USERS MANUAL

FOR

FENCO - ICE MODEL

LUCIEN POMAR
COMPUTER PROGRAMMER
HYDROLOGIC MODELLING BRANCH
WATER RESOURCES DIVISION
DEPT. OF ENVIRONMENT
ST. JOHN'S, NFLD.
JANUARY 1986.

TABLE OF CONTENTS

SECTION			PAGE
1.0	-	INTRODUCTION	1
2.0	-	DESCRIPTION OF CHANGES	2
3.0	-	HOW TO USE THE PROGRAM " ICE "	3
4.0	-	EXPLANATION OF INPUT DATA	5
5.0	-	APPENDICES	
5.1	-	LISTING OF " ICE " IN IBM BASIC	7
5.2	-	LISTING OF " ICE " IN VAX FORTRAN 77	15
5.3	-	SAMPLE DATA OUTPUT ON IBM BASIC	27
5.4	-	ORIGINAL SAMPLE OF DATA OUTPUT	32

1 . INTRODUCTION.

ICE , is a modelling program used to simulate the upstream progression of the ice cover from goodyear's dam to exploits dam. The program was originally written on a VAX/781 using fortran 77 and has been translated into the basic language to make it operational on the IBM PC microprocessor.

A portion of the original sample data output as well as the corresponding section of data outputed by the translated version of the program can be examined in sections 5.3 and 5.4 pages 28 and 33 respectively. Apart from the internal computation techniques inherent to the machine such as word size used in floating point computation, the outputed data from the two programs are identical.

This report is not meant to be a theoretical explanation of all the computational techniques of the program, but rather an explanation regarding the modifications of the original model, the method of translation and the input/output operations of the program.

additional information regarding the program or the actual simulation of the ice progression as well as detailed computing techniques is available on the report on the hydrotechnical study of the Badger and Rushy pond areas.

There are five sections to this report.

The section ONE discusses the changes and modifications needed during the course of the translation from fortran 77 to IBM basic. The section TWO is a complete explanation on how to use the program on both the IBM PC and the VAX 11/781. The section THREE provides an explanation of the input/output operations. The section FOUR contains a listing of the program in both fortran 77 and IBM advanced basic. The section FIVE contains sample outputs of the original and translated version.

2. DESCRIPTION OF CHANGES.

1.) MAJOR CHANGES.

The major changes done during the actual translation from fortran to basic are as follows :

i) - The two data files, containing the historical data and data pertaining to the particular river respectively have been concatenated hence the path to the device holding the data disk is the same in both reading cases.

ii) - Two functions, one to compute the number of days in a given month and the other to compute the number of days from the year 1900 up to the current date were originally system oriented functions. These fuctions have been written in fortran IV and later translatted into basic.

iii) - The main reading loop in the procedure "read_data" has been changed from a while loop to the more primitive form of " if then goto ". This was a necessary changes due to some input problems in the original program.

iv) - All of the subroutine calls had to be changed to the basic syntax of 'GOSUB linenumber'.

v) - When a subroutine used a parameter passing technique, the parameter hadto be included into the translated subroutine since basic does not allow subroutines with arguements.

vi) - All the common block was totally ignored, but since basic considers all of its variable to be global, the fortran 's common block technique did notpresent any problems.

vii) - All of the fortran syntax was changed to basic syntax.

viii) - Two main structures of 'If then else if...' were changed to several 'If then else 'due to redundancy.

ix) - One subroutine was completely eliminated due to unecessity.

x) - Several nested 'If' structures were changed into small bloks of data which were accessed through the basic 'If<cond> goto... else goto....' structure.

3. HOW TO USE THE PROGRAM.

1) - VAX VERSION.

The following files contained on in-house tape 'I7' must be dumped onto disk space.

i) ICEPROG.FOR - source code of ice model program. (1)
ii) ICEPROG.DAT - data file needed for execution. (2)
iii) NEWICE.CMN - Common block file. (3)
iv) ICE.OUT1 - Sample data output. (4)

If any changes are necessary to the input data files, the section regarding input/output procedures should be consulted and the data changed accordingly using the Vax System editor,otherwise assuming that the files were dumped onto disk under the same names, execution is performed as follows :

```
$ fortran iceprog.for
$ link iceprog
$ run iceprog
```

To view the data :

```
$ type ice.out  - for output to screen.
$ print ice.out - for output to default printer.
```

NOTE : the dollar sign is the VAX system prompt.

2) - IBM PC VERSION.

For the IBM, the current drive should be drive 'A' and the following command should be typed :

INICE

The screen will clear and the message

ANY CHANGES IN SEASON VARIABLES Y/N ?

If any changes are necessary type 'y' and then each variables will in turn be prompted on the screen under its actual name. To change the variable, the new value should be typed in otherwise, the <return> key should be pressed.

If 'N' was typed, each daily data variable will be prompted and their value must be entered.

At this point the program is ready to be executed. The following command should be entered.

ice

The program will executes leaving and updating a message on the screen to indicates the stage of the computation. When execution is completed, the output of the program can be reviewed by typing the command 'ice.out' since the data has been saved on disk under the name 'ice.out'

NOTE : The data drive 'B' should contain a data disk.

EXPLANATION OF INPUT DATA

1). SEASON VARIABLES .

Start year, start month, end year, end month.

The date of the month and year for wich the current analysis is beeing performed should be entered as well as the date of the month and year for which the particular analysis is to end.

start season, end season.

the month of the start of the analysis and the month of the end of the analysis should be entered.

qflush, poradj, texadj, bdradj, frdadj, voladj.

These variables can be changed to adjust the rate of ice cover progression along the river.

qflush - is the flow rate in (m^3/s) which is known to flush ice from the river.

poradj - is a non-dimensional multiplier used to adjust the porosity of the ice cover. The porosity is set within the model to 0.5, hence a value of 1.4 for poradj will set the porosity at 0.7.

texadj - is a value in degree celcius used to change the temperature of discharge entering the river from Exploits dam. This is set at 3.0 degree celcius within the model, hence a value of 1.0 for texadj will set this discharge temperature to 4.0 degree celcius.

bdradj - is a non dimensional multiplier used to increase or decrease the rate of border ice growth (and thus closure) according to the field observations in a given year. If there are no observations, the value is set at 1.0. A value greater then 1.0 will speed the growth. A value less than 1.0 will slow the rate of border ice growth.

frdadj - is a non dimensional multiplier used to change the date of ice closure at the Rushy Pond ice boom. The date of closure is set within the model on a day in which there is no frazil slush passing the boom and Froude number at the boom is 0.08. A positive value above 1.0 will slow closure by increasing the value of the critical Froude number. A value less than 1.o, will speed the closure.

voladj - is a non dimensional multiplier used to ensure that closure at the ice boom takes place on a date when there is sufficient slush to form the start of an ice field at the boom. A value of 500,000 m^3 is assumed in the model, but it may be increased or decreased by corresponding changes in voladj. A value of 1.0 is recommended.

HISTORICAL VARIABLES, (metereological data).

year, month, day, ta, tdew, wind, sun, qe, qgfall.

Year - current year.

month - current month.

day - current day.

ta - air temperature on this particular date.

tdew - equilibrium temperature.

wind - wind chill factor (w/m^3).

sun - coefficient of surface heat exchange.

qe - flow rate at exploit dam.

qgfall - flow rate at grand falls dam.

FENCO - ICE PROGRAM

Developed by Lucien J Pomar
Hydrologic Modelling Branch, Dept. of Environment.
Using IBM PC/8088 processor
December 1985.

```
10 KEY OFF:CLS    ' MAIN PROGRAM SECTION'
20 DIM PLTDAY!(1000),PLTMON!(1000),PLTVOL!(1000),PLTYR!(1000)
30 DIM ICCD!(50),ICCDELA!(50),ICCA!(50),ICCIV!(50),ICCX!(50)
   COVER(50),CCD!(50),CCA!(50),CCIV!(50),CCT!(50),CCX!(50)
   CCDELA(50),CCQ(50)
40 DIM TOT121!(10),OLD121!(10)
50 OPEN "b:ice.out" FOR OUTPUT AS #2
60 GOSUB 320:GOSUB 210:DONE=0:KEY(1) ON
65 LOCATE 10,26:COLOR 22,0:PRINT "COMPUTATION IN PROCESS";:COLOR
   7,0:PRINT
66 LOCATE 12,25:PRINT "DATE   : ";
70 WHILE(DONE = 0)
80      LOCATE 12,35:PRINT CURYR;" ";CURMON;" ";CURDAY
90      ON KEY(1) GOSUB 170
100     GOSUB 720:GOSUB 930
110     IF(FLUSH=1)THEN GOSUB 2690:GOTO 130 ELSE GOSUB 1010
120     IF(ICEPROD=1) THEN GOSUB 2560
130     GOSUB 1370:GOSUB 510
140     LOCATE 12,35:PRINT "
150 WEND
160 GOSUB 2970
170 CLOSE 1:CLOSE 2
180 END
190 REM
200 REM ************************************************************
210 REM
220 REM subroutine init
230 DIM NDAY(12)
240 FOR I=1 TO 12:READ NDAY(I):NEXT
250 NPLT=0:CURDAY=1:CURMON=STARTSEASON:CURYR=STARTYR
260 GOSUB 1680:SEASONDAY=1:TEXPLD!=3!+TEXADJ!:DT!=3600!:E!=.5*PORADJ!
270 TSTEP!=24!*60!*60!:VSCRIT!=VOLADJ!*500000!
280 RETURN
290 REM
300 REM ************************************************************
310 REM
320 REM subrouitne read control
330 OPEN "b:ice.dat" FOR INPUT AS #1
340 INPUT #1,STARTYR,STARTMON,ENDYR,ENDMON
```

```
350 INPUT #1,STARTSEASON,ENDSEASON
360 INPUT #1,QFLUSH!,PORADJ!,TEXADJ!,BDRADJ!,FRDADJ!,VOLADJ!
370 INPUT #1,NCCT,ICCT!,ICOVER!
371 PRINT #2,USING "#### ";STARTYR;:PRINT #2,USING "##
    ";STARTMON;:PRINT #2,USING "#### ";ENDYR;:PRINT #2,USING
    "##";ENDMON
372 PRINT #2, USING "## ";STARTSEASON;ENDSEASON
373 PRINT #2,USING "###.# ";QFLUSH!;:PRINT #2,USING "##.## ";
    PORADJ!;TEXADJ!;BDRADJ!;VOLADJ;
374 PRINT #2,USING "## ";NCCT;:PRINT #2,USING "##.## ";ICCT!;ICOVER!
375 FOR I=1 TO NCCT
376 INPUT #1,NCC,ICCD!(I),ICCA!(I),CCIV!(I),ICCX!(I),ICCDELA!(I)
377 PRINT #2,USING "## "; NCC;:PRINT #2,USING "#.# ";ICCD!(I);:PRINT
    #2,USING "########.# ";ICCA!(I);:PRINT #2,USING "#######.#
    CCIV!(I);:PRINT #2,USING "##.# ";ICCX!(I);:PRINT #2,USING
    "#####.#";ICCDELA!(I)
378 IF(NCC <> I)THEN PRINT #2, " ERROR IN RIVER DATA - NCC ";I:END
379 NEXT
465 PRINT #2," ":PRINT #2," "
470 RETURN
480 REM
490 REM ************************************************************
500 REM
510 REM subroutine check time
520 YEAR=CURYR:MONTH=CURMON:GOSUB 3110
530 IF(CURMON=ENDSEASON)AND(CURDAY=DAYINMON)THEN 540 ELSE 580
540 GOSUB 1590:GOSUB 1680:SEASONDAY=1:CURDAY=1
550 CURMON=STARTSEASON
560 IF(ENDSEASON >= STARTSEASON)THEN CURYR=CURYR + 1
570 GOTO 630
580 SEASONDAY=SEASONDAY + 1:CURDAY=CURDAY + 1
590 IF(CURDAY > DAYINMON)THEN 600 ELSE 630
600 CURDAY = 1:CURMON = CURMON + 1
610 IF(CURMON > 12) THEN 620 ELSE 630
620 CURMON =1 :CURYR=CURYR + 1
630 IF(CURYR > ENDYR)OR((CURYR = ENDYR)AND(CURMON > ENDMON))THEN 640 ELSE
640 IF(STARTSEASON <> CURMON)THEN 650 ELSE 670
650 CURMON=ENDMON:CURYR = ENDYR:SEASONDAY = SEASONDAY - 1
660 GOSUB 1580
670 DONE = 1
680 RETURN
690 REM
700 REM ************************************************************
710 REM
720 REM subroutine readdata
730 REM
740 INPUT #1,YR,MON,DAY,TA!,TDEW!,WIND!,SUN!,QE!,QGFALL!
750 IYEAR=YR:IMONTH=MON:IDAY=DAY:GOSUB 3180:FIRST=DAYNUM
760 IYEAR=CURYR:IMONTH=CURMON:IDAY=CURDAY:GOSUB 3180:SECOND=DAYNUM
770 IF(FIRST >= SECOND) THEN 800
780 INPUT #1,YR,MON,DAY,TA!,TDEW!,WIND!,SUN!,QE!,QGFALL!
790 GOTO 750
```

```
800 IF(YR<>CURYR)OR(MON<>CURMON)OR(DAY<>CURDAY)THEN 810 ELSE 830
810 PRINT #2, "ALL DATA MISSING  FOR  : ";YR;MON;DAY
820 TA!=-999:TDEW!=-999:WIND!=-999:SUN!=-999:QE!=-999:QGFALL!=-999
830 IF(TA!=-999!)THEN TA!=PREVTA! ELSE PREVTA!=TA!
840 IF(TDEW!=-999!)THEN TDEW!=PREVTDEW! ELSE PREVTDEW!=TDEW!
850 IF(WIND!=-999!)THEN WIND!=PREVWIND! ELSE PREVWIND!=WIND!
860 IF(SUN!=-999!)THEN SUN!=4!
870 IF(QE!=-999!)THEN QE!=PREVQE! ELSE PREVQE!=QE!
880 IF(QGFALL!=-999!)THEN QGFALL!=QE!/35.3147
890 RETURN
900 REM
910 REM ************************************************************
920 REM
930 REM subroutine conv
940 TAIR!=((9!*TA!)/5!)+32!:TD!=((9!/5!)*TDEW!)+32!:QEXPL!=
    QE!/35.3147:QGF!=QGFALL!:WS!=.62137*WIND!
950 IF(QEXPL!>=QFLUSH!)OR(QGF!>=QFLUSH!)THEN FLUSH=1 ELSE FLUSH=0
960 FOR I=1 TO NCCT:CCQ!(I)=QEXPL!:NEXT
970 RETURN
980 REM
990 REM ************************************************************
1000 REM
1010 REM boolean function iceprod
1020 ICEPROD=0:GOSUB 1280:CCT!(1)=TEXPLD!
1030 FOR J=1 TO 24
1040     FOR I=2 TO NCCT
1050       CCV!=CCD!(I)*CCA!(I)
1060 CCT!(I)=(CCV!*CCT!(I)+CCQ!(I)*DT!*CCT!(I-1)+CSHEMS!*CCA!(I)*DT!*
     ETC!)/(CCV! +CCQ!(I)*DT!+CSHEMS!*CCA!(I)*DT!)
1070 IF(CCT!(I)< -.01)THEN CCT!(I)=-.01
1080 NEXT
1090 NEXT
1100 IF(TA! < .1)THEN 1110 ELSE SLS121! = 0:GOTO 1170
1110 IF(BLOCKFLAG = 0)THEN GOSUB 2830
1120 IF(BLOCKFLAG = 1) THEN 1130 ELSE 1170
1130 GOSUB 2440
1140 IF(ACCUMULATEFLAG = 0)THEN 1150 ELSE 1160
1150 IF(DAYSLS! > VSCRIT!)THEN ACCUMULATEFLAG = 1:PRINT #2,"RIVER
     CLOSURE AT BOOM." ELSE BLOCKFLAG = 0
1160 IF(ACCUMULATEFLAG = 1) THEN ICEPROD=1
1170 FOR I=1 TO 10
1180   TOT121!(I)=TOT121!(I)+SLS121!-OLD121!(I)
1190 NEXT
1200 FOR I=10 TO 2 STEP -1
1210   OLD121!(I)=OLD121!(I-1)
1220 NEXT
1230 OLD121!(1)=SLS121!
1240 RETURN
1250 REM
1260 REM ************************************************************
1270 REM
```

```
 1280 REM subroutine heatcalc
 1290 GOSUB 1820:GOSUB 2020:DEGDY!=0!-TA!
 1300 IF(DEGDY!<0!)THEN DEGDY!=0!
 1310 DEGSUM!=DEGSUM! + DEGDY!
 1320 WCF!=(6.127001*SQR(WIND!)+12.15-(.3229*WIND!))*(33!-TA!)
 1330 RETURN
 1340 REM
 1350 REM ************************************************************
 1360 REM
 1370 REM subroutine dayoutput
 1380 PRINT #2, " DATE (year/mon/day) :
      ";CURYR;"/";CURMON;"/";CURDAY;" AIR TEMPERATURE :  ";:PRINT #2
      ,USING "####.#-";TA:PRINT #2," "
  390 PRINT #2, " ET ETC WCF CSHE CSHEMS CSHEWM DEGDY DEGSUM QEXPL QGF"
 1400 PRINT #2,USING "####.#-
      ";ET!;ETC!;WCF!;CSHE!;CSHEMS!;CSHEWM!;DEGDY!;DEGSUM!;QEXPL!;QGF!:PRIN
      #2," "
 1410 PRINT #2,"COOLING CANAL RIVER SEGMENTS TEMPERATURE :"
 1420 FOR I=1 TO 9:PRINT #2,USING "###.##- ";CCT(I);:NEXT
 1430 FOR I=10 TO 19:PRINT #2,USING "###.##- ";CCT(I);:NEXT
 1440 FOR I=20 TO 29:PRINT #2,USING "###.##- ";CCT(I);:NEXT
 1441 FOR I=30 TO 32:PRINT #2,USING "###.##- ";CCT(I);:NEXT:PRINT #2,"
 1450 PRINT #2," ":PRINT #2,"TOTAL SLUSH GENERATED THIS DAY :";DAYSLS
 1460 PRINT #2,"TOTAL SLUSH GENERATED THIS DAY SEGS 1 TO 21 :";SLS121
 1470 PRINT #2,"TOTAL SLUSH GENERATED TO DATE : ";YRSLST:PRINT #2," "
 1480 PRINT #2,"RUNNING NDAY TOTAL SLUSH PRODUCED IN SEGS 1 TO
      21":PRINT #2," "
 1490 PRINT #2,"DAY :  1            2            3            4            5"
 1500 FOR I=1 TO 5:PRINT #2,USING "######.# ";TOT121(I);:NEXT:PRINT #2,
      " ";:PRINT #2," "
 1510 PRINT #2,"DAY :  6            7            8            9           10"
 1520 FOR I=6 TO 10:PRINT #2,USING "######.# ";TOT121(I);:NEXT:PRINT
      #2," "
 1530 PRINT #2," ":PRINT #2,"FRONT OF ICE FIELD TODAY IS AT RIVER
      SECTION :";ICELOC
 1540 PRINT #2," ":PRINT #2,"-----------------------------------------
 1550 IF(PLTCNT=3)THEN
      NPLT=NPLT+1:PLTVOL(NPLT)=ICELOC:PLTDAY(NPLT)=CURDAY:PLTMON(NPLT)=
      CURMON:PLTYR(NPLT)=CURYR:PLTCNT=1  ELSE PLTCNT=PLTCNT+1
 1560 RETURN
 1570 REM ************************************************************
 1580 REM
 1590 REM subroutine SEASONREPT
 1600 PRINT  #2,"TOTAL DEGREE DAYS FOR SEASON ";DEGSUM!
 1610 PRINT #2,"SEASON ENDING ";CURYR;CURMON;
 1620 PRINT #2, " HAD ";SEASONDAY;"  DAYS"
 1630 PRINT #2," ":PRINT #2," "
 1640 RETURN
 1650 REM
 1660 REM ************************************************************
 1670 REM
 1680 REM subroutine INITSEASON
 1690 FOR I=1 TO NCCT
 1700   CCT!(I)=ICCT!:COVER!(I)=ICOVER!:CCD!(I)=ICCD!(I):CCA!(I)=ICCA!(I)
```

```
1710   CCX!(I)=ICCX!(I):CCDELA!(I)=ICCDELA!(I)
1720 NEXT
1730 ICELOC=NCCT:YRSLST!=0!:DAYSLS!=0!:SLS121!=0!:BLOCKFLAG=0
1740 BDRICE!=0!:BDRMAX!=(.5*CCA!(NCCT))/2500!:DEGSUM!=0!
1750 FOR I=1 TO 10
1760   TOT121!(I)=0!:OLD121!(I)=0!
1770 NEXT
1780 RETURN
1790 REM
1800 REM ************************************************************
1810 REM
1820 REM subroutine CALCSOLAR
1830 B!=47.15:C!=1.74
1840 IYEAR=CURYR:IMONTH=CURMON:IDAY=CURDAY:GOSUB 3180
     :FIRST=DAYNUM:IMONTH=1:IDAY=1:GOSUB 3180:SECOND=DAYNUM
1850 DAYJUL=FIRST-SECOND+1
1860 IF(CURMON=12)THEN A!=26.39:SSMAX!=8.32
1870 IF(CURMON=1) THEN A!=29.46:SSMAX!=8.770001
1880 IF(CURMON=2)AND(CURDAY<10)THEN A!=39.61:SSMAX!=9.479999 ELSE
     IF(CURMON=2) THEN A!=52.03:SSMAX!=10.2
1890 IF(CURMON=3) THEN A!=52.03:SSMAX!=11.9
1900 SSRATIO!=SUN!/SSMAX!
1910 IF(SSRATIO!>1!)THEN SSRATIO!=1!
1920 CC!=(1!-SSRATIO!)*10!
1930 ARG!=(.01717*DAYJUL+C)*.017453293#
1940 IF CURMON=3 THEN SRHO!=9.717 + .1265*CURDAY ELSE
     SRHO!=(A!-B!*SIN(ARG!))*(24!/88.114)
1950 SRO!=(1!-.0071*CC!*CC!)*SRHO!
1960 HS!=SRO!*87.97799
1970 RETURN
1980 REM
1990 REM
2000 REM ************************************************************
2010 REM
2020 REM subroutine keidsr
2030 LESP!=TAIR!+460!:LESP!=LESP!*LESP!:LESP!=LESP!*LESP!
2040 ET!=TD!:FW!=70!+(.7*WS!*WS!):HA!=3.1872E-08*LESP!:I=0:DDN=0
2050 WHILE(DDN=0)
2060   TSTAR!=(ET!+TD!)/2!
2070 TST!=TSTAR!*TSTAR!
2080   BETA!=.255-(8.500001E-03*TSTAR!)+(.000204*TST!)
2090   CSHE!=15.7+(.26+BETA!)*FW!
2100   ETP!=(HS!+HA!-1801!)/CSHE!+(CSHE!-15.7)*(.26*TAIR!+BETA!*TD!)
      /(CSHE!*(.26+BETA!))
2110 IF(ABS(ETP!-ET!)<.05)THEN DDN=1 ELSE IF(I<50) THEN ET!=ETP!:I=I+1
     ELSE PRINT #2,"DOES NOT CONVERGE";CURYR;CURMON;CURDAY:DDN=1
2120 WEND
2130 ETC!=5!*(ET!-32!)/9!:CSHEWM!=.2364*CSHE!:CSHEMS!=5.65E-08*CSHE!
2140 RETURN
2150 REM
```

```
2160 REM ************************************************************
2170 REM
2180 REM subroutine ICECALCS
2190 CCT!(1)=TEXPLD!
2200 FOR J=1 TO 24
2210     FOR I=2 TO NCCT
2220       CCV!=CCD!(I)*CCA!(I)
2230 CCT!(I)=(CCV!*CCT!(I)+CCQ!(I)*DT!*CCT!(I-1)+CSHEMS!*
     CCA!(I)*DT*ETC)/(CCV!+CCQ!(I)*DT!+CSHEMS!*CCA!(I)*DT!)
2240 IF(CCT!(I)< -.01)THEN CCT!(I)=-.01
2250 NEXT
2260 NEXT
2270 IF(TA! < .1)THEN 2280 ELSE SLS121! = 0:GOTO 2340
2280 IF(BLOCKFLAG = 0)THEN GOSUB 2830
2290 IF(BLOCKFLAG = 1) THEN 2300 ELSE 2280
2300 GOSUB 2440
2310 IF(ACCUMULATEFLAG = 0)THEN 2320 ELSE 2330
2320 IF(DAYSLS! > 500000!)THEN ACCUMULATEFLAG = 1:PRINT #2,"RIVER
     CLOSURE AT BOOM." ELSE BLOCKFLAG = 0
2330 IF(ACCUMULATEFLAG = 1) THEN GOSUB 2560
2340 FOR I=1 TO 10
2350   TOT121!(I)=TOT121!(I)+SLS121!-OLD121!(I)
2360 NEXT
2370 FOR I=10 TO 2 STEP -1
2380   OLD121!(I)=OLD121!(I-1)
2390 NEXT
2400 OLD121!(1)=SLS121!
2410 RETURN
2420 REM
2430 REM ************************************************************
2440 REM subroutine ICEPRODUCTION
2450 DAYSLS!=0!:SLS121!=0!
2460 FOR I=1 TO NCCT
2470 IF(CCT!(I) < 0!) THEN 2480 ELSE 2520
2480 TSUBF! = 0! - TA!:VFRAZ! = (CSHEWM! * CCA!(I) * TSUBF!* TSTEP!)
     /(900 * 334000!)
2490 SEGSLS!= (VFRAZ!/E!)*(1 - COVER!(I))
2500 DAYSLS!= DAYSLS!+ SEGSLS!
2510 IF(I <= 21) THEN SLS121!= SLS121!+ SEGSLS!
2520 NEXT
2530 RETURN
2540 REM
2550 REM ************************************************************
2560 REM subroutine ICEACCUMULATION
2570 IF(DAYSLS!<=0!)THEN RETURN
2580 TEMPSLS!=DAYSLS!:YRSLST!=YRSLST!+TEMPSLS!
2590 WHILE(TEMPSLS!>0!)AND(ICELOC>1)
2600    CAP!=(1!-COVER!(ICELOC))*CCIV!(ICELOC)
2610    IF(CAP!> TEMPSLS!)THEN 2620 ELSE 2640
2620 CAP!=CAP!-TEMPSLS!:COVER!(ICELOC)=1! -
     CAP!/CCIV!(ICELOC):TEMPSLS!=0
2630    GOTO 2650
```

```
2640    TEMPSLS!= TEMPSLS!- CAP!:COVER!(ICELOC)= 1:ICELOC=ICELOC - 1
2650 WEND
2660 RETURN
2670 REM
2680 REM ************************************************************
2690 REM subroutine FLUSHRIVER
2700 PRINT  #2,"HIGH FLOW  -  ICE FLUSHED FROM RIVER"
2710 ICELOC=NCCT:YRSLST!=0!:DAYSLS!=0!
2720 FOR I=1 TO NCCT
2730   COVER!(I)=0!:CCT!(I)=2!
2740 NEXT
2750 BLOCKFLAG=0:ACCUMULATEFLAG=0:BDRICE!=0!
2760 FOR I=1 TO 10
2770   TOT121!(I)=0!
2780   OLD121!(I)=0!
2790 NEXT
2800 RETURN
2810 REM
2820 REM ************************************************************
2830 REM subroutine block RIVER
2840 IF(CURYR>1974)OR((CURYR=1974)AND(CURMON>=12))THEN
     BDRVEL!=.0012*QGF! ELSE BDRVEL!=.0012*QGF! +.1
2850 BDRICE!=BDRICE!+((.54/BDRVEL!^1.5)*DEGDY!)*BDRADJ!
2860 IF(BDRICE!>BDRMAX!)THEN 2870 ELSE 2890
2870 BLOCKFLAG = 1:ACCUMULATEFLAG = 1
2880 PRINT #2, "RIVER CLOSURE DUE TO BORDER ICE."
2890 IF(BLOCKFLAG = 0)THEN 2900 ELSE 2930
2900 IF(CURYR>1974)OR((CURYR=1974)AND(CURMON>=12))THEN
     DBOOM!=1.79+.00259 * QGF!:VBOOM!=6.150001E-04*QGF! ELSE DBOOM!=
    .00457 * QGF!:VBOOM!= .33 +.000251 *QGF!
2910 FROUDE!= (VBOOM!/SQR(9.8 * DBOOM!))*FRDADJ!
2920 IF(FROUDE!<= .08)THEN BLOCKFLAG=1
2930 RETURN
2940 REM
2950 REM ************************************************************
2960 REM
2970 REM subroutine print PLOT
2980 PRINT #2," '1' ICE FRONT PLOT - THE LOCATION OF THE ICE FRONT EACH DA
     REPRESENTED BY THE RIGHT MOST ASTERISK "
2990 PRINT #2,"SECTION #   ";
2991 FOR J=0 TO 3:PRINT #2,USING "#         ";J;:NEXT:PRINT #2," "
2992 PRINT #2,"SECTION #   ";:FOR K=1 TO 4:FOR J=0 TO 9:PRINT #2,USING "#"
     ;J;:NEXT J:NEXT K
2993 PRINT #2," "
3000 FOR I=1 TO NPLT
3010 PRINT #2,USING "## ";PLTDAY(I);PLTMON(I);:PRINT #2,USING "#### ";PLTY
3020 FOR J=1 TO PLTVOL(I)
3030 PRINT #2,"*";
3040 NEXT
3050 PRINT #2," "
3060 NEXT
```

```
3070 RETURN
3080 REM
3090 REM ************************************************************
3100 REM
3110 REM subroutine DAYINMON
3120 IF(MONTH=2)THEN 3130 ELSE 3150
3130 IF(YEAR MOD 4 = 0)THEN DAYINMON = 29 ELSE DAYINMON = 28
3140 RETURN
3150 DAYINMON=NDAY(MONTH):RETURN
3160 REM
3170 REM ************************************************************
3180 REM FUNCTION DAYNUM
3190 BASEYR=1952:BASEDY=1:BASEMT=1:IDIFF=IYEAR-BASEYR:NDAYS=IDIFF*365
3200 LASTYR=IYEAR-1
3210 FOR IDX=BASEYR TO LASTYR
3220   IF(IDX MOD 4=0) THEN NDAYS=NDAYS+1
3230 NEXT
3240 JMONTH=IMONTH-1
3250 IF JMONTH <= 0 THEN 3300
3260 FOR IDX=1 TO JMONTH
3270    YEAR=IYEAR:MONTH=IDX:GOSUB 3110
3280    NDAYS=NDAYS+DAYINMON
3290 NEXT
3300 NDAYS=NDAYS+IDAY:DAYNUM=NDAYS
3310 RETURN
3320 DATA 31,27,31,30,31,30,31,31,30,31,30,31
```

FENCO - ICE PROGRAM (Original)

```
C      MAIN PROGRAM
       INCLUDE 'NEWICE.CMN/LIST'
       LOGICAL*1 DONE ,ICE_PROD
       OPEN(UNIT=8,FILE='ICE.OUT',STATUS='NEW')
       CALL READ_CONTROL
       CALL INIT
       DONE=.FALSE.
       DO WHILE(.NOT.DONE)
          CALL READ_DATA
          CALL CONV
          IF(FLUSH)THEN
               CALL FLUSH_RIVER
          ELSE IF(ICE_PROD())THEN
               CALL ICE_ACCUMULATION
          END IF
          CALL DAY_OUTPUT
          CALL CHECK_TIME(DONE)
       END DO
       CALL FINAL_OUTPUT
       CLOSE(UNIT=2)
       CLOSE(UNIT=8)
       STOP
       END
C*************************************************************
       SUBROUTINE INIT
       INCLUDE 'NEWICE.CMN'
       NPLT=0
       CUR_DAY=1
       CUR_MON=START_SEASON
       CUR_YR=START_YR
       CALL INIT_SEASON
       SEASON_DAY=1
       TEXPLD=3.+TEXADJ
       DT=3600.
       E=.5*PORADJ
       TSTEP=1.*24.*60.*60.
       VSCRIT=VOLADJ*500000.
       RETURN
       END
C*************************************************************************
       SUBROUTINE READ_CONTROL
       INCLUDE 'NEWICE.CMN'
       INTEGER NCC,I
       OPEN(UNIT=2,FILE='ICE.DAT',STATUS='OLD')
```

```
      READ(2,*)START_YR,START_MON,END_YR,END_MON
      READ(2,*)START_SEASON,END_SEASON
      READ(2,*)QFLUSH,PORADJ,TEXADJ,BDRADJ,FRDADJ,VOLADJ
      READ(2,*)NCCT,ICCT,ICOVER
      WRITE(8,810)START_YR,START_MON,END_YR,END_MON
  810 FORMAT(' ',I4,1X,I2,1X,I4,1X,I2)
      WRITE(8,820)START_SEASON,END_SEASON
  820 FORMAT(' ',I2,1X,I2)
      WRITE(8,825)QFLUSH,PORADJ,TEXADJ,BDRADJ,FRDADJ,VOLADJ
  825 FORMAT(' ',F6.0,5(1X,F4.2))
      WRITE(8,830)NCCT,ICCT,ICOVER
  830 FORMAT(' ',I2,1X,F4.1,1X,F4.2)
      DO I=1,NCCT
         READ(2,*)NCC,ICCD(I),ICCA(I),CCIV(I),ICCX(I),ICCDELA(I)
         WRITE(8,840)NCC,ICCD(I),ICCA(I),CCIV(I),ICCX(I),ICCDELA(I)
  840    FORMAT(' ',I2,1X,F4.1,1X,F9.0,1X,F10.1,1X,F5.1,1X,F5.0)
         IF(NCC.NE.I)THEN
            WRITE(8,*)'ERROR IN RIVER DATA - NCC ',I
         END IF
      END DO
      return
      END
C*************************************************************

      SUBROUTINE CHECK_TIME(DONE)
      INCLUDE 'NEWICE.CMN'
      LOGICAL*1 DONE
      INTEGER DAY_IN_MON
      IF(CUR_MON.EQ.END_SEASON.AND.
     &CUR_DAY.EQ.DAY_IN_MON(CUR_YR,CUR_MON))THEN
         CALL SEASON_REPT
         CALL INIT_SEASON
         SEASON_DAY=1
         CUR_DAY=1
         CUR_MON=START_SEASON
         IF(END_SEASON.GE.START_SEASON)THEN
            CUR_YR=CUR_YR+1
         END IF
      ELSE
         SEASON_DAY=SEASON_DAY+1
         CUR_DAY=CUR_DAY+1
         IF(CUR_DAY.GT.DAY_IN_MON(CUR_YR,CUR_MON))THEN
            CUR_DAY=1
            CUR_MON=CUR_MON+1
            IF(CUR_MON.GT.12)THEN
               CUR_MON=1
               CUR_YR=CUR_YR+1
            END IF
        END IF
      END IF
      if(cur_yr.gt.end_yr.or.
```

```
      &(cur_yr.eq.end_yr.and.cur_mon.gt.end_mon))then
         IF(START_SEASON.NE.CUR_MON)THEN
            CUR_MON=END_MON
            CUR_YR=END_YR
            SEASON_DAY=SEASON_DAY-1
            CALL SEASON_REPT
         END IF
         DONE=.TRUE.
      END IF
      RETURN
      END
C*****************************************************************

   SUBROUTINE READ_DATA
   INCLUDE 'NEWICE.CMN'
   INTEGER  DAY_NUM
           ,YR/1900/,MON/1/,DAY/1/
       REAL      PREV_TA,PREV_TDEW,PREV_WIND,PREV_QE,PREV_QGFALL
       READ(2,*)YR,MON,DAY,TA,TDEW,WIND,SUN,QE,QGFALL
 501   IF(DAY_NUM(YR,MON,DAY).GE.DAY_NUM(CUR_YR,CUR_MON,CUR_DAY))
      & GOTO 500
          READ(2,*)YR,MON,DAY,TA,TDEW,WIND,SUN,QE,QGFALL
          GOTO 501
 500      CONTINUE
       IF(YR.NE.CUR_YR.OR.MON.NE.CUR_MON.OR.DAY.NE.CUR_DAY)THEN
          WRITE(8,*)'ALL DATA MISSING FOR ',YR,MON,DAY
          TA=-999.
          TDEW=-999.
          WIND=-999.
          SUN=-999.
          QE=-999.
          QGFALL=-999.
       END IF
       IF(TA.EQ.-999.)THEN
          TA=PREV_TA
       ELSE
         PREV_TA=TA
       END IF
       IF(TDEW.EQ.-999.)THEN
           TDEW=PREV_TDEW
       ELSE
          PREV_TDEW=TDEW
       END IF
       IF(WIND.EQ.-999.)THEN
         WIND=PREV_WIND
       ELSE
         PREV_WIND=WIND
       END IF
       IF(SUN.EQ.-999.)THEN
         SUN=4.0
       END IF
```

```
      IF(QE.EQ.-999.)THEN
        QE=PREV_QE
      ELSE
        PREV_QE=QE
      END IF
      IF(QGFALL.EQ.-999.)THEN
        QGFALL=QE/35.3147
      END IF
      RETURN
      END
C*****************************************************************

      SUBROUTINE CONV
      INCLUDE 'NEWICE.CMN'
      INTEGER I
      TAIR=(9./5.)*TA+32.
      TD=(9./5.)*TDEW+32.
      QEXPL=QE/35.3147
      QGF=QGFALL
      WS=.62137*WIND
      IF(QEXPL.GE.QFLUSH.OR.QGF.GE.QFLUSH)THEN
         FLUSH=.TRUE.
      ELSE
         FLUSH=.FALSE.
      END IF
      DO I=1,NCCT
         CCQ(I)=QEXPL
      END DO
      RETURN
      END
C*****************************************************************

      LOGICAL*1 FUNCTION ICE_PROD
      INCLUDE 'NEWICE.CMN'
      REAL CCV
      INTEGER I,J
      ICE_PROD=.FALSE.
      CALL HEAT_CALC
      CCT(1)=TEXPLD
      DO J=1,24
         DO I=2,NCCT
            CCV=CCD(I)*CCA(I)
            CCT(I)=(CCV*CCT(I)+CCQ(I)*DT*CCT(I-1)+CSHEMS*CCA(I)*DT
     &      *ETC)/(CCV+CCQ(I)*DT+CSHEMS*CCA(I)*DT)
            IF(CCT(I).LT.-.01)THEN
               CCT(I)=-.01
            END IF
         END DO
      END DO
      IF(TA.LT.0.1)THEN
         IF(.NOT.BLOCK_FLAG)THEN
```

```
            CALL BLOCK_RIVER
         END IF
         IF(BLOCK_FLAG)THEN
            CALL ICE_PRODUCTION
            IF(.NOT.ACCUMULATE_FLAG)THEN
               IF(DAYSLS.GT.VSCRIT)THEN
                  ACCUMULATE_FLAG=.TRUE.
                  WRITE(8,*)' *** RIVER CLOSURE AT BOOM ***'
               ELSE
                  BLOCK_FLAG=.FALSE.
               END IF
            END IF
            IF(ACCUMULATE_FLAG)THEN
               ICE_PROD=.TRUE.
            END IF
         END IF
      ELSE
         SLS121=0.0
      END IF
      DO I=1,10
         TOT121(I)=TOT121(I)+SLS121-OLD121(I)
      END DO
      DO I=10,2,-1
         OLD121(I)=OLD121(I-1)
      END DO
      OLD121(1)=SLS121
      RETURN
      END
C*************************************************************

      SUBROUTINE HEAT_CALC
      INCLUDE 'NEWICE.CMN'
      CALL CALC_SOLAR
      CALL KEIDSR
      DEGDY=0.-TA
      IF(DEGDY.LT.0.)THEN
         DEGDY=0.0
      END IF
      DEGSUM=DEGSUM+DEGDY
      WCF=(6.127*SQRT(WIND)+12.15-(.3229*WIND))*(33.-TA)
      RETURN
      END
C*************************************************************

      SUBROUTINE DAY_OUTPUT
      INCLUDE 'NEWICE.CMN'
      INTEGER I,PLTCNT/3/
      WRITE(8,805)'YEAR','MO','DA','TA','ET','ETC','WCF','CSHE',
     &'CSHEMS','CSHEWM','DEGDY','DEGSUM','QEXPL','QGF'
  805 FORMAT(' ',A4,2(1X,A2),11(1X,A6))
      WRITE(8,810)CUR_YR,CUR_MON,CUR_DAY,TA,ET,ETC,WCF,CSHE,CSHEMS,
```

```
     &CSHEWM,DEGDY,DEGSUM,QEXPL,QGF
  810 FORMAT(' ',I4,2(1X,I2),11(1X,F6.1))
      WRITE(8,*)'COOLING CANAL/RIVER SEGMENT TEMPERATURES'
      WRITE(8,820)(CCT(I),I=1,16)
      WRITE(8,820)(CCT(I),I=17,32)
  820 FORMAT(' ',16(1X,F5.2))
      WRITE(8,*)'TOTAL SLUSH GENERATED THIS DAY = ',DAYSLS
      WRITE(8,*)'TOTAL SLUSH GENERATED THIS DAY SEG 1 TO 21 = ',SLS121
      WRITE(8,*)'TOTAL SEASON SLUSH GENERATED TO DATE = ',YRSLST
      WRITE(8,*)'RUNNING NDAY TOTAL SLUSH PRODUCED IN SEG 1 TO 21'
      WRITE(8,825)'DAY',(I,I=1,10)
  825 FORMAT(' ',A3,10(1X,3X,I2,3X))
      WRITE(8,830)((TOT121(I)),I=1,10)
  830 FORMAT(' ',3X,10(1X,F8.0))
      WRITE(8,*)'FRONT OF ICE FIELD TODAY IS AT RIVER SECTION',ICE_LOC
      WRITE(8,*)
      WRITE(8,*)
      IF(PLTCNT.EQ.3)THEN
          NPLT=NPLT+1
          PLTVOL(NPLT)=ICE_LOC
          PLTDAY(NPLT)=CUR_DAY
          PLTMON(NPLT)=CUR_MON
          PLTYR(NPLT)=CUR_YR
          PLTCNT=1
      ELSE
          PLTCNT=PLTCNT+1
      END IF
      RETURN
      END
C*************************************************************

      SUBROUTINE FINAL_OUTPUT
      INCLUDE 'NEWICE.CMN'
      CALL PRINT_PLOT
      RETURN
      END
C*************************************************************

      SUBROUTINE SEASON_REPT
      INCLUDE 'NEWICE.CMN'
      WRITE(8,*)'TOTAL DEGREE DAYS FOR SEASON = ',DEGSUM
      WRITE(8,*)'SEASON ENDING ',CUR_YR,CUR_MON,
     &' HAD ',SEASON_DAY,' DAYS'
      WRITE(8,*)
      WRITE(8,*)
      RETURN
      END
C*************************************************************

      SUBROUTINE INIT_SEASON
      INCLUDE 'NEWICE.CMN'
```

```
      INTEGER  I
      DO I=1,NCCT
         CCT(I)=ICCT
         COVER(I)=ICOVER
         CCD(I)=ICCD(I)
         CCA(I)=ICCA(I)
         CCX(I)=ICCX(I)
         CCDELA(I)=ICCDELA(I)
      END DO
      ICE_LOC=NCCT
      YRSLST=0.0
      DAYSLS=0.0
      SLS121=0.0
      BLOCK_FLAG=.FALSE.
      ACCUMULATE_FLAG=.FALSE.
      BDRICE=0.0
      BDRMAX=(0.5*CCA(NCCT))/2500.
      DEGSUM=0.0
      DO I=1,10
         TOT121(I)=0.0
         OLD121(I)=0.0
      END DO
      RETURN
      END
C*****************************************************************

      SUBROUTINE CALC_SOLAR
      INCLUDE 'NEWICE.CMN'
      REAL B/47.15/
     &    ,C/1.74/
     &    ,SSRATIO,A,SSMAX,CC,SRHO
      INTEGER  DAYJUL
     &        ,DAY_NUM
      DAYJUL=DAY_NUM(CUR_YR,CUR_MON,CUR_DAY)-DAY_NUM(CUR_YR,1,1)+1
      IF(CUR_MON.EQ.12)THEN
         A=26.39
         SSMAX=8.32
      ELSE IF(CUR_MON.EQ.1)THEN
         A=29.46
         SSMAX=8.77
      ELSE IF(CUR_MON.EQ.2.AND.CUR_DAY.LT.10)THEN
         A=39.61
         SSMAX=9.48
      ELSE IF(CUR_MON.EQ.2)THEN
         A=52.03
         SSMAX=10.2
      ELSE IF(CUR_MON.EQ.3)THEN
         A=52.03
         SSMAX=11.9
      ELSE
        !?
```

```
      END IF
      SSRATIO=SUN/SSMAX
      IF(SSRATIO.GT.1.0)ssratio=1.0
      CC=(1.0-SSRATIO)*10.0
      IF(CUR_MON.EQ.3)THEN
         SRHO=9.717+0.1265*CUR_DAY
      ELSE
         SRHO=(A-B*SIND(0.01717*DAYJUL+C))*(24.0/88.114)
      END IF
      SRO=(1.0-0.0071*CC*CC)*SRHO
      HS=SRO*87.978
      RETURN
      END
C*****************************************************************

      SUBROUTINE KEIDSR
      INCLUDE 'NEWICE.CMN'
      LOGICAL*1 DONE
      INTEGER I
      REAL HA,TSTAR,BETA,FW,ETP
      ET=TD
      FW=70.0+(.7*WS*WS)
      HA=3.1872E-8*(TAIR+460.)**4.
      I=0
      DONE=.FALSE.
      DO WHILE(.NOT.DONE)
         TSTAR=(ET+TD)/2.
         BETA=.255-(.0085*TSTAR)+(.000204*TSTAR*TSTAR)
         CSHE=15.7+(.26+BETA)*FW
         ETP=(HS+HA-1801.)/CSHE + (CSHE-15.7)*(.26*TAIR+BETA*TD) /
     &   (CSHE*(.26+BETA))
         IF(ABS(ETP-ET).LT..05)THEN
            DONE=.TRUE.
         ELSE IF(I.LT.50)THEN
            ET=ETP
             I=I+1
         ELSE
            WRITE(8,*)'*******************************',
     &EIDSR DOES NOT CONVERGE IN 50 ITERATIONS',CUR_YR,CUR_MON,
     &CUR_DAY
            DONE=.TRUE.
         END IF
      END DO
      ETC=5.*(ET-32.)/9.
      CSHEWM=.2364*CSHE
      CSHEMS=5.65E-8*CSHE
      return
      end
C*****************************************************************

 SUBROUTINE ICE_CALCS
```

```
      INCLUDE 'NEWICE.CMN'
      REAL CCV
      INTEGER I,J
      CCT(1)=TEXPLD
      DO J=1,24
         DO I=2,NCCT
            CCV=CCD(I)*CCA(I)
            CCT(I)=(CCV*CCT(I)+CCQ(I)*DT*CCT(I-1)+CSHEMS*CCA(I)*DT
     &      *ETC)/(CCV+CCQ(I)*DT+CSHEMS*CCA(I)*DT)
            IF(CCT(I).LT.-.01)THEN
               CCT(I)=-.01
            END IF
         END DO
      END DO
      IF(TA.LT.0.1)THEN
         IF(.NOT.BLOCK_FLAG)THEN
            CALL BLOCK_RIVER
         END IF
         IF(BLOCK_FLAG)THEN
            CALL ICE_PRODUCTION
            IF(.NOT.ACCUMULATE_FLAG)THEN
               IF(DAYSLS.GT.500000.)THEN
                  ACCUMULATE_FLAG=.TRUE.
                  WRITE(8,*)' *** RIVER CLOSURE AT BOOM ***'
               ELSE
                  BLOCK_FLAG=.FALSE.
               END IF
            END IF
            IF(ACCUMULATE_FLAG)THEN
               CALL ICE_ACCUMULATION
            END IF
         END IF
      ELSE
         SLS121=0.0
      END IF
      DO I=1,10
         TOT121(I)=TOT121(I)+SLS121-OLD121(I)
      END DO
      DO I=10,2,-1
         OLD121(I)=OLD121(I-1)
      END DO
      OLD121(1)=SLS121
      RETURN
      END
C*****************************************************************

      SUBROUTINE ICE_PRODUCTION
      INCLUDE 'NEWICE.CMN'
      INTEGER I
      REAL  TSUBF,VFRAZ,SEGSLS
      DAYSLS=0.0
```

```
      SLS121=0.0
      DO I=1,NCCT
         IF(CCT(I).LT.0.0)THEN
            TSUBF=0.0-TA
            VFRAZ=(CSHEWM*CCA(I)*TSUBF*TSTEP)/(900.*334000.)
            SEGSLS=(VFRAZ/E)*(1.0-COVER(I))
C           COVER(I)=0.0
            DAYSLS=DAYSLS+SEGSLS
            IF(I.LE.21)THEN
               SLS121=SLS121+SEGSLS
            END IF
         END IF
      END DO
      RETURN
      END
C*************************************************************

      SUBROUTINE ICE_ACCUMULATION
      INCLUDE 'NEWICE.CMN'
      INTEGER I
      REAL    CAP,TEMPSLS
      IF(DAYSLS.GT.0.0)THEN
         TEMPSLS=DAYSLS
         YRSLST=YRSLST+TEMPSLS
         DO WHILE(TEMPSLS.GT.0.0.AND.ICE_LOC.GT.1)
            CAP=(1.-COVER(ICE_LOC))*CCIV(ICE_LOC)
            IF(CAP.GT.TEMPSLS)THEN
               CAP=CAP-TEMPSLS
               COVER(ICE_LOC)=1.-CAP/CCIV(ICE_LOC)
               TEMPSLS=0.0
            ELSE
               TEMPSLS=TEMPSLS-CAP
               COVER(ICE_LOC)=1.0
               ICE_LOC=ICE_LOC-1
            END IF
         END DO
      END IF
      RETURN
      END
C*************************************************************

      SUBROUTINE FLUSH_RIVER
      INCLUDE 'NEWICE.CMN'
      INTEGER I
      WRITE(8,*)'HIGH FLOW - ICE FLUSHED FROM RIVER'
      ICE_LOC=NCCT
      YRSLST=0.0
      DAYSLS=0.0
      DO I=1,NCCT
         COVER(I)=0.0
         CCT(I)=2.0
```

```
      END DO
      BLOCK_FLAG=.FALSE.
      ACCUMULATE_FLAG=.FALSE.
      BDRICE=0.0
      DO I=1,10
         TOT121(I)=0.0
         OLD121(I)=0.0
      END DO
      RETURN
      END
C*****************************************************************

      SUBROUTINE BLOCK_RIVER
      INCLUDE 'NEWICE.CMN'
      REAL  BDRVEL,DBOOM,VBOOM,FROUDE
      IF(CUR_YR.GT.1974.OR.(CUR_YR.EQ.1974.AND.CUR_MON.GE.12))THEN
         BDRVEL=0.0012*QGF
      ELSE
         BDRVEL=0.0012*QGF+0.1
      END IF
      BDRICE=BDRICE+((0.54/BDRVEL**1.5)*DEGDY)*BDRADJ
      IF(BDRICE.GT.BDRMAX)THEN
         BLOCK_FLAG=.TRUE.
         ACCUMULATE_FLAG=.TRUE.
         WRITE(8,*)' *** RIVER CLOSURE DUE TO BORDER ICE ***'
      END IF
      IF(.NOT.BLOCK_FLAG)THEN
         IF(CUR_YR.GT.1974.OR.(CUR_YR.EQ.1974.AND.CUR_MON.GE.12))THEN
            DBOOM=1.79+0.00259*QGF
            VBOOM=0.000615*QGF
         ELSE
            DBOOM=0.00457*QGF
            VBOOM=0.33+0.000251*QGF
         END IF
         FROUDE=(VBOOM/SQRT(9.8*DBOOM))*FRDADJ
         IF(FROUDE.LE.0.08)THEN
            BLOCK_FLAG=.TRUE.
         END IF
      END IF
      RETURN
      END
C*****************************************************************

      SUBROUTINE PRINT_PLOT
      INCLUDE 'NEWICE.CMN'
      INTEGER I,J,K
      WRITE(8,805)
  805 FORMAT('1','ICE FRONT PLOT - THE LOCATION OF THE ICE FRONT EACH',
     &' DAY IS REPRESENTED BY THE RIGHT-MOST ASTERISK')
      WRITE(8,806)'SECTION #',(J,J=0,3)
  806 FORMAT(' ',A10,1X,4(I1,9X))
```

```
      WRITE(8,807)'SECTION #',((J,J=0,9),K=1,4)
  807 FORMAT(' ',A10,1X,50I1)
      DO I=1,NPLT
         WRITE(8,810)PLTDAY(I),PLTMON(I),PLTYR(I),('*',J=1,PLTVOL(I))
  810    FORMAT(' ',I2,1X,I2,1X,I4,2X,120(A))
      END DO
      RETURN
      END
C*************************************************************

      INTEGER FUNCTION DAY_IN_MON(YEAR,MONTH)
      INTEGER YEAR,MONTH,NDAY(12)/31,27,31,30,31,30,31,31,30,31,30,31
      IF(MONTH.EQ.2)THEN
        IF(MOD(YEAR,4).EQ..0)THEN
           DAY_IN_MON=29
        ELSE
           DAY_IN_MON=28
        ENDIF
       ELSE
          DAY_IN_MON=NDAY(MONTH)
      ENDIF
      RETURN
      END
C*************************************************************
      INTEGER FUNCTION DAY_NUM(IYEAR,IMONTH,IDAY)
      INTEGER IYEAR,IMONTH,IDAY,BASEYR,BASEMT,BASEDY
      INTEGER DAY_IN_MON,IDIFF,NDAYS,IDX,LASTYR,JMONTH
      BASEYR=1900
      BASEMT=1
      BASEDY=1
      IDIFF=IYEAR-BASEYR
      NDAYS=IDIFF*365
      LASTYR=IYEAR-1
      DO 173 IDX=BASEYR,LASTYR
          IF(MOD(IDX,4).EQ.0)NDAYS=NDAYS+1
173   CONTINUE
      JMONTH=IMONTH-1
      DO 174 IDX=1,JMONTH
           NDAYS=NDAYS+DAY_IN_MON(IYEAR,IDX)
174   CONTINUE
      NDAYS=NDAYS+IDAY
      DAY_NUM=NDAYS
      RETURN
      END
```

```
1963   12   1964   2
12   2
540   1.6   1   .4   9.99   1
32   4   0
 1   1.5      490000              .1    81            60
 2   1.5      310000              .1    77.5         354
 3     1      350000              .1    75           363
 4   1.5      350000              .1    72.5         373
 5   2.5      325000              .1    70           383
 6   2.5      325000              .1    67.4         393
 7   2.5      335000              .1    65           446
 8   2.5      335000              .1    62.5         483
 9     2      515000              .1    60           493
10   1.5      515000              .1    57.5         502
11   1.5      490000              .1    55          1315
12   1.5      490000              .1    52.5        1369
13     1      620000       620000.0     50          1375
14   1.5      620000       620000.0     47.5        1383
15   1.5      500000       500000.0     45          1390
16     1      500000       500000.0     42.5        1396
17   1.5      515500       515500.0     40          1404
18   1.5      580000      1160000.0     37.5        1410
19     1      696000      1045000.0     35          1418
20   1.5      644000      1610000.0     32.5        1425
21     1      644000       773000.0     30          2339
22     1      637000      1456000.0     27.5        2403
23   1.8      683000      2400000.0     25          2438
24     2      618000      2236000.0     22.5        2448
25   1.5      786000      1965000.0     20          2569
26   1.5      760000      2020000.0     17.5        2620
27   1.5     1082000      1623000.0     15          2677
28     1     1418000      1918000.0     12.5        2766
29     1      773000      1773000.0     10          2778
30     2      871000      1871000.0      7.5        3302
31     4     1750000      1050000.0      5          3317
32     5      722000       515000.0      2.5        3368
```

```
 DATE (year/mon/day) : 1964 / 1 / 30   AIR  TEMPERATURE  : 12.5-
 ET = 9.8  .  ETC = -12.3  .  WCF = 1565.9

  CSHE   CSHEMS  CSHEWM   DEGDY  DEGSUM   QEXPL     QGF
 118.6     0.0    28.0    12.5   433.2   141.6   223.0

 COOLING CANAL RIVER SEGMENTS TEMPERATURE :

 4.00 3.76 3.50 3.24 3.01 2.78 2.54 2.31 1.96 1.62 1.31 1.00 0.62
 0.26 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01-
 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01-


 TOTAL SLUSH GENERATED THIS DAY :   472931.3
 TOTAL SLUSH GENERATED THIS DAY SEGS 1 TO 21 : 472931.3
 TOTAL SLUSH GENERATED TO DATE  :1.969012E+07

 RUNNING NDAY TOTAL SLUSH PRODUCED IN SEGS 1 TO 21

 DAY :  1          2          3          4          5
 472931.3    740917.3    836220.8    836220.5    836220.0

 DAY :  6          7          8          9         10
 836220.5    836221.0    836221.0    836221.0    836221.0

 FRONT OF ICE FIELD TODAY IS AT RIVER SECTION #     20

 DATE (year/mon/day) : 1964 / 2 / 10   AIR  TEMPERATURE  :10.3-
 ET = 11.7  .  ETC = 11.3-  .  WCF = 1682.3

  CSHE   CSHEMS  CSHEWM   DEGDY  DEGSUM   QEXPL     QGF
 309.3     0.0    73.1    10.3   512.8   130.8   159.0

 COOLING CANAL RIVER SEGMENTS TEMPERATURE :

 4.00 3.39 2.74 2.11 1.55 1.02 0.49 0.01- 0.01- 0.01- 0.01- 0.01-
 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01-
 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01-

 TOTAL SLUSH GENERATED THIS DAY :  1858788
 TOTAL SLUSH GENERATED THIS DAY SEGS 1 TO 21 :   1858788
 TOTAL SLUSH GENERATED TO DATE  :2.183712E+07

 RUNNING NDAY TOTAL SLUSH PRODUCED IN SEGS 1 TO 21

DAY :  1          2          3          4          5
%1858787.9    %2146997.2    %2146997.2    %2146997.0    %2146996.5
```

```
   DAY :  6            7            8            9           10
   %2146997.0    %2146997.5    %2146997.5    %2146997.5    %2146997.5

   FRONT OF ICE FIELD TODAY IS AT RIVER SECTION #     19

   ---------------------------------------------------------------

   DATE (year/mon/day) : 1964 / 2 / 11   AIR  TEMPERATURE  :  11.7-
   ET = 10.8  .  ETC = 11.8-  .  WCF = 1697.9

 CSHE  CSHEMS  CSHEWM   DEGDY  DEGSUM   QEXPL     QGF
246.2     0.0    58.2    11.7   524.5   150.1   160.0

 COOLING CANAL RIVER SEGMENTS TEMPERATURE :
 4.00 3.56 3.08 2.61 2.19 1.78 1.37 0.97 0.39 0.01- 0.01- 0.01- 0.01-
 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01-
 0.01- 0.01- 0.01- 0.01- 0.01- 0.01- 0.01-


 TOTAL SLUSH GENERATED THIS DAY :    1249990
 TOTAL SLUSH GENERATED THIS DAY SEGS 1 TO 21 :    1249990
 TOTAL SLUSH GENERATED TO DATE  :2.308711E+07

 RUNNING NDAY TOTAL SLUSH PRODUCED IN SEGS 1 TO 21

 DAY :  1            2            3            4            5
 %1249990.4    %3108778.0    %3396987.5    %3396987.5    %3396987.0

 DAY :  6            7            8            9           10
 %3396987.5    %3396988.0    %3396988.0    %3396988.0    %3396988.0

 FRONT OF ICE FIELD TODAY IS AT RIVER SECTION #     18


 ---------------------------------------------------------------

 DATE (year/mon/day)  :   1964 / 2 / 29     AIR  TEMPERATURE  :    13.7-
 ET = 13.0  .  ETC = 10.6-  .  WCF = 1403.0

 CSHE  CSHEMS  CSHEWM   DEGDY  DEGSUM   QEXPL     QGF
 68.9     0.0    16.3    13.7   632.9   160.0   159.0

 COOLING CANAL RIVER SEGMENTS TEMPERATURE :
 4.00 3.89 3.77 3.65 3.54 3.43 3.31 3.20 3.02 2.85 2.68 2.52 2.31 2.08
 1.89 1.72 1.51 1.26 1.00 0.70 0.45 0.21 0.00 0.01- 0.01- 0.01- 0.01-
 0.01- 0.01- 0.01- 0.01- 0.01-

 TOTAL SLUSH GENERATED THIS DAY : 0
 TOTAL SLUSH GENERATED THIS DAY SEGS 1 TO 21 : 0
```

TOTAL SLUSH GENERATED TO DATE : 2.460991E+07

RUNNING NDAY TOTAL SLUSH PRODUCED IN SEGS 1 TO 21

DAY	1	2	3	4	5
	0.0	351454.6	351454.6	351454.6	351454.1
DAY	6	7	8	9	10
	351454.4	351454.1	351454.8	351454.8	351454.8

FRONT OF ICE FIELD TODAY IS AT RIVER SECTION # 15

TOTAL DEGREE DAYS FOR SEASON 632.9002
SEASON ENDING 1964 2 HAD 91 DAYS

'1' ICE FRONT PLOT - THE LOCATION OF THE ICE FRONT EACH DAY IS REPRESENTED BY THE RIGHT MOST ASTERISK SECTION #

```
 4   12   1963 **********************************
 7   12   1963 **********************************
10   12   1963 **********************************
13   12   1963 **********************************
16   12   1963 **********************************
19   12   1963 **********************************
22   12   1963 **********************************
25   12   1963 **********************************
28   12   1963 **********************************
31   12   1963 **********************************
 3    1   1964 ********************************
 6    1   1964 ********************************
 9    1   1964 ***************************
12    1   1964 ************************
15    1   1964 ************************
18    1   1964 **********************
21    1   1964 **********************
24    1   1964 **********************
27    1   1964 **********************
30    1   1964 **********************
 2    2   1964 **********************
 5    2   1964 **********************
 8    2   1964 **********************
11    2   1964 ******************
14    2   1964 ******************
17    2   1964 ****************
20    2   1964 ****************
23    2   1964 ****************
26    2   1964 ****************
29    2   1964 ***************
```

```
1963  12  1964  2
12  2
540  1.6  1  .4  9.99  1
32  4  0
 1  1.5    490000            .1   81          60
 2  1.5    310000            .1   77.5       354
 3    1    350000            .1   75         363
 4  1.5    350000            .1   72.5       373
 5  2.5    325000            .1   70         383
 6  2.5    325000            .1   67.4       393
 7  2.5    335000            .1   65         446
 8  2.5    335000            .1   62.5       483
 9    2    515000            .1   60         493
10  1.5    515000            .1   57.5       502
11  1.5    490000            .1   55        1315
12  1.5    490000            .1   52.5      1369
13    1    620000       620000.0  50        1375
14  1.5    620000       620000.0  47.5      1383
15  1.5    500000       500000.0  45        1390
16    1    500000       500000.0  42.5      1396
17  1.5    515500       515500.0  40        1404
18  1.5    580000      1160000.0  37.5      1410
19    1    696000      1045000.0  35        1418
20  1.5    644000      1610000.0  32.5      1425
21    1    644000       773000.0  30        2339
22    1    637000      1456000.0  27.5      2403
23  1.8    683000      2400000.0  25        2438
24    2    618000      2236000.0  22.5      2448
25  1.5    786000      1965000.0  20        2569
26  1.5    760000      2020000.0  17.5      2620
27  1.5   1082000      1623000.0  15        2677
28    1   1418000      1918000.0  12.5      2766
29    1    773000      1773000.0  10        2778
30    2    871000      1871000.0   7.5      3302
31    4   1750000      1050000.0   5        3317
32    5    722000       515000.0   2.5      3368
```

```
YEAR MO DA      TA      ET     ETC     WCF    CSHE CSHEMS CSHEWM  DEGDY DEGSUM  QEXPL     QGF
1964  1 30   -12 5     9 8   -12.3 1565.9   118.6     0.0    28.0    12.5  433 2   141.6   223.0
COOLING CANAL/RIVER SEGMENT TEMPERATURES
   1.00   3.76   3.50   3 21   3 01   2.78   2.54   2.31   1.96   1.62   1.31   1 00   0.62   0.26 -0.01 -0 01
  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01
TOTAL SLUSH GENERATED THIS DAY =     472931.2
TOTAL SLUSH GENERATED THIS DAY SEG 1 TO 21 =     472931.2
TOTAL SEASON SLUSH GENERATED TO DATE =    1.9690122E+07
RUNNING NDAY TOTAL SLUSH PRODUCED IN SEG 1 TO 21
DAY      1         2         3         4         5         6         7         8         9        10
     472931.   740918.   836221.   836220.   836220.   836221.   836221.   836221.   836221.   836221.
FRONT OF ICE FIELD TODAY IS AT RIVER SECTION          20


YEAR MO DA      TA      ET     ETC     WCF    CSHE CSHEMS CSHEWM  DEGDY DEGSUM  QEXPL     QGF
1964  2 10   -10.3    11.7   -11.3 1682.3   309.3     0 0    73.1    10.3  512 8   130.8   159.0
COOLING CANAL/RIVER SEGMENT TEMPERATURES
   1 00   3.39   2.74   2 11   1.55   1.02   0 49  -0.01  -0 01  -0 01  -0 01  -0 01  -0.01  -0.01  -0 01  -0.01
  -0 01  -0.01  -0 01  -0.01  -0 01  -0.01  -0.01  -0.01  -0 01  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01  -0.01
TOTAL SLUSH GENERATED THIS DAY =     1958788
TOTAL SLUSH GENERATED THIS DAY SEG 1 TO 21 =     1958788
TOTAL SEASON SLUSH GENERATED TO DATE =    2.1837129E+07
RUNNING NDAY TOTAL SLUSH PRODUCED IN SEG 1 TO 21
DAY      1         2         3         4         5         6         7         8         9        10
     1958788.  2146998.  2146998.  2146997.  2146997.  2146998.  2146998.  2146998.  2146998.  2146998.
FRONT OF ICE FIELD TODAY IS AT RIVER SECTION          19
```

```
YEAR MO DA      TA      ET     ETC     WCF    CSHE CSHEMS CSHENM  DEGDY DEGSUM  QEXFL     QGF
1964  2 11   -11.7    10.8  -11.8 1697.9   246.2     0.0    58.2   11.7   524.5   150.1   160.0
COOLING CANAL/RIVER SEGMENT TEMPERATURES
  4.00  3.56  3.08  2.61  2.19  1.78  1.37  0.97  0.39 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01
 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01
TOTAL SLUSH GENERATED THIS DAY =     1249990.
TOTAL SLUSH GENERATED THIS DAY SEG 1 TO 21 =     1249990.
TOTAL SEASON SLUSH GENERATED TO DATE =    2.3067110E+07
RUNNING NDAY TOTAL SLUSH PRODUCED IN SEG 1 TO 21
DAY      1         2         3         4         5         6         7         8         9        10
     1249991. 3108779. 3396988. 3396987. 3396987. 3396988. 3396988. 3396988. 3396988. 3396989.
FRONT OF ICE FIELD TODAY IS AT RIVER SECTION           18


YEAR MO DA      TA      ET     ETC     WCF    CSHE CSHEMS CSHENM  DEGDY DEGSUM  QEXFL     QGF
1964  2 29   -13.7    13.0  -10.6 1403.0    68.9     0.0    16.3   13.7   632.9   160.0   159.0
COOLING CANAL/RIVER SEGMENT TEMPERATURES
  4.00  3.89  3.77  3.65  3.54  3.43  3.31  3.20  3.02  2.85  2.68  2.52  2.31  2.08  1.89  1.72
  1.51  1.26  1.00  0.70  0.45  0.21  0.00 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01
TOTAL SLUSH GENERATED THIS DAY =    0.0000000E+00
TOTAL SLUSH GENERATED THIS DAY SEG 1 TO 21 =    0.0000000E+00
TOTAL SEASON SLUSH GENERATED TO DATE =    2.4609910E+07
RUNNING NDAY TOTAL SLUSH PRODUCED IN SEG 1 TO 21
DAY      1         2         3         4         5         6         7         8         9        10
         0.   351455.   351455.   351455.   351455.   351455.   351455.   351455.   351456.   351456.
FRONT OF ICE FIELD TODAY IS AT RIVER SECTION           15


TOTAL DEGREE DAYS FOR SEASON =     632.9001
SEASON ENDING          1964          2 MAD          91 DAYS
```

ICE FRONT PLOT - THE LOCATION OF THE ICE FRONT EACH DAY IS REPRESENTED BY THE RIGHT-MOST ASTERISK

```
 SECTION #  0          1          2          3
 SECTION #  0123456789012345678901234567890123456789
  1  12 1963  **********************************
  4  12 1963  **********************************
  7  12 1963  **********************************
 10  12 1963  **********************************
 13  12 1963  **********************************
 16  12 1963  **********************************
 19  12 1963  **********************************
 22  12 1963  **********************************
 25  12 1963  **********************************
 28  12 1963  **********************************
 31  12 1963  **********************************
  3   1 1964  **********************************
  6   1 1964  **********************************
  9   1 1964  **********************************
 12   1 1964  **********************************
 15   1 1964  **********************************
 18   1 1964  **********************************
 21   1 1964  **********************************
 24   1 1964  **********************************
 27   1 1964  **********************************
 30   1 1964  **********************************
  2   2 1964  **********************************
  5   2 1964  **********************************
  8   2 1964  **********************************
 11   2 1964  **********************************
 14   2 1964  **********************************
 17   2 1964  **********************************
 20   2 1964  **********************************
 23   2 1964  **********************************
 26   2 1964  **********************************
 29   2 1964  **********************************
```

INICE - DATA FILE HANDLING PROGRAM
HYDROLOGIC MODELLING BRANCH
DEPARTMENT OF ENVIRONMENT
DEVELOPED BY L.J. POMAR

```
10 KEY OFF:CLS:NEWFILE=0:TOP=0
20 DIM VAR$(20),VAR(20),MET$(20),MET(20)
30 FOR I=1 TO 15:READ VAR$(I):NEXT:FOR I=1 TO 9:READ MET$(I):NEXT
40 LOCATE 5,5:PRINT "DO YOU WISH TO CHANGE SEASON VARIABLES Y/N ?";
   X=52:YC=5
50 GOSUB 1000
60 IF(FLG=1)THEN NEWFILE=1:TOP=1:IDX=15 ELSE NEWFILE=0:TOP=0:
   IDX=9:GOTO 100
70 GOSUB 210
80 IF(CORRECT=0)THEN GOSUB 370
90 IF(TOP=1)THEN GOSUB 530
100 DONE=0
110 IF(NEWFILE=1)THEN  OPEN "A:ICEBOT" FOR OUTPUT AS #2
120 WHILE(DONE =0)
130     GOSUB 210:IF(CORRECT=0)THEN GOSUB 370
140     GOSUB 630
150     LOCATE 25,50:PRINT "MORE Y/N ?";:YC=25:X=64:GOSUB 1000
160     IF(FLG=0)THEN DONE=1
170 WEND
180 CLOSE 2
190 IF(NEWFILE=1)THEN GOSUB 660
200 CLS:SYSTEM
210 REM MAIN SUBROUTINE
220 CLS:IF(TOP=1)THEN TITLE$="SEASON DATA" ELSE TITLE$=
    "METEOROLOGICAL DATA"
230 LOCATE 3,25:PRINT TITLE$;
240 YY=7
250 FOR Y=7 TO IDX+7 STEP 2
260      IF(TOP=1)THEN V$=VAR$(Y-6) ELSE V$=MET$(Y-6)
270      LOCATE YY,1:PRINT Y-6;") ";:LOCATE YY,7:PRINT V$;" ";:
280      IF(TOP=1)THEN VAR(Y-6)=V:V$=VAR$(Y-5) ELSE MET(Y-6)=V
         :V$=MET$(Y-5)
290      IF(Y>=IDX+7-1)THEN 320
300      LOCATE YY,40:PRINT Y-5;") ";:LOCATE YY,45:PRINT V$;" ";
310      IF(TOP=1)THEN VAR(Y-5)=V ELSE MET(Y-5)=V
320      YY=YY+1
330 NEXT
340 LOCATE YY+2,1:PRINT "IS ALL DATA ENTERED CORRECTLY Y/N ?";:X=38:
    YC=YY+2:GOSUB 1000
350 IF(FLG=1)THEN  CORRECT=1 ELSE CORRECT=0
360 RETURN
370 REM subroutine to correct data
380 LOCATE YY+4,1:PRINT "TO END DATA CORRECTION, TYPE  ";:COLOR 0,7:
```

```
    PRINT "QUIT";:COLOR 7,0:PRINT;
 390 LOCATE YY+6,1:PRINT "ENTER NUMBER ";:INPUT N$
 400 IF (ASC(LEFT$(N$,1))>=58)OR(ASC(LEFT$(N$,1))<=47)THEN RETURN
 405 IF(VAL(N$)>IDX)THEN SOUND 3000,1:LOCATE 25,2:PRINT
     "ILLEGAL NUMBER";:GOTO 390
 410 IF TOP=1 THEN GOSUB 450 ELSE GOSUB 490
 420 GOTO 390
 430 RETURN
 440 REM SUBROUTINE SCAN1
 450 REM SUB SCAN1
 460 LOCATE YY+6,40:PRINT "ENTER NEW VALUE : ";:INPUT VAR(VAL(N$))
 470 LOCATE YY+6,18:PRINT "                                                    ";
 480 RETURN
 490 REM SUBROUTINE SCAN2
 500 LOCATE YY+6,40:PRINT "ENTER NEW VALUE : ";:INPUT MET(VAL(N$))
 510 LOCATE YY+6,18:PRINT "                                                    ";
 520 RETURN
 530 REM SUBROUTINE TO CREATE TOP PORTION OF FILE
 540 OPEN "A:ICETOP" FOR OUTPUT AS #2
 550 PRINT #2,USING "#### ";VAR(1);:PRINT #2,USING "## ";VAR(2);
     PRINT #2,USING "####";VAR(3):PRINT #2,USING "##";VAR(4)
 560 PRINT #2,USING "## ";VAR(5);VAR(6)
 570 PRINT #2,USING "###.#";VAR(7);:PRINT #2,USING "##.##";VAR(8);
     VAR(9);VAR(10);VAR(11);VAR(12);
 580 PRINT #2,USING "###.#";VAR(13);:PRINT #2,USING "##.##";VAR(14)
     VAR(15);'
 590 CLOSE 2:TOP=0:IDX=9
600 RETURN
610 DATA "START YEAR","START MONTH","END YEAR","END MONTH",
    "START SEASON",END SEASON","QFLUSH","PORADJ","TEXADJ","BDRADJ"
    "FRADADJ","VOLADJ","NCCT","ICCT",ICOVER"
620 DATA "YEAR","MONTH","DAY","TA","TDEW","WIND","SUN","QE","QGFALL"
630 REM SUBROUTINE TO CREATE BOTTOM PART OF FILE
640 PRINT #2,USING "#### ";MET(1);:PRINT #2,USING "## ";MET(2);MET(3);:PRINT
    #2,USING "###.#- ";MET(7);:PRINT #2,USING "#####.# ";MET(8);:
    PRINT #2,USING "####.#";MET(9);
650 RETURN
660 OPEN "B:ICE.DAT" FOR OUTPUT AS #2
670 OPEN "A:ICETOP"  FOR INPUT AS #1
680 FOR I=1 TO 15
690    INPUT #1,VAR(I)
700 NEXT
710 GOSUB 550:CLOSE 1:CLOSE 2
720 OPEN "b:ice.dat" FOR APPEND AS #2
730 OPEN "A:icemid" FOR INPUT AS #1
740 IF EOF(1) THEN 780
750     INPUT #1,VAL1,VAL2,VAL3,VAL4,VAL5,VAL6
760     PRINT #2,USING "## ";VAL1;:PRINT #2,USING "#.# ";VAL2;:
        PRINT #2,USING "########.#";VAL(3):PRINT #4,USING "####.##";
        ";VAL4;:PRINT #2,USING "##.# ";VAL5;:PRINT #2,USING "####.# "
        ;VAL(6)
770 GOTO 740
780 CLOSE 1:CLOSE 2
790 OPEN "B:ICE.DAT" FOR APPEND AS #2
```

```
800 OPEN "A:ICEBOT" FOR INPUT AS #1
810 IF EOF(1) THEN 860
820   FOR I=1 TO 9
830      INPUT #1,MET(I)
840   NEXT
850 GOSUB 630:GOTO 810
860 CLOSE 1:CLOSE 2
870 KILL "A:icetop":KILL "A:icebot"
880 RETURN
1000 REM KEY POLL
1010 LOCATE YC,X:COLOR 22,0:PRINT CHR$(95);:COLOR 7,0:PRINT;
1011 K$=INKEY$:IF K$="" THEN 1011
1020 SOUND 3000,1
1030 IF(K$="Y")OR(K$="y")THEN FLG=1 ELSE IF(K$="N")OR(K$="n")THEN
     FLG=0  ELSE GOTO 1011
1035 LOCATE YC,X:PRINT K$
1040 RETURN
```