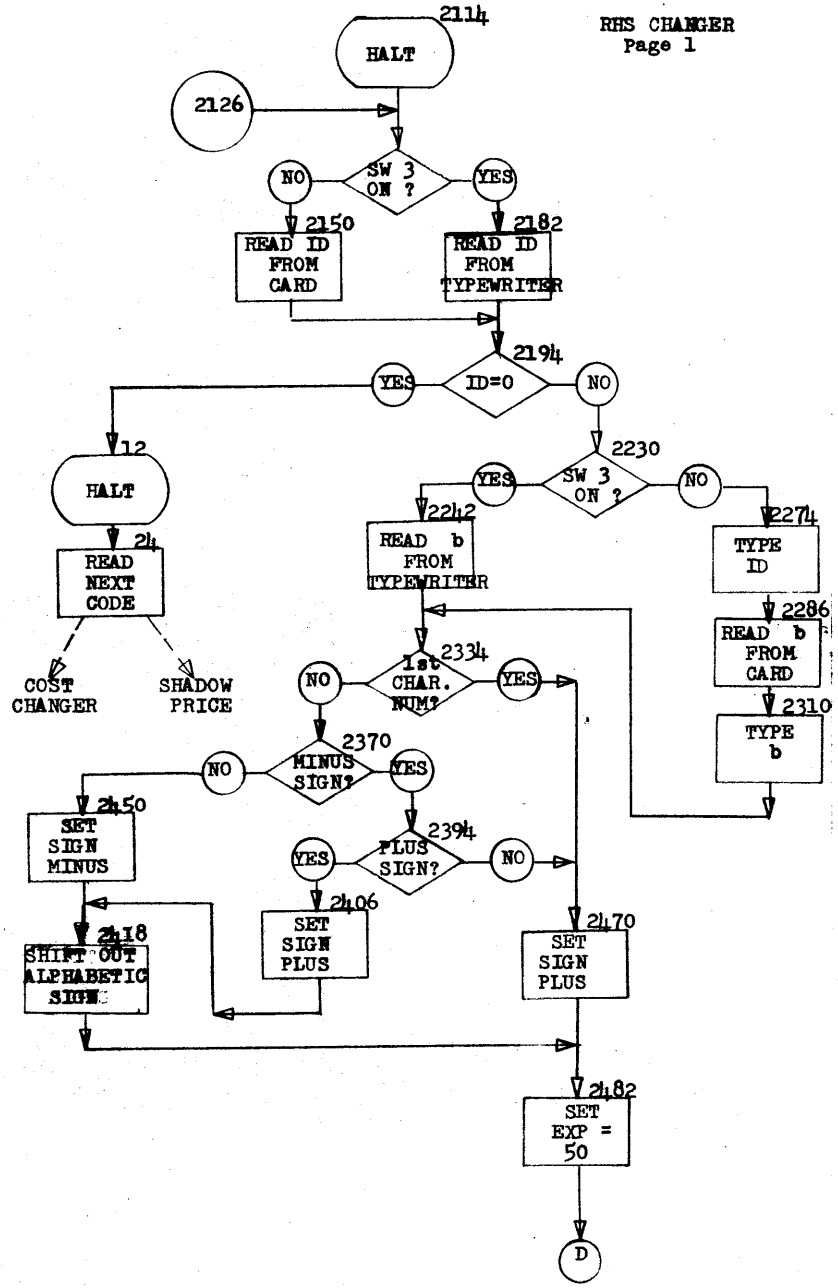


COST CHANGER
Page 2

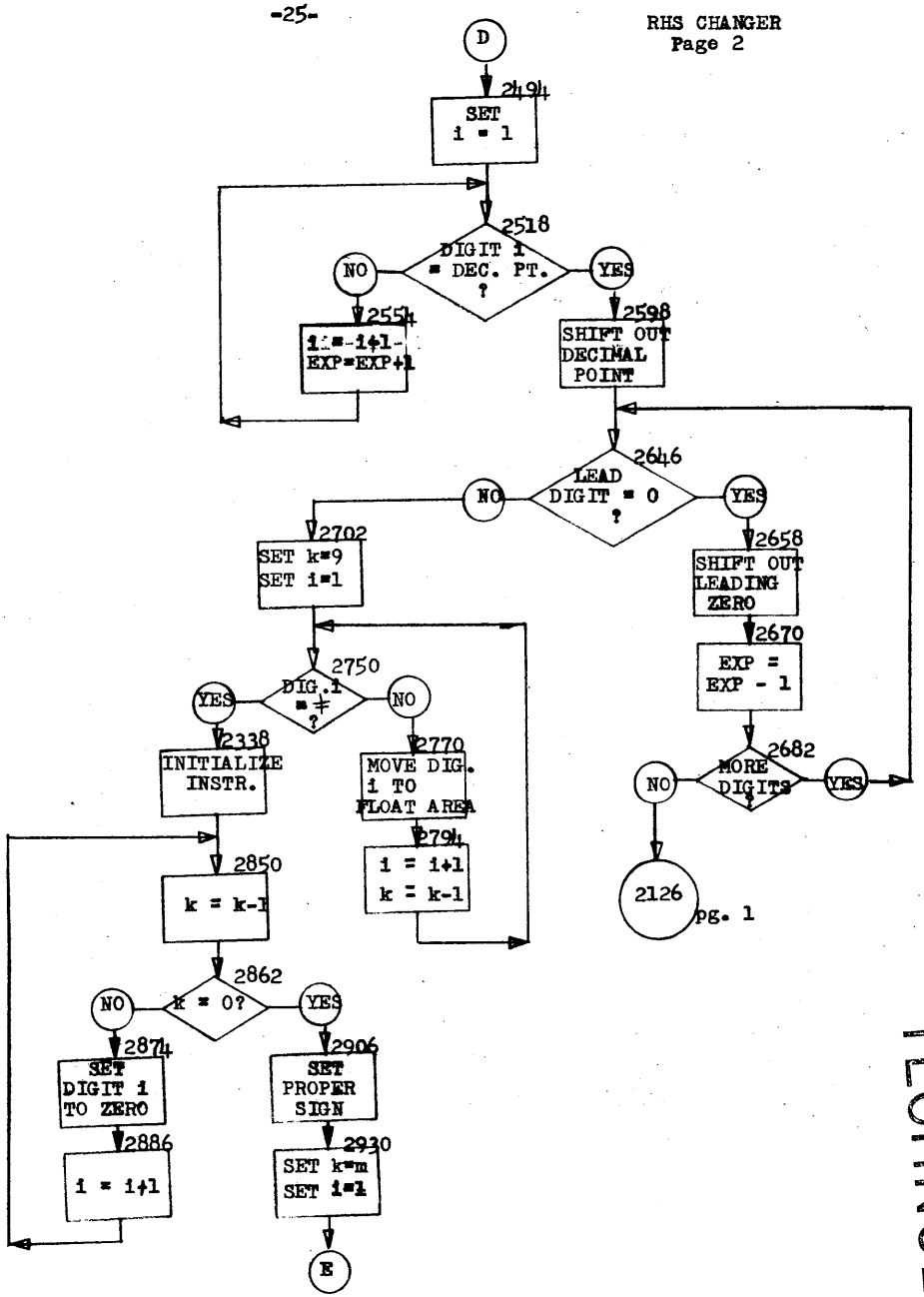


pg. 1

RHS CHANGER
Page 1

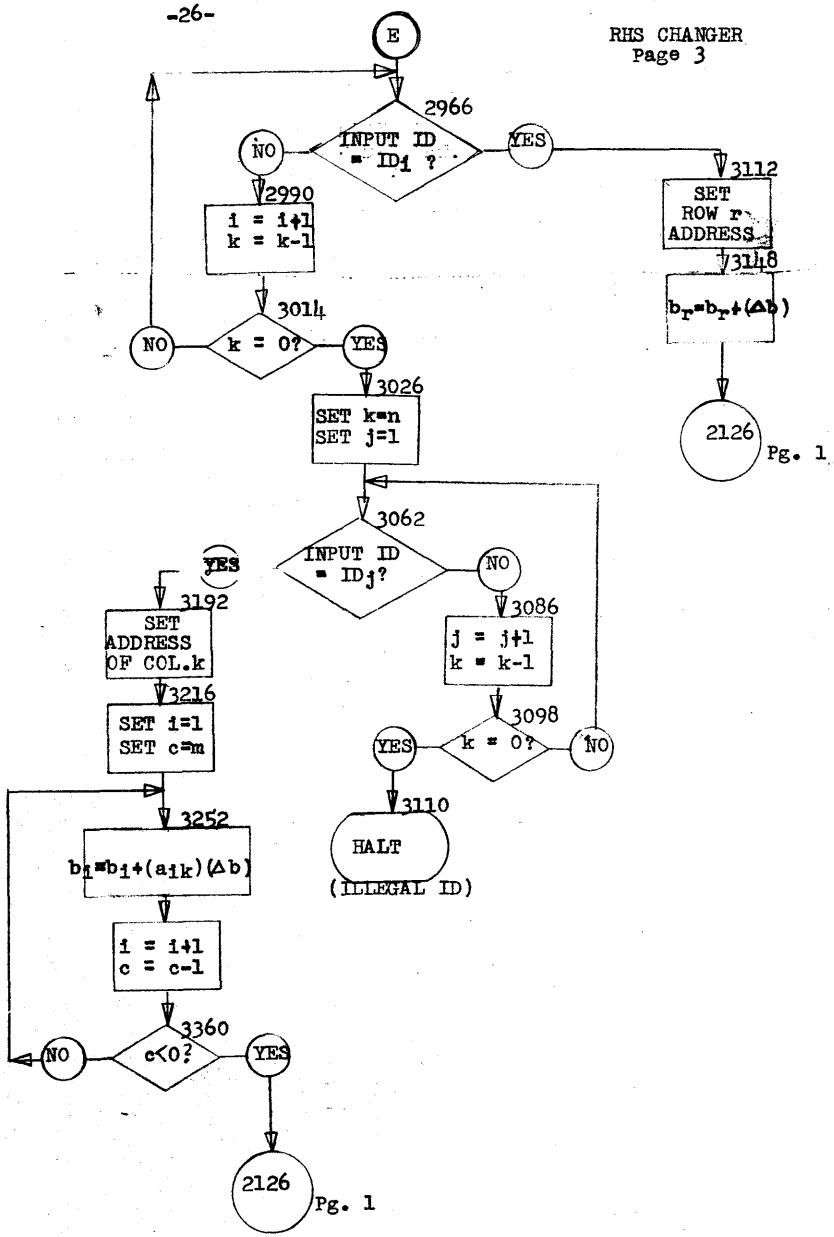


RHS CHANGER
Page 2

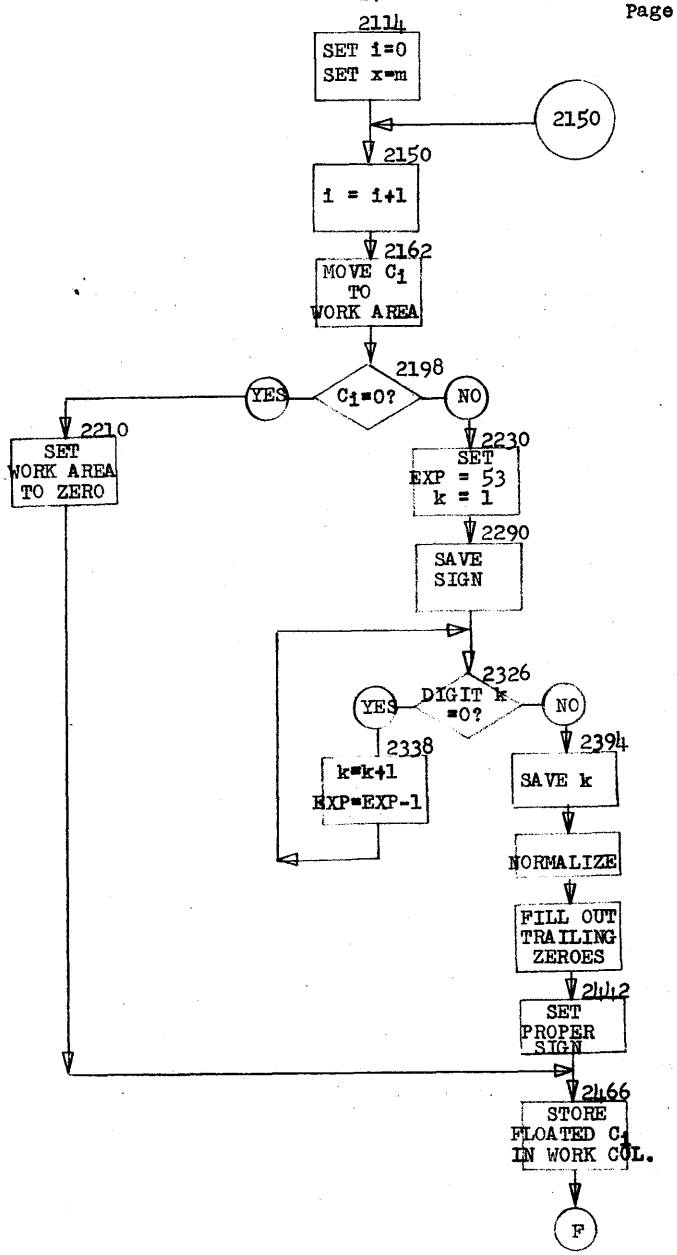


COMPUTER TECHNOLOGY

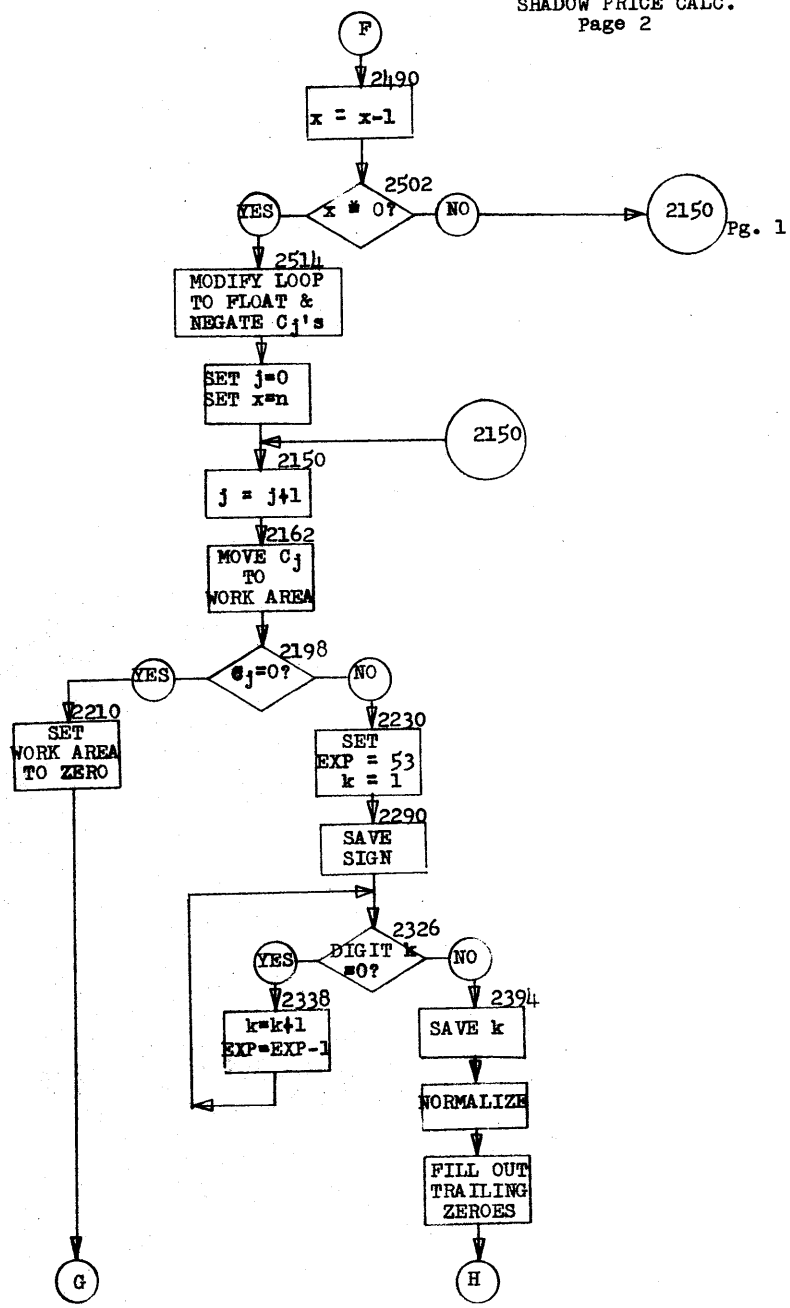
RHS CHANGER
Page 3



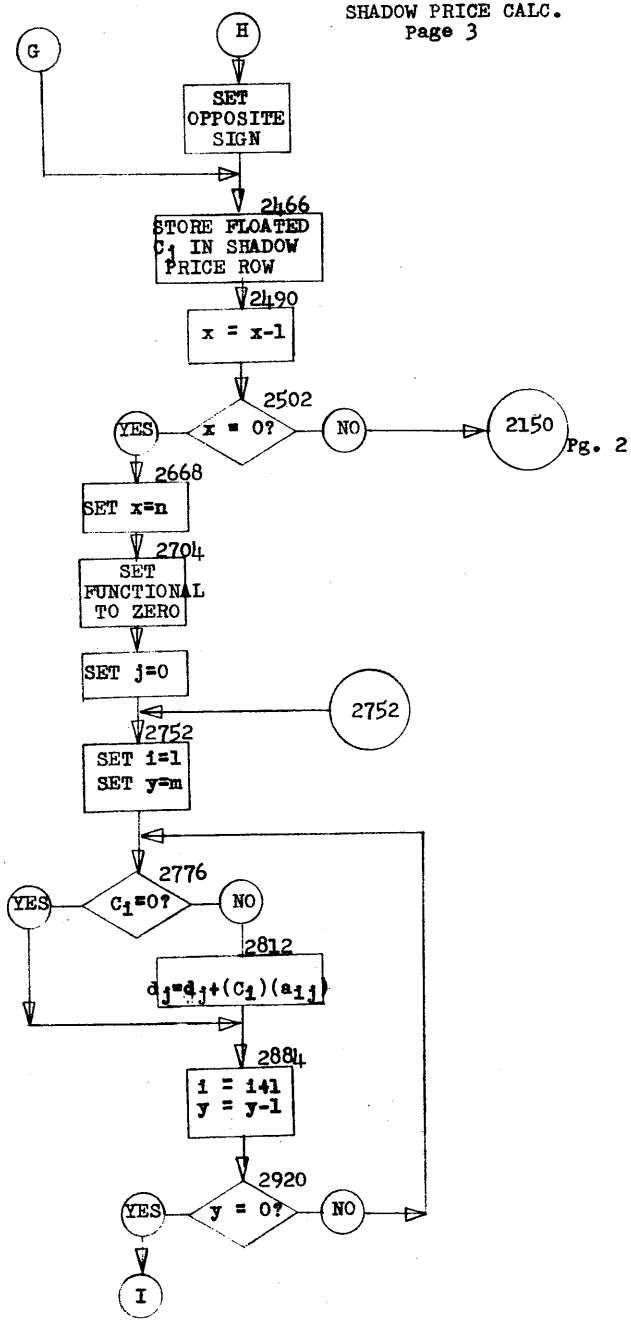
SHADOW PRICE CALC.
Page 1



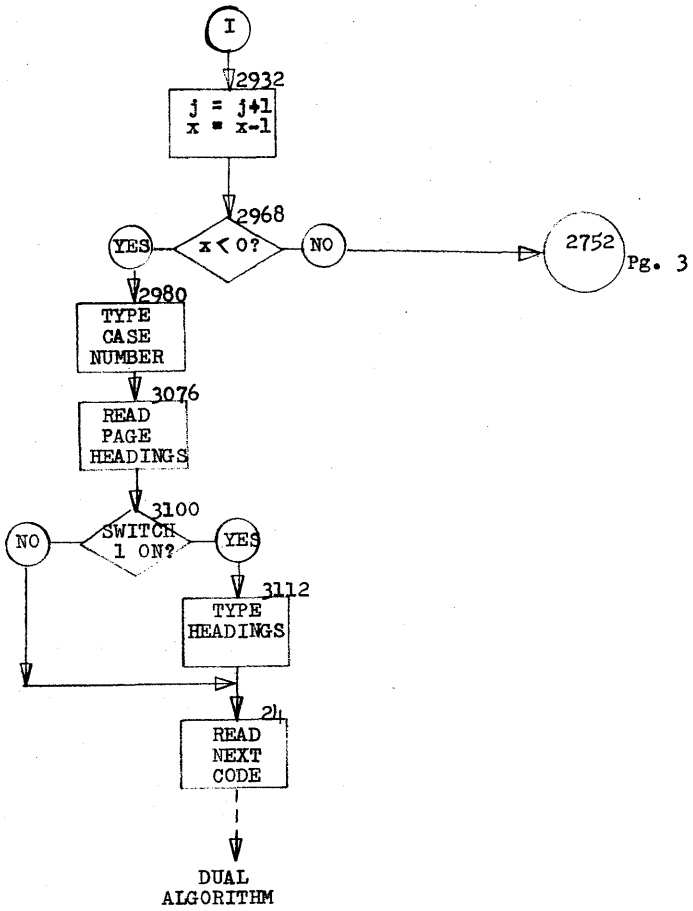
SHADOW PRICE CALC.
Page 2



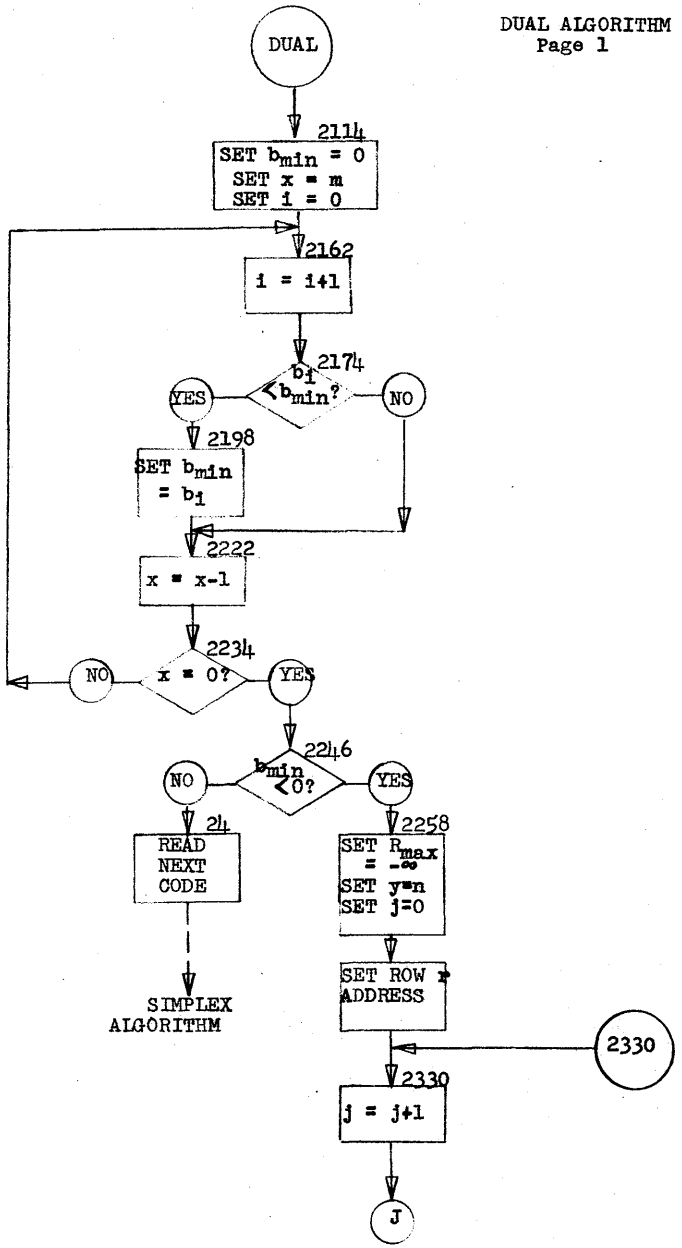
SHADOW PRICE CALC.
Page 3



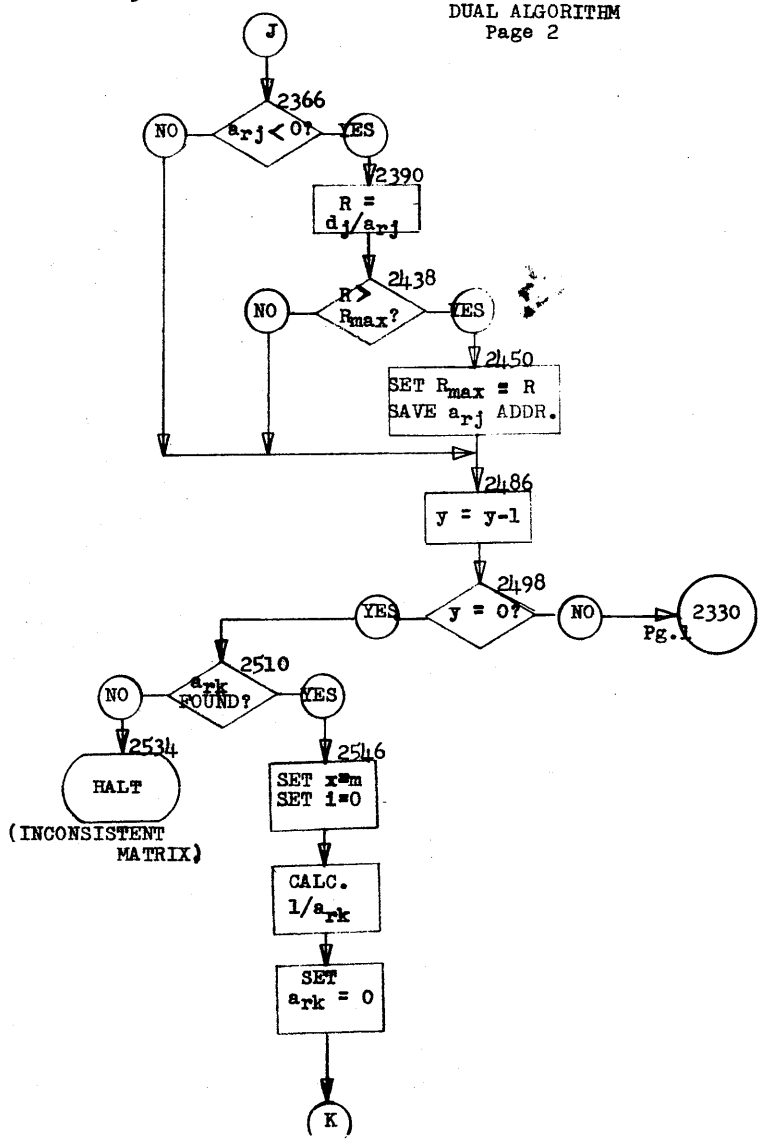
SHADOW PRICE CALC.
Page 4



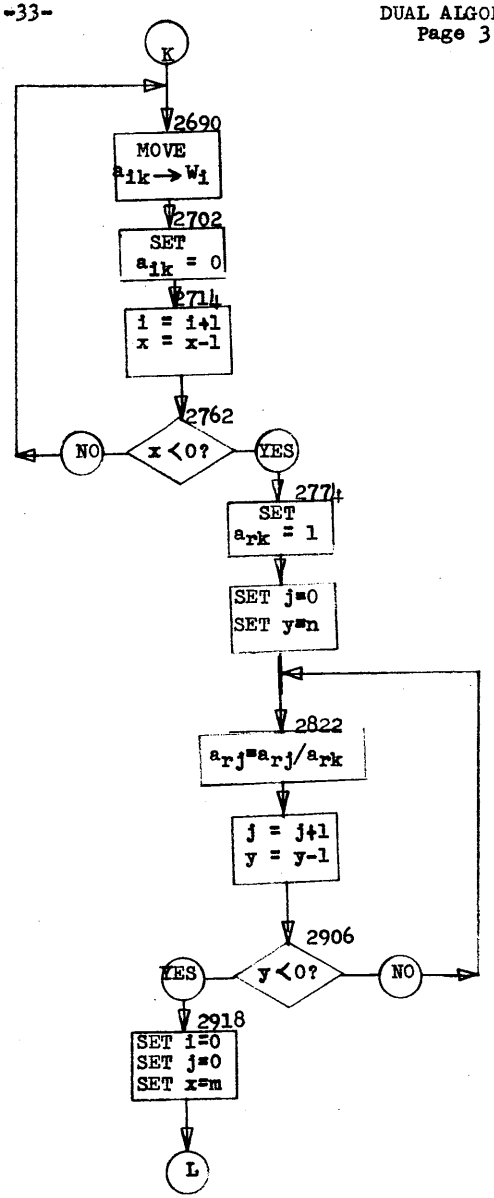
DUAL ALGORITHM
Page 1



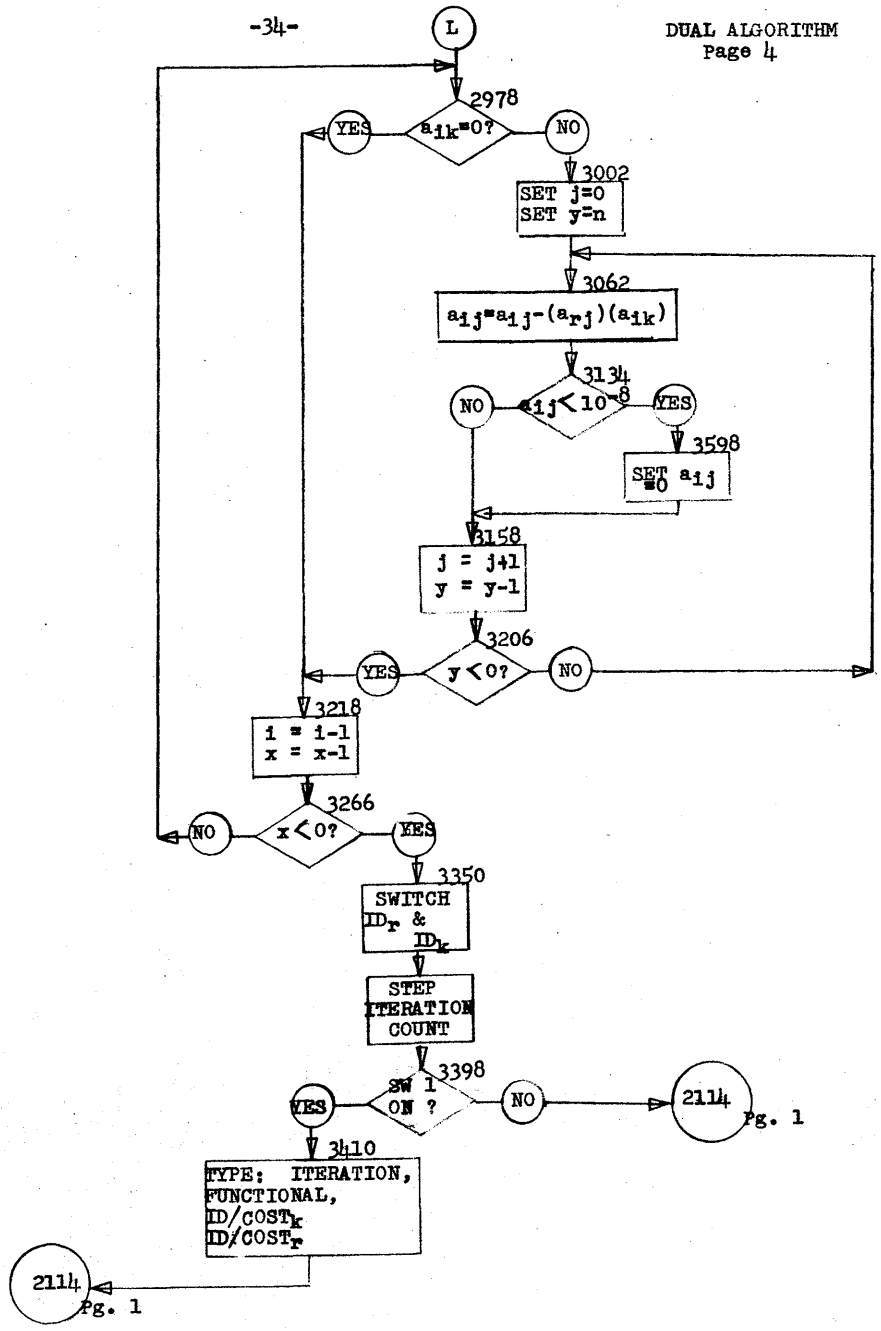
DUAL ALGORITHM
Page 2



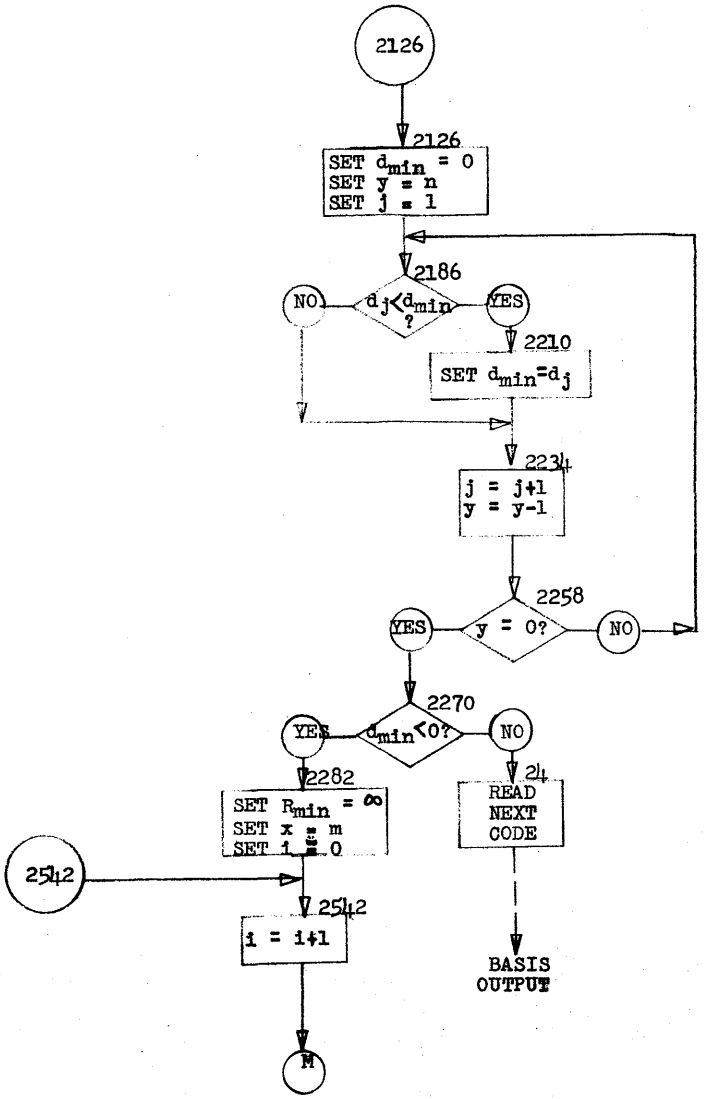
DUAL ALGORITHM
Page 3



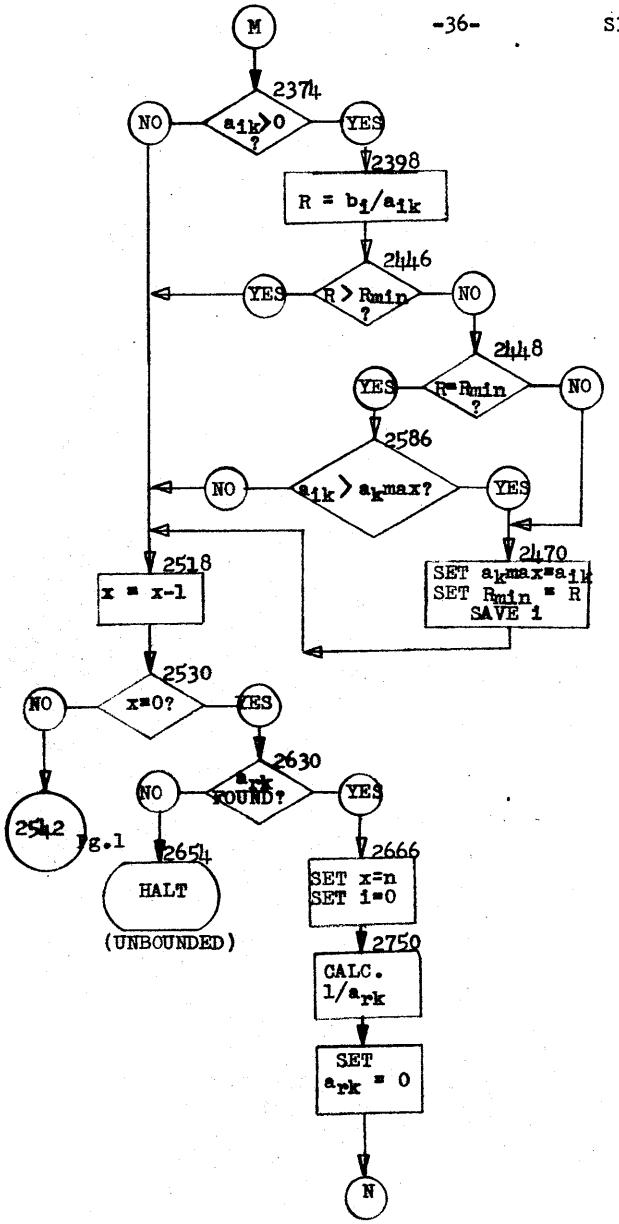
DUAL ALGORITHM
Page 4



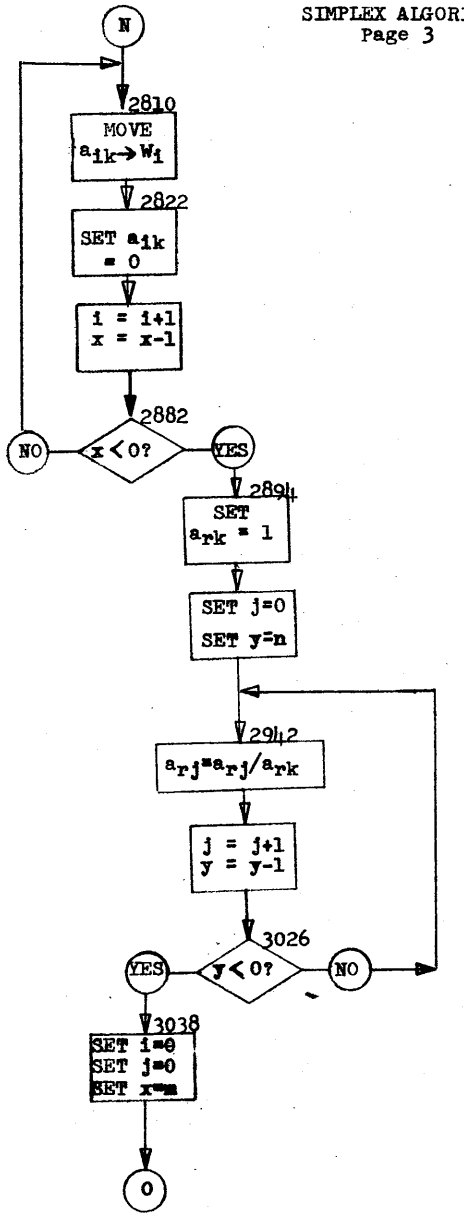
SIMPLEX ALGORITHM
Page 1



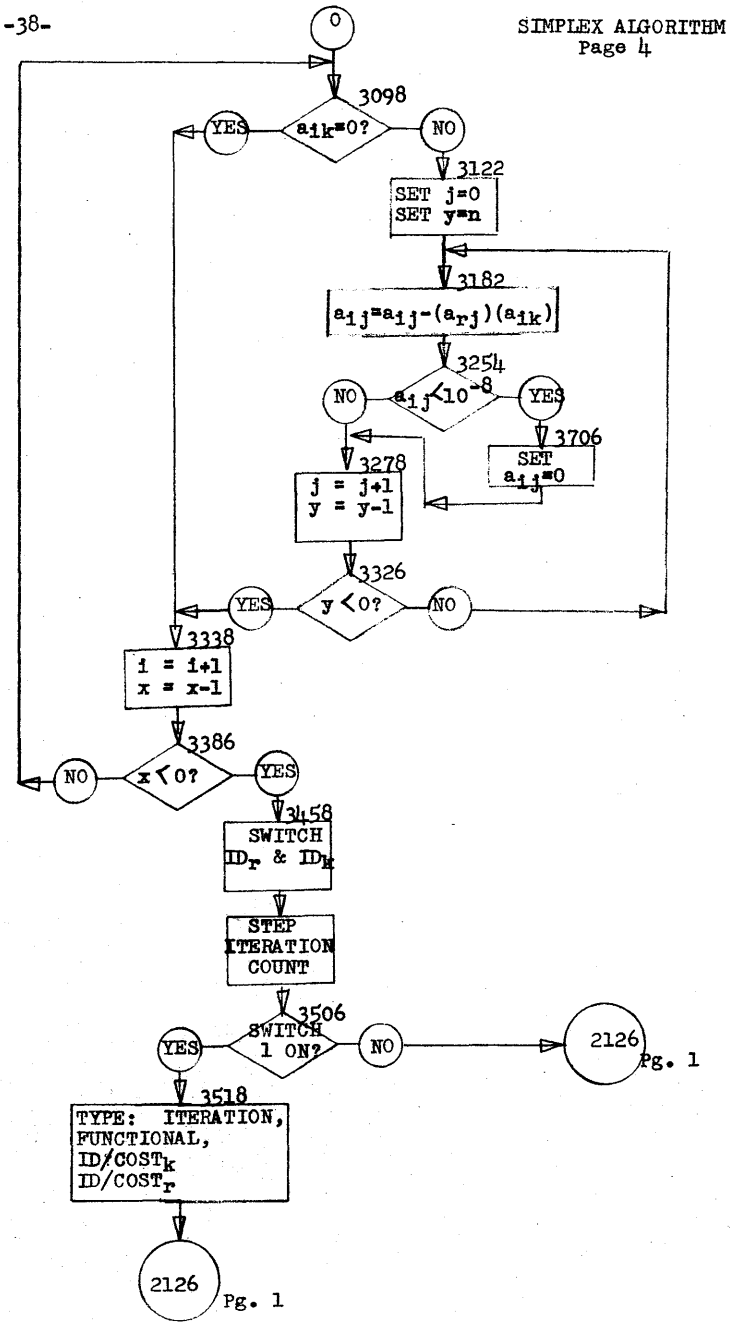
SIMPLEX ALGORITHM
Page 2



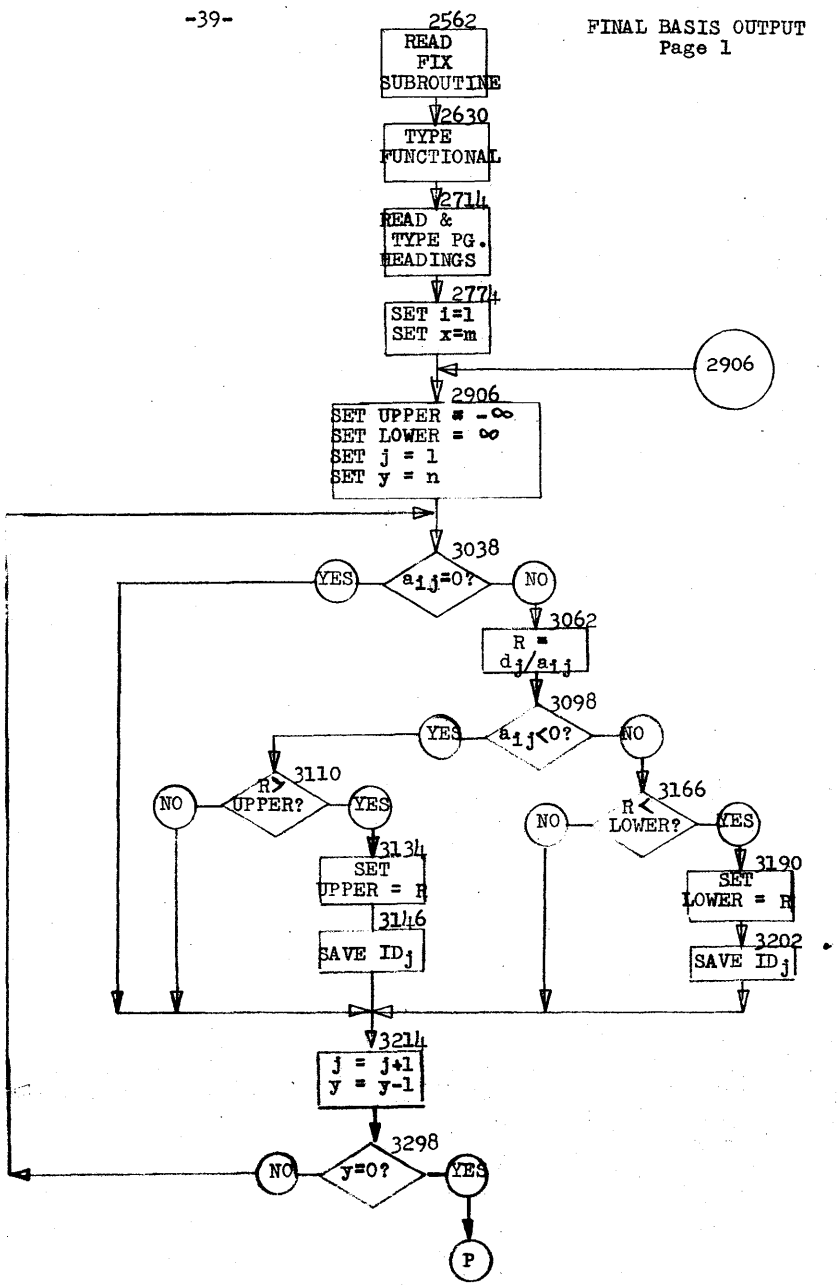
SIMPLEX ALGORITHM
Page 3



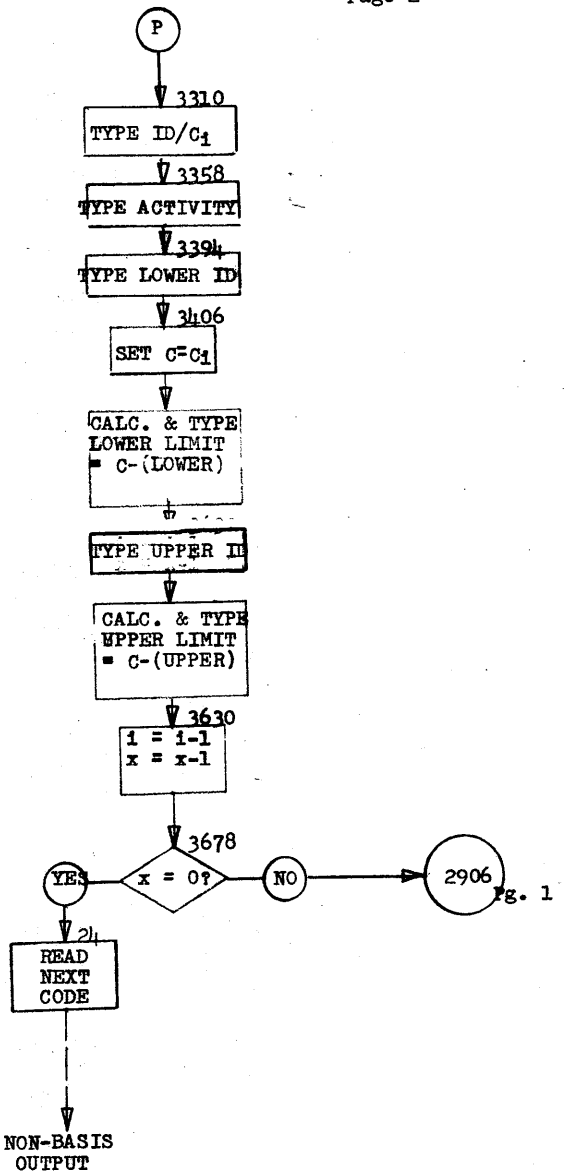
SIMPLEX ALGORITHM
Page 4



FINAL BASIS OUTPUT
Page 1

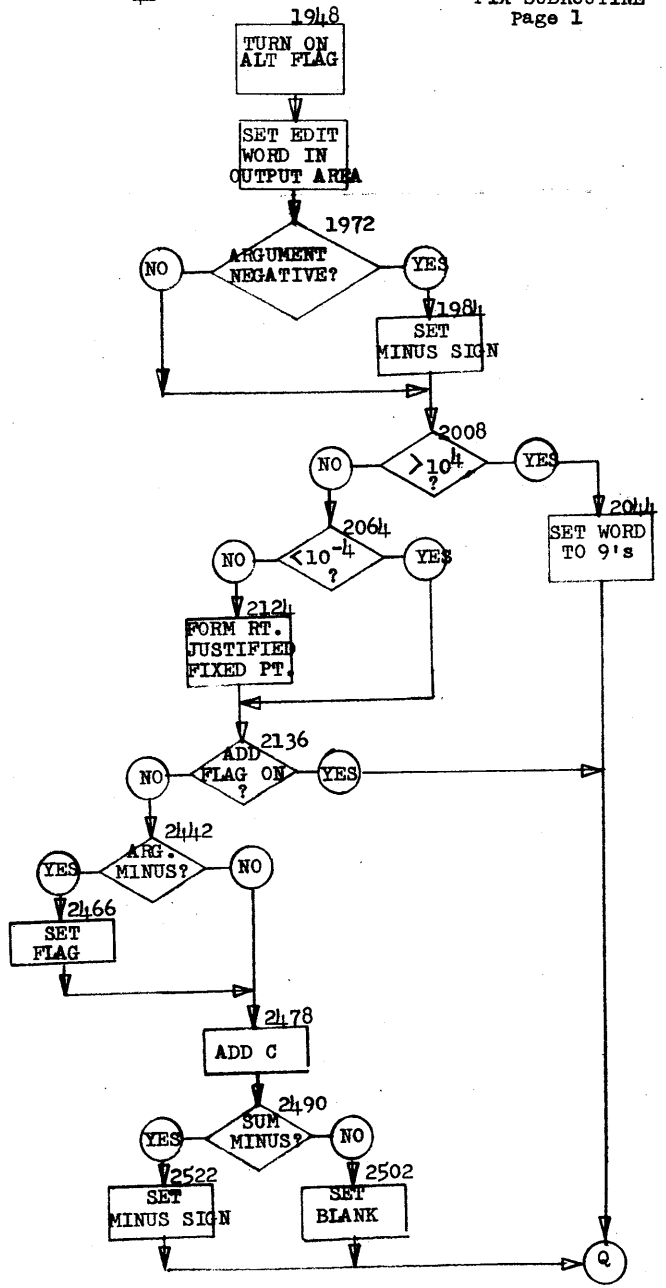


FINAL BASIS OUTPUT
Page 2

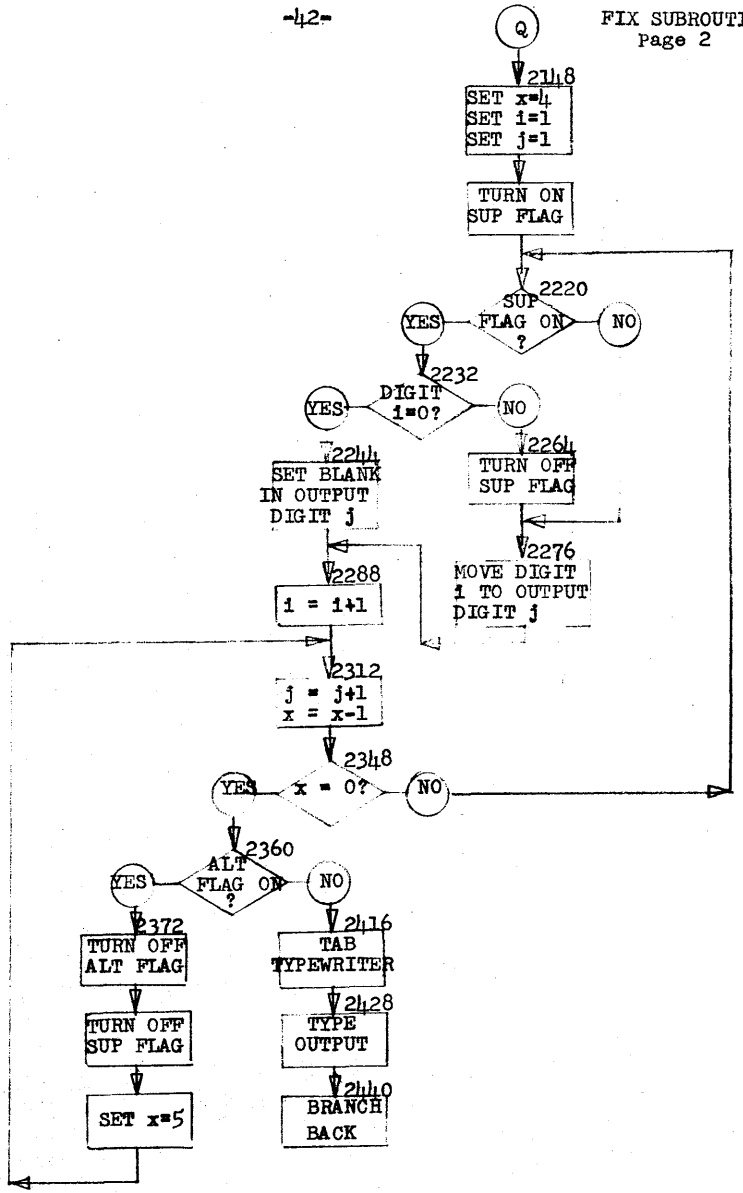


Pg. 1

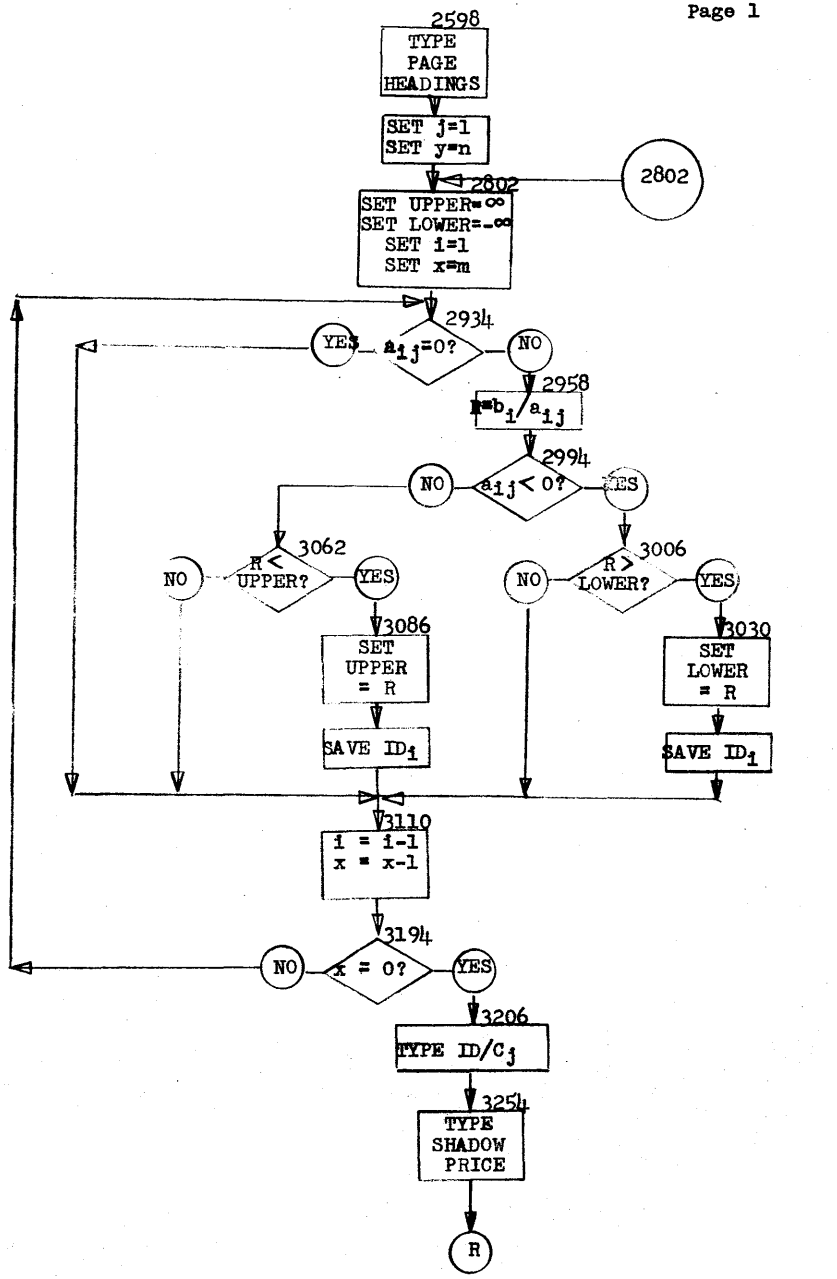
FIX SUBROUTINE
Page 1

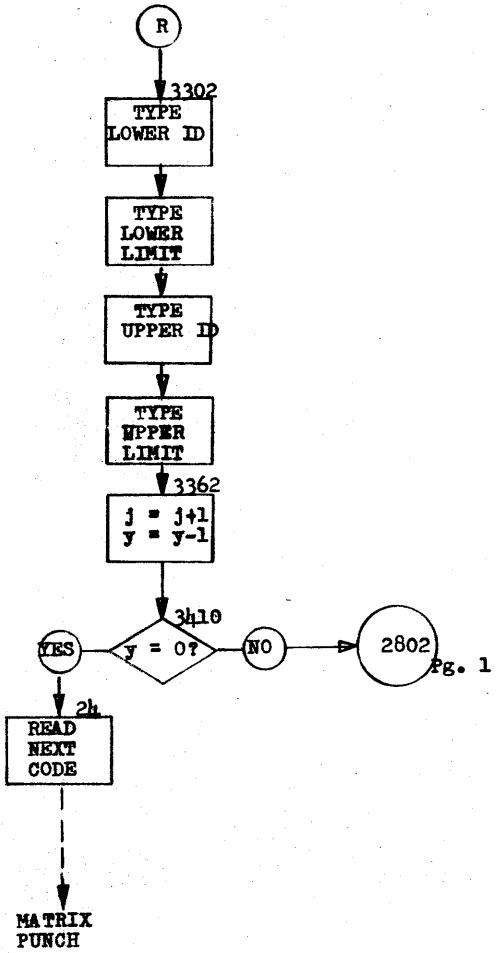


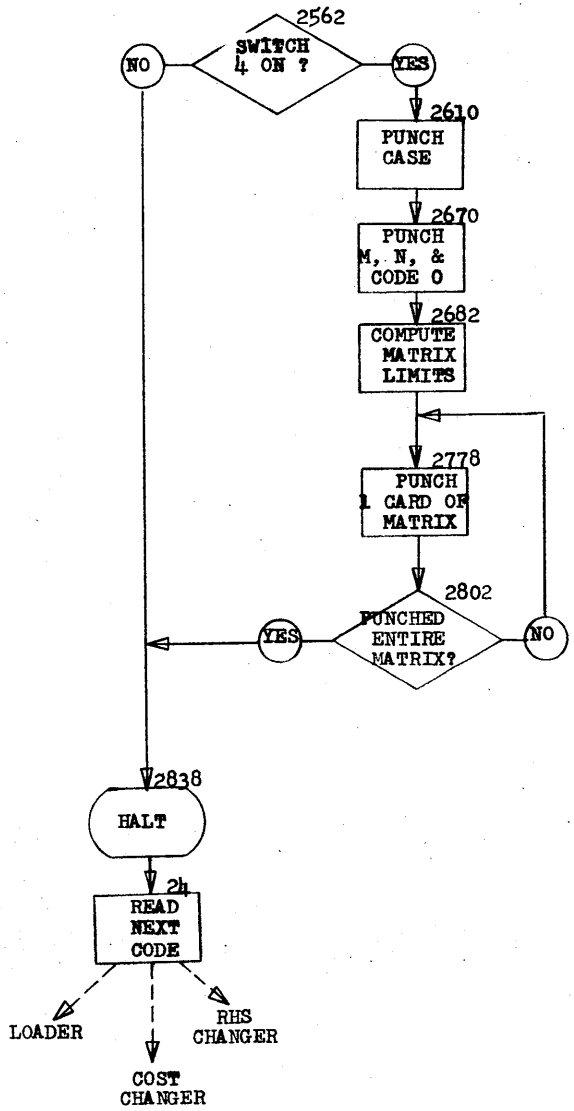
FIX SUBROUTINE
Page 2



NON-BASIS OUTPUT
Page 1




R
3302
TYPE LOWER ID
TYPE LOWER LIMIT
TYPE UPPER ID
TYPE UPPER LIMIT
3362
j = j+1
y = y-1
3410
y = 0?
YES
NO
2802
Pg. 1
24
READ NEXT CODE
MATRIX PUNCH


2562
SWITCH 4 ON ?
NO
YES
2610
PUNCH CASE
2670
PUNCH M, N, & CODE 0
2682
COMPUTE MATRIX LIMITS
2778
PUNCH 1 CARD OF MATRIX
2802
PUNCHED ENTIRE MATRIX?
YES
NO
2838
HALT
24
READ NEXT CODE
LOADER
COST CHANGER
RHS CHANGER

PRE-LOAD CARDS

```
360383800500
150391800002490382 6
3100012038874903826                    *                480000000000360382600500490382 6
```

ARITHMETIC TABLES

```
3100100038584903826 000000000000102030400020406080003060902100408021610050015102
3100160038584903826 006021814200704112820080614223009081726300000000005060708090
3100220038584903826 012141618151811242720242822363520353045403632484455324946536
3100280038584903826 048465462754453627180123456789123456789023456789OJ34567890JK
3100340038584903826 4567890JKL567890JKLM67890JKLMN7890JKLMNO890JKLMNOP90JKLMNOPQ
3100402038584903826 160053600538490043401600536005582601197000002601112004451601
3100462038584903826 124000001101124000014600494014004901128004692601176000002601
3100522038584903826 187018274900000022011860117649005700260118601176150084700001
3100582038584903826 160085701187260109201169220117801189470070201100260117601197
3100642038584903826 260119701186260118601176210109201189260117801189320117800000
3100702038584903826 150119800000140117800000Q4701094013004400774011973201198000000
3100762038584903826 330119700000210085701178440082201186330118600000150084700002
3100822038584903826 150117800000150118900000210119800000025010930119833011980000O
3100882038584903826 430098601189430103401190460096601200310119001191250119800557
3100942038584903826 120109200001460089401300260119701826490110602601198011973301
3101002038584903826 190000001101092000014601126014001501189000001101198000054300
3101062038584903826 986011894401094010933201197000002601189010922600000011974900
3101122038584903826 000048260115100469260051700000490050600000000000000000000000
3101182039004903826                    *                          *  00000000000000002
3101200038584903826 160053601256150175300002160111200099320119000000490045804301
3101260038584903826 276011904900966043012900116948220118901168250132501176330117
3101320038584903826 600000150116800000160134700000110134700001210117601176430139
3101380038584903826 401168490135001601441018372501428011691101441O *002301175000
3101440038584903826 002601477014411201477000042601166000002201166000933201162000
3101500038584903826 002301176011652601186019372201186000963201178000002301186011
3101560038584903826 863300082000001500081000OJ21000900118626011860009023011860l1
3101620038584903826 652100095000951201347000014701622012002601186000944401694013
3101680038584903826 253201186000002301186011971101189000N025010930009926010920l1
3101740038584903826 892601198000924601774013004900966047008820140049011260000000
3101800038584903826 000000000000000000000000000000J90692R080J74077P568J60128040 5J4
3101860038584903826 8249N490J38013M759J29099M164J21267L674J14332L266J08147K922J0
3101920038584903826 2597K630J000000000160053601970150175300003490122404301990011
3101980038584903826 904900966043020100116949009660320116900000120116800ON1230119
3102040038934903826                    *                          701176210118901168490171 8
```

```
3102114038584903826 480000000000370375900500250376900400310004803758360383800500
3102174038584903826 2102196038411600064039502600079038421103842000K0260006903842
3102234038584903826 260007403847430232603848210228800064360383800500310000103838
3102294038584903826 1102288000Q047022700090049000120360364300500140364500000 4603
3102354038584903826 1740120023036480006921000990006933000990000021000980364521 00
3102414038584903826 0980006333036460000032000950000026031600009944024900364826 02
3102474038584903826 1250365849031540440251003645490247004903418000001403759000P0
3102534038584903826 460264601300140375900 0K0460262601200430264603759330260500000
3102594038584903826 320376000000310375803760490265803202605000004902594033026050
3102654038584903826 0000160211700 0N0160271203759160270003760320000000000140000 00
3102714038584903826 0003460277401200110271200002110270000002110211700001490269 40
3102774038584903826 260282102700260281602712120281600001310000000000430292603759
3102834038584903826 310375803760120211700001450282203759490330200000000000000000
3102894038584903826 0000000000000000000000000000000000160265700009160298503759160 3
3102954038584903826 0000211816030050375945029940000049030620250000000000110298 50
3103014038584903826 0002110300000001110300500002120265700001490297402603104030 00
3103074038584903826 1202657000014603130012001500000000001103104000014903074044 03
3103134038584903826 1540260532021250000026000000212549033020260325200064120325 20
3103194038584903826 00J9230006900073320009500000210009900069210009900069320000 00
3103254038704903826 0000110325200 0J0120009900 0J0470324601200490001 20
3103302038584903826 3103758037983103798038383103838038783103878039183103643036 63
3103362038584903826 4503394036431602516034184902326016025160345449023380000031 03
3103422038664903826 75803500310383803500370375900500320375800004902522 0
3103500038584903826 ------------------------------------------------------------
3103560038974903826                  *                        ----------Z
2503723004004902126
```

```
360383800500
150391800002490382 6
3100100038584903826 0000000000001020304000204060800030609021004080216100500151020
3100160038584903826 0060218142007041128200806142230090817263000000000050607080900
3100220038584903826 0121416181518112427202428223635203530454036324844553249465360
3100280038584903826 0484654627544536271801234567891234567890234567890J34567890JK
3100340038584903826 4567890JKL567890JKLM67890JKLMN7890JKLMN0890JKLMN0P90JKLMN0PQ
3100402038584903826 1600536005384900434016005360055826011970000026011120044516010
3100462038584903826 1240000011011240000146004940140049011280046926011760000026010
3100522038584903826 1870182749000000220118601176490057002601186011761500847000010
3100582038584903826 1600857011872601092011892201178011894700702011002601176011970
3100642038584903826 2601197011862601186011762101092011892601178011893201178000000
3100702038584903826 1501198000001401178000094701094013004400774011973201198000000
3100762038584903826 3301197000002100857011784400822011863301186000001500847000020
3100822038584903826 1501178000001501189000002101198000002501093011983301198000000
3100882038584903826 4300986011894301034011904600956012003101190011912501198005570
3100942038584903826 1201092000001460089401300260119701826490110602601198011973301
3101002038584903826 1900000011010920000146011260140015011890000011011980000543000
3101062038584903826 9860118944010940109332011970000026011890109226000000119749000
3101122038584903826 0000482601151004692600517000004900506000000000000000000000000
3101182039004903826                    *                                   * 000000000000000002
3101200038584903826 1600536012561501753000021601112000993201190000004900458043010
3101260038584903826 2760119049009660430129001169482201189011682501325011763301170
3101320036584903826 6000001501168000001601347000001101347000012101176011764301390
3101380038584903826 4011684901350015014410183725014280116911014410 *002301175000
3101440038584903826 0026014770144112014770000426011660000022011660009332011620000
3101500038584903826 0023011760116526011660193722011860009632011780000023011860110
3101560038584903826 8633000820000015000810000J21000900118626011860009023011860110
3101620038584903826 6521000950009512013470000147016220120026011860009444016940130
3101680038584903826 2532011860000023011860119711011890000N02501093000992601092011
3101740038584903826 8926011980009246017740130049009660470088201400490112600000000
3101800038584903826 000000000000000000000000000000J90692R080J74077P568J6012804054J4
3101860038584903826 8249N490J38013M759J29099M164J21267L674J14332L266J08147K922J0
3101920038584903826 2597K630J000000000160053601970150175300003490122404301990011
3101980038584903826 9049009660430201001169490096603201169000001201168000N1230119
3102040038934903826                    *                   701176210118901168490171 8
3102114038584903826 4806000000004602170002003603838005002503848004004902194034000
3102174038584903826 0000010236038380010032038380000014038410000046000120120046020
3102234038584903826 2660020034000000010238038380010026022050007926023080006411020
3102294038584903826 3080000424000000384146024600120011023080000J0120220500 0J04702
3102354038584903826 3020120026022050007426024040230821024040006924000000384146020
3102414038584903826 4800120021024040006912022050 0J04702398012004826025100230849
3102474038764903826                    *  0249202602510024041102510000062600000003847
3102516038744902126                    *260254602510120254600005330000000000049021260
```

FLOATING POINT ROUTINES

FLOATING POINT ROUTINES

FLOATING POINT ROUTINES

```
360383800500
150391800002Z4903826
3100100038584903826 0000000000001020304000204060800030609021004080216100500015102
3100160038584903826 0060218142007041128200806142230090817263000000000050607080 90
3100220038584903826 0121416181518112427202428223635203530454036324844553249465 36
3100280038584903826 0484654627544536271801234567891234567890234567890J34567890JK
3100340038584903826 4567890JKL567890JKLM67890JKLMN7890JKLMNO890JKLMNOP90JKLMNOPQ
3100402038584903826 1600536005384900434016005360055826011970000026011120044516 01
3100462038584903826 1240000011011240000146004940140049011280046926011760000026 01
3100522038584903826 1870182749000000220118601176490057002601186011761500847000 01
3100582038584903826 1600857011872601092011892201178011894700702011002601176011 97
3100642038584903826 2601197011862601186011762101092011892601178011893201178000 00
3100702038584903826 1501198000001401178000Q4701094013004400774011973201198000 00
3100762038584903826 3301197000002100857011784400822011863301186000001500847000 02
3100822038584903826 1501178000001501189000002101198000002501093011983301198000 00
3100882038584903826 4300986011894301034011904600966012003101190011912501198005 57
3100942038584903826 1201092000014600894013002601197018264901106026011980119733 01
3101002038584903826 1900000011010920000146011260140015011890000011011980000543 00
3101062038584903826 9860118944010940109332011970000026011890109226000000011974900
3101122038584903826 0000482601151004692600517000000490050600000000000000000000000000
3101182039004903826                    *                                *  000000000000000002
3101200038584903826 1600536012561501753000021601112000993201190000004900458043 01
3101260038584903826 2760119049009660430129001169482201189011682501325011763301 17
3101320038584903826 6000001501168000001601347000001101347000012101176011764301 39
3101380038584903826 4011884901350016014410183725014280116911014410 *002301175000
3101440038584903826 0026014770144112014770000426011660000022011660009332011620 00
3101500038584903826 0023011760116526011860193722011860009632011780000023011860 11
3101560038584903826 8633000820000015000810000J21000900118626011860009023011860 11
3101620038584903826 6521000950009512013470000147016220120026011860009440169401 3
3101680038584903826 2532011860000023011860119711011890000N02501093000992601092011
3101740038584903826 8926011980009246017740130049009660470088201400490112600000 00
3101800038584903826 0000000000000000000000000000000J90692R080J74077P568J6012804 05J4
3101860038584903826 8249N490J38013M759J29099M164J21267L674J14332L266J08147K922J0
3101920038584903826 2597K630J000000000160053601970150175300003490122404301990011
3101980038584903826 9049009660430201001169490096603201169000001201168000N1230119
3102040038934903826                    *                          701176210118901168490171 8
```

REQUIREMENT
CHANGE
ROUTINE

```
3102114038584903826 480000000000340000000102460218200300360340000500250340400400
3102174038584903826 490219403603400001003203400000000140340300000460001201200470 2
3102234038584903826 274003003400000000108370375900100490233403803400001003703759 0
3102294038584903826 050041000000000034000000010839037590010032037580000014037590
3102354038584903826 00P0460247001300140375900 0K0460245001200430247003759330248 10
3102414038584903826 000032037600000031037580376049024820320248100000490241803302
3102474038584903826 481000001602117000N0160253603759160252403760320376000000 1403
3102534038584903826 759000034602598012001102536000021102524000021102117000014902
3102594038584903826 518026026450252426026400253612026400000013103758037604302702 0
3102654038584903826 375931037580376012021170000145026460375949021260160220500009
3102714038584903826 160276103759160277602118160278103759450277003759490283802502
3102774038584903826 118037591102761000021102776000011102781000021202205000014902
3102834038584903826 750026028800277612022050000146029060120015021180000011028800
3102894038584903826 000149028500440293002481320212500000026022050007926029720006 4
3102954038584903826 110297200004240000003403460311201200110297200 0J01202205000 Jn
3103014038584903826 470296601200260220500074260306802972210306800069240000003403
3103074038584903826 460319201200120220500 0J0470305001200482603171029722103171000
3103134038584903826 691103171000061600469031831600445000004900422021254902126026
3103194038584903826 032750306811032750000 626033110006421033110006926022050007916
3103254038584903826 004690328726011970000049019380212516004690332316004450000049
3103314038584903826 004220009911032750 0 0J0110331100 0J01202205000J04603252013 049
3103374039124902126                                                          021260
```

PRE-LOAD CARDS

```
360383800500
1503918000Z4903826
3102114038584903826 160247203940260217300064260225300079110217300OJ0260391800000
3102174038584903826 320391300000220391801822470223001200260392001826490246601603
3102234038584903826 912000N3330391300000160233703913160242901818160243603919250 3
3102294038584903826 910039183303918000002503919004004302394000001102337000011203
3102354038584903826 912000011102429000011202436000014902326026024170233731039130
3102414038584903826 000026039200181833039190000044024660391032039200000026000000
3102474038584903826 392011024720000J01202253000J0470215001200260247200064210247 0
3102534038584903826 006921024720006911021730 00J0260216100069260248900069260225 30
3102594038584903826 007416024480264816024550 00M1260252002647490215 0M902668320392
3102654038584903826 000000490246602602659000742602710000642102710000692600000018
3102714038584903826 262602871027102602847027101102847000J0160279903940260225300 0
3102774038584903826 792603920018262103920000004602884012001600469028472601197039
3102834038584903826 204901938000001600469028831600445000004900422000991102799000
3102894038584903826 J01102847000J01202253000J0470277601200110284700 K02102871000
3102954038584903826 691202659000J0460275201300370356300500250358300400340000000 1
3103014038584903826 023903563001003900049001003400000001023400000001021602659000
3103074038584903826 043703563005002503583004004703136001003903563001003400000001
3103134038734902114 0812026590000147030760120034000000010249000 24
CASE
ITER NO
FUNCTIONAL
  VAR OUT
  VAR IN
```

PRE-LOAD CARDS. DUAL ALGORITHM ROUTINE

```
360383800500
1503918000Z4903826
3102114038584903826 260002301826260344000079260218500064210218500069110218500 0J0
3102174038584903826 240002300000470222201 Q02602221021852600023000001203440000J0
3102234038584903826 470216201200440002400023260002302205160356000000260341600074
3102294038584903826 260237202221260241300064210241300069210237200069260242502372
3102354038584903826 210241300069240000001826460848601300160046902425260119700000
3102414038584903826 490120000000240009900023470248601100260002300099260352402372
3102474038584903826 260356002413120341600 0J0470233001200140356000000470254601200
3102534038584903826 48N1100000002602684035242602708035602603440000792602665035 24
3102594038584903826 160269603930260278003524260270103560160046902665260119702545
3102654038584903826 490120000000260374000099260000001826260393000000260000001826
3102714038584903826 1102696000J01102701000J01102708000J0120344000 0J0460269001300
3102774038584903826 260000002545260285702221260286402221260341600074160046902857
3102834038584903826 260119703740490193800000260000000099210285700069210286400069
3102894038584903826 1203416000J0460282201300160298403930260302500064210302500069
```

SOLUTION

PRE-LOAD CARDS

SIMPLEX ALGORITHM ROUTINE

```
3102954038584903826 160309703930260344000079220393001826460321801200260308502221
3103014038584903826 160312100000260341600074260314003121120314000008160046903097
3103074038584903826 260119700000490193803930160046903133160044500000490040200099
3103134038584903826 1400000000M3470359801300210308500069210314000069210312100069
3103194038584903826 1203416000J 460306201300110298400OJ0110309700OJ0110302500OJ0
3103254038584903826 1203440000J 460297801300260337302221260336103560120336100OJ0
3103314038584903826 260336803361220337300069260338003373260002300000260000000000
3103374038584903826 260000000023110363100001470211400100340000000010238036290010O
3103434038584903826 340000000108260357703373260354103361260350500064210350500069
3103494038584903826 260391700000380390800100340000000108260391700000380390800100
3103554038584903826 340000000108260391700000380390800100490211402603616031212600
3103614039004902114                    *                              *  000018264903158000
360383800500
1503918000OZ4903826
3102114038584903826 250366803747260002301826260354800074260219700064210219700069
3102174038584903826 210219700069240002300000470223401100260223302197260002300000
3102234038584903826 210219700069120354800OJ0470218601200440002400023260002303522
3102294038584903826 160348100000260363200079260238002233260243302233260242100064
3102354038584903826 210242100069490254202400000018264702518011001600469024332601
3102414038584903826 197000004901200000002400099000234602518011004602586012002602
3102474038584903826 493023802603746000002600023000992603481024211203632000J04602
3102534038584903826 630012001102380000J0110242100OJ0110243300OJ0490237402602609O
3102594038584903826 238024037460000047024700130049025180140348100000470266601200
3102654038584903826 48N1100000002602804024932602828022332603632000792602785024 93
3102714038584903826 160281603930260290002493260282102233160046902785260119702665
3102774038584903826 490120000000260392000099260000001826260393000000260000001826
3102834038584903826 1102816000J0110282100OJ0110282800OJ0120363200OJ0460281001300
3102894038584903826 260000002665260297703481260298403481260354800074160046902977
3102954038584903826 260119703920490193800000260000000099210297700069210298400069
3103014038584903826 120354800OJ0460294201300160310403930260314500064210314500069
3103074038584903826 160321703930260363200079220393001[illegible]4603338012002603205034 81
3103134038584903826 160324100000260354800074260326003241120326000008160046903217
3103194038584903826 260119700000490193803930160046903253160044500000490040200099
3103254038584903826 140000000OM3470370601300210320500069210326000069210324100069
3103314038584903826 120354800OJ0460318201300110310400OJ0110321700OJ0110314500OJ0
3103374038584903826 120363200OJ0460309801300260346902233120346900OJ026034760346 9
3103434038584903826 220348100069260348803481260002300000260000000000260000000023
3103494038584903826 110366700001470212600100340000000102380366500100340000000108
3103554038584903826 260368503481260364903469260361300064210361300069260374600000
3103614038584903826 380373700100340000000108260374600000380373700100340000000108
3103674038584903826 260374600000380373700100490212602603724032412600000018264903
3103734039044902114                    *                              *  27800000000OZ
```

PRE-LOAD CARDS

```
360383800500
1503918000Z4903826
3102562038584903826 360382600500490382603400100001022602653000642102653000693703
3102622038584903826 759005003903759001002603917000003400010001081803908001003400
3102682038584903826 000001023400000001021603533000063703759005003903759001001203
3102742038584903826 533000014702714012003400000001022603321000641103321000J02603
3102802038584903826 369026531103369000J026030130336921030130007926030010301311 03
3102862038584903826 001000J026029650301311029650 0K026035450007926007500266260 0
3102922038584903826 740025931600725000001600715000001603044000002603097029652603
3102982038584903826 109029651603085000001603157000002603213030132603533000742200
3103042038584903826 000037064603214012001600469030972601197000004901200000004403
3103102038584903826 166000002400099007504703214011002600750000992600730000004903
3103162038584903826 214024000990074046032140130026007400009926007200000021030440
3103222038584903826 006921030850006921030970006921031090006921031570006921032130
3103282038584903826 006912035330 0J047030380120026039180000034000000001023803909 0
3103342038584903826 010032019550000027019480000026 0-918007143400000000 1 0638039150
3103402038584903826 010026034290332126007600000 03.C07550000044034860076033007600
3103462038584903826 000032035210000049034980330352100000260076900760150077000000
3103522038584903826 330195500000320074000000270194800740260391800724340000000108
3103582038584903826 380391500100330075000000330195500000027019480075011029650 0J0
3103642038584903826 110332100 0J011033690 0J012035450 0J047029060120049000240000 0
3103702038904902562                    *                    0000070707070037070700000Z0
360383800500
1503918000Z4903826
3101948038584903826 320195600000310389803707440200801947330194700000160391700 0K0
3102008038584903826 260076003706140193900 0N44702064011001600757R999949021480120 1
3102068038584903826 939000M747021360130016021350194021021350193915019390000021 0 0
3102128038584903826 760019474402442019551602219000041602287007531602282038991602
3102188038584903826 243007531602250038993201957000004402276019574302264007531603
3102248038584903826 759000004902288033019570000025038990075311022870000111022430
3102308038584903826 000111022820000211022500000212022190000147022200120044024160
3102368038584903826 195633019560000033019570000016022190000549023120340000000108
3102428038584903826 390389900100421403917000K047024780120032007600000021007600 7
3102488038584903826 704702522013001603917000004902148016039170 0 0K049021487070707
3102548039044903826                    *                    *       00370707070000
2503919004004902582
```

PAGE HEADINGS

```
FUNCTIONALZ
 VAR/COST     Z
ACTIVITY  Z
LIM VAR      Z
LOWER LIM  Z
LIM VAR   Z
UPPER LIM     Z
```

```
360383800500
150391800002Z4903826
310256203858490382 6 340001000102L400100001021603253000063703759005003903759001 00
310262203858490382 6 120325300001470259801200340000000102260321700064210321700069
310268203858490382 6 210321700069260326503217120321700 0J0260286103217110286100 0K0
310274203858490382 6 260289702861220289700069260290902897220290900069260322400074
310280203858490382 6 260075002585260074002574160071500000160072500000160294000000
310286203858490382 6 260299302861260300502861160298100000160305300000260310902909
310292203858490382 6 260325300079220000003438460311001200160046902993260119700000
310298203858490382 6 490120000000440306200000240009900740470311001100260074000099
310304203858490382 6 260072000000490311002400099007504603110013002600750000992600
310310203858490382 6 730000001102940000J0110298100 J0110299300 J0110300500 J01103
310316203858490382 6 053000J0110310900 J0120325300 J0470293401200260391800000340 0
310322203858490382 6 000001023803909001003201955000002701948000002603918007143400
310328203858490382 6 000001083803915001002701948007402603918007243400000001083803
310334203858490382 6 915001002701948007502102861000692103217000692103265000691203
310340203881490256 2                    *       224000J0470280201200490002400000000000
 VAR/COST   Z
SHAD PRICE Z
LIM VAR      Z
LOWER LIM  Z
LIM VAR   Z
UPPER LIM      Z
360383800500
150391800002Z4903826
310256203858490382 6 470282600400310356303401310356200048150357300000390356300400
310262203858490382 6 310356303481260356700079260357200074150357300000380356300400
310268203858490382 6 260278400064110278400001230006900073320009500000210009900069
310274203858490382 6 210009800078210009902784260281300099380000000400110278400090
310280203858490382 6 140278400000470277801300410000000J00488888888888490002400002
310340103858490382 6 ----------------------------------------------------------
310346103858490382 6 ----------------------------------------------------------
310352103878490382 6                    *     ----------------------------------
250356100400250372300400490256 2
```

PAGE HEADINGS

PRE-LOAD CARDS

MATRIX PUNCH ROUTINE

COMPUTER TECHNOLOGY

Linear Programming Code for the Card 1620
with Punched Card Option for Final Output 10.1.006

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

1620 Correction

Feb.1,1965

10.1.006

0101

"Linear Programming Code for the Card 1620 with Punched card Option for Final Output"

An error in the subject program was reported to the 1620 Users Group and subsequently to us. The enclosed changes should eliminate the error.

The Right Hand Side Changer Program (cards numbered 200 - 260) should be modified to eliminate a stray flag bit left in memory position 03513. The following changes should be made.

1. Card #229 of the RHS changer program. Change cols. 1 - 7 from 4800000 to 4903484.

2. Card #259 of the RHS changer program. Change from col. 33 on to contain:
330351300000480000000000≠000000̄1̄0̄3̄4̄5̄2̄0̄3̄5̄0̄8̄0̄0̄0̄2̄5̄9

3. Card #260 of the RHS changer program. Change cols. 1 - 13 to 490212600000≠ and 63 - 80 to
0̄1̄0̄3̄5̄0̄8̄0̄3̄5̄2̄0̄0̄0̄0̄2̄6̄0

4. Add Card #261 to the RHS changer program. Card cols. and contents as follows
Col 1 - 69 - all zeros
Cols 70 - 80 - 02114000261

5. Appendix A, Section D3 should read:
"RHS changer itself loads and a stop at 03507 is executed."

A complete corrected RHS Changer deck has been distributed since

Feb.2,1965

File No. 10.1

LINEAR PROGRAMMING CODE FOR THE CARD 1620 WITH PUNCHED CARD OPTION FOR FINAL OUTPUT

Lou Davis and Art Nickel
IBM
401 Grand Avenue
Oakland, California

"Linear Programming Code for the Card 1620 with Punched card Option for Final Output"

An error in the subject program was reported to the 1620 Users Group and subsequently to us. The enclosed changes should eliminate the error.

The Right Hand Side Changer Program (cards numbered 200 - 260) should be modified to eliminate a stray flag bit left in memory position 03513. The following changes should be made.

1. Card #229 of the RHS changer program. Change cols. 1 - 7 from 4800000 to 4903484.

2. Card #259 of the RHS changer program. Change from col. 33 on to contain:
   330351300000480000000000+000000103452035080000259

3. Card #260 of the RHS changer program. Change cols. 1 - 13 to 4902126000000+ and 63 - 80 to
   0103508035200000260

4. Add Card #261 to the RHS changer program. Card cols. and contents as follows
   Col 1 - 69 - all zeros
   Cols 70 - 80 - 02114000261

5. Appendix A, Section D3 should read:
   "RHS changer itself loads and a stop at 03507 is executed."

A complete corrected RHS Changer deck has been distributed since

Feb.2,1965

DECK KEY

1. Data Loader - 000-053
2. Cost Changer - 100-107
3. RHS Changer - 200- 260
4. Solution Deck - 300-916 (not continuous)

A.	Floating Point Subroutine	300-328
B.	Shadow Price	400-423
C.	Dual	500-526
D.	Simplex	600-628
E.	Final Basis	700-720
F.	Fix and Print	750-768
G.	Non Basis	800-821
H.	Matrix Punch	900-916

5. The deck mentioned in Section I. on Page 10

PROGRAM ABSTRACT

Title: Linear Programming Code for the Card 1620 with Punched Card Option for Final Output

Subject Calssification: 10.1.006

Authors: Lou Davis and Art Nickel
IBM
401 Grand Avenue
Oakland 10, California

Direct Inquiries to: Lou Davis or Art Nickel
IBM
401 Grand Avenue
Oakland 10, California
TEmplebar 4-7070

Purpose: Solution of linear programming problems with output of detailed results. Given coefficients $a_{i,j}$, cost coefficients c_j, and requirements b_i, determine x_j such that

$$\sum_j a_{i,j} x_j = b_i \text{ with } x_j \geq 0$$

and

$$\sum_j c_j x_j = \text{maximum}.$$

Mathematical Method: Computations are performed by the Dual Algorithm until a feasible solution is obtained. Control is then given to the Simplex Algorithm for optimization. Many, many things go on before this stage is reached and after. It is quite important to read the instructions for order of program input (Appendix A) and data input carefully.

a. Accuracy: All computations are performed in 2- and -8 floating point arithmetic.

b. Derivation-Reference: Some (Nichols') notation and techniques were derived from the writeup of the "Linear Programming Code for the Augmented 650" by O.R. Perry. Reference is also made to C. R. Nichols' writeup for the 1620 paper tape input/output version.

Restrictions, Range:

a. Requires a 1622 Card Read-Punch Unit. This program was rewritten for a 20K machine. Certain changes in the program deck are necessary to enable it to run on a 40K or 60K machine. These changes are indicated in Appendix E. The size of the problem which can be handled is restricted by the following relationship:

$$(m+2)\ (n+3) \leq \frac{\text{memory} - 3920}{10}$$

where m is the number of restrictions,
n is the number of non-basis independent variables, and memory is 20K, 40K, or 60K.

b. Data must be prepared in the format specified in Appendix B.

c. Output may be either on the typewriter or on cards. The optional final matrix punchout is on cards. (See Addendum No. 1 to program writeup).

Storage Requirements:

Any size memory - see Restrictions.

Equipment Specifications:

Basic 1620 with Card input and output.

Source Language:

The original tape program and the modification for card input and output have been coded in 1620 SPS.

Program Execution Time:

The precise time required per iteration depends on the size and density of the matrix. As an approximation, a problem with 30 equations and 40 non-basis variables requires about 20 seconds per iteration.

Check Out Status:

This program has been widely and successfully used in the field for some time now on many problems of different sizes and descriptions.

COMMENTS

This program and its documentation were written by an IBM employee. It was developed for a specific purpose and submitted for general distribution to interested parties in the hope that it might prove helpful to other members of the data processing community. The program and its documentation are essentially in the author's original form. IBM serves as the distribution agency in supplying this program. Questions concerning the use of the program should be directed to the author's attention. **For certain problems it has been found that the value for "essential zero" located in memory positions 3144 and 3145 may need to be increased to some value greater than the 10-8 value to which it is set in the program.**

TABLE OF CONTENTS

1620 LINEAR PROGRAMMING CODE FOR CARD INPUT/OUTPUT

Nichols, Nickel and Davis

Addendums to 1620 Linear Programming Code - File Number 10.1.006

I. To avoid an overflow condition and certain loading inefficiency when using the Data Loader program, make the following change in the program deck and listings for the Data Loader:

Object Program

Card No.	Location	From	To
32	01946	4902022	4902090

Symbolic Listing (Data Loader)

Label	Location	From	To
TSTRC + 12	01946	B Fill	B Signst

The subject "overflow" causes no apparent error but only slows down the execution of the program. If the above change is made, data cards must not contain record marks.

II. A second item that it would be well to note is the following:
When using negative (minus) data values, the minus sign (11 punch) must be in the first and only first column of the ten (10) column field allowed for the number. That is, column 9, 29, 49 and/or 69.
If this is not followed, the value will be treated as positive.

III. To convert the "final output - punch option" deck for use on a 40K or 60K 1620, the following cards and columns should be changed in the manner prescribed in the program writeup:

Card Number	Columns
700	3 and 51
701	3, 15, and 39
702	27 and 39
712	47
718	43
800	39 and 51
814	23
912	15

These columns contain ones (1's) as the first digit of 19xxx addresses and should be changed to threes (3's) or fives (5's) for the 40K or 60K machine.

I. Reference: Item 3. "Restrictions", part c.

The Final Basis and Non-Basis Output may be punched on cards rather than typed by replacing cards numbered 700 through 916 of the solution deck with a different card deck bearing the same first and last number (700 and 916) and approximately the same numbers in between. (These cards are the last section in the deck.)

II. Reference: Appendix C - Output
Items 2 and 3

2.1 and 3.1
Final Basis and Non-Basis Output using "Punch Deck" for cards 700 - 916. The final value of the profit function is typed in floating point form as before.

The same information is punched as is given with the typed output. The card format is as follows:

Card Columns

1 - 2	Code 11 - Final Basis 22 - Non-Basis
3 - 10	Zeros
11 - 20	Identification/Cost Coefficient
21 - 24	Zeros
25 - 32	Activity - Final Basis or Shadow Price - Non-Basis
33	Record Mark 0 - 2 - 8 punch
34	Zero
35 - 38	Limiting variable

39 - 40 Zeros

41 - 48 Lower Limit on Cost - Final Basis
or Lower Limit on Activity - Non-Basis

49 Record Mark 0 - 2 - 8 punch

50 Zero

51 - 54 Limiting variable

55 - 56 Zeros

57 - 64 Upper Limit on Cost - Final Basis
or Upper Limit on Activity - Non-Basis

65 Record Mark 0 - 2 - 8 punch

66 - 80 Zeros

Notes:

1. The card output is numeric so there are no decimal points or minus signs.

2. Negative numbers are denoted by a flag (11 punch) over the low order digit i.e., card columns 48 or 64.

3. The Activity, Shadow Price and Limits are 8 digit numbers, and the decimal point is assumed in the middle -- that is, 4 positions from the right or left, so (xxxx.xxxx).

4. Card columns shown as containing zeros may contain "most anything" on the first card punched, but all columns of any interest will be correct.

G. N. Nickel

1. Identification:

a. 1620 Linear Programing Code for card input/output.

b. This program is an amended version of C. R. Nichols' 1620 Linear Programing Code for paper tape input/output. Most of his original program and much of his writeup have been retained unchanged for this edition. This version edited by Art Nickel and Lou Davis.

c. IBM Branch Office, Oakland, California.

2. Purpose: Solution of linear programming problems with output of detailed results. Given coefficients $a_{i,j}$, cost coefficients c_j, and requirements b_i, determine x_j such that

$$\sum_j a_{i,j}x_j = b_i \quad \text{with } x_j \geq 0$$

and

$$\sum_j c_j x_j = \text{maximum.}$$

3. Restrictions:

a. Requires a 1622 Card Read-Punch Unit. This program was rewritten for a 20K machine. Certain changes in the program deck are necessary to enable it to run on a 40K or 60K machine. These changes are indicated in Appendix E. The size of the problem which can be handled is restricted by the following relationship:

$$(m+2)(n+3) \leq \frac{\text{memory} - 3920}{10}$$

where m is the number of restrictions,

n is the number of non-basis independent variables, and

memory is 20K, 40K, or 60K.

b. Data must be prepared in the format specified in Appendix B.

c. All output is on the typewriter except an optional final matrix punch out which is on cards.

4. Method: Computations are performed by the Dual Algorithm until a feasible solution is obtained. Control is then given to the Simplex Algorithm for optimization. Many, many things go on before this stage is reached and after. It is quite important to read the instructions for order of program input (Appendix A) and data input carefully.

a. Accuracy: All computations are performed in 2-and-8 floating point arithmetic.

b. Derivation-Reference: Some (Nichols') notation and techniques were derived from the writeup of the "Linear Programming Code for the Augmented 650" by O. R. Perry. Reference is also made to C. R. Nichols' writeup for the 1620 paper tape input/output version.

5. Miscellaneous:

a. Special Features:

(1) Cost changes in terms of the original data can be made without the necessity of reloading or resolving the original matrix.

(2) Requirements can also be changed--as in (1) above.

(3) The overall program consists of several relatively independent subprograms which can be deleted or altered in many respects without interference with other subprograms. Sequential loading of the subprograms is automatic, with intermediate programmed stops at points where it might be desireable to manipulate order of input decks.

APPENDIX A

OPERATOR INFORMATION

A. Preliminary:

1. Set typewriter margins at 15 and 95; tabs at 27, 39, 51, 63, and 72.

B. Load Matrix:

1. Clear Memory -- turn off all console switches.
2. Place DATA LOADER deck followed by data deck in 1622 Card Reader.
3. Make sure computer is in manual mode.
4. Depress LOAD on card-reader.
5. DATA LOADER itself loads and a halt at 00413 is executed. (All addresses given for halts will be the address displayed in the Memory Address Register.)
6. Depress START on console.
7. If all goes well the data will load and a normal halt at 02265 will be executed.
8. Operation may now proceed to the COST CHANGER, RHS CHANGER, or SOLUTION.

C. Cost Changes:

1. Place COST CHANGER deck (followed by cost changes if input is by cards) in card-reader hopper.
2. Press START on console and on card-reader.
3. COST CHANGER itself loads and a stop at 02125 is executed.
4. a. If cost changes are on cards make sure Sense Switch 2 is off and depress START on console.

b. If cost changes are to be entered from the typewriter, turn Sense Switch 2 on. (RELEASE and START must be depressed after

each entry--don't forget last entry of 0000¢.)

5. If all goes well data will load and program will branch to the subroutine to load the next program in the card hopper. If there is no program in the hopper, the computer will hang up on a READER NO FEED with a MAR address of *19851. In this case put the appropriate deck in the hopper and push START on the reader.

   It is possible for all not to go well because a change has been entered for a non-existent identification number. In this case the computer will halt at 02457.

   a. Error Procedure:

      (1) If data are being entered through the typewriter, depress RESET and INSERT, type 4902126, depress RELEASE and START. Corrected or new change data may now be entered normally.

      (2) If data are being entered through the card reader, NON PROCESS RUNOUT cards in hopper, find offending card, correct it, put cards back in hopper at that point (i.e., the corrected card followed by cards yet to be processed), press START on card reader, depress RESET and INSERT, type 4902126, depress RELEASE and START.

      Alternatively, after card(s) has been corrected it is possible to reload the entire deck, including the COST CHANGER deck, and go through the procedures in (2) except type 4*19840.

6. Operation may now proceed to RHS CHANGER, or SOLUTION (or COST CHANGER again).

D. Right Hand Side (Requirement) Changes:

   a. Place RHS CHANGER deck (followed by RHS change data deck if input is

A-2

by cards) in card reader hopper.

2. Press START on console and on card reader.

3. RHS CHANGER itself loads and a stop at 02125 is executed.

4. a. If RHS changes are on cards make sure Sense Switch 3 is off and depress START on console.

   b. If RHS changes are to be entered from the typewriter, turn Sense Switch 3 on, and press START on console. (RELEASE and START must be depressed after each entry--don't forget last entry of 0000¢).

5. If all goes well the data will load and the program will stop at 03475.

   Just like the COST CHANGER it is possible for strange happenings because a change has been entered for a non-existent identification number. In this case the computer will halt at 03201.

   a. Error Procedure:

      (1) If data are being entered through the typewriter, depress RESET and INSERT, type 4902126, depress RELEASE and START. Corrected or new change data may now be entered normally.

      (2) If data are being entered through the card reader, NON PROCESS RUNOUT cards in hopper, find offending card, correct it, put cards back in hopper at that point (i.e., the corrected card followed by cards yet to be processed), press START on card reader, depress RESET and INSERT, type 4902126, depress RELEASE and START.

      No alternate, ~~unless you take out cards already processed. (See above C, 5., a., (2)).~~

6. Operation may now proceed to COST CHANGER or SOLUTION (or RHS CHANGER again).

A-3

E. Solve the Problem:

1. Place SOLUTION deck in card reader hopper.
2. Set Sense Switch 1 on to type iterations, off to omit iteration typeout.
3. Set Sense Switch 4 on to obtain punchout of final matrix, off to suppress punching.
4. Depress START on console (and if needed on card reader).
5. Error Stops:
   a. 02545 Dual Algorithm-Inconsistent Matrix
   b. 02665 Simplex Algorithm-Unbounded Solution

   It may be possible to obtain the solution existing at the time of either of these stops by NON PROCESS RUNOUT the card reader, put back in the hopper that portion of the SOLUTION deck beginning with card number 700, RESET and INSERT, type 4919840, RELEASE and START on the console and START on the card reader.
6. After all output is complete, a halt at location 02845 will be executed. Operation may now proceed to DATA LOADER, COST CHANGER, or RHS CHANGER.

F. Other Stops:

01137 Floating Point Overflow

01299 Floating Point Attempted Division by Zero

While overflow causes a halt at either of the two above addresses, underflow causes the result to be set to zero, and processing continues.

APPENDIX B

INPUT FORMAT

Prologue: There are three basic types of data input:

Input for DATA LOADER (A below)

Input for COST CHANGER (B below)

Input for RHS CHANGER (C below)

A. Input for DATA LOADER:

1. First Data Input Card - Header Card. The first card of the data deck must be a card containing the case identification. Only the first five columns (columns 1-5) are read by the program. Explicitly, columns 1-5 of the first data card may contain any alphanumeric characters to identify the problem being run.
2. Second Data Input Card - Parameter Card. The second data card must contain certain information needed by the program. Namely,

Columns	Information	Example
1-3	Number of restrictions.	006
4	Blank	
5-7	Number of non-basis variables	008
8	Blank	
9	Input type code (one digit) as follows: 1 for row-column, fixed point input 0 for a complete (include zeros) floating point matrix	

3. The data, following the above two cards, can be in one of two formats:
   a. Row-Column, Fixed Point Input: Up to four elements with row-column designations may be entered on each card for this type of input.

In general, the format is as follows:

```
i   j   element
XXXbXXXbXXXXXXXXXX
```

where b stands for blank column.

There are four fields per card for data:

	Columns for i	Columns for j	Columns for element
Field 1	1-3	5-7	9-18
Field 2	21-23	25-27	29-38
Field 3	41-43	45-47	49-58
Field 4	61-63	65-67	69-78

All other columns are not used by the program and may be used for anything else.

ANY field which has the middle digit of the three digit i value left blank will be ignored and the other columns of that field may be used for anything else. Hence, data cards do not have to have a complete set of four elements, nor is it required that the left hand fields be used first, nor is it required that consecutive fields be used.

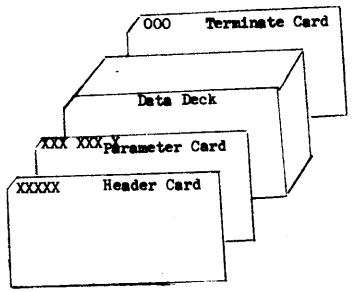
→ Matrix elements may contain up to eight numeric digits with optional leading sign and mandatory decimal point. Loading of zero elements is optional.

IMPORTANT: The signal to the DATA LOADER to quit loading is a three (zeros) digit value of i=000. This must be in the last data card in a field to/the right (if other values on the card) of the last element. The best thing is to have a card with 000 punched in columns 1-3 and then have this card as the last card in the data deck.

(Fixed Point Input continued)

Item	Description	Row-Column	Remarks-Examples (i j matrix element)
$a_{i,j}$	Matrix element in the ith row and of the jth non-basis vector.	XXX XXX	010 004 36.075 003 004 -4.0 008 017 -.6 008 017 -0.6
b_i	Element in the ith row of the requirement vector j = 000	XXX 000	008 000 6.025
c_i	Cost per unit of the basis variable for row i j = 00$\bar{1}$ = 00J where J is a flagged one.	XXX 00J	Expressed in fixed point with the following format: ID COST XXX 00J XXXXXXXXX Decimal point in cost assumed here. Negative cost denoted by flag (11 punch) over units position of cost. 012 00J 0003001500 Here cost is 1.5 012 00J 0016005999 Here cost is -5.999
c_j	Cost per unit of the jth non-basis vector i = 00$\bar{1}$ = 00J where J is a flagged one.	00J XXX	Remarks same as above. 00J 024 0034123456 Here cost is 123.456

Summary of DATA LOAD input for row-column, fixed point input.



b. Input for Floating Point Matrix: In the floating point mode, the entire matrix (including zero elements) is loaded in column order. (See storage layout in Appendix E). Seven ten digit floating point numbers must be punched in matrix/column order in each card in columns 1-70. Elements of a new matrix column follow immediately the last element of the old matrix column (i.e., a new matrix column does not automatically start a new card). A record mark (0,2,8) punch must be placed in column 71 of each card. If the last card does not require a full 7 elements, the record mark must instead follow directly behind the last digit of the last element. ALSO, the last card must have a record mark in column 72 (or else you will not terminate loading). Columns 73-80 you have for your own pleasure. The floating point representation is the same as used in 1620 SPS. Note that the optional matrix punch out of the SOLUTION deck prepares a datadeck in this format.

APPENDIX C

OUTPUT

1. Results of iterations are typed only if Sense Switch 1 is on. If it is desired to monitor the course of the solution only occasionally, this may be done by setting Sense Switch 1 on and off periodically. The following points should be borne in mind:

   a. Page headings for iteration output are typed only if Sense Switch 1 is on at the beginning of the solution.

   b. The iteration count is reset when the dual algorithm is finished and control passes to the simplex algorithm.

   The information which may be typed includes the iteration count, the current value of the profit function (floating point), the last variable to leave the basis, and the variable which entered the basis.

2. Final Basis Output:

   The final value of the profit function is given in floating point.

   For each final basis variable, the following information is supplied (in fixed point):

   a. Identification/cost coefficient.

   b. Activity.

   c. The limits of the cost coefficient over which the current solution is optimal.

   d. The variables which limit the range of the cost coefficient and will enter the basis if a limit is exceeded.

3. Final Non-Basis Output:

   For each final non-basis variable, the following information is given (in fixed point):

a. Identification/cost coefficient.

b. Shadow price - the penalty to the total system if a unit of this variable is forced into the final solution.

c. The limits of activity over which the shadow price applies.

d. The variables which limit the range of applicability of the shadow price. The uper limiting variable is the one which would leave the basis if the associated non-basis variable were forced into the solution.

5. A punchout of the complete final matrix is optional under control of Sense Switch 4. The deck punched out contains the Header and the Parameter cards and is in a format which is suitable for direct reloading by the DATA LOADER routine.

B. Input for COST CHANGER:

1. Cards: One element only per card. For each cost coefficient to be changed, a ten digit ID/new cost element is punched in columns 1-10 in the following format.

ID New Cost

XXXXXXXXXX

Decimal point assumed here.

Negative cost is signified by a flag over the units positions.

2. Typewriter: As above--ten digits typed in without any spacing between digits, but immediately followed by a record mark.

3. IMPORTANT: Termination of loading is recognized by a field of four zeros, 0000. These are punched in columns 1-4 if entry is by card or typed in as 0000# if entry is by typewriter.

C. Input for RHS CHANGER:

For each requirement element to be changed, a four digit identification and the change amount are entered. The identification entered is that for the original basis variable for the row whose requirement is to be changed. The identification format is XXXX.

The second entry for each change is the amount by which the original requirement is to be changed. Its format is XXXXXXXXXX.

1. Cards: One element only per card.

Columns 1-4	Columns 5-8	Columns 9-18	Example	
ID	Blank	Change Amount		
XXXX	bbbb	XXXXXXXXXX	0012	-0.75

The change amount may be any number (fixed point) up to eight digits with mandatory decimal point and optional leading sign.

but a record mark must immediately follow each entry.

2. Typewriter. Same as above/ RELEASE and START will have to be depressed between entry of the ID and the Change Amount.
3. IMPORTANT: Termination of loading is recognized by a field of four zeros, 0000. These are punched in columns 1-4 if entry is by card or typed in as 0000%if entry is by typewriter.
4. Cost changes and/or requirement changes may be made in unlimited numbers on any single run. If both types of changes are to be made, the order in which the two changers are operated is of no consequence. It should be noted that changes may be initiated under the procedures given in Appendix A either immediately after loading a matrix or at the conclusion of the output phase. If the output phase is interrupted before its completion, however, the computer is not conditioned to accept changes.

25

APPENDIX D

SAMPLE PROBLEM

$$\begin{aligned} Y_1 + 3Y_2 &\geq 6 \\ 2Y_1 + Y_2 &\geq 4 \\ -Y_1 + 3Y_2 &\leq 10 \\ Y_1 + Y_2 &\leq 6 \\ Y_1 + Y_2 &\leq 2 \\ Y_1 + 2Y_2 &= \min \\ Y_1 \geq 0 \quad Y_2 &\geq 0 \end{aligned}$$

This is converted to –

$$\begin{aligned} Y_1 + 3Y_2 - Y_3 + Y_5 &= 6 \\ 2Y_2 + Y_4 - Y_4 + Y_6 &= 4 \\ -Y_1 + 3Y_2 + Y_7 &= 10 \\ Y_1 + Y_2 + Y_8 &= 6 \\ Y_1 - Y_2 + Y_9 &= 2 \\ -Y_1 - 2Y_2 &= \max \\ Y_1 \geq 0 \quad Y_2 &\geq 0 \end{aligned}$$

and finally changed to:

$$\begin{aligned} Y_1 + 3Y_2 - Y_3 + Y_5 &= 6 \\ 2Y_2 + Y_4 - Y_4 + Y_6 &= 4 \\ -Y_1 + 3Y_2 + Y_7 &= 10 \\ Y_1 + Y_2 + Y_8 &= 6 \\ -Y_1 - Y_2 + Y_9 &= 2 \\ -Y_1 - 2Y_2 - 999.999Y_8 - 999.999Y_9 &= \max \\ Y_1 \geq 0 \quad Y_2 &\geq 0 \end{aligned}$$

to force out artificials.

SAMPLE CASE DATA (80-80 LISTING OF DATA CARDS)

```
TEST1
005 004 1
001 001 1.          001 002 3.          001 003 -1.          002 001 2.0
                    002 002 1.0         002 004 -1.0
003 001 -1.                             003 002 3.
                                                             004 001 1.0
004 002 1.
005 001 1.
005 002 -1.
001 000 6.
002 000 4.
003 000 10.
004 000 6.
005 000 2.
001 00J 0005999999R  002 00J 0006999999R  003 00J 0007000000  004 00J 0008000000
                    005 00J 0009000000
00J 001 0001001000                      00J 002 0002002000   00J 003 0003000000
                                                             00J 004 0004000000
000
```

This does NOT imply that the requirement vector is one per card. Any or all of the four fields per card may be used. Also, it is not necessary for elements to be in their row-column sequence

TYPEWRITER OUTPUT FOR SAMPLE CASE

```
CASE TEST1

ITER NO     FUNCTIONAL     VAR OUT      VAR IN
0001        5420039980     0005999995   0002002000
0002        5144000000     0006999995   0001001000
FUNCTIONAL  5144000000

 VAR/COST      ACTIVITY  LIM VAR     LOWER LIM  LIM VAR   UPPER LIM

0002002000       1.6000  0004          3.0000-  0003         .5000-
0001001000       1.2000  0003          4.0001-  0004         .6667-
0007000000       6.4000  0003           .4285-  0004         .1666
0008000000       3.2000  0004           .5000-  0006      2499.4975
0009000000       2.4000  0004           .2500-  0003         1.0000

 VAR/COST    SHAD PRICE  LIM VAR     LOWER LIM  LIM VAR   UPPER LIM

0006999995      999.7990  0009          3.0000-  0001         2.0000
0005999995      999.3989  0007          4.5714-  0002         4.0000
0003000000        .6000   0002          4.0000-  0007         4.5714
0004000000        .2000   0001          2.0000-  0009         3.0000
```

This output optional depending on Sense switch 1.

RHS CHANGE SAMPLE DATA

```
0009    1.
0004    -1.5
0000
```

COST CHANGE SAMPLE DATA

```
0001001000
0002002000
0000
```

29

D-4

APPENDIX E

STORAGE LAYOUT AND LISTINGS

1. STORAGE USED:

Memory locations from 00048 through 3747 are used by the program. Locations 19840-19963 are used for a program load routine. Locations 19964, 19965-00044 are used for a card input area for data, headings, and program input (it is, however, possible to use 00000-000047 for inserting). Memory from 3749 up to the required location (or *19839) is used for the matrix and its manipulation. The matrix is stored in column sequence and uses memory from 03749 upward. The following diagram indicates the manner in which matrix elements are stored:

3760			S+10(2m+3)	ID/Cj	S+10((n+1)(m+2)+1)
Working Column	S	Functional	S+10(2m+4)		S+10((n+1)(m+2))
	ID/Ci	S+10(m+3)	S+10(2m+5)		S+10((n+1)(m+2)+1)
		bi	Increasing i → Increasing j; $a_{i,j}$: S+10((j+1)(m+2)+i)		
3760+10(m+1)	S+10m	S+10(2m+2)	S+10(3m+4)		S+10((n+2)(m+2)-2)

30

S is computed for each problem and is equal to 03760 + 10(m+3). All addresses are field addresses.

2. PROGRAM ORDER (Card numbers are in columns 78-80 of program decks)

a. DATA LOADER 000-053

b. COST CHANGER 100-107

c. RHS CHANGER 200-260

E-1

d. SOLUTION 300-916 (Not continuous)

(1) Floating Point Subroutines	300-328
(2) Shadow Price	400-423
(3) Dual	500-526
(4) Simplex	600-628
(5) Final Basis	700-720
(6) Fix and Print	750-768
(7) Non Basis	800-821
(8) Matrix Punch	900-916

3. 80-80 listings, with remarks, of the machine language program decks follow. These cards are in a format similar to the condensed output. Specifically:

Columns	Contents
1-60	Instructions or constants
61	# (record mark) if full 60 columns are used. Other wise the record mark is at end of the data being punched.
62	1
63-64	01 Loader Code
65-69	Address of where the first character on the card is to be placed in memory.
70-74	Address of where the last character will fall in memory
75	Flag
76-77	Not used
78-80	Card sequence numbers

This listing was made on a 407 Accounting Machine. Since the 407 does not have flags, flagged digits will appear as alphabetical characters in the J-R range. A flagged zero will appear as a - (minus) or a zero. (The 407 was a bit recalcitrant.) A record mark prints out as a Z.

E-2

4. In order to allow this program to run on a machine with greater than 20K memory, certain changes have to be made. Any digit in the attached list that is marked with an * must be changed. In each case this digit will be a 1 (one) or a J (flagged one). This 1 (or flagged 1) must be changed to a 3 (or flagged 3) for 40 K memory, or a 5 (or flagged 5) for 60K. I. e., add two to the digit for each additional 20K of storage added.

E-3

DATA LOADER Card Nos 000 - 053

```
360007200500360020100500440001200275260005900274250001100000260009000269000000000 First Card DATA LOADER
260009500264310000000200260011400274250000000011490001200008000000000000 - 1
480000000000160231300001370231500500250232500400310004802314Z00100402004620000002
150374200000360231400500250231700400310374302314210053203745Z00100462005220000003
160006403790150374600000260007903746110374600OK0260006903746Z00100522005820000004
250232100400310374302318210062803745160007400000250374502322Z00100582006420000005
430073403745210068400064360231400500031000010231411006840000P0Z00100642007020000006
→ 450066602385480000000000491984000000Z00000000000000 -100702007380000007
370231500500250235100400250239100400250243100400250247100400Z00100734007940000008
430095002316140231300004470087801200160231300001160092502316Z00100794008540000009
160094902314490073400000110231300001110092500040110094900040Z00100854009140000010
430093802316490080600000310231402314250374002315250374102317Z00100914009740000011
250374202319320231800000140231900OP0460103401300320374200000Z00100974010340000012
250374302323250374402325250374502327320232600000140232700OP0Z00101034010940000013
460111801300320374500000330231800000330232600000320374000000Z00101094011540000014
140374200000460213401200320374300000230374500069210009900069Z00101154012140000015
330009900000210009803742210009800063490227400000260212000099Z00101214012740000016
440147403745160231100000250041602331110130400001110130900002Z00101274013340000017
110231100001140231100OJ0470129801200320234800000140234900OP0Z00101334013940000018
460141801300320042500000330235000000160130400416160130902331Z00101394014540000019
320041600000490211400000Z000000 0 -10145401478000020
440149403742490128600000Z000000 0 -10147401498000021
250235100400310372602330320372600000140372700OP0460165401300Z00101494015540000022
```

The data load routine reads and stores parameters for the problem. It then proceeds to load and store the matrix in the proper format. In the case of fixed point input, storage locations are computed from the row/column designations, and the input elements are converted to floating point notations as they are read and stored.

E-4

```
1403727000K0460163401200430165403727330166500000320372800000Z00101554016140000023
310372603728490166600000Z000000 0 -10161401638000024
320166500000490160200000Z000000 0 -10163401658000025
330166500000160041700ON0160172003727160170803728320170200000Z00101654017140000026
140171400003460178201200110172000002110170800002110041700001Z00101714017740000027
490170200000Z0000000000000 -10177401786000028
260182901708260182401720120182400001310181801818430188603727Z00101782018420000029
310372603728120041700001450183003727490080600000Z0000000000000010184201890000030
160146500009160194503727160196000418160196503727450195400000Z00101886019460000031
490202200000Z000000000000 -10194601958000032
250000000000110194500002110196000001110196500002120146500001Z00101954020140000033
490193400000Z000000000000 -10201402026000034
260206401960120146500001460209001200150205800000110206400001Z00102022020820000035
490203400000Z0000000000000 -10208202094000036
440211401665320042500000260211400425490080600000Z0000000000000010209002138000037
260221200064120221200OJ923000690007332000950000021000990006 9Z00102134021940000038
210009900069320220600000110221200OJ012000990000J04702206012002Z00102194022540000039
→ 480000000000491984000000Z000000 -10225402278000040
330374300000320009500000490126200000Z00000000000000 -10227402310000041
0000Z000000000000000 1010231002314000042
480000000000Z000000000000 -10231402326000043
→ 361996500500441994400039261988700038251985119876261990600033Z001J9840J9900000044
→ 311990019965261993000038251992419851491984000000Z0000000000000001J9900J9948000045
```

E-5

```
→ 261996200038491995600000Z000000 0                                    -1J9944J9968000046
  L60000000500490000Z00000000000                                       -10009600115000047
  3600100005003600172005003600244005003600316005003600000005000000000000000000 0 48
  0000000000001020304000204060800030609021004080216100500151020060218142002Z0000049
  704112820080614223009081726300000000005060708090012141618151811242720242Z0000050
  822363520353045403632484455324946536048465462754453627180123456789123456Z0000051
  789023456789OJ34567890JK4567890JKL567890JKLM67890JKLMN7890JKLMNO890JKLMNZ0000052
  410000000000490040200P90JKLMNOPQZ0000L10038800019M900000000000M9000360000000000053   Last Card  DATA LOADER
```

35

COST CHANGER Card Nos 100 - 107

```
  480000000000460215800200360251100500490218203400000001023602Z101021140217400010O   First Card  COST CHANGER
→ 5110010032025110000014025140000046198400120049022540020034002Z101021740223400010 1
  0000010238037370010026021930007926022960006411022960000424002Z101022340229400010 2
  0000251446024480120011022960000J01202193000J04702290012002602Z101022940235400010 3
  1930007426023920229621023920006924000000251446024680120021022Z101023540241400010 4
  3920006912021930000J04702386012004826024980229649024800260249Z101024140247400010 5
  8023921102498000062600000025204902126ZZ0000000000000000000000  1010247402512000106
  000000000000                                                    -2114000107  Last Card  COST CHANGER
```

The cost change routine reads ID/cost elements, searches the matrix for a corresponding identification number, and stores the new cost coefficient in the proper location.

36

RHS CHANGER Card Nos 200 - 260

```
160053600538490043401600536005582601197000002601112004451601Z10100402004620002000
124000001101124000014600494014004901128004692601176000002601Z10100462005220002010
187018274900000022011860117649005700260118601176150084700001Z10100522005820002020
160085701187260109201189220117801189470070201100260117601197Z10100582006420002030
260119701186260118601176210109201189260117801189320117800000Z10100642007020002040
150119800000140117800000470109401300440077401197320119800000Z10100702007620002050
330119700000210085701178440082201186330118600000150084700002Z10100762008220002060
150117800000150118900000210119800000250109301198330119800000Z10100822008820002070
430098601189430103401190460096601200310119001191250119800557Z10100882009420002080
120109200001460089401300260119701826490110602601198011973301Z10100942010020002090
190000001101092000014601126014001501189000001101198000054300Z10101002010620002100
986011894401094010933201197000002601189010922600000011974900Z10101062011220002110
000048260115100469260051700000490050600000000000000000  0   Z10101122011820002120
00000000000000000ZZ000000000000                              10101182012000002130
160053601256150175300002160111200099320119000000490045804301Z10101200012600002140
276011904900966043012900116948220118901168250132501176330117Z10101260013200002150
600000150116800000160134700000110134700001210117601176430139Z10101320013800002160
401168490135001601441018372501428011691101441000002301175000Z10101380014400002170
002601477014411201477000042601166000002201166000933201162000Z10101440015000002180
002301176011652601186019372201186000963201178000002301186011Z10101500015600002190
863300082000001500081000OJ2100090011862601186000902301186011Z10101560016200002200
652100095000951201347000014701622012002601186000944401694013Z10101620016800002210
253201186000002301186011971101189000N025010930009926010920112Z10101680017400002220
```

First Card - RHS CHANGER

The requirement change routine reads variable identification numbers and changes for the associated requirement element. The change is converted to floating point notations, the matrix is searched for the identification number, and the requirement vector is updated.

37

E-8

```
892601198000924601774013004900966047008820140049011260000000Z10101740018000002230
0n000000000000  -          J90692R080J74077P568J60128O405J4Z10101800018600002240
8249N490J38013M759J29099M164J21267L674J14332L266J08147K922J0Z10101860019200002250
2597K630J0000000001600536019701501753000034901224043019900l1Z10101920019800002260
904900966043020100116949009660320116900000120116800ON1230119Z10101980020400002270
701176210118901168490171822000000000                        10102040020660002280
480000000000460221800300370348500500250229203485250229303487Z00102114021740002290
250229403489250229503491250229600400490224200000Z0000000000000010217402222000230
340000000102360229200100320229200000140229500000460346401200Z00102218022780002310
470232200300340000000108370372700100490241400000Z0000000000000010227802326000232
320350000000140350100OP0460239001300430239003501250352100400Z00102322023820002330
490240200000Z000000000000                                   -10238202394000234
250351900400310372603500320372600000140372700OP0460255001300Z00102390024500002350
140372700OK0460253001200430255003727330213700000320372800000Z00102450025100002360
310372603728490256200000Z000000 0                           -10251002534000237
320213700000490249800000Z000000 0                           -10253002554000238
330213700000160211700ON0160261603727160260403728320259800000Z00102550026100002390
140261000003460267801200110261600002110260400002110211700001Z00102610026700002400
490259800000Z000000000000                                   -10267002682000241
260272502604260272002616120272000001310271402714430278203727Z00102678027380002420
310372603728120211700001450272603727490212600000Z0000000000000010273802786000243
160225300009160284103727160285602118160286103727450285000000Z00102782028420002440
490291800000Z000000000000                                   -10284202854000245
```

38

E-9

```
250285002850110284100002110285600001110286100002120225300001Z00102850029100002460
490283000000Z000000000000                                    -10291002922000247
260296002856120225300001460298601200150295400000110296000001Z00102918029780002480
490293000000Z000000000000                                    -10297802990000249
440301002137320212500000260225300079260305200064110305200004Z00102986030460002500
240304602295460319201200110305200OJ01202253000J04703046012002Z00103046031060002510
260225300074260314803052210314800069240314202295460327201200Z00103106031660002520
1202253000J04703130012004800000000002Z00000000000000        -10316603202000253
260325103052210325100069110325100006160046903263160044503240Z00103192032520002540
490042202125490212600000Z000000 0                            -10325203276000255
260335503148110335500OJ62603391000642103391000692602253000079Z00103272033320002560
160046903367160119703344490193802125160046903403160044503380Z00103332033920002570
490042200099110335500OJ01103391000J01202253000J04603332013002Z00103392034520002580
490212600000480000000000491984000000Z00000000000000        -10345203488000259
000000000000                                                 -2114000260    Last Card  Rtn Character
```

E-10

SOLUTION Cards Nos. 300 - 916 (Not continuous)

```
160053600538490043401600536005582601197000002601112004451601Z10100402004620003001 First Card
124000001101124000014600494014004901128004692601176000002601Z10100462005220003011 First Card
187018274900000022011860117649005700260118601176150084700001Z10100522005820003021
160085701187260109201189220117801189470070201100260117601197Z10100582006420003031
260119701186260118601176210109201189260117801189320117800000Z10100642007020003041
150119800000140117800004701094013004400774011973201198000002Z10100702007620003051
330119700000210085701178440082201186330118600000150084700002Z10100762008220003061
150117800000150118900000210119800000250109301198330119800000Z10100822008820003071
430098601189430103401190460096601200310119001191250119800557Z10100882009420003081
120109200001460089401300260119701826490110602601198011973301Z10100942010020003091
190000001101092000014601126014001501189000001101198000054300Z10101002010620003101
986011894401094010933201197000002601189010922600000011974900Z10101062011220003111
000048260115100469260051700000490050600000000000000000000 0 Z10101122011820003121
0000000000000000ZZ0000000000 0                               1010118201200003131
160053601256150175300002160111200099320119000000490045804301Z10101200012600003141
276011904900966043012900116948220118901168250132501176330117Z10101260013200003151
600000150116800000160134700000110134700001210117601176430139Z10101320013800003161
401168490135001601441018372501428011691101441000002301175000Z10101380014400003171
002601477014411201477000042601166000002201166000933201162000Z10101440015000003181
002301176011652601186019372201186000963201178000002301186011Z10101500015600003191
863300082000001500081000OJ21000900118626011860009023011860112Z10101560016200003201
652100095000951201347000014701622012002601186000944401694013Z10101620016800003211
253201186000002301186011971101189000N02501093000992601092011Z10101680017400003221
```

E-11

```
8926011980009246017740130049009660470088201400490112600000002101017400180000323
000000000000-     -              J90692R080J74077P568J601280405J4Z101018000186000324
8249N490J38013M759J29099M164J21267L674J14332L266J08147K922J0Z101018600192000325
2597K630J000000000016005360197015017530000349012240430199001IZ101019200198000326
90490096604302010011694900966032011690000012011680000N1230119Z101019800204000327
701176210118901168490171 8ZZ00000000                    1010204002066000328  Last Card - Floating Point Subroutines
160247203780260217300064260225300079110217300OJ026037580000OZ101021140217400 0400  First Card - Shadow Price
320375300000220375801822470223001200260376001826490246601603Z101021740223400 0401
752000N33303753000001602337037531602429018181602436037592503Z101022340229400 0402
750037583303758000002503759031554302394000001102337000011203Z101022940235400 0403
752000011102429000011202436000014902326026024170233731037530Z101023540241400 0404
000026037600181833037590000044024660375032037600000026000000Z101024140247400 0405
37601102472000J01202253000J04702150012002602472000642102472OZ101024740253400 0406
00692102472000691102173000J02602161000692602489000692602253OZ101025340259400 0407
00741602448026481602455000M126025200264749021 50M902668320376Z101025940265400 0408
000000490246602602659000742602710000642102710000692600000018Z101026540271400 0409
26260287102710260284702710110284700OJ016027990378026022530000Z101027140277400 0410
79260376001826210376000000460288401200160046902847260119703 7Z101027740283400 0411
60490193800000160046902883160044500000490042200099110279900OZ101028340289400 0412
J01102847000J01202253000J04702776012001102847000K021028710002101028940295400 0413
691202659000J04602752013003619964005003400000001023919965001Z101029540301400 0414
0039000490010034000000010234000000010216026590000436199640052101030140307400 0415
0047031120010039199650010034000000010812026590000147030640122101030740313400 0416
```

E-12

```
0034000000010249198402Z00000000                  1010313403156000417
0000000000000                                    -2114000418
43416245000Z0000000000000                        419
49634559005556OZ000000000000                     420
46645543634956554153OZ000000000                  421
000065415900566463OZ10000000000                  422
000065415900495502000000000000                   423  Last Card - Shadow Price
260002301826260344000079260218500064210218500069110218500OJ0Z101021140217400 0500  First Card - Dual
240002300000470222201100260222102185260002300000120344000OJ0Z101021740223400 0501
4702162012004419840000232600023022051603560000002603416000 74Z101022340229400 0502
260237202221260241300064210241300069210237200069260242502372Z101022940235400 0503
210241300069240000001826460248601300160046902425260119700000Z101023540241400 0504
490120000000240009900023470248601100260002300099260352402372Z101024140247400 0505
260356002413120341600OJ04702330012001403560000004702546012002101024740253400 0506
48N110000000260268403524260270803560260344000079260266503524Z101025340259400 0507
160269603770260278003524260270103560160046902665260119702545Z101025940265400 0508
490120000000260376000099260000001826260377000000260000001826Z101026540271400 0509
1102696000J01102701000J01102708000J01203440000J04602690013002101027140277400 0510
260000002545260285702221260286402221260341600074160046902857Z101027740283400 0511
260119703760490193800000260000000992102857000692102864000692101028340289400 0512
1203416000J04602822013001602984037702603025000642103025000692101028940295400 0513
160309703770260344000079220377001826460321801200260308502221Z101029540301400 0514
160312100000260341600074260314003121120314000008160046903097Z101030140307400 0515
```

The shadow price calculator converts all cost coefficients to floating point notation and evaluates $Z_j - c_j$ for each non-basis variable.

The dual algorithm computes a feasible solution for the problem. Its operation may be monitored allowing the typing of the value of the functional and the identifications of the variables being entered into and removed from the basis. It should be noted in this regard that the objectives of feasibility and optimality may not be compatible at this point, and therefore, the functional may actually decrease from one iteration to the next while the dual algorithm is in control

During the reduction of the matrix of coefficients a test is made to determine whether the resulting quantity has a magnitude less than an amout called 'essential zero.' Any element falling into this category is set to zero in order to avoid computation with numbers which are not significant. 'Essential zero' has been set to 10^{-8}. Its value (the corresponding floating point exponent) is located in memory addresses

```
2601197000004901938037701600469031331600445000004900402000992101030740313400051 6
1400000000M34703598013002103085000692103140000692103121000692101031340319400051 7
1203416000J04603062013001102984000J01103097000J01103025000J0Z101031940325400051 8
1203440000J04602978013002603373022212603361035601203361000J0Z101032540331400051 9
2603368033612203373000692603380033732600023000002600000000002101033140337400052 0
2600000000231103631000014702114001003400000001023803629001002101033740343400052 1
3400000001082603577033732603541033612603505000642103505000692101034340349400052 2
2603746000003803737001003400000001082603746000003803737001002101034940355400052 3
3400000001082603746000003803737001004902114026036160312126002101035540361400052 4
0000182649031580002Z00000000000                                    1010361403633000525
000000000000-                                                      -2114000526  Last Card - Dual
2503668037472600023018262603548000742602197000642102197000692101021140217400060 0  First Card - Simplex
2102197000692400023000004702234011002602233021972600023000002101021740223400060 1
→2102197000691203548000J04702186012004419840000232600023035222101022340229400060 2
1603481000002603632000792602380022332602433022332602421000642101022940235400060 3
2102421000694902542024000000182647025180110016004690243326012101023540241400060 4
1970000049012000000024000990002346025180110046025860120026022101024140247400060 5
4930238026037460000026000230009926034810242112036320002J04602Z101024740253400060 6
6300120011023800002J01102421000J01102433000J04902374026026090Z101025340259400060 7
2380240374600000470247001300490251801403481000004702666012002Z101025940265400060 8
48N1100000002602804024932602828022332603632000792602785024932Z101026540271400060 9
1602816037702602900024932602821022331600469027852601197026652101027140277400061 0
4901200000002603760000992600000018262603770000002600000001826Z101027740283400061 1
```

03144 and 03145 (Card No. 517) and may be changed if desired. If this change is made, it is important to note that the high order digit must carry a flag.

The simplex algorithm starts with any feasible solution and computes the optimal feasible solution. Its operation may be monitored by allowing iteration typeout as in the case of the dual algorithm.

The discussion of 'essential zero' as applied to the dual algorithm is equally valid for the simplex algorithm. In this case, the test value is located in memory addresses 03264 and 03265 (Card No. 619)


```
1102816000J01102821000J01102828000J01203632000J04602810013002101028340289400061 2
2600000026652602977034812602984034812603548000741600469029772101028940295400061 3
2601197037604901938000002600000009921029770006921029840006921Z101029540301400061 4
1203548000J04602942013001603104037702603145000642103145000692101030140307400061 5
1603217037702603632000792203770018264603338012002603205034812101030740313400061 6
1603241000002603548000742603260032411203260000081600469032172101031340319400061 7
2601197000004901938037701600469032531600445000004900402000992101031940325400061 8
1400000000M34703706013002103205000692103260000692103241000692101032540331400061 9
1203548000J04603182013001103104000J01103217000J01103145000J0Z101033140337400062 0
1203632000J04603098013002603469022331203469000J02603476034692101033740343400062 1
2203481000692603488034812600023000002600000000002600000000232101034340349400062 2
1103667000014702126001003400000001023803665001003400000001082101034940355400062 3
2603685034812603649034692603613000642103613000692603746000002101035540361400062 4
3803737001003400000001082603746000003803737001003400000001082101036140367400062 5
2603746000003803737001004902126026037240324126000000182649032101036740373400062 6
2780000000000ZZ00000000000                                         1010373403748000627
0000000000000                                                      -2114000628  Last Card  Simplex
→4919840003003400100001022602645000642102645000693619964005002101025620262200070 0  First Card  Final Basis
→3919965001002619984000003400010001081819975001003400000001022101026220268200070 1
→3400000001021603525000063619964005003919965001001203525000012101026820274200070 2
4702706012003400000001022603313000641103313000J02603361026452101027420280200070 3
1103361000J02603005033612103005000792602993030051102993000J0Z101028020286200070 4
2602957030051102957000K02603537000792600750026582600740025852101028620292200070 5
```

In the case of a tie in the quotient, normally used as the criterion for choosing the variable to leave the basis on any iteration, the choice is resolved by using the largest denominator as a secondary criterion.

```
1600725000001600715000001603036000002603089029572603101029572101029220298200070 6
1603077000001603149000002603205030052603525000742200000036982101029820304200070 7
4603206012001600469030892601197000004901200000004403158000002101030420310200070 8
2400099007504703206011002600750000992600730000004903206024002101031020316200070 9
0990074046032060130026007400009926007200000021030360006921032101031620322200071 0
0770006921030890006921031010006921031490006921032050006912032101032220328200071 1
→ 525000J047030300120026199820000034000000010238199730010032012101032820334200071 2
9550000027019480000026008000071434000000010838007970010026032101033420340200071 3
4210331326007600000032007550000044034780076033007600000032032101034020346200071 4
5130000049034900330351300000260076900760150077000000330195502101034620352200071 5
0000320074000000270194800740260080000724340000000108380079702101035220358200071 6
0100330075000000330195500000270194800750110295700OJ011033130Z101035820364200071 7
→ 00J011033610000J012035370000J047028980120049198400000000000707Z101036420370200071 8
0707003707070700002Z00000000000 1010370203721000719
00000000000 -2562000720 Last Card Final Basis
3201956000003100780036994402008019473301947000001600799000K0Z101019480200800075 0 First Card Fix + Print
2600760036981401939000N44702064011001600757R99994902148012012101020080206800075 1
9390000M747021360130016021350194021021350193915019390000021002101020680212800075 2
7600194744024420195516022190000416022870075316022820078116022101021280218800075 3
2430075316022500078132019570000044022760195743022640075316002101021880224800075 4
7810000049022880330195700000250078100753110228700001110224302101022480230800075 5
0001110228200002110225000002120221900001470222001200440241602101023080236800075 6
1956330195600000330195700000160221900005490231203400000001082101023680242800075 7
```

E-16

```
3900781001004214007990OOK047024780120032007600000021007600072101024280248800075 8
7047025220130016007990000049021480160079900OK049021487070707Z101024880254800075 9
0037070707002Z000000000000 1010254802561000760
0000000000000 -2574000761
4664554363495655415302000000000 762
0065415921435662630000020000000 763
0000414363496549636800020000000 764
0053495400654159000000020000000 765
0000005356664559005349540000020000000 766
5349540065415900000002000000000 767
6457574559005349540200000000000 768 Last Card Fix + Print
→ 3400010001021.4001000010216032530000636199640050039199650010021012562026220008 0 0 First Card Non Basis
1203253000014702598012003400000010226032170006421032170006921012626202682000801
2103217000692603265032171203217000J026028610321711028610000K0Z101026820274200080 2
2602897028612202897000692602909028972202909000692603224000 74Z101027420280200080 3
2600750025852600740025741600715000001600725000001602940000002101028020286200080 4
2602993028612603005028611602981000001603053000002603109029092101028620292200080 5
2603253000792200000034384603110012001600469029932601197000002101029220298200080 6
4901200000004403062000002400099007404703110011002600740000992101029820304200080 7
2600720000004903110024000990075046031100130026007500009926002101030420310200080 8
7300000011029400001011029810001011029930001011030050001011032101031020316200080 9
0530001011031090001012032530001047029340120026008000000034002101031620322200081 0
0000010238007910010032019550000027019480000026008000071434002101032220328200081 1
```

E-17

```
0000010838007970010027019480074026008000072434000000010838002101032820334200081 2
7970010027019480075021028610006921032170006921032650006912032101033420340200081 3
→224000J0470280201200491984000000000002Z000000000000000000 0  1010340203440000814
0000000000000                                                          -2562000815
0065415921435662630000020000000                                          816
0062484144005759494345020000000                                          817
0053495400654159000000020000000                                          818
0000005356664559005349540000020000000                                    819
534954006541590000000Z000000000                                          820
6457574559005349540Z000000000000                                         821   Last Card  Non Basis
4702834004003102860000481602874000001102144000051402144030240Z001021140217400090 0  First Card  Matrix Punch
4702138012003902861004002500079004003102860000762500074004002001021740223400090 1
3102864000711502868000002502863028582502867028582502869028582001022340229400090 2
1102288000011402288029364702282012001602939000003302936000002001022940235400090 3
3802860004003202936000002602629000641102629000012602593026292001023540241400090 4
2300069000733200095000002100099000692100098000782100099026292001024140247400090 5
2602492000992502486004002602569000991500079000001500074000002001024740253400090 6
1102939000011102593000701402593025584602714013002502859025822001025340259400090 7
2602612025932502606004003102860026183302936000003802860004002001025940265400090 8
3202936000002602684025932502678028592602629025934902534000002001026540271400090 9
4602774012002502860028581102732000011402732029304702726012002001027140277400091 0
2602797026293102860027862502931004003302936000003802860004002001027740283400091 1
→480000000000491984000000Z000000 0                                    -10283402858000912
```

E-18

```
Z0000000000000                                                  1010285802859000913
0Z0000000000000                                                 1010285902860000914
4100000000002000000000000                                       -10286002872000915
0000000000000                                                   -2114000916   Last Card  Matrix Punch
                                                                              Last Card  SOLUTION
```

E-19

THE FOLLOWING PAGES ARE SPS LISTINGS FOR

1. DATA LOADER
2. MATRIX PUNCH OUT
3. RIGHT HAND SIDE CHANGER

SPS listings for the remaining routines do not exist as the original program was read into memory from paper tape and then punched out on cards in the format for condensed SPS assembled cards. These cards were corrected, patched, etc., to make the present version for card I/O.

49

DATA LOADER - SPS LISTING

```
                                36000 720050 0360 020100500440001200275260005900274250001100000260009000269000
                                26000 950026 4310 000000200260011400274250000000011490001200000000000000000000 -
00402                                     DORG 402Z                                                              0
00402 48 00000 00000                      H     Z                                                                0
00414 16 02313 000-1                      TFM   CNTR+1+10Z                                                       0
00426 37 02315 00500              START   RACD  AREA+1Z                                                         0
00438 25 02325 00400                      TD    AREA+11+400Z                                                     0
00450 31 00048 02314                      TR    CASE-10+AREAZ                                                    0
00462 15 03742 0000-                      TDM   INPUT-4+0+11Z                                                    0
00474 36 02314 00500                      RNCD  AREAZ                                                            0
00486 25 02317 00400                      TD    AREA+3+400Z                                                      0
00498 31 03743 02314                      TR    INPUT-3+AREAZ                                                    0
00510 21 00532 03745                      A     SSTR+10+INPUT-1Z                                                 0
00522 16 00064 -3790              SSTR    TFM   SADDR+W+30+7Z                                                    0
00534 15 03746 00000                      TDM   INPUT+0Z                                                         0
00548 26 00079 03746              MSTR    TF    M+INPUTZ                                                         0
00558 11 03746 000K0                      AM    INPUT+20+10Z                                                     0
00570 26 00069 03746              M2STR   TF    MAND2+INPUTZ                                                     0
00582 25 02321 00400                      TD    AREA+7+400Z                                                      0
00594 31 03743 02318                      TR    INPUT-3+AREA+4Z                                                  0
00606 21 00628 03745                      A     NSTR+10+INPUT-1Z                                                 0
00618 16 00074 -0000              NSTR    TFM   NZ                                                               0
```

50

```
00630 25 03745 02322              TD   INPUT-1,AREA+8Z                    0
00642 43 00734 03745              BD   LOAD,INPUT-1Z                      0
00654 21 00684 00064              A    MTXTR+6,SADDRZ                     0
00666 36 02314 00500      MTXRD   RNCD AREAZ                              0
00678 31 -0001 02314      MTXTR   TR   1,AREA,2Z                          0
00690 11 00684 000P0              AM   MTXTR+6,70,10Z                     0
00702 45 00666 02385              BNR  MTXRD,AREA+71Z                     0
00714 48 00000 00000              H     Z                                 0
00726 49 19840 00000              B    LSTARTZ                            0
00734                             DORG *-3Z                               0
00734 37 02315 00500      LOAD    RACD AREA+1Z                            0
00746 25 02351 00400              TD   AREA+37,400Z                       0
00758 25 02391 00400              TD   AREA+77,400Z                       0
00770 25 02431 00400              TD   AREA+117,400Z                      0
00782 25 02471 00400              TD   AREA+157,400Z                      0
00794 43 00950 02316              BD   ELOAD,AREA+2Z                      0
00806 14 02313 000-4      ALOAD   CM   CNTR,4,10Z                         0
00818 47 00878 01200              BNE  BLOADZ                             0
00830 16 02313 000-1              TFM  CNTR,1,10Z                         0
00842 16 00925 -2316              TFM  CLOAD+11,AREA+2Z                   0
00854 16 00949 -2314              TFM  DLOAD+11,AREAZ                     0
00866 49 00734 00000              B    LOADZ                              0
00878 11 02313 000-1      BLOAD   AM   CNTR,1,10Z                         0
```

E-22

```
00890 11 00925 -0040              AM   CLOAD+11,40Z                       0
00902 11 00949 -0040              AM   DLOAD+11,40Z                       0
00914 43 00938 -2316      CLOAD   BD   DLOAD,AREA+2,7Z                   0
00926 49 00806 00000              B    ALOADZ                             0
00938 31 02314 -2314      DLOAD   TR   AREA,AREA,7Z                      0
00950 25 03740 02315      ELOAD   TD   INPUT-6,AREA+1Z                   0
00962 25 03741 02317              TD   INPUT-5,AREA+3Z                    0
00974 25 03742 02319              TD   INPUT-4,AREA+5Z                    0
00986 32 02318 00000              SF   AREA+4Z                            0
00998 14 02319 000P0              CM   AREA+5,70,10Z                      0
01010 46 01034 01300              BNL  TRCOLZ                             0
01022 32 03742 00000              SF   INPUT-4Z                           0
01034 25 03743 02323      TRCOL   TD   INPUT-3,AREA+9Z                    0
01046 25 03744 02325              TD   INPUT-2,AREA+11Z                   0
01058 25 03745 02327              TD   INPUT-1,AREA+13Z                   0
01070 32 02326 00000              SF   AREA+12Z                           0
01082 14 02327 000P0              CM   AREA+13,70,10Z                     0
01094 46 01118 01300              BNL  ACLFLZ                             0
01106 32 03745 00000              SF   INPUT-1Z                           0
01118 33 02318 00000      ACLFL   CF   AREA+4Z                            0
01130 33 02326 00000              CF   AREA+12Z                           0
01142 32 03740 00000              SF   INPUT-6Z                          0
01154 14 03742 00-00              CM   INPUT-4,0,9Z                       0
```

E-23

```
01166 46 02134 01200          BE    OUTZ                    0
01178 32 03743 00000          SF    INPUT-3Z                0
01190 23 03745 00069          M     INPUT-1,MAND2Z          0
01202 21 00099 00069          A     99,MAND2Z               0
01214 33 00099 00000          CF    99Z                     0
01226 21 00098 03742          A     98,INPUT-4Z             0
01238 21 00098 00063          A     98,SADDR-1Z             0
01250 49 02274 00000          B     XCLRZ                   0
01262 26 02120 00099          TF    STORE+6,99Z             0
01274 44 01474 03745          BNF   ROWMI1,INPUT-1Z         0
01286 16 02311 000-0    INITI TFM   I,0,10Z                 0
01298 25 00416 02331    TRID  TD    EXP-1,AREA+17Z          0
01310 11 01304 000-1          AM    TRID+6,1,10Z            0
01322 11 01309 000-2          AM    TRID+11,2,10Z           0
01334 11 02311 000-1          AM    I,1,10Z                 0
01346 14 02311 000J0          CM    I,10,10Z                0
01358 47 01298 01200          BNE   TRIDZ                   0
01370 32 02348 00000          SF    AREA+34Z                0
01382 14 02349 000P0          CM    AREA+35,70,10Z          0
01394 46 01418 01300          BNL   BCLFLZ                  0
01406 32 00425 00000          SF    EXP+8Z                  0
01418 33 02350 00000    BCLFL CF    AREA+36Z                0
01430 16 01304 -0416          TFM   TRID+6,EXP-1Z           0
```

E-24

```
01442 16 01309 -2331          TFM   TRID+11,AREA+17Z        0
01454 32 00416 00000    FLSET SF    EXP-1Z                  0
01466 49 02114 00000          B     STOREZ                  0
01474                         DORG  *-3Z                    0
01474 44 01494 03742    ROWMI1 BNF  TREL,INPUT-4Z           0
01486 49 01286 00000          B     INITIZ                  0
01494                         DORG  *-3Z                    0
01494 25 02351 00400    TREL  TD    AREA+37,00400Z          0
01506 31 03726 02330          TR    INPUT-20,AREA+16Z       0
01518 32 03726 00000          SF    INPUT-20Z               0
01530 14 03727 000P0          CM    INPUT-19,70,10Z         0
01542 46 01654 01300          BNL   CLRFLGZ                 0
01554 14 03727 000K0          CM    INPUT-19,20,10Z         0
01566 46 01634 01200          BE    NEGZ                    0
01578 43 01654 03727          BD    CLRFLG,INPUT-19Z        0
01590 33 01665 00000          CF    SIGNZ                   0
01602 32 03728 00000    SHFTSN SF   INPUT-18Z               0
01614 31 03726 03728          TR    INPUT-20,INPUT-18Z      0
01626 49 01666 00000          B     FLOATZ                  0
01634                         DORG  *-3Z                    0
01634 32 01665 00000    NEG   SF    SIGNZ                   0
01646 49 01602 00000          B     SHFTSNZ                 0
01654                         DORG  *-3Z                    0
```

E-25

```
01654 33 01665 00000      CLRFLG CF   SIGNZ                          0
01666 16 00417 000N0      FLOAT  TFM  EXP,50,10Z                     0
01678 16 01720 -3727           TFM  DEC+6,INPUT-19Z                  0
01690 16 01708 -3728           TFM  FLDDEF+6,INPUT-18Z               0
01702 32 01702 00000      FLDDEF SF   *Z                             0
01714 14 01714 000-3      DEC    CM   *,3,10Z                        0
01726 46 01782 01200           BE   SHFTDCZ                          0
01738 11 01720 000-2           AM   DEC+6,2,10Z                      0
01750 11 01708 000-2           AM   FLDDEF+6,2,10Z                   0
01762 11 00417 000-1           AM   EXP,1,10Z                        0
01774 49 01702 00000           B    FLDDEFZ                          0
01782                          DORG *-3Z                             0
01782 26 01829 01708      SHFTDC TF   *+47,FLDDEF+6Z                 0
01794 26 01824 01720           TF   *+30,DEC+6Z                      0
01806 12 01824 000-1           SM   *+18,1,10Z                       0
01818 31 01818 01818           TR   *,*Z                             0
01830 43 01886 03727      ZRTST  BD   TRSMT,INPUT-19Z                0
01842 31 03726 03728           TR   INPUT-20,INPUT-18Z               0
01854 12 00417 000-1           SM   EXP,1,10Z                        0
01866 45 01830 03727           BNR  ZRTST,INPUT-19Z                  0
01878 49 00806 00000           B    ALOADZ                           0
01886                          DORG *-3Z                             0
01886 16 01465 000-9      TRSMT  TFM  CTR,9,10Z                      0
```



```
01898 16 01945 -3727           TFM  TSTRC+11,INPUT-19Z               0
01910 16 01960 -0418           TFM  STRDIG+6,EXP+1Z                  0
01922 16 01965 -3727           TFM  STRDIG+11,INPUT-19Z              0
01934 45 01954 00000      TSTRC  BNR  STRDIGZ                        0
01946 49 02022 00000           B    FILLZ                            0
01954                          DORG *-3Z                             0
01954 25 00000 00000      STRDIG TD   Z                              0
01966 11 01945 000-2           AM   TSTRC+11,2,10Z                   0
01978 11 01960 000-1           AM   STRDIG+6,1,10Z                   0
01990 11 01965 000-2           AM   STRDIG+11,2,10Z                  0
02002 12 01465 000-1           SM   CTR,1,10Z                        0
02014 49 01934 00000           B    TSTRCZ                           0
02022                          DORG *-3Z                             0
02022 26 02064 01960      FILL   TF   STRZR+6,STRDIG+6Z              0
02034 12 01465 000-1           SM   CTR,1,10Z                        0
02046 46 02090 01200           BZ   SIGNSTZ                          0
02058 15 02058 00000      STRZR  TDM  *,0Z                           0
02070 11 02064 000-1           AM   STRZR+6,1,10Z                    0
02082 49 02034 00000           B    FILL+12Z                         0
02090                          DORG *-3Z                             0
02090 44 02114 01665      SIGNST BNF  STORE,SIGNZ                    0
02102 32 00425 00000           SF   EXP+8Z                           0
02114 26 02114 00425      STORE  TF   *,EXP+8Z                       0
```

```
02126 49 00806 00000          B     ALOADZ                                  0
02134                         DORG  *-3Z                                    0
02134 26 02212 00064    OUT   TF    FLG+6,SADDRZ                            0
02146 12 02212 000J9          SM    FLG+6,19,10Z                            0
02158 23 00069 00073          M     MAND2,N-1Z                              0
02170 32 00095 00000          SF    95Z                                     0
02182 21 00099 00069          A     99,MAND2Z                               0
02194 21 00099 00069          A     99,MAND2Z                               0
02206 32 02206 00000    FLG   SF    *Z                                      0
02218 11 02212 000J0          AM    FLG+6,10,10Z                            0
02230 12 00099 000J0          SM    99,10,10Z                               0
02242 47 02206 01200          BNZ   FLGZ                                    0
02254 48 00000 00000          H     Z                                       0
02266 49 19840 00000          B     LSTARTZ                                 0
02274                         DORG  *-3Z                                    0
02274 33 03743 00000    XCLR  CF    INPUT-3Z                                0
02286 32 00095 00000          SF    95Z                                     0
02298 49 01262 00000          B     INITI-24Z                               0
03760                   W     DS    ,3760Z                                  0
00079                   M     DS    ,79Z                                    0
00074                   N     DS    ,74Z                                    0
00069                   MAND2 DS    ,69Z                                    0
00064                   SADDR DS    ,64Z                                    0
```

E-28

```
03746                   INPUT DS    ,3746Z                                  0
00058                   CASE  DS    ,58Z                                    0
01665                   SIGN  DS    ,CLRFLG+11Z                             0
01465                   CTR   DS    ,FLSET+11Z                              0
00417                   EXP   DS    ,START-9Z                               0
02311         2 00000   I     DC    2,0Z                                    0
                Z000 0000 000                                               -602310023120
02313         2 00000   CNTR  DC    2,0Z                                    0
                Z000 0000 0000                                              -602312023140
02314 48 00000 00000    AREA  H     Z                                       0
19840                         DORG  19840Z                                  0
19840 36 19965 00500    LSTART RNCD LAREAZ                                  0
19852 44 19944 00039          BNF   LOUT,LAREA+74Z                          0
19864 26 19887 00048          TF    LCHA+11,LAREA+73Z                       0
19876 25 19851 19876    LCHA  TD    LSTART+11,*Z                            0
19888 26 19906 00033          TF    LTR+6,LAREA+68Z                         0
19900 31 19900 19965    LTR   TR    *,LAREAZ                                0
19912 26 19930 00038          TF    LBACK+6,LAREA+73Z                       0
19924 25 19924 19851    LBACK TD    *,LSTART+11Z                            0
19936 49 19840 00000          B     LSTARTZ                                 0
19944                         DORG  *-3Z                                    0
19944 26 19962 00038    LOUT  TF    LGO+6,LAREA+73Z                         0
19956 49 19956 00000    LGO   B     *Z                                      0
```

E-29

```
19965                              DORG *-2Z                                          0
19965        1 00000         LAREA DS   1Z                                          0
00402                              DEND 00402Z                                        0
                                 L600 0000 0500490000020000000000                    -800096001150
                          36001 000050 0360 0172005003600244005003690316005003600000005000000000000000000
                                       102 0304000204060800030609021004080216100500151020060218142002000
                          70411 282008 0614 2230090817263000000000050607080900121416181518112427202422000
                          82236 352035 3045 4036324844553249465360484654627544536271801234567891234562000
                          78902 345678 90J3 4567890JK4567890JKL567890JKLM67890JKLMN7890JKLMNO890JKLMNZ000
                          M8000 000000 0490 04020P90JKLMNOPQZ0000L10038800019M90000000000M90003600000000
```

E-30

RHS CHANGER SPS Listing

```
                          ~~36000 720050 0560 0201005004400012002752600059002742500011000002600090002000000~~
                          ~~28000 950026 4910 0000002002600114002742500000000114900012000000000000000000000~~
02114                              DORG 2114Z                                         0
02114 48 00000 00000               H    Z                                             0
02126 46 02218 00300          GO   BC3  READTYZ                                       0
02138 37 03485 00500          READCD RACD RAREA+1Z                                   0
02150 25 02292 03485               TD   IDIN-3,RAREA+1Z                             0
02162 25 02293 03487               TD   IDIN-2,RAREA+3Z                             0
02174 25 02294 03489               TD   IDIN-1,RAREA+5Z                             0
02186 25 02295 03491               TD   IDIN,RAREA+7Z                               0
02198 25 02296 00400               TD   IDIN+1,400Z                                 0
02210 49 02242 00000               B    GONEZ                                       0
02218                              DORG *-3Z                                          0
02218 34 00000 00102          READTY RCTY Z                                         0
02230 36 02292 00100               RNTY IDIN-3Z                                     0
02242 32 02292 00000          GONE SF   IDIN-3Z                                     0
02254 14 02295 0-000               CM   IDIN,0,8Z                                   0
02266 46 03464 01200               BE   RHSOUTZ                                     0
02278 47 02322 00300               BNC3 TRCHNGZ                                     0
02290 34 00000 00108          TABTY TBTY Z                                          0
02302 37 03727 00100               RATY INP-19Z                                     0
02314 49 02414 00000               B    FISHINZ                                     0
02322                              DORG *-3Z                                          0
```

E-31

```
02322 32 03500 00000      TRCHNG SF   RAREA+16Z                    0
02334 14 03501 000P0      SINTST CM   RAREA+17,70,10Z               0
02346 46 02390 01300      LEADIG BNL  SET35Z                       0
02358 43 02390 03501      LEADEC BD   SET35,RAREA+17Z              0
02370 25 03521 00400             TD   RAREA+37,400Z                0
02382 49 02402 00000             B    MOVERCZ                      0
02390                            DORG *-3Z                         0
02390 25 03519 00400      SET35  TD   RAREA+35,400Z                0
02402 31 03726 03500      MOVERC TR   INP-20,RAREA+16Z             0
02414 32 03726 00000      FISHIN SF   INP-20Z                      0
02426 14 03727 000P0             CM   INP-19,70,10Z                0
02438 46 02550 01300             BNL  CLRFLGZ                      0
02450 14 03727 000K0             CM   INP-19,20,10Z                0
02462 46 02530 01200             BE   NEGZ                         0
02474 43 02550 03727             BD   CLRFLG,INP-19Z               0
02486 33 02137 00000             CF   SIGNZ                        0
02498 32 03728 00000      SHFTSN SF   INP-18Z                      0
02510 31 03726 03728             TR   INP-20,INP-18Z               0
02522 49 02562 00000             B    FLOATZ                       0
02530                            DORG *-3Z                         0
02530 32 02137 00000      NEG    SF   SIGNZ                        0
02542 49 02498 00000             B    SHFTSNZ                      0
02550                            DORG *-3Z                         0
```

E-32

```
02550 33 02137 00000      CLRFLG CF   SIGNZ                        0
02562 16 02117 000N0      FLOAT  TFM  EXP,50,10Z                   0
02574 16 02616 -3727             TFM  DEC+6,INP-19Z                0
02586 16 02604 -3728             TFM  FLDDEF+6,INP-18Z             0
02598 32 02598 00000      FLDDEF SF   *Z                           0
02610 14 02610 000-3      DEC    CM   *,3,10Z                      0
02622 46 02678 01200             BE   SHFTDCZ                      0
02634 11 02616 -0002             AM   DEC+6,2,7Z                   0
02646 11 02604 -0002             AM   FLDDEF+6,2,7Z                0
02658 11 02117 000-1             AM   EXP,1,10Z                    0
02670 49 02598 00000             B    FLDDEFZ                      0
02678                            DORG *-3Z                         0
02678 26 02725 02604      SHFTDC TF   *+47,FLDDEF+6Z               0
02690 26 02720 02616             TF   *+30,DEC+6Z                  0
02702 12 02720 -0001             SM   *+18,1,7Z                    0
02714 31 02714 02714             TR   *,*Z                         0
02726 43 02782 03727      ZRTST  BD   TRSMT,INP-19Z                0
02738 31 03726 03728             TR   INP-20,INP-18Z               0
02750 12 02117 000-1             SM   EXP,1,10Z                    0
02762 45 02726 03727             BNR  ZRTST,INP-19Z                0
02774 49 02126 00000             B    GOZ                          0
02782                            DORG *-3Z                         0
02782 16 02253 000-9      TRSMT  TFM  CTR,9,10Z                    0
```

E-33

```
02794 16 02841 -3727              TFM  TSTRC+11,INP-19Z          0
02806 16 02856 -2118              TFM  STRDIG+6,EXP+1Z           0
02818 16 02861 -3727              TFM  STRDIG+11,INP-19Z         0
02830 45 02850 00000       TSTRC  BNR  STRDIGZ                   0
02842 49 02918 00000              B    FILLZ                    0
02850                             DORG *-3Z                     0
02850 25 02850 02850       STRDIG TD   *,*Z                     0
02862 11 02841 -0002              AM   TSTRC+11,2,7Z            0
02874 11 02856 -0001              AM   STRDIG+6,1,7Z            0
02886 11 02861 -0002              AM   STRDIG+11,2,7Z           0
02898 12 02253 000-1              SM   CTR,1,10Z                0
02910 49 02830 00000              B    TSTRCZ                   0
02918                             DORG *-3Z                     0
02918 26 02960 02856       FILL   TF   STRZR+6,STRDIG+6Z        0
02930 12 02253 000-1              SM   CTR,1,10Z                0
02942 46 02986 01200              BZ   SIGNSTZ                  0
02954 15 02954 00000       STRZR  TDM  *,0Z                     0
02966 11 02960 -0001              AM   STRZR+6,1,7Z             0
02978 49 02930 00000              B    FILL+12Z                 0
02986                             DORG *-3Z                     0
02986 44 03010 02137       SIGNST BNF  *+24,SIGNZ               0
02998 32 02125 00000              SF   EXP+8Z                   0
03010 26 02253 00079              TF   CTR,MZ                   0
```

E-34

```
03022 26 03052 00064              TF   ILP+6,SADDRZ             0
03034 11 03052 -0004              AM   ILP+6,4,7Z               0
03046 24 03046 02295       ILP    C    *,IDINZ                  0
03058 46 03192 01200              BE   FOUNDIZ                  0
03070 11 03052 000J0              AM   ILP+6,10,10Z             0
03082 12 02253 000J0              SM   CTR,10,10Z               0
03094 47 03046 01200              BNZ  ILPZ                     0
03106 26 02253 00074              TF   CTR,NZ                   0
03118 26 03148 03052              TF   JLP+6,ILP+6Z             0
03130 21 03148 00069              A    JLP+6,MAND2Z             0
03142 24 03142 02295       JLP    C    *,IDINZ                  0
03154 46 03272 01200              BE   FOUNDJZ                  0
03166 12 02253 000J0              SM   CTR,10,10Z               0
03178 47 03130 01200              BNZ  JLP-12Z                  0
03190 48 00000 00000              H    Z                        0
03192                             DORG *-9Z                     0
03192 26 03251 03052       FOUNDI TF   ADD+FP,ILP+6Z            0
03204 21 03251 00069              A    ADD+FP,MAND2Z            0
03216 11 03251 -0006              AM   ADD+FP,6,7Z              0
03228 16 00469 -3263       ADD    TFM  00469,*+35Z              0
03240 16 00445 -3240              TFM  00445,*Z                 0
03252 49 00422 -2125              B    00422,EXP+8,7Z           0
03264 49 02126 00000              B    GOZ                      0
```

E-35

```
03272                                  DORG  *-3Z                                  0
03272 26 03355 03148        FOUNDJ     TF    ELLP1+FP,JLP+6Z                      0
03284 11 03355 000J6                   AM    ELLP1+FP,16,10Z                      0
03296 26 03391 00064                   TF    ELLP2+FP,SADDRZ                      0
03308 21 03391 00069                   A     ELLP2+FP,MAND2Z                      0
03320 26 02253 00079                   TF    CTR,MZ                               0
03332 16 00469 -3367        ELLP1      TFM   00469,*+35Z                          0
03344 16 01197 -3344                   TFM   01197,*Z                             0
03356 49 01938 -2125                   B     01938,EXP+8,7Z                       0
03368 16 00469 -3403        ELLP2      TFM   00469,*+35Z                          0
03380 16 00445 -3380                   TFM   00445,*Z                             0
03392 49 00422 -0099                   B     00422,00099,7Z                       0
03404 11 03355 000J0                   AM    ELLP1+FP,10,10Z                      0
03416 11 03391 000J0                   AM    ELLP2+FP,10,10Z                      0
03428 12 02253 000J0                   SM    CTR,10,10Z                           0
03440 46 03332 01300                   BNM   ELLP1Z                               0
03452 49 02126 00000                   B     GOZ                                  0
03464 48 00000 00000        RHSOUT     H      Z                                   0
03476 49 19840 00000                   B     LSTARTZ                              0
03484                                  DORG  *-3Z                                 0
00079                       M          DS    ,79Z                                 0
00074                       N          DS    ,74Z                                 0
00069                       MAND2      DS    ,69Z                                 0
```

E-36

```
00064                       SADDR      DS    ,64Z                                 0
00023                       FP         DS    ,23Z                                 0
02295                       IDIN       DS    ,TABTY+5Z                            0
03746                       IMP        DS    ,3746Z                               0
02117                       EXP        DS    ,GO-9Z                               0
02253                       CTR        DS    ,GONE+11Z                            0
02137                       SIGN       DS    ,GO+11Z                              0
19840                       LSTART     DS    ,19840Z                              0
03484 41 00000 00000        RAREA      NOP    Z                                   0
02114                                  DEND  02114Z                               0
```

E-37

MATRIX PUNCH OUT - SPS LISTING

~~36000 720050 0360 020100500440001200275260005900274250001100000260009000269~~ 0

~~26000 950026 4310 000000200260011400274250000000011490001200000~~ 0

```
02114                          DORG 02114Z                          0
02114 47 02834 00400     START BNC4 OUTZ                           0
02126 31 02860 00048           TR   SLUSH,48Z                      0
02138 16 -2874 -0000     DID   TFM  SLUSH+14,0,27Z                  0
02150 11 02144 -0005           AM   DID+6,5Z                       0
02162 14 02144 -3024           CM   DID+6,SLUSH+164Z               0
02174 47 02138 01200           BNE  DIDZ                           0
02186 39 02861 00400           WACD SLUSH+1Z                       0
02198 25 00079 00400           TD   M,400Z                         0
02210 31 02860 00076           TR   SLUSH,M-3Z                     0
02222 25 00074 00400           TD   N,400Z                         0
02234 31 02864 00071           TR   SLUSH+4,N-3Z                   0
02246 15 02868 00000           TDM  SLUSH+8,0Z                     0
02258 25 02863 02858           TD   SLUSH+3,GLUGZ                  0
02270 25 02867 02858           TD   SLUSH+7,GLUGZ                  0
02282 25 -2869 02858     YOU   TD   SLUSH+9,GLUG,2Z                0
02294 11 02288 -0001           AM   YOU+6,1Z                       0
02306 14 02288 -2936           CM   YOU+6,SLUSH+76Z                0
02318 47 02282 01200           BNE  YOUZ                           0
02330 16 02939 0-000           TFM  SLUSH+79,0,8Z                  0
02342 33 02936 00000           CF   SLUSH+76Z                      0
```

E-38

67

```
02354 38 02860 00400           WNCD SLUSHZ                         0
02366 32 02936 00000           SF   SLUSH+76Z                      0
02378 26 02629 00064           TF   GOOD+11,SADDRZ                 0
02390 11 02629 -0001           AM   GOOD+11,1Z                     0
02402 26 02593 02629           TF   NICKEL+11,GOOD+11Z             0
02414 23 00069 00073           M    MAND2,N-1Z                     0
02426 32 00095 00000           SF   95Z                            0
02438 21 00099 00069           A    99,MAND2Z                      0
02450 21 00098 00078           A    98,M-1Z                        0
02462 21 00099 02629           A    99,GOOD+11Z                    0
02474 26 02492 00099           TF   KNOW+6,99Z                     0
02486 25 02486 00400     KNOW  TD   *,400Z                         0
02498 26 02569 00099           TF   ART+11,99Z                     0
02510 15 00079 00000           TDM  M,0Z                           0
02522 15 00074 00000           TDM  N,0Z                           0
02534 11 02939 0-001     THAT  AM   SLUSH+79,1,8Z                  0
02546 11 02593 -0070           AM   NICKEL+11,70Z                  0
02558 14 02593 -2558     ART   CM   NICKEL+11,*Z                  0
02570 46 02714 01300           BNL  YESZ                           0
02582 25 -2859 02582     NICKEL TD  HOLD,*,2Z                      0
02594 26 02612 02593           TF   DOES+6,NICKEL+11Z              0
02606 25 02606 00400     DOES  TD   *,400Z                         0
02618 31 02860 02618     GOOD  TR   SLUSH,*Z                       0
```

E-39

68

```
02630 33 02936 00000          CF    SLUSH+76Z                    0
02642 38 02860 00400          WNCD  SLUSHZ                       0
02654 32 02936 00000          SF    SLUSH+76Z                    0
02666 26 02684 02593          TF    WORK+6,NICKEL+11Z            0
02678 25 02678 02859    WORK  TD    *,HOLDZ                      0
02690 26 02629 02593          TF    GOOD+11,NICKEL+11Z           0
02702 49 02534 00000          B     THATZ                        0
02714 46 02774 01200    YES   BE    RIGHTZ                       0
02726 25 -2860 02858    THATS TD    SLUSH,GLUG,2Z                0
02738 11 02732 -0001          AM    THATS+6,1Z                   0
02750 14 02732 -2930          CM    THATS+6,SLUSH+70Z            0
02762 47 02726 01200          BNE   THATSZ                       0
02774 26 02797 02629    RIGHT TF    KIDDO+11,GOOD+11Z            0
02786 31 02860 02786    KIDDO TR    SLUSH,*Z                     0
02798 25 02931 00400          TD    SLUSH+71,400Z                0
02810 33 02936 00000          CF    SLUSH+76Z                    0
02822 38 02860 00400          WNCD  SLUSHZ                       0
02834 48 00000 00000    OUT   H     Z                            0
02846 49 19840 00000          B     19840Z                       0
00079                   M     DS    ,79Z                         0
00074                   N     DS    ,74Z                         0
00064                   SADDR DS    ,64Z                         0
02858         1         GLUG  DNR   1Z                           0
```

E-X

69

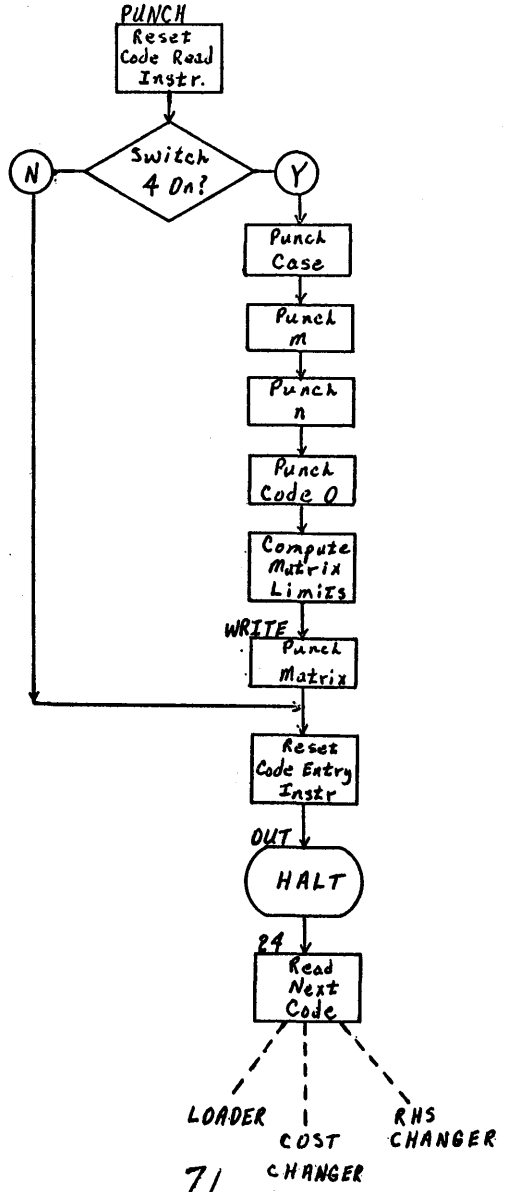
```
00069                   MAND2 DS    ,69Z                   -702858028590
02859         1         HOLD  DC    1,0Z                         0
                        Z                                        0
02860 41 00000 00000    SLUSH NOP   Z                  0602859028600
                              DEND  Z                            0
                                                                 0
```

[illegible]

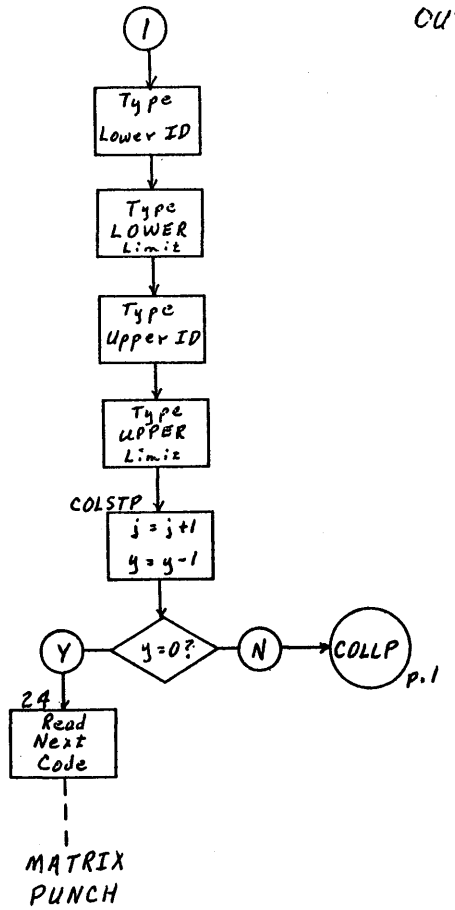
70

E-Y

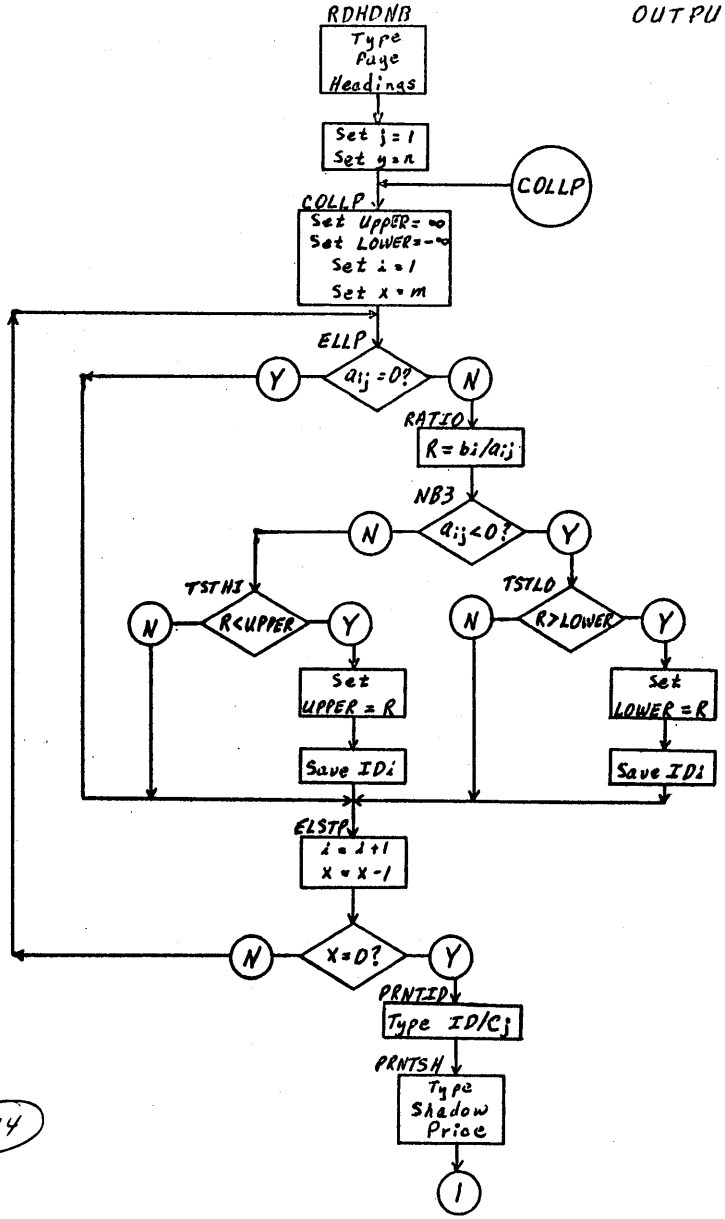
MATRIX
PUNCH



NON
BASIS
OUTPUT - 2

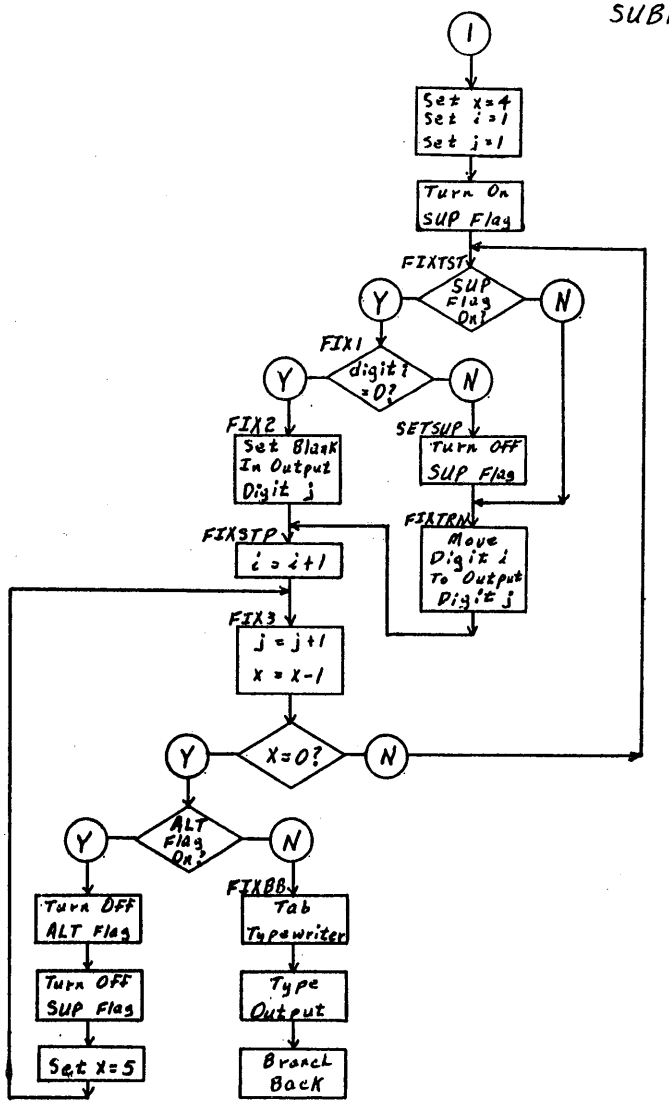


NON
BASIS
OUTPUT - 1



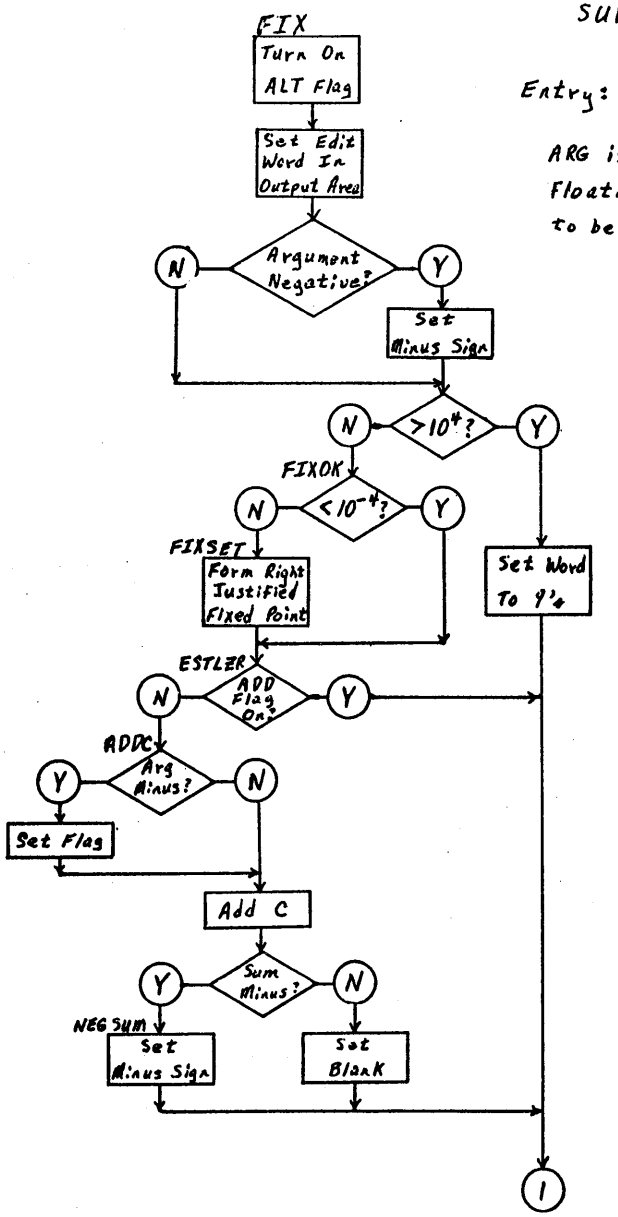
14

FIX
SUBROUTINE - 2



15

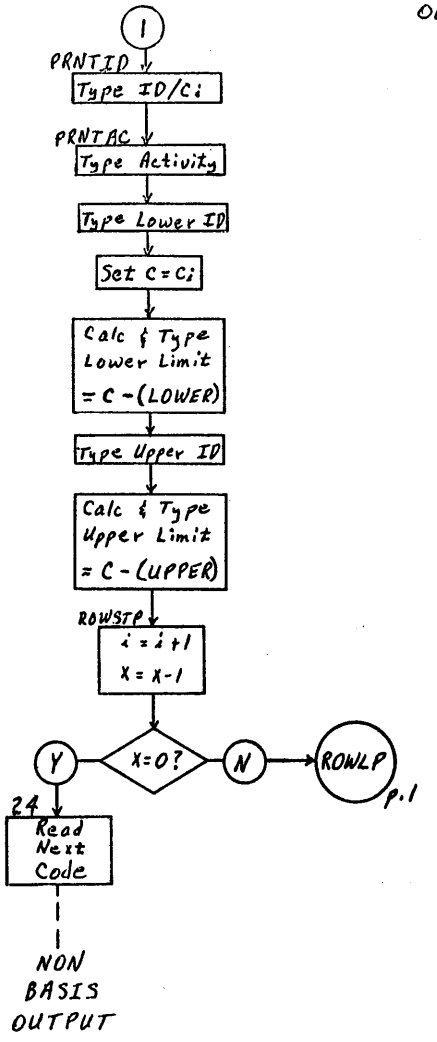
FIX SUBROUTINE -1



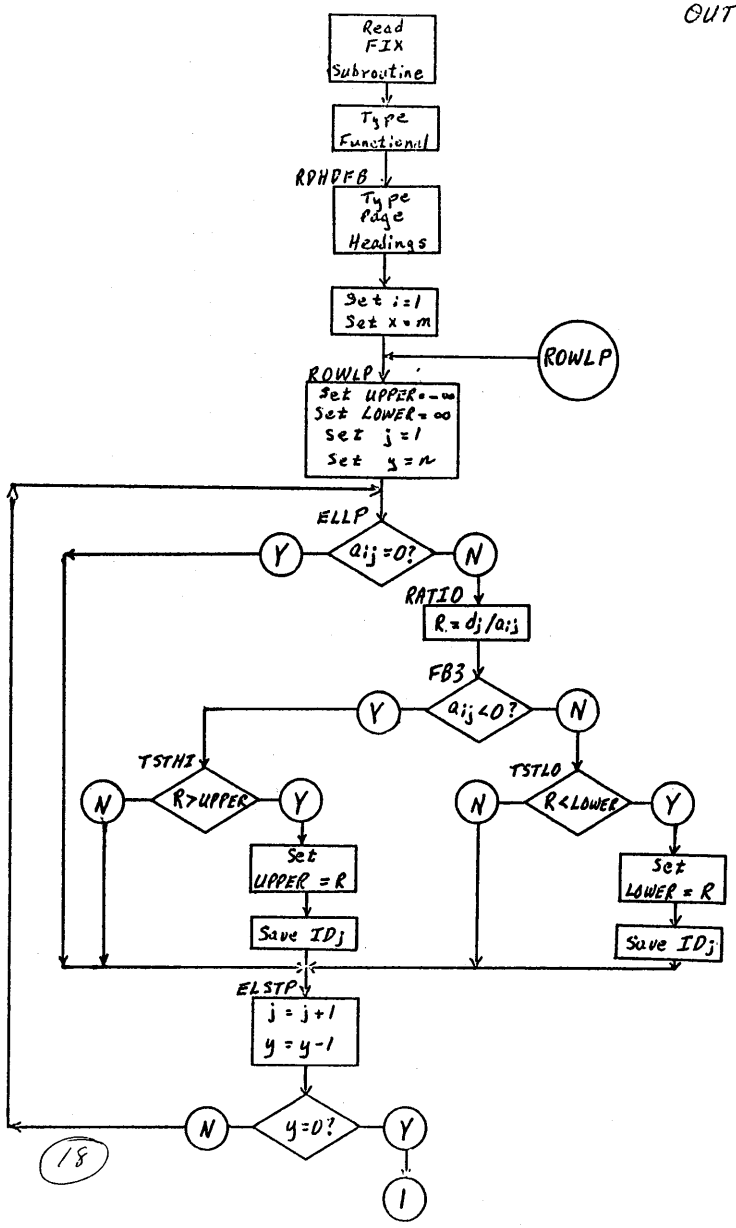
Entry: BT FIX, ARG

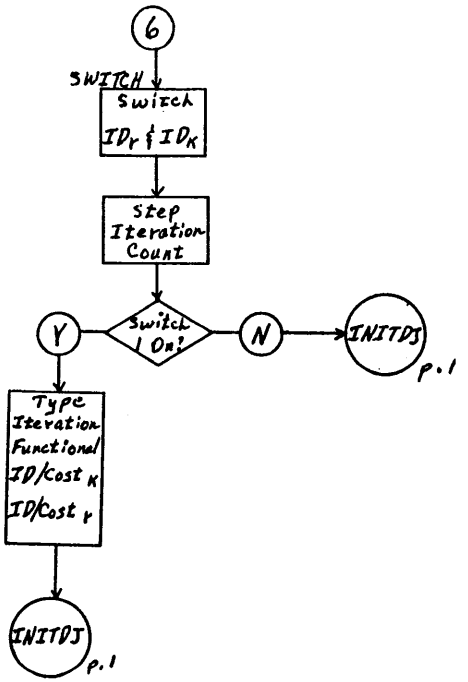
ARG is location of Floating point field to be converted and typed.

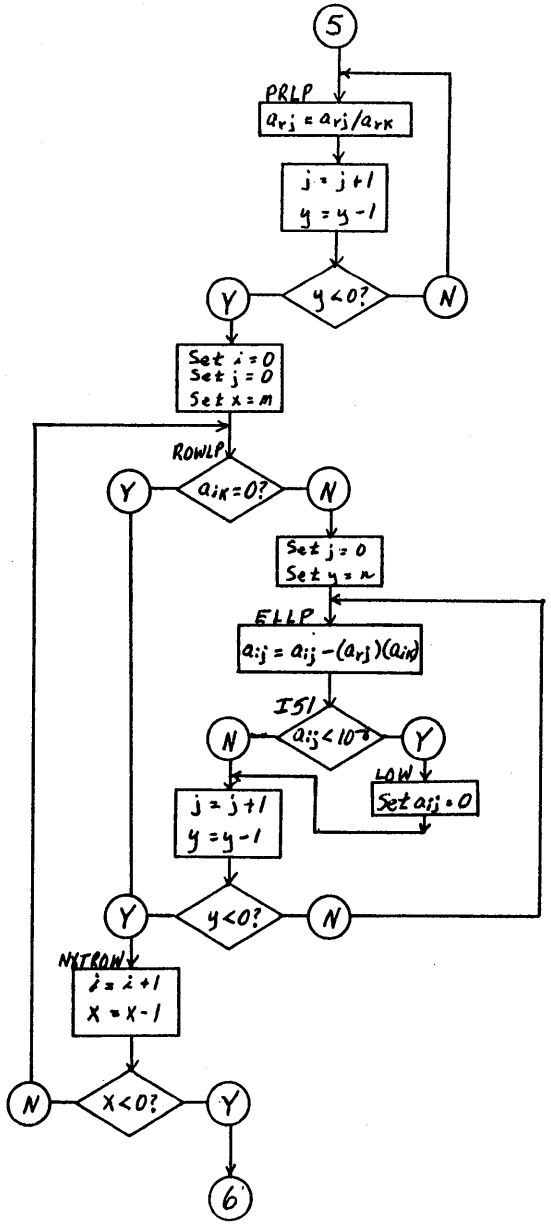
FINAL BASIS OUTPUT -2



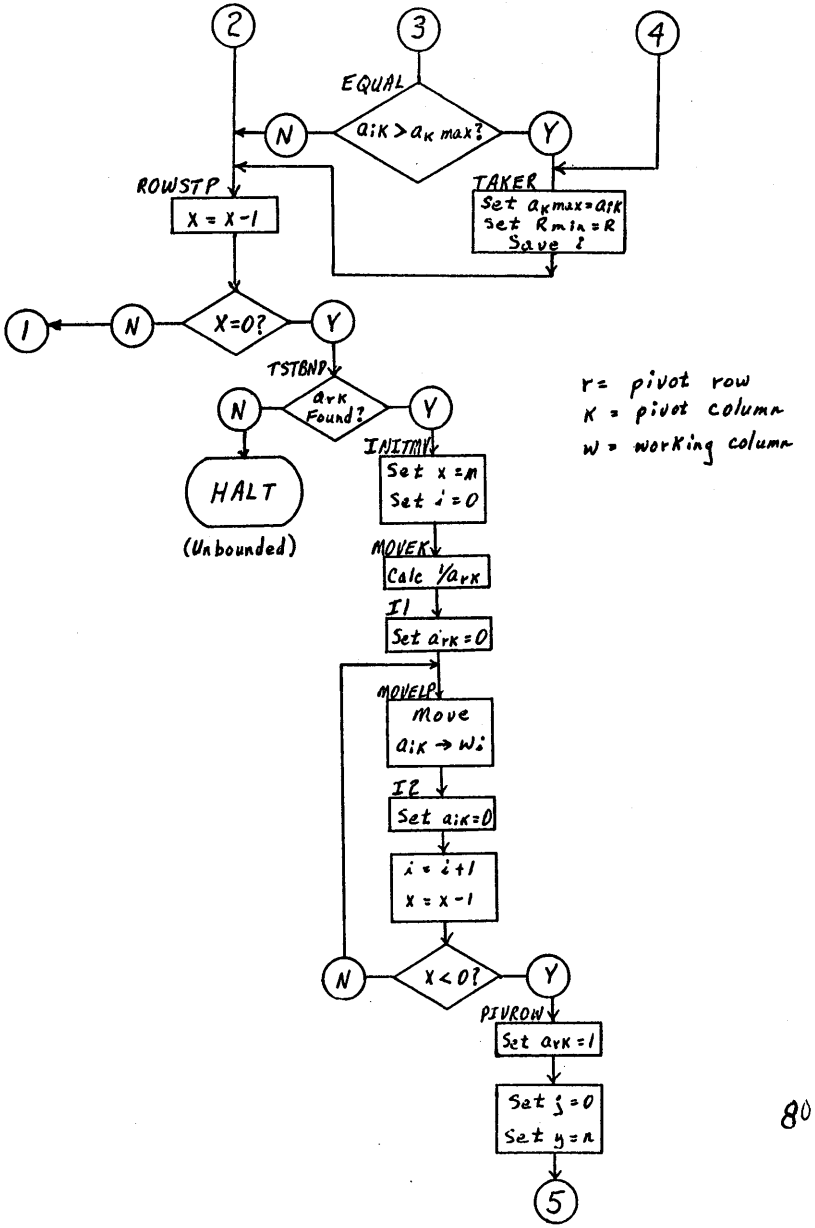
FINAL
BASIS
OUTPUT - 1



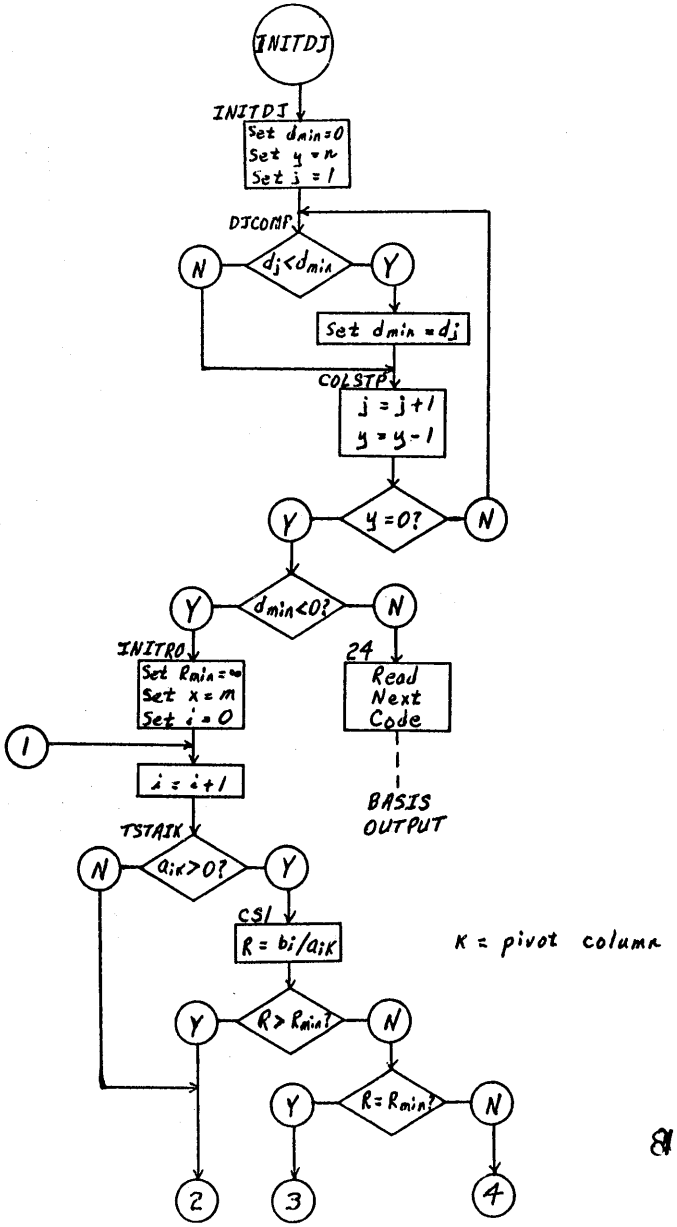


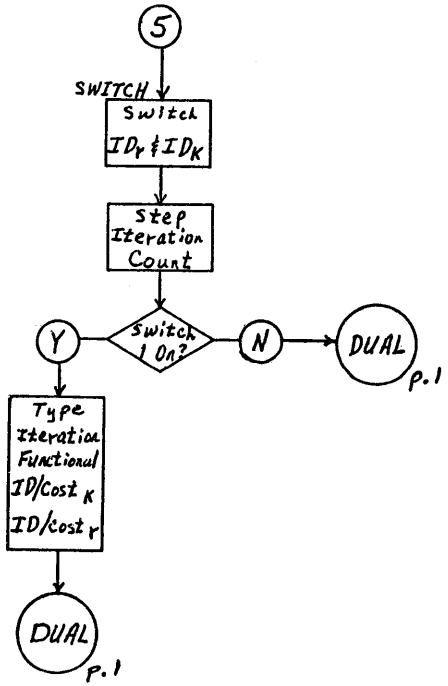


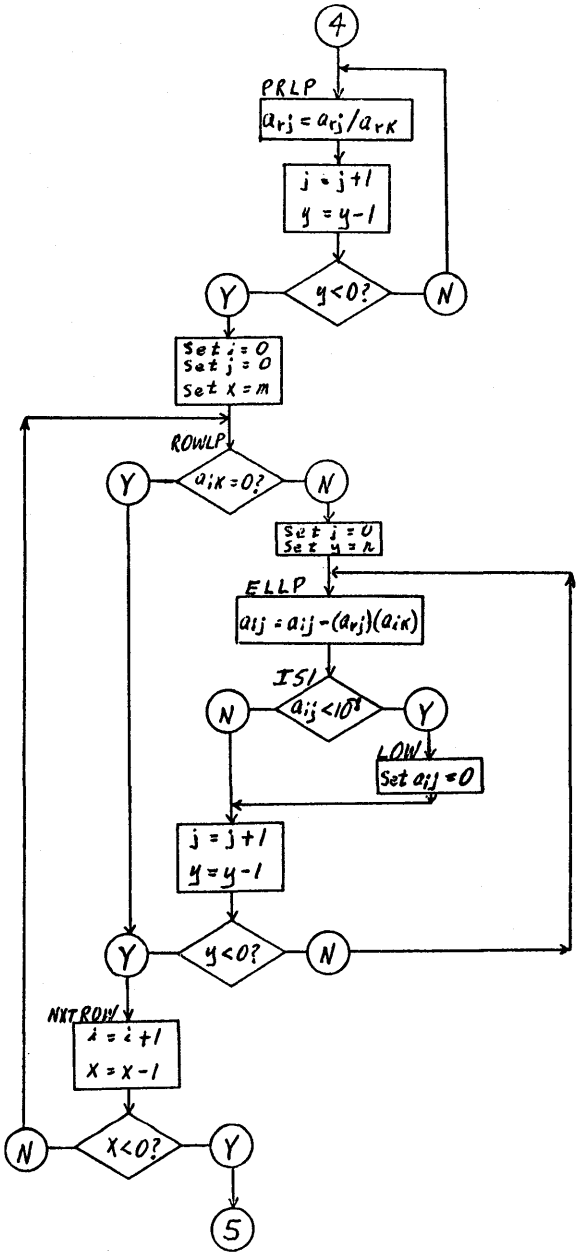
(20)

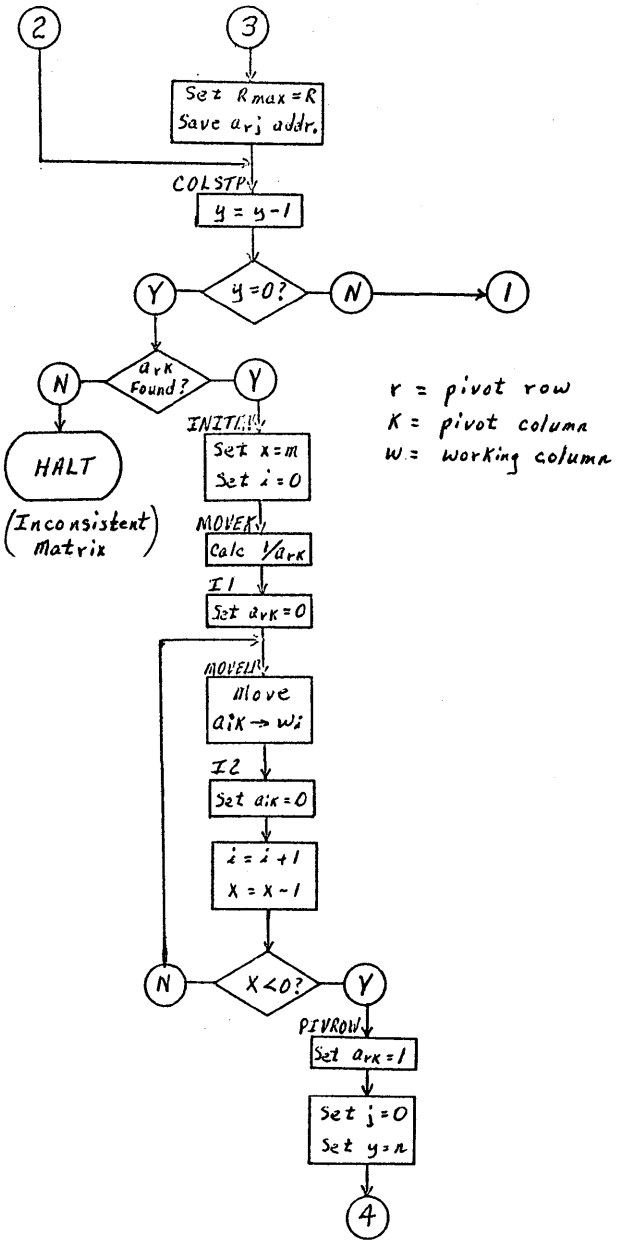


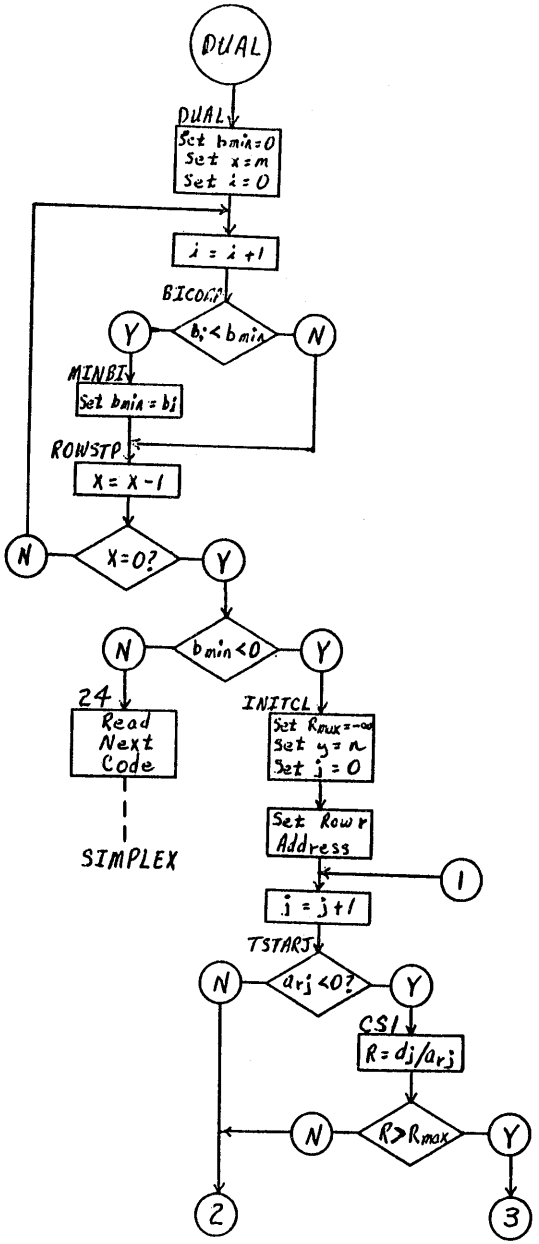
r = pivot row
K = pivot column
w = working column

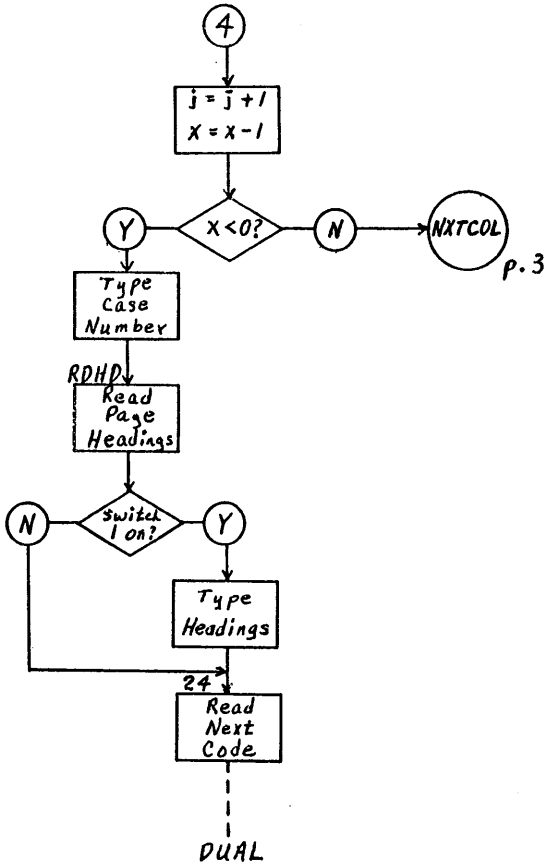

INITDJ
INITDJ
Set dmin=0
Set y = n
Set j = 1
DJCOMP
dj<dmin
N
Y
Set dmin = dj
COLSTP
j = j+1
y = y-1
y=0?
Y
N
dmin<0?
Y
N
INITRO
Set Rmin = ∞
Set x = m
Set i = 0
24
Read Next Code
BASIS OUTPUT
1
i = i+1
TSTAIK
aiK>0?
N
Y
CSI
R = bi/aiK
K = pivot column
R>Rmin?
Y
N
R=Rmin?
Y
N
2
3
4

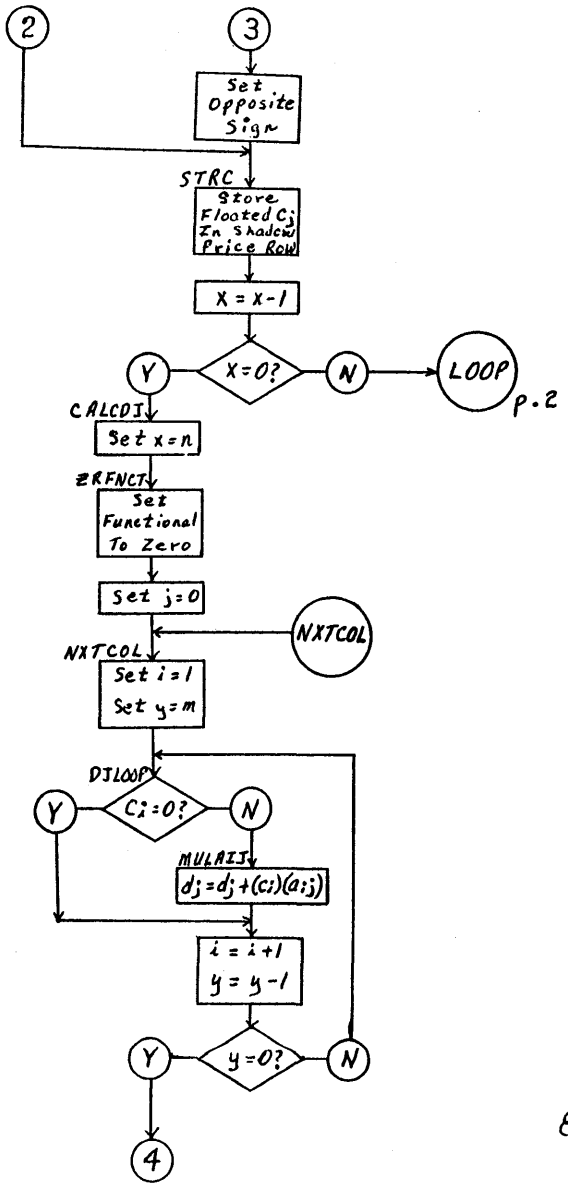

5
SWITCH
Switch IDr & IDK
Step Iteration Count
Y
switch 1 On?
N
DUAL
p.1
Type Iteration Functional
ID/cost K
ID/cost r
DUAL
p.1

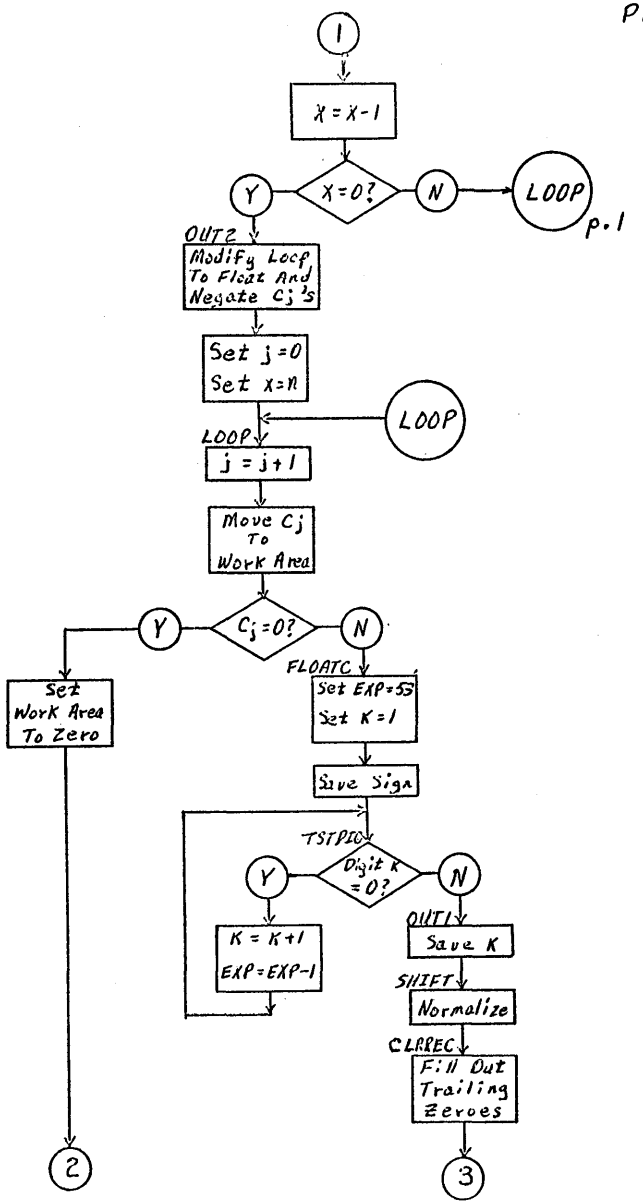

4
PRLP
$a_{rj} = a_{rj}/a_{rK}$
j = j+1
y = y-1
y<0?
Y
N
Set i = 0
Set j = 0
Set x = m
ROWLP
a_{iK}=0?
Set j = 0
Set y = n
ELLP
$a_{ij} = a_{ij} - (a_{rj})(a_{iK})$
ISI
a_{ij}<10^{s}
LOW
Set a_{ij} = 0
j = j+1
y = y-1
y<0?
NXTROW
i = i+1
X = X-1
X<0?
5


2
3
Set Rmax = R
Save a_{rj} addr.
COLSTP
y = y-1
y=0?
1
a_{rK} Found?
HALT
(Inconsistent Matrix)
INITIN
Set x = m
Set i = 0
MOVER
Calc 1/a_{rK}
I1
Set a_{rK} = 0
MOVELP
Move $a_{iK} \rightarrow w_i$
I2
Set a_{iK} = 0
i = i+1
X = X-1
X<0?
PIVROW
Set a_{rK} = 1
Set j = 0
Set y = n
4
r = pivot row
K = pivot column
w = working column

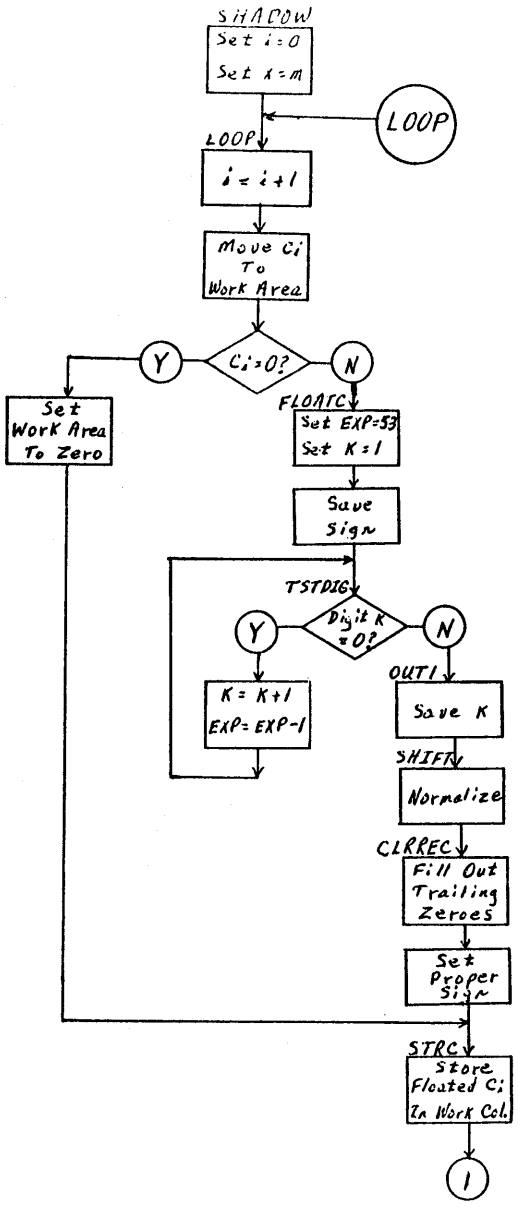

DUAL
DUAL
Set bmin=0
Set x=m
Set i=0
i = i+1
BICOMP
$b_i < b_{min}$
Y
N
MINBI
Set bmin = bi
ROWSTP
X = X-1
X=0?
N
Y
$b_{min} < 0$
N
Y
24
Read Next Code
SIMPLEX
INITCL
Set $R_{max} = -\infty$
Set y = n
Set j = 0
Set Row r Address
1
j = j+1
TSTARJ
$a_{rj} < 0$?
N
Y
CSI
$R = d_j / a_{rj}$
$R > R_{max}$
N
Y
2
3


4
j = j+1
X = X-1
Y
X<0?
N
NXTCOL
p. 3
Type Case Number
RDHD
Read Page Headings
N
Switch 1 on?
Y
Type Headings
24
Read Next Code
DUAL

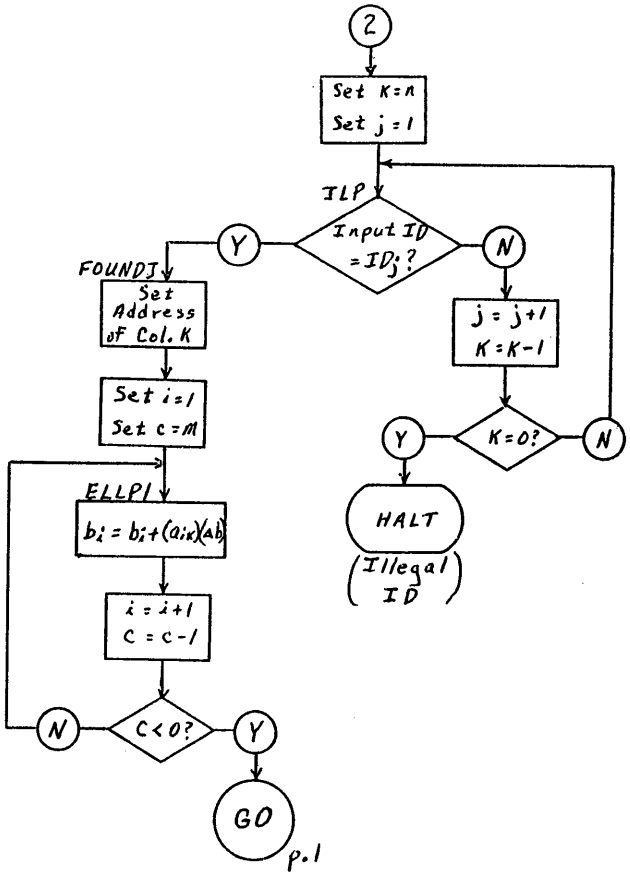

2
3
Set Opposite Sign
STRC
Store Floated Cj In Shadow Price Row
X = X-1
Y
X=0?
N
LOOP
p.2
CALCDJ
Set x=n
ZRFNCT
Set Functional To Zero
Set j=0
NXTCOL
NXTCOL
Set i=1
Set y=m
DJLOOP
Y
Ci=0?
N
MULAIJ
dj=dj+(ci)(aij)
i = i+1
y = y-1
Y
y=0?
N
4


1
X = X-1
Y
X=0?
N
LOOP
p.1
OUT2
Modify Loop To Float And Negate Cj's
Set j=0
Set x=n
LOOP
LOOP
j = j+1
Move Cj To Work Area
Y
Cj=0?
N
Set Work Area To Zero
FLOATC
Set EXP=53
Set K=1
Save Sign
TSTPIC
Y
Digit K =0?
N
K = K+1
EXP=EXP-1
QUIT1
Save K
SHIFT
Normalize
CLRREC
Fill Out Trailing Zeroes
2
3

SHADOW
PRICE -1

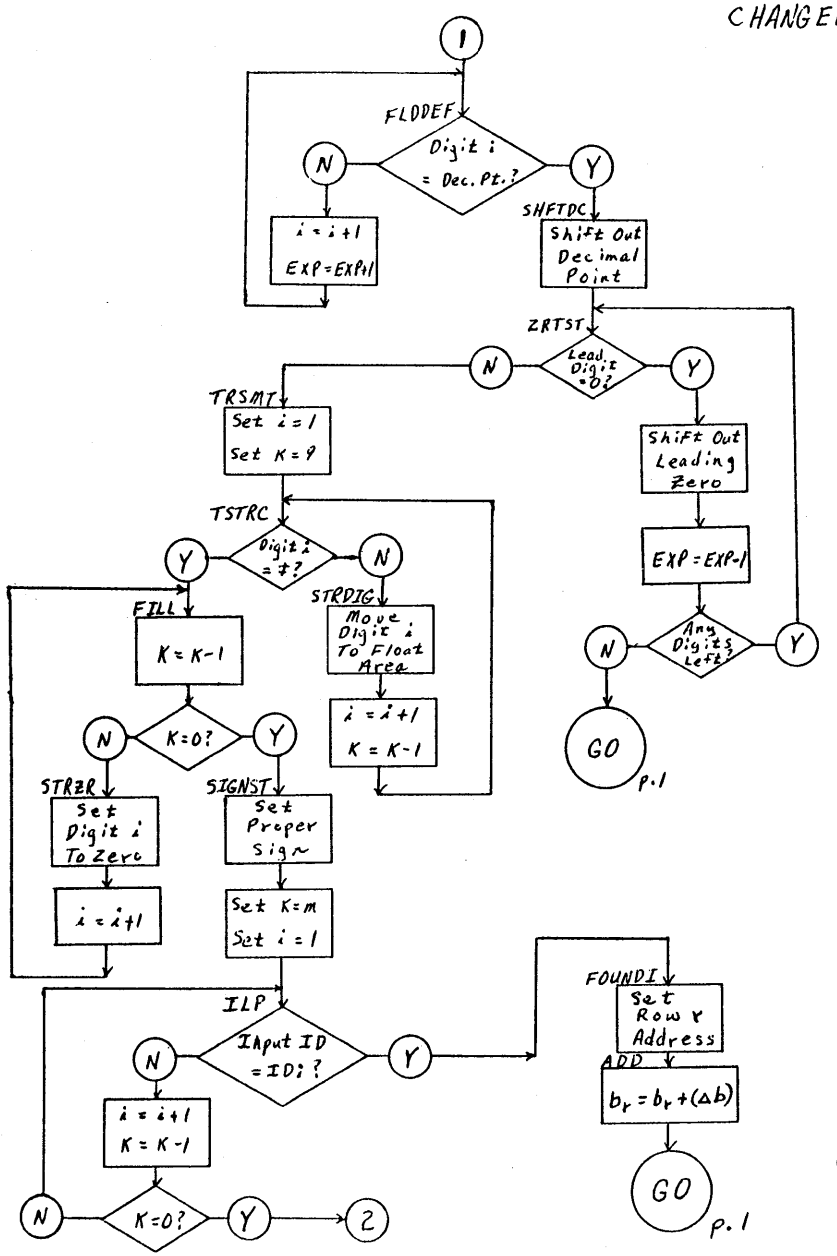


RHS
CHANGER - 3



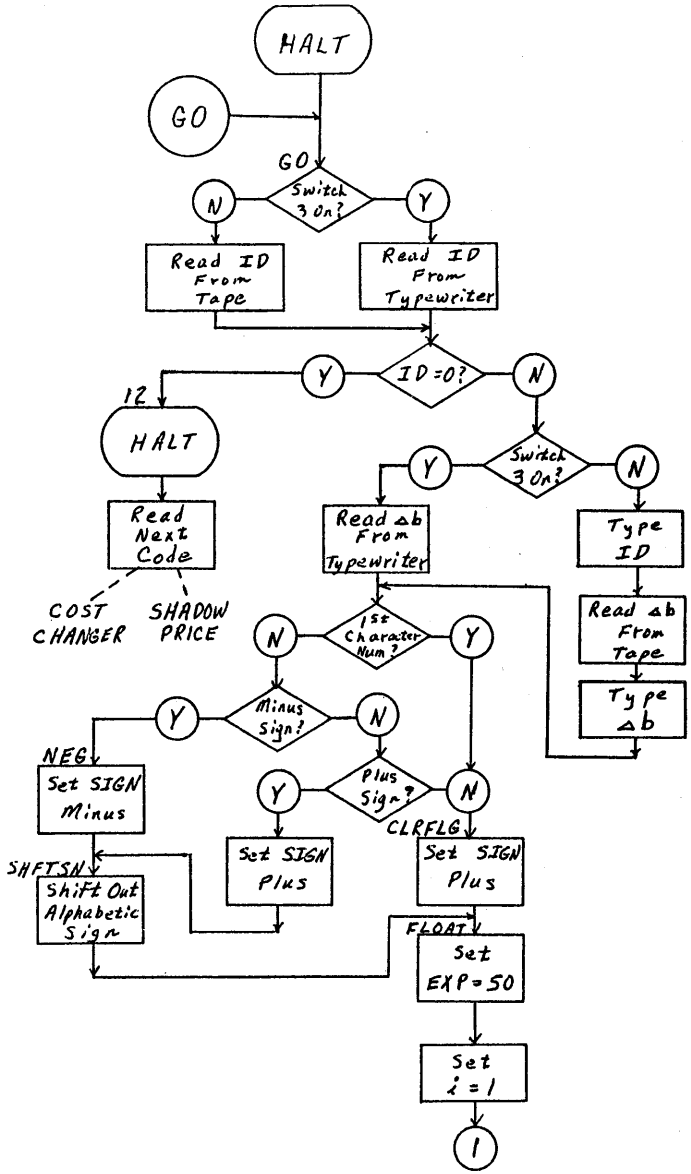
p.1

RHS
CHANGER - 2



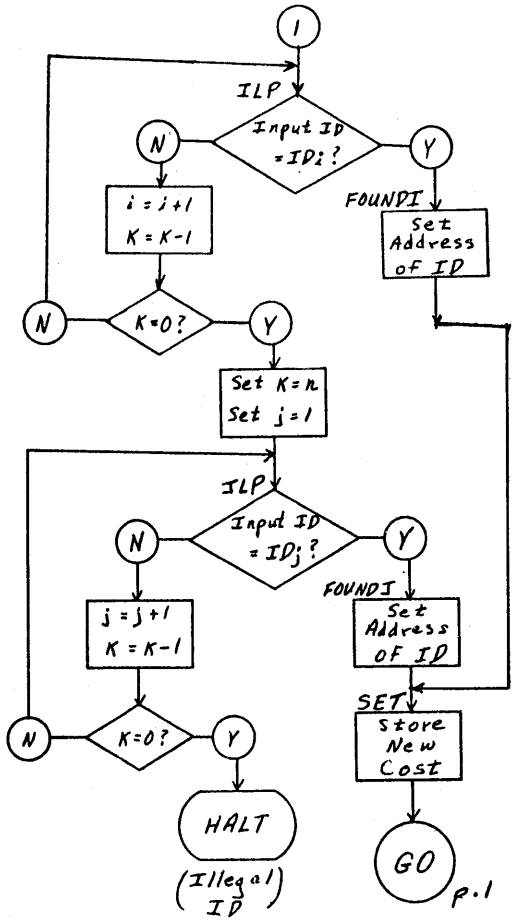
9l

RHS
CHANGER - 1

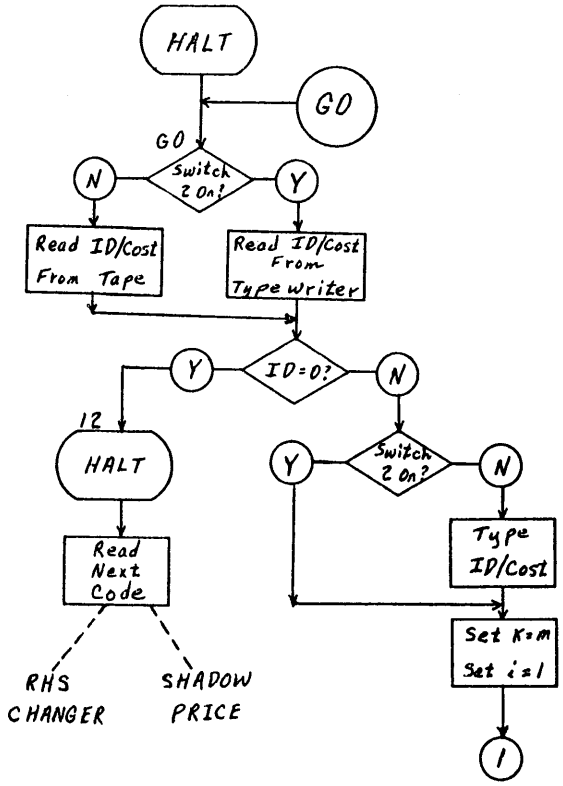


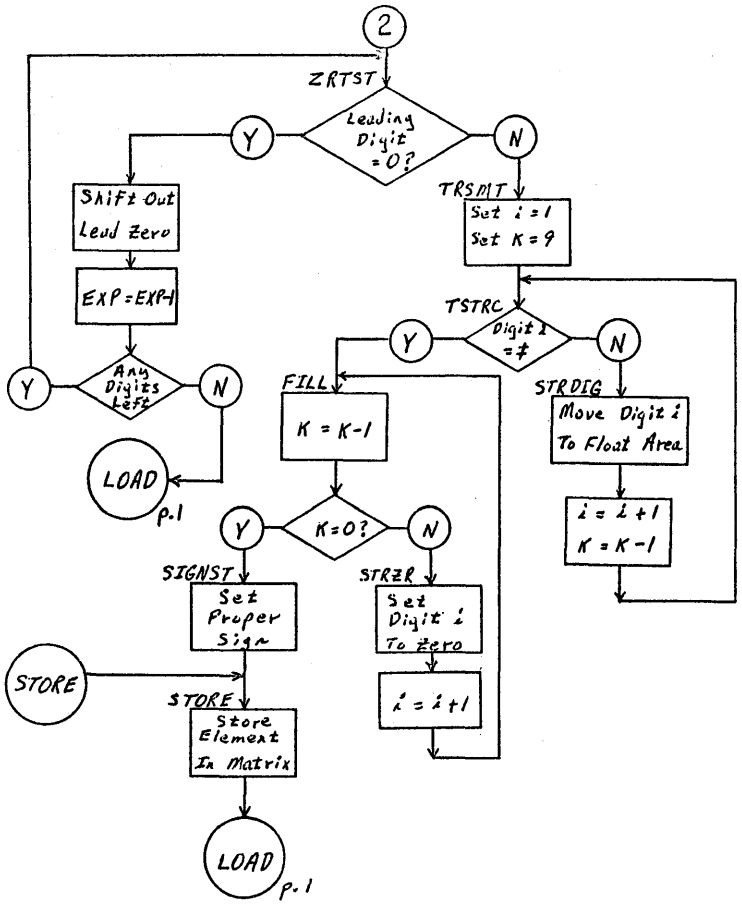
9r

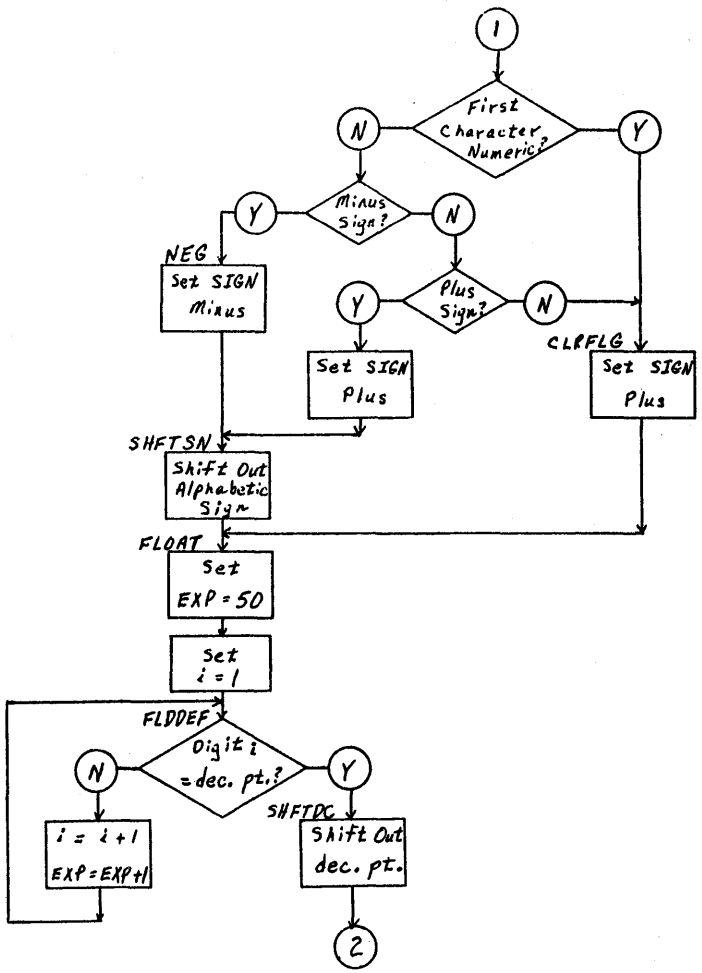
COST
CHANGER - 2

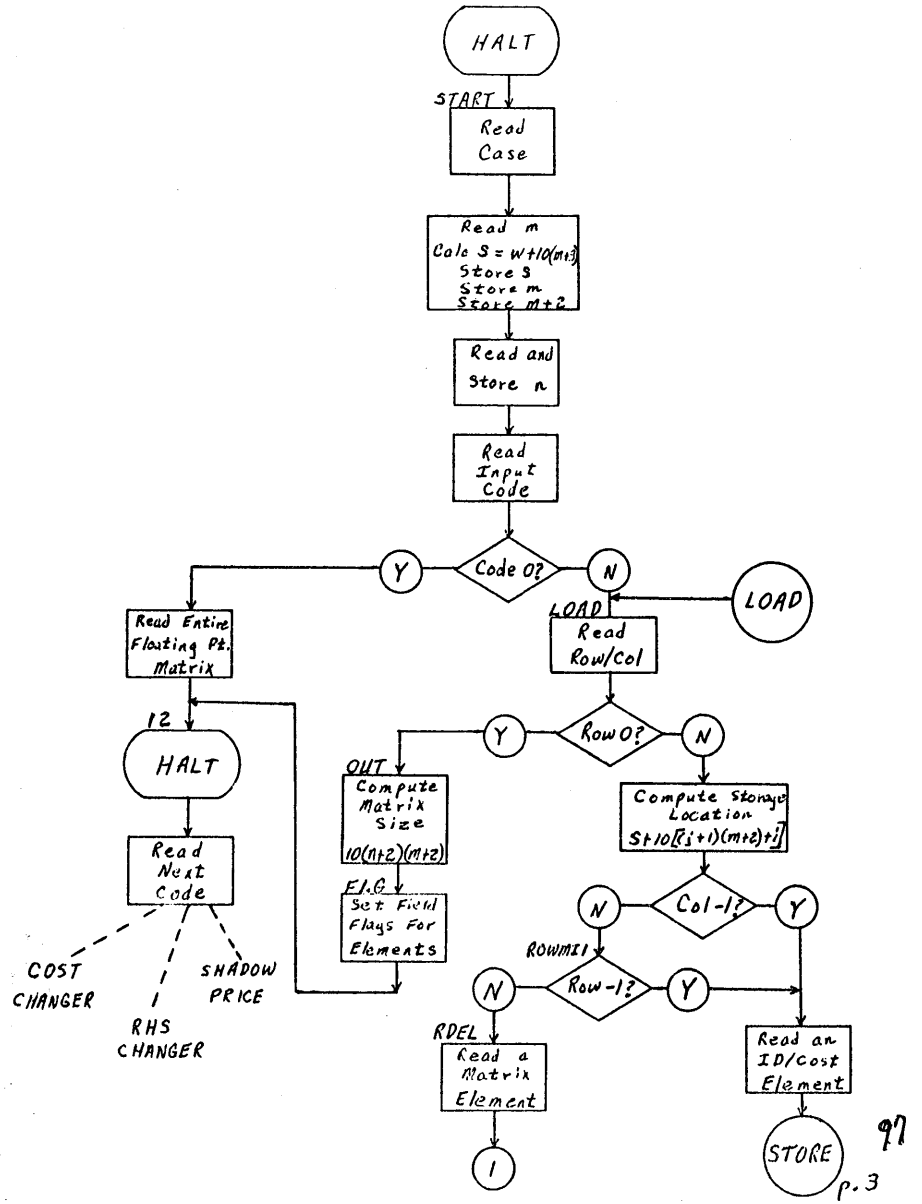


COST
CHANGER - 1




2
ZRTST
Leading Digit = 0?
Y
N
Shift Out Lead Zero
EXP = EXP-1
Any Digits Left?
Y
N
LOAD p.1
TRSMT
Set i = 1
Set K = 9
TSTRC
Digit i = ≠
Y
N
FILL
K = K-1
K=0?
Y
N
SIGNST
Set Proper Sign
STORE
STORE
Store Element In Matrix
LOAD p.1
STRZR
Set Digit i To Zero
i = i + 1
STRDIG
Move Digit i To Float Area
i = i + 1
K = K-1


1
First Character Numeric?
N
Y
Minus Sign?
Y
N
NEG
Set SIGN Minus
Plus Sign?
Y
N
Set SIGN Plus
CLRFLG
Set SIGN Plus
SHFTSN
Shift Out Alphabetic Sign
FLOAT
Set EXP = 50
Set i = 1
FLDDEF
Digit i = dec. pt.?
N
Y
i = i + 1
EXP = EXP+1
SHFTDC
Shift Out dec. pt.
2


HALT
START
Read Case
Read m
Calc S = W+10(m+3)
Store S
Store m
Store m+2
Read and Store n
Read Input Code
Code 0?
Y
N
LOAD
LOAD
Read Row/Col
Row 0?
Y
N
OUT
Compute Matrix Size
10(n+2)(m+2)
E.O.C.
Set Field Flags For Elements
Compute Storage Location
S+10[(i+1)(m+2)+j]
Col -1?
N
Y
ROWM1
Row -1?
N
Y
RDEL
Read a Matrix Element
1
Read an ID/Cost Element
STORE
Read Entire Floating Pt. Matrix
12
HALT
Read Next Code
COST CHANGER
RHS CHANGER
SHADOW PRICE

97

COMPUTER
TECHNOLOGY