8024

CONTRACTOR REPORT

SAND87—7096
Unlimited Release
UC—94e

RS-8232-2 65986
C. 1
8232-2//065986
00000001

AUTSKED: REVISITED

Dallas W. Sasser
Creative Computer Services
5301 Central, Suite 915
Albuquerque, NM 87108

and

Dale K. Buchanan
Division 6257
Sandia National Laboratories
Albuquerque, NM 87185

Prepared by Sandia National Laboratories Albuquerque, New Mexico 87185 and Livermore, California 94550 for the United States Department of Energy under Contract DE-AC04-76DP00789

Printed April 1987

1090075

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors or subcontractors.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy: A04
Microfiche copy: A01

Category UC-94e

SAND87-7096
Unlimited Release
Printed April 1987

AUTSKED: REVISITED

Dallas W. Sasser
Creative Computer Services
5301 Central, Suite 915
Albuquerque, New Mexico 87108

and

Dale K. Buchanan
Division 6257
Sandia National Laboratories
Albuquerque, New Mexico 87185

ABSTRACT

AUTSKED is a FORTRAN program created to automate scheduling of leaching and oil filling of salt caverns at the Strategic Petroleum Reserve. There have been a number of revisions to the AUTSKED program since the first version. This report documents the latest embellishments to the program.

DISCLAIMER OF WARRANTY

This program is presented without any express or implied warranties whatsoever. Because of the diversity of conditions and hardware under which this program may be used, no warranty of fitness for a particular purpose is offered. The user is advised to test the program thoroughly before relying upon it. The user must assume the entire risk of using the program.

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
II.	PROGRAM STRUCTURE	2
III.	NEW FEATURES IN AUTSKED	6
IV.	OPERATIONAL CONSIDERATIONS	11
V.	SUMMARY	13
REFERENCES		14
APPENDICES		
A.	Program SPRSKD Listing	A-1
B.	NETWRK File Listing	B-1
C.	CDC NOS Procedure File	C-1
D.	Sample SPRSKD Input/Output	D-1

LIST OF FIGURES

2.1.	Program Flow Diagram	3
4.1.	Program Interactions	11

I. INTRODUCTION

The purpose of this report is to describe the latest set of changes and additions to the AUTSKED program. The original AUTSKED program was documented in a Sandia National Laboratories (SNL) report published in March, 1984.[1] After the program was in use for several months, the AUTSKED user community requested several additional features. These additional features were implemented and subsequently documented in an SNL report published in April, 1985.[2] During the year 1986, more requests for modifications to AUTSKED were made. This report documents the implementation of most of these requests. A detailed discussion of these modifications is found in Section III.

AUTSKED is used in conjunction with two other SNL created programs, EDITOR and PREDICT.[3] Previously, these three programs had been executed separately, each one providing output to be used by the others. Now, the three programs AUTSKED, EDITOR, and PREDICT, have been incorporated as subprograms of a single program, SPRSKD. AUTSKED also uses a FORTRAN software package called Simulation Language for Alternative Modeling (SLAM), which is described in Section IV.[5,6]

AUTSKED, EDITOR and PREDICT are FORTRAN programs, written in ANSI standard FORTRAN 77. It is the author's understanding that the currently available version of SLAM (SLAM II) is also written in ANSI standard FORTRAN 77. Thus, there should not be much difficulty in adapting these programs to other computers which have a FORTRAN 77 compiler. Efforts have been made to ensure that the FORTRAN statements in AUTSKED, EDITOR and PREDICT do not depend on word size, which varies from computer to computer. There are no mathematical calculations in any of these programs which require high precision.

II. PROGRAM STRUCTURE

In this section an overview of the program structure and operation is presented. The program is highly interactive, requiring decisions and input by the user for each of the three subprograms EDITOR, PREDICT and AUTSKED. In this discussion, as well as in the program itself, it is assumed that the user is knowledgeable about the various stages in cavern development as described in References 1 and 3.

A program flow diagram is displayed in Figure 2.1. At program start, the user is given the following four choices:

1. Edit old site file or create new site file using EDITOR
2. Develop individual cavern schedules using PREDICT
3. Develop site schedule using AUTSKED
4. Exit from program

Option 4 needs no explanation. The other options will be discussed in more detail.

EDITOR is an interactive FORTRAN subprogram which guides the user through a sequence of interrogations to create a new site file or to edit an existing or old site file. The user is asked for a file name which is restricted to eight or fewer alphanumeric characters, the first of which must be alpha. The site file contains characteristics of the Strategic Petroleum Reserve (SPR) site, the number of caverns, present state of each cavern, desired leaching processes, days of workover, brine production schedules, stage end volumes, etc. This file is used as input to PREDICT. In editing an old site file, these various attributes can be changed for each individual cavern. The edited file can replace the old file or it can be given a new name. In creating a new site file, the user can input the requested information for each cavern. Alternatively, the user can input this information for one cavern, and this becomes the default data for all subsequent caverns of the site. Individual caverns may then be edited as desired.

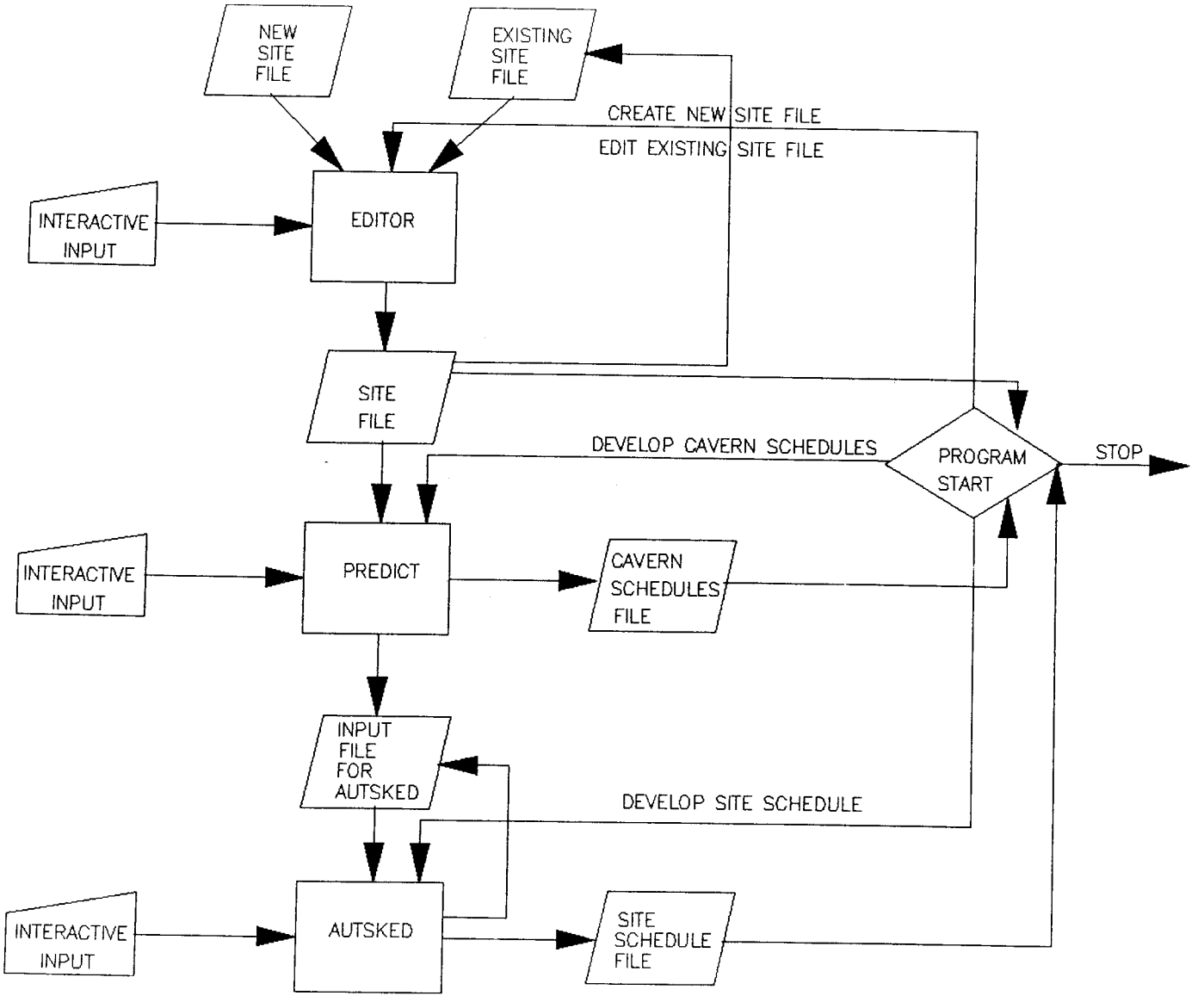


Figure 2.1. Program Flow Diagram

PREDICT is also an interactive FORTRAN subprogram which requires for input a file created by EDITOR. The user is requested to provide the name of this file. PREDICT uses the information in the EDITOR site file to develop individual cavern schedules for each cavern at a site. These schedules are created based upon predicted leaching efficiencies as a function of brine production flow rates. The scientific support for these leaching efficiencies is described in Reference 4. The important thing to observe here is that PREDICT produces individual cavern schedules independent of any constraints such as brine disposal capability, oil availability, equipment available for workovers, etc. These schedules are meaningful only if all caverns at a site can be developed simultaneously, which is not possible. Thus, AUTSKED was developed to cope with this problem.

There are two output files generated by PREDICT. One is a listing of the individual cavern schedules and the second is used as an input file for AUTSKED. The user is asked for a name for the first file, or instructed that the default option is a listing at the user's terminal. The user is also instructed to insert a name for the file to be used as an input to AUTSKED.

AUTSKED is another interactive FORTRAN subprogram which requires for input a file created by PREDICT. The user is requested to provide the name of this file. AUTSKED uses the individual cavern schedules created by PREDICT and, along with a cavern priority ranking, creates a site schedule subject to the following constraints:

1. Maximum allowable brine flow from a site
2. Maximum allowable oil flow to a site
3. Maximum allowable brine flow from each cavern
4. Maximum allowable oil flow to each cavern
5. Number of workover rigs available

The critical element in this site schedule development is the cavern priority ranking. The original version of AUTSKED[1] made an attempt to

find a dynamic cavern prioritization scheme which would minimize the time for completion of a site. The site-scheduling problem is a specific instance of the general problem of activity scheduling with resource allocation constrained by limited resource availability. There are no methods known for optimally solving this problem. In Reference 1 several heuristic methods were examined, one of which appeared to be superior over the others. This scheme was referred to in Reference 1 as "shortest time remaining (STR)." In site-scheduling terminology, at each breakpoint, caverns are prioritized according to the time needed to complete the cavern's development (leaching and oil filling), with the cavern having the shortest time to completion being given first priority, the next shortest having second priority, etc. This cavern prioritization is one of the options available in AUTSKED. The other prioritization option is for the user to provide an initial ranking of caverns. Previously this user-ranking of caverns persisted throughout the site development. The current version of AUTSKED allows some flexibility in changing cavern rankings during site development. Section III discusses this further.

The output file of AUTSKED (name of file is requested of the user) contains the site schedule. Two user-selected options for additional information in this file are:

1. A tabulation, at each breakpoint and for each cavern, of the current cavern and oil volume and, at each breakpoint, a site total cavern and oil volume.

2. A "clock" for each cavern which displays at each breakpoint the accumulated number of days the cavern was in active development or was idle (zero brine flow rate, as scheduled by PREDICT). This "clock" time does not include time during which the cavern was idle because AUTSKED did not schedule it due to brine flow, oil flow or workover constraints.

Option 1 is self-explanatory. Option 2 is clarified in Section III.

III. NEW FEATURES IN AUTSKED

This section describes the latest embellishments to AUTSKED. There were eight changes requested for AUTSKED which became the basis of modifications made in the program. These requested changes are stated below, followed by a discussion of how they were or why they were not implemented.

1. Permit the user to specify the oil/water ratio in leach/fill mode.

   In the leach/fill mode of cavern development, cavern leaching and oil filling occur simultaneously during some stages of development. Previously, the oil/water ratio during these stages was determined from SANSMIC predictions.[4] This has been changed so that the user can either implement the SANSMIC ratios or specify different ratios. Implementation has been made in the form of an option in the interactive input to PREDICT (see Figure 2.1). There are no restrictions upon the user in his selection of ratios and no caveats (except one), so discretion should be used since unrealistic ratios may produce unrealistic results.

2. Permit the temporary storage (surge storage) of oil and its removal concurrent with and subsequent to the leaching process.

   When this option was first proposed there was discussion about the feasibility of doing this in AUTSKED. It is closely related to the problem of "drawdown" (i.e., the removal of oil from the SPR when it is needed). AUTSKED was not developed for this purpose and a major restructuring of the program would be required in order to have this option, therefore it has not been implemented.

3. Provide the user with the capability of reassigning cavern priorities at any time during the site development.

   This feature permits the user to allow for events that occur during site development, e.g., equipment failure at individual

caverns. This request has been accommodated in the following manner. In the interactive input to AUTSKED (see Figure 2.1), the user may specify that site scheduling be interrupted after a number of days in order to change cavern priorities. Changes in cavern priorities to be instigated at this time are requested from the user. The user may then specify that site scheduling be interrupted again at some later time for another change in priorities. This process goes on until the user chooses to continue scheduling to site completion, with no further changes in priorities. AUTSKED currently allows for 60 different cavern priority specifications (this means 59 changes after the initial prioritization). The value of 60 is arbitrary and can be changed with appropriate dimension statement modifications. For a site development time of 1200 days, this means cavern priorities could be changed every 20 days on the average.

All of this information, time of priority change and new priorities, is saved in the original input file for AUTSKED (see Figure 2.1) which was created by PREDICT. For discussion purposes, we shall call this file AUTIN. This is the reason for the feedback loop from the block labeled "AUTSKED" to the block labeled "INPUT FILE FOR AUTSKED" in Figure 2.1. If file AUTIN is used again as the input file to AUTSKED, the priority changes made previously will be part of the scheduling in AUTSKED and the user may specify some subsequent priority changes if desired. AUTSKED will then rerun and add on the priority changes.

This option is available only if the user-specified priority option is selected in AUTSKED, i.e., it is not available if the STR priority scheme (see page 5, line 8) is chosen.

4. EDITOR and PREDICT should have the same time reference as AUTSKED. If AUTSKED indicates that a change is required at day 1001 in cavern 9, the user must be able to relate day 1001 in AUTSKED to day 1001 in EDITOR and PREDICT.

This is the original statement of the request, and the first author's interpretation is as follows. Recall that PREDICT creates individual cavern schedules independent of any constraints. Thus the PREDICT schedule for cavern 9 may say that from day 200 to 300 there should be a brine flow rate of 100,000 barrels per day. "Day 200 to 300" is relative to the development of this cavern. In the site schedule produced by AUTSKED, this activity for cavern 9 may occur on day 800 to 900 (days now being relative to site development). The objective of this request is to relate day 800 to 900 in the AUTSKED site schedule to day 200 to 300 in the PREDICT schedule for cavern 9. This will allow the user, after examining the site schedule, to make changes, via EDITOR, in the PREDICT schedules in order to achieve a reduced site completion time. This correlation between day intervals in the site schedule and day intervals in the individual cavern schedules is meaningful provided that AUTSKED adheres to the original cavern schedules created by PREDICT. However, because of constraints, AUTSKED does not adhere to these schedules. For example, AUTSKED may decide that it can allocate only 50,000 barrels per day to cavern 9. The site schedule will then show a flow rate of 50,000 barrels per day to cavern 9 from day 800 to day 1000. The extra 100 days are to allow for the reduced flow rate.

Complete correlation between day intervals in the site schedule and day intervals in PREDICT cavern schedules is not possible. What has been implemented in AUTSKED is the following. A "clock" is associated with each cavern. This clock is "running" if, in AUTSKED, the cavern is being actively developed or is idle (zero brine flow rate as scheduled in PREDICT). Otherwise the clock is not running. This means that the clock is not running if AUTSKED has not scheduled this cavern for active development because of constraints. The clock is running during idle time specified in PREDICT because this is a part of the PREDICT schedule for this cavern.

This "cavern clock" is implemented as an option in the interactive input to AUTSKED. The user may or may not select it, as desired.

5. Provide a capability in AUTSKED to redefine the maximum flow rate for each cavern.

As a result of schedule changes during development, it may be necessary to redefine maximum flow rates to some caverns to effect site completion time. This has been implemented in the following way. In the interactive input to AUTSKED, the user specifies a default maximum cavern brine flow and a default maximum cavern oil flow. These default values may be used for all caverns at a site or, alternatively, the user may specify different maximum flow rates for individual caverns.

6. Tabulate the total oil stored and the total storage volume created per cavern and per site at each breakpoint.

This has been implemented in the interactive input to AUTSKED. It is an option which can be selected at the user's discretion. It was included to provide running totals of these parameters.

7. Maintain maximum site flow rates as long as possible even if it means flow to a cavern that has a zero flow rate requested by EDITOR.

Specification of zero flow rate to a cavern for a number of days in EDITOR was originally intended to allow active development of some caverns to begin after the start date of site development. This option has also been used to stop development of a cavern at any time in order to stagger development of caverns in the hope of reducing site completion time. The purpose of this request is to reactivate flow to some of these caverns at zero flow when there are not enough caverns being leached to use up the maximum allowable site brine flow. However, there may be other reasons for

maintaining zero flow to the caverns for the specified number of days. Therefore this request has been converted into an option in the interactive input to AUTSKED. The user may "force" zero flow rates to be maintained or opt to make them dependent on the maximum site flow utilization. This dichotomy applies to all caverns at zero flow, rather than on an individual basis.

8. Provide for a reduced flow to the lower priority caverns in lieu of reducing one or more caverns to zero flow.

This request is accommodated by number 5 where a maximum cavern brine flow can be specified for each cavern. The flow automatically will revert to the default maximum cavern brine flow when necessary to use as much of the maximum site brine flow as possible. This feature allows the continued development of caverns that would otherwise be reduced to zero flow rate for some period of time.

IV. OPERATIONAL CONSIDERATIONS

Figure 4.1 contains a diagram depicting the interactions among the components of the complete program. The master program SPRSKD contains as subprograms EDITOR, PREDICT and AUTSKED. As mentioned before, these three programs had previously been executed separately but are now all part of the main program, SPRSKD. The final two components are SLAM and NETWRK.

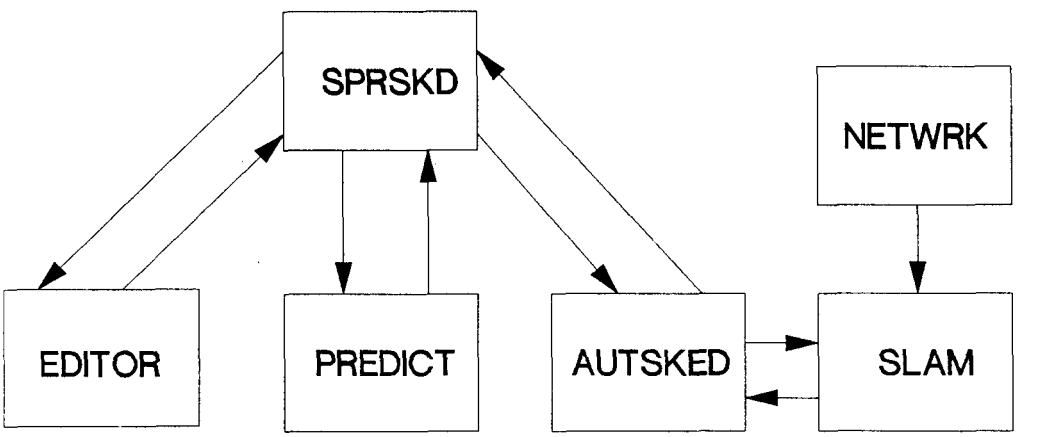


Figure 4.1. Program Interactions

SLAM is a proprietary FORTRAN program distributed by Pritsker and Associates, West Lafayette, Indiana.[5,6] SLAM provides network symbols for building graphical models which are easily translated into input statements for direct computer processing. It contains subprograms that support both discrete event and continuous model developments. The subprogram AUTSKED utilizes all three of these modeling aspects of SLAM: network, discrete event and continuous simulation. The network aspect is incorporated into NETWRK. NETWRK is a graphical description, using the SLAM syntax, of the scheduling of leaching and filling activities at an SPR site. The NETWRK file is an input to SLAM. When executing SPRSKD, this file must have the name NETWRK. The discrete event and continuous simulation aspects are in AUTSKED's subroutines, which SLAM calls when needed.

The program begins with SPRSKD. This program calls EDITOR, PREDICT or AUTSKED at the user's option. EDITOR and PREDICT are stand-alone subprograms. There is no interaction between either of them and SLAM. When AUTSKED is called, it calls SLAM which is then in control. SLAM reads and interprets the NETWRK file and site scheduling begins. There is considerable interaction between SLAM and AUTSKED. The subroutines in AUTSKED provide information to SLAM-- information which is specific to this particular application of SLAM. In computer jargon, AUTSKED is a "library" of subroutines which is a supplement of SLAM.

It is important to emphasize again that EDITOR, PREDICT, AUTSKED and SLAM are all FORTRAN programs written in ANSI FORTRAN 77. NETWRK is a special file written in a syntax understood by SLAM. The structure of this file would have to be changed only if the sequence of events involved in the development of an SPR site is changed from that described in Reference 1.

V. SUMMARY

The latest embellishments to the EDITOR, PREDICT and AUTSKED programs, which have been described in this report, provide the user with increased flexibility to modify the scheduling of leaching and filling of caverns at an SPR site. Eight requests for changes have been addressed and all but one have been implemented. The one excepted was considered to be outside the capability of the current structure of the programs and would require major changes in the structure. The result now is the master program, SPRSKD, which incorporates all three subprograms EDITOR, PREDICT and AUTSKED (see Figure 4-1).

REFERENCES

1. Dallas W. Sasser, "AUTSKED: Automated Scheduling Technique for Leaching and Filling Activities at the Strategic Petroleum Reserve," Sandia National Laboratories, Report No. SAND83-2583, March 1984.

2. Dallas W. Sasser, "AUTSKED: An Addendum," Sandia National Laboratories, Report No. SAND83-2583/2, April 1985.

3. Robert J. Gross, "User Instructions for Programs NEDIT and NPRED," Sandia National Laboratories, Report No. SAND82-2189, February 1983.

4. A. J. Russo, "User's Manual for the Solution Mining Code SANSMIC," Sandia National Laboratories, Report No. SAND83-1150, September 1983.

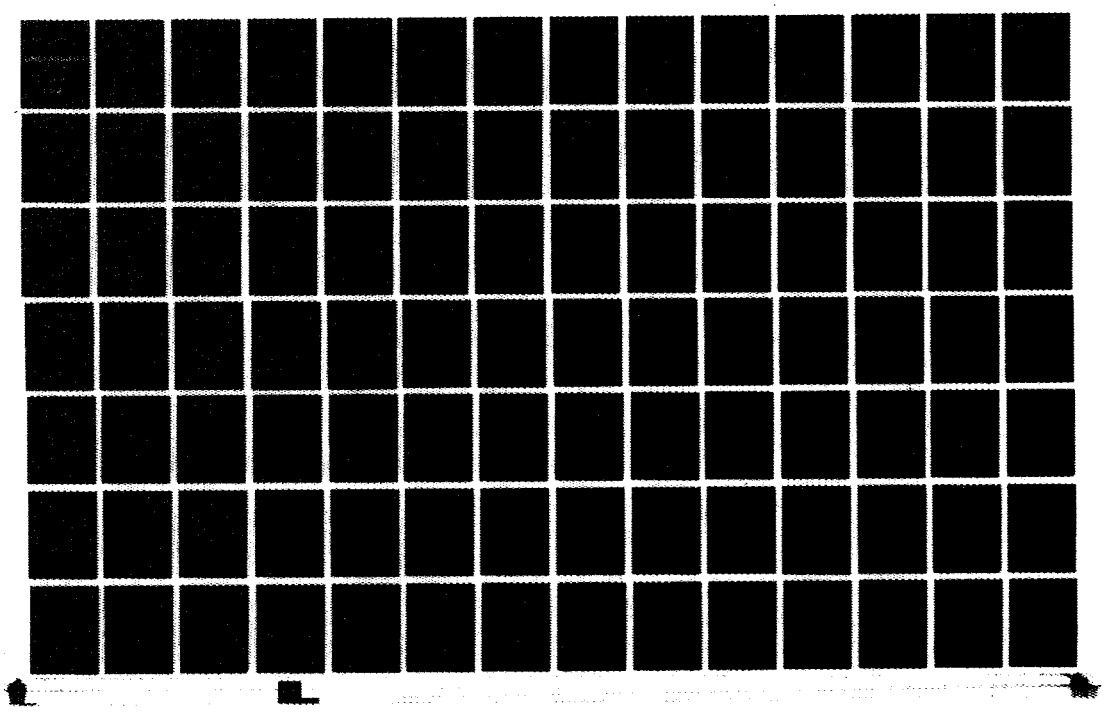
5. A. A. B. Pritsker and C. D. Pegden, "Introduction to Simulation and SLAM," Halsted Press, John Wiley, New York, and Systems Publishing Corp., West Lafayette, Indiana, 1979.

6. A. A. B. Pritsker, "Introduction to Simulation and SLAM II," Second Edition, Halsted Press, John Wiley, New York, and Systems Publishing Corp., West Lafayette, Indiana, 1984.

APPENDIX A

PROGRAM SPRSKD LISTING

This appendix contains a listing of SPRSKD, which combines the previously separate EDITOR, PREDICT and AUTSKED programs into one program.



SECURITY CLASSIFICATION

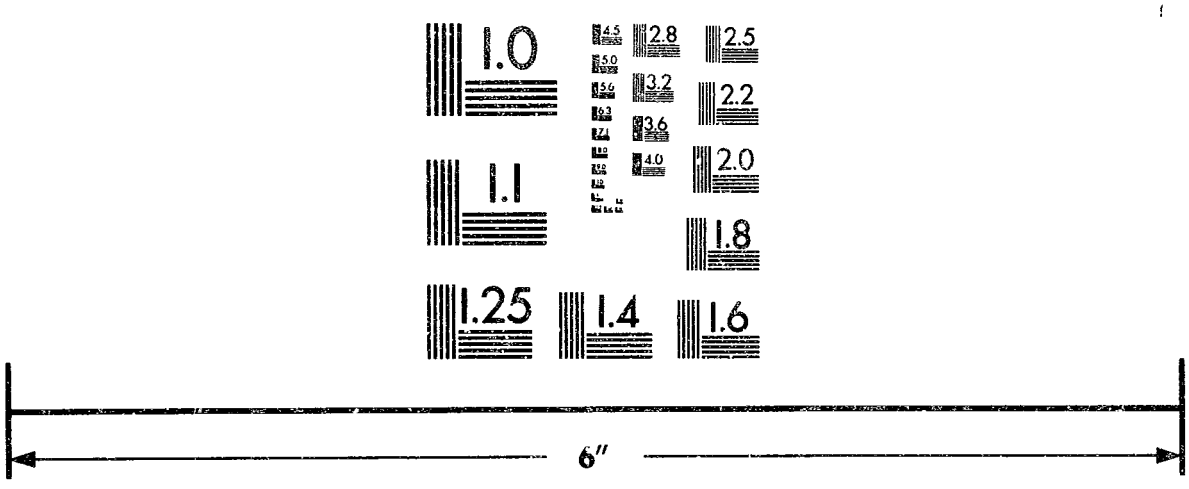
UNCLASSIFIED

DATE OF MICROFILMING
7/22/87

MICROFILMED BY
O. ARMIJO

SANDIA NATIONAL LABS

IMAGE EVALUATION TEST TARGET (MT-3)



PHOTOGRAPHIC SCIENCES CORPORATION
770 BASKET ROAD
P.O. BOX 338
WEBSTER, NEW YORK 14580
(716) 265-1600

```
      PROGRAM MAIN
C
C     PROGRAM SPRSKD
C           "PROGRAM MAIN" IS USED FOR COMPATIBILITY WITH SLAM
C
      COMMON /FILES/OUTFIL,PRDFIL,INFIL
      CHARACTER*8 OUTFIL,PRDFIL,INFIL
      OPEN(5,FILE='INPUT')
      OPEN(6,FILE='OUTPUT')
    1 WRITE(6,8000)
 8000 FORMAT(1H ,///,30X,28H********** MENU ************///
     *               5X,24H1. EDIT OLD SITE FILE OR
     *                 ,34H CREATE NEW SITE FILE USING EDITOR//
     *               5X,21H2. DEVELOP INDIVIDUAL
     *                 ,31H CAVERN SCHEDULES USING PREDICT//
     *               5X,38H3. DEVELOP SITE SCHEDULE USING AUTSKED//
     *               5X,20H4. EXIT FROM PROGRAM///
     *               1X,22HENTER NUMBER OF CHOICE)
      READ(5,8010,END=8001,IOSTAT=IEOF)ICH
 8001 IF(IEOF.NE.0.OR.ICH.LT.1.OR.ICH.GT.4) THEN
         REWIND 5
         WRITE(6,8002)
 8002    FORMAT(1H ,19HNOT PROPER RESPONSE///)
         GO TO 1
      ENDIF
 8010 FORMAT(I1)
      GO TO(9001,9002,9003,9004),ICH
 9001 CALL EDITOR
      WRITE(6,8040)OUTFIL
 8040 FORMAT(1H ,//27H**** EDITING COMPLETED ****/
     *       10X,27HSITE FILE WITH CHANGES IS  .A8)
      GO TO 1
 9002 CALL PREDCT
      WRITE(6,8030)OUTFIL
 8030 FORMAT(1H ,//44H**** PREDICT CAVERN SCHEDULES COMPLETED ****/
     *       10X,30HFILE FOR INPUT TO AUTSKED IS  ,A8)
      IF(PRDFIL.NE.'OUTPUT') THEN
         WRITE(6,8031)PRDFIL
 8031    FORMAT(1H ,10X,34HPREDICT CAVERN SCHEDULES IN FILE  ,A8)
      ENDIF
      GO TO 1
 9003 CALL AUTSKD
      WRITE(6,8020)
 8020 FORMAT(1H ,//33H**** SITE SCHEDULE COMPLETED ****)
      IF(OUTFIL.NE.'OUTPUT') THEN
         WRITE(6,8021)OUTFIL
 8021    FORMAT(1H ,10X,23HSITE SCHEDULE IN FILE  ,A8)
      ENDIF
      GO TO 1
 9004 END
           SUBROUTINE AUTSKD
C
```

```
C*********************************************************************
C
C              CAVERN ATTRIBUTE VALUES (THE ATRIB(.) ARRAY)
C
C     ATRIB NUMBER                DEFINITION
C          1          # OF DAYS LEACHING IN SUMP(S) STAGE
C          2          # OF DAYS IN WORKOVER AFTER SUMP STAGE
C          3          # OF DAYS LEACHING IN SUMP/CHIMNEY(SC) STAGE
C          4          # OF DAYS IN WORKOVER AFTER SUMP/CHIMNEY STAGE
C          5          # OF DAYS LEACHING IN REVERSE1(R1) STAGE
C          6          # OF DAYS IN WORKOVER AFTER REVERSE1 STAGE
C          7          # OF DAYS IN OIL FILL(F1) AFTER REVERSE1 STAGE
C          8          # OF DAYS LEACHING IN REVERSE2(R2) STAGE
C          9          # OF DAYS IN WORKOVER AFTER REVERSE2 STAGE
C         10          # OF DAYS IN OIL FILL(F2) AFTER REVERSE2 STAGE
C         11          # OF DAYS LEACHING IN REVERSE3(R3) STAGE
C         12          # OF DAYS IN WORKOVER AFTER REVERSE3 STAGE
C         13          # OF DAYS IN OIL FILL(F3) AFTER REVERSE3 STAGE
C         14          BRINE FLOW RATF IN S STAGE
C         15          OIL FLOW RATE IN S STAGE
C         16          BRINE FLOW RATE IN SC STAGE
C         17          OIL FLOW RATE IN SC STAGE
C         18          BRINE FLOW RATE IN R1 STAGE
C         19          OIL FLOW RATE IN R1 STAGE
C         20          OIL AND BRINE FLOW RATES IN F1
C         21          BRINE FLOW RATE IN R2 STAGE
C         22          OIL FLOW RATE IN R2 STAGE
C         23          OIL AND BRINE FLOW RATES IN F2
C         24          BRINE FLOW RATE IN R3 STAGE
C         25          OIL FLOW RATE IN R3 STAGE
C         26          OIL AND BRINE FLOW RATES IN F3
C         27          USED BRIEFLY IN INITIALIZING THE NETWORK AND THEN
C                     BECOMES THE CAVERN RANKING
C         28          INITIALLY STORES THE CAVERN RANKING DURING NETWORK
C                     INITIALIZATION AND THEN IS TRANSFERRED TO ATRIB(27)
C                     ATRIB(28) IS SUBSEQUENTLY USED TO SAVE THE NOMINAL
C                     BRINE FLOW RATE (FROM PREDICT) DURING LEACHING WHEN
C                     A SCHEDULING CHANGE IN THIS RATE IS MADE. ATRIB(40)
C                     IS USED TO SAVE THE NUMBER OF FLOW DAYS AT THE
C                     NOMINAL FLOW RATE.
C         29          CAVERN NUMBER (101, 102, ETC.)
C         30          A FLAG, EQUAL TO -1 WHEN THERE ARE INSUFFICIENT
C                     RESOURCES (BRINE FLOW OR OIL FLOW) TO ALLOW
C                     LEACHING OF THIS CAVERN, EQUAL TO 0 OTHERWISE
C         31          STORES THE TIME WHEN WORK FIRST BEGINS ON THE
C                     CAVERN IN THE SCHEDULE. USED TO COLLECT STATISTICS
C                     ON CAVERN COMPLETION TIMES.
C      32,33,34       REVERSE2 STAGE CAN HAVE TWO SUBSTAGES, R2-1, R2-2.
C                     IF BOTH OF THESE OCCUR, BRINE FLOW RATE, OIL FLOW
C                     RATE AND # OF DAYS FOR R2-2 ARE STORED IN 32,33,34,
C                     RESPECTIVELY.
C         35          STORES # OF DAYS IN WORKOVER AFTER R2-2, UNDER THE
C                     ABOVE ASSUMPTIONS.
```

```
C      36,37,38      REVERSE3 STAGE CAN HAVE TWO SUBSTAGES, R3-1, R3-2.
C                    VALUES ARE STORED IN 36,37,38 ANALOGOUS TO 32,33,34
C         39         STORES # OF DAYS IN WORKOVER AFTER R3-2.
C         40         SEE ATRIB(28) ABOVE. ATRIB(40) IS ALSO USED IN THE
C                    WORKOVER SUBROUTINES WO1 AND WO2 AND IS ALSO USED
C                    AS A FLAG (VALUE=-2) TO INDICATE THAT A CAVERN
C                    ORIGINALLY AT ZERO BRINE FLOW RATE HAS BEEN BROUGHT
C                    ON LINE.
C
C*********************************************************************
C
      CHARACTER*5 ST(21),STSV(21),OF(21)
      CHARACTER*8 BDAT,EDATE
      CHARACTER*8 OUTFIL,INFIL,PRDFIL
      DIMENSION NSET(8000)
      COMMON/FILES/OUTFIL,PRDFIL,INFIL
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/XCOM1/ MFE(100),MLE(100),NQ(100)
      COMMON/GCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
     1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
     2TTSET,XXI(100),TTTS,TTTF
      COMMON/XCOM4/ NXACT(100),NXCNT(100),NXGAT(25),NXRSC(75)
      COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
     *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
      COMMON/UCOM2/ST,STSV,BDAT,OF
      COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),OVOL(20)
     *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
      COMMON/UCOM4/RANC(20,60),RT(60),ITR
      COMMON QSET(8000)
      EQUIVALENCE (NSET(1),QSET(1))
      OUTFIL='OUTPUT'
      INFIL='TAPE55'
      CLOSE(5)
      CLOSE(6)
      OPEN(45,FILE='INPUT')
      WRITE(45,8030)
   12 WRITE(45,8000)
 8000 FORMAT(1X,45HENTER NAME OF AUTSKED INPUT FILE FROM PREDICT)
      READ(45,8010,END=10,IOSTAT=IEOF)INFIL
 8010 FORMAT(A8)
   10 IF(IEOF.NE.0) THEN
         REWIND 45
         WRITE(45,11)
   11    FORMAT(1H ,38HTHERE IS NO DEFAULT FOR THIS FILE NAME)
         GO TO 12
      ENDIF
   13 WRITE(45,8020)
 8020 FORMAT(1X,33HENTER NAME OF AUTSKED OUTPUT FILE)
      READ(45,8010,END=20,IOSTAT=IEOF)OUTFIL
   20 IF(IEOF.NE.0) THEN
         REWIND 45
         WRITE(45,11)
```

```
         GO TO 13
      ENDIF
      WRITE(45,8030)
 8030 FORMAT(///)
C*********************************************************************
C                              TAPE UNITS
C     TAPE5                INPUT
C     TAPE6                OUTPUT
C     TAPE7                SLAM SCRATCH TAPE
C     TAPE8                DATA STORAGE FOR SLAM LINEPRINTER PLOT
C     TAPE45               INPUT/OUTPUT
C     TAPE46,...,TAPE50    STORAGE FOR CAVERN LEACHED VOLUMES AND CAVERN
C                          OIL VOLUMES
C     TAPE51               STORAGE FOR SCHEDULE
C     TAPE55               INPUT FROM PROGRAM PREDICT
C*********************************************************************
C
      OPEN(5,FILE='NETWRK')
      REWIND 5
      OPEN(6,FILE=OUTFIL)
      REWIND 6
      OPEN(7,STATUS='SCRATCH')
      OPEN(8,STATUS='SCRATCH',FORM='UNFORMATTED')
      OPEN(46,STATUS='SCRATCH')
      OPEN(47,STATUS='SCRATCH')
      OPEN(48,STATUS='SCRATCH')
      OPEN(49,STATUS='SCRATCH')
      OPEN(50,STATUS='SCRATCH')
      OPEN(51)
      REWIND 51
      OPEN(55,FILE=INFIL,STATUS='OLD')
      NNSET=8000
      NCRDR=5
      NPRNT=51
      NTAPE=7
      XX(30)=0.
      XX(31)=0.
      XX(32)=0.
C     DVOL=0.
      DO 30 I=1,21
         ST(I)='    '
         STSV(I)='    '
         OF(I)='    '
         OR(I)=0.
         OLF(I)=0.
   30 CONTINUE
      DO 40 I=1,20
         IS(I)=1
         ISC(I)=1
         IR1(I)=1
         IR2(I)=1
         IR3(I)=1
         CVOL(I)=0.
```

```
      OVOL(I)=0.
      CLK(I)=0.
      DO 51 L=1,60
         RANC(I,L)=0.
         RT(L)=0.
51    CONTINUE
      DO 50 J=1,27
      DO 50 K=1,16
         ATR(I,J,K)=-1.
50    CONTINUE
40 CONTINUE
   ITR=0
   CALL SLAM
   CLOSE(5)
   CLOSE(6)
   CLOSE(7)
   DO 60 I=45,51
      CLOSE(I)
60 CONTINUE
   CLOSE(55)
   OPEN(5,FILE='INPUT')
   OPEN(6,FILE='OUTPUT')
   RETURN
   END
         SUBROUTINE INTLC
   CHARACTER*5 ST(21),STSV(21),OF(21)
   CHARACTER*8 BDAT,EDATE
   CHARACTER*8 OUTFIL,PRDFIL,INFIL
   CHARACTER*1 ITUT,ICH,IDEL,ECH,ICDF,IODF,ICK,ICOV,IFZER
   DIMENSION RANK(20),JPR(20)
   DIMENSION NSET(8000)
   COMMON /FILES/OUTFIL,PRDFIL,INFIL
   COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
  1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
   COMMON/XCOM1/ MFE(100),MLE(100),NQ(100)
   COMMON/SCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
  1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
  2TTSET,XXI(100),TTTS,TTTF
   COMMON/XCOM4/ NXACT(100),NXCNT(100),NXGAT(25),NXRSC(75)
   COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
  *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
   COMMON/UCOM2/ST,STSV,BDAT,OF
   COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),DVOL(20)
  *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
   COMMON/UCOM4/RANC(20,60),RT(60),ITR
   COMMON QSET(8000)
   EQUIVALENCE (NSET(1),QSET(1))
   DATA RANK/1.E4,2.E4,3.E4,4.E4,5.E4,6.E4,7.E4,8.E4,
  *          9.E4,10.E4,11.E4,12.E4,13.E4,14.E4,15.E4,16.E4,
  *         17.E4,18.E4,19.E4,20.E4/
   IER=0
   NPRNT=6
   DO 5 I=1,8
```

```
         BACKSPACE51
    5 CONTINUE
      BDAT(1:8)='01/01/83/'
    1 REWIND55
      READ(55,99999)FACTOR,ISLIP
      READ(55,99994)R21,R22,R31,R32
      FACTOR=FACTOR/100.+1.
      DO 8000 KEY=1,20
         READ (55,99998) (IACC(I,KEY),I=1,12)
         READ (55,99997) (IACC(I,KEY),I=13,25)
         READ (55,99996) (RACC(I,KEY),I=1,8)
         READ (55,99996) (RACC(I,KEY),I=9,16)
         READ (55,99996) (RACC(I,KEY),I=17,24)
         READ (55,99995) (RACC(I,KEY),I=25,31)
 8000 CONTINUE
99999 FORMAT(F6.2,I6)
99998 FORMAT(12I6)
99997 FORMAT(13I6)
99996 FORMAT(8F9.2)
99995 FORMAT(7F9.2)
99994 FORMAT(4F11 6)
      NC=0
  500 DO 50 J=1,40
         ATRIB(J)=-1.
   50 CONTINUE
      ATRIB(30)=0.
      JS=0
      JSC=0
      JR1=0
      JR2=0
      JR3=0
      IFRS=0
      JFRS=0
      READ(55,*,END=102)OILI,VOLUM
      IF(OILI.GT.0) THEN
         BACKSPACE 55
         GO TO 100
      ENDIF
 5000 READ(55,*,END=5800,IOSTAT=IEOF)ISTG,ITEMP,IDAY,FLOW,OIL,NCV
C
C     ISTG=1   SUMP
C         =2   SUMP/CHIMNEY
C         =3   REVERSE 1
C         =4   REVERSE 2-1
C         =5   REVERSE 2-2
C         =6   REVERSE 3-1
C         =7   REVERSE 3-2
C     ITEMP    FIRST DAY OF ACTIVITY
C     IDAY     LAST DAY OF ACTIVITY
C     FLOW     BRINE FLOW RATE
C     OIL      OIL FLOW RATE
C     NCV      NUMBER OF CAVERN (101,102,ETC.)
C
```

```
      IF(ISTG.LE.0) THEN
         BACKSPACE55
         BACKSPACE55
         GO TO 5800
      ELSEIF(ISTG.GT.7) THEN
         BACKSPACE 55
         BACKSPACE 55
         BACKSPACE 55
         GO TO 5800
      ENDIF
      ATRIB(29)=NCV
      KK=NCV-100
      DAYS=IDAY-ITEMP+1
      GO TO (5100,5200,5300,5400,5500,5600,5700),ISTG
 5100 IF(FLOW.GT.-1.AND.IDAY.GT.0) THEN
         JS=JS+1
         ATR(KK,14,JS)=AINT(FLOW*1000.)
         ATR(KK,15,JS)=AINT(OIL*1000.)
         ATR(KK,1,JS)=DAYS
         IF(JS.EQ.1) THEN
            ATRIB(14)=ATR(KK,14,JS)
            ATRIB(15)=ATR(KK,15,JS)
            ATRIB(1)=ATR(KK,1,JS)
            ATR(KK,14,JS)=-1.
            ATR(KK,15,JS)=-1.
            ATR(KK,1,JS)=-1.
         ENDIF
      ELSEIF(ITEMP.NE.0) THEN
         ATRIB(2)=DAYS
      ENDIF
      GO TO 5000
 5200 IF(FLOW.GT.-1.AND.IDAY.GT.0) THEN
         JSC=JSC+1
         ATR(KK,16,JSC)=AINT(FLOW*1000.)
         ATR(KK,17,JSC)=AINT(OIL*1000.)
         ATR(KK,3,JSC)=DAYS
         IF(JSC.EQ.1) THEN
            ATRIB(16)=ATR(KK,16,JSC)
            ATRIB(17)=ATR(KK,17,JSC)
            ATRIB(3)=ATR(KK,3,JSC)
            ATR(KK,16,JSC)=-1.
            ATR(KK,17,JSC)=-1.
            ATR(KK,3,JSC)=-1.
         ENDIF
      ELSEIF(ITEMP.NE.0) THEN
         ATRIB(4)=DAYS
      ENDIF
      GO TO 5000
 5300 IF(FLOW.GT.-1.AND.IDAY.GT.0) THEN
         IF(FLOW.EQ.OIL.AND.FLOW.NE.0) THEN
            ATRIB(20)=AINT(FLOW*1000.)
            ATRIB(7)=DAYS
         ELSE
```

```
            JR1=JR1+1
            ATR(KK,18,JR1)=AINT(FLOW*1000.)
            ATR(KK,19,JR1)=AINT(OIL*1000.)
            ATR(KK,5,JR1)=DAYS
            IF(JR1.EQ.1) THEN
               ATRIB(18)=ATR(KK,18,JR1)
               ATRIB(19)=ATR(KK,19,JR1)
               ATRIB(5)=ATR(KK,5,JR1)
               ATR(KK,18,JR1)=-1.
               ATR(KK,19,JR1)=-1.
               ATR(KK,5,JR1)=-1.
            ENDIF
         ENDIF
      ELSEIF(ITEMP.NE.0) THEN
         ATRIB(6)=DAYS
      ENDIF
      GO TO 5000
5400  IF(FLOW.GT.-1.AND.IDAY.GT.0) THEN
         IF(FLOW.EQ.OIL.AND.FLOW.NE.0) THEN
            ATRIB(23)=AINT(FLOW*1000.)
            ATRIB(10)=DAYS
         ELSE
            JR2=JR2+1
            ATR(KK,21,JR2)=AINT(FLOW*1000.)
            ATR(KK,22,JR2)=AINT(OIL*1000.)
            ATR(KK,8,JR2)=DAYS
            IF(JR2.EQ.1) THEN
               ATRIB(21)=ATR(KK,21,JR2)
               ATRIB(22)=ATR(KK,22,JR2)
               ATRIB(8)=ATR(KK,8,JR2)
               ATR(KK,21,JR2)=-1.
               ATR(KK,22,JR2)=-1.
               ATR(KK,8,JR2)=-1.
            ENDIF
         ENDIF
      ELSEIF(ITEMP.NE.0) THEN
         ATRIB(9)=DAYS
      ENDIF
      GO TO 5000
5500  IF(FLOW.GT.-1.AND.IDAY.GT.0) THEN
         IF(FLOW.EQ.OIL.AND.FLOW.NE.0) THEN
            ATRIB(23)=AINT(FLOW*1000.)
            ATRIB(10)=DAYS
         ELSE
            JR2=JR2+1
            ATR(KK,21,JR2)=AINT(FLOW*1000.)
            ATR(KK,22,JR2)=AINT(OIL*1000.)
            ATR(KK,8,JR2)=DAYS
            IFRS=IFRS+1
            IF(IFRS.EQ.1) THEN
               IF(JR2.NE.1) THEN
                  ATRIB(32)=ATR(KK,21,JR2)
                  ATRIB(33)=ATR(KK,22,JR2)
```

```
                ATRIB(34)=ATR(KK,8,JR2)
                ATR(KK,21,JR2)=-1.
                ATR(KK,22,JR2)=-1.
                ATR(KK,8,JR2)=-1.
             ELSE
                ATRIB(21)=ATR(KK,21,JR2)
                ATRIB(22)=ATR(KK,22,JR2)
                ATRIB(8)=ATR(KK,8,JR2)
                ATR(KK,21,JR2)=-1.
                ATR(KK,22,JR2)=-1.
                ATR(KK,8,JR2)=-1.
             ENDIF
          ENDIF
       ENDIF
    ELSEIF(ITEMP.NE.0.AND.JR2.NE.1) THEN
       ATRIB(35)=DAYS
    ELSEIF(ITEMP.NE.0) THEN
       ATRIB(9)=DAYS
    ENDIF
    GO TO 5000
5600 IF(FLOW.GT.-1.AND.IDAY.GT.0) THEN
       IF(FLOW.EQ.OIL.AND.FLOW.NE.0) THEN
          ATRIB(26)=AINT(FLOW*1000.)
          ATRIB(13)=DAYS
       ELSE
          JR3=JR3+1
          ATR(KK,24,JR3)=AINT(FLOW*1000.)
          ATR(KK,25,JR3)=AINT(OIL*1000.)
          ATR(KK,11,JR3)=DAYS
          IF(JR3.EQ.1) THEN
             ATRIB(24)=ATR(KK,24,JR3)
             ATRIB(25)=ATR(KK,25,JR3)
             ATRIB(11)=ATR(KK,11,JR3)
             ATR(KK,24,JR3)=-1.
             ATR(KK,25,JR3)=-1.
             ATR(KK,11,JR3)=-1.
          ENDIF
       ENDIF
    ELSEIF(ITEMP.NE.0) THEN
       ATRIB(12)=DAYS
    ENDIF
    GO TO 5000
5700 IF(FLOW.GT.-1.AND.IDAY.GT.0) THEN
       IF(FLOW.EQ.OIL.AND.FLOW.NE.0) THEN
          ATRIB(26)=AINT(FLOW*1000.)
          ATRIB(13)=DAYS
          GO TO 5800
       ELSE
          JR3=JR3+1
          ATR(KK,24,JR3)=AINT(FLOW*1000.)
          ATR(KK,25,JR3)=AINT(OIL*1000.)
          ATR(KK,11,JR3)=DAYS
          JFRS=JFRS+1
```

```
            IF(JFRS.EQ.1) THEN
               IF(JR3.NE.1) THEN
                  ATRIB(36)=ATR(KK,24,JR3)
                  ATRIB(37)=ATR(KK,25,JR3)
                  ATRIB(38)=ATR(KK,11,JR3)
                  ATR(KK,24,JR3)=-1.
                  ATR(KK,25,JR3)=-1.
                  ATR(KK,11,JR3)=-1.
               ELSE
                  ATRIB(24)=ATR(KK,24,JR3)
                  ATRIB(25)=ATR(KK,25,JR3)
                  ATRIB(11)=ATR(KK,11,JR3)
                  ATR(KK,24,JR3)=-1.
                  ATR(KK,25,JR3)=-1.
                  ATR(KK,11,JR3)=-1.
               ENDIF
            ENDIF
         ENDIF
      ELSEIF(ITEMP.NE.0.AND.JR3.NE.1) THEN
         ATRIB(39)=DAYS
      ELSEIF(ITEMP.NE.0) THEN
         ATRIB(12)=DAYS
      ENDIF
      GO TO 5000
 5800 NC=NC+1
      NCV=ATRIB(29)
      KK=NCV-100
      NUMCAV(NC)=ATRIB(29)
C
C   INITIALIZE CAVERN VOLUME, OIL VOLUME
C
      SS(NC)=RACC(1,KK)
      SS(NC+40)=RACC(2,KK)
      ATRIB(30)=0.
      ATRIB(27)=0.
C
C   IFRS=JR2 MEANS NO R2-1 LEACHING FOR THIS CAVERN
C   JFRS=JR3 MEANS NO R3-1 LEACHING FOR THIS CAVERN
C   ATRIB(32) AND ATRIB(36) ARE USED AS FLAGS
C   TO DENOTE THESE FACTS
C
      IF(IFRS.EQ.JR2) ATRIB(32)=-2.
      IF(JFRS.EQ.JR3) ATRIB(36)=-2.
  101 CALL FILEM(7,ATRIB)
      IF(ISTG.GT.7) GO TO 100
      IF(IEOF.EQ.0) GO TO 500
      GO TO 102
  100 CONTINUE
12010 ITR=ITR+1
      DO 12000 I=1,NC
         READ(55,*,END=12999)RANC(I,ITR),RT(ITR)
12000 CONTINUE
      GO TO 12010
```

```
12999 ITR=ITR-1
  102 IF(ITR.NE.0) IPR=2
      WRITE(45,3000)
 3000 FORMAT(///,25X,27H*****  A U T S K E D  *****//)
  200 WRITE(45,2000)
 2000 FORMAT(1X,37HENTER MAX BRINE FLOW FOR SITE(MB/DAY))
      READ(45,*,END=2001)XX(1)
      XX(1)=XX(1)*1000.
      GO TO 2009
 2001 REWIND 45
      GO TO 200
 2009 WRITE(45,2010)
 2010 FORMAT(1X,35HENTER MAX OIL FLOW FOR SITE(MB/DAY))
      READ(45,*,END=2011)XX(2)
      XX(2)=XX(2)*1000.
      GO TO 320
 2011 REWIND 45
      GO TO 2009
  320 WRITE(45,2020)
 2020 FORMAT(1X,50HENTER MAX BRINE FLOW (DEFAULT) FOR CAVERNS(MB/DAY)
     *       /5X,21H(MUST BE AT LEAST 40))
      READ(45,*,END=2021)XX(3)
      IF(XX(3).LT.40) GO TO 2021
      XX(3)=XX(3)*1000.
 2027 WRITE(45,2022)
 2022 FORMAT(1X,53HUSE DEFAULT BRINE FLOW FOR ALL CAVERNS?(Y/N,DEFAULT=Y
     *         ,1H))
      ICDF='Y'
      READ(45,3010,END=2025)ICDF
      IF(ICDF.NE.'Y'.AND.ICDF.NE.'N') THEN
         WRITE(45,2026)
 2026    FORMAT(1X,24HANSWER  Y  OR  N  PLEASE)
         GO TO 2027
      ENDIF
 2025 REWIND 45
      GO TO 321
 2021 REWIND 45
      GO TO 320
  321 IF(XX(3).GT.XX(1)) THEN
         IJI=XX(1)/1000.
         WRITE(45,6010)IJI
 6010    FORMAT(5X,39HMAX CAVERN BRINE FLOW SHOULD NOT EXCEED/
     *          5X,20HMAX SITE BRINE FLOW(,I4,13H)---TRY AGAIN)
         GO TO 320
      ENDIF
  330 WRITE(45,2030)
 2030 FORMAT(1X,48HENTER MAX OIL FLOW (DEFAULT) FOR CAVERNS(MB/DAY))
      READ(45,*,END=2031)XX(4)
      XX(4)=XX(4)*1000.
 2037 WRITE(45,2032)
 2032 FORMAT(1X,52HUSE DEFAULT OIL FLOW FOR ALL CAVERNS?(Y/N,DEFAULT=Y))
      IODF='Y'
      READ(45,3010,END=2035)IODF
```

```
      IF(IODF.NE.'Y'.AND.IODF.NE.'N') THEN
         WRITE(45,2036)
2036     FORMAT(1X,24HANSWER  Y  OR  N  PLEASE)
         GO TO 2037
      ENDIF
2035  REWIND 45
      GO TO 331
2031  REWIND 45
      GO TO 330
 331  IF(XX(4).GT.XX(2)) THEN
         IJI=XX(2)/1000.
         WRITE(45,6020)IJI
6020     FORMAT(5X,37HMAX CAVERN OIL FLOW SHOULD NOT EXCEED/
     *          5X,18HMAX SITE OIL FLOW(,I4,14H)--- TRY AGAIN)
         GO TO 330
      ENDIF
2049  WRITE(45,2050)
2050  FORMAT(1X,29HENTER NUMBER OF WORKOVER RIGS)
      READ(45,*,END=2051)IWORK
      XX(5)=IWORK
      XX(6)=XX(5)+1
      GO TO 340
2051  REWIND 45
      GO TO 2049
 340  IF(ITR.NE.0) GO TO 9001
      WRITE(45,2040)
2040  FORMAT(1X,15HPRIORITY CHOICE/
     *      10X,26H1. SHORTEST TIME REMAINING/
     *      10X,22H2. USER INPUT PRIORITY/
     *      10X,26H3. CAVERN NUMBER MINUS 100//
     *       1X,22HENTER NUMBER OF CHOICE)
      READ(45,*,END=2041)IPR
      GO TO 341
2041  REWIND 45
      GO TO 340
 341  IF(IPR.NE.1.AND.IPR.NE.2.AND.IPR.NE.3) THEN
         WRITE(45,6000)
         GO TO 340
      ENDIF
9001  WRITE(45,9000)
9000  FORMAT(1X,52HDO YOU WANT A CLOCK FOR EACH CAVERN?(Y/N, DEFAULT=N))
      ICK='N'
8011  READ(45,3010,END=8010)ICK
8010  REWIND 45
      IF(ICK.NE.'Y'.AND.ICK.NE.'N') THEN
         WRITE(45,8020)
8020     FORMAT(1X,24HPLEASE ANSWER  Y  OR  N )
         ICK='N'
         GO TO 8011
      ENDIF
      IF(ICK.EQ.'Y') THEN
         CLOK=1.
      ELSE
```

```
         CLOK=0.
      ENDIF
      WRITE(45,11000)
11000 FORMAT(1X,50HDO YOU WANT A CAVERN/OIL VOLUME TABULATION AT EACH/
     *          28H BREAKPOINT?(Y/N, DEFAULT=N))
      ICOV='N'
11011 READ(45,3010,END=11010)ICOV
11010 REWIND 45
      IF(ICOV.NE.'Y'.AND.ICOV.NE.'N') THEN
         WRITE(45,11020)
11020    FORMAT(1X,24HPLEASE ANSWER  Y  OR  N )
         ICOV='N'
         GO TO 11011
      ENDIF
      IF(ICOV.EQ.'Y') THEN
         COVT=1.
      ELSE
         COVT=0.
      ENDIF
      WRITE(45,13000)
13000 FORMAT(1X,45HDO YOU WANT ZERO FLOW RATES MAINTAINED (Y) OR/
     *          45H (N) DEPENDENT ON MAX SITE FLOW UTILIZATION ?
     *         ,16H(Y/N, DEFAULT=Y))
      IFZER='Y'
13011 READ(45,3010,END=13010)IFZER
13010 REWIND 45
      IF(IFZER.NE.'Y'.AND.IFZER.NE.'N') THEN
         WRITE(45,13020)
13020    FORMAT(1X,24HPLEASE ANSWER  Y  OR  N )
         IFZER='Y'
         GO TO 13011
      ENDIF
      IF(IFZER.EQ.'Y') THEN
         FZER=1.
      ELSE
         FZER=0.
      ENDIF
  300 WRITE(45,2060)
 2060 FORMAT(1X,28HENTER START DATE FOR PROJECT/
     *       1X,23HIN THE FORMAT  MM/DD/YY/
     *       1X,18HEXAMPLE:  04/08/83)
      READ(45,2070,END=2061)BDAT
      GO TO 351
 2061 REWIND 45
      GO TO 300
  351 CALL CHKDAT(BDAT,ERR)
      IF(ERR.EQ.1) THEN
         WRITE(45,6000)
 6000    FORMAT(1X,21HBAD INPUT---TRY AGAIN)
         GO TO 300
      ENDIF
      IF(IPR.EQ.1) THEN
         XX(10)=0
```

```
      GO TO 601
    ENDIF
 400 WRITE(45,4000)
4000 FORMAT(1X,18HSCHEDULING OPTIONS/
   *       6X,38H1. DEVELOP SCHEDULE TO SITE COMPLETION/
   *       6X,39H2. STOP SCHEDULE AFTER A NUMBER OF DAYS/
   *       6X,33H     AND REPRIORITIZE THE CAVERNS//
   *       1X,22HENTER NUMBER OF CHOICE)
    READ(45,4010,END=4011)ISCH
    GO TO 401
4011 REWIND 45
    GO TO 400
4010 FORMAT(I1)
 401 IF(ISCH.NE.1.AND.ISCH.NE.2) THEN
      WRITE(45,6000)
      GO TO 400
    ENDIF
    IF(ISCH.EQ.2) THEN
 410    WRITE(45,4020)
4020    FORMAT(1X,24HENTER THE NUMBER OF DAYS)
        READ(45,4030,END=4031)IDAYS
        GO TO 411
4031    REWIND 45
        GO TO 410
4030    FORMAT(I4)
 411    IF(ITR.NE.0) THEN
           IDYS=RT(ITR)
        ELSE
           IDYS=0
        ENDIF
        IF(IDAYS.LE.IDYS) THEN
           WRITE(45,4040)IDYS
4040       FORMAT(1X,36HNUMBER OF DAYS MUST BE GREATER THAN ,I4,3H --/
   *             1X,9HTRY AGAIN)
           GO TO 410
        ENDIF
        XX(10)=IDAYS
    ELSE
        XX(10)=0
    ENDIF
 601 CONTINUE
2070 FORMAT(A8)
3010 FORMAT(A1)
    ICVMX=XX(3)/1000.
    IOVMX=XX(4)/1000.
    IF(ITR.EQ.0.AND.IPR.NE.1)ITR=ITR+1
    DO 7000 I=1,NC
       KK=NUMCAV(I)
       KKK=KK-100
       CVMAX(I)=XX(3)
       COMAX(I)=XX(4)
       X=KK
       JJ=NFIND(1,NCLNR,29,0,X,0)
```

```
      CALL RMOVE(JJ,NCLNR,ATRIB)
      IF(IPR.EQ.1) THEN
         ATRIB(28)=0.
         DO 7001 MM=1,13
            ATRIB(28)=ATRIB(28)+AMAX1(0.,ATRIB(MM))
            DO 7002 NN=1,16
               ATRIB(28)=ATRIB(28)+AMAX1(0.,ATR(KKK,MM,NN))
7002        CONTINUE
7001     CONTINUE
         ATRIB(28)=ATRIB(28)+AMAX1(0.,ATRIB(34))
         ATRIB(28)=ATRIB(28)+AMAX1(0.,ATRIB(35))
         ATRIB(28)=ATRIB(28)+AMAX1(0.,ATRIB(38))
         ATRIB(28)=ATRIB(28)+AMAX1(0.,ATRIB(39))
         IF(ICDF.EQ.'Y') GO TO 7711
7601     WRITE(45,7600)KK,ICVMX
7600     FORMAT(1X,41HENTER MAXIMUM BRINE FLOW RATE FOR CAVERN ,I3,
     *            9H (MB/DAY)/
     *         5X,10H(DEFAULT= ,I3,22H, MUST BE AT LEAST 40))
         READ(45,*,END=7610,IOSTAT=IEOF)CVMAX(I)
7610     IF(IEOF.NE.0) THEN
            REWIND 45
         ELSEIF(CVMAX(I).LT.40) THEN
            GO TO 7601
         ELSE
            CVMAX(I)=CVMAX(I)*1000.
         ENDIF
7711     IF(IODF.EQ.'Y') GO TO 7611
         WRITE(45,7700)KK,IOVMX
7700     FORMAT(1X,39HENTER MAXIMUM OIL FLOW RATE FOR CAVERN ,I3,
     *            9H (MB/DAY)/
     *         5X,10H(DEFAULT= ,I3,22H, MUST BE AT LEAST 40))
         READ(45,*,END=7710,IOSTAT=IEOF)COMAX(I)
7710     IF(IEOF.NE.0) THEN
            REWIND 45
         ELSE
            COMAX(I)=COMAX(I)*1000.
         ENDIF
7611     CALL FILEM(7,ATRIB)
      ELSEIF(IPR.EQ.2)THEN
         IF(ITR.GT.1) GO TO 7515
         IF(I.EQ.1) THEN
            WRITE(45,7500)
7500        FORMAT(1X,25HENTER CAVERN PRIORITIES--/
     +            1X,23H1 HIGHEST, 2 NEXT, ETC./)
         ENDIF
7004     WRITE(45,7510)KK
7510     FORMAT(1X,26HENTER PRIORITY FOR CAVERN ,I3)
7005     READ(45,*,END=7006)JPR(KKK)
         GO TO 7007
7006     REWIND 45
         GO TO 7004
7007     INV=0
         DO 7010 JJ=1,NC
```

```
            IF(JPR(KKK).EQ.JJ) THEN
               INV=1
            ENDIF
            IF(JJ.LT.I) THEN
               JJJ=NUMCAV(JJ)-100
               IF(JPR(JJJ).EQ.JPR(KKK)) THEN
                  WRITE(45,7520)JPR(KKK)
 7520             FORMAT(1X,14HYOU HAVE USED ,I2,9H ALREADY-
     +                    ,11H--TRY AGAIN)
                  GO TO 7004
               ENDIF
            ENDIF
7010     CONTINUE
         IF(INV.EQ.0) THEN
            L=1
            WRITE(45,7530)L,NC
7530        FORMAT(1X,32HPRIORITY SHOULD BE IN THE RANGE ,I2,
     +                 4H TO ,I2,11H--TRY AGAIN)
            GO TO 7005
         ENDIF
 7515    IF(ITR.EQ.1) THEN
            ATRIB(28)=RANK(JPR(KKK))
            IF(ISCH.EQ.2) THEN
               RANC(I,ITR)=ATRIB(28)
               RT(ITR)=TNOW
               WRITE(55,*)RANC(I,ITR),RT(ITR)
            ENDIF
         ELSE
            ATRIB(28)=RANC(I,1)
            JPR(KKK)=INT(ATRIB(28)/10000.)
         ENDIF
         IF(ICDF.EQ.'Y') GO TO 7540
 7513    WRITE(45,7511)KK,ICVMX
 7511    FORMAT(1X,41HENTER MAXIMUM BRINE FLOW RATE FOR CAVERN ,I3,
     *            9H (MB/DAY)/
     *         5X,10H(DEFAULT= ,I3,22H, MUST BE AT LEAST 40))
         READ(45,*,END=7512,IOSTAT=IEOF)CVMAX(I)
 7512    IF(IEOF.NE.0) THEN
            REWIND 45
         ELSEIF(CVMAX(I).LT.40) THEN
            GO TO 7513
         ELSE
            CVMAX(I)=CVMAX(I)*1000.
         ENDIF
 7540    IF(IODF.EQ.'Y') GO TO 7514
         WRITE(45,7541)KK,IOVMX
 7541    FORMAT(1X,39HENTER MAXIMUM OIL FLOW RATE FOR CAVERN ,I3,
     *            9H (MB/DAY)/
     *         5X,10H(DEFAULT= ,I3,1H))
         READ(45,*,END=7542,IOSTAT=IEOF)COMAX(I)
 7542    IF(IEOF.NE.0) THEN
            REWIND 45
         ELSE
```

```
            COMAX(I)=COMAX(I)*1000.
         ENDIF
7514     CALL FILEM(7,ATRIB)
      ELSE
         ATRIB(28)=RANK(KKK)
         IF(ISCH.EQ.2) THEN
            RANC(I,ITR)=ATRIB(28)
            RT(ITR)=TNOW
            WRITE(55,*)RANC(I,ITR),RT(ITR)
         ENDIF
         JPR(KKK)=KKK
         IF(ICDF.EQ.'Y') GO TO 7840
7830     WRITE(45,7800)KK,ICVMX
7800     FORMAT(1X,41HENTER MAXIMUM BRINE FLOW RATE FOR CAVERN ,I3,
    *                9H (MB/DAY)/
    *             5X,10H(DEFAULT= ,I3,22H, MUST BE AT LEAST 40))
         READ(45,*,END=7810,IOSTAT=IEOF)CVMAX(I)
7810     IF(IEOF.NE.0) THEN
            REWIND 45
         ELSEIF(CVMAX(I).LT.40)THEN
            GO TO 7830
         ELSE
            CVMAX(I)=CVMAX(I)*1000.
         ENDIF
7840     IF(IODF.EQ.'Y') GO TO 7811
         WRITE(45,7841)KK,IOVMX
7841     FORMAT(1X,39HENTER MAXIMUM OIL FLOW RATE FOR CAVERN ,I3,
    *                9H (MB/DAY)/
    *             5X,10H(DEFAULT= ,I3,1H))
         READ(45,*,END=7842,IOSTAT=IEOF)COMAX(I)
7842     IF(IEOF.NE.0) THEN
            REWIND 45
         ELSE
            COMAX(I)=COMAX(I)*1000.
         ENDIF
7811     CALL FILEM(7,ATRIB)
      ENDIF
7000 CONTINUE
     NCAV=0
6049 WRITE(45,6050)
6050 FORMAT(5X,36HWOULD YOU LIKE AN ECHO OF YOUR INPUT/
    *       5X,48HAND THE OPTION TO MAKE CHANGES? (Y/N, DEFAULT=N))
     ECH='N'
6053 READ(45,3010,END=6051)ECH
6051 REWIND 45
     IF(ECH.NE.'Y'.AND.ECH.NE.'N') THEN
        WRITE(45,6052)
6052    FORMAT(1X,23HPLEASE ANSWER  Y  OR  N)
        ECH='N'
        GO TO 6053
     ENDIF
     IF(ECH.EQ.'Y') THEN
        IQT=0
```

```
      CALL ECHO(IWORK,IPR,IDEL,JPR,IQT,ICDF,IODF)
      IF(IQT.EQ.1) STOP
   ENDIF
   DO 20 I=1,40
      ATRIB(I)=-1.
 20 CONTINUE
   ATRIB(27)=1.
   CALL FILEM(7,ATRIB)
   WRITE(45,2080)
2080 FORMAT(1X,/31HSCHEDULING PROGRAM IS EXECUTING)
   ENDFILE45
   RETURN
   END
        SUBROUTINE ECHO(IWORK,IPR,IDEL,JPR,IQT,ICDF,IODF)
   CHARACTER*5 ST(21),STSV(21),OF(21)
   CHARACTER*8 BDAT,EDATE,DAT
   CHARACTER*1 ITUT,ICH,IDEL,ECH,ICDF,IODF,ICK,ICOV,IFZER
   CHARACTER*3 PRI
   DIMENSION RANK(20),JPR(20),ICAV(20)
   DIMENSION NSET(8000)
   COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
  1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
   COMMON/XCOM1/ MFE(100),MLE(100),NQ(100)
   COMMON/GCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
  1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
  2TTSET,XXI(100),TTTS,TTTF
   COMMON/XCOM4/ NXACT(100),NXCNT(100),NXGAT(25),NXRSC(75)
   COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
  *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
   COMMON/UCOM2/ST,STSV,BDAT,OF
   COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),OVOL(20)
  *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
   COMMON/UCOM4/RANC(20,60),RT(60),ITR
   COMMON QSET(8000)
   EQUIVALENCE (NSET(1),QSET(1))
   DATA RANK/1.E4,2.E4,3.E4,4.E4,5.E4,6.E4,7.E4,8.E4,
  *9.E4,10.E4,11.E4,12.E4,13.E4,14.E4,15.E4,16.E4,
  *17.E4,18.E4,19.E4,20.E4/
2005 I1=XX(1)/1000.
   I2=XX(2)/1000.
   I3=XX(3)/1000.
   I4=XX(4)/1000.
   I5=XX(5)
   WRITE(45,2000)
2000 FORMAT(//15X,19H** ECHO OF INPUT **//)
   WRITE(45,2010)I1
2010 FORMAT(5X,27HMAXIMUM BRINE FLOW FOR SITE,T45,I4,7H MB/DAY)
   WRITE(45,2020)I2
2020 FORMAT(5X,25HMAXIMUM OIL FLOW FOR SITE,T45,I4,7H MB/DAY)
   WRITE(45,2030)I3
2030 FORMAT(5X,35HMAX BRINE FLOW(DEFAULT) FOR CAVERNS,T45,I4,7H MB/DAY)
   WRITE(45,2040)I4
2040 FORMAT(5X,33HMAX OIL FLOW(DEFAULT) FOR CAVERNS,T45,I4,7H MB/DAY)
```

```
      WRITE(45,2050)I5
 2050 FORMAT(5X,23HNUMBER OF WORKOVER RIGS,T45,I4)
      WRITE(45,2060)BDAT
 2060 FORMAT(5X,18HPROJECT START DATE,T46,A8)
      IF(CLOK.EQ.1) THEN
         WRITE(45,2061)
 2061    FORMAT(5X,19HCAVERN CLOCK OPTION,T48,3HYES)
      ELSE
         WRITE(45,2062)
 2062    FORMAT(5X,19HCAVERN CLOCK OPTION,T48,2HNO)
      ENDIF
      IF(COVT.EQ.1) THEN
         WRITE(45,2161)
 2161    FORMAT(5X,28HCAVERN/OIL VOLUME TABULATION,T48,3HYES)
      ELSE
         WRITE(45,2162)
 2162    FORMAT(5X,28HCAVERN/OIL VOLUME TABULATION,T48,2HNO)
      ENDIF
      IF(FZER.EQ.1) THEN
         WRITE(45,2261)
 2261    FORMAT(5X,23HFORCED ZERO FLOW OPTION,T48,3HYES//)
      ELSE
         WRITE(45,2262)
 2262    FORMAT(5X,23HFORCED ZERO FLOW OPTION,T48,2HNO//)
      ENDIF
      PAUSE 'PRESS RETURN TO CONTINUE'
      WRITE(45,2071)
 2071 FORMAT(21X,37HMAX FLOW RATES AND INITIAL PRIORITIES)
      WRITE(45,2070)
 2070 FORMAT(//5X,13HCAVERN NUMBER,T25,8HPRIORITY,T40,14HMAX BRINE FLOW,
     *                            T60,12HMAX OIL FLOW)
      DO 100 I=1,NC
         NUM=NUMCAV(I)
         J=NUM-100
         JBR=CVMAX(I)/1000.
         JOL=COMAX(I)/1000.
         PRI='STR'
         IF(IPR.EQ.1) THEN
            WRITE(45,2080)NUM,PRI,JBR,JOL
 2080       FORMAT(5X,I3,T25,A3,T43,I3,T63,I3)
         ELSE
            K=JPR(J)
            WRITE(45,2090)NUM,K,JBR,JOL
 2090       FORMAT(5X,I3,T25,I2,T43,I3,T63,I3)
         ENDIF
  100 CONTINUE
      WRITE(45,2099)
 2099 FORMAT(//)
      PAUSE 'PRESS RETURN TO CONTINUE'
      IF(MCH.EQ.15) GO TO 16000
 2095 WRITE(45,3000)
 3000 FORMAT(//5X,27HMAKE A CHOICE FROM THE MENU//)
 2096 WRITE(45,3010)
```

```
 3010 FORMAT(1X,T20,12H*** MENU ***//
     +      10X,29H1. CHANGE MAX SITE BRINE FLOW/
     +      10X,27H2. CHANGE MAX SITE OIL FLOW/
     +      10X,45H3. CHANGE MAX BRINE FLOW(DEFAULT) FOR CAVERNS/
     +      10X,43H4. CHANGE MAX OIL FLOW(DEFAULT) FOR CAVERNS/
     +      10X,37H5. CHANGE MAX BRINE FLOW FOR A CAVERN/
     +      10X,35H6. CHANGE MAX OIL FLOW FOR A CAVERN/
     +      10X,33H7. CHANGE NUMBER OF WORKOVER RIGS/
     +      10X,29H8. CHANGE CAVERN CLOCK OPTION/
     +      10X,38H9. CHANGE CAVERN/OIL VOLUME TABULATION/
     +       9X,34H10. CHANGE FORCED ZERO FLOW OPTION/
     +       9X,29H11. CHANGE PROJECT START DATE/
     +       9X,37H12. CHANGE PRIORITY TYPE(STR OR USER)/
     +       9X,31H13. CHANGE CAVERN USER PRIORITY/
     +       9X,41H14. DISPLAY CURRENT CAVERN PRIORITY TABLE/
     +       9X,23H15. DO INPUT ECHO AGAIN/
     +       9X,29H16. EXIT TO PROGRAM EXECUTION/
     +       9X,22H17. TERMINATE THIS RUN)
  150 READ(45,*,END=151)MCH
      GO TO 99999
  151 REWIND 45
      GO TO 2095
99999 IF(MCH.EQ.0) GO TO 2096
      GO TO(5000,6000,7000,8000,18000,19000,9000,20000,22000,23000,
     +      10000,12000,13000,21000,14000,15000,17000),MCH
      WRITE(45,3020)
 3020 FORMAT(1X,25HINVALID CHOICE--TRY AGAIN)
      GO TO 150
 5000 WRITE(45,4000)
 4000 FORMAT(1X,37HENTER MAX BRINE FLOW FOR SITE(MB/DAY))
      READ(45,*,END=4001)XX(1)
      XX(1)=XX(1)*1000.
      GO TO 4003
 4001 REWIND 45
      GO TO 5000
 4003 IF(XX(1).LT.XX(3)) THEN
         WRITE(45,4002)
 4002    FORMAT(1X,43HMAX BRINE FLOW(DEFAULT) FOR CAVERNS EXCEEDS,/
     *          1X,44HNEW MAX BRINE FLOW FOR SITE---PLEASE CORRECT//)
      ENDIF
      GO TO 16000
 6000 WRITE(45,4010)
 4010 FORMAT(1X,35HENTER MAX OIL FLOW FOR SITE(MB/DAY))
      READ(45,*,END=4012)XX(2)
      GO TO 4013
 4012 REWIND 45
      GO TO 6000
 4013 XX(2)=XX(2)*1000.
      IF(XX(2).LT.XX(4)) THEN
         WRITE(45,4011)
 4011    FORMAT(1X,41HMAX OIL FLOW(DEFAULT) FOR CAVERNS EXCEEDS,/
     *          1X,42HNEW MAX OIL FLOW FOR SITE---PLEASE CORRECT//)
      ENDIF
```

```
      GO TO 16000
 7000 WRITE(45,4020)
 4020 FORMAT(1X,49HENTER MAX BRINE FLOW(DEFAULT) FOR CAVERNS(MB/DAY))
      READ(45,*,END=4022)XX(3)
      GO TO 4023
 4022 REWIND 45
      GO TO 7000
 4023 XX(3)=XX(3)*1000.
      IF(XX(3).GT.XX(1)) THEN
         IJI=XX(1)/1000.
         WRITE(45,4021)IJI
 4021    FORMAT(1X,46HMAX BRINE FLOW(DEFAULT) FOR CAVERNS SHOULD NOT/
     +            27HEXCEED MAX SITE BRINE FLOW(,I4,12H)--TRY AGAIN)
          GO TO 7000
      ENDIF
      IF(ICDF.EQ.'Y') THEN
         DO 4024 IJI=1,NC
            CVMAX(IJI)=XX(3)
 4024   CONTINUE
      ENDIF
      GO TO 16000
 8000 WRITE(45,4030)
 4030 FORMAT(1X,47HENTER MAX OIL FLOW(DEFAULT) FOR CAVERNS(MB/DAY))
      READ(45,*,END=4032)XX(4)
      GO TO 4033
 4032 REWIND 45
      GO TO 8000
 4033 XX(4)=XX(4)*1000.
      IF(XX(4).GT.XX(2)) THEN
         IJI=XX(2)/1000.
         WRITE(45,4031)IJI
 4031    FORMAT(1X,44HMAX OIL FLOW(DEFAULT) FOR CAVERNS SHOULD NOT/
     +            25HEXCEED MAX SITE OIL FLOW(,I4,12H)--TRY AGAIN)
         GO TO 8000
      ENDIF
      IF(IODF.EQ.'Y') THEN
         DO 4034 IJI=1,NC
            COMAX(IJI)=XX(4)
 4034   CONTINUE
      ENDIF
      GO TO 16000
 9000 WRITE(45,9010)
 9010 FORMAT(1X,29HENTER NUMBER OF WORKOVER RIGS)
      READ(45,*,END=9011)IWORK
      GO TO 9012
 9011 REWIND 45
      GO TO 9000
 9012 XX(5)=IWORK
      XX(6)=XX(5)+1
      GO TO 16000
10000 WRITE(45,10010)
10010 FORMAT(1X,28HENTER START DATE FOR PROJECT)
      READ(45,10020,END=10011)BDAT
```

```
      GO TO 10012
10011 REWIND 45
      GO TO 10000
10012 CALL CHKDAT(BDAT,ERR)
      IF(ERR.EQ.1) THEN
         WRITE(45,10022)
10022    FORMAT(21HBAD INPUT---TRY AGAIN)
         GO TO 10000
      ENDIF
10020 FORMAT(A8)
10030 GO TO 16000
12000 IF(ITR.GE.1) THEN
         WRITE(45,12900)
12900    FORMAT(1X,20HOPTION NOT AVAILABLE)
         GO TO 16000
      ENDIF
      WRITE(45,12010)
12010 FORMAT(1X,15HPRIORITY CHOICE/10X,26H1. SHORTEST TIME REMAINING/
     +      10X,22H2. USER INPUT PRIORITY//1X,22HENTER NUMBER OF CHOICE)
      READ(45,*,END=12011)IPRCH
      GO TO 12013
12011 REWIND 45
      GO TO 12000
12013 IF(IPRCH.NE.1.AND.IPRCH.NE.2) THEN
         WRITE(45,12020)
12020    FORMAT(1X,21HBAD INPUT---TRY AGAIN)
         GO TO 12000
      ENDIF
      IF(IPR.EQ.IPRCH) THEN
12029    WRITE(45,12030)
12030    FORMAT(1X,34HPRIORITY TYPE REQUESTED IS SAME AS/
     +         1X,50HCURRENT SELECTION. DO YOU WANT TO TRY AGAIN? (Y/N))
         READ(45,11060,END=12012)ICH
         GO TO 12014
11060    FORMAT(A1)
12012    REWIND 45
         GO TO 12029
12014    IF(ICH.EQ.'Y') THEN
            GO TO 12000
         ELSE
            GO TO 16000
         ENDIF
      ENDIF
      IF(IPRCH.EQ.1) THEN
         GO TO 12500
      ELSE
         GO TO 12700
      ENDIF
12500 DO 12510 I=1,NC
         XNCV=NUMCAV(I)
         IJI=NFIND(1,NCLNR,29,0,XNCV,0)
         CALL RMOVE(IJI,NCLNR,ATRIB)
         KK=XNCV-100.
```

```
         ATRIB(28)=0.
         DO 12520 KKK=1,13
            ATRIB(28)=ATRIB(28)+AMAX1(0.,ATRIB(KKK))
            DO 12521 JJJ=1,16
               ATRIB(28)=ATRIB(28)+AMAX1(0.,ATR(KK,KKK,JJJ))
12521       CONTINUE
12520    CONTINUE
         ATRIB(28)=ATRIB(28)+AMAX1(0.,ATRIB(34))
         ATRIB(28)=ATRIB(28)+AMAX1(0.,ATRIB(35))
         ATRIB(28)=ATRIB(28)+AMAX1(0.,ATRIB(38))
         ATRIB(28)=ATRIB(28)+AMAX1(0.,ATRIB(39))
         CALL FILEM(7,ATRIB)
12510 CONTINUE
      IPR=IPRCH
      GO TO 16000
12700 DO 12710 I=1,NC
         XNCV=NUMCAV(I)
         KK=XNCV-100
         NCV=XNCV
         IJI=NFIND(1,NCLNR,29,0,XNCV,0)
         CALL RMOVE(IJI,NCLNR,ATRIB)
12719    WRITE(45,12720)NCV
12720    FORMAT(1X,25HENTER PRIORITY FOR CAVERN,I4,
     +            25H, 1 HIGHEST, 2 NEXT, ETC.)
         READ(45,*,END=12721)JPR(KK)
         GO TO 12722
12721    REWIND 45
         GO TO 12719
12722    ATRIB(28)=RANK(JPR(KK))
         CALL FILEM(7,ATRIB)
12710 CONTINUE
      IPR=IPRCH
      GO TO 16000
13000 IF(ITR.GT.1) THEN
         WRITE(45,13900)
13900    FORMAT(1X,20HOPTION NOT AVAILABLE)
         GO TO 16000
      ELSEIF(ITR.EQ.1) THEN
         GO TO 13005
      ENDIF
      IF(IPR.EQ.1) THEN
         WRITE(45,13010)
13010    FORMAT(1X,43HCURRENT PRIORITY TYPE IS STR. USE OPTION 12/
     +          1X,24HTO CHANGE PRIORITY TYPE.)
         GO TO 16000
      ENDIF
13005 WRITE(45,13020)
13020 FORMAT(1X,10HOPTIONS---//
     +       1X,37H   1. REDO ALL CAVERN USER PRIORITIES/
     +       1X,44H   2. CHANGE SELECTED CAVERN USER PRIORITIES/
     +       1X,22HENTER NUMBER OF CHOICE)
      READ(45,*,END=13021)IJI
      GO TO 13022
```

```
13021 REWIND 45
      GO TO 13005
13022 IF(IJI.NE.1.AND.IJI.NE.2) THEN
         WRITE(45,12020)
         GO TO 13005
      ENDIF
      IF(IJI.EQ.1) THEN
         GO TO 13500
      ELSE
         GO TO 13700
      ENDIF
13500 DO 13510 I=1,NC
         XNCV=NUMCAV(I)
         KK=XNCV-100
         NCV=XNCV
         IJI=NFIND(1,NCLNR,29,0,XNCV,0)
         CALL RMOVE(IJI,NCLNR,ATRIB)
13519    WRITE(45,13520)NCV
13520    FORMAT(1X,25HENTER PRIORITY FOR CAVERN,I4,
     +            25H, 1 HIGHEST, 2 NEXT, ETC.)
         READ(45,*,END=13521)JPR(KK)
         GO TO 13522
13521    REWIND 45
         GO TO 13519
13522    ATRIB(28)=RANK(JPR(KK))
         CALL FILEM(7,ATRIB)
13510 CONTINUE
      GO TO 16000
13700 WRITE(45,13710)
13710 FORMAT(1X,38HENTER CAVERN NUMBER FOR WHICH PRIORITY/
     +       1X,45HIS TO BE CHANGED OR ENTER 0 TO RETURN TO MENU)
      READ(45,*,END=13713)NCV
      GO TO 13715
13713 REWIND 45
      GO TO 13700
13715 IF(NCV.EQ.0) GO TO 16000
      INV=0
      DO 13711 I=1,NC
         IJI=NUMCAV(I)
         IF(NCV.EQ.IJI) THEN
            INV=1
            GO TO 13714
         ENDIF
13711 CONTINUE
13714 IF(INV.EQ.0) THEN
         WRITE(45,13712)
13712    FORMAT(1X,32HINVALID CAVERN NUMBER--TRY AGAIN)
         GO TO 13700
      ENDIF
13719 WRITE(45,13720)
13720 FORMAT(1X,40HENTER PRIORITY--1,HIGHEST, 2,NEXT,..ETC.)
      KK=NCV-100
      READ(45,*,END=13721)JPR(KK)
```

```
      GO TO 13722
13721 REWIND 45
      GO TO 13719
13722 XNCV=NCV
      IJI=NFIND(1,NCLNR,29,0,XNCV,0)
      CALL RMOVE(IJI,NCLNR,ATRIB)
      ATRIB(28)=RANK(JPR(KK))
      CALL FILEM(7,ATRIB)
13749 WRITE(45,13750)
13750 FORMAT(1X,42HCHANGE ANOTHER CAVERN USER PRIORITY? (Y/N))
      READ(45,11060,END=13751)ICH
      GO TO 13752
13751 REWIND 45
      GO TO 13749
13752 IF(ICH.EQ.'Y') THEN
         GO TO 13700
      ELSE
         GO TO 16000
      ENDIF
14000 GO TO 2005
15000 CALL CHKINP(IPR,JPR,ERR)
      IF(ERR.NE.0) THEN
         GO TO 2005
      ELSE
         RETURN
      ENDIF
16000 WRITE(45,16001)
16001 FORMAT(//1X,33HMAKE ANOTHER CHOICE FROM THE MENU/
     +         5X,27H(ENTER 0 TO SEE MENU AGAIN))
      READ(45,*,END=16002)MCH
      GO TO 99999
16002 REWIND 45
      GO TO 16000
17000 IQT=1
      RETURN
18000 WRITE(45,18700)
18700 FORMAT(1X,47HENTER NUMBER OF CAVERN TO CHANGE MAX BRINE FLOW)
      READ(45,*,END=18010,IOSTAT=IEOF)NOC
18010 IF(IEOF.NE.0) THEN
         REWIND 45
         GO TO 18000
      ENDIF
      DO 18020 I=1,NC
         JCV=I
         IF(NUMCAV(I).EQ.NOC) GO TO 18030
18020 CONTINUE
      WRITE(45,18710)
18710 FORMAT(1X,37HNOT A VALID CAVERN NUMBER---TRY AGAIN)
      GO TO 18000
18030 WRITE(45,18720)NOC
18720 FORMAT(1X,32HENTER MAX BRINE FLOW FOR CAVERN ,I3,9H (MB/DAY)
     *       /5X,21H(MUST BE AT LEAST 40))
      READ(45,*,END=18040,IOSTAT=IEOF)CVMAX(JCV)
```

```
18040 IF(IEOF.NE.0.OR.CVMAX(JCV).LT.40) THEN
         REWIND 45
         GO TO 18030
      ENDIF
      CVMAX(JCV)=CVMAX(JCV)*1000.
      IF(CVMAX(JCV).GT.XX(1)) THEN
         WRITE(45,18730)
18730    FORMAT(1X,46HMAX CAVERN BRINE FLOW MUST NOT EXCEED MAX SITE,
     *            11H BRINE FLOW)
         GO TO 18030
      ENDIF
      GO TO 16000
19000 WRITE(45,19700)
19700 FORMAT(1X,45HENTER NUMBER OF CAVERN TO CHANGE MAX OIL FLOW)
      READ(45,*,END=19010,IOSTAT=IEOF)NOC
19010 IF(IEOF.NE.0) THEN
         REWIND 45
         GO TO 19000
      ENDIF
      DO 19020 I=1,NC
         JCV=I
         IF(NUMCAV(I).EQ.NOC) GO TO 19030
19020 CONTINUE
      WRITE(45,19710)
19710 FORMAT(1X,37HNOT A VALID CAVERN NUMBER---TRY AGAIN)
      GO TO 19000
19030 WRITE(45,19720)NOC
19720 FORMAT(1X,30HENTER MAX OIL FLOW FOR CAVERN ,I3,9H (MB/DAY))
      READ(45,*,END=19040,IOSTAT=IEOF)COMAX(JCV)
19040 IF(IEOF.NE.0) THEN
         REWIND 45
         GO TO 19030
      ENDIF
      COMAX(JCV)=COMAX(JCV)*1000.
      IF(COMAX(JCV).GT.XX(2)) THEN
         WRITE(45,19730)
19730    FORMAT(1X,48HMAX CAVERN OIL FLOW MUST NOT EXCEED MAX SITE OIL,
     *            5H FLOW)
         GO TO 19030
      ENDIF
      GO TO 16000
20000 WRITE(45,20010)
20010 FORMAT(1X,42HENTER CAVERN CLOCK OPTION (Y/N, DEFAULT=N))
      ICK='N'
      READ(45,20020,END=20030,IOSTAT=IEOF)ICK
20020 FORMAT(A1)
20030 IF(IEOF.NE.0) THEN
         REWIND 45
         GO TO 20040
      ENDIF
      IF(ICK.NE.'Y'.AND.ICK.NE.'N') THEN
         WRITE(45,20050)
20050    FORMAT(1X,22HPLEASE ENTER  Y  OR  N)
```

```
         GO TO 20000
      ENDIF
20040 IF(ICK.EQ.'Y') THEN
         CLOK=1
      ELSE
         CLOK=0
      ENDIF
      GO TO 16000
21000 IF(ITR.LE.1) THEN
         WRITE(45,21010)
21010    FORMAT(1X,17HTHERE IS NO TABLE)
         GO TO 16000
      ENDIF
      CALL PRTABL
      GO TO 16000
22000 WRITE(45,22010)
22010 FORMAT(1X,47HENTER CAVERN/OIL VOLUME OPTION (Y/N, DEFAULT=N))
      ICOV='N'
      READ(45,22020,END=22030,IOSTAT=IEOF)ICOV
22020 FORMAT(A1)
22030 IF(IEOF.NE.0) THEN
         REWIND 45
         GO TO 22040
      ENDIF
      IF(ICOV.NE.'Y'.AND.ICOV.NE.'N') THEN
         WRITE(45,22050)
22050    FORMAT(1X,22HPLEASE ENTER  Y  OR  N)
         GO TO 22000
      ENDIF
22040 IF(ICOV.EQ.'Y') THEN
         COVT=1
      ELSE
         COVT=0
      ENDIF
      GO TO 16000
23000 WRITE(45,23010)
23010 FORMAT(1X,46HENTER FORCED ZERO FLOW OPTION (Y/N, DEFAULT=Y))
      IFZER='Y'
      READ(45,23020,END=23030,IOSTAT=IEOF)IFZER
23020 FORMAT(A1)
23030 IF(IEOF.NE.0) THEN
         REWIND 45
         GO TO 23040
      ENDIF
      IF(IFZER.NE.'Y'.AND.IFZER.NE.'N') THEN
         WRITE(45,23050)
23050    FORMAT(1X,22HPLEASE ENTER  Y  OR  N)
         GO TO 23000
      ENDIF
23040 IF(IFZER.EQ.'Y') THEN
         FZER=1
      ELSE
         FZER=0
```

```
      ENDIF
      GO TO 16000
      END
      SUBROUTINE PRTABL
      COMMON/FILES/OUTFIL,PRDFIL,INFIL
      COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
     *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
      COMMON/UCOM4/RANC(20,60),RT(60),ITR
      M=(ITR-1)/15+1
      DO 100 I=1,M
         K=1+(I-1)*15
         L=MIN0(I*15,ITR)
         WRITE(45,10000)(INT(RT(J)),J=K,L)
10000    FORMAT(1X,3HCAV,15(2X,3HDAY)/
     *          1X,3HNUM,15(1X,I4))
         DO 110 J=1,NC
            KCV=NUMCAV(J)
            WRITE(45,10010)KCV,(INT(RANC(J,KKK)/10000.),KKK=K,L)
10010       FORMAT(1X,I3,15(1X,I4))
  110    CONTINUE
      PAUSE 'PRESS RETURN TO CONTINUE'
  100 CONTINUE
      RETURN
      END
            SUBROUTINE CHKINP(IPR,JPR,ERR)
      CHARACTER*5 ST(21),STSV(21),OF(21)
      CHARACTER*8 BDAT,EDATE,DAT
      CHARACTER*1 ITUT,ICH,IDEL,ECH
      CHARACTER*3 PRI
      DIMENSION RANK(20),JPR(20),ICAV(20)
      DIMENSION NSET(8000)
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/XCOM1/ MFE(100),MLE(100),NQ(100)
      COMMON/GCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
     1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
     2TTSET,XXI(100),TTTS,TTTF
      COMMON/XCOM4/ NXACT(100),NXCNT(100),NXGAT(25),NXRSC(75)
      COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
     *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
      COMMON/UCOM2/ST,STSV,BDAT,OF
      COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),OVOL(20)
     *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
      COMMON QSET(8000)
      EQUIVALENCE (NSET(1),QSET(1))
      DATA RANK/1.E4,2.E4,3.E4,4.E4,5.E4,6.E4,7.E4,8.E4,
     *9.E4,10.E4,11.E4,12.E4,13.E4,14.E4,15.E4,16.E4,
     *17.E4,18.E4,19.E4,20.E4/
      ERR=0.
C
C     CHECK MAX FLOW RATES
C
      IF(XX(3).GT.XX(1)) THEN
```

```
         WRITE(45,10000)
10000    FORMAT(1X,33HMAX BRINE FLOW FOR CAVERN EXCEEDS/
     *          1X,34HMAX FLOW FOR SITE---PLEASE CORRECT/)
         ERR=1.
      ENDIF
      IF(XX(4).GT.XX(2)) THEN
         WRITE(45,10010)
10010    FORMAT(1X,31HMAX OIL FLOW FOR CAVERN EXCEEDS/
     *          1X,34HMAX FLOW FOR SITE---PLEASE CORRECT/)
         ERR=1.
      ENDIF
      IF(IPR.EQ.1) THEN
         RETURN
      ELSE
         DO 200 I=1,NC
            KK=NUMCAV(I)
            KKK=KK-100
            IF(I.EQ.1) GO TO 200
            LKL=I-1
            DO 210 JJ=1,LKL
               JJJ=NUMCAV(JJ)-100
               IF(JPR(JJ).EQ.JPR(KKK)) THEN
                  WRITE(45,10040)
10040             FORMAT(/1X,20HTWO OR MORE CAVERNS ,
     *                     22HHAVE THE SAME PRIORITY/
     *                 1X,14HPLEASE CORRECT)
                  ERR=2.
                  RETURN
               ENDIF
210         CONTINUE
200      CONTINUE
      ENDIF
      END
            FUNCTION ZTETA(ISTG,FLOW,V)
      Q= FLOW
      GO TO (10,20,30,40,40,50,50), ISTG
   10 ZTETA= 0.1004 - 4.2952E-4*Q + 5.7859E-7*Q*Q
      RETURN
   20 ZTETA= 0.15505 - 3.3054E-4*Q - 6.11076E-4*V + 1.1394E-5*Q*V
      RETURN
   30 ZTETA= 0.17067 - 1.60865E-4*Q - 6.87857E-4*V + 4.6665E-6*Q*V
      RETURN
   40 ZTETA= 0.14227 - 6.5627E-5*Q + 5.3743E-4*V - 3.0271E-7*Q*V
      RETURN
   50 ZTETA= 0.083139 - 2.2865E-5*Q + 0.0024031*V -5.3335E-6*Q*V
      RETURN
      END
C
            FUNCTION ZTFETA(ISTG,FLOW,V)
      Q=FLOW
      GO TO (10,20,30,40,40,50,50),ISTG
   10 ZTFETA= 0.12059 - 7.1948E-4*Q + 1.6487E-6*Q*Q
      RETURN
```

```
         WRITE(45,10000)
10000    FORMAT(1X,33HMAX BRINE FLOW FOR CAVERN EXCEEDS/
     *          1X,34HMAX FLOW FOR SITE---PLEASE CORRECT/)
         ERR=1.
      ENDIF
      IF(XX(4).GT.XX(2)) THEN
         WRITE(45,10010)
10010    FORMAT(1X,31HMAX OIL FLOW FOR CAVERN EXCEEDS/
     *          1X,34HMAX FLOW FOR SITE---PLEASE CORRECT/)
         ERR=1.
      ENDIF
      IF(IPR.EQ.1) THEN
         RETURN
      ELSE
         DO 200 I=1,NC
            KK=NUMCAV(I)
            KKK=KK-100
            IF(I.EQ.1) GO TO 200
            LKL=I-1
            DO 210 JJ=1,LKL
               JJJ=NUMCAV(JJ)-100
               IF(JPR(JJ).EQ.JPR(KKK)) THEN
                  WRITE(45,10040)
10040             FORMAT(/1X,20HTWO OR MORE CAVERNS ,
     *                      22HHAVE THE SAME PRIORITY/
     *                  1X,14HPLEASE CORRECT)
                  ERR=2.
                  RETURN
               ENDIF
210         CONTINUE
200      CONTINUE
      ENDIF
      END
            FUNCTION ZTETA(ISTG,FLOW,V)
      Q= FLOW
      GO TO (10,20,30,40,40,50,50), ISTG
   10 ZTETA= 0.1004 - 4.2952E-4*Q + 5.7859E-7*Q*Q
      RETURN
   20 ZTETA= 0.15505 - 3.3054E-4*Q - 6.11076E-4*V + 1.1394E-5*Q*V
      RETURN
   30 ZTETA= 0.17067 - 1.60865E-4*Q - 6.87857E-4*V + 4.6665E-6*Q*V
      RETURN
   40 ZTETA= 0.14227 - 6.5627E-5*Q + 5.3743E-4*V - 3.0271E-7*Q*V
      RETURN
   50 ZTETA= 0.083139 - 2.2865E-5*Q + 0.0024031*V -5.3335E-6*Q*V
      RETURN
      END
C
            FUNCTION ZTFETA(ISTG,FLOW,V)
      Q=FLOW
      GO TO (10,20,30,40,40,50,50),ISTG
   10 ZTFETA= 0.12059 - 7.1948E-4*Q + 1.6487E-6*Q*Q
      RETURN
```

```
20 ZTFETA= 0.144205 - 2.40987E-4*Q + 2.12414E-4*V + 4.05896E-6*Q*V
   RETURN
30 ZTFETA= 0.16660 - 1.74333E-4*Q - 7.23541E-5*V + 5.57366E-6*Q*V
   RETURN
40 ZTFETA= 0.176549 - 1.77964E-4*Q - 1.37806E-3*V + 1.10771E-5*Q*V
   RETURN
50 ZTFETA= 0.172931 - 1.49309E-4*Q - 1.11127E-3*V + 8.9116E-6*Q*V
   RETURN
   END
        SUBROUTINE EVENT(IX)
   DIMENSION IPWO(15),AATR(100)
   CHARACTER*5 ST(21),STSV(21),OF(21)
   CHARACTER*8 BDAT,EDATE
   CHARACTER*5 STD
   CHARACTER*10 IPHASE(22)
   CHARACTER*2 CH
   LOGICAL FILL,ANS
   DIMENSION NSET(8000)
   COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
  1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
   COMMON/XCOM1/ MFE(100),MLE(100),NQ(100)
   COMMON/GCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
  1NNAPD,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
  2TTSET,XXI(100),TTTS,TTTF
   COMMON/XCOM4/ NXACT(100),NXCNT(100),NXGAT(25),NXRSC(75)
   COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
  *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
   COMMON/UCOM2/ST,STSV,BDAT,OF
   COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),DVOL(20)
  *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
   COMMON QSET(8000)
   EQUIVALENCE (NSET(1),QSET(1))
   DATA IPHASE/'         S','       S/C','        R1',
  *'        R2','        R3','        F1',
  *'        F2',         F3',
  *'       WO1','       WO3','       WO2',
  *'     END S','   END S/C','    END R1',
  *'    END F1','    END R2','    END F2',
  *'    END R3','    END F3','   END WO2',
  *'   END WO1','   END WO3'/
   DATA IFRST/0/
   DATA IPWO/2,2,2,4,4,4,6,6,6,9,9,9,12,12,12/
   DATA TLST,TLST1/-1.,-1./
   KK=ATRIB(29)-100.
   FILL=.FALSE.
   DO 10 I=1,NC
      IF(NUMCAV(I).EQ.ATRIB(29)) THEN
         K=I
         GO TO 15
      ENDIF
10 CONTINUE
15 CONTINUE
   ATRIB(30)=0.
```

```
      BRI=0.
      OIL=0.
      STD(1:5)='     '
      IF(TLST.NE.TNOW) THEN
         IY=0
         DO 1 I=1,NC
            ST(I)=STSV(I)
            OF(I)='     '
    1    CONTINUE
      ENDIF
      TLST=TNOW
      IF(IFRST.EQ.1)GO TO 100
      IFRST=1
      IF(XX(9).EQ.1) THEN
         WRITE(NPRNT,10000)
10000    FORMAT(///,10H   CAVERN#,3X,10H      TNOW,3X,10H     PHASE,3X,
     *              10H     BRINE,3X,10H       OIL,3X,10H  DURATION,3X,
     *              10H  NXRSC(1),3X,10H ATRIB(27))
      ENDIF
  100 IF(IX.GE.9.AND.IX.LE.38)THEN
         IF(ATRIB(31).EQ.-1) ATRIB(31)=TNOW
         IF(IX-2*(IX/2).NE.0) THEN
            IP=IX-3*(IX/3)+9
            JJ=IX/2-3
            L=IPWO(JJ)
            IF(IX-3*(IX/3).EQ.0) THEN
               I=ATRIB(L)
C
C   ATRIB(28) USED TEMPORARILY TO SAVE ATRIB(40) DURING WORKOVER
C
               ATRIB(28)=ATRIB(40)
C
               ATRIB(40)=I-3*(I/3)
               ATRIB(L)=(I-2)/3+1
            ENDIF
            DUR=ATRIB(L)
            IF(IP.EQ.9) THEN
               ST(K)='  W1 '
               STSV(K)=ST(K)
            ELSEIF(IP.EQ.10) THEN
               ST(K)='  W3 '
               STSV(K)=ST(K)
            ELSE
               ST(K)='  W2 '
               STSV(K)=ST(K)
            ENDIF
            X=NXRSC(1)
            GO TO 20000
         ELSE
C
C   TURN CAVERN CLOCK OFF
C
            CLK(K)=0.
```

```
            IP=IX-3*(IX/3)+20
            ST(K)=' '
            STSV(K)=ST(K)
            JJ=IX/2-4
            ATRIB(27)=ATRIB(27)-ATRIB(IPWD(JJ))
            I=ATRIB(IPWD(JJ))
            IF(IP.EQ.20) CALL WD2(I,ATRIB(40))
            IF(IP.EQ.21) CALL WD1(I,ATRIB(40))
            ATRIB(IPWD(JJ))=I
            IF(IP.EQ.21.OR.IP.EQ.22) THEN
               CALL FREE(2,1)
            ELSE
               CALL FREE(3,1)
            ENDIF
            RES=NXRSC(1)
            CALL TRCE(ATRIB(29),TNOW,IPHASE(IP),0.,0.,0.,RES,ATRIB(27))
         ENDIF
         IF(IX-6*((IX-2)/6).EQ.2) THEN
C
C   RESTORE ATRIB(40)
C
            ATRIB(40)=ATRIB(28)
            ATRIB(28)=0.
C
C   TURN CAVERN CLOCK OFF
C
            CLK(K)=0.
            CALL RESTRT
         ENDIF
         RETURN
      ELSE
         GO TO 150
      ENDIF
  150 IF(IX.GE.51.AND.IX.LE.58) IX=IX-42
      IF(IX.GE.59.AND.IX.LE.66) THEN
         CALL PREVUE(IY)
         RETURN
      ENDIF
      IF(IX.GE.67.AND.IX.LE.71) IX=IX-50
      GO TO(200,400,600,800,1000,1200,1400,1600,
     *      1800,2000,2200,2400,2600,2800,3000,3200,3400,3600,3800,4000,
     *      4200),IX
  200 IJ=ATRIB(1)
      JJ=ATRIB(14)
      KJ=ATRIB(15)
      LJ=2
      MJ=12
      IS(KK)=IS(KK)+1
      JJJ=IS(KK)
      ATRIB(1)=ATR(KK,1,JJJ)
      ATRIB(14)=ATR(KK,14,JJJ)
      ATRIB(15)=ATR(KK,15,JJJ)
      GO TO 50000
```

```
 400 IJ=ATRIB(3)
     JJ=ATRIB(16)
     KJ=ATRIB(17)
     LJ=4
     MJ=13
     ISC(KK)=ISC(KK)+1
     JJJ=ISC(KK)
     ATRIB(3)=ATR(KK,3,JJJ)
     ATRIB(16)=ATR(KK,16,JJJ)
     ATRIB(17)=ATR(KK,17,JJJ)
     GO TO 50000
 600 IJ=ATRIB(5)
     JJ=ATRIB(18)
     KJ=ATRIB(19)
     LJ=6
     MJ=14
     IR1(KK)=IR1(KK)+1
     JJJ=IR1(KK)
     ATRIB(5)=ATR(KK,5,JJJ)
     ATRIB(18)=ATR(KK,18,JJJ)
     ATRIB(19)=ATR(KK,19,JJJ)
     GO TO 50000
 800 IJ=ATRIB(7)
     JJ=ATRIB(20)
     KJ=ATRIB(20)
     LJ=0
     MJ=15
     FILL=.TRUE.
     GO TO 50000
1000 IJ=ATRIB(8)
     JJ=ATRIB(21)
     KJ=ATRIB(22)
     LJ=9
     MJ=16
     IR2(KK)=IR2(KK)+1
     JJJ=IR2(KK)
     ATRIB(8)=ATR(KK,8,JJJ)
     ATRIB(21)=ATR(KK,21,JJJ)
     ATRIB(22)=ATR(KK,22,JJJ)
     GO TO 50000
1200 IJ=ATRIB(10)
     JJ=ATRIB(23)
     KJ=ATRIB(23)
     LJ=0
     MJ=17
     FILL=.TRUE.
     GO TO 50000
1400 IJ=ATRIB(11)
     JJ=ATRIB(24)
     KJ=ATRIB(25)
     LJ=12
     MJ=18
     IR3(KK)=IR3(KK)+1
```

```
      JJJ=IR3(KK)
      ATRIB(11)=ATR(KK,11,JJJ)
      ATRIB(24)=ATR(KK,24,JJJ)
      ATRIB(25)=ATR(KK,25,JJJ)
      GO TO 50000
 1600 IJ=ATRIB(13)
      JJ=ATRIB(26)
      KJ=ATRIB(26)
      LJ=0
      MJ=19
      FILL=.TRUE.
      CTIME=TNOW-ATRIB(31)
      CALL COLCT(CTIME,1)
      NCAV=NCAV+1
C
C   WHEN NCAV=NC, SITE IS COMPLETED. CALL TRCE TO CREATE LAST ENTRY
C   ON SCHEDULE LISTING AND THEN SET MSTOP=-1 TO STOP.
C
      IF(NCAV.EQ.NC) THEN
         CALL TRCE(0.,TNOW,0,0.,0.,0.,0 .,0.)
         MSTOP=-1
      ENDIF
      IF(JJ.LE.0) RETURN
      GO TO 50000
 1800 IJ=14
      JJ=15
      KJ=1
      CH='SU'
      IP=1
      GO TO 60000
 2000 IJ=16
      JJ=17
      KJ=3
      CH='SC'
      IP=2
      GO TO 60000
 2200 IJ=18
      JJ=19
      KJ=5
      CH='R1'
      IP=3
      GO TO 60000
 2400 IJ=20
      JJ=20
      KJ=7
      CH='F1'
      IP=6
      GO TO 60000
 2600 IJ=21
      JJ=22
      KJ=8
      CH='R2'
      IP=4
```

```
      GO TO 60000
 2800 IJ=23
      JJ=23
      KJ=10
      CH='F2'
      IP=7
      GO TO 60000
 3000 IJ=24
      JJ=25
      KJ=11
      CH='R3'
      IP=5
      GO TO 60000
 3200 IJ=26
      JJ=26
      KJ=13
      CH='F3'
      IP=8
      GO TO 60000
 3400 V=RACC(3,KK)/1000.
      YZY=ATRIB(14)/1000.
      LLL=IACC(2,KK)
      IF(LLL.EQ.0) THEN
         EF=ZTETA(1,YZY,V)
      ELSE
         EF=ZTFETA(1,YZY,V)
      ENDIF
      EVOL=RACC(19,KK)
      IF(SS(K)+EF*YZY*ATRIB(1).LE.EVOL) THEN
         ATRIB(40)=0
      ELSE
         X=AINT((EVOL-SS(K))/(EF*YZY)+.5)
         IF(X.LE.0) THEN
            ATRIB(1)=1
            ATRIB(40)=0
         ELSE
            ATRIB(1)=X
            ATRIB(40)=0
         ENDIF
      ENDIF
      RETURN
 3600 V=RACC(3,KK)/1000.
      YZY=ATRIB(16)/1000.
      LLL=IACC(2,KK)
      IF(LLL.EQ.0) THEN
         EF=ZTETA(2,YZY,V)
      ELSE
         EF=ZTFETA(2,YZY,V)
      ENDIF
      EVOL=RACC(20,KK)
      IF(SS(K)+EF*YZY*ATRIB(3).LE.EVOL) THEN
         ATRIB(40)=0
      ELSE
```

```
         X=AINT((EVOL-SS(K))/(EF*YZY)+.5)
         IF(X.LE.0) THEN
            ATRIB(3)=1
            ATRIB(40)=0
         ELSE
            ATRIB(3)=X
            ATRIB(40)=0
         ENDIF
      ENDIF
      RETURN
3800  V=RACC(3,KK)/1000.
      YZY=ATRIB(18)/1000.
      LLL=IACC(2,KK)
      IF(LLL.EQ.0) THEN
         EF=ZTETA(3,YZY,V)
      ELSE
         EF=ZTFETA(3,YZY,V)
      ENDIF
      EVOL=RACC(21,KK)
      IF(SS(K)+EF*YZY*ATRIB(5).LE.EVOL) THEN
         ATRIB(40)=0
      ELSE
         X=AINT((EVOL-SS(K))/(EF*YZY)+.5)
         IF(X.LE.0) THEN
            ATRIB(5)=1
            ATRIB(40)=0
         ELSE
            ATRIB(5)=X
            ATRIB(40)=0
         ENDIF
      ENDIF
      RETURN
4000  V=RACC(3,KK)/1000.
      YZY=ATRIB(21)/1000.
      LLL=IACC(2,KK)
      IF(LLL.EQ.0) THEN
         EF=ZTETA(4,YZY,V)
      ELSE
         EF=ZTFETA(4,YZY,V)
      ENDIF
      IF(ATRIB(32).EQ.-2) THEN
         EVOL=RACC(23,KK)
      ELSE
         EVOL=RACC(22,KK)
      ENDIF
      IF(SS(K)+EF*YZY*ATRIB(8).LE.EVOL) THEN
         ATRIB(40)=0
      ELSE
         X=AINT((EVOL-SS(K))/(EF*YZY)+.5)
         IF(X.LE.0) THEN
            ATRIB(8)=1
            ATRIB(40)=0
         ELSE
```

```
            ATRIB(8)=X
            ATRIB(40)=0
         ENDIF
      ENDIF
      RETURN
 4200 V=RACC(3,KK)/1000.
      YZY=ATRIB(24)/1000.
      LLL=IACC(2,KK)
      IF(LLL.EQ.0) THEN
         EF=ZTETA(6,YZY,V)
      ELSE
         EF=ZTFETA(6,YZY,V)
      ENDIF
      IF(ATRIB(36).EQ.-2) THEN
         EVOL=RACC(25,KK)
      ELSE
         EVOL=RACC(24,KK)
      ENDIF
      IF(SS(K)+EF*YZY*ATRIB(11).LE.EVOL) THEN
         ATRIB(40)=0
      ELSE
         X=AINT((EVOL-SS(K))/(EF*YZY)+.5)
         IF(X.LE.0) THEN
            ATRIB(11)=1
            ATRIB(40)=0
         ELSE
            ATRIB(11)=X
            ATRIB(40)=0
         ENDIF
      ENDIF
      RETURN
20000 CONTINUE
C
C   TURN CAVERN CLOCK ON
C
      CLK(K)=1.
      XX(50)=0.
      IF(ANS) THEN
         IF(STD.NE.'     '.AND.ATRIB(IJ).NE.0)STSV(K)=STD
      ELSE
         IF(STD.NE.'     ')STSV(K)=STD
      ENDIF
      ANS=.FALSE.
      CALL TRCE(ATRIB(29),TNOW,IPHASE(IP),BRI,OIL,DUR,X,ATRIB(27))
      DO 20010 M=1,NC
         IF(ST(M).EQ.'  F1 '.OR.ST(M).EQ.'  F2 '.OR.ST(M).EQ.'  F3 '
     *     .OR.STSV(M).EQ.'  F1 '.OR.STSV(M).EQ.'  F2 '
     *     .OR.STSV(M).EQ.'  F3 '.OR.OF(M).NE.'     ') THEN
              XX(50)=1.
              RETURN
         ENDIF
20010 CONTINUE
      RETURN
```

```
50000 ATRIB(27)=ATRIB(27)-IJ
      CALL FREE(1,JJ)
      IF(KJ.NE.0.AND..NOT.FILL) THEN
         CALL FREE(4,KJ)
         CALL FREE(1,KJ)
      ELSEIF(KJ.NE.0) THEN
         CALL FREE(4,KJ)
      ENDIF
      IF(.NOT.FILL.AND.ATRIB(40).NE.-2) THEN
         XX(32)=XX(32)-OLF(K)
         IF(ATRIB(28).GT.0) THEN
            XX(31)=XX(31)-ATRIB(28)-OLF(K)
         ELSE
            XX(31)=XX(31)-JJ-KJ
         ENDIF
         OLF(K)=0
      ENDIF
C
C   TURN CAVERN CLOCK OFF
C
      CLK(K)=0.
50010 ST(K)=' '
      STSV(K)=ST(K)
      OF(K)=ST(K)
      OR(K)=0.
      RES=NXRSC(1)
      ATRIB(28)=0.
      CALL TRCE(ATRIB(29),TNOW,IPHASE(MJ),0.,0.,0.,RES,ATRIB(27))
      CALL RESTRT
      RETURN
60000 IF(IJ.EQ.20.OR.IJ.EQ.23.OR.IJ.EQ.26) FILL=.TRUE.
      IF(.NOT.FILL) THEN
         XX(20)=ATRIB(IJ)
      ELSE
C
C   ADJUST OIL FLOW IF MAXIMUM CAVERN OIL FLOW RATE IS EXCEEDED
C
         IF(ATRIB(IJ).GT.COMAX(K)) THEN
            ATRIB(27)=ATRIB(27)-ATRIB(KJ)
            ATRIB(KJ)=AINT(ATRIB(IJ)*ATRIB(KJ)/COMAX(K)+.5)
            ATRIB(27)=ATRIB(27)+ATRIB(KJ)
            ATRIB(IJ)=COMAX(K)
         ENDIF
         IF(NXRSC(4).LE.XX(32)) THEN
C
C   THERE IS INSUFFICIENT OIL FOR FILLING, PUT CAVERN ON HOLD.
C
            GO TO 61100
         ENDIF
         ATRIB(28)=ATRIB(IJ)
         ATRIB(40)=ATRIB(KJ)
         ATRIB(IJ)=AMIN1(ATRIB(IJ),NXRSC(4)-XX(32))
         ATRIB(JJ)=ATRIB(IJ)
```

```
         XX(20)=0.
         IF(ATRIB(IJ).EQ.ATRIB(28)) THEN
            ATRIB(28)=0.
            ATRIB(40)=0.
         ELSE
            ATRIB(27)=ATRIB(27)-ATRIB(KJ)
            ATRIB(KJ)=AINT(ATRIB(28)*ATRIB(KJ)/ATRIB(IJ)+.5)
            ATRIB(27)=ATRIB(27)+ATRIB(KJ)
         ENDIF
      ENDIF
      XX(21)=ATRIB(JJ)
      IF(NXRSC(1).GE.XX(20)+XX(21).AND.NXRSC(4).GE.XX(21)) THEN
C
C        THERE ARE SUFFICIENT BRINE AND OIL FLOWS AVAILABLE
C
         IF(.NOT.FILL.AND.ATRIB(28).EQ.0.AND.ATRIB(40).NE.-2) THEN
            OLF(K)=ATRIB(JJ)
         ELSEIF(FILL) THEN
            OLF(K)=0
         ENDIF
         GO TO 61000
      ENDIF
      IF(NXRSC(1).LT.40000.AND.IJ.NE.14.AND.IJ.NE.16.AND..NOT.FILL.OR.
     *   NXRSC(4).EQ.0.AND.XX(21).GT.0.OR.NXRSC(1).EQ.0)
C
C        IF NOT SUMP, NOT SUMP CHIMNEY, NOT FILL AND BRINE FLOW
C        AVAILABLE IS < 40000   OR
C        IF FILL (OR LEACH/FILL) AND OIL FLOW AVAILABLE IS 0   OR
C        IF BRINE FLOW AVAILABLE IS 0    THEN
C        PUT CAVERN ON HOLD
C
     *                                                GO TO 61100
C
      IF(NXRSC(1).LT.XX(20)+XX(21).AND.NXRSC(4).GE.XX(21)) THEN
C
C        THERE IS SUFFICIENT OIL FLOW AVAILABLE BUT NOT BRINE FLOW.
C        USE BRINE FLOW AVAILABLE AND ADJUST OIL FLOW AND NUMBER OF
C        DAYS ACCORDINGLY.
C
         ATRIB(28)=ATRIB(IJ)
         ATRIB(40)=ATRIB(KJ)
         ATRIB(IJ)=AINT(NXRSC(1)*XX(20)/(XX(20)+XX(21)))
         ATRIB(27)=ATRIB(27)-ATRIB(KJ)
         IF(FILL) THEN
            ATRIB(KJ)=AINT(ATRIB(28)*ATRIB(KJ)/NXRSC(1)+.5)
         ELSE
            NON=KJ/2+1
            LLL=IACC(2,KK)
            V=RACC(3,KK)/1000.
            YZY=ATRIB(IJ)/1000.
            ZYZ=ATRIB(28)/1000.
            IF(LLL.EQ.0)THEN
               EN=ZTETA(NON,YZY,V)
```

```
            EO=ZTETA(NON,ZYZ,V)
         ELSE
            EN=ZTFETA(NON,YZY,V)
            EO=ZTFETA(NON,ZYZ,V)
         ENDIF
         ATRIB(KJ)=AINT(EO*ATRIB(29)*ATRIB(KJ)/
     *                  (EN*ATRIB(IJ))+.5)
      ENDIF
      ATRIB(27)=ATRIB(27)+ATRIB(KJ)
      IF(.NOT.FILL) THEN
         OLF(K)=ATRIB(JJ)
      ELSE
         OLF(K)=0.
      ENDIF
      ATRIB(JJ)=NXRSC(1)-ATRIB(IJ)
    ELSEIF(NXRSC(1).GE.XX(20)+XX(21)) THEN
C
C     THERE IS SUFFICIENT BRINE FLOW AVAILABLE BUT NOT OIL.
C     USE OIL FLOW AVAILABLE AND ADJUST BRINE FLOW AND NUMBER
C     OF DAYS ACCORDINGLY
C
      IF(NXRSC(4)*ATRIB(IJ)/ATRIB(JJ).LT.40000.AND.IJ.NE.14
     *   .AND.IJ.NE.16.AND..NOT.FILL)
C
C     IF NOT SUMP, NOT SUMP CHIMNEY, NOT FILL AND OIL FLOW AVAILABLE
C     WOULD FORCE A BRINE FLOW < 40000 THEN PUT CAVERN ON HOLD.
C
     *                                 GO TO 61100
      ATRIB(28)=ATRIB(IJ)
      ATRIB(40)=ATRIB(KJ)
      ATRIB(IJ)=AINT(NXRSC(4)*ATRIB(IJ)/ATRIB(JJ))
      IF(.NOT.FILL) THEN
         OLF(K)=ATRIB(JJ)
      ELSE
         OLF(K)=0.
      ENDIF
      ATRIB(JJ)=NXRSC(4)
      ATRIb(27)=ATRIB(27)-ATRIB(KJ)
      IF(FILL) THEN
         ATRIB(KJ)=AINT(ATRIB(28)*ATRIB(KJ)/ATRIB(IJ)+.5)
      ELSE
         NON=KJ/2+1
         LLL=IACC(2,KK)
         V=RACC(3,KK)/1000.
         YZY=ATRIB(IJ)/1000.
         ZYZ=ATRIB(28)/1000.
         IF(LLL.EQ.0)THEN
            EN=ZTETA(NON,YZY,V)
            EO=ZTETA(NON,ZYZ,V)
         ELSE
            EN=ZTFETA(NON,YZY,V)
            EO=ZTFETA(NON,ZYZ,V)
         ENDIF
```

```
            ATRIB(KJ)=AINT(EO*ATRIB(28)*ATRIB(KJ)/
     *                    (EN*ATRIB(IJ))+.5)
         ENDIF
         ATRIB(27)=ATRIB(27)+ATRIB(KJ)
      ELSE
C
C        INSUFFICIENT BRINE AND OIL FLOW AVAILABLE.
C        USE BRINE FLOW AVAILABLE OR OIL FLOW AVAILABLE,
C        WHICHEVER IS FEASIBLE, AND ADJUST THE OTHER FLOW
C        AND NUMBER OF DAYS ACCORDINGLY.
C
         X=NXRSC(4)*XX(20)/(XX(20)+XX(21))
         IF(X.LE.NXRSC(1)) THEN
C
C        USING ALL OIL FLOW AVAILABLE THERE IS SUFFICIENT BRINE
C        FLOW AVAILABLE.
C
            IF(X.LT.40000.AND.IJ.NE.14.AND.IJ.NE.16.AND..NOT.FILL)
C
C        IF NOT SUMP, NOT SUMP CHIMNEY, NOT FILL AND OIL FLOW AVAILABLE
C        WOULD FORCE A BRINE FLOW < 40000 THEN PUT CAVERN ON HOLD.
C
     *                                                GO TO 61100
            ATRIB(28)=ATRIB(IJ)
            ATRIB(40)=ATRIB(KJ)
            ATRIB(IJ)=AINT(X)
            IF(.NOT.FILL) THEN
               OLF(K)=ATRIB(JJ)
            ELSE
               OLF(K)=0.
            ENDIF
            ATRIB(JJ)=NXRSC(4)
            ATRIB(27)=ATRIB(27)-ATRIB(KJ)
            IF(FILL) THEN
               ATRIB(KJ)=AINT(ATRIB(28)*ATRIB(KJ)/ATRIB(IJ)+.5)
            ELSE
               NON=KJ/2+1
               LLL=IACC(2,KK)
               V=RACC(3,KK)/1000.
               YZY=ATRIB(IJ)/1000.
               ZYZ=ATRIB(28)/1000.
               IF(LLL.EQ.0)THEN
                  EN=ZTETA(NON,YZY,V)
                  EO=ZTETA(NON,ZYZ,V)
               ELSE
                  EN=ZTFETA(NON,YZY,V)
                  EO=ZTFETA(NON,ZYZ,V)
               ENDIF
               ATRIB(KJ)=AINT(EO*ATRIB(28)*ATRIB(KJ)/
     *                    (EN*ATRIB(IJ))+.5)
            ENDIF
            ATRIB(27)=ATRIB(27)+ATRIB(KJ)
         ELSE
```

```
C
C       USING ALL OIL FLOW AVAILABLE THERE IS NOT SUFFICIENT BRINE
C       FLOW AVAILABLE. USE BRINE FLOW AVAILABLE AND ADJUST OIL FLOW
C       AND NUMBER OF DAYS ACCORDINGLY.
C
          ATRIB(28)=ATRIB(IJ)
          ATRIB(40)=ATRIB(KJ)
          ATRIB(IJ)=AINT(NXRSC(1)*XX(20)/(XX(20)+XX(21)))
          ATRIB(27)=ATRIB(27)-ATRIB(KJ)
          IF(FILL) THEN
             ATRIB(KJ)=AINT(ATRIB(28)*ATRIB(KJ)/ATRIB(IJ)+.5)
          ELSE
             NON=KJ/2+1
             LLL=IACC(2,KK)
             V=RACC(3,KK)/1000.
             YZY=ATRIB(IJ)/1000.
             ZYZ=ATRIB(28)/1000.
             IF(LLL.EQ.0)THEN
                EN=ZTETA(NON,YZY,V)
                EO=ZTETA(NON,ZYZ,V)
             ELSE
                EN=ZTFETA(NON,YZY,V)
                EO=ZTFETA(NON,ZYZ,V)
             ENDIF
             ATRIB(KJ)=AINT(EO*ATRIB(28)*ATRIB(KJ)/
     *                      (EN*ATRIB(IJ))+.5)
          ENDIF
          ATRIB(27)=ATRIB(27)+ATRIB(KJ)
          IF(.NOT.FILL) THEN
             OLF(K)=ATRIB(JJ)
          ELSE
             OLF(K)=0.
          ENDIF
          ATRIB(JJ)=NXRSC(1)-ATRIB(IJ)
       ENDIF
    ENDIF
61000 I=ATRIB(IJ)/1000.+.5
    CALL CHFINT(I,STD,CH)
    ST(K)=STD
    STD='  '//CH//' '
    STSV(K)=STD
    IF(ATRIB(IJ).EQ.0) THEN
       ST(K)='  0  '
       STSV(K)=ST(K)
    ENDIF
    IXX=OR(K)/1000.+.5
    IF(IXX.EQ.I.AND.ATRIB(IJ).NE.0) THEN
       ST(K)=STD
    ELSEIF(ATRIB(JJ).NE.0.AND..NOT.FILL) THEN
       I=ATRIB(JJ)/1000+.5
       CALL CHFINT(I,OF(K),'LF')
    ENDIF
    BRI=ATRIB(IJ)
```

```
      OIL=ATRIB(JJ)
      DUR=ATRIB(KJ)
      IF(ATRIB(31).EQ.-1) ATRIB(31)=TNOW
      IF(.NOT.FILL) THEN
         XX(20)=ATRIB(IJ)
         V=RACC(3,KK)/1000.
         NON=KJ/2+1
         LLL=IACC(2,KK)
         YZY=XX(20)/1000.
         IF(LLL.EQ.0)THEN
            EN=ZTETA(NON,YZY,V)
         ELSE
            EN=ZTFETA(NON,YZY,V)
         ENDIF
         CVOL(K)=EN*XX(20)
         OVOL(K)=ATRIB(JJ)
         FILL=.FALSE.
      ELSE
         IY=ATRIB(29)
         XX(20)=0.
         OVOL(K)=ATRIB(JJ)
      ENDIF
      XX(21)=ATRIB(JJ)
      XX(20)=XX(20)+XX(21)
      X=NXRSC(1)-XX(20)
      ANS=.TRUE.
      GO TO 20000
61100 ATRIB(30)=-1.
      OR(K)=0.
      FILL=.FALSE.
      RETURN
      END
            FUNCTION USERF(IZ)
      DIMENSION NSET(8000)
      DIMENSION IFLO(5),IOIL(5),IDAY(5)
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/XCOM1/ MFE(100),MLE(100),NQ(100)
      COMMON/GCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
     1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
     2TTSET,XXI(100),TTTS,TTTF
      COMMON/XCOM4/ NXACT(100),NXCNT(100),NXGAT(25),NXRSC(75)
      COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
     *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
      COMMON/UCOM2/ST,STSV,BDAT,OF
      COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),OVOL(20)
     *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
      COMMON QSET(8000)
      EQUIVALENCE (NSET(1),QSET(1))
      DATA IFLO/14,16,18,21,24/
      DATA IOIL/15,17,19,22,25/
      DATA IDAY/1,3,5,8,11/
      USERF=0.
```

```
      DO 10 I=1,NC
         IF(NUMCAV(I).EQ.ATRIB(29)) THEN
            K=I
            GO TO 15
         ENDIF
   10 CONTINUE
   15 CONTINUE
      KK=NUMCAV(K)-100
      IJ=IFLO(IZ)
      JJ=IOIL(IZ)
      KJ=IDAY(IZ)
C
C   ADJUST FLOW RATES IF MAXIMUM CAVERN BRINE FLOW RATE IS EXCEEDED.
C
      IF(ATRIB(IJ)+ATRIB(JJ).GT.CVMAX(K)) THEN
         ATRIB(27)=ATRIB(27)-ATRIB(KJ)
         ATRIB(28)=ATRIB(IJ)
         A=ATRIB(IJ)/(ATRIB(IJ)+ATRIB(JJ))
         ATRIB(IJ)=AINT(A*CVMAX(K))
         ATRIB(JJ)=CVMAX(K)-ATRIB(IJ)
         NON=KJ/2+1
         LLL=IACC(2,KK)
         V=RACC(3,KK)/1000.
         YZY=ATRIB(IJ)/1000.
         ZYZ=ATRIB(28)/1000.
         IF(LLL.EQ.0)THEN
            EN=ZTETA(NON,YZY,V)
            EO=ZTETA(NON,ZYZ,V)
         ELSE
            EN=ZTFETA(NON,YZY,V)
            EO=ZTFETA(NON,ZYZ,V)
         ENDIF
         ATRIB(KJ)=AINT(EO*ATRIB(28)*ATRIB(KJ)/
     *                 (EN*ATRIB(IJ))+.5)
         ATRIB(28)=0.
         ATRIB(27)=ATRIB(27)+ATRIB(KJ)
      ENDIF
      RETURN
      END
            SUBROUTINE PREVUE(IY)
C ******************************************************************
C     WHEN THE NUMBER OF UNCOMPLETED CAVERNS IS SUCH THAT THE SITE MAXI-
C     MUM BRINE FLOW RATE CAN NOT BE ACHIEVED USING THE NOMINAL CAVERN
C     FLOW RATES, SUBROUTINE PREVUE IS CALLED TO SEE IF THESE NOMINAL
C     RATES CAN BE INCREASED (SUBJECT TO THE MAXIMUM DEFAULT CAVERN FLOW
C     RATE CONSTRAINT). ALSO ANY CAVERNS AT ZERO FLOW RATE ARE BROUGHT
C     ON LINE, IF THE FORCED ZERO FLOW OPTION IS NOT CHOSEN.
C ******************************************************************
      DIMENSION IDAYS(8),IRATE(8),AATR(100),IOIL(8),IZFLO(20),ATRB(100)
      CHARACTER*5 ST(21),STSV(21),OF(21),STD
      CHARACTER*8 BDAT,EDATE
      CHARACTER*2 CH
      LOGICAL FILL
      DIMENSION NSET(8000)
```

```
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/XCOM1/ MFE(100),MLE(100),NQ(100)
      COMMON/GCOM1/ JJCDR,FKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
     1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
     2TTSET,XXI(100),TTTS,TTTF
      COMMON/XCOM4/ NXACT(100),NXCNT(100),NXGAT(25),NXRSC(75)
      COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
     *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
      COMMON/UCOM2/ST,STSV,BDAT,OF
      COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),OVOL(20)
     *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
      COMMON QSET(8000)
      EQUIVALENCE (NSET(1),QSET(1))
      DATA TLST,TLAST/-1.,0./
      DATA IDAYS/1,3,5,7,8,10,11,13/
      DATA IRATE/14,16,18,20,21,23,24,26/
      DATA IOIL/15,17,19,20,22,23,25,26/
      DO 200 I=1,20
         IZFLO(I)=0
  200 CONTINUE
      KZF=0
      N=NNQ(4)
      IF(N.EQ.0) RETURN
      BRI=NXRSC(1)-XX(31)
      OIL=NXRSC(4)-XX(32)
      DO 100 LJL=1,50
         IF(LJL.GT.N) GO TO 300
         CALL RMOVE(LJL,4,AATR)
         KK=AATR(29)-100.
         DO 10 I=1,NC
            IF(NUMCAV(I).EQ.AATR(29)) THEN
            KCV=I
            GO TO 15
            ENDIF
   10    CONTINUE
   15    CONTINUE
         IPT=AATR(NNAM1)
         IPT=NSET(IPT+2)
         IK=NSET(IPT+4)-50
         I=IDAYS(IK)
         J=IRATE(IK)
         K=IOIL(IK)
         IF(AATR(J).EQ.0.AND.FZER.EQ.0) THEN
            KZF=KZF+1
            IZFLO(KZF)=AATR(29)
            CALL FFILE(4,AATR)
            GO TO 90
         ENDIF
         IF(IK.EQ.1.OR.IK.EQ.2
     *       .OR.AATR(J)+AATR(K).GE.XX(3)
     *       .OR.AATR(J).EQ.0) THEN
            CALL FFILE(4,AATR)
```

```
          GO TO 90
        ENDIF
 300    IF(AATR(K).EQ.0) THEN
          S=BRI+AATR(J)
          X=AMIN1(XX(3),S)
          AATR(28)=AATR(J)
C
C*********************************************************************
C
C    PROCEDURE FOR BRINGING A CAVERN AT ZERO FLOW ON LINE
C
          IF(AATR(28).EQ.0) THEN
            IF(X.LT.40000) THEN
C
C   BRINE FLOW RATE IN R1, R2, OR R3 MUST BE AT LEAST 40000 MB/DAY
C   DO NOT BRING CAVERN ON LINE
C
              CALL FFILE(4,AATR)
              GO TO 90
            ENDIF

            AATR(40)=-2.
            AATR(J)=X
            IF(J.LE.18.OR.IACC(2,KK).GT.0) THEN
C
C   SUMP, SUMP/CHIMNEY, R1 OR NOT LEACH/FILL
C
              IST=J/2+12
              IF(J.EQ.21.AND.AATR(32).EQ.-2
     *            .OR.J.EQ.24.AND.AATR(36).EQ.-2) IST=IST+1
              V=RACC(3,KK)/1000.
              YZY=X/1000.
              JST=IST-18
              EF=ZTFETA(JST,YZY,V)
              IF(SS(KCV)+EF*YZY*AATR(I).GT.RACC(IST,KK)) THEN
C
C   REDUCE NUMBER OF FLOW DAYS IF CAVERN END VOLUME FOR THIS STAGE
C   WOULD BE EXCEEDED
C
                AATR(28)=AATR(I)
                AATR(27)=AATR(27)-AATR(I)
                AATR(I)=AINT((RACC(IST,KK)-SS(KCV))/(EF*YZY)+.5)
                AATR(28)=AATR(I)-AATR(28)
                AATR(27)=AATR(27)+AATR(I)
              ENDIF
              BRI=BRI-X
              CALL FFILE(4,AATR)
              IF(BRI.EQ.0) RETURN
              GO TO 90
            ENDIF
C
C   LEACH/FILL
C
```

```
            IF(OIL.LE.0.AND.IY.EQ.0) THEN
C
C   NO OIL AVAILABLE AND NO CAVERN IN FILL MODE
C   CANNOT BRING CAVERN ON LINE
C
              AATR(J)=0
              AATR(K)=0
              AATR(40)=0
              CALL FFILE(4,AATR)
              GO TO 90
            ENDIF
C
C   DETERMINE STAGE, R2-1, R2-2, R3-1 OR R3-2
C
            IF(J.EQ.21) THEN
C
C   LEACH STAGE R2
C
              IF(AATR(32).NE.-2) THEN
C
C   LEACH STAGE R2-1
C
                OBR=R21
                IST=22
              ELSE
C
C   LEACH STAGE R2-2
C
                OBR=R22
                IST=23
              ENDIF
            ELSEIF(J.EQ.24) THEN
C
C   LEACH STAGE R3
C
              IF(AATR(36).NE.-2) THEN
C
C   LEACH STAGE R3-1
C
                OBR=R31
                IST=24
              ELSE
C
C   LEACH STAGE R3-2
C
                OBR=R32
                IST=25
              ENDIF
            ENDIF
            V=RACC(3,KK)/1000.
            JST=IST-18
C
C   APPORTION FLOWS SO THAT MAXIMUM CAVERN BRINE FLOW IS NOT EXCEEDED
```

```
C
            AATR(K)=AATR(J)*OBR
            A=AATR(J)/(AATR(J)+AATR(K))
            Y=X-AINT(A*X)
            X=AINT(A*X)
            ZYZ=Y/1000.
C
C   ADJUST OIL FLOW RATE IF FINAL OIL VOLUME WOULD BE EXCEEDED
C
            IF(ZYZ*AATR(I)+SS(KCV+40).GT.RACC(3,KK))THEN
               SY=Y
               Y=AINT((RACC(3,KK)-SS(KCV+40))*1000./AATR(I))
               X=X+SY-Y
            ENDIF
            IF(Y.GT.OIL.AND.IY.EQ.0) THEN
C
C   IF NOT ENOUGH OIL AVAILABLE AND NO CAVERN IN FILL MODE
C   USE OIL AVAILABLE
C
               SY=Y
               Y=OIL
               X=X+SY-Y
            ELSEIF(Y.GT.OIL) THEN
C
C   IF NOT ENOUGH OIL AVAILABLE FOR THIS LEACH/FILL, THEN
C   REDUCE THE OIL FLOW OF THE LOWEST PRIORITY CAVERN IN A
C   FILL ACTIVITY
C
               Z=Y-OIL
C
C   REDUCE OIL FLOW OF CAVERN IN FILL MODE
C
               XIY=IY
               NTRY=MFE(NCLNR)
               IJI=NFIND(-NTRY,NCLNR,29,0,XIY,0)
               CALL RMOVE(-IJI,NCLNR,ATRB)
               KLK=IJI+NNAM1
               IPT=-QSET(KLK)
               IPT=NSET(IPT+2)
               IPT=NSET(IPT+4)
               IJI=20 +(IPT/2-2)*3
               IKI=7+(IPT/2-2)*3
               IF(Z.GE.ATRB(IJI)) THEN
C
C   DO NOT REDUCE CAVERN IN FILL MODE TO ZERO FLOW
C   PUT CURRENT CAVERN BACK TO ZERO FLOW
C
                  AATR(J)=0
                  AATR(K)=0
                  AATR(40)=0
                  CALL FFILE(4,AATR)
                  CALL FFILE(NCLNR,ATRB)
                  GO TO 90
```

```
                ENDIF
                DO 400 JRJ=1,NC
                   IF(NUMCAV(JRJ).EQ.IY) THEN
                      LCV=JRJ
                      GO TO 410
                   ENDIF
 400            CONTINUE
 410            CONTINUE
                IF(IPT.EQ.4) THEN
                   CH='F1'
                ELSEIF(IPT.EQ.6) THEN
                   CH='F2'
                ELSEIF(IPT.EQ.8) THEN
                   CH='F3'
                ENDIF
                NXRSC(1)=NXRSC(1)+Z
                NXRSC(4)=NXRSC(4)+Z
                BRI=BRI+Z
                OIL=OIL+Z
                ATRB(27)=ATRB(27)-ATRB(IKI)
                IF(ATRB(28).LE.0) THEN
                   ATRB(28)=ATRB(IJI)
                   ATRB(40)=ATRB(IKI)
                ENDIF
                ATRB(IJI)=ATRB(IJI)-Z
                ATRB(IKI)=AINT(((ATRB(IJI)+Z)/ATRB(IJI))*ATRB(IKI)
     *                        +.5)
                ATRB(27)=ATRB(27)+ATRB(IKI)
                ATRB(NNATR)=TNOW+ATRB(IKI)
                IMI=ATRB(IJI)/1000.+.5
                CALL CHFINT(IMI,STD,CH)
                ST(LCV)=STD
                STSV(LCV)='  '//CH//' '
                CALL FFILE(NCLNR,ATRB)
             ENDIF
             YZY=X/1000.
             EF=ZTETA(JST,YZY,V)
C
C   REDUCE NUMBER OF FLOW DAYS IF CAVERN END VOLUME FOR THIS STAGE
C   WOULD BE EXCEEDED
C
             IF(SS(KCV)+EF*YZY*AATR(I).GT.RACC(JST,KK)) THEN
                AATR(28)=AATR(I)
                AATR(I)=AINT((RACC(JST,KK)-SS(KCV))/(EF*YZY)+.5)
C               Y=AMIN1(X*OBR,Y*AATR(28)/AATR(I))
                AATR(28)=AATR(I)-AATR(26)
             ENDIF
             AATR(J)=X
             AATR(K)=Y
             BRI=BRI-X-Y
             OIL=OIL-Y
             CALL FFILE(4,AATR)
             IF(BRI.EQ.0) RETURN
```

```
            GO TO 90
          ENDIF
C
C*********************************************************************
          AATR(40)=AATR(I)
          AATR(27)=AATR(27)-AATR(I)
          NON=I/2+1
          LLL=IACC(2,KK)
          V=RACC(3,KK)/1000.
          YZY=X/1000.
          ZYZ=AATR(28)/1000.
          IF(LLL.EQ.0)THEN
            EN=ZTETA(NON,YZY,V)
            EO=ZTETA(NON,ZYZ,V)
          ELSE
            EN=ZTFETA(NON,YZY,V)
            EO=ZTFETA(NON,ZYZ,V)
          ENDIF
          AATR(I)=AINT(EO*AATR(28)*AATR(I)/(EN*X)+.5)
          AATR(27)=AATR(27)+AATR(I)
          AATR(J)=X
          BRI=BRI-X+AATR(28)
          CALL FFILE(4,AATR)
          IF(BRI.NE.0) THEN
            GO TO 90
          ELSE
            RETURN
          ENDIF
        ELSE
          IF(OIL.EQ.0.AND.IY.EQ.0) THEN
            CALL FFILE(4,AATR)
            GO TO 90
          ENDIF
          S=BRI+AATR(J)+AATR(K)
          OLF(KCV)=AATR(K)
          X=AMIN1(XX(3),S)
          A=AATR(J)/(AATR(J)+AATR(K))
          Y=X-AINT(A*X)
          X=AINT(A*X)
          Z=Y-AATR(K)
          IF(Z.GT.OIL.AND.IY.EQ.0) THEN
C
C   NOT ENOUGH OIL AVAILABLE FOR THIS LEACH/FILL AND THERE IS NO
C   CAVERN IN FILL MODE. USE OIL AVAILABLE.
C
            Y=OIL+AATR(K)
            X=AINT(A*Y/(1.-A))
            GO TO 95
          ELSEIF(Z.GT.OIL) THEN
C
C   IF NOT ENOUGH OIL AVAILABLE FOR THIS LEACH/FILL, THEN
C   REDUCE THE OIL FLOW OF THE LOWEST PRIORITY CAVERN IN A
C   FILL ACTIVITY
```

```
C
            Z=Z-OIL
C
C   REDUCE OIL FLOW OF CAVERN IN FILL MODE
C
            XIY=IY
            NTRY=MFE(NCLNR)
            IJI=NFIND(-NTRY,NCLNR,29,0,XIY,0)
            CALL RMOVE(-IJI,NCLNR,ATRB)
            KLK=IJI+NNAM1
            IPT=-QSET(KLK)
            IPT=NSET(IPT+2)
            IPT=NSET(IPT+4)
            IJI=20 +(IPT/2-2)*3
            IKI=7+(IPT/2-2)*3
            IF(Z.GE.ATRB(IJI)) THEN
C
C   DO NOT REDUCE CAVERN IN FILL MODE TO ZERO FLOW
C   USE OIL AVAILABLE
C
               Y=OIL+AATR(K)
               X=AINT(A*Y/(1.-A))
               CALL FFILE(NCLNR,ATRB)
               GO TO 95
            ENDIF
            DO 500 JRJ=1,NC
               IF(NUMCAV(JRJ).EQ.IY) THEN
                  LCV=JRJ
                  GO TO 510
               ENDIF
 500        CONTINUE
 510        CONTINUE
            IF(IPT.EQ.4) THEN
               CH='F1'
            ELSEIF(IPT.EQ.6) THEN
               CH='F2'
            ELSEIF(IPT.EQ.8) THEN
               CH='F3'
            ENDIF
            NXRSC(1)=NXRSC(1)+Z
            NXRSC(4)=NXRSC(4)+Z
            BRI=BRI+Z
            OIL=OIL+Z
            ATRB(27)=ATRB(27)-ATRB(IKI)
            IF(ATRB(28).LE.0) THEN
               ATRB(28)=ATRB(IJI)
               ATRB(40)=ATRB(IKI)
            ENDIF
            ATRB(IJI)=ATRB(IJI)-Z
            ATRB(IKI)=AINT(((ATRB(IJI)+Z)/ATRB(IJI))*ATRB(IKI)
     *                    +.5)
            ATRB(27)=ATRB(27)+ATRB(IKI)
            ATRB(NNATR)=TNOW+ATRB(IKI)
```

```
            IMI=ATRB(IJ1)/1000.+.5
            CALL CHFINT(IMI,STD,CH)
            ST(LCV)=STD
            STSV(LCV)='  '//CH//' '
            CALL FFILE(NCLNR,ATRB)
         ENDIF
95       AATR(28)=AATR(J)
         AATR(40)=AATR(I)
         AATR(27)=AATR(27)-AATR(I)
         NON=I/2+1
         LLL=IACC(2,KK)
         V=RACC(3,KK)/1000.
         YZY=X/1000.
         ZYZ=AATR(28)/1000.
         IF(LLL.EQ.0)THEN
             EN=ZTETA(NON,YZY,V)
             EO=ZTETA(NON,ZYZ,V)
         ELSE
             EN=ZTFETA(NON,YZY,V)
             EO=ZTFETA(NON,ZYZ,V)
         ENDIF
         AATR(I)=AINT(EO*AATR(28)*AATR(I)/(EN*X)+.5)
         AATR(27)=AATR(27)+AATR(I)
         BRI=BRI-X+AATR(28)-Y+AATR(K)
         OIL=OIL-Y+AATR(K)
         AATR(J)=X
         AATR(K)=Y
         CALL FFILE(4,AATR)
         IF(BRI.NE.0) THEN
            GO TO 90
         ELSE
            RETURN
         ENDIF
      ENDIF
90    IF(LJL.LT.N) GO TO 100
      IF(KZF.EQ.0.OR.LJL.EQ.N+KZF) RETURN
      KKK=LJL+1-N
      XJ=IZFLO(KKK)
      NTRY=MFE(4)
      JJJ=NFIND(-NTRY,4,29,0,XJ,0)
      CALL RMOVE(-JJJ,4,AATR)
      KK=AATR(29)-100.
      DO 91 I=1,NC
         IF(NUMCAV(I).EQ.AATR(29)) THEN
         KCV=I
         GO TO 92
         ENDIF
91    CONTINUE
92    IPT=AATR(NNAM1)
      IPT=NSET(IPT+2)
      IK=NSET(IPT+4)-50
      I=IDAYS(IK)
      J=IRATE(IK)
```

```
      K=IOIL(IK)
  100 CONTINUE
      RETURN
      END
            SUBROUTINE RESTRT
C ***********************************************************************
C     AT EACH BREAKPOINT (OTHER THAN A TRANSITION FROM WORKOVER 1 TO
C     WORKOVER 2 OR FROM WORKOVER 2 TO WORKOVER 3) SUBROUTINE RESTRT IS
C     CALLED TO HALT ALL ACTIVITY AND SEND EACH CAVERN BACK THROUGH THE
C     SCHEDULING ALGORITHM.
C ***********************************************************************
      DIMENSION IDAYS(8),IRATE(8),AATR(100),IOIL(8)
      CHARACTER*5 ST(21),STSV(21),OF(21)
      CHARACTER*8 BDAT,EDATE
      LOGICAL FILL
      DIMENSION NSET(8000)
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/XCOM1/ MFE(100),MLE(100),NQ(100)
      COMMON/GCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
     1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
     2TTSET,XXI(100),TTTS,TTTF
      COMMON/XCOM4/ NXACT(100),NXCNT(100),NXGAT(25),NXRSC(75)
      COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
     *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
      COMMON/UCOM2/ST,STSV,BDAT,OF
      COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),OVOL(20)
     *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
      COMMON QSET(8000)
      EQUIVALENCE (NSET(1),QSET(1))
      DATA TLST,TLAST/-1.,0./
      DATA IDAYS/1,3,5,7,8,10,11,13/
      DATA IRATE/14,16,18,20,21,23,24,26/
      DATA IOIL/15,17,19,20,22,23,25,26/
      CALL ENTER(9,ATRIB)
C     DVOL=0.
      DO 100 I=1,NC
         CVOL(I)=0.
         OVOL(I)=0.
  100 CONTINUE
      NTRY=MFE(NCLNR)
   10 IF(NTRY.EQ.0) THEN
         TLST=TNOW
         TLAST=TNOW
         RETURN
      ENDIF
      CALL COPY(-NTRY,NCLNR,AATR)
      IF(AATR(NNATR).EQ.TNOW) GO TO 20
      K=NTRY+NNAM1
      IPT=-QSET(K)
      IPT=NSET(IPT+2)
      IF(NSET(IPT).NE.15) GO TO 20
      IK=NSET(IPT+4)
```

```
      IF(IK.LT.1.OR.IK.GT.8) GO TO 20
      FILL=.FALSE.
      IF(IK.EQ.4.OR.IK.EQ.6.OR.IK.EQ.8)FILL=.TRUE.
      CALL RMOVE(-NTRY,NCLNR,AATR)
      I=IDAYS(IK)
      J=IRATE(IK)
      K=IOIL(IK)
      DO 30 IJI=1,NC
         IF(NUMCAV(IJI).EQ.AATR(29))THEN
            KKK=IJI
            GO TO 35
         ENDIF
   30 CONTINUE
   35 LL=AATR(29)-100.
      IF(TLST.NE.TNOW) THEN
         OR(KKK)=AATR(J)
      ENDIF
      JJ=AATR(K)
      IF(AATR(28).GT.0) THEN
         AATR(27)=AATR(27)-AATR(I)
         IF(FILL) THEN
            AATR(I)=AINT((AATR(40)*AATR(28)-(TNOW-TLAST)*AATR(J))/
     *                    AATR(28)+.5)
         ELSE
            NON=I/2+1
            LLL=IACC(2,LL)
            V=RACC(3,LL)/1000.
            YZY=AATR(J)/1000.
            ZYZ=AATR(28)/1000.
            IF(LLL.EQ.0)THEN
               EN=ZTETA(NON,YZY,V)
               EO=ZTETA(NON,ZYZ,V)
            ELSE
               EN=ZTFETA(NON,YZY,V)
               EO=ZTFETA(NON,ZYZ,V)
            ENDIF
            AATR(I)=AINT((EO*AATR(40)*AATR(28)-EN*(TNOW-TLAST)
     *                    *AATR(J))/(EO*AATR(28))+.5)
            AATR(K)=OLF(KKK)
         ENDIF
         AATR(27)=AATR(27)+AATR(I)
         IJ=AATR(J)
         AATR(J)=AATR(28)
      ELSE
         IJ=AATR(J)
         AATR(27)=AATR(27)-AATR(I)+AATR(NNATR)-TNOW
         AATR(I)=AATR(NNATR)-TNOW
C
         IF(AATR(40).EQ.-2) THEN
C
C     CAVERN WAS AT ZERO FLOW
C
            AATR(J)=0.
```

```
            AATR(K)=0.
            AATR(27)=AATR(27)-AATR(28)
            AATR(I)=AATR(I)-AATR(28)
         ENDIF
C
      ENDIF
      CALL FREE(1,IJ)
      IF(JJ.NE.0) THEN
         CALL FREE(4,JJ)
         IF(.NOT.FILL) CALL FREE(1,JJ)
      ENDIF
      ST(LL)=' '
      STSV(LL)=' '
      OF(LL)='      '
      AATR(28)=0.
C
C   TURN CAVERN CLOCK OFF
C
      CLK(KKK)=0.
      CALL ENTER(IK,AATR)
   20 NTRY=NSUCR(NTRY)
      GO TO 10
      END
            SUBROUTINE CHKDAT(D,ERR)
      CHARACTER*8 D
      CHARACTER*5 ST(21),STSV(21),OF(21)
      CHARACTER*8 BDAT,EDATE
      CHARACTER*1 ITUT,ICH,IDEL,ECH
      DIMENSION RANK(20),JPR(20)
      DIMENSION NSET(8000)
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/XCOM1/ MFE(100),MLE(100),NQ(100)
      COMMON/GCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
     1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
     2TTSET,XXI(100),TTTS,TTTF
      COMMON/XCOM4/ NXACT(100),NXCNT(100),NXGAT(25),NXRSC(75)
      COMMON/UCOM1/NC,NCAV,DR(21),DLF(21),NUMCAV(21),R21,R22,R31,R32,
     *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
      COMMON/UCOM2/ST,STSV,BDAT,OF
      COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),OVOL(20)
     *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
      COMMON QSET(8000)
      EQUIVALENCE (NSET(1),QSET(1))
      ERR=0.
      IF(D(1:1).EQ.'/') THEN
         D(1:8)=BDAT(1:8)
         ERR=2.
         RETURN
      ENDIF
      DO 10 I=1,8
         IF(I.EQ.3.OR.I.EQ.6) THEN
            IF(D(I:I).NE.'/') THEN
```

```
            ERR=1.
         ENDIF
      ELSEIF(D(I:I).LT.'0'.OR.D(I:I).GT.'9') THEN
         ERR=1.
      ENDIF
10 CONTINUE
   RETURN
   END
         SUBROUTINE CHFINT(I,C,ST)
   CHARACTER*2 ST
   CHARACTER*5 C
   C(1:5)='     '
   J1=I/100
   IF(J1.EQ.0) THEN
      C(1:1)=' '
   ELSE
      C(1:1)=CHAR(J1+16)
   ENDIF
   J2=(I-100*J1)/10
   IF(J1.EQ.0.AND.J2.EQ.0) THEN
      C(2:2)=' '
   ELSE
      C(2:2)=CHAR(J2+16)
   ENDIF
   J3=(I-100*J1-10*J2)
   C(3:3)=CHAR(J3+16)
   C(4:4)=ST(1:1)
   C(5:5)=ST(2:2)
   RETURN
   END
         SUBROUTINE WD1(I,X)
   IF(X.EQ.1) THEN
      I=I+1
   ELSEIF(X.EQ.2) THEN
      I=I-1
   ENDIF
   RETURN
   END
         SUBROUTINE WD2(I,X)
   IF(X.EQ.1) THEN
      I=I-1
   ELSEIF(X.EQ.2) THEN
      I=I+1
   ENDIF
   RETURN
   END
         SUBROUTINE TRCE(A24,TN,IPH,BRI,OIL,DUR,RES,A27)
   CHARACTER*5 ST(21),STSV(21),OF(21)
   CHARACTER*5 SST(21),SSTSV(21),SOF(21)
   CHARACTER*8 BDAT,EDATE
   CHARACTER*10 IPH
   CHARACTER*20 FMT(4)
   DIMENSION ISS(20)
```

```
      COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
     *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
      COMMON/UCOM2/ST,STSV,BDAT,OF
      COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),DVOL(20)
     *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      DATA TLST,TLAST/-1.,0./
      DATA FMT/'(2E11.5,2E12.5)','(4E11.5,2E12.5)','(6E11.5,2E12.5)',
     *         '(8E11.5,2E12.5)'/
C     TTN=TN
C     IF(TN.EQ.TLST) TTN=0.
C     IF(XX(9).EQ.1) THEN
C        WRITE(NPRNT,1000)A24,TTN,IPH,BRI,OIL,DUR,RES,A27
C1000    FORMAT(/,2(F10.2,3X),A10,3X,5(F10.2,3X))
C     ENDIF
      IF(TN.EQ.TLAST.AND.NCAV.LT.NC)THEN
         DO 200 I=1,NC
            SST(I)=ST(I)
            SSTSV(I)=STSV(I)
            SOF(I)=OF(I)
            J=I+80
            ISS(I)=SS(J)
 200     CONTINUE
      ELSE
         TL=TLAST
         CALL TIMDAT(BDAT,TL,EDATE)
         WRITE(51,1010)EDATE,TL,(SST(J),J=1,NC)
         WRITE(51,1020)(SSTSV(J),J=1,NC)
         WRITE(51,1020)(SOF(J),J=1,NC)
         IF(CLOK.EQ.1) THEN
            WRITE(51,1060)(ISS(J),J=1,NC)
 1060       FORMAT(20I4)
         ENDIF
         DO 210 I=1,NC
            SST(I)=ST(I)
            SSTSV(I)=STSV(I)
            SOF(I)=OF(I)
            J=I+80
            ISS(I)=SS(J)
  210    CONTINUE
         TLAST=TN
      ENDIF
 1010 FORMAT(A8,F8.1,20A5)
 1020 FORMAT(20A5)
      IF(TN.EQ.TLST.OR.COVT.EQ.0) RETURN
      K=NC/5
      L=NC-5*K
      ITN=TN
 1050 FORMAT(I4)
      DO 100 I=1,K
         JTAPE=45+I
         WRITE(JTAPE,1050)ITN
```

```
      KKK=5*(I-1)+1
      LLL=KKK+4
      WRITE(JTAPE,1030)(SS(J),SS(J+40),J=KKK,LLL)
1030  FORMAT(10E11.5)
 100 CONTINUE
    JTAPE=46+K
    IF(L.NE.0) THEN
      WRITE(JTAPE,1050)ITN
      KKK=5*K+1
      LLL=KKK+L-1
      WRITE(JTAPE,FMT(L))(SS(J),SS(J+40),J=KKK,LLL),XX(42),XX(43)
    ELSE
      WRITE(JTAPE,1050)ITN
      WRITE(JTAPE,1040)XX(42),XX(43)
    ENDIF
1040 FORMAT(2E12.6)
    TLST=TN
    RETURN
    END
          SUBROUTINE TIMDAT(BDATE,XDAYS,EDATE)
    CHARACTER*8 BDATE,EDATE
    CHARACTER*1 B,C
    INTEGER M,D,Y
    DAYS=XDAYS
    B=BDATE(1:1)
    C=BDATE(2:2)
    M=10*(ICHAR(B)-16)+ICHAR(C)-16
    B=BDATE(4:4)
    C=BDATE(5:5)
    D=10*(ICHAR(B)-16)+ICHAR(C)-16
    B=BDATE(7:7)
    C=BDATE(8:8)
    Y=10*(ICHAR(B)-16)+ICHAR(C)-16+1900
    IF(M.LE.2) THEN
      YP=Y-1
      MP=M+13
    ELSE
      YP=Y
      MP=M+1
    ENDIF
    J=INT(365.25*YP)+INT(30.6001*MP)+D+DAYS
    YP=INT((J-122.1)/365.25)
    MP=INT((J-INT(365.25*YP))/30.6001)
    D=J-INT(365.25*YP)-INT(30.6001*MP)
    IF(MP.GE.14) THEN
      M=MP-13
      Y=YP+1
    ELSE
      M=MP-1
      Y=YP
    ENDIF
    IF(Y.LE.1999) THEN
      Y=Y-1900
```

```
ELSE
   Y=Y-2000
ENDIF
J1=M/10
J2=M-10*J1
EDATE(1:1)=CHAR(J1+16)
EDATE(2:2)=CHAR(J2+16)
EDATE(3:3)='/'
J1=D/10
J2=D-J1*10
EDATE(4:4)=CHAR(J1+16)
EDATE(5:5)=CHAR(J2+16)
EDATE(6:6)='/'
J1=Y/10
J2=Y-J1*10
EDATE(7:7)=CHAR(J1+16)
EDATE(8:8)=CHAR(J2+16)
RETURN
END
      SUBROUTINE D1D2DY(BDATE,EDATE,XDAY)
CHARACTER*8 BDATE,EDATE
CHARACTER*1 B,C
INTEGER M,D,Y
B=BDATE(1:1)
C=BDATE(2:2)
M=10*(ICHAR(B)-16)+ICHAR(C)-16
B=BDATE(4:4)
C=BDATE(5:5)
D=10*(ICHAR(B)-16)+ICHAR(C)-16
B=BDATE(7:7)
C=BDATE(8:8)
Y=10*(ICHAR(B)-16)+ICHAR(C)-16+1900
IF(M.LE.2) THEN
   MP=M+13
   YP=Y-1
ELSE
   MP=M+1
   YP=Y
ENDIF
J1=INT(365.25*YP)+INT(30.6001*MP)+D
B=EDATE(1:1)
C=EDATE(2:2)
M=10*(ICHAR(B)-16)+ICHAR(C)-16
B=EDATE(4:4)
C=EDATE(5:5)
D=10*(ICHAR(B)-16)+ICHAR(C)-16
B=EDATE(7:7)
C=EDATE(8:8)
Y=10*(ICHAR(B)-16)+ICHAR(C)-16+1900
IF(M.LE.2) THEN
   MP=M+13
   YP=Y-1
ELSE
```

```
      MP=M+1
      YP=Y
    ENDIF
    J2=INT(365.25*YP)+INT(30.6001*MP)+D
    XDAY=J2-J1
    RETURN
    END
          SUBROUTINE STATE
    CHARACTER*5 ST(21),STSV(21),OF(21)
    CHARACTER*8 BDAT,EDATE
    DIMENSION NSET(8000)
    COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
   1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
    COMMON/XCOM1/ MFE(100),MLE(100),NQ(100)
    COMMON/GCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
   1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
   2TTSET,XXI(100),TTTS,TTTF
    COMMON/XCOM4/ NXACT(100),NXCNT(100),NXGAT(25),NXRSC(75)
    COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
   *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
    COMMON/UCOM2/ST,STSV,BDAT,OF
    COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),OVOL(20)
   *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
    COMMON/UCOM4/RANC(20,60),RT(60),ITR
    COMMON QSET(8000)
    EQUIVALENCE (NSET(1),QSET(1))
    DATA TLAST/-1./
    IF(TNOW.EQ.0) ITTR=2
C
C       XX(40)=SITE BRINE FLOW RATE
C       XX(41)=SITE OIL FLOW RATE
C       XX(42)=SITE LEACHED VOLUME
C       XX(43)=SITE OIL VOLUME
C
        XX(40)=(XX(1)-NXRSC(1))/1000.
        XX(41)=(XX(2)-NXRSC(4))/1000.
        XX(42)=0.
        XX(43)=0.
    DO 100 I=1,NC
C
C     CAVERN LEACHED VOLUMES
C
        J=20+I
        SS(J)=CVOL(I)
        SS(I)=SSL(I)+SSL(J)*DTNOW/1000.
        XX(42)=XX(42)+SS(I)
C
C     CAVERN OIL VOLUMES
C
        K=40+I
        L=60+I
        SS(L)=OVOL(I)
        SS(K)=SSL(K)+SSL(L)*DTNOW/1000.
```

```
         XX(43)=XX(43)+SS(K)
  100 CONTINUE
C
C     CAVERN CLOCKS
C
      IF(TNOW.NE.TLAST) THEN
         DO 200 I=1,NC
            J=I+80
            SS(J)=SSL(J)+CLK(I)
  200    CONTINUE
      ENDIF
      IF(TNOW.NE.TLAST) LL=0
      IF(TNEXT.EQ.TNOW) THEN
         TLAST=TNOW
         RETURN
      ENDIF
      IF(ITR.GT.1.AND.LL.EQ.0) THEN
         IF(ITTR.LE.ITR) THEN
            IF(RT(ITTR).EQ.TNOW) THEN
               LL=ITTR
               CALL RERANK(LL)
               ITTR=ITTR+1
               TLAST=TNOW
               RETURN
            ELSE
               GO TO 300
            ENDIF
         ENDIF
      ENDIF
  300 CONTINUE
      IF(XX(10).NE.0.AND.XX(10).EQ.TNOW) THEN
         CALL RERANK(LL)
C        CALL TRCE(0.,TNOW,0,0.,0.,0.,0.,0.)
      ENDIF
      TLAST=TNOW
      RETURN
      END
            SUBROUTINE RERANK(LL)
      DIMENSION NSET(8000),AATR(100)
      DIMENSION RANK(20),JPR(20)
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/XCOM1/ MFE(100),MLE(100),NQ(100)
      COMMON/GCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
     1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
     2TTSET,XXI(100),TTTS,TTTF
      COMMON/XCOM4/ NXACT(100),NXCNT(100),NXGAT(25),NXRSC(75)
      COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
     *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
      COMMON/UCOM2/ST,STSV,DDAT,OF
      COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),OVOL(20)
     *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
      COMMON/UCOM4/RANC(20,60),RT(60),ITR
```

```
      COMMON QSET(8000)
      EQUIVALENCE (NSET(1),QSET(1))
      DATA RANK/1.E4,2.E4,3.E4,4.E4,5.E4,6.E4,7.E4,8.E4,
     *9.E4,10.E4,11.E4,12.E4,13.E4,14.E4,15.E4,16.E4,
     *17.E4,18.E4,19.E4,20.E4/
      ITN=TNOW
      IF(LL.NE.0) GO TO 2001
      ITR=ITR+1
      WRITE(45,100)ITN
  100 FORMAT(//1X,26HSCHEDULING STOPPED AT DAY ,I4,12H IN ORDER TO/
     +          25HCHANGE CAVERN PRIORITIES.//)
      DO 7000 I=1,NC
         KK=NUMCAV(I)
         KKK=KK-100
         IF(I.EQ.1) THEN
            WRITE(45,7500)
7500        FORMAT(1X,25HENTER CAVERN PRIORITIES--/
     +             1X,23H1 HIGHEST, 2 NEXT, ETC./)
         ENDIF
 7004    WRITE(45,7510)KK
 7510    FORMAT(1X,26HENTER PRIORITY FOR CAVERN ,I3)
 7005    READ(*5,*,END=7006)JPR(KKK)
         GO TO 7007
 7006    REWIND 45
         GO TO 7004
 7007    INV=0
         DO 7010 JJ=1,NC
            IF(JPR(KKK).EQ.JJ) THEN
               INV=1
            ENDIF
            IF(JJ.LT.I) THEN
               JJJ=NUMCAV(JJ)-100
               IF(JPR(JJJ).EQ.JPR(KKK)) THEN
                  WRITE(45,7520)JPR(KKK)
 7520             FORMAT(1X,14HYOU HAVE USED ,I2,9H ALREADY-
     +                    ,11H--TRY AGAIN)
                  GO TO 7004
               ENDIF
            ENDIF
7010     CONTINUE
         IF(INV.EQ.0) THEN
            L=1
            WRITE(45,7530)L,NC
7530        FORMAT(1X,32HPRIORITY SHOULD BE IN THE RANGE ,I2,
     +                4H TO ,I2,11H--TRY AGAIN)
            GO TO 7005
         ENDIF
C        ATRIB(28)=RANK(JPR(KKK))
         RANC(I,ITR)=RANK(JPR(KKK))
         RT(ITR)=TNOW
         WRITE(55,*)RANC(I,ITR),RT(ITR)
10000    FORMAT(2F10.1)
 7000 CONTINUE
```

```
2001 IF(LL.EQ.0) THEN
        JJ=ITR
     ELSE
        JJ=LL
     ENDIF
     DO 2000 III=1,NCLNR
        IF(III.EQ.2.OR.III.GT.6.AND.III.LT.NCLNR) GO TO 2000
        NTRY=MFE(III)
1000    NXT=NSUCR(NTRY)
        IF(NTRY.EQ.0) GO TO 2000
        CALL RMOVE(-NTRY,III,AATR)
        DO 1010 I=1,NC
           IF(NUMCAV(I).EQ.AATR(29)) THEN
              K=I
              GO TO 1020
           ENDIF
1010    CONTINUE
1020    AATR(27)=RANC(K,JJ)
        NTRY=NXT
        CALL FFILE(III,AATR)
        GO TO 1000
2000 CONTINUE
     IF(LL.NE.0) GO TO 5001
     IF(ITR.EQ.60) THEN
        WRITE(45,4060)
4060    FORMAT(1X,45HTHERE IS NO STORAGE LEFT FOR SCHEDULING STOPS/
    *            44H SCHEDULING WILL CONTINUE TO SITE COMPLETION)
        XX(10)=0.
        GO TO 5002
     ENDIF
 400 WRITE(45,4000)
4000 FORMAT(1X,18HSCHEDULING OPTIONS/
    *       6X,38H1. DEVELOP SCHEDULE TO SITE COMPLETION/
    *       6X,39H2. STOP SCHEDULE AFTER A NUMBER OF DAYS/
    *       6X,33H     AND REPRIORITIZE THE CAVERNS//
    *       1X,22HENTER NUMBER OF CHOICE)
     READ(45,4010,END=4011)ISCH
     GO TO 401
4011 REWIND 45
     GO TO 400
4010 FORMAT(I1)
 401 IF(ISCH.NE.1.AND.ISCH.NE.2) THEN
        WRITE(45,6000)
6000    FORMAT(1X,21HBAD INPUT---TRY AGAIN)
        GO TO 400
     ENDIF
     IF(ISCH.EQ.2) THEN
 410    WRITE(45,4020)
4020    FORMAT(1X,24HENTER THE NUMBER OF DAYS)
        READ(45,4030,END=4031)IDAYS
        GO TO 411
4031    REWIND 45
        GO TO 410
```

```
4030    FORMAT(I4)
 411    IF(IDAYS.LE.XX(10)) THEN
           IDAYS=XX(10)
           WRITE(45,4040)IDAYS
4040       FORMAT(1X,36HNUMBER OF DAYS MUST BE GREATER THAN ,I5/
    *            1X,9HTRY AGAIN)
           GO TO 410
        ENDIF
        XX(10)=IDAYS
     ELSE
        XX(10)=0
     ENDIF
5002 WRITE(45,5000)
5000 FORMAT(//1X,21HSCHEDULING CONTINUING)
     ENDFILE45
5001 CALL RESTRT
     RETURN
     END
          SUBROUTINE OTPUT
     CHARACTER*5 ST(21),STSV(21),OF(21)
     CHARACTER*8 BDAT,EDATE
     CHARACTER*80 FMT1(4)
     CHARACTER*160 FMT2(4),FMT4(4)
     CHARACTER*20 FMT3(4)
     DIMENSION IX(20),ISS(20)
     DIMENSION NSET(8000)
     COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,NSTOP,NCLNR
    1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
     COMMON/XCOM1/ MFE(100),MLE(100),NQ(100)
     COMMON/GCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
    1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
    2TTSET,XX1(100),TTTS,TTTF
     COMMON/XCOM4/ NXACT(100),NXCNT(100),NXGAT(25),NXRSC(75)
     COMMON/UCOM1/NC,NCAV,OR(21),OLF(21),NUMCAV(21),R21,R22,R31,R32,
    *IS(20),ISC(20),IR1(20),IR2(20),IR3(20),ATR(20,27,16)
     COMMON/UCOM2/ST,STSV,BDAT,OF
     COMMON/UCOM3/IACC(25,20),RACC(31,20),DVOL,FACTOR,CVOL(20),OVOL(20)
    *,CVMAX(20),COMAX(20),CLK(20),CLOK,COVT,FZER
     COMMON QSET(8000)
     EQUIVALENCE (NSET(1),QSET(1))
     DATA FMT1/
    *'(7HCAVERN#,10X,I3,20X,11HSITE TOTALS)',
    *'(7HCAVERN#,10X,I3,23X,I3,20X,11HSITE TOTALS)',
    *'(7HCAVERN#,10X,I3,23X,I3,23X,I3,20X,11HSITE TOTALS)',
    *'(7HCAVERN#,10X,I3,23X,I3,23X,I3,23X,I3,20X,11HSITE TOTALS)'/
     DATA FMT2/
    *'(5H#DAYS,4X,6HCAVERN,8X,3HOIL,9X,6HCAVERN,8X,3HOIL)',
    *'(5H#DAYS,4X,6HCAVERN,8X,3HOIL,9X,6HCAVERN,8X,3HOIL,
    *9X,6HCAVERN,8X,3HOIL)',
    *'(5H#DAYS,4X,6HCAVERN,8X,3HOIL,9X,6HCAVERN,8X,3HOIL,
    *9X,6HCAVERN,8X,3HOIL,9X,6HCAVERN,8X,3HOIL)',
    *'(5H#DAYS,4X,6HCAVERN,8X,3HOIL,9X,6HCAVERN,8X,3HOIL,
    *9X,6HCAVERN,8X,3HOIL,9X,6HCAVERN,8X,3HOIL,9X,6HCAVERN,8X,3HOIL)'/
```

```
      DATA FMT3/'(2E11.5,2E12.6)','(4E11.5,2E12.6)','(6E11.5,2E12.6)',
     *          '(8E11.5,2E12.6)'/
      DATA FMT4/
     *'(I4,2X,E11.5,2X,E11.5,2X,E12.6,2X,E12.6)',
     *'(I4,2X,E11.5,2X,E11.5,2X,E11.5,2X,E11.5,2X,E12.6,
     *2X,E12.6)',
     *'(I4,2X,E11.5,2X,E11.5,2X,E11.5,2X,E11.5,2X,E11.5,
     *2X,E11.5,2X,E12.6,2X,E12.6)',
     *'(I4,2X,E11.5,2X,E11.5,2X,E11.5,2X,E11.5,2X,E11.5,
     *2X,E11.5,2X,E11.5,2X,E11.5,2X,E12.6,2X,E12.6)'/
      NNN=NNQ(5)
      WRITE(NPRNT,1040)
      WRITE(NPRNT,2000)
 2000 FORMAT(40X,25H**** S C H E D U L E ****//)
      I1=XX(1)/1000.
      I2=XX(2)/1000.
      I3=XX(3)/1000.
      I4=XX(4)/1000.
      I5=XX(5)
      WRITE(NPRNT,2010)I1
 2010 FORMAT(35X,27HMAXIMUM BRINE FLOW FOR SITE,T70,I4,7H MB/DAY)
      WRITE(NPRNT,2020)I2
 2020 FORMAT(35X,25HMAXIMUM OIL FLOW FOR SITE,T70,I4,7H MB/DAY)
      WRITE(NPRNT,2030)I3
 2030 FORMAT(35X,29HMAXIMUM BRINE FLOW FOR CAVERN,T70,I4,7H MB/DAY)
      WRITE(NPRNT,2040)I4
 2040 FORMAT(35X,27HMAXIMUM OIL FLOW FOR CAVERN,T70,I4,7H MB/DAY)
      WRITE(NPRNT,2050)I5
 2050 FORMAT(35X,23HNUMBER OF WORKOVER RIGS,T70,I4)
      WRITE(NPRNT,1040)
      WRITE(NPRNT,2060)
 2060 FORMAT(47X,11HL E G E N D//)
      WRITE(NPRNT,2070)
 2070 FORMAT(35X,2HSU,10X,4HSUMP/
     *       35X,2HSC,10X,12HSUMP/CHIMNEY/
     *       35X,2HR1,10X,9HREVERSE 1/
     *       35X,2HR2,10X,9HREVERSE 2/
     *       35X,2HR3,10X,9HREVERSE 3/
     *       35X,2HF1,10X,23HINTERMEDIATE FILL PHASE/
     *       35X,2HF2,10X,23HINTERMEDIATE FILL PHASE/
     *       35X,2HF3,10X,16HFINAL FILL PHASE/
     *       35X,2HLF,10X,19HLEACH/FILL OIL FLOW/
     *       35X,2HW1,10X,16HWORKOVER PHASE 1/
     *       35X,2HW2,10X,16HWORKOVER PHASE 2/
     *       35X,2HW3,10X,16HWORKOVER PHASE 3/
     *       35X,2HF ,10X,25HSOME CAVERN IN FILL PHASE)
      WRITE(NPRNT,1040)
      WRITE(NPRNT,1000)
 1000 FORMAT(18X,13HCAVERN NUMBER)
      WRITE(NPRNT,1010)(NUMCAV(I),I=1,NC)
 1010 FORMAT(2X,4HDATE,3X,6H# DAYS,3X,20(I3,3X))
      REWIND51
   20 READ(51,1020,END=100)EDATE,TN,(ST(J),J=1,NC)
```

```
      READ (51,1050)(STSV(J),J=1,NC)
      READ(51,1050)(OF(J),J=1,NC)
      IF(CLOK.EQ.1) THEN
         READ(51,1080)(ISS(J),J=1,NC)
1080     FORMAT(20I4)
      ENDIF
1050 FORMAT(20A5)
1020 FORMAT(A8,F8.1,20A5)
      ST(21)=' '
      DO 30 I=1,NC
         IF(ST(I).EQ.'  F1 '.OR.ST(I).EQ.'  F2 '
     *          .OR.ST(I).EQ.'  F3 '
     *          .OR.STSV(I).EQ.'  F1 '.OR.STSV(I).EQ.'  F2 '
     *          .OR.STSV(I).EQ.'  F3 '.OR.OF(I).NE.'     ') THEN
            ST(21)='   F '
         ENDIF
   30 CONTINUE
      ITN=TN
      WRITE(NPRNT,1030)EDATE,ITN,(ST(J),J=1,NC),ST(21)
      DO 40 I=1,NC
         IF(OF(I).NE.'     ') THEN
            WRITE(NPRNT,1070)(OF(KKK),KKK=1,NC),'     '
            GO TO 1095
         ENDIF
   40 CONTINUE
1070 FORMAT(17X,21(A5,1H+))
1095 IF(CLOK.EQ.1) THEN
         WRITE(NPRNT,1090)(ISS(J),J=1,NC)
1090     FORMAT(2X,12HCAVERN CLOCK,3X,21(I5,1H+))
      ENDIF
      GO TO 20
1030 FORMAT(A8,1X,I6,2X,21(A5,1H+))
  100 CONTINUE
      WRITE(NPRNT,1040)
1040 FORMAT(/////)
      ITN=TNOW
      WRITE(NPRNT,1060)ITN
1060 FORMAT(20X,27H**** TOTAL TIME FOR SITE = ,I5,10H DAYS ****///)
       CALL PRNTC(0)
       CALL PRNTT(0)
       WRITE(NPRNT,1100)
1100   FORMAT(//,1X,29HFILE 1 IS F1 OR F2 FILL QUEUE//
     *          1X,23HFILE 3 IS F3 FILL QUEUE//
     *          1X,24HFILE 4 IS LEACHING QUEUE//
     *          1X,31HFILE 5 IS WORKOVER 1 OR 3 QUEUE//
     *          1X,26HFILE 6 IS WORKOVER 2 QUEUE//)
      CALL PRNTF(1)
      CALL PRNTF(3)
      CALL PRNTF(4)
      CALL PRNTF(5)
      CALL PRNTF(6)
      CALL PRNTR(1)
      CALL PRNTR(2)
```

```
      CALL PRNTR(3)
      CALL PRNTR(4)
      WRITE(NPRNT,1040)
      CALL PRNTP(0)
      IF(COVT.EQ.0) RETURN
      WRITE(NPRNT,1040)
      WRITE(NPRNT,1200)
 1200 FORMAT(53X,23H**** V O L U M E S ****)
      WRITE(NPRNT,1040)
      REWIND 46
      REWIND 47
      REWIND 48
      REWIND 49
      REWIND 50
      K=NC/5
      L=NC-5*K
      DO 200 I=1,K
         KKK=5*(I-1)+1
         LLL=KKK+4
         WRITE(NPRNT,1220)(NUMCAV(J),J=KKK,LLL)
 1220    FORMAT(1H ,7HCAVERN#,10X,I3,23X,I3,23X,I3,23X,I3,23X,I3)
         WRITE(NPRNT,1230)
 1230    FORMAT(1H ,5H#DAYS,4X,6HCAVERN,8X,3HOIL,9X,6HCAVERN,8X,3HOIL,
     *                     9X,6HCAVERN,8X,3HOIL,9X,6HCAVERN,8X,3HOIL,
     *                     9X,6HCAVERN,8X,3HOIL)
         JTAPE=45+I
  220    READ(JTAPE,1210,END=210)ITN
 1210    FORMAT(I4)
         READ(JTAPE,1240,END=210)(SS(J),SS(J+40),J=KKK,LLL)
 1240    FORMAT(10E11.5)
         WRITE(NPRNT,1250)ITN,(SS(J),SS(J+40),J=KKK,LLL)
 1250    FORMAT(1H ,I4,2X,E11.5,2X,E11.5,2X,E11.5,2X,E11.5,2X,E11.5,
     *                 2X,E11.5,2X,E11.5,2X,E11.5,2X,E11.5,2X,E11.5)
         GO TO 220
  210    WRITE(NPRNT,1040)
  200 CONTINUE
      JTAPE=46+K
      IF(L.NE.0) THEN
         KKK=5*K+1
         LLL=KKK+L-1
         WRITE(NPRNT,FMT1(L))(NUMCAV(J),J=KKK,LLL)
         WRITE(NPRNT,FMT2(L))
  230    READ(JTAPE,1210,END=240)ITN
         READ(JTAPE,FMT3(L),END=240)(SS(J),SS(J+40),J=KKK,LLL),
     *        XX(42),XX(43)
         WRITE(NPRNT,FMT4(L))ITN,(SS(J),SS(J+40),J=KKK,LLL),
     *        XX(42),XX(43)
         GO TO 230
  240    CONTINUE
      ELSE
         WRITE(NPRNT,1260)
 1260    FORMAT(1H ,14X,11HSITE TOTALS)
         WRITE(NPRNT,1270)
```

```
1270    FORMAT(1H ,5H#DAYS,9X,6HCAVERN,9X,3HOIL)
 260    READ(JTAPE,1210,END=250)ITN
        READ(JTAPE,1280)XX(42),XX(43)
1280    FORMAT(2E12.6)
        WRITE(NPRNT,1290)ITN,XX(42),XX(43)
1290    FORMAT(1H ,I4,2X,E12.6,2X,E12.6)
        GO TO 260
 250    CONTINUE
      ENDIF
      RETURN
      END
            SUBROUTINE EDITOR
C       ***********************************************************
C                              ** EDITOR **
C
C          THIS IS THE MAIN CONTROL MODULE FOR THE EDITOR PROGRAM
C          FOR THE SPR SCHEDULE FILE.
C
C       ***********************************************************
C
      INTEGER INP, OUT, ERR,OLD,NEW
      INTEGER TRU, FALS
      CHARACTER WYE,ENN,QMARK,BUF(80),ICHR
      CHARACTER*8 INFIL,OUTFIL,PRDFIL
      COMMON /FILES/OUTFIL,PRDFIL,INFIL
      COMMON /INPUT/ LINLMT
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /ALPHAS/ WYE, ENN,QMARK,BUF
      COMMON /LOGICS/ TRU, FALS
      COMMON /BLK1/ OILVOL(3)
      IPRVU = FALS
      ICHNGD = FALS
      INFIL='TAPE1'
      WRITE (OUT,99999)
      WRITE (OUT,99998)
      WRITE (OUT,99997)
      WRITE (OUT,99996)
      WRITE(5,4000)
 4000 FORMAT(1H ,///37HENTER INPUT FILE NAME OR PRESS RETURN
     *            /,20HIF CREATING NEW FILE/)
      READ(5,4010,END=4011,IOSTAT=IEOF)INFIL
 4011 IF(IEOF.NE.0) REWIND 5
 4010 FORMAT(A8)
      OPEN(OLD,FILE=INFIL)
      CALL INFILL(NEWFLG,INFIL)
      CLOSE(OLD)
 4021 WRITE(5,4020)
 4020 FORMAT(1H ,22HENTER OUTPUT FILE NAME)
      IF(INFIL.NE.'TAPE1')WRITE(5,4030)INFIL
 4030 FORMAT(1H ,10X,11H(DEFAULT = ,A8,1H)/)
      READ(5,4010,END=4031,IOSTAT=IEOF)OUTFIL
 4031 IF(IEOF.NE.0) THEN
         REWIND 5
```

```
         IF(INFIL.EQ.'TAPE1')GO TO 4021
         OUTFIL=INFIL
      ENDIF
      OPEN(NEW,FILE=OUTFIL)
      IF(INFIL.EQ.'TAPE1') THEN
         WRITE(5,4040) OUTFIL
 4040    FORMAT(1H ,/10HNEW FILE  ,A8,17H  WILL BE CREATED)
      ELSE
         WRITE(5,4050)INFIL,OUTFIL
 4050    FORMAT(1H ,/17HCHANGES TO FILE  ,A8,19H  WILL BE ON FILE  ,A8)
      ENDIF
   10 CONTINUE
      IF (NEWFLG.EQ.TRU) GO TO 40
      WRITE (OUT,99994)
      READ (INP,99995,END=10001,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10001 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 30
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 30
      ICHR = BUF(IPTR)
      IF (ICHR.NE.WYE) GO TO 20
      WRITE (OUT,99993)
      CALL PRVIEW
      IPRVU = TRU
      GO TO 40
   20 CONTINUE
      IF (ICHR.EQ.ENN) GO TO 30
      WRITE (OUT,99992)
      GO TO 10
   30 CONTINUE
      WRITE (OUT,99991)
   40 CONTINUE
C     WRITE (OUT,99990)
C     READ (INP,99995,END=10002,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
C10002 IF(IEOF.NE.FALS) THEN
C        REWIND INP
C        GO TO 60
C     ENDIF
C     IPTR = ISKIPB(BUF,LINLMT,1)
C     IF (IPTR.GT.LINLMT) GO TO 60
C     ICHR = BUF(IPTR)
C     IF (ICHR.NE.WYE) GO TO 50
C     WRITE (OUT,99993)
C     CALL EDSITE(IPRVU)
C     ICHNGD = TRU
C     GO TO 70
C  50 CONTINUE
C     IF (ICHR.EQ.ENN) GO TO 60
C     WRITE (OUT,99992)
C     GO TO 40
C  60 CONTINUE
```

```
C     WRITE (OUT,99991)
   70 CONTINUE
      WRITE (OUT,99989)
      READ (INP,99995,END=10003,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10003 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 90
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 90
      ICHR = BUF(IPTR)
      IF (ICHR.NE.WYE) GO TO 80
      WRITE (OUT,99993)
      CALL EDCVRN
      ICHNGD = TRU
      GO TO 100
   80 CONTINUE
      IF (ICHR.EQ.ENN) GO TO 90
      WRITE (OUT,99992)
      GO TO 70
   90 CONTINUE
      WRITE (OUT,99991)
  100 CONTINUE
      IF (ICHNGD.EQ.FALS) GO TO 130
      WRITE (OUT,99988)
      READ (INP,99995,END=10004,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10004 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 120
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 120
      ICHR = BUF(IPTR)
      IF (ICHR.NE.WYE) GO TO 110
      WRITE (OUT,99993)
      CALL PRVIEW
      GO TO 130
  110 CONTINUE
      IF (ICHR.EQ.ENN) GO TO 120
      WRITE (OUT,99992)
      GO TO 100
  120 CONTINUE
      WRITE (OUT,99991)
  130 CONTINUE
      WRITE (OUT,99987)
      READ (INP,99995,END=10005,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10005 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 130
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 130
      ICHR = BUF(IPTR)
```

```
      ICHNGD=TRU
      IF (ICHR.NE.WYE) GO TO 160
      IF (ICHNGD.EQ.FALS) GO TO 180
      IF (NEWFLG.EQ.FALS) GO TO 140
      GO TO 150
  140 CONTINUE
  150 CONTINUE
      CALL DMPFIL
      GO TO 180
  160 CONTINUE
      IF (ICHR.EQ.ENN) GO TO 170
      WRITE (OUT,99992)
      GO TO 130
  170 CONTINUE
      WRITE (OUT,99991)
      GO TO 10
  180 CONTINUE
99999 FORMAT (//////////)
99998 FORMAT (1H , 24X, 27HSPR SCHEDULE EDITOR PROGRAM)
99997 FORMAT (1H , 33X, 12HVERSION FOUR)
99996 FORMAT (//49H=================================================)
99995 FORMAT (80A1)
99994 FORMAT (//1X, 31H FILE PREVIEW??(Y/N, N=DEFAULT))
99993 FORMAT (5H  YES)
99992 FORMAT (22H  ERROR: ILLEGAL INPUT)
99991 FORMAT (4H  NO)
99990 FORMAT (//1X,37H  ALTER SITE VALUES??(Y/N, N=DEFAULT))
99989 FORMAT (//1X,39H  ALTER CAVERN VALUES??(Y/N, N=DEFAULT))
99988 FORMAT (//1X,33H  FILE POSTVIEW??(Y/N, N=DEFAULT))
99987 FORMAT (//1X, 24H DONE WITH EDITOR??(Y/N))
99986 FORMAT (24H  OLD FILE BEING ALTERED)
99985 FORMAT (/1X,24H  NEW FILE BEING CREATED)
99984 FORMAT (//1X,26H NORMAL PROGRAM COMPLETION)
      CLOSE(NEW)
      RETURN
      END
C
C       ************ END OF MAIN BODY OF EDITOR *********************
C
C       ***************************************************************
C                              ** DATBLK **
C
C               THIS IS NOT A SUBPROGRAM. THIS IS A DATA BLOCK FOR THE
C       PURPOSE OF INITIALIZING VIA A CENTRALIZED PIECE OF CODE. ALL
C       FIRST TIME DEFAULT VALUES AND CONDITIONS ARE LOCATED HERE FOR
C       EASY ACCESS AND MODIFICATION.
C
C       ***************************************************************
C
      BLOCK DATA BLOKDT
      CHARACTER WYE, ENN, QMARK, BUFFER(80)
      INTEGER STDIN, STDOUT, OLDFIL, NEWFIL
      INTEGER TRU, FALS
```

```
      COMMON /INPUT/ LINLMT
      COMMON /STATS/ IENDBP(15), BPRS(15), BPRMIN, BPRMAX
      COMMON /PARAMS/ IDNUM, MODE, CURRCV, CURROV, FINLOV, FOVMIN,
     * FOVMAX
      COMMON /ALPHAS/ WYE, ENN, QMARK, BUFFER
      COMMON /DELAYS/ IWODLY(7), IWOSLF(7), IWOLLF(7), IWOSLT(7),
     * IWOLLT(7)
      COMMON /ENDPTS/ PENDPV(7), DVCSLF(7,2), DVCLLF(7,2), DVCSLT(7,2),
     * DVCLLT(7,2), DVCSIF(7,2)
      COMMON /ACCUMS/ IACCUM(25,20), RACCUM(31,20)
      COMMON /LUNITS/ STDIN, STDOUT, OLDFIL, NEWFIL
      COMMON /SITECM/ FACTOR, ISLIP
      COMMON /LOGICS/ TRU, FALS
      COMMON /BLK1/ OILVOL(3)
      COMMON /BLK2/ FLRT(3)
      DATA LINLMT /80/
      DATA WYE, ENN, QMARK /'Y','N','?'/
      DATA DVCSIF /0.0,3103.0,3897.0,3400.0,0.0,800.0,0.0,0.0,310.3,
     * 389.7,340.0,0.0,80.0,0.0/
      DATA DVCSLT /0.0,3108.0,2016.0,1999.0,1760.0,2317.0,0.0,0.0,310.8,
     * 201.6,199.9,176.0,231.7,0.0/
      DATA DVCLLT /0.0,3516.0,2937.0,2599.0,2453.0,3195.0,0.0,0.0,
     * 264.3,221.0,195.3,184.3,240.3,0.0/
      DATA DVCSLF /0.0,2513.0,3186.0,1359.0,2507.0,817.5,817.5,0.0,
     * 251.3,318.6,135.9,250.7,81.75,81.75/
      DATA DVCLLF /0.0,3386.0,4027.0,1745.0,3259.5,1141.25,1141.25,
     * 0.0,254.7,302.7,131.3,245.2,85.6,85.6/
      DATA PENDPV /0.0,2513.0,5699.0,7058.0,9565.0,10382.5,11200.0/
      DATA IWODLY /0,20,20,0,20,0,20/
      DATA IWOSLF /0,20,20,0,20,0,20/
      DATA IWOLLF /0,20,20,0,20,0,20/
      DATA IWOSLT /0,20,20,0,20,0,20/
      DATA IWOLLT /0,20,20,0,20,0,20/
      DATA BPRS /150.0,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0./
      DATA OILVOL /2000.,6000.,10000./
      DATA FLRT /50.0,50.0,50.0/
      DATA IENDBP /-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1/
      DATA STDIN, STDOUT /5,6/
      DATA OLDFIL, NEWFIL /1,1/
      DATA FACTOR, ISLIP /0.0,0/
      DATA MODE, CURRCV, CURROV, FINLOV /0,0.0,0.0,10000.0/
      DATA BPRMIN, BPRMAX /50.0,225.0/
      DATA FOVMIN, FOVMAX /10000.0,15000.0/
      DATA TRU, FALS /-1,0/
      END
C
C       ***********END OF .DATA. BLOCK ******************************
C
C       ************************************************************
C                              ** INFILL **
C
C               THIS ROUTINE READS THE 'OLD' FILE INTO THE ARRAY
C       ACCUMULATORS.
```

```
C
C       ************************************************************
C
            SUBROUTINE INFILL(NEWFLG,INFIL)
      INTEGER NEWFLG, OUT, OLD
      INTEGER TRU, FALS
      CHARACTER*8 INFIL
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /ACCUMS/ IACCUM(25,20), RACCUM(31,20)
      COMMON /SITECM/ FACTOR, ISLIP
      COMMON /BLK1/ OILVOL(3)
      COMMON /BLK2/ FLRT(3)
      DATA TRU, FALS /-1,0/
      REWIND OLD
      READ (OLD,99999,END=999,IOSTAT=IEOF) FACTOR,ISLIP
  999 IF (IEOF.NE.FALS) THEN
         REWIND OLD
         GO TO 20
      ENDIF
      WRITE (OUT,99994)INFIL
      DO 10 KEY=1,20
         READ (OLD,99998) (IACCUM(I,KEY),I=1,12)
         READ (OLD,99997) (IACCUM(I,KEY),I=13,25)
         READ (OLD,99996) (RACCUM(I,KEY),I=1,8)
         READ (OLD,99996) (RACCUM(I,KEY),I=9,16)
         READ (OLD,99996) (RACCUM(I,KEY),I=17,24)
         READ (OLD,99995) (RACCUM(I,KEY),I=25,31)
   10 CONTINUE
      NEWFLG = FALS
      GO TO 60
   20 CONTINUE
      WRITE (OUT,99993)
      NEWFLG = TRU
      DO 50 KEY=1,20
         DO 30 I=1,25
            IACCUM(I,KEY) = 0
   30    CONTINUE
         DO 40 I=1,31
            RACCUM(I,KEY) = 0.0
   40    CONTINUE
   50 CONTINUE
   60 CONTINUE
99999 FORMAT (F6.2, I6)
99998 FORMAT (12I6)
99997 FORMAT (13I6)
99996 FORMAT (8F9.2)
99995 FORMAT (7F9.2)
99994 FORMAT (17H  OLD SITE FILE  ,A8,16H  WILL BE EDITED)
99993 FORMAT (31H  NEW SITE FILE WILL BE CREATED)
      RETURN
      END
C
C       *************** END OF INFILL ROUTINE ************************
```

```
C
C       ************************************************************
C                              ** PRVIEW **
C
C               THIS ROUTINE PRINTS OUT A FORMATTED LISTING OF THE
C       CAVERN ACCUMULATORS REQUESTED INTERACTIVELY.
C
C       ************************************************************
C
            SUBROUTINE PRVIEW
      INTEGER ERR, OUT
      INTEGER ALLFLG, CUTFLG
      INTEGER TRU, FALS
      CHARACTER BUF(80),WYE,ENN,QMARK
      DIMENSION ICHOOS(20)
      COMMON /INPUT/ LINLMT
      COMMON /ALPHAS/ WYE, ENN, QMARK, BUF
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /ACCUMS/ IACCUM(25,20), RACCUM(31,20)
      COMMON /SITECM/ FACTOR, ISLIP
      COMMON /LOGICS/ TRU, FALS
   10 CONTINUE
      DO 20 I=1,20
         ICHOOS(I) = FALS
   20 CONTINUE
      WRITE (OUT,99999)
      READ (INP,99998,END=10006,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10006 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 10
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 30
      CUTFLG = FALS
      ALLFLG = IALL(IPTR)
      IF (ALLFLG.EQ.TRU) GO TO 40
      CALL GETCVS(IPTR, ICHOOS, ERR)
      IF (ERR.EQ.TRU) GO TO 10
      GO TO 60
   30 CONTINUE
      CUTFLG = TRU
      GO TO 60
   40 CONTINUE
      DO 50 I=1,20
         ICHOOS(I) = TRU
   50 CONTINUE
   60 CONTINUE
C     WRITE (OUT,99997) FACTOR
C     WRITE (OUT,99996) ISLIP
      IF (CUTFLG.EQ.TRU) GO TO 90
      KNT = 1
   70 CONTINUE
      IF (KNT.GT.20) GO TO 90
```

```
      IF (ICHOOS(KNT).EQ.FALS) GO TO 80
      CALL PRNTCV(KNT)
   80 CONTINUE
      KNT = KNT + 1
      GO TO 70
   90 CONTINUE
99999 FORMAT (//1X,24H  ENTER CAVERN#S OR ALL:)
99998 FORMAT (80A1)
99997 FORMAT (/1X,19H  CONTING. FACT. = , F5.2, 2H %)
99996 FORMAT (21H  DAYS INTO SCHED =  , I5)
      RETURN
      END
C
C       ************** END OF PRVIEW ROUTINE ************************
C
C       ************************************************************
C                              ** GETCVS **
C
C               THIS ROUTINE PERFORMS THE INTERACTION NEEDED TO INPUT
C       THE CAVERN NUMBERS REQUESTED FOR PRVIEW PRINTOUT.
C
C       ************************************************************
C
            SUBROUTINE GETCVS(IPTR, ICHOOS, ERR)
      INTEGER ERR, OUT
      INTEGER TRU, FALS
      CHARACTER WYE,ENN,QMARK,BUF(80)
      DIMENSION ICHOOS(1)
      COMMON /INPUT/ LINLMT
      COMMON /ALPHAS/ WYE,ENN,QMARK,BUF
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /LOGICS/ TRU, FALS
   10 CONTINUE
      ITMP = IDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 30
      ITMP = ITMP - 100
      IF ((ITMP.GT.0) .AND. (ITMP.LE.20)) GO TO 20
      WRITE (OUT,99999)
      ERR = TRU
      GO TO 30
   20 CONTINUE
      ICHOOS(ITMP) = TRU
      IPTR = ISKIPB(BUF,LINLMT,IPTR)
      IF (IPTR.GT.LINLMT) GO TO 30
      GO TO 10
   30 CONTINUE
99999 FORMAT (31H  ERROR:  ILLEGAL CAVERN NUMBER)
      RETURN
      END
C
C       ************* END OF GETCVS ROUTINE *************************
C
C       ************************************************************
```

```
C                          ** PRNTCV **
C
C              THIS ROUTINE PRINTS, IN FORMATTED OUTPUT, THE INITIAL
C       CONDITIONS FOR A SINGLE SELECTED CAVERN SPECIFIED BY THE
C       ARGUMENT 'KEY'.
C
C       ************************************************************
C
            SUBROUTINE PRNTCV(KEY)
      INTEGER OUT
      INTEGER TRU
      COMMON /ACCUMS/ IACCUM(25,20), RACCUM(31,20)
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /LOGICS/ TRU
      WRITE (OUT,99997)
      ITMP = KEY + 100
      IF (IACCUM(1,KEY).EQ.TRU) GO TO 10
      WRITE (OUT,99999) ITMP
      GO TO 60
   10 CONTINUE
      WRITE (OUT,99998) ITMP
      IF (IACCUM(2,KEY).EQ.0) WRITE (OUT,99995)
      IF (IACCUM(2,KEY).EQ.1) WRITE (OUT,99996)
      IF (IACCUM(2,KEY).EQ.2) WRITE (OUT,99994)
      WRITE (OUT,99991) RACCUM(1,KEY)
      WRITE (OUT,99990) RACCUM(2,KEY)
      WRITE (OUT,99989) RACCUM(3,KEY)
      I = 1
   20 CONTINUE
      IF (IACCUM(I+3,KEY).GE.0) GO TO 30
      WRITE (OUT,99986) I, RACCUM(I+3,KEY)
      GO TO 35
   30 CONTINUE
      WRITE (OUT,99985) I, RACCUM(I+3,KEY), IACCUM(I+3,KEY)
      I = I + 1
      IF (I.LE.15) GO TO 20
   35 CONTINUE
      IF (IACCUM(2,KEY).EQ.2) GO TO 40
      WRITE (OUT,99981) RACCUM(29,KEY)
      GO TO 50
   40 CONTINUE
      WRITE (OUT,99980) RACCUM(29,KEY)
      WRITE (OUT,99979) RACCUM(30,KEY)
      WRITE (OUT,99978) RACCUM(31,KEY)
   50 CONTINUE
      WRITE (OUT,99992)
      WRITE (OUT,99993) (RACCUM(I,KEY),I=19,25)
      WRITE (OUT,99988) (IACCUM(I,KEY),I=19,25)
      IF (IACCUM(2,KEY).EQ.2) WRITE (OUT,99987) (RACCUM(I,KEY),I=26,28)
      IF (IACCUM(2,KEY) .NE. 0) GO TO 60
      COV = CAVOIL(RACCUM(3,KEY),RACCUM(1,KEY),IFIN)
      WRITE (OUT,99984) COV
      IF (IFIN.EQ.1) WRITE (OUT,99983)
```

```
      IF (IFIN.EQ.0) WRITE (OUT,99982)
   60 CONTINUE
99999 FORMAT (1H , I3, 12H IS INACTIVE)
99998 FORMAT (1H , I3, 10H IS ACTIVE)
99997 FORMAT (//)
99996 FORMAT (16H     MODE IS LTF)
99995 FORMAT (16H     MODE IS L/F)
99994 FORMAT (16H     MODE IS I/F)
99993 FORMAT (12H      EVS = , 7F9.2, 3H MB)
99992 FORMAT (//15X, 4HSUMP, 3X, 7HSUMP-CH, 4X, 4HREV1, 3X, 6HREV2-1,
     * 3X, 6HREV2-2, 3X, 6HREV3-1, 3X, 6HREV3-2)
99991 FORMAT (/1X,11H     CCV = , F8.2, 3H MB)
99990 FORMAT (/1X,11H     COV = , F8.2, 3H MB)
99989 FORMAT (/1X,11H     FOV = , F8.2, 3H MB)
99988 FORMAT (13H     DELAYS =, I5, 6I9, 7H   DAYS)
99987 FORMAT (14H     OIL VOL =, 15X, 2F9.2, 9X, F9.2, 12H          MB)
99986 FORMAT (9H     BPR#, I2, 3H = , F8.2, 24H,   *   (MB/DAY,LASTDAY#,
     * 1H))
99985 FORMAT (9H     BPR#, I2, 3H = , F8.2, 1H,, I6, 15H (MB/DAY,LASTDA,
     * 3HY#))
99984 FORMAT (//1X, 33H     THE FORMAL SCHEDULE PREDICTS, F8.2,
     * 10H MB OF OIL)
99983 FORMAT (1H , 29H     OR GREATER IN THE CAVERN/)
99982 FORMAT (1H , 18H     IN THE CAVERN/)
99981 FORMAT (1H /20H     OIL FILL RATE =,F8.2,3H MB)
99980 FORMAT (1H /23H     OIL FILL RATE #1 =,F8.2,3H MB)
99979 FORMAT (1H ,22H    OIL FILL RATE #2 =,F8.2,3H MB)
99978 FORMAT (1H ,22H    OIL FILL RATE #3 =,F8.2,3H MB)
      RETURN
      END
C
C       ************* END OF PRNTCV ROUTINE *************************
C
C       ************************************************************
C                              ** EDSITE **
C
C       ************************************************************
C
            SUBROUTINE EDSITE(IPRVU)
      INTEGER INP, OUT, OLD, NEW
      INTEGER TRU, FALS, ERR
      CHARACTER WYE,ENN,QMARK,BUF(80)
      COMMON /INPUT/ LINLMT
      COMMON /ALPHAS/ WYE,ENN,QMARK,BUF
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /SITECM/ FACTOR, ISLIP
      COMMON /ACCUMS/ IACCUM(25,20), RACCUM(31,20)
      COMMON /LOGICS/ TRU, FALS
      IF (IPRVUE.EQ.TRU) GO TO 10
      WRITE (OUT,99999) FACTOR
      WRITE (OUT,99998) ISLIP
   10 CONTINUE
      WRITE (OUT,99996)
```

```
      READ (INP,99997,END=10007,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10007 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 30
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 30
      RTMP = RDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 10
      IF ((RTMP.GE.0.0) .AND. (RTMP.LE.99.0)) GO TO 20
      WRITE (OUT,99994)
      GO TO 10
   20 CONTINUE
      FACTOR = RTMP
   30 CONTINUE
      WRITE (OUT,99995) FACTOR
   40 CONTINUE
      WRITE (OUT,99993)
      READ (INP,99997,END=10008,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10008 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 60
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 60
      ITMP = IDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 40
      IF (ITMP.GE.0) GO TO 50
      WRITE (OUT,99991)
      GO TO 40
   50 CONTINUE
      ISLIP = ITMP
   60 CONTINUE
      WRITE (OUT,99992) ISLIP
99999 FORMAT (/1X,18H CONTING. FACT. = , F5.2, 2H %)
99998 FORMAT (/1X,22H   DAYS INTO SCHED =  , I5)
99997 FORMAT (80A1)
99996 FORMAT (//1X,28H FACTOR??(DEFAULT=NO CHANGE))
99995 FORMAT (2H  , F6.2, 8H PERCENT)
99994 FORMAT (38H  RANGE ERROR: MUST BE [ 0.0 TO 99.0 ])
99993 FORMAT (/1X,40H DAYS INTO SCHEDULE??(DEFAULT=NO CHANGE))
99992 FORMAT (2H  , I5, 5H DAYS)
99991 FORMAT (51H  ERROR: MUST BE GREATER THAN OR EQUAL TO ZERO DAYS)
      RETURN
      END
C
C       ************** END OF EDSITE SUBROUTINE ***********************
C
C       ************************************************************
C                              ** EDCVRN **
C
C              THIS ROUTINE CONTROLS THE EDITING INTERACTION FOR THE
C       PARAMETERS THAT DESCRIBE THE INITIAL CONDITIONS OF A CAVERN
```

```
C       AND ITS LEACH SCHEDULE.
C
C       ************************************************************
C
            SUBROUTINE EDCVRN
      CHARACTER WYE, ENN, QMARK, BUF(80),ICHAR
      INTEGER INP, OUT, OLD, NEW
      INTEGER ALLFLG,ACTV,ERR
      INTEGER TRU, FALS
      COMMON /INPUT/ LINLMT
      COMMON /ALPHAS/ WYE, ENN, QMARK, BUF
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /PARAMS/ IDNUM, MODE, CURRCV, CURROV, FINLOV, FOVMIN,
     * FOVMAX
      COMMON /ACCUMS/ IACCUM(25,20)
      COMMON /LOGICS/ TRU, FALS
      COMMON /BLK1/ OILVOL(3)
      COMMON /BLK2/ FLRT(3)
   10 CONTINUE
      WRITE (OUT,99998)
      READ (INP,99999,END=10009,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10009 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 200
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 200
      ITMP = IDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 10
      IF ((ITMP.GE.101) .AND. (ITMP.LE.120)) GO TO 20
      WRITE (OUT,99996)
      GO TO 10
   20 CONTINUE
      IDNUM = ITMP - 100
      WRITE (OUT,99997) ITMP
   30 CONTINUE
      WRITE (OUT,99995) ITMP
      READ (INP,99999,END=10010,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10010 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 50
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 50
      ICHAR = BUF(IPTR)
      IF (ICHAR.EQ.WYE) GO TO 40
      IF (ICHAR.EQ.ENN) GO TO 50
      WRITE (OUT,99994)
      GO TO 30
   40 CONTINUE
      ACTV = FALS
      WRITE (OUT,99993) ITMP
      GO TO 190
```

```
   50 CONTINUE
      WRITE (OUT,99992)
      IF (IACCUM(1,IDNUM).EQ.FALS) GO TO 60
      CALL LOADCV
   60 CONTINUE
      ACTV = TRU
      IWARN1 = FALS
      IWARN2 = FALS
      IWARN3 = FALS
   70 CONTINUE
      WRITE (OUT,99991)
      READ (INP,99999,END=10011,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10011 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 80
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 80
      ALLFLG = IALL(IPTR)
      IF (ALLFLG.EQ.TRU) GO TO 110
      IF (BUF(IPTR).EQ.QMARK) GO TO 80
      ITMP = IDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 70
      IF ((ITMP.EQ.6) .AND. (MODE.NE.2)) GO TO 90
      IF ((ITMP.GE.0) .AND. (ITMP.LE.7)) GO TO 100
      WRITE (OUT,99988)
   80 CONTINUE
      CALL CVHELP
      GO TO 70
   90 CONTINUE
      WRITE (OUT,99986)
      WRITE (OUT,99985)
      GO TO 70
  100 CONTINUE
      WRITE (OUT,99990) ITMP
      GO TO 120
  110 CONTINUE
      WRITE (OUT,99989)
  120 CONTINUE
      IF (ITMP.NE.0) GO TO 130
      IF ((IWARN1.EQ.FALS) .AND. (IWARN2.EQ.FALS) .AND.
     *    (IWARN3.EQ.FALS)) GO TO 190
      WRITE (OUT,99987)
      GO TO 70
  130 CONTINUE
      IF ((ALLFLG.EQ.FALS) .AND. (ITMP.NE.1)) GO TO 140
      CALL EDMODE(IWARN1, IWARN2, IWARN3)
  140 CONTINUE
      IF ((ALLFLG.EQ.FALS) .AND. (ITMP.NE.2)) GO TO 150
      CALL EDEPVS(IWARN1, IWARN2)
  150 CONTINUE
      IF ((ALLFLG.EQ.FALS) .AND. (ITMP.NE.3)) GO TO 160
      CALL EDVOLS(IWARN2, IWARN3)
```

```
  160 CONTINUE
      IF ((ALLFLG.EQ.FALS) .AND. (ITMP.NE.4)) GO TO 170
      CALL EDDLYS
  170 CONTINUE
      IF ((ALLFLG.EQ.FALS) .AND. (ITMP.NE.5)) GO TO 180
      CALL EDBPRS
  180 CONTINUE
      IF ((ALLFLG.EQ.FALS) .AND. (ITMP.NE.6)) GO TO 185
      IF (MODE.NE.2) GO TO 185
      WRITE (OUT,99984)
      WRITE (OUT,99983)
      WRITE (OUT,99982)
      WRITE (OUT,99981)
      CALL EDOIL
  185 CONTINUE
      IF ((ALLFLG.EQ.FALS) .AND. (ITMP.NE.7)) GO TO 70
      CALL FWOIL
      GO TO 70
  190 CONTINUE
      CALL SAVECV(ACTV)
      GO TO 10
  200 CONTINUE
 1000 FORMAT (//)
99999 FORMAT (80A1)
99998 FORMAT (//1X,31H CAVERN ID# ??(DEFAULT=NO MORE))
99997 FORMAT (8H  ID# = , I3)
99996 FORMAT (37H  RANGE ERROR: MUST BE [ 101 TO 120 ])
99995 FORMAT (//1X,16H DELETE CAVERN# , I3,19H ??(Y/N, N=DEFAULT))
99994 FORMAT (34H  INPUT ERROR: INCORRECT CHARACTER)
99993 FORMAT (10H  CAVERN #, I3, 8H DELETED)
99992 FORMAT (4H  NO)
99991 FORMAT (//1X,44H ENTER ? OR OPT# OR ALL:(DEFAULT=?, FOR HELP,
     *            13H ---- 0=EXIT))
99990 FORMAT (2H  , I1)
99989 FORMAT (5H  ALL)
99988 FORMAT (32H  RANGE ERROR:  MUST BE <0 TO 6!)
99987 FORMAT (46H  ERROR:  UNRESOLVED PARAMETER CONFLICTS EXIST)
99986 FORMAT (//1X, 39HTHIS IS NOT AN INTERMITTENT FILL CAVERN)
99985 FORMAT (39H  YOU CANNOT USE THIS OPTION. TRY AGAIN)
99984 FORMAT (/46H  WARNING!!! I/F OIL FILL VOLUMES ARE INTIMATELY)
99983 FORMAT (51H  DEPENDENT UPON HAVING THE PROPER STATE ENDVOLUMES)
99982 FORMAT (47H  IF YOU CHANGE OIL FILL VOLUMES YOU SHOUD ALSO)
99981 FORMAT (49H  RECOMPUTE STAGE ENDVOLUMES IN PROPER PROPORTION)
99980 FORMAT (//1X,27HTHIS IS A LEACH/FILL CAVERN)
      RETURN
      END
C
C       *********** END OF EDCVRN SUBROUTINE ************************
C
C       ************************************************************
C                              ** LOADCV **
C       ************************************************************
C
```

```
      SUBROUTINE LOADCV
      COMMON /PARAMS/ IDNUM, MODE, CURRCV, CURROV, FINLOV
      COMMON /ENDPTS/ PENDPV(7), DVCSLF(7,2), DVCLLF(7,2), DVCSLT(7,2),
     * DVCLLT(7,2), DVCSIF(7,2)
      COMMON /STATS/ IENDBP(15), BPRS(15), BPRMIN, BPRMAX
      COMMON /DELAYS/ IWODLY(7), IWOSLF(7), IWOLLF(7), IWOSLT(7),
     * IWOLLT(7)
      COMMON /ACCUMS/ IACCUM(25,20), RACCUM(31,20)
      COMMON /BLK1/ OILVOL(3)
      COMMON /BLK2/ FLRT(3)
      KEY = IDNUM
      MODE = IACCUM(2,KEY)
      CURRCV = RACCUM(1,KEY)
      CURROV = RACCUM(2,KEY)
      FINLOV = RACCUM(3,KEY)
      DO 10 I=1,7
         IWODLY(I) = IACCUM(I+18,KEY)
         PENDPV(I) = RACCUM(I+18,KEY)
   10 CONTINUE
      DO 20 I=1,15
         IENDBP(I) = IACCUM(I+3,KEY)
         BPRS(I) = RACCUM(I+3,KEY)
   20 CONTINUE
      DO 30 I=1,3
         OILVOL(I) = RACCUM(I+25,KEY)
         FLRT(I) = RACCUM(I+28,KEY)
   30 CONTINUE
      RETURN
      END
C
C       *********** END OF LOADCV ROUTINE *************************
C
C       ************************************************************
C                              ** CVHELP **
C       ************************************************************
C
      SUBROUTINE CVHELP
      INTEGER OUT
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      WRITE (OUT,99999)
      WRITE (OUT,99998)
      WRITE (OUT,99997)
      WRITE (OUT,99996)
      WRITE (OUT,99995)
      WRITE (OUT,99994)
      WRITE (OUT,99993)
      WRITE (OUT,99990)
      WRITE (OUT,99992)
      WRITE (OUT,99991)
99999 FORMAT (//1X, 5X, 14H0   ----- EXIT)
99998 FORMAT (1H , 5X, 20H1   ----- MODE + FOV)
99997 FORMAT (1H , 5X, 18H2   ----- END VOLS)
99996 FORMAT (1H , 5X, 22H3   ----- CURRENT VOLS)
```

```
99995 FORMAT (1H , 5X, 19H4   ----- WORKOVERS)
99994 FORMAT (1H , 5X, 22H5   ----- RPR SCHEDULE)
99993 FORMAT (1H , 5X, 22H6   ----- I/F OIL VOLS)
99990 FORMAT (1H , 5X, 24H7   ----- OIL FILL RATES)
99992 FORMAT (1H , 5X, 18HALL ----- 1 THRU 7)
99991 FORMAT (1H , 5X, 18H?   ----- OPT HELP)
      RETURN
      END
C
C       **************** END OF CVHELP ROUTINE ************************
C
C       ************************************************************
C                              ** EDMODE **
C       ************************************************************
C
            SUBROUTINE EDMODE(IWARN1, IWARN2, IWARN3)
      INTEGER INP, OUT, OLD, NEW
      INTEGER TRU, FALS, ERR
      CHARACTER WYE, ENN, QMARK, BUF(80), ICHR
      COMMON /INPUT/ LINLMT
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /ENDPTS/ PENDPV(7), DVCSLF(7,2), DVCLLF(7,2), DVCSLT(7,2),
     * DVCLLT(7,2), DVCSIF(7,2)
      COMMON /DELAYS/ IWODLY(7), IWOSLF(7), IWOLLF(7), IWOSLT(7),
     * IWOLLT(7)
      COMMON /PARAMS/ IDNUM, MODE, CURRCV, CURROV, FINLOV, FOVMIN,
     * FOVMAX
      COMMON /LOGICS/ TRU, FALS
      COMMON /ALPHAS/ WYE, ENN,QMARK,BUF
      COMMON /BLK1/ OILVOL(3)
      ICHNGD = FALS
      IWARN1 = FALS
      IWARN2 = FALS
      IWARN3 = FALS
      IF (MODE.EQ.0) WRITE (OUT,99998)
      IF (MODE.EQ.1) WRITE (OUT,99999)
      IF (MODE.EQ.2) WRITE (OUT,99997)
   10 CONTINUE
      WRITE (OUT,99995)
      READ (INP,99996,END=10012,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10012 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 30
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 30
      ITMP = IDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 10
      IF ((ITMP.GE.0) .OR. (ITMP.LE.2)) GO TO 20
      WRITE (OUT,99991)
      GO TO 10
   20 CONTINUE
      IF (MODE.EQ.ITMP) GO TO 30
```

```
      MODE = ITMP
      ICHNGD = TRU
   30 CONTINUE
      IF (MODE.EQ.0) WRITE (OUT,99993)
      IF (MODE.EQ.1) WRITE (OUT,99994)
      IF (MODE.EQ.2) WRITE (OUT,99992)
   40 CONTINUE
      WRITE (OUT,99990) FINLOV
   50 CONTINUE
      WRITE (OUT,99989)
      READ (INP,99996,END=10013,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10013 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 70
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 70
      RTMP = RDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 50
      IF (RTMP.GE.0.0) GO TO 60
      WRITE (OUT,99988)
      GO TO 50
   60 CONTINUE
      IF (RTMP.EQ.FINLOV) GO TO 70
      ICHNGD = TRU
      FINLOV = RTMP
   70 CONTINUE
      WRITE (OUT,99985) FINLOV
      IF (FINLOV.GE.CURROV) GO TO 80
      WRITE (OUT,99987)
      IWARN3 = TRU
   80 CONTINUE
      IF (FINLOV.LE.PENDPV(7)) GO TO 90
      WRITE (OUT,99986)
      IWARN1 = TRU
   90 CONTINUE
      IF (MODE.EQ.2) GO TO 100
      IF (MODE.NE.1) GO TO 140
      IF (FINLOV.GE.13000.0) GO TO 120
  100 CONTINUE
      DO 110 I=1,7
         IWODLY(I) = IWOSLT(I)
  110 CONTINUE
      GO TO 180
  120 CONTINUE
      DO 130 I=1,7
         IWODLY(I) = IWOLLT(I)
  130 CONTINUE
      GO TO 180
  140 CONTINUE
      IF (FINLOV.GE.13000.0) GO TO 160
      DO 150 I=1,7
         IWODLY(I) = IWOSLF(I)
```

```
  150 CONTINUE
      GO TO 180
  160 CONTINUE
      DO 170 I=1,7
         IWODLY(I) = IWOLLF(I)
  170 CONTINUE
  180 CONTINUE
      WRITE (OUT,99976)
      WRITE (OUT,99974)
      WRITE (OUT,99975) (IWODLY(I),I=1,7)
      GO TO 200
  190 CONTINUE
      IF (IWARN1.EQ.FALS) GO TO 270
  200 CONTINUE
      WRITE (OUT,99982)
      READ (INP,99996,END=10014,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10014 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 210
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 210
      ICHR = BUF(IPTR)
      IF (ICHR.EQ.WYE) GO TO 220
      IF (ICHR.EQ.ENN) GO TO 210
      WRITE (OUT,99981)
      GO TO 200
  210 CONTINUE
      WRITE (OUT,99980)
      GO TO 240
  220 CONTINUE
      WRITE (OUT,99979)
      IF ((FINLOV.GE.FOVMIN) .AND. (FOVMAX.LE.FOVMAX)) GO TO 230
      WRITE (OUT,99984) FOVMIN
      WRITE (OUT,99983) FOVMAX
      GO TO 40
  230 CONTINUE
      CALL ENDVOL
      IWARN1 = FALS
      IF (PENDPV(7).GE.CURRCV) GO TO 240
      WRITE (OUT,99978)
      WRITE (OUT,99977)
      IWARN2 = TRU
  240 CONTINUE
      IF (MODE.NE.2) GO TO 270
      WRITE (OUT,99973)
      WRITE (OUT,99970)
      WRITE (OUT,99969)
      READ (INP,99996,END=10015,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10015 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 250
      ENDIF
```

```
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 250
      ICHR = BUF(IPTR)
      IF (ICHR.EQ.WYE) GO TO 260
      IF (ICHR.EQ.ENN) GO TO 250
      WRITE (OUT,99981)
      GO TO 240
  250 CONTINUE
      WRITE (OUT,99980)
      GO TO 270
  260 CONTINUE
      WRITE (OUT,99979)
      OILVOL(1) = 2000.*PENDPV(3)/7000.
      OILVOL(2) = 6000.*PENDPV(4)/10400.
      OILVOL(3) = FINLOV
      WRITE (OUT,99972)
      WRITE (OUT,99974)
      WRITE (OUT,99971) (OILVOL(I),I=1,3)
  270 CONTINUE
99999 FORMAT (//1X,22H THE LEACH MODE IS LTF)
99998 FORMAT (//1X,22H THE LEACH MODE IS L/F)
99997 FORMAT (//1X,22H THE LEACH MODE IS I/F)
99996 FORMAT (80A1)
99995 FORMAT (/1X,50H MODE ??  <0=L/F, 1=LTF, 2=I/F>(DEFAULT=NO CHANGE))
99994 FORMAT (17H  THE MODE IS LTF)
99993 FORMAT (17H  THE MODE IS L/F)
99992 FORMAT (17H  THE MODE IS I/F)
99991 FORMAT (41H  RANGE ERROR: ONLY 0, 1, OR 2 IS ALLOWED)
99990 FORMAT (//1X,7H FOV = , F8.2, 3H MB)
99989 FORMAT (//25H FOV??(DEFAULT=NO CHANGE))
99988 FORMAT (40H  RANGE ERROR: MUST BE GREATER THAN ZERO)
99987 FORMAT (47H  WARNING: MUST BE > OR = TO CURRENT OIL VOLUME)
99986 FORMAT (48H  WARNING: MUST BE < OR = TO FINAL CAVERN VOLUME)
99985 FORMAT (2H  , F8.2, 3H MB)
99984 FORMAT (36H  RANGE ERROR: FOV MUST BE > OR = TO, F8.2)
99983 FORMAT (36H                        AND < OR = TO, F8.2)
99982 FORMAT (/1X,46H DO YOU WANT THE STAGE ENDVOLUMES RECOMPUTED??/
     *        20X,16H(Y/N, N=DEFAULT))
99981 FORMAT (22H  ERROR: ILLEGAL INPUT)
99980 FORMAT (4H  NO)
99979 FORMAT (5H  YES)
99978 FORMAT (44H  WARNING: FINAL ENDVOLUME MUST BE > OR = TO)
99977 FORMAT (38H                  CURRENT CAVERN VOLUME)
99976 FORMAT (//1X,25H   THE NEW WORKOVERS ARE:)
99975 FORMAT (4H    , 7I9, 6H  DAYS)
99974 FORMAT (1H , 8X, 4HSUMP, 3X, 7HSUMP-CH, 4X, 4HREV1, 3X, 6HREV2-1,
     *            3X, 6HREV2-2, 3X, 6HREV3-1, 3X, 6HREV3-2)
99973 FORMAT (/51H  DO YOU WANT INTERMITTENT OIL VOLUMES RECOMPUTED??/
     *        20X,16H(Y/N, N=DEFAULT))
99972 FORMAT (//1X,36H  THE NEW INTERMITTENT OIL VOLS ARE:)
99971 FORMAT (11H  OIL VOL =, 12X, 2F9.2, 9X, F9.2,12H          MB)
99970 FORMAT (50H  (NOTE: IF YOU HAVE CHANGED THE STAGE ENDVOLUMES,)
99969 FORMAT (45H  YOU SHOULD ALSO CHANGE THE I/F OIL VOLUMES))
```

```
      RETURN
      END
C
C       *********** END OF EDMODE SUBROUTINE ************************
C
C       ************************************************************
C                              ** ENDVOL **
C       ************************************************************
C
            SUBROUTINE ENDVOL
      INTEGER INP, OUT, OLD, NEW
      REAL VCTEMP(7,2)
      COMMON /PARAMS/ IDNUM, MODE, CURRCV, CURROV, FINLOV
      COMMON /ENDPTS/ PENDPV(7), DVCSLF(7,2), DVCLLF(7,2), DVCSLT(7,2),
     * DVCLLT(7,2), DVCSIF(7,2)
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      SIZE = (FINLOV-10000.0)*0.001
      IF (MODE.EQ.0) GO TO 40
      IF (MODE.EQ.2) GO TO 80
      IF (SIZE.GE.3.0) GO TO 20
      PENDPV(1) = DVCSLT(1,1)
      DO 10 I=2,7
         PENDPV(I) = PENDPV(I-1) + DVCSLT(I,1) + DVCSLT(I,2)*SIZE
   10 CONTINUE
      GO TO 100
   20 CONTINUE
      SIZE = (FINLOV-13300.)*.001
      PENDPV(1) = DVCLLT(1,1)
      DO 30 I=2,7
         PENDPV(I) = PENDPV(I-1) + DVCLLT(I,1) + DVCLLT(I,2)*SIZE
   30 CONTINUE
      GO TO 100
   40 CONTINUE
      IF (SIZE.GE.3.0) GO TO 60
      PENDPV(1) = DVCSLF(1,1)
      DO 50 I=2,7
         PENDPV(I) = PENDPV(I-1) + DVCSLF(I,1) + DVCSLF(I,2)*SIZE
   50 CONTINUE
      GO TO 100
   60 CONTINUE
      SIZE = (FINLOV-13300.)*.001
      PENDPV(1) = DVCLLF(1,1)
      DO 70 I=2,7
         PENDPV(I) = PENDPV(I-1) + DVCLLF(I,1) + DVCLLF(I,2)*SIZE
   70 CONTINUE
      GO TO 100
   90 CONTINUE
       PENDPV(1) = DVCSIF(1,1)
      DO 90 I=2,7
         PENDPV(I) = PENDPV(I-1) + DVCSIF(I,1) + DVCSIF(I,2)*SIZE
   90 CONTINUE
  100 CONTINUE
      WRITE (OUT,99999)
```

```
      WRITE (OUT,99997)
      WRITE (OUT,99998) (PENDPV(I),I=1,7)
99999 FORMAT (/1X,26H   THE NEW ENDVOLUMES ARE:)
99998 FORMAT (5H     , 7F9.2, 3H MB)
99997 FORMAT (1H , 8X, 4HSUMP, 3X, 7HSUMP-CH, 4X, 4HREV1, 3X, 6HREV2-1,
     *            3X, 6HREV2-2, 3X, 6HREV3-1, 3X, 6HREV3-2)
      RETURN
      END
C
C       ************** END OF ENDVOL ROUTINE *************************
C
C       ************************************************************
C                                ** EDDLYS **
C
C               THIS ROUTINE PERFORMS THE EDIT INTERACTION FOR THE
C       WORKOVER DELAY PARAMETERS.
C
C       ************************************************************
C
            SUBROUTINE EDDLYS
      INTEGER TRU, FALS, ERR, OUT, OLD
      CHARACTER WYE,ENN,QMARK,BUF(80)
      COMMON /INPUT/ LINLMT
      COMMON /ALPHAS/ WYE,ENN,QMARK,BUF
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /DELAYS/ IWODLY(7), IWOSLF(7), IWOLLF(7), IWOSLT(7),
     * IWOLLT(7)
      COMMON /LOGICS/ TRU, FALS
      KNT = 1
      WRITE (OUT,99997)
      WRITE (OUT,99994)
      WRITE (OUT,99993) (IWODLY(I),I=1,7)
   10 CONTINUE
      WRITE (OUT,99999) KNT
      READ (INP,99998,END=10016,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10016 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 30
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 30
      ITMP = IDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 10
      IF (ITMP.GE.0) GO TO 20
      WRITE (OUT,99996)
      GO TO 10
   20 CONTINUE
      IWODLY(KNT) = ITMP
   30 CONTINUE
      WRITE (OUT,99995) IWODLY(KNT)
      KNT = KNT + 1
      IF (KNT.GT.7) GO TO 40
      GO TO 10
```

```
   40 CONTINUE
99999 FORMAT (2H #, I1, 2H ?)
99998 FORMAT (80A1)
99997 FORMAT (//1X,12H WORKOVERS =)
99996 FORMAT (37H  RANGE ERROR: MUST BE > OR = TO ZERO)
99995 FORMAT (2H  , I5, 5H DAYS)
99994 FORMAT (1H , 8X, 4HSUMP, 3X, 7HSUMP-CH, 4X, 4HREV1, 3X, 6HREV2-1,
     *            3X, 6HREV2-2, 3X, 6HREV3-1, 3X, 6HREV3-2)
99993 FORMAT (4H    , 7I9, 6H  DAYS
     *        //,10X,26HPRESS RETURN FOR NO CHANGE)
      RETURN
      END
C
C       ********* END OF EDDLYS ROUTINE ******************************
C
C       ************************************************************
C                                ** EDVOLS **
C       ************************************************************
C
            SUBROUTINE EDVOLS(IWARN2, IWARN3)
      INTEGER INP, OUT, OLD, NEW
      INTEGER TRU, FALS, ERR
      CHARACTER WYE,ENN,QMARK,BUF(80)
      COMMON /INPUT/ LINLMT
      COMMON /ALPHAS/WYE,ENN,QMARK,BUF
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /PARAMS/ IDNUM, MODE, CURRCV, CURROV, FINLOV
      COMMON /ENDPTS/ PENDPV(7), DVCSLF(7,2), DVCLLF(7,2), DVCSLT(7,2),
     * DVCLLT(7,2), DVCSIF(7,2)
      COMMON /LOGICS/ TRU, FALS
      WRITE(OUT,99999)CURRCV,CURROV
      IWARN2 = FALS
      IWARN3 = FALS
   10 CONTINUE
      WRITE (OUT,99997)
      READ (INP,99998,END=10017,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10017 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 20
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 20
      RTMP = RDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 10
      GO TO 30
   20 CONTINUE
      RTMP = CURRCV
   30 CONTINUE
      IF (RTMP.GE.0.0) GO TO 40
      WRITE (OUT,99996)
      GO TO 10
   40 CONTINUE
      CURRCV = RTMP
```

```
      WRITE (OUT,99994) CURRCV
      IF (RTMP.LE.PENDPV(7)) GO TO 50
      IWARN2 = TRU
      WRITE (OUT,99995)
   50 CONTINUE
      WRITE (OUT,99993)
      READ (INP,99998,END=10018,IOSTAT=IEOF) (BUF(I),I=1.LINLMT)
10018 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 60
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 60
      RTMP = RDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 50
      GO TO 70
   60 CONTINUE
      RTMP = CURROV
   70 CONTINUE
      IF ((RTMP.GE.0.0) .AND. (RTMP.LE.CURRCV)) GO TO 80
      WRITE (OUT,99992)
      GO TO 50
   80 CONTINUE
      IF (RTMP.LE.FINLOV) GO TO 90
      WRITE (OUT,99991)
      IWARN3 = TRU
   90 CONTINUE
      CURROV = RTMP
      WRITE (OUT,99994) CURROV
99999 FORMAT (//1X,7H CCV = , F8.2, 12H MB   COV = , F8.2, 3H MB)
99998 FORMAT (80A1)
99997 FORMAT (/1X,26H  CCV??(DEFAULT=NO CHANGE))
99996 FORMAT (40H  RANGE ERROR: MUST BE GREATER THAN ZERO)
99995 FORMAT (54H  RANGE WARNING: MUST BE < OR = TO FINAL CAVERN VOLUME)
99994 FORMAT (2H  , F8.2, 3H MB)
99993 FORMAT (/1X,26H  COV??(DEFAULT=NO CHANGE))
99992 FORMAT (52H  RANGE ERROR: MUST BE > OR = TO 0 AND < OR = TO CCV)
99991 FORMAT (51H  RANGE WARNING: MUST BE < OR = TO FINAL OIL VOLUME)
      RETURN
      END
C
C       *********** END OF EDVOLS SUBROUTINE ************************
C
C       ************************************************************
C                              ** EDBPRS **
C
C               THIS ROUTINE PERFORMS THE INTERACTIVE EDITTING FOR THE
C       BRINE_PRODUCTION_RATE/INTERVAL_ENDING_DAY PAIRS FOR THE LEACH
C       SCHEDULE FORECAST.  FIFTEEN INTERVALS ARE THE LIMIT.  THE
C       INTERVAL_ENDING_DAY WILL DEFAULT TO THE LAST COMPUTED FILL DAY
C       FOR THE CAVERN.
C
C       ************************************************************
```

```
C
          SUBROUTINE EDBPRS
      CHARACTER STAR,BUF(80),ICHR,WYE,ENN,QMARK
      INTEGER TRU, FALS, ERR, OUT
      COMMON /INPUT/ LINLMT
      COMMON /ALPHAS/WYE,ENN,QMARK,BUF
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /STATS/ IENDBP(15), BPRS(15), BPRMIN, BPRMAX
      COMMON /PARAMS/ IDNUM, MODE
      COMMON /LOGICS/ TRU, FALS
      DATA STAR /'*'/
      KNT = 1
      WRITE (OUT,99997)
   10 CONTINUE
      IF (IENDBP(KNT).GE.0) GO TO 20
      WRITE (OUT,99996) BPRS(KNT)
      GO TO 30
   20 CONTINUE
      WRITE (OUT,99995) BPRS(KNT), IENDBP(KNT)
      KNT = KNT + 1
      IF (KNT.LE.15) GO TO 10
   30 CONTINUE
      WRITE(OUT,99989)
      KNT = 1
   40 CONTINUE
      WRITE (OUT,99999) KNT
      READ (INP,99998,END=10019,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10019 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 110
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 110
      RTMP = RDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 40
      IF (RTMP.EQ.0.0) GO TO 60
      IF ((RTMP.GE.BPRMIN) .AND. (RTMP.LE.BPRMAX)) GO TO 50
      WRITE (OUT,99994) BPRMIN, BPRMAX
      GO TO 40
   50 CONTINUE
      IF (MODE.EQ.1) GO TO 60
      IF (RTMP.GE.100.0) GO TO 60
      WRITE (OUT,99990)
   60 CONTINUE
      IPTR = ISKIPB(BUF,LINLMT,IPTR)
      IF (IPTR.GT.LINLMT) GO TO 100
      ICHR = BUF(IPTR)
      IF (ICHR.NE.STAR) GO TO 70
      BPRS(KNT) = RTMP
      GO TO 120
   70 CONTINUE
      ITMP = IDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 40
```

```
      IF (KNT.GT.1) GO TO 80
      IF (ITMP.GT.0) GO TO 90
      WRITE (OUT,99993)
      GO TO 40
   80 CONTINUE
      IF (ITMP.GE.IENDBP(KNT-1)) GO TO 90
      WRITE (OUT,99992) IENDBP(KNT-1)
      GO TO 40
   90 CONTINUE
      IENDBP(KNT) = ITMP
  100 CONTINUE
      BPRS(KNT) = RTMP
  110 CONTINUE
      IF ((KNT.GE.15) .OR. (IENDBP(KNT).LT.0)) GO TO 120
      WRITE (OUT,99995) BPRS(KNT), IENDBP(KNT)
      KNT = KNT + 1
      GO TO 40
  120 CONTINUE
      IENDBP(KNT) = -1
      WRITE (OUT,99996) BPRS(KNT)
      IF (BPRS(KNT).GT.0.0) GO TO 130
      WRITE (OUT,99991)
      GO TO 40
  130 CONTINUE
      KNT = KNT + 1
      IF (KNT.GT.15) GO TO 140
      BPRS(KNT) = 0.0
      IENDBP(KNT) = -1
      GO TO 130
  140 CONTINUE
99999 FORMAT (2H #, I2, 2H ?)
99998 FORMAT (80A1)
99997 FORMAT (//1X,7H BPRS =)
99996 FORMAT (2H  , F8.2, 4H,  *)
99995 FORMAT (2H  , F8.2, 1H,, I5)
99994 FORMAT (31H  RANGE ERROR: MUST BE 0.0 OR <, F7.2, 4H TO , F7.2,
     *        2H !)
99993 FORMAT (44H  RANGE ERROR: MUST BE GREATER THAN DAY ZERO)
99992 FORMAT (33H  RANGE ERROR: MUST BE > OR = TO , I5)
99991 FORMAT (51H  ERROR: 0 BRINE RATE NOT ALLOWED FOR LAST INTERVAL)
99990 FORMAT (48H  WARNING: THIS BPR MAY PRODUCE LEACH ERRORS >5%)
99989 FORMAT (//,5X,26HPRESS RETURN FOR NO CHANGE//)
      RETURN
      END
C
C       *********** END OF EDBPRS ROUTINE ***************************
C
C       ************************************************************
C                              ** EDEPVS **
C
C               THIS ROUTINE PROVIDES THE EDITING INTERACTION FOR THE
C       PHASE ENDPOINT VOLUMES.
C
```

```
C       ************************************************************
C
            SUBROUTINE EDEPVS(IWARN1, IWARN2)
      CHARACTER WYE,ENN,QMARK,BUF(80)
      INTEGER TRU, FALS, ERR, OUT
      COMMON /INPUT/ LINLMT
      COMMON /ALPHAS/WYE,ENN,QMARK,BUF
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /PARAMS/ IDNUM, MODE, CURRCV, CURROV, FINLOV
      COMMON /ENDPTS/ PENDPV(7), DVCSLF(7,2), DVCLLF(7,2), DVCSLT(7,2),
     * DVCLLT(7,2), DVCSIF(7,2)
      COMMON /LOGICS/ TRU, FALS
      IWARN1 = FALS
      IWARN2 = FALS
      KNT = 1
      WRITE (OUT,99997)
      WRITE (OUT,99990)
      WRITE (OUT,99989) (PENDPV(I),I=1,7)
   10 CONTINUE
      WRITE (OUT,99999) KNT
      READ (INP,99998,END=10020,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10020 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 40
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 40
      RTMP = RDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 10
      IF (KNT.GT.1) GO TO 20
      IF (RTMP.GE.0.0) GO TO 30
      WRITE (OUT,99996)
      GO TO 10
   20 CONTINUE
      IF (RTMP.GE.PENDPV(KNT-1)) GO TO 30
      WRITE (OUT,99995) PENDPV(KNT-1)
      GO TO 10
   30 CONTINUE
      PENDPV(KNT) = RTMP
   40 CONTINUE
      WRITE (OUT,99994) PENDPV(KNT)
      KNT = KNT + 1
      IF (KNT.GT.7) GO TO 50
      GO TO 10
   50 CONTINUE
      IF (FINLOV.LE.PENDPV(7)) GO TO 60
      WRITE (OUT,99992)
      WRITE (OUT,99991)
      IWARN1 = TRU
   60 CONTINUE
      IF (CURRCV.LE.PENDPV(7)) GO TO 70
      WRITE (OUT,99993)
      IWARN2 = TRU
```

```
   70 CONTINUE
99999 FORMAT (2H #, I1, 2H ?)
99998 FORMAT (80A1)
99997 FORMAT (//1X,13H ENDVOLUMES =)
99996 FORMAT (51H  RANGE ERROR: MUST BE GREATER THAN OR EQUAL TO 0.0)
99995 FORMAT (48H  RANGE ERROR: MUST BE GREATER THAN OR EQUAL TO , F8.2)
99994 FORMAT (2H  , F8.2, 3H MB)
99993 FORMAT (52H  RANGE WARNING: FINAL VOL. MUST BE >= TO CURR. VOL.)
99992 FORMAT (52H  RANGE WARNING: FINAL CAVERN VOL. MUST BE > OR = TO)
99991 FORMAT (36H                    FINAL OIL VOLUME)
99990 FORMAT (1H , 8X, 4HSUMP, 3X, 7HSUMP-CH, 4X, 4HREV1, 3X, 6HREV2-1,
     *            3X, 6HREV2-2, 3X, 6HREV3-1, 3X, 6HREV3-2)
99989 FORMAT (5H     , 7F9.2, 3H MB
     *       //,10X,26HPRESS RETURN FOR NO CHANGE)
      RETURN
      END
C
C       ************** END OF EDEPVS ROUTINE *************************
C
C       ************************************************************
C
C                              ** EDOIL **
C
C                 THIS ROUTINE PROVIDES THE EDITING INTERACTION FOR THE
C       INTERMITTENT OIL VOLUMES IF THE MODE OF LEACHING IS THE
C       INTERMITTENT FILL MODE
C
C
C       ************************************************************
C
            SUBROUTINE EDOIL
      CHARACTER WYE,ENN,QMARK,BUF(80)
      INTEGER INP, OUT, OLD, NEW
      INTEGER TRU, FALS, ERR
      COMMON /INPUT/ LINLMT
      COMMON /ALPHAS/WYE,ENN,QMARK,BUF
      COMMON /BLK1/ OILVOL(3)
      COMMON /PARAMS/ IDNUM, MODE, CURRCV, CURROV, FINLOV
      COMMON /ENDPTS/ PENDPV(7), DVCSLF(7,2), DVCLLF(7,2), DVCSLT(7,2),
     * DVCLLT(7,2), DVCSIF(7,2)
      COMMON /LOGICS/ TRU, FALS
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      KNT = 1
      WRITE (OUT,99999)
      WRITE (OUT,99998)
      WRITE (OUT,99997)
      WRITE (OUT,99996) OILVOL(1), OILVOL(2), OILVOL(3)
   10 CONTINUE
      WRITE (OUT,99995) KNT
      READ (INP,99994,END=10021,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10021 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 30
```

```
      ENDIF
      IPTR = ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 30
      RTMP = RDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 10
      IF (RTMP.GE.0.) GO TO 20
      WRITE (OUT,99993)
   20 CONTINUE
      OILVOL(KNT) = RTMP
   30 CONTINUE
      WRITE (OUT,99992) OILVOL(KNT)
      KNT = KNT + 1
      IF (KNT.GT.2) GO TO 40
      GO TO 10
   40 CONTINUE
      OILVOL(3) = FINLOV
      WRITE (OUT,99998)
      WRITE (OUT,99991) OILVOL(3)
99999 FORMAT (/1X,37H   INTERMITTENT OIL FILL VOLUMES ARE:)
99998 FORMAT (/)
99997 FORMAT (1H , 10X, 4HSUMP, 3X, 7HSUMP-CH, 4X, 4HREV1, 3X, 6HREV2-1,
     *             3X, 6HREV2-2, 3X, 6HREV3-1, 3X, 6HREV3-2)
99996 FORMAT (11H  OIL VOL =, 14X, 2F9.2, 9X, F9.2,12H         MB
     *       //,10X,26HPRESS RETURN FOR NO CHANGE)
99995 FORMAT (2H #, I1, 2H ?)
99994 FORMAT (80A1)
99993 FORMAT (39H  RANGE ERROR: MUST BE GREATER THAN 0.0)
99992 FORMAT (2H  , F9.2, 3H MB)
99991 FORMAT (48H THE THIRD INTERMITTENT OIL VOLUME MUST BE EQUAL,/
     *        44H TO THE FINAL OIL VOLUME (FOV). EDITOR WILL ,/
     *        55H INTERNALLY FIX THE THIRD INTERMITTENT OIL VOLUME TO BE
     *        ,//25X,F9.2,3H MB)
      RETURN
      END
C
C*********************************************************************
C
C                              ** FWOIL **
C
C     THIS ROUTINE PROVIDES THE EDITTING INTERACTION FOR THE
C INTERMITTENT OIL FILL RATES FOR EACH OF THE THREE LEACHING MODES
C
C
C ****************************************************************
C
            SUBROUTINE FWOIL
      CHARACTER WYE,ENN,QMARK,BUF(80)
      INTEGER TRU,FALS,ERR,OUT
      COMMON /BLK2/ FLRT(3)
      COMMON /INPUT/ LINLMT
      COMMON /ALPHAS/WYE,ENN,QMARK,BUF
      COMMON /LUNITS/ INP,OUT,OLD,NEW
      COMMON /PARAMS/ IDNUM,MODE
```

```
      COMMON /LOGICS/ TRU,FALS
      WRITE (OUT,199)
      IF (MODE .EQ. 2) GO TO 60
   10 CONTINUE
      WRITE (OUT,200) FLRT(1)
      WRITE (OUT,199)
      WRITE (OUT,201)
      READ (INP,99998,END=10022,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10022 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 40
      ENDIF
      IPTR= ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 40
      RTMP= RDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 10
      IF (RTMP.GE. 0.) GO TO 20
      GO TO 10
   20 CONTINUE
      FLRT(1)=RTMP
      FLRT(2)=0.0
      FLRT(3)=0.0
   40 CONTINUE
      WRITE (OUT,202) FLRT(1)
      GO TO 150
   60 CONTINUE
      KNT=1
      WRITE (OUT,210) FLRT(1)
      WRITE (OUT,211) FLRT(2)
      WRITE (OUT,212) FLRT(3)
   70 CONTINUE
      WRITE (OUT,99999) KNT
      READ (INP,99998,END=10023,IOSTAT=IEOF) (BUF(I),I=1,LINLMT)
10023 IF(IEOF.NE.FALS) THEN
         REWIND INP
         GO TO 90
      ENDIF
      IPTR=ISKIPB(BUF,LINLMT,1)
      IF (IPTR.GT.LINLMT) GO TO 90
      RTMP= RDCODE(BUF,LINLMT,IPTR,ERR)
      IF (ERR.EQ.TRU) GO TO 70
   80 CONTINUE
      FLRT(KNT)=RTMP
   90 CONTINUE
      WRITE (OUT,213) FLRT(KNT)
      KNT=KNT+1
      IF (KNT.GT.3) GO TO 150
      GO TO 70
  150 CONTINUE
  199 FORMAT(1H )
  200 FORMAT(24H   FINAL OIL FILL RATE =,F8.2,3H MB)
  201 FORMAT(33H   FILL RATE??(DEFAULT=NO CHANGE))
99998 FORMAT(80A1)
```

SECURITY CLASSIFICATION

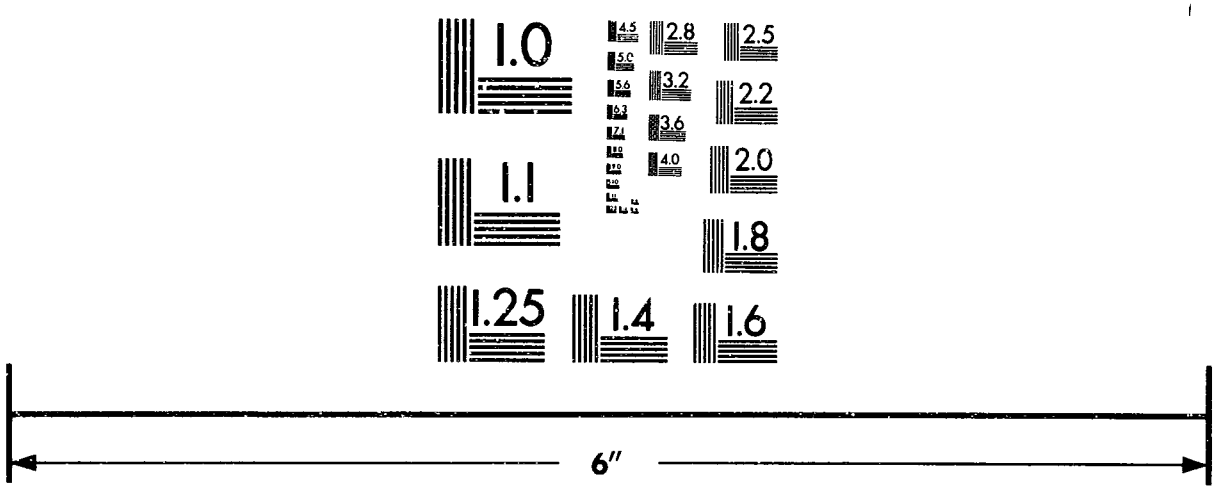
UNCLASSIFIED

DATE OF MICROFILMING
7/22/87

MICROFILMED BY
O. ARMIJO

SANDIA NATIONAL LABS

IMAGE EVALUATION TEST TARGET (MT-3)



PHOTOGRAPHIC SCIENCES CORPORATION
770 BASKET ROAD
P.O. BOX 338
WEBSTER, NEW YORK 14580
(716) 265-1600

```
      GO TO 90
   70 CONTINUE
      DO 80 I=1,3
         RACCUM(I+25,KEY) = 0.0
         RACCUM(I+28,KEY) = FLRT(I)
   80 CONTINUE
   90 CONTINUE
      RETURN
      END
C
C       ************ END OF SAVECV SUBROUTINE ************************
C
C       ************************************************************
C                              ** DMPFIL **
C
C               THIS ROUTINE WRITES TO THE MASTER SCHEDULE FILE.
C
C       ************************************************************
C
            SUBROUTINE DMPFIL
      INTEGER OUT, OLD
      INTEGER TRU, FALS
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      COMMON /SITECM/ FACTOR, ISLIP
      COMMON /ACCUMS/ IACCUM(25,20), RACCUM(31,20)
      COMMON /LOGICS/ TRU, FALS
      REWIND NEW
      WRITE (NEW,99999) FACTOR, ISLIP
      DO 10 J=1,20
         WRITE (NEW,99998) (IACCUM(I,J),I=1,12)
         WRITE (NEW,99997) (IACCUM(I,J),I=13,25)
         WRITE (NEW,99996) (RACCUM(I,J),I=1,8)
         WRITE (NEW,99996) (RACCUM(I,J),I=9,16)
         WRITE (NEW,99996) (RACCUM(I,J),I=17,24)
         WRITE (NEW,99995) (RACCUM(I,J),I=25,31)
   10 CONTINUE
99999 FORMAT (F6.2, I6)
99998 FORMAT (12I6)
99997 FORMAT (13I6)
99996 FORMAT (8F9.2)
99995 FORMAT (7F9.2)
      RETURN
      END
C
C       ********** END OF DMPFIL ROUTINE ***************************
C
C       ************************************************************
C                              ** IDCODE **
C
C               THIS FUNCTION DECODES THE NEXT NON-BLANK STRING IN THE
C       INPUT BUFFER, 'BFR'.  THE SIZE LIMIT OF THE BUFFER IS IN 'BLMT'.
C       THE ARGUMENT, 'PTR', POINTS TO THE NEXT ELEMENT, IN THE BUFFER,
C       TO BE USED.  THE ARRAY, 'ITEMP', PROVIDES TEMPORARY CHARACTER
```

```
C       STORAGE DURING THE PARSING --- MUST BE LARGE ENOUGH TO HOLD UP
C       TO THE MAX NUMBER OF DIGITS ALLOWED AS AN INTEGER IN THE HOST
C       MACHINE.  AT PRESENT, THE VARIABLE, 'MXDGIT', IS LIMITED FOR
C       A PDP11 INTEGER SIZE --- ( -32,768 TO +32,767! ==> 5 DIGITS.
C
C       ************************************************************
C
            FUNCTION IDCODE(BFR, BFRLMT, IPTR, ERR)
      INTEGER INP, OUT, OLD, NEW,BFRLMT,ERR,DIGLMT
      INTEGER TRU, FALS
      CHARACTER BFR(1),BLANK,COMMA,SLASH,SEMIC,COLON,MINUS,PLUS
      DIMENSION ITEMP(10)
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      DATA BLANK, COMMA, SLASH, SEMIC, COLON /' ',',','/',';',':'/
      DATA MINUS, PLUS /'-','+'/
      DATA TRU, FALS /-1,0/
      DATA DIGLMT /5/
      DO 10 I=1,10
         ITEMP(I) = 0
   10 CONTINUE
      ERR = FALS
      MFLG = FALS
      NXT = IPTR
      NXT = ISKIPB(BFR,BFRLMT,NXT)
      IF (NXT.GT.BFRLMT) GO TO 90
      IF (BFR(NXT).EQ.PLUS) GO TO 20
      IF (BFR(NXT).NE.MINUS) GO TO 30
      MFLG = TRU
   20 CONTINUE
      NXT = NXT + 1
      IF (NXT.GT.BFRLMT) GO TO 90
C                       ** FIRST NUMBER **
   30 CONTINUE
      JPTR = 1
      IF (NMBTST(BFR(NXT),IVAL).EQ.FALS) GO TO 100
      ITEMP(JPTR) = IVAL
      NXT = NXT + 1
      JPTR = JPTR + 1

C                       ** THE REST OF THE NUMBERS **
   40 CONTINUE
      IF (NXT.GT.BFRLMT) GO TO 80
      IF (NMBTST(BFR(NXT),IVAL).EQ.FALS) GO TO 60
      IF (JPTR.GT.DIGLMT) GO TO 110
      ITEMP(JPTR) = IVAL
      IF (JPTR.NE.DIGLMT) GO TO 50
      IF (MAXTST(ITEMP,JPTR).EQ.TRU) GO TO 110
   50 CONTINUE
      NXT = NXT + 1
      JPTR = JPTR + 1
      GO TO 40
   60 CONTINUE
      IF ((BFR(NXT).EQ.BLANK) .OR. (BFR(NXT).EQ.COMMA)) GO TO 70
```

```
      IF ((BFR(NXT).EQ.SLASH) .OR. (BFR(NXT).EQ.SEMIC)) GO TO 70
      IF (BFR(NXT).EQ.COLON) GO TO 70
      GO TO 120
   70 CONTINUE
      NXT = NXT + 1
   80 CONTINUE
      IPTR = NXT
      JPTR = JPTR - 1
      IDCODE = ICNVRT(ITEMP,JPTR)
      IF (MFLG.EQ.TRU) IDCODE = -IDCODE
      GO TO 140
C                        ** ERROR MESSAGES **
   90 CONTINUE
      WRITE (OUT,99999)
      GO TO 130
  100 CONTINUE
      WRITE (OUT,99998)
      GO TO 130
  110 CONTINUE
      WRITE (OUT,99997)
      GO TO 130
  120 CONTINUE
      WRITE (OUT,99996)
  130 CONTINUE
      ERR = TRU
      IPTR = NXT
      IDCODE = 0
  140 CONTINUE
99999 FORMAT (35H  INPUT ERROR:  NO CHARACTERS FOUND)
99998 FORMAT (41H  FORMAT ERROR: CHARACTER IS NOT A NUMBER)
99997 FORMAT (44H  RANGE ERROR: ATTEMPT TO EXCEED MAX INTEGER)
99996 FORMAT (43H  FORMAT ERROR: CHAR IS NOT LEGAL DELIMITER)
      RETURN
      END
C
C       ************** END OF IDCODE FUNCTION ******************
C
C       ************************************************************
C                                ** ICNVRT **
C       ************************************************************
C
            FUNCTION ICNVRT(BUF, DSIZ)
      INTEGER BUF(1), DSIZ
      ITMP = 0
      I = 1
   10 CONTINUE
      ITMP = (10*ITMP) + BUF(I)
      I = I + 1
      IF (I.LE.DSIZ) GO TO 10
      ICNVRT = ITMP
      RETURN
      END
C
```

```
C       ************** END OF ICNVRT FUNCTION ************************
C
C       ************************************************************
C                                ** MAXTST **
C       ************************************************************
C
            FUNCTION MAXTST(ITEMP, NUM)
      INTEGER TRU, FALS
      DIMENSION MAXINT(10), ITEMP(1)
      DATA MAXINT /3,2,7,6,7,0,0,0,0,0/
      DATA TRU, FALS /-1,0/
      LMT = 5
      MAXFLG = FALS
      IPTR = 1
      IF (NUM.LT.LMT) GO TO 30
   10 CONTINUE
      IF (ITEMP(IPTR).LT.MAXINT(IPTR)) GO TO 30
      IF (ITEMP(IPTR).EQ.MAXINT(IPTR)) GO TO 20
      MAXFLG = TRU
      GO TO 30
   20 CONTINUE
      IPTR = IPTR + 1
      IF (IPTR.GT.NUM) GO TO 30
      GO TO 10
   30 CONTINUE
      MAXTST = MAXFLG
      RETURN
      END
C
C       ************** END OF MAXTST ******************************
C
C       ************************************************************
C                                ** NMBTST **
C
C               THIS FUNCTION RETURNS 'TRUE' IF THE ARGUMENT 'ICHAR'
C       CONTAINS AN ASCII DIGIT IN < '0'...'9'!.  THE ARGUMENT 'IVAL'
C       WILL CONTAIN THE IN :GER VALUE OF THE LEGAL DIGIT FOUND IN
C       'ICHAR'.
C
C       ************************************************************
C
            FUNCTION NMBTST(ICHAR, IVAL)
      INTEGER FLAG
      INTEGER TRU, FALS
      CHARACTER ICHAR,NUMBS(10)
      DATA NUMBS /'0','1','2','3','4','5','6','7','8','9'/
      DATA TRU, FALS /-1,0/
      FLAG = FALS
      I = 1
   10 IF (I.GT.10) GO TO 30
      IF (ICHAR.EQ.NUMBS(I)) GO TO 20
      I = I + 1
      GO TO 10
```

```
   20 CONTINUE
      IVAL = I - 1
      FLAG = TRU
      GO TO 40
   30 CONTINUE
      IVAL = -1
   40 CONTINUE
      NMBTST = FLAG
      RETURN
      END
C
C       ***************** END OF NMBTST FUNCTION *********************
C
C       ************************************************************
C                              ** RDCODE **
C
C               THIS FUNCTION ATTEMPTS TO DECODE THE NEXT CHARACTER
C       STRING IN THE INPUT BUFFER, 'BFR', AS A REAL NUMBER.  THE
C       MAXIMUM NUMBER OF LEGAL DIGITS IS DETERMINED BY THE VALUE
C       IN THE VARIABLE, 'LMT' --- THIS IS MACHINE DEPENDENT AND
C       SHOULD BE ADJUSTED TO THE MAXIMUM NUMBER OF SIGNIFICANT
C       DIGITS ON THE HOST MACHINE.  THE SIZE OF THE ARRAY, 'ITEMP',
C       MAY HAVE TO BE ADJUSTED ALSO.
C
C       ************************************************************
C
            FUNCTION RDCODE(BFR, BLMT, PTR, ERR)
      INTEGER BLMT, PTR, ERR
      CHARACTER BLANK,COMMA,SLASH,SEMIC,COLON,DOT,MINUS,PLUS,BFR(1)
      INTEGER INP, OUT, OLD, NEW
      INTEGER TRU, FALS
      DIMENSION ITEMP(10)
      COMMON /LUNITS/ INP, OUT, OLD, NEW
      DATA BLANK, COMMA, SLASH, SEMIC, COLON, DOT /' ',',','/',';',':',
     * '.'/
      DATA MINUS, PLUS /'-','+'/
      DATA TRU, FALS /-1,0/
      LMT = 7
      DO 10 I=1,10
         ITEMP(I) = 0
   10 CONTINUE
      ERR = FALS
      MFLG = FALS
      I = PTR
      J = 0
      NWHOLE = 0
      NFRACT = 0
      NFACT = 0
C               * REMOVE LEADING BLANKS *
   20 IF (I.GT.BLMT) GO TO 180
      IF (BFR(I).NE.BLANK) GO TO 30
      I = I + 1
      GO TO 20
```

```
 30 CONTINUE
    IF (BFR(I).EQ.MINUS) GO TO 40
    IF (BFR(I).EQ.PLUS) GO TO 50
    GO TO 60
 40 CONTINUE
    MFLG = TRU
 50 CONTINUE
    I = I + 1
    IF (I.GT.BLMT) GO TO 170
 60 CONTINUE
    IF (BFR(I).EQ.DOT) GO TO 90
    IF (NMBTST(BFR(I),IVAL).EQ.FALS) GO TO 170
    J = 1
    NWHOLE = 1
    ITEMP(J) = IVAL
    I = I + 1
    J = J + 1
 70 CONTINUE
    IF (I.GT.BLMT) GO TO 170
    IF (NMBTST(BFR(I),IVAL).EQ.FALS) GO TO 80
    IF (J.GT.LMT) GO TO 170
    ITEMP(J) = IVAL
    NWHOLE = NWHOLE + 1
    I = I + 1
    J = J + 1
    GO TO 70
 80 CONTINUE
    IF (BFR(I).NE.DOT) GO TO 140
    I = I + 1
    GO TO 100
 90 CONTINUE
    I = I + 1
    IF (I.GT.BLMT) GO TO 170
    IF (NMBTST(BFR(I),IVAL).EQ.FALS) GO TO 170
    NFRACT = 1
    J = 1
    ITEMP(J) = IVAL
    I = I + 1
    J = J + 1
100 CONTINUE
    IF (I.GT.BLMT) GO TO 160
    IF (NMBTST(BFR(I),IVAL).EQ.FALS) GO TO 110
    IF (J.GT.LMT) GO TO 120
    ITEMP(J) = IVAL
    NFACT = NFACT + 1
    I = I + 1
    J = J + 1
    GO TO 100
110 CONTINUE
    GO TO 140
120 CONTINUE
    WRITE (OUT,99999) LMT
    ERR = TRU
```

```
      GO TO 180
  130 CONTINUE
      IF ((I.GT.BLMT) .OR. (NMBTST(BFR(I),IVAL).EQ.FALS)) GO TO 140
      I = I + 1
      GO TO 130
  140 CONTINUE
      IF ((BFR(I).EQ.BLANK) .OR. (BFR(I).EQ.SLASH)) GO TO 150
      IF ((BFR(I).EQ.COMMA) .OR. (BFR(I).EQ.SEMIC)) GO TO 150
      IF (BFR(I).EQ.COLON) GO TO 150
      GO TO 170
  150 CONTINUE
      I = I + 1
  160 CONTINUE
      PTR = I
      RTMP = RCNVRT(ITEMP,NWHOLE,NFACT)
      IF (MFLG.EQ.TRU) RTMP = -RTMP
      GO TO 190
  170 CONTINUE
      ERR = TRU
      WRITE (OUT,99998)
  180 CONTINUE
      RTMP = 0.0
  190 CONTINUE
      RDCODE = RTMP
99999 FORMAT (44H  RANGE ERROR:  EXCEEDED THE DIGIT LIMIT OF , I2)
99998 FORMAT (50H  INPUT ERROR:  INCORRECT REAL FORMAT OR CHARACTER)
      RETURN
      END
C
C       ************* END OF RDCODE FUNCTION ************************
C
C       ************************************************************
C                              ** RCNVRT **
C
C               THIS FUNCTION PERFORMS THE ACTUAL CONVERSION AN A
C       LEGAL PARSED STRING THAT CONTAINS THE INTEGERS THAT MAKE UP
C       A REAL NUMBER.  THE NUMBER OF DIGIITS THAT ARE TO THE LEFT
C       OF THE DECIMAL POINT OR THE FIRST LEGAL DELIMITER ARE PASSED
C       IN THE ARGUMENT, 'WSIZ', AND THE NUMBER OF DIGITS TO THE
C       RIGHT OF THE DECIMAL POINT ARE PASSED IN THE ARGUMENT, 'FSIZ'.
C       THE SUM OF WSIZ AND FSIZ SHOULD NOT EXCEED THE MAXIMUM NUMBER
C       OF SIGNIFICANT DIGITS FOR THE HOST SYSTEM NOR THE SIZE OF THE
C       ARRAY, 'BUF', AS DECLARED IN THE CALLING ROUTINE.
C
C       ************************************************************
C
            FUNCTION RCNVRT(BUF, WSIZ, FSIZ)
      INTEGER BUF(1), WSIZ, FSIZ, WHOLE
      WHOLE = WSIZ - 1
      RTMP = 0.0
      FACTOR = 0.10
      I = 1
   10 CONTINUE
```

```
      IF (I.GT.WSIZ) GO TO 20
      RTMP = RTMP + FLOAT(BUF(I))*(10.0**WHOLE)
      WHOLE = WHOLE - 1
      I = I + 1
      GO TO 10
   20 CONTINUE
      J = 1
   30 CONTINUE
      IF (J.GT.FSIZ) GO TO 40
      RTMP = RTMP + FLOAT(BUF(I))*(FACTOR**J)
      I = I + 1
      J = J + 1
      GO TO 30
   40 CONTINUE
      RCNVRT = RTMP
      RETURN
      END
C
C       ********* END OF RCNVRT FUNCTION *************************
C
C       ************************************************************
C                              ** EDUTIL **
C                        ** UTILITY STORAGE FILE **
C
C               THIS UTILITY FILE IS FOR THE SPR EDITOR PROGRAM.
C
C       ************************************************************
C
            FUNCTION ISKIPB(BUF, LMT, IPT)
      INTEGER PTR
      CHARACTER BUF(1),BLANK
      DATA BLANK /' '/
      PTR = IPT
   10 CONTINUE
      IF ((PTR.GT.LMT) .OR. (BUF(PTR).NE.BLANK)) GO TO 20
      PTR = PTR + 1
      GO TO 10
   20 CONTINUE
      ISKIPB = PTR
      RETURN
      END
C
C       ************** END OF ISKIPB FUNCTION ************************
C
C       ************************************************************
C                              ** IALL **
C
C               THIS FUNCTION TESTS THE INPUT BUFFER FOR THE STRING
C       'ALL' STARTING AT THE ELEMENT POINTED TO BY IPTR.
C
C       ************************************************************
C
            FUNCTION IALL(IPTR)
```

```
      CHARACTER BUF(80),AAY,ELL,WYE,ENN,QMARK
      INTEGER TRU, FALS
      COMMON /INPUT/ LINLMT
      COMMON /ALPHAS/WYE,ENN,QMARK,BUF
      COMMON /LOGICS/ TRU, FALS
      DATA AAY, ELL /'A','L'/
      I = IPTR
      IFLG = FALS
      IF (BUF(I).NE.AAY) GO TO 10
      I = I + 1
      IF (I.GT.LINLMT) GO TO 10
      IF (BUF(I).NE.ELL) GO TO 10
      I = I + 1
      IF (I.GT.LINLMT) GO TO 10
      IF (BUF(I).NE.ELL) GO TO 10
      IFLG = TRU
   10 CONTINUE
      IALL = IFLG
      RETURN
      END
C
C       ***************** END OF IALL FUNCTION **********************
C
            FUNCTION CAVOIL(CFV, V, IFIN)
      DIMENSION C1(6), C2(6), C3(6), C4(6), P(7)
      DATA C1 /0.,-12.05,-296.398,-86.878,-1577.06,-1537.70/
      DATA C2 /0.,-0.02245,-0.5112,-.15752,-2.77392,-2.78227/
      DATA C3 /0.,0.09395,1.0025,0.4707,3.36372,3.3649/
      DATA C4 /0.,1.7692E-8,-1.07716E-7,-1.2817E-9,-1.8516E-7,1.2366E-7/
      IFIN = 0
      VD = CFV - 10000.
      P(1) = 0.0
      P(2) = 2513. + 0.238*VD
      P(3) = 5700. + 0.540*VD
      P(4) = 7058. + 0.6684*VD
      P(5) = 9565. + 0.9058*VD
      P(6) = 10383. + 0.9832*VD
      P(7) = 11200. + 1.0606*VD
      DO 10 I=1,6
         J = I
         IF (V.LT.P(I+1)) GO TO 20
   10 CONTINUE
   20 CAVOIL = C1(J) + C2(J)*CFV + C3(J)*V + C4(J)*V*CFV
      IF (V.GE.P(7)) CAVOIL = 8340. + 0.7900*VD
      IF (V.GE.P(7)) IFIN = 1
      RETURN
      END
            SUBROUTINE PREDCT
C
C*********************************************************************
C
C       PREDICT LEACHING SCHEDULE FOR SPR SITES
C
```

```
C***********************************************************************
C
      INTEGER CAVERN(20),DELTAT,MORDER(360)
      INTEGER WKOVER,LUNTO,LUNTI,INCDAY
      INTEGER NEXTBP,LASTBP,MAXBP,MAXCAV,LLIMIT
      INTEGER OLDFIL,NEWFIL,ERROR
      CHARACTER YES,NO,ANSWER(80)
      CHARACTER*2 LCHPHS
      CHARACTER*8 INFIL,OUTFIL,PRDFIL
      LOGICAL SITOUT,CAVOUT,INCOUT,WARN
      REAL CONFAC,ENDCAV,ENDOIL
      REAL TOTCAV,TOTOIL,DELOIL,BPR,TOTOFL
      COMMON /FILES/OUTFIL,PRDFIL
      COMMON /BLOK1/ IACC(25,20), RACC(31,20)
      COMMON /BLOK2/ CURVOL,DELCAV,CUROIL,DELOIL,QOIL,IDAY,DELTAT
      COMMON /BLOK3/ CAVOUT,LUNLP,LASTDY
      COMMON /BLOK4/ LCHPHS(2,7)
      COMMON /BLOK5/ FLOW,ICN,IST6,IPOINT,IZER,NCV,R21,R22,R31,R32
      COMMON /BLOK6/ CAVVOL(28,20),OILVOL(28,20),OILFLO(28,20),
     1 ITIME(28,20),BRNFLO(28,20),FINOIL(20)
      COMMON /SITCM/ CONFAC,INIDAY
      COMMON /LUNTS/ LUNTI,LUNTO,OLDFIL,NEWFIL
      DATA IACT,IMODE,IWO,IENDBP,LASTTI
     +    /  1,    2, 18,     3,     0/
      DATA IENDPV,IBPR,ICURCV,ICUROV,IENDOV
     +    /    18,   3,     1,     2,     3/
      DATA NUMCAV,MAXBP, YES,  NO,MAXCAV,LLIMIT,INCDAY
     +    /    20,   28, 'Y', 'N',    20,    80,    90/
C
C.........INITIALIZE
C
      LUNLP=2
      PRDFIL='OUTPUT'
 5013 WRITE(5,5000)
 5000 FORMAT(///,1H ,33HENTER INPUT FILE NAME FROM EDITOR)
      READ(5,5010,END=5011,IOSTAT=IEOF)INFIL
 5011 IF(IEOF.NE.0) THEN
         REWIND 5
         WRITE(5,5012)
 5012    FORMAT(1H ,38HTHERE IS NO DEFAULT FOR THIS FILE NAME)
         GO TO 5013
      ENDIF
 5010 FORMAT(A8)
      OPEN(1,FILE=INFIL)
      REWIND1
      WRITE(5,5030)
 5030 FORMAT(1H ,34HENTER FILE NAME FOR PREDICT OUTPUT,/
     *10X,36H(DEFAULT =OUTPUT, I.E. THE TERMINAL))
      READ(5,5010,END=5040,IOSTAT=IEOF)PRDFIL
 5040 IF(IEOF.NE.0) REWIND 5
      OPEN(2,FILE=PRDFIL)
      REWIND 2
 5023 WRITE(5,5020)
```

```
 5020 FORMAT(1H ,33HENTER FILE NAME FOR AUTSKED INPUT)
      READ(5,5010,END=5021,IOSTAT=IEOF)OUTFIL
 5021 IF(IEOF.NE.0) THEN
         REWIND 5
         WRITE(5,5022)
 5022    FORMAT(1H ,38HTHERE IS NO DEFAULT FOR THIS FILE NAME)
         GO TO 5023
      ENDIF
      IF(OUTFIL.EQ.PRDFIL) THEN
         WRITE(5,5029)
 5029    FORMAT(1H ,41HAUTSKED INPUT FILE NAME MUST BE DIFFERENT/
     *              32H   FROM PREDICT OUTPUT FILE NAME)
         GO TO 5023
      ENDIF
      OPEN(55,FILE=OUTFIL)
      REWIND 55
      DO 10 I=1,20
   10 CAVERN(I)=I
      LUNTO=5
      LUNTI=5
      OLDFIL=1
      NEWFIL=1
      IFLAG =0
      R21=0.1333
      R22=0.06667
      R31=0.3333
      R32=0.3333
      CALL INPHIL (ERROR)
      CONFAC=CONFAC/100.+1.
C
C.........ASK FOR COMPUTING AND OUTPUT FUNCTIONS
C
      WRITE (LUNTO,9001)
      READ  (LUNTI,9101,END=21,IOSTAT=IEOF) ANSWER
   21 IF (IEOF.NE.0) THEN
         REWIND LUNTI
         GO TO 110
      ENDIF
   20 IF (ANSWER(1).NE.NO) GO TO 110
C
C.........GET CAVERN NUMBERS
C
      JUMP=1
      WRITE (LUNTO,9002)
   30 WRITE (LUNTO,9003)
      READ  (LUNTI,9101,END=41,IOSTAT=IEOF) ANSWER
   41 IF (IEOF.NE.0) THEN
         REWIND LUNTI
         GO TO 8902
      ENDIF
   40 NUMCAV=0
      I=1
   50 I=ISCIPB (ANSWER,LLIMIT,I)
```

```
      IF (I.GT.LLIMIT) GO TO 60
      NUMCAV=NUMCAV+1
      CAVERN(NUMCAV)= IDKODE(ANSWER,LLIMIT,I,ERROR)
      IF (ERROR.EQ.-1) GO TO 8901
      IF (NUMCAV.LT.MAXCAV) GO TO 50
   60 IF (NUMCAV.LT.1) GO TO 8902
      DO 70 J=1,NUMCAV
         IF (CAVERN(J).LT.101.OR.CAVERN(J).GT.120) GO TO 8901
   70 CONTINUE
   71 WRITE (LUNTO,9004) (CAVERN(K),K=1,NUMCAV)
      WRITE (LUNTO,9005)
      READ  (LUNTI,9101,END=81,IOSTAT=IEOF) ANSWER
   81 IF (IEOF.NE.0) THEN
         REWIND LUNTI
         GO TO 90
      ENDIF
   80 IF (ANSWER(1).EQ.NO) GO TO 60
   90 DO 100 J=1,NUMCAV
         CAVERN(J)=CAVERN(J)-100
  100 CONTINUE
C
C.........GET OUTPUT INSTRUCTIONS
C
  110 CAVOUT=.TRUE.
      WRITE (LUNTO,9006)
      READ  (LUNTI,9101,END=121,IOSTAT=IEOF) ANSWER
  121 IF (IEOF.NE.0) THEN
         REWIND LUNTI
         GO TO 130
      ENDIF
  120 IF (ANSWER(1).EQ.NO) CAVOUT=.FALSE.
  130 SITOUT=.FALSE.
      WRITE (LUNTO,9007)
      READ  (LUNTI,9101,END=141,IOSTAT=IEOF) ANSWER
  141 IF (IEOF.NE.0) THEN
         REWIND LUNTI
         GO TO 150
      ENDIF
  140 IF (ANSWER(1).EQ.YES) SITOUT=.TRUE.
  150 INCOUT=.FALSE.
      WRITE (LUNTO,9008)
      READ  (LUNTI,9101,END=161,IOSTAT=IEOF) ANSWER
  161 IF (IEOF.NE.0) THEN
         REWIND LUNTI
         GO TO 170
      ENDIF
  160 IF (ANSWER(1).EQ.YES) INCOUT=.TRUE.
  170 IF (.NOT.INCOUT) GO TO 200
  180 WRITE (LUNTO,9009)
      READ  (LUNTI,9101,END=191,IOSTAT=IEOF) ANSWER
  191 IF (IEOF.NE.0) THEN
         REWIND LUNTI
         GO TO 200
```

```
      ENDIF
  190 I=1
      I=ISCIPB(ANSWER,LLIMIT,I)
      IF (I.GT.LLIMIT) GO TO 200
      JUMP=2
      INCDAY= IDKODE(ANSWER,LLIMIT,I,ERROR)
      IF (ERROR.EQ.-1) GO TO 8901
      IF (INCDAY.LT.1) GO TO 71
  200  DO 3050 I=1,20
         IF(IACC(1,I).EQ.-1.AND.IACC(2,I).EQ.0) GO TO 3000
 3050 CONTINUE
      GO TO 220
 3000 WRITE(LUNTO,4000)
 4000 FORMAT(///,1H ,48HTHE DEFAULT OIL/BRINE RATIOS IN STAGES R2 AND R3,
     *            /,39H DURING LEACH/FILL MODE ARE AS FOLLOWS:,
     *                //,15X,18HR2-1        0.1333/
     *                   15X,19HR2-2        0.06667/
     *                   15X,18HR3-1        0.3333/
     *                   15X,18HR3-2        0.3333//
     *               30HPRESS RETURN FOR DEFAULT VALUE)
      WRITE(LUNTO,4010)
 4010 FORMAT(1H ,10HR2-1 RATIO)
      READ(LUNTI,4020,END=3010,IOSTAT=IEOF)R21
 4020 FORMAT(F10.4)
 3010 IF(IEOF.NE.0) REWIND LUNTI
      WRITE(LUNTO,4020)R21
      WRITE(LUNTO,4030)
 4030 FORMAT(1H ,10HR2-2 RATIO)
      READ(LUNTI,4020,END=3020,IOSTAT=IEOF)R22
 3020 IF(IEOF.NE.0) REWIND LUNTI
      WRITE(LUNTO,4020)R22
      WRITE(LUNTO,4040)
 4040 FORMAT(1H ,10HR3-1 RATIO)
      READ(LUNTI,4020,END=3030,IOSTAT=IEOF)R31
 3030 IF(IEOF.NE.0) REWIND LUNTI
      WRITE(LUNTO,4020)R31
      WRITE('UNTO,4050)
 4050 FORMAT(1H ,10HR3-2 RATIO)
      READ(LUNTI,4020,END=3040,IOSTAT=IEOF)R32
 3040 IF(IEOF.NE.0) REWIND LUNTI
      WRITE(LUNTO,4020)R32
C
C*********************************************************************
C.........COMPUTE BREAKPOINTS FOR ALL CAVERNS
C*********************************************************************
C
  220 DO 350 IC=1,NUMCAV
        NCV=CAVERN(IC)+100
        IZER=0
      WARN=.FALSE.
      ICN=CAVERN(IC)
      R1OIL= RACC(3,ICN)*0.07151515 - 451.151515
      IF (IACC(IACT,ICN).EQ.0) GO TO 350
```

```
      IF (.NOT.CAVOUT) GO TO 230
      ITEMP= ICN+100
      WRITE (LUNLP,9501) ITEMP
      WRITE (LUNLP,9502)
      WRITE (LUNLP,9503)
      WRITE (LUNLP,9504)
  230 CURVOL= RACC(ICURCV,ICN)
      CUROIL= RACC(ICUROV,ICN)
      IDAY=   INIDAY
      ENDCAV= RACC(IENDPV+7,ICN)
      ENDOIL= RACC(IENDOV,ICN)
      V= ENDOIL/1000.
      IPOINT= 0
      IF(CAVOUT) THEN
         WRITE (LUNLP,9505) IDAY,CURVOL,CUROIL
         XXX=-CUROIL
         WRITE(55,*)XXX,CURVOL
      ENDIF
      CALL STORE(CURVOL,0.,CUROIL,0.,0.,IDAY,0)
C
C*************************************************************
C
C   CHECK TO DETERMINE IF THE CAVERN IS TO BE INITIALLY IDLE
C
      ISTG=1
      CALL ZIDLE(0)
C
C*************************************************************
C
C   DETERMINE THE STAGE
C
  240 I2= 18+ISTG
      IF (CURVOL .LT. RACC(I2,ICN)) GO TO 260
      IF (CURVOL .GT. RACC(I2,ICN)) THEN
         WRITE(55,*)ISTG,IZER,IZER,IZER,IZER,NCV
         GO TO 250
      ENDIF
C
C*************************************************************
C
C   CALL THE WORKOVER SUBROUTINE
C
      CALL WORKOR(I2)
C
C*************************************************************
C
C   DETERMINE IF A FILL IS NECESSARY.  IF ONE IS, THEN
C   BRANCH TO THE FILL SUBROUTINE WHICH PERFORMS THIS FUNCTION
C
      IF (IACC(2,ICN) .EQ. 2 .AND. ISTG .EQ. 3)
     1    CALL FILL(26,29,CONFAC)
      IF (IACC(2,ICN) .EQ. 2 .AND. ISTG .EQ. 5)
     1    CALL FILL(27,30,CONFAC)
```

```
      IF (CURVOL .EQ. RACC(25,ICN)) GO TO 300
      IF(ISTG.EQ.1) THEN
         WRITE(55,*)ISTG,IZER,IZER,IZER,IZER,NCV
      ENDIF
  250 ISTG= ISTG+1
      GO TO 240
C
C*************************************************************
C
C  DETERMINE THE LEACHING EFFICIENCIES
C
  260 IF (IACC(2,ICN) .GT. 0) EF= ZTFETA(ISTG,FLOW,V)
      IF (IACC(2,ICN) .EQ. 0) EF= ZTETA(ISTG,FLOW,V)
C
C*************************************************************
C
C   COMPUTE THE CAVERN VOLUME AND LEACHING TIME
C
      DELCAV= RACC(I2,ICN)-CURVOL
      DELTAT= (DELCAV*CONFAC)/(EF*FLOW) + 1.0
C
C*************************************************************
C
C   DETERMINE WHETHER THE PRESENT STAGE IS FINISHED.  IF
C   NOT, FIRST FIND THE INPUTTED LEACHING TIME AND THEN
C   COMPUTE THE CAVERN VOLUMF LEACHED OUT BASED ON THAT TIME.
C
      IF (LASTDY .LT. 0) GO TO 270
      I4= IDAY+DELTAT
      IF (I4 .LE. LASTDY) GO TO 270
      DELTAT= LASTDY-IDAY
      DELCAV= EF*FLOW*FLOAT(DELTAT)/CONFAC
  270 CONTINUE
C
C*************************************************************
C
C   DETERMINE THE OIL FLOW RATE AND TOTAL OIL VOLUME
C   IN THE CAVERN
C
      CALL OILDET(R1OIL,CONFAC)
C
C*************************************************************
C
C   CALL THE PRINT SUBROUTINE
C
      CALL ZPRNT(I2)
      GO TO 240
C
C*************************************************************
C
C   THESE ARE THE CALLS TO FILL THE CAVERN IF AN
C   INTERMITTENT FILL IS USER SPECIFIED
C
```

```
  300 IF (IACC(2,ICN) .EQ. 2) CALL FILL(28,31,CONFAC)
      IF (IACC(2,ICN) .NE. 2) CALL FILL(3,29,CONFAC)
      IPOINT= IPOINT+1
      ITEMP = ICN+100
      IF (WARN) WRITE (LUNLP,9527) ITEMP
  350 CONTINUE
      IF (.NOT.SITOUT) GO TO 900
C*********************************************************************
C.........OUTPUT SITE BREAKPOINTS
C*********************************************************************
      TOTCAV=0.
      TOTOIL=0.
      TOTOFL=0.
      WRITE (LUNLP,9512)
      WRITE (LUNLP,9513)
      WRITE (LUNLP,9514)
      WRITE (LUNLP,9515)
      WRITE (LUNLP,9516)
C
C.........PRINT INITIAL CONDITIONS
C
      DO 400 IC=1,NUMCAV
         ICN=CAVERN(IC)
         IF (IACC(IACT,ICN).NE.-1) GO TO 400
         TOTCAV=TOTCAV+CAVVOL(1,ICN)
         TOTOIL=TOTOIL+OILVOL(1,ICN)
  400 CONTINUE
      IDAY=INIDAY
      WRITE (LUNLP,9517) IDAY,TOTCAV,TOTOIL
      MORDER(1)= IDAY
      IPOINT= 2
  500 MORDER(IPOINT)= 10000000
      DO 600 IC=1,NUMCAV
         IF (IACC(1,IC) .NE. -1) GO TO 600
         DO 520 J=1,28
            IF (ITIME(J,IC) .LE. MORDER(IPOINT-1)) GO TO 520
            IF (ITIME(J,IC) .GT. MORDER(IPOINT)  ) GO TO 600
            MORDER(IPOINT)= ITIME(J,IC)
  520    CONTINUE
  600 CONTINUE
      IF ( MORDER(IPOINT) .EQ. 10000000) GO TO 650
      IPOINT= IPOINT+1
      GO TO 500
  650 IPOINT= IPOINT-1
      DO 800 ICNT=2,IPOINT
         TOTOFL= 0.0
         BPR   = 0.0
         DO 700 IC=1,NUMCAV
            IF (IACC(1,IC) .NE. -1) GO TO 700
            DO 670 J=2,28
               IF (ITIME(J,IC) .LT. MORDER(ICNT)) GO TO 670
               TEMP1= MORDER(ICNT) - MORDER(ICNT-1)
               TEMP2= ITIME(J,IC) - ITIME(J-1,IC)
```

```
              RATIO= TEMP1/TEMP2
              TOTCAV= TOTCAV + RATIO*(CAVVOL(J,IC) - CAVVOL(J-1,IC))
              TOTOIL= TOTOIL + RATIO*(OILVOL(J,IC) - OILVOL(J-1,IC))
              TOTOFL= TOTOFL + OILFLO(J,IC)
              BPR   = BPR    + BRNFLO(J,IC)
              GO TO 700
  670       CONTINUE
  700    CONTINUE
         ITEMP= MORDER(ICNT-1) + 1
         WRITE(LUNLP,9520) ITEMP,MORDER(ICNT),TOTCAV,TOTOIL,
     *   TOTOFL,BPR
  800 CONTINUE
  900 IF (.NOT.INCOUT) GO TO 1300
C*********************************************************************
C.........OUTPUT SITE INTERVALS
C*********************************************************************
      WRITE (LUNLP,9521)
      WRITE (LUNLP,9522)
      WRITE (LUNLP,9523)
      WRITE (LUNLP,9524)
      WRITE (LUNLP,9525)
C
C.........PRINT INITIAL CONDITIONS
C
      TOTCAV= 0.
      TOTOIL= 0.
      DO 1000 IC=1,NUMCAV
         IF (IACC(IACT,IC).NE.-1) GO TO 1000
         TOTCAV=TOTCAV+CAVVOL(1,IC)
         TOTOIL=TOTOIL+OILVOL(1,IC)
 1000 CONTINUE
      IDAY=   INIDAY
      SAVOIL= 0.0
      WRITE (LUNLP,9518) IDAY,TOTCAV,TOTOIL
 1050 TOTCAV= 0.
      TOTOIL= 0.
      IF ( (IDAY+INCDAY) .LT. MORDER(IPOINT)) GO TO 1080
      INCDAY= MORDER(IPOINT) - IDAY
 1080 IDAY= IDAY + INCDAY
      DO 1200 IC=1,NUMCAV
         IF (IACC(1,IC) .NE. -1) GO TO 1200
         DO 1100 J=2,28
            IF ( ITIME(J,IC) .LT. IDAY) GO TO 1090
            TEMP1= IDAY - ITIME(J-1,IC)
            TEMP2= ITIME(J,IC) - ITIME(J-1,IC)
            RATIO= TEMP1/TEMP2
            IF (RATIO .LT. 0.0) RATIO= 0.0
            TOTCAV= TOTCAV + CAVVOL(J-1,IC)
     *         + RATIO*(CAVVOL(J,IC) - CAVVOL(J-1,IC))
            TOTOIL= TOTOIL + OILVOL(J-1,IC)
     *         + RATIO*(OILVOL(J,IC) - OILVOL(J-1,IC))
            GO TO 1200
 1090       IF (J .EQ. 28) THEN
```

```
              TOTCAV=TOTCAV+RACC(25,IC)
              TOTOIL= TOTOIL + FINOIL(IC)
            ENDIF
 1100    CONTINUE
 1200 CONTINUE
      DELOIL= TOTOIL - SAVOIL
      OILRAT= DELOIL/FLOAT(INCDAY)
      SAVOIL= TOTOIL
C
C.........OUTPUT SITE INTERVAL
C
      WRITE (LUNLP,9526) IDAY,TOTCAV,TOTOIL,DELOIL,OILRAT
C
      IF (IDAY .LT. MORDER(IPOINT)) GO TO 1050
C
C.........TERMINATE
C
 1300 CLOSE(1)
      CLOSE(2)
      IF(PRDFIL.EQ.'OUTPUT') OPEN(6,FILE='OUTPUT')
      CLOSE(55)
      RETURN
C
C.........ERROR MESSAGES
C
 8901 WRITE (LUNTO,9801)
      GO TO (30,180),JUMP
 8902 WRITE (LUNTO,9802)
      GO TO 1300
C
C.........FORMATS
C
 9001 FORMAT (//43HDO YOU WISH TO EVALUATE ALL ACTIVE CAVERNS?/
     *        10X,23H(Y  OR  N, DEFAULT = Y)
     *       /48H  NOTE: ANSWER Y IF YOU ARE GOING TO RUN AUTSKED)
 9002 FORMAT (50H ENTER THE CAVERN NUMBERS OF THOSE TO BE EVALUATED)
 9003 FORMAT (31H (ANY OF THE SET [101,...,120]))
 9004 FORMAT (1H ,I3,19I4)
 9005 FORMAT (37H IS THE ABOVE DATA CORRECT? [Y/N D:Y])
 9006 FORMAT (45H DO YOU WANT BREAKPOINT DATA FOR EACH CAVERN?/
     *        10X,23H(Y  OR  N, DEFAULT = Y)/
     *        48H  NOTE: ANSWER Y IF YOU ARE GOING TO RUN AUTSKED)
 9007 FORMAT (41H DO YOU WANT THE SITE BREAKPOINT SUMMARY?/
     *        10X,23H(Y  OR  N, DEFAULT = N))
 9008 FORMAT (40H DO YOU WANT A SITE INCREMENTAL SUMMARY?/
     *        10X,23H(Y  OR  N, DEFAULT = N))
 9009 FORMAT (38H ENTER REPORT INCREMENT IN DAYS [D:90])
 9010 FORMAT (53H DO YOU WANT A LINE PRINTER LISTING? [Y/N D:TERMINAL])
 9101 FORMAT (80A1)
 9501 FORMAT (//////6HCAVERN,I4,12H BREAKPOINTS)
 9502 FORMAT (52H      DAY    LEACH    BRINE     OIL    CAVERN    OIL)
 9503 FORMAT (54H   INTERVAL  STAGE    FLOW     FLOW    VOLUME   VOLUME)
 9504 FORMAT (54H  ---------   ----  -------  -------  -------  -------)
```

```
 9505 FORMAT (1H ,6X,I4,25X,2F9.0)
 9508 FORMAT (1H ,I5,1H-,I4,3X,4HIDLE)
 9509 FORMAT (1H ,I5,1H-,I4,3X,2A2,F9.1,F9.2,2F9 0)
 9510 FORMAT (1H ,I5,1H-,I4,3X,3HW/O)
 9512 FORMAT (//////22H---SITE BREAKPOINTS---)
 9513 FORMAT (//45H              TOTAL     TOTAL    DAILY   BRINE)
 9514 FORMAT (49H    DAY       CAVERN     OIL     OIL   PRODUCTION)
 9515 FORMAT (45H  INTERVAL    VOLUME   VOLUME    FLOW    RATE)
 9516 FORMAT (47H ---------- -------  -------  -------  -------)
 9517 FORMAT (1H ,6X,I4,2F9.0)
 9518 FORMAT (1H ,I5,2F9.0)
 9520 FORMAT (1H ,I5,1H-,I4,2F9.0,F9.2,F9.1)
 9521 FORMAT (//////18H---SITE SUMMARY---)
 9522 FORMAT (//41H       TOTAL     TOTAL            AVERAGE)
 9523 FORMAT (40H        CAVERN     OIL     OIL       OIL)
 9524 FORMAT (41H   DAY  VOLUMES  VOLUMES   USED      FLOW)
 9525 FORMAT (42H  ----  -------  -------  -------  -------)
 9526 FORMAT (1H ,I5,3F9.0,F9.2)
 9527 FORMAT (//19H***WARNING: CAVERN ,I3
     +        ./,52H BRINE FLOW OUT OF RANGE 100-200MB/DAY DURING LEACH.
     +        ,/,27H MAY CAUSE EXCESSIVE ERROR.)
 9801 FORMAT (41H ***IMPROPER DATA ENTRY! PLEASE TRY AGAIN)
 9802 FORMAT (39H ***NO CAVERNS ENTERED! RUN TERMINATING)
      END
      BLOCK DATA BLOCDT
      CHARACTER*2 LCHPHS(2,7)
      COMMON /BLOK4/ LCHPHS
      DATA LCHPHS(1,1),LCHPHS(2,1)/'SU','MP'/
     +    ,LCHPHS(1,2),LCHPHS(2,2)/'S/','C '/
     +    ,LCHPHS(1,3),LCHPHS(2,3)/'R1','  '/
     +    ,LCHPHS(1,4),LCHPHS(2,4)/'R2','-1'/
     +    ,LCHPHS(1,5),LCHPHS(2,5)/'R2','-2'/
     +    ,LCHPHS(1,6),LCHPHS(2,6)/'R3','-1'/
     +    ,LCHPHS(1,7),LCHPHS(2,7)/'R3','-2'/
      END
C
C*****************************************************************
C
C          SUBROUTINE WORKOR DETERMINES WHETHER A WORKOVER IS
C          NECESSARY AND PERFORMS THE NECESSARY ACTIONS TO INSERT
C          A WORKOVER IN THE SCHEDULE DURING THE USER-SPECIFIED
C          TIME
C
C*****************************************************************
C
           SUBROUTINE WORKOR(I2)
      INTEGER WKOVER
      LOGICAL CAVOUT
      COMMON /BLOK1/ IACC(25,20), RACC(31,20)
      COMMON /BLOK2/ CURVOL,DELCAV,CUROIL,DELOIL,QOIL,IDAY,DELTAT
      COMMON /BLOK3/ CAVOUT,LUNLP,LASTDY
      COMMON /BLOK5/ FLOW,ICN,ISTG,IPOINT,IZER,NCV,R21,R22,R31,R32
      WKOVER= IACC(I2,ICN)
```

```
      IF (WKOVER .EQ. 0) RETURN
      TEMP= FLOW
      FLOW= 0.0
      CALL STORE(CURVOL,0.,CUROIL,0.,0.,IDAY,WKOVER)
      FLOW= TEMP
      ITEMP= IDAY+1
      IDAY= IDAY+WKOVER
      IF(CAVOUT) THEN
         WRITE(LUNLP,10) ITEMP,IDAY
         XIZER=-1.
         WRITE(55,*)ISTG,ITEMP,IDAY,XIZER,XIZER,NCV
      ENDIF
      CALL ZIDLE(1)
   10 FORMAT(1H ,I5,1H-,I4,3X,3HW/O)
      RETURN
      END
C
C*************************************************************
C
C          SUBROUTINE BRINFL FINDS THE PROPER FLOW OF BRINE
C          DEPENDENT UPON THE DAY IN THE SCHEDULE AND USER INPUT
C
            SUBROUTINE BRINFL(FLOW,IDAY,LASTDY,ICN)
      COMMON /BLOK1/ IACC(25,20), RACC(31,20)
      IFL= 1
   10 I1 = 3+IFL
      IF (IACC(I1,ICN) .GT. IDAY
     *    .OR.IACC(I1,ICN) .LT. 0)    GO TO 20
      IFL= IFL+1
      GO TO 10
   20 LASTDY= IACC(I1,ICN)
      FLOW  = RACC(I1,ICN)
      RETURN
      END
C
C*************************************************************
C
C         SUBROUTINE OILDET DETERMINES THE PROPER OIL FLOW RATE
C         AND AMOUNT OF OIL TO BE PLACED IN THE CAVERN BASED ON
C         USER INPUT, THE LEACHING MODE, AND THE TIME OF THE
C         BREAKPOINT EVENT
C
            SUBROUTINE OILDET(R1OIL,CONFAC)
      LOGICAL CAVOUT
      INTEGER DELTAT
      COMMON /BLOK1/ IACC(25,20), RACC(31,20)
      COMMON /BLOK2/ CURVOL,DELCAV,CUROIL,DELOIL,QOIL,IDAY,DELTAT
      COMMON /BLOK5/ FLOW,ICN,ISTG,IPOINT,IZER,NCV,R21,R22,R31,R32
      DELOIL= 0.0
      QOIL  = 0.0
      Q= FLOW
      IF (IACC(2,ICN) .GT. 0 .AND. ISTG .NE. 3) RETURN
      GO TO (70,70,10,20,30,40,50), ISTG
```

```
   10 DELOIL= (((CURVOL+DELCAV)-RACC(20,ICN))/(RACC(21,ICN)-
     *        RACC(20,ICN)))*R1OIL - CUROIL
      IF (DELOIL .LE. 0.) DELOIL = 0.0
      QOIL= DELOIL/FLOAT(DELTAT)
      GO TO 70
   20 QOIL= R21*Q
      GO TO 60
   30 QOIL= R22*Q
      GO TO 60
   40 QOIL= R31*Q
      GO TO 60
   50 QOIL= R32*Q
   60 DELOIL= QOIL*FLOAT(DELTAT)/CONFAC
      IF ( (CUROIL+DELOIL) .LE. RACC(3,ICN) ) GO TO 70
      DELOIL= RACC(3,ICN) - CUROIL
      QOIL  = DELOIL*CONFAC/FLOAT(DELTAT)
   70 CONTINUE
      RETURN
      END
C
C*************************************************************
C
C          SUBROUTINE ZPRNT STORES AND PRINTS OUT THE PERTINENT
C          INFORMATION REQUIRED FOR THE USER
C
            SUBROUTINE ZPRNT(I2)
      LOGICAL CAVOUT
      INTEGER DELTAT
      CHARACTER*2 LCHPHS(2,7)
      COMMON /BLOK1/ IACC(25,20), RACC(31,20)
      COMMON /BLOK2/ CURVOL,DELCAV,CUROIL,DELOIL,QOIL,IDAY,DELTAT
      COMMON /BLOK3/ CAVOUT,LUNLP,LASTDY
      COMMON /BLOK4/ LCHPHS
      COMMON /BLOK5/ FLOW,ICN,ISTG,IPOINT,IZER,NCV,R21,R22,R31,R32
      CALL STORE(CURVOL,DELCAV,CUROIL,DELOIL,QOIL,IDAY,DELTAT)
      ITEMP= IDAY + 1
      IDAY= IDAY + DELTAT
      CURVOL= CURVOL + DELCAV
      IF(ABS(CURVOL-RACC(I2,ICN)).LT.1.E-2)CURVOL=RACC(I2,ICN)
      CUROIL= CUROIL + DELOIL
      IF(CAVOUT) THEN
         WRITE(LUNLP,10) ITEMP,IDAY,(LCHPHS(K,ISTG),K=1,2),
     *                   FLOW,QOIL,CURVOL,CUROIL
         FLO=FLOW
         OIL=QOIL
         WRITE(55,*)ISTG,ITEMP,IDAY,FLO,OIL,NCV
      ENDIF
      IF (CURVOL .NE. RACC(I2,ICN)) CALL ZIDLE(0)
   10 FORMAT(1H ,I5,1H-,I4,3X,2A2,F9.1,F9.2,2F9.0)
      RETURN
      END
C
C*****************************************************************
```

```
C
C         SUBROUTINE ZIDLE PERFORMS THE REQUISITE STEP NECESSARY
C         WHEN AN IDLE IS CALLED FOR BY THE INPUT DATA
C
            SUBROUTINE ZIDLE(IJI)
      LOGICAL CAVOUT
      INTEGER DELTAT
      COMMON /BLOK2/ CURVOL,DELCAV,CUROIL,DELOIL,QOIL,IDAY,DELTAT
      COMMON /BLOK3/ CAVOUT,LUNLP,LASTDY
      COMMON /BLOK5/ FLOW,ICN,ISTG,IPOINT,IZER,NCV,R21,R22,R31,R32
      CALL BRINFL(FLOW,IDAY,LASTDY,ICN)
      IF (FLOW .NE. 0.) RETURN
      DELTAT= LASTDY - IDAY
      CALL STORE(CURVOL,0.,CUROIL,0.,0.,IDAY,DELTAT)
      ITEMP= IDAY + 1
      IDAY= IDAY + DELTAT
      IF(CAVOUT) THEN
         IJIJ=ISTG
         IF(IJI.NE.0)IJIJ=IJIJ+1
         WRITE(LUNLP,10) ITEMP,IDAY
         WRITE(55,*)IJIJ,ITEMP,IDAY,IZER,IZER,NCV
      ENDIF
   10 FORMAT(1H ,I5,1H-,I4,3X,4HIDLE)
      CALL BRINFL(FLOW,IDAY,LASTDY,ICN)
      RETURN
      END
C
C*************************************************************
C
C         SUBROUTINE FILL PERFORMS ALL OF THE STEPS NECESSARY
C         TO FILL THE CAVERN WITH OIL
C
            SUBROUTINE FILL(I6,I7,CONFAC)
      LOGICAL CAVOUT
      INTEGER DELTAT
      COMMON /BLOK1/ IACC(25,20), RACC(31,20)
      COMMON /BLOK2/ CURVOL,DELCAV,CUROIL,DELOIL,QOIL,IDAY,DELTAT
      COMMON /BLOK3/ CAVOUT,LUNLP,LASTDY
      COMMON /BLOK5/ FLOW,ICN,ISTG,IPOINT,IZER,NCV,R21,R22,R31,R32
   10 TVOL= RACC(I6,ICN) - 1.0
      IF (CUROIL .GE. TVOL) RETURN
      QOIL= RACC(I7,ICN)
      FLOW= QOIL
      IF (FLOW .EQ. 0.0) GO TO 15
      DELOIL= RACC(I6,ICN) - CUROIL
      DELTAT= (DELOIL*CONFAC)/FLOW + 1.0
      IF (LASTDY .LT. 0) GO TO 20
      I5= IDAY + DELTAT
      IF ( I5 .LE. LASTDY) GO TO 20
      DELTAT= LASTDY - IDAY
      DELOIL= FLOW*FLOAT(DELTAT)/CONFAC
      GO TO 20
   15 DELOIL= 0.0
```

```
      DELTAT= 1.0
   20 CALL STORE(CURVOL,0.,CUROIL,DELOIL,QOIL,IDAY,DELTAT)
      ITEMP= IDAY+1
      IDAY= IDAY + DELTAT
      CUROIL= CUROIL + DELOIL
      IF(CAVOUT) THEN
         WRITE(LUNLP,30) ITEMP,IDAY,FLOW,QOIL,CURVOL,CUROIL
         FLO=FLOW
         OIL=QOIL
         WRITE(55,*)ISTG,ITEMP,IDAY,FLO,OIL,NCV
      ENDIF
   30 FORMAT(1H ,I5,1H-,I4,3X,4HFILL,F9.1,F9.2,2F9.0)
      IF (CUROIL .GE. TVOL) RETURN
      IF (FLOW .EQ. 0.) RETURN
      CALL ZIDLE(0)
      GO TO 10
      END
C
C*************************************************************
C
C         SUBROUTINE STORE STORES ALL OF THE PERTINENT VALUES
C         FOR LATER PRINTOUT
C
          SUBROUTINE STORE(CURVOL,DELCAV,CUROIL,DELOIL,QOIL,IDAY,DELTAT)
      INTEGER DELTAT
      COMMON /BLOK5/ FLOW,ICN,ISTG,IPOINT,IZER,NCV,R21,R22,R31,R32
      COMMON /BLOK6/ CAVVOL(28,20),OILVOL(28,20),OILFLO(28,20),
     1 ITIME(28,20),BRNFLO(28,20),FINOIL(20)
      IPOINT= IPOINT+1
      CAVVOL(IPOINT,ICN)= CURVOL+DELCAV
      OILVOL(IPOINT,ICN)= CUROIL+DELOIL
      OILFLO(IPOINT,ICN)= QOIL
      BRNFLO(IPOINT,ICN)= FLOW
      ITIME(IPOINT,ICN)= IDAY+DELTAT
      FINOIL(ICN)= OILVOL(IPOINT,ICN)
      RETURN
      END
           SUBROUTINE INPHIL(ERROR)
      INTEGER ERROR, OUT, OLD
      INTEGER TRU, FALS
      COMMON /LUNTS/ INP, OUT, OLD, NEW
      COMMON /BLOK1/ IACCUM(25,20), RACCUM(31,20)
      COMMON /BLOK5/ FLOW,ICN,ISTG,IPOINT,IZER,NCV,R21,R22,R31,R32
      COMMON /SITCM/ FACTOR, ISLIP
      DATA TRU, FALS /-1,0/
      REWIND OLD
      READ (OLD,99999) FACTOR, ISLIP
      WRITE(55,99999)FACTOR,ISLIP
      WRITE(55,99993)R21,R22,R31,R32
      DO 10 KEY=1,20
         READ (OLD,99998) (IACCUM(I,KEY),I=1,12)
         READ (OLD,99997) (IACCUM(I,KEY),I=13,25)
         READ (OLD,99996) (RACCUM(I,KEY),I=1,8)
```

```
      READ (OLD,99996) (RACCUM(I,KEY),I=9,16)
      READ (OLD,99996) (RACCUM(I,KEY),I=17,24)
      READ (OLD,99995) (RACCUM(I,KEY),I=25,31)
      WRITE (55,99998) (IACCUM(I,KEY),I=1,12)
      WRITE (55,99997) (IACCUM(I,KEY),I=13,25)
      WRITE (55,99996) (RACCUM(I,KEY),I=1,8)
      WRITE (55,99996) (RACCUM(I,KEY),I=9,16)
      WRITE (55,99996) (RACCUM(I,KEY),I=17,24)
      WRITE (55,99995) (RACCUM(I,KEY),I=25,31)
   10 CONTINUE
      ERROR = FALS
99999 FORMAT (F6.2, I6)
99998 FORMAT (12I6)
99997 FORMAT (13I6)
99996 FORMAT (8F9.2)
99995 FORMAT (7F9.2)
99993 FORMAT(4F11.6)
      RETURN
      END
C                              ** IDKODE **
C
C              THIS FUNCTION DECODES THE NEXT NON-BLANK STRING IN THE
C       INPUT BUFFER, 'BFR'.  THE SIZE LIMIT OF THE BUFFER IS IN 'BLMT'.
C       THE ARGUMENT, 'PTR', POINTS TO THE NEXT ELEMENT, IN THE BUFFER,
C       TO BE USED.  THE ARRAY, 'ITEMP', PROVIDES TEMPORARY CHARACTER
C       STORAGE DURING THE PARSING --- MUST BE LARGE ENOUGH TO HOLD UP
C       TO THE MAX NUMBER OF DIGITS ALLOWED AS AN INTEGER IN THE HOST
C       MACHINE.  AT PRESENT, THE VARIABLE, 'MXDGIT', IS LIMITED FOR
C       A PDP11 INTEGER SIZE ---   -32,768 TO +32,767  ==> 5 DIGITS.
C
C       ************************************************************
C
          FUNCTION IDKODE(BFR, BFRLMT, IPTR, ERR)
      INTEGER BFRLMT, ERR, DIGLMT
      INTEGER INP, OUT
      INTEGER TRU, FALS
      CHARACTER BFR(1),BLANK,COMMA,SLASH,SEMIC,COLON,MINUS,PLUS
      DIMENSION ITEMP(10)
      COMMON /LUNTS/ INP, OUT, OLD, NEW
      DATA BLANK, COMMA, SLASH, SEMIC, COLON /' ',',','/',';',':'/
      DATA MINUS, PLUS /'-','+'/
      DATA TRU, FALS /-1,0/
      DATA DIGLMT /5/
      DO 10 I=1,10
         ITEMP(I) = 0
   10 CONTINUE
      ERR = FALS
      MFLG = FALS
      NXT = IPTR
      NXT = ISCIPB(BFR,BFRLMT,NXT)
      IF (NXT.GT.BFRLMT) GO TO 90
      IF (BFR(NXT).EQ.PLUS) GO TO 20
      IF (BFR(NXT).NE.MINUS) GO TO 30
```

```
      MFLG = TRU
   20 CONTINUE
      NXT = NXT + 1
      IF (NXT.GT.BFRLMT) GO TO 90
C                        ** FIRST NUMBER **
   30 CONTINUE
      JPTR = 1
      IF (NUMTST(BFR(NXT),IVAL).EQ.FALS) GO TO 100
      ITEMP(JPTR) = IVAL
      NXT = NXT + 1
      JPTR = JPTR + 1
C                        ** THE REST OF THE NUMBERS **
   40 CONTINUE
      IF (NXT.GT.BFRLMT) GO TO 80
      IF (NUMTST(BFR(NXT),IVAL).EQ.FALS) GO TO 60
      IF (JPTR.GT.DIGLMT) GO TO 110
      ITEMP(JPTR) = IVAL
      IF (JPTR.NE.DIGLMT) GO TO 50
      IF (MACTST(ITEMP,JPTR).EQ.TRU) GO TO 110
   50 CONTINUE
      NXT = NXT + 1
      JPTR = JPTR + 1
      GO TO 40
   60 CONTINUE
      IF(     BFR(NXT).EQ.BLANK
     *    .OR.BFR(NXT).EQ.COMMA
     *    .OR.BFR(NXT).EQ.SLASH
     *    .OR.BFR(NXT).EQ.SEMIC
     *    .OR.BFR(NXT).EQ.COLON)     THEN
                                          GO TO 70
      ELSE
         GO TO 120
      ENDIF
   70 CONTINUE
      NXT = NXT + 1
   80 CONTINUE
      IPTR = NXT
      JPTR = JPTR - 1
      IDKODE = IKNVRT(ITEMP,JPTR)
      IF (MFLG.EQ.TRU) IDKODE = -IDKODE
      GO TO 140
C                        ** ERROR MESSAGES **
   90 CONTINUE
      WRITE (OUT,99999)
      GO TO 130
  100 CONTINUE
      WRITE (OUT,99998)
      GO TO 130
  110 CONTINUE
      WRITE (OUT,99997)
      GO TO 130
  120 CONTINUE
      WRITE (OUT,99996)
```

```
  130 CONTINUE
      ERR = TRU
      IPTR = NXT
      IDKODE = 0
  140 CONTINUE
99999 FORMAT (35H  INPUT ERROR:  NO CHARACTERS FOUND)
99998 FORMAT (41H  FORMAT ERROR: CHARACTER IS NOT A NUMBER)
99997 FORMAT (44H  RANGE ERROR: ATTEMPT TO EXCEED MAX INTEGER)
99996 FORMAT (43H  FORMAT ERROR: CHAR IS NOT LEGAL DELIMITER)
      RETURN
      END
C                              ** IKNVRT **
C
C       ************************************************************
C
            FUNCTION IKNVRT(BUF, DSIZ)
      INTEGER BUF(1), DSIZ
      ITMP = 0
      I = 1
   10 CONTINUE
      ITMP = (10*ITMP) + BUF(I)
      I = I + 1
      IF (I.LE.DSIZ) GO TO 10
      IKNVRT = ITMP
      RETURN
      END
C                              ** MACTST **
C       ************************************************************
C
            FUNCTION MACTST(ITEMP, NUM)
      INTEGER TRU, FALS
      DIMENSION MAXINT(10), ITEMP(1)
      DATA MAXINT /3,2,7,6,7,0,0,0,0,0/
      DATA TRU, FALS /-1,0/
      LMT = 5
      MAXFLG = FALS
      IPTR = 1
      IF (NUM.LT.LMT) GO TO 30
   10 CONTINUE
      IF (ITEMP(IPTR).LT.MAXINT(IPTR)) THEN
         GO TO 30
      ELSEIF (ITEMP(IPTR).EQ.MAXINT(IPTR)) THEN
         GO TO 20
      ELSE
         MAXFLG = TRU
         GO TO 30
      ENDIF
   20 CONTINUE
      IPTR = IPTR + 1
      IF (IPTR.GT.NUM) GO TO 30
      GO TO 10
   30 CONTINUE
      MACTST = MAXFLG
```

```
      RETURN
      END
C                              ** NUMTST **
C
C               THIS FUNCTION RETURNS 'TRUE' IF THE ARGUMENT 'ICHAR'
C       CONTAINS AN ASCII DIGIT IN   '0'...'9' .  THE ARGUMENT 'IVAL'
C       WILL CONTAIN THE INTEGER VALUE OF THE LEGAL DIGIT FOUND IN
C       'ICHAR'.
C
C       ************************************************************
C
            FUNCTION NUMTST(ICHAR, IVAL)
      INTEGER FLAG
      INTEGER TRU, FALS
      CHARACTER NUMBS(10),ICHAR
      DATA NUMBS /'0','1','2','3','4','5','6','7','8','9'/
      DATA TRU, FALS /-1,0/
      FLAG = FALS
      I = 1
   10 IF (I.GT.10) GO TO 30
      IF (ICHAR.EQ.NUMBS(I)) GO TO 20
      I = I + 1
      GO TO 10
   20 CONTINUE
      IVAL = I - 1
      FLAG = TRU
      GO TO 40
   30 CONTINUE
      IVAL = -1
   40 CONTINUE
      NUMTST = FLAG
      RETURN
      END
            FUNCTION ISCIPB(BUF, LMT, IPT)
      INTEGER PTR
      CHARACTER BUF(1),BLANK
      DATA BLANK /' '/
      PTR = IPT
   10 CONTINUE
      IF ((PTR.GT.LMT) .OR. (BUF(PTR).NE.BLANK)) GO TO 20
      PTR = PTR + 1
      GO TO 10
   20 CONTINUE
      ISCIPB = PTR
      RETURN
      END
```

UNCLASSIFIED

END

DATE
FILMED

JULY 87

SANDIA NATIONAL LABS

APPENDIX B

NETWRK FILE LISTING

This appendix contains a listing of the file NETWRK which is an input file used by SLAM. The statements in this file are a description, in SLAM syntax, of the scheduling of leaching and filling activities at an SPR site.

SECURITY CLASSIFICATION

UNCLASSIFIED

DATE OF MICROFILMING

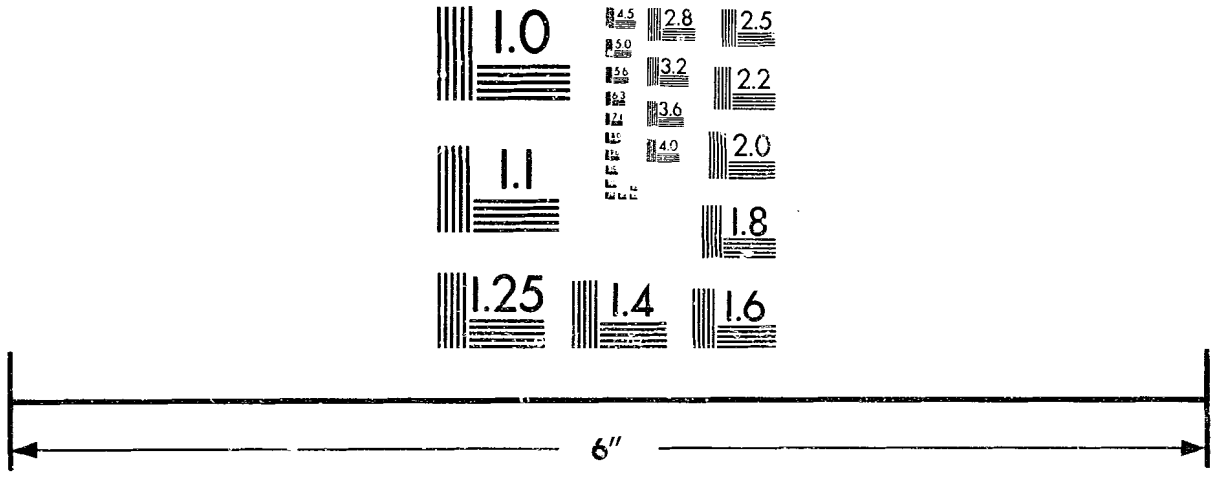
5-18-87

MICROFILMED BY

F. Mitchell

SANDIA NATIONAL LABS

IMAGE EVALUATION TEST TARGET (MT-3)



PHOTOGRAPHIC SCIENCES CORPORATION
770 BASKET ROAD
P.O. BOX 338
WEBSTER, NEW YORK 14580
(716) 265-1600

```
;
;                    NETWRK FILE
;
;  SLAM NETWORK DESCRIPTION OF STRATEGIC PETROLEUM RESERVE SCHEDULING
;
;       CONTROL STATEMENTS
;
GEN,D W SASSER,SPR,10/01/86,,N,N,,N,N;
LIMITS,13,40,60;
STAT,1,CAVERN DONE TIME;
TIMST,XX(50),IF FILL 1 ELSE 0;
PRIORITY/1,LVF(27)/2,LVF(27)/3,LVF(27)/4,LVF(27)/?,LVF(27);
PRIORITY/5,LVF(27)/6,LVF(27)/NCLNR,LVF(27);
CONT,0,100,1,1;
RECORD,TNOW,TIME,B,B,10,,,N;
VAR,XX(40),B,BRINE FLOW,0,2000;
VAR,XX(41),O,OIL FLOW,0,1000;
VAR,XX(42),C,CAVERN VOLUME,0,200000;
VAR,XX(43),V,OIL VOLUME,0,200000;
;
;
;
NETWORK;
;
; RESOURCES AND GATES
;
      RESOURCE/BRINE(0),9;
      RESOURCE/WO1 OR 3(0),5;
      RESOURCE/WO2(0),6;
      RESOURCE/OIL(0),2;
      RESOURCE/DUM(1000),7;
      RESOURCE/HOLD1(0),1;
      RESOURCE/HOLD3(0),3;
      RESOURCE/HOLD4(0),4;
      GATE/GATE1,CLOSE,10;
      GATE/GATE2,CLOSE,11;
      GATE/GATE3,CLOSE,12;
      GATE/GATE4,CLOSE,13;
;
;            START OF NETWORK ROUTING
;
;   EACH CAVERN WITH ATRIB(27)=0 ARRIVES AT THIS AWAIT NODE
;   AND THEN IS PLACED IN ITS STARTING PLACE IN THE NETWORK
;   VIA THE BRANCHES FROM THE ASSIGN NODE G1 BELOW.
;
AW7   AWAIT(7),DUM/1,1;
      ACT,,ATRIB(27).LE.0,G1;
      ACT;
;
;   A DUMMY CAVERN WITH ATRIB(27)=1 ARRIVES, AFTER ALL CAVERNS
;   HAVE BEEN PLACED, TO ALTER THE BRINE, OIL AND WORKOVER
;   RESOURCES (WHICH ARE INITIALLY ZERO) TO THEIR APPROPRIATE
```

```
;   VALUES.
;
      ALTER,OIL/XX(2);
      ALTER,BRINE/XX(1);
      ALTER,WO1 OR 3/XX(6);
      ALTER,WO2/XX(5);
;
;   IF CAVERNS AWAITING F1 OR F2, RELEASE ONE
;   DESTROY DUMMY CAVERN
;
      ACT,,NNQ(1).GT.0,H1;
;
;   ELSE IF CAVERNS AWAITING F3, RELEASE ONE
;   DESTROY DUMMY CAVERN
;
      ACT,,NNQ(3).GT.0,H3;
;
;   ELSE IF CAVERNS AWAITING LEACHING AND TOTAL BRINE REQUIREMENTS
;   FOR THESE CAVERNS IS LESS THAN MAXIMUM SITE BRINE FLOW, CALL
;   SUBROUTINE PREVUE TO SEE IF ANY CAVERN BRINE FLOW RATES CAN BE
;   INCREASED. THEN RELEASE ONE CAVERN AND DESTROY DUMMY CAVERN.
;
      ACT,,XX(31).LT.NNRSC(1).AND.NNQ(4).NE.0,E66;
;
;   ELSE IF CAVERNS AWAITING LEACHING, RELEASE ONE
;   DESTROY DUMM. CAVERN
;
      ACT,,NNQ(4).GT.0,H4;
E66   EVENT,66;             CALLS SUBROUTINE PREVUE
      ACT,,,H4;
H1    ALTER,HOLD1/1;
      TERM;
H3    ALTER,HOLD3/1;
      TERM;
H4    ALTER,HOLD4/1;
      TERM;
;
;   AFTER A RESTART, A DUMMY CAVERN IS ENTERED HERE TO BEGIN THE
;   SCHEDULING OF CAVERNS AGAIN. A RESTART OCCURS AT EACH BREAKPOINT.
;
      ENTER,9,1;
      GOON,1;
      ACT,,NNQ(1).NE.0,H1;
      ACT,,NNQ(3).NE.0,H3;
      ACT,,XX(31).LT.NNRSC(1).AND.NNQ(4).NE.0,E59;
      ACT,,NNQ(4).NE.0,H4;
E59   EVENT,59;
      ACT,,,H4;
G1    ASSIGN,ATRIB(27)=ATRIB(28),ATRIB(31)=-1.,ATRIB(28)=0.,1;
      ACT,,ATRIB(14).GT.-1.,G00;
      ACT,,ATRIB(2).GT.-1.,W1;
      ACT,,ATRIB(16).GT.-1.,G06;
      ACT ,ATRIB(4).GT.-1.,W2;
```

```
      ACT,,ATRIB(18).GT.-1.,G07;
      ACT,,ATRIB(6).GT.-1.,W3;
      ACT,,ATRIB(20).GT.-1.,EN4;
      ACT,,ATRIB(21).GT.-1.,G08;
      ACT,,ATRIB(9).GT.-1.,W4;
      ACT,,ATRIB(23).GT.-1.,EN6;
      ACT,,ATRIB(24).GT.-1.,G09;
      ACT,,ATRIB(12).GT.-1.,W5;
      ACT,,ATRIB(26).GT.-1.,EN8;
;
;   XX(32) ACCUMULATES THE CURRENT OIL FLOW REQUIREMENTS FOR
;   ALL CAVERNS AWAITING LEACHING. THIS IS USED TO REDUCE THE
;   OIL FLOW AVAILABLE FOR OIL FILLING.
;
G00   GOON,1;
      ACT,,ATRIB(40).LE.-2.,E67;
      ACT,,,G00A;
E67   EVENT,67;
G00A  ASSIGN,II=USERF(1),XX(32)=XX(32)+ATRIB(15);
;
;   XX(31) ACCUMULATES THE CURRENT OIL FLOW AND BRINE FLOW REQUIREMENTS
;   FOR ALL CAVERNS AWAITING LEACHING. THIS IS USED TO SEE IF OIL AND
;   BRINE FLOW REQUIREMENTS ARE LESS THAN THE SITE MAXIMUM BRINE FLOW.
;
      ASSIGN,XX(31)=XX(31)+ATRIB(14),XX(31)=XX(31)+ATRIB(15);
;
;   CAVERNS AWAITING SUMP LEACH ENTER HERE AFTER A RESTART.
;
EN1   ENTER,1;
;
;   CAVERNS AWAITING SUMP LEACH ARE HELD HERE AND RELEASED ONE AT
;   A TIME.
;
HS    AWAIT(4),HOLD4/1;
;
;   EVENT 51 DETERMINES IF THERE ARE SUFFICIENT BRINE AND OIL FLOWS
;   AVAILABLE FOR A CAVERN AND, IF NOT, REDUCES THE FLOWS TO THE
;   CAVERN IF POSSIBLE.
;
E51   EVENT,51,1;
;
;   IF INSUFFICIENT BRINE OR OIL FLOWS ARE AVAILABLE FOR A CAVERN,
;   THE CAVERN IS NOT SCHEDULED.
;
      ACT,,ATRIB(30).EQ.-1,G01;
;
;   OTHERWISE CAVERN IS SCHEDULED. XX(21) IS OIL FLOW, XX(20) IS
;   SUMP LEACHING BRINE FLOW PLUS OIL FLOW (THE TOTAL BRINE FLOW).
;
      ACT,,XX(21).NE.0.AND.XX(20).NE.0,OS;
      ACT,,XX(21).EQ.0.AND.XX(20).NE.0,BS;
      ACT,,,ZF1;
G01   GOON,1;
```

```
;
;   CAVERN CANNOT BE SCHEDULED. IF OTHER CAVERNS AWAIT SUMP LEACHING,
;   RELEASE ONE AND SEND THE UNSCHEDULED CAVERN TO GATE1 TO BE HELD
;   UNTIL ALL CAVERNS AWAITING SUMP LEACHING ARE EXAMINED.
;
      ACT,,NNQ(4).NE.0,GT1;
;
;   CAVERN CANNOT BE SCHEDULED AND NO OTHER CAVERNS AWAIT SUMP LEACHING.
;   OPEN GATE1, SENDING ALL UNSCHEDULED CAVERNS TO AWAIT HOLD4.
;
      ACT,,NNQ(4).EQ.0;
      OPEN,GATE1;
      CLOSE,GATE1;
      ACT,,,HS;
GT1   ALTER,HOLD4/1;
      AWAIT(10),GATE1;
      ACT,,,HS;
;
;   ALLOCATE BRINE AND OIL FLOWS TO CAVERN FOR SUMP LEACHING
;
OS    AWAIT(2),OIL/XX(21);
BS    AWAIT(9),BRINE/XX(20);
ZF1   GOON,1;
;
;   CAVERNS AWAITING SUMP LEACH, RELEASE ONE.
;
      ACT,,NNQ(4).NE.0,FR6;
;
;   NO CAVERNS AWAITING SUMP LEACH, OPEN GATE1.
;
      ACT,,NNQ(4).EQ.0;
      OPEN,GATE1;
      CLOSE,GATE1;
      ACT,,,G6;
FR6   ALTER,HOLD4/1;
G6    GOON;
;
;   BEGIN SUMP LEACH;
;
      ACT,ATRIB(1);
;
;   SUMP LEACH COMPLETED. EVENT 1 DOES SOME NECESSARY RECORD KEEPING
;   AND THEN CALLS SUBROUTINE RESTRT TO BEGIN SCHEDULING OF CAVERNS
;   AGAIN.
;
E1    EVENT,1,1;
;
;   SEND CAVERN TO NEXT STAGE OF DEVELOPMENT.
;
      ACT,,ATRIB(1).GT.-1.,G00;
      ACT,,ATRIB(2).GT.-1.,W1;
      ACT,,ATRIB(16).GT.-1.,G06;
      ACT,,ATRIB(18).GT.-1.,G07;
```

```
      ACT,,ATRIB(20).GT.-1.,EN4;
      ACT,,ATRIB(21).GT.-1.,GO8;
      ACT,,ATRIB(23).GT.-1.,EN6;
      ACT,,ATRIB(24).GT.-1.,GO9;
      ACT,,ATRIB(26).GT.-1.,EN8;
      ACT,,,E8;
;
;   WAIT FOR WO1
;
W1    AWAIT(5),WO1 OR 3/1;
E9    EVENT,9;            DETERMINES WORKOVER TIME
;
;   BEGIN WO1
;
      ACT,ATRIB(2);
;
;   END WO1, WAIT FOR WO2
;
E10   EVENT,10;           RECORD KEEPING
      AWAIT(6),WO2/1;
E11   EVENT,11;           DETERMINES WORKOVER TIME
;
;   BEGIN WO2
;
      ACT,ATRIB(2);
;
;   END WO2, WAIT FOR WO3
;
E12   EVENT,12;           RECORD KEEPING
      AWAIT(5),WO1 OR 3/1;
E13   EVENT,13;           DETERMINES WORKOVER TIME
;
;   BEGIN WO3
;
      ACT,ATRIB(2);
;
;   END WO3
;
E14   EVENT,14,1;         RECORD KEEPING, CALLS RESTRT
;
;   SEND CAVERN TO NEXT STAGE OF DEVELOPMENT.
;
      ACT,,ATRIB(16).GT.-1.,GO6;
      ACT,,ATRIB(18).GT.-1.,GO7;
      ACT,,ATRIB(20).GT.-1.,EN4;
      ACT,,ATRIB(21).GT.-1.,GO8;
      ACT,,ATRIB(23).GT.-1.,EN6;
      ACT,,ATRIB(24).GT.-1.,GO9;
      ACT,,ATRIB(26).GT.-1.,EN8;
      ACT,,,E8;
;
;*************************************************************************
;             THE COMMENTS ABOVE DESCRIBE THE LOGIC FOR THE ROUTE OF A
```

```
;            CAVERN THROUGH SUMP LEACH AND THE SUBSEQUENT WORKOVER. THE
;            LOGIC IS SIMILAR FOR THE OTHER LEACH STAGES AND THEREFORE
;            COMMENTS WILL NOT BE REPEATED.
;*********************************************************************
;
G06   GOON,1;
      ACT,,ATRIB(40).LE.-2.,E68;
      ACT,,,G06A;
E68   EVENT,68;
G06A  ASSIGN,II=USERF(2),XX(32)=XX(32)+ATRIB(17);
AS9   ASSIGN,XX(31)=XX(31)+ATRIB(16),XX(31)=XX(31)+ATRIB(17);
EN2   ENTER,2;
HSC   AWAIT(4),HOLD4/1;
E52   EVENT,52,1;
      ACT,,ATRIB(30).EQ.-1,G02;
      ACT,,XX(21).NE.0.AND.XX(20).NE.0,OSC;
      ACT,,XX(21).EQ.0.AND.XX(20).NE.0,BSC;
      ACT,,,ZF2;
G02   GOON,1;
      ACT,,NNQ(4).NE.0,GT2;
      ACT,,NNQ(4).EQ.0;
      OPEN,GATE1;
      CLOSE,GATE1;
      ACT,,,HSC;
GT2   ALTER,HOLD4/1;
      AWAIT(10),GATE1;
      ACT,,,HSC;
OSC   AWAIT(2),OIL/XX(21);
BSC   AWAIT(9),BRINE/XX(20);
ZF2   GOON,1;
      ACT,,NNQ(4).NE.0,FR7;
      ACT,,NNQ(4).EQ.0;
      OPEN,GATE1;
      CLOSE,GATE1;
      ACT,,,G7;
FR7   ALTER,HOLD4/1;
G7    GOON;
      ACT,ATRIB(3);
E2    EVENT,2,1;
      ACT,,ATRIB(3).GT.-1.,G06;
      ACT,,ATRIB(4).GT.-1.,W2;
      ACT,,ATRIB(18).GT.-1.,G07;
      ACT,,ATRIB(20).GT.-1.,EN4;
      ACT,,ATRIB(21).GT.-1.,G08;
      ACT,,ATRIB(23).GT.-1.,EN6;
      ACT,,ATRIB(24).GT.-1.,G09;
      ACT,,ATRIB(26).GT.-1.,EN8;
      ACT,,,E8;
W2    AWAIT(5),W01 OR 3/1;
E15   EVENT,15;
      ACT,ATRIB(4);
E16   EVENT,16;
      AWAIT(6),W02/1;
```

```
E17   EVENT,17;
      ACT,ATRIB(4);
E18   EVENT,18;
      AWAIT(5),W01 OR 3/1;
E19   EVENT,19;
      ACT,ATRIB(4);
E20   EVENT,20,1;
      ACT,,ATRIB(18).GT.-1.,G07;
      ACT,,ATRIB(20).GT.-1.,EN4;
      ACT,,ATRIB(21).GT.-1.,G08;
      ACT,,ATRIB(23).GT.-1.,EN6;
      ACT,,ATRIB(24).GT.-1.,G09;
      ACT,,ATRIB(26).GT.-1.,EN8;
      ACT,,,E8;
G07   GOON,1;
      ACT,,ATRIB(40).LE.-2.,E69;
      ACT,,,G07A;
E69   EVENT,69;
G07A  ASSIGN,II=USERF(3),XX(32)=XX(32)+ATRIB(19);
AS3   ASSIGN,XX(31)=XX(31)+ATRIB(18),XX(31)=XX(31)+ATRIB(19);
EN3   ENTER,3;
HR1   AWAIT(4),HOLD4/1;
E53   EVENT,53,1;
      ACT,,ATRIB(30).EQ.-1,G03;
      ACT,,XX(21).NE.0.AND.XX(20).NE.0,OR1;
      ACT,,XX(21).EQ.0.AND.XX(20).NE.0,BR1;
      ACT,,,ZF3;
G03   GOON,1;
      ACT,,NNQ(4).NE.0,GT3;
      ACT,,NNQ(4).EQ.0;
      OPEN,GATE1;
      CLOSE,GATE1;
      ACT,,,HR1;
GT3   ALTER,HOLD4/1;
      AWAIT(10),GATE1;
      ACT,,,HR1;
OR1   AWAIT(2),OIL/XX(21);
BR1   AWAIT(9),BRINE/XX(20);
ZF3   GOON,1;
      ACT,,NNQ(4).NE.0,FR8;
      ACT,,NNQ(4).EQ.0;
      OPEN,GATE1;
      CLOSE,GATE1;
      ACT,,,G8;
FR8   ALTER,HOLD4/1;
G8    GOON;
      ACT,ATRIB(5);
E3    EVENT,3,1;
      ACT,,ATRIB(5).GT.-1.,G07;
      ACT,,ATRIB(6).GT.-1.,W3;
      ACT,,ATRIB(20).GT.-1.,EN4;
      ACT,,ATRIB(21).GT.-1.,G08;
      ACT,,ATRIB(23).GT.-1.,EN6;
```

```
      ACT,,ATRIB(24).GT.-1.,GO9;
      ACT,,ATRIB(26).GT.-1.,EN8;
      ACT,,,E8;
W3    AWAIT(5),WO1 OR 3/1;
E21   EVENT,21;
      ACT,ATRIB(6);
E22   EVENT,22;
      AWAIT(6),WO2/1;
E23   EVENT,23;
      ACT,ATRIB(6);
E24   EVENT,24;
      AWAIT(5),WO1 OR 3/1;
E25   EVENT,25;
      ACT,ATRIB(6);
E26   EVENT,26,1;
      ACT,,ATRIB(20).GT.-1.,EN4;
      ACT,,ATRIB(21).GT.-1.,GO8;
      ACT,,ATRIB(23).GT.-1.,EN6:
      ACT,,ATRIB(24).GT.-1.,GO9;
      ACT,,ATRIB(26).GT.-1.,EN8;
      ACT,,,E8;
EN4   ENTER,4;
;
;   CAVERNS AWAITING OIL FILLING (F1 OR F2) ARE HELD HERE AND
;   ARE RELEASED ONE AT A TIME.
;
HO1   AWAIT(1),HOLD1/1;
;
;   EVENT 54 DETERMINES IF THERE IS SUFFICIENT OIL FLOW AVAILABLE FOR
;   SCHEDULING OF OIL FILLING OF CAVERN.
;
E54   EVENT,54,1;
;
;   CAVERN CANNOT BE SCHEDULED FOR F1 OIL FILLING. OTHER CAVERNS AWAIT
;   F1 OR F2 OIL FILLING, RELEASE ONE AND SEND THE UNSCHEDULED CAVERN
;   TO GATE2 TO BE HELD UNTIL ALL CAVERNS AWAITING F1 OR F2 OIL FILLING
;   ARE EXAMINED.
;
      ACT,,ATRIB(30).EQ.-1.AND.NNQ(1).NE.0,FR1A;
      ACT;
;
;   CAVERN CANNOT BE SCHEDULED FOR F1 OIL FILLING AND NO OTHER CAVERNS
;   AWAIT F1 OR F2 OIL FILLING. OPEN GATE2, SENDING ALL CAVERNS AWAITING
;   F1 OR F2 OIL FILLING TO AWAIT HOLD1.
;
      OPEN,GATE2;
      CLOSE,GATE2,1;
;
;   CAVERN CANNOT BE SCHEDULED FOR F1 OIL FILLING AND NO OTHER CAVERNS
;   AWAIT F1 OR F2 OIL FILLING. CAVERNS AWAIT F3 OIL FILLING, RELEASE
;   ONE AND SEND UNSCHEDULED CAVERN TO AWAIT HOLD..
;
      ACT,,ATRIB(30).EQ.-1.AND.NNQ(3).NE.0,FR1;
```

```
;
;   CAVERN CANNOT BE SCHEDULED FOR F1 OIL FILLING AND NO OTHER CAVERNS
;   AWAIT F1 OR F2 OR F3 OIL FILLING. CAVERNS AWAITING LEACH, RELEASE
;   ONE AND SEND UNSCHEDULED CAVERN TO AWAIT HOLD1. IF TOTAL BRINE FLOW
;   NEEDED FOR ALL CAVERNS IS LESS THAN SITE MAXIMUM BRINE CALL PREVUE
;   TO SEE IF ANY CAVERN BRINE FLOWS CAN BE INCREASED.
;
      ACT,,ATRIB(30).EQ.-1.AND.NNQ(4).NE.0,FR2;
;
;   CAVERN CANNOT BE SCHEDULED FOR F1 OIL FILLING AND NO OTHER CAVERNS
;   AWAIT F1 OR F2 OR F3 OIL FILLING OR LEACH. SEND UNSCHEDULED CAVERN
;   TO AWAIT HOLD1.
;
      ACT,,ATRIB(30).EQ.-1,HO1;
;
;   CAVERN SCHEDULED FOR F1 OIL FILLING
;
      ACT,,,OF1;
FR1A  ALTER,HOLD1/1;
      AWAIT(11),GATE2;
      ACT,,,HO1;
FR1   ALTER,HOLD3/1;
      ACT,,,HO1;
FR2   GOON,1;
      ACT,,XX(31).GE.NNRSC(1),AL1;
      ACT;
E60   EVENT,60;             CALLS SUBROUTINE PREVUE
AL1   ALTER,HOLD4/1;
      ACT,,,EN4;
;
;   ALLOCATE BRINE AND OIL FLOWS FOR F1 OIL FILLING
;
OF1   AWAIT(2),OIL/XX(21);
BF1   AWAIT(9),BRINE/XX(20),1;
;
;   IF CAVERNS AWAIT F1 OR F2 OIL FILLING, RELEASE ONE
;
      ACT,,NNQ(1).NE.0,FRB;
;
;   IF NO CAVERNS AWAIT F1 OR F2 OIL FILLING BUT CAVERNS AWAIT F3
;   OIL FILLING, RELEASE ONE.
;
      ACT,,NNQ(3).NE.0,FRC;
;
;   IF NO CAVERNS AWAIT F1 OR F2 OR F3 OIL FILLING BUT CAVERNS AWAIT
;   LEACH, RELEASE ONE.
;
      ACT,,NNQ(4).NE.0,FRD;
;
;   NO CAVERNS AWAIT OIL FILLING OR LEACH ACTIVITIES, CONTINUE
;
      ACT,,,GB;
FRB   ALTER,HOLD1/1;
```

```
      ACT,,,GB;
FRC   ALTER,HOLD3/1;
      ACT,,,GB;
FRD   GOON,2;
      ACT,,,GB;
      ACT,,XX(31).LT.NNRSC(1),E63;
      ACT,,,AL4;
;
;   IF TOTAL BRINE FLOW REQUIREMENT FOR ALL CAVERNS IS LESS THAN
;   SITE MAXIMUM BRINE FLOW, CALL PREVUE.
;
E63   EVENT,63;
AL4   ALTER,HOLD4/1;
      TERM;
;
;   BEGIN F1 OIL FILLING
;
GB    GOON;
      ACT,ATRIB(7);
;
;   F1 OIL FILLING COMPLETED
;
E4    EVENT,4,1;            RECORD KEEPING, CALLS SUBROUTINE PREVUE
;
;   SEND CAVERN TO NEXT STAGE OF DEVELOPMENT
;
      ACT,,ATRIB(21).GT.-1.,G08;
      ACT,,ATRIB(23).GT.-1.,EN6;
      ACT,,ATRIB(24).GT.-1.,G09;
      ACT,,ATRIB(26).GT.-1.,EN8;
      ACT,,,E8;
;
;*********************************************************************
;             LOGIC FOR F2 OR F3 OIL FILLING IS SIMILAR AND THEREFORE
;             COMMENTS WILL NOT BE REPEATED.
;*********************************************************************
;
G08   GOON,1;
      ACT,,ATRIB(40).LE.-2.,E70;
      ACT,,,G08A;
E70   EVENT,70;
G08A  ASSIGN,II=USERF(4),XX(32)=XX(32)+ATRIB(22);
AS5   ASSIGN,XX(31)=XX(31)+ATRIB(21),XX(31)=XX(31)+ATRIB(22);
EN5   ENTER,5;
HR2   AWAIT(4),HOLD4/1;
E55   EVENT,55,1;
      ACT,,ATRIB(30).EQ.-1,G04;
      ACT,,XX(21).NE.0.AND.XX(20).NE.0,OR2;
      ACT,,XX(21).EQ.0.AND.XX(20).NE.0,BR2;
      ACT,,,ZF4;
G04   GOON,1;
      ACT,,NNQ(4).NE.0,GT4;
      ACT,,NNQ(4).EQ.0;
```

```
      OPEN,GATE1;
      CLOSE,GATE1;
      ACT,,,HR2;
GT4   ALTER,HOLD4/1;
      AWAIT(10),GATE1;
      ACT,,,HR2;
OR2   AWAIT(2),OIL/XX(21);
BR2   AWAIT(9),BRINE/XX(20);
ZF4   GOON,1;
      ACT,,NNQ(4).NE.0,FR9;
      ACT,,NNQ(4).EQ.0;
      OPEN,GATE1;
      CLOSE,GATE1;
      ACT,,,G9;
FR9   ALTER,HOLD4/1;
G9    GOON;
      ACT,ATRIB(8);
E5    EVENT,5,1;
      ACT,,ATRIB(8).GT.-1.,GO8;
      ACT,,ATRIB(9).GT.-1.,W4;
      ACT,,ATRIB(32).GT.-1.,AS1;
      ACT,,ATRIB(23).GT.-1.,EN6;
      ACT,,ATRIB(24).GT.-1.,GO9;
      ACT,,ATRIB(26).GT.-1.,EN8;
      ACT,,,E8;
AS1   ASSIGN,ATRIB(21)=ATRIB(32),ATRIB(22)=ATRIB(33);
      ASSIGN,ATRIB(8)=ATRIB(34),ATRIB(9)=ATRIB(35),ATRIB(32)=-2.;
      ACT,,,GO8;
W4    AWAIT(5),WO1 OR 3/1;
E27   EVENT,27;
      ACT,ATRIB(9);
E28   EVENT,28;
      AWAIT(6),WO2/1;
E29   EVENT,29;
      ACT,ATRIB(9);
E30   EVENT,30;
      AWAIT(5),WO1 OR 3/1;
E31   EVENT,31;
      ACT,ATRIB(9);
E32   EVENT,32,1;
      ACT,,ATRIB(32).GT.-1.,AS1;
      ACT,,ATRIB(23).GT.-1.,EN6;
      ACT,,ATRIB(24).GT.-1.,GO9;
      ACT,,ATRIB(26).GT.-1.,EN8;
      ACT,,,E8;
EN6   ENTER,6;
HO2   AWAIT(1),HOLD1/1;
E56   EVENT,56,1;
      ACT,,ATRIB(30).EQ.-1.AND.NNQ(1).NE.0,FR3A;
      ACT;
      OPEN,GATE3;
      CLOSE,GATE3,1;
      ACT,,ATRIB(30).EQ.-1.AND.NNQ(3).NE.0,FR3;
```

```
      ACT,,ATRIB(30).EQ.-1.AND.NNQ(4).NE.0,FR4;
      ACT,,ATRIB(30).EQ.-1,EN6;
      ACT,,,OF2;
FR3A  ALTER,HOLD1/1;
      AWAIT(12),GATE3;
      ACT,,,EN6;
FR3   ALTER,HOLD3/1;
      ACT,,,EN6;
FR4   GOON,1;
      ACT,,XX(31).GE.NNRSC(1),AL2;
      ACT;
E61   EVENT,61;
AL2   ALTER,HOLD4/1;
      ACT,,,EN6;
OF2   AWAIT(2),OIL/XX(21);
BF2   AWAIT(9),BRINE/XX(20),1;
      ACT,,NNQ(1).NE.0,FRE;
      ACT,,NNQ(3).NE.0,FRF;
      ACT,,NNQ(4).NE.0,FRG;
      ACT,,,GC;
FRE   ALTER,HOLD1/1;
      ACT,,,GC;
FRF   ALTER,HOLD3/1;
      ACT,,,GC;
FRG   GOON,2;
      ACT,,,GC;
      ACT,,XX(31).LT.NNRSC(1),E64;
      ACT,,,AL5;
E64   EVENT,64;
AL5   ALTER,HOLD4/1;
      TERM;
GC    GOON;
      ACT,ATRIB(10);
E6    EVENT,6,1;
      ACT,,ATRIB(24).GT.-1.,GO9;
      ACT,,ATRIB(26).GT.-1.,EN8;
      ACT,,,E8;
GO9   GOON,1;
      ACT,,ATRIB(40).LE.-2.,E71;
      ACT,,,GO9A;
E71   EVENT,71;
GO9A  ASSIGN,II=USERF(5),XX(32)=XX(32)+ATRIB(25);
AS7   ASSIGN,XX(31)=XX(31)+ATRIB(24),XX(31)=XX(31)+ATRIB(25);
EN7   ENTER,7;
HR3   AWAIT(4),HOLD4/1;
E57   EVENT,57,1;
      ACT,,ATRIB(30).EQ.-1,GO5;
      ACT,,XX(21).NE.0.AND.XX(20).NE.0,BR3;
      ACT,,XX(21).EQ.0.AND.XX(20).NE.0,BR3;
      ACT,,,ZF5;
GO5   GOON,1;
      ACT,,NNQ(4).NE.0,GT5;
      ACT,,NNQ(4).EQ.0;
```

```
      OPEN,GATE1;
      CLOSE,GATE1;
      ACT,,,HR3;
GT5   ALTER,HOLD4/1;
      AWAIT(10),GATE1;
      ACT,,,HR3;
OR3   AWAIT(2),OIL/XX(21);
BR3   AWAIT(9),BRINE/XX(20);
ZF5   GOON,1;
      ACT,,NNQ(4).NE.0,FRA;
      ACT,,NNQ(4).EQ.0;
      OPEN,GATE1;
      CLOSE,GATE1;
      ACT,,,GA;
FRA   ALTER,HOLD4/1;
GA    GOON;
      ACT,ATRIB(11);
E7    EVENT,7,1;
      ACT,,ATRIB(11).GT.-1.,G09;
      ACT,,ATRIB(12).GT.-1.,W5;
      ACT,,ATRIB(36).GT.-1.,AS2;
      ACT,,ATRIB(26).GT.-1.,EN8;
      ACT,,,E8;
AS2   ASSIGN,ATRIB(24)=ATRIB(36),ATRIB(25)=ATRIB(37);
      ASSIGN,ATRIB(11)=ATRIB(38),ATRIB(12)=ATRIB(39),ATRIB(36)=-2.;
      ACT,,,G09;
W5    AWAIT(5),W01 OR 3/1;
E33   EVENT,33,
      ACT,ATRIB(12);
E34   EVENT,34;
      AWAIT(6),W02/1;
E35   EVENT,35;
      ACT,ATRIB(12);
E36   EVENT,36;
      AWAIT(5),W01 OR 3/1;
E37   EVENT,37;
      ACT,ATRIB(12);
E38   EVENT,38,1;
      ACT,,ATRIB(36).GT.-1.,AS2;
      ACT,,ATRIB(26).GT.-1.,EN8;
      ACT,,,E8;
EN8   ENTER,8;
H03   AWAIT(3),HOLD3/1;
E58   EVENT,58,1;
      ACT,,ATRIB(30).EQ.-1.AND.NNQ(3).NE.0,FR5A;
      ACT;
      OPEN,GATE4;
      CLOSE,GATE4,1;
      ACT,,ATRIB(30).EQ.-1.AND.NNQ(4).EQ.0,EN8;
      ACT,,ATRIP(30).EQ.-1.AND.NNQ(4).NE.0,FR5;
      ACT,,,OF3;
FR5A  ALTER,HOLD3/1;
      AWAIT(13),GATE4;
```

```
      ACT,,,EN8;
FR5   GOON,1;
      ACT,,XX(31).GE.NNRSC(1),AL3;
      ACT;
E62   EVENT,62;
AL3   ALTER,HOLD4/1;
      ACT,,,EN8;
OF3   AWAIT(2),OIL/XX(21);
BF3   AWAIT(9),BRINE/XX(20),1;
      ACT,,NNQ(3).NE.0,FRH;
      ACT,,NNQ(4).NE.0,FRI;
      ACT,,,GD;
FRH   ALTER,HOLD3/1;
      ACT,,,GD;
FRI   GOON,2;
      ACT,,,GD;
      ACT,,XX(31).LT.NNRSC(1),E65;
      ACT,,,AL6;
E65   EVENT,65;
AL6   ALTER,HOLD4/1;
      TERM;
GD    GOON;
      ACT,ATRIB(13);
E8    EVENT,8,1;
      TERM;
      END;
FIN;
```

UNCLASSIFIED

END

DATE
FILMED

MAY 87

SANDIA NATIONAL LABS

APPENDIX C

CDC NOS PROCEDURE FILE

This appendix contains a listing of the procedure file which is used on the CDC time-sharing system at SNL to execute the program SPRSKD.

APPENDIX C

CDC NOS PROCEDURE FILE

This appendix contains a listing of the procedure file which is used on the CDC time-sharing system at SNL to execute the program SPRSKD.

BSLAM is the FORTRAN-compiled program SLAM; BSPRSKD is the FORTRAN-compiled program SPRSKD; and NETWRK is the network input file to SLAM. The version of SLAM used by SNL utilizes overlay structure with FILE2 as the primary file in this structure. Overlay structure may not be necessary on other computers.

```
.PROC,SPRSKED.
REWIND,*.
GET,BSPRSKD/UN=DKBUCHA.
GET,BSLAM/UN=DKBUCHA.
GET,NETWRK/UN=DKBUCHA.
NOEXIT.
LIBEDIT,P=BSLAM,N-NSLAM,I=0,B=BSPRSKD,L=0.
ONEXIT.
RETURN,BSLAM.
RENAME,BSLAM=NSLAM.
LIBGEN,F=BSPRSKD,P=YYYSLIB.
LDSET(MAP=SBX/YYYYMAP,LIB=YYYSLIB,USEP=BLOKDT,USEP=BLOCDT,PRESET=0)
LOAD(BSLAM)
NOGO.
FILE1.
REVERT. SCHEDULE PROGRAM TERMINATED NORMALLY
EXIT.
REVERT,ABORT.  RUN ABORTED
```

APPENDIX D

SAMPLE SPRSKD INPUT/OUTPUT

This appendix is provided to illustrate a sample interactive session between the user and SPRSKD. The narrative portion discusses the inputs required, and the actual computer output follows. The narrative discusses each input as the reader follows along using the output listing.

A SPRSKD session is initiated on SNL's computer by entering the command BEGIN,, SPRSKED. (SPRSKED is a CDC/NOS procedure file and is listed in Appendix C.) All user inputs are underlined and preceded with a ? prompt. When a RETURN key entry is the only requirement to enter the proper information, e.g. the default case, then no input is shown--just the ? prompt.

In response to the BEGIN,, SPRSKED command, SPRSKD displays a menu (see page D-15). This menu indicates that three subprograms are available (EDITOR, PREDICT and AUTSKED), and that the user has the option to select any of them. If the user is starting from scratch to develop a site schedule, then initial data must be entered into EDITOR. The output of EDITOR is then used by PREDICT to develop individual cavern schedules. Finally, the output of PREDICT is used by AUTSKED to develop a schedule for an entire site.

The example used in this appendix is the development of a two-cavern site. Most input is similar to that which is required to develop a larger site, such as 14 caverns at Big Hill. (Note that the output listing is intended to be printed on 15-inch wide paper.)

In this example an initial file will be developed by EDITOR, so 1 is selected from the main menu. The program next displays "SPR Schedule Editor Program Version Four" and "Enter input file name or press return if creating new file" (see Page D-15). Here the return is pressed (the default) because a new file is to be created. If an existing EDITOR file

is to be changed, then that file must be made available in the local file space before SPRSKD is activated. That file name would then be entered in response to EDITOR's first interrogative. For our example here, the name of the new file to be created by EDITOR is SAMPLE. Thus, SAMPLE is entered as the output file name for the next interrogative.

The next question, "ALTER CAVERN VALUES?" must be answered yes to activate the change routine in EDITOR. This change routine includes prompts that ask for the required data entry (page D-16). Identification of caverns is limited to the numbers 101 through 120. This series was chosen to be compatible with other SPR sites. As noted at the top of page D-16, the number 1 was entered intentionally to invoke the error message. After entering the proper number, 101, the user is asked if the cavern is to be deleted ("DELETE CAVERN # 101?"). This seems inappropriate at a time when creating is the only consideration. However, as time moves forward and this cavern is completed, then it will be necessary to delete it. At this time no is entered. The delete may be invoked with a yes.

At the next prompt, "ENTER? OR OPT# OR ALL: (DEFAULT = ?, FOR HELP -----0 = EXIT)," the user is expected to enter the necessary data into the file SAMPLE in seven steps. At this point the default will display the data entry fields as illustrated next on page D-16. Data may be entered by entering the OPT# and then the corresponding data in seven steps or the user may enter ALL and then be prompted for the seven data fields one at a time. The seven data field OPT#'s plus the three other OPT#'s are explained below.

This is also the point in the EDITOR program at which changes to an existing data file may be made. The user may call the necessary OPT#'s and enter changes as required. To retain existing data in any OPT#, enter the default (press just the enter key).

"0 ---- Exit" - To exit from this subroutine.

"1 ---- Mode + FOV" - Modes are: 0 = leach/fill, L/F; 1 = leach-then-fill, LTF; 2 = intermittent leach then fill, I/F. FOV = final oil volume.

"2 ---- END VOLS" - The total cavern volume at the end of each development stage - sump, sump/chimney, 1st reverse (REV1), 2nd reverse (REV2-1, REV2-2) and 3rd reverse (REV3-1, REV3-2).

"3 ---- Current Vols" - Current cavern volume and current oil volume. When starting, of course, these are zero.

"4 ---- Workovers" - The amount of downtime (in DAYS) to do a workover at the end of each development stage.

"5 ---- BPR Schedule" - This provides for the brine production rates during development and oil fill. Up to nine entries may be made.

"6 ---- I/F Oil Vols" - This is used to define the intermittent oil volumes when that mode is selected for data field 1 above.

"7 ---- Oil Fill Rates" - For final oil fill or I/F.

"All ---- 1 through 7" - When this option is selected, the program automatically sequences through the above seven steps displaying current values in the data fields (which are default values in this example with a newly created file) and then displaying prompts for input of new values. This is the proper response to the next prompt for inserting original data into all fields. For changing data in any specific data field, the number of that field would be the proper response to the next prompt.

"? ---- Opt Help" - This will redisplay the data fields menu. This is also the default.

With the request "MODE??" data entry begins with the leaching mode. The default mode is L/F, but for the sample here, the leach-then-fill mode (LTF) is chosen by entering a 1. "FOV??" - the final oil volume, is the next input request. It has a 10,000.00 MB default. For our sample

here, the FOV is changed to 11500 MB. (Note the warning which is issued for values above 10,000 MB on page D-16.)

A default workover schedule is displayed next and the user is asked if the stage endvolumes are to be recomputed. Since the endvolumes for the sample problem have not been entered, the response to this question is yes. The default endvolumes are displayed and redisplayed at the top of page D-17. Following these displays, the individual endvolumes are sequentially displayed along with prompts to allow changing as desired. The new endvolumes are entered one at a time as illustrated on page D-17. Since the caverns in the sample are developed with only two reverse stages, the inputs for numbers 6 and 7 are the same as input number 5. The endvolumes selected for this sample problem are as follows:

#1 ?	470 MB	This is the total cavern volume at the end of sump development and replaces the .00 MB as the default sump volume.
#2 ?	2770 MB	Total volume at the end of sump/chimney. (Replaces the 3574.20-MB default.)
#3 ?	6094 MB	Total volume at the end of first reverse (REV1). (Replaces the 5892.60-MB default.)
#4 ?	9578 MB	A number between those used in 3 (REV1) and 5 (REV2-2) to satisfy the program. There used to be a workover at this point so the program requires data here at REV2-1. (Replaces the 8191.45-MB default.)
#5 ?	12800 MB	Total volume at the end of 2nd reverse (REV2-2). (Replaces the 10215.45-MB default.)
#6 ?	12800 MB	Since the cavern is fully developed at the end of Rev2-2, there is no further leaching. Steps 6 and 7 are left in the program in case a three-reverse system needs to be investigated. (Replaces the 12880.00-MB default.)
#7 ?	12800 MB	Same comment as for 6.

"CCV ??" - the current cavern volume, and "COV ??" - the current oil volume, are the next requests. Since the user is starting a new cavern, both the current cavern volume (CCV) and the current oil volume (COV) are .00 MB. (In each case the default was used.) This option also permits restarting a cavern for a current volume other than zero.

The previously displayed default workover schedule is displayed again. For the sample case, a different workover series is input. Therefore, seven entries are made in response to the following prompts:

"#1 ?"	10 Days.	At the end of sump. (Replaces the 0 day default.)
"#2 ?"	10 Days.	After sump/chimney. (Replaces the 20 days default.)
"#3 ?"	15 Days.	After 1st reverse (REV1). (Replaces the 20 days default.)
"#4 ?"	0 Days.	After REV2-1 because no workover is planned. (REV2-1 is used in the leach/fill mode.)
"#5 ?"	15 Days.	After completion of second reverse (REV2-2). This is the end of cavern development. (Replaces the 20 days default.)
"#6 ?"	0 Days.	After REV3-1.
"#7 ?"	0 Days.	After REV3-2. (Replaces the 20 days default.)

Next, the brine production rate schedule (BPRS) in thousands of bbl/day is requested as on page D-18 ("BPRS="). The line "BPRS=" through the line "#1?" are next displayed. The default value of brine production rate is 150 MBD. Since the average design flow rate is 175 MBD for Big Hill caverns, that rate will be used. If the default brine production rate is not to be used, as in this example, then the desired rate along with its number of days is sequentially entered. When values for one time interval are entered, a prompt is displayed for the next interval until the entry with the *, designating cavern completion, has been made. Note that times entered for each interval are total days from beginning of sump development. The brine production rate schedule, displayed on page D-18 and following, reflects that the sump will be

developed at 50 MBD for 95 days. At the end of 95 days the cavern volume is calculated to be 421 MB. At that time the second well in the cavern will be opened (since coalescence will have occurred), and the flow will be increased to 85 MBD. The 85 MBD in the two-well configuration can be accomplished at the same raw water manifold pressure as 50 MBD in one well. The intent of the 85 MBD for varying lengths of time will permit using the workover rigs continuously without taking caverns off-line. This will result in different-sized sumps but this is corrected in sump/chimney development by going to similar total volumes in all caverns. The BPRS is entered as follows:

50,95	--	This means 50 MBD for 95 days.
85,103	--	85 MBD from day 95 through day 103.
175,392	--	175 MBD through day 392.
0,480	--	0 MBD through day 480 to illustrate that zero flow may be programmed into the schedule at any time.
175,*	--	175 MBD for the remainder of the time.

The program will discontinue calculations for each stage of cavern development when the previously defined endvolume for that stage is reached regardless of whether the days required corresponds to that defined above.

The next prompt, page D-18, is for the final oil fill rate, for which the default value is 50 MBD. Again, since Big Hill design was for 175 MBD oil fill rate, that is used in the sample. At this point, data have been entered in the seven data fields of "Editor" for cavern 101. At the next prompt, 0 is entered to exit the menu (pages D-2, D-3, and D-16) and return to the prompt for the ID of the next cavern. The next cavern is entered as 102. When the "DELETE CAVERN #102??" prompt is answered NO, then all the data entered in Cavern 101 are now duplicated in Cavern 102. It is easier to go into the 102 data file and change endvolumes and the brine production rate schedule than to enter all data. Again, referring back to the information required in the seven data fields (see page D-16), one may find the numbers of the data fields

in which changes are to be made for Cavern 102. We want to change the sump, sump/chimney and REV1 endvolumes, so we enter a 2 and the program displays the endvolumes stored for Cavern 102, page D-19. (These are the same as are stored for Cavern 101.) After the endvolumes are displayed, they may be changed one at a time by entering new numbers as shown on page D-19. Note that the first three endvolumes were changed by inputs and that the last four were left unchanged by using the default. After the last endvolume is changed the prompt again asks in which of the seven data fields are changes to be made? To change the brine production rate schedule (BPRS), 5 is entered. The program responds with the data stored in the BPRS file. This is the same information copied from the data input for Cavern 101. Again, the operator is given a chance to change each entry line by line. The sump development at 50 MBD for 95 days repeats for Cavern 102 so the default is used. For step two, the 85 MBD from day 95 through 138 is entered and replaces the BPRS used for Cavern 101. This is entered by 85,138. The cavern is to be completed at a BPRS of 175 MBD. To indicate that this is the last entry in the BPRS field, the * is used. The 175* was entered without the comma so that the error message would be displayed. Finally, the 175,* was entered which EDITOR understood.

The data option request line is displayed again asking if more changes to Cavern 102 are desired. Since all corrections have been made, a 0 is entered and the next cavern to be added is requested. If another cavern were entered, it would have the same data in its file as Cavern 102. Since two caverns are sufficient for this sample, the default is entered and a request for a "File Postview?" is displayed. This request is answered by entering YES. (See page D-20.) The request to view all caverns is entered and the response indicates that 2 of 20 caverns are to be developed by displaying all data for Caverns 101 and 102, and by indicating 18 caverns are inactive. (See pages D-20 and D-21.) The user is next asked, "Done with EDITOR?" Answer YES. This completes the session with the EDITOR subprogram in SPRSKD, and the user is notified that the file containing all the required data is called SAMPLE (see page D-21).

The main menu is displayed again and at this point PREDICT is selected by entering 2. The prompt is for the input file name from the EDITOR subprogram, so the entry is SAMPLE as illustrated on page D-22. The next prompt is for a "File Name For PREDICT Output." By inputting a file name here the output is saved in a file by that name, thus saving the time of having it printed on the screen. The file can be printed later when it is necessary to exit from SPRSKD. The PREDICT output file has been named PREOUT for this illustration. The next prompt is to assign a "File Name For AUTSKED Input." This is entered as AUINPUT. PREDICT next queries, "Do You Wish To Evaluate All Active Caverns?" The note displayed at this point requires that the question be answered yes if AUTSKED is to be run. If all caverns are not evaluated by PREDICT, then the input to AUTSKED will include only those evaluated by the running of PREDICT during this session. Here the default is set to provide the yes input. File AUINPUT provides AUTSKED with individual cavern schedules. The breakpoint prompt must be answered yes also so that AUTSKED can operate on all caverns simultaneously. Again the default is yes. The next two prompts are answered yes so that the output can be examined in this Appendix. (Normally these two questions would be answered with the default which is no.) All PREDICT output is shown on pages D-26 and D-27 following completion of the interactive session with SPRSKD. The PREDICT subprogram runs very quickly but displays no output if an output file name has been previously assigned.

After PREDICT completes its calculations, the main menu is displayed again (see page D-22) and 3 is selected this time to run AUTSKED. The first prompt asks for the "Name Of The AUTSKED Input File From PREDICT." This file name is entered as AUINPUT. The prompt then asks for the "Name Of The AUTSKED Output File." This file name is entered as AUOUT, and stands for the final schedule or product we desired when we started this procedure at step 1 in the main menu. The AUOUT file is illustrated at the end of this Appendix.

AUTSKED prompts for the following:

1. "Maximum Brine Flow For Site." 1400 MBD is used here since it is the design value for Big Hill.
2. "Maximum Oil Flow For Site." 220 MBD is used because it is a Big Hill design value also.
3. "Maximum Brine Flow (Default) For Caverns." (There is a warning that it must exceed 40 MBD to produce valid predictions.) This will permit caverns to receive up to the default amount as a maximum in the event that the maximum brine flow for the site falls below the value specified in 1 above. Another Big Hill design value is entered here 175 MBD.
4. "Use Default Brine Flow For All Caverns?" The default of yes here means that the maximum flow to all caverns is that listed in item 3. If no is entered, then the user has an opportunity to specify the maximum value for each cavern later.
5. "Maximum Oil Flow (Default) For Caverns." This input affects oil limits the same way item 3 affects brine limits. 175 MBD is the design maximum oil flow per cavern.
6. "Use Default Oil Flow For All Caverns?" Similar to item 4.
7. "Enter Number Of Workover Rigs." Enter the number of workover rigs that are available during site development. The example here assumes two rigs will be available. Therefore, 2 is entered.
8. "Priority Choice."

   a. "Shortest Time Remaining" - this is an automatic priority setting that AUTSKED assigns during its computations.

   b. "User Input Priority" - this permits the user to assign priorities. In the sample case, spacing between wells at the bottom of the hole was used.

   c. "Cavern Number Minus 100" - this will automatically finish Cavern 101 first, 102 second, etc. For our sample case, priority 2 was selected.

9. "Do You Want A Clock For Each Cavern?" This option permits the user to view the difference between the individual cavern development time calculated in PREDICT and the time required to complete each cavern in AUTSKED. This difference is more apparent as more caverns are added. In our sample the clock option was selected, but with only two caverns the effect of the clock is not very apparent.
10. "Do You Want A Cavern/Oil Volume Tabulation At Each Breakpoint?" This option lists the total cavern and oil volumes for each cavern at each time breakpoint and also the totals for the entire site.
11. "Do You Want Zero Flow Rates Maintained (Y) Or (N) Dependent On Max Site Flow Utilization?" In our sample case we use some zero flow time in EDITOR and PREDICT. This is recognized in PREDICT as "idle" time. However, in AUTSKED we ask the subprogram to depend upon maximum site flow utilization. As a result, Cavern 101 is completed in 718 days instead of the 806 days that PREDICT calculates. The 88 days of zero flow or idle time that was input into EDITOR (from day 392 to 480) is overridden by AUTSKED. If YES is selected in the sample problem, then the site demand for maximum brine flow does not override the request for idle time, and Cavern 101 is developed in 806 days.
12. "Enter Project Start Date" (see page D-23). We use 07/01/87 for our sample problem.
13. "Scheduling Options." Option 1 will permit AUTSKED to run through completion of the site. Option 2 permits the user to stop AUTSKED at any future date and to reassign priorities. We select 2.
14. If option 2 is selected, then the user must input the "Number Of Days" on which priority changes will be made. In our sample, 150 days is selected. For subsequent changes, the total number of days from the start of leaching must be used.
15. "Enter Cavern Priorities." 1=highest, 2=next, etc. No input is required here.

16. "Enter Priority For Cavern 101." We enter 1 here.
17. "Enter Priority For Cavern 102." We enter 2.
18. "Would You Like An Echo Of Your Input And The Option To Make Changes?" Yes was selected. The echo is illustrated on page D-24, and shows the inputs as given by the user. Since the echo is more than a screen full, it is divided into three sections and requires pressing the RETURN key to continue.
19. Make a choice from the menu (see page D-24). Since everything is as we had requested, selection 16 is chosen and the program begins to execute.
20, 21. "Enter Cavern Priorities." The program stops after 150 days to permit the user to change the priorities assigned to each cavern. The priority for Cavern 101 is changed to 2, and for Cavern 102, to 1. See page D-25.
22. Scheduling option is again displayed and option 2 is the intended choice. However, the user anticipates the next question and enters 375 by mistake. The program responds with a "Bad Input" error message and the user enters 2.
23. "Enter Number Of Days." This is the total number of days from day 1, not the last priority change day. The user enters 375. This sequence (Steps 20 through 22) is repeated one more time and the user selects development completion by entering a 1 at the last opportunity. See page D-25.

The program completes its calculation, and the main menu is displayed once again. The exit choice, 4, is selected and the schedule created by AUTSKED is written to the file AUOUT.

To review the data that was selected as input for this sample exercise, see the EDITOR file on page D-20. This lists all of the data for Caverns 101 and 102 and states that the other 18 caverns are inactive. Next, print out the file PREOUT which contains the data in PREDICT. This file is printed out starting on page D-26 and continues to the page D-27. The Caverns 101 and 102 breakpoints provide all the data needed by AUTSKED. The site breakpoints on page D-27 of the Attachment

are shown as examples of data available in PREDICT, and indicate how long it would take to develop the site provided there are no changes in schedule, i.e., zero flow time is not overridden by AUTSKED. They also list total cavern and oil volumes, average daily oil flow and brine production rates. The final listing, Site Summary on page D-27 indicates total cavern and oil volumes, oil use and average flow during each 90 day interval. These data are listed at 90-day intervals because that is what the user requested when inputs were given to PREDICT.

Finally, the output from the AUTSKED subprogram is printed from the AUTSKED output file AUOUT. This output follows the PREDICT output (starts on page D-28). The file starts with a review of some of the input data, and a legend to explain the letters and numbers used in the listing.

The first listing is in pairs of lines because we asked for the cavern clock option. The first line contains the date, the number of days from day 0, the brine production rate for Cavern 101 and its development stage and finally the brine production rate for Cavern 102 and its development stage. The cavern clock on line two lists the development day from PREDICT that corresponds to where that cavern is in its development. If a cavern is delayed in AUTSKED due to constraints, the day number on line one will continue with the real clock, but the cavern clock will stop. The cavern clock will resume when cavern development resumes, and the difference between the day number on line one and the cavern clock will indicate how many days the cavern was off-line or not being developed. If it is necessary to go back to EDITOR and change zero flow times, water flow rates or oil flows at some day into the schedule, then cavern clock time must be used when figuring where to enter the change in EDITOR.

The total time for site development is shown to be 772 days on page D-30. This is 34 days less than PREDICT calculated per pages D-26 and D-27. The difference between the cavern clock and the number of days

from day zero (start of site development) is due to the fact that the AUTSKED subprogram overrode the idle time that was entered into EDITOR and PREDICT for Cavern 101. There is another consideration that affects this difference, and that is the oil fill rate of Cavern 102. PREDICT indicates a completion time of 742 days for Cavern 102. This is based upon a final oil fill rate of 175 MBD for 54 days. However, since the entire site can flow only 220 MBD of oil at any time and since Cavern 101, which is being filled at a rate of 175 MBD, has higher priority than Cavern 102, only 45 MBD of oil are available to fill Cavern 102 until Cavern 101 completes filling. This results in the 30-day difference between the cavern clock and the time from PREDICT.

The next information on page D-30 includes statistics for variables based on observation. These data are calculated from the cavern completion times calculated in PREDICT. (718 days for cavern 101 and 772 days for cavern 102.) These values are self explanatory.

The statistics for time-persistent variables pertain to the intermittent oil fill mode. Since our sample illustration is for the leach-then-fill mode, these statistics do not apply. In the intermittent oil fill mode, these statistics would apply to the first oil fill period.

The remaining information on page D-30 defines the various files used in the program. Note that there is no file 2 and that F2 is used as final fill for the leach-then-fill mode. Therefore, File 3 does not apply here. (It is used in the intermittent oil fill mode.)

On pages D-31 and D-32 printouts for each of the five files listed on page D-30 are displayed. TNOW is the time that the table is printed. QQTIM is the last time the file was accessed. These printouts are all for data contained in the files at the end of the run where the time is 772 days. These files may be used as diagnostic during the run but at the end of the run they are all empty and of no value. The Resource Statistics listed on page D-32 provide the statistical data defined for each of the four resources numbered and labeled.

The next listing titled table number 1, pages D-32, D-33, and D-34, defines the total site brine flow rate (B), the total site oil flow rate (O), the total site cavern volume (C), and the total site oil volume (V). These are instantaneous values at ten-day intervals. These data are plotted on page D-35. Note that the duplicates in the right column mean the pairs of letters listed have the same values on the plot. One of the letters is printed on the plot but the other is not, simply to avoid overstriking the two letters.

The final listing on page D-36 includes total cavern and oil volumes for Cavern 101 in columns one and two; the same for Cavern 102 in columns 3 and 4; and site totals in columns 5 and 6.

This completes the sample interactive session using SPRSKD.

```
/begin,,sprsked

                    ********** MENU ************

    1.  EDIT OLD SITE FILE OR CREATE NEW SITE FILE USING EDITOR

    2.  DEVELOP INDIVIDUAL CAVERN SCHEDULES USING PREDICT

    3.  DEVELOP SITE SCHEDULE USING AUTSKED

    4.  EXIT FROM PROGRAM

 ENTER NUMBER OF CHOICE
? 1


                    SPR SCHEDULE EDITOR PROGRAM
                          VERSION FOUR

 ENTER INPUT FILE NAME OR PRESS RETURN
 IF CREATING NEW FILE

? NEW SITE FILE WILL BE CREATED
 ENTER OUTPUT FILE NAME

? Sample

 NEW FILE SAMPLE WILL BE CREATED

   ALTER CAVERN VALUES??(Y/N, N=DEFAULT)
? Y
  YES
```

```
 CAVERN ID# ??(DEFAULT=NO MORE)
?1
 RANGE ERROR:  MUST BE [ 101 TO 120 ]

 CAVERN ID# ??(DEFAULT=NO MORE)
?101
 ID# = 101

 DELETE CAVERN# 101 ??(Y/N, N=DEFAULT)
?n
 NO

 ENTER ? OR OPT# OR ALL:(DEFAULT=?, FOR HELP ---- O=EXIT)
?


    0   ----- EXIT
    1   ----- MODE + FOV
    2   ----- END VOLS
    3   ----- CURRENT VOLS
    4   ----- WORKOVERS
    5   ----- BPR SCHEDULE
    6   ----- I/F OIL VOLS
    7   ----- OIL FILL RATES
    ALL ----- 1 THRU 7
    ?   ----- OPT HELP

 ENTER ? OR OPT# OR ALL:(DEFAULT=?, FOR HELP ---- O=EXIT)
?all
 ALL

 THE LEACH MODE IS L/F

 MODE ??  <0=L/F, 1=LTF, 2=I/F>(DEFAULT=NO CHANGE)
?1
 THE MODE IS LTF

 FOV = 10000.00 MB

 FOV??(DEFAULT=NO CHANGE)
?11500
 11500.00 MB
 WARNING:  MUST BE < OR = TO FINAL CAVERN VOLUME

   THE NEW WORKOVERS ARE:
        SUMP   SUMP-CH   REV1   REV2-1   REV2-2   REV3-1   REV3-2
           0       20     20         0       20        0       20  DAYS

 DO YOU WANT THE STAGE ENDVOLUMES RECOMPUTED??
                  (Y/N, N=DEFAULT)
?y
 YES
```

```
   THE NEW ENDVOLUMES ARE:
        SUMP   SUMP-CH    REV1   REV2-1   REV2-2   REV3-1   REV3-2
          .00  3574.20  5892.60  8191.45 10215.45 12880.00 12880.00 MB

 ENDVOLUMES =
        SUMP   SUMP-CH    REV1   REV2-1   REV2-2   REV3-1   REV3-2
          .00  3574.20  5892.60  8191.45 10215.45 12880.00 12880.00 MB

         PRESS RETURN FOR NO CHANGE
 #1 ?
? 470
    470.00 MB
 #2 ?
? 2770
   2770.00 MB
 #3 ?
? 6094
   6094.00 MB
 #4 ?
? 9578
   9578.00 MB
 #5 ?
? 12800
  12800.00 MB
 #6 ?
? 12800
  12800.00 MB
 #7 ?
? 12800
  12800.00 MB

  CCV =       .00 MB   COV =        .00 MB

   CCV??(DEFAULT=NO CHANGE)
?
       .00 MB

   COV??(DEFAULT=NO CHANGE)
?
       .00 MB

 WORKOVERS =
        SUMP   SUMP-CH    REV1   REV2-1   REV2-2   REV3-1   REV3-2
           0       20      20        0       20        0       20  DAYS

         PRESS RETURN FOR NO CHANGE
 #1 ?
? 10
     10 DAYS
 #2 ?
? 10
     10 DAYS
 #3 ?
? 15
     15 DAYS
```

```
 #4 ?
? 0
      0 DAYS
 #5 ?
15
?    15 DAYS
 #6 ?
? 0
      0 DAYS
 #7 ?
0
?       0 DAYS

 BPRS =
   150.00,  *

    PRESS RETURN FOR NO CHANGE

 # 1 ?
? 50,95
     50.00,   95
 # 2 ?
? 85,103
     85.00,  103
 # 3 ?
? 175,392
    175.00,  392
 # 4 ?
? 0,480
        .00,  480
 # 5 ?
? 175,*
    175.00,  *

   FINAL OIL FILL RATE =  50.00 MB

   FILL RATE??(DEFAULT=NO CHANGE)
? 175
    175.00 MB

 ENTER ? OR OPT# OR ALL:(DEFAULT=?, FOR HELP ---- O=EXIT)
?0
 0

 CAVERN ID# ??(DEFAULT=NO MORE)
?102
 ID# = 102

 DELETE CAVERN# 102 ??(Y/N, N=DEFAULT)
?
 NO

 ENTER ? OR OPT# OR ALL:(DEFAULT=?, FOR HELP ---- O=EXIT)
?2
 2
```

```
 ENDVOLUMES =
        SUMP    SUMP-CH     REV1    REV2-1    REV2-2    REV3-1    REV3-2
      470.00   2770.00   6094.00   9578.00 12800.00 12800.00 12800.00 MB

         PRESS RETURN FOR NO CHANGE
 #1 ?
? 679
    679.00 MB
 #2 ?
? 2710
   2710.00 MB
 #3 ?
? 6092
   6092.00 MB
 #4 ?
?
   9578.00 MB
 #5 ?
?
  12800.00 MB
 #6 ?
?
  12800.00 MB
 #7 ?
?
  12800.00 MB

  ENTER ? OR OPT# OR ALL:(DEFAULT=?, FOR HELP ---- O=EXIT)
? 5
  5

  BPRS =
     50.00,   95
     85.00,  103
    175.00,  392
       .00,  480
    175.00,  *

     PRESS RETURN FOR NO CHANGE

 # 1 ?
?
     50.00,   95
 # 2 ?
? 85,138
     85.00,  138
 # 3 ?
? 175*
  INPUT ERROR:  INCORRECT REAL FORMAT OR CHARACTER
 # 3 ?
? 175,*
    175.00,  *
```

```
  ENTER ? OR OPT# OR ALL:(DEFAULT=?, FOR HELP ---- O=EXIT)
? 0
  0

  CAVERN ID# ??(DEFAULT=NO MORE)
?
  FILE POSTVIEW??(Y/N, N=DEFAULT)
?y
 YES

  ENTER CAVERN#S OR ALL:
? all

 101 IS ACTIVE
     MODE IS LTF

      CCV =       .00 MB

      COV =       .00 MB

      FOV = 11500.00 MB
     BPR# 1 =    50.00,      95 (MB/DAY,LASTDAY#)
     BPR# 2 =    85.00,     103 (MB/DAY,LASTDAY#)
     BPR# 3 =   175.00,     392 (MB/DAY,LASTDAY#)
     BPR# 4 =      .00,     480 (MB/DAY,LASTDAY#)
     BPR# 5 =   175.00,     *   (MB/DAY,LASTDAY#)

     OIL FILL RATE =  175.00 MB

               SUMP    SUMP-CH    REV1    REV2-1    REV2-2    REV3-1    REV3-2
      EVS =    470.00  2770.00  6094.00 9578.00  12800.00 12800.00 12800.00 MB
     DELAYS =   10       10       15       0        15        0         0   DAYS

 102 IS ACTIVE
     MODE IS LTF

      CCV =       .00 MB

      COV =       .00 MB

      FOV = 11500.00 MB
     BPR# 1 =    50.00,      95 (MB/DAY,LASTDAY#)
     BPR# 2 =    85.00,     138 (MB/DAY,LASTDAY#)
     BPR# 3 =   175.00,     *   (MB/DAY,LASTDAY#)

     OIL FILL RATE = 175.00 MB

               SUMP    SUMP-CH    REV1    REV2-1    REV2-2    REV3-1    REV3-2
      EVS =    679.00  2710.00  6092.00 9578.00  12800.00 12800.00 12800.00 MB
     DELAYS =   10       10       15       0        15        0         0   DAYS

 103 IS INACTIVE

 104 IS INACTIVE
```

```
 105 IS INACTIVE

 106 IS INACTIVE

 107 IS INACTIVE

 108 IS INACTIVE

 109 IS INACTIVE

 110 IS INACTIVE

 111 IS INACTIVE

 112 IS INACTIVE

 113 IS INACTIVE

 114 IS INACTIVE

 115 IS INACTIVE

 116 IS INACTIVE

 117 IS INACTIVE

 118 IS INACTIVE

 119 IS INACTIVE

 120 IS INACTIVE

  DONE WITH EDITOR??(Y/N)

? Y

**** EDITING COMPLETED ****
          SITE FILE WITH CHANGES IS SAMPLE


                              ********** MENU ************

    1.  EDIT OLD SITE FILE OR CREATE NEW SITE FILE USING EDITOR

    2.  DEVELOP INDIVIDUAL CAVERN SCHEDULES USING PREDICT

    3.  DEVELOP SITE SCHEDULES USING AUTSKED

    4.  EXIT FROM PROGRAM

 ENTER NUMBER OF CHOICE
? 2
```

```
 ENTER INPUT FILE NAME FROM EDITOR
? sample
 ENTER FILE NAME FOR PREDICT OUTPUT
          (DEFAULT =OUTPUT, I.E. THE TERMINAL)
? preout
 ENTER FILE NAME FOR AUTSKED INPUT
? auinput

DO YOU WISH TO EVALUATE ALL ACTIVE CAVERNS?
          (Y OR N, DEFAULT = Y)
  NOTE:  ANSWER Y IF YOU ARE GOING TO RUN AUTSKED
?
 DO YOU WANT BREAKPOINT DATA FOR EACH CAVERN?
          (Y OR N, DEFAULT = Y)
  NOTE:  ANSWER Y IF YOU ARE GOING TO RUN AUTSKED
?
 DO YOU WANT THE SITE BREAKPOINT SUMMARY?
          (Y OR N, DEFAULT = N)
? y
 DO YOU WANT A SITE INCREMENTAL SUMMARY?
          (Y OR N, DEFAULT = N)
? y
 ENTER REPORT INCREMENT IN DAYS [D:90]
?

**** PREDICT CAVERN SCHEDULES COMPLETED ****
          FILE FOR INPUT TO AUTSKED IS AUINPUT
           PREDICT CAVERN SCHEDULES IN FILE PREOUT


                          ********** MENU ************

    1.  EDIT OLD SITE FILE OR CREATE NEW SITE FILE USING EDITOR

    2.  DEVELOP INDIVIDUAL CAVERN SCHEDULES USING PREDICT

    3.  DEVELOP SITE SCHEDULE USING AUTSKED

    4.  EXIT FROM PROGRAM

 ENTER NUMBER OF CHOICE
? 3

 ENTER NAME OF AUTSKED INPUT FILE FROM PREDICT
? auinput
 ENTER NAME OF AUTSKED OUTPUT FILE
? auout
```

```
 ENTER MAX BRINE FLOW FOR SITE(MB/DAY)
? 1400
 ENTER MAX OIL FLOW FOR SITE(MB/DAY)
? 220
 ENTER MAX BRINE FLOW (DEFAULT) FOR CAVERNS(MB/DAY)
     (MUST BE AT LEAST 40)
? 175
 USE DEFAULT BRINE FLOW FOR ALL CAVERNS?(Y/N,DEFAULT=Y)
?
 ENTER MAX OIL FLOW (DEFAULT) FOR CAVERNS(MB/DAY)
? 175
 USE DEFAULT OIL FLOW FOR ALL CAVERNS?(Y/N,DEFAULT=Y)
?
 ENTER NUMBER OF WORKOVER RIGS
? 2
 PRIORITY CHOICE
          1.  SHORTEST TIME REMAINING
          2.  USER INPUT PRIORITY
          3.  CAVERN NUMBER MINUS 100

 ENTER NUMBER OF CHOICE
? 2
 DO YOU WANT A CLOCK FOR EACH CAVERN?(Y/N, DEFAULT=N)
? y
 DO YOU WANT A CAVERN/OIL VOLUME TABULATION AT EACH
 BREAKPOINT?(Y/N, DEFAULT=N)
? y
 DO YOU WANT ZERO FLOW RATES MAINTAINED (Y) OR
 (N) DEPENDENT ON MAX SITE FLOW UTILIZATION ?(Y/N, DEFAULT=Y)
? n
 ENTER START DATE FOR PROJECT
 IN THE FORMAT  MM/DD/YY
 EXAMPLE:  04/08/83
? 07/01/87
 SCHEDULING OPTIONS
      1.  DEVELOP SCHEDULE TO SITE COMPLETION
      2.  STOP SCHEDULE AFTER A NUMBER OF DAYS
            AND REPRIORITIZE THE CAVERNS

 ENTER NUMBER OF CHOICE
? 2
 ENTER THE NUMBER OF DAYS
? 150
 ENTER CAVERN PRIORITIES--
 1 HIGHEST, 2 NEXT, ETC.

 ENTER PRIORITY FOR CAVERN 101
? 1
 ENTER PRIORITY FOR CAVERN 102
? 2
     WOULD YOU LIKE AN ECHO OF YOUR INPUT
     AND THE OPTION TO MAKE CHANGES? (Y/N, DEFAULT=N)
? y
```

```
                ** ECHO OF INPUT **

     MAXIMUM BRINE FLOW FOR SITE              1400 MB/DAY
     MAXIMUM OIL FLOW FOR SITE                 220 MB/DAY
     MAX BRINE FLOW(DEFAULT) FOR CAVERNS       175 MB/DAY
     MAX OIL FLOW(DEFAULT) FOR CAVERNS         175 MB/DAY
     NUMBER OF WORKOVER RIGS                     2
     PROJECT START DATE                       07/01/87
     CAVERN CLOCK OPTION                        YES
     CAVERN/OIL VOLUME TABULATION               YES
     FORCED ZERO FLOW OPTION                    NO

  PAUSE PRESS RETURN TO CONTINUE
?
                    MAX FLOW RATES AND INITIAL PRIORITIES

    CAVERN NUMBER     PRIORITY      MAX BRINE FLOW     MAX OIL FLOW
    101                1                175                175
    102                2                175                175

  PAUSE PRESS RETURN TO CONTINUE
?

    MAKE A CHOICE FROM THE MENU

                   *** MENU ***

          1.  CHANGE MAX SITE BRINE FLOW
          2.  CHANGE MAX SITE OIL FLOW
          3.  CHANGE MAX BRINE FLOW(DEFAULT) FOR CAVERNS
          4.  CHANGE MAX OIL FLOW(DEFAULT) FOR CAVERNS
          5.  CHANGE MAX BRINE FLOW FOR A CAVERN
          6.  CHANGE MAX OIL FLOW FOR A CAVERN
          7.  CHANGE NUMBER OF WORKOVER RIGS
          8.  CHANGE CAVERN CLOCK OPTION
          9.  CHANGE CAVERN/OIL VOLUME TABULATION
         10.  CHANGE FORCED ZERO FLOW OPTION
         11.  CHANGE PROJECT START DATE
         12.  CHANGE PRIORITY TYPE(STR OR USER)
         13.  CHANGE CAVERN USER PRIORITY
         14.  DISPLAY CURRENT CAVERN PRIORITY TABLE
         15.  DO INPUT ECHO AGAIN
         16.  EXIT TO PROGRAM EXECUTION
         17.  TERMINATE THIS RUN

? 16

SCHEDULING PROGRAM IS EXECUTING

 SCHEDULING STOPPED AT DAY 150 IN ORDER TO
CHANGE CAVERN PRIORITIES.

 ENTER CAVERN PRIORITIES--
 1 HIGHEST, 2 NEXT, ETC.
```

```
 ENTER PRIORITY FOR CAVERN 101
? 2
 ENTER PRIORITY FOR CAVERN 102
1
?  SCHEDULING OPTIONS
      1.  DEVELOP SCHEDULE TO SITE COMPLETION
      2.  STOP SCHEDULE AFTER A NUMBER OF DAYS
            AND REPRIORITIZE THE CAVERNS

 ENTER NUMBER OF CHOICE
? 375
 BAD INPUT---TRY AGAIN
 SCHEDULING OPTIONS
      1.  DEVELOP SCHEDULE TO SITE COMPLETION
      2.  STOP SCHEDULE AFTER A NUMBER OF DAYS
            AND REPRIORITIZE THE CAVERNS

 ENTER NUMBER OF CHOICE
? 2
 ENTER THE NUMBER OF DAYS
? 375

 SCHEDULING CONTINUING

 SCHEDULING STOPPED AT DAY 375 IN ORDER TO
CHANGE CAVERN PRIORITIES.

 ENTER CAVERN PRIORITIES--
 1 HIGHEST, 2 NEXT, ETC.

 ENTER PRIORITY FOR CAVERN 101
? 1
 ENTER PRIORITY FOR CAVERN 102
? 2
 SCHEDULING OPTIONS
      1.  DEVELOP SCHEDULE TO SITE COMPLETION
      2.  STOP SCHEDULE AFTER A NUMBER OF DAYS
            AND REPRIORITIZE THE CAVERNS

 ENTER NUMBER OF CHOICE
? 1

 SCHEDULING CONTINUING

**** SITE SCHEDULE COMPLETED ****
          SITE SCHEDULE IN FILE AUOUT


                          ********** MENU ************

    1.  EDIT OLD SITE FILE OR CREATE NEW SITE FILE USING EDITOR

    2.  DEVELOP INDIVIDUAL CAVERN SCHEDULES USING PREDICT
```

```
    3.  DEVELOP SITE SCHEDULES USING AUTSKED

    4.  EXIT FROM PROGRAM

 ENTER NUMBER OF CHOICE
? 4
REVERT, SCHEDULE PROGRAM TERMINATED NORMALLY
/ra
  15 FILES PROCESSED.
/list,f=preout
```

CAVERN 101 BREAKPOINTS

DAY INTERVAL	LEACH STAGE	BRINE FLOW	OIL FLOW	CAVERN VOLUME	OIL VOLUME
0				0.	0.
1- 95	SUMP	50.0	.00	422.	0.
96- 103	SUMP	85.0	.00	470.	0.
104- 113	W/O				
114- 230	S/C	175.0	.00	2770.	0.
231- 240	W/O				
241- 370	R1	175.0	2.86	6094.	371.
371- 385	W/O				
386- 392	R2-1	175.0	.00	6280.	371.
393- 480	IDLE				
481- 605	R2-1	175.0	.00	9578.	371.
606- 727	R2-2	175.0	.00	12800.	371.
728- 742	W/O				
743- 806	FILL	175.0	175.00	12800.	11500.

CAVERN 102 BREAKPOINTS

DAY INTERVAL	LEACH STAGE	BRINE FLOW	OIL FLOW	CAVERN VOLUME	OIL VOLUME
0				0.	0.
1- 95	SUMP	50.0	.00	422.	0.
96- 138	SUMP	85.0	.00	679.	0.
139- 148	W/O				
149- 252	S/C	175.0	.00	2710.	0.
253- 262	W/O				
263- 394	R1	175.0	2.81	6092.	371.
395- 409	W/O				
410- 541	R2-1	175.0	.00	9578.	371.
542- 663	R2-2	175.0	.00	12800.	371.
664- 678	W/O				
679- 742	FILL	175.0	175.00	12800.	11500.

---SITE BREAKPOINTS---

DAY INTERVAL	TOTAL CAVERN VOLUME	TOTAL OIL VOLUME	DAILY OIL FLOW	BRINE PRODUCTION RATE
0	0.	0.		
1- 95	843.	0.	.00	100.0
96- 103	939.	0.	.00	170.0
104- 113	999.	0.	.00	85.0
114- 138	1640.	0.	.00	260.0
139- 148	1837.	0.	.00	175.0
149- 230	5050.	0.	.00	350.0
231- 240	5246.	0.	.00	175.0
241- 252	5787.	34.	2.86	350.0
253- 262	6043.	63.	2.86	175.0
263- 370	11571.	675.	5.67	350.0
371- 385	11955.	717.	2.81	175.0
386- 392	12321.	737.	2.81	350.0
393- 394	12372.	743.	2.81	175.0
395- 409	12372.	743.	.00	.0
410- 480	14247.	743.	.00	175.0
481- 541	17467.	743.	.00	350.0
542- 605	20846.	743.	.00	350.0
606- 663	23910.	743.	.00	350.0
664- 678	24306.	743.	.00	175.0
679- 727	25600.	9263.	175.00	350.0
728- 742	25600.	11871.	175.00	175.0
743- 806	25600.	23000.	175.00	175.0

---SITE SUMMARY---

DAY	TOTAL CAVERN VOLUMES	TOTAL OIL VOLUMES	OIL USED	AVERAGE OIL FLOW
0	0.	0.		
90	799.	0.	0.	.00
180	3091.	0.	0.	.00
270	6452.	108.	108.	1.20
360	11059.	618.	510.	5.67
450	13455.	743.	124.	1.38
540	17415.	743.	0.	.00
630	22167.	743.	0.	.00
720	25415.	8046.	7303.	81.15
806	25600.	23000.	14954.	173.89

/list.f=auout

**** S C H E D U L E ****

MAXIMUM BRINE FLOW FOR SITE	1400 MB/DAY
MAXIMUM OIL FLOW FOR SITE	220 MB/DAY
MAXIMUM BRINE FLOW FOR CAVERN	175 MB/DAY
MAXIMUM OIL FLOW FOR CAVERN	175 MB/DAY
NUMBER OF WORKOVER RIGS	2

L E G E N D

SU	SUMP
SC	SUMP/CHIMNEY
R1	REVERSE 1
R2	REVERSE 2
R3	REVERSE 3
F1	INTERMEDIATE FILL PHASE
F2	INTERMEDIATE FILL PHASE
F3	FINAL FILL PHASE
LF	LEACH/FILL OIL FLOW
W1	WORKOVER PHASE 1
W2	WORKOVER PHASE 2
W3	WORKOVER PHASE 3
F	SOME CAVERN IN FILL PHASE

		CAVERN NUMBER		
DATE	# DAYS	101	102	
07/01/87	0	50SU+	50SU+	+
CAVERN CLOCK		0+	0+	
10/04/87	95	85SU+	85SU+	+
CAVERN CLOCK		95+	95+	
10/12/87	103	W1 +	SU +	+
CAVERN CLOCK		103+	103+	
10/15/87	106	W2 +	SU +	+
CAVERN CLOCK		106+	106+	
10/19/87	110	W3 +	SU +	+
CAVERN CLOCK		110+	110+	
10/22/87	113	175SC+	SU +	+
CAVERN CLOCK		113+	113+	
11/16/87	138	SC +	W1 +	+
CAVERN CLOCK		138+	138+	
11/19/87	141	SC +	W2 +	+
CAVERN CLOCK		141+	141+	
11/23/87	145	SC +	W3 +	+
CAVERN CLOCK		145+	145+	
11/26/87	148	SC +	175SC+	+
CAVERN CLOCK		148+	148+	
11/28/87	150	SC +	SC +	+
CAVERN CLOCK		150+	150+	

02/16/88	230	W1 +	SC +		+
CAVERN CLOCK		230+	230+		
02/19/88	233	W2 +	SC +		+
CAVERN CLOCK		233+	233+		
02/23/88	237	W3 +	SC +		+
CAVERN CLOCK		237+	237+		
02/26/88	240	172R1+	SC +	F	+
		3LF+	+		+
CAVERN CLOCK		240+	240+		
03/09/88	252	R1 +	W1 +		+
CAVERN CLOCK		252+	252+		
03/12/88	255	R1 +	W2 +		+
CAVERN CLOCK		255+	255+		
03/16/88	259	R1 +	W3 +		+
CAVERN CLOCK		259+	259+		
03/19/88	262	R1 +172R1+		F	+
		+	3LF+		+
CAVERN CLOCK		262+	262+		
07/07/88	372	W1 +	R1 +		+
CAVERN CLOCK		372+	372+		
07/10/88	375	W1 +	R1 +		+
CAVERN CLOCK		375+	375+		
07/12/88	377	W2 +	R1 +		+
CAVERN CLOCK		377+	377+		
07/17/88	382	W3 +	R1 +		+
CAVERN CLOCK		382+	382+		
07/22/88	387	175R2+	R1 +		+
CAVERN CLOCK		387+	387+		
07/29/88	394	175R2+	R1 +		+
CAVERN CLOCK		394+	394+		
07/31/88	396	R2 +	W1 +		+
CAVERN CLOCK		396+	396+		
08/05/88	401	R2 +	W2 +		+
CAVERN CLOCK		401+	401+		
08/10/88	406	R2 +	W3 +		+
CAVERN CLOCK		406+	406+		
08/15/88	411	R2 +175R2+			+
CAVERN CLOCK		411+	411+		
10/25/88	482	175R2+	R2 +		+
CAVERN CLOCK		482+	482+		
11/29/88	517	175R2+	R2 +		+
CAVERN CLOCK		517+	517+		
12/25/88	543	R2 +175R2+			+
CAVERN CLOCK		543+	543+		
03/31/89	639	W1 +	R2 +		+
CAVERN CLOCK		639+	639+		
04/05/89	644	W2 +	R2 +		+
CAVERN CLOCK		644+	644+		
04/10/89	649	W3 +	R2 +		+
CAVERN CLOCK		649+	649+		
04/15/89	654	175F2+	R2 +	F	+
CAVERN CLOCK		654+	654+		
04/26/89	665	F2 +	W1 +	F	+
CAVERN CLOCK		665+	665+		

```
05/01/89     670     F2 +  W2 +   F +
  CAVERN CLOCK       670+  670+
05/06/89     675     F2 +  W3 +   F +
  CAVERN CLOCK       675+  675+
05/11/89     680     F2 + 45F2+   F +
  CAVERN CLOCK       680+  680+
06/18/89     718        +175F2+   F +
  CAVERN CLOCK       718+  718+
08/11/89     772        +     +     +
  CAVERN CLOCK       718+  772+
```

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
CAVERN DONE TIME	.7450E+03	.3818E+02	.5125E-01	.7180E+03	.7720E+03	2

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
IF FILL 1 ELSE 0	.3109E+00	.4629E+00	.0000E+00	.1000E+01	.7720E+03	.1000E+01

FILE 1 IS F1 OR F2 FILL QUEUE

FILE 3 IS F3 FILL QUEUE

FILE 4 IS LEACHING QUEUE

FILE 5 IS WORKOVER 1 OR 3 QUEUE

FILE 6 IS WORKOVER 2 QUEUE

```
      PRINTOUT OF FILE NUMBER   1
          TNOW =    .7720E+03
          QQTIM=    .7180E+03

TIME PERIOD FOR STATISTICS  .7720E+03
AVERAGE NUMBER IN FILE          .0000
STANDARD DEVIATION              .0000
MAXIMUM NUMBER IN FILE         2

          THE FILE IS EMPTY


      PRINTOUT OF FILE NUMBER   3
          TNOW =    .7720E+03
          QQTIM=    .0000E+00

TIME PERIOD FOR STATISTICS  .7720E+03
AVERAGE NUMBER IN FILE          .0000
STANDARD DEVIATION              .0000
MAXIMUM NUMBER IN FILE         0

          THE FILE IS EMPTY


      PRINTOUT OF FILE NUMBER   4
          TNOW =    .7720E+03
          QQTIM=    .6540E+03

TIME PERIOD FOR STATISTICS  .7720E+03
AVERAGE NUMBER IN FILE          .0000
STANDARD DEVIATION              .0000
MAXIMUM NUMBER IN FILE         2

          THE FILE IS EMPTY


      PRINTOUT OF FILE NUMBER   5
          TNOW =    .7720E+03
          QQTIM=    .6750E+03

TIME PERIOD FOR STATISTICS  .7720E+03
AVERAGE NUMBER IN FILE          .0000
STANDARD DEVIATION              .0000
MAXIMUM NUMBER IN FILE         1

          THE FILE IS EMPTY


      PRINTOUT OF FILE NUMBER   6
          TNOW =    .7720E+03
          QQTIM=    .6700E+03
```

TIME PERIOD FOR STATISTICS	.7720E+03
AVERAGE NUMBER IN FILE	.0000
STANDARD DEVIATION	.0000
MAXIMUM NUMBER IN FILE	1

THE FILE IS EMPTY

RESOURCE STATISTICS

RESOURCE NUMBER	RESOURCE LABEL	CURRENT CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	MAXIMUM UTILIZATION	CURRENT UTILIZATION
1	BRINE	******	271981.8653	98266.1722	350000	0

RESOURCE STATISTICS

RESOURCE NUMBER	RESOURCE LABEL	CURRENT CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	MAXIMUM UTILIZATION	CURRENT UTILIZATION
2	WO1 OR 3	3	.0829	.2757	1	0

RESOURCE STATISTICS

RESOURCE NUMBER	RESOURCE LABEL	CURRENT CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	MAXIMUM UTILIZATION	CURRENT UTILIZATION
3	WO2	2	.0466	.2108	1	0

RESOURCE STATISTICS

RESOURCE NUMBER	RESOURCE LABEL	CURRENT CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	MAXIMUM UTILIZATION	CURRENT UTILIZATION
4	OIL	220000	29924.6528	68303.5158	220000	0

1

TABLE NUMBER 1
RUN NUMBER 1

TIME	BRINE FLOW	OIL FLOW	CAVERN VOLUME	OIL VOLUME
.0000E+00	.1400E+04	.2200E+03	.0000E+00	.0000E+00
.1000E+02	.1000E+03	.0000E+00	.8874E+02	.0000E+00

.2000E+02	.1000E+03	.0000E+00	.1775E+03	.0000E+00
.3000E+02	.1000E+03	.0000E+00	.2662E+03	.0000E+00
.4000E+02	.1000E+03	.0000E+00	.3550E+03	.0000E+00
.5000E+02	.1000E+03	.0000E+00	.4437E+03	.0000E+00
.6000E+02	.1000E+03	.0000E+00	.5324E+03	.0000E+00
.7000E+02	.1000E+03	.0000E+00	.6212E+03	.0000E+00
.8000E+02	.1000E+03	.0000E+00	.7099E+03	.0000E+00
.9000E+02	.1000E+03	.0000E+00	.7986E+03	.0000E+00
.1000E+03	.1700E+03	.0000E+00	.9037E+03	.0000E+00
.1100E+03	.8500E+02	.0000E+00	.9825E+03	.0000E+00
.1200E+03	.2600E+03	.0000E+00	.1181E+04	.0000E+00
.1300E+03	.2600E+03	.0000E+00	.1439E+04	.0000E+00
.1400E+03	.1750E+03	.0000E+00	.1685E+04	.0000E+00
.1500E+03	.3500E+03	.0000E+00	.1921E+04	.0000E+00
.1600E+03	.3500E+03	.0000E+00	.2315E+04	.0000E+00
.1700E+03	.3500E+03	.0000E+00	.2710E+04	.0000E+00
.1800E+03	.3500E+03	.0000E+00	.3104E+04	.0000E+00
.1900E+03	.3500E+03	.0000E+00	.3498E+04	.0000E+00
.2000E+03	.3500E+03	.0000E+00	.3892E+04	.0000E+00
.2100E+03	.3500E+03	.0000E+00	.4287E+04	.0000E+00
.2200E+03	.3500E+03	.0000E+00	.4681E+04	.0000E+00
.2300E+03	.3500E+03	.0000E+00	.5075E+04	.0000E+00
.2400E+03	.1750E+03	.0000E+00	.5272E+04	.0000E+00
.2500E+03	.3500E+03	.2810E+01	.5722E+04	.2810E+02
.2600E+03	.1750E+03	.2810E+01	.6014E+04	.5620E+02
.2700E+03	.3500E+03	.5578E+01	.6469E+04	.1064E+03
.2800E+03	.3500E+03	.5578E+01	.6975E+04	.1622E+03
.2900E+03	.3500E+03	.5578E+01	.7480E+04	.2180E+03
.3000E+03	.3500E+03	.5578E+01	.7986E+04	.2738E+03
.3100E+03	.3500E+03	.5578E+01	.8492E+04	.3296E+03
.3200E+03	.3500E+03	.5578E+01	.8997E+04	.3853E+03
.3300E+03	.3500E+03	.5578E+01	.9503E+04	.4411E+03
.3400E+03	.3500E+03	.5578E+01	.1001E+05	.4969E+03
.3500E+03	.3500E+03	.5578E+01	.1051E+05	.5527E+03
.3600E+03	.3500E+03	.5578E+01	.1102E+05	.6085E+03
.3700E+03	.3500E+03	.5578E+01	.1152E+05	.6642E+03
.3800E+03	.1750E+03	.2768E+01	.1183E+05	.6975E+03
.3900E+03	.3500E+03	.2768E+01	.1216E+05	.7252E+03
.4000E+03	.1750E+03	.0000E+00	.1258E+05	.7418E+03
.4100E+03	.1750E+03	.0000E+00	.1284E+05	.7418E+03
.4200E+03	.3500E+03	.0000E+00	.1335E+05	.7418E+03
.4300E+03	.3500E+03	.0000E+00	.1388E+05	.7418E+03
.4400E+03	.3500E+03	.0000E+00	.1441E+05	.7418E+03
.4500E+03	.3500E+03	.0000E+00	.1494E+05	.7418E+03
.4600E+03	.3500E+03	.0000E+00	.1547E+05	.7418E+03
.4700E+03	.3500E+03	.0000E+00	.1601E+05	.7418E+03
.4800E+03	.3500E+03	.0000E+00	.1654E+05	.7418E+03
.4900E+03	.3500E+03	.0000E+00	.1707E+05	.7418E+03
.5000E+03	.3500E+03	.0000E+00	.1760E+05	.7418E+03
.5100E+03	.3500E+03	.0000E+00	.1813E+05	.7418E+03
.5200E+03	.3500E+03	.0000E+00	.1866E+05	.7418E+03
.5300E+03	.3500E+03	.0000E+00	.1920E+05	.7418E+03
.5400E+03	.3500E+03	.0000E+00	.1973E+05	.7418E+03
.5500E+03	.3500E+03	.0000E+00	.2026E+05	.7418E+03
.5600E+03	.3500E+03	.0000E+00	.2079E+05	.7418E+03

.5700E+03	.3500E+03	.0000E+00	.2132E+05	.7418E+03
.5800E+03	.3500E+03	.0000E+00	.2185E+05	.7418E+03
.5900E+03	.3500E+03	.0000E+00	.2238E+05	.7418E+03
.6000E+03	.3500E+03	.0000E+00	.2292E+05	.7418E+03
.6100E+03	.3500E+03	.0000E+00	.2345E+05	.7418E+03
.6200E+03	.3500E+03	.0000E+00	.2398E+05	.7418E+03
.6300E+03	.3500E+03	.0000E+00	.2451E+05	.7418E+03
.6400E+03	.1750E+03	.0000E+00	.2501E+05	.7418E+03
.6500E+03	.1750E+03	.0000E+00	.2528E+05	.7418E+03
.6600E+03	.3500E+03	.1750E+03	.2555E+05	.1792E+04
.6700E+03	.1750E+03	.1750E+03	.2568E+05	.3542E+04
.6800E+03	.1750E+03	.1750E+03	.2568E+05	.5292E+04
.6900E+03	.2200E+03	.2200E+03	.2568E+05	.7492E+04
.7000E+03	.2200E+03	.2200E+03	.2568E+05	.9692E+04
.7100E+03	.2200E+03	.2200E+03	.2568E+05	.1189E+05
.7200E+03	.1750E+03	.1750E+03	.2568E+05	.1400E+05
.7300E+03	.1750E+03	.1750E+03	.2568E+05	.1575E+05
.7400E+03	.1750E+03	.1750E+03	.2568E+05	.1750E+05
.7500E+03	.1750E+03	.1750E+03	.2568E+05	.1925E+05
.7600E+03	.1750E+03	.1750E+03	.2568E+05	.2100E+05
.7700E+03	.1750E+03	.1750E+03	.2568E+05	.2275E+05
.7720E+03	.1750E+03	.1750E+03	.2568E+05	.2310E+05
MINIMUM	.8500E+02	.0000E+00	.0000E+00	.0000E+00
MAXIMUM	.1400E+04	.2200E+03	.2568E+05	.2310E+05

PLOT NUMBER 1
RUN NUMBER 1

SCALES OF PLOT

B=BRINE FLOW	.0000E+00	.5000E+03	.1000E+04	.1500E+04	.2000E+04
O=OIL FLOW	.0000E+00	.2500E+03	.5000E+03	.7500E+03	.1000E+04
C=CAVERN VOLUM	.0000E+00	.5000E+05	.1000E+06	.1500E+06	.2000E+06
V=OIL VOLUME	.0000E+00	.5000E+05	.1000E+06	.1500E+06	.2000E+06

```
                0   5  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80  85  90  95 100 DUPLICATES
TIME
 .0000E+00  C                       O  +                   +                   B  +                   + CV
 .1000E+02  O    B                     +                   +                      +                   + OC OV
 .2000E+02  O    B                     +                   +                      +                   + OC OV
 .3000E+02  O    B                     +                   +                      +                   + OC OV
 .4000E+02  O    B                     +                   +                      +                   + OC OV
 .5000E+02  O    B                     +                   +                      +                   + OC OV
 .6000E+02  O    B                     +                   +                      +                   + OC OV
 .7000E+02  O    B                     +                   +                      +                   + OC OV
 .8000E+02  O    B                     +                   +                      +                   + OC OV
 .9000E+02  O    B                     +                   +                      +                   + OC OV
 .1000E+03  O        B                 +                   +                      +                   • OC OV
 .1100E+03  O   B                      +                   +                      +                   + OC OV
 .1200E+03  OC           B             +                   +                      +                   • OV
 .1300E+03  OC           B             +                   +                      +                   + OV
 .1400E+03  OC       B                 +                   +                      +                   + OV
 .1500E+03  OC                   B     +                   +                      +                   + OV
 .1600E+03  OC                   B     +                   +                      +                   + OV
 .1700E+03  OC                   B     +                   +                      +                   + OV
 .1800E+03  O C                  B     +                   +                      +                   + OV
 .1900E+03  O C                  B     +                   +                      +                   + OV
 .2000E+03  O C                  B     +                   +                      +                   + OV
 .2100E+03  O C                  B     +                   +                      +                   + OV
 .2200E+03  O C                  B     +                   +                      +                   + OV
 .2300E+03  O  C                 B     +                   +                      +                   + OV
 .2400E+03  O  C     B                 +                   +                      +                   + OV
 .2500E+03  O  C                 B     +                   +                      +                   + OV
 .2600E+03  O  C     B                 +                   +                      +                   + OV
 .2700E+03  VO C                 B     +                   +                      +                   +
 .2800E+03  VO C                 B     +                   +                      +                   +
 .2900E+03  VO  C                B     +                   +                      +                   +
 .3000E+03  VO  C                B     +                   +                      +                   +
 .3100E+03  VO  C                B     +                   +                      +                   +
 .3200E+03  VO  C                B     +                   +                      +                   +
 .3300E+03  VO   C               B     +                   +                      +                   +
 .3400E+03  VO   C               B     +                   +                      +                   +
 .3500E+03  VO   C               B     +                   +                      +                   +
 .3600E+03  VO    C              B     +                   +                      +                   +
 .3700E+03  VO    C              B     +                   +                      +                   +
 .3800E+03  O     C  B                 +                   +                      +                   • OV
 .3900E+03  O     C               B     +                   +                      +                   + OV
 .4000E+03  O     C  B                 +                   +                      +                   + OV
 .4100E+03  O     C  B                 +                   +                      +                   + OV
 .4200E+03  O      C              B     +                   +                      +                   • OV
 .4300E+03  O      C              B     +                   +                      +                   + OV
 .4400E+03  O      C              B     +                   +                      +                   + OV
 .4500E+03  O      C              B     +                   +                      +                   + OV
 .4600E+03  O       C             B     +                   +                      +                   + OV
 .4700E+03  O       C             B     +                   +                      +                   + OV
 .4800E+03  O       C             B     +                   +                      +                   + OV
 .4900E+03  O        C            B     +                   +                      +                   + OV
 .5000E+03  O        C            B     +                   +                      +                   + OV
 .5100E+03  O        C            B     +                   +                      +                   + OV
 .5200E+03  O        C            B     +                   +                      +                   + OV
 .5300E+03  O         C           B     +                   +                      +                   + OV
 .5400E+03  O         C           B     +                   +                      +                   + OV
 .5500E+03  O         C           B     +                   +                      +                   + OV
 .5600E+03  O         C           B     +                   +                      +                   + OV
 .5700E+03  O          C          B     +                   +                      +                   + OV
 .5800E+03  O          C          B     +                   +                      +                   + OV
 .5900E+03  O          C          B     +                   +                      +                   + OV
 .6000E+03  O          C          B     +                   +                      +                   + OV
 .6100E+03  O           C         B     +                   +                      +                   • OV
 .6200E+03  O           C         B     +                   +                      +                   + OV
 .6300E+03  O           C         B     +                   +                      +                   + OV
 .6400E+03  O        B   C            +                   +                      +                   + OV
 .6500E+03  O        B   C            +                   +                      +                   + OV
 .6600E+03  +V           C         B  +                   +                      +                   + BO
 .6700E+03  + V      B   C         O  +                   +                      +                   +
 .6800E+03  +  V     B   C         O  +                   +                      +                   +
 .6900E+03  +   V    B C           O  +                   +                      +                   +
 .7000E+03  +    V   B C           O  +                   +                      +                   +
 .7100E+03  +     V  B C           O  +                   +                      +                   +
 .7200E+03  +      V B   C         O  +                   +                      +                   +
 .7300E+03  +      VB    C         O  +                   +                      +                   +
 .7400E+03  +        B   C         O  +                   +                      +                   + BV
 .7500E+03  +        BV  C         O  +                   +                      +                   +
 .7600E+03  +        B V C         O  +                   +                      +                   +
 .7700E+03  +        B VVC         O  +                   +                      +                   +
                0   5  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80  85  90  95 100 DUPLICATES
TIME
```

OUTPUT CONSISTS OF 79 POINT SETS (316 POINTS)

**** V O L U M E S ****

AVERN#	101		102		SITE TOTALS	
DAYS	CAVERN	OIL	CAVERN	OIL	CAVERN	OIL
0	.00000E+00	.00000E+00	.00000E+00	.00000E+00	.000000E+00	.000000E+00
95	.42150E+03	.00000E+00	.42150E+03	.00000E+00	.843010E+03	.000000E+00
103	.47002E+03	.00000E+00	.47002E+03	.00000E+00	.940040E+03	.000000E+00
106	.47002E+03	.00000E+00	.48821E+03	.00000E+00	.958230E+03	.000000E+00
110	.47002E+03	.00000E+00	.51247E+03	.00000E+00	.982490E+03	.000000E+00
113	.47002E+03	.00000E+00	.53066E+03	.00000E+00	.100070E+04	.000000E+00
138	.96284E+03	.00000E+00	.68227E+03	.00000E+00	.164510E+04	.000000E+00
141	.10220E+04	.00000E+00	.68227E+03	.00000E+00	.170420E+04	.000000E+00
145	.11008E+04	.00000E+00	.68227E+03	.00000E+00	.178310E+04	.000000E+00
148	.11600E+04	.00000E+00	.68227E+03	.00000E+00	.184220E+04	.000000E+00
150	.11994E+04	.00000E+00	.72170E+03	.00000E+00	.192110E+04	.000000E+00
230	.27764E+04	.00000E+00	.22987E+04	.00000E+00	.507510E+04	.000000E+00
233	.27764E+04	.00000E+00	.23578E+04	.00000E+00	.513420E+04	.000000E+00
237	.27764E+04	.00000E+00	.24367E+04	.00000E+00	.521310E+04	.000000E+00
240	.27764E+04	.00000E+00	.24958E+04	.00000E+00	.527220E+04	.000000E+00
252	.30797E+04	.33720E+02	.27324E+04	.00000E+00	.581210E+04	.337200E+02
255	.31555E+04	.42150E+02	.27324E+04	.00000E+00	.588790E+04	.421500E+02
259	.32566E+04	.53390E+02	.27324E+04	.00000E+00	.598900E+04	.533900E+02
262	.33325E+04	.61820E+02	.27324E+04	.00000E+00	.606480E+04	.618200E+02
372	.61127E+04	.37092E+03	.55132E+04	.30448E+03	.116260E+05	.675400E+03
375	.61127E+04	.37092E+03	.55891E+04	.31278E+03	.117020E+05	.683700E+03
377	.61127E+04	.37092E+03	.56396E+04	.31832E+03	.117520E+05	.689240E+03
382	.61127E+04	.37092E+03	.57661E+04	.33216E+03	.118790E+05	.703080E+03
387	.61127E+04	.37092E+03	.58925E+04	.34600E+03	.120050E+05	.716920E+03
394	.62987E+04	.37092E+03	.60694E+04	.36538E+03	.123680E+05	.736300E+03
396	.63519E+04	.37092E+03	.61200E+04	.37091E+03	.124720E+05	.741830E+03
401	.64848E+04	.37092E+03	.61200E+04	.37091E+03	.126050E+05	.741830E+03
406	.66176E+04	.37092E+03	.61200E+04	.37091E+03	.127380E+05	.741830E+03
411	.67505E+04	.37092E+03	.61200E+04	.37091E+03	.128700E+05	.741830E+03
482	.86372E+04	.37092E+03	.80067E+04	.37091E+03	.166440E+05	.741830E+03
517	.95673E+04	.37092E+03	.89368E+04	.37091E+03	.185040E+05	.741830E+03
543	.10258E+05	.37092E+03	.96277E+04	.37091E+03	.198860E+05	.741830E+03
639	.12809E+05	.37092E+03	.12179E+05	.37091E+03	.249880E+05	.741830E+03
644	.12809E+05	.37092E+03	.12312E+05	.37091E+03	.251210E+05	.741830E+03
649	.12809E+05	.37092E+03	.12445E+05	.37091E+03	.252540E+05	.741830E+03
654	.12809E+05	.37092E+03	.12577E+05	.37091E+03	.253870E+05	.741830E+03
665	.12809E+05	.22959E+04	.12870E+05	.37091E+03	.256790E+05	.266680E+04
670	.12809E+05	.31709E+04	.12870E+05	.37091E+03	.256790E+05	.354180E+04
675	.12809E+05	.40459E+04	.12870E+05	.37091E+03	.256790E+05	.441680E+04
680	.12809E+05	.49209E+04	.12870E+05	.37091E+03	.256790E+05	.529180E+04
718	.12809E+05	.11571E+05	.12870E+05	.20809E+04	.256790E+05	.136520E+05
772	.12809E+05	.11571E+05	.12870E+05	.11531E+05	.256790E+05	.231020E+05

Distribution:

US DOE SPR PMO (12)
900 Commerce Road East
New Orleans, LA 70123
Attn: E. E. Chapple, PR-63 (5)
D. Guier, PR-622.2
J. Kunkel, PR-622.2
L. Rousseau, PR-622.2
D. Spence, PR-62
J. Wagoner, PR-61
Technical Documentation (2)

Boeing Petroleum Services, Inc. (10)
850 S. Clearview Parkway
New Orleans, LA 70123
Attn: R. Dyer, EF-31
J. Lamare, EF-16 (5)
K. E. Mills, EF-91 (2)
R. L. Mitchell, EF-26 (2)

PB/KBB (7)
880 W. Commerce Road, Suite 301
New Orleans, LA 70123
Attn: T. Eyerman (2)
H. Lombard, EF-90 (4)
J. M. Mackenzie, EF-90

Walk Haydel and Associates
600 Carondelette Street
New Orleans, LA 70130
Attn: J. Mayes

US Department of Energy
Oak Ridge Operations Office
P.O. Box E
Oak Ridge, TN 37831
Attn: John Milloway, Assistant
Manager for SPR

D. W. Sasser, Consultant (5)
1221 Parsifal NE
Albuquerque, NM 87112

Creative Computer Services, Inc. (5)
5301 Central Ave., NE, Suite 915
Albuquerque, NM 87108
Attn: J. Chapman (5)

1512	A. J. Russo
1513	R. J. Gross
6200	V. L. Dugan
6201	D. Engi
6201	K. Jensen
6250	B. W. Marshall
6257	J. K. Linn (5)
6257	D. K. Buchanan (5)
6257	K. L. Goin
6257	J. L. Todd
8024	P. W. Dean
3141	S. A. Landenberger (5)
3144	G. R. Dalphin (5)
3151	W. L. Garner (3)
3154-1	C. H. Dalin (28)(for DOE/OSTI Unlimited Release)