Staff Papers Series

STAFF PAPER P82-15 SEPTEMBER 1982

Matrix Generator and Optionals (MGAO): Users Guide

Howard McDowell

Department of Agricultural and Applied Economics

University of Minnesota
Institute of Agriculture, Forestry and Home Economics
St. Paul, Minnesota 55108

Matrix Generator and Optionals (MGAO):

Users Guide

Howard McDowell

September 1982

MGAO - Operation Outline

H. McDowell

Table of Contents

Preface

Matrix Generator and Optionals (MGAO) is a computer software package developed by Paul Chang and Terry L. Roe. The program is designed to generate input data for a linear programming problem approximating a non-linear programming problem, submit the generated problem to an optimization package, from which the user receives standard computer output.

This paper results directly from efforts by the author to utilize the program and is the first comprehensive documentation written on the program. It is hoped that this paper will make available a useful computer program to those interested. Criticism and suggestions are welcome.

Terry L. Roe provided a significant contribution in the theoretical section and in the general organization of the paper. Reviews by Jeff Apland, Vernon Eidman, and Boyd Buxton are appreciated.

I. INTRODUCTION

Matrix Generator and Optionals (MGAO) is a fortran computer program developed to generate input matrices for mathematical programming algorithm.[1] Of primary importance is its capacity to generate a linear programming problem approximating a nonlinear programming problem.

Specifically, the program is capable of generating matrices for solving linear approximations of nonlinear programming problems incorporating linear or nonlinear supply and demand functions, linear and nonlinear production functions having multiple inputs, and substitutability in demand.

The program operates in conjunction with Multi Purpose Optimization System, MPOS, a system of mathematical programming algorithms developed for solving optimization problems on CDC 6000/CYBER computers. The system includes various linear programming (LP), integer programming (IP), and quadratic programming (QP) algorithms, and an interface with CDC's APEX, a system designed for solving large scale linear programming problems.[2]

For purposes of exposition each mathematical program may be viewed as being composed of two parts, a nonaugmented and an augmented section. The nonaugmented portion is perhaps best illustrated or characterized by most traditional linear programming problems. Following Intrilligator, this portion of the problem may be stated as "choosing nonnegative values of certain variables so as to maximize or minimize a given linear function subject to a given set of linear inequality constraints....

... $\max_{\underline{X}} F(\underline{X}) = \underline{CX}$

Subject to

$\underline{AX} \leq \underline{b}$, $X \geq 0$

(where $\underline{A}$ is mxn, $\underline{X}$, nx1, $\underline{C}$, 1xn, $\underline{b}$, nx1) "or, written out in full:

$$\max_{x_1, x_2 \ldots x_n} F(x_1, x_2, \ldots x_n) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

subject to:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2$$
$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m$$
$$x_1 \geq 0,\ x_2 \geq 0, \ldots,\ x_n \geq 0.$$"[3]

Clearly, a problem of this nature requires nothing more than defining the activities (x's) the coefficients (c's and a's) and the right-hand-side (RHS) parameters (b's). Therefore in this respect, MGAO is simply a means of entering the data for a linear programming problem, or the linear portion of a nonlinear programming problem. This specification is referred to as the nonaugmented problem, i.e. it has not been augmented to include a nonlinear function.

The augmented portion of the matrix is that portion generated by the program from input data in linear functional form. The principle involved is that a nonlinear function may be approximated by a number of linear steps each of which is a separate linear programming activity. Hence, this technique is also known as separable programming. As the number of steps increases the loss in accuracy decreases. The nonlinear programming problem is stated by Intrilligator below.

"The nonlinear programming problem is that of choosing nonnegative values of certain variables so as to maximize (minimize) a given quasi-concave (convex) function subject to a set of inequality constraints....

... $\max_{\underline{X}} F(\underline{X})$ subject to $\underline{g}(\underline{X}) \leq \underline{b} \quad \underline{X} \geq \underline{0}$

or written out in full:

$$\max_{x_1 \dots x_n} F(x_1 \dots x_n) \text{ subject to}$$

$$g_1(x_1 \dots x_n) \leq b_1$$

$$g_m(x_1 \dots x_n) \leq b_m$$

$$x_1 \geq 0, \dots, x_n \geq 0."[4]$$

This portion of the problem requires entering the objective function $F(\underline{X})$, and the constraints $\underline{g}(\underline{X})$, in nonlinear form. MGAO then defines discrete linear programming activities with the appropriate objective and constraint activities according to the instructions provided by the user.

The augmented portion of the matrix is also referred to as the extended portion of the matrix.

In specification of problems with both nonaugmented and augmented matrices, the user is advised to design the matrices such that the nonaugmented portion of the matrix, i.e. that part not containing linear approximations of nonlinear equations, is in the upper left hand portion of the matrix and that all transfer or summary columns from the generated rows of the matrix containing the linear approximations of nonlinear functions be on the left hand side. This will prevent respecification to fit the program input format. This will become apparent with examination of the same problems.

II. THEORETICAL REVIEW

Although the program can be used in solving many different types of problems it was designed to facilitate the solution of sectoral models. The user is referred to Duloy and Norton, and Klein and Roe.[5]

The concept is that given "well-behaved" supply and demand functions, a market equilibrium price and quantity may be found by maximizing the area bounded on the right by the supply and demand curves.

Referring to Figure 1. Equilibrium Solution, the equilibrium price and quantity, p*, q*, may be found by discovering the quantity that maximizes (area A and area B).

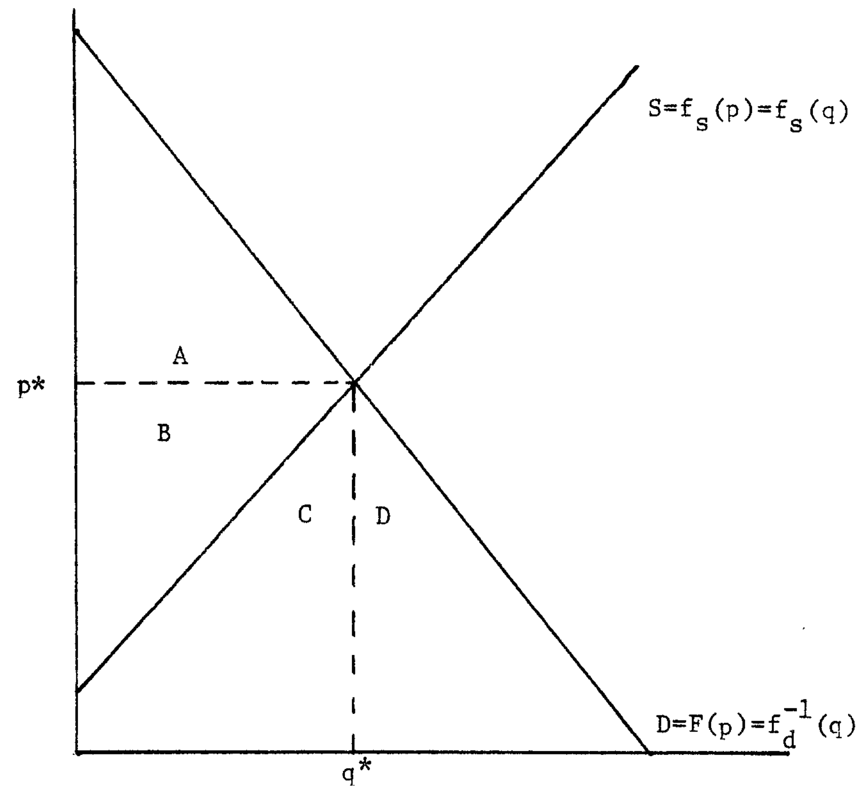


Figure 1. Equilibrium Solution.

Area A is the portion of the integral under the demand curve above p*, area B is the portion of the rectangle p*q* above the integral under the supply function. Under certain conditions these areas are commonly referred to as consumer and producer surplus, respectively. The total area, A + B = Z may be stated as follows:

$$(1) \quad Z = \int_0^{q^*} f_d^{-1}(q)dq - p^*q^* + p^*q^* - \int_0^{q^*} f_s(q)dq$$

where:

$p = f_d^{-1}(q)$ is an inverse demand function, where price is a function of quantity demanded; and

$MC = f_s(q)$ marginal cost is a function of the quantity supplied.

Rearranging the equation for Z,

$$(2) \quad Z = \int_0^{q^*} f_d^{-1}(q)dq - \int_0^{q^*} f_s \quad (q)dq.$$

Provided that Z is a quasi-concave function, is twice continuously differentiable, and in the domain of real numbers, q* may be found by maximizing Z with respect to q.

Applying the Kuhn-Tucker theorem, the necessary conditions for a maximum to exist are stated as follows:

$$(3) \quad \frac{\partial Z}{\partial q} = f_d^{-1}(q) - f_s(q) \leq 0 \text{ and } \frac{\partial Z}{\partial q}q \equiv 0$$

Rearranging, $f_d^{-1}(q) = f_s(q)$

Substituting p for $f_d^{-1}(q)$ and MC for $f_s(q)$ results in the competitive solution of price and marginal cost being equal.

One may easily complicate this problem by moving to an interregional trade problem. Similarly, the total cost function, $\int_0^q f_s(q)dq$ could be replaced with input supply functions and a production function.

Following Klein and Roe, the following simple nonlinear programming problem is specified, and then converted to a linear programming problem. Both equation and tableau specifications are provided for the linear problem. For a simple case, deriviation of the economic information embodied in the dual variables of the LP problem is provided.[6]

Let the demand function for the j^{th} commodity, $j = 1,\ldots, J$, be specified in inverse form as:

(4) $p_j = a_j - b_j q_j$

where q_j is the quantity demanded, a_j is the intercept, and b_j is the change in the quantity of q_j demanded given a change in its own price, p_j.

Let the supply side be specified by the following total cost and conversion equations.

(5) Let $q_j = m_j x_j$, $j = 1,\ldots, J$

where $m > 0$ is the conversion factor for x into q.

Let c_j be the unit cost of x_j, $j = 1,\ldots J$.

The nonlinear programming specification of this problem is

(6) $$\max_{q,x} Z = \sum_j \int_0^q (a_j - b_j q_j) dq_j - \sum_j c_j x_j + \sum_j \lambda_j (m_j x_j - q_j)$$

or in matrix form,

(7) $$\max_{Q,X} Z = Q'(A - .5BQ) - C'X + \lambda((MX)' - Q')$$

where:

Q is Jx1 of elements q_j

B is JxJ of elements b_j

C is Jx1 of elements c_j

X is Jx1 of elements x_j

λ is Jx1 of elements λ_j

M is JxJ of elements m_{ij}, all $m_{ij} = 0$ for $i \neq j$.

The procedure for linearizing the problem is to find the definite integral of each of the j demand equations,

$p_j = a_j - b_j q_j$, and evaluate the integrals for q_j varying over i, or

q_{ji}, $i = 1, \ldots, I$, over $j = 1, \ldots, J$, or

$w_{ji} = a_j q_{ji} - 0.5 b_j q^2$

For each commodity, q_j, the area under its demand curve is found for $i = 1$ to I steps. Each of these steps, w_{ji}, are to be activities in the linear programming format, and enter the solution at levels a_{ji}. Certain restrictions (to be explained) are placed on the a_{ji} in order to insure feasibility.

The linear programming problem may be stated as follows:

(8.1) $\max_{a,\,x} Z^\circ = \sum_{j}\sum_{i} a_{ji} w_{ji} - \sum_j c_j x_j$

Subject to the J commodity balance constraints

(8.2) $m_j x_j - \sum_i a_{ji} q_{ji} \geq 0 \quad j = 1, \ldots, J,$

and J convexity constraints,

(8.3) $\sum_i a_{ji} \leq 1 \quad j = 1, \ldots, J$

or $\max_{a,\,x} Z^\circ = \sum_{j}\sum_{i} a_{ji} w_{ji} - \sum_j c_j x_j + \sum_j \lambda (m_j x_j - \sum a_{ji} q_{ji}) + \sum_j \lambda_j^* (1 - \sum_i a_{ji})$

This problem is shown in tableau form below in Table 1..

The convexity constraints are crucial to the problem. Duloy and Norton have shown that if the nonlinear problem is concave, a nontrivial solution will exist where the following will hold for each of the j activities. Either,

(a) $a_{ji} = 1$, all other $a_{js} = 0$ for a particular j,

(b) $a_{ji} < 1$, all other $a_{js} = 0$ for a particular j, or

(c) $a_{ji} + a_{j(i+1)} = 1$ and all other $a_{js} = 0$, $s \neq i$, $i+1$. [7]

Table 1. Specification of Commodity Market Demand in Linear Programming Format.

Constraint Constants			Supply Activities (x)	Demand Activities (λ)	Dual
Commodity Balance	0	$\leq$	m	$-q_1$ $-q_2 \ldots$ $-q_I$	Market Price (π)
Convexity Constraint	1	$\geq$		1 1 ... 1	Consumer Surplus (π*)
Objective Function	Z =		− c	w_1 w_2 ... w_I	Consumer Plus Producer Surplus

Source: Klein, Harold E. and Terry L. Roe, "Agriculture Sector Analysis Model Design: The Influence of Administrative Infrustructure Characteristics," Table A.1, p. 299.

The implication is that depending on the difference between segments q_{ji} and $q_{j(i+1)}$ the solution to the linear problem, Z°, can be shown to be an arbitrarily close approximation of the solution to the nonlinear problem Z.

Given this arbitrary closeness of the linear to the nonlinear problem, it can be shown that the duals of commodity balance rows are equal to the prices, and that the duals of convexity constraints are equal to consumer surpluses. Case (a) is used for simplicity, otherwise the problem is complicated by combination of a_{ji}, or fractional values of a_{ji}.

For a positive a_{ji}, the Kuhn-Tucker conditions require that,

$$\frac{\partial Z^{\circ}}{\partial a_{ji}} = w_{ji} - \lambda_j q_{ji} - \lambda_j^* = 0.$$

For a basis variable, it follows from the nonlinear problem that

$$\frac{\partial Z}{\partial q_{ji}} = \frac{\partial (w_{ji})}{\partial q_{ji}} - \lambda_j = 0$$

$$= p_{ji} - \lambda_j = 0.$$

Therefore λ_j, the shadow price or dual for the commodity balance row is equal to the equilibrium commodity price.

Since a_{ji} is assumed to be one, and Z° is an approximation of Z, p_{ji} may be substituted for λ_j and

$$\frac{\partial Z^{\circ}}{\partial a_{ji}} = w_{ji} - p_{ji} q_{ji} - \lambda_j^* = 0.$$

That is, λ_j^*, the shadow price on the convexity constraint is shown to be the consumer surplus for q_{ji} at p_{ji}.

These results can be extended to the production side in the case of total cost expressed as an integral of marginal cost instead of average cost times quantity. In the case of production functions, it is asserted that the shadow prices on the convexity constraints are producer surpluses accrued to the holder of the processes .

It should be pointed out that fixed factors having a positive opportunity cost are also included in calculations of other relevant shadow prices. The same is true for any other form of price or quantity restriction. In order to determine exactly what is involved in the determination of a dual value, Kuhn-Tucker conditions should be stated for each problem, from which expressions for all dual values may be derived.

In summary:

1. The shadow prices on commodity balance constraints for demand functions are equilibrium market prices for the commodities.
2. The shadow prices on convexity constraints for demand functions are consumer surpluses associated with the commodities.
3. The shadow prices on factor balance constraints for supply functions are equilibrium market prices for the factors.
4. The shadow prices on convexity constraints for supply functions are producer surpluses associated with the factors.
5. The shadow prices on convexity constraints for production functions are producer surpluses associated with production of the commodities.

III. DATA ENTRY

In proceeding to the section explaining the data entry it should be useful for the user to have a broad view of how the program operates.

The first block of information includes the dimensions of the nonaugmented portion of the matrix, the algorithm and/or system desired (one of several MPOS algorithms or APEX). The objective function, constraints, and if an integer program, the integer variables are read in.

The second possible block of information is in conjunction with an option to read in a second data set to be inserted some place within the data set previously read in for the initial models. This option could be useful in the case of expanding the number of columns or rows somewhere in the middle of the nonaugmented portion of the matrix, without having to repunch a new data deck.

The third possible block of data includes the information necessary to generate linear activities approximating a nonlinear function. This block is further divided into two groups of functions and associated procedures.

The simpler of the two entails taking linear steps of a single variable function, and calculating the coefficients for the objective function and the row constraints. Examples of this type of function include supply and demand curves where quantity is a function of price. The program calculates the area under the curve at each quantity increment. These values are then placed into the objective and appropriate constraint specification by the algorithm.

The more complex of the two nonlinear functions involves the generation of an input substitution surface. An isoquant defining the relationship of an output, Q, two inputs X_1, and X_2, in Cobb-Douglas functional form is provided for. It is also conceivable that if Q were viewed as a composite consumption good, the surface could represent how X_1 and X_2 substitute in the consumption of Q. For example Q could be fruit, X_1 oranges, and X_2 apples, the program will calculate as many activities as necessary to satisfy the steps in Q desired.

Card Format

In moving through the data input cards, the user may wish to refer to the listing of variable names and options, the flow chart, and the program listing found in Appendices A, B, and C, respectively.

Input cards are listed in read statement form, each with its fortran format given. A short explanation is given where program branches occur, or where an explanation may otherwise be helpful.

1. READ (5,500) IDM, M1, COL, ROW

   500 FORMAT [I1, I2,2I5]

   IDM =0 for maximum

   =1 for minimum

   M1, algorithm within MPOS

   = 01, - REGULAR -, 2-phase simplex (LP)

   = 02, - REVISED -, revised simplex (LP)

   = 03, - DUAL -, dual simplex (LP)

   = 04, - MINIT -, primal-dual (LP)

   = 05, - BBMIP -, branch and bound mixed integer program (IP)

   = 06, - DSZ1IP -, direct search 0-1 integer program (IP)

   = 07, - GOMORY -, Gomory's cutting plane (IP)

   = 08, - WOLFE -, Wolfe's quadratic simplex (QP)

   = 09, - BEALE -, Beale's algorithm (QP)

   = 10, - LEMKE -, Lemke's complementary pivot algorithm (QP)

   = 11, - APEX 1 -, MPOS-APEX data file interface (GENERAL)

   = 12, - APEX 2 -, MPOS-APEX data file interface (GENERAL)

   COL, number of columns in nonaugmented matrix

   ROW, number of rows in nonaugmented matrix.

2. READ (5,501) TITLE

   501 FORMAT (8A10)

3. If the problem is an integer programming problem, the following cards are punched indicating the number of integer variables and variable names. If the problem is not IP, then the card block is left out.

   READ (5,503) N2, (ACT(I1), I1 = 2,N2)

   503 FORMAT (I3, 11A7/(3X, 11A7))

   N2, the member of integer variables

   ACT (I2), the activity names

4. Read in the nonaugmented or traditional LP activities

   READ (5,505) (ACT(IA), IA = 1, COL)

   505 FORMAT (3X, 11A7)

   ACT(IA), activity names

   COL, number of columns in nonaugmented matrix

5. Read in the nonzero coefficients of the objective function of the nonaugmented matrix. Activities such as transfer columns having no objective value need not be entered.

   READ (5,506) (ICOL(IB), SIGN(IB), COEF(IB), IB = J1, J1+4)

   506 FORMAT (5(I4, A1, F11.2))

   ICOL(IB), the integer number of the activity, ACT(IA) for which an objective value is entered. Numbers begin with the left hand side of the matrix with 1, and run consecutively up through COL.

   SIGN(IB), the sign, + or -, of the objective value

   COEF(IB), the real value of the objective function.

   Note that up to five such entries may be entered on each card.

   FLAG - Once all objective values are read in, or if there are no nonzero values associated with the nonaugmented matrix, then ICOL = -999. So at least one card, with entry -999 in the first four columns is necessary if any nonaugmented activities are entered.

6. Read in the constraints for the nonaugmented matrix.

   READ (5,506) (ICOL(IF), SIGN(IF), COEF(IF), IF = J2, J2+4)

   506 FORMAT (5(I4, A1, F11.2))

   Exactly as in the case of the objective function, only the nonzero coefficients need be entered. In order to signify the completion of input for each constraint, three possible values may be assigned to ICOL. These values coincide with the nature of the constraints.

ICOL = -100, ——— ≤ RHS constraint

ICOL = -200, ——— = RHS constraint

ICOL = -300, ——— ≥ RHS constraint

Just as in the case of the column coefficients, the right hand side parameter is entered with SIGN and COEF along with the appropriate ICOL value. No other indicator is necessary to signify the completion of constraint input.

If this block of cards complete the data input, it is followed by an end-of-file (EOF) card. This card is multiple punched, 7-8-9, in the first column, and completes the input.

7. Read in data for the insertion option.

```
READ (5,511) ISID
511 FORMAT (I5)
```

If new activities are to be inserted, ISID is given the value of 99999, and a subroutine called INSERT is called. If the user does not desire to use the insert option, a blank card is necessary.

If the insert option is used, the cards following ISID, and used by the subroutine INSERT are listed below.

1. Location of insertion

   ```
   READ (5,511) NINS
   511 FORMAT (I5)
   ```

   NINS is the column number of the existing nonaugmented matrix at which the new activities are to be inserted.

2. Number and name of inserted activities

   ```
   READ (5,503) NAA, (AACT(IA), IA = 1, NAA)
   503 FORMAT (I3, 11A7/(3X, 11A7))
   ```

NAA, the number of new activites to be inserted.

AACT, names of the new activities.

3. Read in objective of inserted activites.

READ (5,506) (AICOL(IB), ASIGN(IB), ACOEF(IB), IB = 1,NAA)

506 FORMAT (5(I4, A1, F11.2))

This input is identical in format to the objective data entered above. However unlike the earlier case in which only nonzero coefficents were entered, an objective value for each inserted activity must be entered.

AICOL, the column number of the inserted activity, beginning with 1.

ASIGN, the sign on the coefficient.

ACOEF, the objective coefficient.

4. Read in number of nonzero coefficients to be inserted.

READ (5,511) NBB

511 FORMAT (I5)

NBB, the number of nonzero constraint coefficients to be inserted.

5. Read in the coefficients

READ (5,512) (AEWROW(IX), AEWCOL(IX), AEWSIGN(IX), AEWCOEF(IX), IX = 1, NBB)

512 FORMAT (4(2I3, A1, F13.2))

AEWROW, row number of the coefficient.

AEWCOL, column number of the coefficient.

AEWSIGN, sign of the coefficient.

AEWCOEF, the coefficient.

Note, this option has not been tested and it is unclear whether or not the numbers for AEWROW and AEWCOL are row and column numbers of the new matrix. However, this appears to be the most logical first choice. As above, if this block of data is final, then an EOF card follows the insertion and the input is completed.

7. Read in information for extended functions from which the augmented portion of the matrix is composed.

This section is characterized by having two options. The first is to generate linear activites from a single nonlinear function, such as a supply or demand function, the second is to generate a substitution relationship between 2 variables according to an exponential function, such as a production function with 2 input variables. Data entry is given for both of these cases.

READ (5,510) EID, RM, IDPV(JA)

510 FORMAT (I5, F10.2, I5)

IDPV, flag for two variable function,

= 0, single variable function,

≠ 0, three variable function.

EID, for IDPV = 0, denotes the number of nonzero coefficients for activities in the nonaugmented matrix in the same row as the generated activities; for IDPV ≠ 0, EID = 3, denoting the number of rows necessary for the exponential function, one row each for X_1, X_2, and Y.

RM, the right-hand-side value for the extended row. This program is designed for the RHS value of an extended row to be either 1.0 or 0.0. For each set of activities generated, a convexity constraint is generated automatically having a RHS value of 1.0. If a RHS value of zero is desired then RM is given a value of zero. Although no example is readily available for which it may be useful, it is possible to enter a negative RHS value but not possible to enter a positive RHS value. In general, RM will be given a value of 0.0.

Case a. Single Variable Function

This type of function will require the use of a single quantity constraint row. In the case of a supply or demand function, the generated activities in the augmented portion of the matrix will require at least one transfer activity in the same row in the nonaugmented portion of the matrix. It is possible, however, to generate augmented activities with no other coefficients in the same row.

In this case, IDPV = 0, EID = K, where K is the number of nonzero coefficients for the row in the nonaugmented portion of the matrix, and RM = 0.0, unless a negative RHS is desired. Note that in the case where all three values equal zero, a blank card is still necessary for the program to proceed.

The following cards are punched in the case of IDPV = 0.

1. If EID ≠ 0, read in coefficients, otherwise, skip this card and proceed to 2.

   READ (5,506) (ICOL(II), SIGN(II), COEF(II), II = 1,EID)

   506 FORMAT (5(I4, A1, F11.2))

      ICO6, the number of columns in which the coefficient is to be entered.

      SIGN, the sign of the coefficient, + or -.

      COEF, the coefficient to be entered.

2. Read in mathematical function to be extended.

   The program is designed for input of exponential functions of the following form:

   $$W = C_1 X^{\alpha_1} + C_2 X^{\alpha_2} + \ldots + C_n X^{\alpha_n}.$$

   Note that in the case of X being a commodity for which a supply or demand function is defined, the equation entered is the integral of the supply or demand function. In the case of supply, the equation above would represent the total cost function associated with a marginal cost or supply function of the form:

$$\frac{\partial W}{\partial X} = MC = \alpha_1 C_1 X^{\alpha_1 - 1} + \alpha_2 C_2 X^{\alpha_2 - 2} + \ldots + \alpha_n C_n X^{\alpha_n - 1} .$$

If the first term were an intercept, α_1 would have the value of 1, so that the value would simply be C_1.

```
READ (5,508) IJ, (CC(IK), IEXP(IK), IK = 1,IJ)
508 FORMAT (I5,5(F10.4, F5.0))
```

IJ, the number of terms in the function.

CC, the coefficients C_i for the function.

IEXP, the exponents α_i for the function.

3. Read in the initial value, the magnitude, and number of steps to be taken in the linearization procedure.

```
READ (5,509) A(1), DELTA Q, STEP
509 FORMAT (2F10.4, I5)
```

Q(1), initial value of the function.

DELTAQ, the increment value, $(Q_i - Q_{i-1}) \; \forall_i = 1, n$.

STEP, the number of steps, n, taken.

From the function and linearization information, the program adds the number of columns consistent with the number of steps, and calculates the area under the function at each step for the objective function. Two constraints are generated, a quantity allocation constraint containing the quantity steps specified, and a convexity constraint.

All quantity steps are generated having negative signs. The direction of the constraint is determined by the program to be, $\leq$, in the case of a supply function, $\geq$, in the case of a demand function.

Case b. Multiple Variable Function

As stated above, the most obvious use of this option is to incorporate a production function where two inputs, X_1 and X_2, are combined in the production of various quantities of some Y, specified by a Cobb-Douglas type function.

The concept used is to define several input ratios, or expansion paths at various levels of Y. From the ratio and Y values, values for X_1 and X_2 are determined. The calculation of the ratios follow:

$Y = AX_1^{\alpha_1}X_2^{\alpha_2}$

$RATIO = (X_1/X_2)$, rearranging

$X_1 = X_2R$, where $R = RATIO$.

Substituting for X_1, and solving for X_2.

$Y = A(X_2R)^{\alpha_1}X_2^{\alpha_2}$

$Y = AX_2^{\alpha_1}R^{\alpha_1}X_2^{\alpha_2}$

$X_2^{\alpha_1+\alpha_2} = YA^{-1}R^{-\alpha_1}$

$X_2 = (YA^{-1}R^{-\alpha_1})^{\frac{1}{\alpha_1+\alpha_2}}$

For each ratio r_i, $i = 1,\ldots,n$, and for varying levels of Y, a unique value of X_2 is calculated which in turn determines the appropriate value of X_1. This grid linearization is illustrated in Figure 2. The Linearized Specification of $Y = f(X_1,X_2)$.

The number of activities generated in the number of steps in Y times the number of ratios. Three quantity constraint rows, one each for X_1, X_2, and Y, and a convexity constraint row are generated by the program. Zeroes are placed in the objective function.

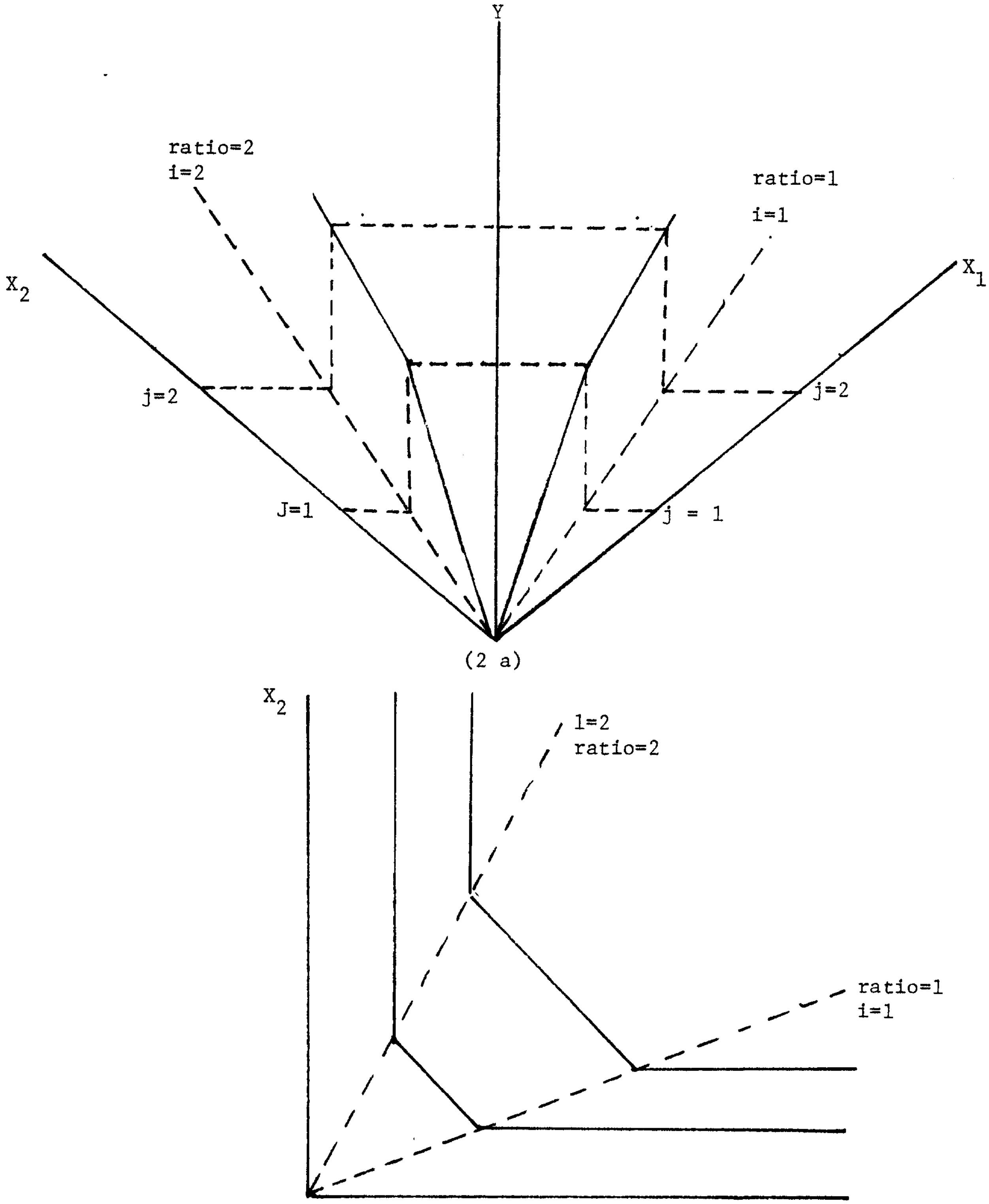


Figure 2. The Linearized Specification of $Y=f(X_1,X_2)$.

Source: Roe, Terry. "Modeling of Nonlinear Functions into A Linear Programming Format". Staff Paper P75-9 [8]

Values for the first card of this group are as follows:

IDPV ≠ 0, the value 99999 is given in some sample decks

EID = 3, the number of quantity rows to be generated

RM = 0.0, right hand side values.

For each of the three rows, the following sequence of cards is necessary.

1. Read in the number of coefficients in the row in the nonaugmented portion of the matrix.

   READ (5,507) IEID

   507 FORMAT (I5).

   IEID, the number of coefficients.

2. Read in the column, sign, and value of the coefficients

   READ (5,506) (ICOL(I1), SIGN(I1), COEF(I1), I1=1, IEID(I3)+1)

   506 FORMAT (5(I4, A1, F11.2))

   ICOL, the column number in which the coefficient is to be entered.

   SIGN, the sign of the coefficient.

   COEF, the value of the coefficient.

   Important! Note that the right hand side value and the constraint type must be entered by the user. Therefore the final entry will have one of the following values for ICOL:

   ICOL = -100 ——— ≤ RHS constraint,

   ICOL = -200 ——— = RHS constraint,

   ICOL = -300 ——— ≥ RHS constraint.

3. Read in information pertaining to the ratios to be used in generating activities.

   READ (5,584) NOR,(RATIO(I2), I2=1, NOR)

   584 FORMAT (I3, 11F7.2/(3X, 11F7.2))

   NOR, the number of ratios.

   RATIO, the ratio, (X_1/X_2).

4. Read in the Cobb-Douglas function parameters

   $Q = AAX_1^{\alpha_1} X_2^{\alpha_2}$

   READ (5,585) AA, ALPH1, ALPH2

   585 FORMAT (3F10.2)

   AA, the multiplicative coefficient

   ALPH1, the exponent on X_1.

   ALPH2, the exponent on X_2.

5. Read in the initial value, the magnitude, and the number of steps to be taken.

   READ (5,509) Q(1), DELTAQ, STEP

   509 FORMAT (2F10.4, I5)

   Q(1), initial value of the function.

   DELTAQ, the increment value.

   STEP, the number of steps taken.

This concludes the data input section. It should be pointed out that once the input is complete, end of file (EOF) card is required. This card is punched 7-8-9 in the first column.

IV. SAMPLE PROBLEMS

For illustrative purposes, two sample problems developed by Roe are provided. The first problem is stated in nonlinear form and then restated in linear form. Results concerning the values of shadow prices on commodity balance and convexity constraints are provided.

Provided for both problems are verbal and mathematical specification, tableau representation, data input deck, MPOS specification, and finally MPOS summary of results.

Problem One

The first problem is one of maximizing the sum of producers' and consumers' surplus, with a variety of perfectly inelastic and elastic, and sloping supply and demand functions, and a production function.

Three commodities which are perfectly inelastically supplied may be combined. One of these inputs and an input supplied with an upward sloping function, may be combined to produce another commodity. The produced commodity faces a downward sloping demand.

The nonlinear programming specification of the problem follows:

$$\text{MAX } Z = .5\text{LACT1} + .9\text{LACT2} + .7\text{LACT3} - 1.5X_1$$

$$- \int_0^{X_2} (X_2)dX_2 + \int_0^{Y} (90-1.2Y)dY$$

$$+ \lambda_1 (90 - .4\text{LACT1} - .3\text{LACT2})$$

$$+ \lambda_2 (80 - .3\text{LACT1} - .2\text{LACT3})$$

$$+ \lambda_3 (200 - .4\text{LACT2} - .9\text{LACT3} - X_1)$$

$$+ \lambda_4 (4X_1^{.3} X_2^{.5} - Y)$$

where:

$\bar{X}_2$ = supply or marginal cost of X_2, and

$90 - 1.2Y$ = Inverse demand function for Y.

The nonlinear programming problem is now converted to a linear programming problem. Notice that each of the constraints stated in Lagrangian form corresponds exactly to a row constraint in the tableau specification of the problem found below.

$$\max_{L,X_1,a} Z^o = .5LACT1 + .9LACT2 + .7LACT3 - 1.5X_1 - \sum_{m=1}^{M} a_m \gamma_m + \sum_{n=1}^{N} a_n w_n$$

$$+ \lambda_1 (90 - .4LACT1 - .3LACT2)$$

$$+ \lambda_2 (80 - .3LACT1 - .2LACT3)$$

$$+ \lambda_3 (200 - .4LACT2 - .9LACT3 - X_1)$$

$$+ \lambda_4 (X2 - \sum_{m=1}^{M} a_m X_{2m})$$

$$+ \lambda_5 (1 - \sum_{m=1}^{M} a_m)$$

$$+ \lambda_6 (\sum_{p=1}^{P} a_p X_{1p} - X_1)$$

$$+ \lambda_7 (\sum_{p=1}^{P} a_p X_{2p} - X_2)$$

$$+ \lambda_8 (\sum_{p=1}^{P} a_p Y_p - Y)$$

$$+ \lambda_9 \left(1 - \sum_{p=1}^{P} a_p\right)$$

$$+ \lambda_{10} \left(Y - \sum_{n=1}^{N} a_n Y_n\right)$$

$$+ \lambda_{11} \left(1 - \sum_{n=1}^{N} a_n\right)$$

Where:

$\gamma_m = \int_0^{X2m} (X2) dX_2 = [0.5X2^2]_0^{X2m}$, the area under marginal cost curve, or total cost, of X2, at the m^{th} quantity of X2.

$W_n = \int_0^{Y_n} (90 - 1.2Y) dY = 90Y - 0.6Y^2$, the area under the demand function (marginal revenue under competitive assumptions), or total revenue for Y, at the n^{th} quantity of Y.

a_m, the level at which the m^{th} quantity steps of X_2 is supplied in the solution

a_n, the level at which the n^{th} quantity step of Y is demanded in the solution.

a_p, the level at which the p^{th} step in the production of Y, from inputs X_1 and X_2, enters the solution. Note that the index P embodies both ratios and quantities. Referring to Figure 2 may be of some help. Given a particular input ratio i, as quantities j of Y are changed, quantities of X_1 and X_2 change accordingly. Therefore the index p runs over both ratio numbers, and the quantity steps in Y, or P = (ratios)(M).

Linearization Parameters, Problem 1.

Supply of X_2:

Total cost: $W = -0.5X2^2$ area under supply

Initial $X_2 = 0$

$\Delta X_2 = 10$

STEPS = 12

Production of Y

$Y = 4X_1^{.3}\ X_2^{.5}$

Initial Y = 10

$\Delta Y = 10$

STEPS = 5

RATIOS

R1 = .4 R2 - .8 R3 = 1.4 R4 = 1.8

Demand for Y

W = 90Y-.6Y (area under demand)

Initial Y = 0

$\Delta Y = 7.5$

STEPS = 12

The matrix, data input, computer specification and results follows:

Table 2. Problem 1 Tableau Specification

No. 1 TEST TRAD. LP, INPUT, SUPPLY, PRODUCTION FUNCTION AND DEMAND

	LACT 1	LACT 2	LACT 3	X1T	X2T	YT	SUPPLY OF X2	PRODUCTION	DEMAND, Y		
OBJECTIVE	.5	.9	.7	-1.5	0	0	$\tilde{Y}_1$ $-\tilde{Y}_2$ .... $-\tilde{Y}_2$	0 0	W_1 .... W_n		
CONSTRAINT							Nonaugmented submatrix				
1.	.4	.3								$\leq$	90
2.	.3		.2							$\leq$	80
3.		.4	.9	1				Augmented submatrix		$\leq$	200
4.					1		$-\tilde{X2}_1$ $-\tilde{X2}_2$.....$-\tilde{X2}_n$			$\leq$	0
5.							1 1 1			$\leq$	1
6.				-1				$X1_1$..... $X1_p$		$\leq$	0
7.					-1			$X2_1$..... $X2_p$		$\leq$	0
8.						-1		$\tilde{Y}_1$ ..... $\tilde{Y}_p$		$\geq$	0
9.								1 1		$\leq$	1
10.						1			$-Y_1$ ....$-Y_n$	$\geq$	0
11.									1 1	$\leq$	1

Problem One Input

```
001      6     3
** NO. 1 = TEST TRAD, LP,INPUT,SUPPLY,PRODUCTION FUNC AND DEMAND SPECIFICATION,
    LACT1  LACT2  LACT3  X1T     X2T     YT
    1+0.5              2+0.9            3+0.7          4-1.5        -999
    1+0.4              2+0.3        -100+90.0
    1+0.3              3+0.2        -100+80.0
    2+0.4              3+0.9            4+1.0       -100+200.0

      1
    5+1.0
      1-0.5        2.0
0.0         10.0           12
     3              99999
      1
    4-1.0          -100+0.0
      1
    5-1.0          -100+0.0
      1
    6-1.0          -300+0.0
   40.4      0.8      1.4      1.8
4.0          0.3           0.5
10.0        10.0           5
      1
    6+1.0
      290.0       1.0    -0.6      2.0
0.0          7.5           12
>
>
>
```

An explanation for each card follows

Problem One, Input Explanation

1. maximum problem, MPOS regular algorithm, 6 columns and 3 rows in the nonaugmented matrix
2. title
3. variable names
4. objective, nonaugmented
5. row 1, coefficients by column, constraint type, RHS value
6. row 2, coefficients by column, constraint type, RHS value
7. row 3, coefficients by column, constraint type, RHS value
8. blank card for no data insertion

   9-12 generate supply of X2

9. 1 nonzero coefficient in nonaugmented portion of row 4
10. entry of row 4, column 5, equal to 1.0.
11. integrated supply function, 1 term, coefficient = -0.5, exponent = 2.0
12. initial step = 0.0, increment = 10, 12 steps

    13-22 generate production surface $Y = AX^{\alpha 1} X^{\alpha 2}$

13. 3 rows generated, 99999 = DEPV subroutine
14. 1 nonzero coefficient in row 6, (X1)
15. row 6, coefficient by column, constraint type, RHS value
16. 1 coefficient row 7, (X2)
17. row 7
18. 1 coefficient row 8, (Y)
19. row 8
20. 4 ratios, $\gamma 1 = 0.4$, $\gamma 2 = 0.8$, $\gamma 3 = 1.4$, $\gamma 4 = 1.8$
21. production function $Y = 4X_1^{0.3} X_2^{0.5}$
22. initial Y = 10, increment = 10, 5 steps

    23-26 generate demand for Y

23. 1 coefficient row 10

24. row 10 coefficient

25. integrated demand function, 2 terms, coefficient 1 = 90, exponent 1 = 1.0, coefficient 2 = -0.6, exponent 2 = 2.0

26. initial Y = 0.0, increment = 7.5, 12 steps

Following is the computer output generated for problem one.

```
MPOS VERSION 4.0        NORTHWESTERN UNIVERSITY
     **********************************
     *                                *
     *            M P O S             *
     *                                *
     *          VERSION 4.0           *
     *                                *
     * MULTI-PURPOSE OPTIMIZATION SYSTEM *
     *                                *
     **********************************

**** PROBLEM NUMBER  1 *****

    REGULAR
    TITLE
    ** NO. 1 # TEST TRAD. LP,INPUT,SUPPLY,PRODUCTION FUNC AND DEMAND SPECIFICATION
    VARIABLES
      LACT1     LACT2     LACT3     X1T       X2T       YT        YA1       YA2
      YA3       YA4       YA5       YA6       YA7       YA8       YA9       YA10
      YA11      YA12      YB1       YB2       YB3       YB4       YB5       YB6
      YB7       YB8       YB9       YB10      YB11      YB12      YB13      YB14
      YB15      YB16      YB17      YB18      YB19      YB20      YC1       YC2
      YC3       YC4       YC5       YC6       YC7       YC8       YC9       YC10
      YC11      YC12
    MAXIMIZE
    +          .50000LACT1   +          .90000LACT2   +          .70000LACT3
    -         1.50000X1T
    +               0YA1     -        50.00000YA2     -       200.00000YA3
    -       450.00000YA4     -       800.00000YA5     -      1250.00000YA6
    -      1800.00000YA7     -      2450.00000YA8     -      3200.00000YA9
    -      4050.00000YA10    -      5000.00000YA11    -      6050.00000YA12
    +               0YB1     +               0YB2     +               0YB3
    +               0YB4     +               0YB5     +               0YB6
    +               0YB7     +               0YB8     +               0YB9
    +               0YB10    +               0YB11    +               0YB12
    +               0YB13    +               0YB14    +               0YB15
    +               0YB16    +               0YB17    +               0YB18
    +               0YB19    +               0YB20    +               0YC1
    +       641.25000YC2     +      1215.00000YC3     +      1721.25000YC4
    +      2160.00000YC5     +      2531.25000YC6     +      2835.00000YC7
    +      3071.25000YC8     +      3240.00000YC9     +      3341.25000YC10
    +      3375.00000YC11    +      3341.25000YC12
    CONSTRAINT
 1. +          .40000LACT1   +          .30000LACT2
      .LE.  +         90.00000
 2. +          .30000LACT1   +          .20000LACT3
      .LE.  +         80.00000
 3. +          .40000LACT2   +          .90000LACT3   +         1.00000X1T
      .LE.  +        200.00000
 4. +         1.00000X2T
    -               0YA1     -        10.00000YA2     -        20.00000YA3
    -        30.00000YA4     -        40.00000YA5     -        50.00000YA6
    -        60.00000YA7     -        70.00000YA8     -        80.00000YA9
    -        90.00000YA10    -       100.00000YA11    -       110.00000YA12
      .LE.  +               0
 5. +         1.00000YA1     +         1.00000YA2     +         1.00000YA3
    +         1.00000YA4     +         1.00000YA5     +         1.00000YA6
    +         1.00000YA7     +         1.00000YA8     +         1.00000YA9
```

```
      MPOS VERSION 4.0        NORTHWESTERN UNIVERSITY

    *********************
    * PROBLEM NUMBER  1 *
    *********************

    USING REGULAR
    ** NO. 1 * TEST TRAD. LP,INPUT,SUPPLY,PRODUCTION FUNC AND DEMAND SPECIFICATION *
              +          1.00000YA10    +          1.00000YA11    +          1.00000YA12
               .LE.      +         1.00000
       6.     -          1.00000X1T
              +          1.77300YB1     +          4.21700YB2     +          7.00000YB3
              +         10.03000YB4     +         13.25600YB5     +          2.73400YB6
              +          6.50300YB7     +         10.79600YB8     +         15.65900YB9
              +         20.44400YB10    +          3.97000YB11    +          9.23700YB12
              +         15.31600YB13    +         21.94500YB14    +         29.00500YB15
              +          4.53900YB16    +         10.79500YB17    +         17.02300YB18
              +         25.67700YB19    +         33.93600YB20
               .LE.      +              0
       7.     -          1.00000X2T
              +          4.43300YB1     +         10.54200YB2     +         17.50100YB3
              +         25.07400YB4     +         33.14100YB5     +          3.41900YB6
              +          8.12900YB7     +         13.99500YB8     +         19.35500YB9
              +         25.55500YB10    +          2.77100YB11    +          6.59000YB12
              +         10.94000YB13    +         15.67500YB14    +         20.71500YB15
              +          2.52200YB16    +          5.99500YB17    +          9.05600YB18
              +         14.26500YB19    +         18.85400YB20
               .LE.      +              0
       8.     -          1.00000YT
              +         10.00000YB1     +         20.00000YB2     +         30.00000YB3
              +         40.00000YB4     +         50.00000YB5     +         10.00000YB6
              +         20.00000YB7     +         30.00000YB8     +         40.00000YB9
              +         50.00000YB10    +         10.00000YB11    +         20.00000YB12
              +         30.00000YB13    +         40.00000YB14    +         50.00000YB15
              +         10.00000YB16    +         20.00000YB17    +         30.00000YB18
              +         40.00000YB19    +         50.00000YB20
               .GE.      +              0
       9.     +          1.00000YB1     +          1.00000YB2     +          1.00000YB3
              +          1.00000YB4     +          1.00000YB5     +          1.00000YB6
              +          1.00000YB7     +          1.00000YB8     +          1.00000YB9
              +          1.00000YB10    +          1.00000YB11    +          1.00000YB12
              +          1.00000YB13    +          1.00000YB14    +          1.00000YB15
              +          1.00000YB16    +          1.00000YB17    +          1.00000YB18
              +          1.00000YB19    +          1.00000YB20
               .LE.      +         1.00000
      10.     +          1.00000YT
              -                0YC1     -          7.50000YC2     -         15.00000YC3
              -         22.50000YC4     -         30.00000YC5     -         37.50000YC6
              -         45.00000YC7     -         52.50000YC8     -         60.00000YC9
              -         67.50000YC10    -         75.00000YC11    -         82.50000YC12
               .GE.      +              0
      11.     +          1.00000YC1     +          1.00000YC2     +          1.00000YC3
              +          1.00000YC4     +          1.00000YC5     +          1.00000YC6
              +          1.00000YC7     +          1.00000YC8     +          1.00000YC9
              +          1.00000YC10    +          1.00000YC11    +          1.00000YC12
               .LE.      +         1.00000
              OPTIMIZE
```

MPOS VERSION 4.0 NORTHWESTERN UNIVERSITY

```
************************
* PROBLEM NUMBER   1 *
************************
```

USING REGULAR
** NO. 1 N TEST TRAD. LP,INPUT,SUPPLY,PRODUCTION FUNC AND DEMAND SPECIFICATION**

SUMMARY OF RESULTS

VAR NO	VAR NAME	ROW NO	STATUS	ACTIVITY LEVEL	OPPORTUNITY COST	LOWER BOUND	UPPER BOUND
1	LACT1	--	LB	0.0000000	.2851852	0.0000	INF
2	LACT2	--	B	300.0000000	0.0000000	0.0000	INF
3	LACT3	--	B	51.1800000	0.0000000	0.0000	INF
4	X1T	--	B	33.9350000	0.0000000	0.0000	INF
5	X2T	--	B	18.6540000	0.0000000	0.0000	INF
6	Y1	--	B	50.0000000	0.0000000	0.0000	INF
7	YA1	--	LB	0.0000000	100.0000000	0.0000	INF
8	YA2	--	B	.1146000	0.0000000	0.0000	INF
9	YA3	--	B	.8854000	0.0000000	0.0000	INF
10	YA4	--	LB	0.0000000	100.0000000	0.0000	INF
11	YA5	--	LB	0.0000000	300.0000000	0.0000	INF
12	YA6	--	LB	0.0000000	600.0000000	0.0000	INF
13	YA7	--	LB	0.0000000	1000.0000000	0.0000	INF
14	YA8	--	LB	0.0000000	1500.0000000	0.0000	INF
15	YA9	--	LB	0.0000000	2100.0000000	0.0000	INF
16	YA10	--	LB	0.0000000	2800.0000000	0.0000	INF
17	YA11	--	LB	0.0000000	3600.0000000	0.0000	INF
18	YA12	--	LB	0.0000000	4500.0000000	0.0000	INF
19	YB1	--	LB	0.0000000	970.4202778	0.0000	INF
20	YB2	--	LB	0.0000000	752.6291667	0.0000	INF
21	YB3	--	LB	0.0000000	548.3462222	0.0000	INF
22	YB4	--	LB	0.0000000	353.0428889	0.0000	INF
23	YB5	--	LB	0.0000000	167.1960000	0.0000	INF
24	YB6	--	LB	0.0000000	957.3042222	0.0000	INF
25	YB7	--	LB	0.0000000	721.6341667	0.0000	INF
26	YB8	--	LB	0.0000000	495.9026667	0.0000	INF
27	YB9	--	LB	0.0000000	280.1444444	0.0000	INF
28	YB10	--	LB	0.0000000	69.7796667	0.0000	INF
29	YB11	--	LB	0.0000000	950.2872778	0.0000	INF
30	YB12	--	LB	0.0000000	704.7578333	0.0000	INF
31	YB13	--	LB	0.0000000	468.8732222	0.0000	INF
32	YB14	--	LB	0.0000000	230.9076111	0.0000	INF
33	YB15	--	LB	0.0000000	15.7237222	0.0000	INF
34	YB16	--	LB	0.0000000	948.0556111	0.0000	INF
35	YB17	--	LB	0.0000000	699.4473667	0.0000	INF
36	YB18	--	LB	0.0000000	460.0468333	0.0000	INF
37	YB19	--	LB	0.0000000	227.3492778	0.0000	INF
38	YB20	--	B	1.0000000	0.0000000	0.0000	INF
39	YC1	--	LB	0.0000000	1417.5000000	0.0000	INF
40	YC2	--	LB	0.0000000	1012.5000000	0.0000	INF
41	YC3	--	LB	0.0000000	675.0000000	0.0000	INF
42	YC4	--	LB	0.0000000	405.0000000	0.0000	INF
43	YC5	--	LB	0.0000000	202.5000000	0.0000	INF
44	YC6	--	LB	0.0000000	67.5000000	0.0000	INF
45	YC7	--	B	.3333333	0.0000000	0.0000	INF

MPOS VERSION 4.0 NORTHWESTERN UNIVERSITY

```
**********************
* PROBLEM NUMBER  1  *
**********************
```

USING REGULAR
** NO. 1 # TEST TRAD. LP,INPUT,SUPPLY,PRODUCTION FUNC AND DEMAND SPECIFICATION**

SUMMARY OF RESULTS

VAR NO	VAR NAME		ROW NO	STATUS	ACTIVITY LEVEL	OPPORTUNITY COST	LOWER BOUND	UPPER BOUND
46	YC8		--	B	.6666667	0.0000000	0.0000	INF
47	YC9		--	LB	0.0000000	67.5000000	0.0000	INF
48	YC10		--	LB	0.0000000	202.5000000	0.0000	INF
49	YC11		--	LB	0.0000000	405.0000000	0.0000	INF
50	YC12		--	LB	0.0000000	675.0000000	0.0000	INF
51	SLACK--	D-	1	LB	0.0000000	1.9629630	0.0000	INF
52	SLACK--	D-	2	B	69.7640000	0.0000000	0.0000	INF
53	SLACK--	D-	3	LB	0.0000000	.7777778	0.0000	INF
54	SLACK--	D-	4	LB	0.0000000	15.0000000	0.0000	INF
55	SLACK--	D-	5	LB	0.0000000	100.0000000	0.0000	INF
56	SLACK--	D-	6	LB	0.0000000	2.2777778	0.0000	INF
57	SLACK--	D-	7	LB	0.0000000	15.0000000	0.0000	INF
58	SLACK--		8	LB	0.0000000	31.5000000	0.0000	INF
69	ARTIF--	D-	8	LB	0.0000000	-31.5000000	0.0000	INF
59	SLACK--	D-	9	LB	0.0000000	1214.8867773	0.0000	INF
60	SLACK--		10	LB	0.0000000	31.5000000	0.0000	INF
71	ARTIF--	D-	10	LB	0.0000000	-31.5000000	0.0000	INF
61	SLACK--	D-	11	LB	0.0000000	1417.5000000	0.0000	INF

MAXIMUM VALUE OF THE OBJECTIVE FUNCTION = 3064.609000

CALCULATION TIME WAS .0930 SECONDS FOR 19 ITERATIONS.

DATA STORAGE MEMORY =002037(OCTAL) TOTAL MEMORY = 050000(OCTAL)
TOTAL TIME FOR THIS PROBLEM WAS .709 SECONDS

<u>Output Interpretation - Problem One</u>

A brief description of the output follows. Those needing further explanation should refer to the MPOS manual.[2]

REGULAR - The particular MPOS algorithm requested by the user.

TITLE - Followed by user provided title

VARIABLES - The names of variables in the nonaugmented position of the matrix are given first, followed by the augmented or generated variables. Variables associated with the supply of X2 YA1 through YA12; production, YB1 to YB20; demand for Y, YC1 to YC12.

MAXIMIZE - The type of optimization requested for the objective function that follows. Note that the sign and objective value for each variable is provided.

CONSTRAINT - Followed by each row constraint in the problem. Variables having zero coefficients are not listed. Note that constraint (4) is the commodity balance row for X2. The values at each step are the quantities associated with the total cost values in the objective function. Constraint (5) is the convexity constraint for X2.

Summary of Results

For each variable the following information is provided.

STATUS - Whether the variable is in the optimal basis at a zero or positive value. LB indicates zero; B positive value.

ACTIVITY LEVEL - The level or value a variable takes on in the optimal solution.

OPPORTUNITY COST - The cost in terms of a change in the objective function given a marginal increase in the particular variable. This item is used synonymously with shadow price or dual.

LOWER, UPPER BOUNDS - The lower and upper limits of a variable within which the opportunity cost is unchanged.

Interpretation of the slack variables of the row constraints is the equivalent of finding the values of the dual problem.

The first three slack variables are associated with row constraints of the nonaugmented portions of the matrix. They are the commodity balance rows of fixed resources.

STATUS - Whether the slack variable is in the optimal basis at a zero or positive level. LB indicates zero; B a positive value. That is, SLACK takes on a positive value only when the resource is not totally exhausted.

OPPORTUNITY COST - The change in the objective function given an additional unit of the commodity constrained, or the value of an additional unit of the commodity.

The resource in row 2 has a positive value (69.76), indicating that it is not used up or is not a constraining resource. Since it is not constraining, its worth or value is zero as indicated in the opportunity cost column.

Additional units of the resources in rows 1 and 3 would be worth \$1.96 and \$0.78 respectively.

The interpretation of the Lagrangians or dual values of the rows of the augmented portion of the matrix was discussed in the theory review above. Again this value is given as the opportunity cost here. These values, taken from the computer output are listed below.

$\lambda 4$ - dual row 4 - price of X_2 -	15.00
$\lambda 5$ - dual row 5 - producer surplus X_2 -	100.00
$\lambda 9$ - dual row 9 - producer surplus Y -	1214.89
$\lambda 10$ - dual row 10 - price of Y -	31.50
$\lambda 11$ - dual row 11 - consumer surplus Y -	1417.50

The value of the objective function is 3064.61.

Problem Two

In problem two, surplus is maximized from two limited resources, which may be combined by two different production functions into a commodity facing a downward sloping demand.

The nonlinear programming specification follows:

$$\text{Max} \quad Z = 2.0X_{11}-2.0X_{12}-1.8X_{21}-1.8X_{22}+\int_0^Y(90-1.2Y)dY$$

$$+\lambda_1\ (25-X_{21}-X_{22})$$

$$+\lambda_2\ (75-X_{11}-X_{12})$$

$$+\lambda_3\ (Y-4X_{11}^{.3}X_{21}^{.5}-3X_{12}^{.6}X_{22}^{.15})$$

Linearization Parameters

Production Y_1

$Y_1 = 4X_{11}^{.3}\ X_{21}^{.5}$

Initial Y_1 = 10

ΔY_1 = 10

STEPS = 5

RATIOS

R1 = .4 R2 = .8 R3 = 1.4 R4 = 1.8

Production Y_2

$Y_2 = 3X_{12}^{.6}\ X_{22}^{.15}$

Initial Y_2 = 10

ΔY_2 = 10

STEPS = 5

Demand for Y

$W = 90Y - .6Y^2$ area under demand

Initial Y = 0

ΔY = 5

STEPS = 12

The matrix, data input, computer specification and results follow:

Table 3. Problem 2 Tableau Specification

NO. 2 TEST TWO PROD. FNTS. AND ONE DEMAND

	Y1T	TY2	TX11	TX12	TX21	TX22	PROD 1	PROD 2	DEMAND		
OBJECTIVE	0	0	-2.0	-2.0	-1.8	-1.8	0...0	0...0	$W_1 \ldots W_n$		
CONSTRAINT											
1.					1	1				$\leq$	25
2.			1	1						$\leq$	75
3.			-1				$X_{11} \ldots X_{1n}$			$\leq$	0
4.					-1		$X_{21} \ldots X_{2n}$			$\leq$	0
5.	-1						$Y_{11} \ldots Y_{1n}$			$\geq$	0
6.							1...1			$\leq$	1
7.				-1				$X_{11} \ldots X_{1n}$		$\leq$	0
8.						-1		$X_{21} \ldots X_{2n}$		$\leq$	0
9.		-1						$Y_{21} \ldots Y_{2n}$		$\geq$	0
10.								1...1		$\leq$	1
11.	1	1							$-Y_1 \ldots -Y_n$	$\geq$	0
12.									1...1	$\leq$	1

Input Deck - Problem 2

```
001     6    2
*** NO. 2 = TEST TWO PROD. FNTS. AND ONE DEMAND ***
    TY2     TY1     TX11    TX12    TX21    TX22
    3-2.0           4-2.0           5-1.8                   6-1.8          -999
    5+1.0           6+1.0          -100+25.0
    3+1.0           4+1.0          -100+75.0

     3            99999
     1
    3-1.0          -100+0.0
     1
    5-1.0          -100+0.0
     1
    2-1.0          -300+0.0
   40.4      0.8     1.4     1.8
4.0          0.3          0.5
10.0         10.0           5
     3            99999
     1
    4-1.0          -100+0.0
     1
    6-1.0          -100+0.0
     1
    1-1.0          -300+0.0
   4+0.4     +0.8    1.4     1.8
3.0          0.6          0.15
10.0         10.0           5
     2
    1+1.0           2+1.0
     2+90.0      1.0  -0.6        2.0
0.0          5.0            18
>
```

Problem Two Input Explanation

1. maximum, MPOS regular algouthm, 6 columns and 2 rows in the nonaugmented matrix.
2. title
3. variable names
4. objective, nonaugmented
5. row 1, coefficients by column, constraint type, RHS value
6. row 2, coefficients by column, constraint type, RHS value
7. blank card for no data insertion

   8-17 generate production surface $Y_1 = 4X_{11}^{.3}\ X_{21}^{.5}$
8. 3 rows generated, 99999 = DEPV subroutine
9. 1 nonzero coefficient in row 3, (XII)
10. row 3, coefficient by column, constraint type, RHS value
11. 1 nonzero coefficient in row 4, (X_{21})
12. row 4, coefficient by column, constraint type, RHS value
13. 1 nonzero coefficient in row 5, (Y_1)
14. row 5, coefficient by column, constraint type, RHS value
15. 4 ratios,
16. production function $Y_1 = 3X_{11}^{0.3}\ X_{21}^{0.5}$
17. initial Y=10, increment =10, 5 steps

   18-27 generate production surface $Y_2 = 3X_{12}^{0.6}\ X_{22}^{0.15}$ similar to 8-17

   28-31 generate demand for Y
28. 2 nonzero coefficients in row 11 (Y_1 and Y_2)
29. row 11, coefficients by column
30. integrated demand function, 2 terms, coefficient 1 = 90, exponent 1 = 1.0, coefficient 2 = -0.6, exponent 2 = 2.0
31. initial Y = 0.0, increment = 5.0, 18 steps

```
MPOS VERSION 4.0        NORTHWESTERN UNIVERSITY

     ***********************************
     *                                 *
     *            M P O S              *
     *                                 *
     *          VERSION 4.0            *
     *                                 *
     * MULTI-PURPOSE OPTIMIZATION SYSTEM *
     *                                 *
     ***********************************

***** PROBLEM NUMBER   1 *****

    REGULAR
    TITLE
    *** NO. 2 # TEST TWO PROD. FNTS, AND ONE DEMAND ***
    VARIABLES
      TY2      TY1      TX11     TX12     TX21     TX22     YA1      YA2
      YA3      YA4      YA5      YA6      YA7      YA8      YA9      YA10
      YA11     YA12     YA13     YA14     YA15     YA16     YA17     YA18
      YA19     YA20     YB1      YB2      YB3      YB4      YB5      YB6
      YB7      YB8      YB9      YB10     YB11     YB12     YB13     YB14
      YB15     YB16     YB17     YB18     YB19     YB20     YC1      YC2
      YC3      YC4      YC5      YC6      YC7      YC8      YC9      YC10
      YC11     YC12     YC13     YC14     YC15     YC16     YC17     YC18
    MAXIMIZE
    -          2.00000TX11    -        2.00000TX12    -         1.80000TX21
    -          1.80000TX22
    +                0YA1     +                0YA2   +               0YA3
    +                0YA4     +                0YA5   +               0YA6
    +                0YA7     +                0YA8   +               0YA9
    +                0YA10    +                0YA11  +               0YA12
    +                0YA13    +                0YA14  +               0YA15
    +                0YA16    +                0YA17  +               0YA18
    +                0YA19    +                0YA20  +               0YB1
    +                0YB2     +                0YB3   +               0YB4
    +                0YB5     +                0YB6   +               0YB7
    +                0YB8     +                0YB9   +               0YB10
    +                0YB11    +                0YB12  +               0YB13
    +                0YB14    +                0YB15  +               0YB16
    +                0YB17    +                0YB18  +               0YB19
    +                0YB20    +                0YC1   +      435.00000YC2
    +        840.00000YC3     +     1215.00000YC4     +     1560.00000YC5
    +       1875.00000YC6     +     2160.00000YC7     +     2415.00000YC8
    +       2640.00000YC9     +     2835.00000YC10    +     3000.00000YC11
    +       3135.00000YC12    +     3240.00000YC13    +     3315.00000YC14
    +       3360.00000YC15    +     3375.00000YC16    +     3360.00000YC17
    +       3315.00000YC18
    CONSTRAINT
 1. +          1.00000TX21    +        1.00000TX22
     .LE.      +        25.00000
 2. +          1.00000TX11    +        1.00000TX12
     .LE.      +        75.00000
 3. -          1.00000TX11
    +          1.77300YA1     +        4.21700YA2     +         7.00000YA3
    +         10.03000YA4     +       13.25600YA5     +         2.73400YA6
    +          6.50300YA7     +       10.79600YA8     +        15.46800YA9
    +         20.44400YA10    +        3.87900YA11    +         9.22700YA12
```

```
     MPOS VERSION 4.0        NORTHWESTERN UNIVERSITY

********************
* PROBLEM NUMBER  1 *
********************

USING REGULAR
*** NO. 2 * TEST TWO PROD. PNTS. AND ONE DEMAND ***

          +          15.31600YA13    +        21.94500YA14    +        29.00500YA15
          +           4.53900YA16    +        10.79600YA17    +        17.92100YA18
          +          25.67700YA19    +        33.93800YA20
        .LE.         +                0
    4.    -           1.00000TX21
          +           4.43300YA1     +        10.54200YA2     +        17.50100YA3
          +          25.07400YA4     +        33.14100YA5     +         3.41800YA6
          +           8.12900YA7     +        13.49500YA8     +        19.33500YA9
          +          25.55500YA10    +         2.77100YA11    +         6.59000YA12
          +          10.94000YA13    +        15.67500YA14    +        20.71800YA15
          +           2.52200YA16    +         5.99600YA17    +         9.95600YA18
          +          14.26500YA19    +        18.85400YA20
        .LE.         +                0
    5.    -           1.00000TY1
          +          10.00000YA1     +        20.00000YA2     +        30.00000YA3
          +          40.00000YA4     +        50.00000YA5     +        10.00000YA6
          +          20.00000YA7     +        30.00000YA8     +        40.00000YA9
          +          50.00000YA10    +        10.00000YA11    +        20.00000YA12
          +          30.00000YA13    +        40.00000YA14    +        50.00000YA15
          +          10.00000YA16    +        20.00000YA17    +        30.00000YA18
          +          40.00000YA19    +        50.00000YA20
        .GE.         +                0
    6.    +           1.00000YA1     +         1.00000YA2     +         1.00000YA3
          +           1.00000YA4     +         1.00000YA5     +         1.00000YA6
          +           1.00000YA7     +         1.00000YA8     +         1.00000YA9
          +           1.00000YA10    +         1.00000YA11    +         1.00000YA12
          +           1.00000YA13    +         1.00000YA14    +         1.00000YA15
          +           1.00000YA16    +         1.00000YA17    +         1.00000YA18
          +           1.00000YA19    +         1.00000YA20
        .LE.         +          1.00000
    7.    -           1.00000TX12
          +           4.14500YB1     +        10.44600YB2     +        17.93700YB3
          +          26.32500YB4     +        35.44400YB5     +         4.76200YB6
          +          11.99900YB7     +        20.60400YB8     +        30.23700YB9
          +          40.71500YB10    +         5.32600YB11    +        13.42100YB12
          +          23.04400YB13    +        33.81800YB14    +        45.53600YB15
          +           5.60000YB16    +        14.11200YB17    +        24.23200YB18
          +          35.58100YB19    +        47.88400YB20
        .LE.         +                0
    8.    -           1.00000TX22
          +          10.36400YB1     +        26.11500YB2     +        44.84200YB3
          +          65.80700YB4     +        88.61000YB5     +         5.95300YB6
          +          14.99900YB7     +        25.75500YB8     +        37.79600YB9
          +          50.89500YB10    +         3.80400YB11    +         9.56600YB12
          +          16.46000YB13    +        24.15500YB14    +        32.52600YB15
          +           3.11100YB16    +         7.84000YB17    +        13.46200YB18
          +          19.75600YB19    +        26.60200YB20
        .LE.         +                0
    9.    -           1.00000TY2
          +          10.00000YB1     +        20.00000YB2     +        30.00000YB3
          +          40.00000YB4     +        50.00000YB5     +        10.00000YB6
```

```
     MPOS VERSION 4.0        NORTHWESTERN UNIVERSITY

*********************
* PROBLEM NUMBER  1 *
*********************

USING REGULAR
*** NO. 2 # TEST TWO PROD. PNTS. AND ONE DEMAND ***

          +        20.00000YB7     +        30.00000YB8     +       40.00000YB9
          +        50.00000YB10    +        10.00000YB11    +       20.00000YB12
          +        30.00000YB13    +        40.00000YB14    +       50.00000YB15
          +        10.00000YB16    +        20.00000YB17    +       30.00000YB18
          +        40.00000YB19    +        50.00000YB20
       .GE.     +              0
   10.    +         1.00000YB1     +         1.00000YB2     +        1.00000YB3
          +         1.00000YB4     +         1.00000YB5     +        1.00000YB6
          +         1.00000YB7     +         1.00000YB8     +        1.00000YB9
          +         1.00000YB10    +         1.00000YB11    +        1.00000YB12
          +         1.00000YB13    +         1.00000YB14    +        1.00000YB15
          +         1.00000YB16    +         1.00000YB17    +        1.00000YB18
          +         1.00000YB19    +         1.00000YB20
       .LE.     +        1.00000
   11.    +         1.00000TY2     +         1.00000TY1
          -                0YC1    -         5.00000YC2     -       10.00000YC3
          -        15.00000YC4     -        20.00000YC5     -       25.00000YC6
          -        30.00000YC7     -        35.00000YC8     -       40.00000YC9
          -        45.00000YC10    -        50.00000YC11    -       55.00000YC12
          -        60.00000YC13    -        65.00000YC14    -       70.00000YC15
          -        75.00000YC16    -        80.00000YC17    -       85.00000YC18
       .GE.     +              0
   12.    +         1.00000YC1     +         1.00000YC2     +        1.00000YC3
          +         1.00000YC4     +         1.00000YC5     +        1.00000YC6
          +         1.00000YC7     +         1.00000YC8     +        1.00000YC9
          +         1.00000YC10    +         1.00000YC11    +        1.00000YC12
          +         1.00000YC13    +         1.00000YC14    +        1.00000YC15
          +         1.00000YC16    +         1.00000YC17    +        1.00000YC18
       .LE.     +        1.00000
       OPTIMIZE
```

MPOS VERSION 4.0   NORTHWESTERN UNIVERSITY

```
*********************
* PROBLEM NUMBER  1 *
*********************
```

USING REGULAR

*** NO. 2 A TEST TWO PROD. FNTS. AND ONE DEMAND ***

SUMMARY OF RESULTS

VAR NO	VAR NAME	ROW NO	STATUS	ACTIVITY LEVEL	OPPORTUNITY COST	LOWER BOUND	UPPER BOUND
1	TT2	--	B	16.4178473	0.0000000	0.0000	INF
2	TT1	--	B	50.0000000	0.0000000	0.0000	INF
3	TX11	--	B	33.9380000	0.0000000	0.0000	INF
4	TX12	--	B	11.0628710	0.0000000	0.0000	INF
5	TX21	--	B	16.8540000	0.0000000	0.0000	INF
6	TX22	--	B	6.1460000	0.0000000	0.0000	INF
7	YA1	--	LB	0.0000000	73.1310074	0.0000	INF
8	YA2	--	LB	0.0000000	62.2906048	0.0000	INF
9	YA3	--	LB	0.0000000	105.2450556	0.0000	INF
10	YA4	--	LB	0.0000000	138.1685041	0.0000	INF
11	YA5	--	LB	0.0000000	179.1071592	0.0000	INF
12	YA6	--	LB	0.0000000	59.3899412	0.0000	INF
13	YA7	--	LB	0.0000000	49.6261726	0.0000	INF
14	YA8	--	LB	0.0000000	51.0180968	0.0000	INF
15	YA9	--	LB	0.0000000	60.4825959	0.0000	INF
16	YA10	--	LB	0.0000000	76.4191000	0.0000	INF
17	YA11	--	LB	0.0000000	51.6957010	0.0000	INF
18	YA12	--	LB	0.0000000	31.3249520	0.0000	INF
19	YA13	--	LB	0.0000000	20.6303785	0.0000	INF
20	YA14	--	LB	0.0000000	16.9559681	0.0000	INF
21	YA15	--	LB	0.0000000	18.8984881	0.0000	INF
22	YA16	--	LB	0.0000000	49.1732345	0.0000	INF
23	YA17	--	LB	0.0000000	25.3274494	0.0000	INF
24	YA18	--	LB	0.0000000	10.6556917	0.0000	INF
25	YA19	--	LB	0.0000000	2.6624229	0.0000	INF
26	YA20	--	B	1.0000000	0.0000000	0.0000	INF
27	YB1	--	LB	0.0000000	109.0173390	0.0000	INF
28	YB2	--	LB	0.0000000	274.6803493	0.0000	INF
29	YB3	--	LB	0.0000000	488.6497764	0.0000	INF
30	YB4	--	LB	0.0000000	738.9451110	0.0000	INF
31	YB5	--	LB	0.0000000	1019.0737103	0.0000	INF
32	YB6	--	LB	0.0000000	42.1805853	0.0000	INF
33	YB7	--	LB	0.0000000	108.2487693	0.0000	INF
34	YB8	--	LB	0.0000000	199.4409761	0.0000	INF
35	YB9	--	LB	0.0000000	314.5167790	0.0000	INF
36	YB10	--	LB	0.0000000	447.5823434	0.0000	INF
37	YB11	--	LB	0.0000000	10.1460935	0.0000	INF
38	YB12	--	LB	0.0000000	25.5615602	0.0000	INF
39	YB13	--	LB	0.0000000	60.8843257	0.0000	INF
40	YB14	--	LB	0.0000000	111.1784284	0.0000	INF
41	YB15	--	LB	0.0000000	173.7922876	0.0000	INF
42	YB16	--	B	.3582153	0.0000000	0.0000	INF
43	YB17	--	B	.6417847	0.0000000	0.0000	INF
44	YB18	--	LB	0.0000000	16.9964119	0.0000	INF
45	YB19	--	LB	0.0000000	46.7808539	0.0000	INF

MPOS VERSION 4.0   NORTHWESTERN UNIVERSITY

```
************************
* PROBLEM NUMBER   1 *
************************
```

USING REGULAR
*** NO. 2 , TEST TWO PROD. PNTS. AND ONE DEMAND ***

SUMMARY OF RESULTS

VAR NO	VAR NAME	ROW NO	STATUS	ACTIVITY LEVEL	OPPORTUNITY COST	LOWER BOUND	UPPER BOUND
46	YB20	--	LB	0.0000000	67.0715348	0.0000	INF
47	YC1	--	LB	0.0000000	2730.0000000	0.0000	INF
48	YC2	--	LB	0.0000000	2340.0000000	0.0000	INF
49	YC3	--	LB	0.0000000	1960.0000000	0.0000	INF
50	YC4	--	LB	0.0000000	1650.0000000	0.0000	INF
51	YC5	--	LB	0.0000000	1350.0000000	0.0000	INF
52	YC6	--	LB	0.0000000	1060.0000000	0.0000	INF
53	YC7	--	LB	0.0000000	840.0000000	0.0000	INF
54	YC8	--	LB	0.0000000	630.0000000	0.0000	INF
55	YC9	--	LB	0.0000000	450.0000000	0.0000	INF
56	YC10	--	LB	0.0000000	300.0000000	0.0000	INF
57	YC11	--	LB	0.0000000	180.0000000	0.0000	INF
58	YC12	--	LB	0.0000000	90.0000000	0.0000	INF
59	YC13	--	LB	0.0000000	30.0000000	0.0000	INF
60	YC14	--	B	.7164305	0.0000000	0.0000	INF
61	YC15	--	B	.2835695	0.0000000	0.0000	INF
62	YC16	--	LB	0.0000000	30.0000000	0.0000	INF
63	YC17	--	LB	0.0000000	90.0000000	0.0000	INF
64	YC18	--	LB	0.0000000	160.0000000	0.0000	INF
65	SLACK-- D-	1	LB	0.0000000	13.6315923	0.0000	INF
66	SLACK-- D-	2	B	29.9991284	0.0000000	0.0000	INF
67	SLACK-- D-	3	LB	0.0000000	2.0000000	0.0000	INF
68	SLACK-- D-	4	LB	0.0000000	15.4315923	0.0000	INF
69	SLACK--	5	LB	0.0000000	9.0000000	0.0000	INF
81	ARTIF-- D-	5	LB	0.0000000	-9.0000000	0.0000	INF
70	SLACK-- D-	6	LB	0.0000000	91.1767587	0.0000	INF
71	SLACK-- D-	7	LB	0.0000000	2.0000000	0.0000	INF
72	SLACK-- D-	8	LB	0.0000000	15.4315923	0.0000	INF
73	SLACK--	9	LB	0.0000000	9.0000000	0.0000	INF
85	ARTIF-- D-	9	LB	0.0000000	-9.0000000	0.0000	INF
74	SLACK-- D-	10	LB	0.0000000	30.7923163	0.0000	INF
75	SLACK--	11	LB	0.0000000	9.0000000	0.0000	INF
87	ARTIF-- D-	11	LB	0.0000000	-9.0000000	0.0000	INF
76	SLACK-- D-	12	LB	0.0000000	2730.0000000	0.0000	INF

MAXIMUM VALUE OF THE OBJECTIVE FUNCTION =   3192.756883

CALCULATION TIME WAS   .2200 SECONDS FOR 30 ITERATIONS.

DATA STORAGE MEMORY =002465(OCTAL) TOTAL MEMORY = 050000(OCTAL)
TOTAL TIME FOR THIS PROBLEM WAS   .991 SECONDS

Problem Two Dual

For problem 2, the dual values are taken from the computer output and listed below;

$\lambda 6$ - dual row 6 - producer surplus Y_1 -	91.18
$\lambda 10$ - dual row 10 - producer surplus Y_2 -	30.79
$\lambda 11$ - dual row 11 - price of Y -	9.00
$\lambda 12$ - dual row 12 - consumer surplus Y -	2730.00

VI. HEADER CARDS

Three groups of JCL header cards are given. The first is to generate maximum output for troubleshooting. The second is to utilize a previously compiled program and to generate only output pertinent to problem solving. The third is to address APEX-1.

1. Maximum output, No LP output

```
NAME,T10.
ACCOUNT,GQM1111,PASSWORD.
BIN CARD IF NECESSARY
RFL(77000)
ACQUIRE(MGAO).
FETCH(MINNLIB/V=MNF)
MNF(I=MGAO,B=CMG)
RETAIN,CMG/CT=PU.
CBR(INPUT,TAPE5)
R,TAPE5.
SETTL(20)
CMG.
COST.
```

$^{7}8_{9}$

Data Deck

$^{7}8_{9}$

$^{6}{}^{7}8_{9}$

$^{6}{}^{7}8_{9}$

2. Reduced output using compiled deck, CMG, with LP output

```
NAME,T10.
ACCOUNT,GQM1111,PASSWORD.
RFL(77000)
FETCH(MINNLIB/V-MNF)
ACQUIRE,CMG.
CBR(INPUT,TAPE5)
R,TAPE5.
SETTL(20)
CMG.
R,TAPE1.
COPYSBF,TAPE1,OUTPUT.
R,TAPE1.
MPOS(TAPE1)
COST.
789
Date Deck
789
6789
6789
```

3. APEX-1

In order to access APEX, the following cards are inserted between the MPOS(TAPE1) and COST cards in deck 2.

```
R,APXFIL.
COPYSBF(APXFIL,OUTPUT)
RETURN,TAPE1.
R,APXFIL.
RENAME,TAPE1=APXFIL.
APEX(SOLV--------)
```

VII. FINAL COMMENTS

This computer program has not been fully tested and problems may be found. However, initial testing and use indicate that the program could be extremely useful in solving certain types of problems. The fortran program itself is rather straight forward and appears to be organized in a way that would facilitate user provided modification.

Some comments regarding the use of MGAO with APEX are in order. Currently, only APEX-I one is operable. APEX-II may be accessed, however format errors are incurred. If the user accesses APEX-I directly, parametric and/or ranging procedures are difficult if not impossible to perform. An alternative is to use MGAO to punch out the data deck and then put together a completely new problem specification for the APEX problem. This may be done with cards or through the use of permanent files and interactive terminals.

The user should not be limited by the mathematical forms of the functions as currently read. One need only to change the format and the read statements to fit different needs and problems.

Bibliography

[1] Program developed by Terry L. Roe and Paul Chang

[2] Multi Purpose Optimization System User's Guide Version 3, Claude Cohen and Jack Stein, Manual No. 320, Copyright 1976, Voeglback Computing Center, Northwestern University, Evanston, Illinois 60201.

APEX-1 Reference Manual

[3] Intrilligator Michael D., Mathematical Optimization and Economics Theory, 1971, Prentice-Hall, Inc., p. 72.

[4] Intrilligator, p.44.

[5] Duloy, John H. and Roger D. Norton, "Prices and Incomes in Linear Programming Models," American Journal of Agricultural Economics, Volume 57, Number 4 (November 1975) pp. 591-600.

Klein, Harold E. and Terry L. Roe, "Agriculture Sector Analysis Model Design: The Influence of Administrative Infrastructure Characteristics," in Planning Processes in Developing Countries: Techniques and Achievements, eds. W.D. Cook and T.E. Kuhn (Amsterdam-London: North-Holland, 1982) pp. 273-308.

[6] Klein and Roe, pp. 297-299.

[7] Duloy, John H. and Roger P. Norton, "The CHAC Demand Structures, Chapter 3 in Programming Studies for Mexican Agricultural Policy, eds. Roger D. Norton and Leopoldo M. Solis, forthcoming.

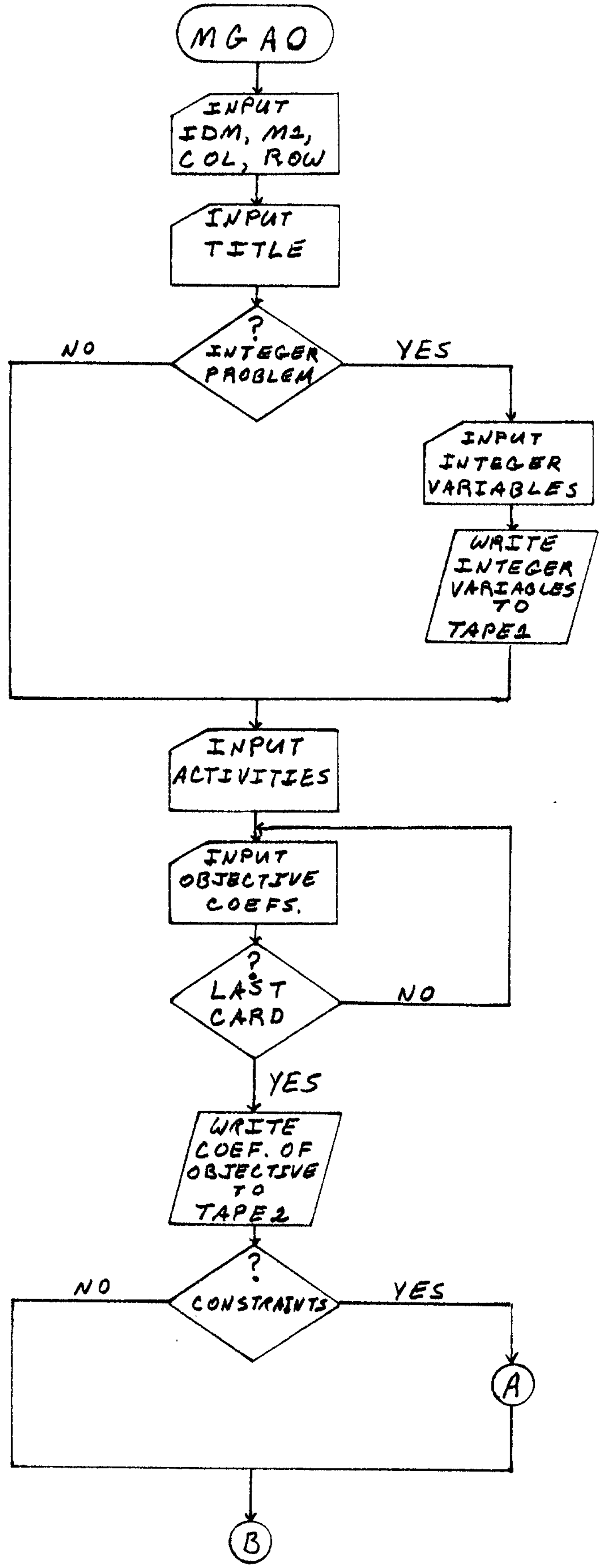
[8] Roe, Terry, "Modelling of Nonlinear Functions into a Linear Programming Format," Staff Paper P75-9, June 1975, Dept. of Ag. and Applied Econ., U of M, St. Paul.

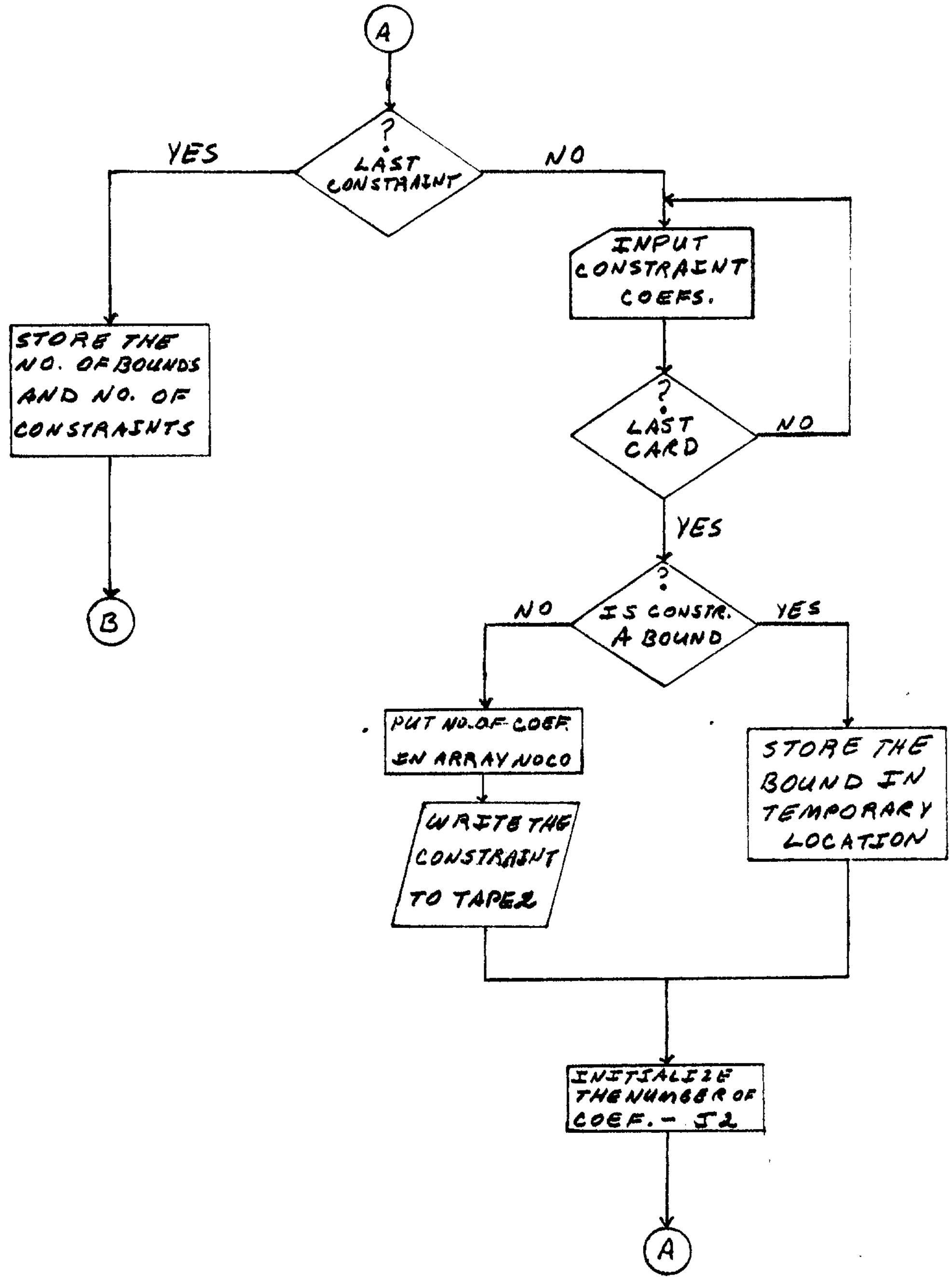
Appendix A. Variable Name

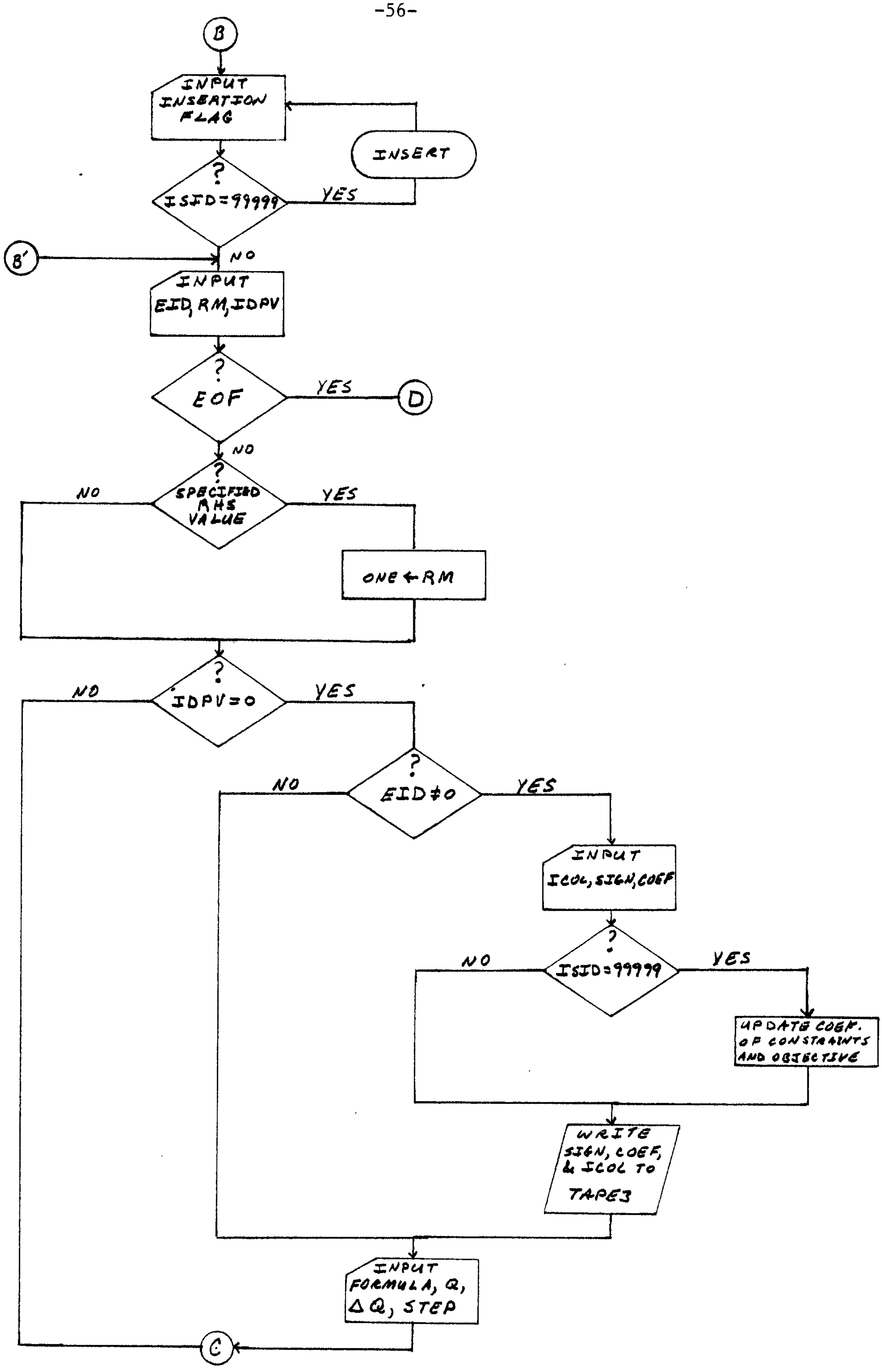
```
******************************************************************************
** METHD(M1) CONTAINS THE 12 ALGORITHMS WHICH ARE USED IN M.P.O.S.           *
**     M1 = 01        ≠REGULAR≠, 2-PHASE SIMPLEX       (L.P.)                *
**     M1 = 02        ≠REVISED≠, REVISED SIMPLEX       (L.P.)                *
**     M1 = 03        ≠DUAL≠, DUAL SIMPLEX             (L.P.)                *
**     M1 = 04        ≠MINIT≠, PRIMAL-DUAL ALG.        (L.P.)                *
**     M1 = 05        ≠BBMIP≠, BRANCH AND BOUND MIXED INT. PROGRAM.(I.P.)    *
**     M1 = 06        ≠DSZ1IP≠, DIRECT SEARCH 0-1 INTEGER PROGRAM. (I.P.)    *
**     M1 = 07        ≠GOMORY≠, GOMORY≠S CUTTING PLANE              (I.P.)   *
**     M1 = 08        ≠WOLFE≠, WOLFE≠S QUARDRATIC SIMPLEX (Q.P.)             *
**     M1 = 09        ≠BEALE≠, BEALE≠S ALGORITHM            (Q.P.)           *
**     M1 = 10        ≠LEMKE≠, LEMKE≠S COMPLEMENTARY PIVOT ALG.(Q.P.)        *
**     M1 = 11        ≠APEX1≠, MPOS-APEX DATA FILE INTERFACE  (GENERAL)      *
**     M1 = 12        ≠APEX2≠, MPOS-APEX DATA FILE INTERFACE  (GENERAL)      *
**                                                                           *
** VARS(N1) ARE THE RESERVED WORDS USED BY ≠MPOS≠.                           *
**     N1 = 1         TITLE                                                  *
**     N1 = 2         INTEGER                                                *
**     N1 = 3         VARIABLES                                              *
**     N1 = 4         MAXIMIZE                                               *
**     N1 = 5         MINIMIZE                                               *
**     N1 = 6         CONSTRAINTS                                            *
**     N1 = 7         BOUNDS                                                 *
**     N1 = 8         PRINT                                                  *
**     N1 = 9         OPTIMIZE                                               *
**     N1 = 10        ENDAPEX                                                *
**     N1 = 11        BNDALL                                                 *
**     N1 = 12        BNDINT                                                 *
**     N1 = 13        RNGOBJ                                                 *
**     N1 = 14        RNGRHS                                                 *
**     N1 = 15        TOLERANCE                                              *
**     N1 = 16        EPSILON                                                *
**     N1 = 17        BNDOBJ                                                 *
**     N1 = 18        LIMIT                                                  *
**     N1 = 19        NOSCALE                                                *
**     N1 = 20        CHECK                                                  *
**     N1 = 21        QCHECK                                                 *
**     N1 = 22        GO                                                     *
**     N1 = 23        STOP                                                   *
**     N1 = 24        RESCALE                                                *
**     N1 = 25        MAXCM                                                  *
**     N1 = 26        MAXECS                                                 *
**                                                                           *
** GFL(M2)            CONTAIN ≠.LE.≠, ≠.EQ.≠, ≠.GE.≠                         *
**         =-100      MEANS LESS THAN OR EQUAL TO                            *
**         =-200      MEANS EQUAL TO                                         *
**         =-300      MEANS GREAT THAN OR EQUAL TO                           *
**                                                                           *
```

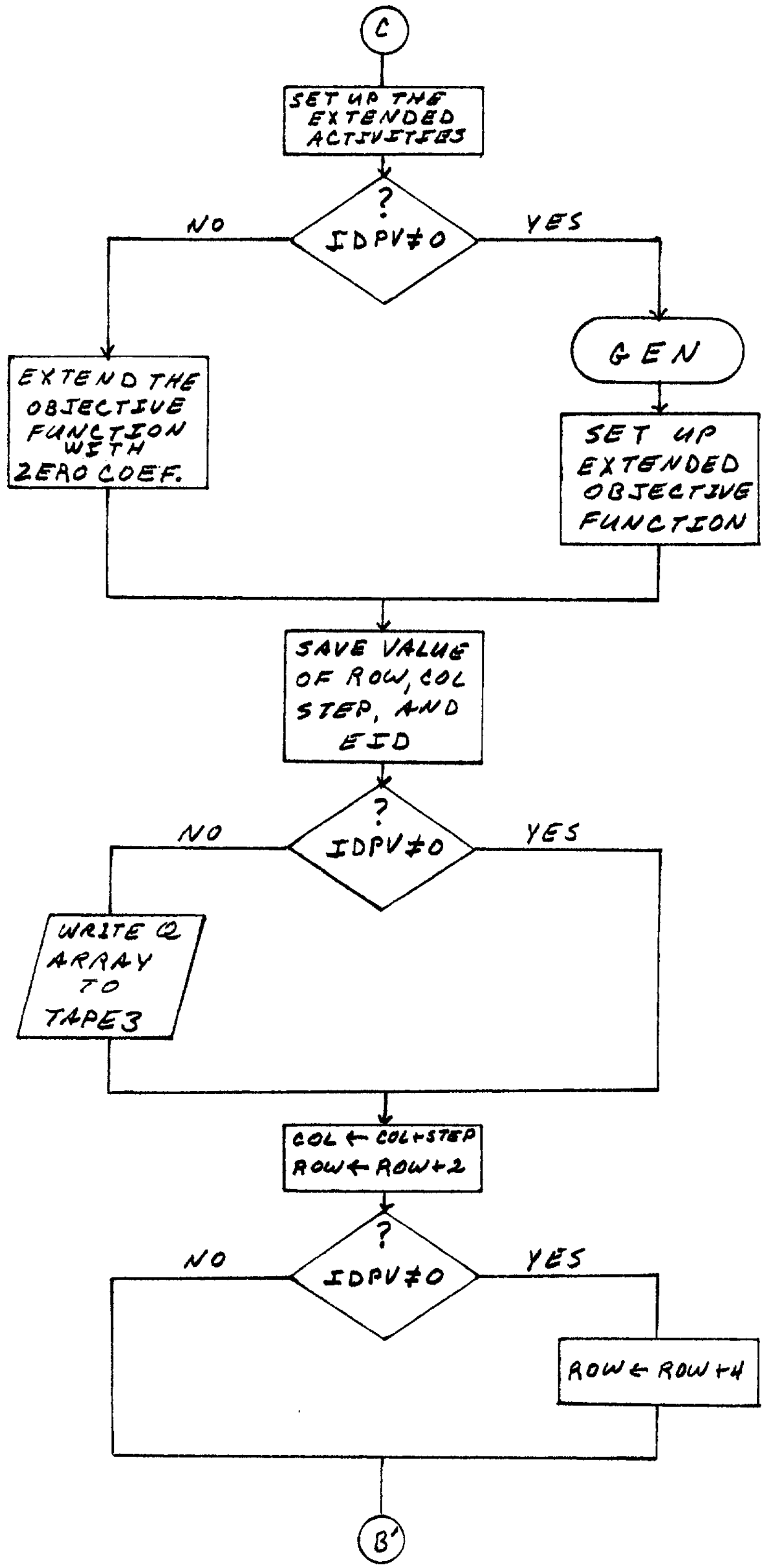
```
** CWRITE              AN ARRAY STORE THE COEF. OF MATRIX TO BE DUMPED       *
** EID                 EXTEDING MODEL FLAG                                   *
**         .EQ. 0      EXTENDING THE MODEL WITHOUT READING THE COEF.         *
**                     (ONLY ADD ONE BLANK CARD)                             *
**         .NE. 0      EXTENDING THE MODEL WITH    READING THE COEF.         *
**                     (PUNCH THE NO. OF COEF. WHICH WILL BE ADDED.)         *
** IDM                 MAX. OR MIN. IDENTIFIER                               *
**         = 0         MEANS MAXIMIZE                                        *
**         = 1         MEANS MINIMIZE                                        *
** IDPV                                                                      *
**         = 99999     2 INDEP. VAR. FUNC. EXTENSION                         *
**         = 0         OTHER CASE                                            *
** ISID                INSERTION FLAG                                        *
**         = 0         NO NEW ACTIVITIES ARE TO BE ADDED.                    *
**         = 99999     NEW ACTIVITIES ARE TO BE INSERTED.                    *
** IWRITE              DUMP MATRIX TABLE FLAG                                *
**         = 0         DON#T DUMP THE TABLE                                  *
**         = 1         DUMP THE TABLE                                        *
**         = 2         DUMP PART OF THE TABLE                                *
** ACT                 ACTIVITIES                                            *
** AACT                NEW ACTIVITIES                                        *
** CC                  COEF. OF THE FORMULA                                  *
** COEF                ABSOLUTE VALUE OF THE COEFFIENT.                      *
** COL                 COLUMN NO. OF THE MATRIX(NO. OF ENTRIES)              *
** DELTAQ              DELTA Q                                               *
** ICOL                COLUMN COORD. OF THE ACTIVITY                         *
** IEXP                EXPONENT OF EACH ITEM OF THE FORMULA                  *
** IJ                  NO. OF ITEMS OF THE FORMULA.                          *
** JA                  NO. OF EXTENDING PROCEDURES.                          *
** N2                  NO. OF INTEGER VARIABLES FOR I.P.                     *
** NAA                 NO. OF ACTIVITIES NEEDS TO BE INSERTED                *
** NBB                 TOTAL NO. OF NON-ZERO COEF.                           *
** NINS                THE PLACEMENT OF NEW ACTIVITIES WILL BE INSERTED      *
** NCOL                ARRAY STORE THE VELUES OF COL  OF EXTENDING           *
** NEID                ARRAY STORE THE VELUES OF EID  OF EXTENDING           *
** NOBD                NO. OF BOUNDS                                         *
** NOCD                NO. OF CONSTRAINTS                                    *
** NOCO                ARRAY STORE THE VELUES OF NOCO OF EXTENDING           *
** NROW                ARRAY STORE THE VELUES OF ROW  OF EXTENDING           *
** NSTEP               ARRAY STORE THE VELUES OF STEP OF EXTENDING           *
** ONE                 #1#, CONSTANT ONE.                                    *
** Q(1)                INITIAL VALUE OF Q                                    *
** ROW                 ROW NO. OF THE MATRIX                                 *
** RM                  THE SPECIFIED R.H.S VALUE DIFFERENT FROM DEFAULT ONE*
** SIGN                SIGN OF THE ACTIVITY                                  *
** STEP                STEPS                                                 *
** TACT                TEMP. LOC. STORES THE ACTIVITIES OF BOUNDS            *
** TCOEF               TEMP. LOC. STORES THE ABSOLUTE VALUES OF BOUNDS       *
** TITLE               TITLEOF THE PROBLEM (RESTRICTED ONE CARD )            *
** TSIGN               TEMP. LOC. STORES THE SIGN OF BOUNDS                  *
** TT1                 TEMP. LOC. STORES THE RELATIONS OF BOUNDS             *
** TT2                 TEMP. LOC. STORES THE SIGN OF R.H.S.                  *
** TT3                 TEMP. LOC. STORES THE ABSOLUTE VALUE OF R.H.S.        *
** W                   THE VALUES OF THE FORMULA WITH PUTTING Q VALUES       *
** WSIGN               THE SIGNS OF W ARRAY                                  *
** XSIGN               #-#, MINUS SIGN.                                      *
** YSIGN               #+#, PLUS  SIGN.                                      *
** ZERO                #0#, CONSTANT ZERO                                    *
******************************************************************************
```

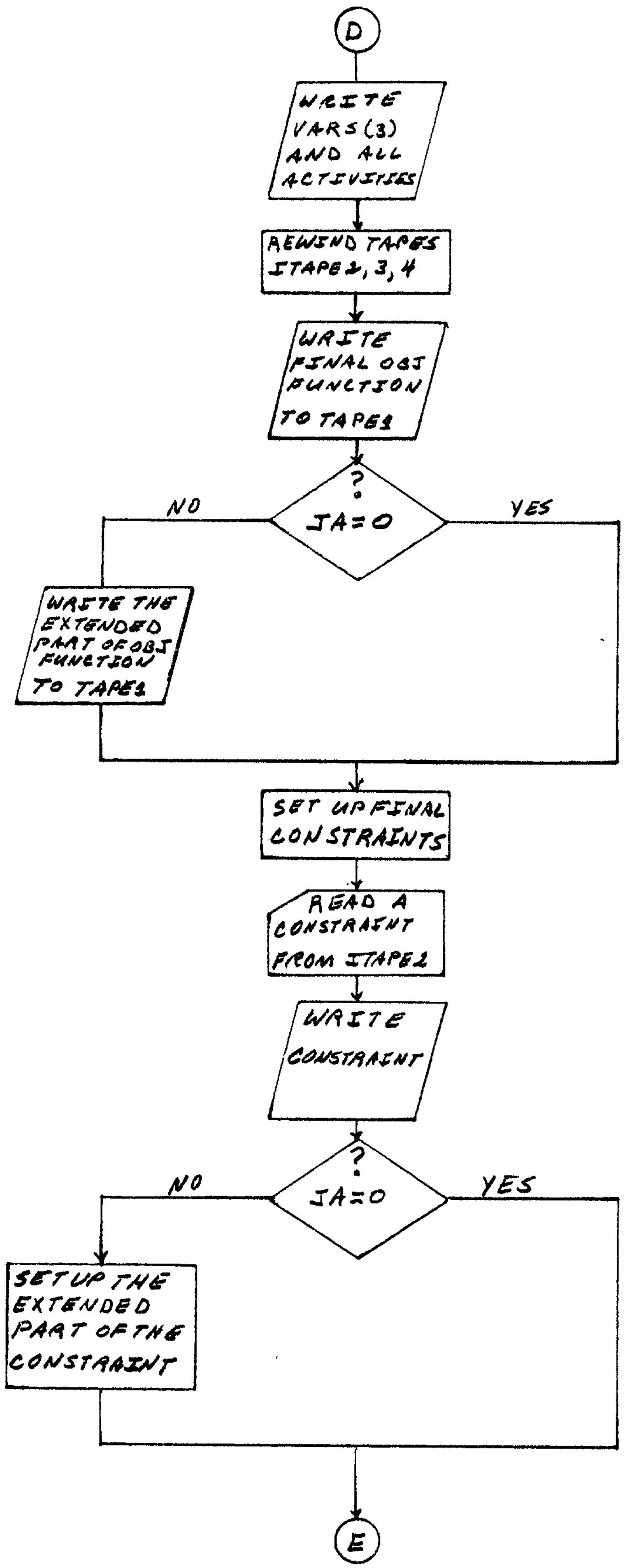
Appendix B. Flow Chart

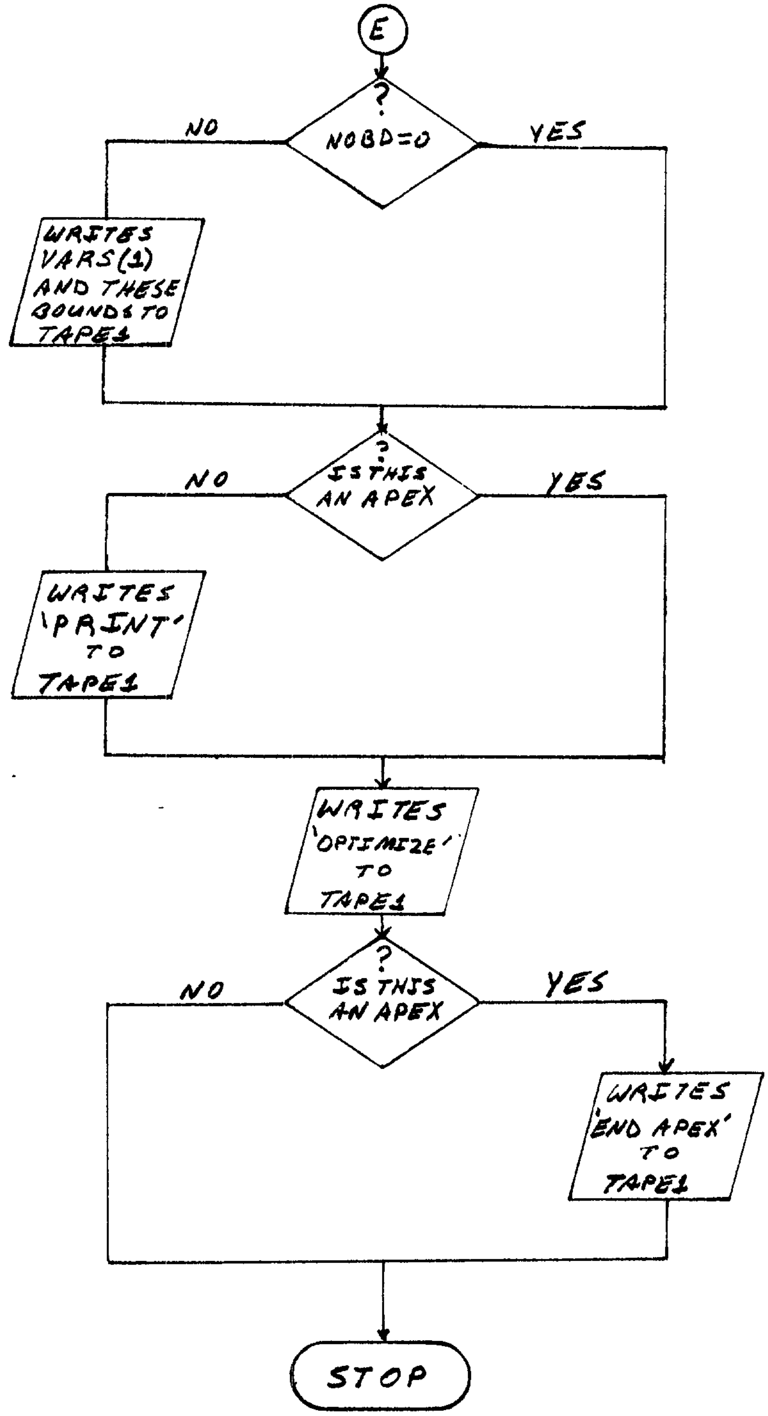


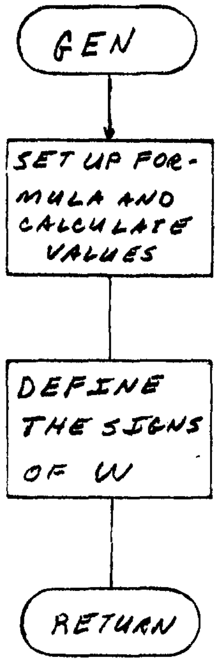

A
LAST CONSTRAINT
YES
NO
INPUT CONSTRAINT COEFS.
STORE THE NO. OF BOUNDS AND NO. OF CONSTRAINTS
LAST CARD
NO
YES
B
IS CONSTR. A BOUND
NO
YES
PUT NO. OF COEF. IN ARRAY NOCO
STORE THE BOUND IN TEMPORARY LOCATION
WRITE THE CONSTRAINT TO TAPE2
INITIALIZE THE NUMBER OF COEF. - I2
A

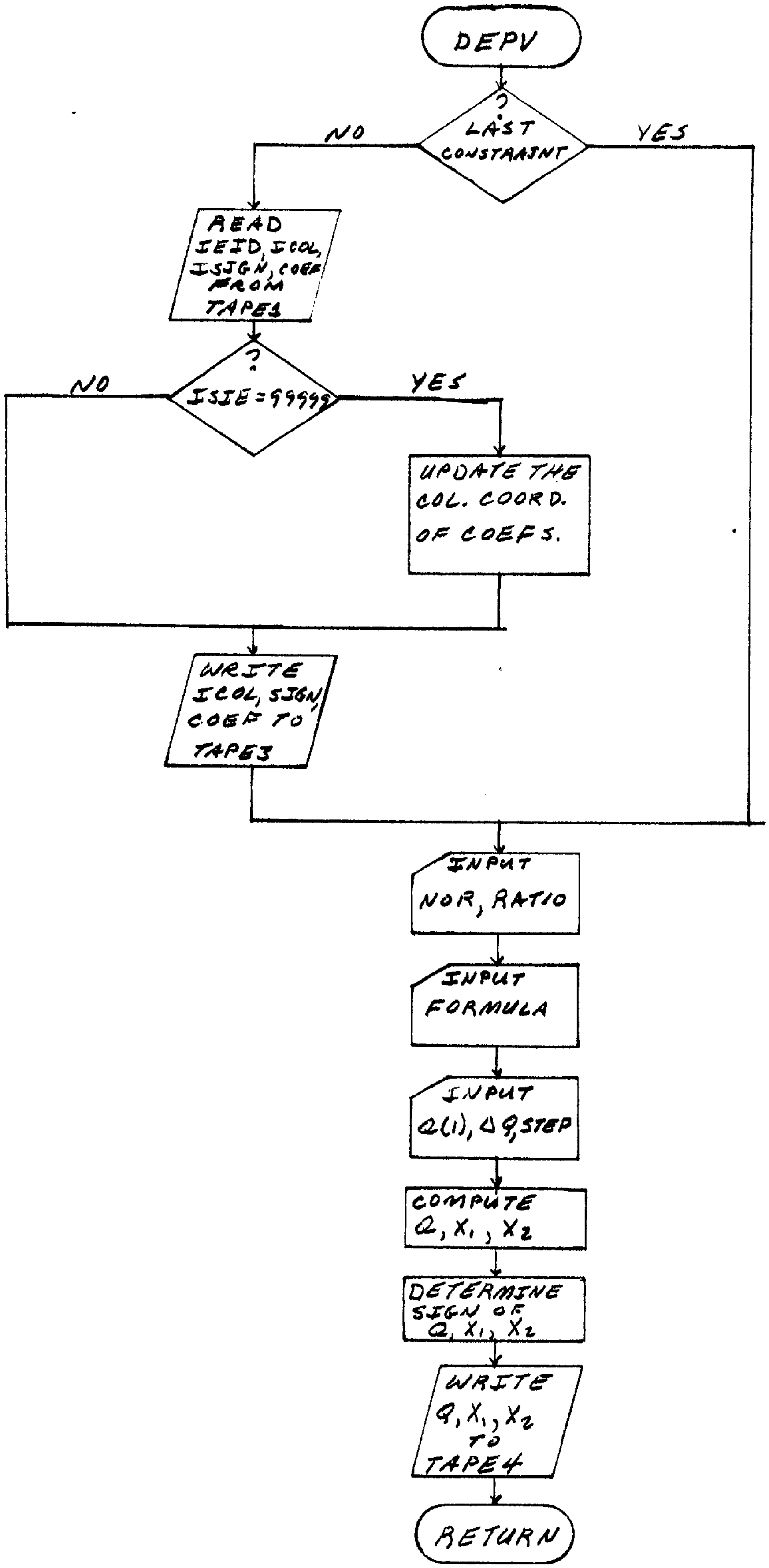

B
INPUT INSERTION FLAG
INSERT
?
ISID=99999
YES
NO
B'
INPUT EID, RM, IDPV
?
EOF
YES
D
NO
?
SPECIFIED RHS VALUE
NO
YES
ONE ← RM
?
IDPV=0
NO
YES
?
EID≠0
NO
YES
INPUT ICOL, SIGN, COEF
?
ISID=99999
NO
YES
UPDATE COEF. OF CONSTRAINTS AND OBJECTIVE
WRITE SIGN, COEF, & ICOL TO TAPE3
INPUT FORMULA, Q, ΔQ, STEP
C

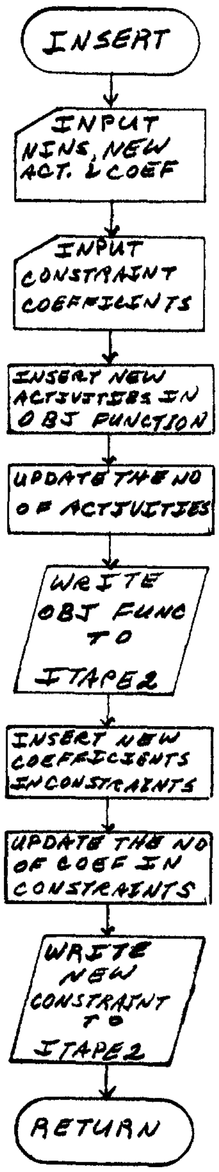

C
SET UP THE EXTENDED ACTIVITIES
?
IDPV≠0
NO
YES
EXTEND THE OBJECTIVE FUNCTION WITH ZERO COEF.
GEN
SET UP EXTENDED OBJECTIVE FUNCTION
SAVE VALUE OF ROW, COL STEP, AND EID
?
IDPV≠0
NO
YES
WRITE Q ARRAY TO TAPE3
COL ← COL+STEP
ROW ← ROW+2
?
IDPV≠0
NO
YES
ROW ← ROW+4
B'


D
WRITE VARS(3) AND ALL ACTIVITIES
REWIND TAPES ITAPE2, 3, 4
WRITE FINAL OBJ FUNCTION TO TAPE1
NO
? JA=0
YES
WRITE THE EXTENDED PART OF OBJ FUNCTION TO TAPE1
SET UP FINAL CONSTRAINTS
READ A CONSTRAINT FROM ITAPE2
WRITE CONSTRAINT
NO
? JA=0
YES
SET UP THE EXTENDED PART OF THE CONSTRAINT
E


E
NOBD=0
NO
YES
WRITES VARS(1) AND THESE BOUNDS TO TAPE1
IS THIS AN APEX
NO
YES
WRITES 'PRINT' TO TAPE1
WRITES 'OPTIMIZE' TO TAPE1
IS THIS AN APEX
NO
YES
WRITES 'END APEX' TO TAPE1
STOP


GEN
SET UP FORMULA AND CALCULATE VALUES
DEFINE THE SIGNS OF W
RETURN


DEPV
?
LAST CONSTRAINT
NO
YES
READ IEID, ICOL, ISIGN, COEF FROM TAPE1
?
ISIE = 99999
NO
YES
UPDATE THE COL. COORD. OF COEFS.
WRITE ICOL, SIGN, COEF TO TAPE3
INPUT NOR, RATIO
INPUT FORMULA
INPUT Q(1), ΔQ, STEP
COMPUTE Q, X1, X2
DETERMINE SIGN OF Q, X1, X2
WRITE Q, X1, X2 TO TAPE4
RETURN


INSERT
INPUT NINS, NEW ACT. & COEF
INPUT CONSTRAINT COEFFICIENTS
INSERT NEW ACTIVITIES IN OBJ FUNCTION
UPDATE THE NO OF ACTIVITIES
WRITE OBJ FUNC TO ITAPE2
INSERT NEW COEFFICIENTS IN CONSTRAINTS
UPDATE THE NO OF COEF IN CONSTRAINTS
WRITE NEW CONSTRAINT TO ITAPE2
RETURN

Appendix C. Program Listing

```
      MGRO      MATRIX GENERATOR AND OPTIONALS                              *
*****************************************************************************
                THE FUNCTION OF THIS ROUTINE IS TO                          *
                READ  A SET OF DATA CARDS IN SPECIFIED FORMATS.             *
                SET UP AN M.P.O.S. MODEL OR EXTED THE MODEL BY              *
                CALLING AN APPROPRIATE SUBROUTINE(GEN).                     *
                THROUGH M.P.O.S. , WE CAN GET THE SOLUTIONS.                *
*****************************************************************************
* METHOD(M1) CONTAINS THE 12 ALGORITHMS WHICH ARE USED IN M.P.O.S.          *
*     M1 = 01        ≠REGULAR≠, 2-PHASE SIMPLEX     (L.P.)                  *
*     M1 = 02        ≠REVISED≠, REVISED SIMPLEX     (L.P.)                  *
*     M1 = 03        ≠DUAL≠, DUAL SIMPLEX           (L.P.)                  *
*     M1 = 04        ≠MINIT≠, PRIMAL-DUAL ALG.      (L.P.)                  *
*     M1 = 05        ≠BBMIP≠, BRANCH AND BOUND MIXED INT. PROGRAM.(I.P.)    *
*     M1 = 06        ≠DSZ1IP≠, DIRECT SEARCH 0-1 INTEGER PROGRAM. (I.P.)    *
*     M1 = 07        ≠GOMORY≠, GOMORY≠S CUTTING PLANE            (I.P.)     *
*     M1 = 08        ≠WOLFE≠, WOLFE≠S QUARDRATIC SIMPLEX (Q.P.)             *
*     M1 = 09        ≠BEALE≠, BEALE≠S ALGORITHM          (Q.P.)             *
*     M1 = 10        ≠LEMKE≠, LEMKE≠S COMPLEMENTARY PIVOT ALG.(Q.P.)        *
*     M1 = 11        ≠APEX1≠, MPOS-APEX DATA FILE INTERFACE  (GENERAL)      *
*     M1 = 12        ≠APEX2≠, MPOS-APEX DATA FILE INTERFACE  (GENERAL)      *
*                                                                           *
* VARS(N1) ARE THE RESERVED WORDS USED BY ≠MPOS≠.                           *
*     N1 = 1         TITLE                                                  *
*     N1 = 2         INTEGER                                                *
*     N1 = 3         VARIABLES                                              *
*     N1 = 4         MAXIMIZE                                               *
*     N1 = 5         MINIMIZE                                               *
*     N1 = 6         CONSTRAINTS                                            *
*     N1 = 7         BOUNDS                                                 *
*     N1 = 8         PRINT                                                  *
*     N1 = 9         OPTIMIZE                                               *
*     N1 = 10        ENDAPEX                                                *
*     N1 = 11        BNDALL                                                 *
*     N1 = 12        BNDINT                                                 *
*     N1 = 13        RNGOBJ                                                 *
*     N1 = 14        RNGRHS                                                 *
*     N1 = 15        TOLERANCE                                              *
*     N1 = 16        EPSILON                                                *
*     N1 = 17        BNDOBJ                                                 *
*     N1 = 18        LIMIT                                                  *
*     N1 = 19        NOSCALE                                                *
*     N1 = 20        CHECK                                                  *
*     N1 = 21        QCHECK                                                 *
*     N1 = 22        GO                                                     *
*     N1 = 23        STOP                                                   *
*     N1 = 24        RESCALE                                                *
*     N1 = 25        MAXCM                                                  *
*     N1 = 26        MAXECS                                                 *
*                                                                           *
* GEL(N2)            CONTAIN ≠.LE.≠, ≠.EQ.≠, ≠.GE.≠                         *
*          =-1.0     MEANS LESS THAN OR EQUAL TO                            *
*          =-2.0     MEANS EQUAL TO                                         *
*          =-3.0     MEANS GREAT THAN OR EQUAL TO                           *
```

```
*
* CWRITE                    AN ARRAY STORE THE COEF. OF MATRIX TO BE DUMPED       *
* EID                       EXTENDING MODEL FLAG                                  *
*          .EQ   0          EXTENDING THE MODEL WITHOUT READING THE COEF.         *
*                           (ONLY ADD ONE BLANK CARD)                             *
*          .NE   0          EXTENDING THE MODEL WITH    READING THE COEF.         *
*                           (PUNCH THE NO. OF COEF. WHICH WILL BE ADDED.)         *
* IMM                       MAX. OR MIN. IDENTIFIER                               *
*          = 0              MEANS MAXIMIZE                                        *
*          = 1              MEANS MINIMIZE                                        *
* INDV                                                                            *
*          = 9.999          2 INDEP. VAR. FUNC. EXTENSION                         *
*          = 0              OTHER CASE                                            *
* ISID                      INSERTION FLAG                                        *
*          = 0              NO NEW ACTIVITIES ARE TO BE ADDED.                    *
*          = 9.999          NEW ACTIVITIES ARE TO BE INSERTED.                    *
* IWRITE                    DUMP MATRIX TABLE FLAG                                *
*          = 0              DON#T DUMP THE TABLE                                  *
*          = 1              DUMP THE TABLE                                        *
*          = 2              DUMP PART OF THE TABLE                                *
* ACT                       ACTIVITIES                                            *
* AACT                      NEW ACTIVITIES                                        *
* CC                        COEF. OF THE FORMULA                                  *
* COEF                      ABSOLUTE VALUE OF THE COEFFIENT.                      *
* COL                       COLUMN NO. OF THE MATRIX(NO. OF ENTRIES)              *
* DELTAQ                    DELTA Q                                               *
* ICOL                      COLUMN COORD. OF THE ACTIVITY                         *
* IEXP                      EXPONENT OF EACH ITEM OF THE FORMULA                  *
* IN                        NO. OF ITEMS OF THE FORMULA.                          *
* JA                        NO. OF EXTENDING PROCEDURES.                          *
* N2                        NO. OF INTEGER VARIABLES FOR I.P.                     *
* NAW                       NO. OF ACTIVITIES NEEDS TO BE INSERTED                *
* NRR                       TOTAL NO. OF NON-ZERO COEF.                           *
* NTMS                      THE PLACEMENT OF NEW ACTIVITIES WILL BE INSERTED      *
* NCOL                      ARRAY STORE THE VELUES OF COL   OF EXTENDING          *
* NEID                      ARRAY STORE THE VELUES OF EID   OF EXTENDING          *
* NOBD                      NO. OF BOUNDS                                         *
* NOCD                      NO. OF CONSTRAINTS                                    *
* NOCO                      ARRAY STORE THE VELUES OF NOCO OF EXTENDING           *
* NROW                      ARRAY STORE THE VELUES OF ROW  OF EXTENDING           *
* NSTEP                     ARRAY STORE THE VELUES OF STEP OF EXTENDING           *
* ONE                       #1#, CONSTANT ONE.                                    *
* Q(I)                      INITIAL VALUE OF Q                                    *
* ROW                       ROW NO. OF THE MATRIX                                 *
* RM                        THE SPECIFIED R.H.S VALUE DIFFERENT FROM DEFAULT ONE*
* SIGN                      SIGN OF THE ACTIVITY                                  *
* STEP                      STEPS                                                 *
* TACT                      TEMP. LOC. STORES THE ACTIVITIES OF BOUNDS            *
* TCOEF                     TEMP. LOC. STORES THE ABSOLUTE VALUES OF BOUNDS       *
* TITLE                     TITLEOF THE PROBLEM (RESTRICTED ONE CARD )            *
* TSIGN                     TEMP. LOC. STORES THE SIGN OF BOUNDS                  *
* TT1                       TEMP. LOC. STORES THE RELATIONS OF BOUNDS             *
* TT2                       TEMP. LOC. STORES THE SIGN OF R.H.S.                  *
* TT3                       TEMP. LOC. STORES THE ABSOLUTE VALUE OF R.H.S.        *
* W                         THE VALUES OF THE FORMULA WITH PUTTING Q VALUES       *
* WSIGN                     THE SIGNS OF W ARRAY                                  *
* XSIGN                     #-#, MINUS SIGN.                                      *
* YSIGN                     #+#, PLUS SIGN.                                       *
* ZERO                      #0#, CONSTANT ZERO                                    *
***********************************************************************************
```

```
********************************************************************************
      PROGRAM  MSAO(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE5,TAPE6=OUT
     *PUT,TAPE7,TAPE8)
      REAL METHD,TEXP,L
      INTEGER     AICOL,AEWCOL,AEWROW
      INTEGER COL,ROW ,FID,STEP
      COMMON /T111/Q(1000),W(1000),CC(10),TEXP(10) ,WSIGN(1000)
      COMMON /T112/DELTAQ,COL,ROW,FID,STEP,IJ,IDM,M1,N2,NOBD ,NOOB
      COMMON  /T113/TITLE(8),GEL(3),ACT(1200),ICOL(1200),SIGN(1200),
     *        COFF(1200),METHD(15),VARS(26)
      COMMON  /T114/TACT(100),TT2(100),TT3(100)
      COMMON  /T115/NSTEP(50),NOCO(500),NROW(50),NCOL(50),NFID(50)
      COMMON  /T116/Y(20,100)
      COMMON   /T118/RATIO(100),C(1000),L(1000),IDPV(50)
      COMMON   /T119/IEID(100),CSIGN(1000),LSIGN(1000),QSIGN(1000)
      COMMON   /T120/XSIGN,YSIGN,ZERO,ONE,JA,JB,JC,NRR,ITAPE1,ITAPE2
      COMMON  /T121/ISID,NAA,NBB,NINS,AACT(20),AICOL(20),ASIGN(20),ISIF,
     *        ACOFF(20),AEWROW(50),AEWSIGN(50),AEWCOFF(50),AEWCOL(50)


      DATA  ZERO,FID,ISIF/0.0,-1.0/
      DATA XSIGN,YSIGN,ONE,JA,JB,JC/≠-≠,≠+≠,1.0,0,0,1/
      DATA METHD/≠REGULAR   ≠,≠REVISED   ≠,≠DUAL       ≠,≠MINIT      ≠,
     *≠DBMIP      ≠,≠DSZ1IP     ≠,≠GOMORY     ≠,≠WOLFF      ≠,≠BEALE      ≠,
     *≠LEMKE      ≠,≠APEX1      ≠,≠APEX2      ≠/
      DATA GEL/≠ .LE. ≠,≠ .EQ. ≠,≠ .GE. ≠/
      DATA VARS/≠TITLE      ≠,≠INTEGER    ≠,≠VARIABLES ≠,≠MAXIMIZE   ≠,
     *≠MINIMIZE   ≠,≠CONSTRAINT≠,≠BOUNDS     ≠,≠PRINT      ≠,≠OPTIMIZE   ≠,
     *≠ENDAP-X    ≠,≠BNDUAL     ≠,≠BNDINT     ≠,≠RNGOBJ     ≠,≠RNGRHS     ≠,
     *≠TOLERANCE  ≠,≠EPSILON    ≠,≠BNDOBJ     ≠,≠LIMIT      ≠,≠NOSCALE    ≠,
     *≠CHECK      ≠,≠NOCHECK    ≠,≠GO         ≠,≠STOP       ≠,≠RESCALE    ≠,
     *≠MAXCMP     ≠,≠MAXEYS     ≠/
      DATA  IDPV,NOBD/≠ ≠,0/
      DATA  TACT,TT2,TT3/200*≠ ≠,100*0.0/


      DATA  (Y(1,MA),MA=1,100)/≠YA1    ≠,≠YA2    ≠,≠YA3    ≠,≠YA4    ≠,
     *≠YA5     ≠,
     *      ≠YA6    ≠,≠YA7    ≠,≠YA8    ≠,≠YA9    ≠,≠YA10   ≠,≠YA11   ≠,
     *      ≠YA12   ≠,≠YA13   ≠,≠YA14   ≠,≠YA15   ≠,≠YA16   ≠,≠YA17   ≠,
     *      ≠YA18   ≠,≠YA19   ≠,≠YA20   ≠,≠YA21   ≠,≠YA22   ≠,
     *      ≠YA23   ≠,≠YA24   ≠,≠YA25   ≠,≠YA26   ≠,≠YA27   ≠,≠YA28   ≠,
     *      ≠YA29   ≠,≠YA30   ≠,≠YA31   ≠,≠YA32   ≠,≠YA33   ≠,≠YA34   ≠,
     *      ≠YA35   ≠,≠YA36   ≠,≠YA37   ≠,≠YA38   ≠,≠YA39   ≠,≠YA40   ≠,
     *      ≠YA41   ≠,≠YA42   ≠,≠YA43   ≠,≠YA44   ≠,≠YA45   ≠,≠YA46   ≠,
     *      ≠YA47   ≠,≠YA48   ≠,≠YA49   ≠,≠YA50   ≠,≠YA51   ≠,≠YA52   ≠,
     *      ≠YA53   ≠,≠YA54   ≠,≠YA55   ≠,≠YA56   ≠,≠YA57   ≠,≠YA58   ≠,
     *      ≠YA59   ≠,≠YA60   ≠,≠YA61   ≠,≠YA62   ≠,≠YA63   ≠,≠YA64   ≠,
     *      ≠YA65   ≠,≠YA66   ≠,≠YA67   ≠,≠YA68   ≠,≠YA69   ≠,≠YA70   ≠,
     *      ≠YA71   ≠,≠YA72   ≠,≠YA73   ≠,≠YA74   ≠,≠YA75   ≠,≠YA76   ≠,
     *      ≠YA77   ≠,≠YA78   ≠,≠YA79   ≠,≠YA80   ≠,≠YA81   ≠,≠YA82   ≠,
     *      ≠YA83   ≠,≠YA84   ≠,≠YA85   ≠,≠YA86   ≠,≠YA87   ≠,≠YA88   ≠,
     *      ≠YA89   ≠,≠YA90   ≠,≠YA91   ≠,≠YA92   ≠,≠YA93   ≠,≠YA94   ≠,
     *      ≠YA95   ≠,≠YA96   ≠,≠YA97   ≠,≠YA98   ≠,≠YA99   ≠,≠YA100  ≠/
      DATA  (Y(2,MA),MA=1,100)/≠YB1    ≠,≠YB2    ≠,≠YB3    ≠,≠YB4    ≠,
```

```
*≠YB5     ≠,
*         ≠YB6    ≠,≠YB7    ≠,≠YB8    ≠,≠YB9    ≠,≠YB10   ≠,≠YB11   ≠,
*         ≠YB12   ≠,≠YB13   ≠,≠YB14   ≠,≠YB15   ≠,≠YB16   ≠,≠YB17   ≠,
*         ≠YB18   ≠,≠YB19   ≠,≠YB20   ≠,         ≠YB21   ≠,≠YB22   ≠,
*         ≠YB23   ≠,≠YB24   ≠,≠YB25   ≠,≠YB26   ≠,≠YB27   ≠,≠YB28   ≠,
*         ≠YB29   ≠,≠YB30   ≠,≠YB31   ≠,≠YB32   ≠,≠YB33   ≠,≠YB34   ≠,
*         ≠YB35   ≠,≠YB36   ≠,≠YB37   ≠,≠YB38   ≠,≠YB39   ≠,≠YB40   ≠,
*         ≠YB41   ≠,≠YB42   ≠,≠YB43   ≠,≠YB44   ≠,≠YB45   ≠,≠YB46   ≠,
*         ≠YB47   ≠,≠YB48   ≠,≠YB49   ≠,≠YB50   ≠,≠YB51   ≠,≠YB52   ≠,
*         ≠YB53   ≠,≠YB54   ≠,≠YB55   ≠,≠YB56   ≠,≠YB57   ≠,≠YB58   ≠,
*         ≠YB59   ≠,≠YB60   ≠,≠YB61   ≠,≠YB62   ≠,≠YB63   ≠,≠YB64   ≠,
*         ≠YB65   ≠,≠YB66   ≠,≠YB67   ≠,≠YB68   ≠,≠YB69   ≠,≠YB70   ≠,
*         ≠YB71   ≠,≠YB72   ≠,≠YB73   ≠,≠YB74   ≠,≠YB75   ≠,≠YB76   ≠,
*         ≠YB77   ≠,≠YB78   ≠,≠YB79   ≠,≠YB80   ≠,≠YB81   ≠,≠YB82   ≠,
*         ≠YB83   ≠,≠YB84   ≠,≠YB85   ≠,≠YB86   ≠,≠YB87   ≠,≠YB88   ≠,
*         ≠YB89   ≠,≠YB90   ≠,≠YB91   ≠,≠YB92   ≠,≠YB93   ≠,≠YB94   ≠,
*         ≠YB95   ≠,≠YB96   ≠,≠YB97   ≠,≠YB98   ≠,≠YB99   ≠,≠YB100  ≠/
 DATA     ,Y(3,MA),MA=1,100)/≠YC1   ≠,≠YC2    ≠,≠YC3    ≠,≠YC4    ≠,
*≠YC5     ≠,
*         ≠YC6    ≠,≠YC7    ≠,≠YC8    ≠,≠YC9    ≠,≠YC10   ≠,≠YC11   ≠,
*         ≠YC12   ≠,≠YC13   ≠,≠YC14   ≠,≠YC15   ≠,≠YC16   ≠,≠YC17   ≠,
*         ≠YC18   ≠,≠YC19   ≠,≠YC20   ≠,         ≠YC21   ≠,≠YC22   ≠,
*         ≠YC23   ≠,≠YC24   ≠,≠YC25   ≠,≠YC26   ≠,≠YC27   ≠,≠YC28   ≠,
*         ≠YC29   ≠,≠YC30   ≠,≠YC31   ≠,≠YC32   ≠,≠YC33   ≠,≠YC34   ≠,
*         ≠YC35   ≠,≠YC36   ≠,≠YC37   ≠,≠YC38   ≠,≠YC39   ≠,≠YC40   ≠,
*         ≠YC41   ≠,≠YC42   ≠,≠YC43   ≠,≠YC44   ≠,≠YC45   ≠,≠YC46   ≠,
*         ≠YC47   ≠,≠YC48   ≠,≠YC49   ≠,≠YC50   ≠,≠YC51   ≠,≠YC52   ≠,
*         ≠YC53   ≠,≠YC54   ≠,≠YC55   ≠,≠YC56   ≠,≠YC57   ≠,≠YC58   ≠,
*         ≠YC59   ≠,≠YC60   ≠,≠YC61   ≠,≠YC62   ≠,≠YC63   ≠,≠YC64   ≠,
*         ≠YC65   ≠,≠YC66   ≠,≠YC67   ≠,≠YC68   ≠,≠YC69   ≠,≠YC70   ≠,
*         ≠YC71   ≠,≠YC72   ≠,≠YC73   ≠,≠YC74   ≠,≠YC75   ≠,≠YC76   ≠,
*         ≠YC77   ≠,≠YC78   ≠,≠YC79   ≠,≠YC80   ≠,≠YC81   ≠,≠YC82   ≠,
*         ≠YC83   ≠,≠YC84   ≠,≠YC85   ≠,≠YC86   ≠,≠YC87   ≠,≠YC88   ≠,
*         ≠YC89   ≠,≠YC90   ≠,≠YC91   ≠,≠YC92   ≠,≠YC93   ≠,≠YC94   ≠,
*         ≠YC95   ≠,≠YC96   ≠,≠YC97   ≠,≠YC98   ≠,≠YC99   ≠,≠YC100  ≠/
 DATA     ,Y(4,MA),MA=1,100)/≠YD1   ≠,≠YD2    ≠,≠YD3    ≠,≠YD4    ≠,
*≠YD5     ≠,
*         ≠YD6    ≠,≠YD7    ≠,≠YD8    ≠,≠YD9    ≠,≠YD10   ≠,≠YD11   ≠,
*         ≠YD12   ≠,≠YD13   ≠,≠YD14   ≠,≠YD15   ≠,≠YD16   ≠,≠YD17   ≠,
*         ≠YD18   ≠,≠YD19   ≠,≠YD20   ≠,         ≠YD21   ≠,≠YD22   ≠,
*         ≠YD23   ≠,≠YD24   ≠,≠YD25   ≠,≠YD26   ≠,≠YD27   ≠,≠YD28   ≠,
*         ≠YD29   ≠,≠YD30   ≠,≠YD31   ≠,≠YD32   ≠,≠YD33   ≠,≠YD34   ≠,
*         ≠YD35   ≠,≠YD36   ≠,≠YD37   ≠,≠YD38   ≠,≠YD39   ≠,≠YD40   ≠,
*         ≠YD41   ≠,≠YD42   ≠,≠YD43   ≠,≠YD44   ≠,≠YD45   ≠,≠YD46   ≠,
*         ≠YD47   ≠,≠YD48   ≠,≠YD49   ≠,≠YD50   ≠,≠YD51   ≠,≠YD52   ≠,
*         ≠YD53   ≠,≠YD54   ≠,≠YD55   ≠,≠YD56   ≠,≠YD57   ≠,≠YD58   ≠,
*         ≠YD59   ≠,≠YD60   ≠,≠YD61   ≠,≠YD62   ≠,≠YD63   ≠,≠YD64   ≠,
*         ≠YD65   ≠,≠YD66   ≠,≠YD67   ≠,≠YD68   ≠,≠YD69   ≠,≠YD70   ≠,
*         ≠YD71   ≠,≠YD72   ≠,≠YD73   ≠,≠YD74   ≠,≠YD75   ≠,≠YD76   ≠,
*         ≠YD77   ≠,≠YD78   ≠,≠YD79   ≠,≠YD80   ≠,≠YD81   ≠,≠YD82   ≠,
*         ≠YD83   ≠,≠YD84   ≠,≠YD85   ≠,≠YD86   ≠,≠YD87   ≠,≠YD88   ≠,
*         ≠YD89   ≠,≠YD90   ≠,≠YD91   ≠,≠YD92   ≠,≠YD93   ≠,≠YD94   ≠,
*         ≠YD95   ≠,≠YD96   ≠,≠YD97   ≠,≠YD98   ≠,≠YD99   ≠,≠YD100  ≠/
 DATA     ,Y(5,MA),MA=1,100)/≠YE1   ≠,≠YE2    ≠,≠YE3    ≠,≠YE4    ≠,
*≠YE5     ≠,
```

```
*       ≠YF6     ≠,≠YF7     ≠,≠YF8     ≠,≠YF9     ≠,≠YF10    ≠,≠YF11    ≠,
*       ≠YF12    ≠,≠YF13    ≠,≠YF14    ≠,≠YF15    ≠,≠YF16    ≠,≠YF17    ≠,
*       ≠YF18    ≠,≠YF19    ≠,≠YF20    ≠,           ≠YF21    ≠,≠YF22    ≠,
*       ≠YF23    ≠,≠YF24    ≠,≠YF25    ≠,≠YF26    ≠,≠YF27    ≠,≠YF28    ≠,
*       ≠YF29    ≠,≠YF30    ≠,≠YF31    ≠,≠YF32    ≠,≠YF33    ≠,≠YF34    ≠,
*       ≠YF35    ≠,≠YF36    ≠,≠YF37    ≠,≠YF38    ≠,≠YF39    ≠,≠YF40    ≠,
*       ≠YF41    ≠,≠YF42    ≠,≠YF43    ≠,≠YF44    ≠,≠YF45    ≠,≠YF46    ≠,
*       ≠YF47    ≠,≠YF48    ≠,≠YF49    ≠,≠YF50    ≠,≠YF51    ≠,≠YF52    ≠,
*       ≠YF53    ≠,≠YF54    ≠,≠YF55    ≠,≠YF56    ≠,≠YF57    ≠,≠YF58    ≠,
*       ≠YF59    ≠,≠YF60    ≠,≠YF61    ≠,≠YF62    ≠,≠YF63    ≠,≠YF64    ≠,
*       ≠YF65    ≠,≠YF66    ≠,≠YF67    ≠,≠YF68    ≠,≠YF69    ≠,≠YF70    ≠,
*       ≠YF71    ≠,≠YF72    ≠,≠YF73    ≠,≠YF74    ≠,≠YF75    ≠,≠YF76    ≠,
*       ≠YF77    ≠,≠YF78    ≠,≠YF79    ≠,≠YF80    ≠,≠YF81    ≠,≠YF82    ≠,
*       ≠YF83    ≠,≠YF84    ≠,≠YF85    ≠,≠YF86    ≠,≠YF87    ≠,≠YF88    ≠,
*       ≠YF89    ≠,≠YF90    ≠,≠YF91    ≠,≠YF92    ≠,≠YF93    ≠,≠YF94    ≠,
*       ≠YF95    ≠,≠YF96    ≠,≠YF97    ≠,≠YF98    ≠,≠YF99    ≠,≠YF100   ≠/
 DATA   (Y(6,MA),MA=1,100)/≠YF1       ≠,≠YF2     ≠,≠YF3     ≠,≠YF4     ≠,
*≠YF5    ≠,
*       ≠YF6     ≠,≠YF7     ≠,≠YF8     ≠,≠YF9     ≠,≠YF10    ≠,≠YF11    ≠,
*       ≠YF12    ≠,≠YF13    ≠,≠YF14    ≠,≠YF15    ≠,≠YF16    ≠,≠YF17    ≠,
*       ≠YF18    ≠,≠YF19    ≠,≠YF20    ≠,           ≠YF21    ≠,≠YF22    ≠,
*       ≠YF23    ≠,≠YF24    ≠,≠YF25    ≠,≠YF26    ≠,≠YF27    ≠,≠YF28    ≠,
*       ≠YF29    ≠,≠YF30    ≠,≠YF31    ≠,≠YF32    ≠,≠YF33    ≠,≠YF34    ≠,
*       ≠YF35    ≠,≠YF36    ≠,≠YF37    ≠,≠YF38    ≠,≠YF39    ≠,≠YF40    ≠,
*       ≠YF41    ≠,≠YF42    ≠,≠YF43    ≠,≠YF44    ≠,≠YF45    ≠,≠YF46    ≠,
*       ≠YF47    ≠,≠YF48    ≠,≠YF49    ≠,≠YF50    ≠,≠YF51    ≠,≠YF52    ≠,
*       ≠YF53    ≠,≠YF54    ≠,≠YF55    ≠,≠YF56    ≠,≠YF57    ≠,≠YF58    ≠,
*       ≠YF59    ≠,≠YF60    ≠,≠YF61    ≠,≠YF62    ≠,≠YF63    ≠,≠YF64    ≠,
*       ≠YF65    ≠,≠YF66    ≠,≠YF67    ≠,≠YF68    ≠,≠YF69    ≠,≠YF70    ≠,
*       ≠YF71    ≠,≠YF72    ≠,≠YF73    ≠,≠YF74    ≠,≠YF75    ≠,≠YF76    ≠,
*       ≠YF77    ≠,≠YF78    ≠,≠YF79    ≠,≠YF80    ≠,≠YF81    ≠,≠YF82    ≠,
*       ≠YF83    ≠,≠YF84    ≠,≠YF85    ≠,≠YF86    ≠,≠YF87    ≠,≠YF88    ≠,
*       ≠YF89    ≠,≠YF90    ≠,≠YF91    ≠,≠YF92    ≠,≠YF93    ≠,≠YF94    ≠,
*       ≠YF95    ≠,≠YF96    ≠,≠YF97    ≠,≠YF98    ≠,≠YF99    ≠,≠YF100   ≠/
 DATA   (Y(7,MA),MA=1,100)/≠YG1       ≠,≠YG2     ≠,≠YG3     ≠,≠YG4     ≠,
*≠YG5    ≠,
*       ≠YG6     ≠,≠YG7     ≠,≠YG8     ≠,≠YG9     ≠,≠YG10    ≠,≠YG11    ≠,
*       ≠YG12    ≠,≠YG13    ≠,≠YG14    ≠,≠YG15    ≠,≠YG16    ≠,≠YG17    ≠,
*       ≠YG18    ≠,≠YG19    ≠,≠YG20    ≠,           ≠YG21    ≠,≠YG22    ≠,
*       ≠YG23    ≠,≠YG24    ≠,≠YG25    ≠,≠YG26    ≠,≠YG27    ≠,≠YG28    ≠,
*       ≠YG29    ≠,≠YG30    ≠,≠YG31    ≠,≠YG32    ≠,≠YG33    ≠,≠YG34    ≠,
*       ≠YG35    ≠,≠YG36    ≠,≠YG37    ≠,≠YG38    ≠,≠YG39    ≠,≠YG40    ≠,
*       ≠YG41    ≠,≠YG42    ≠,≠YG43    ≠,≠YG44    ≠,≠YG45    ≠,≠YG46    ≠,
*       ≠YG47    ≠,≠YG48    ≠,≠YG49    ≠,≠YG50    ≠,≠YG51    ≠,≠YG52    ≠,
*       ≠YG53    ≠,≠YG54    ≠,≠YG55    ≠,≠YG56    ≠,≠YG57    ≠,≠YG58    ≠,
*       ≠YG59    ≠,≠YG60    ≠,≠YG61    ≠,≠YG62    ≠,≠YG63    ≠,≠YG64    ≠,
*       ≠YG65    ≠,≠YG66    ≠,≠YG67    ≠,≠YG68    ≠,≠YG69    ≠,≠YG70    ≠,
*       ≠YG71    ≠,≠YG72    ≠,≠YG73    ≠,≠YG74    ≠,≠YG75    ≠,≠YG76    ≠,
*       ≠YG77    ≠,≠YG78    ≠,≠YG79    ≠,≠YG80    ≠,≠YG81    ≠,≠YG82    ≠,
*       ≠YG83    ≠,≠YG84    ≠,≠YG85    ≠,≠YG86    ≠,≠YG87    ≠,≠YG88    ≠,
*       ≠YG89    ≠,≠YG90    ≠,≠YG91    ≠,≠YG92    ≠,≠YG93    ≠,≠YG94    ≠,
*       ≠YG95    ≠,≠YG96    ≠,≠YG97    ≠,≠YG98    ≠,≠YG99    ≠,≠YG100   ≠/
 DATA   (Y(8,MA),MA=1,100)/≠YH1       ≠,≠YH2     ≠,≠YH3     ≠,≠YH4     ≠,
```

```
*≠YH5     ≠,
*         ≠YH6     ≠,≠YH7     ≠,≠YH8     ≠,≠YH9     ≠,≠YH10    ≠,≠YH11    ≠,
*         ≠YH12    ≠,≠YH13    ≠,≠YH14    ≠,≠YH15    ≠,≠YH16    ≠,≠YH17    ≠,
*         ≠YH18    ≠,≠YH19    ≠,≠YH20    ≠,          ≠YH21    ≠,≠YH22    ≠,
*         ≠YH23    ≠,≠YH24    ≠,≠YH25    ≠,≠YH26    ≠,≠YH27    ≠,≠YH28    ≠,
*         ≠YH29    ≠,≠YH30    ≠,≠YH31    ≠,≠YH32    ≠,≠YH33    ≠,≠YH34    ≠,
*         ≠YH35    ≠,≠YH36    ≠,≠YH37    ≠,≠YH38    ≠,≠YH39    ≠,≠YH40    ≠,
*         ≠YH41    ≠,≠YH42    ≠,≠YH43    ≠,≠YH44    ≠,≠YH45    ≠,≠YH46    ≠,
*         ≠YH47    ≠,≠YH48    ≠,≠YH49    ≠,≠YH50    ≠,≠YH51    ≠,≠YH52    ≠,
*         ≠YH53    ≠,≠YH54    ≠,≠YH55    ≠,≠YH56    ≠,≠YH57    ≠,≠YH58    ≠,
*         ≠YH59    ≠,≠YH60    ≠,≠YH61    ≠,≠YH62    ≠,≠YH63    ≠,≠YH64    ≠,
*         ≠YH65    ≠,≠YH66    ≠,≠YH67    ≠,≠YH68    ≠,≠YH69    ≠,≠YH70    ≠,
*         ≠YH71    ≠,≠YH72    ≠,≠YH73    ≠,≠YH74    ≠,≠YH75    ≠,≠YH76    ≠,
*         ≠YH77    ≠,≠YH78    ≠,≠YH79    ≠,≠YH80    ≠,≠YH81    ≠,≠YH82    ≠,
*         ≠YH83    ≠,≠YH84    ≠,≠YH85    ≠,≠YH86    ≠,≠YH87    ≠,≠YH88    ≠,
*         ≠YH89    ≠,≠YH90    ≠,≠YH91    ≠,≠YH92    ≠,≠YH93    ≠,≠YH94    ≠,
*         ≠YH95    ≠,≠YH96    ≠,≠YH97    ≠,≠YH98    ≠,≠YH99    ≠,≠YH100   ≠/
 DATA     (Y(9,MA),MA=1,100)/≠YI1    ≠,≠YI2    ≠,≠YI3    ≠,≠YI4    ≠,
*≠YI5     ≠,
*         ≠YI6     ≠,≠YI7     ≠,≠YI8     ≠,≠YI9     ≠,≠YI10    ≠,≠YI11    ≠,
*         ≠YI12    ≠,≠YI13    ≠,≠YI14    ≠,≠YI15    ≠,≠YI16    ≠,≠YI17    ≠,
*         ≠YI18    ≠,≠YI19    ≠,≠YI20    ≠,          ≠YI21    ≠,≠YI22    ≠,
*         ≠YI23    ≠,≠YI24    ≠,≠YI25    ≠,≠YI26    ≠,≠YI27    ≠,≠YI28    ≠,
*         ≠YI29    ≠,≠YI30    ≠,≠YI31    ≠,≠YI32    ≠,≠YI33    ≠,≠YI34    ≠,
*         ≠YI35    ≠,≠YI36    ≠,≠YI37    ≠,≠YI38    ≠,≠YI39    ≠,≠YI40    ≠,
*         ≠YI41    ≠,≠YI42    ≠,≠YI43    ≠,≠YI44    ≠,≠YI45    ≠,≠YI46    ≠,
*         ≠YI47    ≠,≠YI48    ≠,≠YI49    ≠,≠YI50    ≠,≠YI51    ≠,≠YI52    ≠,
*         ≠YI53    ≠,≠YI54    ≠,≠YI55    ≠,≠YI56    ≠,≠YI57    ≠,≠YI58    ≠,
*         ≠YI59    ≠,≠YI60    ≠,≠YI61    ≠,≠YI62    ≠,≠YI63    ≠,≠YI64    ≠,
*         ≠YI65    ≠,≠YI66    ≠,≠YI67    ≠,≠YI68    ≠,≠YI69    ≠,≠YI70    ≠,
*         ≠YI71    ≠,≠YI72    ≠,≠YI73    ≠,≠YI74    ≠,≠YI75    ≠,≠YI76    ≠,
*         ≠YI77    ≠,≠YI78    ≠,≠YI79    ≠,≠YI80    ≠,≠YI81    ≠,≠YI82    ≠,
*         ≠YI83    ≠,≠YI84    ≠,≠YI85    ≠,≠YI86    ≠,≠YI87    ≠,≠YI88    ≠,
*         ≠YI89    ≠,≠YI90    ≠,≠YI91    ≠,≠YI92    ≠,≠YI93    ≠,≠YI94    ≠,
*         ≠YI95    ≠,≠YI96    ≠,≠YI97    ≠,≠YI98    ≠,≠YI99    ≠,≠YI100   ≠/
 DATA     (Y(10,MA),MA=1,100)/≠YJ1   ≠,≠YJ2    ≠,≠YJ3    ≠,≠YJ4    ≠,
*≠YJ5     ≠,
*         ≠YJ6     ≠,≠YJ7     ≠,≠YJ8     ≠,≠YJ9     ≠,≠YJ10    ≠,≠YJ11    ≠,
*         ≠YJ12    ≠,≠YJ13    ≠,≠YJ14    ≠,≠YJ15    ≠,≠YJ16    ≠,≠YJ17    ≠,
*         ≠YJ18    ≠,≠YJ19    ≠,≠YJ20    ≠,          ≠YJ21    ≠,≠YJ22    ≠,
*         ≠YJ23    ≠,≠YJ24    ≠,≠YJ25    ≠,≠YJ26    ≠,≠YJ27    ≠,≠YJ28    ≠,
*         ≠YJ29    ≠,≠YJ30    ≠,≠YJ31    ≠,≠YJ32    ≠,≠YJ33    ≠,≠YJ34    ≠,
*         ≠YJ35    ≠,≠YJ36    ≠,≠YJ37    ≠,≠YJ38    ≠,≠YJ39    ≠,≠YJ40    ≠,
*         ≠YJ41    ≠,≠YJ42    ≠,≠YJ43    ≠,≠YJ44    ≠,≠YJ45    ≠,≠YJ46    ≠,
*         ≠YJ47    ≠,≠YJ48    ≠,≠YJ49    ≠,≠YJ50    ≠,≠YJ51    ≠,≠YJ52    ≠,
*         ≠YJ53    ≠,≠YJ54    ≠,≠YJ55    ≠,≠YJ56    ≠,≠YJ57    ≠,≠YJ58    ≠,
*         ≠YJ59    ≠,≠YJ60    ≠,≠YJ61    ≠,≠YJ62    ≠,≠YJ63    ≠,≠YJ64    ≠,
*         ≠YJ65    ≠,≠YJ66    ≠,≠YJ67    ≠,≠YJ68    ≠,≠YJ69    ≠,≠YJ70    ≠,
*         ≠YJ71    ≠,≠YJ72    ≠,≠YJ73    ≠,≠YJ74    ≠,≠YJ75    ≠,≠YJ76    ≠,
*         ≠YJ77    ≠,≠YJ78    ≠,≠YJ79    ≠,≠YJ80    ≠,≠YJ81    ≠,≠YJ82    ≠,
*         ≠YJ83    ≠,≠YJ84    ≠,≠YJ85    ≠,≠YJ86    ≠,≠YJ87    ≠,≠YJ88    ≠,
*         ≠YJ89    ≠,≠YJ90    ≠,≠YJ91    ≠,≠YJ92    ≠,≠YJ93    ≠,≠YJ94    ≠,
*         ≠YJ95    ≠,≠YJ96    ≠,≠YJ97    ≠,≠YJ98    ≠,≠YJ99    ≠,≠YJ100   ≠/
```

```
*** INPUT THE FLAGS, NO. OF COLUMNS AND ROWS.
    READ (5,500) IOM,M1,COL,ROW

*** INPUT THE TITLE OF YOUR PROBLEM.
    READ (5,501) TITLE

*** WRITE OUT THE ALGORITHM≠S NAME AND TITLE
    WRITE (1,100) METHOD(M1),VARS(1),TITLE

*** TEST WHETHER THIS IS AN I.P. PROBLEM.
    IF IT IS AN I.P. PROBLEM, READ THE INT. VARS AND WRITE TO TAPE 1.
    O.W. CONTINUE THE PROCESS.
    IF ((M1 .LT. 5) .OR. (M1 .GT. 7)) GO TO 5
    READ (5,503) N2,(ACT(I1),I1=1,N2)
    WRITE (1,101) VARS(2), (ACT(I1),I1=1,N2)
  5 CONTINUE

*** READ INTO ALL OF THE ACTIVITIES
    READ (5,505) (ACT(IA),IA=1,COL)

*** READ INTO THE COEFFIENTS OF OBJECT FUNCTION.
    J1=1
 10 CONTINUE
    READ (5,506) (ICOL(IB),SIGN(IB),COFF(IB),IB=J1,J1+4)
    DO 15 IC=J1,J1+4
    IF (ICOL(IC) .EQ. -999) GO TO 20
 15 CONTINUE
    J1=J1+5
    GO TO 10

*** NOOB IS THE NO. OF NON-ZERO COFF. OF OBJECTIVE FUNCTION.
 20 CONTINUE
    NOOB = IC - 1

*** WRITE OUT THE COEF. OF OBJ. FUNCTION TO PERMENANT FILE (TAPE 2)
    WRITE (2,201) (SIGN(ID),COFF(ID),ICOL(ID),ID=1,NOOB)
```

```
*** TEST IF THERE IS NO ANY CONSTRAINT IN NON-ARGUMENTED
    CONSTRAINT MATRIX.
    IF (ROW .EQ. 0) GO TO 43

*** READ INTO THE COEFFIENTS OF CONSTRAINTS.
    J2=1 $ K1=0  $IEW=0
 25 CONTINUE
    DO 50 IF=1,ROW
 27 CONTINUE
    READ (4,506) (ICOL(IF),SIGN(IF),COEFF(IF),IF=J2,J2+4)
    DO 30 IG=J2,J2+4
    IF (ICOL(IG) .LT. 0) GO TO 40
 30 CONTINUE
    J2=J2+5
    GO TO 27
 40 CONTINUE

*** TEST IF THIS ROW CONTAINS ONLY ONE COEF. WITH *LE* RELATION.
    IF YES, THEN PUT IT IN TEMP. LOC.
    OTHERWISE CONTINUE THE PROCESS.
    IF (IG .GT. 2)    GO TO 45
    IF (ICOL(IG) .LT. -100 ) GO TO 45
    K1=K1+1
    IACT(K1)  = ACT(ICOL(1))
    TT2(K1)=SIGN(IG)
    TT3(K1)=COEFF(IG)/COEF(1)
    GO TO 48
 45 CONTINUE

*** NOCO CANTAINS THE NO. OF NON-ZERO COEFF. OF CONSTRAINT.
    IEW=IEW+1
    NOCO(IEW) = IG - 1

*** WRITE OUT THE COEFF. AND R.H.S. OF EACH CONSTRAINT.
    WRITE (2,201) (SIGN(IH),COEFF(IH),ICOL(IH),IH=1,NOCO(IEW))
    IGFL=ICOL(IG)/(-100)
    WRITE (2,202) GEL(IGFL),SIGN(IG),COEFF(IG)

 48 CONTINUE
    J2=1
 50 CONTINUE

*** NORD CANTAINS THE NO. OF BOUNDS.
*** NOCO CANTAINS THE NO. OF CONSTRAINTS.
    NORD = K1      $    NOCO = ROW-NORD
```

```
 43 CONTINUE
*** TEST IF THERE ARE SOME NEW ACTIVITIES NEED TO BE INSERTED INTO
    THE #OLD# MATRIX.   (ISTO = 99999)
    IF YES, THEN CALL SUBROUTINE #INSERT#,
    OTHERWISE, CONTINUE THE NORMAL PROCESS.
    LOOP = 1
    ITAPE2 = 2
 44 CONTINUE
    READ (2,511) ISTO
    IF (EOF(5) .NE. 0.) GO TO 65
    IF (ISTO .NE. 99999) GO TO 52
    ILOOP = LOOP/2   $ILOOP=ILOOP*2   $ITAPE1 = 2   $ITAPE2 = 7
    IF (ILOOP .NE. LOOP) GO TO 44
    ITAPE1 = 7   $ITAPE2 = 2
 44 CONTINUE
    CALL  INSERT
    GO TO 44
 52 CONTINUE

*** TEST IF THE PROBLEM NEEDS TO BE EXTENDED (CHECK EOF)
 53 CONTINUE
    JA=JA+1
    READ (2,510) EID,RW ,IDPV(JA)
    IF (EOF(5) .NE. 0.) GO TO 65
    IF (ABS(RW) .LE. 10.**(-10)) GO TO 54
    ONE = RW
 54 CONTINUE

*** TEST IF THIS IS AN EXTENTION OF TWO DEP. VAR. FUNCTIONS
    IF YES  THEN CALL SUBROUTINE #DEPVA
    O.W. CONTINUE THE PROCESS.
    WRITE (3,507) EID,ONE
    IF (IDPV(JA) .EQ. 0) GO TO 55
    CALL  DEPV
    GO TO 57
 55 CONTINUE
    IF (EID .EQ. 0) GO TO 56
    READ (2,506) (ICOL(II),SIGN(II),COFF(II),II=1,EID)

*** UPDATE THE COL. COORD. OF NON-ZERO COEF. IF COL. .GT. NINS.
    IF (ISTO .NE. 99999) GO TO 46
    DO 42 II=1,EID
    IF (ICOL(II) .GT. NINS) ICOL(II)=ICOL(II)+NAA
 42 CONTINUE

 46 CONTINUE
    WRITE (3,201) (SIGN(II),COFF(II),ICOL(II),II=1,EID)
 56 CONTINUE
```

```
*** PROCESS THE EXTENDING PROCEDURE.
    READ INTO THE FORMULA
    READ (5,800) IJ,(CC(IK),IEXP(IK),IK=1,IJ)

*** READ THE INITIAL VALUE OF Q, DELTA Q AND STEPS.
    READ (5,800) Q(1),DELTAQ,STEP

*** SET UP  THE NEW ACTIVITIES.
 57 CONTINUE
    DO 58 JA=1,STEP
    ACT(KA,COL)=Y(JA,KA)
 58 CONTINUE

*** CALL THE SUBROUTINE GEN TO GENERATE THE VALUES OF Q AND W.
    IF (IJ.V(JA) .NE. 0) GO TO 62
    CALL GEN

*** EXTENDING THE OBJECT FUNCTION.
    IQ=1
    DO 60 IP= COL+1, COL+STEP
    SIGN(IQ)=WSIGN(IQ)
    COFF(IQ)=ABS(W(IQ))     $ ICOL(IP)= COL  +IQ
    IQ=IQ+1
 60 CONTINUE
    GO TO 64

 62 CONTINUE
    IQ=1
    DO 63 IP=COL+1,COL+STEP
    SIGN(IQ) = ≠+≠
    COEFF(IQ) = 0.0     $ICOL(IP) = COL+IQ
    IQ = IQ+1
 63 CONTINUE

*** STORE THE VALUES OF ROW ,COL, AND STEP
 64 CONTINUE
    NROW(JA)=ROW   $NCOL(JA)=COL   $NSTEP(JA)=STEP $NEID(JA)=EID

*** EXTENDING THE CONSTRAINTS.
    IF (IJ.V(JA) .NE. 0) GO TO 66
    WRITE (3,800) (W(II),II=1,NSTEP(JA))
```

```
*** UPDATE THE VALUES OF COL AND ROW.
 66 CONTINUE
    COL=COL+STEP    -    ROW = ROW+2
    IF (IDAV(JA) .NE. 0) ROW = ROW+4
    GO TO 63
 65 CONTINUE
    JA=JA-1

*** SET UP THE VARIABLES.
    WRITE (1,103) VARS(3)
    WRITE (1,504) (ACT(JN),JN=1,COL)

*** REWIND THE  3  PERMENANT FILES (ITAPE2.TAPE3.TAPE4)
    REWIND ITAPE2
    REWIND 3
    REWIND 4

*** SET UP THE FINAL OBJECT FUNCTION.
    IDM=IDM+4
    WRITE (1,103) VARS(IDM)
    READ (ITAPE2,201) (SIGN(ID),COFF(ID),ICOL(ID) ,ID=1,NOOB)
    WRITE(1,102) (SIGN(ID),COEF(ID),ACT(ICOL(ID)),ID=1,NOOB)
    IF (JA .EQ. 0) GO TO 68
    WRITE (1,102) (SIGN(ID),COFF(ID),ACT(ICOL(ID)),ID=NCOL(1)+1,COL)

*** SET UP THE FINAL CONSTRAINTS.
 68 CONTINUE
    WRITE (1,103) VARS(6)
    DO 70 JC=1,NOCO
    READ (ITAPE2,201) (SIGN(IH),COFF(IH),ICOL(IH),IH=1,NOCO(JC))
    WRITE(1,102) (SIGN(IH),COEF(IH),ACT(ICOL(IH)),IH=1,NOCO(JC))
    READ (ITAPE2,202) TTT1,TTT2,TTT3
    WRITE(1,202) TTT1,TTT2,TTT3
 70 CONTINUE
    IF (JA .EQ. 0) GO TO 85
    IAW = NCOL(1)
    DO 80 JK=1,JA
    READ (3,507) FID,ONE
    IF (IDAV(JK) .EQ. 0) GO TO 74
    DO 840 I3=1,FID
    READ (3,511) IFID(I3)
    IIF = IFID(I3) + 1
    READ (4,506) (ICOL(I1),SIGN(I1),COFF(I1),I1=1,IIF)
    WRITE (1,102) (SIGN(I1),COFF(I1),ACT(ICOL(I1)),I1=1,IFID(I3))
```

```
      ICT = NCOL(IIE)/(-100)
      READ  (4,599) (CSIGN(IB),C(IB),IB=1,NSTEP(JK))
      WRITE (1,102) (CSIGN(IB),C(IB),ACT(NCOL(JK)+IB),IB=1,NSTEP(JK))
      WRITE (1,202) GEL(ICT),SIGN(IIE),COEF(IIE)
  840 CONTINUE
      OONE = 1.0
      WRITE (1,102)  (YSIGN,OONE,ACT(NCOL(JK)+I2),I2=1,NSTEP(JK))
      WRITE (1,202) GEL(1),YSIGN,ONE
      GO TO 80
   74 CONTINUE
      IF (EID .EQ. 0) GO TO 75
      READ (3,201) (SIGN(II),COEF(II),ICOL(II),II=1,EID)
      WRITE(1,102) (SIGN(II),COEF(II),ACT(ICOL(II)),II=1,EID)
   75 CONTINUE
      READ  (3,300) (Q(IT),IT=1,NSTEP(JK))
      WRITE(1,102)(XSIGN,Q(IT-NCOL(JK)) ,ACT(II),II=NCOL(JK)+1,NCOL(JK)+
     *NSTEP(JK))
      IV=3
      IWW=NSTEP(JK)/2 + 1 +IWW
      IF (SIGN(IWW) .EQ. ≠+≠) GO TO 77
      IV=1
   77 CONTINUE
      IWW = IWW+NSTEP(JK)/2-1
      WRITE(1,202) GEL(IV),YSIGN,ZERO
      WRITE(1,102) (YSIGN,ONE,ACT(II),II=NCOL(JK)+1,NCOL(JK)+NSTEP(JK))
      WRITE (1,202) GEL(1),YSIGN,ONE
   80 CONTINUE


*** IF NOBD EQUAL TO 0, THEN THERE IS NO BOUND EXITS.
    ELSE PRINT OUT THE VERB ≠BOUND≠ AND SET UP THE BOUND SECTION.
   85 CONTINUE
      IF (NOBD .EQ. 0) GO TO 90
      WRITE (1,103) VARS(7)
      DO 88 KB=1,NOBD
      WRITE (1,104) TACT(KB),GEL(1),TT2(KB),TT3(KB)
   88 CONTINUE
   90 CONTINUE


*** PRINT OUT THE VERB ≠OPTIMIZE≠.
      WRITE (1,103) VARS(9)


*** IF IT IS FOR MPOS-APEX, THEN PRINT ≠ENDAPEX≠ AND STOP
    ELSE STOP.
      IF ((MI .NE. 11) .AND. (MI .NE. 12)) STOP
      WRITE (1,103) VARS(10)


*** FORMATS.
```

```
00     FORMAT (A10/A10/8A10)
01     FORMAT (A10/8(A7,3X))
02     FORMAT (3(A1,F16.5,A7))
03     FORMAT (A10)
04     FORMAT (A7,A10,A1,F16.5)
01     FORMAT (5(A1,F16.5,I4))
02     FORMAT (A10,A1,F16.5)
00     FORMAT (8F16.2)
00     FORMAT (I1,I2,2I5)
01     FORMAT (8A10)
02     FORMAT (F10.2)
03     FORMAT (I3,11A7/(3X,11A7))
04     FORMAT (8(3X,A7))
05     FORMAT (3X,11A7)
06     FORMAT (5(I4,A1,F11.2))
07     FORMAT (I5,F16.8)
08     FORMAT (I5,5(F10.4,F6.0))
09     FORMAT (2F10.4,I5)
10     FORMAT (I5,F10.2,I6)
11     FORMAT (I5)
99     FORMAT (5(A1,F15.3))
       STOP
       END


       SUBROUTINE  GEN
       INTEGER COL,ROW ,FID,STEP
       REAL W-THO,TEXP,C
       COMMON /T111/Q(1000),W(1000),CC(10),TEXP(10) ,WSIGN(1000)
       COMMON /T112/DELTAQ,COL,ROW,FID,STEP,IJ,IDM,M1,N2,NOBD ,NDOR

*** SET UP AND CALCULATE THE FORMULA
       DO 8 IL=1,STEP
       W(IL)=0.0
       DO 3 IM=1,IJ
       W(IL) = W(IL) + CC(IM)*Q(IL)**TEXP(IM)
     3 CONTINUE
       Q(IL+1)=Q(IL)+DELTAQ
     8 CONTINUE

*** DEFINE THE SIGNS OF W.
       DO 13 IN=1,STEP
       WSIGN(IN)=1H+
       IF (W(IN) .GE. 0.0) GO TO 13
       WSIGN(IN)=1H-
    13 CONTINUE
       RETURN
       END
```

```
      SUBROUTINE  DERV
      INTEGER COL,ROW ,FID,STEP
      REAL METHD,IEXP,L
      COMMON /T111/Q(1000),A(1000),CC(10),IEXP(10) ,WSIGN(1000)
      COMMON /T112/DELTAQ,COL,ROW,FID,STEP,IJ,IOW,M1,N2,NOBD ,NOOB
      COMMON  /T113/TITLE(8),SEL(3),ACT(1200),ICOL(1200),SIGN(1200),
     *        COFF(1200),METHD(12),VARS(26)
      COMMON  /T118/RATIO(100),C(1000),L(1000),IDPV(50)
      COMMON  /T119/IEID(100),CSIGN(1000),LSIGN(1000),DSIGN(1000)
      COMMON  /T121/ISID,NAA,NBB,NINS,AACT(20),AICOL(20),ASIGN(20),ISIF.
     *        ACOFF(20),AEWROW(50),AEWSIGN(50),AEWCOFF(50),AEWCOL(50)

*** READ INTO THE NO. OF NON-ZERO COEF. OF EACH ROW
      DO 800 I3=1,EID
      READ (-,507) IEID(I3)
      READ (-,506) (ICOL(I1),SIGN(I1),COFF(I1),I1=1,IEID(I3)+1)

*** UPDATE THE COL. COORD. OF NON-ZERO COEF. IF COL. .GT. NINS.
      IF (ISIF .NE. 99999) GO TO 701
      DO 700 I1=1,IEID(I3)
      IF (ICOL(I1) .GT. NINS) ICOL(I1)=ICOL(I1)+NAA
700   CONTINUE
701   CONTINUE

*** WRITE OUT TO PERMENANT FILE (TAPE 3)
      WRITE (3,507) IEID(I3)
      WRITE (3,506) (ICOL(I1),SIGN(I1),COFF(I1),I1=1,IEID(I3)+1)
800   CONTINUE

*** READ INTO THE NO. OF RATIOS DESIRED (NOR) AND RATIOS (RATIO)
      READ (-,584) NOR,(RATIO(I2),I2=1,NOR)

*** READ INTO THE FORMULA.
      READ (-,585) AA,ALPH1,ALPH2
      XXX = ALPH1 + ALPH2
      XXA = ./XXX
      XXB = _ALPH1/XXX

*** READ INTO THE INITIAL VALUES OF Q, DELTA Q, AND STEPS.
      READ (-,509) Q(1),DELTAQ,STEP
```

```
*** COMPUTE THE VALUES OF Q, L, AND C.
      DO 806  I5=1,NOR
      DO 805  I6=1,STEP
      III = I6+STEP*I5-STEP
      Q(III+1) = Q(III)+DELTAQ
      L(III) = (Q(III)**XXA) *(AA **(-XXA)) *(RATIO(I5)**(XXB))
      C(III) = RATIO(I5)*L(III)
805 CONTINUE
      Q(III+1) = Q(1)
806 CONTINUE


*** DETERMINE THE SIGNS OF ARRAYS Q,C, AND L.
      NRR = NOR*STEP
      DO 820 IN=1,NRR
      QSIGN(IN) = ≠+≠
      CSIGN(IN) = ≠+≠
      LSIGN(IN) = ≠+≠
      IF (Q(IN) .GE. 0.) GO TO 816
      Q(IN) = ≠-≠
816 CONTINUE
      IF (C(IN) .GE. 0.) GO TO 817
      C(IN) = ≠-≠
817 CONTINUE
      IF (L(IN) .GE. 0.) GO TO 818
      L(IN) = ≠-≠
818 CONTINUE
820 CONTINUE
      STEP = NRR


*** WRITE OUT TO PERMENANT FILE (TAPE 4)
      WRITE (4,599) (CSIGN(I8),C(I8),I8=1,NRR)
      WRITE (4,599) (LSIGN(I9),L(I9),I9=1,NRR)
      WRITE (4,599) (QSIGN(I1),Q(I1),I1=1,NRR)
```

```
*** FORMAT.
02    FORMAT (3(A1,F16.5,A7))
03    FORMAT (A10)
01    FORMAT (5(A1,F16.5,I4))
02    FORMAT (A10,A1,F16.5)
05    FORMAT (3X,11A7)
06    FORMAT (5(I4,A1,F11.2))
07    FORMAT (I5)
09    FORMAT (2F10.4,I5)
84    FORMAT (I3,11F7.2/(3X,11F7.2))
85    FORMAT (3F10.2)
99    FORMAT (5(A1,F15.3))
      RETURN
      END


      SUBROUTINE  INSERT
      INTEGER COL ,ROW ,FID,STEP
      INTEGER    AICOL,AEWCOL,AEWROW
      COMMON /T112/DELTAG,COL,ROW,EID,STEP,ID,IDM,M1,N2,NOBD ,NOOB
      COMMON  /T113/TITLE(8),GEL(3),ACT(1200),ICOL(1200),SIGN(1200),
     *        COFF(1200),METHD(15),VARS(26)
      COMMON  /T115/NSTEP(50),NOCO(500),NROW(50),NCOL(50),NEID(50)
      COMMON   /T120/XSIGN,YSIGN,ZERO,ONE,JA,JB,JC,NRR,ITAPE1,ITAPE2
      COMMON  /T121/ISID,NAA,NBB,NINS,AACT(20),AICOL(20),ASIGN(20),ISIE,
     *        ACOFF(20),AEWROW(50),AEWSIGN(50),AEWCOEF(50),AEWCOL(50)


*** READ INTO THE PLACEMENT OF NEW ACTIVITIES(NINS) IN
      OLD MATRIX, NAMES OF NEW ACTIVITIES(AACT), AND COFF.
      OF OBJECTIVE FUNCTION.
      ISIE = 99999
      READ (-,511) NINS
      READ (-,503) NAA,(AACT(IA),IA=1,NAA)
      READ (-,506) (AICOL(IB),ASIGN(IB),ACOEF(IB),IB=1,NAA)


*** READ INTO THE COEF. OF CONTRAINTS.
      READ (-,511) NBB
      READ (-,512) (AEWROW(IX),AEWCOL(IX),AEWSIGN(IX),AEWCOEF(IX),IX=
     *1,NBB)
```

```
*** INSERT THE NEW ACTIVITIES INTO #OLD# OBJ. FUCTION.
    DO 30 K1=1,COL
    IF (NINS .GE. K1) GO TO 40
 30 CONTINUE
 40 CONTINUE
    DO 45 K2=1,COL-NINS
    KAD=COL+NAA+1-K2
    KAE=COL+1-K2
    ACT(KAD)=ACT(KAE)
 45 CONTINUE
    DO 47 K2=1,NAA
    KAC=NINS+K2
    ACT(KAC)=AACT(K2)
 47 CONTINUE

*** UPDATE THE NO. OF ACTIVITIES.
    COL=COL+NAA

*** READ INTO THE #OLD# OBJECTIVE FUNCTION FROM ITAPE1.
    REWIND ITAPE1
    READ (ITAPE1,201) (SIGN(ID),COFF(ID),ICOL(ID),ID=1,NOOB)

*** START TO INSERT THE NEW ACTIVITIES INTO THE OLD OBJ. FUNC.
    DO 50 K1=1,NOOB
    IF (NINS .GE. ICOL(K1) .AND. NINS .LT. ICOL(K1+1)) GO TO 55
 50 CONTINUE
    GO TO 63

 55 CONTINUE
    DO 60 KAB =1,NOOB-NINS
    KAD = NOOB+NAA+1-KAB
    KAE = NOOB+1-KAB
    SIGN(KAD)=SIGN(KAE)
    COFF(KAD)=COFF(KAE)
    ICOL(KAD)=ICOL(KAE)+NAA
 60 CONTINUE

 63 CONTINUE
    DO 65 K2 = 1,NAA
    KAC = NINS + K2
    SIGN(KAC)=ASIGN(K2) $COFF(KAC)=ACOFF(K2)  $ICOL(KAC)=AICOL(K2)
 65 CONTINUE

*** UPDATE THE NO. OF COFF. OF OBJ. FUNC.(NOOB)
    NOOB = NOOB+NAA

*** WRITE OUT THE #NEW# OBJ. FUNC. TO ITAPE2.
    WRITE(ITAPE2,201) (SIGN(ID),COFF(ID),ICOL(ID),ID=1,NOOB)
```

```
*** DO THE INSERTION FOR CONSTRAINTS.
    DO 80 IF=1,ROW
    READ (ITAPE1,201) (SIGN(IH),COFF(IH),ICOL(IH),IH=1,NOCO(IF))
    READ (ITAPE1,202) TTT1,TTT2,TTT3
*** UPDATE THE COL. COORD. OF NON-ZERO COEF. IF COL. .GT. NINS.
    DO 71 I1=1,NOCO(IF)
    IF (ICOL(I1) .GT. NINS) ICOL(I1)=ICOL(I1)+NAA
 71 CONTINUE

    DO 70 ID=1,NBB
    IF (AEWROW(ID) .NE. IE) GO TO 70
    DO 67 IF = 1,NOCO(IE)
    IF (AEWCOL(ID) .GE. ICOL(IF) .AND. AEWCOL(ID) .LT. ICOL(IF+1)) GO
   * TO 68
 67 CONTINUE
    GO TO 75
 68 CONTINUE
    DO 69 IG=1,NOCO(IF)-IF
    IH = NOCO(IF)+1-IG
    SIGN(IH+1) = SIGN(IH)
    COFF(IH+1) = COEF(IH)
    ICOL(IH+1) = ICOL(IH)
 69 CONTINUE
 75 CONTINUE
    SIGN(IF+1)=AEWSIGN(ID)
    COFF(IF+1)=AEWCOFF(ID)
    ICOL(IF+1)=AEWCOL(ID)

*** UPDATE THE NO. OF COFF. OF CONSTRAINT.
    NOCO(IE) = NOCO(IE)+1
 70 CONTINUE

*** WRITE OUT THE *NEW* CONSTRAINTS TO ITAPE2.
    WRITE (ITAPE2,201) (SIGN(IH),COEF(IH),ICOL(IH),IH=1,NOCO(IF))
    WRITE (ITAPE2,202) TTT1,TTT2,TTT3
 80 CONTINUE

*** REWIND THE PERMENNAT FILE (ITAF2)
    REWIND ITAPE2

*** FORMATS.
01  FORMAT (5(A1,F16.5,I4))
02  FORMAT (A10,A1,F16.5)
03  FORMAT (I3,11A7/(3X,11A7))
06  FORMAT (5(I4,A1,F11.2))
11  FORMAT (I5)
12  FORMAT (4(2I3,A1,F13.2))
    RETURN
    END
```