CENTRO AGRONOMICO TROPICAL DE
INVESTIGACION Y ENSEÑANZA

PROGRAMA MANEJO INTEGRADO DE
RECURSOS NATURALES

AREA DE MANEJO DE CUENCAS

PUBLICACIONES DEL PROYECTO RENARM/MANEJO DE CUENCAS

UNA INTERFACE FORTRAN ENTRE EL SIG IDRISI Y EL MODELO HIDROSEDIMENTOLOGICO DISTRIBUIDO ANSWERS

Preparado por:
Gregorio Leclerc

CATIE, TURRIALBA
1994

UNA INTERFACE FORTRAN ENTRE EL SIG IDRISI
Y EL MODELO HIDROSEDIMENTOLOGICO DISTRIBUIDO ANSWERS

Elaborado por Dr. Gregorio Leclerc
Consultor Internacional

Unidad de Análisis de Información Geográfico
Laboratorio de SIG y Teledetección
Proyecto RENARM-CUENCAS, CATIE

15 de diciembre de 1994

Antecedentes.

La necesidad de determinar de manera más exacta las tasas de erosión en las cuencas hidrográficas de los países centroamericanos nos llevó a considerar los modelos hidrosedimentológicos distribuidos tales como AGNPS o ANSWERS. ANSWERS (Beasley, 1989) fue escogido por su capacidad de simular a la escala del evento los procesos de erosión-deposición y escorrentía, y calcular las concentraciones de sedimentos y el caudal en un punto dado de una cuenca. Estos modelos demandan un enorme trabajo de entrada de datos, ya que para cada celda en que se partió la cuenca se tiene que entrar los datos correspondientes (suelos, pendiente, lluvia, etc...). La construcción de una interface entre un SIG de formato cuadrícula y el modelo ANSWERS consiste en usar el SIG para generar los mapas que representan los diversos parámetros que require el modelo, la interface siendo usada para transferir estos datos en un formato compatible con el modelo. En este caso, la interface permite generar, a partir de los mapas de entrada, el archivo de entrada del modelo. Además, la interface puede leer el archivo de salida del modelo y generar mapas de erosión-deposición, y mapas de deposición en los canales (o ríos). Esta interface ha sido aplicada a la cuenca del río Pejibaye, Costa Rica, en apoyo a la tesis de M. Sc. del Ingeniero Rolando Portilla Pastor (CATIE, 1995). Los programas se van a modificar en el futuro, y se hará una actualización de este manual periódicamente.

Descripción.

Esta interface fue escrita en FORTRAN Microsoft 5.1, por lo que se necesitó adaptar al FORTRAN los módulos que permiten escribir y leer los archivos imagen de IDRISI. El código fuente de estos módulos existe solamente en PASCAL y BASIC, y fue adaptado al FORTRAN en 1992-1994. Las subrutinas se encuentran en NEWMOD4.FOR, y usan subrutinas de utilitarios MESUTILS.FOR. Los programas READANSW Y WRITANS leen el archivo IDDRISI.ENV para localizar las imagens y los archivos. Por lo tanto estos programas estan ubicados en el directorio IDRISI. ANS77W tiene que estar en el mismo directorio que el directorio de datos.

READANSW.FOR: Usa los archivos INCLUDE: IDRIVAR.TXT, ANSWVARW.TXT.
Este programa, que corre bajo ambiente WINDOWS, permite leer el archivo de entrada y de salida de ANSWERS, y crear imágenes en formato IDRISI.

WRITANS.FOR: Usa los archivos INCLUDE: IDRIVAR.TXT, ANSWVAR.TXT.
Este programa permite escribir el archivo de entrada de ANSWERS, a partir de imágenes IDRISI cuyas nombres están en el archivo inicial PINIT.ANS. PINIT.ANS, a la diferencia de PINITW.ANS, contiene un campo "elem_number" escribir un archivo con el número secuencial de cada elemento y su ubicación fila/columna, y facilitar la localización de celdas con problemas.

Notas:

a)El archivo de suelos es utilizado para determinar el área de la cuenca: el fondo tiene que ser 0 ya que se hace un test con este valor de fondo.

b)Todas las imágenes de pinit.ans o pinitw.ans tienen que existir, aunque pueden tener un valor de 0.
c)Todas las imágenes son de tipo byte, a exepción de las de pendiente y de pendiente de canales, y aspecto, que son de tipo entero. No importa si son binarias o ascii.
d)Los archivos de salida son de tipo ascii, se actualiza el titulo con respecto al nombre de la cuenca.
e) Se puede cambiar el numero máximo de elementos cambiando el valor de 45000 en los archivos *.TXT y READANSW.FOR y ANS77W.FOR (45000 elementos corresponde a una máquina con 4MB de RAM. Usar los comandos de búsqueda del procesador de texto). Se puede cambiar el intervalo de tiempo DT buscando "DT=" con el procesador de texto, dentro del archivo ANS77W.FOR.

Problemas conocidos:
-Se pierde la georeferenciación al leer los archivos ANSWERS, las unidades son PIXEL.
-Si el contorno de la cuenca no coincide con los límites de la imagen, la imagen creada por WRITANS no tendrá los mismos números de filas y columnas, pero será cortada al contorno de la cuenca.
-Algunos módulos de IDRISI no leen bien el archivo de documentación generado por WRITANS, en particular el campo "ref. units.". Se corrige con DOCUMENT.
-Una celda tipo canal con pendiente=0 genera un mensaje de error al correr ANS77W.
-Si una celda tipo canal esta conectada a la orilla de la cuenca, se tendrá otro error al correr ANS77W.

Utilización de los programas.

WRITANS, para escribir un archivo de entrada de ANSWERS a partir de una secuencia de imágenes.
uso:
este programa es de DOS, y se corre desde la linea de commando:

C:\ANSWERS> WRITANS [archivo initial] (sin extensión).
donde [archivo inicial] es equivalente a pinit.ans.
El programa pide el nombre de la cuenca (entrar 'cuenca XXX' , con las comillas), y luego pide la fila y columna de la salida de la cuenca (que corresponden las filas y columnas de IDRISI, más 1).

READANSW, para leer un archivo de entrada o de salida hasta 45000 elementos, dentro de WINDOWS.
uso:
-seleccionar el icono WRITANSW
-seleccionar "File, Properties"
-a la línea "Command Line" entrar:
READANSW [archivo inicial (sin extensión .ans)]
donde [archivo inicial] es equivalente a pinitw.ans
-'Working Directory" es el directorio C:\IDRISI> (donde se ubica IDRISI.ENV)
-Doble click sobre el icono para arrancar el proceso.

ANS77W: versión para WINDOWS de ANSWERS. Esta versión permite 45000 elementos (necesita 4MB de memoria), y se puede ver el progreso del cálculo en tiempo real. Nota: el número de elementos es el número de pixeles dentro de la cuenca + número de pixeles de canales. El intervalo de tiempo DT de modelación es 10 secundas.

uso:

-seleccionar el icono ans77w

-seleccionar "File Properties"

-a la línea "command line" poner:

ANS77W [archivo entrada (sin extensión .ans)] [archivo salida (sin extensión .ans)] donde [archivo entrada] es el archivo generado por writans (archivo en formato de entrada para ANSWERS), y [archivo salida] es el archivo generado por ANSWERS.

-"Working Directory" es el directorio donde estan los datos (donde se ubica ANS77W.EXE)

Compilación de los programas:

La compilación de los programas genera archivos .EXE que corren bajo ambiente WINDOWS o DOS.

(se tuvo que ambientar el FORTRAN con NEW-VARS.BAT si se apagó la máquina.)

```
fl /MW1 /W0 ans77w.for mesutils.for (/MW1: para WINDOWS)
fl /Gt /W0 writans.for mesutils.for newmod4.for (DOS)
fl /MW1 /W0 /Gt readansw.for mesutils.for newmod4.for (/MW1: para WINDOWS)
```

ANS77W.FOR

```
C************************************************************************
C************************************************************************
C************************************************************************
C*****                                                             ******
C*****                                                             ******
C*****                       A N S W E R S                         ******
C*****                                                             ******
C*****             (AREAL NONPOINT SOURCE WATERSHED                ******
C*****              ENVIRONMENT RESPONSE SIMULATION)               ******
C*****                                                             ******
C*****                                                             ******
C*****  AUTHORS:  D. B. BEASLEY, L. F. HUGGINS, AND J. R. BURNEY   ******
C*****                                                             ******
C*****   AGRICULTURAL ENGINEERING DEPARTMENT, PURDUE UNIVERSITY    ******
C*****               WEST LAFAYETTE, INDIANA  47907                ******
C*****                                                             ******
C*****                                                             ******
C************************************************************************
C************************************************************************
C************************************************************************
C*****                                                             ******
C*****       CURRENT MODIFICATION DATE:   15 FEBRUARY, 1988        ******
C*****                                                             ******
C*****    THE COMPONENTS CONTAINED IN THIS PROGRAM HAVE BEEN       ******
C*****    THOROUGHLY TESTED USING OBSERVED INFORMATION FROM        ******
C*****    BOTH PLOT AND WATERSHED RESEARCH AREAS.  SINCE THE       ******
C*****    MODEL IS STILL UNDERGOING DEVELOPMENT, THE ADDITION      ******
C*****    OR MODIFICATION OF COMPONENT RELATIONSHIPS SHOULD        ******
C*****    BE EXPECTED FROM ONE RELEASE TO ANOTHER.  BECAUSE OF     ******
C*****    THIS, SLIGHTLY DIFFERENT SIMULATION RESULTS MAY BE       ******
C*****    OBTAINED.  ALWAYS USE THE MOST CURRENT RELEASE!!         ******
C*****                                                             ******
C****************************************************** ****************
C************************************************************************
C*****                                                             ******
C*****      THIS VERSION OF ANSWERS CONTAINS:                      ******
C*****                                                             ******
C*****      1)  MEMORY CONSERVATION EQUIVALENCING                  ******
C*****      2)  3-PER-PASS ALGORITHM                               ******
C*****      3)  IMPROVED DATA VERIFICATION DIAGNOSTICS             ******
C*****      4)  STRUCTURAL PRACTICES                               ******
C*****      5)  MODIFIED INPUT FORMATS (SEE USER'S MANUAL)         ******
C*****      6)  MODIFIED DETACHMENT AND TRANSPORT RELATIONSHIPS    ******
C*****      7)  MODIFIED OUTPUT FORMATS                            ******
C*****                                                             ******
C*****      THE FOLLOWING ARE NEW WITH THIS RELEASE:               ******
C*****                                                             ******
C*****      8)  1977 ANSI STANDARD FORTRAN CODING                  ******
C*****          (SHOULD YOUR PARTICULAR COMPILER NOT BE ANSI-      ******
C*****          1977 COMPATIBLE, SIMPLY REMOVE THE CHARACTER       ******
C*****          DEFINITION STATEMENTS AT THE BEGINNING OF THE      ******
C*****          MAIN PROGRAM AND ALL SUBROUTINES)                  ******
C*****      9)  PROBLEM WITH MULTIPLE RAIN GAGES IN THE DATA       ******
C*****          SUBROUTINE INITIALIZATION ROUTINE CORRECTED        ******
C*****     10)  FLOATING UNDERFLOW PROBLEM CORRECTED               ******
C*****                                                             ******
C************************************************************************
C************************************************************************
C
C
C ****** DISTRIBUTED PARAMETER MATHEMATICAL MODEL OF A RAINFALL
C ****** EVENT ON A CATCHMENT, WITH EROSION AND DEPOSITION.
C
C **** MAXIMUM NUMBER OF SOIL TYPES IS 20.
C
      COMMON /CSOIL/ A(20),P(20),FC(20),GWC(20),SKDR(20)
C
C **** MAXIMUM NUMBER OF SURFACE AND CROP TYPES IS 20.
C
      COMMON /CROUGH/ ROUGH(20),HU(20),DIR(21),PIT(5,20),PER(20),CDR(20)
C
C **** MAXIMUM NUMBER OF RAINGAGES IS 4 WITH 35 VALUES PER GAGE.
C
      COMMON /CRGAGE/ RC(4,35),TC(4,35),R(4,20),FRA(4),JTR(4),RATE(4),SR
     1(4),NF(4)
C
C **** MAXIMUM NUMBER OF OVERLAND ELEMENTS PLUS CHANNEL ELEMENTS
C **** IS 42500 = NMAX.
C
C ****** IT IS EXPECTED THAT ARRAY "IEL" (IN SUBROUTINE DATA) WILL
```

```
C ****** BE OF SUCH A SIZE THAT IT WILL OVERLAY (BE EQUIVALENCED TO)
C ****** THE SPACE IN ARRAYS SI AND QI TOGETHER.  THEREFORE IT IS
C ****** NECESSARY THAT THESE TWO ARRAYS BE KEPT ADJACENT IN THEIR
C ****** COMMON BLOCK.  NOTE: THE ACTUAL NUMBER OF ELEMENTS THAT
C ****** CAN BE DIMENSIONED IN IEL WILL DEPEND ON THE WORD LENGTH
C ****** OF THE MACHINE BEING USED, E.G. ON A MACHINE WHICH USES
C ****** A SINGLE WORD INTEGER AND A DOUBLE WORD REAL, THE NUMBER
C ****** OF ELEMENTS IN IEL CAN BE FOUR TIMES THE NUMBER OF ELEMENTS
C ****** IN ARRAY SI.
C
      COMMON /CFLOW/ Q(42500),RFL(42500),FLINS(42500),SS(42500),
     &PIV(42500),B(42500),NR(42500),NC(42500),DR(42500),S(42500),
     &SL(42500),SEL(42500),SI(43000),QI(43000),DIN(42500),SST(42500)
C==================
      character*8 cmd_line_args(10), cmd_line_files(10)
      character*1 cmd_line_options(10)
      INTEGER*2 cmd_line_err
      INTEGER*4 cmd_line_args_num,cmd_line_options_num,
     &cmd_line_files_num
      character*12 entrada_file,salida_file
C=====================

C
C ****** ARRAYS SI AND QI MUST BE DIMENSIONED TO A SIZE = NMAX+ISTRUC+2
C ****** TO HOLD, IN ORDER, SEDIMENT AND FLOW FROM THE WATERSHED OUTLET
C ****** ELEMENT, STRUCTURAL PRACTICES AND ANY "LEAKY" ELEMENTS.
C
      EQUIVALENCE (FILTS(1),CWID(1))
      DIMENSION CWID(42500), FILTS(42500)
      EQUIVALENCE (TIAL(1),RANE(1)), (SUR(1),SOIL(1))
      COMMON /CSURF/ SUR(42500),RANE(42500)
      INTEGER SUR,SOIL(42500),TIAL(42500),RANE
C
C **** NUMBER OF PRINT AND PLOT POINTS IS 101 MAXIMUM.
C
      DIMENSION T(101), Q1(101), RW(101), SSI(101), SSCON(101)
      DIMENSION PP(14), QA(300), TT(20)
      CHARACTER*4 PP, TT
      DATA PP(1),PP(2),PP(3),PP(4),PP(5),PP(6),PP(7),PP(8),PP(9),PP(10),
     1PP(11),PP(12),PP(13),PP(14)/' IN.','/HR.',' AC.',' FT.',' LB.',
     2' PPM','/AC ','  MM','/H  ',' HA ',' M  ','  KG','MG/L','/HA '/
C
C **** NEW TRANSPORT AND DETACHMENT CONSTANTS.
C
C
C **** TRANSPORT COEFFICIENTS (CE1 AND CE2) ARE 10 PERCENT LARGER THAN
C **** THOSE USED IN THE ORIGINAL VERSION OF THE MODEL.  THESE CHANGES
C **** WERE PRECIPITATED BY ADDITIONAL INFORMATION FROM SEVERAL RAIN-
C **** FALL SIMULATOR PLOT STUDIES THAT INDICATED THAT YIELDS WERE ON
C **** THE ORDER OF 10 PERCENT HIGHER THAN PREDICTED WHEN THE WATER-
C **** SHED WAS IN A "TRANSPORT LIMITED" SITUATION.
C
C **** DETACHMENT COEFFICIENT CE3 (RAINFALL) WAS INCREASED BY A FACTOR
C **** OF 4 IN THE MARCH 15, 1982 VERSION OF ANSWERS.  THE REASON FOR
C **** THIS LARGE INCREASE WAS THAT A NUMBER OF THE RAINFALL SIMULATOR
C **** PLOTS THAT WERE USED IN COEFFICIENT CALCULATION HAD DEPOSITION
C **** AREAS.  HOWEVER, AFTER CLOSER EXAMINATION OF PHOTOGRAPHS AND
C **** SURVEY INFORMATION, THE DETACHMENT COEFFICIENT WAS DEEMED TO
C **** BE TOO HIGH.  THUS, THE CURRENT ACCEPTED VALUE OF CE3 IS TWICE
C **** THE ORIGINAL VALUE (GASP-IV VERSION OF ANSWERS).
C
C **** DETACHMENT COEFFICIENT CE4 (FLOW) WAS INCREASED BY A FACTOR OF
C **** 50 IN THE MARCH 15, 1982 VERSION OF ANSWERS.  THE REASON FOR
C **** THIS DRAMATIC INCREASE WAS SOME RAINFALL SIMULATOR DATA THAT
C **** SHOWED THE DIFFERENCE BETWEEN RAINFALL-ONLY AND RAINFALL PLUS
C **** UPSLOPE FLOW SEDIMENT YIELDS.  WHILE THE YIELDS INCREASED
C **** SUBSTANTIALLY WITH THE INCREASED FLOW, IT APPEARS THAT THE
C **** MAJOR SOURCE OF SEDIMENT WAS WASHOFF OF UNATTACHED PARTICLES,
C **** NOT DETACHMENT OF COHESIVE PARTICLES.  A RE-EXAMINATION OF THE
C **** FLOW DETACHMENT EQUATION HAS LED TO THE CONCLUSION THAT THE
C **** FLOW DETACHMENT COEFFICIENT SHOULD BE APPROXIMATELY 5 TIMES
C **** GREATER THAN THE ORIGINAL VALUE (NOT 50 TIMES).
C
C **** WHILE THE C AND K FACTORS IN THE USLE ARE USED TO DESCRIBE
C **** THE RELATIVE DEGREE OF ERODIBILITY OF A PARTICULAR SOIL IN
C **** THIS MODEL, THE IMPACTS OF SURFACE COMPACTION, ROUGHNESS,
C **** TEMPERATURE, ETC. ARE NOT TAKEN INTO ACCOUNT.  THUS, WHILE
C **** THE EROSION EQUATIONS WORK FOR THOSE SOIL SERIES FOR WHICH
C **** WE HAVE RAINFALL SIMULATOR AND WATERSHED DATA, THEY MAY NOT
C **** DO AN ADEQUATE JOB ON OTHER TYPES OF TOPOGRAPHY, SOIL TEXTURE,
C **** SURFACE CONDITION, ETC.  FOR THESE REASONS, CE3 AND CE4 SHOULD
```

```
C **** BE CONSIDERED AS POTENTIAL VARIABLES.  RESEARCH NOW BEING
C **** CONDUCTED SHOULD YIELD BETTER DESCRIPTIONS OF THE DETACHMENT
C **** PROCESS AND THE COEFFICIENTS ASSOCIATED WITH IT.  WHILE IT
C **** IS NOT POSSIBLE TO GIVE EXACT INSTRUCTIONS ON MODIFICATIONS
C **** THAT SHOULD BE MADE TO COEFFICIENTS WHEN SIMULATED AND OBSERVED
C **** RESULTS DON'T AGREE, WE WILL CERTAINLY BE WILLING TO DISCUSS
C **** THE PROBLEM AND MAKE SUGGESTIONS FOR LOGICAL MODEL MODIFICATIONS.
C
C **** TO EITHER MAKE SUGGESTIONS OR RECEIVE FURTHER INFORMATION, CONTACT:
C
C ****                    DAVID B. BEASLEY, PH.D., P.E.
C ****                    AGRICULTURAL ENGINEERING DEPARTMENT
C ****                    UNIVERSITY OF GEORGIA -- CPES
C ****                    TIFTON, GA  31793-0748
C ****                    PHONE: (912) 386-3377
C
      DATA CE1,CE2,CE3,CE4,CE5,CE6/5603.,4.26,62208.0,0.1,.00833333,62.3
     174/
C!!!!!!!!!!!!!!!MODIFICATION ENTREE: UNIT=1, SORTIE: UNIT=2
      call read_cmd_line_args(cmd_line_options_num,cmd_line_options,
     &cmd_line_files_num,cmd_line_files)
      IF (cmd_line_files_num .NE. 2) THEN
          WRITE(*,*) 'ERROR: se necesitan 2 archivos!'
          CALL HELP
          STOP
      END IF

      entrada_file(1:8)=cmd_line_files(1)
      entrada_file(9:12)='.ans'
      salida_file(1:8)=cmd_line_files(2)
      salida_file(9:12)='.ans'


C=================================
      OPEN (1, FILE=entrada_file)
      OPEN (2, FILE=salida_file)
      READ (1,280) (TT(I),I=1,19)
      WRITE (6,290) (TT(I),I=1,19)
      WRITE (2,290) (TT(I),I=1,19)
C
C **** READ, TRANSFORM AND RETURN INPUT INFORMATION.
C
      CALL DATA (NDT,KPR,N,CONV,CU,SF,IT,NN,ICR,NFI,CU2,ISTRUC,SB,TMIN,T
     1MAX,NRG,DX,GRF,NEXP,DC,PP,FILTS,CWID,AREA,DT,NMAX)
C
C **** COMPUTE THE PIECE-WISE LINEAR SEGMENTS FOR USE IN MANNING'S
C **** EQUATION.
C
      SC=((SF*CONV/SB)**.6)/300.
      D=0.
      DO 10 I=1,300
      QA(I)=D**1.66667
   10 D=D+SC
      SC=1./SC
C
C **** INITIALIZE VARIABLES.
C **** SET RAINFALL INITIAL VALUES.
C
      DO 20 I=1,NRG
      JTR(I)=1
      IF (TC(I,2).EQ.TMIN) JTR(I)=2
      SR(I)=0.
   20 NF(I)=NFI
      N1=N+1
      N2=NN-1
      CHN=N2-N
C
C **** EROSION CONSTANTS.
C
      IF (IT.LE.0) GO TO 30
C
C **** METRIC UNITS.
C
      CE1=9.66155E+5
      CE2=2.0847E+1
      CE3=3.26932E+6
      CE4=5.25545
      CE5=7.7419E-4
      CE6=1.E+3
C
C **** INITIALIZE VALUES.
```

```
C
   30 VOL=0.
      SSI(1)=0.
      SDR=0.
      CHDR=0.
      SSCON(1)=0.
      RW(1)=0.
      Q1(1)=0.
      RMAX=0.
      QMAX=0.
      CMAX=0.
      PREC=0.
      DTM=DT/60.
      T(1)=TMIN
C
C **** WRITE HYDROGRAPH HEADING AND INITIAL VALUE.
C
      WRITE (6,300) PP(IT+6),PP(IT+1),PP(IT+2),PP(IT+1),PP(IT+2),PP(IT+5
     1)
      WRITE (6,310) T(1),RW(1),Q1(1),SSI(1),SSCON(1)
      WRITE (2,300) PP(IT+6),PP(IT+1),PP(IT+2),PP(IT+1),PP(IT+2),PP(IT+5
     1)
      WRITE (2,310) T(1),RW(1),Q1(1),SSI(1),SSCON(1)
C
C **** START COMPUTATION FOR EACH HYDROGRAPH PRINT LINE AT DT*KPR.
C
      DO 220 L=2,NDT
      LM1=L-1
      T(L)=T(LM1)
C
C **** CONTINUITY EQUATION FOR TIME INCREMENTS DT.
C
      DO 170 J=1.KPR
      SPT=SI(NN)
      T(L)=T(L)+DTM
C
C **** CALCULATE NET RAINFALL FOR EACH GAGE AND SURFACE CONDITION AND
C **** UPDATE INFILTRATION CAPACITIES WITHIN GAGE AREA ON TIME OR NET
C **** RAINFALL CHANGE.
C
      DO 90 JJ=1,NRG
      NF(JJ)=NF(JJ)-1
      ITR=JTR(JJ)
      ITRM1=ITR-1
      IF (T(L)-TC(JJ,ITR)) 60,60,40
   40 IF (T(L)-TMAX) 50,230,230
C
C **** NEW RAINFALL RATE, ALLOW FOR DTM BRIDGING TC VALUE.
C
   50 DI=T(L)-TC(JJ,ITR)
      ITRP1=ITR+1
      RATE(JJ)=CU*(RC(JJ,ITRP1)*DI+RC(JJ,ITR)*(DTM-DI))/DTM
      JTR(JJ)=JTR(JJ)+1
      ITR=ITRP1
C
C **** ADD WHOLE HISTOGRAM BLOCK TO TOTAL PRECIPITATION IN
C **** PROPORTION TO WATERSHED AREA COVERED.
C
      PREC=PREC+RC(JJ,ITR)*(TC(JJ,ITR)-TC(JJ,ITR-1))*FRA(JJ)/60.
C
C **** CALCULATE NET RAINFALL FOR EACH COVER.
C
   60 DO 70 I=1,ICR
      R(JJ,I)=RAIN(RATE(JJ),PIT(JJ,I),PER(I))
      IF (R(JJ,I).EQ.SR(JJ).AND.NF(JJ).GT.0) GO TO 70
      SR(JJ)=R(JJ,I)
      NF(JJ)=-NFI
   70 CONTINUE
      RATE(JJ)=RC(JJ,ITR)*CU
      IF (NF(JJ).GT.0) GO TO 90
C
C **** CALCULATION OF INFILTRATION CAPACITY FOR EACH OVERLAND ELEMENT.
C
      DO 80 M=1,N
      IF (MOD(RANE(M),256).NE.JJ) GO TO 80
      K=MOD(SUR(M),256)
      KK=SOIL(M)/256
      FILTS(M)=FILT(A(KK),PIV(M),P(KK),FC(KK),GWC(KK),DR(M),S(M),R(JJ,K)
     1,CU2,ROUGH(K),HU(K),NEXP)
   80 CONTINUE
      NF(JJ)=NFI
   90 CONTINUE
C
C **** CONTINUITY EQUATION EXPLICIT SOLUTION FOR EACH ELEMENT DURING
C **** TIME INCREMENT, DT.
C
      DO 170 M=1,N2
      SSTOR=S(M)+SS(M)
      IF (SSTOR.LT.0.) SSTOR=0.
      IF (M.GT.N) GO TO 100
C
C **** OVERLAND ELEMENT.
C
      I=MOD(RANE(M),256)
      K=MOD(SUR(M),256)
      KK=SOIL(M)/256
      SUPP=.5*SSTOR+QI(M)+R(I,K)
      FIL=FILTS(M)
      IF (FIL.GT.SUPP) FIL=SUPP
      PIV(M)=PIV(M)+DR(M)-FIL
      SDR=SDR+DR(M)
      FLIN=QI(M)+R(I,K)-FIL
      GO TO 110
C
C **** CHANNEL ELEMENT.
C
  100 K=21
      FLIN=QI(M)+CHDR+DIN(M)
C
C **** COMBINE INITIAL INFLOW, OUTFLOW AND STORAGE WITH ACCUMULATED
C **** INFLOW.
C
  110 FHS=FLINS(M)+FLIN
      IF (SSTOR.GT.DIR(K)) GO TO 130
C
C **** NO RUNOFF FROM ELEMENT.
C
  120 S(M)=FHS
      SS(M)=0.
      FLINS(M)=FLIN+FHS
      IF (Q(M).EQ.0.) GO TO 170
      D=-Q(M)
      Q(M)=0.
      GO TO 150
C
C **** DIRECT SOLUTION OF CONTINUITY EQUATION BY LINEARIZATION.
C
  130 Y=SC*(SSTOR-DIR(K))
      IY=Y+1.
C!!!!!!!!IF (IY.LT.300) ORIGINALEMENT
      IF (IY.LT.300) GO TO 140
      WRITE (6,330) M
      WRITE (2,330) M
      STOP
  140 Y=IY-1
      QL=B(M)*QA(IY)
      QD=B(M)*(QA(IY+1)-QA(IY))
      SSTOR=(FHS-QL+QD*(Y+DIR(K)*SC))/(1.+QD*SC)
      IF (SSTOR.LE.DIR(K)) GO TO 120
      Q2=QL+QD*((SSTOR-DIR(K))*SC-Y)
      IF (Q2.LT.1.E-20) Q2=0.
      D=Q2-Q(M)
      Q(M)=Q2
      SS(M)=SSTOR-S(M)
      IF (SSTOR.LT.1.E-20) SSTOR=0.
      S(M)=SSTOR
      FLINS(M)=FLIN+SSTOR-Q2
C
C **** SEDIMENT CALCULATION.
C
  150 IF (M.LE.N) GO TO 160
C
C **** COMPUTE TRANSPORT/DEPOSITION FOR CHANNEL FLOW.
C
      CALL SED (CWID(M),DX,0.,SL(M),Q(M),1.,1.,SI(M),SEL(M),SE,S(M),M,N,
     1SST(M),CE1,CE2,CE3,CE4,CE5)
C
C **** REMEMBER ALL CHANNEL FLOW MOVES WITH ITS 'COLUMN' DESIGNATOR.
C
      K=NC(M)
      QI(K)=QI(K)+D
      SI(K)=SI(K)+SE
      GO TO 170
C
C **** COMPUTE TRANSPORT/DEPOSITION FOR OVERLAND FLOW.
C
  160 CALL SED (DX,DX,R(I,K),SL(M),Q(M),CDR(K),SKDR(KK),SI(M),SEL(M),SE,
     1S(M),M,N,SST(M),CE1,CE2,CE3,CE4,CE5)
C
C **** PROPORTION OUTFLOW AND SEDIMENT TO DOWNSLOPE ADJACENT ROW AND
C **** COLUMN ELEMENTS.
C
      DRA=D*RFL(M)
      SRA=SE*RFL(M)
      I=NR(M)
      K=NC(M)
      QI(I)=QI(I)+DRA
      QI(K)=QI(K)+D-DRA
      SI(I)=SI(I)+SRA
      SI(K)=SI(K)+SE-SRA
  170 CONTINUE
      IF (CHN.LT.1..OR.SDR.EQ.0.) GO TO 180
C
C **** CALCULATE TILE DRAINAGE AND GROUNDWATER CONTRIBUTION.
C
      XPR=KPR
      CALL DRAIN (DR,DC,DIN,N,N1,N2,STD,TIAL,RFL,NR,NC)
      SDR=SDR-STD*XPR
      CHDR=SDR*GRF/XPR/CHN
      SDR=SDR*(1.-GRF)
C
C **** OUTPUT PRINT SECTION.
C
  180 Q1(L)=QI(NN)/CONV
      SSI(L)=SI(NN)*DT
      IF (QI(NN).GT.0.) GO TO 190
      SSCON(L)=0.
      GO TO 200
  190 SSCON(L)=(SI(NN)-SPT)/(SI(NN)-SPT+QI(NN)*CE6)*1000000.
  200 IF (Q1(L).GT.QMAX) QMAX=Q1(L)
      IF (SSCON(L).GT.CMAX) CMAX=SSCON(L)
      VOL=VOL+Q1(L)
      RW(L)=0.
      DO 210 I=1,NRG
      J=JTR(I)
  210 RW(L)=RW(L)+RC(I,J)*FRA(I)
      IF (RW(L).GT.RMAX) RMAX=RW(L)
C
C **** PRINT ONE HYDROGRAPH LINE.....
C
      WRITE (6,310) T(L),RW(L),Q1(L),SSI(L),SSCON(L)
      WRITE (2,310) T(L),RW(L),Q1(L),SSI(L),SSCON(L)
  220 CONTINUE
C
C **** END OF HYDROGRAPH.  PRINT TOTAL RUNOFF AND RAINFALL.
C
      L=NDT+1
  230 VOL=(VOL-.5*Q1(L-1))*DT*FLOAT(KPR)/3600.
      X=SSI(L-1)/AREA
      WRITE (6,320) PREC,PP(IT+1),VOL,PP(IT+1),X,PP(IT+5),PP(IT+7)
      WRITE (2,320) PREC,PP(IT+1),VOL,PP(IT+1),X,PP(IT+5),PP(IT+7)
C
C **** DISPLAY STRUCTURAL PRACTICE EFFECTIVENESS.
C
      K=NMAX+2
      M=K+ISTRUC-1
      DO 240 I=K,M
      IF (SI(I).EQ.0.) GO TO 240
      SI(I)=SI(I)*DT
      J=I-K+1
      WRITE (6,370) J,SI(I),PP(IT+5)
      WRITE (2,370) J,SI(I),PP(IT+5)
  240 CONTINUE
C
C **** INDIVIDUAL ELEMENT SEDIMENT LOSS (-) OR GAIN (+).
C
      X=10000./DX/DX
      IF (IT.EQ.0) X=X*4.356
      WRITE (6,340) (PP(IT+5),PP(IT+7),I=1,4)
      WRITE (2,340) (PP(IT+5),PP(IT+7),I=1,4)
C
C **** OUTPUT INDIVIDUAL ELEMENT NET SEDIMENTATION AMOUNTS AND GROSS
C **** STATISTICAL VALUES.
```

```
C
      SPAERO=0.
      SPADEP=0.
      SPASUM=0.
      SPASS=0.
C
C **** COMPUTE STATISTICS ON OVERLAND FLOW ELEMENTAL SEDIMENT YIELDS.
C
      DO 250 I=1,N
      SEL(I)=SEL(I)*DT*X
      IF (SEL(I).GT.SPADEP) SPADEP=SEL(I)
      IF (SEL(I).LT.SPAERO) SPAERO=SEL(I)
      SPASUM=SPASUM+SEL(I)
  250 SPASS=SPASS+SEL(I)*SEL(I)
      WRITE (6,360) (I,SEL(I),I=1,N)
      WRITE (2,360) (I,SEL(I),I=1,N)
      NM1=N-1
      SPASD=SQRT((SPASS-SPASUM*SPASUM/FLOAT(N))/FLOAT(NM1))
      SPAERO=-SPAERO
      WRITE (6,350) SPAERO,PP(IT+5),PP(IT+7),SPADEP,PP(IT+5),PP(IT+7),SP
     1ASD,PP(IT+5),PP(IT+7),PP(IT+5)
      WRITE (2,350) SPAERO,PP(IT+5),PP(IT+7),SPADEP,PP(IT+5),PP(IT+7),SP
     1ASD,PP(IT+5),PP(IT+7),PP(IT+5)
C
C **** NOW, OUTPUT NET DEPOSITION FOR CHANNEL AREAS.
C
      J=N+1
      DO 260 I=J,N2
  260 SEL(I)=SEL(I)*DT
      WRITE (6,360) (NR(I),SEL(I),I=J,N2)
      WRITE (2,360) (NR(I),SEL(I),I=J,N2)
C
C **** PLOTTING SECTION.  THIS SECTION OF CODE WILL CREATE THE INPUT
C **** FILE FOR SUBROUTINE HYPLT ON DEVICE 8.  SOME OF THE COMMANDS
C **** ARE MACHINE DEPENDENT AND ALL ARE PRESENTLY DISABLED.  TO USE,
C **** SIMPLY REMOVE THE C IN COLUMN 1, ADD SUBROUTINE HYPLT TO THE
C **** PROGRAM, AND APPEND THE CALCOMP LIBRARY TO THE INPUT FILE.
C **** THERE ARE TWO FORMAT STATEMENTS (380 AND 390) THAT MUST ALSO
C **** HAVE THE COMMENT DESIGNATION REMOVED!
C
C     L=L-1
C     REWIND 8
C     WRITE (8,380) L1,RMAX,QMAX,CMAX,IT,PP
C
C **** COPY HYDROGRAPH TO STORAGE TAPE.
C
C     DO 270 I=1,L
C 270 WRITE (8,390) T(I),RW(I),Q1(I),SSCON(I)
C     CALL HYPLT (L1,T,RW,Q1,SSCON,RMAX,QMAX,CMAX,IT,PP)
      STOP
C
C **** FORMATS.
C
  280 FORMAT (19A4)
  290 FORMAT (1H1,52H DISTRIBUTED HYDROLOGIC AND WATER QUALITY SIMULATIO
     1N/16X,23HBY ANSWERS VER 4.880215/19A4)
  300 FORMAT (/,15X,'OUTLET HYDROGRAPHS--VER 4.880215',/,31X,'YIELD',9X,
     1'CONCENTRATIONS -',A4,/,2X,' TIME',2X,'RAINFALL',2X,'RUNOFF',4X,'S
     2EDIMENT',3X,'SEDIMENT  PHOSPHORUS  NITROGEN',/,1X,'  MIN.',2X,2A4,
     31X,2A4,5X,A4,18X,'(N/A)',6X,'(N/A)')
  310 FORMAT (1X,F7.1,F8.2,F10.4,2F11.0)
  320 FORMAT (4X,28HRUNOFF VOLUME PREDICTED FROM,F7.2,A4,14H OF RAINFALL
     1 =,F7.3,A4/15X,19HAVERAGE SOIL LOSS =,F7.0,1X,2A4)
  330 FORMAT (///5X,48HMEAN FLOW DEPTH GREATER THAN EXPECTED AT ELEMENT,
     1I5/56H CONDITION OCCURRED BECAUSE THIS ELEMENT'S SLOPE IS MUCH,31H
     2 LESS THAN WATERSHED AVERAGE OR,/,28H CIRCULAR FLOW PATTERNS ARE ,
     358H PRESENT IN THIS VICINITY.  RECOMMENDED CORRECTIVE ACTION:,/60H
     4 INCREASE EXPECTED PEAK RUNOFF VALUE (SF) IN SUBROUTINE DATA,10H O
     5R MODIFY,/,24HELEMENT FLOW DIRECTIONS.)
  340 FORMAT (//19X,36HINDIVIDUAL ELEMENT NET SEDIMENTATION/1X,4(2X,16HE
     1LEMENT SEDIMENT)/1X,4(4X,3HNO.,3X,2A4))
  350 FORMAT (1X,'MAX EROSION RATE =',F7.0,2A4,2X,'MAX DEPOSITION RATE =
     1',F7.0,2A4,/,23X,'STD. DEV. =',F7.0,2A4,//,24X,'CHANNEL DEPOSITION
     2 --',A4,/,4(4X,'NO.     AMOUNT'))
  360 FORMAT (4(I7,F11.0))
  370 FORMAT (21H  STRUCTURAL PRACTICE,I3,32H REDUCED TOTAL SEDIMENT YIE
     1LD BY,F9.0,A4)
C 380 FORMAT (I4,2F7.2,F7.0,I3/12A4)
C 390 FORMAT (3F10.2,F10.0)
C
      END
      SUBROUTINE DATA (NDT,KPR,N,CONV,CU,SF,IT,NN,ICR,NFI,CU2,ISTRUC,SB,
     1TMIN,TMAX,NRG,DX,GRF,NEXP,DC,PP,FILTS,CWID,AREA,DT,NMAX)
C
C ****** SUBROUTINE TO INPUT WATERSHED DATA.
C
C **** MAXIMUM NUMBER OF SOIL TYPES IS 20.
C
      COMMON /CSOIL/ A(20),P(20),FC(20),GWC(20),SKDR(20)
      DIMENSION TP(20), DF(20), ASM(20), FCAP(20)
C
C **** MAXIMUM NUMBER OF SURFACE AND CROP TYPES IS 20.
C
      COMMON /CROUGH/ ROUGH(20),HU(20),DIR(21),PIT(5,20),PER(20),CDR(20)
C
C **** MAXIMUM NUMBER OF OVERLAND ELEMENTS PLUS CHANNEL ELEMENTS
C **** IS 42500.
C
C ****** IT IS EXPECTED THAT ARRAY "IEL" (IN SUBROUTINE DATA) WILL
C ****** BE OF SUCH A SIZE THAT IT WILL OVERLAY (BE EQUIVALENCED TO)
C ****** THE SPACE IN ARRAYS SI AND QI TOGETHER.  THEREFORE IT IS
C ****** NECESSARY THAT THESE TWO ARRAYS BE KEPT ADJACENT IN THEIR
C ****** COMMON BLOCK.  NOTE: THE ACTUAL NUMBER OF ELEMENTS THAT
C ****** CAN BE DIMENSIONED IN IEL WILL DEPEND ON THE WORD LENGTH
C ****** OF THE MACHINE BEING USED, E.G. ON A MACHINE WHICH USES
C ****** A SINGLE WORD INTEGER AND A DOUBLE WORD REAL, THE NUMBER
C ****** OF ELEMENTS IN IEL CAN BE FOUR TIMES THE NUMBER OF ELEMENTS
C ****** IN ARRAY SI.
C
      COMMON /CFLOW/ Q(42500),RFL(42500),FLINS(42500),SS(42500),
     &PIV(42500),B(42500),NR(42500),NC(42500),DR(42500),S(42500),
     &SL(42500),SEL(42500),SI(43000),QI(43000),DIN(42500),SST(42500)
C
C ****** ARRAYS SI AND QI MUST BE DIMENSIONED TO A SIZE = NMAX+ISTRUC+2
C ****** TO HOLD, IN ORDER, SEDIMENT AND FLOW FROM THE WATERSHED OUTLET
C ****** ELEMENT, STRUCTURAL PRACTICES AND ANY "LEAKY" ELEMENTS.
C
      EQUIVALENCE (TP(1),SST(1)), (DF(1),SST(21)), (ASM(1),SST(41))
      EQUIVALENCE (FCAP(1),SST(61)), (ITEMP(1),SST(81))
      EQUIVALENCE (IRR(1),SST(101))
      EQUIVALENCE (RN(1),SEL(1))
      EQUIVALENCE (WID(1),SEL(41)), (CN(1),SEL(51))
      EQUIVALENCE (CBAR(1),SEL(80)), (SPER(1),SEL(101)), (CROP(1,1),SEL
     1(121)), (NSTRUC(1),SEL(161))
      DIMENSION CROP(20,2), RN(20), DIRM(20), CBAR(20), SPER(20), NSTRUC
     1(4), STRNAM(3,4)
      EQUIVALENCE (DIRM(1),DIR(1))
C
C **** MAXIMUM NUMBER OF RAINGAGES IS 4 WITH 35 VALUES PER GAGE.
C
      COMMON /CRGAGE/ RC(4,35),TC(4,35),R(4,20),FRA(4),JTR(4),RATE(4),SR
     1(4),NF(4)
      DIMENSION IRR(4), IG(4), DATE(2)
      EQUIVALENCE (IEL(1,1,1),SI(1))
      DIMENSION IEL(3,303,11), ITEMP(11)
      DIMENSION IELC(3,303,2), ITEMPC(2)
      DIMENSION FILTS(42500), CWID(42500)
      EQUIVALENCE (TIAL(1),RANE(1)), (SUR(1),SOIL(1))
      EQUIVALENCE (DIN(1),CHAN(1))
      COMMON /CSURF/ SUR(42500),RANE(42500)
      INTEGER SUR,SOIL(42500),TIAL(42500),RANE,CHAN(42500)
C
C **** MAXIMUM NUMBER OF CHANNEL TYPES IS 10.
C
      DIMENSION WID(10), CN(10), PP(14), TITLE(11)
      LOGICAL STRUC
      CHARACTER*4 C1, C3, C4, C5, C6, PRI, UN, UNITS, PR, TEST
      CHARACTER*4 PP, TITLE, STRNAM, DATE
      CHARACTER*2 IG, IELC, ITEMPC, ISTL
      CHARACTER JBEG
      DATA C1,C3,C4,C5,C6,PRI,UN/' RAI','  SO','  SU','  CH','  EL','PRI
     1N','METR'/
      DATA ISTL/'TI'/
C
C **** NOW, STORE THE NAMES OF THE STRUCTURAL PRACTICES.
C
      DATA STRNAM/'PTO ','TERR','ACES','POND','S, L','AKES','G. W','ATER
     1','WAYS','FIEL','D BO','RDER'/
      STRUC=.FALSE.
C
C ****** NUMBER OF STRUCTURAL PRACTICES PERMITTED.  ARRAYS STRNAM AND
C ****** NSTRUC MUST BE REDIMENSIONED IF ISTRUC IS MODIFIED.  ALSO, THE
```

```
C ****** ADITIONAL STRUCTURE NAMES MUST BE ADDED TO THE DATA STATEMENT.
C
      ISTRUC=4
      IT=0
      OUTSID=0.
      TMAX=0.
      TMIN=1.E+10
C
C **** INPUT UNITS USED IN SIMULATION AND OUTPUT PRINT CONTROL.
C
      READ (1,800) UNITS,PR
C
C **** INPUT NUMBER OF RAINGAGES AND DATE OF EVENT.
C
      READ (1,810) TEST,NRG,DATE
      IF (NRG.GT.4) GO TO 540
      IF (TEST.NE.C1) GO TO 580
C
C **** INPUT SEPARATE RAINFALL HYETOGRAPHS FOR EACH RAINGAGE.
C
      DTMIN=900.
      TINT=DTMIN
      DO 20 I=1,NRG
      FRA(I)=0.
      READ (1,830) IG(I)
      K=2
      KM1=1
   10 READ (1,740) JBEG,TC(I,K),RC(I,K)
      IF (K.GT.2) TINT=TC(I,K)-TC(I,KM1)
      IF (TINT.LT.DTMIN) DTMIN=TINT
      K=K+1
      KM1=K-1
      IF (JBEG.EQ.' '.OR.JBEG.EQ.'0') GO TO 10
      IF (JBEG.NE.'1') GO TO 570
      IF (K.GT.35) GO TO 540
      IF (TC(I,2).LT.TMIN) TMIN=TC(I,2)
      IF (TC(I,KM1).GT.TMAX) TMAX=TC(I,KM1)
   20 IRR(I)=K
C
C **** INSERT SAME START AND FINISH TIME FOR EACH RAINGAGE RECORD.
C
      DO 30 I=1,NRG
      K=IRR(I)
      KM1=K-1
      TC(I,1)=TMIN
      RC(I,1)=0.
      IF (TC(I,KM1).EQ.TMAX) IRR(I)=IRR(I)-1
      TC(I,K)=TMAX
   30 RC(I,K)=0.
C
C ****** DEFINE DEFAULT SIMULATION REQUIREMENTS.  MAXIMUM NUMBER OF
C ****** HYDROGRAPH PRINT POINTS IS 101 (THIS IS THE NUMBER THAT WILL BE
C ****** OUTPUT).  NORMAL TIME STEP IS 60 SECONDS AND NORMAL TIME STEP
C ****** FOR INFILTRATION IS 180 SECONDS.  MAXIMUM EXPECTED RUNOFF RATE
C ****** IS 2 INCHES (50.8 MM) PER HOUR.  IF A SEGMENTED CURVE ERROR
C ****** OCCURS DURING SIMULATION, INCREASE SF BY 50 PERCENT UNTIL THAT
C ****** PROBLEM CEASES (IT MAY NOT BE THE ONLY PROBLEM, THOUGH).
C ****** FOR WATERSHEDS WITH LARGE ELEMENTS (GREATER THAN 5 ACRES),
C ****** MILD TOPOGRAPHY (LESS THAN 1 PERCENT AVERAGE SLOPES), OR
C ****** MANY ELEMENTS (MORE THAN 1000), THE SIMULATION TIME STEP, DT,
C ****** SHOULD BE INCREASED TO NO MORE THAN 300 SECONDS (5 MINUTES).
C ****** SIMILARLY, FOR SMALL ELEMENTS (LESS THAN 1 ACRE), SEVERE
C ****** TOPOGRAPHY, OR WATERSHEDS WITH ONLY A FEW ELEMENTS, THE
C ****** SIMULATION TIME STEP SHOULD BE DECREASED TO 15 - 30 SECONDS.
C
      NDT=101
c!!!!!!! DT=60.
      DT=10.
      NFI=180
c!!!!!!! SF=2.
      SF=2.
c!!!!!!! units=un=>SF=50.8
      IF (UNITS.EQ.UN) SF=50.8
      IF (UNITS.EQ.UN) IT=7
      IF (PRI.NE.PR) GO TO 50
      WRITE (6,660) DATE
      WRITE (2,660) DATE
      DO 40 I=1,NRG
      L=IRR(I)
      WRITE (6,670) IG(I),PP(IT+1),PP(IT+2),(TC(I,K),RC(I,K),K=2,L)
   40 WRITE (2,670) IG(I),PP(IT+1),PP(IT+2),(TC(I,K),RC(I,K),K=2,L)
   50 IF (DT.GT.DTMIN*60.) THEN
      WRITE (6,880)
      WRITE (2,880)
      END IF
      KPR=(TMAX-TMIN)/DT/FLOAT(NDT)*60.+1.
      IF (PRI.EQ.PR) THEN
      WRITE (6,630) DT
      WRITE (2,630) DT
      END IF
      NFI=NFI/IFIX(DT)
C
C **** INPUT INFILTRATION AND SOIL DATA.
C
      READ (1,810) TEST
      IF (TEST.NE.C3) GO TO 580
      READ (1,780) ISR
      IF (PRI.EQ.PR) THEN
      WRITE (6,750) PP(IT+1),PP(IT+2),PP(IT+1),PP(IT+2),PP(IT+1)
      WRITE (2,750) PP(IT+1),PP(IT+2),PP(IT+1),PP(IT+2),PP(IT+1)
      END IF
      IF (ISR.GT.20) GO TO 530
      ASMBAR=0.
      FPBAR=0.
      DO 60 I=1,ISR
      READ (1,790) TP(I),FCAP(I),FC(I),A(I),P(I),DF(I),ASM(I),SKDR(I)
      SPER(I)=0.
      IF (PRI.EQ.PR) THEN
      WRITE (6,640) I,TP(I),FCAP(I),FC(I),A(I),P(I),DF(I),ASM(I),SKDR(I)
      WRITE (2,640) I,TP(I),FCAP(I),FC(I),A(I),P(I),DF(I),ASM(I),SKDR(I)
      END IF
   60 CONTINUE
C
C **** INPUT DRAINAGE AND GROUNDWATER CONSTANTS.
C
      READ (1,980) DC,GRF
      NEXP=3
      IF (PRI.EQ.PR) THEN
      WRITE (6,990) DC,PP(IT+1),GRF
      WRITE (2,990) DC,PP(IT+1),GRF
      END IF
C
C **** INPUT CROP AND SURFACE ROUGHNESS DATA.
C
      READ (1,810) TEST
      IF (TEST.NE.C4) GO TO 580
      READ (1,940) ICR
      IF (PRI.EQ.PR) THEN
      WRITE (6,950) PP(IT+1),PP(IT+1),PP(IT+1)
      WRITE (2,950) PP(IT+1),PP(IT+1),PP(IT+1)
      END IF
      IF (ICR.GT.20) GO TO 550
      DO 70 I=1,ICR
      CBAR(I)=0.
      READ (1,620) CROP(I,1),CROP(I,2),PIT(1,I),PER(I),ROUGH(I),HU(I),RN
     1(I),CDR(I)
      IF (ROUGH(I).GT.1.0.OR.ROUGH(I).LE.0.) GO TO 590
      IF (PRI.EQ.PR) THEN
      WRITE (6,960) I,CROP(I,1),CROP(I,2),PIT(1,I),PER(I),ROUGH(I),
     &HU(I),RN(I),CDR(I)
      WRITE (2,960) I,CROP(I,1),CROP(I,2),PIT(1,I),PER(I),ROUGH(I),
     &HU(I),RN(I),CDR(I)
      END IF
   70 CONTINUE
C
C **** INPUT CHANNEL DATA.
C
      READ (1,810) TEST
      IF (TEST.EQ.C6) GO TO 80
      IF (TEST.NE.C5) GO TO 580
      READ (1,920) M
      IF (M.GT.10) GO TO 510
      READ (1,760) (WID(I),CN(I),I=1,M)
      IF (PRI.EQ.PR) THEN
      WRITE (6,650) PP(IT+4),(I,WID(I),CN(I),I=1,M)
      WRITE (2,650) PP(IT+4),(I,WID(I),CN(I),I=1,M)
      END IF
C
C **** INPUT OUTFLOW ELEMENT POSITION.
C
      READ (1,820) TEST,TITLE
      IF (TEST.NE.C6) GO TO 580
   80 READ (1,610) DX,NIOUT,NJOUT
C
C **** EVALUATE CONSTANTS FOR USE WITH METRIC OR ENGLISH UNITS.
C **** METRIC UNITS.
C
      DX2=DX*DX
      AREA=DX2/1.E+4
      CU1=DX2/1.E+3
      CU2=DT/DX2*500.
      CU=DX2/3.6E+6
      CONST=DX/(2./DT*DX2)**1.6667
      IF (UNITS.EQ.UN) GO TO 90
C
C **** CONVERT TO ENGLISH UNITS.
C
      CU1=CU1/.012
      CU=CU/.012
      CU2=CU2*.012
      CONST=1.486*CONST
      AREA=AREA/4.3560
C
C **** INPUT INDIVIDUAL ELEMENT TOPOGRAPHICAL DATA.
C
   90 NPAR=13
      NPAR2=11
C
C **** CHANGE DIMENSION STATEMENT BELOW IF JMAX IS CHANGED.
C
      JMAX=303
      NMAX=42500
      N=0
      II=0
      SCMIN=9.
      SCMAX=0.
      SCBAR=0.
      SMIN=9.
      SMAX=0.
      SBAR=0.
      TBAR=0.
      DO 100 J=1,JMAX
  100 IEL(3,J,3)=0
C
C **** INPUT FIRST ROW OF ELEMENTAL DATA.
C
      READ (1,680) (ITEMP(K),K=1,7),(ITEMPC(L),L=1,2),(ITEMP(K),K=8,11)
      CALL RELEM (IEL,ITEMP,N,MOUT,NIOUT,NJOUT,ISR,ICR,NMAX,JMAX,NPAR,
     1IELC,ITEMPC,NPAR2)
C
C **** PUT WATERSHED ELEMENTAL DATA INTO SINGLE DIMENSIONED ARRAYS.
C
  110 CALL RELEM (IEL,ITEMP,N,MOUT,NIOUT,NJOUT,ISR,ICR,NMAX,JMAX,NPAR,
     1IELC,ITEMPC,NPAR2)
      JS=IEL(2,1,2)
      DO 270 J=1,JS
      JM1=J-1
      I=IEL(2,J,3)
      IF (I.EQ.0) GO TO 270
      SL(I)=FLOAT(IEL(2,J,4))/1000.
      IF (SL(I).LT.SMIN) SMIN=SL(I)
      IF (SL(I).GT.SMAX) SMAX=SL(I)
      SBAR=SBAR+SL(I)
      CHAN(I)=IEL(2,J,6)/100
      IF (CHAN(I).GT.10) THEN
      WRITE (6,1020) CHAN(I),I
      WRITE (2,1020) CHAN(I),I
      END IF
      SS(I)=FLOAT(IEL(2,J,8))/1000.
C
C **** IF CHANNEL SLOPE NOT SPECIFIED, ASSUME IT'S HALF OVERLAND SLOPE.
C
      IF (SS(I).LE.0.) SS(I)=.5*SL(I)
      TIAL(I)=0
      IF (IELC(2,J,2).NE.ISTL) GO TO 120
      TIAL(I)=256
      TBAR=TBAR+1.
  120 M=FLOAT(IEL(2,J,5))/90.+1.
      MM1=M-1
C
C **** EVALUATE OUTFLOW PROPORTIONS TO ADJACENT COLUMN AND ROW ELEMENTS.
C
      ANG=(FLOAT(IEL(2,J,5))-90.*FLOAT(MM1))*.01745329
      X=SIN(ANG)+COS(ANG)
      IX=CHAN(I)
      IF (IX.EQ.0) GO TO 130
C
C **** EVALUATE CONVEYANCE FOR CHANNEL ELEMENTS.
C
      II=II+1
      CWID(II)=WID(IX)
      SS(II)=SS(I)
      IF (SS(I).LT.SCMIN) SCMIN=SS(I)
      IF (SS(I).GT.SCMAX) SCMAX=SS(I)
      SCBAR=SCBAR+SS(I)
      PIV(II)=CONST/CN(IX)/X*(DX/WID(IX)/X)**0.6667*SQRT(SS(I))
C
C **** NOW DETERMINE THE ELEMENT(S) THAT RECEIVE OUTFLOW FROM THE
C **** CURRENT ELEMENT.  NOTE:  IS IS LEGAL FOR AN ELEMENT WITH A
C **** SHADOW CHANNEL ELEMENT TO SHOW FLOW, AT THIS TEST POINT, THAT
C **** WOULD OTHERWISE BE OUTSIDE THE CATCHMENT.
C
  130 GO TO (140,150,150,140,140), M
  140 IF ((J.GE.JMAX.OR.IEL(2,J+1,3).EQ.0).AND.CHAN(I).EQ.0.AND.IEL(2,J,
     15).NE.270.AND.I.NE.MOUT) THEN
      WRITE (6,770) IEL(2,J,1),J
      WRITE (2,770) IEL(2,J,1),J
      END IF
      NR(I)=IEL(2,J+1,3)
      GO TO (160,160,170,170,160), M
  150 IF ((J.LE.1.OR.IEL(2,JM1,3).EQ.0).AND.IEL(2,J,5).NE.90.AND.I.NE.MO
     1UT.AND.CHAN(I).EQ.0) THEN
      WRITE (6,770) IEL(2,J,1),J
      WRITE (2,770) IEL(2,J,1),J
      END IF
      NR(I)=IEL(2,JM1,3)
      GO TO (160,160,170,170,160), M
  160 IF (IEL(1,J,3).EQ.0.AND.IEL(2,J,5).NE.0.AND.CHAN(I).EQ.0.AND.IEL(2
     1,J,5).NE.360.AND.I.NE.MOUT) THEN
      WRITE (6,770) IEL(2,J,1),J
      WRITE (2,770) IEL(2,J,1),J
      END IF
      NC(I)=IEL(1,J,3)
      GO TO 180
  170 IF (IEL(3,J,3).EQ.0.AND.IEL(2,J,5).NE.180.AND.I.NE.MOUT.AND.CHAN(I
     1).EQ.0) THEN
      WRITE (6,770) IEL(2,J,1),J
      WRITE (2,770) IEL(2,J,1),J
      END IF
      NC(I)=IEL(3,J,3)
  180 IF (ANG.GT..78539816) GO TO 190
      RFL(I)=.5*SIN(ANG)/COS(ANG)
      GO TO 200
  190 RFL(I)=1.-.5*SIN(1.5707963-ANG)/COS(1.5707963-ANG)
  200 GO TO (210,220,210,220,210), M
  210 RFL(I)=1.-RFL(I)
C
C **** ELIMINATE FALSE RECEIVING ELEMENTS WHICH MAY CAUSE OUT-OF-RANGE
C **** SUBSCRIPTS FOR SOME BOUNDARY ELEMENTS.
C
  220 IF (RFL(I).LT.0.01) NR(I)=NC(I)
      IF (RFL(I).GT.0.99) NC(I)=NR(I)
C
C **** "LEAKY" ELEMENTS (THOSE WITH PARTIAL FLOW OUTSIDE THE WATERSHED)
C **** MUST DIVERT THAT PARTIAL FLOW INTO A SPECIAL PSUEDO ELEMENT.
C
      IF (NC(I).GT.0.OR.I.EQ.MOUT) GO TO 230
C
C **** THIS ELEMENT LEAKS, DIVERT IT INTO SPECIAL "BOTTOMLESS PIT".
C
      NC(I)=NMAX+ISTRUC+2
C
C **** ADD TO TOTAL NON-CONTRIBUTING AREA.
C
      OUTSID=OUTSID+1.-RFL(I)
  230 IF (NR(I).GT.0.OR.I.EQ.MOUT) GO TO 240
      NR(I)=NMAX+ISTRUC+2
      OUTSID=OUTSID+RFL(I)
C
C **** GET CROP/MGMT NUMBER.
C
  240 I1=IEL(2,J,7)
      CBAR(I1)=CBAR(I1)+1.
C
```

```
C **** PUT CROP/MANAGEMENT NUMBER IN LOW BYTE AND SOIL TYPE NUMBER IN
C **** NEXT BYTE OF (SOIL:SUR).
C
      K=MOD(IEL(2,J,6),100)
      SPER(K)=SPER(K)+1.
      SOIL(I)=(K*256)+I1
      ASMBAR=ASMBAR+ASM(K)
      FPBAR=FPBAR+FCAP(K)
      B(I)=CONST*SQRT(SL(I))*X/RN(I1)
C
C **** MAKE SPECIAL ADJUSTMENTS TO ACCOUNT FOR STRUCTURAL PRACTICES,
C **** BUT FIRST SEE IF ANY ARE PRESENT IN THIS ELEMENT.
C
      IF (IEL(2,J,9).NE.0) CALL STRUCT (I,J,NC(I),NR(I),RFL(I),IEL,JMAX,
     1NPAR,NMAX,STRUC,NSTRUC,ISTRUC,X,DX,WID,SS(II+1),SS(I),PIV(II+1),CN
     2,CWID(II+1),CHAN(I),CONST,SL(I),II,SCMIN,SCMAX,SCBAR,ANG,IELC,NPAR
     32)
C
C **** RENUMBER RAINGAGES TO 1,2,..,NRG IN ORDER OF HYETOGRAPH INPUTS.
C
      DO 250 K=1,NRG
      IF (IELC(2,J,1).EQ.IG(K)) GO TO 260
  250 CONTINUE
      WRITE (6,600) IELC(2,J,1),IEL(2,J,1),J,IG(1)
      WRITE (2,600) IELC(2,J,1),IEL(2,J,1),J,IG(1)
      K=1
C
C **** PUT RAINGAGE NUMBER IN LOW BYTE AND TILE NUMBER IN NEXT BYTE
C **** OF (TIAL:RANE).
C
  260 RANE(I)=TIAL(I)+K
  270 CONTINUE
      JS=IEL(3,1,2)
      IF (ITEMP(3).NE.999.AND.IEL(3,JS,1).NE.ITEMP(1)) GO TO 110
      ITEMP(3)=999
      IF (JS.NE.JMAX) GO TO 110
      IF (N+II.GT.NMAX) GO TO 520
      X=N
      ASMBAR=ASMBAR/X
      FPBAR=FPBAR/X
      SB=AREA
      AREA=AREA*(X-OUTSID)
      CONV=CU*(X-OUTSID)
      SBAR=SBAR/X
      IF (II.GT.0) SCBAR=SCBAR/FLOAT(II)
      NN=N+1
C
C **** OUTPUT STATISTICAL SUMMARY OF WATERSHED CHARACTERISTICS.
C
      TBAR=TBAR/X
      WRITE (6,690) TITLE,SB,PP(IT+3),N,II,AREA,PP(IT+3),SMIN,SBAR,SMAX,
     1SCMIN,SCBAR,SCMAX,TBAR,DC,PP(IT+1),ASMBAR,FPBAR,GRF,MOUT,NIOUT,NJO
     2UT
      WRITE (6,700) PP(IT+1),PP(IT+2),PP(IT+1),PP(IT+2),PP(IT+1)
      WRITE (2,690) TITLE,SB,PP(IT+3),N,II,AREA,PP(IT+3),SMIN,SBAR,SMAX,
     1SCMIN,SCBAR,SCMAX,TBAR,DC,PP(IT+1),ASMBAR,FPBAR,GRF,MOUT,NIOUT,NJO
     2UT
      WRITE (2,700) PP(IT+1),PP(IT+2),PP(IT+1),PP(IT+2),PP(IT+1)
      DC=DC*CU/24.
      SB=CONST*SQRT(SBAR)/RN(1)
      J=0
      DO 330 I=1,ICR
      IF (CBAR(I).LE.0..AND.I.LT.ICR) GO TO 330
      CBAR(I)=CBAR(I)/X
      IF (J.GE.ISR) GO TO 320
  280 J=J+1
      DO 300 JJ=J,ISR
      IF (SPER(JJ).LE.0.) GO TO 300
      FPBAR=FC(JJ)+A(JJ)*(1.-ASM(JJ))**P(JJ)
      SPER(JJ)=SPER(JJ)/X
      IF (CBAR(I).LE.0.) GO TO 290
      WRITE (6,710) CROP(I,1),CROP(I,2),CBAR(I),PER(I),RN(I),CDR(I),JJ,S
     1PER(JJ),FC(JJ),FPBAR,DF(JJ),SKDR(JJ)
      WRITE (2,710) CROP(I,1),CROP(I,2),CBAR(I),PER(I),RN(I),CDR(I),JJ,S
     1PER(JJ),FC(JJ),FPBAR,DF(JJ),SKDR(JJ)
      CBAR(I)=0.
      GO TO 310
  290 WRITE (6,720) JJ,SPER(JJ),FC(JJ),FPBAR,DF(JJ),SKDR(JJ)
      WRITE (2,720) JJ,SPER(JJ),FC(JJ),FPBAR,DF(JJ),SKDR(JJ)
      GO TO 310
  300 CONTINUE
```

```
      J=ISR
      GO TO 320
  310 J=JJ
      IF (I.LT.ICR) GO TO 330
      IF (J.LT.ISR) GO TO 280
  320 IF (CBAR(I).GT.0.) THEN
      WRITE (6,730) CROP(I,1),CROP(I,2),CBAR(I),PER(I),RN(I),CDR(I)
      END IF
  330 CONTINUE
      NR(MOUT)=NN
      NC(MOUT)=NN
      IF (II.NE.0) GO TO 340
      N2=N
      GO TO 410
C
C **** DETERMINE SHADOW ELEMENT CONTINUITY.
C **** FIND CHANNEL SEGMENTS.
C
  340 DO 350 J=1,N
      IF (CHAN(J).EQ.0) GO TO 350
C
C **** USE THE ROW FLOW POINTER TO REMEMBER ORIGINAL ELEMENT NUMBER
C **** OF THIS CHANNEL ELEMENT, SINCE THE FLOW COMPONENT IN THE ROW
C **** DIRECTION IS 0.
C
      NR(NN)=J
      NN=NN+1
  350 CONTINUE
C
C **** MOVE CHANNEL PARAMETERS TO END OF ARRAYS.
C
      N2=NN-1
      N1=N+1
      DO 390 I=N1,N2
      I1=I-N
      B(I)=PIV(I1)
      CWID(I)=CWID(I1)
      SL(I)=SS(I1)
      J=NR(I)
      I1=NC(J)
      I2=NR(J)
C
C **** IF CERTAIN STRUCTURES ARE PRESENT IN AN ELEMENT WITH A SHADOW
C **** ELEMENT, IT IS LIKELY THAT THE RECEIVING CHANNEL ELEMENT WILL
C **** NOT BE GETTING THE MAJOR OUTFLOW.
C
      IF (I1.GT.NMAX) GO TO 360
      IF (I2.GT.NMAX) GO TO 380
C
C **** THIS ELEMENT DOES NOT CONTAIN A STRUCTURE; THEREFORE, THE
C **** RECEIVING CHANNEL ELEMENT SHOULD BE IN THE DIRECTION OF THE
C **** PREDOMINANT FLOW COMPONENT.
C
      IF (RFL(J).LT.0.207107) GO TO 380
      IF (RFL(J).GT.0.792893) GO TO 360
C
C **** FLOW DIRECTION IS PREDOMINANTLY DIAGONAL.
C ****** IF ROW FLOW DESTINATION NUMBER IS LESS THAN CURRENT ELEMENT
C ****** NUMBER, THE DIAGONAL POINTS TO THE LEFT AND THE DIAGONAL
C ****** DESTINATION ELEMENT CAN BE COMPUTED BY SUBTACTING ONE FROM
C ****** THE CONVENTIONAL OVERLAND FLOW COLUMN DESTINATION NUMBER.
C
      IF (I2.LT.J) GO TO 370
      I1=I1+1
      GO TO 380
  360 I1=I2
      GO TO 380
  370 I1=I1-1
C
C **** MAKE CERTAIN THE RECEIVING ELEMENT IS A CHANNEL ELEMENT.
C
  380 IF (CHAN(I1).LT.1.AND.J.NE.MOUT) GO TO 560
C
C **** TEMPORARILY ASSIGN THE ORIGINAL OVERLAND FLOW ELEMENT NUMBER
C **** AS THE DESTINATION FOR THE SHADOW OUTFLOW.  THIS IS NECESSARY
C **** UNTIL NEW NUMBERS ARE ASSIGNED TO ALL SHADOW ELEMENTS.
C
      NC(I)=I1
C
C **** MAKE ALL OVERLAND FLOW FROM THIS ELEMENT GO INTO ITS SHADOW
C **** ELEMENT, UNLESS IT CONTAINS A STRUCTURAL PRACTICE.
```

```
C
      IF (NR(J).LE.NMAX) NR(J)=I
      IF (NC(J).LE.NMAX) NC(J)=I
  390 CONTINUE
C
C **** FIND REAL CHANNEL SEGMENT NUMBER INTO WHICH EACH CHANNEL
C **** SEGMENT FLOWS.
C
      DO 400 J=N1,N2
      I=NC(J)
      NC(J)=NR(I)
C
C **** IF THIS ELEMENT CONTAINS A STRUCTURAL MEASURE, ITS CORRECT
C **** CHANNEL ELEMENT NUMBER MAY BE PRESENT ONLY IN ARRAY NC.
C
      IF (NC(J).GT.NMAX) NC(J)=NC(I)
C
C **** FORCE ALL CHANNEL FLOW TO USE ONLY COLUMN FLOW DIRECTIONS.
C
  400 RFL(J)=0.
      J=NR(MOUT)
      NC(J)=NN
C
C **** OUTPUT DATA CONCERNING ANY STRUCTURAL PRACTICES.
C
  410 IF (.NOT.STRUC) GO TO 430
      WRITE (6,1000)
      WRITE (2,1000)
      DO 420 I=1,ISTRUC
      IF (NSTRUC(I).NE.0) THEN
      WRITE (6,1010) I,(STRNAM(J,I),J=1,3),NSTRUC(I)
      WRITE (2,1010) I,(STRNAM(J,I),J=1,3),NSTRUC(I)
      END IF
  420 CONTINUE
C
C **** EVALUATE INITIAL CONDITIONS.
C
  430 DO 440 I=1,N2
      S(I)=0.
  440 FLINS(I)=0.
C
C **** CONVERT SOIL CONSTANTS.
C
      DO 450 I=1,ISR
      FC(I)=CU*FC(I)
      TP(I)=TP(I)*CU1*DF(I)
      A(I)=CU*A(I)*(DT/TP(I))**P(I)
  450 GWC(I)=(1.-FCAP(I))*TP(I)/DT
C
C **** INITIALIZE VALUES SPECIFIC TO INDIVIDUAL ELEMENTS.
C
      Y=1./X
      DO 460 I=1,N
      K=2
      IS=SOIL(I)/256
      IC=MOD(SUR(I),256)
      PIV(I)=(1.-ASM(IS))*TP(IS)/DT
C
C **** CONTINUE FOR SURFACE INITIAL CONDITION.
C
      J=MOD(RANE(I),256)
      IF (TC(J,2).LT.(TMIN+1.1)) K=3
      FRA(J)=FRA(J)+Y
      SUPP=RC(J,K)*(1.-PER(IC))*CU
      X=FILT(A(IS),PIV(I),P(IS),FC(IS),GWC(IS),DR(I),S(I),SUPP,CU2,ROUGH
     1(IC),HU(IC),NEXP)
      FILTS(I)=X
      IF (X.GT.SUPP) X=SUPP
  460 FLINS(I)=SUPP-X
C
C **** CONVERT SURFACE VALUES.
C
      DO 480 I=1,ICR
      pitini=PIT(1,I)
      DIRM(I)=0.10*HU(I)
      DO 470 J=1,NRG
  470 PIT(J,I)=pitini*CU1/DT
      ADIR=HU(I)*ROUGH(I)*(DIRM(I)/HU(I))**(1./ROUGH(I))
  480 DIR(I)=ADIR*2.*CU1/DT
C
C **** SET CHANNEL RETENTION TO ZERO.
```

```
C
      DIR(21)=0.
      J=NMAX+ISTRUC+2
      DO 500 I=1,J
      IF (I.GT.NMAX) GO TO 490
      Q(I)=0.
      SS(I)=0.
      SEL(I)=0.
      SST(I)=0.
      DIN(I)=0.
  490 QI(I)=0.
  500 SI(I)=0.
      RETURN
C
C **** ERROR MESSAGES.
C
  510 WRITE (6,930)
      WRITE (2,930)
      STOP
  520 WRITE (6,840)
      WRITE (2,840)
      STOP
  530 WRITE (6,860)
      WRITE (2,860)
      STOP
  540 WRITE (6,850)
      WRITE (2,850)
      STOP
  550 WRITE (6,870)
      WRITE (2,870)
      STOP
  560 WRITE (6,890) J
      WRITE (2,890) J
      STOP
  570 WRITE (6,900) NRG,J
      WRITE (2,900) NRG,J
      STOP
  580 WRITE (6,910) TEST
      WRITE (2,910) TEST
      STOP
  590 WRITE (6,970) ROUGH(I),CROP(I,1),CROP(I,2)
      WRITE (2,970) ROUGH(I),CROP(I,1),CROP(I,2)
      STOP
C
C **** FORMATS.
C
  600 FORMAT (1X,27HRAIN DATA MISSING FOR GAGE ,A2,12H, AT ELEMENT,I4,1H
     1,,I4,7H: GAGE ,A2,10H DATA USED)
  610 FORMAT (16X,F6.1/17X,I4,8X,I4)
  620 FORMAT (11X,2A4,6X,F3.2,6X,F3.2,5X,F3.2,5X,F4.2,4X,F4.3,4X,F3.2)
  630 FORMAT (/1X,27HSIMULATION TIME INCREMENT =,F5.0,8H SECONDS)
  640 FORMAT (I4,2PF9.1,F11.1,0PF11.2,F8.2,F7.2,F9.1,2PF10.1,0PF9.2)
  650 FORMAT (/1X,18HCHANNEL PROPERTIES/1X,4HTYPE,3X,5HWIDTH,3X,11HMANNI
     1NG'S N/9X,A4/(I4,F8.1,F11.3))
  660 FORMAT (//5X,33HRAINFALL HYETOGRAPH FOR EVENT OF ,2A4)
  670 FORMAT (/5X,12HGAGE NUMBER ,A2/5X,11HTIME - MIN.,7X,15HRAINFALL RA
     1TE -,2A4/(F14.1,F24.2))
  680 FORMAT (2I3,I2,I3,3I4,3X,A2,1X,A2,2X,I4,I3,2I4)
  690 FORMAT (/,5X,11A4,/,5X,'WATERSHED CHARACTERISTICS',/,' NUMBER OF',
     1F6.2,A4,' OVERLAND FLOW ELEMENTS =',I5,/,1X,'NUMBER OF CHANNEL SEG
     2MENTS = ',I3,/,1X,'AREA OF CATCHMENT =',F8.1,A4,/,1X,'CATCHMENT SL
     3OPE: MIN =',2PF7.2,'  AVE =',F7.2,'  MAX =',F7.2,' PERCENT',/,1X,'
     4CHANNEL SLOPE:   MIN =',F7.2,'  AVE =',F7.2,'  MAX =',F7.2,' PERCE
     5NT',/,1X,'PERCENT OF AREA TILED =',F6.1,' WITH A D.C. OF',0PF5.2,A
     64,'/24H',/,' MEAN ANTECEDENT SOIL MOISTURE =',2PF4.0,', FIELD CAPA
     7CITY =',F4.0,' PERCENT SATURATION',/,' GROUNDWATER RELEASE FRACTIO
     8N =',0PF7.4,/,1X,'OUTLET IS ELEMENT',I5,' AT ROW',I4,' COL',I4)
  700 FORMAT (/,' SURFACE COVER/MANAGEMENT CONDITIONS',8X,'SOIL ASSOCIAT
     1ION PROPERTIES',/,3X,'CROP  PERCENT PERCENT  N',4X,'C',5X,'NO. PER
     2CENT   FC',4X,'INITIAL  CONTROL  K',/,9X,'PRESENT  COVER',18X,'PRE
     3SENT',4A4,' DEPTH',A4)
  710 FORMAT (1X,2A4,2PF6.1,F7.0,0PF6.3,F6.2,I5,2PF7.1,0PF7.1,1X,2F8.1,F
     17.2)
  720 FORMAT (I39,2PF7.1,0PF7.1,1X,2F8.1,F7.2)
  730 FORMAT (1X,2A4,2PF6.1,F7.0,0PF6.3,F6.2)
  740 FORMAT (A1,F9.0,F10.0)
  750 FORMAT (//1X,15HSOIL PROPERTIES/1X,4HSOIL,2X,8HPOROSITY,2X,10HFIEL
     1D CAP.,2X,22HINFILTRATION CONSTANTS,2X,7HCONTROL,2X,10HANTECEDENT,
     21X,7HEROSION/7X,8H(PERCENT,3X,8H(PERCENT,6X,2HFC,7X,1HA,6X,1HP,5X,
     34HZONE,5X,8HMOISTURE,3X,6HCONST./9X,5HVOL.),6X,5HSAT.),4X,2A4,2A4,
     49X,A4,3X,13H(PERCENT SAT))
  760 FORMAT (18X,F4.0,27X,F5.0)
  770 FORMAT (8H ELEMENT,I4,1H,,I4,27H FLOWS OUT OF THE WATERSHED)
  780 FORMAT (18X,I4)
  790 FORMAT (10X,F3.2,6X,F3.2,6X,F5.2,5X,F5.3,5X,F3.2,6X,F5.1,7X,F3.2,5
     1X,F3.2)
  800 FORMAT (1X,A4,52X,A4)
  810 FORMAT (A4,15X,I1,25X,2A4)
  820 FORMAT (A4,24X,11A4)
  830 FORMAT (16X,A2)
  840 FORMAT (37H NUMBER OF SHED+CHAN ELEMENTS EXCEEDS,10H DIMENSION)
  850 FORMAT (32H RAINFALL DATA EXCEEDS DIMENSION)
  860 FORMAT (31H NO. OF SOILS EXCEEDS DIMENSION)
  870 FORMAT (36H NO. OF CROPS EXCEEDS DIMENSION SPEC)
  880 FORMAT (47H ANALYSIS IS NOT ACCURATE IF RAINFALL INTENSITY,28H INT
     1ERVALS ARE LESS THAN DT.)
  890 FORMAT (39HCHANNELS DISCONTINUOUS NEAR ELEMENT NO.,I5)
  900 FORMAT (1X,37HHYETOGRAPH DATA MISSING OR INCORRECT,,24H FIRST COLU
     1MN NOT 0 OR 1/I4,40H GAGES REQUESTED. BAD LINE BEGINS WITH: ,A2)
  910 FORMAT (24HINCORRECT INPUT SEQUENCE,36H OR HEADER CARD.  CARD BEGI
     1NS WITH: ,A4)
  920 FORMAT (30X,I3)
  930 FORMAT (39H NO. OF CHANNEL TYPES EXCEEDS DIMENSION)
  940 FORMAT (31X,I3)
  950 FORMAT (/7H COVER/,20HMANAGEMENT PRACTICES/3X,4HCROP,6X,9HMAX. POT
     1.,3X,7HPERCENT,2X,6HROUGH.,2X,6HROUGH.,2X,9HMANNING'S,3X,7HEROSION
     2/11X,12HINTERCEPTION,3X,5HCOVER,3X,6HCOEFF.,2X,6HHEIGHT,6X,1HN,8X,
     36HCONST./14X,A4,25X,A4,16X,A4)
  960 FORMAT (1X,I2,1X,2A4,F7.2,2PF12.0,0PF8.2,F8.1,F10.3,F10.2)
  970 FORMAT (20H ROUGHNESS COEFF. OF,F8.2,27H IS OUT OF RANGE FOR CROP:
     1 ,2A4)
  980 FORMAT (39X,F5.2/31X,E10.3)
  990 FORMAT (/1X,22HTILE DRAINAGE COEFF. =,F5.2,A4,4H/24H/1X,30HGROUNDW
     1ATER RELEASE FRACTION =,E10.3)
 1000 FORMAT (/3X,28HSTRUCTURAL MEASURES INCLUDED,/10X,4HTYPE,9X,6HNUMBE
     1R)
 1010 FORMAT (I7,2X,3A4,I6)
 1020 FORMAT (1X,11HCHANNEL NO.,I5,15H AT ELEMENT NO.,I5)
C
      END
      SUBROUTINE STRUCT (I,J,NC,NR,RFL,IEL,JMAX,NPAR,NMAX,STRUC,NSTRUC,I
     1STRUC,X,DX,WID,SSII,SSI,PIV,CN,CWID,CHAN,CONST,SL,II,SCMIN,SCMAX,S
     2CBAR,ANG,IELC,NPAR2)
C
C ****** SUBROUTINE TO ADJUST PARAMETERS TO REFLECT STRUCTURAL PRACTICES
C ****** INSTALLED WITHIN AN ELEMENT.
C
      DIMENSION IEL(3,JMAX,NPAR2), NSTRUC(ISTRUC), WID(10), CN(10)
      DIMENSION IELC(3,JMAX,2)
      INTEGER CHAN,PRACT
      LOGICAL STRUC
      CHARACTER*2 IELC
C
C **** SWITCH TO APPROPRIATE HANDLER FOR EACH STRUCTURAL TYPE.
C
      PRACT=IEL(2,J,9)
      IF (PRACT.GT.ISTRUC.OR.PRACT.LT.0) GO TO 90
      STRUC=.TRUE.
      NSTRUC(PRACT)=NSTRUC(PRACT)+1
      GO TO (10,60,70,80), PRACT
C
C **** HANDLE PONDS AND TILE-OUTLET TERRACES BY USING A TRAP EFFICIENCY
C **** APPROACH, FOR BOTH SEDIMENT AND WATER.
C
C **** CASE 1 IS FOR A PTO.
C
   10 TRAP=.90
C
C **** CHECK FOR A POSSIBLE SHADOW CHANNEL ELEMENT.
C
   20 IF (CHAN.EQ.0) GO TO 40
C
C **** IT'S A CHANNEL ELEMENT, DOES IT REQUIRE DIAGONAL FLOW?
C
      IF (ANG.LT..3926991.OR.ANG.GT.1.178097) GO TO 40
C
C **** FLOW IS DIAGONAL, CHANGE DESTINATION ELEMENT NUMBERS.
C
      IF (NR.LT.I) GO TO 30
      NR=NC+1
      NC=NC+1
      GO TO 40
   30 NR=NC-1
      NC=NC-1
C
C **** THE PREDOMINANT OVERLAND DIRECTION IS MAINTAINED AND THAT
C **** ELEMENT WILL RECEIVE THE UNTRAPPED FLOW AND SEDIMENT.
C
   40 IF (RFL.GT..5) GO TO 50
      RFL=TRAP
      NR=NMAX+1+PRACT
      RETURN
   50 RFL=1.-TRAP
      NC=NMAX+1+PRACT
      RETURN
C
C **** PONDS ARE SIMILAR TO PTO'S, BUT HAVE A HIGHER TRAP EFFICIENCY.
C
   60 TRAP=.95
      GO TO 20
C
C **** GRASSED WATERWAYS DIRECTLY AFFECT ONLY THE VEGETAGED AREA OF
C **** THE ELEMENT IN WHICH THEY ARE LOCATED, BUT THEY MUST ALSO ASSURE
C **** THAT THIS ELEMENT HAS A SHADOW CHANNEL ELEMENT.
C
   70 IF (CHAN.NE.0) GO TO 80
C
C **** CURRENT ELEMENT DOES NOT HAVE A SHADOW CHANNEL ELEMENT, MAKE ONE.
C
      CHAN=IEL(2,J,11)
      IF (CHAN.EQ.0) CHAN=1
      II=II+1
      CWID=WID(CHAN)
      PIV=CONST/CN(CHAN)/X*(DX/CWID/X)**.6667*SQRT(SSI)
      SSII=SSI
      IF (SSI.LT.SCMIN) SCMIN=SSI
      IF (SSI.GT.SCMAX) SCMAX=SSI
      SCBAR=SCBAR+SSI
C
C **** NOW ACCOUNT FOR VEGETATED AREA BY REDUCING THE SEDIMENT
C **** DETACHMENT BY FLOW FOR THIS ELEMENT BY AN AMOUNT PROPORTIONAL
C **** TO THE VEGETATED AREA.  SINCE FLOW DETACHMENT IS DIRECTLY
C **** PROPORTIONAL TO THE OVERLAND SLOPE, ADJUST THAT PARAMETER.
C
C **** FIELD BORDERS HAVE A SIMILAR EFFECT TO THE VEGETATED AREA
C **** OF GRASSED WATERWAYS.
C
   80 TRAP=FLOAT(IEL(2,J,10))/DX
      IF (TRAP.GT..5) TRAP=.5
      SL=SL*(1.-TRAP)
      RETURN
C
C **** CHECK TO SEE IF IT'S A MANAGEMENT PRACTICE BEFORE SPOUTING OFF.
C
   90 IF (PRACT.GT.10.AND.PRACT.LT.13) RETURN
      WRITE (6,100) IEL(2,J,9),IEL(2,J,1),J
      WRITE (2,100) IEL(2,J,9),IEL(2,J,1),J
      RETURN
C
  100 FORMAT (14H  PRACTICE NO.,I3,7H IN ROW,I4,5H, COL,I4,20H ILLEGAL A
     1ND IGNORED)
C
      END
      SUBROUTINE DRAIN (DR,DC,DIN,N,N1,N2,STD,TIAL,RFL,NR,NC)
C
C ****** SUBROUTINE FOR SUBSURFACE DRAINAGE.
C
      DIMENSION DR(42500), DIN(42500), RFL(42500)
      INTEGER NR(42500),NC(42500),TIAL(42500)
C
C **** SET ALL CHANNEL INFLOWS TO ZERO.
C
      DO 10 I=N1,N2
   10 DIN(I)=0.
      STD=0.
C
C **** ROUTE DRAINAGE FROM TILES.
C
      DO 50 I=1,N
      DRANE=0.
      IF (TIAL(I).LT.256) GO TO 40
      IF (DR(I).GT.DC) GO TO 20
      DRANE=DR(I)
      GO TO 30
   20 DRANE=DC
   30 STD=STD+DRANE
   40 DRANE=DRANE+DIN(I)
      DD=RFL(I)*DRANE
      J=NR(I)
      K=NC(I)
      DIN(J)=DIN(J)+DD
      DIN(K)=DIN(K)-DD+DRANE
   50 DIN(I)=0.
      RETURN
C
      END
      FUNCTION FILT(A,PIV,P,FC,GWC,DR,S,R,CU2,ROUGH,HU,NEXP)
C
C ****** CALCULATION OF INFILTRATION CAPACITY.
C
C **** POTENTIAL INFILTRATION CAPACITY -- WHOLE SURFACE COVERED.
C
      IF (PIV) 30,40,10
C
C **** UNSATURATED INFILTRATION ZONE.
C
   10 FMAX=A*PIV**P+FC
      IF (PIV.LT.GWC) GO TO 20
      DR=0.
      GO TO 50
   20 DR=FC*(1.-PIV/GWC)**NEXP
      GO TO 50
C
C **** INFILTRATION ZONE SATURATED.
C
   30 PIV=0.
   40 DR=FC
      FMAX=FC
C
C **** ADJUST INFILTRATION ACCORDING TO FRACTION OF AREA INUNDATED.
C **** REMAINING AREA INFILTRATES AT RAINFALL RATE.
C
   50 IF (R.GE.FMAX.OR.HU.LE.0.) GO TO 70
      DEP=S*CU2
      IF (DEP.GT.1.E-10) GO TO 60
      FWA=0.
      GO TO 90
   60 FH=DEP/HU/ROUGH
      IF (FH.LT.1.) GO TO 80
C
C **** ENTIRE SURFACE INUNDATED OR RAINFALL RATE EXCEEDS SOIL
C **** INFILTRATION CAPACITY.
C
   70 FILT=FMAX
      RETURN
C
C **** INFILTRATION CAPACITY REDUCED BELOW ITS POTENTIAL VALUE.
   80 FWA=FH**(1.-ROUGH)
   90 FILT=FWA*FMAX+(1.-FWA)*R
      RETURN
C
      END
      FUNCTION RAIN(RATE,PIT,PER)
C
C ****** DETERMINATION OF NET RAINFALL RATE.
C
      IF (PIT) 40,50,10
   10 RIT=PER*RATE
      IF (RIT-PIT) 20,30,30
   20 RAIN=RATE-RIT
      PIT=PIT-RIT
      RETURN
   30 RAIN=RATE-PIT
      PIT=0.
      RETURN
   40 PIT=0.
   50 RAIN=RATE
      RETURN
C
      END
      SUBROUTINE SED (ZW,ZL,R,SL,Q,CDR,SKDR,SI,SEL,SE,S,M,N,SST,CE1,CE2,
     1CE3,CE4,CE5)
C
C ****** SUBROUTINE FOR SEDIMENT DETACHMENT AND TRANSPORT CALCULATIONS.
```

```
C
      IF (Q.GT.0.) GO TO 10
C
C **** NO OUTFLOW, ALL SEDIMENT ASSUMED DEPOSITED.
C
      SEL=SEL+(SST+SI)/2.
      SST=SI
      SE=0.
      SI=0.
      RETURN
C
C **** OUTFLOW.
C
   10 QX=Q/ZW
      IF (QX.LE.CE5) GO TO 20
      TF=CE1*QX*Q*SL
      GO TO 30
   20 TF=CE2*SL*SQRT(Q*ZW)
   30 C=CDR*SKDR
      DR=C*CE3*R*R/ZL/ZW
      DF=C*CE4*SL*Q*ZL
C
C **** ONLY ALLOW PICKUP OF DEPOSITED SEDIMENT IN CHANNELS.
C
      IF (M.GT.N.AND.DF.GT.SEL) DF=SEL
      DS=SI+DR+DF
      S2=(SST+DS)/(Q/S+1.)
      SE=S2*Q/S
      IF (SE.LT.TF) GO TO 40
C
C **** SEDIMENT SUPPLY EXCEEDS TRANSPORT CAPACITY.
C
      ZI2=TF*(1.+S/Q)-SST
      SEL=SEL+SI-ZI2
      SST=ZI2+TF*(S/Q-1.)
      SE=TF
      GO TO 50
C
C **** SEDIMENT SUPPLY LESS THAN TRANSPORT CAPACITY.
C
   40 IF (SE.LT.0.) SE=0.
      SST=DS-SE+S2
      SEL=SEL-DF-DR
   50 SI=0.
      RETURN
C
      END
      SUBROUTINE RELEM (IEL,ITEMP,N,MOUT,NIOUT,NJOUT,ISR,ICR,NMAX,JMAX,N
     1PAR,IELC,ITEMPC,NPAR2)
C
C ******  SUBROUTINE TO SET UP NEXT ROW OF WATERSHED ELEMENTAL DATA.
C **** INTO THE PROPER POSITION OF THE "3-ROW PER PASS" ARRAY.
C
      DIMENSION IEL(3,JMAX,NPAR2), ITEMP(NPAR2)
      DIMENSION IELC(3,JMAX,2), ITEMPC(2)
      CHARACTER*2 IELC, ITEMPC
C
C **** "RIPPLE" ROW 2 INTO ROW 1 AND ROW 3 INTO ROW 2, THEN ZERO
C **** THE THIRD ROW.
C
      DO 20 J=1,JMAX
      DO 10 I=1,NPAR-2
      IEL(1,J,I)=IEL(2,J,I)
   10 IEL(2,J,I)=IEL(3,J,I)
   20 IEL(3,J,3)=0
      DO 25 J=1,JMAX
      DO 23 I=1,2
      IELC(1,J,I)=IELC(2,J,I)
   23 IELC(2,J,I)=IELC(3,J,I)
   25 CONTINUE
C
C **** SET UP POSSIBLE LAST ROW TEST FLAG.
C
      IEL(3,1,2)=JMAX
      IF (ITEMP(3).EQ.999) RETURN
C
C **** NOW TRANSFER CURRENT WATERSHED ELEMENTAL DATA INTO THE THIRD
C **** ROW OF THE "3-ROW PER PASS" ARRAY.
C
C ****** IEL(I,J,3) CONTAINS THE POSITION NUMBER FOR THAT ELEMENT IN
C ****** THE SINGLE DIMENSION ARRAYS USED FOR SIMULATION ANALYSIS.
```

```
C ****** IEL(I,1,2) CONTAINS THE COLUMN NUMBER OF THE LAST WATERSHED
C ****** ELEMENT IN THE ROW.
C
   30 J=ITEMP(2)
      K=MOD(ITEMP(6),100)
      ITEMP(6)=ITEMP(6)/100*100+K
      IF (K.LE.0.OR.K.GT.ISR) GO TO 80
      IF (ITEMP(7).LE.0.OR.ITEMP(7).GT.ICR) GO TO 90
      IF (J.GT.JMAX) GO TO 50
C
C **** TRANSFER PARAMETER DATA FROM A SINGLE ELEMENT.
C
      DO 40 I=1,NPAR-2
   40 IEL(3,J,I)=ITEMP(I)
      DO 45 I=1,2
   45 IELC(3,J,I)=ITEMPC(I)
C
C **** REMEMBER AS POSSIBLE LAST ELEMENT IN CURRENT ROW.
C
      IEL(3,1,2)=J
C
C **** REMEMBER ROW NUMBER OF THIS ELEMENT.
C
      IC=ITEMP(1)
C
C **** SAVE ELEMENT'S SEQUENCE NUMBER.
C
      N=N+1
      IF (N.GT.NMAX) GO TO 60
      IEL(3,J,3)=N
      IF (ITEMP(1).EQ.NIOUT.AND.J.EQ.NJOUT) MOUT=N
      IF (ITEMP(3).NE.0) RETURN
C
C **** NOW READ PARAMETERS FOR NEXT ELEMENT.
C
      READ (1,100) (ITEMP(K),K=1,7),(ITEMPC(L),L=1,2),(ITEMP(K),K=8,11)
      IF (ITEMP(1).LT.IC.OR.ITEMP(1).GT.IC+1.OR.(ITEMP(2).LE.J.AND.ITEMP
     1(1).EQ.IC)) GO TO 70
      IF (ITEMP(1).EQ.IC) GO TO 30
      RETURN
   50 WRITE (6,110) ITEMP(1),J
      WRITE (2,110) ITEMP(1),J
      STOP
C
C **** ERROR MESSAGES.
C
   60 WRITE (6,120) ITEMP(1),J
      WRITE (2,120) ITEMP(1),J
      STOP
   70 WRITE (6,130) ITEMP(1),ITEMP(2)
      WRITE (2,130) ITEMP(1),ITEMP(2)
      STOP
   80 WRITE (6,140) K,ITEMP(1),J
      WRITE (2,140) K,ITEMP(1),J
      STOP
   90 WRITE (6,150) ITEMP(7),ITEMP(1),J
      WRITE (2,150) ITEMP(7),ITEMP(1),J
      STOP
C
  100 FORMAT (2I3,I2,I3,3I4,3X,A2,1X,A2,2X,I4,I3,2I4)
  110 FORMAT (23H COLUMN NO. FOR ELEMENT,I4,1H,,I4,24H EXCEEDS IEL() DIM
     1ENSION)
  120 FORMAT (45H NO. OF ELEMENTS EXCEEDS DIMENSION AT ELEMENT,I4,1H,,I4
     1)
  130 FORMAT (40H ELEMENT DATA OUT OF SEQUENCE AT ELEMENT,I4,1H,,I4)
  140 FORMAT (1X,9HSOIL TYPE,I4,22H SPECIFIED FOR ELEMENT,I4,1H,,I4,15H
     1IS NOT DEFINED)
  150 FORMAT (1X,9HCROP TYPE,I4,22H SPECIFIED FOR ELEMENT,I4,1H,,I4,15H
     1IS NOT DEFINED)
C
      CLOSE(1)
      CLOSE(2)
      END

      SUBROUTINE HELP
      write(*,*) 'cambiar /File/Properties/Command line/ de ANS77W por:'
      WRITE(*,*) 'ANS77W [archivo entrada] [archivo salida]'
      STOP
      END
```

READANSW.FOR

```
$NOTRUNCATE
$DEBUG
      PROGRAM READANSW

      INCLUDE 'IDRIVAR.TXT'
      INCLUDE 'ANSWVARW.TXT'
      integer*1 errcod,action
      character*6 str6
      CHARACTER*2 MOT2,MOT2B
      write(*,1000)
1000  format(' Lire les données ENTREE (1), ou SORTIE (2)?: '\)
      read(*,*) action


      CALL IDRINIT

      CALL read_env_file

      CALL read_cmd_line_args(cmd_line_options_num,cmd_line_options,
     &cmd_line_files_num,cmd_line_files)

      IF (cmd_line_files_num .NE. 1) THEN
          WRITE(*,*) 'ERREUR: besoin de 1 filière !'
          CALL HELP
          STOP
      END IF


      INFO(1)=' '
      INFO(2)=drive
      INFO(3)=path
      INFO(5)='.ANS'

      init_file=cmd_line_files(1)
      INFO(4)=init_file
      CALL COLLE(5,INFO)
      init_file_file=INFO(1)

      n_files=0
      WRITE(*,*) 'lit la filière initialisation...'
      OPEN(1,FILE=init_file_file)
2     READ(1,108,END=1002) description,files(n_files+1)
      n_files=n_files+1
108   FORMAT(A14,A8)
      write(*,*) '|',FILES(n_files),'|'
      GOTO 2
1002  CLOSE(1)

      IF (n_files .NE. 17) THEN
        WRITE(*,*) 'nb de filières initialisation: il y a ',n_files,
     &  ' filières au lieu de 17..vérifier!'
        STOP
      END IF

      DO I=1,n_files
        INFO(4)=files(I)
        CALL COLLE(5,INFO)
        filenames(I)=INFO(1)
      END DO

c....files: sans le path, filenames: avec le path
      ans_elem=files(2)
      ans_out=files(3)
      idri_soil=files(7)
      idri_slope=files(4)
      idri_aspect=files(5)
      idri_chan_cat=files(6)
      idri_crop_man=files(8)
      idri_gauge=files(9)
      idri_tile=files(10)
      idri_chan_slope=files(11)
      idri_bmpi=files(12)
      idri_bmp1=files(13)
      idri_bmp2=files(14)
      idri_elev=files(15)
      idri_sed=files(16)
      idri_cdep=files(17)

      ans_elem_file=filenames(2)
      ans_out_file=filenames(3)

c...verifier: cete filiere est ouverte 2 fois
      OPEN(1,FILE=ans_elem_file)
      write(*,*) 'lit la filière d'éléments individuels de ANSWERS...'
      READ(1,101) ligne
101   FORMAT(A80)
      ws_name=ligne(29:72)
      READ(1,102) elem_dim,elem_units

c........il faut compter le nombre de canaux...
102   FORMAT(17X,F5.3,1X,A2)
      READ(1,101) ligne

c.........on commence a lire les éléments
      elem_nb=1
      chan_nb=0

      max_row=0
      min_row=10000
      max_col=0
      min_col=10000
1     READ(1,104,END=1001) row_nb(elem_nb),col_nb(elem_nb),tmp_a2
      IF (tmp_A2 .NE. '  ') THEN
        chan_nb=chan_nb+1
      END IF
104   FORMAT(2I3,9x,A2)
      max_row=MAX0(row_nb(elem_nb),max_row)
      max_col=MAX0(col_nb(elem_nb),max_col)
      min_row=MIN0(row_nb(elem_nb),min_row)
      min_col=MIN0(col_nb(elem_nb),min_col)
      elem_nb=elem_nb+1
      GOTO 1
1001  CLOSE(1)

      elem_nb=elem_nb-1

      rows=max_row-min_row+1
      cols=max_col-min_col+1
      IF (elem_units(1:2) .EQ. 'FT') THEN
        ref_units='ft'
      ELSE IF(elem_units(1:1) .EQ. 'M') THEN
        ref_units='m'
      ELSE
        ref_units(1:2)=elem_units(1:2)
      END IF
c       WRITE(*,*) 'ELEM DIM=', elem_dim
      wxmin=(min_col-1)*elem_dim
      wxmax=(max_col)*elem_dim
      wymin=(min_row-1)*elem_dim
      wymax=(max_row)*elem_dim
      resolution=(wxmax-wxmin)/cols
      n_Z=cols*rows

      SELECT CASE(action)
      CASE(1)
C......LECTURE/ECRITURE DES PARAMETRES ELEMENTAIRES...
C..DO I=1,COLS??...
C.......DO I=1,N_Z

      DO I=1,4096
        ZBYTE(I)=0
        ZREAL(I)=0
        ZINT(I)=0
      END DO
c%%%%%%
           DO K=1,cols
             slope(K)=0
             aspect(K)=0
             chan_cat(K)=0
             soil_type(K)=0
             crop_man(K)=0
             gauge(K)=0
             tile(K)=0
             chan_slope(K)=0
             bmpi(K)=0
             bmp1(K)=0
             bmp2(K)=0
             elev(K)=0
             ZINT(K)=0
           END DO
```

```
c$$$$$$

      OPEN(1,FILE=ans_elem_file)
      READ(1,101) ligne
      READ(1,101) ligne
      READ(1,101) ligne

c%%%%%%% ici, on peut avoir un probleme avec pente et aspect REAL
      min_slope=0
      min_aspect=0
      min_chan_cat=0
      min_soil_type=0
      min_crop_man=0
      min_gauge=0
      min_tile=0
      min_chan_slope=0
      min_bmpi=0
      min_bmp1=0
      min_bmp2=0
      min_elev=0
      max_slope=0
      max_aspect=0
      max_chan_cat=0
      max_soil_type=0
      max_crop_man=0
      max_gauge=0
      max_tile=0
      max_chan_slope=0
      max_bmpi=0
      max_bmp1=0
      max_bmp2=0
      max_elev=0


        new_col_pos=0
        new_file_type=1
        new_image=idri_slope
        CALL open_new_image_file(5)
        new_image=idri_aspect
        call open_new_image_file(6)
        new_image=idri_chan_cat
        call open_new_image_file(7)
        new_image=idri_soil
        CALL open_new_image_file(2)
        new_image=idri_crop_man
        CALL open_new_image_file(8)
        new_image=idri_gauge
        call open_new_image_file(9)
        new_image=idri_tile
        CALL open_new_image_file(10)
        new_image=idri_chan_slope
        CALL open_new_image_file(11)
        new_image=idri_bmpi
        call open_new_image_file(12)
        new_image=idri_bmp1
        call open_new_image_file(13)
        new_image=idri_bmp2
        call open_new_image_file(14)
        new_image=idri_elev
        call open_new_image_file(15)

c$$$$$$$
      DO L=1,elem_nb

      READ(1,110,END=1003) ligne
c      write(*,*) ligne
      write(str6,111) ligne(1:3)
111   format(A6)
112   format(I6)
      read(str6,112) J
      write(str6,111) ligne(4:6)
      read(str6,112) I
      I1=I-min_col+1
      J1=(J-min_row)*cols
c      WRITE(*,*) J,I,I1,J1
      write(str6,111) ligne(9:11)
      read(str6,112) slope(I1+J1)
      write(str6,111) ligne(12:15)
      read(str6,112) aspect(I1+J1)
c      write(*,*) slope(I1+J1), aspect(I1+J1)
```

```
if (ligne(16:17) .NE. '  ') then
  write(str6,111) ligne(16:17)
  read(str6,112) chan_cat(I1+J1)
end if
write(str6,111) ligne(18:19)
read(str6,112) soil_type(I1+J1)

if (ligne(20:23) .NE. '    ') then
  write(str6,111) ligne(20:23)
  read(str6,112) crop_man(I1+J1)
end if
write(str6,111) ligne(28:28)
read(str6,112) gauge(I1+J1)

mot2=ligne(30:31)
call upcase(mot2,mot2B)
if (mot2B .EQ. "TI") then
  tile(I1+J1)=1
end if

if (ligne(34:37) .NE. '    ') then
  write(str6,111) ligne(34:37)
  read(str6,112) chan_slope(I1+J1)
else
  chan_slope(I1+J1)=slope(I1+J1)
end if


if (ligne(39:40) .NE. '  ') then
  write(str6,111) ligne(39:40)
  read(str6,112) bmpi(I1+J1)
end if

if (ligne(41:44) .NE. '    ') then
  write(str6,111) ligne(41:44)
  read(str6,112) bmp1(I1+J1)
end if

if (ligne(45:48) .NE. '    ') then
  write(str6,111) ligne(45:48)
  read(str6,112) bmp2(I1+J1)
end if

if (ligne(64:70) .NE. '       ') then
  write(str6,111) ligne(64:70)
  read(str6,112) elev(I1+J1)
end if
END DO

write(*,*) 'Ecriture des images...'
DO L=1,n_z
      min_slope=AMIN1(min_slope,slope(L))
      max_slope=AMAX1(max_slope,slope(L))
      min_aspect=AMIN1(min_aspect,aspect(L))
      max_aspect=AMAX1(max_aspect,aspect(L))
      min_chan_cat=AMIN1(min_chan_cat,chan_cat(L))
      max_chan_cat=AMAX1(max_chan_cat,chan_cat(L))
      min_soil_type=AMIN1(min_soil_type,soil_type(L))
      max_soil_type=AMAX1(max_soil_type,soil_type(L))
      min_crop_man=AMIN1(min_crop_man,crop_man(L))
      max_crop_man=AMAX1(max_crop_man,crop_man(L))
      min_gauge=AMIN1(min_gauge,gauge(L))
      max_gauge=AMAX1(max_gauge,gauge(L))
      min_tile=AMIN1(min_tile,tile(L))
      max_tile=AMAX1(max_tile,tile(L))
      min_chan_slope=AMIN1(min_chan_slope,chan_slope(L))
      max_chan_slope=AMAX1(max_chan_slope,chan_slope(L))
      min_bmpi=AMIN1(min_bmpi,bmpi(L))
      max_bmpi=AMAX1(max_bmpi,bmpi(L))
      min_bmp1=AMIN1(min_bmp1,bmp1(L))
      max_bmp1=AMAX1(max_bmp1,bmp1(L))
      min_bmp2=AMIN1(min_bmp2,bmp2(L))
      max_bmp2=AMAX1(max_bmp2,bmp2(L))
      min_elev=AMIN1(min_elev,elev(L))
      max_elev=AMAX1(max_elev,elev(L))
  END DO

  DO J=1,rows
      J0=(J-1)*cols+1
      DO I=J0,J0+cols-1
```

```
        ZINT(I-J0+1)=slope(I)
      END DO
      new_data_type=0
      CALL fill_Z_data(new_data_type,cols,ZINT,ZREAL,ZBYTE)
      CALL write_image_data(5,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
&new_file_type,new_data_type)
      new_col_pos=0

      DO I=J0,J0+cols-1
        ZINT(I-J0+1)=aspect(I)
      END DO
      new_data_type=0
      CALL fill_Z_data(new_data_type,cols,ZINT,ZREAL,ZBYTE)
      CALL write_image_data(6,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
&new_file_type,new_data_type)
      new_col_pos=0

      DO I=J0,J0+cols-1
        ZBYTE(I-J0+1)=chan_cat(I)
      END DO
      new_data_type=2
      CALL fill_Z_data(new_data_type,cols,ZINT,ZREAL,ZBYTE)
      CALL write_image_data(7,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
&new_file_type,new_data_type)
      new_col_pos=0

      DO I=J0,J0+cols-1
        ZBYTE(I-J0+1)=soil_type(I)
      END DO
      new_data_type=2
      CALL fill_Z_data(new_data_type,cols,ZINT,ZREAL,ZBYTE)
      CALL write_image_data(2,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
&new_file_type,new_data_type)
      new_col_pos=0

      DO I=J0,J0+cols-1
        ZBYTE(I-J0+1)=crop_man(I)
      END DO
      new_data_type=2
      CALL fill_Z_data(new_data_type,cols,ZINT,ZREAL,ZBYTE)
      CALL write_image_data(8,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
&new_file_type,new_data_type)
      new_col_pos=0

      DO I=J0,J0+cols-1
        ZBYTE(I-J0+1)=gauge(I)
      END DO
      new_data_type=2
      CALL fill_Z_data(new_data_type,cols,ZINT,ZREAL,ZBYTE)
      CALL write_image_data(9,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
&new_file_type,new_data_type)
      new_col_pos=0

      DO I=J0,J0+cols-1
        ZBYTE(I-J0+1)=tile(I)
      END DO
      new_data_type=2
      CALL fill_Z_data(new_data_type,cols,ZINT,ZREAL,ZBYTE)
      CALL write_image_data(10,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
&new_file_type,new_data_type)
      new_col_pos=0

      DO I=J0,J0+cols-1
        ZINT(I-J0+1)=chan_slope(I)
      END DO
      new_data_type=0
      CALL fill_Z_data(new_data_type,cols,ZINT,ZREAL,ZBYTE)
      CALL write_image_data(11,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
&new_file_type,new_data_type)
      new_col_pos=0

      DO I=J0,J0+cols-1
        ZBYTE(I-J0+1)=bmpi(I)
      END DO
      new_data_type=2
      CALL fill_Z_data(new_data_type,cols,ZINT,ZREAL,ZBYTE)
      CALL write_image_data(12,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
&new_file_type,new_data_type)
      new_col_pos=0

      DO I=J0,J0+cols-1
```

```
              ZBYTE(I-J0+1)=bmp1(I)
            END DO
            new_data_type=2
            CALL fill_Z_data(new_data_type,cols,ZINT,ZREAL,ZBYTE)
            CALL write_image_data(13,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     &new_file_type,new_data_type)
            new_col_pos=0

            DO I=J0,J0+cols-1
              ZBYTE(I-J0+1)=bmp2(I)
            END DO
            new_data_type=2
            CALL fill_Z_data(new_data_type,cols,ZINT,ZREAL,ZBYTE)
            CALL write_image_data(14,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     &new_file_type,new_data_type)
            new_col_pos=0

            DO I=J0,J0+cols-1
              ZINT(I-J0+1)=elev(I)
            END DO
            new_data_type=0
            CALL fill_Z_data(new_data_type,cols,ZINT,ZREAL,ZBYTE)
            CALL write_image_data(15,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     &new_file_type,new_data_type)
            new_col_pos=0

        END DO


        CLOSE(2)
        CLOSE(5)
        CLOSE(6)
        CLOSE(7)
        CLOSE(8)
        CLOSE(9)
        CLOSE(10)
        CLOSE(11)
        CLOSE(12)
        CLOSE(13)
        CLOSE(14)
        CLOSE(15)

        info(2)=ws_name


c..ici on a besoin des fichiers sans extension (arranger ca plus haut)
c... changer val_units au lieu de titre, faire legendes

        new_image=idri_slope
        min=min_slope
        max=max_slope
        info(1)=' '
        info(2)=ws_name
        info(3)=': SLOPE'
        call colle(3,info)
        title=info(1)
        val_units='x0.1%'
        legend=0
        legend_text(1)=' '
        new_data_type=0
        CALL create_new_documentation_file

        new_image=idri_aspect
        min=min_aspect
        max=max_aspect
        info(1)=' '
        info(2)=ws_name
        info(3)=': ASPECT'
        call colle(3,info)
        title=info(1)
        val_units='deg'
        legend=0
        legend_text(1)=' '
        new_data_type=0
        CALL create_new_documentation_file

        new_image=idri_chan_cat
        min=min_chan_cat
        max=max_chan_cat
        info(1)=' '
        info(2)=ws_name
```

```
       info(3)=': CHANNEL CATEGORY'
       call colle(3,info)
       title=info(1)
       val_units='category'
       legend=0
       legend_text(1)=' '
       new_data_type=2
       CALL create_new_documentation_file

       new_image=idri_soil
       min=min_soil_type
       max=max_soil_type
       info(1)=' '
       info(2)=ws_name
       info(3)=': SOIL TYPE'
       call colle(3,info)
       title=info(1)
       val_units='category'
       legend=0
       legend_text=' '
C         legend=int(max-min)
C         legend_text(1)=' '
C         do l=int(min),int(max)
C           write(legend_text(L+1),2000) L
C2000      format(I2)
C           end do
       new_data_type=2
       CALL create_new_documentation_file

       new_image=idri_crop_man
       min=min_crop_man
       max=max_crop_man
       info(1)=' '
       info(2)=ws_name
       info(3)=': CROP MANAGEMENT'
       call colle(3,info)
       title=info(1)
       val_units='category'
       legend=0
       legend_text(1)=' '
       new_data_type=2
       CALL create_new_documentation_file

       new_image=idri_gauge
       min=min_gauge
       max=max_gauge
       info(1)=' '
       info(2)=ws_name
       info(3)=': RAIN GAUGES'
       call colle(3,info)
       title=info(1)
       val_units='gauge #'
       legend=0
       legend_text(1)=' '
       new_data_type=2
       CALL create_new_documentation_file

       new_image=idri_tile
       min=min_tile
       max=max_tile
       info(1)=' '
       info(2)=ws_name
       info(3)=': TILAGE'
       call colle(3,info)
       title=info(1)
       val_units='boolean'
       legend=0
       legend_text(1)=' '
       new_data_type=2
       CALL create_new_documentation_file

       new_image=idri_chan_slope
       min=min_chan_slope
       max=max_chan_slope
       info(1)=' '
       info(2)=ws_name
       info(3)=': CHANNEL SLOPE'
       call colle(3,info)
       title=info(1)
       val_units='x0.1%'
       legend=0
```

```
        legend_text(1)=' '
        new_data_type=0
        CALL create_new_documentation_file

        new_image=idri_bmpi
        min=min_bmpi
        max=max_bmpi
        info(1)=' '
        info(2)=ws_name
        info(3)=': BMP ID'
        call colle(3,info)
        title=info(1)
        val_units='ID'
        legend=0
        legend_text(1)=' '
        new_data_type=2
        CALL create_new_documentation_file

        new_image=idri_bmp1
        min=min_bmp1
        max=max_bmp1
        info(1)=' '
        info(2)=ws_name
        info(3)=': BMP #1'
        call colle(3,info)
        title=info(1)
        val_units='category'
        legend=0
        legend_text(1)=' '
        new_data_type=2
        CALL create_new_documentation_file

        new_image=idri_bmp2
        min=min_bmp2
        max=max_bmp2
        info(1)=' '
        info(2)=ws_name
        info(3)=': BMP #2'
        call colle(3,info)
        title=info(1)
        val_units='category'
        legend=0
        legend_text(1)=' '
        new_data_type=2
        CALL create_new_documentation_file

        new_image=idri_elev
        min=min_elev
        max=max_elev
        info(1)=' '
        info(2)=ws_name
        info(3)=': ELEVATION'
        call colle(3,info)
        title=info(1)
c...ici on presuppose que si le systeme de reference est en
c...systeme anglais, les elevations le sont aussi
        val_units=ref_units
        legend=0
        legend_text(1)=' '
        new_data_type=0
        CALL create_new_documentation_file


c$$$$$$$$$$$$$$$
c.....[verifier si il peut y avoit plus de 9 types de sols.(??)]
c....16=channel, 18=crop
c--------109   FORMAT(2I3,2X,I3,1X,I3,2X,I2)
110   FORMAT(A80)
1003  CLOSE(1)
c...enlever le stop pour continuer la lecture /ecriture des
c...fichiers de sedimentation....
c      STOP==============================

      CASE(2)
c*******initialiser legende
      call initlegend
      elem_nb=elem_nb-1

c....initialise, de telle sorte que si sed(i)=0. ou cdep(i)=0., on
c....puisse le voir quand même dans idrisi.
       do i=1,elem_nb
```

```
        sed(i)=0
        cdep(i)=0
      end do

      OPEN(1,FILE=ans_out_file)
      write(*,*) 'lit la filière de sortie de ANSWERS...'
      DO WHILE
     &(ligne(20:55) .NE. 'INDIVIDUAL ELEMENT NET SEDIMENTATION')
      READ(1,101) ligne
c      write(*,*) ligne
      END DO

      READ(1,101) ligne
      READ(1,101) ligne

      N=1
      elem_nb_m4=MOD(elem_nb,4)
      elem_nb_4=elem_nb-elem_nb_m4
c      WRITE(*,*) elem_nb,elem_nb_4,elem_nb_m4
      DO WHILE (N .LT. elem_nb_4)
        READ(1,103) N,sed(N),N,sed(N),N,sed(N),N,sed(N)
c       WRITE(*,103) N-3,sed(N-3),N-2,sed(N-2),N-1,sed(N-1),N,sed(N)
103   FORMAT(1X,4(I7,F11.0))
      END DO
c      WRITE(*,*) 'SORT DE LA BOUCLE'
c      WRITE(*,*) 'elem_nb_m4:',elem_nb_m4

      SELECT CASE(elem_nb_m4)
      CASE(1)
        READ(1,105) N,sed(N)
105   FORMAT(1X,1(I7,F11.0))
      CASE(2)
C....... write(*,*) 'CASE(2) OK..'
        READ(1,106) N,sed(N),N,sed(N)
106   FORMAT(1X,2(I7,F11.0))
      CASE(3)
        READ(1,107) N,sed(N),N,sed(N),N,sed(N)
107   FORMAT(1X,3(I7,F11.0))
      END SELECT

C...... WRITE(*,*) 'DERNIER CANAL:',N
c.......         PAUSE

C......LIT LA DEPOSITION DANS CANAUX...

      READ(1,101) ligne
      READ(1,101) ligne
      READ(1,101) ligne
      READ(1,101) ligne
      READ(1,101) ligne

      N1=1
      chan_nb_m4=MOD(chan_nb,4)
      chan_nb_4=chan_nb-chan_nb_m4
      DO WHILE (N1 .LT. chan_nb_4)
        READ(1,103) N,cdep(N),N,cdep(N),N,cdep(N),N,cdep(N)
        N1=N1+4
c.........         write(*,*) N
      END DO

      SELECT CASE(chan_nb_m4)
      CASE(1)
        READ(1,105) N,cdep(N)
      CASE(2)
        READ(1,106) N,cdep(N),N,cdep(N)
      CASE(3)
        READ(1,107) N,cdep(N),N,cdep(N),N,cdep(N)
      END SELECT
      WRITE(*,*) chan_nb,N
C.....  PAUSE


      CLOSE(1)

c........ecriture des images...

      min_sed=1E10
      max_sed=-1e10
      min_cdep=1E10
      max_cdep=-1e10
      DO N=1,elem_nb
```

```
        min_sed=AMIN1(sed(N),min_sed)
        max_sed=AMAX1(sed(N),max_sed)
        min_cdep=AMIN1(cdep(N),min_cdep)
        max_cdep=AMAX1(cdep(N),max_cdep)
      END DO

C****

      N0=1

      write(*,*) 'place les cellules dans Z...attendez un peu...'
      min_max_flag=0
      DO J=min_row,max_row
        compteur=10*INT1(10*SQRT(REAL(J)/rows))
        SELECT CASE(compteur)
        CASE(10)
        WRITE(*,*) '...10% terminé...'
        CASE(20)
        WRITE(*,*) '...20% terminé...'
        CASE(30)
        WRITE(*,*) '...30% terminé...'
        CASE(40)
        WRITE(*,*) '...40% terminé...'
        CASE(50)
        WRITE(*,*) '...50% terminé...'
        CASE(60)
        WRITE(*,*) '...60% terminé...'
        CASE(70)
        WRITE(*,*) '...70% terminé...'
        CASE(80)
        WRITE(*,*) '...80% terminé...'
        CASE(90)
        WRITE(*,*) '...90% terminé...'
        END SELECT
        DO I=min_col,max_col
          K=(J-min_row)*cols+(I-min_col+1)
          DO N=N0,elem_nb
c           write(*,*) N,col_nb(N),row_nb(N)
            IF ((I .EQ. col_nb(N)) .AND. (J .EQ. row_nb(N))) THEN
              ZREAL3(K)=sed(N)
              ZREAL2(K)=cdep(N)
              N0=N0+1
            EXIT
            ELSE
              ZREAL3(K)=0
              ZREAL2(K)=0
              IF (min_max_flag .EQ. 0) THEN
                sed_min=AMIN1(sed_min,0)
                sed_max=AMAX1(sed_max,0)
                cdep_min=AMIN1(cdep_min,0)
                cdep_max=AMAX1(cdep_max,0)
                min_max_flag=1
              END IF
c           write(*,*) ZREAL3(K),ZREAL2(K)
            END IF
          END DO
        END DO
      END DO

      WRITE(*,*) ' '
      WRITE(*,*) 'écrit les données...'
C.....................
      new_image=idri_sed
      min=min_sed
      max=max_sed
      new_file_type=0
c..........data type =2 originalement
      new_data_type=1
      info(1)=' '
      info(2)=ws_name
      info(3)=': SEDIMENT CONCENTRATION'
      call colle (3,info)
      title=info(1)
      val_units='mg/l'

      CALL create_new_documentation_file
C      CALL fill_Z_data(1,n_Z,ZINT,ZREAL,ZBYTE)
      CALL open_new_image_file(4)
      do j=min_row,max_row
        do i=min_col,max_col
          ZREAL(i-min_col+1)=ZREAL3((j-min_row)*cols+(i-min_col+1))
```

```
c.....          write(*,*) ZREAL(i-min_col)
        end do
        new_col_pos=0
        CALL write_image_data(4,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     &new_file_type,new_data_type)
      end do
      CLOSE(4)
C....................
      new_image=idri_cdep
      min=min_cdep
      max=max_cdep
c..pour raslnvec.exe, il faut des entiers...
c..peur etre appeler ce programme d'ici?
      new_data_type=1
      info(1)=' '
      info(2)=ws_name
      info(3)=': CHANNEL DEPOSITION'
      call colle (3,info)
      title=info(1)
      val_units='kg'

      CALL create_new_documentation_file
C      CALL fill_Z_data(1,n_Z,ZINT,ZREAL2,ZBYTE)
      CALL open_new_image_file(4)
      do j=min_row,max_row
        do i=min_col,max_col
          ZREAL(i-min_col+1)=ZREAL2((j-min_row)*cols+(i-min_col+1))
        end do
        new_col_pos=0
        CALL write_image_data(4,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     &new_file_type,new_data_type)
      end do
      CLOSE(4)
C...........FIN DU PROGRAMME!............
      END SELECT
      END

      SUBROUTINE help
      WRITE(*,*) 'syntaxe: readans [-option -option..] initfile'
c     &predata ansout sedimg chanimg'
      WRITE(*,*) 'initfile = filière initialisation de READANS'
c
c      WRITE(*,*) 'predata = filière predata de ANSWERS'
c      WRITE(*,*) 'ansout = filière sortie de ANSWERS'
c      WRITE(*,*) 'sedimg = filière sédiments IDRISI'
C........WRITE(*,*) 'chanimg = filière "channel deposition" IDRISI'
      WRITE(*,*) 'options: ? -> écrit ce message d`aide'
      END

      SUBROUTINE initlegend
      INCLUDE 'IDRIVAR.TXT'
      legend=0
      do l=1,256
       legend_text(l)=' '
      end do
      END
```

```
$NOTRUNCATE
$DEBUG
      PROGRAM WRITANS

      INCLUDE 'IDRIVAR.TXT'
      INCLUDE 'ANSWVAR.TXT'
      character*6 str6
      character*4 str4
      character*2 str2
      character*1 str1
      CHARACTER*2 MOT2,MOT2B
      character*79 ligneblanche
      INTEGER*2 outflow_row, outflow_col,elem_written
      integer*1 slope_data_type,aspect_data_type,chan_cat_data_type,
     &soil_data_type,crop_man_data_type,gauge_data_type,tile_data_type,
     &chan_slope_data_type,bmpi_data_type,bmp1_data_type,
     &bmp2_data_type,elev_data_type,elem_nb_data_type
      integer*1 slope_file_type,aspect_file_type,chan_cat_file_type,
     &soil_file_type,crop_man_file_type,gauge_file_type,tile_file_type,
     &chan_slope_file_type,bmpi_file_type,bmp1_file_type,
     &bmp2_file_type,elev_file_type,elem_nb_file_type
      integer*2 first_col(4096),last_col(4096)
      REAL*4 resolx, resoly
      integer*1 flag

      WRITE(*,1000)
1000  format(' Nom de la cuenca (43 car. max.)? '\)
      READ(*,*) ws_name
      WRITE(*,1001)
1001  format(' Pixel de outflow (ligne,col)? '\)
      READ(*,*) outflow_row, outflow_col

      CALL read_env_file

      CALL read_cmd_line_args(cmd_line_options_num,cmd_line_options,
     &cmd_line_files_num,cmd_line_files)

      IF (cmd_line_files_num .NE. 1) THEN
         WRITE(*,*) 'ERREUR: besoin de 1 filière !'
         CALL HELP
         STOP
      END IF

      INFO(1)=' '
      INFO(2)=drive
      INFO(3)=path
      INFO(5)='.ANS'

      init_file=cmd_line_files(1)
      INFO(4)=init_file
      CALL COLLE(5,INFO)
      init_file_file=INFO(1)

      n_files=0
      WRITE(*,*) 'lit la filière initialisation...'
      OPEN(1,FILE=init_file_file)
2     READ(1,108,END=1002) description,files(n_files+1)
      n_files=n_files+1
108   FORMAT(A14,A8)
      write(*,*) '|',FILES(n_files),'|'
      GOTO 2
1002  CLOSE(1)

      IF (n_files .NE. 18) THEN
        WRITE(*,*) 'nb de filières initialisation: il y a ',n_files,
     &  ' filières au lieu de 18..vérifier!'
        STOP
      END IF

      DO I=1,n_files
        INFO(4)=files(I)
        CALL COLLE(5,INFO)
        filenames(I)=INFO(1)
      END DO

c....files: sans le path, filenames: avec le path
      ans_elem=files(2)
      ans_out=files(3)
      idri_soil=files(7)
      idri_slope=files(4)
      idri_aspect=files(5)
      idri_chan_cat=files(6)
      idri_crop_man=files(8)
      idri_gauge=files(9)
      idri_tile=files(10)
      idri_chan_slope=files(11)
      idri_bmpi=files(12)
      idri_bmp1=files(13)
      idri_bmp2=files(14)
      idri_elev=files(15)
      idri_sed=files(16)
      idri_cdep=files(17)


      ans_elem_file=filenames(2)
      ans_out_file=filenames(3)
      elem_nb_file=filenames(18)

      old_image=idri_soil
      call read_documentation_file
        resolx=(wxmax-wxmin)/cols
        resoly=(wymax-wymin)/rows
        if (resolx .NE. resoly) then
          write(*,*) 'ERREUR: le pixel n'est pas carré!..Vérifier!'
          stop
c       else
c         resolution=resolx
        end if
      n_z=rows*cols
      do i=1,cols
        ZREAL(i)=0
        ZINT(i)=0
        ZBYTE(i)=0
      end do

c....calculer les limites de chaque ligne
c....ouvrir fichier pour ecrire elem nb.
      open(18, file=elem_nb_file)
      call open_old_image_file(2)
      new_col_pos=0
c++++++++++++++++++++++++++++++++
      write(*,*) 'calcule les limites de la cuenca....'
      elem_nb=0
      do J=1,rows
        CALL read_image_data(2,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     &  old_file_type,old_data_type)
        new_col_pos=0
        flag=0
        do I=1,cols
          if (ZBYTE(I) .NE. 0) then
            elem_nb=elem_nb+1
            write(18,*) elem_nb,I,J
          end if
        end do
      end do
      CLOSE(2)
      close(18)
c=========================
c       CALL read_image_data(2,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
c     &        old_file_type,old_data_type)
c       new_col_pos=0
c       flag=0
c       do I=1,cols
c       if (ZBYTE(I) .NE. 0) then
c         if (flag .EQ. 0) then
c           first_col(J)=I
c           flag=1
c         else
c         if(I .EQ. cols) then
c          last_col(J)=cols
c         end if
c         end if
c       else
c         if (flag .EQ. 1) then
c         last_col(J)=I-1
c         flag=0
c         end if
c       end if
c       end do
c      end do
c      CLOSE(2)
c=========================
      do i=1,79
        ligneblanche(i:i)=' '
      end do

c...ecriture des trois premieres lignes
      OPEN(1,FILE=ans_elem_file)
      ligne=ligneblanche
      ligne(1:28)='  ELEMENT SPECIFICATIONS FOR'
      ligne(29:72)=ws_name

      WRITE(1,101) ligne
101   FORMAT(A79)

c.....ca ne marche pas...
      ligne=ligneblanche
      write(str6,'(F6.1)') resolution
      read(str6,'(A6)') ligne(17:22)
      ligne(1:16)=' EACH ELEMENT IS'
      ligne(23:23)=' '
      str2='  '
      call upcase(ref_units,str2)
      ligne(24:25)=str2
      ligne(26:33)='. SQUARE'
      WRITE(1,101) ligne

      ligne=ligneblanche
      ligne(1:17)=' OUTFLOW FROM ROW'
      ligne(22:29)=' COLUMN '
      WRITE(str4,'(I4)') outflow_row
      READ(str4,'(A4)') ligne(18:21)
      WRITE(str4,'(I4)') outflow_col
      READ(str4,'(A4)') ligne(30:33)
      WRITE(1,101) ligne

      write(*,*) 'ouverture des fichiers...'
        old_image=idri_slope
        call read_documentation_file
        slope_file_type=old_file_type
        slope_data_type=old_data_type
c       write(*,*) 'slope',old_data_type
        IF (slope_data_type .NE. 0) then
        write(*,*) 'El archivo ',idri_slope,
     &             ' no esta en formato INTEGER..Utilizer CONVERT'
        stop
        end if
        CALL open_old_image_file(5)

        old_image=idri_aspect
        call read_documentation_file
        aspect_file_type=old_file_type
        aspect_data_type=old_data_type
c       write(*,*) 'aslect',old_data_type
        IF (aspect_data_type .NE. 0) then
        write(*,*) 'El archivo ',idri_aspect,
     &             ' no esta en formato INTEGER..Utilizer CONVERT'
        stop
        end if
        call open_old_image_file(6)

        old_image=idri_chan_cat
        call read_documentation_file
        chan_cat_file_type=old_file_type
        chan_cat_data_type=old_data_type
c       write(*,*) 'chan cat',old_data_type
        IF (chan_cat_data_type .NE. 2) then
        write(*,*) 'El archivo ',idri_chan_cat,
     &             'no esta en formato BYTE..Utilizer CONVERT'
        stop
        end if
        call open_old_image_file(7)

        old_image=idri_soil
        call read_documentation_file
        soil_file_type=old_file_type
        soil_data_type=old_data_type
c       write(*,*) 'soil',old_data_type
        IF (soil_data_type .NE. 2) then
        write(*,*) 'El archivo ',idri_soil,
     &             'no esta en formato BYTE..Utilizer CONVERT'
        stop
        end if
      CALL open_old_image_file(2)

      old_image=idri_crop_man
      call read_documentation_file
      crop_man_file_type=old_file_type
      crop_man_data_type=old_data_type
c     write(*,*) 'crop man',old_data_type
      IF (crop_man_data_type .NE. 2) then
      write(*,*) 'El archivo ',idri_crop_man,
     &            'no esta en formato BYTE..Utilizer CONVERT'
      stop
      end if
      CALL open_old_image_file(8)

      old_image=idri_gauge
      call read_documentation_file
      gauge_file_type=old_file_type
      gauge_data_type=old_data_type
c     write(*,*) 'gauge',old_data_type
      IF (gauge_data_type .NE. 2) then
      write(*,*) 'El archivo ',idri_gauge,
     &            'no esta en formato BYTE..Utilizer CONVERT'
      stop
      end if
      call open_old_image_file(9)

      old_image=idri_tile
      tile_file_type=old_file_type
      tile_data_type=old_data_type
c     write(*,*) 'tile',old_data_type
      IF (tile_data_type .NE. 2) then
      write(*,*) 'El archivo ',idri_tile,
     &            'no esta en formato BYTE..Utilizer CONVERT'
      stop
      end if
      CALL open_old_image_file(10)

      old_image=idri_chan_slope
      call read_documentation_file
      chan_slope_file_type=old_file_type
      chan_slope_data_type=old_data_type
c     write(*,*) 'chan slope',old_data_type
      IF (chan_slope_data_type .NE. 0) then
      write(*,*) 'El archivo ',idri_chan_slope,
     &            ' no esta en formato INTEGER..Utilizer CONVERT'
      stop
      end if
      CALL open_old_image_file(11)

      old_image=idri_bmpi
      call read_documentation_file
      bmpi_file_type=old_file_type
      bmpi_data_type=old_data_type
c     write(*,*) 'bmpi',old_data_type
      IF (bmpi_data_type .NE. 2) then
      write(*,*) 'El archivo ',idri_bmpi,
     &            ' no esta en formato BYTE..Utilizer CONVERT'
      stop
      end if
      call open_old_image_file(12)

      old_image=idri_bmp1
      call read_documentation_file
      bmp1_file_type=old_file_type
      bmp1_data_type=old_data_type
c     write(*,*) 'bmp1',old_data_type
      IF (bmp1_data_type .NE. 2) then
      write(*,*) 'El archivo ',idri_bmp1,
     &            ' no esta en formato BYTE..Utilizer CONVERT'
      stop
      end if
      call open_old_image_file(13)

      old_image=idri_bmp2
      call read_documentation_file
      bmp2_file_type=old_file_type
      bmp2_data_type=old_data_type
c     write(*,*) 'bmp2',old_data_type
      IF (bmp2_data_type .NE. 2) then
      write(*,*) 'El archivo ',idri_bmp2,
     &            ' no esta en formato BYTE..Utilizer CONVERT'
```

```
       stop
       end if
       call open_old_image_file(14)

       old_image=idri_elev
       call read_documentation_file
       elev_file_type=old_file_type
       elev_data_type=old_data_type
c      write(*,*) 'elev',old_data_type
       IF (elev_data_type .NE. 0) then
       write(*,*) 'El archivo ',idri_elev,
     &            ' no esta en formato INTEGER..Utilizer CONVERT'
       stop
       end if
       call open_old_image_file(15)
c=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
      write(*,*) 'lecture des images...'
      new_col_pos=0
      endflag=0
      elem_written=0
      do J=1,rows
       ligne=ligneblanche
c      write(*,*) 'slope'
       old_file_type=slope_file_type
       old_data_type=slope_data_type
       CALL read_image_data(5,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     & old_file_type,old_data_type)
       new_col_pos=0
       do I=1,cols
         slope(I)=ZINT(I)
       end do

c      write(*,*) 'aspect'
       old_file_type=aspect_file_type
       old_data_type=aspect_data_type
       CALL read_image_data(6,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     & old_file_type,old_data_type)
       new_col_pos=0
       do I=1,cols
         aspect(I)=ZINT(I)
       end do

c      write(*,*) 'chan. cat.'
       old_file_type=chan_cat_file_type
       old_data_type=chan_cat_data_type
       CALL read_image_data(7,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     & old_file_type,old_data_type)
       new_col_pos=0
       do I=1,cols
         chan_cat(I)=ZBYTE(I)
       end do


c      write(*,*) 'soil'
       old_file_type=soil_file_type
       old_data_type=soil_data_type
       CALL read_image_data(2,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     & old_file_type,old_data_type)
       new_col_pos=0
       do I=1,cols
         soil_type(I)=ZBYTE(I)
       end do

c      write(*,*) 'crop man.'
       old_file_type=crop_man_file_type
       old_data_type=crop_man_data_type
       CALL read_image_data(8,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     & old_file_type,old_data_type)
       new_col_pos=0
       do I=1,cols
         crop_man(I)=ZBYTE(I)
       end do

c      write(*,*) 'gauge'
       old_file_type=gauge_file_type
       old_data_type=gauge_data_type
c      write(*,*) 'old data type=',old_data_type
       CALL read_image_data(9,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     & old_file_type,old_data_type)
       new_col_pos=0
       do I=1,cols
```

```
        gauge(I)=ZBYTE(I)
      end do
c     write(*,*) gauge(15)

c     write(*,*) 'tile'
      old_file_type=tile_file_type
      old_data_type=tile_data_type
c     write(*,*) 'old data type =',old_data_type
      CALL read_image_data(10,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     & old_file_type,old_data_type)
      new_col_pos=0
      do I=1,cols
        tile(I)=ZBYTE(I)
      end do
c     write(*,*) tile(15)

c     write(*,*) 'chan. slope'
      old_file_type=chan_slope_file_type
      old_data_type=chan_slope_data_type
      CALL read_image_data(11,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     & old_file_type,old_data_type)
      new_col_pos=0
      do I=1,cols
        chan_slope(I)=ZINT(I)
      end do
c     write(*,*) chan_slope(I)

c     write(*,*) 'bmpi'
      old_file_type=bmpi_file_type
      old_data_type=bmpi_data_type
      CALL read_image_data(12,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     & old_file_type,old_data_type)
      new_col_pos=0
      do I=1,cols
        bmpi(I)=ZBYTE(I)
      end do

c     write(*,*) 'bmp1'
      old_file_type=bmp1_file_type
      old_data_type=bmp1_data_type
      CALL read_image_data(13,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     & old_file_type,old_data_type)
      new_col_pos=0
      do I=1,cols
        bmp1(I)=ZBYTE(I)
      end do

c     write(*,*) 'bmp2'
      old_file_type=bmp2_file_type
      old_data_type=bmp2_data_type
      CALL read_image_data(14,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     & old_file_type,old_data_type)
      new_col_pos=0
      do I=1,cols
        bmp2(I)=ZBYTE(I)
      end do


c     write(*,*) 'elev'
      old_file_type=elev_file_type
      old_data_type=elev_data_type
      CALL read_image_data(15,new_col_pos,cols,ZINT,ZREAL,ZBYTE,
     & old_file_type,old_data_type)
      new_col_pos=0
      do I=1,cols
         elev(I)=ZINT(I)
      end do

c     write(*,*) 'ecriture de la ligne....'
c     do I=first_col(J),last_col(J)
      do I=1,cols
        if(soil_type(I) .NE. 0) then
        write(str6,112) J
        read(str6,111) ligne(1:3)
        write(str6,112) I
        read(str6,111) ligne(4:6)
        write(str6,112) slope(I)
        read(str6,111) ligne(9:11)
        write(str6,112) aspect(I)
        read(str6,111) ligne(12:15)
        if (chan_cat(I) .EQ. 0) then
```

```
            ligne(16:17)='  '
          else
            write(str6,112) chan_cat(I)
            read(str6,111) ligne(16:17)
            ligne(18:18)='0'
          end if
          write(str6,112) soil_type(I)
          read(str6,111) ligne(19:19)
          write(str6,112) crop_man(I)
          read(str6,111) ligne(20:23)
          ligne(27:27)='R'
          write(str1,114) gauge(I)
          read(str1,113) ligne(28:28)
          if (tile(I) .EQ. 1) then
            ligne(30:33)='TILE'
          else
            ligne(30:33)='    '
          end if
          if(chan_slope(I) .LE. 0) then
            ligne(34:37)='    '
          else
            write(str6,112) chan_slope(I)
            read(str6,111) ligne(34:37)
          end if
          if(bmpi(I) .EQ. 0) then
            ligne(39:40)='  '
          else
            write(str6,112) bmpi(I)
            read(str6,111) ligne(39:40)
          end if
          if (bmp1(I) .EQ. 0) then
            ligne(41:44)='    '
          else
            write(str6,112) bmp1(I)
            read(str6,111) ligne(41:44)
          end if
          if (bmp2(I) .EQ. 0) then
            ligne(45:48)='    '
          else
            write(str6,112) bmp2(I)
            read(str6,111) ligne(45:48)
          end if
          if (elev(I) .EQ. 0) then
            ligne(64:70)='       '
          else
            write(str6,112) elev(I)
            read(str6,111) ligne(64:70)
          end if
          if (elem_written .EQ. elem_nb-1) then
              ligne(8:8)='9'
          else
              ligne(8:8)=' '
          end if
          write(1,101) ligne
          elem_written=elem_written+1
        end if
111   format(a6)
112   format(I6)
113   format(a1)
114   format(I1)
        end do
      end do
        CLOSE(1)
        CLOSE(2)
        CLOSE(5)
        CLOSE(6)
        CLOSE(7)
        CLOSE(8)
        CLOSE(9)
        CLOSE(10)
        CLOSE(11)
        CLOSE(12)
        CLOSE(13)
        CLOSE(14)
        CLOSE(15)
      END

            SUBROUTINE help
      WRITE(*,*) 'syntaxe: writans [-option -option..] initfile'
      WRITE(*,*) 'initfile = filière initialisation de WRITANS'
      WRITE(*,*) 'options: ? -> écrit ce message d'aide'
```

NEWMOD4.FOR

```
$NOTRUNCATE
$DEBUG

C*********************************************************
      SUBROUTINE create_new_documentation_file
      INCLUDE 'IDRIVAR.TXT'

      CHARACTER*80 docname
      INTEGER*2 I,J

      docname=' '
      INFO(1)=docname
      INFO(2)=drive
      INFO(3)=path
      INFO(4)=new_image
      INFO(5)=image_docfile_extension
      CALL COLLE(5,INFO)
      docname=INFO(1)
      OPEN(3,FILE=docname)

      IF((new_data_type .EQ. 2) .AND. (new_file_type .EQ. 0)) THEN
      new_data_type=0
      END IF

      WRITE(3,800,ERR=2002) 'file title  : ',title
800   FORMAT(A14,A66)

      SELECT CASE (new_data_type)
         CASE(0)
           WRITE(3,821,ERR=2002) 'data type   : integer'
821   FORMAT(A21)
818   FORMAT(A18)
         CASE(1)
           WRITE(3,818,ERR=2002) 'data type   : real'
         CASE(2)
           WRITE(3,818,ERR=2002) 'data type   : byte'
      END SELECT


      SELECT CASE (new_file_type)
         CASE(0)
           WRITE(3,819,ERR=2002) 'file type   : ascii'
819   FORMAT(A19)
         CASE(1)
           WRITE(3,820,ERR=2002) 'file type   : binary'
820   FORMAT(A20)
         CASE(2)
           WRITE(3,827,ERR=2002) 'file type   : packed binary'
827   FORMAT(A27)
      END SELECT


      WRITE(3,804,ERR=2002) 'columns     : ',cols
804   FORMAT(A14,I4)
      WRITE(3,804,ERR=2002) 'rows        : ',rows

      WRITE(3,800,ERR=2002) 'ref. system : ',ref_system
      WRITE(3,807,ERR=2002) 'ref. units  : ',ref_units
807   FORMAT(A14,A3)
C  {*** note that both ref_system and ref_units MUST be in LOWER CASE ***}
          WRITE(3,801,ERR=2002) 'unit dist.  : ',unit_dist
          WRITE(3,801,ERR=2002) 'min. X      : ',wxmin
          WRITE(3,801,ERR=2002) 'max. X      : ',wxmax
          WRITE(3,801,ERR=2002) 'min. Y      : ',wymin
          WRITE(3,801,ERR=2002) 'max. Y      : ',wymax

808   FORMAT(A14,I9)
801   FORMAT(A14,1PG32.16)

      IF(posn_error_flag .EQ. 1) THEN
        IF(posn_error .EQ. INT(posn_error)) THEN
               WRITE(3,808,ERR=2002)'pos n error : ',INT(posn_error)
        ELSE
               WRITE(3,801,ERR=2002)'pos n error : ',posn_error
        END IF
      ELSE
          WRITE(3,821,ERR=2002)'pos n error : unknown'
      END IF
      IF(resolution_flag .EQ. 1) THEN
              WRITE(3,801,ERR=2002) 'resolution  : ',resolution
      ELSE
          WRITE(3,821,ERR=2002) 'resolution  : unknown'
      END IF
      IF(new_data_type .EQ. 1) THEN
               WRITE(3,801,ERR=2002) 'min. value   : ',min
      ELSE
          WRITE(3,805,ERR=2002) 'min. value   : ',INT(min)
      END IF
      IF(new_data_type .EQ. 1) THEN
               WRITE(3,801,ERR=2002) 'max. value   : ',max
      ELSE
          WRITE(3,805,ERR=2002) 'max. value   : ',INT(max)
805   FORMAT(A14,I9)
      END IF
      IF(val_units .EQ. ' ') THEN
          WRITE(3,825,ERR=2002) 'value units : unspecified'
825   FORMAT(A25)
      ELSE
          WRITE(3,800,ERR=2002) 'value units : ',val_units
      END IF

C  {*** note that val_units MUST be in LOWER CASE ***}
      IF(val_error_flag .EQ. 1) THEN
               WRITE(3,801,ERR=2002) 'value error : ',val_error
      ELSE
          WRITE(3,821,ERR=2002) 'value error : unknown'
      END IF
      IF(flag_flag .EQ. 1) THEN
        IF (new_data_type .EQ. 1) THEN
               WRITE(3,801,ERR=2002) 'flag value   : ',flag_value
        ELSE
               WRITE(3,805,ERR=2002) 'flag value   : ',INT(flag_value)
        END IF
      ELSE
          WRITE(3,818,ERR=2002) 'flag value   : none'
      END IF

      IF (flag_defn .EQ. ' ') THEN
          WRITE(3,818,ERR=2002) 'flag def n   : none'
      ELSE
          WRITE(3,800,ERR=2002) 'flag def n   : ',flag_defn
      END IF


      WRITE(3,806,ERR=2002)'legend cats : ',legend
806   FORMAT(A14,I3)
      IF (legend .NE. 0) THEN
       IF (legend .GT. 255) THEN
           legend=255
       END IF
       DO 1 i=1,legend
             J=i-1
             WRITE(3,802,ERR=2002) 'category',J,' : ',legend_text(i)
1      CONTINUE
      END IF
802   FORMAT(A8,I3,A3,A66)
      CLOSE(3)
      RETURN
2002  WRITE(*,*) '** erreur d`écriture: documentation file **'
      CLOSE(3)

      END

C*******************************************************
      SUBROUTINE IDRINIT

      INCLUDE 'IDRIVAR.TXT'

      ref_units='m'
      val_units= 'default'
      title='default'
      ref_system='plane'
      flag_defn='background'
      val_recs=5
      rows=10
      cols=10
      val_fields=2
      val_file_type=0
      legend=1
      data_format=0
      geo_type=0
      old_data_type=2
      old_file_type=1
      new_data_type=2
      new_file_type=1
      wxmin=0
      wxmax=10
      wymin=0
      wymax=10
      unit_dist=1
      flag_value=0
      resolution=10
      cellx=1
      celly=1
      posn_error=0
      val_error=10
      min=0
      max=10
      posn_error_flag=0
      resolution_flag=0
      val_error_flag=0
      flag_flag=1
      DO 1 I=1,256
      legend_text(I)=' '
1     CONTINUE
      legend_text(1)='default '
      path=' '
      digi_port='COM2'
      plot_port='COM1'
      prn_port='LPT1'
      drive=' '
      units='m'
      old_image='default'
      new_image='default'
      image_docfile_extension='.doc'
      image_file_extension='.img'
      vector_docfile_extension='.vct'
      vector_file_extension='.dvc'
      values_docfile_extension='.dvl'
      values_file_extension='.avl'
      DO 2 I=1,10
      INFO(I)=' '
2     CONTINUE

      END

C*************************************************
      SUBROUTINE read_documentation_file
      INCLUDE 'IDRIVAR.TXT'

      CHARACTER*80 docname
      CHARACTER*14 description
      INTEGER*2 i,err_code
      CHARACTER*66 tmpstr,tmpstr2

      docname=' '
      INFO(1)=docname
      INFO(2)=drive
      INFO(3)=path
      INFO(4)=old_image
      INFO(5)=image_docfile_extension

      CALL COLLE(5,INFO)
      docname=INFO(1)
      OPEN(3,FILE=docname)
      READ(3, 800,ERR=2001) description,tmpstr
      title=tmpstr
800   FORMAT(A14,A66)
      READ(3,800,ERR=2001) description,tmpstr
      IF (tmpstr(1:7) .EQ. 'integer') THEN
            old_data_type=0
      ELSE IF(tmpstr(1:4) .EQ. 'real') THEN
            old_data_type=1
      ELSE IF (tmpstr(1:4) .EQ. 'byte') THEN
            old_data_type=2
      ELSE IF (tmpstr(1:4) .EQ. 'word') THEN
            old_data_type=3
      ELSE
            old_data_type=999
      END IF

      IF(old_data_type .GT. 3) THEN
```

```
             WRITE(*,830) 'Error : The ',description,' data type is not
     &supported by this module'
830   FORMAT(A12,A14,A42)
             STOP
      END IF

      READ(3,800,ERR=2001) description,tmpstr
      IF(tmpstr(1:5) .EQ. 'ascii') THEN
            old_file_type=0
      ELSE IF (tmpstr(1:6) .EQ. 'binary') THEN
            old_file_type=1
      ELSE IF (tmpstr(1:13) .EQ. 'packed binary') THEN
            old_file_type=2
      ELSE
            old_file_type=999
      END IF
      IF(old_file_type .GT. 2) THEN
            WRITE(*,830) 'Error : The ',description,' file type is not
     &supported by this module'
            STOP
      END IF

      READ(3,804,ERR=2001) description,cols
      READ(3,804,ERR=2001) description,rows
804   FORMAT(A14,I4)
      READ(3,800,ERR=2001) description,tmpstr
      ref_system=tmpstr
      READ(3,800,ERR=2001) description,tmpstr
      CALL UPCASE(tmpstr,tmpstr2)
      IF ((tmpstr2(1:1) .EQ. 'M') .OR. (tmpstr2(1:2) .EQ. 'FT') .OR.
    &(tmpstr2(1:2) .EQ. 'MI') .OR. (tmpstr2(1:2) .EQ. 'KM') .OR.
    &(tmpstr2(1:3) .EQ. 'DEG') .OR. (tmpstr2(1:3) .EQ. 'RAD')) THEN
      ref_units=tmpstr
      END IF

      READ(3,801,ERR=2001) description,unit_dist
      READ(3,801,ERR=2001) description,wxmin
      READ(3,801,ERR=2001) description,wxmax
      READ(3,801,ERR=2001) description,wymin
      READ(3,801,ERR=2001) description,wymax
801   FORMAT(A14,1PG32.16)

      cellx=(wxmax-wxmin)/cols
      celly=(wymax-wymin)/rows

      READ(3,800,ERR=2001) description,tmpstr
      posn_error_flag=0
      posn_error=0
      CALL VAL(tmpstr,posn_error,err_code)
      IF(err_code .EQ. 0) THEN
      posn_error_flag=1
      END IF

      READ(3,800,ERR=2001) description,tmpstr
      resolution_flag=0
      resolution=0
      CALL VAL(tmpstr,resolution,err_code)
      IF(err_code .EQ. 0) THEN
      resolution_flag=1
      END IF
      READ(3,800,ERR=2001) description,tmpstr
      CALL VAL(tmpstr,min,err_code)
      READ(3,800,ERR=2001) description,tmpstr
      CALL VAL(tmpstr,max,err_code)
      READ(3,800,ERR=2001) description,tmpstr
      val_units=tmpstr
      READ(3,800,ERR=2001) description,tmpstr
      val_error_flag=0
      val_error=0
      CALL VAL(tmpstr,val_error,err_code)
      IF(err_code .EQ. 0) THEN
      val_error_flag=1
      END IF

      READ(3,800,ERR=2001) description,tmpstr
      flag_flag=0
      flag_value=0
      CALL VAL(tmpstr,flag_value,err_code)
      IF(err_code .EQ. 0) THEN
      flag_flag=1
      END IF
      READ(3,800,ERR=2001)description,tmpstr
      flag_defn=tmpstr
      READ(3,806,ERR=2001)description,legend
806   FORMAT(A14,I3)

      IF (legend .NE. 0) THEN
            IF (legend .GT. 255) THEN
            legend=255
            END IF
            DO 6 I=1,legend
            READ(3,800,ERR=2001) description,tmpstr
            legend_text(I)=tmpstr
6     CONTINUE
      END IF

      new_data_type=old_data_type
      new_file_type=old_file_type
      CLOSE(3)
      RETURN
2001  WRITE(*,*) '** erreur de lecture: documentation file **'
      CLOSE(3)

      END

C***********************************************************
      SUBROUTINE read_env_file

      INCLUDE 'IDRIVAR.TXT'

      CHARACTER*80 temp
      CHARACTER*40 env_txt

      path=' '
      drive=' '
      temp='idrisi.env'

      OPEN(3,FILE=temp)
      READ(3,899,ERR=2001) temp
      READ(3,899,ERR=2001) temp
899   FORMAT(A80)
      READ(3,900,ERR=2001) env_txt,drive
900   FORMAT(A40,A2)
      READ(3,901,ERR=2001) env_txt,path
901   FORMAT(A40,A40)
      READ(3,902,ERR=2001) env_txt,image_file_extension
902   FORMAT(A40,A4)
      READ(3,902,ERR=2001) env_txt,image_docfile_extension
      READ(3,902,ERR=2001) env_txt,vector_file_extension
      READ(3,902,ERR=2001) env_txt,vector_docfile_extension
      READ(3,902,ERR=2001) env_txt,values_file_extension
      READ(3,902,ERR=2001) env_txt,values_docfile_extension
      READ(3,903,ERR=2001) env_txt,ref_units
903   FORMAT(A40,A3)
      READ(3,899,ERR=2001) temp
      READ(3,899,ERR=2001) temp
      READ(3,899,ERR=2001) temp
      READ(3,902,ERR=2001) env_txt,digi_port
      READ(3,902,ERR=2001) env_txt,plot_port
      READ(3,902,ERR=2001) env_txt,prn_port
      IF(path(1:4) .EQ. 'none') THEN
             path=' '
      END IF
      IF(drive(1:2) .EQ. 'no') THEN
             drive=' '
      ELSE IF(drive(2:2) .NE. ':') THEN
             drive=' '
      END IF
      CLOSE(3)
      RETURN
2001  WRITE(*,*) '** erreur de lecture: environment file **'
      CLOSE(3)

      END


C*****************************
      SUBROUTINE read_image_data(file_unit,N1,ndata,ZINT,ZREAL,ZBYTE,
     &file_type,data_type)

      INTEGER*2 data_type,file_type,file_unit
      INTEGER*4 ndata,N1
      INTEGER*4 I0
      REAL*4 WREAL,NWREAL,WREAL1,NWREAL1
      REAL*4 ZREAL(*)
      INTEGER*2 WINT,NWINT,WINT1,NWINT1,I1
      INTEGER*2 ZINT(*)
      INTEGER*1 WBYTE,NWBYTE,WBYTE1,NWBYTE1
      INTEGER*1 ZBYTE(*)

C............I0=compteur du nombre total de donnes lues,
C............I1=compteur de la position (ndata) dans la lecture de
C...............de fichiers
C...............comprimés (prolème possible si non initialisé).
C............Question: I1,NWINT1(pourrait être NWINT),WINT1(..WINT)
C............sont-ils conservés entre deux appels de la sous-routine?

      I0=0

      SELECT CASE (data_type)
      CASE(0)
          IF (file_type .EQ. 0) THEN
              DO J=1,ndata
                READ(file_unit,*,ERR=2001) ZINT(J+N1)
              END DO
          ELSE IF (file_type .EQ. 1) THEN
              DO J=1,ndata
                READ(file_unit,ERR=2001) ZINT(J+N1)
              END DO
          ELSE
              IF (I1 .NE. NWINT1) THEN
                DO I=I1+1,NWINT1
                    ZINT(I-I1+N1)=WINT1
                    I0=I0+1
                END DO
              END IF
              DO WHILE (I0 .LT. ndata)
                READ(file_unit,ERR=2001) NWINT
                READ(file_unit,ERR=2001) WINT
                NWINT1=NWINT
                WINT1=WINT
                DO I=1,NWINT
                    I0=I0+1
                    I1=I-1
                    IF (I0 .GT. ndata) THEN
                      EXIT
                    ELSE
                      ZINT(I0+N1)=WINT
                    END IF
                END DO
              END DO
          END IF
      CASE(1)
          IF (file_type .EQ. 0) THEN
              DO J=1,ndata
                READ(file_unit,*,ERR=2001) ZREAL(J+N1)
              END DO
          ELSE IF (file_type .EQ. 1) THEN
              DO J=1,ndata
                READ(file_unit,ERR=2001) ZREAL(J+N1)
              END DO
          ELSE
              IF (I1 .NE. NWREAL1) THEN
                DO I=I1+1,NWREAL1
                    ZREAL(I-I1+N1)=WREAL1
                    I0=I0+1
                END DO
              END IF
              DO WHILE (I0 .LT. ndata)
                READ(file_unit,ERR=2001) NWREAL
                READ(file_unit,ERR=2001) WREAL
                DO I=1,NWREAL
                    I0=I0+1
                    I1=I-1
                    NWREAL1=NWREAL
                    WREAL1=WREAL
                    IF (I0 .GT. ndata) THEN
                      EXIT
                    ELSE
                      ZREAL(I0+N1)=WREAL
                    END IF
                END DO
              END DO
          END IF
      CASE(2)
          IF (file_type .EQ. 0) THEN
              DO J=1,ndata
                READ(file_unit,*,ERR=2001) ZBYTE(J+N1)
              END DO
          ELSE IF(file_type .EQ. 1) THEN
              DO J=1,ndata
                READ(file_unit,ERR=2001) ZBYTE(J+N1)
              END DO
          ELSE
              IF (I1 .NE. NWBYTE1) THEN
                DO I=I1+1,NWBYTE1
                    ZBYTE(I-I1+N1)=WBYTE1
                    I0=I0+1
                END DO
              END IF

              DO WHILE (I0 .LT. ndata)
                READ(file_unit,ERR=2001) NWBYTE
                READ(file_unit,ERR=2001) WBYTE
                DO I=1,NWBYTE
                    I0=I0+1
                    I1=I-1
                    NWBYTE1=NWBYTE
                    WBYTE1=WBYTE
                    IF (I0 .GT. ndata) THEN
                      EXIT
                    ELSE
                      ZBYTE(I0+N1)=WBYTE
                    END IF
                END DO
              END DO
          END IF
      END SELECT
      N1=N1+ndata
      RETURN
2001  WRITE(*,*) '** erreur de lecture: image file **'
      CLOSE(file_unit)

      END

C**********************************************************
      SUBROUTINE open_new_image_file(file_unit)

      INCLUDE 'IDRIVAR.TXT'

      CHARACTER*80 imgname
      INTEGER*2 file_unit

      imgname=' '
      INFO(1)=imgname
      INFO(2)=drive
      INFO(3)=path
      INFO(4)=new_image
      INFO(5)=image_file_extension
      CALL COLLE(5,INFO)
      imgname=INFO(1)


      SELECT CASE (new_file_type)

            CASE(0)
            OPEN(file_unit,FILE=imgname)
            RETURN

            CASE(1)
            OPEN(file_unit,FILE=imgname,FORM='BINARY')
            RETURN

            CASE(2)
            OPEN(file_unit,FILE=imgname,FORM='BINARY')
            RETURN

      END SELECT

      END

C**********************************************************
      SUBROUTINE open_old_image_file(file_unit)
      INCLUDE 'IDRIVAR.TXT'

      CHARACTER*80 imgname
      INTEGER*2 file_unit

      imgname=' '
      INFO(1)=imgname
      INFO(2)=drive
      INFO(3)=path
      INFO(4)=old_image
      INFO(5)=image_file_extension
      CALL COLLE(5,INFO)
      imgname=INFO(1)


      SELECT CASE (old_file_type)

            CASE(0)
            OPEN(file_unit,FILE=imgname)
            RETURN

            CASE(1)
            OPEN(file_unit,FILE=imgname,FORM='BINARY')
            RETURN

            CASE(2)
            OPEN(file_unit,FILE=imgname,FORM='BINARY')
            RETURN

      END SELECT

      END
C*******************************
      SUBROUTINE write_image_data(file_unit,N1,ndata,ZINT,ZREAL,ZBYTE,
     &file_type,data_type)

      INTEGER*2 data_type,file_type,file_unit
      INTEGER*4 ndata,N1
      REAL*4 ZREAL(*)
      INTEGER*2 ZINT(*)
      INTEGER*1 ZBYTE(*)


      SELECT CASE (data_type)
      CASE(0)
          IF (file_type .EQ. 0) THEN
              DO J=1,ndata
              WRITE(file_unit,100,ERR=2002) ZINT(J+N1)
100   FORMAT(I5)
              END DO
          ELSE IF (file_type .EQ. 1) THEN
              DO J=1,ndata
              WRITE(file_unit,ERR=2002) ZINT(J+N1)
              END DO
          ELSE
              WRITE(*,*) 'désolés, nous ne pouvons pas écrire les filières
     & comprimées..'
              WRITE(*,*) 'Utiliser CONVERT'
          END IF
      CASE(1)
          IF (file_type .EQ. 0) THEN
              DO J=1,ndata
              WRITE(file_unit,200,ERR=2002) ZREAL(J+N1)
200   FORMAT(1PE13.6)
              END DO
          ELSE IF (file_type .EQ. 1) THEN
              DO J=1,ndata
              WRITE(file_unit,ERR=2002) ZREAL(J+N1)
              END DO
          ELSE
              WRITE(*,*) 'désolés, nous ne pouvons pas écrire les filières
     & comprimées..'
              WRITE(*,*) 'Utiliser CONVERT'
          END IF
      CASE(2)
          IF (file_type .EQ. 0) THEN
              DO J=1,ndata
              WRITE(file_unit,300,ERR=2002) ZBYTE(J+N1)
300   FORMAT(I3)
              END DO
          ELSE IF(file_type .EQ. 1) THEN
              DO J=1,ndata
              WRITE(file_unit,ERR=2002) ZBYTE(J+N1)
              END DO
          ELSE
              WRITE(*,*) 'désolés, nous ne pouvons pas écrire les filières
     & comprimées..'
              WRITE(*,*) 'Utiliser CONVERT'
          END IF
      END SELECT
      N1=N1+ndata
      RETURN
2002  WRITE(*,*) '** erreur d‘écriture: image file **'
      CLOSE(file_unit)

      END

C*****************************
      SUBROUTINE fill_Z_data(initial_data_type,n_data,ZINT,ZREAL,ZBYTE)

      INTEGER*2 ZINT(*)
      REAL*4 ZREAL(*)
      INTEGER*1 ZBYTE(*)
      INTEGER*2 initial_data_type

      DO I=1,n_data
          SELECT CASE(initial_data_type)
          CASE(0)
            ZREAL(I)=REAL(ZINT(I))
C............mettre un flag pour valeurs tronquees!
            IF (ZINT(I) .GT. 127) THEN
              ZBYTE(I)=127
            ELSE IF (ZINT(I) .LT. -128) THEN
              ZBYTE(I)=-128
            ELSE
              ZBYTE(I)=INT1(ZINT(I))
            END IF
          CASE(1)
            IF (ZREAL(I) .GT. 32767) THEN
              ZINT(I)=32767
            ELSE IF (ZREAL(I) .LT. -32768) THEN
              ZINT(I)=-32768
            ELSE
              ZINT(I)=INT2(ZREAL(I))
            END IF
            IF (ZREAL(I) .GT. 127) THEN
              ZBYTE(I)=127
            ELSE IF (ZREAL(I) .LT. -128) THEN
              ZBYTE(I)=-128
            ELSE
              ZBYTE(I)=INT1(ZREAL(I))
            END IF
          CASE(2)
            ZINT(I)=INT2(ZBYTE(I))
            ZREAL(I)=REAL(ZBYTE(I))
          END SELECT
      END DO

      END
```

MESUTILS.FOR

```
$NOTRUNCATE

      SUBROUTINE COLLE(NTEXT,TEXT)
      CHARACTER TEXT(*)*(*),TEXT1*80
      INTEGER NTEXT,LEN(80),L(80,2)
      TEXT1=' '
      L(1,1)=0
      L(1,2)=0
      DO 1 I=1,NTEXT
      LEN(I)=LEN_TRIM(TEXT(I))
1     CONTINUE
      I=2
      L(I,1)=1
      L(I,2)=LEN(I)+L(I,1)-1
      DO 2 I=3,NTEXT
      L(I,1)=L(I-1,2)+1
      L(I,2)=LEN(I)+L(I,1)-1
2     CONTINUE
      DO 3 I=2,NTEXT
      TEXT1(L(I,1):L(I,2))=TEXT(I)
3     CONTINUE
      TEXT(1)=TEXT1
      END
C**************************************
      SUBROUTINE VAL(STRING,DOBLVALUE,ERRCODE)
      INTEGER*1 ERRCODE,RFLAG
      INTEGER N,IN,L,INTVALUE
      CHARACTER STRING*(*),STR*80
      REAL*8 DOBLVALUE
      STR=STRING
      ERRCODE=0
      RFLAG=0
      L=LEN_TRIM(STRING)
      DO 1 N=1,L
      IN=ICHAR(STRING(N:N))
      IF ((IN .GT. 57) .OR. (IN .LE. 47)) THEN
         IF(IN .NE. 46) THEN
            ERRCODE=1
              EXIT
           ELSE
             IF (RFLAG .EQ. 1) THEN
             ERRCODE=1
             EXIT
             ELSE
             RFLAG=1
             END IF
         END IF
      END IF
1     CONTINUE
      IF (ERRCODE .EQ. 0) THEN
           WRITE(STR,800) STRING
           IF (RFLAG .EQ. 1) THEN
               READ(STR,801) DOBLVALUE
           ELSE
               READ(STR,802) INTVALUE
               DOBLVALUE=DBLE(INTVALUE)
           END IF
800   FORMAT(A80)
801   FORMAT(D80.39)
802   FORMAT(I80)
      ELSE
           DOBLVALUE=0
      END IF
      END

C**************************************
      SUBROUTINE UPCASE(STRING1,STRING2)
      CHARACTER*(*) STRING1,STRING2
      INTEGER L,ASCII
      STRING2=' '
      L=LEN_TRIM(STRING1)
      DO 1 I=1,L
      ASCII=ICHAR(STRING1(I:I))
      IF((ASCII .LE. 122) .AND. (ASCII .GT. 96)) THEN
         ASCII=ASCII-32
      END IF
      STRING2(I:I)=CHAR(ASCII)
1     CONTINUE
      END
```

```
C******************************
      subroutine read_cmd_line_args(cmd_line_options_num,
     &cmd_line_options,cmd_line_files_num,cmd_line_files)

      CHARACTER*1 cmd_line_options(10)
      CHARACTER*8 cmd_line_files(10),cmd_line_args(10)
      INTEGER*2 cmd_line_err
      INTEGER*4 cmd_line_args_num,cmd_line_options_num,
     &cmd_line_files_num

      cmd_line_options_num=0
      cmd_line_files_num=0
      cmd_line_args_num=NARGS( )-1

      IF(cmd_line_args_num .EQ. 0) THEN
      CALL HELP
      STOP
      END IF

      DO I=1,cmd_line_args_num
      CALL GETARG (I,cmd_line_args(I),cmd_line_err)
C      WRITE(*,*) 'ARGUMENT ',I,':',cmd_line_args(I)

      IF (cmd_line_err .EQ. -1) THEN
      WRITE(*,*) 'ERREUR: CALL GETARG(I,cmd_line_args(I),cmd_line_err)'
      CALL HELP
      STOP
      END IF

      SELECT CASE(cmd_line_args(I)(1:1))
      CASE('-')
      cmd_line_options_num=cmd_line_options_num+1
      cmd_line_options(cmd_line_options_num)=cmd_line_args(I)(2:2)
      CASE('?')
      CALL HELP
      STOP
      CASE DEFAULT
C......... initialiser ici les valeurs default des parametres..
      END SELECT
      cmd_line_files_num=cmd_line_files_num+1
      cmd_line_files(cmd_line_files_num)=cmd_line_args(I)
      END DO

      DO I=1,cmd_line_options_num
      SELECT CASE(cmd_line_options(I))
      CASE('?')
      CALL HELP
c .....ici initialiser ce que font les parametres par default......
      END SELECT
      END DO

      DO I=1,cmd_line_files_num
      IF (SCAN(cmd_line_files(I),':') .NE. 0) THEN
      WRITE(*,*) 'ERREUR CMD LINE: NE PAS SPECIFIER DE DISQUE'
      STOP
      ELSE IF(SCAN(cmd_line_files(I),'\') .NE. 0) THEN
      WRITE(*,*) 'ERREUR CMD LINE: NE PAS SPECIFIER DE PATH'
      STOP
      ELSE IF(SCAN(cmd_line_files(I),'.') .NE. 0) THEN
      WRITE(*,*) 'ERREUR CMD LINE: NE PAS SPECIFIER D'EXTENSION'
      STOP
      END IF
      END DO

      END
C*************************************
```

ANSWVARW.TXT

```
 CHARACTER*80 ligne,mot
 character*5 elem_units
 real*4 elem_dim
 real*4 sed(45000),min_sed,max_sed
 real*4 cdep(45000),min_cdep,max_cdep
 REAL*4 ZREAL2(45000),ZREAL3(45000)
 INTEGER*2 N,N0,N1,elem_nb,min_row,max_row,min_col,max_col
 integer*2 chan_nb
 integer*2 row_nb(45000),col_nb(45000)
 CHARACTER*43 ws_name

 CHARACTER*80 filenames(64),ans_elem_file,ans_out_file,
&init_file_file,elem_nb_file

 CHARACTER*8  ans_elem,ans_out,idri_sed,init_file,idri_cdep

 INTEGER*4 n_Z
 INTEGER*1 compteur
 INTEGER*2 elem_nb_4,elem_nb_m4,chan_nb_4,chan_nb_m4
 CHARACTER*14 description
 CHARACTER*8 files(64)
 integer*1 min_max_flag
 integer*2 row,col,J0

 integer*1 chan_cat(4096),soil_type(4096),crop_man(4096),
&gauge(4096),tile(4096),bmpi(4096),
&bmp1(4096),bmp2(4096)
 integer*2 slope(4096),aspect(4096),chan_slope(4096),elev(4096)
 integer*2 min_slope,max_slope,min_aspect,max_aspect,
&min_elev,max_elev
&min_chan_slope,max_chan_slope
 integer*1 min_soil_type,max_soil_type,
&min_chan_cat,max_chan_cat,
&min_crop_man,min_gauge,min_tile,
&min_bmpi,min_bmp1,
&min_bmp2,
&max_crop_man,max_gauge,max_tile,
&max_bmpi,max_bmp1,
&max_bmp2
 character*8 idri_slope,idri_aspect,idri_chan_cat,idri_soil,
&idri_crop_man,idri_gauge,idri_tile,idri_chan_slope,
&idri_bmpi,idri_bmp1,idri_bmp2,idri_elev
```

ANSWVAR.TXT

```
 CHARACTER*80 ligne,mot
 character*5 elem_units
 real*4 elem_dim
 real*4 sed(1700),min_sed,max_sed
 real*4 cdep(1700),min_cdep,max_cdep
 REAL*4 ZREAL2(4096)
 INTEGER*2 N,N0,N1,elem_nb,min_row,max_row,min_col,max_col
 integer*2 chan_nb
 integer*2 row_nb(1700),col_nb(1700)
 CHARACTER*43 ws_name

 CHARACTER*80 filenames(64),ans_elem_file,ans_out_file,
&init_file_file,elem_nb_file

 CHARACTER*8  ans_elem,ans_out,idri_sed,init_file,idri_cdep

 INTEGER*4 n_Z
 INTEGER*1 compteur
 INTEGER*2 elem_nb_4,elem_nb_m4,chan_nb_4,chan_nb_m4
 CHARACTER*14 description
 CHARACTER*8 files(64)
 integer*1 min_max_flag
 integer*2 row,col,J0

 integer*1 chan_cat(4096),soil_type(4096),crop_man(4096),
&gauge(4096),tile(4096),bmpi(4096),
&bmp1(4096),bmp2(4096)
 integer*2 slope(4096),aspect(4096),chan_slope(4096),elev(4096)
 integer*2 min_slope,max_slope,min_aspect,max_aspect,
&min_elev,max_elev
&min_chan_slope,max_chan_slope
 integer*1 min_soil_type,max_soil_type,
&min_chan_cat,max_chan_cat,
&min_crop_man,min_gauge,min_tile,
&min_bmpi,min_bmp1,
&min_bmp2,
&max_crop_man,max_gauge,max_tile,
&max_bmpi,max_bmp1,
&max_bmp2
 character*8 idri_slope,idri_aspect,idri_chan_cat,idri_soil,
&idri_crop_man,idri_gauge,idri_tile,idri_chan_slope,
&idri_bmpi,idri_bmp1,idri_bmp2,idri_elev
```

IDRIVAR.TXT

```
 CHARACTER*3 ref_units
 CHARACTER*66 val_units,title,ref_system,flag_defn
 INTEGER*4 val_recs,rows,cols
 INTEGER*1 val_fields,val_file_type
 INTEGER*2 legend,data_format,geo_type
 INTEGER*2 old_data_type,old_file_type
 INTEGER*2 new_data_type,new_file_type
 REAL*8 wxmin,wxmax,wymin,wymax,unit_dist
 REAL*8 flag_value,resolution,cellx,celly
 REAL*8 posn_error,val_error,min,max
 INTEGER*1 posn_error_flag,resolution_flag,val_error_flag
 INTEGER*1 flag_flag
 CHARACTER*66 legend_text
 CHARACTER*40 path
 CHARACTER*4 digi_port,plot_port,prn_port
 CHARACTER*2 drive,units
 CHARACTER*8 old_image,new_image
 CHARACTER*4 image_docfile_extension
 CHARACTER*4 image_file_extension
 CHARACTER*4 vector_docfile_extension
 CHARACTER*4 vector_file_extension
 CHARACTER*4 values_docfile_extension
 CHARACTER*4 values_file_extension
 CHARACTER*80 INFO

 COMMON /IDRIDAT/ ref_units,val_units,title,ref_system,flag_defn,
&val_recs,rows,cols,val_fields,val_file_type,legend,data_format,
&geo_type,old_data_type,old_file_type,new_data_type,new_file_type,
&wxmin,wxmax,wymin,wymax,unit_dist,flag_value,resolution,cellx,
&celly,posn_error,val_error,min,max,posn_error_flag,
&resolution_flag,val_error_flag, flag_flag,legend_text(256),
&path,
&digi_port,plot_port,prn_port,drive,units,old_image,new_image,
&image_docfile_extension,image_file_extension,
&vector_docfile_extension,vector_file_extension,
&values_docfile_extension,values_file_extension,
&INFO(10)

 REAL*4 ZREAL(4096)
 INTEGER*2 ZINT(4096)
 INTEGER*1 ZBYTE(4096)
 INTEGER*4 new_col_pos,old_col_pos,col_pos
 CHARACTER*8 cmd_line_files(10), cmd_line_args(10)
 INTEGER*4 cmd_line_options_num,cmd_line_files_num
```

PINITW.ANS

```
predata.....: ppredat
elements....: pdat
sortie......: pout7
slope.......: ppendi2
aspect......: paspe3
ch..size.cat: pccat5
soil.type...: psuelos
crop/manag..: puso
rain.gauge..: ppluvio
tile.flag...: ptile
ch..slope...: pcpendi2
BMP.ID......: pbmpid
BMP.descr..1: pbmp1
BMP.descr..2: pbmp2
mean.elev...: pelev
sediments...: psed7
ch..depos...: pcdep7
```