January 1976

User Oriented Systems Analysis for Regional Municipal Water Supply Planning

Paul E. Pugner

Trevor C. Hughes

USER ORIENTED SYSTEMS ANALYSIS FOR REGIONAL MUNICIPAL WATER SUPPLY PLANNING

Paul E. Pugner

Trevor C. Hughes

OPTIMAL WATER PLANNING SERIES

PRWA23-1

Utah Water Research Laboratory
College of Engineering
Utah State University
Logan, Utah 84322

ly 1976

USER ORIENTED SYSTEMS ANALYSIS FOR REGIONAL MUNICIPAL WATER SUPPLY PLANNING

by

Paul E. Pugner and Trevor C. Hughes

Utah Water Research Laboratory
College of Engineering
Utah State University
Logan, Utah 84322
July 1976

PRWA23-1

ABSTRACT

An interactive data and model generator is developed that is intended to bridge the gap between planning engineers and the mathematical programming systems approach to municipal and regional water supply planning.

The optimization objective is to minimize total annual cost of existing and future alternative source-related water supply facilities with respect to capital investment and operation and maintenance costs. A matrix generator is developed which formulates the necessary hydrologic, demographic and stochastic municipal water supply data into the format of a mixed integer linear programming problem for system optimization. The program then calls the integer programming algorithm, solves the optimization problem, and outputs a report in a format and language designed specifically for the problem at hand. All of this is accomplished in interactive mode with the user simply answering questions which are asked by the program.

ACKNOWLEDGMENTS

This is a report on work which was supported by the State of Utah (UWRL project WA23) and supplemented by funds provided by the Department of the Interior, Office of Water Research and Technology, under P.L. 88-379, Project Number B-125-UTAH. The work reported herein represents, with some minor additions and minor modifications, the M.S. Thesis of Paul E. Pugner.

The authors wish to acknowledge the valuable assistance of the following persons: Mr. Jesse Grodnick, Burroughs Corporation, and Mr. Karl A. Fugal and Michael R. Stephenson, USU Computer Center.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Planning for use of multiple water sources for municipal water supply is becoming an increasingly more difficult task for the engineer and planner. With the growing complexity of the systems, regional and community interconnections, and the rapidly increasing costs of materials, labor, and power, the engineer and planner need to use a systems or operations research approach to regional water supply planning in order to select the best combination of source related water supply facilities for their municipality or region.

This report is the second resulting from a program at Utah Water Research Laboratory investigating the capabilities of integer programming in water resource planning applications. A previous Utah Water Research Laboratory (UWRL) publication (Hughes et al., 1976) reported on a project in which two types of problems, a regional water supply planning model and a river basin management model, were optimized by using several different integer programming algorithms. One conclusion was that for these types of models, MXINT, the integer programming (IP) algorithm included in the TEMPO mathematical programming package on the Burroughs B6700 system, has computational efficiency and flexibility which is at least equal to any of the other IP codes which were evaluated.

The computational experience reported in the earlier publication verified that IP is a viable tool for optimizing large regional water resource problems. The advantages of the discrete decision-making capability of IP as compared to LP will be discussed later.

One of the difficulties in applying IP (or any other optimizing procedure) to large problems is that manual construction of the mathematical model is time consuming. Hundreds of variables need to be defined and given individual names for input to the computer. Also, hundreds of constraints are required to be constructed and named, and constants such as right hand sides must be manually calculated. This procedure not only requires much manual effort but also provides considerable opportunity for errors in the input data. A principal objective of the work reported herein was the development of a procedure which totally eliminates the need for manual construction of the IP model.

A very comprehensive discussion of the integer programming algorithms, a state of the art survey, was written by Geoffrion and Marsten (1972). An updated version of this original work explores "recent practical advances in integer linear programming" (Geoffrion, 1975, p. 1). Geoffrion (1975) also discusses improvements in the simplex algorithms which are used by most mixed integer programming (MIP) algorithms as their optimizing tool. The literature reveals that in recent years some research has been done to apply these algorithms to water resource planning problems (Hughes, 1973; Bishop et al., 1974; Bishop et al., 1975; Hughes et al., 1976; and Klemetson and Grenney, 1975). However, the gap between research and actual field applications is wide due mainly to the amount of time required to understand the model applications and the lack of user oriented input interface programs.

This report describes an interactive data and model generator that is intended to bridge the gap between the water resource engineer and planner and the mathematical programming systems approach to optimization of water resources for municipal and regional use. The procedure should allow the planner who is not skilled in mathematical programming to obtain optimal solutions to his planning problems with much less time and cost than is now required by a skilled programmer who must manually construct the model. The matrix generator will formulate municipal water supply data in the necessary format to run as a MIP model for system optimization.

Data input is divided into nine segments as follows: 1) regional zone number and name definition, 2) seasonal information, zonal populations, and demand requirements, 3) existing wells, 4) existing springs, 5) existing treatment plants, 6) existing and proposed interzonal connections, 7) proposed wells, 8) proposed springs, and 9) proposed treatment plants. One to four seasonal divisions and peak day supply and demand functions are available. Segments for proposed facilities allow multiple alternatives to be considered for each zone defined.

The program was written to guide the user through each segment and aid in eliminating errors. Default values for many of the input data have been included. For example, capital costs of proposed wells are calculated for the user as a function of well capacity. Pipe costs, power costs for pumping, operation and maintenance, and other dollar figures may be input by the user or left to default. The default values easily can be updated as changes in the economy dictate.

Upon completion of this data input phase of model definition, the user then executes the mathematical programming package GAMMA-TEMPO available through the Utah State University (USU) Computer Center on the Burroughs B 6700 computer system to obtain the optimal solution of the plan alternatives. Output consists of the activity levels of all variables and sensitivity analysis for objective function cost values and constraint values for the non-zero activities.

An added feature of the matrix generator is that if no proposed facilities are input (implying an existing system), a unique form of the MIP problem is formed. Since no integer variables are present, the model becomes a strict LP problem which is efficient to run. The optimization then will be for the operation of the existing system.

Figure 1 gives an overall picture of the main functional operations and data flow that are used in this system. The first step is to input the hydrologic, demographic, and stochastic data, and to make any revisions necessary to the MODELDATA file. Then the TEMPO mathematical programming system is executed to input the model data, generate the model and report and input the problem. The LP and MIP solutions are obtained next, followed by the sensitivity analysis of RANGE. Data flow to and from disk files and to the system printer at each stage also is indicated.

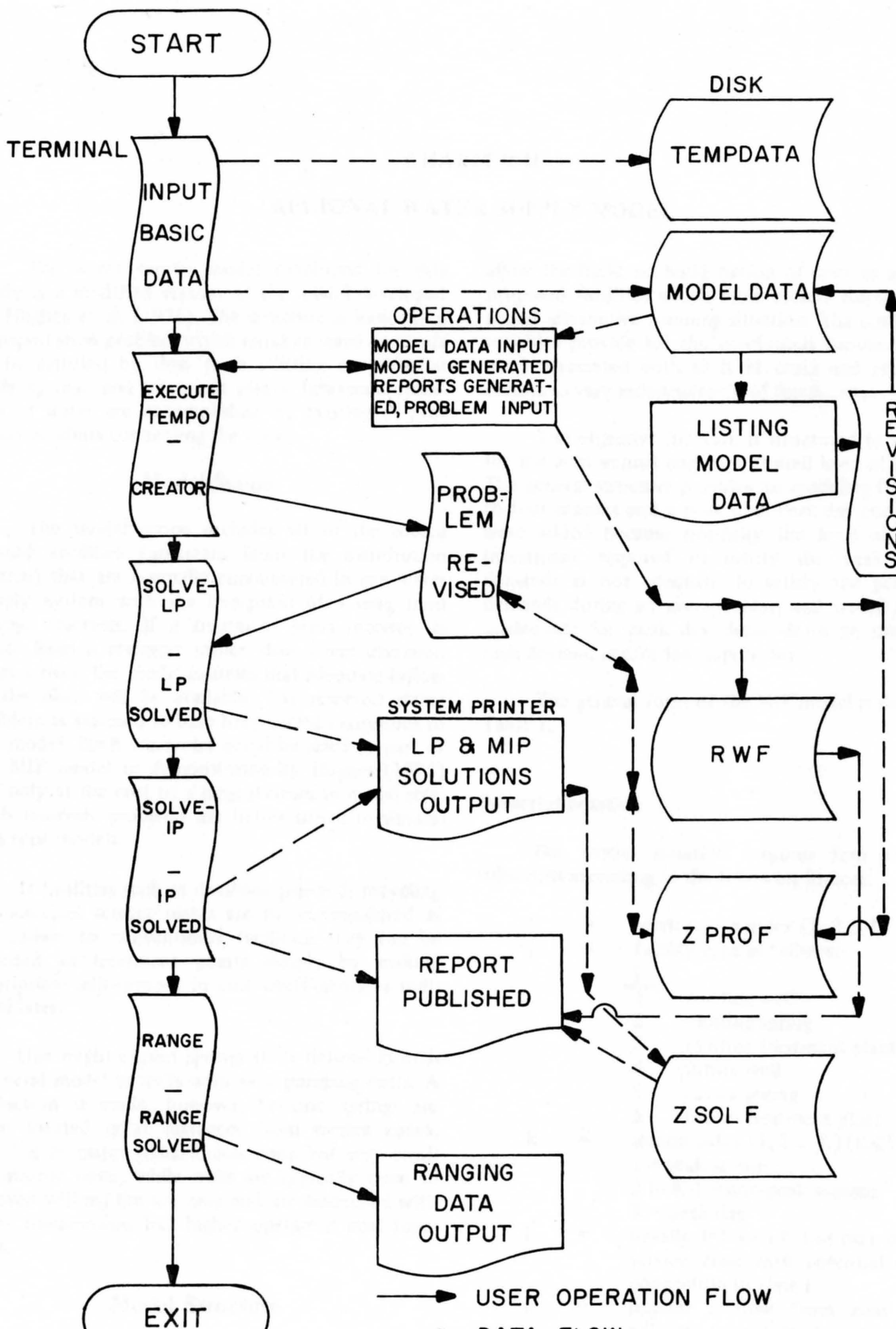

Figure 1. Main functional user operations and data flow.

CHAPTER II

REGIONAL WATER SUPPLY MODEL

The water supply model developed for this study is a modified version of the model developed by Hughes et al. (1976). The structure is basically a transportation problem which requires zonal demands to be satisfied by flow from existing or potential wells, springs, and treatment plants. Interzonal transfers of water are accomplished by existing or proposed conduits connecting the zones.

Model Scope

The model scope includes all of the source related facilities (upstream from the distribution system) that are normally encountered in any water supply system with the exception of a long term storage reservoir. If a treatment plant receives its water from a reservoir rather than direct diversion from a river, the model assumes that adequate inflow to the plant will be available. The reservoir sizing problem is assumed to have been solved exogenous to the model. Such reservoirs could be sized as part of the MIP model as demonstrated by Hughes (1972) but only at the cost of a large increase in model size. Such reservoir problems are better suited to simulation type models.

If facilities such as desalting plants or recycling of municipal sewage plants are to be considered as alternatives to conventional facilities, they can be included as treatment plants simply by making appropriate adjustments in cost coefficients as indicated later.

One might expect springs to be defined in such a general model as wells with zero pumping costs. A distinction is made, however, because springs are often located great distances from service zones, resulting in major transmission costs but very small operational costs, while wells are typically close to (or even within) the use area and are associated with lower transmission but higher operation cost functions.

Model Structure

A MIP model structure was chosen to enable the separation of capital investment costs (integer variables) and operation and maintenance (O & M) costs (continuous variables). The integer variables allow the build/no build option of discrete sizes for proposed facilities which more closely duplicate the actual alternative planning situation. The continuous variables provide for the continuous function (cost/unit) associated with O & M costs and allow use factors to vary independently of fixed costs.

The objective function is structured to provide for the least annual cost at a desired level of service. The general structure provides for modeling from one to four seasons and a peak day. Peak day constraints were added because normally the level of capital investment required to satisfy the peak season demands is not adequate to satisfy the peak day demands during an average year, and would also be inadequate for peak day demands in an unusually high demand and/or low supply day.

The general form of the MIP model is shown in Table 1.

Subscript notation

The model notation requires four variable subscripts according to the following indices:

i	=	service zone index (1, 2 ... I)
j	=	facility type as follows:
		j
		1 existing well
		2 existing spring
		3 existing treatment plant
		4 future well
		5 future spring
		6 future treatment plant
k	=	season index (1, 2 ... K) (K≤5)
		1 = peak season
		2 to K-1 = non-peak seasons
		K = peak day
i'	=	usually i+1 or i-1, but may be any service zone with potential direct connection to zone i
ii'	=	implies a flow from zone i to adjacent zone i' (and conversely $i'i$ represents flow into zone i)
m	=	alternate size of facilities (1, 2 ... M) M≤10 for existing facilities and M≤4 for proposed facilities.

Table 1. General form mixed integer programming (MIP) water supply model.

Objective Function Minimize Total Annual Cost	$\sum_{i}\sum_{j}\hat{c}_{ijm} I_{ijm} + \sum_{i,i'}\sum_{j}\hat{c}_{ii'm} I_{ii'm} + \sum_{i}\sum_{j}\sum_{k} c_{ijkm} x_{ijkm} + \sum_{ii'}\sum_{k} c_{ii'jkm} Z_{ii'jkm}$	
Demands[a]	$\sum_{j} x_{ijk} + \sum_{i'} (Z_{i'ik} - Z_{ii'k}) \geqslant d_{ik}$	(i = 1, 2, ...I) (j = 1, 2, ...6) (k = 1, 2, ...K)
Existing Supply Sources Wells Springs Treatment plants Zonal transfers	 $x_{i1km} \leqslant b_{i1km}$ $x_{i2k} = b_{i2k}$ $x_{i3km} \leqslant b_{i3km}$ $Z_{ii'k} \leqslant b_{ik}$ $Z_{i'ik} \leqslant b_{ik}$	(i = 1, 2, ...I) (k = 1, 2, ...K) (m = 0, 1, ...M)
Proposed Supply Sources Wells Springs Treatment plants Zonal transfers	 $x_{i4km}\ a_{i4km} I_{i4m} \leqslant 0$ $x_{i5km} - a_{i5km} I_{i5m} = 0$ $x_{i6km} - a_{i6km} I_{i6m} \leqslant 0$ $Z_{ii'km} - a_{ii'km} I_{ii'km} \leqslant 0$ $Z_{i'ikm} - a_{i'ikm} I_{i'im} \leqslant 0$	(i = 1, 2, ...I) (k = 1, 2, ...K) (m = 0, 1, ...M)

[a]This is a generalized form of notation in which the peak day is defined as one of the seasons. See subscript notation for index k.

Variable description

A description of the variables used in Table 1 is as follows:

Integer variables

I_{ijm} = integer variable denoting development of a new well, spring, or treatment plant j in zone i alternate size m. Activity level indicates the number of new facilities built. Values are 0 or 1 for springs and treatment plants, but higher integer values are possible for wells.

$I_{ii'm}$ = 0 or 1 variable denoting construction of a particular size m of pipeline between zone i and adjacent zone i'.

Continuous variables

x_{ijkm} = flow in millions of gallons (MG) from either existing or proposed facilities j (wells, springs, or treatment plants) in zone i during season k for alternate size m.

$Z_{ii'km}$ = flow in MG from zone i to i' during season k for alternate size m.

Technical coefficients for integer variables

a_{ijkm} = average capacity of source facility I_{ijm} during season k for alternate size m.

$a_{ii'km}$ = capacity of pipeline $I_{ii'm}$ during season k for alternate size m.

Note: All a_{ijkm} and $a_{ii'km}$ continuous variables are either zero or unity and therefore are not shown explicitly in constraints.

Cost coefficients

$\hat{c}_{ijm}$ and $\hat{c}_{ii'm}$ = annual loan service cost (fixed cost) in dollars for constructing facilities I_{ijm} and $I_{ii'm}$ respectively.

c_{ijkm} and $c_{ii'km}$ = unit cost (variable cost) in \$/MG of operating facilities x_{ijkm} and $Z_{ii'km}$ respectively.

Right hand side

d_{ik} = average demand for zone i during season k.

b_{ijkm} = average capacity of existing source facility j in zone i during season k for alternate size m.

Note: Capacities of new facilities are represented by "A" matrix coefficients as described previously rather than being in the RHS vector.

Computer variable names

In order to provide variable names which are easily recognizable and which are acceptable to the naming conventions of the algorithms used, the formal notation used previously in defining the model was modified. An X prefix indicates a seasonal continuous variable, a P prefix indicates a peak day continuous variable, and an I prefix indicates an integer variable. For example:

1. The j index was represented by the following descriptive letters rather than numbers:

j	Computer Notation
1	W (existing wells)
2	S (existing springs)
3	TP (existing treatment plants)
4	FW (proposed wells)
5	FS (proposed springs)
6	FTP (proposed treatment plants)

2. The m index (alternate sizes) was represented by the following descriptive letters rather than numbers:

This notation would lead to the following examples for variable names:

XW22B2 = continuous variable for the supply from an existing well in zone 22 of size B (the second entered) for season 2

XFS14A3 = continuous variable for the supply from a proposed spring in zone 14 of size A (the first entered) for season 3

IFW22A = integer variable for a proposed well in zone 22 of size A

IZT1422K = integer variable for proposed zonal transfer conduit from zone 14 to zone 22 of size K (indicating a 36 inch diameter conduit)

PEW22A = continuous variable for the peak day supply from an existing well in zone 22 of size A

PFW22A = continuous variable for the peak day supply for a proposed well in zone 22 of size A

m	Computer Notation		
1	A	Order of entry in data input phase for wells, springs, and treatment plants.	Increasing pipe size for zonal transfers
2	B		
3	C		
4	D		
5	E		
6	F		
7	G		
8	H		
9	I		
10	J		
11	K		
	X	Existing pipe for zonal transfer	

CHAPTER III

INTERACTIVE INPUT DATA GENERATOR

WITH SAMPLE PROBLEM

The interactive input data generator was devised to be an easy means for the planner who is not familiar with mathematical programming to formulate the necessary hydrologic, demographic and stochastic data into the required format for input to a mathematical model without having to know the actual form of the model.

Sequence of Segments

Data input is divided into nine segments: 1) regional zone number and name definition, 2) season information, zonal populations, and demand requirements, 3) existing wells, 4) existing springs, 5) existing treatment plants, 6) existing and proposed interzonal connections, 7) proposed wells, 8) proposed springs, and 9) proposed treatment plants.

Input Data Structure

The zone definitions identify separate municipal water systems in a regional problem, or separate service areas within a single municipality which have significantly different water delivery unit costs due to elevation differences, or combinations of both.

The structure is such that one to four seasonal divisions may be input depending on the required detail of the analysis. For large problems of 20 or more zones, it might be advisable to limit the model to two seasons. With four seasons the LP portion of the model will become quite large and even though the LP portion is not the limiting factor (as noted later), it can cause an increase in run times and costs.

Season one is always considered as the peak season for computational purposes, and the peak day constraints will be based on this season. Peak season is defined as the season with the largest per capita demand. Appendix A contains a complete sample run of the interactive data input phase for a sample problem, and Appendix B contains a complete listing of all the programs and subroutines.

Sample Problem Description

The sample problem used in this study is hypothetical but was chosen to include at least one facility type or condition to indicate to the user the type of data required, the method and sequence of input, and the output data used by the model. The problem consists of five service zones located as shown in Figure 2. Interzonal connections with probable directions of flow are indicated as well. Zone 01 is intended to be a small community situated in the foothills. Zone 14 is a large metropolitan area at the base of the foothills. Zone 32 is a suburb community located several miles from the metropolitan area. Zone 22 is a farming community located in the valley floor, and zone 56 is a small service community. Worksheets useful in preparing the required input data are shown in Appendix C together with the appropriate data for the sample problem.

Appendix A is the remote terminal interactive record for the sample problem.

Program Execution

The program is accessed by the command EXECUTE (480030)MATRIX. The program title and introductory information is then given. The question DO YOU WANT INSTRUCTIONS? 〈YES/NO〉 will always require the first user input response. A YES response will give the general sequence of segments and the required input data.

After instructions are listed or an answer NO is given to the instructions question, IS THIS A RESTART OF ANOTHER RUN? 〈YES/NO〉 will be asked. The implication of the answer NO is that this is a new model being input and the answer YES implies that a model has been started previously, but all the segments have not been finished. Data input may be stopped after the completion of any segment and restarted at the *next* segment without the loss of any previous data.

Segment 1 (zone number and name definition)

All zones must be numbered using two digits. If zones are to be numbered 1 to 9, the leading zero

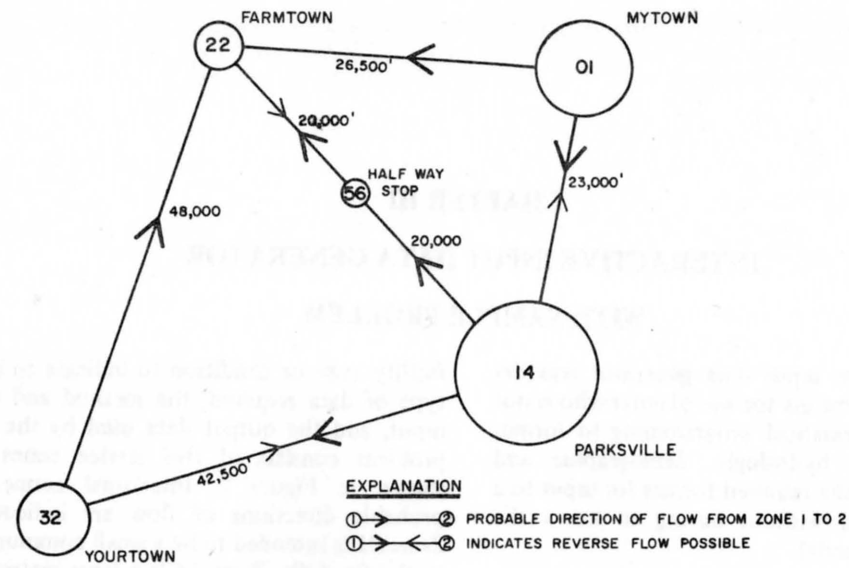

Figure 2. Sample problem service zone locations.

need not be included, since the program will prefix the zone number. Zone numbers of 01 to 98 may be used in any order, but the program will not reorder the input sequence throughout the remainder of the segments, so that continuity of input order will be maintained. The zone name may consist of any sequence of characters on the key board up to a maximum of 18. Multiple spaces are allowed between the zone number and name, but single spaces only within the name. The total maximum number of zone numbers, spaces, and characters in zone name is 30. For example:

Acceptable Zone Designations:

1 UPPER LOGAN
14PARKSVILLE
  02   HALF WAY STOP

Unacceptable Zone Designations:

142 UPPER LOGAN
  zone number exceeds 98
22 LOWER   LOGAN
  will be truncated to 22 LOWER because of multiple spaces in name
39 THIS NAME IS TOO LONG
  will be truncated to 39 THIS NAME IS TOO L because the name exceeds 18 characters
45   THIS USES TOO MANY
  will be truncated to 45 THIS USES TOO M because the 30 character space limit was exceeded

Zone numbers and names should not be repeated as this will cause errors in the model generation at later stages. A zone number of 99 should be entered when all desired zones have been input. The maximum number of zones is 40.

The zone numbers and names will next be printed out for user examination and corrections. To correct a zone name or number, input the row of the change when asked and the corrected zone number and name when asked. If a zone has been omitted, enter a row number greater than any listed and the new zone when asked. All revisions will be printed back for inspection. The data can again be listed in entirety if desired.

After completion of each segment, the question DO YOU WANT TO CONTINUE? ⟨YES/NO⟩ will be asked. A NO answer will lock all the required data files and terminate execution of the program. A restart may then be initiated at some later date. If, however, the restart and completion to solution of the entire model is not to be completed the same day, special provisions must be made with the USU computer center to have data files saved for the duration of the time required to reach completion.

Segment 2 (seasonal information, zone populations, and demand requirements)

The number of seasons in the model, the number of days in each season, and the term of the season are input first. Season 1 must be the peak season in the model. The term information is required for clarity in the solution report and may consist of any sequence of 15 or less characters and/or spaces except two consecutive spaces. The term is normally a month abbreviation and a day for the start and finish of the season. For example: May 1 - July 31. The season, day, and term data will be output for inspection and correction if required.

The population, reservoir elevation in feet from mean sea level (MSL), and demand per person per day for each season in gallons per day (GPD) will next be input for each zone. These input data will be output for inspection and correction if required.

The standard peak day demand multiplier (stochastic portion of model) is output, and, if acceptable, a YES response is input. If the standard value is unacceptable, a response of NO will allow a constant for all zones to be input or if no constant for all zones is acceptable, each zone's peak day multiplier must be entered. Acceptable peak day demand multipliers are greater than or equal to one.

The remaining segments (3-9) do not necessarily require execution, and prior to the possible execution of each, the program will ask if any facilities of this type are included in the model. An answer of NO will cause that segment to be skipped and the next to be questioned for execution.

Segment 3 (existing well information)

A special request to group all existing wells in a zone, if possible, is made to assist in representing the model data in the simplest form possible. The fewer the number of facilities, the easier it is to solve the problem. Examples of wells that could be combined would be those that 1) pump from the same elevation (well water surface level), 2) have approximately the same operating and maintenance (O & M) and pumping costs, and 3) are not in some other way so different that they cannot be combined. If, however, it is necessary to consider the wells in a zone separately, provision is made in the program to have up to ten separate well representations in each zone, and a maximum of 90 for all zones combined.

Each zone name will be output followed by a request for the number of wells or groups of wells in that zone. A zero response will cause continuation to the next zone. A non-zero response will cause requests for well elevation (well water surface level) and well capacity in GPM for each of the number of wells in that zone. The wells will be alpha named rather than numbered in the order of input. The first "A," second "B," etc., up to the maximum of ten or "J."

The standard peak day supply constant multiplier will be output next for acceptance or rejection.

The standard O & M well costs ($/MG) will be output next for acceptance or rejection. The O & M costs include all costs except power charges for pumping. Standard power costs for pumping ($/MG/100FT.) will be output next for acceptance or rejection. The input data will be output and any required changes made at this time.

Segment 4 (existing spring information)

For model simplicity all existing springs in a zone *must* be input as one source. Since spring supply is usually the least unit cost source and the first to be used to supply demand, this requirement should not change the model solution. For each zone, spring supply in CFS for each season is input following the output of the zone identity. Peak day supply constant multiplier and O & M costs are input next. Output of data, required changes, and output of calculated data follow.

Segment 5 (existing treatment plant information)

Again as in segment 3 for model simplicity, all existing treatment plants in a zone are grouped as one source if possible. However, if this is not possible, 10 treatment plants per zone with a total maximum for the model of 80 may be input.

Peak day supply constant multiplier is input next. Then for each zone, the number of existing treatment plants, the plant capacity (MGD), and O & M costs ($/MG) for each season are input. Output of data and corrections follow.

Segment 6 (existing and proposed interzonal connections)

The following four segments (6-9) are for input of new facilities. The user should be aware that each new facility input creates one integer variable (column), several continuous variables (columns), and several rows in the model matrix structure. The number of integer variables in a model will determine the overall difficulty in finding the optimal solution and increase the cost of finding that solution accordingly. Therefore, it is strongly suggested that the number of proposed alternate facilities be kept to the minimum that are in reality feasible alternatives.

A maximum of 80 interconnections (existing and proposed) and a maximum total of 120 alternate size options may be defined in the model.

A calculating formula for the total capital cost of zonal transfer facilities is used based on pipe cost per foot and an installation factor based on the difficulty of installation as shown in the sample run. Standard costs and factors are output for acceptance or rejection. If any of the constants are not acceptable, all must be input even though some may not change. If the total capital cost of the facility is known, this may be entered after not accepting the standard value or entering required user input constants. The capital cost will be requested at a later time.

The capital recovery factor constants of interest rate (R) and amortization period in years (N) is then required either by default or by user input. Each zone number and name will then be output followed by a request for the zone number it is connected to, or for which there is a proposed connection, or both. An important note on the way the program selects the sizes of alternate proposed connections is that the probable direction of flow is implied by the way connections are input. The demand for the zone number given in the answer to the question CONNECTED TO ZONE ?? will be used in calculating the pipe diameter required to carry the majority of the total demand for that zone.

For example, in the first request for input for zone MYTOWN, the response was that zone 01 was connected to zone 14. This implies that the most probable direction of flow will be from zone 01 to 14, and the two alternate pipe sizes later requested will be based on the majority of the total demand of zone 14 being supplied by the alternate pipes. A pipe diameter that will carry the demand based on a velocity of 6 fps is picked. The next smaller diameter is then used for the starting size of alternates and each successively larger size input until the proper number of alternatives is chosen. For example, if zone 14 demand could be supplied by a 36 inch diameter pipe and two alternates were required, the model would pick alternate "J" equal to 30 inch diameter and alternate "K" as 36 inch diameter. If the user wishes to have alternate sizes different than those the program will pick, there are two alternatives available. The first, if smaller sizes are required and zone 01 has a smaller demand, is to wait until zone 14's connection pattern is requested and then input zone 01. This would require reverse flow if the actual direction will be from 01 to 14. The second is to input the zones as the most probable direction of flow and then make revisions to the calculated data output for review.

Output will consist of a summary of the input data followed by a list of calculated data. Revisions can be made in both lists.

Segment 7 (proposed future wells)

Four alternate size future wells per zone and a maximum total of 120 for the model are allowed. The calculating formula for total capital cost of proposed wells is based on an exponential formula as shown in the sample run. The user can input other constants or reject all constants and input known total capital costs. Capital recovery factor values are also required in this segment by default or user input.

Each zone number and name will be output followed by the request for the number of optional wells required. A zero response will cause output of the next zone. If several wells of the same capacity, pumping head, capital cost, and other factors are proposed for a zone, they may be indicated by inputing as one size with several possible wells of that particular size as shown for 01 MYTOWN. Even though only one integer variable is created for each well size, the algorithm handles the larger integer bounds (more than one per size) as if there were an integer variable for each possible. So three wells of size "A" in a zone is comparable to three integer variables. The user should always minimize the number of integer variables when possible.

Output consists of input data and calculated capital costs followed by peak day supplies and water costs. Revisions can be made to either list. The possibility of a long transmission line from a well was not provided for in the capital cost or in the O & M costs. If unusual situations occur, revisions to the output data can be made to account for these.

The program was terminated after segment 7 to show how to stop and restart the program if desired. The restart is the same as the original start; however, one would not ask for instructions and a restart would be indicated with the *next* segment in the sequence as the starting point.

Segment 8 (proposed future springs)

Four alternate size proposed springs per zone and a maximum total of 80 for the model are allowed. If more than one alternate size conduit is required for a proposed spring, then each alternate conduit will have to be defined as a separate spring since only one conduit per spring is allowed. This will require a change in the model structure (reference CHAPTER IV MODEL GENERATION AND SOLUTION) since with the present structure all the alternate springs may be developed even though only one spring really exists.

To change the model structure, the user would add a constraint that limits the development of these special alternate springs to only one per zone. For example, if zone 22 had a large spring and the user wanted to review the possibility of having three different conduits connected to this spring, three springs would be defined, each with a different flow capacity (based on conduit size). A constraint would be added in the model revision section of the form:

$$IFS22A + IFS22B + IFS22C \leqslant 1.$$

A small spring supplying large demands should not use this modification since the total flow would be developed and only one conduit size would be defined. The modifications should only be used where a large spring may supply a small demand and partial development may be the least cost alternative.

The calculating formula for total capital cost of proposed springs is the same as for proposed inter-zonal transfers except for the addition of a user supplied on site spring development cost as shown in the sample run. Capital recovery factor values are also required by default or user input.

Input data required are shown in the sample run. Output consists of input and calculated data as shown in the sample run followed by a request for the peak day supply constant multiplier (if applicable) and output of the peak day supply and O & M cost data. All listed data can be revised.

Segment 9 (proposed future treatment plants)

Four alternate size proposed treatment plants per zone and a maximum total of 80 for the model are allowed. The standard peak day supply constant multiplier and capital recovery factor values are input first.

Each zone number and name will be output followed by a request for the number of proposed treatment plants for this zone. The plant capacities and O & M costs for each season are input as shown followed by the total capital cost for the treatment plant. Each zone in succession will be output followed by the request for necessary input data.

Output consists of input data as shown and as usual revisions can be made at this time.

Summary Suggestions

After completion of segment 9 the program will give some suggestions for running the model based on the total number of integer variables created. These are only suggestions and need not be followed explicitly if the user is familiar with the algorithm used for model solution. This concludes the data input phase.

Prior to execution of the next phase (model generation), the user should obtain a listing of the data the model generator will use. This is accomplished by listing the file MODELDATA (created by the interactive input data generator) at the user's terminal or at the system printer. The output for the sample problem is shown at the end of Appendix A. The system printer is recommended as the listing can become quite long. Corrections or revisions to data can also be made at this time by using the system editing feature CANDE (Burrough's, 1972).

CHAPTER IV

MODEL GENERATION AND SOLUTION

GAMMA Description

The mathematical model for the problem must be generated in a specific format as required by the algorithm used. This is accomplished by use of GAMMA, a general system matrix generator and report writer for use with the TEMPO mathematical programming system. Appendix D contains a listing of the data, matrix, and report sections of GAMMA used to input the required data from the file MODELDATA, generate the matrix structure, and publish the solution report. The user need not become familiar with the GAMMA code unless changes to the model structure or report format are required.

Matrix generation

GAMMA's usefulness in solving mathematical programming models is due primarily to the following two concepts:

a. Tables of numerical data and lists of character data are input and maintained separately from the structure of the matrix (variables and constraints). Tabular data may, therefore, be updated periodically without changing the structural definition of variables in the model. This simplifies obtaining solutions when costs, prices, requirements, etcetera change. Also errors are less likely since data can be scanned in the table, making errors more apparent.

b. GAMMA constructs have an implied iteration on all elements of a list when the list name appears in a statement. This permits many rows, columns, coefficients, etcetera, to be generated with only a few statements. Thus, both the input required to define a model, and the possibility of error, are greatly reduced. (Burroughs GAMMA Manual, November, 1975, p. xi.)

The GAMMA listing shown in Appendix D is general and will always generate the proper model matrix for the input data from the interactive data generator for any given combination or number of facilities. A few brief examples from the listing in Appendix D will further explain this statement. Line 110000 is used to create *all* the seasonal zone demand constraints no matter the number of zones or seasons. D(ZO)(S),G states the row name and starts with the letter D (demand), and rows will be created by the replication on lists (ZO) and (S). If list (ZO) contained the elements 10, 22, and 56, and list (S) contained the elements 1 and 2, the matrix rows (constraints) shown below would be created, each as being greater than or equal to inequality (,G).

The associated line 110100 is for the demand row right hand side (RHS) definition. The name of the RHS vector is named first (RHS1) followed by a RHS definition statement which will retrieve from the table DEMAND, row (ZO) and column DEMAND(S) the proper seasonal demand to insert for the proper matrix row. Again, list replication will allow this simple statement to obtain from previously defined tables and lists values of all the necessary data required for the demand RHS vector. For example:

List (ZO)
01
22
56
List (S)
1
2
TABLE DEMAND

*	DEMAND1	DEMAND2
01	500	300
22	2400	1800
56	950	625

would generate the following row constraints

ROW NAMES		RHS
D011	⩾	500
D012	⩾	300
D221	⩾	2400
D222	⩾	1800
D561	⩾	950
D562	⩾	625

All the proposed well integer variables (line 118300), the objective function values (line 118400), the continuous variables for flow (line 118500), the peak day continuous variables for flow (line 118600), and an upper bound for the number of proposed wells of this size (line 118700) would be generated from these few statements no matter the actual number of facilities (0 to number in the list (FWL)).

Similar constructs are used to generate the model from the user's data supplied in the data input phase and standard lists and tables shown in lines 100000-108400.

Report description

A report of the problem solution in a format which is easier to read than the standard solution output is generated in a similar manner. Each print line is specified depending on the solution to the problem, and output consists of the activity levels of the non-zero variables. The report output for the sample problem is shown in Appendix E.

The solution output consists of first a summary of the construction of proposed facilities by facility type. Total zonal added capacities for each season and peak day are given in addition to the number of facilities and annual capital costs. A zone by zone analysis of the source of supply and associated costs to meet the required seasonal and peak day demand is then output. The import/export supply source can become confusing because the order of zone names was originally determined by the most probable direction of flow. The easiest way to keep import/export correct is to recall that the analysis is being done for a specific zone. Import means bringing water into the zone being analyzed from another zone, and export means sending water to another zone from the analyzed zone. For example, in the analysis for the zone PARKSVILLE import is shown PARKSVILLE to YOURTOWN which means water is being imported to PARKSVILLE from YOURTOWN. In the zone YOURTOWN the corresponding export PARKSVILLE to YOURTOWN means water is being exported from YOURTOWN to PARKSVILLE.

TEMPO Execution and Model Generation

The matrix structure and the report structure are generated using the following procedures:

a. E $MPS/SPECIAL: a special interactive version of the TEMPO mathematical programming system is executed which contains all the necessary procedures to solve the water supply planning problem.
b. $FILE MACROLIB = (480030) MACROLIB1: a series of TEMPO control language (TCL) statements specifically designed to accomplish all the necessary steps in problem solution are input to the user's file. A listing of these TCL statements is shown in Appendix F.
c. MACRO CREATER RESTORE: restores the matrix and report generator subset of the above MACROLIB with TCL statements designed to generate the data, matrix, and reports and input the problem in the permanent problem file (ZPROF), and output the model structure, row and column names.
d. CREATOR: initiates the set of TCL statements to accomplish (c) above.

Output from CREATOR will be sent to the system printer and the user's terminal. The system printer will receive all the long printout consisting of the GAMMA input data and model structure statements. The user's terminal will receive only the key statements required to indicate that proper data, model, and report generation have occurred. The model will then be input and saved on the problem file.

Model Solution and Revision

After model input to the problem file, the user may proceed with any of the following three procedures:

a. Stop the run at this point by entering EXIT and continue with steps (b) and/or (c) at some later date. All required data, model structure, and report formats have been saved on the user's files. If the user is not going to complete the model solution in one day, special provisions must be made with the USU Computer Center to have all files saved. When restarting, the user would need to reinput the E $MPS/SPECIAL and $FILE MACROLIB=(480030)MACROLIB1 statements as before.
b. Activate the revised option of TEMPO to change the model structure by adding constraints (rows) which would be in addition to the general model structure. This may be a requirement if, for example, the quality of a source of supply will not meet quality standards and has to be mixed in some ratio with another source. In the sample problem, for example, if the proposed well in zone 32 size "A" contained a high level of total dissolved solids and prior to use should be mixed with the water from the existing spring in zone 32 at a 50 percent ratio, the sequence of input operations and responses shown in Appendix G would be required to change the model structure. The change would consist of

adding the following constraints, which state that the flow from proposed well "A" in zone 32 for each season and peak day must be less than or equal to the flow from the existing spring(s) in zone 32:

XFW32A1 - XS321≤0
XFW32A2 - XS322≤0
XFW32A3 - XS323≤0
XFW32A4 - XS324≤0
PFW32A - PES32≤0

The user could stop the run at this point as in (a) above or continue to (c) below.

c. Continue to the LP solution phase of the problem. An LP solution is required prior to obtaining MIP solutions.

LP Solution Phase

The LP solution phase consists of setting up the model structure in the proper format for an LP solution. This phase is initiated by the following steps:

1. ZNAME = "MODEL NAME":
   The name of the model, MODEL if the standard model structure is used or REVMODEL if revisions are made as in (b) above.
2. MACRO SOLVELP RESTORE:
   restores the LP subset of MACROLIB with TCL statements designed to obtain an LP solution.
3. SOLVELP:
   initiates the set of TCL statements to solve the LP model.

As in the generate phase, the long printout will be sent to the system printer and only key statements to indicate proper program flow are output at the user's terminal. After the optimal LP solution is found and the system indicates a SAVE and OUTPUT have occurred, it will respond with READY.

An added feature of the matrix generator is that if no proposed facilities are required in the problem (implying an existing system), a unique form of the MIP problem is formed. Since no integer variables are present, the model becomes a strict LP problem. The optimization then is for the operation of an existing system. The procedure for this special case is to input the following TCL statements which will publish the optimal LP solution:

ZREPNM="MODEL"
PUBLISH

This step concludes the model run unless revisions are to be made. If, however, there are proposed facilities to be considered, the problem is an MIP problem, and it is now ready to be solved by MIP procedures.

MIP Solution Phase

The MIP solution phase consists of solving the problem using the Branch and Bound mathematical technique to obtain integer solutions from an LP starting solution. The sequence of TCL statements used for this procedure have incorporated a unique feature of TEMPO. Most problems have many suboptimal MIP solutions. To allow an exhaustive search of all suboptimal solutions requires considerably more run time and cost than finding only a few suboptimal and an optimal MIP solution. The feature allows for a CUTOFF value (projected integer solutions with an objective function value greater than CUTOFF are discarded) to be input so that many of the suboptimal solutions will be discarded. The technique used by this structure is for the user to input the percentage of improvement desired in the objective function of the suboptimal MIP solutions. The new CUTOFF then is based on this percentage and only the MIP solutions with projected objective function values lower than CUTOFF are explored.

After all MIP solutions using CUTOFF have been exhausted, the branch node system below the last MIP solution is reloaded and an exhaustive search for the optimal MIP solution is made. This procedure has been shown (Hughes et al., 1976) to obtain the optimal MIP solution with a considerable savings in run time and cost. The recommended percentage value is from 0.03 to 0.08 for large models (40 or more integer variables) and from 0.00 to 0.05 for small models (less than 40 integer variables). When using the larger values, the user should know from previous problems the range of acceptable values for that particular problem type because TEMPO may not find any MIP solutions with the required improvement and will restart at the beginning with an exhaustive search. A procedure for overcoming this difficulty on large problems will be described later.

The MIP solution phase is initiated by the following steps after the LP solution phase is completed:

1. URPERCNT = decimal value:
   decimal value representing percent improvement in MIP solution objective function desired
2. MACRO SOLVEIP RESTORE:
   restores the MIP subset of MACROLIB with TCL statements designed to obtain an optimal MIP solution
3. SOLVEIP:
   initiates the set of TCL statements to obtain the optimal MIP solution

Output to the system printer will consist of all suboptimal MIP solutions found, the optimal MIP solution, and the published report of the optimal MIP solution. The optimal MIP solution for the sample problem is shown in Appendix H and the associated published report is shown in Appendix E.

The integer variable IFTP14A shown in the columns section of output Appendix H is not an integer value, but equal to 0.99938. This problem may occur as tolerances are built into the TEMPO code to allow integer variables with values within the specified tolerance limit (ZTOLIN) of integer values to be considered as integers. The default value is 0.01 but in the MACRO SOLVEIP, ZTOLIN was reset to 0.001 to allow less "cheating" by TEMPO. Since input data normally are not within this level of accuracy, any error introduced by this problem will be less significant than the data error. The problem becomes more significant if the solution is an activity of 0.00099 because then the cheating has built a very small portion of a facility at a very small cost to meet some small demand requirement. Smaller values of ZTOLIN could be input but computational time to obtain MIP solutions increases dramatically, and a tradeoff is required between computer run cost and numerical accuracy. For report publication purposes, all integer variables are rounded to the nearest integer value prior to use computationally. A detailed description of the meaning of the TEMPO output shown in Appendix H is available in the TEMPO manual, pages 4-26 to 4-29.

Output to the user's terminal will consist of the number of the solution, the active integer variables, OUTPUT and SAVEFIle indicators, and the new cutoff (ZBIOBJ) for each MIP solution found. The final indicator of MIP completion will be an indication of the report being published (---PUBLIS---) followed by a READY.

If the user wants to have suboptimal MIP solutions published, as may be required when reviewing several close alternatives, these solutions are saved on the solution file (ZSOLF). To publish the reports, the following sequence of TCL statements are input following the last READY.

ZSOLNO = integer value:
the number of the MIP solution the user wishes to publish
PUBLISH:
the command to publish the report of the solution number ZSOLNO

The above two statements are repeated for each MIP solution the user wishes to be published.

Sensitivity Analysis

Sensitivity analysis determines the range over which cost coefficients and right hand sides can vary without changing the optimal solution. Both row and column variables are ranged. The user should be aware that ranging for integer variables can sometimes be meaningless if the range of values does not contain integers. All ranging output is sent to the system printer.

Two methods are available to do sensitivity analysis on the optimal MIP solution. The first method of sensitivity analysis is to input the TCL command RANGE following the last READY after the optimal MIP solution is found. This will do a range of all rows and columns in the model. Selected RANGE output for the sample problem is shown in Appendix I. The explanation of the headings of RANGE output as defined by the TEMPO manual is as follows:

a. Ranging data for rows:
   1) The first heading indicates the solution status of the rows:
      a) Number–the internal number of the row.
      b) Row–the name of the row.
      c) Status–a code indicating activity status:
         1. BS–activity at intermediate level.
         2. EQ–activity at fixed level.
         3. UL–activity at upper limit.
         4. LL–activity at lower limit.
      d) Activity–the value of the row activity. Computed as the difference between the right hand side and the slack activity.
      e) Slack activity–the activity of the logical variable for the row.
   2) Two lines of output are printed for the remaining headings. The upper line indicates the activity-cost relationship for activity decrease per cost increase. The lower line indicates the activity-cost relationship for activity increase per cost decrease.
      a) Lower limit–the input lower limit for the row. Specified or implicit.
      b) Upper limit–the input upper limit for the row. Specified or implicit.
      c) Lower activity–the row activity may be decreased to this level at a cost per unit of decrease indicated by unit cost. Decrease below this level has a different unit cost.

d) Upper activity–the row activity may be increased to this level at a cost per unit of increase indicated by unit cost. Increase above this level has a different unit cost.

e) Unit cost–the change in the objective function per unit decrease in the row activity (upper line).

f) Unit cost–the change in the objective function per unit increase in the row activity (lower line).

g) Limiting process–the name of the row or column that would change its status if the activity level of this row were decreased below lower activity. If this row is basic, the limiting process enters the basis; otherwise, the limiting process leaves the basis (upper line).

h) Limiting process–the name of the row or column that would change its status if the activity level of this row were increased above upper activity. If this row is basic, the limiting process enters the basis; otherwise, the limiting process leaves the basis (lower line).

i) Status–the status of the limiting process (upper and lower lines):

1. LL–the limiting row or column leaves or enters the basis at lower limit.
2. UL–the limiting row or column leaves or enters the basis at upper limit.

b. Ranging data for columns:

1) The first heading indicates the solution status of the column:

a) Number–the internal number of the column.

b) Row–the name of the row.

c) Status–a code indicating the activity status:

1. BS–in the basis at intermediate level.
2. EQ–nonbasic at the fixed level.
3. UL–nonbasic at the upper limit.
4. LL–nonbasic at the lower limit.
5. FR–nonbasic, free.

d) Activity–the value of the column activity.

e) Input cost–the unit cost of the variable as specified on input.

2) Two lines of output are printed for each remaining heading. The upper line indicates the activity-cost relationship for activity decrease per cost increase. The lower line indicates the activity-cost relationship for activity increase per cost decrease.

a) Lower limit–the input lower bound of the column.

b) Upper limit–the input upper bound of the column.

c) Lower activity–the activity level that would result from changing the cost coefficient from input cost to upper cost.

d) Upper activity–the activity level that would result from changing the cost coefficient from input cost to lower cost.

e) Unit cost–the change in the objective function per unit decrease in the column activity. Unit cost is not valid for activity decrease below lower activity (upper line).

f) Unit cost–the change in the objective function per unit increase in the column activity. Unit cost is not valid for activity increase above upper activity (lower line).

g) Upper cost–the highest cost coefficient the column can have without changing its activity. If the cost coefficient is increased above upper cost, the activity level would be decreased to lower activity.

h) Lower cost–the lowest cost coefficient the column can have without changing its activity. If the cost coefficient is decreased below lower cost, the activity level would be increased to upper activity.

i) Limiting process–the name of the row or column that would change its status if the activity level of this column were decreased below lower activity. If this column is basic, the limiting process enters the basis; otherwise, the limiting process leaves the basis (upper line).

j) Limiting process–the name of the row or column that would change its status if the activity level of this column were increased above upper activity. If this column is basic, the limiting process enters the basis; otherwise, the limiting process leaves the basis (lower line).

k) Status–the status of the limiting process (upper and lower lines):
   1. LL–the limiting row or column leaves or enters the basis at lower limit.
   2. UL–the limiting row or column leaves or enters the basis at upper limit. (Burroughs TEMPO Manual, 1975, pp. 4.51-4.54)

The second method of sensitivity analysis is to revise the model problem structure to remove all integer variables with zero activity levels and their corresponding continuous variables and rows. The remaining integer variables are fixed at their present integer activity levels (or rounded to integer values) and the problem is restructured as an all LP problem. The problem is resolved in this form, with the solution being identical to the optimal MIP solution (unless non-integer values existed in the MIP solution). Ranging is then accomplished on an LP solution with integer variables fixed as constants rather than an MIP solution with integer variables. This is accomplished by invoking the macro RANGES.

The advantages of this method are 1) the range is for an LP solution which is more representative of the actual ranges. However, where only a few proposed facilities are constructed, the range is very limited in its analysis since these facilities will be the limiting factors (for example, if wells but no springs or treatment plants are built in the optimal solution, then the O & M cost of the wells can range to infinity since no other facilities are available to enter the solution), and 2) there are fewer rows and columns to range over since many have been deleted, thus simplifying the ranging.

The main disadvantage with the LP range is that no information on the sensitivity of the deleted rows and columns is available as in the first method. Costs or RHS values of these deleted rows and columns may be very sensitive and would have entered the solution if a slight change was made. No recommendations are made as to the method of sensitivity analysis since it is necessary for the user to determine the type of information needed and the most applicable method after viewing the solution.

Batch TEMPO Execution

All of the above procedures may be run by batch (cards) rather than interactive (terminal) with a few modifications.

1. The E $MPS/SPECIAL execution statement is changed to E MPS/ SPECIAL.
2. The $FILE MACROLIB = (480030) MACROLIB1 statement is deleted because the statements as listed would cause premature termination of TEMPO (reference Appendix F). All statements, except MACRO, END MACRO, EXIT, $FILE INPUT, ZCONSOLE ZPRINTER, are punched in standard TEMPO format. Additionally, URPERCNT = decimal value (improvement in MIP solutions) is included prior to line 420. If method one ranging is used, RANGE is inserted between lines 730 and 740 (line 740 EXIT is not deleted) and lines 810 to 990 are not used. If method two ranging is used, lines 810 to 990 are between lines 730 and 740, with the before mentioned statements deleted (MACRO, END MACRO, etc.).
3. The REVISE(REMOTE) statement is changed to REVISE with the revision data being input by cards punched in standard TEMPO format (reference TEMPO manual, Chapter V).

Large Problem Modifications

For very large problems of more than 50 integer variables, it is suggested that the macro SOLVEIP not be used but that the user interactively input the following TCL statements after the completion of the macro SOLVELP.

```
ZCONSOLE = .False.
ZPRINTER = .TRUE.
ZTOLIN = 0.001
ZSOLNM = "IPSOL"
TITLE "INTEGER SOLUTIONS FOR MODEL"
MXINT(NOPRINT)
```

After each integer solution is found, control will be returned to the user when NO is answered to the question STANDARD? ⟨YES/NO⟩. Then the following sequence of TCL statements should be input after the READY.

```
ZSOLNO = integer value:
the number of the MIP solution just found
OUTPUT(FILE)
SAVEFILE
ZBIOBJ = cutoff value:
the value of the user desired CUTOFF
UR1 = ZCUROB
```

These five statements should be repeated for each MIP solution. In this manner the user can input a CUTOFF of varying decreasing amounts depending on the frequency of integer solutions and possible knowledge of the magnitude of the optimal solution. The user should never input a CUTOFF smaller than the optimal LP solution, since it is impossible for a MIP solution to be smaller than its correspondent LP solution.

When TEMPO returns with "NO REMAINING VALID NODES" for the above set of procedures, the user should input the following:

ZBIOBJ = UR1
MXINT(NOPRINT, RESTART)

which will restart the search for MIP solutions in the region between the last solution and any solution overlooked because of CUTOFF.

When TEMPO now returns with the demand exit request, the user should answer NO and input ZSOLNO = solution number, OUTPUT(FILE) and RETURN. This will save the latest solution for possible publishing.

The "NO REMAINING VALID NODES" after this sequence of TEMPO operations will indicate that the last solution is the optimal MIP solution, and the user can continue with PUBLISH for any solutions desired.

Decomposing Problems for Large Models

Another and more efficient means to solve large problems (Hughes et al., 1976) is to decompose the problem into two smaller problems with a common zone in both problems to allow manual interfacing after the solutions. It has been shown that improvements in run time and costs of one order of magnitude can be made if the problem division is possible (Hughes et al., 1976). The technique involves inputing one half of the model through the interactive input data generator and solving the problem to completion. The remaining half of the model then is input and solved in a similar manner. Following this step, manual adjustments are made to capital costs, O & M costs, and supply levels to remove duplication between the solutions for the two halves. The result is the optimal solution for the total model.

User Completion Procedures

When the user has completed the problem and wishes to terminate the TEMPO execution the TCL statement EXIT will terminate the run.

When the user is finished with the problem and does not want to maintain *any* remnants of the generated model or report structure, problem data, format and solutions, all TEMPO, GAMMA and user files should be removed. Files not removed and maintained on the system will incur daily charges. The files to remove are as follows:

RWF: GAMMA file containing ali report instructions for publish.
ZPROF: TEMPO file containing all problems (original and revised) by name.
ZSOLF: TEMPO file containing all solutions (LP and MIP) by name and number.
MODREV: USER file containing data for revision to LP problem for RANGE (if that alternative is chosen).
TEMPDATA: USER file containing temporary data used by interactive data generator.
MODELDATA: USER file containing all input data created by interactive data generator for GAMMA GENERATE.

All files on the user's account can be listed by the CANDE system command FILES.

CHAPTER V

RESULTS AND CONCLUSIONS

Sample Problem Results

The interactive data generator phase of the sample problem required approximately 1 1/2 hours of actual user time for input and revisions. If the data work sheets are completed accurately prior to execution of this phase even large models may be input in less than a day. The central processing unit (CPU) and input/output (IO) times for input of the sample problem were 12.2 seconds and 8.5 seconds respectively. The current rates for these chargeable items at the Utah State University Computer Center are $.08 and $.05 per second, respectively. This resulted in a charge of approximately $1.40 for the input of data. Additional charges for memory used amounted to a total cost of approximately $1.80. An additional charge of $2.50/hour for connect time may be incurred depending on the input line used.

These are very favorable costs considering the alternatives of manually calculating the data, creating the model structure (constraints, variable names, and coefficients), and key punching or creating a program to calculate the required data and its associated output. Even for very large models the total input portion should cost less than $20.00. Costs will be increased in proportion to the number of revisions required, the number of restarts, and, of course, the size of the problem. The user need not hurry through the interactive data generator phase, since on the Burroughs system charges are incurred only for actual machine use. Thus, except for connect charges, if applicable, the user is not charged for sitting and contemplating the answer to the next question (or taking a short break).

The model generation, LP problem solution, MIP problem, report publication, range from MIP solution, revision for LP range, and LP range of the sample problem took approximately 40 minutes of actual user time. The CPU time was 5.20 minutes and IO time was 1.6 minutes. The actual cost breakdown using normal priority rates is as follows:

CPU Time	$24.50
IO Time	4.50
Memory	25.00
Lines Printed	4.50
Total	$58.50

Of this total, approximately 25 percent was spent in the generate phase, 5 percent in the LP solution phase, 60 percent in the MIP phase, and the remaining 10 percent in the report, revision, and ranges phase. These proportions would change depending on the model size. The larger the model size and the larger the number of integer variables, the larger percentage of time the MIP phase will require.

Previous experience (Hughes et al., 1976) has shown that for very large problems the cost of a run (not including generate, report, or ranging) can exceed $1000 if made with normal priority rates. However, a very low cost rate (10 percent of normal priority rates) is available through the USU Computer Center by special arrangements for large problems of this sort. Therefore, the charges for even very large problems can be well within most budget limitations which should remove cost as a limiting factor for using mathematical programming to solve municipal water resource problems.

Recommendations

The publish report format of GAMMA is advantageous in that solution output can be phrased in a format which is understandable to persons who have only a limited knowledge of mathematical programming. Headings are used which are descriptive of the specific water supply facilities involved while the normal TEMPO output is generalized for any type of problem. The range output, however, can be confusing and a report format should be written to analyze ranging data and output in a simpler and more understandable form. The output may consist of actual facility names (existing well zone 22 alternate "A") rather than variable names. Other terminology may be used for the output headings and a zone by zone analysis performed rather than the present form. This ranging output revision would add in the presentation of the sensitivity analysis so that detailed descriptions would not have to be prepared in addition to the standard TEMPO output.

GAMMA has available a provision to allow comparisons of solutions by extracting data from the required solutions. This capability would be needed, for example, when a case study is being done that compares the results obtained from different right hand sides. Another possible use of the GAMMA capability would be to automatically revise and output the solutions to a problem that had been decomposed because of its size, as one output, with manual adjustments for overlap zones being left to GAMMA. It is conceivable that extremely large problems may be divided into more than two parts to insure fast run times and low costs. With the generality available with GAMMA, the number of divisions would be optional, depending upon solution efficiency and convenience.

It is recommended that if the user intends to become extensively involved with revising existing or writing new GAMMA and TEMPO procedures, that they become familiar with not only the GAMMA and TEMPO manuals, but also CANDE, FORTRAN, I/O SUBSYSTEMS, SOFTWARE, manuals, and other sources which reference disk and tape file handling. Disk and tape files are used extensively in the procedure described by this report, and must be used efficiently to avoid high costs.

The purpose of developing the procedure described herein is to assist in water supply planning. The final and most important recommendation is that a means in addition to publication should be used to communicate to the consulting engineer and others engaged in water supply planning the value and practicality of this tool. A series of seminars and short courses would effectively accomplish this needed information transfer.

Summary

The principal objectives of this research effort were: 1) to develop a computer capability which would eliminate much of the manual effort required in developing the detailed optimizing model matrix for regional and individual municipal water supply planning problems, and 2) develop a report writer which would produce a more convenient form of solution output which is written in the language of water supply planners.

Both of these objectives have been accomplished. The planning methodology developed uses several data manipulating Fortran subroutines, a model generator, and report writer called GAMMA, which is coupled to the TEMPO mathematical programming package on the Burroughs B6700 computer system at Utah State University. In addition to being available on Burroughs computers, software very similar to GAMMA is becoming available on a growing number of other computers (including UNIVAC 1108).

The planning system developed herein uses an integer programming algorithm which has several characteristics which are desirable for water supply planning problems. The system is designed for use by planning engineers who are not necessarily skilled at mathematical programming. The procedure developed avoids having to manually construct the objective function and constraints required for use by the optimizing algorithm. This is accomplished completely by the computer. The user merely calls the program from a remote keyboard type terminal (usable from any telephone), and then inputs the problem data in response to instructions from the program. The type and form of data that is requested is essentially that which would be required for a manual approach to the problem.

The degree of success with which the research objectives were accomplished represents in itself a potential problem. An important caveat is therefore appropriate. The extreme ease with which planning problems can be solved via the computer package developed herein invites its misuse by those who are not familiar with mathematical programming. One advantage of manually constructing a planning model is that the planner is intimately involved in determining the assumptions and criteria by which the alternatives are defined and compared. The greater the share of the model structure generation and computational effort that is done by the computer, the less the planner is required to become acquainted with the details of the problem.

This is not to say that those who are totally unfamiliar with mathematical programming in general and with IP in particular should not utilize the package developed herein. Indeed, an important objective of this research was to enlarge the group of users of such optimization tools to include those who are now unfamiliar with mathematical programming concepts. It is hoped, however, that the ease with which this package solves planning problems will first entice nonprogrammers to experiment with sample problems and then having experienced success with using the black box, they will be motivated to become familiar with at least the basic principles upon which it operates. Users should become acquainted with the simplex algorithm, for instance, in order to use this powerful tool in a rational manner in actual planning situations.

It also would be very foolish, to input a real world planning problem using only the program default values for cost and unit demand coefficients and to solve the problem in batch mode thereby obtaining only the "optimal" solution. One should always estimate his own rule of thumb version of unit costs and water demands as a check on default values. Also, in an actual planning situation several interim IP solutions in addition to the optimal solution should be output. These are usually invaluable in answering

questions such as why some other intuitive solution is not optimal or how close it is to optimal. Such other good solutions may in fact represent the true optimum when considered in conjunction with political or social considerations which the planner was unable to include in the mathematical model. Also, because of the difficulties related to interpreting the range output of an IP problem, these non-optimal integer solutions may represent the most useful type of parametric information that is available.

Before presenting a solution to a client as the recommended alternative, a planner should be prepared to answer questions concerning the implications of certain revisions to the solution. This implies that the planner should at the very least understand enough about sensitivity analysis (if not parametric analysis) to interpret the TEMPO output in regard to dual activities and upper and lower limits.

In summary, it is hoped that the extreme ease with which this package allows anyone to solve water supply planning problems will provide also the motivation to become familiar enough with the tool that it will become less of a black box and that it therefore will be used properly.

SELECTED BIBLIOGRAPHY

Bishop, A. Bruce, William J. Grenney, Rangesan Narayanan, and Stanley L. Klemetson. 1974. Evaluating water reuse alternatives in water resources planning. Utah Water Research Laboratory, PRWG123-1, Utah State University, Logan, Utah.

Bishop, A. Bruce, Rangesan Narayanan, Suravuth Pratishthananda, Stanley L. Klemetson, and William J. Grenney. 1975. Optimization of water allocation, wastewater treatment and reuse considering nonlinear costs, seasonal variations and stochastic supplies. Utah Water Research Laboratory, PRWG123-2, Utah State University, Logan, Utah.

Burroughs Corporation. 1972. B 6700 / B 7700 Command and Edit (CANDE) Language Information Manual.

Burroughs Corporation. 1973. B 6700 / B 7700 System Software Handbook.

Burroughs Corporation. 1974. B 6700 / B 7700 Fortran Reference Manual.

Burroughs Corporation. 1974. B 6700 Input/Output Subsystem Information Manual.

Burroughs Corporation. 1975. GAMMA User's Manual.

Burroughs Corporation. 1975. B 6700 TEMPO Reference Manual.

Dantzig, G. B., and R. M. Van Slyke. 1967. Generalized upper bounding techniques. Journal of Computer and System Sciences 1:213-226.

Driebeek, N. J. 1966. An algorithm for the solution of mixed integer programming problems. Management Science 12:576-587.

Garfinkel, Robert S., and George L. Nemhauser. 1972. Integer programming. John Wiley and Sons. New York.

Geoffrion, A. M., and R. E. Marsten. 1972. Integer programming: A framework and state-of-the-art survey. Management Science 18(9):465-491 (May).

Geoffrion, Arthur M. 1975. A guided tour of recent practical advances in integer linear programming. Western Management Science Institute, University of California, Los Angeles, Working Paper No. 220. Los Angeles, California.

Gomory, R. E. 1958. Outline of an algorithm for interger solutions to linear programs. Bulletin of the American Mathematical Society 64:275-278.

Gomory, R. E. 1963. An algorithm for integer solutions to linear programs. Recent Advances in Mathematical Programming. R. L. Graves and P. Wolfe, Editors, McGraw-Hill, New York. pp. 269-302.

Hall, W. A., and J. A. Dracup. 1970. Water resource systems engineering. McGraw-Hill Book Company, New York.

Hughes, T. C. 1972. A mixed integer programming approach to planning multiple water sources for municipal water supply. Utah Water Research Laboratory, PRWG73-3, Utah State University, Logan, Utah.

Hughes, Trevor C., and Calvin G. Clyde. 1973. Municipal water planning–mixed integer approach. Paper of the Technical Council on Water Resources Planning and Management, Journal of the Hydraulics Division, ASCE 99(HY11):2079-2095.

Hughes, Trevor C., William J. Grenney, A. Bruce Bishop, Calvin G. Clyde, Rangesan Narayanan, and Paul E. Pugner. 1976. Capability of integer programming algorithms in solving water resource planning problems. Utah Water Research Laboratory, PRWG175-1, Utah State University, Logan, Utah.

Klemetson, Stanley L., and William J. Grenney. 1975. Development of a dynamic programming model for the regionalization and staging of wastewater treatment plants. Utah Water Reserach Laboratory PRWA20-2, Utah State University, Logan, Utah.

Kolesar, Peter J., Kenneth L. Rider, Thomas, B. Crabill and Warren E. Walker. 1975. A queuing-linear programming approach to scheduling police patrol cars. Operations Research 23(6):1045-1061 (November-December).

McMillan, Claude, Jr. 1970 lst Ed., 1975 2nd Ed. Mathematical programming. Graduate School of Business and Administration, University of Colorado. John Wiley & Sons, Inc., New York, London, Sydney, Toronto.

Williams, H. P. 1974. Experiments in the formulation of integer programming problems. Mathematical Programming Study 2, December.

Appendix A

Interactive Data Generator Sample Run (Using Input Data from Appendix C) and Listing of MODELDATA File for This Problem which is Created by Data Generator Program (Appendix B)

```
        MATHEMATICAL PROGRAMMING USING THE MIXED INTEGER APPROACH
                   FOR MUNICIPAL WATER SOURCE PLANNING

                                   BY
                             PAUL E. PUGNER


              *************DATA INPUT PHASE*************


    THE SOUND OF THE BELL INDICATES DATA INPUT REQUIRED.
    TERMS INCLOSED IN <> INDICATE THE REQUIRED INPUT DATA.


    DO YOU WANT INSTRUCTIONS? <YES/NO>
#?
YES


    THE DATA INPUT PHASE IS DIVIDED INTO NINE SEGMENTS AS FOLLOWS:

          [1] ZONE NUMBERS AND ZONE NAMES.**
          [2] ZONE POPULATIONS, RESERVOIR ELEVATIONS AND
              SEASONAL DEMANDS.**
          [3] EXISTING WELL FACILITIES.
          [4] EXISTING SPRING FACILITIES.
          [5] EXISTING TREATMENT PLANT FACILITIES.
          [6] EXISTING AND PROPOSED CONNECTION STRUCTURE BETWEEN ZONES.
          [7] PROPOSED FUTURE WELLS.
          [8] PROPOSED FUTURE SPRINGS.
          [9] PROPOSED FUTURE TREATMENT PLANTS.
                ** THIS SEGMENT MUST BE EXECUTED.

    YOU MAY STOP AFTER ANY SEGMENT AND RESTART AT THE NEXT
    SEGMENT AT A LATER DATE WITHOUT LOSS OF ANY PREVIOUS DATA.


    THE DATA REQUIRED AND UNITS FOR THE SEGMMENTS ARE AS FOLLOWS:

SEGMENT 1
   ZONE NUMBER - ANY INTEGER VALUE BETWEEN 01 AND 98.
                 A MAXIMUM OF 40 ZONES CAN BE CONSIDERED.
   ZONE NAME - ANY STRING OF UP TO 18 CHARACTERS.

              ZONE NUMBER + ZONE NAME + SPACES <= 20 CHARACTERS
SEGMENT 2
   NUMBER OF SEASONS IN YOUR MODEL - MAXIMUM OF 4.
   NUMBER OF DAYS IN EACH SEASON.
   TERM OF THE SEASONS.  EX: JAN 15 - MAR 22.
   POPULATION OF EACH ZONE.
   RESERVOIR ELEVATION OF EACH ZONE - FEET FROM MEAN SEA LEVEL (MSL).
   SEASONAL DEMAND FOR EACH ZONE - GALLONS/PERSON/DAY.
   PEAK DAY MULTIPLIER CONSTANT.

SEGMENT 3
   WELL ELEVATION - FEET FROM MSL.
   WELL CAPACITY - GALLONS PER MINUTE (GPM).
   PEAK DAY MULTIPLIER CONSTANT.
   OPERATION AND MAINTENANCE COSTS (O&M) - $/MG.
   PUMPING POWER COSTS - $/MG/100 FT.

SEGMENT 4
   SPRING FLOWS - CUBIC FEET PER SECOND (CFS) EACH SEASON.
   PEAK DAY MULTIPLIER CONSTANT.
   O&M COSTS - $/MG EACH SEASON.

SEGMENT 5
   TREATMENT PLANT CAPACITY - MILLION GALLONS/DAY (MGD) EACH SEASON.
   PEAK DAY MULTIPLIER CONSTANT.
   O&M COSTS - $/MG/SEASON.

SEGMENT 6
   CAPITAL COSTS FOR EACH PIPE SIZE CONSIDERED.
   CAPITAL RECOVERY FACTOR TERMS - YEARS AND INTEREST RATE.
   ZONAL CONNECTION PATTERN - EXISTING AND PROPOSED.
   SIZE OF EXISTING PIPE - INCHES.
   DISTANCE BETWEEN ZONES - FEET.
   REVERSE FLOWS ALLOWED - EX: 01 TO 02  AND  02 TO 01.
   NUMBER OF SIZE OPTIONS ALLOWED IN THE MODEL - THE
      MAXIMUM NUMBER RECOMMENDED FOR MOST MODELS IS TWO.
   TYPE OF PIPE INSTALLATION AND BACKFILL.
   O&M TRANSFER COSTS - $/MG.
   PUMPING POWER COSTS - $/MG/100 FT.

SEGMENT 7
   CAPITAL COSTS FOR EACH WELL SIZE CONSIDERED.
   CAPITAL RECOVERY FACTOR TERMS - YEARS, RATE.
   WELL CAPACITY FOR EACH WELL SIZE OPTION - GPM.
   NUMBER OF WELLS OF A PARTICULAR SIZE ALLOWED PER ZONE.
   PEAK DAY MULTIPLIER CONSTANT.
   WELL ELEVATION - FEET FROM MSL.
   O&M COSTS - $/MG.
   PUMPING POWER COSTS - $/MG/100 FT.

SEGMENT 8
   CAPITAL COSTS FOR PROPOSED SPRINGS.
   CAPITAL RECOVERY FACTOR TERMS - YEARS, RATE.
   DISTANCE FROM SPRING TO CONNECTION - FEET.
   SPRING FLOW - CFS EACH SEASON.
   PIPE SIZE - INCHES.
   TYPE OF PIPE INSTALLATION AND BACKFILL.
   PEAK DAY MULTIPLIER CONSTANT.
   O&M COSTS - $/MG EACH SEASON.

SEGMENT 9
   CAPITAL COSTS FOR EACH SIZE TREATMENT PLANT.
   CAPITAL RECOVERY FACTOR TERMS - YEARS, RATE.
   TREATMENT PLANT CAPACITY - MGD EACH SEASON.
   PEAK DAY MULTIPLIER CONSTANT.
   O&M COSTS - $/MG EACH SEASON.

         [NOTE] THE MODEL CONTAINS DEFAULT VALUES FOR MANY OF
         THE ABOVE.  THESE VALUES WILL BE GIVEN IN MORE DETAIL
         WHEN EACH SEGMENT IS EXECUTED.  IF YOU USE A DEFAULT VALUE
         OR SUPPLY YOUR OWN CONSTANT IN ITS PLACE IN THE FIRST
         SEGMENT IT IS ASKED FOR THIS VALUE WILL BE ASSUMED ACCEPTABLE
         FOR ALL OTHER SEGMENTS WHERE IT IS REQUIRED. (EX: PEAK
         DAY SUPPLY CONSTANT MULTIPLIER)


*************************************************************
    THIS MODEL ASSUMES THAT SEASON 1 IS THE PEAK SEASON
    WITH RESPECT TO DEMAND AND SUPPLY.  PLEASE CONFORM TO
    THIS CONVENTION!!!
*************************************************************


    YOU WILL BE GIVEN EVERY OPPORTUNITY TO
    CORRECT INPUT DATA THROUGHT THE DATA INPUT
    PHASE.  ALL DATA IS INPUT FREE FORMAT.


    IS THIS A RESTART OF ANOTHER RUN? <YES/NO>
NO
```

```
********** SEGMENT 1 **********

   ENTER EACH ZONE NUMBER AND ZONE NAME FOLLOWED BY A
   RETURN.  ENTER A ZONE NUMBER OF 99 WHEN COMPLEATED
        EXAMPLE: 01 UPPER LOGAN  <RETURN>


01 MYTOWN
   14PARKSVILLE
 32 YOURTOWN
22 FARMTOWN
56 HALFWAYSTOP
99

   THE FOLLOWING IS A LIST OF YOUR INPUT OF ZONES AND THEIR NAMES.

 ROW  ZONE        **** ZONE NAME ****

  1   01          MYTOWN
  2   14          PARKSVILLE
  3   32          YOURTOWN
  4   22          FARMTOWN
  5   56          HALFWAYSTOP


   ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
YES
   ENTER ROW OF CHANGE
5
   ENTER THE NEW ZONE AND NAME
56 HALF WAY STOP
   5   56          HALF WAY STOP
   MORE CHANGES? <YES/NO>
NO
   DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>
YES

   THE FOLLOWING IS A LIST OF YOUR INPUT OF ZONES AND THEIR NAMES.

 ROW  ZONE        **** ZONE NAME ****

  1   01          MYTOWN
  2   14          PARKSVILLE
  3   32          YOURTOWN
  4   22          FARMTOWN
  5   56          HALF WAY STOP


   ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
NO

   DO YOU WANT TO CONTINUE? <YES/NO>
YES
```



```
********** SEGMENT 2 **********


   ENTER THE NUMBER OF SEASONS TO BE CONSIDERED IN YOUR MODEL. <1 - 4>
4
   ENTER THE NUMBER OF DAYS IN SEASON 1.
70
   ENTER THE MONTH TERM (EX: JAN 15 - MAR 31) FOR SEASON 1 (15 MAX).
JUN 1 - AUG 9
   ENTER THE NUMBER OF DAYS IN SEASON 2.
100
   ENTER THE MONTH TERM (EX: JAN 15 - MAR 31) FOR SEASON 2 (15 MAX).
AUG 10 - NOV 17
   ENTER THE NUMBER OF DAYS IN SEASON 3.
90
   ENTER THE MONTH TERM (EX: JAN 15 - MAR 31) FOR SEASON 3 (15 MAX).
NOV 18 - FEB 16
   ENTER THE NUMBER OF DAYS IN SEASON 4.
105
   ENTER THE MONTH TERM (EX: JAN 15 - MAR 31) FOR SEASON 4 (15 MAX).
FEB 17 - JUN 30

   SEASON   DAYS   TERM
      1       70   JUN 1 - AUG 9
      2      100   AUG 10 - NOV 17
      3       90   NOV 18 - FEB 16
      4      105   FEB 17 - JUN 30


   ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
NO


   NOW ENTER THE POPULATION OF EACH ZONE,
   THE RESERVOIR ELEVATION IN FEET FROM MSL AND
   THE DEMAND PER PERSON PER SEASON IN GALLONS
   PER DAY ALL SEPERATED BY COMMAS.


 *****01  MYTOWN                 *****
25000,4000,300,250,200,250

 *****14  PARKSVILLE             *****
100000,3700,250,200,150,190

 *****32  YOURTOWN               *****
17000,3600,275,250,200,250

 *****22  FARMTOWN               *****
8500,3500,900,700,400,800

 *****56  HALF WAY STOP          *****
12000,190,190,190,190
0
```

THE FOLLOWING IS A LIST OF YOUR INPUT DATA.

ROW	ZONE	POPULATION	RESERVOIR ELEVATION	DEMAND S1	DEMAND S2	DEMAND S3	DEMAND S4
1	01	25000	4000	300.0	250.0	200.0	250.0
2	14	100000	3700	250.0	200.0	150.0	190.0
3	32	17000	3600	275.0	250.0	200.0	250.0
4	22	8500	3500	900.0	700.0	400.0	800.0
5	56	12000	190	190.0	190.0	190.0	0.0

ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
YES
ENTER ROW OF CHANGE.
5
ENTER NEW POPULATION, RESERVOIR ELEVATION AND DEMANDS.
1120,3600,190,180,180,190
5 56 1120 3600 190.0 180.0 180.0 190.0
MORE CHANGES? <YES/NO>
NO
DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>
YES

THE FOLLOWING IS A LIST OF YOUR INPUT DATA.

ROW	ZONE	POPULATION	RESERVOIR ELEVATION	DEMAND S1	DEMAND S2	DEMAND S3	DEMAND S4
1	01	25000	4000	300.0	250.0	200.0	250.0
2	14	100000	3700	250.0	200.0	150.0	190.0
3	32	17000	3600	275.0	250.0	200.0	250.0
4	22	8500	3500	900.0	700.0	400.0	800.0
5	56	1120	3600	190.0	180.0	180.0	190.0

ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
NO

THE STANDARD PEAK DAY DEMAND IS 1.10 TIMES GREATER THAN THE PEAK SEASON DAILY DEMAND. IS THIS ACCEPTABLE FOR ALL ZONES? <YES/NO>
NO
IS THERE A CONSTANT VALUE THAT IS ACCEPTABLE? <YES/NO>
YES
ENTER PEAK DAY MULTIPLIER CONSTANT.
1.2

DO YOU WANT TO CONTINUE? <YES/NO>
YES

DO YOU HAVE ANY EXISTING WELLS IN YOUR MODEL? <YES/NO>
YES

********** SEGMENT 3 **********

***** ENTER EXISTING WELL INFORMATION *****

[NOTE] FOR MODEL SIMPLICITY IT IS ADVISIBLE TO GROUP ALL EXISTING WELLS IN A ZONE AND INPUT AS ONE SOURCE IF POSSIBLE.

*****01 MYTOWN *****

ENTER THE NUMBER OF EXISTING WELLS (OR GROUPS) IN THIS ZONE. <0 - 10>
1
ENTER WELL ELEVATION FROM MSL AND WELL CAPACITY <FEET,GPM> FOR ZONE 01 WELL "A".
3800,1000

*****14 PARKSVILLE *****

ENTER THE NUMBER OF EXISTING WELLS (OR GROUPS) IN THIS ZONE. <0 - 10>
2
ENTER WELL ELEVATION FROM MSL AND WELL CAPACITY <FEET,GPM> FOR ZONE 14 WELL "A".
3600,500
ENTER WELL ELEVATION FROM MSL AND WELL CAPACITY <FEET,GPM> FOR ZONE 14 WELL "B".
3500,2000

*****32 YOURTOWN *****

ENTER THE NUMBER OF EXISTING WELLS (OR GROUPS) IN THIS ZONE. <0 - 10>
0

*****22 FARMTOWN *****

ENTER THE NUMBER OF EXISTING WELLS (OR GROUPS) IN
THIS ZONE. <0 - 10>

0

*****56 HALF WAY STOP *****

ENTER THE NUMBER OF EXISTING WELLS (OR GROUPS) IN
THIS ZONE. <0 - 10>

1

ENTER WELL ELEVATION FROM MSL AND WELL CAPACITY
<FEET,GPM> FOR ZONE 56 WELL "A".

3600,3600

THE STANDARD PEAK DAY SUPPLY IS 0.70 TIMES
THE PEAK SEASONAL DAILY CAPACITY.
IS THIS ACCEPTABLE FOR ALL YOUR ZONES? <YES/NO>

NO

IS THERE A CONSTANT VALUE THAT IS ACCEPTABLE? <YES/NO>

NO

ENTER PEAK DAY MULTIPLIER FOR ZONE 01 WELL "A".

.9

ENTER PEAK DAY MULTIPLIER FOR ZONE 14 WELL "A".

.85

ENTER PEAK DAY MULTIPLIER FOR ZONE 14 WELL "B".

.9

ENTER PEAK DAY MULTIPLIER FOR ZONE 56 WELL "A".

.5

THE STANDARD OPERATION AND MAINTENANCE WELL
COSTS FOR THIS MODEL ARE $8.50/MG. (FOR PUMPS,
PIPELINE, ECT.) IS THIS ACCEPTABLE FOR ALL YOUR
ZONES? <YES/NO>

NO

IS THERE A CONSTANT VALUE THAT IS ACCEPTABLE? <YES/NO>

YES

ENTER O&M CONSTANT.

9.25

THE STANDARD POWER COSTS FOR PUMPING IN THIS
MODEL ARE $15.00/MG/100FT. IS THIS ACCEPTABLE
FOR ALL YOUR ZONES? <YES/NO>

YES

THE FOLLOWING IS A LIST OF YOUR DATA.

ROW	ZONE	ALT	COL 1 WELL ELEV FEET	COL 2 WELL CAPACITY GPM	COL 3 WELL CAPACITY MGD	COL 4 PEAK DAY CAPACITY MGD	COL 5 TOTAL COST $/MG
1	01	A	3800	1000	1.44	1.30	$39.25
2	14	A	3600	500	0.72	0.61	$24.25
3	14	B	3500	2000	2.88	2.59	$39.25
4	56	A	3600	3600	5.18	2.59	$9.25

ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>

YES

ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.

4,2

ENTER NEW WELL CAPACITY <GPM>.

100

4 56 A 3600 100 0.14 0.10 $9.25

MORE CHANGES? <YES/NO>

NO

DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>

NO

DO YOU WANT TO CONTINUE? <YES/NO>

YES

DO YOU HAVE ANY EXISTING SPRINGS IN YOUR MODEL? <YES/NO>
YES

********** SEGMENT 4 **********

***** ENTER EXISTING SPRING INFORMATION *****

[NOTE] FOR MODEL SIMPLICITY ALL EXISTING SPRINGS IN A ZONE MUST BE GROUPED AS ONE SOURCE. SINCE SPRING FLOWS ARE USUALLY THE LEAST UNIT COST SOURCE AND THE FIRST TO BE USED THIS WILL NOT CHANGE YOUR MODEL SOLUTION.

ENTER EXISTING SPRING FLOWS <CSF> FOR EACH SEASON SEPARATED BY COMMAS. ENTER ZERO FOR ALL SEASONS IF A ZONE DOES NOT HAVE EXISTING SPRINGS.

*****01 MYTOWN *****
0,0,0,0
*****14 PARKSVILLE *****
0,0,0,0
*****32 YOURTOWN *****
10,8,5,9
*****22 FARMTOWN *****
15,12,10,8
*****56 HALF WAY STOP *****
0,0,0,0

THE STANDARD PEAK DAY SUPPLY IS 0.70 TIMES THE PEAK SEASONAL DAILY CAPACITY.
IS THIS ACCEPTABLE FOR ALL YOUR ZONES? <YES/NO>

YES

THE STANDARD OPERATION AND MAINTENANCE SPRING COSTS FOR THIS MODEL ARE $6.60/MG (FOR CHLORINATION, DESANDER CLEANING AND MISC. O&M). IS THIS ACCEPTABLE FOR ALL YOUR ZONES? <YES/NO>

YES

THE FOLLOWING IS A LIST OF YOUR DATA.

ROW	ZONE	COL 1 COST $/MG	COL 2 PEAK DAY SPRING FLOW MGD	COL 3 SPRING FLOW S1 CSF	COL 4 SPRING FLOW S2 CSF	COL 5 SPRING FLOW S3 CSF	COL 6 SPRING FLOW S4 CSF
1	32	$6.60	4.52	10.00	8.00	5.00	9.00
2	22	$6.60	6.79	15.00	12.00	10.00	8.00

ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
YES
ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.
2,5
ENTER NEW SPRING FLOW FOR ZONE 22 SEASON 3.
8
2 22 $6.60 6.79 15.00 12.00 8.00 8.00
MORE CHANGES? <YES/NO>
YES
ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.
2,6
ENTER NEW SPRING FLOW FOR ZONE 22 SEASON 4.
10
2 22 $6.60 6.79 15.00 12.00 8.00 10.00
MORE CHANGES? <YES/NO>
NO
DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>
NO

DO YOU WANT TO CONTINUE? <YES/NO>
YES

DO YOU HAVE ANY EXISTING TREATMENT PLANTS IN YOUR MODEL?<YES/NO>
YES

********** SEGMENT 5 **********

***** ENTER DATA FOR EXISTING TREATMENT PLANTS *****

[NOTE] FOR MODEL SIMPLICITY IT IS ADVISIBLE TO GROUP
ALL EXISTING TREATMENT PLANTS IN A ZONE AND INPUT AS ONE
SOURCE IF POSSIBLE.
ALSO, SEASON ONE IS CONSIDERED THE PEAK SEASON.

*****01 MYTOWN *****

ENTER THE NUMBER OF EXISTING TREATMENT PLANTS (OR GROUPS)
IN THIS ZONE. <0 - 10>
0

*****14 PARKSVILLE *****

ENTER THE NUMBER OF EXISTING TREATMENT PLANTS (OR GROUPS)
IN THIS ZONE. <0 - 10>
1

ENTER EXISTING TREATMENT PLANT CAPACITIES <MGD> FOR
EACH SEASON SEPARATED BY COMMAS FOR ZONE 14 TREATMENT PLANT "A".
2,2,2,2
ENTER O&M COSTS <$/MG> FOR EACH SEASON FOR ZONE 14 PLANT "A".
123,123,123,123

*****32 YOURTOWN *****

ENTER THE NUMBER OF EXISTING TREATMENT PLANTS (OR GROUPS)
IN THIS ZONE. <0 - 10>
0

*****22 FARMTOWN *****

ENTER THE NUMBER OF EXISTING TREATMENT PLANTS (OR GROUPS)
IN THIS ZONE. <0 - 10>
0

*****56 HALF WAY STOP *****

ENTER THE NUMBER OF EXISTING TREATMENT PLANTS (OR GROUPS)
IN THIS ZONE. <0 - 10>
0

THE FOLLOWING IS A LIST OF YOUR DATA.

ROW	ZONE	ALT	COL 1 PEAK DAY CAP MGD	COL 2 PLANT CAP S-1 MGD	COL 2 PLANT CAP S-2 MGD	COL 2 PLANT CAP S-3 MGD	COL 2 PLANT CAP S-4 MGD	COL 3 O&M S-1 $/MG	COL 3 O&M S-2 $/MG	COL 3 O&M S-3 $/MG	COL 3 O&M S-4 $/MG
1	14	A	1.4	2.0	2.0	2.0	2.0	123.0	123.0	123.0	123.0

ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
NO

DO YOU WANT TO CONTINUE? <YES/NO>
YES

```
    DO YOU HAVE EXISTING OR PROPOSED INTERZONAL
    CONNECTIONS IN YOUR MODEL? <YES/NO>
YES

                ********** SEGMENT 6 **********


        *****ENTER THE ZONAL TRANSFER INFORMATION*****

                  ***** IMPORTANT NOTE *****
    A MAXIMUM OF 80 INTERCONNECTIONS AND 120 TOTAL PIPE
    OPTIONS ARE ALLOWED.
    A PROBABLE DIRECTION OF FLOW WILL BE IMPLIED IN THIS
    SECTION BY THE WAY YOU INDICATE CONNECTIONS.  FOR
    EXAMPLE:   IF YOU CONNECT ZONE 01 TO ZONE 14 THE
    IMPLICATION IS THAT THE MOST PROBABLE DIRECTION OF
    FLOW WILL BE TO ZONE 14 AND THIS MODEL WILL CALCULATE
    THE OPTIONAL PIPE SIZES BASED ON ZONE 14 DEMAND EVEN
    THOUGH REVERSE FLOW COULD BE ALLOWED.

    *******************************************************
    IF A ZONAL CONNECTION HAS BEEN PREVIOUSLY DEFINED OR A
    ZONE IS NOT TO BE CONNECTED TO ANY OTHER ZONES ENTER
    <RETURN> WHEN QUESTIONED ABOUT THIS ZONE.
    *******************************************************


    THE CALCULATING FORMULA FOR CAPITAL COSTS OF
    ZONAL TRANSFER FACILITIES IN THIS MODEL IS OF THE FORM:
      C = X*PIPE COST  +  K1*FTOR*X*D**E1      WHERE:
        C = TOTAL CAPITAL COST
        D = PIPE DIAMETER IN INCHES
        X = LENGTH OF LINE IN FEET
        PIPE COST = COST OF              PIPE PER FOOT
            6 INCH DIA. =    $2.30/FOOT
            8 INCH DIA. =    $3.40/FOOT
           10 INCH DIA. =    $4.95/FOOT
           12 INCH DIA. =    $6.79/FOOT
           14 INCH DIA. =    $8.26/FOOT
           16 INCH DIA. =   $10.20/FOOT
           18 INCH DIA. =   $14.97/FOOT
           20 INCH DIA. =   $18.19/FOOT
           24 INCH DIA. =   $25.34/FOOT
           30 INCH DIA. =   $39.71/FOOT
           36 INCH DIA. =   $56.79/FOOT

        K1 = INSTALLATION CONSTANT MULTIPLIER (DEFAULT K1=0.1426)
        FTOR = PIPE INSTALLATION DIFFICULTY FACTOR:
          NORMAL EXCAVATION AND BACKFILL (DEFAULT 1.0)
          ROUGHER EXCAVATION (BUT NO RIPPING) AND SELECT BACKFILL
            (DEFAULT 1.7)
          ROCK EXCAVATION AND BACKFILL FROM BORROW (DEFAULT 6.0)
          BELOW WATER EXCAVATION WITH GRAVEL BACKFILL (DEFAULT 3.0)
        E1 = INSTALLATION SCALE FACTOR EXPONENT (DEFAULT E1= 0.700)
     DEFAULT VALUES GIVE A CAPITAL COST OF ABOUT   $2,700
     FOR A 1000 FOOT  6 INCH LINE WITH A  500 GPM
     CAPACITY AND NORMAL EXCAVATION.


     WILL THESE DEFAULT VALUES BE ACCEPTABLE FOR ALL
     YOUR FUTURE INTERZONAL TRANSFER FACILITIES? <YES/NO>
YES
```

```
    THE CAPITAL RECOVERY FACTOR (CRF) FORMULA
    IS  CRF = R+ {R/[(1+R)**N - 1]} WHERE:
         R = INTEREST RATE (DEFAULT R = 0.060)
         N = NUMBER OF YEARS (DEFAULT N = 40)
    ARE THESE VALUES ACCEPTABLE FOR ALL YOUR FUTURE ZONAL
    TRANSFER FACILITIES? <YES/NO>


NO
    ARE THERE VALUES THAT WILL BE ACCEPTABLE FOR ALL
    YOUR FUTURE INTERZONAL TRANSFER FACILITIES? <YES/NO>
YES
    ENTER THE VALUES <N,R>.
40,.07

***** 01    MYTOWN                        *****
    CONNECTED TO ZONE ??
14
    IS THERE AN EXISTING CONNECTION? <YES/NO>
NO
    WHAT IS THE DISTANCE FROM
    MYTOWN  TO  PARKSVILLE                ? <FEET>
23000
    IS REVERSE FLOW ALLOWED? <YES/NO)
YES
    HOW MANY PIPE OPTIONS DO YOU WANT TO LOOK AT
    IN YOUR MODEL FOR THIS ZONAL TRANSFER? <1 - 4>
2
    INDICATE THE TYPE OF PIPE INSTALLATION <1-4>.

        <1> NORMAL EXCAVATION AND NORMAL BACKFILL
        <2> ROUGHER EXCAVATION (BUT NO RIPPING) AND SELECT BACKFILL
        <3> ROCK EXCAVATION AND BACKFILL FROM BORROW
        <4> BELOW WATER EXCAVATION WITH GRAVEL BACKFILL
1

***** 01    MYTOWN                        *****
    CONNECTED TO ZONE ??
22
    IS THERE AN EXISTING CONNECTION? <YES/NO>
YES
    WHAT IS THE SIZE OF YOUR EXISTING PIPE
    FROM MYTOWN  TO  FARMTOWN                ?
    < 6, 8,10,12,14,16,18,20,24,30,36> (INCH)
6
    DO YOU ALSO WANT TO LOOK AT NEW CONNECTIONS? <YES/NO>
YES
    WHAT IS THE DISTANCE FROM
    MYTOWN  TO  FARMTOWN                  ? <FEET>
26500
    IS REVERSE FLOW ALLOWED? <YES/NO)
NO
    HOW MANY PIPE OPTIONS DO YOU WANT TO LOOK AT
    IN YOUR MODEL FOR THIS ZONAL TRANSFER? <1 - 4>
1
    INDICATE THE TYPE OF PIPE INSTALLATION <1-4>.
2

***** 01    MYTOWN                        *****
    CONNECTED TO ZONE ??


***** 14    PARKSVILLE                    *****
    CONNECTED TO ZONE ??
56
```

```
    IS THERE AN EXISTING CONNECTION? <YES/NO>
NO
    WHAT IS THE DISTANCE FROM
    PARKSVILLE  TO  HALF WAY STOP             ? <FEET>
20000
    IS REVERSE FLOW ALLOWED? <YES/NO)
NO
    HOW MANY PIPE OPTIONS DO YOU WANT TO LOOK AT
    IN YOUR MODEL FOR THIS ZONAL TRANSFER? <1 - 4>
3
    INDICATE THE TYPE OF PIPE INSTALLATION <1-4>.
3

***** 14   PARKSVILLE              *****
    CONNECTED TO ZONE ??
32
    IS THERE AN EXISTING CONNECTION? <YES/NO>
YES
    WHAT IS THE SIZE OF YOUR EXISTING PIPE
    FROM PARKSVILLE  TO  YOURTOWN                  ?
    < 6, 8,10,12,14,16,18,20,24,30,36> (INCH)
8
    DO YOU ALSO WANT TO LOOK AT NEW CONNECTIONS? <YES/NO>
YES
    WHAT IS THE DISTANCE FROM
    PARKSVILLE  TO  YOURTOWN                  ? <FEET>
4200
    IS REVERSE FLOW ALLOWED? <YES/NO)
YES
    HOW MANY PIPE OPTIONS DO YOU WANT TO LOOK AT
    IN YOUR MODEL FOR THIS ZONAL TRANSFER? <1 - 4>
1
    INDICATE THE TYPE OF PIPE INSTALLATION <1-4>.
4

***** 14   PARKSVILLE              *****
    CONNECTED TO ZONE ??


***** 32   YOURTOWN                *****
    CONNECTED TO ZONE ??
32
    YOU CAN NOT CONNECT A ZONE TO ITSELF.


***** 32   YOURTOWN                *****
    CONNECTED TO ZONE ??
22
    IS THERE AN EXISTING CONNECTION? <YES/NO>
NO
    WHAT IS THE DISTANCE FROM
    YOURTOWN  TO  FARMTOWN                    ? <FEET>
48000
    IS REVERSE FLOW ALLOWED? <YES/NO)
NO
    HOW MANY PIPE OPTIONS DO YOU WANT TO LOOK AT
    IN YOUR MODEL FOR THIS ZONAL TRANSFER? <1 - 4>
3
    INDICATE THE TYPE OF PIPE INSTALLATION <1-4>.
2

***** 32   YOURTOWN                *****
    CONNECTED TO ZONE ??
```

```
***** 22   FARMTOWN                *****
    CONNECTED TO ZONE ??


***** 56   HALF WAY STOP           *****
    CONNECTED TO ZONE ??
22
    IS THERE AN EXISTING CONNECTION? <YES/NO>
NO
    WHAT IS THE DISTANCE FROM
    HALF WAY STOP  TO  FARMTOWN               ? <FEET>
20000
    IS REVERSE FLOW ALLOWED? <YES/NO)
YES
    HOW MANY PIPE OPTIONS DO YOU WANT TO LOOK AT
    IN YOUR MODEL FOR THIS ZONAL TRANSFER? <1 - 4>
3
    INDICATE THE TYPE OF PIPE INSTALLATION <1-4>.
2

***** 56   HALF WAY STOP           *****
    CONNECTED TO ZONE ??


    THE FOLLOWING IS A LIST OF YOUR DATA.

           COL       COL       COL       COL       COL       COL      COL
            1         2         3         4         5         6        7
         CONNECTED DISTANCE EXISTING REVERSE NO. PIPE EXISTING  SOIL
ROW ZONE  TO ZONE   A TO B    PIPE     FLOW  OPTIONS  PIPE SIZE TYPE
  1  01     14      23000    NO       YES       2        0.      1
  2  01     22      26500    YES      NO        1        6.      2
  3  14     56      20000    NO       NO        3        0.      3
  4  14     32       4200    YES      YES       1        8.      4
  5  32     22      48000    NO       NO        3        0.      2
  6  56     22      20000    NO       YES       3        0.      2


    ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
YES
    ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE
4,2
    WHAT IS THE DISTANCE FROM
    PARKSVILLE  TO  YOURTOWN                  ? <FEET>
42500
  4  14     32      42500    YES      YES       1        8.      4
    MORE CHANGES? <YES/NO>
YES
    ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE
5,3
    IS THERE AN EXISTING CONNECTION? <YES/NO>
NO
  5  32     22      48000    NO       NO        3        0.      2
    MORE CHANGES? <YES/NO>
YES
    ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE
5,5
    HOW MANY PIPE OPTIONS DO YOU WANT TO LOOK AT
    IN YOUR MODEL FOR THIS ZONAL TRANSFER? <1 - 4>
2
  5  32     22      48000    NO       NO        2        0.      2
    MORE CHANGES? <YES/NO>
NO
    DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>
NO
```

THE FOLLOWING IS A LIST OF CALCULATED DATA.

ROW	ZONE A	TO B	COL 1 PIPE DIA. INCH	COL 2 COST PER YEAR	COL 3 AB S1	COL 4 AB S2	COL 5 AB S3	COL 6 AB S4	COL 7 BA S1	COL 8 BA S2	COL 9 BA S3	COL 10 BA S4
					OPERATION AND MAINTENANCE COSTS - $/MG							
1	01	14	30	$71160	7.7	7.7	7.7	7.7	52.7	52.7	52.7	52.7
2	01	14	36	$100990	6.5	6.5	6.5	6.5	51.5	51.5	51.5	51.5
3	01	22	6	$0	115.1	115.1	115.1	115.1	0.0	0.0	0.0	0.0
4	01	22	20	$40080	12.8	12.8	12.8	12.8	0.0	0.0	0.0	0.0
5	14	56	6	$7940	30.3	30.3	30.3	30.3	0.0	0.0	0.0	0.0
6	14	56	8	$10600	22.6	22.6	22.6	22.6	0.0	0.0	0.0	0.0
7	14	56	10	$13850	18.2	18.2	18.2	18.2	0.0	0.0	0.0	0.0
8	14	32	8	$0	63.0	63.0	63.0	63.0	48.0	48.0	48.0	48.0
9	14	32	14	$34980	28.9	28.9	28.9	28.9	43.9	43.9	43.9	43.9
10	32	22	20	$72590	23.2	23.2	23.2	23.2	0.0	0.0	0.0	0.0
11	32	22	24	$99300	19.7	19.7	19.7	19.7	0.0	0.0	0.0	0.0
12	56	22	20	$30240	9.7	9.7	9.7	9.7	24.7	24.7	24.7	24.7
13	56	22	24	$41370	8.2	8.2	8.2	8.2	23.2	23.2	23.2	23.2
14	56	22	30	$63500	6.7	6.7	6.7	6.7	21.7	21.7	21.7	21.7

ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
NO

DO YOU WANT TO CONTINUE? <YES/NO>
YES

DO YOU HAVE PROPOSED FUTURE WELLS IN YOUR MODEL? <YES/NO>
YES

********** SEGMENT 7 **********

***** ENTER DATA FOR FUTURE WELLS *****

[NOTE] YOU ARE ALLOWED UP TO FOUR ALTERNATE SIZE WELLS PER ZONE
AND A MAXIMUM TOTAL OF 120.

THE CALCULATING FORMULA FOR CAPITAL COSTS OF WELLS
IN THIS MODEL IS OF THE FORM Y=A(X)**Z WHERE:
Y = CAPITAL COST
A = CONSTANT MULTIPLIER (DEFAULT A = 2010.)
X = THE FLOW OF THE WELL IN GALLONS PER MINUTE
Z = THE SCALE FACTOR EXPONENT (DEFAULT Z =0.453)
DEFAULT VALUES GIVE A CAPITAL COST OF ABOUT $45,900 FOR A
WELL OF 1000 GPM CAPACITY.

WILL THESE DEFAULT VALUES BE ACCEPTABLE FOR ALL
YOUR FUTURE WELLS? <YES/NO>
YES

THE CAPITAL RECOVERY FACTOR (CRF) FORMULA
IS CRF = R + {R/[(1+R)**N - 1]} WHERE:
R = INTEREST RATE (DEFAULT R = 0.060)
N = NUMBER OF YEARS (DEFAULT N = 25)
ARE THESE VALUES ACCEPTABLE FOR ALL YOUR FUTURE WELLS?
<YES/NO>

YES

***** 01 MYTOWN *****

HOW MANY OPTIONAL WELLS DO YOU WANT TO LOOK
AT FOR THIS ZONE? <0 - 4>
1

ENTER FLOW <GPM> FOR WELL OPTION "A" ZONE 01.
1000
ENTER THE NUMBER OF POSSIBLE WELLS OF THIS SIZE ALLOWED <NUMBER>.
1
ENTER WELL ELEVATION FROM MSL. <FEET>
37000

```
***** 14   PARKSVILLE            *****

    HOW MANY OPTIONAL WELLS DO YOU WANT TO LOOK
    AT FOR THIS ZONE? <0 - 4>
2
    ENTER FLOW <GPM> FOR WELL OPTION  "A"  ZONE 14.
2500
    ENTER THE NUMBER OF POSSIBLE WELLS OF THIS SIZE ALLOWED <NUMBER>.
1
    ENTER WELL ELEVATION FROM MSL. <FEET>
3550

    ENTER FLOW <GPM> FOR WELL OPTION  "B"  ZONE 14.
2500
    ENTER THE NUMBER OF POSSIBLE WELLS OF THIS SIZE ALLOWED <NUMBER>.
1
    ENTER WELL ELEVATION FROM MSL. <FEET>
3500

***** 32   YOURTOWN              *****

    HOW MANY OPTIONAL WELLS DO YOU WANT TO LOOK
    AT FOR THIS ZONE? <0 - 4>
1
    ENTER FLOW <GPM> FOR WELL OPTION  "A"  ZONE 32.
1500
    ENTER THE NUMBER OF POSSIBLE WELLS OF THIS SIZE ALLOWED <NUMBER>.
2
    ENTER WELL ELEVATION FROM MSL. <FEET>
3400

***** 22   FARMTOWN              *****

    HOW MANY OPTIONAL WELLS DO YOU WANT TO LOOK
    AT FOR THIS ZONE? <0 - 4>
1
    ENTER FLOW <GPM> FOR WELL OPTION  "A"  ZONE 22.
1500
    ENTER THE NUMBER OF POSSIBLE WELLS OF THIS SIZE ALLOWED <NUMBER>.
2
    ENTER WELL ELEVATION FROM MSL. <FEET>
3500

***** 56   HALF WAY STOP         *****

    HOW MANY OPTIONAL WELLS DO YOU WANT TO LOOK
    AT FOR THIS ZONE? <0 - 4>
0
```

THE FOLLOWING IS A LIST OF YOUR DATA

ROW	ZONE	ALT	COL 1 CAPITAL TOTAL	COL 2 CAPITAL PERYEAR	COL 3 FLOW GPM	COL 4 FLOW MGD	COL 5 MAX NUMBER	COL 6 ELEVATION MSL
1	01	A	$45,900	$3,590	1000	1.44	1	****
2	14	A	$69,500	$5,440	2500	3.60	1	3550
3	14	B	$69,500	$5,440	2500	3.60	1	3500
4	32	A	$55,200	$4,310	1500	2.16	2	3400
5	22	A	$55,200	$4,310	1500	2.16	2	3500

```
    ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
YES
    ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.
1,6
    ENTER WELL ELEVATION FROM MSL. <FEET>
3700
  1  01   A   $45,900   $3,590   1000  1.44    1          3700
    MORE CHANGES? <YES/NO>
YES
    ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.
1,5
    ENTER THE NUMBER OF POSSIBLE WELLS OF THIS SIZE ALLOWED <NUMBER>.
3
  1  01   A   $45,900   $3,590   1000  1.44    3          3700
    MORE CHANGES? <YES/NO>
YES
    ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.
2,3
    ENTER FLOW <GPM> FOR WELL OPTION  "A"  ZONE 14.
1500
  2  14   A   $55,200   $4,310   1500  2.16    1          3550
    MORE CHANGES? <YES/NO>
NO
    DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>
YES
```

THE FOLLOWING IS A LIST OF YOUR DATA

ROW	ZONE	ALT	COL 1 CAPITAL TOTAL	COL 2 CAPITAL PERYEAR	COL 3 FLOW GPM	COL 4 FLOW MGD	COL 5 MAX NUMBER	COL 6 ELEVATION MSL
1	01	A	$45,900	$3,590	1000	1.44	3	3700
2	14	A	$55,200	$4,310	1500	2.16	1	3550
3	14	B	$69,500	$5,440	2500	3.60	1	3500
4	32	A	$55,200	$4,310	1500	2.16	2	3400
5	22	A	$55,200	$4,310	1500	2.16	2	3500

```
    ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
NO
```

```
    THE FOLLOWING IS A LIST OF THE CALCULATED WELL DATA.

                           COL          COL
                            1            2
                   AVG.   PEAK DAY     WATER
                   FLOW     FLOW       COST
ROW ZONE ALT       MGD      MGD        $/MG
  1  01   A        1.44     1.01       54.25
  2  14   A        2.16     1.51       31.75
  3  14   B        3.60     2.52       39.25
  4  32   A        2.16     1.51       39.25
  5  22   A        2.16     1.51        9.25


    ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
YES
    ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.
5,2
    ENTER O&M COSTS FOR ZONE 22 OPTION  "A".
35.50
    ENTER PUMP POWER COSTS FOR ZONE 22 OPTION  "A".
12
  5  22   A      2.16      1.51       35.50
    MORE CHANGES? <YES/NO>
NO
    DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>
NO

    DO YOU WANT TO CONTINUE? <YES/NO>
NO
```



```
E (480030)MATRIX
#RUNNING 9909

**********************************************************************

      MATHEMATICAL PROGRAMMING USING THE MIXED INTEGER APPROACH
                  FOR MUNICIPAL WATER SOURCE PLANNING

                                 BY
                           PAUL E. PUGNER


              *************DATA INPUT PHASE*************

    THE SOUND OF THE BELL INDICATES DATA INPUT REQUIRED.
    TERMS INCLOSED IN <> INDICATE THE REQUIRED INPUT DATA.


    DO YOU WANT INSTRUCTIONS? <YES/NO>
#?
NO

    IS THIS A RESTART OF ANOTHER RUN? <YES/NO>
YES
    AT WHAT SEGMENT WOULD YOU LIKE TO RESTART?<1 TO 9>
8
```

```
   DO YOU HAVE PROPOSED FUTURE SPRINGS IN YOUR MODEL? <YES/NO>
YES

         ********** SEGMENT 8 **********

      ***** ENTER DATA FOR FUTURE SPRINGS *****

   THE CALCULATING FORMULA FOR CAPITAL COSTS OF
   FUTURE SPRINGS IN THIS MODEL IS OF THE FORM:
    C = X*PIPE COST  +  K1*FTOR*X*D**E1  +  SDEV   WHERE:
      C = TOTAL CAPITAL COST
      D = PIPE DIAMETER IN INCHES
      X = LENGTH OF LINE IN FEET
      PIPE COST = COST OF           PIPE PER FOOT
         6 INCH DIA. =    $2.30/FOOT
         8 INCH DIA. =    $3.40/FOOT
        10 INCH DIA. =    $4.95/FOOT
        12 INCH DIA. =    $6.79/FOOT
        14 INCH DIA. =    $8.26/FOOT
        16 INCH DIA. =   $10.20/FOOT
        18 INCH DIA. =   $14.97/FOOT
        20 INCH DIA. =   $18.19/FOOT
        24 INCH DIA. =   $25.34/FOOT
        30 INCH DIA. =   $39.71/FOOT
        36 INCH DIA. =   $56.79/FOOT

      K1 = INSTALLATION CONSTANT MULTIPLIER (DEFAULT K1=0.1426)
      FTOR = PIPE INSTALLATION DIFFICULTY FACTOR:
        NORMAL EXCAVATION AND BACKFILL (DEFAULT 1.0)
        ROUGHER EXCAVATION (BUT NO RIPPING) AND SELECT BACKFILL
          (DEFAULT 1.7)
        ROCK EXCAVATION AND BACKFILL FROM BORROW (DEFAULT 6.0)
        BELOW WATER EXCAVATION WITH GRAVEL BACKFILL (DEFAULT 3.0)
     E1 = INSTALLATION SCALE FACTOR EXPONENT (DEFAULT E1= 0.700)
      SDEV = ON SITE SPRING DEVELOPMENT COSTS (NO DEFAULT)
   DEFAULT VALUES GIVE A CAPITAL COST OF ABOUT   $2,300 EXCLUDING
   SPRING DEVELOPMENT COSTS FOR A 1000 FOOT  6 INCH LINE WITH
   A 1.11 CFS CAPACITY AND NORMAL EXCAVATION.

   WILL THESE DEFAULT VALUES BE ACCEPTABLE FOR ALL
   YOUR FUTURE SPRINGS? <YES/NO>
YES
```

```
   THE CAPITAL RECOVERY FACTOR (CRF) FORMULA
   IS  CRF = R+ {R/[(1+R)**N - 1]} WHERE:
       R = INTEREST RATE (DEFAULT R = 0.060)
       N = NUMBER OF YEARS (DEFAULT N = 40)
   ARE THESE VALUES ACCEPTABLE FOR ALL YOUR FUTURE SPRINGS?<YES/NO>

YES

   [NOTE]  SEASON ONE IS CONSIDERED THE PEAK SEASON FOR THIS MODEL
   AND THE MAXIMUM TOTAL NUMBER OF SPRINGS IS 80.

   ***** 01   MYTOWN                  *****
   ENTER THE NUMBER OF POTENTIAL SPRINGS FOR THIS ZONE. <0-4>
1

   ENTER DISTANCE <FEET> FROM SPRING TO RESERVOIR
   OR CONNECTION FOR SPRING   "A"   ZONE 01.
10000
   ENTER SPRING FLOW <CFS> FOR EACH SEASON.
15,12,7,9
   YOUR MAXIMUM SEASONAL FLOW OF  15.00 CFS
   MAY BE TRANSFERED BY A 24 INCH DIAMETER PIPE.
   DO YOU AGREE? <YES/NO>
YES
   INDICATE THE TYPE OF PIPE INSTALLATION <1-4>.

         <1> NORMAL EXCAVATION AND NORMAL BACKFILL
         <2> ROUGHER EXCAVATION (BUT NO RIPPING) AND SELECT BACKFILL
         <3> ROCK EXCAVATION AND BACKFILL FROM BORROW
         <4> BELOW WATER EXCAVATION WITH GRAVEL BACKFILL
1

   ENTER ON SITE DEVELOPMENT COSTS FOR SPRING  "A"  ZONE 01.
20000
```

```
      ***** 14   PARKSVILLE               *****
      ENTER THE NUMBER OF POTENTIAL SPRINGS FOR THIS ZONE. <0-4>
1


      ENTER DISTANCE <FEET> FROM SPRING TO RESERVOIR
      OR CONNECTION FOR SPRING  "A"  ZONE 14.
150000
      ENTER SPRING FLOW <CFS> FOR EACH SEASON.
25,20,12,21
      YOUR MAXIMUM SEASONAL FLOW OF  25.00 CFS
      MAY BE TRANSFERED BY A 30 INCH DIAMETER PIPE.
      DO YOU AGREE? <YES/NO>
NO
      WHAT SIZE WOULD YOU RECOMMEND?< 6 TO 36>
36
      INDICATE THE TYPE OF PIPE INSTALLATION <1-4>.
3
      ENTER ON SITE DEVELOPMENT COSTS FOR SPRING  "A"  ZONE 14.
43000



      ***** 32   YOURTOWN                 *****
      ENTER THE NUMBER OF POTENTIAL SPRINGS FOR THIS ZONE. <0-4>
1


      ENTER DISTANCE <FEET> FROM SPRING TO RESERVOIR
      OR CONNECTION FOR SPRING  "A"  ZONE 32.
200000
      ENTER SPRING FLOW <CFS> FOR EACH SEASON.
8,6,4,7
      YOUR MAXIMUM SEASONAL FLOW OF   8.00 CFS
      MAY BE TRANSFERED BY A 16 INCH DIAMETER PIPE.
      DO YOU AGREE? <YES/NO>
NO
      WHAT SIZE WOULD YOU RECOMMEND?< 6 TO 36>
14
      THE SIZE YOU PICKED IS TOO SMALL TO TRANSFER
      THE FLOW REQUIRED AT NORMAL OPERATING CONDITIONS.
      DO YOU WANT TO MAINTAIN THAT SIZE? <YES/NO>
YES
      INDICATE THE TYPE OF PIPE INSTALLATION <1-4>.
1
      ENTER ON SITE DEVELOPMENT COSTS FOR SPRING  "A"  ZONE 32.
12000
```

```
      ***** 22   FARMTOWN                 *****
      ENTER THE NUMBER OF POTENTIAL SPRINGS FOR THIS ZONE. <0-4>
0



      ***** 56   HALF WAY STOP            *****
      ENTER THE NUMBER OF POTENTIAL SPRINGS FOR THIS ZONE. <0-4>
0


      THE FOLLOWING IS A LIST OF YOUR DATA

                 COL    COL  COL     COL      COL           COL
                  1      2    3       4        5             6    >>>>
                 PIPE   PIPE PIPE
                LENGTH  SIZE INST CAPITAL  CAPITAL    FLOW  FLOW  FLOW  FLOW
ROW ZONE ALT     FEET   INCH TYPE  TOTAL   PERYEAR    CFS   CFS   CFS   CFS
  1  01   A     10000    24    1   $286500  $19040    15.0  12.0   7.0   9.0
  2  14   A    150000    36    3 $10138200 $673800    25.0  20.0  12.0  21.0
  3  32   A    200000    14    1  $1844900 $122610     8.0   6.0   4.0   7.0


      ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
YES
      ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.
1,1

      ENTER DISTANCE <FEET> FROM SPRING TO RESERVOIR
      OR CONNECTION FOR SPRING  "A"  ZONE 01.
100000
  1  01   A    100000  24    1  $2685900 $178500    15.0  12.0   7.0   9.0
      MORE CHANGES? <YES/NO>
NO
      DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>
YES


      THE FOLLOWING IS A LIST OF .OUR DATA

                 COL    COL  COL     COL      COL           COL
                  1      2    3       4        5             6    >>>>
                 PIPE   PIPE PIPE
                LENGTH  SIZE INST CAPITAL  CAPITAL    FLOW  FLOW  FLOW  FLOW
ROW ZONE ALT     FEET   INCH TYPE  TOTAL   PERYEAR    CFS   CFS   CFS   CFS
  1  01   A    100000    24    1  $2685900 $178500    15.0  12.0   7.0   9.0
  2  14   A    150000    36    3 $10138200 $673800    25.0  20.0  12.0  21.0
  3  32   A    200000    14    1  $1844900 $122610     8.0   6.0   4.0   7.0
```

ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
NO

THE STANDARD PEAK DAY SUPPLY IS 0.70 TIMES THE
PEAK SEASONAL DAILY CAPACITY. IS THIS ACCEPTABLE FOR ALL
YOUR ZONES? <YES/NO>

NO
IS THERE A CONSTANT THAT IS ACCEPTABLE? <YES/NO>
YES
ENTER PEAK DAY MULTIPLIER CONSTANT. <PDC>
.6

THE FOLLOWING IS A LIST OF CALCULATED DATA.

				COL 1	COL 2
			SPRING	PEAK DAY	O&M
		SPRING	FLOW	FLOW	COST
ROW	ZONE	ALT	S1-MGD	MGD	$/MG
1	01	A	9.69	5.82	6.60
2	14	A	16.16	9.69	6.60
3	32	A	5.17	3.10	6.60

ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
NO

DO YOU WANT TO CONTINUE? <YES/NO>
YES

DO YOU HAVE PROPOSED FUTURE TREATMENT
PLANTS IN YOUR MODEL? <YES/NO>
YES

********** SEGMENT 9 **********

***** ENTER DATA FOR FUTURE TREATMENT PLANTS *****

[NOTE] YOU ARE ALLOWED UP TO FOUR ALTERNATE SIZE PLANTS PER ZONE.

THE CAPITAL RECOVERY FACTOR (CRF) FORMULA
IS CRF = R+ (R/[(1+R)**N - 1]) WHERE:
R = INTEREST RATE (DEFAULT R = 0.060)
N = NUMBER OF YEARS (DEFAULT N = 25)
ARE THESE VALUES ACCEPTABLE FOR ALL YOUR FUTURE PLANTS? <YES/NO>

YES

[NOTE] SEASON ONE IS CONSIDERED THE PEAK SEASON FOR THIS MODEL.

*****01 MYTOWN *****

ENTER THE NUMBER OF PROPOSED TREATMENT PLANTS IN THIS ZONE. <0 - 4>
0

```
*****14  PARKSVILLE           *****


   ENTER THE NUMBER OF PROPOSED TREATMENT PLANTS IN THIS ZONE. <0 - 4>
1


   ENTER FUTURE TREATMENT PLANT CAPACITIES <MGD> FOR
   EACH SEASON SEPARATED BY COMMAS FOR ZONE 14 TREATMENT PLANT  "A".
20,20,20,20
   ENTER O&M COSTS <$/MG> FOR EACH SEASON FOR ZONE 14 PLANT  "A".
65,73,81,70
   ENTER TOTAL CAPITAL COSTS FOR ZONE 14 PLANT  "A".
4530000


*****32  YOURTOWN             *****


   ENTER THE NUMBER OF PROPOSED TREATMENT PLANTS IN THIS ZONE. <0 - 4>
1


   ENTER FUTURE TREATMENT PLANT CAPACITIES <MGD> FOR
   EACH SEASON SEPARATED BY COMMAS FOR ZONE 32 TREATMENT PLANT  "A".
5,5,5,5
   ENTER O&M COSTS <$/MG> FOR EACH SEASON FOR ZONE 32 PLANT  "A".
78,94,123,82
   ENTER TOTAL CAPITAL COSTS FOR ZONE 32 PLANT  "A".
1050000


*****22  FARMTOWN             *****


   ENTER THE NUMBER OF PROPOSED TREATMENT PLANTS IN THIS ZONE. <0 - 4>
0


*****56  HALF WAY STOP        *****


   ENTER THE NUMBER OF PROPOSED TREATMENT PLANTS IN THIS ZONE. <0 - 4>
0
```

```
    THE FOLLOWING IS A LIST OF YOUR DATA.

               COL      COL   COL   COL   COL   COL   COL   COL   COL   COL
                1        2     3     3     3     3     4     4     4     4
                        PEAK PLANT PLANT PLANT PLANT  O&M   O&M   O&M   O&M
    ZONE                DAY   CAP   CAP   CAP   CAP   S-1   S-2   S-3   S-4
     &   CAPITAL        CAP   S-1   S-2   S-3   S-4  $/MG  $/MG  $/MG  $/MG
ROW  ALT  TOTAL         MGD   MGD   MGD   MGD   MGD
  1 14 A$4530000       12.0  20.0  20.0  20.0  20.0  65.0  73.0  81.0  70.0
  2 32 A$1050000        3.0   5.0   5.0   5.0   5.0  78.0  94.0 123.0  82.0


    ARE THERE ANY CHANGES REQUIRED IN THIS DATA? <YES/NO>
YES
    ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.
1,2
    ENTER PEAK DAY MULTIPLIER FOR ZONE 14 PLANT  "A".
.8
  1 14 A$4530000       16.0  20.0  20.0  20.0  20.0  65.0  73.0  81.0  70.0
    MORE CHANGES? <YES/NO>
YES
    ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.
2,2
    ENTER PEAK DAY MULTIPLIER FOR ZONE 32 PLANT  "A".
.9
  2 32 A$1050000        4.5   5.0   5.0   5.0   5.0  78.0  94.0 123.0  82.0
    MORE CHANGES? <YES/NO>
NO
    DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>
NO


      THE FOLLOWING ARE SUGGESTIONS FOR RUNNING YOUR MODEL.


      YOUR MODEL CONTAINS LESS THAN FORTY INTEGER VARIABLES
      AND SHOULD BE CLASSIFIED AS A SMALL MODEL.  THE RUN TIMES
      WILL PROBABLY BE LESS THAN 30 MINUTES OF CPU.  THERE
      SHOULD BE NO NEED TO RUN THIS MODEL BY MODIFIED INTERACTIVE
      OR MODIFIED BATCH BUT IF DESIRED REFER TO USERS MANUAL
      CHAPTER IV.


      THIS ENDS THE DATA INPUT PHASE.  THE NEXT PHASE IS TO
      GENERATE THE MODEL MATRIX BY EXECUTING "TEMPO" AND
      INVOKE THE MACRO SOLVELP.  (REF. USERS MANUAL CHAPTER IV)
```

Listing of File MODELDATA.

```
ELEMENT DEMAND
    TABLE DEMAND
          *     PEAKDEM   DEMAND1   DEMAND2   DEMAND3   DEMAND4
          01       9.00       525       625       450       656
          14      30.00      1750      2000      1350      1995
          32       5.61       327       425       306       446
          22       9.18       536       595       306       714
          56       0.26        15        20        18        22
    LIST(ZO),T=20
          01       "MYTOWN              "
          14       "PARKSVILLE          "
          32       "YOURTOWN            "
          22       "FARMTOWN            "
          56       "HALF WAY STOP       "
ENDATA
ELEMENT SEASON
    TABLE SEASONS,ZERO
          *    DAYS
          1     70
          2    100
          3     90
          4    105
    LIST(S),T=15
          1    "JUN 1 - AUG 9  "
          2    "AUG 10 - NOV 17"
          3    "NOV 18 - FEB 16"
          4    "FEB 17 - JUN 30"
ENDATA
ELEMENT EXWELL
    TABLE EXSTWELL,ZERO
          *     FLOW      PDFLOW     COST
          01A   1.44        1.30    39.25
          14A   0.72        0.61    24.25
          14B   2.88        2.59    39.25
          56A   0.14        0.07     9.25
ENDATA
ELEMENT EXSPRG
    TABLE EXSTSPRG,ZERO
          *     COST      PDFLOW     FLOW1      FLOW2      FLOW3      FLOW4
          32    6.60        4.52      6.46       5.17       3.23       5.82
          22    6.60        6.79      9.69       7.76       5.17       6.46
ENDATA
ELEMENT EXTRPL
    TABLE EXSTTRPL,ZERO
          *    PDFLOW     CAP1      CAP2      CAP3      CAP4
          14A    1.40      2.0       2.0       2.0       2.0
          *              CST1     CST2     CST3     CST4
          14A            98.6    112.3    122.5    109.4
ENDATA
ELEMENT PIPENT
    TABLE PIPENET,ZERO
          *       CAPTL    CAPAC      AB1       AB2       AB3       AB4
          0114J   71160    18.72     7.65      7.65      7.65      7.65
          0114K  100990    26.93     6.46      6.46      6.46      6.46
          0122X       0     0.72   115.12    115.12    115.12    115.12
          0122H   40080     8.35    12.84     12.84     12.84     12.84
          1456A    7940     0.72    30.28     30.28     30.28     30.28
          1456B   10600     1.35    22.58     22.58     22.58     22.58
          1456C   13850     2.16    18.17     18.17     18.17     18.17
          1432X       0     1.35    62.98     62.98     62.98     62.98
          1432E   34980     4.03    28.88     28.88     28.88     28.88
          3222H   72590     8.35    23.25     23.25     23.25     23.25
          3222I   99300    11.95    19.68     19.68     19.68     19.68
          5622H   30240     8.35     9.69      9.69      9.69      9.69
          5622I   41370    11.95     8.20      8.20      8.20      8.20
          5622J   63500    18.72     6.66      6.66      6.66      6.66
          *                  BA1       BA2       BA3       BA4
          0114J            52.65     52.65     52.65     52.65
          0114K            51.46     51.46     51.46     51.46
          0122X             0.00      0.00      0.00      0.00
          0122H             0.00      0.00      0.00      0.00
          1456A             0.00      0.00      0.00      0.00
          1456B             0.00      0.00      0.00      0.00
          1456C             0.00      0.00      0.00      0.00
          1432X            47.98     47.98     47.98     47.98
          1432E            43.88     43.88     43.88     43.88
          3222H             0.00      0.00      0.00      0.00
          3222I             0.00      0.00      0.00      0.00
          5622H            24.69     24.69     24.69     24.69
          5622I            23.20     23.20     23.20     23.20
          5622J            21.66     21.66     21.66     21.66
    LIST (BTOA1)
          1401J
          1401K
          2201X
          2201H
          5614A
          5614B
          5614C
          3214X
          3214E
          2232H
          2232I
          2256H
          2256I
          2256J
    LIST(ATOB),T=40
          0114     "MYTOWN  TO  PARKSVILLE                  "
          0122     "MYTOWN  TO  FARMTOWN                    "
          1456     "PARKSVILLE  TO  HALF WAY STOP           "
          1432     "PARKSVILLE  TO  YOURTOWN                "
          3222     "YOURTOWN  TO  FARMTOWN                  "
          5622     "HALF WAY STOP  TO  FARMTOWN             "
ENDATA
ELEMENT FUTWEL
    TABLE FWELL,ZERO
          *       CAPTL      CAP       PCAP     OANDM      NUM
          01A      3590     1.44       1.01     54.25       3
          14A      4310     2.16       1.51     31.75       1
          14B      5440     3.60       2.52     39.25       1
          32A      4310     2.16       1.51     39.25       2
          22A      4310     2.16       1.51     35.50       2
ENDATA
ELEMENT FUTSPG
    TABLE FSPRG,ZERO
          *  PIPE  CAPTL    COST  PDFLOW     FLOW1    FLOW2    FLOW3    FLOW4
          01A  24  178500   6.60    5.82      9.69     7.76     4.52     5.82
          14A  36  673800   6.60    9.69     16.16    12.93     7.76    13.57
          32A  14  122610   6.60    3.10      5.17     3.88     2.59     4.52
ENDATA
ELEMENT FUTRPL
    TABLE FUTRTRPL,ZERO
          *       CAPTL  PDFLOW      CAP1      CAP2      CAP3      CAP4
          14A    354360    16.0     20.00     20.00     20.00     20.00
          32A     82130     4.5      5.00      5.00      5.00      5.00
          *            CST1     CST2     CST3     CST4
          14A         65.00    73.00    81.00    70.00
          32A         78.00    94.00   123.00    82.00
ENDATA
```

Appendix B

Interactive Data Generator Program Listing

MAIN PROGRAM - CONTROL SUBROUTINES

```
      FILE  5=FILE5
      FILE  6=DUMP
      FILE  10(TITLE="MODELDATA",KIND=PACK,MAXRECSIZE=20,BUFFERS=2,
     *PROTECTION=PROTECTED,AREAS=1000,AREASIZE=450,SAVEFACTOR=999)
      FILE  11(TITLE="TEMPDATA",KIND=PACK,MAXRECSIZE=20,BUFFERS=2,
     *PROTECTION=PROTECTED,AREAS=1000,AREASIZE=450,SAVEFACTOR=999)
      DIMENSION LISTZO(40,22),PLIST(11),PFLOW(11),RESELV(40)
      DIMENSION SDAYS(4),DEM(40,4),FTOR(4),PALT(11)
      LOGICAL PRSNT
      INTEGER S,PLIST,TF,SAVED,PDD
      COMMON IN,IO,IF,TF,LISTZO,N,S,IIII
      COMMON /CMN1/ PALT,PLIST,PFLOW,PCOST,FTOR
      COMMON /CMN2/ RESELV,DEM,SDAYS
      COMMON /CMN3/ OM1,PC1,PD1,PD2,OM2
      COMMON /CMN4/ R5,N5,AK1,AE1,AXPMP
      COMMON /CMN5/ AA1,ZZ1,R6,N6
      COMMON /CMN6/ R7,N7,AK2,AE2
      COMMON /CMN7/ R8,N8
      COMMON /CMN8/ TIME
      CALL DATA
      WRITE(IO,100)
100   FORMAT(////1X,71('*')//8X,'MATHEMATICAL PROGRAMMING '
     *'USING THE MIXED INTEGER APPROACH'/19X,'FOR MUNICIPAL WATER'
     *' SOURCE PLANNING'//36X,'BY'/30X
     *'PAUL E. PUGNER'/////16X,13('*'),'DATA INPUT PHASE',
     *13('*')//)
      WRITE(IO,105)
105   FORMAT(5X,'THE SOUND OF THE BELL INDICATES DATA INPUT ',
     *'REQUIRED.'/5X,'TERMS INCLOSED IN <> INDICATE THE REQU'
     *'IRED INPUT DATA.'////)
      WRITE(IO,110)
110   FORMAT(5X,'DO YOU WANT INSTRUCTIONS? <YES/NO>?')
3     READ(IN,1)ANS
1     FORMAT(A6)
      IF(ANS.EQ.'YES')GO TO 120
      IF(ANS.EQ.'NO')GO TO 200
      WRITE(IO,2)
2     FORMAT(5X,'PLEASE ANSWER YES OR NO.?')
      GO TO 3
120   WRITE(IO,130)
130   FORMAT(///5X,'THE DATA INPUT PHASE IS DIVIDED INTO NINE '
     *'SEGMENTS AS FOLLOWS:'//
     *10X,'[1] ZONE NUMBERS AND ZONE NAMES.**'/
     *10X,'[2] ZONE POPULATIONS, RESERVOIR ELEVATIONS AND'/
     *14X,'SEASONAL DEMANDS.**'/
     *10X,'[3] EXISTING WELL FACILITIES.'/
     *10X,'[4] EXISTING SPRING FACILITIES.'/
     *10X,'[5] EXISTING TREATMENT PLANT FACILITIES.'/
     *10X'[6] EXISTING AND PROPOSED CONNECTION STRUCTURE BETWEEN ZONES.'
     */10X,'[7] PROPOSED FUTURE WELLS.'/
     *10X,'[8] PROPOSED FUTURE SPRINGS.'/
     *10X,'[9] PROPOSED FUTURE TREATMENT PLANTS.'/
     *15X,'** THIS SEGMENT MUST BE EXECUTED.'//
     *5X,'YOU MAY STOP AFTER ANY SEGMENT AND RESTART AT THE NEXT'/
     *5X,'SEGMENT AT A LATER DATE WITHOUT LOSS OF ANY PREVIOUS DATA.'//)
      WRITE(IO,140)
140   FORMAT(//5X,'THE DATA REQUIRED AND UNITS FOR THE SEGM',
     *'MENTS ARE AS FOLLOWS:'//
     *1X,'SEGMENT 1'/
     *4X,'ZONE NUMBER - ANY INTEGER VALUE BETWEEN 01 AND 98.'/
     *19X,'A MAXIMUM OF 40 ZONES CAN BE CONSIDERED.'/
     *4X,'ZONE NAME - ANY STRING OF UP TO 18 CHARACTERS.'//
     *15X,'ZONE NUMBER + ZONE NAME + SPACES <= 30 CHARACTERS'/
     *1X,'SEGMENT 2'/
     *4X,'NUMBER OF SEASONS IN YOUR MODEL - MAXIMUM OF 4.'/
     *4X,'NUMBER OF DAYS IN EACH SEASON.'/
     *4X,'TERM OF THE SEASONS.  EX: JAN 15 - MAR 22.'/
     *4X,'POPULATION OF EACH ZONE.'/
     *4X,'RESERVOIR ELEVATION OF EACH ZONE - FEET FROM MEAN SEA LEVEL'
     *' (MSL).'/
     *4X,'SEASONAL DEMAND FOR EACH ZONE - GALLONS/PERSON/DAY.'/
     *4X,'PEAK DAY MULTIPLIER CONSTANT.'//
     *1X,'SEGMENT 3'/
     *4X,'WELL ELEVATION - FEET FROM MSL.'/
     *4X,'WELL CAPACITY - GALLONS PER MINUTE (GPM).'/
     *4X,'PEAK DAY MULTIPLIER CONSTANT.'/
     *4X,'OPERATION AND MAINTENANCE COSTS (O&M) - $/MG.'/
     *4X,'PUMPING POWER COSTS - $/MG/100 FT.'//
     *1X,'SEGMENT 4'/
     *4X,'SPRING FLOWS - CUBIC FEET PER SECOND (CFS) EACH SEASON.'/
     *4X,'PEAK DAY MULTIPLIER CONSTANT.'/
     *4X,'O&M COSTS - $/MG EACH SEASON.'//
     *1X,'SEGMENT 5'/
     *4X,'TREATMENT PLANT CAPACITY - MILLION GALLONS/DAY (MGD) EACH'
     *' SEASON.'/
     *4X,'PEAK DAY MULTIPLIER CONSTANT.'/
     *4X,'O&M COSTS - $/MG/SEASON.'//
     *1X,'SEGMENT 6'/
     *4X,'CAPITAL COSTS FOR EACH PIPE SIZE CONSIDERED.'/
     *4X,'CAPITAL RECOVERY FACTOR TERMS - YEARS AND INTEREST RATE.'/
     *4X,'ZONAL CONNECTION PATTERN - EXISTING AND PROPOSED.'/
     *4X,'SIZE OF EXISTING PIPE - INCHES.'/
     *4X,'DISTANCE BETWEEN ZONES - FEET.'/
     *4X,'REVERSE FLOWS ALLOWED - EX: 01 TO 02  AND  02 TO 01.'/
     *4X,'NUMBER OF SIZE OPTIONS ALLOWED IN THE MODEL - THE'/
     *7X,'MAXIMUM NUMBER RECOMMENDED FOR MOST MODELS IS TWO.'/
     *4X,'TYPE OF PIPE INSTALLATION AND BACKFILL.'/
     *4X,'O&M TRANSFER COSTS - $/MG.'/
     *4X,'PUMPING POWER COSTS - $/MG/100 FT.'//
     *1X,'SEGMENT 7'/
     *4X,'CAPITAL COSTS FOR EACH WELL SIZE CONSIDERED.'/
     *4X,'CAPITAL RECOVERY FACTOR TERMS - YEARS, RATE.'/
     *4X,'WELL CAPACITY FOR EACH WELL SIZE OPTION - GPM.'/
     *4X,'NUMBER OF WELLS OF A PARTICULAR SIZE ALLOWED PER ZONE.'/
     *4X,'PEAK DAY MULTIPLIER CONSTANT.'/
     *4X,'WELL ELEVATION - FEET FROM MSL.'/
     *4X,'O&M COSTS - $/MG.'/
     *4X,'PUMPING POWER COSTS - $/MG/100 FT.'//
     *1X,'SEGMENT 8'/
     *4X,'CAPITAL COSTS FOR PROPOSED SPRINGS.'/
     *4X,'CAPITAL RECOVERY FACTOR TERMS - YEARS, RATE.'/
     *4X,'DISTANCE FROM SPRING TO CONNECTION - FEET.'/
     *4X,'SPRING FLOW - CFS EACH SEASON.'/
     *4X,'PIPE SIZE - INCHES.'/
     *4X,'TYPE OF PIPE INSTALLATION AND BACKFILL.'/
     *4X,'PEAK DAY MULTIPLIER CONSTANT.'/
     *4X,'O&M COSTS - $/MG EACH SEASON.'//
     *1X,'SEGMENT 9'/
     *4X,'CAPITAL COSTS FOR EACH SIZE TREATMENT PLANT.'/
```

```
     *4X,'CAPITAL RECOVERY FACTOR TERMS - YEARS, RATE.'/
     *4X,'TREATMENT PLANT CAPACITY - MGD EACH SEASON.'/
     *4X,'PEAK DAY MULTIPLIER CONSTANT.'/
     *4X,'O&M COSTS - $/MG EACH SEASON.'///
     *10X,'[NOTE] THE MODEL CONTAINS DEFAULT VALUES FOR MANY OF'/
     *10X,'THE ABOVE.  THESE VALUES WILL BE GIVEN IN MORE DETAIL'/
     *10X,'WHEN EACH SEGMENT IS EXECUTED.  IF YOU USE A DEFAULT VALUE'/
     *10X,'OR SUPPLY YOUR OWN CONSTANT IN ITS PLACE IN THE FIRST'/
     *10X'SEGMENT IT IS ASKED FOR THIS VALUE WILL BE ASSUMED ACCEPTABLE'
     */10X,'FOR ALL OTHER SEGMENTS WHERE IT IS REQUIRED. (EX: PEAK'/
     *10X'DAY SUPPLY CONSTANT MULTIPLIER)'////
     *1X,60('*')/5X,'THIS MODEL ASSUMES THAT SEASON 1 IS THE'
     *' PEAK SEASON'/
     *5X,'WITH RESPECT TO DEMAND AND SUPPLY.  PLEASE CONFORM TO'/
     *5X,'THIS CONVENTION!!!'/1X,60('*')//)
      WRITE(IO,150)
150   FORMAT(5X,'YOU WILL BE GIVEN EVERY OPPORTUNITY TO '/
     *5X,'CORRECT INPUT DATA THROUGHT THE DATA INPUT'/
     *5X,'PHASE.  ALL DATA IS INPUT FREE FORMAT.'//)
200   WRITE(IO,210)
210   FORMAT(//5X,'IS THIS A RESTART OF ANOTHER RUN? <YES/NO>?')
220   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 230
      IF(ANS.EQ.'NO')GO TO 300
      WRITE(IO,2)
      GO TO 220
230   INQUIRE(IF,PRESENT=PRSNT)
      IF(PRSNT)GO TO 231
      GO TO 232
231   INQUIRE(TF,PRESENT=PRSNT)
      IF(PRSNT)GO TO 237
232   WRITE(IO,234)
234   FORMAT(5X,'I CAN NOT FIND A RESTART FILE.  PLEASE DO A'/
     *5X,'LIST FILES OR VERIFY THIS IS THE SAME ACCOUNT NUMBER'/
     *5X,'YOU STARTED ON LAST EXECUTION.'////)
      GO TO 4000
237   WRITE(IO,240)
240   FORMAT(5X,'AT WHAT SEGMENT WOULD YOU LIKE TO RESTART?'
     *'<1 TO 9>?')
      READ(IN,/)START
      DO 250 M=1,9
      IF(START.EQ.M)GO TO(300,400,500,600,700,700,700,700,700)START
250   CONTINUE
      WRITE(IO,260)
260   FORMAT(5X,'UNACCEPTABLE SEGMENT NUMBER.')
      GO TO 237
300   CALL ZONENM(SAVED)
      WRITE(IO,301)
301   FORMAT(//5X,'DO YOU WANT TO CONTINUE? <YES/NO>?')
330   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 420
      IF(ANS.EQ.'NO')GO TO 2000
      WRITE(IO,2)
      GO TO 330
400   READ(TF,401)N
401   FORMAT(I2)
      READ(TF,402)((LISTZO(I,J),J=1,22),I=1,N)
402   FORMAT(110A1)
420   CALL POPUL(SAVED,PDFTR)
      WRITE(IO,301)
430   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 520
      IF(ANS.EQ.'NO')GO TO 2000
      WRITE(IO,2)
      GO TO 430
500   READ(TF,501)N,S,IIII
501   FORMAT(3I6)
      READ(TF,402)((LISTZO(I,J),J=1,22),I=1,N)
      READ(TF,502)(RESELV(I),I=1,N)
502   FORMAT(20I5)
      READ(TF,503)((DEM(I,J),J=1,S),I=1,N)
503   FORMAT(16I7)
      READ(TF,504)PDD,PDFTR
      READ(TF,503)(SDAYS(I),I=1,S)
504   FORMAT(I2,F8.4)
520   WRITE(IO,521)
521   FORMAT(//5X,'DO YOU HAVE ANY EXISTING WELLS IN YOUR'
     *' MODEL? <YES/NO>?')
522   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 530
      IF(ANS.EQ.'NO')GO TO 550
      WRITE(IO,2)
      GO TO 522
530   CALL EXWELL(SAVED,              PC,PUMPC              )
      GO TO 545
550   ANSWE='NO'
      CALL EXW(ANSWE,SAVED)
545   WRITE(IO,301)
540   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 620
      IF(ANS.EQ.'NO')GO TO 2000
      WRITE(IO,2)
      GO TO 540
600   READ(TF,501)N,S,IIII
      READ(TF,402)((LISTZO(I,J),J=1,22),I=1,N)
      READ(TF,502)(RESELV(I),I=1,N)
      READ(TF,503) ((DEM(I,J),J=1,S),I=1,N)
      READ(TF,504)PDD,PDFTR
      READ(TF,503)(SDAYS(I),I=1,S)
      READ(TF,601)              PC,PUMPC
601   FORMAT(   I2,F8.3)
620   WRITE(IO,621)
621   FORMAT(//5X,'DO YOU HAVE ANY EXISTING SPRINGS IN YOUR '
     *'MODEL? <YES/NO>?')
622   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 630
      IF(ANS.EQ.'NO')GO TO 650
      WRITE(IO,2)
      GO TO 622
630   CALL EXSPRG(SAVED                              )
      GO TO 645
650   ANSWE='NO'
      CALL EXS(ANSWE,SAVED)
645   WRITE(IO,301)
640   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 720
      IF(ANS.EQ.'NO')GO TO 2000
      WRITE(IO,2)
      GO TO 640
700   READ(TF,501)N,S,IIII
      READ(TF,402)((LISTZO(I,J),J=1,22),I=1,N)
      READ(TF,502)(RESELV(I),I=1,N)
```

```
      READ(TF,503) ((DEM(I,J),J=1,S),I=1,N)
      READ(TF,504)PDD,PDFTR
      READ(TF,503)(SDAYS(I),I=1,S)
      READ(TF,601)               PC,PUMPC

      GO TO(720,820,920,1020,1120)START-4
720   WRITE(IO,721)
721   FORMAT(//5X,'DO YOU HAVE ANY EXISTING TREATMENT PLANTS '
     *'IN YOUR MODEL?<YES/NO>?')
722   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 730
      IF(ANS.EQ.'NO')GO TO 750
      WRITE(IO,2)
      GO TO 722
730   CALL EXTRPL(SAVED              )
      GO TO 745
750   ANSWE='NO'
      CALL EXT(ANSWE,SAVED)
745   WRITE(IO,301)
740   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 820
      IF(ANS.EQ.'NO')GO TO 2000
      WRITE(IO,2)
      GO TO 740
820   WRITE(IO,821)
821   FORMAT(//5X,'DO YOU HAVE EXISTING OR PROPOSED INTERZONAL'/
     *5X,'CONNECTIONS IN YOUR MODEL? <YES/NO>?')
822   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 830
      IF(ANS.EQ.'NO')GO TO 850
      WRITE(IO,2)
      GO TO 822
830   CALL CONECT(SAVED,PC,PUMPC,PDD,PDFTR)
      GO TO 845
850   ANSWE='NO'
      CALL CON(ANSWE,SAVED)
845   WRITE(IO,301)
840   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 920
      IF(ANS.EQ.'NO')GO TO 2000
      WRITE(IO,2)
      GO TO 840
920   WRITE(IO,921)
921   FORMAT(//5X,'DO YOU HAVE PROPOSED FUTURE WELLS IN YOUR '
     *'MODEL? <YES/NO>?')
922   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 930
      IF(ANS.EQ.'NO')GO TO 950
      WRITE(IO,2)
      GO TO 922
930   CALL FWELLS(SAVED                           )
      GO TO 945
950   ANSWE='NO'
      CALL FWE(ANSWE,SAVED)
945   WRITE(IO,301)
940   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 1020
      IF(ANS.EQ.'NO')GO TO 2000
      WRITE(IO,2)
      GO TO 940
1020  WRITE(IO,1021)
1021  FORMAT(//5X,'DO YOU HAVE PROPOSED FUTURE SPRINGS IN '
     *'YOUR MODEL? <YES/NO>?')
1022  READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 1030
      IF(ANS.EQ.'NO')GO TO 1050
      WRITE(IO,2)
      GO TO 1022
1030  CALL FSPNGS(SAVED                           )
      GO TO 1045
1050  ANSWE='NO'
      CALL FSP(ANSWE,SAVED)
1045  WRITE(IO,301)
1040  READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 1120
      IF(ANS.EQ.'NO')GO TO 2000
      WRITE(IO,2)
      GO TO 1040
1120  WRITE(IO,1121)
1121  FORMAT(//5X,'DO YOU HAVE PROPOSED FUTURE TREATMENT'/
     *5X,'PLANTS IN YOUR MODEL? <YES/NO>?')
1122  READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 1130
      IF(ANS.EQ.'NO')GO TO 1150
      WRITE(IO,2)
      GO TO 1122
1130  CALL FTRPLS(SAVED    ,           )
      GO TO 1200
1150  ANSWE='NO'
      CALL FTP(ANSWE,SAVED)
1200  CALL QUEST(IIII)
2000  LOCK IF
      LOCK TF
4000  CALL EXIT
      END
```

MATRIX - BINDING SUBROUTINES

```
HOST IS OBJECT/MAINPROGRAM;
BIND DATA FROM CANDE/DATA;
BIND ZONENM FROM CANDE/ZONENM;
BIND POPUL FROM CANDE/POPUL;
BIND EXWELL FROM CANDE/EXWELL;
BIND EXSPRG FROM CANDE/EXSPRG;
BIND EXTRPL FROM CANDE/EXTRPL;
BIND CONECT FROM CANDE/CONECT;
BIND FWELLS FROM CANDE/FWELLS;
BIND FSPNGS FROM CANDE/FSPNGS;
BIND FTRPLS FROM CANDE/FTRPLS;
BIND QUEST FROM CANDE/QUEST;
```

```
      *4X,'CAPITAL RECOVERY FACTOR TERMS - YEARS, RATE.'/
      *4X,'TREATMENT PLANT CAPACITY - MGD EACH SEASON.'/
      *4X,'PEAK DAY MULTIPLIER CONSTANT.'/
      *4X,'O&M COSTS - $/MG EACH SEASON.'///
      *10X,'[NOTE] THE MODEL CONTAINS DEFAULT VALUES FOR MANY OF'/
      *10X,'THE ABOVE.  THESE VALUES WILL BE GIVEN IN MORE DETAIL'/
      *10X,'WHEN EACH SEGMENT IS EXECUTED.  IF YOU USE A DEFAULT VALUE'/
      *10X,'OR SUPPLY YOUR OWN CONSTANT IN ITS PLACE IN THE FIRST'/
      *10X'SEGMENT IT IS ASKED FOR THIS VALUE WILL BE ASSUMED ACCEPTABLE'
      */10X,'FOR ALL OTHER SEGMENTS WHERE IT IS REQUIRED. (EX: PEAK'/
      *10X'DAY SUPPLY CONSTANT MULTIPLIER)'////
      *1X,60('*')/5X,'THIS MODEL ASSUMES THAT SEASON 1 IS THE'
      *' PEAK SEASON'/
      *5X,'WITH RESPECT TO DEMAND AND SUPPLY.  PLEASE CONFORM TO'/
      *5X,'THIS CONVENTION!!!'/1X,60('*')//)
       WRITE(IO,150)
 150   FORMAT(5X,'YOU WILL BE GIVEN EVERY OPPORTUNITY TO '/
      *5X,'CORRECT INPUT DATA THROUGHT THE DATA INPUT'/
      *5X,'PHASE.  ALL DATA IS INPUT FREE FORMAT.'//)
 200   WRITE(IO,210)
 210   FORMAT(//5X,'IS THIS A RESTART OF ANOTHER RUN? <YES/NO>?')
 220   READ(IN,1)ANS
       IF(ANS.EQ.'YES')GO TO 230
       IF(ANS.EQ.'NO')GO TO 300
       WRITE(IO,2)
       GO TO 220
 230   INQUIRE(IF,PRESENT=PRSNT)
       IF(PRSNT)GO TO 231
       GO TO 232
 231   INQUIRE(TF,PRESENT=PRSNT)
       IF(PRSNT)GO TO 237
 232   WRITE(IO,234)
 234   FORMAT(5X,'I CAN NOT FIND A RESTART FILE.  PLEASE DO A'/
      *5X,'LIST FILES OR VERIFY THIS IS THE SAME ACCOUNT NUMBER'/
      *5X,'YOU STARTED ON LAST EXECUTION.'////)
       GO TO 4000
 237   WRITE(IO,240)
 240   FORMAT(5X,'AT WHAT SEGMENT WOULD YOU LIKE TO RESTART?'
      *'<1 TO 9>?')
       READ(IN,/)START
       DO 250 M=1,9
       IF(START.EQ.M)GO TO(300,400,500,600,700,700,700,700,700)START
 250   CONTINUE
       WRITE(IO,260)
 260   FORMAT(5X,'UNACCEPTABLE SEGMENT NUMBER.')
       GO TO 237
 300   CALL ZONENM(SAVED)
       WRITE(IO,301)
 301   FORMAT(//5X,'DO YOU WANT TO CONTINUE? <YES/NO>?')
 330   READ(IN,1)ANS
       IF(ANS.EQ.'YES')GO TO 420
       IF(ANS.EQ.'NO')GO TO 2000
       WRITE(IO,2)
       GO TO 330
 400   READ(TF,401)N
 401   FORMAT(I2)
       READ(TF,402)((LISTZO(I,J),J=1,22),I=1,N)
 402   FORMAT(110A1)
 420   CALL POPUL(SAVED,PDFTR)
       WRITE(IO,301)
 430   READ(IN,1)ANS
       IF(ANS.EQ.'YES')GO TO 520
       IF(ANS.EQ.'NO')GO TO 2000
       WRITE(IO,2)
       GO TO 430
 500   READ(TF,501)N,S,IIII
 501   FORMAT(3I6)
       READ(TF,402)((LISTZO(I,J),J=1,22),I=1,N)
       READ(TF,502)(RESELV(I),I=1,N)
 502   FORMAT(20I5)
       READ(TF,503)((DEM(I,J),J=1,S),I=1,N)
 503   FORMAT(16I7)
       READ(TF,504)PDD,PDFTR
       READ(TF,503)(SDAYS(I),I=1,S)
 504   FORMAT(I2,F8.4)
 520   WRITE(IO,521)
 521   FORMAT(//5X,'DO YOU HAVE ANY EXISTING WELLS IN YOUR'
      *' MODEL? <YES/NO>?')
 522   READ(IN,1)ANS
       IF(ANS.EQ.'YES')GO TO 530
       IF(ANS.EQ.'NO')GO TO 550
       WRITE(IO,2)
       GO TO 522
 530   CALL EXWELL(SAVED,            PC,PUMPC.            )
       GO TO 545
 550   ANSWE='NO'
       CALL EXW(ANSWE,SAVED)
 545   WRITE(IO,301)
 540   READ(IN,1)ANS
       IF(ANS.EQ.'YES')GO TO 620
       IF(ANS.EQ.'NO')GO TO 2000
       WRITE(IO,2)
       GO TO 540
 600   READ(TF,501)N,S,IIII
       READ(TF,402)((LISTZO(I,J),J=1,22),I=1,N)
       READ(TF,502)(RESELV(I),I=1,N)
       READ(TF,503) ((DEM(I,J),J=1,S),I=1,N)
       READ(TF,504)PDD,PDFTR
       READ(TF,503)(SDAYS(I),I=1,S)
       READ(TF,601)            PC,PUMPC
 601   FORMAT(  I2,F8.3)
 620   WRITE(IO,621)
 621   FORMAT(//5X,'DO YOU HAVE ANY EXISTING SPRINGS IN YOUR '
      *'MODEL? <YES/NO>?')
 622   READ(IN,1)ANS
       IF(ANS.EQ.'YES')GO TO 630
       IF(ANS.EQ.'NO')GO TO 650
       WRITE(IO,2)
       GO TO 622
 630   CALL EXSPRG(SAVED            )
       GO TO 645
 650   ANSWE='NO'
       CALL EXS(ANSWE,SAVED)
 645   WRITE(IO,301)
 640   READ(IN,1)ANS
       IF(ANS.EQ.'YES')GO TO 720
       IF(ANS.EQ.'NO')GO TO 2000
       WRITE(IO,2)
       GO TO 640
 700   READ(TF,501)N,S,IIII
       READ(TF,402)((LISTZO(I,J),J=1,22),I=1,N)
       READ(TF,502)(RESELV(I),I=1,N)
```

```
      READ(TF,503) ((DEM(I,J),J=1,S),I=1,N)
      READ(TF,504)PDD,PDFTR
      READ(TF,503)(SDAYS(I),I=1,S)
      READ(TF,601)              PC,PUMPC

      GO TO(720,820,920,1020,1120)START-4
720   WRITE(IO,721)
721   FORMAT(//5X,'DO YOU HAVE ANY EXISTING TREATMENT PLANTS '
     *'IN YOUR MODEL?<YES/NO>?')
722   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 730
      IF(ANS.EQ.'NO')GO TO 750
      WRITE(IO,2)
      GO TO 722
730   CALL EXTRPL(SAVED                 )
      GO TO 745
750   ANSWE='NO'
      CALL EXT(ANSWE,SAVED)
745   WRITE(IO,301)
740   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 820
      IF(ANS.EQ.'NO')GO TO 2000
      WRITE(IO,2)
      GO TO 740
820   WRITE(IO,821)
821   FORMAT(//5X,'DO YOU HAVE EXISTING OR PROPOSED INTERZONAL'/
     *5X,'CONNECTIONS IN YOUR MODEL? <YES/NO>?')
822   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 830
      IF(ANS.EQ.'NO')GO TO 850
      WRITE(IO,2)
      GO TO 822
830   CALL CONECT(SAVED,PC,PUMPC,PDD,PDFTR)
      GO TO 845
850   ANSWE='NO'
      CALL CON(ANSWE,SAVED)
845   WRITE(IO,301)
840   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 920
      IF(ANS.EQ.'NO')GO TO 2000
      WRITE(IO,2)
      GO TO 840
920   WRITE(IO,921)
921   FORMAT(//5X,'DO YOU HAVE PROPOSED FUTURE WELLS IN YOUR '
     *'MODEL? <YES/NO>?')
922   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 930
      IF(ANS.EQ.'NO')GO TO 950
      WRITE(IO,2)
      GO TO 922
930   CALL FWELLS(SAVED                                 )
      GO TO 945
950   ANSWE='NO'
      CALL FWE(ANSWE,SAVED)
945   WRITE(IO,301)
940   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 1020
      IF(ANS.EQ.'NO')GO TO 2000
      WRITE(IO,2)
      GO TO 940
1020  WRITE(IO,1021)
1021  FORMAT(//5X,'DO YOU HAVE PROPOSED FUTURE SPRINGS IN '
     *'YOUR MODEL? <YES/NO>?')
1022  READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 1030
      IF(ANS.EQ.'NO')GO TO 1050
      WRITE(IO,2)
      GO TO 1022
1030  CALL FSPNGS(SAVED                      )
      GO TO 1045
1050  ANSWE='NO'
      CALL FSP(ANSWE,SAVED)
1045  WRITE(IO,301)
1040  READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 1120
      IF(ANS.EQ.'NO')GO TO 2000
      WRITE(IO,2)
      GO TO 1040
1120  WRITE(IO,1121)
1121  FORMAT(//5X,'DO YOU HAVE PROPOSED FUTURE TREATMENT'/
     *5X,'PLANTS IN YOUR MODEL? <YES/NO>?')
1122  READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 1130
      IF(ANS.EQ.'NO')GO TO 1150
      WRITE(IO,2)
      GO TO 1122
1130  CALL FTRPLS(SAVED    .       )
      GO TO 1200
1150  ANSWE='NO'
      CALL FTP(ANSWE,SAVED)
1200  CALL QUEST(IIII)
2000  LOCK IF
      LOCK TF
4000  CALL EXIT
      END
```

MATRIX - BINDING SUBROUTINES

```
HOST IS OBJECT/MAINPROGRAM;
BIND DATA FROM CANDE/DATA;
BIND ZONENM FROM CANDE/ZONENM;
BIND POPUL FROM CANDE/POPUL;
BIND EXWELL FROM CANDE/EXWELL;
BIND EXSPRG FROM CANDE/EXSPRG;
BIND EXTRPL FROM CANDE/EXTRPL;
BIND CONECT FROM CANDE/CONECT;
BIND FWELLS FROM CANDE/FWELLS;
BIND FSPNGS FROM CANDE/FSPNGS;
BIND FTRPLS FROM CANDE/FTRPLS;
BIND QUEST FROM CANDE/QUEST;
```

ZONE NUMBER AND NAME SUBROUTINE

```
      $SET SEPARATE
            SUBROUTINE ZONENM(SAVED)
            DIMENSION NAME(30), LISTZO(40,22),NUM(10)
            INTEGER ROW,TF
            COMMON IN,IO,IF,TF,LISTZO,N
            DATA  NUM/'0','1','2','3','4','5','6','7','8','9'/
            NN = '    '
            INAME = '9    '
            DO 80 I=1,40
            DO 80 J=1,22
      80    LISTZO(I,J)='    '
            WRITE(IO,90)
      90    FORMAT(///10X,10('*'),' SEGMENT 1 ',10('*'))
            WRITE(IO,100)
      100   FORMAT(//5X,'ENTER EACH ZONE NUMBER AND ZONE NAME FOLLOWED BY A'/
           * 5X,'RETURN.  ENTER A ZONE NUMBER OF 99 WHEN COMPLEATED'/
           * 10X,'EXAMPLE: 01 UPPER LOGAN  <RETURN>?'//!/)
      105   READ(IN,110)  NAME
      110   FORMAT(30A1)
            DO 120 I=1,29
            IF(NAME(I).IS.NN)GO TO 120
            IF(NAME(I).IS.INAME .AND. NAME(I+1).IS.INAME)GO TO 150
            IF(NAME(I+1).NE.NN)GO TO 140
      116   IR = IR + 1
            LISTZO(IR,1) = '0'
            LISTZO(IR,2) = NAME(I)
            GO TO 130
      120   CONTINUE
            GO TO 105
      140   DO 143 LL=1,10
            IF(NAME(I+1) .EQ. NUM(LL))GO TO 145
      143   CONTINUE
            GO TO 116
      145   IR = IR + 1
            LISTZO(IR,1) = NAME(I)
            LISTZO(IR,2) = NAME(I+1)
            I = I + 1
      130   M = I + 1
            DO 160 I=M,30
            II = I
            IF(NAME(I).IS.NN)GO TO 160
            GO TO 165
      160   CONTINUE
      165   DO 168 J=II,29
            JJ = J
            IF(NAME(J).IS.NN .AND. NAME(J+1).IS.NN)GO TO 180
      168   CONTINUE
      180   K = 2
            DO 170 L=II,JJ
            K = K + 1
            IF(K.GT.20)GO TO 171
      170   LISTZO(IR,K) = NAME(L)
      171   DO 200 I=1,30
      200   NAME(I) = '    '
            I = IR
            IF(ROW .NE. 0) GO TO 155
            GO TO 105
      150   WRITE(IO,151)
      151   FORMAT(//  5X,'THE FOLLOWING IS A LIST OF YOUR INPUT',
           * ' OF ZONES AND THEIR NAMES.'  //)
            WRITE(IO,152)
      152   FORMAT(2X,'ROW', 2X,'ZONE',6X,'**** ZONE NAME ****'  /)
            DO 155 I=1,IR
      155   WRITE(IO,156) I, (LISTZO(I,J),J=1,22)
      156   FORMAT(3X,I2,3X,2A1,8X,20A1   )
            IF (ROW .NE. 0) GO TO 197
            N=IR
            WRITE(IO,190)
      190   FORMAT(////5X,'ARE THERE ANY CHANGES REQUIRED IN THIS DATA?'
           * 1X,'<YES/NO>?'  )
      193   READ(IN,191) ANS
      191   FORMAT(A6)
            IF(ANS.EQ.'NO' .AND. ROW.EQ.0)GO TO 205
            IF(ANS.EQ.'NO' .AND. ROW.NE.0)GO TO 199
            IF(ANS.EQ.'YES')GO TO 192
            WRITE(IO,194)
      194   FORMAT(5X,'PLEASE ANSWER YES OR NO?'  )
            GO TO 193
      192   WRITE(IO,195)
      195   FORMAT(5X,'ENTER ROW OF CHANGE?' )
            READ(IN,/) ROW
            IF (ROW.LE.0)GO TO 197
            IF(ROW.GT.N)N=N+1
            IF(ROW.GT.N)ROW=N
            IR=ROW-1
            WRITE(IO,196)
      196   FORMAT(5X,'ENTER THE NEW ZONE AND NAME?')
            DO 215 J=1,22
      215   LISTZO(ROW,J)='    '
            GO TO 105
      197   WRITE(IO,198)
      198   FORMAT(5X,'MORE CHANGES? <YES/NO>?' )
            GO TO 193
      199   WRITE (IO,201)
      201   FORMAT(5X,'DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>?')
      202   READ(IN,191)ANS
            IF(ANS.EQ.'NO')GO TO 205
            IF(ANS.EQ.'YES')GO TO 206
            WRITE(IO,194)
            GO TO 202
      206   IR = N
            ROW=0
            GO TO 150
      205   WRITE(TF,300)N
      300   FORMAT(I2)
            WRITE(TF,310) ((LISTZO(I,J),J=1,22),I=1,N)
      310   FORMAT(110A1)
            WRITE(IF,320)
      320   FORMAT('START OF MODELDATA FILE')
            SAVED=1
            RETURN
            END
```

SEASON AND ZONE DEMAND SUBROUTINE

```
      $SET SEPARATE
            SUBROUTINE POPUL(SAVED,PDFTR)
            DIMENSION POP(40), DEM(40,4), RESELV(40), NDEM(4), NS(4)
            DIMENSION PDEM(40),SDAYS(4),STERM(4,15),TEMP(16)
            DIMENSION LISTZO(40,22)
            INTEGER ROW, S, TF,SAVED,PDD
            COMMON IN,IO,IF,TF,LISTZO,N,S
            COMMON /CMN2/ RESELV,DEM,SDAYS
            COMMON /CMN3/ DM1,PC1,PD1,PD2,DM2
            DATA NDEM/4*'DEMAND'/
            DATA NS/'S1','S2','S3','S4'/
            IF(PDD.EQ.0)PDFTR=PD2
            DO 60 I=1,4
            DO 60 J=1,15
      60    STERM(I,J)='  '
            WRITE(IO,90)
      90    FORMAT(///10X,10('*'),' SEGMENT 2 ',10('*'))
            WRITE(IO,205)
      205   FORMAT(////5X,'ENTER THE NUMBER OF SEASONS TO BE CONSIDERED',
           *' IN YOUR MODEL. <1 - 4>?')
      206   READ(IN,/)S
            IF(S.GE.1.AND.S.LE.4)GO TO 208
            WRITE(IO,207)
      207   FORMAT(5X,'PLEASE ENTER  1 - 4  ONLY.?')
            GO TO 206
      208   DO 215 I=1,S
      209   WRITE(IO,216)I
      216   FORMAT(5X,'ENTER THE NUMBER OF DAYS IN SEASON ',I1,'.?')
            READ(IN,/)SDAYS(I)
            IF(SDAYS(I).GE.1 .AND. SDAYS(I).LE.365)GO TO 211
            WRITE(IO,212)SDAYS(I)
      212   FORMAT(5X,'UNACCEPTABLE NUMBER OF DAYS.',I6)
            GO TO 209
      211   WRITE(IO,217)I
      217   FORMAT(5X,'ENTER THE MONTH TERM (EX: JAN 15 - MAR 31) FOR SEASON '
           * I1,' (15 MAX).?')
            READ(IN,218)TEMP
      218   FORMAT(16A1)
            K=0
            DO 219 J=1,15
            IF(TEMP(J).IS.' ' .AND. TEMP(J+1).IS.' ')GO TO 219
            K=K+1
            STERM(I,K)=TEMP(J)
      219   CONTINUE
            IF(SS.NE.0)GO TO 115
      215   CONTINUE
      99    WRITE(IO,100)
      100   FORMAT(///5X,'SEASON    DAYS    TERM')
            DO 110 I=1,S
      115   WRITE(IO,120)I,SDAYS(I),(STERM(I,J),J=1,15)
      120   FORMAT(8X,I1,6X,I3,3X,15A1)
            IF(SS.NE.0)GO TO 180
      110   CONTINUE
            WRITE(IO,291)
      130   READ(IN,293)ANS
            IF(ANS.IS.'YES')GO TO 140
            IF(ANS.IS.'NO')GO TO 190
            WRITE(IO,294)
            GO TO 130
      140   WRITE(IO,150)
      150   FORMAT(5X,'ENTER SEASON OF CHANGE.?')
      160   READ(IN,/)SS
            IF(SS.GE.1 .AND. SS.LE.S)GO TO 170
            WRITE(IO,207)
            GO TO 160
      170   I=SS
            GO TO 209
      180   WRITE(IO,299)
      182   READ(IN,293)ANS
            IF(ANS.IS.'YES')GO TO 140
            IF(ANS.IS.'NO')GO TO 185
            WRITE(IO,294)
            GO TO 182
      185   SS=0
            GO TO 99
      190   WRITE(IO,210)
      210   FORMAT(//5X,'NOW ENTER THE POPULATION OF EACH ZONE,'/
           *5X,'THE RESERVOIR ELEVATION IN FEET FROM MSL AND '/
           *5X,'THE DEMAND PER PERSON PER SEASON IN GALLONS'/
           *5X,'PER DAY ALL SEPERATED BY COMMAS.'////)
            DO 220 I=1,N
            WRITE(IO,230) (LISTZO(I,J),J=1,22)
      230   FORMAT(/3X,5('*'),2A1,2X,20A1,5('*')'?')
            READ(IN,/) POP(I), RESELV(I), (DEM(I,J),J=1,S)
      220   CONTINUE
      235   WRITE(IO,240)
      240   FORMAT(///5X,'THE FOLLOWING IS A LIST OF YOUR INPUT DATA.'//)
            WRITE(IO,250) (NDEM(I),I=1,S)
      250   FORMAT(23X,'RESERVOIR',2X,4(A6,2X))
            WRITE(IO,260) (NS(I),I=1,S)
      260   FORMAT(1X,'ROW',1X,'ZONE',2X,'POPULATION',2X,'ELEVATION',
           *2X,4(2X,A2,4X))
            DO 280 I=1,N
      280   WRITE(IO,290) I,(LISTZO(I,J),J=1,2),POP(I),RESELV(I),
           *(DEM(I,J),J=1,S)
      290   FORMAT(2X,I2,2X,2A1,3X,I8,6X,I5,5X,4(F6.1,2X))
            WRITE(IO,291)
      291   FORMAT(////5X,'ARE THERE ANY CHANGES REQUIRED IN THIS DATA?'
           *' <YES/NO>?' )
      292   READ(IN,293) ANS
      293   FORMAT(A6)
            IF(ANS.EQ.'NO' .AND. ROW.EQ.0)GO TO 700
            IF(ANS.EQ.'NO' .AND. ROW.NE.0)GO TO 310
            IF(ANS.EQ.'YES')GO TO 295
            WRITE(IO,294)
      294   FORMAT(5X,'PLEASE ANSWER YES OR NO.?')
            GO TO 292
      295   WRITE(IO,296)
      296   FORMAT(5X,'ENTER ROW OF CHANGE.?')
            READ(IN,/) ROW
            IF(ROW.LE.0 .OR. ROW.GT.N)GO TO 340
            WRITE(IO,297)
      297   FORMAT(5X,'ENTER NEW POPULATION, RESERVOIR ELEVATION AND',
           *' DEMANDS.?')
            I = ROW
            READ(IN,/) POP(I),RESELV(I),(DEM(I,J),J=1,S)
            WRITE(IO,290) I,(LISTZO(I,J),J=1,2),POP(I),RESELV(I),
           *(DEM(I,J),J=1,S)
      298   WRITE(IO,299)
```

```
 299   FORMAT(5X,'MORE CHANGES? <YES/NO>??')
       GO TO 292
 340   WRITE(IO,341)
 341   FORMAT(5X'UNACCEPTABLE ROW NUMBER.')
       GO TO 295
 310   WRITE(IO,320)
 320   FORMAT(5X,'DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>??')
 321   READ(IN,293)ANS
       IF(ANS.EQ.'NO')GO TO 700
       IF(ANS.EQ.'YES')GO TO 326
       WRITE(IO,294)
       GO TO 321
 326   ROW = 0
       GO TO 235
 700   WRITE(IO,710)PDFTR
 710   FORMAT(//5X,'THE STANDARD PEAK DAY DEMAND IS 'F4.2' TIMES '
      *'GREATER THAN THE PEAK'/5X,'SEASON DAILY DEMAND.  IS THIS'
      *' ACCEPTABLE FOR ALL ZONES? <YES/NO>?')
 720   READ(IN,293)ANS
       IF(ANS.EQ.'YES')GO TO 400
       IF(ANS.EQ.'NO')GO TO 730
       WRITE(IO,294)
       GO TO 720
 730   WRITE(IO,740)
 740   FORMAT(5X,'IS THERE A CONSTANT VALUE THAT IS ACCEPTABLE?'
      *' <YES/NO>?')
 750   READ(IN,293)ANS
       IF(ANS.EQ.'YES')GO TO 760
       IF(ANS.EQ.'NO')GO TO 401
       WRITE(IO,294)
       GO TO 750
 760   WRITE(IO,770)
 770   FORMAT(5X,'ENTER PEAK DAY MULTIPLIER CONSTANT.?')
       READ(IN,/)PDFTR
       IF(PDFTR.GE.1)GO TO 400
       WRITE(IO,775)
 775   FORMAT(5X,'A PEAK DAY MULTIPLIER LESS THAN 1 FOR'/
      *5X,'DEMANDS IS UNACCEPTABLE.')
       GO TO 760
 400   PDD=1
 401   X=PDFTR
       DO 410 I=1,N
       IF(PDD.EQ.1)GO TO 405
 407   WRITE(IO,402) (LISTZO(I,K),K=1,2)
 402   FORMAT(5X,'ENTER PEAK DAY MULTIPLIER FOR ZONE ',2A1,
      *'.?')
       READ(IN,/)X
       IF(X.GE.1)GO TO 405
       WRITE(IO,775)
       GO TO 407
 405   PDEM(I)=DEM(I,1)*POP(I)*X/1.0E6
       DO 410 J=1,S
 410   DEM(I,J)=DEM(I,J)*SDAYS(J)*POP(I)/1.0E6
       REWIND(IF)
       WRITE(IF,510)
 510   FORMAT('ELEMENT DEMAND')
       WRITE(IF,520)
 520   FORMAT(4X,'TABLE DEMAND')
       WRITE(IF,530) (NDEM(I),I,I=1,S)
 530   FORMAT(9X,'*',4X,'PEAKDEM',4(3X,A6,I1))
       DO 540 I=1,N
 540   WRITE(IF,550) (LISTZO(I,J),J=1,2),PDEM(I),(DEM(I,J),J=1,S)
 550   FORMAT(9X,2A1,3X,F7.2,4(3XI7))
       WRITE(IF,551)
 551   FORMAT(4X,'LIST(ZO),T=20')
       DO 552 I=1,N
 552   WRITE(IF,553)(LISTZO(I,J),J=1,22)
 553   FORMAT(9X,2A1,5X,'"',20A1,'"')
       WRITE(IF,620)
       WRITE(IF,560)
 560   FORMAT('ELEMENT SEASON')
       WRITE(IF,570)
 570   FORMAT(4X,'TABLE SEASONS,ZERO')
       WRITE(IF,575)
 575   FORMAT(9X,'*   DAYS')
       WRITE(IF,580) (I,SDAYS(I),I=1,S)
 580   FORMAT(9X,I1,3X,I4)
       WRITE(IF,590)
 590   FORMAT(4X,'LIST(S),T=15')
       DO 600 I=1,S
 600   WRITE(IF,610) I,(STERM(I,J),J=1,15)
 610   FORMAT(9X,I1,3X,'"',15A1,'"')
       WRITE(IF,620)
 620   FORMAT('ENDATA')
       REWIND(TF)
       WRITE(TF,630)N,S,IIII
 630   FORMAT(3I6)
       WRITE(TF,640) ((LISTZO(I,J),J=1,22),I=1,N)
 640   FORMAT(110A1)
 645   WRITE(TF,650) (RESELV(I),I=1,N)
 650   FORMAT(20I5)
       WRITE(TF,660) ((DEM(I,J),J=1,S),I=1,N)
 660   FORMAT(16I7)
       WRITE(TF,670)PDD,PDFTR
 670   FORMAT(I2,F8.4)
       WRITE(TF,660)(SDAYS(I),I=1,S)
       SAVED=1
       RETURN
       END
```

EXISTING WELL SUBROUTINE

```
      $SET SEPARATE
            SUBROUTINE EXWELL(SAVED                    PC,PUMPC          )
            DIMENSION WELL(90,8),NWELL(4),LISTZO(40,22),WELELV(90)
            DIMENSION RESELV(40),ALTER(10)
            INTEGER S,ROW,COL,TF,OMP,PC,SAVED
            COMMON IN,IO,IF,TF,LISTZO,N,S
            COMMON /CMN2/ RESELV
            COMMON /CMN3/ OM1,PC1,PD1
            DATA NWELL/4*'WELL'/
            DATA ALTER/'A','B','C','D','E','F','G','H','I','J'/
                          OANDMP=OM1
                         PUMPC=PC1
                          PDFTOR=PD1
            WRITE(IO,90)
      90    FORMAT(///17X,10('*'),' SEGMENT 3 ',10('*'))
            WRITE(IO,100)
      100   FORMAT(///12X,5('*'),' ENTER EXISTING WELL INFORMATION ',
           *5('*')/)
            WRITE(IO,105)
      105   FORMAT(//5X'[NOTE] FOR MODEL SIMPLICITY IT IS ADVISIBLE TO GROUP'/
           *5X,'ALL EXISTING WELLS IN A ZONE AND INPUT AS ONE SOURCE'/
           *5X,'IF POSSIBLE.'////)
            DO 130 I=1,N
            WRITE(IO,120) (LISTZO(I,J),J=1,22)
      120   FORMAT(///3X,5('*'),2A1,2X,20A1,5('*')//)
      114   WRITE(IO,115)
      115   FORMAT(5X,'ENTER THE NUMBER OF EXISTING WELLS (OR GROUPS) IN'/
           *5X,'THIS ZONE. <0 - 10>?')
            READ(IN,/)NW
            IF(NW)126,130,1128
      126   WRITE(IO,127)
      127   FORMAT(5X,'PLEASE ENTER A NON-NEGATIVE NUMBER.')
            GO TO 114
      1128  IF(NW.GT.10)GO TO 114
      128   DO 129 K=1,NW
            WRITE(IO,110)(LISTZO(I,J),J=1,2),ALTER(K)
      110   FORMAT(5X,'ENTER WELL ELEVATION FROM  MSL  AND WELL CAPACITY'/
           *5X,'<FEET,GPM> FOR ZONE '2A1' WELL  "'A1'".?')
      125   READ(IN,/)X,Y
            IF(Y.LT.0)GO TO 140
            IF(Y.EQ.0)GO TO 130
            II=II+1
            WELELV(II)=X
            WELL(II,1)=LISTZO(I,1)
            WELL(II,2)=LISTZO(I,2)
            WELL(II,7)=I
            WELL(II,8)=ALTER(K)
            WELL(II,3)=Y
            WELL(II,4)=0.00144*Y
      129   CONTINUE
      130   CONTINUE
            III=II
            GO TO 180
      140   WRITE(IO,150)Y
      150   FORMAT(5X,'YOU HAVE ENTERED A NEGATIVE WELL CAPACITY ',F6.1/
           *5X,'PLEASE REENTER WELL ELEVATION AND CAPACITY.?')
            GO TO 125
      180   WRITE(IO,190)PDFTOR
      190   FORMAT(//5X,'THE STANDARD PEAK DAY SUPPLY IS ',F4.2,' TIMES'/
           *5X,'THE PEAK SEASONAL DAILY CAPACITY.'/
           *5X,'IS THIS ACCEPTABLE FOR ALL YOUR ZONES? <YES/NO>?'////)
      720   READ(IN,1)ANS
            IF(ANS.EQ.'YES')GO TO 200
            IF(ANS.EQ.'NO')GO TO 730
            WRITE(IO,2)
            GO TO 720
      730   WRITE(IO,740)
      740   FORMAT(5X,'IS THERE A CONSTANT VALUE THAT IS ACCEPTABLE?'
           *' <YES/NO>?')
      750   READ(IN,1)ANS
            IF(ANS.EQ.'YES')GO TO 760
            IF(ANS.EQ.'NO')GO TO 201
            WRITE(IO,2)
            GO TO 750
      760   WRITE(IO,770)
      770   FORMAT(5X,'ENTER PEAK DAY MULTIPLIER CONSTANT.?')
            READ(IN,/)PDFTOR
            IF(PDFTOR.LE.1)GO TO 200
            WRITE(IO,775)
      775   FORMAT(5X,'A PEAK DAY MULTIPLIER GREATER THAN 1 FOR'/
           *5X,'SUPPLY IS UNACCEPTABLE.')
            GO TO 760
      200   NPD=1
            Z=PDFTOR
      201   DO 202 I=1,II
            IF(NPD.EQ.1)GO TO 205
      207   WRITE(IO,203) (WELL(I,K),K=1,2),WELL(I,8)
      203   FORMAT(5X,'ENTER PEAK DAY MULTIPLIER FOR ZONE '2A1' WELL  "'A1
           *'".?')
            READ(IN,/)Z
            IF(Z.LE.1)GO TO 205
            WRITE(IO,775)
            GO TO 207
      205   WELL(I,5)=WELL(I,4)*Z
      202   CONTINUE
            WRITE(IO,210)OANDMP
      210   FORMAT(//5X,'THE STANDARD OPERATION AND MAINTENANCE WELL'/
           *5X,'COSTS FOR THIS MODEL ARE ',$F6.2,'/KG. (FOR PUMPS,'/
           *5X,'PIPELINE, ECT.)  IS THIS ACCEPTABLE FOR ALL YOUR'/
           *5X,'ZONES? <YES/NO>?'////)
      215   READ(IN,1)ANS
      1     FORMAT(A6)
            IF(ANS.EQ.'YES')GO TO 220
            IF(ANS.EQ.'NO')GO TO 230
            WRITE(IO,2)
      2     FORMAT(5X,'PLEASE ANSWER YES OR NO?')
            GO TO 215
      230   WRITE(IO,240)
      240   FORMAT(5X,'IS THERE A CONSTANT VALUE THAT IS ACCEPTABLE?'
           *' <YES/NO>?')
      245   READ(IN,1)ANS
            IF(ANS.EQ.'YES')GO TO 250
            IF(ANS.EQ.'NO')GO TO 270
            WRITE(IO,2)
            GO TO 245
      250   WRITE(IO,260)
      260   FORMAT(5X,'ENTER O&M CONSTANT.?')
            READ(IN,/)OANDMP
            IF(OANDMP.LE.0)GO TO 250
```

```
220   OMP=1
270   WRITE(IO,280)PUMPC
280   FORMAT(//5X,'THE STANDARD POWER COSTS FOR PUMPING IN THIS'/
     *5X,'MODEL ARE ',$F6.2,'/MG/100FT.  IS THIS ACCEPTABLE '/
     *5X,'FOR ALL YOUR ZONES? <YES/NO>?'////)
290   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 340
      IF(ANS.EQ.'NO')GO TO 300
      WRITE(IO,2)
      GO TO 290
300   WRITE(IO,240)
310   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 320
      IF(ANS.EQ.'NO')GO TO 350
      WRITE(IO,2)
      GO TO 310
320   WRITE(IO,330)
330   FORMAT(5X,'ENTER POWER PUMP COSTS.?')
      READ(IN,/)PUMPC
      IF(PUMPC.LE.0)GO TO 320
340   PC=1
      X=OANDMP
      Y=PUMPC
350   DO 410 I=1,II
356   IF(OMP.EQ.1)GO TO 380
355   WRITE(IO,360) (WELL(I,K),K=1,2),WELL(I,8)
360   FORMAT(5X,'ENTER O&M COSTS FOR ZONE '2A1' WELL  "'A1'".?')
      READ(IN,/)X
      IF(X.LE.0)GO TO 355
      IF(ROW.GT.0)GO TO 385
380   IF(PC.EQ.1)GO TO 390
385   WRITE(IO,370) (WELL(I,K),K=1,2),WELL(I,8)
370   FORMAT(5X,'ENTER PUMP POWER COSTS FOR ZONE '2A1' WELL  "'A1'".?')
      READ(IN,/)Y
      IF(Y.LE.0)GO TO 385
390   DELELV=RESELV(WELL(I,7))-WELELV(I)
      IF(DELELV)421,421,422
422   WELL(I,6)=Y*DELELV/100. + X
      GO TO 405
421   WELL(I,6)=X
405   IF(ROW.NE.0)GO TO 451
410   CONTINUE
420   WRITE(IO,430)
430   FORMAT(///5X,'THE FOLLOWING IS A LIST OF YOUR DATA.'//)
      WRITE(IO,440) (I,I=1,5)
440   FORMAT(17X,'COL',5X,'COL',2(7X,'COL'),8X,'COL'/
     *18X,I1,7X,I1,2(9X,I1),10X,I1)
      WRITE(IO,450)
450   FORMAT(17X,'WELL',4X,'WELL',6X,'WELL',4X,'PEAK DAY',
     *4X,'TOTAL'/1X,'ROW',2X,'ZONE',2X,'ALT',2X,'ELEV',3(2X,'CAPACITY'),
     *4X,'COST'/17X,'FEET',4X,'GPM',2(7X,'MGD'),7X,'$/MG')
      DO 451 KJ=1,II
451   WRITE(IO,452)KJ,(WELL(KJ,J),J=1,2),WELL(KJ,8),WELELV(KJ),(WELL(KJ,
     *J),J=3,6)
452   FORMAT(2X,I2,3X,2A1,4X,A1,2X,I5,3X,I5,2(5X,F5.2),4X,$F7.2)
      IF(ROW.NE.0)GO TO 560
      WRITE(IO,460)
460   FORMAT(////5X,'ARE THERE ANY CHANGES REQUIRED IN THIS DATA?'
     *' <YES/NO>?')
470   READ(IN,1)ANS
      IF(ANS.EQ.'NO' .AND. ROW.EQ.0)GO TO 800
      IF(ANS.EQ.'NO' .AND. ROW.NE.0)GO TO 480
      IF(ANS.EQ.'YES')GO TO 520
      WRITE(IO,2)
      GO TO 470
480   WRITE(IO,490)
490   FORMAT(5X,'DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>?')
500   READ(IN,1)ANS
      IF(ANS.EQ.'NO')GO TO 800
      IF(ANS.EQ.'YES')GO TO 510
      WRITE(IO,2)
      GO TO 500
510   ROW = 0
      COL = 0
      II = III
      GO TO 420
520   WRITE(IO,530)
530   FORMAT(5X,'ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.?')
      READ(IN,/)ROW,COL
      IF(ROW.GT.0 .AND. ROW.LE.III .AND. COL.GT.0 .AND. COL.LE.5)
     *GO TO 550
      WRITE(IO,540)
540   FORMAT(5X,'UNACCEPTABLE ROW OR COLUMN.')
      GO TO 520
550   KJ = ROW
      II = ROW
      K=WELL(ROW,7)
      I=ROW
      GO TO( 600,620,620,640,355)COL
560   WRITE(IO,570)
570   FORMAT(5X,'MORE CHANGES? <YES/NO>?')
      GO TO 470
600   WRITE(IO,610)
610   FORMAT(5X,'ENTER NEW WELL ELEVATION <FEET>.?')
      READ(IN,/)WELELV(ROW)
      ROW=-1
      GO TO 356
620   WRITE(IO,630)
630   FORMAT(5X,'ENTER NEW WELL CAPACITY <GPM>.?')
      READ(IN,/)WELL(ROW,3)
      WELL(ROW,4)=WELL(ROW,3)*0.00144
      WELL(ROW,5)=WELL(ROW,4)*PDFTOR
      GO TO 451
640   WRITE(IO,203)(WELL(ROW,K),K=1,2)
      READ(IN,/)X
      IF(X.LE.1)GO TO 650
      WRITE(IO,775)
      GO TO 640
650   WELL(ROW,5)=X*WELL(ROW,4)
      GO TO 451
      ENTRY EXW(ANSWE,SAVED)
800   IF(SAVED.EQ.1)GO TO 801
      INQUIRE(IF,LASTRECORD=N2)
      N2=N2+2
      WRITE(IF=N2,810)
      GO TO 802
801   WRITE(IF,810)
810   FORMAT('ELEMENT EXWELL')
802   WRITE(IF,820)
820   FORMAT(4X,'TABLE EXSTWELL,ZERO')
      WRITE(IF,830)
830   FORMAT(9X,'*',5X,'FLOW',5X,'PDFLOW',5X,'COST')
```

```
      IF(ANSWE.EQ.'NO')GO TO 855
      DO 840 I=1,III
840   WRITE(IF,850)(WELL(I,J),J=1,2),WELL(I,8),(WELL(I,J),J=4,6)
850   FORMAT(9X,3A1,2X,F5.2,6X,F5.2,3X,F6.2)
855   IF(ANSWE.IS.'NO')WRITE(IF,856)
856   FORMAT(9X,'DUM')
      WRITE(IF,860)
860   FORMAT('ENDATA')
      IF(SAVED.EQ.1)GO TO 865
      INQUIRE(TF,LASTRECORD=N2)
      N2=N2+2
      WRITE(TF=N2,870)              'C,PUMPC
      GO TO 880
865   WRITE(TF,870)              PC,PUMPC
870   FORMAT(  I2,F8.3)
880   SAVED=1
      RETURN
      END
```

EXISTING SPRING SUBROUTINE

```
$SET SEPARATE
      SUBROUTINE EXSPRG(SAVED                              )
      DIMENSION SPRNG(40,13),NSPRNG(4),LISTZO(40,22),NS(4)
      DIMENSION TEMP(4),COLN(4),FLOW(4),CSF(4)
      INTEGER ROW,COL,S,TF,SAVED,OMS
      COMMON IN,IO,IF,TF,LISTZO,N,S
      COMMON /CMN3/ OM1,PC1,PD1,PD2,OM2
      DATA NS/'S1','S2','S3','S4'/
      DATA NSPRNG/4*'SPRING'
      DATA COLN/4*'COL'/,FLOW/4*'FLOW'/,CSF/4*'CSF'/
                        'DFTOR=PD1
                        OANDMS=OM2
      WRITE(IO,90)
90    FORMAT(////19X,10('*'),' SEGMENT 4 ',10('*'))
      WRITE(IO,100)
100   FORMAT(///12X,5('*'),' ENTER EXISTING SPRING INFORMATION ',
     *5('*')////)
      WRITE(IO,105)
105   FORMAT(5X'(NOTE) FOR MODEL SIMPLICITY ALL EXISTING SPRINGS IN A'/
     *5X'ZONE MUST BE GROUPED AS ONE SOURCE.  SINCE SPRING FLOWS ARE'/
     *5X'USUALLY THE LEAST UNIT COST SOURCE AND THE FIRST TO BE USED'/
     *5X'THIS WILL NOT CHANGE YOUR MODEL SOLUTION.'////)
      WRITE(IO,110)
110   FORMAT(5X,'ENTER EXISTING SPRING FLOWS <CSF> FOR EACH SEASON'/
     *5X,'SEPARATED BY COMMAS.  ENTER ZERO FOR ALL SEASONS IF A'/
     *5X,'ZONE DOES NOT HAVE EXISTING SPRINGS.'////)
      DO 130 I=1,N
      K=0
      WRITE(IO,120)(LISTZO(I,J),J=1,22)
120   FORMAT(3X,5('*'),2A1,2X,20A1,5('*'),'?')
      READ(IN,/)(TEMP(J),J=1,S)
      II=II+1
      SPRNG(II,1)=LISTZO(I,1)
      SPRNG(II,2)=LISTZO(I,2)
      DO 125 J=1,S
      IF(TEMP(J).LE.0)GO TO 125
      K=K+1
      SPRNG(II,J+2)=TEMP(J)
      SPRNG(II,J+6)=0.6463*TEMP(J)
125   CONTINUE
      IF(K.NE.0)GO TO 130
      II=II-1
130   CONTINUE
      III=II

180   WRITE(IO,190)PD,OR
190   FORMAT(//5X,'THE STANDARD PEAK DAY SUPPLY IS ',F4.2,' TIMES'/
     *5X,'THE PEAK SEASONAL DAILY CAPACITY.'/
     *5X,'IS THIS ACCEPTABLE FOR ALL YOUR ZONES? <YES/NO>?'////)
720   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 200
      IF(ANS.EQ.'NO')GO TO 730
      WRITE(IO,2)
      GO TO 720
730   WRITE(IO,740)
740   FORMAT(5X,'IS THERE A CONSTANT VALUE THAT IS ACCEPTABLE?'
     *' <YES/NO>?')
750   READ(IN,1)ANS
```

```
      IF(ANS.EQ.'YES')GO TO 760
      IF(ANS.EQ.'NO')GO TO 201
      WRITE(IO,2)
      GO TO 750
760   WRITE(IO,770)
770   FORMAT(5X,'ENTER PEAK DAY MULTIPLIER CONSTANT.?')
      READ(IN,/)PDFTOR
      IF(PDFTOR.LE.1)GO TO 200
      WRITE(IO,775)
775   FORMAT(5X,'A PEAK DAY MULTIPLIER GREATER THAN 1 FOR'/
     *5X,'SUPPLY IS UNACCEPTABLE.')
      GO TO 760
200   NPD=1
201   X=PDFTOR
      DO 202 I=1,III
      IF(NPD.EQ.1)GO TO 205
207   WRITE(IO,203) (SPRNG(I,K),K=1,2)
203   FORMAT(5X,'ENTER PEAK DAY MULTIPLIER FOR ZONE ',2A1,
     *'.?')
      READ(IN,/)X
      IF(X.LE.1)GO TO 205
      WRITE(IO,775)
      GO TO 207
205   SPRNG(I,11)=X*SPRNG(I,7)
202   CONTINUE
      WRITE(IO,210)OANDMS
210   FORMAT(//5X,'THE STANDARD OPERATION AND MAINTENANCE SPRING COSTS'/
     *5X,'FOR THIS MODEL ARE ',$F6.2,'/MG (FOR CHLORINATION,'/
     *5X,'DESANDER CLEANING AND MISC. O&M).  IS THIS ACCEPTABLE'/
     *5X,'FOR ALL YOUR ZONES? <YES/NO>?'////)
215   READ(IN,1)ANS
1     FORMAT(A6)
      IF(ANS.EQ.'YES')GO TO 220
      IF(ANS.EQ.'NO')GO TO 230
      WRITE(IO,2)
2     FORMAT(5X,'PLEASE ANSWER YES OR NO?')
      GO TO 215
230   WRITE(IO,240)
240   FORMAT(5X,'IS THERE A CONSTANT VALUE THAT IS ACCEPTABLE?'
     *' <YES/NO>?')
245   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 250
      IF(ANS.EQ.'NO')GO TO 270
      WRITE(IO,2)
      GO TO 245
250   WRITE(IO,260)
260   FORMAT(5X,'ENTER O&M CONSTANT.?')
      READ(IN,/)OANDMS
      IF(OANDMS.LE.0)GO TO 250
220   OMS=1
270   X=OANDMS
      DO 410 I=1,III
      IF(OMS.EQ.1)GO TO 370
380   WRITE(IO,360) (SPRNG(I,J),J=1,2)
360   FORMAT(5X,'ENTER O&M COSTS FOR ZONE ',2A1,'.?')
      READ(IN,/)X
      IF(X.LE.0)GO TO 380
370   SPRNG(I,12)=X
      IF(ROW.NE.0)GO TO 451
410   CONTINUE
420   WRITE(IO,430)
430   FORMAT(///5X,'THE FOLLOWING IS A LIST OF YOUR DATA.'//)
      WRITE(IO,440) (COLN(J),J=1,S)
440   FORMAT(12X,'COL',5X,'COL',6X,4(A3,5X))
      WRITE(IO,441)1,2,(I,I=3,S+2)
441   FORMAT(13X,I1,7X,I1,8X,4(I1,7X))
      WRITE(IO,442)(NSPRNG(I),I=1,S)
442   FORMAT(18X,'PEAK DAY',4(2X,A6))
      WRITE(IO,443)(FLOW(I),I=1,S)
443   FORMAT(1X,'ROW',2X,'ZONE',2X,'COST',3X,'SPRING',4(4X,A4))
      WRITE(IO,444)(NS(I),I=1,S)
444   FORMAT(12X,'$/MG',4X,'FLOW',4(6X,A2))
      WRITE(IO,445)(CSF(I),I=1,S)
445   FORMAT(20X,'MGD',7X,4(A3,5X))
      DO 451 KJ=1,II
451   WRITE(IO,452)KJ,(SPRNG(KJ,J),J=1,2),SPRNG(KJ,12),SPRNG(KJ,11),
     *(SPRNG(KJ,J),J=3,S+2)
452   FORMAT(2X,I2,3X,2A1,1X,$F6.2,2X,F6.2,1X,4(2X,F6.2))
      IF(ROW.NE.0)GO TO 560
      WRITE(IO,460)
460   FORMAT(////5X,'ARE THERE ANY CHANGES REQUIRED IN THIS DATA?'
     *' <YES/NO>?')
470   READ(IN,1)ANS
      IF(ANS.EQ.'NO' .AND. ROW.EQ.0)GO TO 800
      IF(ANS.EQ.'NO' .AND. ROW.NE.0)GO TO 480
      IF(ANS.EQ.'YES')GO TO 520
      WRITE(IO,2)
      GO TO 470
480   WRITE(IO,490)
490   FORMAT(5X,'DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>?')
500   READ(IN,1)ANS
      IF(ANS.EQ.'NO')GO TO 800
      IF(ANS.EQ.'YES')GO TO 510
      WRITE(IO,2)
      GO TO 500
510   ROW = 0
      COL = 0
      II = III
      GO TO 420
520   WRITE(IO,530)
530   FORMAT(5X,'ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.?')
      READ(IN,/)ROW,COL
      IF(ROW.GT.0 .AND. ROW.LE.III .AND. COL.GT.0 .AND. COL.LE.S+2)
     *GO TO 550
      WRITE(IO,540)
540   FORMAT(5X,'UNACCEPTABLE ROW OR COLUMN.')
      GO TO 520
550   KJ = ROW
      II = ROW
      GO TO (620,600,610,610,610,610)COL
560   WRITE(IO,570)
570   FORMAT(5X,'MORE CHANGES? <YES/NO>?')
      GO TO 470
600   WRITE(IO,203)(SPRNG(ROW,J),J=1,2)
      READ(IN,/)X
      IF(X.LE.1)GO TO 605
      WRITE(IO,775)
      GO TO 600
605   SPRNG(ROW,11)=SPRNG(ROW,7)*X
      GO TO 451
610   WRITE(IO,611)(SPRNG(ROW,J),J=1,2),COL-2
611   FORMAT(5X,'ENTER NEW SPRING FLOW FOR ZONE ',2A1,' SEASON ',
```

```
     *I1,'.?')
      READ(IN,/)X
      IF(X.LT.0)GO TO 610
      SPRNG(ROW,COL)=X
      SPRNG(ROW,COL+4)=0.6463*X
      IF(COL.NE.3)GO TO 451
      IF(NPD.NE.1)GO TO 600
      SPRNG(ROW,11)=SPRNG(ROW,7)*PDFTOR
      GO TO 451
620   WRITE(IO,360) (SPRNG(ROW,J),J=1,2)
      READ(IN,/) SPRNG(ROW,12)
      IF(SPRNG(ROW,12).LE.0)GO TO 620
      GO TO 451
800   IF(SAVED.EQ.1)GO TO 801
      INQUIRE(IF,LASTRECORD=N2)
      N2=N2+2
      WRITE(IF=N2,810)
      GO TO 802
      ENTRY EXS(ANSWE,SAVED)
801   WRITE(IF,810)
810   FORMAT('ELEMENT EXSPRG')
802   WRITE(IF,820)
820   FORMAT(4X,'TABLE EXSTSPRG,ZERO')
      WRITE(IF,830) (FLOW(I),I,I=1,S)
830   FORMAT(9X,'*',5X,'COST',5X,'PDFLOW',4(5X,A4,I1))
      IF(ANSWE.EQ.'NO')GO TO 855
      DO 840 I=1,III
840   WRITE(IF,850) (SPRNG(I,J),J=1,2),SPRNG(I,12),SPRNG(I,11),
     *(SPRNG(I,J),J=7,S+6)
850   FORMAT(9X,2A1,F8.2,5X,F6.2,4(4X,F6.2))
855   IF(ANSWE.IS.'NO')WRITE(IF,856)
856   FORMAT(9X,'DU')
      WRITE(IF,860)
860   FORMAT('ENDATA')

890   SAVED=1
      RETURN
      END
```

EXISTING TREATMENT PLANT SUBROUTINE

```
$SET SEPARATE
      SUBROUTINE EXTRPL(SAVED              )
      DIMENSION ALTER(10),TEMP(4),TRPL(60,20),LISTZO(40,22)
      DIMENSION PLANT(4),CAP(4),MGD(4),NS(4),OAM(4),CST(4),CS(4),COLN(4)
      INTEGER ROW,COL,TF
      COMMON IN,IO,IF,TF,LISTZO,N,S
      COMMON /CMN3/ OM1,PC1,PD1
      DATA ALTER/'A','B','C','D','E','F','G','H','I','J'/
      DATA PLANT/4*'PLANT'/,CAP/4*' CAP '/,MGD/4*' MGD '/,COLN/4*'COL'/
      DATA CS/4*' CST '/OAM/4*' O&M '/CST/4*' $/MG'/
      DATA NS/' S-1 ',' S-2 ',' S-3 ',' S-4 '/
                PDFTOR=PD1
      WRITE(IO,90)
90    FORMAT(////17X,10('*'),' SEGMENT 5 ',10('*'))
      WRITE(IO,100)
100   FORMAT(////7X,'***** ENTER DATA FOR EXISTING TREATMENT PLANTS '
     *'*****'////)

180   WRITE(IO,190)PDFTOR
190   FORMAT(//5X,'THE STANDARD PEAK DAY SUPPLY IS ',F4.2,' TIMES'/
     *5X,'THE PEAK SEASONAL DAILY CAPACITY.'/
     *5X,'IS THIS ACCEPTABLE FOR ALL YOUR ZONES? <YES/NO>?'////)
720   READ(IN,1)ANS
1     FORMAT(A6)
      IF(ANS.EQ.'YES')GO TO 200
      IF(ANS.EQ.'NO')GO TO 730
      WRITE(IO,2)
2     FORMAT(5X,'PLEASE ANSWER YES OR NO')
      GO TO 720
730   WRITE(IO,740)
740   FORMAT(5X,'IS THERE A CONSTANT VALUE THAT IS ACCEPTABLE?'
     *' <YES/NO>?')
750   READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 760
      IF(ANS.EQ.'NO')GO TO 201
      WRITE(IO,2)
      GO TO 750
760   WRITE(IO,770)
770   FORMAT(5X,'ENTER PEAK DAY MULTIPLIER CONSTANT.?')
      READ(IN,/)PDFTOR
      IF(PDFTOR.LE.1)GO TO 200
      WRITE(IO,775)
775   FORMAT(5X,'A PEAK DAY MULTIPLIER GREATER THAN 1 FOR '
     *'SUPPLY IS UNACCEPTABLE.')
      GO TO 760
200   NPD=1
201   WRITE(IO,103)
103   FORMAT(//5X,'[NOTE] FOR MODEL SIMPLICITY IT IS ADVISIBLE TO GROUP'
     */5X,'ALL EXISTING TREATMENT PLANTS IN A ZONE AND INPUT AS ONE'/
     *5X,'SOURCE IF POSSIBLE.'/
     *5X,'ALSO, SEASON ONE IS CONSIDERED THE PEAK SEASON.'////)
      DO 110 I=1,N
      WRITE(IO,115)(LISTZO(I,L),L=1,22)
115   FORMAT(///3X,5('*'),2A1,2X,20A1,5('*')///)
116   WRITE(IO,117)
117   FORMAT(5X,'ENTER THE NUMBER OF EXISTING TREATMENT PLANTS (OR '
     *'GROUPS)'/5X,'IN THIS ZONE. <0 - 10>?')
      READ(IN,/)NTP
```

```
      IF(NTP)118,110,130
118   WRITE(IO,119)
119   FORMAT(5X,'PLEASE ENTER A NON-NEGATIVE NUMBER.')
      GO TO 116
1130  IF(NTP.GT.10)GO TO 116
130   DO 125 K=1,NTP
      KK=0
      X=PDFTOR
      WRITE(IO,105)(LISTZO(I,L),L=1,2),ALTER(K)
105   FORMAT(///5X,'ENTER EXISTING TREATMENT PLANT CAPACITIES <MGD> FOR'
     */5X,'EACH SEASON SEPARATED BY COMMAS FOR ZONE '2A1' TREATMENT '
     *'PLANT  "'A1'".?')
      READ(IN,/)(TEMP(J),J=1,S)
      II=II+1
      TRPL(II,1)=LISTZO(I,1)
      TRPL(II,2)=LISTZO(I,2)
      DO 120 J=1,S
      IF(TEMP(J).LE.0)GO TO 120
      KK=KK+1
      TRPL(II,J+2)=TEMP(J)
120   CONTINUE
      IF(KK.NE.0)GO TO 135
      II=II-1
      GO TO 125
135   TRPL(II,12)=ALTER(K)
      IF(NPD.EQ.1)GO TO 150
137   WRITE(IO,140)(TRPL(II,L),L=1,2),TRPL(II,12)
140   FORMAT(5X,'ENTER PEAK DAY MULTIPLIER FOR ZONE '2A1' PLANT  "'
     *A1'".?')
      READ(IN,/)X
      IF(X.LE.1)GO TO 150
      WRITE(IO,775)
      GO TO 137
150   TRPL(II,7)=X * TRPL(II,3)
      WRITE(IO,160)(TRPL(II,L),L=1,2),TRPL(II,12)
160   FORMAT(5X,'ENTER O&M COSTS <$/MG> FOR EACH SEASON FOR ZONE '2A1
     *' PLANT  "'A1'".?')
      READ(IN,/)(TEMP(L),L=1,S)
      DO 165 J=1,S
      IF(TEMP(J).LE.0)GO TO 165
      TRPL(II,J+7)=TEMP(J)
165   CONTINUE
125   CONTINUE
110   CONTINUE
      III=II
420   WRITE(IO,400)
400   FORMAT(////5X,'THE FOLLOWING IS A LIST OF YOUR DATA.'///)
      WRITE(IO,410)(COLN(L),L=1,S),(COLN(L),L=1,S)
410   FORMAT(14X,'COL',1X,8(3X,A3))
      WRITE(IO,415)(2,L=1,S),(3,L=1,S)
415   FORMAT(15X,'1',2X,8(4X,I1,1X))
      WRITE(IO,421)(PLANT(L),L=1,S),(DAM(L),L=1,S)
421   FORMAT(14X,'PEAK',1X,8(1X,A5))
      WRITE(IO,425)(CAP(L),L=1,S),(NS(L),L=1,S)
425   FORMAT(14X,'DAY',2X,8(1X,A5))
      WRITE(IO,430)(NS(L),L=1,S),(CST(L),L=1,S)
430   FORMAT(14X,'CAP',2X,8(1X,A5))
      WRITE(IO,435)(MGD(L),L=1,S)
435   FORMAT(' ROW ZONE ALT MGD',2X,4(1X,A5))
      DO 440 KJ=1,II
440   WRITE(IO,445)KJ,(TRPL(KJ,J),J=1,2),TRPL(KJ,12),TRPL(KJ,7),
     *(TRPL(KJ,J),J=3,S+2),(TRPL(KJ,J),J=8,S+7)
445   FORMAT(2X,I2,2X,2A1,3X,A1,F7.1,8F6.1)
      IF(ROW.NE.0)GO TO 560
      WRITE(IO,460)
460   FORMAT(////5X,'ARE THERE ANY CHANGES REQUIRED IN THIS DATA?'
     *' <YES/NO>?')
470   READ(IN,1)ANS
      IF(ANS.EQ.'NO' .AND. ROW.EQ.0)GO TO 800
      IF(ANS.EQ.'NO' .AND. ROW.NE.0)GO TO 480
      IF(ANS.EQ.'YES')GO TO 520
      WRITE(IO,2)
      GO TO 470
480   WRITE(IO,490)
490   FORMAT(5X,'DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>?')
500   READ(IN,1)ANS
      IF(ANS.EQ.'NO')GO TO 800
      IF(ANS.EQ.'YES')GO TO 510
      WRITE(IO,2)
      GO TO 500
510   ROW = 0
      COL = 0
      II = III
      GO TO 420
520   WRITE(IO,530)
530   FORMAT(5X,'ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.?')
      READ(IN,/)ROW,COL
      IF(ROW.GT.0 .AND. ROW.LE.III .AND. COL.GT.0 .AND. COL.LE.3)
     *GO TO 550
      WRITE(IO,540)
540   FORMAT(5X,'UNACCEPTABLE ROW OR COLUMN.')
      GO TO 520
550   KJ = ROW
      II = ROW
      GO TO(600,620,640)COL
560   WRITE(IO,570)
570   FORMAT(5X,'MORE CHANGES? <YES/NO>?')
      GO TO 470
600   WRITE(IO,140)(TRPL(ROW,J),J=1,2),TRPL(ROW,12)
      READ(IN,/)X
      IF(X.LE.1)GO TO 601
      WRITE(IO,775)
      GO TO 600
601   TRPL(ROW,7)=TRPL(ROW,3) * X
      GO TO 440
620   WRITE(IO,105)(TRPL(ROW,J),J=1,2),TRPL(ROW,12)
      READ(IN,/)(TRPL(ROW,J),J=3,S+2)
      IF(NPD.NE.1)GO TO 600
      TRPL(ROW,7)=PDFTOR * TRPL(ROW,3)
      GO TO 440
640   WRITE(IO,160)(TRPL(ROW,J),J=1,2),TRPL(ROW,12)
      READ(IN,/)(TRPL(ROW,J),J=8,S+7)
      GO TO 440
      ENTRY EXT(ANSWE,SAVED)
800   IF(SAVED.EQ.1)GO TO 801
      INQUIRE(IF,LASTRECORD=N2)
      N2=N2+2
      WRITE(IF=N2,810)
810   FORMAT('ELEMENT EXTRPL')
      GO TO 802
801   WRITE(IF,810)
802   WRITE(IF,820)
```

```
820   FORMAT(4X,'TABLE EXSTTRPL,ZERO')
      WRITE(IF,830)(CAP(J),J,J=1,S)
830   FORMAT(9X,'*      PDFLOW',4(4X,A4,I1))
      IF(ANSWE.EQ.'NO')GO TO 855
      DO 840 I=1,III
840   WRITE(IF,850)(TRPL(I,J),J=1,2),TRPL(I,12),TRPL(I,7),
     *(TRPL(I,J),J=3,S+2)
850   FORMAT(9X,3A1,2X,F6.1,1X,4F9.2)
      WRITE(IF,842)(CS(J),J,J=1,S)
842   FORMAT(9X,'*',8X,4(4X,A4,I1))
      DO 843 I=1,III
843   WRITE(IF,844)(TRPL(I,J),J=1,2),TRPL(I,12),(TRPL(I,J),J=8,S+7)
844   FORMAT(9X,3A1,6X,4F9.2)
855   IF(ANSWE.IS.'NO')WRITE(IF,856)
856   FORMAT(9X,'DUM')
      WRITE(IF,860)
860   FORMAT('ENDATA')
      SAVED=1
      RETURN
      END
```

INTERZONAL CONNECTION SUBROUTINE

```
$SET SEPARATE
      SUBROUTINE CONECT(SAVED,PC,PUMPC,PDD,PDFTR)
      DIMENSION LISTZO(40,22),CONNTO(10),ATOB(80,13),NS(4),PPFLOW(11)
      DIMENSION PFLOW(11),RESELV(40),MIX1(80),MIX2(80),ALTER(11)
      DIMENSION TXATOB(80,40),DEM(40,4),SDAYS(4),FTOR(4)
      DIMENSION AB(4),BA(4),COLN(4),ATOB1(120,17),PCOST(11)
      REAL K1,K2
      INTEGER S,ROW,PLIST(11),COL,SNS(4,8),PDD,TF
      COMMON IN,IO,IF,TF,LISTZO,N,S,IIII
      COMMON /CMN1/ ALTER,PLIST,PPFLOW,PCOST,FTOR
      COMMON /CMN2/ RESELV,DEM,SDAYS
      COMMON /CMN3/ OM1,PC1,PD1,PD2,OM2
      COMMON /CMN4/ R5,N5,AK1,AE1,AXPMP
      DATA AB/4*'AB'/,BA/4*'BA'/
      DATA COLN/4*'COL'/
      DATA NS/'S1','S2','S3','S4'/
      DATA SNS/4*8,12,3*9,0,12,10,10,0,13,12,11,0,0,13,12,0,0,
     *14,13,0,0,0,14,0,0,0,15/
      DO 47 I=1,80
      DO 47 J=1,40
47    TXATOB(I,J)='      '
      DO 1001 I=1,11
1001  PFLOW(I)=PPFLOW(I)*0.00144
      CRF1=R5 + (R5/((1.0+R5)**N5 - 1.0))
      CRF=CRF1
      DO 1000 I=1,N
      DO 1000 J=1,S
1000  DEM(I,J)=DEM(I,J)/SDAYS(J)
      K1=AK1
      E1=AE1
      XPMP=AXPMP
      CAP6=INT((1000.*PCOST(1) + K1*FTOR(1)*1000.*6.**E1)/100.)*100
      IF(PC.EQ.0)PUMPC=PC1
      IF(PDD.EQ.0)PDFTR=PD2
      WRITE(IO,90)
90    FORMAT(///17X,10('*'),' SEGMENT 6 ',10('*'))
      WRITE(IO,400)
400   FORMAT(///10X,'*****ENTER THE ZONAL TRANSFER INFORMATION*****'//
     *19X,'***** IMPORTANT NOTE *****'/
     *5X,'A MAXIMUM OF 80 INTERCONNECTIONS AND 120 TOTAL PIPE'/
     *5X,'OPTIONS ARE ALLOWED.'/
     *5X,'A PROBABLE DIRECTION OF FLOW WILL BE IMPLIED IN THIS'/
     *5X,'SECTION BY THE WAY YOU INDICATE CONNECTIONS.  FOR'/
     *5X,'EXAMPLE:   IF YOU CONNECT ZONE 01 TO ZONE 14 THE'/
     *5X,'IMPLICATION IS THAT THE MOST PROBABLE DIRECTION OF'/
     *5X,'FLOW WILL BE TO ZONE 14 AND THIS MODEL WILL CALCULATE'/
     *5X,'THE OPTIONAL PIPE SIZES BASED ON ZONE 14 DEMAND EVEN'/
     *5X'THOUGH REVERSE FLOW COULD BE ALLOWED.'/
     *//4X,56('*')/
     *5X'IF A ZONAL CONNECTION HAS BEEN PREVIOUSLY DEFINED OR A '/
     *5X'ZONE IS NOT TO BE CONNECTED TO ANY OTHER ZONES ENTER'/
     *5X'<RETURN> WHEN QUESTIONED ABOUT THIS ZONE.'/
     *4X,56('*')////)
      WRITE(IO,10)(PLIST(I),PCOST(I),I=1,11),K1,(FTOR(I),I=1,4),
     *E1,CAP6,6,500
10    FORMAT(5X,'THE CALCULATING FORMULA FOR CAPITAL COSTS OF'/
     *5X,'ZONAL TRANSFER FACILITIES IN THIS MODEL IS OF THE FORM:'/
     *5X,' C = X*PIPE COST   +  K1*FTOR*X*D**E1      WHERE:'/
```

```
     *8X,'C = TOTAL CAPITAL COST'/
     *8X,'D = PIPE DIAMETER IN INCHES'/
     *8X,'X = LENGTH OF LINE IN FEET'/
     *8X,'PIPE COST = COST OF PIPE PER FOOT'/
     *11(10X,I2,' INCH DIA. = ',$F7.2,'/FOOT'/)/
     *8X,'K1 = INSTALLATION CONSTANT MULTIPLIER (DEFAULT K1='F6.4')'/
     *8X,'FTOR = PIPE INSTALLATION DIFFICULTY FACTOR:'/
     *10X,'NORMAL EXCAVATION AND BACKFILL (DEFAULT'F4.1')'/
     *10X,'ROUGHER EXCAVATION (BUT NO RIPPING) AND SELECT BACKFILL '/
     *12X,'(DEFAULT'F4.1')'/
     *10X,'ROCK EXCAVATION AND BACKFILL FROM BORROW (DEFAULT'F4.1')'/
     *10X,'BELOW WATER EXCAVATION WITH GRAVEL BACKFILL (DEFAULT'F4.1')'/
     *8X,'E1 = INSTALLATION SCALE FACTOR EXPONENT (DEFAULT E1='F6.3')'/
     *5X,'DEFAULT VALUES GIVE A CAPITAL COST OF ABOUT 'K$I8/
     *5X,'FOR A 1000 FOOT 'I2' INCH LINE WITH A 'I4' GPM'/
     *5X,'CAPACITY AND NORMAL EXCAVATION.'////)
      WRITE(IO,120)
120   FORMAT(5X,'WILL THESE DEFAULT VALUES BE ACCEPTABLE FOR ALL'/
     *5X,'YOUR FUTURE INTERZONAL TRANSFER FACILITIES? <YES/NO>?')
130   READ(IN,140)ANS
140   FORMAT(A6)
      IF(ANS.EQ.'NO')GO TO 155
      IF(ANS.EQ.'YES')GO TO 35
      WRITE(IO,145)
145   FORMAT(5X,'PLEASE ANSWER YES OR NO.?')
      GO TO 130
155   WRITE(IO,156)
156   FORMAT(5X,'ARE THERE VALUES THAT WILL BE ACCEPTABLE FOR ALL'/
     *5X,'YOUR FUTURE INTERZONAL TRANSFER FACILITIES? <YES/NO>?')
157   READ(IN,140)ANS
      IF (ANS.EQ.'NO')GO TO 162
      IF(ANS.EQ.'YES')GO TO 160
      WRITE(IO,145)
      GO TO 157
160   WRITE(IO,161)
161   FORMAT(5X,'ENTER THE VALUES FOR <PIPECOST(1-11),FTOR(1-4),K1,E1>.?
     *')
      READ(IN,/)PCOST,FTOR,K1,E1
35    KE=1
162   WRITE(IO,163)R5,N5
163   FORMAT(//5X,'THE CAPITAL RECOVERY FACTOR (CRF) FORMULA'/
     *5X,'IS  CRF = R+ +R/[(1+R)**N - 1]! WHERE:'/10X,'R = INTEREST'
     *' RATE (DEFAULT R = ',F5.3,')'/10X,'N = NUMBER OF YEARS (DEFAU'
     *'LT N =',I3,')'/5X,'ARE THESE VALUES ACCEPTABLE FOR ALL YOUR'
     *' FUTURE ZONAL'/5X,'TRANSFER FACILITIES? <YES/NO>?'////)
165   READ(IN,140)ANS
      IF(ANS.EQ.'NO')GO TO 166
      IF(ANS.EQ.'YES')GO TO 168
      WRITE(IO,145)
      GO TO 165
166   WRITE(IO,156)
167   READ(IN,140)ANS
      IF(ANS.EQ.'NO')GO TO 170
      IF(ANS.EQ.'YES')GO TO 169
      WRITE(IO,145)
      GO TO 167
169   WRITE(IO,164)
164   FORMAT(5X,'ENTER THE VALUES <N,R>.?')
      READ(IN,/)N1,R1
      IF(N1.LE.0 .OR. R1.LE.0)GO TO 169
      CRF1=R1 + (R1/((1.0+R1)**N1 - 1.0))
      CRF=CRF1
168   NR=1
170   DO 512 I=1,N
403   WRITE(IO,405) (LISTZO(I,J),J=1,22)
405   FORMAT(//1X,5('*'),1X,2A1,3X,20A1,1X,5('*'))
406   WRITE(IO,410)
410   FORMAT(5X,'CONNECTED TO ZONE ???')
      READ(IN,415)CONNTO
415   FORMAT(10A1)
      DO 420 K=1,10
      KK=K
      IF(CONNTO(K).EQ.' ')GO TO 420
      IF(CONNTO(K+1).NE.' ')GO TO 425
      II=II+1
      CONNTO(K+1)=CONNTO(K)
      CONNTO(K)='        '
      ATOB(II,3)='0'
      ATOB(II,4)=CONNTO(K)
      GO TO 430
420   CONTINUE
      IF(COL.EQ.0)GO TO 512
      II=II+1
      GO TO 518
425   II=II+1
      ATOB(II,3)=CONNTO(K)
      ATOB(II,4)=CONNTO(K+1)
430   ATOB(II,1)=LISTZO(I,1)
      ATOB(II,2)=LISTZO(I,2)
      ATOB(II,13)=I
      T1=ATOB(II,3)
      T2=ATOB(II,4)
      T3=ATOB(II,1)
      T4=ATOB(II,2)
      IF(T1.EQ.T3 .AND. T2.EQ.T4)GO TO 442
      GO TO 443
442   WRITE(IO,441)
441   FORMAT(5X,'YOU CAN NOT CONNECT A ZONE TO ITSELF.'/)
      II=II-1
      GO TO 403
443   IF(II.EQ.1)GO TO 434
      DO 431 M=1,II-1
      IF(T1.EQ.ATOB(M,1) .AND. T2.EQ.ATOB(M,2) .AND. T3.EQ.ATOB(M,
     *3) .AND. T4.EQ.ATOB(M,4))GO TO 432
      IF(T1.EQ.ATOB(M,3).AND.T2.EQ.ATOB(M,4).AND.T3.EQ.ATOB(M,1)
     *.AND.T4.EQ.ATOB(M,2))GO TO 432
431   CONTINUE
      GO TO 434
432   WRITE(IO,433) T1,T2,T3,T4
433   FORMAT(5X,'YOU HAVE PREVIOUSLY IDENTIFIED A CONNECTION'/
     *5X,'BETWEEN ZONES ',2A1,' AND ',2A1 /)
      II=II-1
      GO TO 403
434   LL=0
      DO 435 L=3,22
      IF(LISTZO(I,L).IS.' ' .AND. LISTZO(I,L+1).IS.' ')GO TO 440
      LL=LL+1
      TXATOB(II,LL)=LISTZO(I,L)
435   CONTINUE
440   LL=LL+1
      TXATOB(II,LL)=' '
      LL=LL+1
```

```
      TXATOB(II,LL)=' '
      LL=LL+1
      TXATOB(II,LL)='T'
      LL=LL+1
      TXATOB(II,LL)='O'
      LL=LL+1
      TXATOB(II,LL)=' '
      LL=LL+1
      TXATOB(II,LL)=' '
      DO 445 M=1,N
      MM=M
      IF(CONNTO(KK).EQ.LISTZO(M,1) .AND. CONNTO(KK+1).EQ.LISTZO(M,2))
     *GO TO 450
445   CONTINUE
      WRITE(IO,446)
446   FORMAT(5X,'INVALID ZONE NUMBER')
      II=II-1
      GO TO 406
450   DO 455 L=3,22
      IF(LISTZO(MM,L).IS.' ' .AND. LISTZO(MM,L+1).IS.' ')GO TO 458
      LL=LL+1
      IF(LL.GT.40)GO TO 458
      TXATOB(II,LL)=LISTZO(MM,L)
455   CONTINUE
458   MIX1(II)=I
      MIX2(II)=MM
460   WRITE(IO,461)
461   FORMAT(5X'IS THERE AN EXISTING CONNECTION? <YES/NO>?')
462   READ(IN,463)ANS
463   FORMAT(A6)
      ATOB(II,6)='NO'
      IF(ANS.EQ.'NO')GO TO 1400
      IF(ANS.EQ.'YES')GO TO 465
      WRITE(IO,464)
464   FORMAT(5X'PLEASE ANSWER YES OR NO?')
      GO TO 462
465   ATOB(II,6)='YES'
1450  WRITE(IO,1460) (TXATOB(II,J),J=1,40),PLIST
1460  FORMAT(5X,'WHAT IS THE SIZE OF YOUR EXISTING PIPE'/5X,
     *'FROM ',40A1,'?'/5X,'<',10(I2,','),I2,'> (INCH)?')
1461  READ(IN,/)ATOB(II,9)
      X=ATOB(II,9)
      DO 1462 J=1,11
      IF(X.EQ.PLIST(J))GO TO 68
1462  CONTINUE
      WRITE(IO,1463)
1463  FORMAT(5X,'PLEASE PICK FROM ONLY THE SIZE OPTIONS LISTED?')
      GO TO 1461
68    ATOB(II,12)=J
      IF(ROW.NE.0)GO TO 518
466   WRITE(IO,66)
66    FORMAT(5X,'DO YOU ALSO WANT TO LOOK AT NEW CONNECTIONS?'
     *' <YES/NO>?')
67    READ(IN,463)ANS
      IF(ANS.NE.'NO')GO TO 2465
      AC='NO'
      GO TO 1465
2465  IF(ANS.EQ.'YES')GO TO 1465
      WRITE(IO,464)
      GO TO 67
1400  ATOB(II,9)=0.0
      IF(ROW.NE.0)GO TO 518
1465  WRITE(IO,1467) (TXATOB(II,J),J=1,40)
1467  FORMAT(5X,'WHAT IS THE DISTANCE FROM '/
     *5X,40A1,'?',' <FEET>?')
      READ(IN,/) ATOB(II,5)
467   IF(ROW.NE.0)GO TO 518
470   WRITE(IO,468)
468   FORMAT(5X,'IS REVERSE FLOW ALLOWED?', ' <YES/NO>?')
469   READ(IN,463)ANS
      IF(ANS.EQ.'NO')GO TO 481
      IF(ANS.EQ.'YES')GO TO 480
      WRITE(IO,464)
      GO TO 469
480   ATOB(II,7)='YES'
      GO TO 485
481   ATOB(II,7)='NO'
485   IF(ROW.NE.0)GO TO 518
490   IF(AC.EQ.'NO')GO TO 510
499   WRITE(IO,491)
491   FORMAT(5X,'HOW MANY PIPE OPTIONS DO YOU WANT TO LOOK AT'/
     *5X,'IN YOUR MODEL FOR THIS ZONAL TRANSFER?'
     *' <1 - 4>?')
492   READ(IN,/) ATOB(II,8)
      IF(ATOB(II,8).LE.4 .AND. ATOB(II,8).GE.1)GO TO 495
      WRITE(IO,493)
493   FORMAT(5X,'PLEASE SELECT 1 THROUGH 4 ONLY.?')
      GO TO 492
495   IF(ROW.NE.0)GO TO 518
496   WRITE(IO,500)
500   FORMAT(5X,'INDICATE THE TYPE OF PIPE INSTALLATION <1-4>.?')
      IF(II.NE.1)GO TO 505
      WRITE(IO,1502)
1502  FORMAT(/10X,'<1> NORMAL EXCAVATION AND NORMAL BACKFILL'/10X,
     *'<2> ROUGHER EXCAVATION (BUT NO RIPPING) AND SELECT BACK'
     *'FILL'/10X,'<3> ROCK EXCAVATION AND BACKFILL FROM BORROW'/10X,
     *'<4> BELOW WATER EXCAVATION WITH GRAVEL BACKFILL?')
505   READ(IN,/)X
      DO 501 J=1,4
      IF(X.NE.J)GO TO 501
      ATOB(II,10)=X
310   IF(NR.EQ.1)GO TO 340
320   WRITE(IO,330)
330   FORMAT(5X,'ENTER CAPITAL RECOVERY FORMULA VALUES <N,R>.?')
      IF(N1.LE.0 .OR. R1.LE.0)GO TO 320
      IF(N1.NE.0)CRF=R1+(R1/((1.0+R1)**N1 - 1.0))
340   ATOB(II,11)=CRF
      CRF=CRF1
      IF(ROW.NE.0 .OR. COL.NE.0)GO TO 518
      GO TO 510
501   CONTINUE
      WRITE(IO,502)
502   FORMAT(5X,'PLEASE ENTER 1,2,3 OR 4 ONLY.?')
      GO TO 505
510   DO 513 IP=1,10
513   CONNTO(IP)='      '
      AC='    '
      IF(ROW.NE.0 .OR. COL.NE.0)GO TO 518
      I=I-1
512   CONTINUE
      III=II
511   WRITE(IO,515)
```

```
515   FORMAT(///5X'THE FOLLOWING IS A LIST OF YOUR DATA.    '//)
      WRITE(IO,516) (I,I=1,7)
516   FORMAT(7X,6(6X,'COL'),4X,'COL'/6X,6(8X,I1),6X,I1)
      WRITE(IO,517)
517   FORMAT(10X,'CONNECTED',1X,'DISTANCE',1X,'EXISTING',
     *1X,'REVERSE',1X,'NO. PIPE',1X,'EXISTING',2X,'SOIL'/
     *1X,'ROW',1X,'ZONE',2X,'TO ZONE',3X,'A TO B',4X'PIPE',5X,
     *'FLOW',2X,'OPTIONS',2X,'PIPE SIZE',1X,'TYPE')
      DO 518 KJ=1,II
518   WRITE(IO,519) KJ,(ATOB(KJ,J),J=1,10)
519   FORMAT(2X,I2,2X,2A1,6X,2A1,5X,I6,4X,A3,6X,A3,
     *6X,I2,7X,F3.0,4X,I2)
      IF(ROW.NE.0 .OR. COL.NE.0)GO TO 580
      WRITE(IO,530)
530   FORMAT(////5X,'ARE THERE ANY CHANGES REQUIRED IN THIS DATA?'
     *' <YES/NO>?')
540   READ(IN,463)ANS
      IF(ANS.EQ.'NO' .AND.  COL.EQ.0)GO TO 899
      IF(ANS.EQ.'NO' .AND. COL.NE.0)GO TO 560
      IF(ANS.EQ.'YES')GO TO 550
      WRITE(IO,464)
      GO TO 540
550   WRITE(IO,551)
551   FORMAT(5X,'ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE?')
552   READ(IN,/)ROW,COL
      IF(ROW.LE.0 .OR. COL.LT.1 .OR. ROW.GT.III .OR. COL.GT.7)
     *GO TO 570
      I=ATOB(ROW,13)
      II = ROW
      KJ=ROW
      GO TO (4,1465,460,470,498,2450,2496)COL
70    READ(IN,463)ANS
      IF(ANS.EQ.'NO')GO TO 580
      IF(ANS.EQ.'YES')GO TO 71
      WRITE(IO,464)
      GO TO 70
71    ROW=0
      GO TO 499
498   IF(ATOB(II,8).GT.0.0)GO TO 499
6     FORMAT(5X,'YOU SHOW THIS ZONAL TRANSFER TO BE SUPPLIED BY'/
     *5X,'AN EXISTING LINE ONLY.  YOU CAN NOT CHANGE THE NUMBER'/
     *5X,'OF PIPE OPTIONS FROM ZERO.  DO YOU NOW WANT TO',
     */5X,'LOOK AT NEW CONNECTIONS? <YES/NO>?'////)
      WRITE(IO,6)
      GO TO 70
2496  IF(ATOB(II,10).GT.0.0)GO TO 496
      WRITE(IO,2497)
2497  FORMAT(5X,'YOU SHOW THIS ZONAL TRANSFER TO BE SUPPLIED BY'/
     *5X,'AN EXISTING LINE ONLY.  THE SOIL TYPE IS NOT REQUIRED.'/
     *5X,'DO YOU NOW WANT TO LOOK AT NEW CONNECTIONS? <YES/NO>?'////)
      GO TO 70
2450  IF(ATOB(II,9).GT.0.0)GO TO 1450
      WRITE(IO,2451)
2451  FORMAT(5X,'YOU HAVE INDICATED THAT FOR THIS ZONAL TRANSFER'/
     *5X,'NO PIPE LINE EXISTS.  EXISTING PIPE SIZE IS NOT'/
     *5X,'REQUIRED.  DO YOU NOW WISH TO ADD AN EXISTING PIPE?'/
     *5X,' <YES/NO>?'////)
710   READ(IN,463)ANS
      IF(ANS.EQ.'NO')GO TO 580
      IF(ANS.EQ.'YES')GO TO 465
      WRITE(IO,464)
      GO TO 710
4     II=II-1
      DO 5 A=1,13
5     ATOB(ROW,A)=0
      DO 7 A=1,40
7     IXATOB(ROW,A)=0
      ROW=0
      GO TO 403
570   WRITE(IO,571)
571   FORMAT(5X,'UNACCEPTABLE ROW OR COLUMN')
      GO TO 550
580   WRITE(IO,581)
581   FORMAT(5X,'MORE CHANGES? <YES/NO>?')
      GO TO 540
560   WRITE(IO,561)
561   FORMAT(5X'DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>?')
563   READ(IN,463)ANS
      IF(ANS.EQ.'NO')GO TO 899
      IF(ANS.EQ.'YES')GO TO 590
      WRITE(IO,464)
      GO TO 563
590   ROW=0
      COL=0
      II=III
      GO TO 511
899   COL=0
      ROW=0
      IJ=0
      DO 900 I=1,III
929   DIF=RESELV(MIX1(I)) - RESELV(MIX2(I))
      IF(DIF)930,940,950
930   COST1=-PUMPC*DIF/100.
      COST2=0
      GO TO 945
940   COST1=0
      COST2=0
      GO TO 945
950   COST1=0
      COST2=PUMPC*DIF/100.
      IF(ROW.EQ.0 .AND. COL.NE.0)GO TO 949
945   IF(ATOB(I,6).EQ.'YES')GO TO 910
946   X=DEM(MIX2(I),1)*PDFTR
      DO 901 J=1,11
      IF(X.GT.PFLOW(J))GO TO 901
      IF(J.EQ.1)J=J+1
948   DO 902 K=J-1,ATOB(I,8)+J-2
      IF(K.GT.11)GO TO 900
      IJ=IJ+1
      IIJJ=IIJJ+1
      ATOB1(IJ,1)=ATOB(I,1)
      ATOB1(IJ,2)=ATOB(I,2)
      ATOB1(IJ,3)=ATOB(I,3)
      ATOB1(IJ,4)=ATOB(I,4)
      ATOB1(IJ,16)=I
949   ATOB1(IJ,5)=ALTER(K)
      ATOB1(IJ,6)=PLIST(K)
      ATOB1(IJ,17)=K
      IF(KE.EQ.1)GO TO 904
827   WRITE(IO,903)(TXATOB(I,L),L=1,40),ALTER(K),PLIST(K)
903   FORMAT(5X,'ENTER KNOWN TOTAL CAPITAL COSTS FOR ZONAL TRANSFER'/
     *5X,40A1/5X,' ALTERNATE SIZE ',A1,' PIPE SIZE ',I2,' INCH.?')
```

```
      READ(IN,/)ATOB1(IJ,7)
      IF(ATOB1(IJ,7).LE.0)GO TO 827
      ATOB1(IJ,7)=INT(ATOB1(IJ,7)*ATOB(I,11)/10.)*10
      IF(COL.EQ.2)GO TO 815
      GO TO 960
904   ATOB1(IJ,7)=INT((ATOB(I,5)*PCOST(K) + K1*FTOR(ATOB(I,10))
     * *ATOB(I,5)*PLIST(K)**E1) * ATOB(I,11)/10.)*10
960   DO 980 M=1,S
      ATOB1(IJ,M+7)=(0.01932*((PFLOW(K)/PDFTR)*365.)**XPMP
     * *ATOB(I,5))/((PFLOW(K)/PDFTR)*365.) + COST1
      IF(ATOB(I,7).NE.'YES')GO TO 980
      ATOB1(IJ,M+11)=(0.01932*((PFLOW(K)/PDFTR)*365.)**XPMP
     * *ATOB(I,5))/((PFLOW(K)/PDFTR)*365.) + COST2
980   CONTINUE
      IF(ROW.NE.0 .OR. COL.NE.0)GO TO 815
902   CONTINUE
      GO TO 900
901   CONTINUE
      J=13. - ATOB(I,8)
      GO TO 948
900   CONTINUE
      IIJ=IJ
      GO TO 809
910   IJ=IJ+1
      ATOB1(IJ,1)=ATOB(I,1)
      ATOB1(IJ,2)=ATOB(I,2)
      ATOB1(IJ,3)=ATOB(I,3)
      ATOB1(IJ,4)=ATOB(I,4)
      ATOB1(IJ,5)='X'
      ATOB1(IJ,6)=PLIST(ATOB(I,12))
      ATOB1(IJ,7)=0.0
      ATOB1(IJ,16)=I
      K=ATOB(I,12)
      ATOB1(IJ,17)=K
      DO 981 M=1,S
      ATOB1(IJ,M+7)=(0.01932*((PFLOW(K)/PDFTR)*365.)**XPMP
     * *ATOB(I,5))/((PFLOW(K)/PDFTR)*365.) + COST2
      IF(ATOB(I,7).NE.'YES')GO TO 981
      ATOB1(IJ,M+11)=(0.01932*((PFLOW(K)/PDFTR)*365.)**XPMP
     * *ATOB(I,5))/((PFLOW(K)/PDFTR)*365.) + COST1
981   CONTINUE
      IF(COL.NE.0)GO TO 815
      IF(ATOB(I,8).EQ.0)GO TO 900
      GO TO 946
809   WRITE(IO,847)
847   FORMAT(///5X,'THE FOLLOWING IS A LIST OF CALCULATED DATA.'//)
      WRITE(IO,810)(COLN(MN),MN=1,S),(COLN(MN),MN=1,S)
810   FORMAT(12X,'COL',5X,'COL',8(3X,A3))
      WRITE(IO,811)(MN,MN=3,S+2),(MN,MN=S+3,2*S+2)
811   FORMAT(13X,'1',7X,'2',7(5X,I1),4X,I2)
      WRITE(IO,812)(AB(MN),MN=1,S),(BA(MN),MN=1,S)
812   FORMAT(12X,'PIPE',4X,'COST',2X,'OPERATION AND MAINTENANCE'
     *' COSTS - $/MG'/6X,'ZONE  DIA.    PER',8(4X,A2))
      WRITE(IO,813)(NS(MN),MN=1,S),(NS(MN),MN=1,S)
813   FORMAT(' ROW A TO B INCH     YEAR',8(3X,A2,1X))
      DO 815 KI=1,IJ
815   WRITE(IO,816)KI,(ATOB1(KI,J),J=1,4),ATOB1(KI,6),
     * (ATOB1(KI,J),J=7,S+7),(ATOB1(KI,J),J=12,S+11)
816   FORMAT(1X,I3,1X,2A1,2X,2A1,2X,I2,$I9,8(1X,F5.1))
      IF(COL.NE.0)GO TO 818
      WRITE(IO,530)
819   READ(IN,463)ANS
      IF(ANS.EQ.'YES')GO TO 820
      IF(ANS.EQ.'NO' .AND. COL.NE.0)GO TO 831
      IF(ANS.EQ.'NO' .AND. COL.EQ.0)GO TO 898
      WRITE(IO,464)
      GO TO 819
820   WRITE(IO,551)
821   READ(IN,/)ROW,COL
      IF(ROW.LT.1.OR.ROW.GT.IIJ.OR.COL.LT.1.OR.COL.GT.2*S+2)GO TO 833
      I=ATOB1(ROW,16)
      IJ=ROW
      KI=ROW
      K=ATOB1(ROW,17)
      GO TO(822,827,828,828,828,828,828,828,828,828)COL
818   WRITE(IO,581)
      GO TO 819
822   WRITE(IO,823) (ATOB1(ROW,J),J=1,4)
823   FORMAT(5X,'ENTER THE DESIRED DIAMETER <INCH> FOR',
     *' TRANSFER ',2A1,' TO ',2A1,'.?')
853   READ(IN,/)X
      DO 824 K=1,11
      IF(X.EQ.PLIST(K))GO TO 825
824   CONTINUE
      WRITE(IO,1463)
      GO TO 853
825   IF(ATOB1(ROW,5).NE.'X')GO TO 826
      IJ=IJ-1
      ATOB(I,12)=K
      GO TO 929
826   ROW=0
      GO TO 929
828   IF(ATOB1(ROW,SNS(S,COL-2)).NE.0)GO TO 882
      WRITE(IO,848)
848   FORMAT(5X,'SINCE REVERSE FLOW IS NOT ALLOWED IN THIS ZONAL'/
     *5X,'TRANSFER NO COSTS CAN BE ASSOCIATED WITH TRANSFERS FROM'/
     *5X,'ZONE B  TO  ZONE A.'////)
      GO TO 818
882   WRITE(IO,829) ROW,COL
829   FORMAT(5X,'ENTER O&M COSTS FOR ROW ',I2,' COL ',I2,'.?')
      READ(IN,/)ATOB1(ROW,SNS(S,COL-2))
      GO TO 815
831   WRITE(IO,561)
832   READ(IN,463)ANS
      IF(ANS.EQ.'YES')GO TO 834
      IF(ANS.EQ.'NO')GO TO 898
      WRITE(IO,464)
      GO TO 832
833   WRITE(IO,571)
      GO TO 820
834   ROW=0
      COL=0
      IJ=IIJ
      GO TO 809
      ENTRY CON(ANSWE,SAVED)
898   IF(SAVED.EQ.1)GO TO 994
      INQUIRE(IF,LASTRECORD=N2)
      N2=N2+2
      WRITE(IF=N2,997)
      GO TO 996
994   WRITE(IF,997)
997   FORMAT('ELEMENT PIPENT')
```

```
  996   WRITE(IF,998)
  998   FORMAT(4X,'TABLE PIPENET,ZERO')
        WRITE(IF,999)(AB(I),I,I=1,S)
  999   FORMAT(9X,'*',6X,'CAPTL',3X,'CAPAC',4(5X,A2,I1))
        IF(ANSWE.EQ.'NO')GO TO 916
        DO 835 I=1,IIJ
  835   WRITE(IF,906)(ATOB1(I,J),J=1,5),ATOB1(I,7),PFLOW(ATOB1(I,17)),
       *(ATOB1(I,J),J=8,S+7)
  906   FORMAT(9X,5A1,I7,1X,F7.2,4F8.2)
        WRITE(IF,962)(BA(I),I,I=1,S)
  962   FORMAT(9X,'*',8X,4(5X,A2,I1))
        DO 963 I=1,IIJ
  963   WRITE(IF,964)(ATOB1(I,J),J=1,5),(ATOB1(I,J),J=12,S+11)
  964   FORMAT(9X,5A1,4X,4F8.2)
  916   IF(ANSWE.IS.'NO')WRITE(IF,917)
  917   FORMAT(9X,'DUMMY')
        WRITE(IF,907)
  907   FORMAT(4X,'LIST (BTOA1)')
        IF(ANSWE.IS.'NO')GO TO 912
        DO 908 I=1,IIJ
  908   WRITE(IF,909)(ATOB1(I,J),J=3,4),(ATOB1(I,J),J=1,2),ATOB1(I,5)
  909   FORMAT(9X,5A1)
  912   IF(ANSWE.IS.'NO')WRITE(IF,917)
        WRITE(IF,1011)
 1011   FORMAT(4X,'LIST(ATOB),T=40')
        IF(ANSWE.EQ.'NO')GO TO 926
        DO 1012 I=1,III
 1012   WRITE(IF,1013)(ATOB(I,J),J=1,4),(TXATOB(I,J),J=1,40)
 1013   FORMAT(9X,4A1,4X,'"',40A1,'"')
  926   IF(ANSWE.IS.'NO')WRITE(IF,927)
  927   FORMAT(9X,'DUMY')
        WRITE(IF,990)
  990   FORMAT('ENDATA')
        IF(ANSWE.EQ.'NO')GO TO 1110
        IIII=IIII+IIJJ
        REWIND(TF)
        WRITE(TF,1120)N,S,IIII
 1120   FORMAT(3I6)
 1110   SAVED=1
        RETURN
        END
```

PROPOSED FUTURE WELLS SUBROUTINE

```
 $SET SEPARATE
        SUBROUTINE FWELLS(SAVED                                )
        DIMENSION LISTZO(40,22),RESELV(40)
        DIMENSION FWELL(120,12),SLIST(4)
        INTEGER S,ROW,COL,PC,OMP,TF
        REAL N1
        COMMON IN,IO,IF,TF,LISTZO,N,S,IIII
        COMMON /CMN2/ RESELV
        COMMON /CMN3/ OM1,PC1,PD1
        COMMON /CMN5/ AA1,ZZ1,R6,N6
        DATA SLIST/'A','B','C','D'/
                    PUMPC=PC1
                    OANDMP=OM1
                    PDFTOR=PD1
        A1=AA1
        Z1=ZZ1
        CRF1=R6 + (R6/((1.0+R6)**N6 - 1.0))
        CRF=CRF1
        CAP10=INT((A1*1000.**Z1)/100.)*100
        WRITE(IO,90)
   90   FORMAT(///14X,10('*'),' SEGMENT 7 ',10('*'))
        WRITE(IO,100)
  100   FORMAT(///10X,'***** ENTER DATA FOR FUTURE WELLS *****'///
       *5X,'[NOTE] YOU ARE ALLOWED UP TO FOUR ALTERNATE SIZE '
       *'WELLS PER ZONE'/5X,'AND A MAXIMUM TOTAL OF 120.'///)
        WRITE(IO,110)A1,Z1,CAP10
  110   FORMAT(5X,'THE CALCULATING FORMULA FOR CAPITAL COSTS OF WE'
       *'LLS'/5X,'IN THIS MODEL IS OF THE FORM  Y=A(X)**Z  WHERE:'/
       *10X,'Y = CAPITAL COST'/10X,'A = CONSTANT MULTIPLIER (DEFA'
       *'ULT  A = ',F5.0,')'/10X,'X = THE FLOW OF THE WELL IN GALLONS '
       *'PER MINUTE'/10X,'Z = THE SCALE FACTOR EXPONENT (DEFAULT Z ='
       *F5.3')'/5X,'DEFAULT VALUES GIVE A CAPITAL COST OF ABOUT 'K$I9,
       *' FOR A'/5X,'WELL OF 1000 GPM CAPACITY.' ////)
        WRITE(IO,120)
  120   FORMAT(5X,'WILL THESE DEFAULT VALUES BE ACCEPTABLE FOR ALL'/
       *5X,'YOUR FUTURE WELLS? <YES/NO>?')
  130   READ(IN,140)ANS
  140   FORMAT(A6)
        IF(ANS.EQ.'NO')GO TO 155
        IF(ANS.EQ.'YES')GO TO 150
        WRITE(IO,145)
  145   FORMAT(5X,'PLEASE ANSWER YES OR NO.?')
        GO TO 130
  155   WRITE(IO,156)
  156   FORMAT(5X,'ARE THERE CONSTANT VALUES THAT WILL BE ACCEPT'
       *'ABLE'/5X,'FOR ALL YOUR FUTURE WELLS? <YES/NO>?')
  157   READ(IN,140)ANS
        IF (ANS.EQ.'NO')GO TO 162
        IF(ANS.EQ.'YES')GO TO 160
        WRITE(IO,145)
        GO TO 157
  160   WRITE(IO,161)
  161   FORMAT(5X,'ENTER THE VALUES. <A,Z>?')
        READ(IN,/)A1,Z1
        IF(A1.LE.0 .OR. Z1.LE.0)GO TO 160
  150   AZ=1.0
  162   WRITE(IO,163)R6,N6
  163   FORMAT(//5X,'THE CAPITAL RECOVERY FACTOR (CRF) FORMULA'/
```

```
      *5X,'IS  CRF = R + +R/[(1+R)**N - 1]! WHERE:'/10X,'R = INTEREST'
      *' RATE (DEFAULT R = ',F5.3,')'/10X,'N = NUMBER OF YEARS (DEFAU'
      *'LT N =',I3,')'/5X,'ARE THESE VALUES ACCEPTABLE FOR ALL YOUR'
      *' FUTURE WELLS?'/5X,'<YES/NO>?'////)
  165 READ(IN,140)ANS
      IF(ANS.EQ.'NO')GO TO 166
      IF(ANS.EQ.'YES')GO TO 168
      WRITE(IO,145)
      GO TO 165
  166 WRITE(IO,156)
  167 READ(IN,140)ANS
      IF(ANS.EQ.'NO')GO TO 170
      IF(ANS.EQ.'YES')GO TO 169
      WRITE(IO,145)
      GO TO 167
  169 WRITE(IO,164)
  164 FORMAT(5X,'ENTER THE VALUES <N,R>.?')
      READ(IN,/)N1,R1
      IF(N1.LE.0 .OR. R1.LE.0)GO TO 169
      CRF1=R1+(R1/((1.0+R1)**N1-1.0))
      CRF=CRF1
  168 NR=1
  170 DO 390 I=1,N
  175 WRITE(IO,180) (LISTZO(I,J),J=1,22)
  180 FORMAT(//10X,5('*'),1X,2A1,3X,20A1,1X,5('*'))
  183 WRITE(IO,181)
  181 FORMAT(//5X,'HOW MANY OPTIONAL WELLS DO YOU WANT TO LOOK'/
      *5X,'AT FOR THIS ZONE? <0 - 4>?')
      READ(IN,/)B
      IF(B.LE.0)GO TO 390
      IF(B.GE.1 .AND. B.LE.4)GO TO 184
      WRITE(IO,182)
  182 FORMAT(5X,'PLEASE ENTER 0 TO 4 ONLY.')
      GO TO 183
  184 DO 380 K=1,B
      CRF=CRF1
  190 WRITE(IO,200) SLIST(K),(LISTZO(I,J),J=1,2)
  200 FORMAT(/5X,'ENTER FLOW <GPM> FOR WELL OPTION  "'A1'"  ZONE ',
      *2A1,'.?')
      READ(IN,/)X
      IF(X.GT.0)GO TO 187
      WRITE(IO,186)
  186 FORMAT(5X,'YOU ENTERED A FLOW LESS THAN OR EQUAL TO ZERO.')
      GO TO 190
  187 IF(ROW.NE.0)II=II-1
      II=II+1
      IF(ROW.NE.0)GO TO 205
      FWELL(II,10)=I
      FWELL(II,1)=LISTZO(I,1)
      FWELL(II,2)=LISTZO(I,2)
      FWELL(II,3)=K
  205 FWELL(II,6)=X
      FWELL(II,7)=0.00144*X
      IF(AZ.EQ.1)GO TO 300
  210 WRITE(IO,220)
  220 FORMAT(5X,'ENTER KNOWN CAPITAL COSTS FOR THIS WELL. <CAP COST>?')
      READ(IN,/)A
      IF(A.GT.0)GO TO 290
      WRITE(IO,226)
  226 FORMAT(5X,'YOU ENTERED A CAPITAL COST LESS THAN ZERO.')
      GO TO 210
  300 FWELL(II,4)=A1*FWELL(II,6)**Z1
      GO TO 310
  290 FWELL(II,4)=A
  310 IF(NR.EQ.1)GO TO 360
  320 WRITE(IO,330)
  330 FORMAT(5X,'ENTER CAPITAL RECOVERY FORMULA VALUES <N,R>.?')
      READ(IN,/)N1,R1
      IF(N1.LE.0 .OR. R1.LE.0)GO TO 320
      IF(N1.GT.0)CRF=R1+(R1/((1.0+R1)**N1-1.0))
  340 FWELL(II,5)=INT(FWELL(II,4)*CRF/10.)*10
      FWELL(II,4)=INT(FWELL(II,4)/100.)*100
      CRF=CRF1
      IF(ROW.NE.0)GO TO 440
  350 WRITE(IO,355)
  355 FORMAT(5X,'ENTER THE NUMBER OF POSSIBLE WELLS OF THIS '
      *'SIZE ALLOWED <NUMBER>.?')
  356 READ(IN,/)NU
      IF(NU.GT.0)GO TO 357
      WRITE(IO,358)
  358 FORMAT(5X,'YOU HAVE ENTERED LESS THAN ONE WELL.  PLEASE'/
      *5X,'REENTER THE NUMBER OF POSSIBLE WELLS.?')
      GO TO 356
  357 FWELL(II,8)=NU
      IF(ROW.NE.0)GO TO 440
  360 WRITE(IO,361)
  361 FORMAT(5X,'ENTER WELL ELEVATION FROM MSL. <FEET>?')
      READ(IN,/) FWELL(II,9)
      IF(ROW.NE.0)GO TO 440
  380 CONTINUE
  391 IF(ROW.EQ.0)GO TO 390
      DO 392 KK=6,9
  392 FWELL(II,KK)=0
      GO TO 440
  390 CONTINUE
      III=II
  400 WRITE(IO,410)
  410 FORMAT(///5X,'THE FOLLOWING IS A LIST OF YOUR DATA'/)
      WRITE(IO,420) (K,K=1,6)
  420 FORMAT(18X,'COL',7X,'COL',5X,'COL',3X,'COL',4X,'COL',7X,
      *'COL'/19X,I1,9X,I1,7X,I1,5X,I1,6X,I1,9X,I1)
      WRITE(IO,430)
  430 FORMAT(16X,2('CAPITAL',3X),2('FLOW',2X),1X,'MAX',4X,'ELEVATION'/
      *1X,'ROW',1X,'ZONE',1X,'ALT',4X,'TOTAL',4X,'PERYEAR',3X,
      *'GPM',3X,'MGD',3X,'NUMBER',5X,'MSL')
      DO 440 KJ=1,II
  440 WRITE(IO,450)KJ,(FWELL(KJ,J),J=1,2),SLIST(FWELL(KJ,3)),(FWELL
      *(KJ,J),J=4,9)
  450 FORMAT(2X,I2,2X,2A1,3X,A1,3X,KSI8,2X,KSI8,2X,I5,1X,F5.2,
      *4X,I1,8X,I4)
      IF(ROW.NE.0)GO TO 560
      WRITE(IO,460)
  460 FORMAT(////5X,'ARE THERE ANY CHANGES REQUIRED IN THIS DATA?'
      *' <YES/NO>?')
  470 READ(IN,140)ANS
      IF(ANS.EQ.'NO' .AND. ROW.EQ.0)GO TO 600
      IF(ANS.EQ.'NO' .AND. ROW.NE.0)GO TO 480
      IF(ANS.EQ.'YES')GO TO 520
      WRITE(IO,145)
      GO TO 470
  480 WRITE(IO,490)
  490 FORMAT(5X,'DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>?')
```

```
500   READ(IN,140)ANS
      IF(ANS.EQ.'NO')GO TO 600
      IF(ANS.EQ.'YES')GO TO 510
      WRITE(IO,145)
      GO TO 500
510   ROW = 0
      COL = 0
      II = III
      GO TO 400
520   WRITE(IO,530)
530   FORMAT(5X,'ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.?')
      READ(IN,/)ROW,COL
      IF(ROW.GT.0 .AND. ROW.LE.III .AND. COL.GT.0 .AND. COL.LE.6)
     *GO TO 550
      WRITE(IO,540)
540   FORMAT(5X,'UNACCEPTABLE ROW OR COLUMN.')
      GO TO 520
550   KJ = ROW
      II = ROW
      K = FWELL(ROW,3)
      I=FWELL(ROW,10)
      GO TO (210,320,190,190,350,360)COL
560   WRITE(IO,570)
570   FORMAT(5X,'MORE CHANGES? <YES/NO>?')
      GO TO 470
600   II=III
1180  WRITE(IO,1190)PDFTOR
1190  FORMAT(//5X,'THE STANDARD PEAK DAY SUPPLY IS ',F4.2,' TIMES'/
     *5X,'THE PEAK SEASONAL DAILY CAPACITY.'/
     *5X,'IS THIS ACCEPTABLE FOR ALL YOUR ZONES? <YES/NO>?'////)
720   READ(IN,140)ANS
      IF(ANS.EQ.'YES')GO TO 1200
      IF(ANS.EQ.'NO')GO TO 730
      WRITE(IO,145)
      GO TO 720
730   WRITE(IO,740)
740   FORMAT(5X,'IS THERE A CONSTANT VALUE THAT IS ACCEPTABLE?'
     *' <YES/NO>?')
750   READ(IN,140)ANS
      IF(ANS.EQ.'YES')GO TO 760
      IF(ANS.EQ.'NO')GO TO 1201
      WRITE(IO,145)
      GO TO 750
760   WRITE(IO,770)
770   FORMAT(5X,'ENTER PEAK DAY MULTIPLIER CONSTANT.?')
      READ(IN,/)PDFTOR
      IF(PDFTOR.LE.1)GO TO 1200
      WRITE(IO,775)
775   FORMAT(5X,'A PEAK DAY MULTIPLIER GREATER THAN 1 FOR'/
     *5X,'SUPPLY IS UNACCEPTABLE.')
      GO TO 760
1200  NPD=1
      Z=PDFTOR
1201  DO 1202 I=1,II
      IF(NPD.EQ.1)GO TO 1205
1207  WRITE(IO,1203) (FWELL(I,K),K=1,2),SLIST(FWELL(I,3))
1203  FORMAT(5X,'ENTER PEAK DAY MULTIPLIER FOR ZONE '2A1' OPTION  "'A1
     *'".?')
      READ(IN,/)Z
      IF(Z.LE.1)GO TO 1205
      WRITE(IO,775)
      GO TO 1207
1205  FWELL(I,11)=FWELL(I,7)*Z
1202  CONTINUE
      WRITE(IO,1210)OANDMP
1210  FORMAT(//5X,'THE STANDARD OPERATION AND MAINTENANCE WELL'/
     *5X,'COSTS FOR THIS MODEL ARE ',$F6.2,'/MG. (FOR PUMPS,'/
     *5X,'PIPELINE, ECT.)  IS THIS ACCEPTABLE FOR ALL YOUR'/
     *5X,'ZONES? <YES/NO>?'////)
1215  READ(IN,140)ANS
      IF(ANS.EQ.'YES')GO TO 1220
      IF(ANS.EQ.'NO')GO TO 1230
      WRITE(IO,145)
      GO TO 1215
1230  WRITE(IO,1240)
1240  FORMAT(5X,'IS THERE A CONSTANT VALUE THAT IS ACCEPTABLE?'
     *' <YES/NO>?')
1245  READ(IN,140)ANS
      IF(ANS.EQ.'YES')GO TO 1250
      IF(ANS.EQ.'NO')GO TO 1270
      WRITE(IO,145)
      GO TO 1245
1250  WRITE(IO,1260)
1260  FORMAT(5X,'ENTER O&M CONSTANT.?')
      READ(IN,/)OANDMP
      IF(OANDMP.LE.0)GO TO 1250
1220  OMP=1
1270  WRITE(IO,1280)PUMPC
1280  FORMAT(//5X,'THE STANDARD POWER COSTS FOR PUMPING IN THIS'/
     *5X,'MODEL ARE ',$F6.2,'/MG/100FT.  IS THIS ACCEPTABLE '/
     *5X,'FOR ALL YOUR ZONES? <YES/NO>?'////)
1290  READ(IN,140)ANS
      IF(ANS.EQ.'YES')GO TO 1340
      IF(ANS.EQ.'NO')GO TO 1300
      WRITE(IO,145)
      GO TO 1290
1300  WRITE(IO,1240)
1310  READ(IN,140)ANS
      IF(ANS.EQ.'YES')GO TO 1320
      IF(ANS.EQ.'NO')GO TO 1350
      WRITE(IO,145)
      GO TO 1310
1320  WRITE(IO,1330)
1330  FORMAT(5X,'ENTER POWER PUMP COSTS.?')
      READ(IN,/)PUMPC
      IF(PUMPC.LE.0)GO TO 1320
1340  PC=1
      X=OANDMP
      Y=PUMPC
1350  DO 1410 I=1,II
      IF(OMP.EQ.1)GO TO 1380
1355  WRITE(IO,1360)(FWELL(I,K),K=1,2),SLIST(FWELL(I,3))
1360  FORMAT(5X,'ENTER O&M COSTS FOR ZONE ',2A1,' OPTION  "'A1'".?')
      READ(IN,/)X
      IF(X.LE.0)GO TO 1355
1380  IF(PC.EQ.1)GO TO 1390
1385  WRITE(IO,1370) (FWELL(I,K),K=1,2),SLIST(FWELL(I,3))
1370  FORMAT(5X'ENTER PUMP POWER COSTS FOR ZONE '2A1' OPTION  "'A1'".?')
      READ(IN,/)Y
```

```
      IF(Y.LE.0)GO TO 1385
1390  DELELV=RESELV(FWELL(I,10))-FWELL(I,9)
      IF(DELELV)1421,1421,1422
1422  FWELL(I,12)=Y*DELELV/100. + X
      GO TO 1410
1421  FWELL(I,12)=X
1410  CONTINUE
609   COL=0
      ROW=0
      II=III
      WRITE(IO,610)
610   FORMAT(////5X,'THE FOLLOWING IS A LIST OF THE CALCUL'
     *'ATED WELL DATA.'//)
      WRITE(IO,611)
611   FORMAT(25X,'COL',7X,'COL'/26X,'1',9X,'2'/16X,'AVG.',3X,
     *'PEAK DAY',3X,'WATER'/16X,'FLOW',5X,'FLOW',5X,'COST'/
     *1X,'ROW ZONE ALT   MGD',6X,'MGD',6X,'$/MG')
      DO 612 KJ=1,II
612   WRITE(IO,613)KJ,(FWELL(KJ,J),J=1,2),SLIST(FWELL(KJ,3)),FWELL(KJ,7)
     *,FWELL(KJ,11), FWELL(KJ,12)
613   FORMAT(1X,I3,2X,2A1,3X,A1,2X,F6.2,3X,F6.2,4X,F7.2)
      IF(ROW.NE.0)GO TO 619
      WRITE(IO,460)
614   READ(IN,140)ANS
      IF(ANS.EQ.'YES')GO TO 617
      IF(ANS.EQ.'NO'.AND.ROW.NE.0)GO TO 615
      IF(ANS.EQ.'NO'.AND.ROW.EQ.0)GO TO 900
      WRITE(IO,145)
      GO TO 614
615   WRITE(IO,490)
616   READ(IN,140)ANS
      IF(ANS.EQ.'YES')GO TO 609
      IF(ANS.EQ.'NO')GO TO 900
      WRITE(IO,145)
      GO TO 616
617   WRITE(IO,530)
      READ(IN,/)ROW,COL
      IF(ROW.GT.0.AND.ROW.LE.III.AND.COL.GT.0.AND.COL.LE.2)GO TO 618
      WRITE(IO,540)
      GO TO 617
618   KJ=ROW
      II=ROW
      GO TO(620,630)COL
619   WRITE(IO,570)
      GO TO 614
620   WRITE(IO,1203)(FWELL(ROW,J),J=1,2),SLIST(FWELL(ROW,3))
      READ(IN,/)X
      IF(X.LE.1)GO TO 621
      WRITE(IO,775)
      GO TO 620
621   FWELL(ROW,11)=FWELL(ROW,7)*X
      GO TO 612
630   WRITE(IO,1360)(FWELL(ROW,J),J=1,2),SLIST(FWELL(ROW,3))
      READ(IN,/)X
      IF(X.LE.0)GO TO 630
635   WRITE(IO,1370)(FWELL(ROW,J),J=1,2),SLIST(FWELL(ROW,3))
      READ(IN,/)Y
      IF(Y.LE.0)GO TO 635
      DELELV=RESELV(FWELL(ROW,10))-FWELL(ROW,9)
      IF(DELELV)631,631,632
632   FWELL(ROW,12)=Y*DELELV/100. + X
      GO TO 612
631   FWELL(ROW,12)=X
      GO TO 612
      ENTRY FWE(ANSWE,SAVED)
900   IF(SAVED.EQ.1)GO TO 901
      INQUIRE(IF,LASTRECORD=N2)
      N2=N2+2
      WRITE(IF'N2,903)
903   FORMAT('ELEMENT FUTWEL')
      GO TO 902
901   WRITE(IF,903)
902   WRITE(IF,904)
904   FORMAT(4X,'TABLE FWELL,ZERO')
      WRITE(IF,905)
905   FORMAT(9X,'*',6X,'  CAPTL      CAP       PCAP    QANDM     NUM')
      IF(ANSWE.EQ.'NO')GO TO 909
      DO 906 I=1,III
906   WRITE(IF,907)(FWELL(I,J),J=1,2),SLIST(FWELL(I,3)),FWELL(I,5),
     * FWELL(I,7),(FWELL(I,J),J=11,12),FWELL(I,8)
907   FORMAT(9X,3A1,I9,3F9.2,I6)
909   IF(ANSWE.EQ.'NO')WRITE(IF,910)
910   FORMAT(9X,'DUM')
      WRITE(IF,908)
908   FORMAT('ENDATA')
      IF(ANSWE.EQ.'NO')GO TO 930
      IIII=IIII+III
      REWIND(IF)
      WRITE(IF,920)N,S,IIII
920   FORMAT(3I6)
930   SAVED=1
      RETURN
      END
```

PROPOSED FUTURE SPRINGS SUBROUTINE

```
       $SET SEPARATE
            SUBROUTINE FSPNGS(SAVED,                              )
            DIMENSION LISTZO(40,22),NS(4),PFLOW(11),PLIST(11),PCOST(11)
            DIMENSION FSPRNG(80,21),FTOR(4),NF(4),NG(4),ALTER(4),PALT(11)
            INTEGER S,ROW,COL,PLIST,TF
            REAL N1,K1,MAX
            COMMON IN,IO,IF,TF,LISTZO,N,S,IIII
            COMMON /CMN1/ PALT,PLIST,PFLOW,PCOST,FTOR
            COMMON /CMN3/ OM1,PC1,PD1,PD2,OM2
            COMMON /CMN6/ R7,N7,AK2,AE2
            DATA NF/4*'FLOW'/NS/'S1','S2','S3','S4'/NG/4*'CFS'/
            DATA ALTER/'A','B','C','D'/
            DO 6 I=1,11
     6      PFLOW(I)=PFLOW(I) / 449.
                          )ANDNS=OM2
                          ..PDFTOR=PD1
            CRF1 = R7 + (R7/((1.0 + R7)**N7 - 1.0))
            CRF=CRF1
            K1=AK2
            E1=AE2
            CAP6=INT((1000.*PCOST(1) + K1*FTOR(1)*6.**E1)/100.)*100
            WRITE(IO,90)
    90      FORMAT(////15X,10('*'),' SEGMENT 8 ',10('*'))
            WRITE(IO,9)
     9      FORMAT(///10X,'***** ENTER DATA FOR FUTURE SPRINGS *****'///)
            WRITE(IO,10)(PLIST(I),PCOST(I),I=1,11),K1,(FTOR(I),I=1,4),
           *E1,CAP6,6,1.11
    10      FORMAT(5X,'THE CALCULATING FORMULA FOR CAPITAL COSTS OF'/
           *5X,'FUTURE SPRINGS IN THIS MODEL IS OF THE FORM:'/
           *5X,' C = X*PIPE COST  +  K1*FTOR*X*D**E1  +  SDEV   WHERE:'/
           *8X,'C = TOTAL CAPITAL COST'/
           *8X,'D = PIPE DIAMETER IN INCHES'/
           *8X,'X = LENGTH OF LINE IN FEET'/
           *8X,'PIPE COST = COST OF PIPE PER FOOT'/
           *11(10X,I2,' INCH DIA. = ',$F7.2,'/FOOT'/)/
           *8X,'K1 = INSTALLATION CONSTANT MULTIPLIER (DEFAULT K1='F6.4')'/
           *8X,'FTOR = PIPE INSTALLATION DIFFICULTY FACTOR:'/
           *10X,'NORMAL EXCAVATION AND BACKFILL (DEFAULT'F4.1')'/
           *10X,'ROUGHER EXCAVATION (BUT NO RIPPING) AND SELECT BACKFILL '/
           *12X,'(DEFAULT'F4.1')'/
           *10X,'ROCK EXCAVATION AND BACKFILL FROM BORROW (DEFAULT'F4.1')'/
           *10X,'BELOW WATER EXCAVATION WITH GRAVEL BACKFILL (DEFAULT'F4.1')'/
           *8X,'E1 = INSTALLATION SCALE FACTOR EXPONENT (DEFAULT E1='F6.3')'/
           *8X,'SDEV = ON SITE SPRING DEVELOPMENT COSTS (NO DEFAULT)'/
           *5X,'DEFAULT VALUES GIVE A CAPITAL COST OF ABOUT 'K$I8' EXCLUDING'/
           *5X,'SPRING DEVELOPMENT COSTS FOR A 1000 FOOT 'I2' INCH LINE WITH'/
           *5X,'A ',F4.2,' CFS CAPACITY AND NORMAL EXCAVATION.'////)
            WRITE(IO,120)
   120      FORMAT(5X,'WILL THESE DEFAULT VALUES BE ACCEPTABLE FOR ALL'/
           *5X,'YOUR FUTURE SPRINGS? <YES/NO>?')
   130      READ(IN,140)ANS
   140      FORMAT(A6)
            IF(ANS.EQ.'NO')GO TO 155
            IF(ANS.EQ.'YES')GO TO 35
            WRITE(IO,145)
   145      FORMAT(5X,'PLEASE ANSWER YES OR NO.?')
            GO TO 130
   155      WRITE(IO,156)
   156      FORMAT(5X,'ARE THERE CONSTANT VALUES THAT WILL BE ACCEPT'
           *'ABLE'/5X,'FOR ALL YOUR FUTURE SPRINGS? <YES/NO>?')
   157      READ(IN,140)ANS
            IF (ANS.EQ.'NO')GO TO 162
            IF(ANS.EQ.'YES')GO TO 160
            WRITE(IO,145)
            GO TO 157
   160      WRITE(IO,161)
   161      FORMAT(5X,'ENTER THE VALUES FOR <PIPECOST(1-11),FTOR(1-4),K1,E1>.?
           *')
            READ(IN,/)PCOST,FTOR,K1,E1
    35      KE=1
   162      WRITE(IO,163)R7,N7
   163      FORMAT(//5X,'THE CAPITAL RECOVERY FACTOR (CRF) FORMULA'/
           *5X,'IS  CRF = R+ +R/[(1+R)**N - 1]! WHERE:'/10X,'R = INTEREST'
           *' RATE (DEFAULT R = ',F5.3,')'/10X,'N = NUMBER OF YEARS (DEFAU'
           *'LT N ='I3')'/5X,'ARE THESE VALUES ACCEPTABLE FOR ALL YOUR'
           *' FUTURE SPRINGS?<YES/NO>?'////)
   165      READ(IN,140)ANS
            IF(ANS.EQ.'NO')GO TO 166
            IF(ANS.EQ.'YES')GO TO 168
            WRITE(IO,145)
            GO TO 165
   166      WRITE(IO,156)
   167      READ(IN,140)ANS
            IF(ANS.EQ.'NO')GO TO 170
            IF(ANS.EQ.'YES')GO TO 169
            WRITE(IO,145)
            GO TO 167
   169      WRITE(IO,164)
   164      FORMAT(5X,'ENTER THE VALUES <N,R>.?')
            READ(IN,/)N1,R1
            IF(N1.LE.0 .OR. R1.LE.0)GO TO 169
            CRF1=R1 + (R1/((1.0+R1)**N1 - 1.0))
            CRF=CRF1
   168      NR=1
   170      WRITE(IO,177)
   177      FORMAT(////5X,'[NOTE]  SEASON ONE IS CONSIDERED THE PEAK SEASON'
           *' FOR THIS MODEL'/5X'AND THE MAXIMUM TOTAL NUMBER OF SPRINGS'
           *' IS 80.'////)
            DO 390 I=1,N
   175      WRITE(IO,180) (LISTZO(I,J),J=1,22)
   180      FORMAT(////5X,5('*'),1X,2A1,3X,20A1,1X,5('*'))
   173      WRITE(IO,171)
   171      FORMAT(5X,'ENTER THE NUMBER OF POTENTIAL SPRINGS FOR THIS'
           *' ZONE. <0-4>?')
            READ(IN,/)BB
            IF(BB.LE.0)GO TO 390
            IF(BB.GE.1.AND.BB.LE.4)GO TO 174
            WRITE(IO,172)
   172      FORMAT(5X,'PLEASE SELECT 0 TO 4 ONLY.')
            GO TO 173
   174      DO 380 K=1,BB
   650      CRF=CRF1
            WRITE(IO,651) ALTER(K),(LISTZO(I,J),J=1,2)
   651      FORMAT(//5X,'ENTER DISTANCE <FEET> FROM SPRING TO RESER'
           *'VOIR'/5X,'OR CONNECTION FOR SPRING "'A1'"  ZONE ',
           *2A1,'.?')
            READ(IN,/)FT
            IF(FT.LE.0)GO TO 391
            IF(ROW.NE.0)II=II-1
```

```
          II=II+1
          IF(ROW.NE.0)GO TO 670
          FSPRNG(II,1)=LISTZO(I,1)
          FSPRNG(II,2)=LISTZO(I,2)
          FSPRNG(II,3)=K
          FSPRNG(II,17)=I
    670   FSPRNG(II,4)=FT
          IF(ROW.NE.0)GO TO 504
    680   WRITE(IO,681)
    681   FORMAT(5X,'ENTER SPRING FLOW <CFS> FOR EACH SEASON.?')
          READ(IN,/) (FSPRNG(II,J),J=5,S+4)
          DO 700 J=5,S+4
          IF(FSPRNG(II,J).GT.0)GO TO 700
          WRITE(IO,720) J-4
    720   FORMAT(5X,'YOU HAVE ENTERED A NEGATIVE FLOW FOR SEASON '
         *I1,'.')
          GO TO 680
    700   CONTINUE
          MAX=0
          DO 730 J=5,S+4
          JJ=J+8
          FSPRNG(II,JJ)=FSPRNG(II,J)*0.6463
          IF(MAX.GE.FSPRNG(II,J))GO TO 730
          MAX=FSPRNG(II,J)
    730   CONTINUE
          DO 740 J=1,11
          IF(MAX.GT.PFLOW(J))GO TO 740
          SIZE=PLIST(J)
          FSPRNG(II,18)=J
          GO TO 750
    740   CONTINUE
          WRITE(IO,745)
    745   FORMAT(5X,'YOUR MAXIMUM SEASONAL FLOW IS LARGER THAN'/
         *5X,'ANY OF THE ELEVEN PIPE SIZES ALLOWED FOR IN THIS'/
         *5X,'MODEL.  PLEASE ENTER THE DIAMETER OF A PIPE THAT'/
         *5X,'WILL ALLOW THIS FLOW. <DIA. INCHES>?')
          READ(IN,/)FSPRNG(II,9)
          MAX=0
          GO TO 496
    750   WRITE(IO,751)MAX,SIZE
    751   FORMAT(5X,'YOUR MAXIMUM SEASONAL FLOW OF ',F6.2,' CFS'/
         *5X,'MAY BE TRANSFERED BY A ',I2,' INCH DIAMETER PIPE.'/
         *5X,'DO YOU AGREE? <YES/NO>?')
    755   READ(IN,140)ANS
          IF(ANS.EQ.'NO')GO TO 790
          IF(ANS.EQ.'YES')GO TO 870
          WRITE(IO,145)
          GO TO 755
    790   WRITE(IO,791)PLIST(1),PLIST(11)
    791   FORMAT(5X,'WHAT SIZE WOULD YOU RECOMMEND?<',I2,' TO ',
         *I2,'>?')
          READ(IN,/)SIZE
          DO 800 KK=1,11
          IF(SIZE.NE.PLIST(KK))GO TO 800
          FSPRNG(II,18)=KK
          IF(ROW.NE.0)GO TO 504
          GO TO 830
    800   CONTINUE
          WRITE(IO,801)
    801   FORMAT(5X,'PLEASE PICK FROM THE LISTED SIZES ONLY')
          GO TO 790
    830   IF(SIZE.GE.PLIST(J))GO TO 870
    840   WRITE(IO,841)
    841   FORMAT(5X,'THE SIZE YOU PICKED IS TOO SMALL TO TRANSFER'/
         *5X,'THE FLOW REQUIRED AT NORMAL OPERATING CONDITIONS.'/
         *5X,'DO YOU WANT TO MAINTAIN THAT SIZE? <YES/NO>?')
    845   READ(IN,140)ANS
          IF(ANS.EQ.'NO')GO TO 790
          IF(ANS.EQ.'YES')GO TO 870
          WRITE(IO,145)
          GO TO 845
    870   FSPRNG(II,9)=SIZE
          IF(ROW.NE.0)GO TO 504
    496   WRITE(IO,519)
    519   FORMAT(5X,'INDICATE THE TYPE OF PIPE INSTALLATION <1-4>.?')
          IF(II.NE.1)GO TO 505
          WRITE(IO,1502)
   1502   FORMAT(/10X,'<1> NORMAL EXCAVATION AND NORMAL BACKFILL'/10X,
         *'<2> ROUGHER EXCAVATION (BUT NO RIPPING) AND SELECT BACK'
         *'FILL'/10X,'<3> ROCK EXCAVATION AND BACKFILL FROM BORROW'/10X,
         *'<4> BELOW WATER EXCAVATION WITH GRAVEL BACKFILL?')
    505   READ(IN,/) X
          DO 501 J=1,4
          IF(X.NE.J)GO TO 501
          FSPRNG(II,10)=X
          GO TO 504
    501   CONTINUE
          WRITE(IO,502)
    502   FORMAT(5X,'PLEASE ENTER 1,2,3 OR 4 ONLY')
          GO TO 505
    504   IF(KE.EQ.1)GO TO 300
    210   WRITE(IO,220)ALTER(K),(FSPRNG(II,J),J=1,2)
    220   FORMAT(5X,'ENTER KNOWN TOTAL CAPITAL COSTS FOR SPRING  "',A1,
         *'"  ZONE ',2A1,'.?')
          READ(IN,/)FSPRNG(II,11)
          IF(FSPRNG(II,11).LE.0)GO TO 210
          GO TO 310
    300   IF(ROW.NE.0)GO TO 301
    302   WRITE(IO,225)ALTER(K),(FSPRNG(II,J),J=1,2)
    225   FORMAT(5X,'ENTER ON SITE DEVELOPMENT COSTS FOR SPRING  "',
         *A1,'"  ZONE ',2A1,'.?')
          READ(IN,/)FSPRNG(II,19)
          IF(FSPRNG(II,19).LE.0)GO TO 302
    301   FSPRNG(II,11)=FSPRNG(II,4)*PCOST(FSPRNG(II,18)) + K1*FTOR(FSPRN
         *G(II,10))*FSPRNG(II,4)*FSPRNG(II,9)**E1 + FSPRNG(II,19)
    310   IF(NR.EQ.1)GO TO 340
    320   WRITE(IO,330)
    330   FORMAT(5X,'ENTER CAPITAL RECOVERY FORMULA VALUES <N,R>.?')
          READ(IN,/)N1,R1
          IF(N1.LE.0 .OR. R1.LE.0)GO TO 320
          IF(N1.NE.0)CRF=R1 + (R1/((1.0+R1)**N1 - 1.0))
    340   FSPRNG(II,12)=INT(FSPRNG(II,11)*CRF/10.)*10
          FSPRNG(II,11)=INT(FSPRNG(II,11)/100.)*100
          CRF=CRF1
          IF(ROW.NE.0)GO TO 440
    380   CONTINUE
    391   IF(ROW.EQ.0)GO TO 390
          DO 392 KK=4,17
    392   FSPRNG(II,KK)=0
          GO TO 440
    390   CONTINUE
          III=II
```

```
400   WRITE(IO,410)
410   FORMAT(///5X,'THE FOLLOWING IS A LIST OF YOUR DATA'/)
      WRITE(IO,420)(K,K=1,6)
420   FORMAT(15X,'COL',3X,'COL',2X,'COL',4X,'COL',5X,'COL',
     *8X,'COL' /16X,I1,5X,I1,4X,I1,6X,I1,7X,I1,10X,I1,4X,
     *4('>'))
      WRITE(IO,430)(NF(J),J=1,S)
430   FORMAT(15X,'PIPE',2X,2('PIPE',1X)/14X,'LENGTH',1X,'SIZE',
     *1X,'INST',1X,2('CAPITAL',1X),2X,4A6)
      WRITE(IO,435)(NG(J),J=1,S)
435   FORMAT(1X,'ROW',1X,'ZONE',1X,'ALT',2X,'FEET',2X,'INCH',
     *1X,'TYPE',2X,'TOTAL',2X,'PERYEAR',3X,4A6)
      DO 440 KJ=1,II
440   WRITE(IO,450)KJ,(FSPRNG(KJ,J),J=1,2),ALTER(FSPRNG(KJ,3)),
     *FSPRNG(KJ,4),(FSPRNG(KJ,J),J=9,12),(FSPRNG(KJ,J),J=5,S+4)
450   FORMAT(2X,I2,2X,2A1,3X,A1,2X,I6,2X,I2,3X,I1,1X,$I9,$I8,1X,
     *4F6.1)
      IF(ROW.NE.0)GO TO 560
      WRITE(IO,460)
460   FORMAT(////5X,'ARE THERE ANY CHANGES REQUIRED IN THIS DATA?'
     *' <YES/NO>?')
470   READ(IN,140)ANS
      IF(ANS.EQ.'NO' .AND. ROW.EQ.0)GO TO 1000
      IF(ANS.EQ.'NO' .AND. ROW.NE.0)GO TO 480
      IF(ANS.EQ.'YES')GO TO 520
      WRITE(IO,145)
      GO TO 470
480   WRITE(IO,490)
490   FORMAT(5X,'DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>?')
500   READ(IN,140)ANS
      IF(ANS.EQ.'NO')GO TO 1000
      IF(ANS.EQ.'YES')GO TO 510
      WRITE(IO,145)
      GO TO 500
510   ROW = 0
      COL = 0
      II = III
      GO TO 400
520   WRITE(IO,530)
530   FORMAT(5X,'ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.?')
      READ(IN,/)ROW,COL
      IF(ROW.GT.0 .AND. ROW.LE.III .AND. COL.GT.0 .AND. COL.LE.6)
     *GO TO 550
      WRITE(IO,540)
540   FORMAT(5X,'UNACCEPTABLE ROW OR COLUMN.')
      GO TO 520
550   KJ = ROW
      II = ROW
      K = FSPRNG(ROW,3)
      I = FSPRNG(ROW,17)
      GO TO(650,790,496,210,320,680)COL
560   WRITE(IO,570)
570   FORMAT(5X,'MORE CHANGES? <YES/NO>?')
      GO TO 470
1000  ROW=0
      COL=0

1180  WRITE(IO,1190)PDFTOR
1190  FORMAT(//5X,'THE STANDARD PEAK DAY SUPPLY IS 'F4.2' TIMES THE'/
     *5X,'PEAK SEASONAL DAILY CAPACITY.  IS THIS ACCEPTABLE FOR ALL'/
     *5X,'YOUR ZONES? <YES/NO>?'////)
1720  READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 1200
      IF(ANS.EQ.'NO')GO TO 1730
      WRITE(IO,2)
      GO TO 1720
1730  WRITE(IO,1740)
1740  FORMAT(5X,'IS THERE A CONSTANT THAT IS ACCEPTABLE? '
     *'<YES/NO>?')
1750  READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 1760
      IF(ANS.EQ.'NO')GO TO 1201
      WRITE(IO,2)
      GO TO 1750
1760  WRITE(IO,1770)
1770  FORMAT(5X,'ENTER PEAK DAY MULTIPLIER CONSTANT. <PDC>?')
      READ(IN,/)PDFTOR
      IF(PDFTOR.LE.1)GO TO 1200
      WRITE(IO,1775)
1775  FORMAT(5X,'A PEAK DAY MULTIPLIER GREATER THAN 1 IS UNACC'
     *'EPTABLE.')
      GO TO 1760
1200  NPD=1
1201  X=PDFTOR
      DO 1202 I=1,III
      IF(NPD.EQ.1)GO TO 1205
1207  WRITE(IO,1203)ALTER(FSPRNG(I,3)),(FSPRNG(I,J),J=1,2)
1203  FORMAT(5X,'ENTER PEAK DAY MULTIPLIER FOR SPRING "'A1'"  ZONE '
     *2A1,'.?')
      READ(IN,/)X
      IF(X.LE.1)GO TO 1205
      WRITE(IO,1775)
      GO TO 1207
1205  FSPRNG(I,20)=X*FSPRNG(I,13)
      IF(ROW.NE.0)GO TO 1465
1202  CONTINUE

      WRITE(IO,1210)OANDMS
1210  FORMAT(//5X,'THE STANDARD OPERATION AND MAINTENANCE SPRING COSTS'/
     *5X,'FOR THIS MODEL ARE '$F6.2'/MG (FOR CHLORINATION, DESANDER'/
     *5X,'CLEANING AND MISC. O&M).  IS THIS ACCEPTABLE FOR ALL YOUR'/
     *5X,'ZONES? <YES/NO>?'////)
1215  READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO1220
      IF(ANS.EQ.'NO')GO TO 1230
      WRITE(IO,2)
2     FORMAT(5X,'PLEASE ANSWER YES OR NO')
1     FORMAT(A6)
      GO TO 1215
1230  WRITE(IO,1740)
1245  READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 1250
      IF(ANS.EQ.'NO')GO TO 1270
      WRITE(IO,2)
      GO TO 1245
1250  WRITE(IO,1260)
1260  FORMAT(5X,'ENTER O&M CONSTANT.?')
      READ(IN,/)OANDMS
      IF(OANDMS.GT.0)GO TO 1220
      WRITE(IO,1371)
      GO TO 1250
1220  OMS=1
```

```
1270  X=OANDMS
      DO 1410 I=1,III
      IF(OMS.EQ.1)GO TO 1370
1380  WRITE(IO,1360)ALTER(FSPRNG(I,3)),(FSPRNG(I,J),J=1,2)
1360  FORMAT(5X,'ENTER O&M COSTS FOR SPRING  "'A1'"  ZONE '2A1'.?')
      READ(IN,/)X
      IF(X.GT.0)GO TO 1370
      WRITE(IO,1371)
1371  FORMAT(5X,'O&M COSTS LESS THAN ZERO ARE UNACCEPTABLE.')
      GO TO 1380
1370  FSPRNG(I,21)=X
      IF(ROW.NE.0)GO TO 1465
1410  CONTINUE
1451  II=III
      WRITE(IO,1460)
1460  FORMAT(////5X,'THE FOLLOWING IS A LIST OF CALCULATED DATA.'///)
      WRITE(IO,1461)
1461  FORMAT(28X,'COL',5X,'COL'/29X,'1',7X,'2'/18X,'SPRING',2X,
     *'PEAK DAY  O&M'/10X,'SPRING   FLOW      FLOW    COST'/
     *' ROW ZONE  ALT    S1-MGD    MGD     $/MG')
      DO 1465 KJ=1,II
1465  WRITE(IO,1466)KJ,(FSPRNG(KJ,J),J=1,2),ALTER(FSPRNG(KJ,3)),
     *FSPRNG(KJ,13),(FSPRNG(KJ,J),J=20,21)
1466  FORMAT(2X,I2,2X,2A1,4X,A1,5X,F6.2,3X,F6.2,1X,F6.2)
      IF(ROW.NE.0)GO TO 1483
      WRITE(IO,460)
1470  READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO 1484
      IF(ANS.EQ.'NO' .AND. ROW.EQ.0)GO TO 900
      IF(ANS.EQ.'NO' .AND. ROW.NE.0)GO TO 1480
      WRITE(IO,2)
      GO TO 1470
1480  WRITE(IO,490)
1481  READ(IN,1)ANS
      IF(ANS.EQ.'YES')GO TO1482
      IF(ANS.EQ.'NO')GO TO 900
      WRITE(IO,2)
      GO TO 1481
1482  ROW=0
      COL=0
      GO TO 1451
1483  WRITE(IO,570)
      GO TO 1470
1484  WRITE(IO,530)
      READ(IN,/)ROW,COL
      IF(ROW.GT.0.AND.ROW.LE.III.AND.COL.GE.1.AND.COL.LE.2)GO TO 1485
      WRITE(IO,540)
      GO TO 1484
1485  KJ=ROW
      II=ROW
      I=ROW
      GO TO(1207,1380)COL
      ENTRY FSP(ANSWE,SAVED)
900   IF(SAVED.EQ.1)GO TO 910
      INQUIRE(IF,LASTRECORD=N2)
      N2=N2+2
      WRITE(IF=N2,920)
920   FORMAT('ELEMENT FUTSPG')
      GO TO 930
910   WRITE(IF,920)
930   WRITE(IF,940)
940   FORMAT(4X,'TABLE FSPRG,ZERO')
      WRITE(IF,950)(NF(I),I,I=1,S)
950   FORMAT(9X,'*    PIPE  CAPTL    COST  PDFLOW',4(3X,A4,I1))
      IF(ANSWE.EQ.'NO')GO TO 980
      DO 960 I=1,III
960   WRITE(IF,970)(FSPRNG(I,J),J=1,2),ALTER(FSPRNG(I,3)),FSPRNG(I,9)
     *,FSPRNG(I,12),FSPRNG(I,21),FSPRNG(I,20),(FSPRNG(I,J),J=13,S+12)
970   FORMAT(9X,3A1,2X,I2,I8,2F7.2,1X,4F8.2)
980   IF(ANSWE.IS.'NO')WRITE(IF,981)
981   FORMAT(9X,'DUM')
      WRITE(IF,990)
990   FORMAT('ENDATA')
      IF(ANSWE.EQ.'NO')GO TO 1010
      IIII=IIII+III
      REWIND(TF)
      WRITE(TF,1001)N,S,IIII
1001  FORMAT(3I6)
1010  SAVED=1
      RETURN
      END
```

PROPOSED FUTURE TREATMENT PLANTS SUBROUTINE

```
      $SET SEPARATE
            SUBROUTINE FTRPLS(SAVED            )
            DIMENSION ALTER(4),TEMP(4),FTRPL(80,14),LISTZO(40,22)
            DIMENSION PLANT(4),CAP(4),MGD(4),NS(4),OAM(4),CST(4),CS(4),COLN(4)
            INTEGER ROW,COL,TF
            COMMON IN,IO,IF,TF,LISTZO,N,S,IIII
            COMMON /CMN3/ OM1,PC1,PD1
            COMMON /CMN7/ R8,N8
            DATA ALTER/'A','B','C','D'/
            DATA PLANT/4*'PLANT'/,CAP/4*' CAP '/,MGD/4*' MGD '/,COLN/4*'COL'/
            DATA CS/4*' CST '/OAM/4*' O&M '/CST/4*' $/MG'/
            DATA NS/' S-1 ',' S-2 ',' S-3 ',' S-4 '/
            CRF1=R8 + (R8/((1.0+R8)**N8 - 1.0))
            CRF=CRF1
                  PDFTOR=PD1
            WRITE(IO,90)
      90    FORMAT(////17X,10('*'),' SEGMENT 9 ',10('*'))
            WRITE(IO,100)
      100   FORMAT(////8X,'***** ENTER DATA FOR FUTURE TREATMENT PLANTS '
           *'*****'////)
            WRITE(IO,101)
      101   FORMAT(5X,'[NOTE] YOU ARE ALLOWED UP TO FOUR ALTERNATE SIZE '
           *'PLANTS PER ZONE.'///)
      180   WRITE(IO,190)PDFTOR
      190   FORMAT(//5X,'THE STANDARD PEAK DAY SUPPLY IS ',F4.2,' TIMES'/
           *5X,'THE PEAK SEASONAL DAILY CAPACITY.'/
           *5X,'IS THIS ACCEPTABLE FOR ALL YOUR ZONES? <YES/NO>?'////)
      720   READ(IN,1)ANS
      1     FORMAT(A6)
            IF(ANS.EQ.'YES')GO TO 200
            IF(ANS.EQ.'NO')GO TO 730
            WRITE(IO,2)
      2     FORMAT(5X,'PLEASE ANSWER YES OR NO')
            GO TO 720
      730   WRITE(IO,740)
      740   FORMAT(5X,'IS THERE A CONSTANT VALUE THAT IS ACCEPTABLE?'
           *' <YES/NO>?')
      750   READ(IN,1)ANS
            IF(ANS.EQ.'YES')GO TO 760
            IF(ANS.EQ.'NO')GO TO 201
            WRITE(IO,2)
            GO TO 750
      760   WRITE(IO,770)
      770   FORMAT(5X,'ENTER PEAK DAY MULTIPLIER CONSTANT.?')
            READ(IN,/)PDFTOR
            IF(PDFTOR.LE.1 .AND. PDFTOR.GT.0)GO TO 200
            IF(PDFTOR.LE.0)GO TO 760
            WRITE(IO,775)
      775   FORMAT(5X,'A PEAK DAY MULTIPLIER GREATER THAN 1 FOR '
           *'SUPPLY IS UNACCEPTABLE.')
            GO TO 760
      200   NPD=1
      201   WRITE(IO,163)R8,N8
      163   FORMAT(//5X,'THE CAPITAL RECOVERY FACTOR (CRF) FORMULA'/
           *5X,'IS  CRF = R+ +R/[(1+R)**N - 1]! WHERE:'/10X,'R = INTEREST'
           *' RATE (DEFAULT R = ',F5.3,')'/10X,'N = NUMBER OF YEARS (DEFAU'
           *'LT N ='I3')'/5X,'ARE THESE VALUES ACCEPTABLE FOR ALL YOUR'
           *' FUTURE PLANTS? <YES/NO>?'////)
      162   READ(IN,1)ANS
            IF(ANS.EQ.'NO')GO TO 166
            IF(ANS.EQ.'YES')GO TO 168
            WRITE(IO,2)
            GO TO 162
      166   WRITE(IO,740)
      167   READ(IN,1)ANS
            IF(ANS.EQ.'NO')GO TO 170
            IF(ANS.EQ.'YES')GO TO 169
            WRITE(IO,2)
            GO TO 167
      169   WRITE(IO,164)
      164   FORMAT(5X,'ENTER THE VALUES <N,R>.?')
            READ(IN,/)N1,R1
            IF(N1.LE.0 .OR. R1.LE.0)GO TO 169
            CRF1=R1 + (R1/((1.0+R1)**N1 - 1.0))
            CRF=CRF1
      168   NR=1
      170   WRITE(IO,177)
      177   FORMAT(////5X,'[NOTE]  SEASON ONE IS CONSIDERED THE PEAK SEASON'
           * ' FOR THIS MODEL.'////)
            DO 110 I=1,N
            WRITE(IO,115)(LISTZO(I,L),L=1,22)
      115   FORMAT(///3X,5('*'),2A1,2X,20A1,5('*')///)
      116   WRITE(IO,117)
      117   FORMAT(5X,'ENTER THE NUMBER OF PROPOSED TREATMENT PLANTS '
           *'IN THIS ZONE. <0 - 4>?')
            READ(IN,/)NTP
            IF(NTP.EQ.0)GO TO 110
            IF(NTP.GT.0 .AND. NTP.LE.4)GO TO 130
      118   WRITE(IO,119)
      119   FORMAT(5X,'PLEASE ENTER ONLY  0 - 4.')
            GO TO 116
      130   DO 125 K=1,NTP
            KK=0
            X=PDFTOR
            WRITE(IO,105)(LISTZO(I,L),L=1,2),ALTER(K)
      105   FORMAT(///5X,'ENTER FUTURE TREATMENT PLANT CAPACITIES <MGD> FOR'
           */5X,'EACH SEASON SEPARATED BY COMMAS FOR ZONE '2A1' TREATMENT '
           *'PLANT  "'A1'".?')
            READ(IN,/)(TEMP(J),J=1,S)
            II=II+1
            FTRPL(II,1)=LISTZO(I,1)
            FTRPL(II,2)=LISTZO(I,2)
            DO 120 J=1,S
            IF(TEMP(J).LE.0)GO TO 120
            KK=KK+1
            FTRPL(II,J+2)=TEMP(J)
      120   CONTINUE
            IF(KK.NE.0)GO TO 135
            II=II-1
            GO TO 125
      135   FTRPL(II,12)=ALTER(K)
            IF(NPD.EQ.1)GO TO 150
      137   WRITE(IO,140)(FTRPL(II,L),L=1,2),FTRPL(II,12)
      140   FORMAT(5X,'ENTER PEAK DAY MULTIPLIER FOR ZONE '2A1' PLANT  "'
           *A1'".?')
            READ(IN,/)X
            IF(X.LE.1)GO TO 150
            WRITE(IO,775)
```

```
      GO TO 137
150   FTRPL(II,7)=X * FTRPL(II,3)
      WRITE(IO,160)(FTRPL(II,L),L=1,2),FTRPL(II,12)
160   FORMAT(5X,'ENTER O&M COSTS <$/MG> FOR EACH SEASON FOR ZONE '2A1
     *' PLANT  "'A1'".?')
      READ(IN,/)(TEMP(L),L=1,S)
      DO 165 J=1,S
      IF(TEMP(J).LE.0)GO TO 165
      FTRPL(II,J+7)=TEMP(J)
165   CONTINUE
290   WRITE(IO,300)(FTRPL(II,L),L=1,2),FTRPL(II,12)
300   FORMAT(5X,'ENTER TOTAL CAPITAL COSTS FOR ZONE '2A1' PLANT  "'
     *A1,'".?')
      READ(IN,/)FTRPL(II,13)
      IF(NR.EQ.1)GO TO 310
304   WRITE(IO,305)(FTRPL(II,L),L=1,2),FTRPL(II,12)
305   FORMAT(5X,'ENTER  CRF  VALUES <N,R> FOR ZONE '2A1' PLANT  "'
     *A1,'".?')
      READ(IN,/)N1,R1
      IF(N1.LE.0 .OR. R1.LE.0)GO TO 304
      CRF=R1 + (R1/((1.0+R1)**N1 - 1.0))
310   FTRPL(II,14)=INT((FTRPL(II,13)*CRF)/10.)*10
      FTRPL(II,13)=INT(FTRPL(II,13)/100.)*100
      IF(ROW.NE.0)GO TO 440
125   CONTINUE
110   CONTINUE
      III=II
420   WRITE(IO,400)
400   FORMAT(////5X,'THE FOLLOWING IS A LIST OF YOUR DATA.'///)
      WRITE(IO,410)(COLN(L),L=1,S),(COLN(L),L=1,S)
410   FORMAT(12X,'COL',5X,'COL',8(3X,A3))
      WRITE(IO,415)(3,L=1,S),(4,L=1,S)
415   FORMAT(13X,'1',7X,'2',8(5X,I1))
      WRITE(IO,421)(PLANT(L),L=1,S),(DAM(L),L=1,S)
421   FORMAT(20X,'PEAK',8(1X,A5))
      WRITE(IO,425)(CAP(L),L=1,S),(NS(L),L=1,S)
425   FORMAT(5X,'ZONE',11X,'DAY',1X,8(1X,A5))
      WRITE(IO,430)(NS(L),L=1,S),(CST(L),L=1,S)
430   FORMAT(6X,'&   CAPITAL   CAP',1X,8(1X,A5))
      WRITE(IO,435)(MGD(L),L=1,S)
435   FORMAT(' ROW  ALT  TOTAL     MGD ',4(1X,A5))
      DO 440 KJ=1,II
440   WRITE(IO,445)KJ,(FTRPL(KJ,J),J=1,2),(FTRPL(KJ,J),J=12,13),
     *FTRPL(KJ,7),(FTRPL(KJ,J),J=3,S+2),(FTRPL(KJ,J),J=8,S+7)
445   FORMAT(2X,I2,1X,2A1,1X,A1,$I8,1X,9F6.1)
      IF(ROW.NE.0)GO TO 560
      WRITE(IO,460)
460   FORMAT(////5X,'ARE THERE ANY CHANGES REQUIRED IN THIS DATA?'
     *' <YES/NO>?')
470   READ(IN,1)ANS
      IF(ANS.EQ.'NO' .AND. ROW.EQ.0)GO TO 800
      IF(ANS.EQ.'NO' .AND. ROW.NE.0)GO TO 480
      IF(ANS.EQ.'YES')GO TO 520
      WRITE(IO,2)
      GO TO 470
480   WRITE(IO,490)
490   FORMAT(5X,'DO YOU WANT THE DATA LISTED AGAIN? <YES/NO>?')
500   READ(IN,1)ANS
      IF(ANS.EQ.'NO')GO TO 800
      IF(ANS.EQ.'YES')GO TO 510
      WRITE(IO,2)
      GO TO 500
510   ROW = 0
      COL = 0
      II = III
      GO TO 420
520   WRITE(IO,530)
530   FORMAT(5X,'ENTER ROW NUMBER , COLUMN NUMBER OF CHANGE.?')
      READ(IN,/)ROW,COL
      IF(ROW.GT.0 .AND. ROW.LE.III .AND. COL.GT.0 .AND. COL.LE.4)
     *GO TO 550
      WRITE(IO,540)
540   FORMAT(5X,'UNACCEPTABLE ROW OR COLUMN.')
      GO TO 520
550   KJ = ROW
      II = ROW
      GO TO(290,600,620,640)COL
560   WRITE(IO,570)
570   FORMAT(5X,'MORE CHANGES? <YES/NO>?')
      GO TO 470
600   WRITE(IO,140)(FTRPL(ROW,J),J=1,2),FTRPL(ROW,12)
      READ(IN,/)X
      IF(X.LE.1)GO TO 601
      WRITE(IO,775)
      GO TO 600
601   FTRPL(ROW,7)=FTRPL(ROW,3) * X
      GO TO 440
620   WRITE(IO,105)(FTRPL(ROW,J),J=1,2),FTRPL(ROW,12)
      READ(IN,/)(FTRPL(ROW,J),J=3,S+2)
      IF(NPD.NE.1)GO TO 600
      FTRPL(ROW,7)=PDFTOR * FTRPL(ROW,3)
      GO TO 440
640   WRITE(IO,160)(FTRPL(ROW,J),J=1,2),FTRPL(ROW,12)
      READ(IN,/)(FTRPL(ROW,J),J=8,S+7)
      GO TO 440
      ENTRY FTP(ANSWE,SAVED)
800   IF(SAVED.EQ.1)GO TO 801
      INQUIRE(IF,LASTRECORD=N2)
      N2=N2+2
      WRITE(IF=N2,810)
810   FORMAT('ELEMENT FUTRPL')
      GO TO 802
801   WRITE(IF,810)
802   WRITE(IF,820)
820   FORMAT(4X,'TABLE FUTRTRPL,ZERO')
      WRITE(IF,830)(CAP(J),J,J=1,S)
830   FORMAT(9X,'*           CAPTL  PDFLOW',1X,4(3X,A4,I1))
      IF(ANSWE.EQ.'NO')GO TO 855
      DO 840 I=1,III
840   WRITE(IF,850)(FTRPL(I,J),J=1,2),FTRPL(I,12),FTRPL(I,14),FTRPL(I,7)
     *, (FTRPL(I,J),J=3,S+2)
850   FORMAT(9X,3A1,2X,I8,2X,F6.1,1X,4F8.2)
      WRITE(IF,851)(CS(J),J,J=1,S)
851   FORMAT(9X,'*',7X,4(3X,A4,I1))
      DO 852 I=1,III
852   WRITE(IF,853)(FTRPL(I,J),J=1,2),FTRPL(I,12),(FTRPL(I,J),J=8,S+7)
853   FORMAT(9X,3A1,5X,4F8.2)
855   IF(ANSWE.IS.'NO')WRITE(IF,856)
856   FORMAT(9X,'DUM')
      WRITE(IF,860)
860   FORMAT('ENDATA')
      IF(ANSWE.EQ.'NO')GO TO 890
```

```
      IIII=IIII+III
      REWIND(TF)
      WRITE(TF,870)N,S,IIII
870   FORMAT(3I6)
890   SAVED=1
      RETURN
      END
```

EXECUTION SUGGESTIONS SUBROUTINE

```
$SET SEPARATE
      SUBROUTINE QUEST(I)
      COMMON IN,IO
      COMMON /CMN8/ TIME
      WRITE(IO,90)
90    FORMAT(////5X,'THE FOLLOWING ARE SUGGESTIONS FOR RUNNING YOUR '
     *'MODEL.'////)
      IF(I.EQ.0)GO TO 150
      IF(I.LE.40)GO TO 120
      WRITE(IO,100)TIME
100   FORMAT(5X,'YOUR MODEL CONTAINS MORE THAN FORTY INTEGER VARIABLES'
     */5X,'AND SHOULD BE CLASSIFIED AS A LARGE MODEL.  THE RUN TIMES'/
     *5X,'WILL PROBABLY BE IN EXCESS OF 'I3' MINUTES OF CPU.  IT IS'/
     *5X,'SUGGESTED THAT AFTER THE MATRIX IS GENERATED AND THE LP'/
     *5X,'SOLUTION IS FOUND FEASIBLE YOU RUN EITHER INTERACTIVE OR'/
     *5X,'MODIFIED BATCH AS SHOWN IN THE USERS MANUAL CHAPTER IV.'////)
      GO TO 130
120   WRITE(IO,110)TIME
110   FORMAT(5X,'YOUR MODEL CONTAINS LESS THAN FORTY INTEGER VARIABLES'
     */5X,'AND SHOULD BE CLASSIFIED AS A SMALL MODEL.  THE RUN TIMES'/
     *5X,'WILL PROBABLY BE LESS THAN 'I3' MINUTES OF CPU.  THERE'/
     *5X'SHOULD BE NO NEED TO RUN THIS MODEL BY MODIFIED INTERACTIVE'/
     *5X'OR MODIFIED BATCH BUT IF DESIRED REFER TO USERS MANUAL'/
     *5X'CHAPTER IV.'////)
      GO TO 130
150   WRITE(IO,155)
155   FORMAT(5X,'YOUR MODEL DOES NOT CONTAIN ANY INTEGER VARIABLES.'/
     *5X,'THEREFORE YOUR MODEL IS OF THE  LP  FORM AND SHOULD RUN'/
     *5X,'QUICKLY.  REFERENCE USERS MANUAL CHAPTER IV.'///)
130   WRITE(IO,140)
140   FORMAT(5X,'THIS ENDS THE DATA INPUT PHASE.  THE NEXT PHASE IS TO'/
     *5X,'GENERATE THE MODEL MATRIX BY EXECUTING  "TEMPO"  AND'/
     *5X,'INVOKE THE MACRO CREATOR.  (REF. USERS MANUAL CHAPTER IV)'
     */////)
      RETURN
      END
```

NUMERIC CONSTANTS SUBROUTINE

```
$SET SEPARATE
      SUBROUTINE DATA
      DIMENSION PLIST(11),PFLOW(11),PCOST(11),FTOR(4),PALT(11)
      INTEGER PLIST,TF
      COMMON IN,IO,IF,TF
      COMMON /CMN1/ PALT,PLIST,PFLOW,PCOST,FTOR
      COMMON /CMN3/ OM1,PC1,PD1,PD2,OM2
      COMMON /CMN4/ R5,N5,AK1,AE1,AXPMP
      COMMON /CMN5/ AA1,ZZ1,R6,N6
      COMMON /CMN6/ R7,N7,AK2,AE2
      COMMON /CMN7/ R8,N8
      COMMON /CMN8/ TIME
C*  ALTERNATE PIPE SIZE DESIGNATIONS
      DATA PALT/'A','B','C','D','E','F','G','H','I','J','K'/
C*  ALTERNATE PIPE DIA. (INCHES)
      DATA PLIST/6,8,10,12,14,16,18,20,24,30,36/
C*  FLOW GPM AT 6 FT/SEC
      DATA PFLOW/500,940,1500,2060,2800,3660,4700,5800,8300,
     * 13000,18700/
C*  COST OF PIPE $/FT.
      DATA PCOST/2.3,3.4,4.95,6.79,8.26,10.20,14.97,18.19,25.34,
     *39.71,56.79/
C*  PIPE INSTALLATION DIFFICULTY FACTORS
      DATA FTOR/1.0,1.7,6.0,3.0/
      IN=5           % INPUT FILE (REMOTE TERMINAL) NUMBER
      IO=6           % OUTPUT FILE (REMOTE TERMINAL) NUMBER
      IF=10          % MODELDATA FILE NUMBER
      TF=11          % TEMPDATA FILE NUMBER
      PC1=15.00      % STANDARD PUMP POWER COSTS
      PD1=0.70       % STANDARD PEAK DAY DEMAND MULTIPLIER
      PD2=1.10       % STANDARD PEAK DAY SUPPLY MULTIPLIER
      OM1=8.50       % STANDARD O&M COSTS WELLS
      OM2=6.60       % STANDARD O&M COSTS SPRINGS
      R5=0.06        % STANDARD CRF INTEREST RATE PIPES
      N5=40          % STANDARD CRF YEARS PIPES
      AK1=0.1426     % STD CONSTANT FOR CAPITAL COSTS INSTL. PIPES
      AE1=0.70       % STD EXPONENT FOR CAPITAL COSTS INSTL. PIPES
      AXPMP=0.535    % STD EXPONENT FOR O&M COSTS PIPE SUPPLY
      AA1=2010.      % STD CONSTANT FOR CAPITAL COSTS WELLS
      ZZ1=0.453      % STD EXPONENT FOR CAPITAL COSTS WELLS
      R6=0.06        % STD CRF INTEREST RATE WELLS
      N6=25          % STD CRF YEARS WELLS
      R7=0.06        % STD CRF INTEREST RATE SPRINGS
      N7=40          % STD CRF YEARS SPRINGS
      AK2=0.1426     % STD CONSTANT FOR CAPITAL COSTS INSTL. PIPES-SPRINGS
      AE2=0.70       % STD EXPONENT FOR CAPITAL COSTS INSTL. PIPES-SPRINGS
      R8=0.06        % STD CRF INTEREST RATE TREATMENT PLANTS
      N8=25          % STD CRF YEARS TREATMENT PLANTS
      TIME=30.       % APPROXIMATE CPU TIME SMALL-LARGE MODEL
      RETURN
      END
```

Appendix C

Sample Problem Data Input Worksheets

DATA INPUT WORKSHEET (SEGMENTS 1 AND 2)

SEASON	1	2	3	4
DAYS	70	100	90	105
TERM	Jun 1-Aug 9	Aug 10-Nov 17	Nov 18-Feb 16	Feb 17-Jun 30

ZONE NO. 1-98	ZONE NAME (<18 CHARACTERS)	POPULATION	RESERVOIR ELEVATION (FEET)	DEMAND (GAL/PERSON/DAY) SEASON 1	2	3	4	PEAK DAY MULTIPLIER
01	Mytown	25000	4000	300	250	200	250	1.2
14	Parksville	100000	3700	250	200	150	190	1.2
32	Yourtown	17000	3600	275	250	200	250	1.2
22	Farmtown	8500	3500	900	700	400	800	1.2
56	Half Way Stop	1120	3600	190	180	180	190	1.2

DATA INPUT WORKSHEET SEGMENT 3 EXISTING WELLS

ZONE NO.	NUMBER PER ZONE	WELL ELEVATION (FEET)	WELL CAPACITY (GPM)	PEAK DAY MULTIPLIER	O & M COSTS ($/MG)	POWER COSTS ($/MG/100')
01	1	3800	1000	0.9	9.25	STD(15.00)
14	2	3600	500	0.85	9.25	STD
----	----	3500	2000	0.9	0.25	STD
32	0	----	----	----	----	----
22	0	----	----	----	----	----
56	1	3600	100	0.5	9.25	STD

DATA INPUT WORKSHEET SEGMENT 4 EXISTING SPRINGS

ZONE NO.	SEASONAL FLOW (cfs) 1	2	3	4	PEAK DAY MULTIPLIER	O & M ($/MG)
01	0	0	0	0	----	----
14	0	0	0	0	----	----
32	10	8	5	9	STD (0.7)	STD (6.60)
22	15	12	8	10	STD	STD
56	0	0	0	0	----	----

DATA INPUT WORKSHEET SEGMENT 5 EXISTING TREATMENT PLANTS

ZONE NO.	NUMBER PER ZONE	SEASONAL CAPACITY (MGD) 1	2	3	4	SEASONAL O & M COSTS ($/MG) 1	2	3	4	PEAK DAY MULTIPLIER
01	0	----	----	----	----	----	----	----	----	----
14	1	2.0	2.0	2.0	2.0	123.	123.	123.	123.	STD(.7)
32	0	----	----	----	----	----	----	----	----	----
22	0	----	----	----	----	----	----	----	----	----
56	0	----	----	----	----	----	----	----	----	----

DATA INPUT WORKSHEET SEGMENT 6 — EXISTING AND PROPOSED TRANSFER FACILITIES

ZONE NO.	CONNECTED TO ZONE	EXISTING SIZE DIA. (INCH)	DISTANCE (FEET)	REVERSE FLOW (YES/NO)	NO. OF PIPE OPTIONS (1-4)	TYPE OF PIPE INSTL. (1-4)	CAPITAL RECOVERY FACTOR VALUES	
							N	R
01	14	None	23000	Yes	2	1	40	0.07
01	22	6 in.	26500	No	1	2	40	0.07
14	56	None	20000	No	3	3	40	0.07
14	32	8 in.	42500	Yes	1	4	40	0.07
32	22	None	48000	No	2	2	40	0.07
56	22	None	20000	Yes	3	2	40	0.07

DATA INPUT WORKSHEET SEGMENT 7 — PROPOSED WELLS

ZONE NO.	NUMBER OF WELLS IN ZONE (0-4)	WELL CAPACITY (GPM)	NUMBER OF WELLS OF THIS SIZE	WELL ELEVATION (FEET)	PEAK DAY MULTIPLIER	O & M COSTS ($/MG)	POWER COSTS ($/MG/100')	CAPITAL RECOVERY FACTOR VALUES	
								N	R
01	1	1000	3	3700	STD(0.70)	9.25	STD(15.00)	STD	STD
14	2	1500	1	3550	STD	9.25	STD	STD	STD
----	----	2500	1	3500	STD	9.25	STD	STD	STD
32	1	1500	2	3400	STD	9.25	STD	STD	STD
22	1	1500	2	3500	STD	9.25	STD	STD	STD
56	0	----	----	----	----	----	----	----	----

DATA INPUT WORKSHEET SEGMENT 8 — PROPOSED SPRINGS

ZONE NO.	CAPITAL RECOVERY FACTOR VALUES		NO. OF SPRINGS IN ZONE (0-4)	PIPE LENGTH (FEET)	SEASONAL CAPACITY (CFS)				PIPE DIA. (INCH)	TYPE OF PIPE INSTL (1-4)	ON SITE DEV. COSTS ($)	PEAK DAY MULTIPLIER	O & M COSTS ($/MG)
	N	R			1	2	3	4					
01	STD	STD	1	100000	15	12	7	9	24	1	20000	0.60	STD (6.60)
14	STD	STD	1	150000	25	20	12	21	36	3	43000	0.60	STD
32	STD	STD	1	200000	8	6	4	7	14	1	12000	0.60	STD
22	----	----	----	----	----	----	----	----	----	----	----	----	----
56	----	----	----	----	----	----	----	----	----	----	----	----	----

DATA INPUT WORKSHEET SEGMENT 9 — PROPOSED TREATMENT PLANT

ZONE NO.	PEAK DAY MULTIPLIER	CAPITAL RECOVERY FACTOR VALUES		NUMBER OF TR. PLNTS. IN ZONE (0-4)	SEASONAL CAPACITY (MGD)				SEASONAL O & M COSTS ($/MG)				TOTAL CAPITAL COST ($)
		N	R		1	2	3	4	1	2	3	4	
01	----	----	----	----	----	----	----	----	----	----	----	----	----
14	.8	STD (25)	STD(0.06)	1	20	20	20	20	65	73	81	70	4.53×10^6
32	.9	STD	STD	1	5	5	5	5	78	94	123	82	1.05×10^6
22	----	----	----	----	----	----	----	----	----	----	----	----	----
56	----	----	----	----	----	----	----	----	----	----	----	----	----

Appendix D

GAMMA Listing For Model And Report Generation

```
DATA
*
*START OF DATA DEFINATION SECTION
*
     LIST (A)
          A
          B
          C
          D
*
*PIPE SIZES
*
     TABLE PIPES
          *    SIZE
          A      6
          B      8
          C     10
          D     12
          E     14
          F     16
          G     18
          H     20
          I     24
          J     30
          K     36
     LIST (P),T=30
          A  " 6 INCH DIA. PIPE"
          B  " 8 INCH DIA. PIPE"
          C  "10 INCH DIA. PIPE"
          D  "12 INCH DIA. PIPE"
          E  "14 INCH DIA. PIPE"
          F  "16 INCH DIA. PIPE"
          G  "18 INCH DIA. PIPE"
          H  "20 INCH DIA. PIPE"
          I  "24 INCH DIA. PIPE"
          J  "30 INCH DIA. PIPE"
          K  "36 INCH DIA. PIPE"
     LIST (R)
          (P)
          X
*
*
*INCLUDING THE DATA FROM DATA INPUT PROGRAM
*
*
INCLUDE DEMAND, FROM TAPE=MODELDATA.
INCLUDE SEASON, FROM TAPE=MODELDATA.
INCLUDE EXWELL, FROM TAPE=MODELDATA.
INCLUDE EXSPRG, FROM TAPE=MODELDATA.
INCLUDE EXTRPL, FROM TAPE=MODELDATA.
INCLUDE PIPENT, FROM TAPE=MODELDATA.
INCLUDE FUTWEL, FROM TAPE=MODELDATA.
INCLUDE FUTSPG, FROM TAPE=MODELDATA.
INCLUDE FUTRPL, FROM TAPE=MODELDATA.
*
*
*CREATE LISTS FROM INCLUDED TABLES
*
*
     LIST (ZEW)
          (EXSTWELL, ,*)
     LIST (ZS)
          (EXSTSPRG, ,*)
     LIST (ZET)
          (EXSTTRPL, ,*)
     LIST (FWL)
          (FWELL, ,*)
     LIST (FSP)
          (FSPRG, ,*)
     LIST (FTP)
          (FUTRTRPL, ,*)
     LIST (ATOB1)
          (PIPENET, ,*)
     LIST (BTOA)
          (ATOB)(ATOB):00111100:
*
*
*TABLE FOR PEAK DAY MULTIPLIER FOR OBJECTIVE FUNCTION VALUES
*
*
     TABLE DATA
          *      DATA
          PDZ    0.01
*
*END OF DATA DEFINATION SECTION
*
****************************************************************
****************************************************************
****************************************************************
*
*START OF MODEL ROW DEFINATION
*
PROBLEM MODEL
*
*OBJECTIVE ROW -- MINIMIZE COST
*
     OBJECT,N
*
*DEMAND ROWS
*
     D(ZO)(S),G
          RHS1,RHS=(DEMAND,(ZO),DEMAND(S))
*
*FLOW FROM EXISTING WELLS -- SUPPLY
*
     FW(ZEW)(S),L,IF((ZEW).NM. DUM)
          RHS1,RHS=(EXSTWELL,(ZEW),FLOW)*(SEASONS,(S),DAYS)
*
*FLOW FROM EXISTING SPRINGS -- SUPPLY
*
     FS(ZS)(S),E,IF((ZS).NM. DU)
          RHS1,RHS=(EXSTSPRG,(ZS),FLOW(S))*(SEASONS,(S),DAYS)
*
*FLOW FROM EXISTING TREATMENT PLANTS -- SUPPLY
*
     FTP(ZET)(S),L,IF((ZET).NM. DUM)
          RHS1,RHS=(EXSTTRPL,(ZET),CAP(S))*(SEASONS,(S),DAYS)
*
*FLOW FROM FUTURE WELLS -- SUPPLY
```

```
*
     FFW(FWL)(S),L,IF((FWL).NM. DUM)
*
*FLOW FROM FUTURE SPRINGS -- SUPPLY
*
     FFS(FSP)(S),E,IF((FSP).NM. DUM)
*
*FLOW FROM FUTURE TREATMENT PLANTS -- SUPPLY
*
     FFTP(FTP)(S),L,IF((FTP).NM. DUM)
*
*INTERZONAL TRANSFERS A TO B  AND  B TO A
*
     Z(ATOB1)(S),L,IF((ATOB1).NM. DUMMY)
          RHS1,RHS=(PIPENET,(ATOB1),CAPAC)*(SEASONS,(S),DAYS)
               ,IF((PIPENET,(ATOB1),CAPTL).EQ.0)
     Z(BTOA1/ATOB1)(S),L,IF((ATOB1).NM. DUMMY .AND. (PIPENET,(ATOB1),
               BA(S)).NE.0)
          RHS1,RHS=(PIPENET,(ATOB1),CAPAC)*(SEASONS,(S),DAYS)
               ,IF((PIPENET,(ATOB1),CAPTL).EQ.0)
*
*
*PEAK DAY DEMANDS
*
     PD(ZO),G
          RHS1,RHS=(DEMAND,(ZO),PEAKDEM)
*
*PEAK DAY SUPPLY
*
     PSW(ZEW),L,IF((ZEW).NM. DUM )
          RHS1,RHS=(EXSTWELL,(ZEW),PDFLOW)
     PSS(ZS),E,IF((ZS).NM. DU)
          RHS1,RHS=(EXSTSPRG,(ZS),PDFLOW)
     PST(ZET),L,IF((ZET).NM. DUM)
          RHS1,RHS=(EXSTTRPL,(ZET),PDFLOW)
     PSFW(FWL),L,IF((FWL).NM. DUM)
     PSFS(FSP),E,IF((FSP).NM. DUM)
     PSFT(FTP),L,IF((FTP).NM. DUM)
*
*PEAK DAY PIPE CAPACITY
*
     PZ(ATOB1),L,IF((ATOB1).NM. DUMMY)
          RHS1,RHS=(PIPENET,(ATOB1),CAPAC)
               ,IF((PIPENET,(ATOB1),CAPTL).EQ.0)
     PZ(BTOA1/ATOB1),L,IF((ATOB1).NM. DUMMY .AND. (PIPENET,(ATOB1),
               BA1).NE.0)
          RHS1,RHS=(PIPENET,(ATOB1),CAPAC)
               ,IF((PIPENET,(ATOB1),CAPTL).EQ.0)
*
*END OF ROW DEFINITION
*
*
*START OF COLUMN AND MATRIX DEFINITION
*
MATRIX
*
*DECISION VARIABLES
*
*START INTEGER SET
*
     SWELLS,INTORG
*
*FUTURE WELLS
*
     IFW(FWL),IF((FWL).NM. DUM)
          OBJECT=(FWELL,(FWL),CAPTL)
          FFW(FWL)(S)=-(SEASONS,(S),DAYS)*(FWELL,(FWL),CAP)
          PSFW(FWL)=-(FWELL,(FWL),PCAP)
          BND,MAX=(FWELL,(FWL),NUM)
     EWELLS,INTEND
*
*END INTEGER SET FOR FUTURE WELLS
*
*START BIVALENT INTEGER SET
*
     SBIVAL,BIVORG
*
*FUTURE SPRINGS
*
     IFS(FSP),IF((FSP).NM. DUM)
          OBJECT=(FSPRG,(FSP),CAPTL)
          FFS(FSP)(S)=-(SEASONS,(S),DAYS)*(FSPRG,(FSP),FLOW(S))
          PSFS(FSP)=-(FSPRG,(FSP),PDFLOW)
*
*FUTURE TREATMENT PLANTS
*
     IFTP(FTP),IF((FTP).NM. DUM)
          OBJECT=(FUTRTRPL,(FTP),CAPTL)
          FFTP(FTP)(S)=-(SEASONS,(S),DAYS)*(FUTRTRPL,(FTP),CAP(S))
          PSFT(FTP)=-(FUTRTRPL,(FTP),PDFLOW)
*
*NETWORK CONDUITS - FUTURE PIPES
*
     IZT(ATOB1),IF((ATOB1).NM. DUMMY.AND.(PIPENET,(ATOB1),CAPTL)
               .NE.0)
          OBJECT=(PIPENET,(ATOB1),CAPTL)
          Z(ATOB1)(S)=-(SEASONS,(S),DAYS)*(PIPENET,(ATOB1),CAPAC)
          Z(BTOA1/ATOB1)(S)=-(SEASONS,(S),DAYS)*(PIPENET,(ATOB1),
               CAPAC),IF((PIPENET,(ATOB1),BA(S)).NE.0)
          PZ(ATOB1)=-(PIPENET,(ATOB1),CAPAC)
          PZ(BTOA1/ATOB1)=-(PIPENET,(ATOB1),CAPAC)
               ,IF((PIPENET,(ATOB1),BA1).NE.0)
     EBIVAL,BIVEND
*
*END INTEGLR VARIABLR SET
*
*START OF CONTINUOUS VARIABLE SET
*
*EXISTING WELLS
*
     XW(ZEW)(S),IF((ZEW).NM. DUM)
          OBJECT=(EXSTWELL,(ZEW),COST)
          D(ZEW)(S):11101:=1
          FW(ZEW)(S)=1
*
*EXISTING SPRINGS
*
     XS(ZS)(S),IF((ZS).NM. DU)
          OBJECT=(EXSTSPRG,(ZS),COST)
          D(ZS)(S)=1
          FS(ZS)(S)=1
*
```

```
*EXISTING TREATMENT PLANTS
*
     XTP(ZET)(S),IF((ZET).NM. DUM)
          OBJECT=(EXSTTRPL,(ZET),CST(S))
          D(ZET)(S):11101:=1
          FTP(ZET)(S)=1
*
*FUTURE WELLS
*
     XFW(FWL)(S),IF((FWL).NM. DUM)
          OBJECT=(FWELL,(FWL),OANDM)
          D(FWL)(S):11101:=1
          FFW(FWL)(S)=1
*
*FUTURE SPRINGS
*
     XFS(FSP)(S),IF((FSP).NM. DUM)
          OBJECT=(FSPRG,(FSP),COST)
          D(FSP)(S):11101:=1
          FFS(FSP)(S)=1
*
*FUTURE TRREATMENT PLANTS
*
     XFTP(FTP)(S),IF((FTP).NM. DUM)
          OBJECT=(FUTRTRPL,(FTP),CST(S))
          D(FTP)(S):11101:=1
          FFTP(FTP)(S)=1
*
*OPERATION OF PIPE NETWORK TRANSFERS
*
     XZ(ATOB1)(S),IF((ATOB1).NM. DUMMY)
          OBJECT=(PIPENET,(ATOB1),AB(S))
          D(ATOB1)(S):1110001:=-1
          D(ATOB1)(S):1001101:=1
          Z(ATOB1)(S)=1
     XZ(BTOA1/ATOB1)(S),IF((ATOB1).NM. DUMMY.AND.(PIPENET,(ATOB1),
               BA(S)).NE.0)
          OBJECT=(PIPENET,(ATOB1),BA(S))
          D(ATOB1)(S):1110001:=1
          D(ATOB1)(S):1001101:=-1
          Z(BTOA1)(S)=1
*
*PEAK EXISTING FLOWS
*
     PEW(ZEW),IF((ZEW).NM. DUM)
          OBJECT=(EXSTWELL,(ZEW),COST)*(DATA,PDZ,DATA)
          PD(ZEW):11110:=1
          PSW(ZEW)=1
     PES(ZS),IF((ZS).NM. DU)
          OBJECT=(EXSTSPRG,(ZS),COST)*(DATA,PDZ,DATA)
          PD(ZS)=1
          PSS(ZS)=1
     PET(ZET),IF((ZET).NM. DUM)
          OBJECT=(EXSTTRPL,(ZET),CST1)*(DATA,PDZ,DATA)
          PD(ZET):11110:=1
          PST(ZET)=1
*
*PEAK FUTURE FLOWS
*
     PFW(FWL),IF((FWL).NM. DUM)
          OBJECT=(FWELL,(FWL),OANDM)*(DATA,PDZ,DATA)
          PD(FWL):11110:=1
          PSFW(FWL)=1
     PFS(FSP),IF((FSP).NM. DUM)
          OBJECT=(FSPRG,(FSP),COST)*(DATA,PDZ,DATA)
          PD(FSP):11110:=1
          PSFS(FSP)=1
     PFT(FTP),IF((FTP).NM. DUM)
          OBJECT=(FUTRTRPL,(FTP),CST1)*(DATA,PDZ,DATA)
          PD(FTP):11110:=1
          PSFT(FTP)=1
*
*PEAK DAY TRANSFERS
*
     PX(ATOB1),IF((ATOB1).NM. DUMMY)
          OBJECT=(PIPENET,(ATOB1),AB1)*(DATA,PDZ,DATA)
          PD(ATOB1):1111000:=-1
          PD(ATOB1):1100110:=1
          PZ(ATOB1)=1
     PX(BTOA1/ATOB1),IF((ATOB1).NM. DUMMY.AND.(PIPENET,(ATOB1)
               ,BA1).NE.0)
          OBJECT=(PIPENET,(ATOB1),BA1)*(DATA,PDZ,DATA)
          PD(ATOB1):1111000:=1
          PD(ATOB1):1100110:=-1
          PZ(BTOA1)=1
*
*END OF MATRIX DEFINITION SECTION
*
*************************************************************************
*************************************************************************
*************************************************************************
*
*
*START OF REPORT SECTION
*
*
REPORT MODEL
     FORMAT,F(FI)="XXX",F(FR)="XXX,XXX,XXS.",F(FS)="XXXXXX.X"
          ,F(FX)="XXXX,XXS.XX",F(FP)="XXXXX.XX"
          ,F(FZ)="XXX,XXS.XXX",F(FY)="XXX.XXXX"
PAGE
     SKIP 5
     LINE,T15="C O N S T R U C T I O N   S C H E D U L E  **  N E W",
          T72="F A C I L I T I E S"
     SKIP 3
     LINE,T32="ALL CAPITAL COSTS ARE IN DOLLARS PER YEAR"
     SKIP 5
     W(CAPITAL)=0
     W(ADDCAP(S))=0
     W(ADDPK)=0
*
*INTIALIZE COUNTERS FOR WELLS AND OUTPUT ACTIVE WELL INFORMATION
*
     W(TOTPK)=0
     W(TOTNUM)=0
     W(TOTCOST)=0
     W(TOTCAP(S))=0
     LINE,T10="* * * * W E L L S * * * *"
     SKIP 4
DO (ZO)(A),IF((ZO)(A).IN.(FWL) .AND. CACT(IFW(ZO)(A)).GT.0.95)
     LINE,IF(W(TOTNUM).EQ.0)
          T26="WELLS"
```

```
            T36="CAPITAL"
            T48="PEAK DAY"
            T64+17(S)="CAPACITY-MGS"
      LINE,IF(W(TOTNUM).EQ.0)
            T8="ZONE"
            T25="DRILLED"
            T37="COST"
            T46="CAPACITY-MGD"
            T63+17(S)=T(S)
      SKIP 1,IF(W(TOTNUM).EQ.0)
  *
      W(NUM)=CACT(IFW(ZO)(A))//RND
      W(PEAK)=(FWELL,(ZO)(A),PCAP)*W(NUM)
      W(TOTNUM)=W(TOTNUM)+W(NUM)
      W(CAP(S))=-W(NUM)*AIJ(FFW(ZO)(A)(S),IFW(ZO)(A))
      W(COST)=W(NUM)*CCOST(IFW(ZO)(A))
  *
      LINE,T5=T(ZO),E(FI)25=W(NUM),T30=(A) ,E(FR)32=W(COST),
            E(FP)47=W(PEAK),E(FS)65+17(S)=W(CAP(S))
      W(TOTPK)=W(TOTPK)+W(PEAK)
  *
      W(TOTCOST)=W(TOTCOST)+W(COST)
      W(TOTCAP(S))=W(TOTCAP(S))+W(CAP(S))
  END DO
  *
  *
  DO,IF(W(TOTNUM).NE.0)
      LINE,T5="------------------------------------------------------------",
            T62="----------------------------------------------------",
            T110="------------------"
      SKIP 1
      LINE,T10="T O T A L S",E(FI)25=W(TOTNUM),E(FR)32=W(TOTCOST),
            E(FP)47=W(TOTPK),E(FS)65+17(S)=W(TOTCAP(S))
      W(CAPITAL)=W(CAPITAL)+W(TOTCOST)
      W(ADDCAP(S))=W(ADDCAP(S))+W(TOTCAP(S))
      W(ADDPK)=W(ADDPK)+W(TOTPK)
  END DO
  *
  *
  DO,IF(W(TOTNUM).EQ.0)
      SKIP 4
      LINE,T15="**** THE CONSTRUCTION OF NEW WELLS WAS NOT REQUIRED. ****"
  END DO
  *
  *
  *INITIALIZE COUNTERS FOR SPRINGS AND OUTPUT ACTIVE SPRING INFORMATION
  *
  *
      W(TOTPK)=0
      W(TOTNUM)=0
      W(TOTCOST)=0
      W(TOTCAP(S))=0
  PAGE
      SKIP 4
      LINE,T10="* * * * S P R I N G S * * * *"
      SKIP 4
  DO (ZO)(A),IF((ZO)(A)  .IM.(FSP) .AND. CACT(IFS(ZO)(A)).GT. 0.95)
      LINE,IF(W(TOTNUM).EQ.0)
          T25="SPRINGS"
            T36="CAPITAL"
            T48="PEAK DAY"
            T64+17(S)="CAPACITY-MGS"
      LINE,IF(W(TOTNUM).EQ.0)
            T8="ZONE"
            T24="DEVELOPED"
            T37="COST"
            T46="CAPACITY-MGD"
            T63+17(S)=T(S)
      SKIP 1,IF(W(TOTNUM).EQ.0)
  *
      W(PEAK)=(FSPRG,(ZO)(A),PDFLOW)
      W(CAP(S))= -AIJ(FFS(ZO)(A)(S),IFS(ZO)(A))
      W(COST)=CCOST(IFS(ZO)(A))
      W(TOTNUM)=W(TOTNUM)+1
  *
      LINE,T5=T(ZO),T28=(A),E(FR)32=W(COST),E(FP)47=W(PEAK),
            E(FS)65+17(S)=W(CAP(S))
  *
      W(TOTPK)=W(TOTPK)+W(PEAK)
      W(TOTCOST)=W(TOTCOST)+W(COST)
      W(TOTCAP(S))=W(TOTCAP(S))+W(CAP(S))
  END DO
  *
  *
  DO,IF(W(TOTNUM).NE.0)
      LINE,T5="------------------------------------------------------------",
            T62="----------------------------------------------------",
            T110="------------------"
      SKIP 1
      LINE,T10="T O T A L S",E(FI)26=W(TOTNUM),E(FR)32=W(TOTCOST),
            E(FP)47=W(TOTPK),E(FS)65+17(S)=W(TOTCAP(S))
      W(CAPITAL)=W(CAPITAL)+W(TOTCOST)
      W(ADDCAP(S))=W(ADDCAP(S))+W(TOTCAP(S))
      W(ADDPK)=W(ADDPK)+W(TOTPK)
  END DO
  *
  *
  DO,IF(W(TOTNUM).EQ.0)
      SKIP 2
      LINE,T15="**** THE CONSTRUCTION OF NEW SPRINGS WAS NOT REQUIRED.",
            T70="****"
  END DO
  *
  *
  *INITIALIZE COUNTERS FOR TREATMENT PLANTS AND OUTPUT ACTIVE TREATMENT
  *PLANT INFORMATION
  *
  *
      W(TOTPK)=0
      W(TOTNUM)=0
      W(TOTCOST)=0
      W(TOTCAP(S))=0
  PAGE
      SKIP 4
      LINE,T10="* * * * T R E A T M E N T   P L A N T S * * * *"
      SKIP 4
  DO (ZO)(A),IF((ZO)(A) .IM.(FTP) .AND. CACT(IFTP(ZO)(A)).GT. 0.95)
      LINE,IF(W(TOTNUM).EQ.0)
          T24="TRMT PLNT"
          T36="CAPITAL"
          T48="PEAK DAY"
          T64+17(S)="CAPACITY-MGS"
```

```
    LINE,IF(W(TOTNUM).EQ.0)
         T8="ZONE"
         T26="BUILT"
         T37="COST"
         T46="CAPACITY-MGD"
         T63+17(S)=T(S)
    SKIP 1,IF(W(TOTNUM).EQ.0)
*
    W(PEAK)=(FUTRTRPL,(ZO)(A),PDFLOW)
    W(CAP(S))=-AIJ(FFTP(ZO)(A)(S),IFTP(ZO)(A))
    W(COST)=CCOST(IFTP(ZO)(A))
    W(TOTNUM)=W(TOTNUM)+1
*
    LINE,T5=T(ZO),T28=(A),E(FR)32=W(COST),E(FP)47=W(PEAK),
              E(FS)65+17(S)=W(CAP(S))
*
    W(TOTPK)=W(TOTPK)+W(PEAK)
    W(TOTCOST)=W(TOTCOST)+W(COST)
    W(TOTCAP(S))=W(TOTCAP(S))+W(CAP(S))
END DO
*
*
DO,IF(W(TOTNUM).NE.0)
    LINE,T5="------------------------------------------------------",
              T62="--------------------------------------------------",
              T110="------------------"
    SKIP 1
    LINE,T10="T O T A L S",E(FI)26=W(TOTNUM),E(FR)32=W(TOTCOST),
              E(FP)47=W(TOTPK),E(FS)65+17(S)=W(TOTCAP(S))
    W(CAPITAL)=W(CAPITAL)+W(TOTCOST)
    W(ADDCAP(S))=W(ADDCAP(S))+W(TOTCAP(S))
    W(ADDPK)=W(ADDPK)+W(TOTPK)
END DO
*
*
DO,IF(W(TOTNUM).EQ.0)
    SKIP 2
    LINE,T15="**** THE CONSTRUCTION OF NEW TREATMENT PLANTS WAS NOT",
              T69="REQUIRED. ****"
END DO
*
*
*INTITIALIZE COUNTERS AND OUTPUT ACTIVE NETWORK SYNTHESIS INFORMATION
*
*
PAGE
    SKIP 4
    LINE, T10="* * * * N E T W O R K   S Y N T H E S I S * * * *"
    SKIP 4
    W(TOTCOST)=0
    W(TOTNUM)=0
DO (ATOB)(P),IF((ATOB)(P).IN.(ATOB1) .AND. CACT(IZT(ATOB)(P)).GT. 0.95)
    LINE,IF(W(TOTNUM).EQ.0)
         T8="ZONAL CONNECTIONS"
         T55="CONDUIT  SIZE"
         T75="CAPITAL COST"
         T91="CAPACITY-MGD"
    SKIP 1,IF(W(TOTNUM).EQ.0)
*
    W(COST)=CCOST(IZT(ATOB)(P))
    W(CAP)=(PIPENET,(ATOB)(P),CAPAC)
    W(TOTCOST)=W(TOTCOST)+W(COST)
*
    LINE,T5=T(ATOB),T48=(P),T53=T(P),E(FR)75=W(COST),E(FS)92=W(CAP)
*
    W(TOTNUM)=W(TOTNUM)+1
END DO
*
*
DO,IF(W(TOTNUM).NE.0)
    LINE,T5="------------------------------------------------------",
              T62="--------------------------------------------------"
    SKIP 1
    LINE,T10="TOTALS",T25="NUMBER =",E(FI)34=W(TOTNUM),
              E(FR)75=W(TOTCOST)
    W(CAPITAL)=W(CAPITAL)+W(TOTCOST)
END DO
*
*
DO,IF(W(TOTNUM).EQ.0)
    SKIP 2
    LINE,T15="**** NETWORK SYNTESIS NOT REQUIRED ****"
END DO
*
*
*SUMMARY OF NEW FACILITIES
*
*
PAGE
    SKIP 4
    LINE,T10="* * * * S U M M A R Y   O F   N E W   F A C I L I T I E",
              T66="S * * * *"
    SKIP 4
DO,IF(W(CAPITAL).EQ.0)
    LINE,T15="**** NO NEW FACILITIES REQUIRED ****"
END DO
DO,IF(W(CAPITAL).NE.0)
    LINE,T34="ADDED CAPACITY",T53+19(S)="ADDED CAPACITY"
    LINE,T39="MGD",T58+19(S)="MGS"
    LINE,T10="CONSTRUCTION COSTS",T37="PEAK DAY",T53+19(S)=T(S)
    SKIP 1
    LINE,E(FR)13=W(CAPITAL),E(FP)36=W(ADDPK),E(FS)55+19(S)=W(ADDCAP(S))
END DO
*
*
*ZONE BY ZONE ANALYSES FOR DEMAND, SUPPLY AND TRANSFERS
*
*
DO (ZO)
PAGE
    SKIP 4
    LINE,T10="* * * * A N A L Y S I S   F O R   Z O N E  ",T58=T(ZO)
    SKIP 4
    LINE,T8="SEASON",T35="DEMAND-MGS",T52="DEMAND-MGD",
              T69="PEAK DAY DEMAND"
    SKIP 1
    LINE (S)
         T8=T(S)
            E(FS)35=(DEMAND,(ZO),DEMAND(S))
            E(FP)52=(DEMAND,(ZO),DEMAND(S))/(SEASONS,(S),DAYS)
            E(FP)71=(DEMAND,(ZO),PEAKDEM),IF((S).EQ.1)
*
```

```
*
    SKIP 5
    LINE,T73="L E V E L   O F",T92="D  AND  M",T105="U N I T"
    LINE,T75="S U P P L Y",T92="T O T A L",T105="C O S T"
    LINE,T4="S E A S O N",T30="S U P P L Y        S O U R C E",
            T73="M G S       M G D",T92="C O S T S",T104="$/1000 GAL"
    SKIP 1
    LINE (P),IF((ZO)(P).IM.(ZEW) .AND. CACT(PEW(ZO)(P)).GT.0.0001)
         T3="PEAK  DAY"
         T21="EXISTING WELL"
         E(FY)81=CACT(PEW(ZO)(P))
         T117="ALTERNATE"
         T128=(P)
    LINE (S)(P),IF((ZO)(P).IM.(ZEW) .AND. CACT(XW(ZO)(P)(S)).GT. 0.01)
         T3=T(S)
         T21="EXISTING WELL"
         E(FS)71=CACT(XW(ZO)(P)(S))
         E(FP)81=CACT(XW(ZO)(P)(S))/(SEASONS,(S),DAYS)
         E(FX)91=CACT(XW(ZO)(P)(S))*(EXSTWELL,(ZO)(P),COST)
         E(FZ)101=(EXSTWELL,(ZO)(P),COST)/1000.
         T117="ALTERNATE"
         T128=(P)
*
    LINE, IF((ZO).IM.(ZS) .AND. CACT(PES(ZO)).GT. 0.0001)
         T3="PEAK  DAY"
         T21="EXISTING SPRINGS"
         E(FY)81=CACT(PES(ZO))
    LINE (S),IF((ZO).IM.(ZS) .AND. CACT(XS(ZO)(S)).GT. 0.01)
         T3=T(S)
         T21="EXISTING SPRINGS  (ALL COMBINED FOR ZONE)"
         E(FS)71=CACT(XS(ZO)(S))
         E(FP)81=CACT(XS(ZO)(S))/(SEASONS,(S),DAYS)
         E(FX)91=CACT(XS(ZO)(S))*(EXSTSPRG,(ZO),COST)
         E(FZ)101=(EXSTSPRG,(ZO),COST)/1000.
*
    LINE (P),IF((ZO)(P).IM.(ZET) .AND. CACT(PET(ZO)(P)).GT. 0.0001)
         T3="PEAK  DAY"
         T21="EXISTING TREATMENT PLANT"
         E(FY)81=CACT(PET(ZO)(P))
         T117="ALTERNATE"
         T128=(P)
    LINE (S)(P),IF((ZO)(P).IM.(ZET) .AND. CACT(XTP(ZO)(P)(S)).GT. 0.01)
         T3=T(S)
         T21="EXISTING TREATMENT PLANT"
         E(FS)71=CACT(XTP(ZO)(P)(S))
         E(FP)81=CACT(XTP(ZO)(P)(S))/(SEASONS,(S),DAYS)
         E(FX)91=CACT(XTP(ZO)(P)(S))*(EXSTTRPL,(ZO)(P),CST(S))
         E(FZ)101=(EXSTTRPL,(ZO)(P),CST(S))/1000.
         T117="ALTERNATE"
         T128=(P)
*
*
    LINE (A),IF((ZO)(A).IM.(FWL) .AND. CACT(PFW(ZO)(A)).GT. 0.0001)
         T3="PEAK  DAY"
         T21="NEW WELL"
         E(FY)81=CACT(PFW(ZO)(A))
         T117="ALTERNATE"
         T128=(A)
    LINE (S)(A),IF((ZO)(A).IM.(FWL) .AND. CACT(XFW(ZO)(A)(S)).GT.0.01)
         T3=T(S)
         T21="NEW WELL"
         E(FS)71=CACT(XFW(ZO)(A)(S))
         E(FP)81=CACT(XFW(ZO)(A)(S))/(SEASONS,(S),DAYS)
         E(FX)91=CACT(XFW(ZO)(A)(S))*(FWELL,(ZO)(A),OANDM)
         E(FZ)101=(FWELL,(ZO)(A),OANDM)/1000.
         T117="ALTERNATE"
         T128=(A)
*
    LINE (A),IF((ZO)(A).IM.(FSP) .AND. CACT(PFS(ZO)(A)).GT. 0.0001)
         T3="PEAK  DAY"
         T21="NEW SPRING"
         E(FY)81=CACT(PFS(ZO)(A))
         T117="ALTERNATE"
         T128=(A)
    LINE (S)(A),IF((ZO)(A).IM.(FSP) .AND. CACT(XFS(ZO)(A)(S)).GT.0.01)
         T3=T(S)
         T21="NEW SPRING"
         E(FS)71=CACT(XFS(ZO)(A)(S))
         E(FP)81=CACT(XFS(ZO)(A)(S))/(SEASONS,(S),DAYS)
         E(FX)91=CACT(XFS(ZO)(A)(S))*(FSPRG,(ZO)(A),COST)
         E(FZ)101=(FSPRG,(ZO)(A),COST)/1000.
         T117="ALTERNATE"
         T128=(A)
*
    LINE (A),IF((ZO)(A).IM.(FTP) .AND. CACT(PFT(ZO)(A)).GT. 0.0001)
         T3="PEAK  DAY"
         T21="NEW TREATMENT PLANT"
         E(FY)81=CACT(PFT(ZO)(A))
         T117="ALTERNATE"
         T128=(A)
    LINE (S)(A),IF((ZO)(A).IM.(FTP) .AND. CACT(XFTP(ZO)(A)(S)).GT.0.01)
         T3=T(S)
         T21="NEW TREATMENT PLANT"
         E(FS)71=CACT(XFTP(ZO)(A)(S))
         E(FP)81=CACT(XFTP(ZO)(A)(S))/(SEASONS,(S),DAYS)
         E(FX)91=CACT(XFTP(ZO)(A)(S))*(FUTRTRPL,(ZO)(A),CST(S))
         E(FZ)101=(FUTRTRPL,(ZO)(A),CST(S))/1000.
         T117="ALTERNATE"
         T128=(A)
*
*EXPORT
*
    LINE (ATOB)(R),IF((ATOB):1100:.EQ.(ZO) .AND. (ATOB)(R).IM.(ATOB1)
              .AND. CACT(PX(ATOB)(R)).GT. 0.0001)
         T3="PEAK  DAY"
         T21="EXPORT"
         T28=T(ATOB)
         E(FY)81=-CACT(PX(ATOB)(R))
         T117="ALTERNATE"
         T128=(R)
    LINE (ATOB)(R)(S),IF((ATOB):1100:.EQ.(ZO) .AND. (ATOB)(R).IM.(ATOB1)
              .AND. CACT(XZ(ATOB)(R)(S)).GT.0.01)
         T3=T(S)
         T21="EXPORT"
         T28=T(ATOB)
         E(FS)71=-CACT(XZ(ATOB)(R)(S))
         E(FP)81=-CACT(XZ(ATOB)(R)(S))/(SEASONS,(S),DAYS)
         E(FX)91= CACT(XZ(ATOB)(R)(S))*(PIPENET,(ATOB)(R),AB(S))
         E(FZ)101=(PIPENET,(ATOB)(R),AB(S))/1000.
         T117="ALTERNATE"
         T128=(R)
    LINE (ATOB)(R),IF((ATOB):0011:.EQ.(ZO) .AND. (ATOB)(R).IM.(ATOB1)
```

```
                   .AND. (PIPENET,(ATOB)(R),BA1).NE.0
                   .AND. CACT(PX(BTOA/ATOB)(R)).GT. 0.0001)
              T3="PEAK  DAY"
              T21="EXPORT"
              T28=T(ATOB)
              E(FY)81=-CACT(PX(BTOA)(R))
              T117="ALTERNATE"
              T128=(R)
         LINE (ATOB)(R)(S),IF((ATOB):0011:.EQ.(ZO) .AND. (ATOB)(R).IM.(ATOB1)
                   .AND. (PIPENET,(ATOB)(R),BA(S)).NE.0
                   .AND. CACT(XZ(BTOA/ATOB)(R)(S)).GT.0.01)
              T3=T(S)
              T21="EXPORT"
              T28=T(ATOB)
              E(FS)71=-CACT(XZ(BTOA)(R)(S))
              E(FP)81=-CACT(XZ(BTOA)(R)(S))/(SEASONS,(S),DAYS)
              E(FX)91= CACT(XZ(BTOA)(R)(S))*(PIPENET,(ATOB)(R),BA(S))
              E(FZ)101=(PIPENET,(ATOB)(R),BA(S))/1000.
              T117="ALTERNATE"
              T128=(R)
*
*IMPORT
*
         LINE (ATOB)(R),IF((ATOB):1100:.EQ.(ZO) .AND. (ATOB)(R).IM.(ATOB1)
                   .AND. (PIPENET,(ATOB)(R),BA1).NE.0
                   .AND. CACT(PX(BTOA/ATOB)(R)).GT. 0.0001)
              T3="PEAK  DAY"
              T21="IMPORT"
              T28=T(ATOB)
              E(FY)81=CACT(PX(BTOA)(R))
              T117="ALTERNATE"
              T128=(R)
         LINE (ATOB)(R)(S),IF((ATOB):1100:.EQ.(ZO) .AND. (ATOB)(R).IM.(ATOB1)
                   .AND. (PIPENET,(ATOB)(R),BA(S)).NE.0
                   .AND. CACT(XZ(BTOA/ATOB)(R)(S)).GT.0.01)
              T3=T(S)
              T21="IMPORT"
              T28=T(ATOB)
              E(FS)71=CACT(XZ(BTOA)(R)(S))
              E(FP)81=CACT(XZ(BTOA)(R)(S))/(SEASONS,(S),DAYS)
              E(FX)91=CACT(XZ(BTOA)(R)(S))*(PIPENET,(ATOB)(R),BA(S))
              E(FZ)101=(PIPENET,(ATOB)(R),BA(S))/1000.
              T117="ALTERNATE"
              T128=(R)
         LINE (ATCB)(R),IF((ATOB):0011:.EQ.(ZO) .AND. (ATOB)(R).IM.(ATOB1)
                   .AND. CACT(PX(ATOB)(R)).GT. 0.0001)
              T3="PEAK  DAY"
              T21="IMPORT"
              T28=T(ATOB)
              E(FY)81=CACT(PX(ATOB)(R))
              T117="ALTERNATE"
              T128=(R)
         LINE (ATOB)(R)(S),IF((ATOB):0011:.EQ.(ZO) .AND. (ATOB)(R).IM.(ATOB1)
                   .AND. CACT(XZ(ATOB)(R)(S)).GT.0.01)
              T3=T(S)
              T21="IMPORT"
              T28=T(ATOB)
              E(FS)71=CACT(XZ(ATOB)(R)(S))
              E(FP)81=CACT(XZ(ATOB)(R)(S))/(SEASONS,(S),DAYS)
              E(FX)91=CACT(XZ(ATOB)(R)(S))*(PIPENET,(ATOB)(R),AB(S))
              E(FZ)101=(PIPENET,(ATOB)(R),AB(S))/1000.
              T117="ALTERNATE"
              T128=(R)
END DO
*
*
********************************************************************
*END OF INITIAL REPORT OUTPUT
********************************************************************
*
*
*
*
********************************************************************
********************************************************************
********************************************************************
*
*
*START OF REPORT REVISE TO MODIFY INTEGER PROBLEM TO A L.P. PROBLEM
*
*
REPORT REVISE
     FORMAT,F(FX)="XXXXXX.XX"
*
*DELETE SEASON ROWS WITH ASSOCIATED INTEGER VARIABLES OF ZERO ACTIVITY
*
     LINE,T1="NAME",T15="REVISE"
     LINE,T1="ROWS"
     LINE,T3="DELETE"
*
     LINE (FWL)(S),IF((FWL).NM. DUM .AND. CACT(IFW(FWL)).LT.0.1)
          T5="FFW"
          T8=(FWL)
          T11=(S)
     LINE (FSP)(S),IF((FSP).NM. DUM .AND. CACT(IFS(FSP)).LT.0.1)
          T5="FFS"
          T8=(FSP)
          T11=(S)
     LINE (FTP)(S),IF((FTP).NM. DUM .AND. CACT(IFTP(FTP)).LT.0.1)
          T5="FFTP"
          T9=(FTP)
          T12=(S)
     LINE (ATOB1)(S),IF((ATOB1).NM. DUMMY .AND. (ATOB1):00001:.NM. X
                .AND. CACT(IZT(ATOB1)).LT. 0.1)
          T5="Z"
          T6=(ATOB1)
          T11=(S)
     LINE (BTOA1)(S),IF((BTOA1).NM.DUMMY.AND.(BTOA1):00001:.NM.X .AND.
                (PIPENET,(ATOB1/BTOA1),BA(S)).NE.0 .AND.
                CACT(IZT(ATOB1/BTOA1)).LT. 0.1)
          T5="Z"
          T6=(BTOA1)
          T11=(S)
*
*DELETE PEAK ROWS WITH ASSOCIATED INTEGER VARIABLES OF ZERO ACTIVITY
*
     LINE (FWL),IF((FWL).NM. DUM .AND. CACT(IFW(FWL)).LT.0.1)
          T5="PSFW"
          T9=(FWL)
     LINE (FSP),IF((FSP).NM. DUM .AND. CACT(IFS(FSP)).LT.0.1)
          T5="PSFS"
          T9=(FSP)
```

```
      LINE (FTP),IF((FTP).NM. DUM .AND. CACT(IFTP(FTP)).LT.0.1)
             T5="PSFT"
             T9=(FTP)
      LINE (ATOB1),IF((ATOB1).NM.DUMMY.AND.(ATOB1):00001:.NM.X .AND.
                    CACT(IZT(ATOB1)).LT. 0.1)
             T5="PZ"
             T7=(ATOB1)
      LINE (BTOA1),IF((BTOA1).NM.DUMMY.AND.(BTOA1):00001:.NM.X .AND.
                    (PIPENET,(ATOB1/BTOA1),BA1) .NE. 0  .AND.
                    CACT(IZT(ATOB1/BTOA1)).LT. 0.1)
             T5="PZ"
             T7=(BTOA1)
*
*DELETE COLUMNS
*
      LINE,T1="COLUMNS"
      LINE,T3="DELETE"
      LINE,T5="SWELLS"
      LINE,T5="EWELLS"
      LINE,T5="SBIVAL"
      LINE,T5="EBIVAL"
DO(FWL),IF((FWL).NM. DUM .AND. CACT(IFW(FWL)).LT.0.1)
      LINE,T5="IFS",T8=(FWL)
      LINE (S)
             T5="XFW"
             T8=(FWL)
             T11=(S)
      LINE,T5="PFW",T8=(FWL)
END DO
*
DO(FSP),IF((FSP).NM. DUM .AND. CACT(IFS(FSP)).LT. 0.1)
      LINE,T5="IFS",T8=(FSP)
      LINE (S)
             T5="XFS"
             T8=(FSP)
             T11=(S)
      LINE,T5="PFS",T8=(FSP)
END DO
*
DO(FTP),IF((FTP).NM. DUM .AND. CACT(IFTP(FTP)).LT. 0.1)
      LINE,T5="IFTP",T9=(FTP)
      LINE (S)
             T5="XFTP"
             T9=(FTP)
             T12=(S)
      LINE,T5="PFT",T8=(FTP)
END DO
*
DO(ATOB1),IF((ATOB1).NM.DUMMY .AND. (ATOB1):00001:.NM.X .AND.
                    CACT(IZT(ATOB1)).LT. 0.1)
      LINE,T5="IZT",T8=(ATOB1)
      LINE (S)
             T5="XZ"
             T7=(ATOB1)
             T12=(S)
      LINE,T5="PX",T7=(ATOB1)
END DO
*
DO(BTOA1),IF((BTOA1).NM.DUMMY .AND. (BTOA1):00001:.NM.X .AND.
                    (PIPENET,(ATOB1/BTOA1),BA1).NE. 0  .AND.
                    CACT(IZT(ATOB1/BTOA1)) .LT. 0.1 )
      LINE (S)
             T5="XZ"
             T7=(BTOA1)
             T12=(S)
      LINE,T5="PX",T7=(BTOA1)
END DO
*
*FIX ACTIVE INTEGER VARIABLES AT THEIR PRESENT ACTIVITIES
*
      LINE,T1="BOUNDS"
      LINE,T3="MODIFY"
*
      LINE (FWL),IF((FWL).NM. DUM .AND. CACT(IFW(FWL)).GT. 0.9)
             T2="FX"
             T5="BND"
             T15="IFW"
             T18=(FWL)
             E(FX)25=CACT(IFW(FWL))//RND
*
      LINE (FSP),IF((FSP).NM. DUM .AND. CACT(IFS(FSP)) .GT. 0.9)
             T2="FX"
             T5="BND"
             T15="IFS"
             T18=(FSP)
             T30="1.00"
*
      LINE (FTP),IF((FTP).NM.DUM .AND. CACT(IFTP(FTP)).GT. 0.9)
             T2="FX"
             T5="BND"
             T15="IFTP"
             T19=(FTP)
             T30="1.00"
*
      LINE (ATOB1),IF((ATOB1).NM.DUMMY .AND. (ATOB1):00001:.NM. X .AND.
                    CACT(IZT(ATOB1)).GT. 0.9)
             T2="FX"
             T5="BND"
             T15="IZT"
             T18=(ATOB1)
             T30="1.00"
*
      LINE,T1="ENDATA"
*
************************************************************************
*END OF REVISE REPORT
************************************************************************
*
ENDATA
```

Appendix E

Published Report For Sample Problem

C O N S T R U C T I O N   S C H E D U L E ** N E W   F A C I L I T I E S

ALL CAPITAL COSTS ARE IN DOLLARS PER YEAR

* * * * W E L L S * * * *

ZONE	WELLS DRILLED		CAPITAL COST	PEAK DAY CAPACITY-MGD	CAPACITY-MGS JUN 1 - AUG 9	CAPACITY-MGS AUG 10 - NOV 17	CAPACITY-MGS NOV 18 - FEB 16	CAPACITY-MGS FEB 17 - JUN 30
MYTOWN	2	A	$7,180.	2.02	201.6	288.0	259.2	302.4
PARKSVILLE	1	A	$4,310.	1.51	151.2	216.0	194.4	226.8
PARKSVILLE	1	B	$5,440.	2.52	252.0	360.0	324.0	378.0
YOURTOWN	2	A	$8,620.	3.02	302.4	432.0	388.8	453.6
FARMTOWN	2	A	$8,620.	3.02	302.4	432.0	388.8	453.6
T O T A L S	8		$34,170.	12.09	1209.6	1728.0	1555.2	1814.4

* * * * S P R I N G S * * * *

ZONE	SPRINGS DEVELOPED	CAPITAL COST	PEAK DAY CAPACITY-MGD	CAPACITY-MGS JUN 1 - AUG 9	CAPACITY-MGS AUG 10 - NOV 17	CAPACITY-MGS NOV 18 - FEB 16	CAPACITY-MGS FEB 17 - JUN 30
MYTOWN	A	$178,500.	5.82	678.3	776.0	406.8	611.1
T O T A L S	1	$178,500.	5.82	678.3	776.0	406.8	611.1

* * * * T R E A T M E N T   P L A N T S * * * *

ZONE	TRMT PLNT BUILT	CAPITAL COST	PEAK DAY CAPACITY-MGD	CAPACITY-MGS JUN 1 - AUG 9	CAPACITY-MGS AUG 10 - NOV 17	CAPACITY-MGS NOV 18 - FEB 16	CAPACITY-MGS FEB 17 - JUN 30
PARKSVILLE	A	$354,360.	16.00	1400.0	2000.0	1800.0	2100.0
YOURTOWN	A	$82,130.	4.50	350.0	500.0	450.0	525.0
T O T A L S	2	$436,490.	20.50	1750.0	2500.0	2250.0	2625.0

* * * * N E T W O R K   S Y N T H E S I S * * * *

ZONAL CONNECTIONS		CONDUIT SIZE	CAPITAL COST	CAPACITY-MGD
PARKSVILLE TO YOURTOWN	E	14 INCH DIA. PIPE	$34,980.	4.0
HALF WAY STOP TO FARMTOWN	H	20 INCH DIA. PIPE	$30,240.	8.4
TOTALS	NUMBER = 2		$65,220.	

* * * * S U M M A R Y   O F   N E W   F A C I L I T I E S * * * *

CONSTRUCTION COSTS	ADDED CAPACITY MGD PEAK DAY	ADDED CAPACITY MGS JUN 1 - AUG 9	ADDED CAPACITY MGS AUG 10 - NOV 17	ADDED CAPACITY MGS NOV 18 - FEB 16	ADDED CAPACITY MGS FEB 17 - JUN 30
$714,380.	38.41	3637.9	5004.0	4212.0	5050.5

**** ANALYSIS FOR ZONE MYTOWN

SEASON	DEMAND-MGS	DEMAND-MGD	PEAK DAY DEMAND
JUN 1 - AUG 9	525.0	7.50	9.00
AUG 10 - NOV 17	625.0	6.25	
NOV 18 - FEB 16	450.0	5.00	
FEB 17 - JUN 30	656.0	6.25	

SEASON	SUPPLY SOURCE	LEVEL OF SUPPLY MGS	LEVEL OF SUPPLY MGD	O AND M TOTAL COSTS	UNIT COST $/1000 GAL	
PEAK DAY	EXISTING WELL		1.3000			ALTERNATE A
NOV 18 - FEB 16	EXISTING WELL	43.2	0.48	$1,695.60	$0.039	ALTERNATE A
FEB 17 - JUN 30	EXISTING WELL	44.9	0.43	$1,762.32	$0.039	ALTERNATE A
PEAK DAY	NEW WELL		1.8800			ALTERNATE A
PEAK DAY	NEW SPRING		5.8200			ALTERNATE A
JUN 1 - AUG 9	NEW SPRING	678.3	9.69	$4,476.78	$0.007	ALTERNATE A
AUG 10 - NOV 17	NEW SPRING	776.0	7.76	$5,121.60	$0.007	ALTERNATE A
NOV 18 - FEB 16	NEW SPRING	406.8	4.52	$2,684.88	$0.007	ALTERNATE A
FEB 17 - JUN 30	NEW SPRING	611.1	5.82	$4,033.26	$0.007	ALTERNATE A

**** ANALYSIS FOR ZONE PARKSVILLE

SEASON	DEMAND-MGS	DEMAND-MGD	PEAK DAY DEMAND
JUN 1 - AUG 9	1750.0	25.00	30.00
AUG 10 - NOV 17	2000.0	20.00	
NOV 18 - FEB 16	1350.0	15.00	
FEB 17 - JUN 30	1995.0	19.00	

SEASON	SUPPLY SOURCE	LEVEL OF SUPPLY MGS	LEVEL OF SUPPLY MGD	O AND M TOTAL COSTS	UNIT COST $/1000 GAL	
PEAK DAY	EXISTING WELL		0.6100			ALTERNATE A
PEAK DAY	EXISTING WELL		2.5900			ALTERNATE B
JUN 1 - AUG 9	EXISTING WELL	50.4	0.72	$1,222.20	$0.024	ALTERNATE A
AUG 10 - NOV 17	EXISTING WELL	72.0	0.72	$1,746.00	$0.024	ALTERNATE A
NOV 18 - FEB 16	EXISTING WELL	64.8	0.72	$1,571.40	$0.024	ALTERNATE A
FEB 17 - JUN 30	EXISTING WELL	75.6	0.72	$1,833.30	$0.024	ALTERNATE A
JUN 1 - AUG 9	EXISTING WELL	201.6	2.88	$7,912.80	$0.039	ALTERNATE B
AUG 10 - NOV 17	EXISTING WELL	288.0	2.88	$11,304.00	$0.039	ALTERNATE B
NOV 18 - FEB 16	EXISTING WELL	259.2	2.88	$10,173.60	$0.039	ALTERNATE B
FEB 17 - JUN 30	EXISTING WELL	302.4	2.88	$11,869.20	$0.039	ALTERNATE B
PEAK DAY	EXISTING TREATMENT PLANT		1.4000			ALTERNATE A
PEAK DAY	NEW WELL		1.5100			ALTERNATE A
PEAK DAY	NEW WELL		2.5200			ALTERNATE B
JUN 1 - AUG 9	NEW WELL	151.2	2.16	$4,800.60	$0.032	ALTERNATE A
AUG 10 - NOV 17	NEW WELL	216.0	2.16	$6,858.00	$0.032	ALTERNATE A
NOV 18 - FEB 16	NEW WELL	194.4	2.16	$6,172.20	$0.032	ALTERNATE A
FEB 17 - JUN 30	NEW WELL	226.8	2.16	$7,200.90	$0.032	ALTERNATE A
JUN 1 - AUG 9	NEW WELL	252.0	3.60	$9,891.00	$0.039	ALTERNATE B
AUG 10 - NOV 17	NEW WELL	360.0	3.60	$14,130.00	$0.039	ALTERNATE B
NOV 18 - FEB 16	NEW WELL	324.0	3.60	$12,717.00	$0.039	ALTERNATE B
FEB 17 - JUN 30	NEW WELL	378.0	3.60	$14,836.50	$0.039	ALTERNATE B
PEAK DAY	NEW TREATMENT PLANT		15.9900			ALTERNATE A
JUN 1 - AUG 9	NEW TREATMENT PLANT	969.6	13.85	$63,024.00	$0.065	ALTERNATE A
AUG 10 - NOV 17	NEW TREATMENT PLANT	972.0	9.72	$70,956.00	$0.073	ALTERNATE A
NOV 18 - FEB 16	NEW TREATMENT PLANT	507.6	5.64	$41,115.60	$0.081	ALTERNATE A
FEB 17 - JUN 30	NEW TREATMENT PLANT	847.1	8.07	$59,297.00	$0.070	ALTERNATE A
PEAK DAY	IMPORT PARKSVILLE TO YOURTOWN		4.0300			ALTERNATE E
PEAK DAY	IMPORT PARKSVILLE TO YOURTOWN		1.3500			ALTERNATE X
JUN 1 - AUG 9	IMPORT PARKSVILLE TO YOURTOWN	125.2	1.79	$5,493.78	$0.044	ALTERNATE E
AUG 10 - NOV 17	IMPORT PARKSVILLE TO YOURTOWN	92.0	0.92	$4,036.96	$0.044	ALTERNATE E
FEB 17 - JUN 30	IMPORT PARKSVILLE TO YOURTOWN	165.1	1.57	$7,244.59	$0.044	ALTERNATE E

* * * * ANALYSIS FOR ZONE YOURTOWN

SEASON	DEMAND-MGS	DEMAND-MGD	PEAK DAY DEMAND
JUN 1 - AUG 9	327.0	4.67	5.61
AUG 10 - NOV 17	425.0	4.25	
NOV 18 - FEB 16	306.0	3.40	
FEB 17 - JUN 30	446.0	4.25	

SEASON	SUPPLY SOURCE	LEVEL OF SUPPLY MGS	LEVEL OF SUPPLY MGD	O AND M TOTAL COSTS	UNIT COST $/1000 GAL	
PEAK DAY	EXISTING SPRINGS		4.5200			
JUN 1 - AUG 9	EXISTING SPRINGS (ALL COMBINED FOR ZONE)	452.2	6.46	$2,984.52	$0.007	
AUG 10 - NOV 17	EXISTING SPRINGS (ALL COMBINED FOR ZONE)	517.0	5.17	$3,412.20	$0.007	
NOV 18 - FEB 16	EXISTING SPRINGS (ALL COMBINED FOR ZONE)	290.7	3.23	$1,918.62	$0.007	
FEB 17 - JUN 30	EXISTING SPRINGS (ALL COMBINED FOR ZONE)	611.1	5.82	$4,033.26	$0.007	
PEAK DAY	NEW WELL		3.0200			ALTERNATE A
NOV 18 - FEB 16	NEW WELL	15.3	0.17	$600.52	$0.039	ALTERNATE A
PEAK DAY	NEW TREATMENT PLANT		3.4500			ALTERNATE A
PEAK DAY	EXPORT PARKSVILLE TO YOURTOWN		-4.0300			ALTERNATE E
PEAK DAY	EXPORT PARKSVILLE TO YOURTOWN		-1.3500			ALTERNATE X
JUN 1 - AUG 9	EXPORT PARKSVILLE TO YOURTOWN	-125.2	-1.79	$5,493.78	$0.044	ALTERNATE E
AUG 10 - NOV 17	EXPORT PARKSVILLE TO YOURTOWN	-92.0	-0.92	$4,036.96	$0.044	ALTERNATE E
FEB 17 - JUN 30	EXPORT PARKSVILLE TO YOURTOWN	-165.1	-1.57	$7,244.59	$0.044	ALTERNATE E

* * * * ANALYSIS FOR ZONE FARMTOWN

SEASON	DEMAND-MGS	DEMAND-MGD	PEAK DAY DEMAND
JUN 1 - AUG 9	536.0	7.66	9.18
AUG 10 - NOV 17	595.0	5.95	
NOV 18 - FEB 16	306.0	3.40	
FEB 17 - JUN 30	714.0	6.80	

SEASON	SUPPLY SOURCE	LEVEL OF SUPPLY MGS	LEVEL OF SUPPLY MGD	O AND M TOTAL COSTS	UNIT COST $/1000 GAL	
PEAK DAY	EXISTING SPRINGS		6.7900			
JUN 1 - AUG 9	EXISTING SPRINGS (ALL COMBINED FOR ZONE)	678.3	9.69	$4,476.78	$0.007	
AUG 10 - NOV 17	EXISTING SPRINGS (ALL COMBINED FOR ZONE)	776.0	7.76	$5,121.60	$0.007	
NOV 18 - FEB 16	EXISTING SPRINGS (ALL COMBINED FOR ZONE)	465.3	5.17	$3,070.98	$0.007	
FEB 17 - JUN 30	EXISTING SPRINGS (ALL COMBINED FOR ZONE)	678.3	6.46	$4,476.78	$0.007	
PEAK DAY	NEW WELL		2.5800			ALTERNATE A
FEB 17 - JUN 30	NEW WELL	43.0	0.41	$1,526.50	$0.036	ALTERNATE A
PEAK DAY	EXPORT HALF WAY STOP TO FARMTOWN		-0.1900			ALTERNATE H
JUN 1 - AUG 9	EXPORT HALF WAY STOP TO FARMTOWN	-5.2	-0.07	$128.39	$0.025	ALTERNATE H
AUG 10 - NOV 17	EXPORT HALF WAY STOP TO FARMTOWN	-6.0	-0.06	$148.14	$0.025	ALTERNATE H
NOV 18 - FEB 16	EXPORT HALF WAY STOP TO FARMTOWN	-5.4	-0.06	$133.33	$0.025	ALTERNATE H
FEB 17 - JUN 30	EXPORT HALF WAY STOP TO FARMTOWN	-7.3	-0.07	$180.24	$0.025	ALTERNATE H

* * * * ANALYSIS FOR ZONE HALF WAY STOP

SEASON	DEMAND-MGS	DEMAND-MGD	PEAK DAY DEMAND
JUN 1 - AUG 9	15.0	0.21	0.26
AUG 10 - NOV 17	20.0	0.20	
NOV 18 - FEB 16	18.0	0.20	
FEB 17 - JUN 30	22.0	0.21	

SEASON	SUPPLY SOURCE	LEVEL OF SUPPLY MGS	LEVEL OF SUPPLY MGD	O AND M TOTAL COSTS	UNIT COST $/1000 GAL	
PEAK DAY	EXISTING WELL		0.0700			ALTERNATE A
JUN 1 - AUG 9	EXISTING WELL	9.8	0.14	$90.65	$0.009	ALTERNATE A
AUG 10 - NOV 17	EXISTING WELL	14.0	0.14	$129.50	$0.009	ALTERNATE A
NOV 18 - FEB 16	EXISTING WELL	12.6	0.14	$116.55	$0.009	ALTERNATE A
FEB 17 - JUN 30	EXISTING WELL	14.7	0.14	$135.97	$0.009	ALTERNATE A
PEAK DAY	IMPORT HALF WAY STOP TO FARMTOWN		0.1900			ALTERNATE H
JUN 1 - AUG 9	IMPORT HALF WAY STOP TO FARMTOWN	5.2	0.07	$128.39	$0.025	ALTERNATE H
AUG 10 - NOV 17	IMPORT HALF WAY STOP TO FARMTOWN	6.0	0.06	$148.14	$0.025	ALTERNATE H
NOV 18 - FEB 16	IMPORT HALF WAY STOP TO FARMTOWN	5.4	0.06	$133.33	$0.025	ALTERNATE H
FEB 17 - JUN 30	IMPORT HALF WAY STOP TO FARMTOWN	7.3	0.07	$180.24	$0.025	ALTERNATE H

Appendix F

TEMPO Procedures–MACROLIB1

```
100 MACRO CREATOR RETAIN
110 $FILE INPUT=(480030)MODELGENERATOR1
120 ZCONSOLE=.FALSE.
130 ZPRINTER=.TRUE.
140 ZLGFRQ=9999
150 GENERATE
160 LOCK(RWF)
170 ZNAME="MODEL"
180 ZDATA="MODEL"
190 INPUT(DISK)
200 BCDOUT
210 EXIT
220 ENDMACRO
230 MACRO SOLVELP RETAIN
240 ZLGFRQ=9999
250 ZCONSOLE=.FALSE.
260 ZPRINTER=.TRUE.
270 ZNAME="MODEL"
280 ZBNDST="BND"
290 SETUP(MIN,LOWER,SUMMARY)
300 ZDONFS=LAB
310 TITLE "LP SOLUTION FOR MODEL"
320 ZRHS="RHS1"
330 ZOBJ="OBJECT"
340 ZSOLNM="LPOPTUM"
350 ZBASNM="LPBASIS"
360 PRIMAL
370 LAB:SAVE
380 OUTPUT(FILE)
390 EXIT
400 ENDMACRO
410 MACRO SOLVEIP RETAIN
420 URPERCNT=1.0-URPERCNT
430 ZLGFRQ=9999
440 ZCONSOLE=.FALSE.
450 ZPRINTER=.TRUE.
460 ZTOLIN=0.001
470 ZNOSOL=LAB6
480 ZINTSOL=LAB1
490 ZSOLNM="IPSOL"
500 TITLE "INTEGER SOLUTIONS FOR MODEL"
510 LAB3:MXINT(NOPRINT)
520 GO TO LAB2
530 LAB1:UI=UI+1
540 ZSOLNO=UI
550 OUTPUT(FILE)
560 SAVEFILE
570 ZBIOBJ=URPERCNT*ZCUROB
580 UR1=ZCUROB
590 DISPLAY "NEW CUTOFF"
600 DISPLAY ZBIOBJ
610 RETURN
620 LAB6:IF(UI .GT. 0)GO TO LAB2
630 DISPLAY "NO INTEGER SOLUTIONS POSSIBLE FOR YOUR MODEL"
640 EXIT
650 LAB2:IF(UI1 .GT. 0)GO TO LAB4
660 ZINTSOL=LAB5
670 ZBIOBJ=UR1
680 DISPLAY "CUTOFF FOR RESTART"
690 DISPLAY ZBIOBJ
700 MXINT(RESTART,NOPRINT)
710 LAB4:ZREPNM="MODEL"
720 PUBLISH
730 REMOVE(BRANCH)
740 EXIT
750 LAB5:UI=UI+1
760 UI1=7
770 ZSOLNO=UI
780 OUTPUT(FILE)
790 GO TO LAB3
800 ENDMACRO
810 MACRO RANGES RETAIN
820 ZCONSOLE=.FALSE.
830 TITLE "REVISION TO LP SOLUTION FOR RANGE"
840 ZREPNM="REVISE"
850 $FILE CARDOUT(KIND=DISK,TITLE=MODREV)
860 ZPRINTER=.FALSE.
870 PUBLISH(PUNCH)
880 ZPRINTER=.TRUE.
890 ZONAME=ZNAME
900 ZNAME="REVISE"
910 ZDATA="REVISE"
920 $FILE CARDIN(KIND=DISK,TITLE=MODREV)
930 REVISE(CARD)
940 SETUP(MIN,SUMMARY)
950 PRIMAL
960 TITLE "RANGE FOR REVISED IP  TO  LP SOLUTION"
970 RANGE
980 EXIT
990 ENDMACRO
```

Appendix G

Model Revision Example

The numbered statements are input by the user. Numbers prefixing the statements are used here for clarification only. If inputing this section, do <u>not</u> use numbers.

```
    READY    (output from the interactive TEMPO system
               requesting input commands)
1   ZDATA = "REVDATA"  (name of revision data to be created)
    READY
2   ZONAME = "MODEL"  (name of old model)
    READY
3   ZNAME = "REVMODEL"  (name of new model)
    READY
4   REVISE(REMOTE)     (revisions to be made from the terminal)
    ---REVISE---
    %  (indicates revise input requested)
5   ROWS
    %
6   AFTER
    %
7   L/MIX1          (adding the row MIX for season 1 as a ≤ row)
    %
8   L/MIX2
    %
9   L/MIX3
    %
10  L/MIX4
    %
11  L/MIXP
    %
12  COLUMNS
    %
13  MODIFY
    %
14  XFW32A1/MIX1/1.0      (places a 1 in the matrix position
                           row MIX1 column XFW32A1)
    %
15  XS321/MIX1/-1.0
    %
16  XFW32A2/MIX2/1.0
    %
17  XS322/MIX2/-1.0
    %
18  XFW32A3/MIX3/1.0
    %
19  XS323/MIX3/-1.0
    %
20  XFW32A4/MIX4/1.0
    %
21  XS324/MIX4/-1.0
    %
22  PFW32A/MIXP/1.0
    %
23  PES32/MIXP/-1.0
    %
24  LIST  (output of all above changes to check for errors)
    NAME           REVDATA
    ROWS
      AFTER
     L  MIX1
     L  MIX2
     L  MIX3
     L  MIX4
     L  MIXP
    COLUMNS
      MODIFY
        XFW32A1           MIX1          1.0
        XS321             MIX1         -1.0
        XFW32A2           MIX2          1.0
        XS322             MIX2         -1.0
        XFW32A3           MIX3          1.0
        XS323             MIX3         -1.0
        XFW32A4           MIX4          1.0
        XS324             MIX4         -1.0
        PFW32A            MIXP          1.0
        PES32             MIXP         -1.0
      ENDATA
    %
25  ENDATA  (indicates the above list of revisions is acceptable)

         "system information on problem statistics"

    READY
```

Appendix H

TEMPO Output For Optimal MIP Solution

MODEL
INTEGER SOLUTIONS FOR MODEL

ROWS SECTION

NUMBER	NAME	STATUS	ACTIVITY	SLACK ACTIVITY	LOWER LIMIT	UPPER LIMIT	DUAL ACTIVITY
1	OBJECT	BS	1142048.45438	-1142048.45438	NONE	NONE	1.00000
2	D011	BS	678.30000	-153.30000	525.00000	NONE	.
3	D012	BS	776.00000	-151.00000	625.00000	NONE	.
4	D013	LL	450.00000	.	450.00000	NONE	-39.25000
5	D014	LL	656.00000	.	656.00000	NONE	-39.25000
6	D141	LL	1750.00000	.	1750.00000	NONE	-65.00000
7	D142	LL	2000.00000	.	2000.00000	NONE	-73.00000
8	D143	LL	1350.00000	.	1350.00000	NONE	-81.00000
9	D144	LL	1995.00000	.	1995.00000	NONE	-70.00000
10	D321	LL	327.00000	.	327.00000	NONE	-21.12000
11	D322	LL	425.00000	.	425.00000	NONE	-29.12000
12	D323	LL	306.00000	.	306.00000	NONE	-39.25000
13	D324	LL	446.00000	.	446.00000	NONE	-26.12000
14	D221	BS	673.10000	-137.10000	536.00000	NONE	.
15	D222	BS	770.00000	-175.00000	595.00000	NONE	.
16	D223	BS	459.90000	-153.90000	306.00000	NONE	.
17	D224	LL	714.00000	.	714.00000	NONE	-35.50000
18	D561	LL	15.00000	.	15.00000	NONE	-24.69000
19	D562	LL	20.00000	.	20.00000	NONE	-24.69000
20	D563	LL	18.00000	.	18.00000	NONE	-24.69000
21	D564	LL	22.00000	.	22.00000	NONE	-60.19000
22	FW01A1	BS	.	100.80000	NONE	100.80000	.
23	FW01A2	BS	.	144.00000	NONE	144.00000	.
24	FW01A3	BS	43.20000	86.40000	NONE	129.60000	.
25	FW01A4	BS	44.90000	106.30000	NONE	151.20000	.
26	FW14A1	UL	50.40000	.	NONE	50.40000	40.75000
27	FW14A2	UL	72.00000	.	NONE	72.00000	48.75000
28	FW14A3	UL	64.80000	.	NONE	64.80000	56.75000
29	FW14A4	UL	75.60000	.	NONE	75.60000	45.75000
30	FW14B1	UL	201.60000	.	NONE	201.60000	25.75000
31	FW14B2	UL	288.00000	.	NONE	288.00000	33.75000
32	FW14B3	UL	259.20000	.	NONE	259.20000	41.75000
33	FW14B4	UL	302.40000	.	NONE	302.40000	30.75000
34	FW56A1	UL	9.80000	.	NONE	9.80000	15.44000
35	FW56A2	UL	14.00000	.	NONE	14.00000	15.44000
36	FW56A3	UL	12.60000	.	NONE	12.60000	15.44000
37	FW56A4	UL	14.70000	.	NONE	14.70000	50.94000
38	FS321	EQ	452.20000	.	452.20000	452.20000	14.52000
39	FS322	EQ	517.00000	.	517.00000	517.00000	22.52000
40	FS323	EQ	290.70000	.	290.70000	290.70000	32.65000
41	FS324	EQ	611.10000	.	611.10000	611.10000	19.52000
42	FS221	EQ	678.30000	.	678.30000	678.30000	-6.60000
43	FS222	EQ	776.00000	.	776.00000	776.00000	-6.60000
44	FS223	EQ	465.30000	.	465.30000	465.30000	-6.60000
45	FS224	EQ	678.30000	.	678.30000	678.30000	28.90000
46	FTP14A1	BS	.	140.00000	NONE	140.00000	.
47	FTP14A2	BS	.	200.00000	NONE	200.00000	.
48	FTP14A3	BS	.	180.00000	NONE	180.00000	.
49	FTP14A4	BS	.	210.00000	NONE	210.00000	.
50	FFW01A1	BS	-201.60000	201.60000	NONE	.	.
51	FFW01A2	BS	-288.00000	288.00000	NONE	.	.
52	FFW01A3	BS	-259.20000	259.20000	NONE	.	.
53	FFW01A4	BS	-302.40000	302.40000	NONE	.	.
54	FFW14A1	UL	.	.	NONE	.	33.25000
55	FFW14A2	UL	.	.	NONE	.	41.25000
56	FFW14A3	UL	.	.	NONE	.	49.25000
57	FFW14A4	UL	.	.	NONE	.	38.25000
58	FFW14B1	UL	.	.	NONE	.	25.75000
59	FFW14B2	UL	.	.	NONE	.	33.75000
60	FFW14B3	UL	.	.	NONE	.	41.75000
61	FFW14B4	UL	.	.	NONE	.	30.75000
62	FFW32A1	BS	-302.40000	302.40000	NONE	.	.
63	FFW32A2	BS	-432.00000	432.00000	NONE	.	.
64	FFW32A3	BS	-373.50000	373.50000	NONE	.	.

MODEL
INTEGER SOLUTIONS FOR MODEL

ROWS SECTION

NUMBER	NAME	STATUS	ACTIVITY	SLACK ACTIVITY	LOWER LIMIT	UPPER LIMIT	DUAL ACTIVITY
65	FFW32A4	BS	-453.60000	453.60000	NONE	.	.
66	FFW22A1	BS	-302.40000	302.40000	NONE	.	.
67	FFW22A2	BS	-432.00000	432.00000	NONE	.	.
68	FFW22A3	BS	-388.80000	388.80000	NONE	.	.
69	FFW22A4	BS	-410.60000	410.60000	NONE	.	.
70	FFS01A1	EQ	.	.	.	.	-6.60000
71	FFS01A2	EQ	.	.	.	.	-6.60000
72	FFS01A3	EQ	.	.	.	.	32.65000
73	FFS01A4	EQ	.	.	.	.	32.65000
74	FFS14A1	EQ	.	.	.	.	58.40000
75	FFS14A2	EQ	.	.	.	.	66.40000
76	FFS14A3	EQ	.	.	.	.	74.40000
77	FFS14A4	EQ	.	.	.	.	63.40000
78	FFS32A1	EQ	.	.	.	.	14.52000
79	FFS32A2	EQ	.	.	.	.	22.52000
80	FFS32A3	EQ	.	.	.	.	32.65000
81	FFS32A4	EQ	.	.	.	.	19.52000
82	FFTP14A1	BS	-429.52500	429.52500	NONE	.	.
83	FFTP14A2	BS	-1026.75000	1026.75000	NONE	.	.
84	FFTP14A3	BS	-1291.27500	1291.27500	NONE	.	.
85	FFTP14A4	BS	-1251.58750	1251.58750	NONE	.	.
86	FFTP32A1	BS	-350.00000	350.00000	NONE	.	.
87	FFTP32A2	BS	-500.00000	500.00000	NONE	.	.
88	FFTP32A3	BS	-450.00000	450.00000	NONE	.	.
89	FFTP32A4	BS	-525.00000	525.00000	NONE	.	.
90	Z0114J1	UL	.	.	NONE	.	57.35000
91	Z0114J2	UL	.	.	NONE	.	65.35000
92	Z0114J3	UL	.	.	NONE	.	34.10000
93	Z0114J4	UL	.	.	NONE	.	23.10000
94	Z0114K1	UL	.	.	NONE	.	58.54000
95	Z0114K2	UL	.	.	NONE	.	66.54000
96	Z0114K3	UL	.	.	NONE	.	35.29000
97	Z0114K4	UL	.	.	NONE	.	24.29000
98	Z0122X1	BS	.	50.40000	NONE	50.40000	
99	Z0122X2	BS	.	72.00000	NONE	72.00000	.
100	Z0122X3	BS	.	64.80000	NONE	64.80000	.
101	Z0122X4	BS	.	75.60000	NONE	75.60000	.
102	Z0122H1	BS	.	.	NONE	.	.
103	Z0122H2	BS	.	.	NONE	.	.
104	Z0122H3	BS	.	.	NONE	.	.
105	Z0122H4	UL	.	.	NONE	.	45.71429
106	Z1456A1	BS	.	.	NONE	.	.
107	Z1456A2	BS	.	.	NONE	.	.
108	Z1456A3	BS	.	.	NONE	.	.
109	Z1456A4	UL	.	.	NONE	.	105.02646
110	Z1456B1	BS	.	.	NONE	.	.
111	Z1456B2	BS	.	.	NONE	.	.
112	Z1456B3	BS	.	.	NONE	.	.
113	Z1456B4	UL	.	.	NONE	.	74.77954
114	Z1456C1	BS	.	.	NONE	.	.
115	Z1456C2	BS	.	.	NONE	.	.
116	Z1456C3	BS	.	.	NONE	.	.
117	Z1456C4	UL	.	.	NONE	.	61.06702
118	Z1432X1	BS	.	94.50000	NONE	94.50000	.
119	Z1432X2	BS	.	135.00000	NONE	135.00000	.
120	Z1432X3	BS	.	121.50000	NONE	121.50000	.
121	Z1432X4	BS	.	141.75000	NONE	141.75000	.
122	Z1432E1	BS	-282.10000	282.10000	NONE	.	.
123	Z1432E2	BS	-403.00000	403.00000	NONE	.	.
124	Z1432E3	BS	-362.70000	362.70000	NONE	.	.
125	Z1432E4	BS	-423.15000	423.15000	NONE	.	.
126	Z3222H1	BS	.	.	NONE	.	.
127	Z3222H2	BS	.	.	NONE	.	.
128	Z3222H3	BS	.	.	NONE	.	.
129	Z3222H4	UL	.	.	NONE	.	82.79441
130	Z3222I1	BS	.	.	NONE	.	.
131	Z3222I2	BS	.	.	NONE	.	.
132	Z3222I3	BS	.	.	NONE	.	.
133	Z3222I4	UL	.	.	NONE	.	79.13927
134	Z5622H1	BS	-584.50000	584.50000	NONE	.	.

MODEL
INTEGER SOLUTIONS FOR MODEL

ROWS SECTION

NUMBER	NAME	STATUS	ACTIVITY	SLACK ACTIVITY	LOWER LIMIT	UPPER LIMIT	DUAL ACTIVITY
135	Z5622H2	BS	-835.00000	835.00000	NONE	.	.
136	Z5622H3	BS	-751.50000	751.50000	NONE	.	.
137	Z5622H4	BS	-876.75000	876.75000	NONE	.	.
138	Z5622I1	BS	.	.	NONE	.	.
139	Z5622I2	BS	.	.	NONE	.	.
140	Z5622I3	BS	.	.	NONE	.	.
141	Z5622I4	BS	.	.	NONE	.	.
142	Z5622J1	BS	.	.	NONE	.	.
143	Z5622J2	BS	.	.	NONE	.	.
144	Z5622J3	BS	.	.	NONE	.	.
145	Z5622J4	BS	.	.	NONE	.	.
146	Z1401J1	BS	.	.	NONE	.	.
147	Z1401J2	BS	.	.	NONE	.	.
148	Z1401J3	BS	.	.	NONE	.	.
149	Z1401J4	BS	.	.	NONE	.	.
150	Z1401K1	BS	.	.	NONE	.	.
151	Z1401K2	BS	.	.	NONE	.	.
152	Z1401K3	BS	.	.	NONE	.	.
153	Z1401K4	BS	.	.	NONE	.	.
154	Z3214X1	BS	.	94.50000	NONE	94.50000	.
155	Z3214X2	BS	.	135.00000	NONE	135.00000	.
156	Z3214X3	BS	.	121.50000	NONE	121.50000	.
157	Z3214X4	BS	.	141.75000	NONE	141.75000	.
158	Z3214E1	BS	-156.90000	156.90000	NONE	.	.
159	Z3214E2	BS	-311.00000	311.00000	NONE	.	.
160	Z3214E3	BS	-362.70000	362.70000	NONE	.	.
161	Z3214E4	BS	-258.05000	258.05000	NONE	.	.
162	Z2256H1	BS	-579.30000	579.30000	NONE	.	.
163	Z2256H2	BS	-829.00000	829.00000	NONE	.	.
164	Z2256H3	BS	-746.10000	746.10000	NONE	.	.
165	Z2256H4	BS	-869.45000	869.45000	NONE	.	.
166	Z2256I1	UL	.	.	NONE	.	1.49000
167	Z2256I2	UL	.	.	NONE	.	1.49000
168	Z2256I3	UL	.	.	NONE	.	1.49000
169	Z2256I4	UL	.	.	NONE	.	1.49000
170	Z2256J1	UL	.	.	NONE	.	3.03000
171	Z2256J2	UL	.	.	NONE	.	3.03000
172	Z2256J3	UL	.	.	NONE	.	3.03000
173	Z2256J4	UL	.	.	NONE	.	3.03000
174	PD01	LL	9.00000	.	9.00000	NONE	-0.54250
175	PD14	LL	30.00000	.	30.00000	NONE	-22148.15000
176	PD32	LL	5.61000	.	5.61000	NONE	-0.78000
177	PD22	LL	9.18000	.	9.18000	NONE	-0.35500
178	PD56	LL	0.26000	.	0.26000	NONE	-0.60190
179	PSW01A	UL	1.30000	.	NONE	1.30000	0.15000
180	PSW14A	UL	0.61000	.	NONE	0.61000	22147.90750
181	PSW14B	UL	2.59000	.	NONE	2.59000	22147.75750
182	PSW56A	UL	0.07000	.	NONE	0.07000	0.50940
183	PSS32	EQ	4.52000	.	4.52000	4.52000	0.71400
184	PSS22	EQ	6.79000	.	6.79000	6.79000	0.28900
185	PST14A	UL	1.40000	.	NONE	1.40000	22147.16400
186	PSFW01A	BS	-0.14000	0.14000	NONE	.	.
187	PSFW14A	UL	.	.	NONE	.	22147.83250
188	PSFW14B	UL	.	.	NONE	.	22147.75750
189	PSFW32A	UL	.	.	NONE	.	0.38750
190	PSFW22A	BS	-0.44000	0.44000	NONE	.	
191	PSFS01A	EQ	.	.	.	.	0.47650
192	PSFS14A	EQ	.	.	.	.	39172.98968
193	PSFS32A	EQ	.	.	.	.	29594.36935
194	PSFT14A	UL	.	.	NONE	.	22147.50000
195	PSFT32A	BS	-1.05000	1.05000	NONE	.	
196	PZ0114J	UL	.	.	NONE	.	22147.53100
197	PZ0114K	UL	.	.	NONE	.	22147.54290
198	PZ0122X	BS	.	0.72000	NONE	0.72000	.
199	PZ0122H	BS	.	.	NONE	.	.
200	PZ1456A	BS	.	.	NONE	.	.
201	PZ1456B	BS	.	.	NONE	.	.
202	PZ1456C	BS	.	.	NONE	.	.
203	PZ1432X	BS	.	1.35000	NONE	1.35000	.
204	PZ1432E	BS	-4.03000	4.03000	NONE	.	.
205	PZ3222H	BS	.	.	NONE	.	.
206	PZ3222I	BS	.	.	NONE	.	.
207	PZ5622H	BS	-8.35000	8.35000	NONE	.	.
208	PZ5622I	BS	.	.	NONE	.	.
209	PZ5622J	BS	.	.	NONE	.	.
210	PZ1401J	BS	.	.	NONE	.	.

MODEL
INTEGER SOLUTIONS FOR MODEL

ROWS SECTION

NUMBER	NAME	STATUS	ACTIVITY	SLACK ACTIVITY	LOWER LIMIT	UPPER LIMIT	DUAL ACTIVITY
211	PZ1401K	BS	.	.	NONE	.	.
212	PZ3214X	UL	1.35000	.	NONE	1.35000	22146.89020
213	PZ3214E	UL	.	.	NONE	.	22146.93120
214	PZ2256H	BS	-8.16000	8.16000	NONE	.	.
215	PZ2256I	UL	.	.	NONE	.	0.01490
216	PZ2256J	UL	.	.	NONE	.	0.03030

MODEL
INTEGER SOLUTIONS FOR MODEL

COLUMNS SECTION

NUMBER	NAME	STATUS	ACTIVITY	INPUT COST	LOWER LIMIT	UPPER LIMIT	REDUCED COST
241	IFW01A	IV	2.00000	3590.00000	.	3.00000	3590.00000
242	IFW14A	IV	1.00000	4310.00000	.	1.00000	-61319.92708
243	IFW14B	IV	1.00000	5440.00000	.	1.00000	-94161.84890
244	IFW32A	IV	2.00000	4310.00000	.	2.00000	4309.41487
245	IFW22A	IV	2.00000	4310.00000	.	2.00000	4310.00000
246	IFS01A	IV	1.00000	178500.00000	.	1.00000	154861.17177
247	IFS14A	IV	.	673800.00000	.	1.00000	.
248	IFS32A	IV	.	122610.00000	.	1.00000	.
249	IFTP14A	IV	0.99938	354360.00000	.	1.00000	.
250	IFTP32A	IV	1.00000	82130.00000	.	1.00000	82130.00000
251	IZT0114J	IV	.	71160.00000	.	1.00000	-643785.46032
252	IZT0114K	IV	.	100990.00000	.	1.00000	-939205.29579
253	IZT0122H	IV	.	40080.00000	.	1.00000	.
254	IZT1456A	IV	.	7940.00000	.	1.00000	.
255	IZT1456B	IV	.	10600.00000	.	1.00000	.
256	IZT1456C	IV	.	13850.00000	.	1.00000	.
257	IZT1432E	IV	1.00000	34980.00000	.	1.00000	-54272.13274
258	IZT3222H	IV	.	72590.00000	.	1.00000	.
259	IZT3222I	IV	.	99300.00000	.	1.00000	.
260	IZT5622H	IV	1.00000	30240.00000	.	1.00000	30240.00000
261	IZT5622I	IV	.	41370.00000	.	1.00000	34870.81445
262	IZT5622J	IV	.	63500.00000	.	1.00000	42796.04878
263	XW01A1	LL	.	39.25000	.	NONE	39.25000
264	XW01A2	LL	.	39.25000	.	NONE	39.25000
265	XW01A3	BS	43.20000	39.25000	.	NONE	.
266	XW01A4	BS	44.90000	39.25000	.	NONE	.
267	XW14A1	BS	50.40000	24.25000	.	NONE	.
268	XW14A2	BS	72.00000	24.25000	.	NONE	.
269	XW14A3	BS	64.80000	24.25000	.	NONE	.
270	XW14A4	BS	75.60000	24.25000	.	NONE	.
271	XW14B1	BS	201.60000	39.25000	.	NONE	.
272	XW14B2	BS	288.00000	39.25000	.	NONE	.
273	XW14B3	BS	259.20000	39.25000	.	NONE	.
274	XW14B4	BS	302.40000	39.25000	.	NONE	.
275	XW56A1	BS	9.80000	9.25000	.	NONE	.
276	XW56A2	BS	14.00000	9.25000	.	NONE	.
277	XW56A3	BS	12.60000	9.25000	.	NONE	.
278	XW56A4	BS	14.70000	9.25000	.	NONE	.
279	XS321	BS	452.20000	6.60000	.	NONE	.
280	XS322	BS	517.00000	6.60000	.	NONE	.
281	XS323	BS	290.70000	6.60000	.	NONE	.
282	XS324	BS	611.10000	6.60000	.	NONE	.
283	XS221	BS	678.30000	6.60000	.	NONE	.
284	XS222	BS	776.00000	6.60000	.	NONE	.
285	XS223	BS	465.30000	6.60000	.	NONE	.
286	XS224	BS	678.30000	6.60000	.	NONE	.
287	XTP14A1	LL	.	98.60000	.	NONE	33.60000
288	XTP14A2	LL	.	112.30000	.	NONE	39.30000
289	XTP14A3	LL	.	122.50000	.	NONE	41.50000
290	XTP14A4	LL	.	109.40000	.	NONE	39.40000
291	XFW01A1	LL	.	54.25000	.	NONE	54.25000
292	XFW01A2	LL	.	54.25000	.	NONE	54.25000
293	XFW01A3	LL	.	54.25000	.	NONE	15.00000
294	XFW01A4	LL	.	54.25000	.	NONE	15.00000
295	XFW14A1	BS	151.20000	31.75000	.	NONE	.
296	XFW14A2	BS	216.00000	31.75000	.	NONE	.
297	XFW14A3	BS	194.40000	31.75000	.	NONE	.
298	XFW14A4	BS	226.80000	31.75000	.	NONE	.
299	XFW14B1	BS	252.00000	39.25000	.	NONE	.
300	XFW14B2	BS	360.00000	39.25000	.	NONE	.

MODEL
INTEGER SOLUTIONS FOR MODEL

COLUMNS SECTION

NUMBER	NAME	STATUS	ACTIVITY	INPUT COST	LOWER LIMIT	UPPER LIMIT	REDUCED COST
301	XFW14B3	BS	324.00000	39.25000	.	NONE	.
302	XFW14B4	BS	378.00000	39.25000	.	NONE	.
303	XFW32A1	LL	.	39.25000	.	NONE	18.13000
304	XFW32A2	LL	.	39.25000	.	NONE	10.13000
305	XFW32A3	BS	15.30000	39.25000	.	NONE	.
306	XFW32A4	LL	.	39.25000	.	NONE	13.13000
307	XFW22A1	LL	.	35.50000	.	NONE	35.50000
308	XFW22A2	LL	.	35.50000	.	NONE	35.50000
309	XFW22A3	LL	.	35.50000	.	NONE	35.50000
310	XFW22A4	BS	43.00000	35.50000	.	NONE	.
311	XFS01A1	BS	678.30000	6.60000	.	NONE	.
312	XFS01A2	BS	776.00000	6.60000	.	NONE	.
313	XFS01A3	BS	406.80000	6.60000	.	NONE	.
314	XFS01A4	BS	611.10000	6.60000	.	NONE	.
315	XFS14A1	BS	.	6.60000	.	NONE	.
316	XFS14A2	BS	.	6.60000	.	NONE	.
317	XFS14A3	BS	.	6.60000	.	NONE	.
318	XFS14A4	BS	.	6.60000	.	NONE	.
319	XFS32A1	BS	.	6.60000	.	NONE	.
320	XFS32A2	BS	.	6.60000	.	NONE	.
321	XFS32A3	BS	.	6.60000	.	NONE	.
322	XFS32A4	BS	.	6.60000	.	NONE	.
323	XFTP14A1	BS	969.60000	65.00000	.	NONE	.
324	XFTP14A2	BS	972.00000	73.00000	.	NONE	.
325	XFTP14A3	BS	507.60000	81.00000	.	NONE	.
326	XFTP14A4	BS	847.10000	70.00000	.	NONE	.
327	XFTP32A1	LL	.	78.00000	.	NONE	56.88000
328	XFTP32A2	LL	.	94.00000	.	NONE	64.88000
329	XFTP32A3	LL	.	123.00000	.	NONE	83.75000
330	XFTP32A4	LL	.	82.00000	.	NONE	55.88000
331	XZ0114J1	BS	.	7.65000	.	NONE	.
332	XZ0114J2	BS	.	7.65000	.	NONE	.
333	XZ0114J3	BS	.	7.65000	.	NONE	.
334	XZ0114J4	BS	.	7.65000	.	NONE	.
335	XZ0114K1	BS	.	6.46000	.	NONE	.
336	XZ0114K2	BS	.	6.46000	.	NONE	.
337	XZ0114K3	BS	.	6.46000	.	NONE	.
338	XZ0114K4	BS	.	6.46000	.	NONE	.
339	XZ0122X1	LL	.	115.12000	.	NONE	115.12000
340	XZ0122X2	LL	.	115.12000	.	NONE	115.12000
341	XZ0122X3	LL	.	115.12000	.	NONE	154.37000
342	XZ0122X4	LL	.	115.12000	.	NONE	118.87000
343	XZ0122H1	LL	.	12.84000	.	NONE	12.84000
344	XZ0122H2	LL	.	12.84000	.	NONE	12.84000
345	XZ0122H3	LL	.	12.84000	.	NONE	52.09000
346	XZ0122H4	LL	.	12.84000	.	NONE	62.30429
347	XZ1456A1	LL	.	30.28000	.	NONE	70.59000
348	XZ1456A2	LL	.	30.28000	.	NONE	78.59000
349	XZ1456A3	LL	.	30.28000	.	NONE	86.59000
350	XZ1456A4	LL	.	30.28000	.	NONE	145.11646
351	XZ1456B1	LL	.	22.58000	.	NONE	62.89000
352	XZ1456B2	LL	.	22.58000	.	NONE	70.89000
353	XZ1456B3	LL	.	22.58000	.	NONE	78.89000
354	XZ1456B4	LL	.	22.58000	.	NONE	107.16954
355	XZ1456C1	LL	.	18.17000	.	NONE	58.48000
356	XZ1456C2	LL	.	18.17000	.	NONE	66.48000
357	XZ1456C3	LL	.	18.17000	.	NONE	74.48000
358	XZ1456C4	LL	.	18.17000	.	NONE	89.04702
359	XZ1432X1	LL	.	62.98000	.	NONE	106.86000
360	XZ1432X2	LL	.	62.98000	.	NONE	106.86000
361	XZ1432X3	LL	.	62.98000	.	NONE	104.73000
362	XZ1432X4	LL	.	62.98000	.	NONE	106.86000
363	XZ1432E1	LL	.	28.88000	.	NONE	72.76000
364	XZ1432E2	LL	.	28.88000	.	NONE	72.76000
365	XZ1432E3	LL	.	28.88000	.	NONE	70.63000
366	XZ1432E4	LL	.	28.88000	.	NONE	72.76000
367	XZ3222H1	LL	.	23.25000	.	NONE	44.37000
368	XZ3222H2	LL	.	23.25000	.	NONE	52.37000
369	XZ3222H3	LL	.	23.25000	.	NONE	62.50000
370	XZ3222H4	LL	.	23.25000	.	NONE	96.66441
371	XZ3222I1	LL	.	19.68000	.	NONE	40.80000
372	XZ3222I2	LL	.	19.68000	.	NONE	48.80000
373	XZ3222I3	LL	.	19.68000	.	NONE	58.93000
374	XZ3222I4	LL	.	19.68000	.	NONE	89.43927
375	XZ5622H1	LL	.	9.69000	.	NONE	34.38000
376	XZ5622H2	LL	.	9.69000	.	NONE	34.38000
377	XZ5622H3	LL	.	9.69000	.	NONE	34.38000

MODEL
INTEGER SOLUTIONS FOR MODEL

COLUMNS SECTION

NUMBER	NAME	STATUS	ACTIVITY	INPUT COST	LOWER LIMIT	UPPER LIMIT	REDUCED COST
378	XZ5622H4	LL	.	9.69000	.	NONE	34.38000
379	XZ5622I1	LL	.	8.20000	.	NONE	32.89000
380	XZ5622I2	LL	.	8.20000	.	NONE	32.89000
381	XZ5622I3	LL	.	8.20000	.	NONE	32.89000
382	XZ5622I4	LL	.	8.20000	.	NONE	32.89000
383	XZ5622J1	LL	.	6.66000	.	NONE	31.35000
384	XZ5622J2	LL	.	6.66000	.	NONE	31.35000
385	XZ5622J3	LL	.	6.66000	.	NONE	31.35000
386	XZ5622J4	LL	.	6.66000	.	NONE	31.35000
387	XZ1401J1	LL	.	52.65000	.	NONE	117.65000
388	XZ1401J2	LL	.	52.65000	.	NONE	125.65000
389	XZ1401J3	LL	.	52.65000	.	NONE	94.40000
390	XZ1401J4	LL	.	52.65000	.	NONE	83.40000
391	XZ1401K1	LL	.	51.46000	.	NONE	116.46000
392	XZ1401K2	LL	.	51.46000	.	NONE	124.46000
393	XZ1401K3	LL	.	51.46000	.	NONE	93.21000
394	XZ1401K4	LL	.	51.46000	.	NONE	82.21000
395	XZ3214X1	LL	.	47.98000	.	NONE	4.10000
396	XZ3214X2	LL	.	47.98000	.	NONE	4.10000
397	XZ3214X3	LL	.	47.98000	.	NONE	6.23000
398	XZ3214X4	LL	.	47.98000	.	NONE	4.10000
399	XZ3214E1	BS	125.20000	43.88000	.	NONE	.
400	XZ3214E2	BS	92.00000	43.88000	.	NONE	.
401	XZ3214E3	LL	.	43.88000	.	NONE	2.13000
402	XZ3214E4	BS	165.10000	43.88000	.	NONE	.
403	XZ2256H1	BS	5.20000	24.69000	.	NONE	.
404	XZ2256H2	BS	6.00000	24.69000	.	NONE	.
405	XZ2256H3	BS	5.40000	24.69000	.	NONE	.
406	XZ2256H4	BS	7.30000	24.69000	.	NONE	.
407	XZ2256I1	BS	.	23.20000	.	NONE	.
408	XZ2256I2	BS	.	23.20000	.	NONE	.
409	XZ2256I3	BS	.	23.20000	.	NONE	.
410	XZ2256I4	BS	.	23.20000	.	NONE	.
411	XZ2256J1	BS	.	21.66000	.	NONE	.
412	XZ2256J2	BS	.	21.66000	.	NONE	.
413	XZ2256J3	BS	.	21.66000	.	NONE	.
414	XZ2256J4	BS	.	21.66000	.	NONE	.
415	PEW01A	BS	1.30000	0.39250	.	NONE	.
416	PEW14A	BS	0.61000	0.24250	.	NONE	.
417	PEW14B	BS	2.59000	0.39250	.	NONE	.
418	PEW56A	BS	0.07000	0.09250	.	NONE	.
419	PES32	BS	4.52000	0.06600	.	NONE	.
420	PES22	BS	6.79000	0.06600	.	NONE	.
421	PET14A	BS	1.40000	0.98600	.	NONE	.
422	PFW01A	BS	1.88000	0.54250	.	NONE	.
423	PFW14A	BS	1.51000	0.31750	.	NONE	.
424	PFW14B	BS	2.52000	0.39250	.	NONE	.
425	PFW32A	BS	3.02000	0.39250	.	NONE	.
426	PFW22A	BS	2.58000	0.35500	.	NONE	.
427	PFS01A	BS	5.82000	0.06600	.	NONE	.
428	PFS14A	LL	.	0.06600	.	NONE	17024.90568
429	PFS32A	LL	.	0.06600	.	NONE	29593.65535
430	PFT14A	BS	15.99000	0.65000	.	NONE	.
431	PFT32A	BS	3.45000	0.78000	.	NONE	.
432	PX0114J	BS	.	0.07650	.	NONE	.
433	PX0114K	BS	.	0.06460	.	NONE	.
434	PX0122X	LL	.	1.15120	.	NONE	1.33870
435	PX0122H	LL	.	0.12840	.	NONE	0.31590
436	PX1456A	LL	.	0.30280	.	NONE	22147.85090
437	PX1456B	LL	.	0.22580	.	NONE	22147.77390
438	PX1456C	LL	.	0.18170	.	NONE	22147.72980
439	PX1432X	LL	.	0.62980	.	NONE	22147.99980
440	PX1432E	LL	.	0.28880	.	NONE	22147.65880
441	PX3222H	LL	.	0.23250	.	NONE	0.65750
442	PX3222I	LL	.	0.19680	.	NONE	0.62180
443	PX5622H	LL	.	0.09690	.	NONE	0.34380
444	PX5622I	LL	.	0.08200	.	NONE	0.32890
445	PX5622J	LL	.	0.06660	.	NONE	0.31350
446	PX1401J	LL	.	0.52650	.	NONE	22148.13400
447	PX1401K	LL	.	0.51460	.	NONE	22148.12210
448	PX3214X	BS	1.35000	0.47980	.	NONE	.
449	PX3214E	BS	4.03000	0.43880	.	NONE	.
450	PX2256H	BS	0.19000	0.24690	.	NONE	.
451	PX2256I	BS	.	0.23200	.	NONE	.
452	PX2256J	BS	.	0.21660	.	NONE	.

Appendix I

RANGE For Optimal MIP Solution

MODEL
INTEGER SOLUTIONS FOR MODEL

ROWS AT LIMIT LEVEL

NUMBER	ROW	ST	ACTIVITY	SLACK ACTIVITY	LOWER LIMIT UPPER LIMIT	LOWER ACTIVITY UPPER ACTIVITY	UNIT COST UNIT COST	UPPER COST LOWER COST	LIMITING PROCESS	ST ST
4	D013	LL	450.00000	.	450.00000	406.80000	-39.25000		XW01A3	LL
					NONE	536.40000	39.25000		FW01A3	UL
5	D014	LL	656.00000	.	656.00000	611.10000	-39.25000		XW01A4	LL
					NONE	762.30000	39.25000		FW01A4	UL
6	D141	LL	1750.00000	.	1750.00000	780.40000	-65.00000		XFTP14A1	LL
					NONE	2179.52500	65.00000		FFTP14A1	UL
7	D142	LL	2000.00000	.	2000.00000	1028.00000	-73.00000		XFTP14A2	LL
					NONE	3026.75000	73.00000		FFTP14A2	UL
8	D143	LL	1350.00000	.	1350.00000	842.40000	-81.00000		XFTP14A3	LL
					NONE	2641.27500	81.00000		FFTP14A3	UL
9	D144	LL	1995.00000	.	1995.00000	1147.90000	-70.00000		XFTP14A4	LL
					NONE	3246.58750	70.00000		FFTP14A4	UL
10	D321	LL	327.00000	.	327.00000	170.10000	-21.12000		Z3214E1	UL
					NONE	452.20000	21.12000		XZ3214E1	LL
11	D322	LL	425.00000	.	425.00000	114.00000	-29.12000		Z3214E2	UL
					NONE	517.00000	29.12000		XZ3214E2	LL
12	D323	LL	306.00000	.	306.00000	290.70000	-39.25000		XFW32A3	LL
					NONE	679.50000	39.25000		FFW32A3	UL
13	D324	LL	446.00000	.	446.00000	187.95000	-26.12000		Z3214E4	UL
					NONE	611.10000	26.12000		XZ3214E4	LL
17	D224	LL	714.00000	.	714.00000	671.00000	-35.50000		XFW22A4	LL
					NONE	1124.60000	35.50000		FFW22A4	UL
18	D561	LL	15.00000	.	15.00000	9.80000	-24.69000		XZ2256H1	LL
					NONE	152.10000	24.69000		D221	LL
19	D562	LL	20.00000	.	20.00000	14.00000	-24.69000		XZ2256H2	LL
					NONE	195.00000	24.69000		D222	LL
20	D563	LL	18.00000	.	18.00000	12.60000	-24.69000		XZ2256H3	LL
					NONE	171.90000	24.69000		D223	LL

MODEL
INTEGER SOLUTIONS FOR MODEL

ROWS AT LIMIT LEVEL

NUMBER	ROW	ST	ACTIVITY	SLACK ACTIVITY	LOWER LIMIT UPPER LIMIT	LOWER ACTIVITY UPPER ACTIVITY	UNIT COST UNIT COST	UPPER COST LOWER COST	LIMITING PROCESS	ST ST
21	D564	LL	22.00000	.	22.00000 NONE	14.70000 432.60000	-60.19000 60.19000		XZ2256H4 FFW22A4	LL UL
26	FW14A1	UL	50.40000	.	NONE 50.40000	. 1020.00000	40.75000 -40.75000		XW14A1 XFTP14A1	LL LL
27	FW14A2	UL	72.00000	.	NONE 72.00000	. 1044.00000	48.75000 -48.75000		XW14A2 XFTP14A2	LL LL
28	FW14A3	UL	64.80000	.	NONE 64.80000	. 572.40000	56.75000 -56.75000		XW14A3 XFTP14A3	LL LL
29	FW14A4	UL	75.60000	.	NONE 75.60000	. 922.70000	45.75000 -45.75000		XW14A4 XFTP14A4	LL LL
30	FW14B1	UL	201.60000	.	NONE 201.60000	. 1171.20000	25.75000 -25.75000		XW14B1 XFTP14A1	LL LL
31	FW14B2	UL	288.00000	.	NONE 288.00000	. 1260.00000	33.75000 -33.75000		XW14B2 XFTP14A2	LL LL
32	FW14B3	UL	259.20000	.	NONE 259.20000	. 766.80000	41.75000 -41.75000		XW14B3 XFTP14A3	LL LL
33	FW14B4	UL	302.40000	.	NONE 302.40000	. 1149.50000	30.75000 -30.75000		XW14B4 XFTP14A4	LL LL
34	FW56A1	UL	9.80000	.	NONE 9.80000	. 15.00000	15.44000 -15.44000		XW56A1 XZ2256H1	LL LL
35	FW56A2	UL	14.00000	.	NONE 14.00000	. 20.00000	15.44000 -15.44000		XW56A2 XZ2256H2	LL LL
36	FW56A3	UL	12.60000	.	NONE 12.60000	. 18.00000	15.44000 -15.44000		XW56A3 XZ2256H3	LL LL
37	FW56A4	UL	14.70000	.	NONE 14.70000	. 22.00000	50.94000 -50.94000		XW56A4 XZ2256H4	LL LL
38	FS321	EQ	452.20000	.	452.20000 452.20000	327.00000 609.10000	14.52000 -14.52000		XZ3214E1 Z3214E1	LL UL

MODEL
INTEGER SOLUTIONS FOR MODEL

ROWS AT LIMIT LEVEL

NUMBER	ROW	ST	ACTIVITY	SLACK ACTIVITY	LOWER LIMIT UPPER LIMIT	LOWER ACTIVITY UPPER ACTIVITY	UNIT COST UNIT COST	UPPER COST LOWER COST	LIMITING PROCESS	ST ST
167	Z2256I2	UL	.	.	NONE .	. 6.00000	1.49000 -1.49000		XZ2256I2 XZ2256H2	LL LL
168	Z2256I3	UL	.	.	NONE .	. 5.40000	1.49000 -1.49000		XZ2256I3 XZ2256H3	LL LL
169	Z2256I4	UL	.	.	NONE .	. 7.30000	1.49000 -1.49000		XZ2256I4 XZ2256H4	LL LL
170	Z2256J1	UL	.	.	NONE .	. 5.20000	3.03000 -3.03000		XZ2256J1 XZ2256H1	LL LL
171	Z2256J2	UL	.	.	NONE .	. 6.00000	3.03000 -3.03000		XZ2256J2 XZ2256H2	LL LL
172	Z2256J3	UL	.	.	NONE .	. 5.40000	3.03000 -3.03000		XZ2256J3 XZ2256H3	LL LL
173	Z2256J4	UL	.	.	NONE .	. 7.30000	3.03000 -3.03000		XZ2256J4 XZ2256H4	LL LL
174	PD01	LL	9.00000	.	9.00000 NONE	7.12000 9.14000	-0.54250 0.54250		PFW01A PSFW01A	LL UL
175	PD14	LL	30.00000	.	30.00000 NONE	25.09114 30.01000	-22148.15000 22148.15000		FFTP14A1 IFTP14A	UL UL
176	PD32	LL	5.61000	.	5.61000 NONE	2.16000 6.66000	-0.78000 0.78000		PFT32A PSFT32A	LL UL
177	PD22	LL	9.18000	.	9.18000 NONE	6.60000 9.62000	-0.35500 0.35500		PFW22A PSFW22A	LL UL
178	PD56	LL	0.26000	.	0.26000 NONE	0.07000 0.70000	-0.60190 0.60190		PX2256H PSFW22A	LL UL
179	PSW01A	UL	1.30000	.	NONE 1.30000	1.16000 3.18000	0.15000 -0.15000		PSFW01A PFW01A	UL LL
180	PSW14A	UL	0.61000	.	NONE 0.61000	0.60000 5.51886	22147.90750 -22147.90750		IFTP14A FFTP14A1	UL UL

MODEL
INTEGER SOLUTIONS FOR MODEL

COLUMNS AT LIMIT LEVEL

NUMBER	COLUMN	ST	ACTIVITY	INPUT COST	LOWER LIMIT UPPER LIMIT	LOWER ACTIVITY UPPER ACTIVITY	UNIT COST UNIT COST	UPPER COST LOWER COST	LIMITING PROCESS	ST ST
264	XW01A2	LL	.	39.25000	.	-151.00000	-39.25000	INFINITY	D012	LL
					NONE	144.00000	39.25000	.	FW01A2	UL
287	XTP14A1	LL	.	98.60000	.	-429.52500	-33.60000	INFINITY	FFTP14A1	UL
					NONE	140.00000	33.60000	65.00000	FTP14A1	UL
288	XTP14A2	LL	.	112.30000	.	-1026.75000	-39.30000	INFINITY	FFTP14A2	UL
					NONE	200.00000	39.30000	73.00000	FTP14A2	UL
289	XTP14A3	LL	.	122.50000	.	-1291.27500	-41.50000	INFINITY	FFTP14A3	UL
					NONE	180.00000	41.50000	81.00000	FTP14A3	UL
290	XTP14A4	LL	.	109.40000	.	-1251.58750	-39.40000	INFINITY	FFTP14A4	UL
					NONE	210.00000	39.40000	70.00000	FTP14A4	UL
291	XFW01A1	LL	.	54.25000	.	-153.30000	-54.25000	INFINITY	D011	LL
					NONE	201.60000	54.25000	.	FFW01A1	UL
292	XFW01A2	LL	.	54.25000	.	-151.00000	-54.25000	INFINITY	D012	LL
					NONE	288.00000	54.25000	.	FFW01A2	UL
293	XFW01A3	LL	.	54.25000	.	-86.40000	-15.00000	INFINITY	FW01A3	UL
					NONE	43.20000	15.00000	39.25000	XW01A3	LL
294	XFW01A4	LL	.	54.25000	.	-106.30000	-15.00000	INFINITY	FW01A4	UL
					NONE	44.90000	15.00000	39.25000	XW01A4	LL
303	XFW32A1	LL	.	39.25000	.	-125.20000	-18.13000	INFINITY	XZ3214E1	LL
					NONE	156.90000	18.13000	21.12000	Z3214E1	UL
304	XFW32A2	LL	.	39.25000	.	-92.00000	-10.13000	INFINITY	XZ3214E2	LL
					NONE	311.00000	10.13000	29.12000	Z3214E2	UL
306	XFW32A4	LL	.	39.25000	.	-165.10000	-13.13000	INFINITY	XZ3214E4	LL
					NONE	258.05000	13.13000	26.12000	Z3214E4	UL
307	XFW22A1	LL	.	35.50000	.	-137.10000	-35.50000	INFINITY	D221	LL
					NONE	302.40000	35.50000	.	FFW22A1	UL
308	XFW22A2	LL	.	35.50000	.	-175.00000	-35.50000	INFINITY	D222	LL
					NONE	432.00000	35.50000	.	FFW22A2	UL

MODEL
INTEGER SOLUTIONS FOR MODEL

COLUMNS AT LIMIT LEVEL

NUMBER	COLUMN	ST	ACTIVITY	INPUT COST	LOWER LIMIT UPPER LIMIT	LOWER ACTIVITY UPPER ACTIVITY	UNIT COST UNIT COST	UPPER COST LOWER COST	LIMITING PROCESS	ST ST
309	XFW22A3	LL	.	35.50000	.	-153.90000	-35.50000	INFINITY	D223	LL
					NONE	388.80000	35.50000	.	FFW22A3	UL
327	XFTP32A1	LL	.	78.00000	.	-125.20000	-56.88000	INFINITY	XZ3214E1	LL
					NONE	156.90000	56.88000	21.12000	Z3214E1	UL
328	XFTP32A2	LL	.	94.00000	.	-92.00000	-64.88000	INFINITY	XZ3214E2	LL
					NONE	311.00000	64.88000	29.12000	Z3214E2	UL
329	XFTP32A3	LL	.	123.00000	.	-373.50000	-83.75000	INFINITY	FFW32A3	UL
					NONE	15.30000	83.75000	39.25000	XFW32A3	LL
330	XFTP32A4	LL	.	82.00000	.	-165.10000	-55.88000	INFINITY	XZ3214E4	LL
					NONE	258.05000	55.88000	26.12000	Z3214E4	UL
339	XZ0122X1	LL	.	115.12000	.	-137.10000	-115.12000	INFINITY	D221	LL
					NONE	50.40000	115.12000	.	Z0122X1	UL
340	XZ0122X2	LL	.	115.12000	.	-175.00000	-115.12000	INFINITY	D222	LL
					NONE	72.00000	115.12000	.	Z0122X2	UL
341	XZ0122X3	LL	.	115.12000	.	-43.20000	-154.37000	INFINITY	XW01A3	LL
					NONE	64.80000	154.37000	-39.25000	Z0122X3	UL
342	XZ0122X4	LL	.	115.12000	.	-44.90000	-118.87000	INFINITY	XW01A4	LL
					NONE	43.00000	118.87000	-3.75000	XFW22A4	LL
343	XZ0122H1	LL	.	12.84000	.	-137.10000	-12.84000	INFINITY	D221	LL
					NONE	.	12.84000	.	Z0122H1	UL
344	XZ0122H?	LL	.	12.84000	.	-175.00000	-12.84000	INFINITY	D222	LL
					NONE	.	12.84000	.	Z0122H2	UL
345	XZ0122H3	LL	.	12.84000	.	-43.20000	-52.09000	INFINITY	XW01A3	LL
					NONE	.	52.09000	-39.25000	Z0122H3	UL
346	XZ0122H4	LL	.	12.84000	.	.	-62.30429	INFINITY	Z0122H2	UL
					NONE	43.00000	62.30429	-49.46429	XFW22A4	LL
347	XZ1456A1	LL	.	30.28000	.	-137.10000	-70.59000	INFINITY	D221	LL
					NONE	.	70.59000	-40.31000	Z1456A1	UL

MODEL
INTEGER SOLUTIONS FOR MODEL

ROWS AT INTERMEDIATE LEVEL

NUMBER	ROW	ST	ACTIVITY	SLACK ACTIVITY	LOWER LIMIT UPPER LIMIT	LOWER ACTIVITY UPPER ACTIVITY	UNIT COST UNIT COST	UPPER COST LOWER COST	LIMITING PROCESS	ST ST
2	D011	BS	678.30000	-153.30000	525.00000	678.30000	12.84000		XZ0122H1	LL
					NONE	779.10000	39.25000		XW01A1	LL
3	D012	BS	776.00000	-151.00000	625.00000	776.00000	12.84000		XZ0122H2	LL
					NONE	920.00000	39.25000		XW01A2	LL
14	D221	BS	673.10000	-137.10000	536.00000	663.30000	15.44000		FW56A1	UL
					NONE	673.10000	12.84000		XZ0122H1	LL
15	D222	BS	770.00000	-175.00000	595.00000	756.00000	15.44000		FW56A2	UL
					NONE	770.00000	12.84000		XZ0122H2	LL
16	D223	BS	459.90000	-153.90000	306.00000	447.30000	15.44000		FW56A3	UL
					NONE	848.70000	35.50000		XFW22A3	LL
22	FW01A1	BS	.	100.80000	NONE	.	INFINITY		NONE	
					100.80000	INFINITY	39.25000		XW01A1	LL
23	FW01A2	BS	.	144.00000	NONE	.	INFINITY		NONE	
					144.00000	INFINITY	39.25000		XW01A2	LL
24	FW01A3	BS	43.20000	86.40000	NONE	.	15.00000		XFW01A3	LL
					129.60000	INFINITY	39.25000		D013	LL
25	FW01A4	BS	44.90000	106.30000	NONE	.	15.00000		XFW01A4	LL
					151.20000	INFINITY	39.25000		D014	LL
46	FTP14A1	BS	.	140.00000	NONE	.	INFINITY		NONE	
					140.00000	969.60000	33.60000		XTP14A1	LL
47	FTP14A2	BS	.	200.00000	NONE	.	INFINITY		NONE	
					200.00000	972.00000	39.30000		XTP14A2	LL
48	FTP14A3	BS	.	180.00000	NONE	.	INFINITY		NONE	
					180.00000	507.60000	41.50000		XTP14A3	LL
49	FTP14A4	BS	.	210.00000	NONE	.	INFINITY		NONE	
					210.00000	847.10000	39.40000		XTP14A4	LL
50	FFW01A1	BS	-201.60000	201.60000	NONE	-INFINITY	35.61508		IFW01A	LL
					.	INFINITY	54.25000		XFW01A1	LL

MODEL
INTEGER SOLUTIONS FOR MODEL

ROWS AT INTERMEDIATE LEVEL

NUMBER	ROW	ST	ACTIVITY	SLACK ACTIVITY	LOWER LIMIT UPPER LIMIT	LOWER ACTIVITY UPPER ACTIVITY	UNIT COST UNIT COST	UPPER COST LOWER COST	LIMITING PROCESS	ST ST
155	Z3214X2	BS	.	135.00000	NONE	.	INFINITY		NONE	
					135.00000	92.00000	4.10000		XZ3214X2	LL
156	Z3214X3	BS	.	121.50000	NONE	.	INFINITY		NONE	
					121.50000	373.50000	6.23000		XZ3214X3	LL
157	Z3214X4	BS	.	141.75000	NONE	.	INFINITY		NONE	
					141.75000	165.10000	4.10000		XZ3214X4	LL
158	Z3214E1	BS	-156.90000	156.90000	NONE	-251.40000	4.10000		XZ3214X1	LL
					.	145.50000	18.13000		XFW32A1	LL
159	Z3214E2	BS	-311.00000	311.00000	NONE	-403.00000	4.10000		XZ3214X2	LL
					.	121.00000	10.13000		XFW32A2	LL
160	Z3214E3	BS	-362.70000	362.70000	NONE	-457.20000	149.63367		IZT1432E	LL
					.	10.80000	2.13000		XZ3214E3	LL
161	Z3214E4	BS	-258.05000	258.05000	NONE	-399.80000	4.10000		XZ3214X4	LL
					.	195.55000	13.13000		XFW32A4	LL
162	Z2256H1	BS	-579.30000	579.30000	NONE	-INFINITY	51.73653		IZT5622H	LL
					.	-579.30000	1.49000		Z2256I1	UL
163	Z2256H2	BS	-829.00000	829.00000	NONE	-INFINITY	36.21557		IZT5622H	LL
					.	-829.00000	1.49000		Z2256I2	UL
164	Z2256H3	BS	-746.10000	746.10000	NONE	-INFINITY	40.23952		IZT5622H	LL
					.	-746.10000	1.49000		Z2256I3	UL
165	Z2256H4	BS	-869.45000	869.45000	NONE	-INFINITY	34.49102		IZT5622H	LL
					.	-869.45000	1.49000		Z2256I4	UL
186	PSFW01A	BS	-0.14000	0.14000	NONE	-INFINITY	3554.45545		IFW01A	LL
					.	1.16000	0.15000		PSW01A	UL
190	PSFW22A	BS	-0.44000	0.44000	NONE	-0.44000	0.31590		PX0122H	LL
					.	INFINITY	0.35500		PD22	LL
195	PSFT32A	BS	-1.05000	1.05000	NONE	-4.50000	2853.91714		IFW32A	LL
					.	1.97000	0.38750		PSFW32A	UL

MODEL
INTEGER SOLUTIONS FOR MODEL

COLUMNS AT INTERMEDIATE LEVEL

NUMBER	COLUMN	ST	ACTIVITY	INPUT COST	LOWER LIMIT UPPER LIMIT	LOWER ACTIVITY UPPER ACTIVITY	UNIT COST UNIT COST	UPPER COST LOWER COST	LIMITING PROCESS	ST ST
247	IFS14A	BS	.	673800.00000	. 1.00000	. 0.59452	INFINITY 164971.33605	INFINITY 508828.66395	NONE PFS14A	 LL
248	IFS32A	BS	.	122610.00000	. 1.00000	. 0.06564	INFINITY 91740.33160	INFINITY 30869.66840	NONE PFS32A	 LL
249	IFTP14A	BS	0.99938	354360.00000	. 1.00000	0.93375 1.08375	215472.48729 354350.24320	569832.48729 9.75680	IZT1432E PZ3214X	LL UL
253	IZT0122H	BS	.	40080.00000	. 1.00000	. 1.14058E 27	INFINITY 40080.00000	INFINITY .	NONE Z0122H4	 UL
254	IZT1456A	BS	.	7940.00000	. 1.00000	. 1.32275E 28	INFINITY 7940.00000	INFINITY .	NONE Z1456A4	 UL
255	IZT1456B	BS	.	10600.00000	. 1.00000	. 7.05467E 27	INFINITY 10600.00000	INFINITY .	NONE Z1456B4	 UL
256	IZT1456C	BS	.	13850.00000	. 1.00000	. 4.40917E 27	INFINITY 13850.00000	INFINITY .	NONE Z1456C4	 UL
258	IZT3222H	BS	.	72590.00000	. 1.00000	. 1.14058E 27	INFINITY 72590.00000	INFINITY .	NONE Z3222H4	 UL
259	IZT3222I	BS	.	99300.00000	. 1.00000	. 7.96972E 26	INFINITY 99300.00000	INFINITY -4.76837E-07	NONE Z3222I4	 UL
265	XW01A3	BS	43.20000	39.25000	. NONE	-216.00000 129.60000	15.00000 39.25000	54.25000 .	XFW01A3 D013	LL LL
266	XW01A4	BS	44.90000	39.25000	. NONE	-257.50000 151.20000	15.00000 39.25000	54.25000 .	XFW01A4 D014	LL LL
267	XW14A1	BS	50.40000	24.25000	. NONE	-379.12500 50.40000	40.75000 INFINITY	65.00000 -INFINITY	FW14A1 NONE	UL
268	XW14A2	BS	72.00000	24.25000	. NONE	-954.75000 72.00000	48.75000 INFINITY	73.00000 -INFINITY	FW14A2 NONE	UL
269	XW14A3	BS	64.80000	24.25000	. NONE	-1226.47500 64.80000	56.75000 INFINITY	81.00000 -INFINITY	FW14A3 NONE	UL

MODEL
INTEGER SOLUTIONS FOR MODEL

COLUMNS AT INTERMEDIATE LEVEL

NUMBER	COLUMN	ST	ACTIVITY	INPUT COST	LOWER LIMIT UPPER LIMIT	LOWER ACTIVITY UPPER ACTIVITY	UNIT COST UNIT COST	UPPER COST LOWER COST	LIMITING PROCESS	ST ST
284	XS222	BS	776.00000	6.60000	. NONE	776.00000 776.00000	INFINITY INFINITY	INFINITY -INFINITY	NONE NONE	
285	XS223	BS	465.30000	6.60000	. NONE	465.30000 465.30000	INFINITY INFINITY	INFINITY -INFINITY	NONE NONE	
286	XS224	BS	678.30000	6.60000	. NONE	678.30000 678.30000	INFINITY INFINITY	INFINITY -INFINITY	NONE NONE	
295	XFW14A1	BS	151.20000	31.75000	. NONE	-278.32500 151.20000	33.25000 INFINITY	65.00000 -INFINITY	FFW14A1 NONE	UL
296	XFW14A2	BS	216.00000	31.75000	. NONE	-810.75000 216.00000	41.25000 INFINITY	73.00000 -INFINITY	FFW14A2 NONE	UL
297	XFW14A3	BS	194.40000	31.75000	. NONE	-1096.87500 194.40000	49.25000 INFINITY	81.00000 -INFINITY	FFW14A3 NONE	UL
298	XFW14A4	BS	226.80000	31.75000	. NONE	-1024.78750 226.80000	38.25000 INFINITY	70.00000 -INFINITY	FFW14A4 NONE	UL
299	XFW14B1	BS	252.00000	39.25000	. NONE	-177.52500 252.00000	25.75000 INFINITY	65.00000 -INFINITY	FFW14B1 NONE	UL
300	XFW14B2	BS	360.00000	39.25000	. NONE	-666.75000 360.00000	33.75000 INFINITY	73.00000 -INFINITY	FFW14B2 NONE	UL
301	XFW14B3	BS	324.00000	39.25000	. NONE	-967.27500 324.00000	41.75000 INFINITY	81.00000 -INFINITY	FFW14B3 NONE	UL
302	XFW14B4	BS	378.00000	39.25000	. NONE	-873.58750 378.00000	30.75000 INFINITY	70.00000 -INFINITY	FFW14B4 NONE	UL
305	XFW32A3	BS	15.30000	39.25000	. NONE	-347.40000 378.00000	70.63000 2.13000	109.88000 37.12000	XZ1432E3 XZ3214E3	LL LL
310	XFW22A4	BS	43.00000	35.50000	. NONE	-63.30000 453.60000	62.30429 35.50000	97.80429 .	XZ0122H4 D224	LL LL
311	XFS01A1	BS	678.30000	6.60000	. NONE	678.30000 728.13746	INFINITY 228.30779	INFINITY -221.70779	NONE IFS01A	 LL

MODEL
INTEGER SOLUTIONS FOR MODEL

COLUMNS AT INTERMEDIATE LEVEL

NUMBER	COLUMN	ST	ACTIVITY	INPUT COST	LOWER LIMIT	LOWER ACTIVITY	UNIT COST	UPPER COST	LIMITING	ST
					UPPER LIMIT	UPPER ACTIVITY	UNIT COST	LOWER COST	PROCESS	ST
419	PES32	BS	4.52000	0.06600	.	4.52000	INFINITY	INFINITY	NONE	
					NONE	4.52000	INFINITY	-INFINITY	NONE	
420	PES22	BS	6.79000	0.06600	.	6.79000	INFINITY	INFINITY	NONE	
					NONE	6.79000	INFINITY	-INFINITY	NONE	
421	PET14A	BS	1.40000	0.98600	.	1.39000	22147.16400	22148.15000	PST14A	UL
					NONE	1.40000	INFINITY	-INFINITY	NONE	
422	PFW01A	BS	1.88000	0.54250	.	1.88000	22147.53100	22148.07350	PZ0114J	UL
					NONE	2.02000	0.15000	0.39250	PSW01A	UL
423	PFW14A	BS	1.51000	0.31750	.	1.50000	22147.83250	22148.15000	PSFW14A	UL
					NONE	1.51000	INFINITY	-INFINITY	NONE	
424	PFW14B	BS	2.52000	0.39250	.	2.51000	22147.75750	22148.15000	PSFW14B	UL
					NONE	2.52000	INFINITY	-INFINITY	NONE	
425	PFW32A	BS	3.02000	0.39250	.	1.97000	0.38750	0.78000	PSFW32A	UL
					NONE	6.47000	2853.91714	-2853.52464	IFW32A	LL
426	PFW22A	BS	2.58000	0.35500	.	2.58000	0.31590	0.67090	PX0122H	LL
					NONE	3.02000	0.35500	.	PD22	LL
427	PFS01A	BS	5.82000	0.06600	.	5.82000	INFINITY	INFINITY	NONE	
					NONE	6.24762	26608.44876	-26608.38276	IFS01A	LL
430	PFT14A	BS	15.99000	0.65000	.	14.94000	13467.03046	13467.68046	IZT1432E	LL
					NONE	16.00000	22146.89020	-22146.24020	PZ3214X	UL
431	PFT32A	BS	3.45000	0.78000	.	-INFINITY	2853.91714	2854.69714	IFW32A	LL
					NONE	4.50000	0.38750	0.39250	PSFW32A	UL
432	PX0114J	BS	.	0.07650	.	-0.01000	22147.53100	22147.60750	PZ0114J	UL
					NONE	.	INFINITY	-INFINITY	NONE	
433	PX0114K	BS	.	0.06460	.	-0.01000	22147.54290	22147.60750	PZ0114K	UL
					NONE	.	INFINITY	-INFINITY	NONE	
448	PX3214X	BS	1.35000	0.47980	.	1.34000	22146.89020	22147.37000	PZ3214X	UL
					NONE	1.35000	INFINITY	-INFINITY	NONE	