

Record Number:

Author, Monographic: Cluis, D.//Laberge, C.//Houle, C.

Author Role:

Title, Monographic: Détection des tendances en qualité de l'eau. Tome III - Organigrammes
et listes

Translated Title:

Reprint Status:

Edition:

Author, Subsidiary:

Author Role:

Place of Publication: Québec

Publisher Name: INRS-Eau

Date of Publication: 1987

Original Publication Date: Aoûr 1987

Volume Identification:

Extent of Work: 285

Packaging Method: pages

Series Editor:

Series Editor Role:

Series Title: INRS-Eau, Rapport de recherche

Series Volume ID: 238

Location/URL:

ISBN: 2-89146-235-1

Notes: Rapport annuel 1987-1988

Abstract: Rapport rédigé pour Environnement Canada
45.00\$

Avec R229, R230, R245

Call Number: R000238

Keywords: rapport/ ok/ dl

DÉTECTION DES TENDANCES
EN QUALITÉ DE L'EAU

III Organigrammes et listes

août 1987

INRS-Eau
C.P. 7500, Sainte-Foy
Québec, Canada
G1V 4C7

Rapport scientifique no 238

par

Daniel CLUIS
Claude LABERGE
Caroline HOULE

pour

Service Protection et Conservation
Environnement Canada - région du Québec -
MAS # KE 303-6-0042/01-SD

Sat 09-12-87 17:47:25
09-12-87 10:05:26 SAISIE.FOR

Pg 1

SAISIE.FOR

QSMODE, QCLEAR, QBORD, ELSEIF, QBEEP, CHAINE

```
C      PROGRAM SAISMAN
      * * * * *
      INTEGER      AN
      REAL          PAR(30)
      CHARACTER*40 F1,F2,F3
      CHARACTER*20 NOMFICH,VIEUXF,TAMP,CHAINE
      CHARACTER*2  LECT

      OPEN(1,FILE='SETUP.PC',STATUS='OLD')
      READ(1,'(4I2/9I2,5I3)')ITMODE,ITFORE,ITBACK,ITCOLOR,IGMODE,
      *   IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,ILETR,ISEC,KIRSYM,ISYMBL,
      *   JROW1,JROW2,JCOL1,JCOL2
      CLOSE(1)
100    CALL QSMODE(ITMODE)
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QBORD(ITCOLOR)
      WRITE(*,*)' '
      WRITE(*,*)' 1) ANNEXER DES DONNEES A UN FICHIER EXISTANT'
      WRITE(*,*)' 2) CREE UN NOUVEAU FICHIER DE DONNEES'
      WRITE(*,*)' '
      WRITE(*,'(A18\)' )' VOTRE CHOIX : '
      READ(*,*,ERR=100)IREP
      IF (IREP.NE.1.AND.IREP.NE.2) GOTO 100
105    IF (IREP.EQ.1) THEN
      WRITE(*,*)' '
      WRITE(*,'(A42\)' )'ENTREZ LE NOM DE L'ANCIEN FICHIER : '
      READ(*,'(A20)' )VIEUXF
      OPEN(1,FILE=VIEUXF,STATUS='OLD',ERR=105)
      ENDIF
107    CALL QSMODE(ITMODE)
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QBORD(ITCOLOR)
      WRITE(*,*)' '
      WRITE(*,'(A38\)' )'ENTREZ LE NOM DU FICHIER DE SORTIE : '
      READ(*,'(A20)' )NOMFICH
      OPEN(2,FILE=NOMFICH,STATUS='NEW')
110    CALL QSMODE(ITMODE)
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QBORD(ITCOLOR)
      WRITE(*,*)' '
      WRITE(*,*)' 1) UN PARAMETRE ET DEBIT SEULEMENT (FORMAT D'ENTREE
      +   DE P1)'
      WRITE(*,*)' 2) UN PARAMETRE ET PAS DE DEBIT (FORMAT D'ENTREE DE
      +   P1)'
      WRITE(*,*)' 3) FORMAT POUR L'UTILITAIRE "DIVISE" '
      WRITE(*,*)' '
      WRITE(*,'(A18\)' )' VOTRE CHOIX : '
      READ (*,*,ERR=110) IFORM
      IF (IFORM.NE.1.AND.IFORM.NE.2.AND.IFORM.NE.3) GOTO 110
      IF (IFORM.EQ.1) THEN
      F1='(12X,I2,1X,I2,1X,I2,16X,2F12.6)'
      F2='(12X,I2,1X,I2,1X,I2,16X)'
      F3='(2F12.6)'
      NBPARG = 2
```

```

      IDAT = 1
ELSEIF (IFORM.EQ.2) THEN
      F1='(12X,I2,1X,I2,1X,I2,16X,F12.6) '
      F2='(12X,I2,1X,I2,1X,I2,16X\)'
      F3='(F12.6) '
      NBPARG = 1
      IDAT = 1
ELSE
120  CALL QSMODE(ITMODE)
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QBORD(ITCOLOR)
      WRITE(*,*) ' '
      WRITE(*,*) ' ENTREZ LE NOMBRE DE PARAMETRE '
      WRITE(*, '(A38\)' ) ' INCLUANT LE DEBIT MAIS PAS LA DATE : '
      READ (*,*,ERR=120) NBPARG
      IF (NBPARG.GT.30.OR.NBPARG.LT.1) GOTO 120
      F1='(1X,I2,1X,I2,1X,I2,1X,30(1X,E12.5)) '
      F2='(1X,I2,1X,I2,1X,I2,1X\)'
125  F3='(30(1X,E12.5)) '
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' 1) PRESENCE DES DATES DANS LE FICHER '
      WRITE(*,*) ' 2) PAS DE DATE DANS LE FICHER '
      WRITE(*,*) ' '
      WRITE(*, '(A18\)' ) ' VOTRE CHOIX : '
      READ(*,*,ERR=125) IDAT
      IF (IDAT.NE.1.AND.IDAT.NE.2) GOTO 125
ENDIF

IF (IREP.EQ.1) THEN
130  IF (IDAT.EQ.1) THEN
      READ(1,F2,END=135,ERR=105) AN,MOIS,JR
      WRITE(2,F2,ERR=105) AN,MOIS,JR
      ENDIF
      READ(1,F3,END=135,ERR=105) (PAR(I),I=1,NBPARG)
      WRITE(2,F3,ERR=105) (PAR(I),I=1,NBPARG)
      GOTO 130
135  CLOSE(1)
ENDIF

140  IF (IDAT.EQ.1) THEN
      LJR =1
      LMOIS=1
      LAN =70
      IF (IREP.NE.1) THEN
          MOIS = 1
          AN = 70
          JR = 1
700  CALL QSMODE(ITMODE)
          CALL QCLEAR(ITBACK,ITFORE)
          CALL QBORD(ITCOLOR)
          WRITE(*,*) ' '
          IF (MOIS.EQ.0) THEN
              MOIS = LMOIS

```

```

      GOTO 710
    ELSE
      LMOIS = MOIS
    ENDIF
    WRITE(*, '(A21, I2, A3\)' ) ' ENTREZ LE MOIS      [', MOIS, ']' : '
    READ(*, '(I2)', ERR=700) MOIS
    IF (MOIS.GT.12.OR.MOIS.LT.1) GOTO 700
    IF (MOIS.LT.LMOIS) AN = AN + 1
    WRITE(*,*) ' '
710   IF (AN.EQ.0) THEN
      AN = LAN
      GOTO 210
    ELSE
      LAN = AN
    ENDIF
    WRITE(*, '(A21, I2, A3\)' ) ' ENTREZ L' 'ANNEE      [', AN, ']' : '
    READ(*, '(I2)', ERR=710) AN
    IF (AN.LT.50.OR.AN.GT.90) GOTO 710
  ENDIF
210  CALL QSMODE(ITMODE)
    CALL QCLEAR(ITBACK,ITFORE)
    CALL QBORD(ITCOLOR)
    WRITE(*,*) ' '
    WRITE(*, '(A36\)' ) ' ENTREZ LE JOUR ("F" POUR FINIR) : '
    READ(*, '(A2)', ERR=210) LECT
    IF (LECT.EQ.'F'.OR.LECT.EQ.'f') THEN
      GOTO 600
    ELSE
      READ(LECT, '(I2)', ERR=210) JR
    ENDIF
    IF (JR.GT.31.OR.JR.LT.1) GOTO 210
    IF (JR.LE.LJR) MOIS = MOIS + 1
    IF (MOIS.GE.13) THEN
      MOIS = 1
      AN = AN + 1
    ENDIF
220  CALL QSMODE(ITMODE)
    CALL QCLEAR(ITBACK,ITFORE)
    CALL QBORD(ITCOLOR)
    WRITE(*,*) ' '
    DO 250 I=1,NBPAR
910   FORMAT (A8, I2, A1, I2, A3, I2)
225   WRITE(*,*) ' '
      WRITE(*, 910) ' DATE : ', JR, '/', MOIS, '/19', AN
      WRITE(*,*) ' '
      WRITE(*, '(A21, I2, A14\)' ) ' ENTREZ LE PARAMETRE ', I, ' (H = HE
      LP) : '
      READ (*, '(A10)', ERR=225) CHAINE
      IF (CHAINE.EQ.'H'.OR.CHAINE.EQ.'h') THEN
        WRITE(*,*) ' '
        WRITE(*,*) ' "D" POUR CORRIGER LA DATE '
        WRITE(*,*) ' "E" POUR EFFACER LE DERNIER ENREGISTREMENT '
        WRITE(*,*) ' "F" POUR FINIR '
        GOTO 225
      ELSEIF (CHAINE.EQ.'F'.OR.CHAINE.EQ.'f') THEN

```

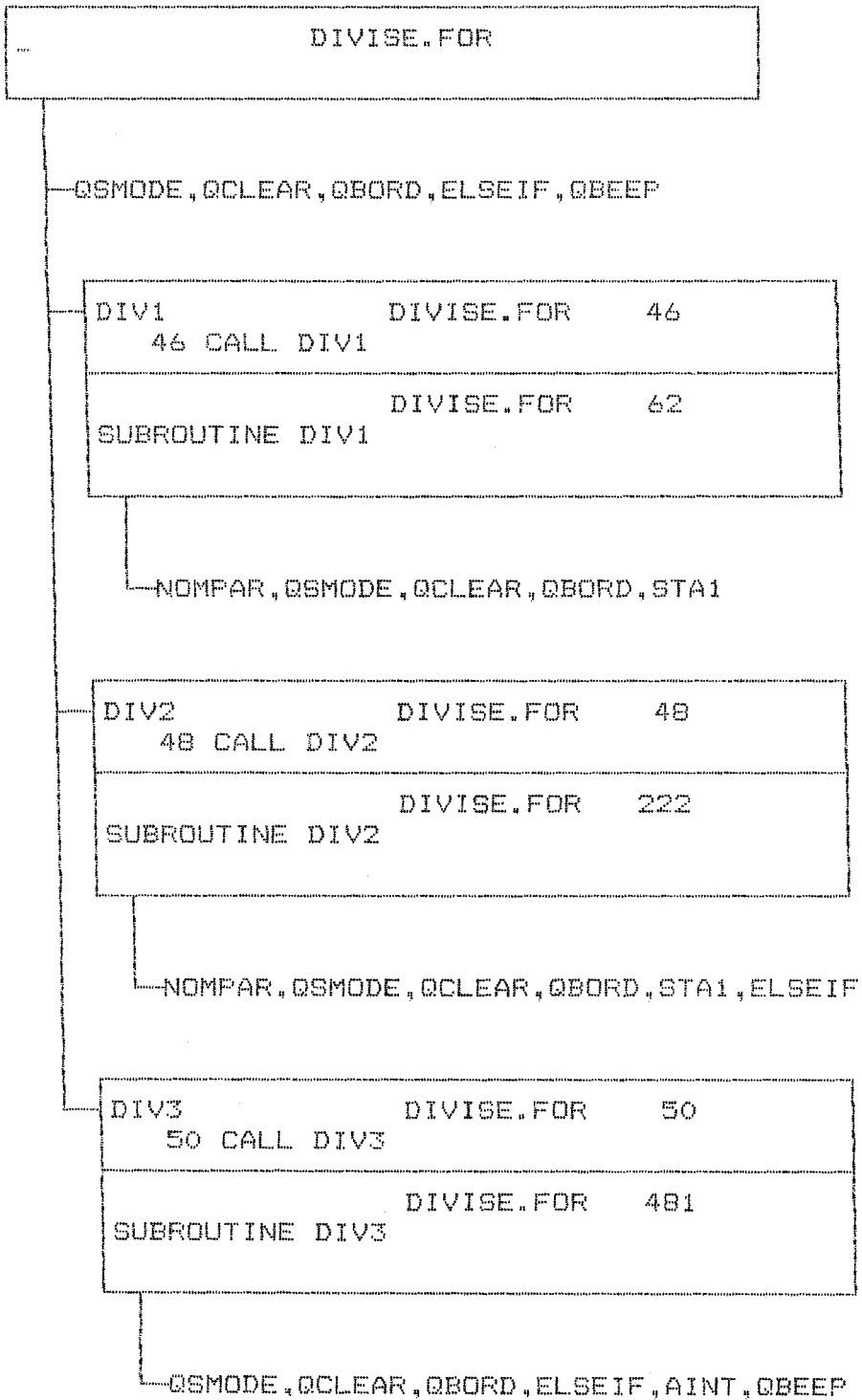
```
CALL QBEEP
GOTO 600
ELSEIF (CHAINE.EQ.'E'.OR.CHAINE.EQ.'e') THEN
CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
WRITE(*,*) ' '
WRITE(*,*) ' ENREGISTREMENT EFFACE '
BACKSPACE(2)
WRITE(*,*) ' '
READ(2,F3) (PAR(II),II=1,NBPAR)
WRITE(*,F3) (PAR(II),II=1,NBPAR)
BACKSPACE(2)
CALL QBEEP
GOTO 220
230 ELSEIF (CHAINE.EQ.'D'.OR.CHAINE.EQ.'d') THEN
WRITE(*,*) ' '
IF (MOIS.EQ.0) THEN
MOIS = LMOIS
GOTO 240
ELSE
LMOIS = MOIS
ENDIF
WRITE(*,'(A21,I2,A3\)' ) ' ENTREZ LE MOIS [',MOIS,'] : '
READ(*,'(I2)',ERR=230) MOIS
IF (MOIS.GT.12.OR.MOIS.LT.1) GOTO 230
IF (MOIS.LT.LMOIS) AN = AN + 1
240 WRITE(*,*) ' '
IF (AN.EQ.0) THEN
AN = LAN
GOTO 220
ELSE
LAN = AN
ENDIF
WRITE(*,'(A21,I2,A3\)' ) ' ENTREZ L' ANNEE [',AN,'] : '
READ(*,'(I2)',ERR=240) AN
IF (AN.LT.50.OR.AN.GT.90) GOTO 240
GOTO 220
ELSE
READ(CHaine,'(E12.0)',ERR=225) PAR(I)
ENDIF
250 CONTINUE
WRITE(2,F1) AN,MOIS,JR,(PAR(I),I=1,NBPAR)
LJR = JR
LMOIS = MOIS
LAN = AN
GOTO 210
300 ELSE
CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
WRITE(*,*) ' '
305 DO 390 I=1,NBPAR
310 WRITE(*,'(A23,I2,A14\)' ) ' ENTREZ LE PARAMETRE : ',I,' (H =
HELP) : '
+
```



```

      READ (*, '(A10)') CHAINE
      IF (CHAINE.EQ.'C'.OR.CHAINE.EQ.'c') THEN
        CALL QBEEP
        GOTO 300
      ELSEIF (CHAINE.EQ.'H'.OR.CHAINE.EQ.'h') THEN
        WRITE(*,*) ' '
        WRITE(*,*) ' "C" POUR CORRECTION DE L'ENREGISTREMENT '
        WRITE(*,*) ' "E" POUR EFFACER LE DERNIER ENREGISTREMENT '
        WRITE(*,*) ' "F" POUR FINIR '
        GOTO 310
      ELSEIF (CHAINE.EQ.'F'.OR.CHAINE.EQ.'f') THEN
        CALL QBEEP
        GOTO 600
      ELSEIF (CHAINE.EQ.'E'.OR.CHAINE.EQ.'e') THEN
        CALL QSMODE(ITMODE)
        CALL QCLEAR(ITBACK,ITFORE)
        CALL QBORD(ITCOLOR)
        WRITE(*,*) ' '
        WRITE(*,*) ' ENREGISTREMENT EFFACE '
        BACKSPACE(2)
        WRITE(*,*) ' '
        READ(2,F3) (PAR(II),II=1,NBPAR)
        WRITE(*,F3) (PAR(II),II=1,NBPAR)
        BACKSPACE(2)
        CALL QBEEP
        GOTO 305
      ELSE
        JPOINT = 0
        DO 370 J=1,20
          IF (CHAINE(J:J).EQ.' ') GOTO 380
          IF (CHAINE(J:J).EQ.' ') THEN
            CHAINE(J:J)=' '
            GOTO 380
          ENDIF
        CONTINUE
        READ(CHAINE,'(E12.5)',ERR=310) PAR(I)
      ENDIF
CONTINUE
CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
WRITE(*,*) ' '
WRITE(*,*) ' LIGNE ENREGISTREE '
WRITE(*,*) ' '
WRITE(2,F3) (PAR(I),I=1,NBPAR)
WRITE(*,F3) (PAR(I),I=1,NBPAR)
GOTO 305
ENDIF

600 ENDFILE(2)
      CLOSE(2)
      END
```



\$DEBUG

PROGRAM DIVISE

```
C *** -----
CHARACTER*1    REF
CHARACTER*20   FDIV
CHARACTER*3    PL(460),PARLET(30)
CHARACTER*5    CA(460)
COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH/ IGMODE,IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,
*   ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2
COMMON /BLOC1/ FDIV
COMMON /BLOC2/ PL,CA,PARLET

I=1
OPEN(1,FILE='CODES.NAQ',STATUS='OLD')
50  READ(1,'(A5,1X,A3)',END=60)CA(I),PL(I)
    I=I+1
    GOTD 50
60  CLOSE(1)

OPEN(2,FILE='SETUP.PC',STATUS='OLD')
READ(2,'(4I2/9I2,5I3)')ITMODE,ITFORE,ITBACK,ITCOLOR,IGMODE,
*   IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,ILETR,ISEC,KIRSYM,ISYMBL,
*   JROW1,JROW2,JCOL1,JCOL2
CLOSE(2)

CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
WRITE(*,*)' '
WRITE(*,'(A39)\')' Entrez le nom du fichier diviser : '
READ(*,'(A20)')FDIV
OPEN(99,FILE=FDIV,STATUS='OLD')
100 CALL QSMODE(ITMODE)
    CALL QCLEAR(ITBACK,ITFORE)
    CALL QBORD(ITCOLOR)
    WRITE(*,*)' '
    WRITE(*,*)' 1) Fichier type NAQUADAT normal'
    WRITE(*,*)' 2) Fichier type NAQUADAT avec des CR apres 80 carac.'
    WRITE(*,*)' 3) Fichier type compagnie '
    WRITE(*,*)' '
    WRITE(*,'(A20)\')' Votre choix : '
    READ(*,'(A1)') REP
    IF (REP.EQ.'1') THEN
        CALL DIV1
    ELSEIF (REP.EQ.'2') THEN
        CALL DIV2
    ELSEIF (REP.EQ.'3') THEN
        CALL DIV3
    ELSE
        GOTD 100
    ENDIF
    WRITE(*,*)' '
    WRITE(*,*)' '
```

```
WRITE(*,*) 'Le traitement est maintenant termin.'  
WRITE(*,*) ' '  
CALL QBEEP  
STOP  
END
```

SUBROUTINE DIV1

C *** -----

```
REAL          DEB,PARAM(30),V(15),VAL  
INTEGER       AN,NVAL(15)  
LOGICAL       PASVIDE  
CHARACTER*1   CAR,REP  
CHARACTER*3   PL(460),PARLET(30)  
CHARACTER*5   CA(460)  
CHARACTER*6   NOSTA  
CHARACTER*12  BID,B12,B1,B2  
CHARACTER*12  STA1,STA2  
CHARACTER*20  FDIV,NOMSFICH(15),NOMPAR  
CHARACTER*24  DASTA
```

```
COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR  
COMMON /GRAPH/ IGMODE,IGBACK,IFAL,IGBORD,IGCOLOR,ILINE,  
*   ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2  
COMMON /BLOC1/ FDIV  
COMMON /BLOC2/ PL,CA,PARLET
```

c *** lecture du nombre de parametres presents dans le fichier

```
READ(99,'(3X,I2,1X\)' )NP
```

```
IF (NP.GT.25) THEN  
*   WRITE(*,*) ' *** CE FICHIER CONTIENT UN TROP GRAND NOMBRE DE PAR  
   AMETRES *** '  
   STOP  
ENDIF
```

c *** constitution des noms de fichiers a partir des codes lus dans le
c *** fichier source

```
DO 150 J=1,NP
```

```
  READ(99,900)NOMPAR
```

```
  IF (NOMPAR(5:5).EQ.' ') THEN
```

c *** cas ou on doit changer de ligne pour continuer a lire le numero du
c *** parametre

```
    DO 100 K=1,5
```

```
      IF (NOMPAR(K:K).EQ.' ') THEN
```

```
        IM=K-1  
        GOTO 110
```

```
      ENDIF
```

100 CONTINUE

110 DO 120 L=1,5-IM

```
      READ(99,'(A1\)' )CAR  
      NOMPAR(IM+L:IM+L)=CAR
```

120 CONTINUE

```
      READ(99,'(4X\)' )
```

```
    ENDIF
```

```
  JJ=1
```

```
c *** on essaie de trouver le terme qui correspond au code
130 IF (NOMPAR.EQ.CA(JJ)) GOTO 140
      IF (JJ.GE.460) THEN
            WRITE(*,*) 'CODE ANALYTIQUE INCONNU'
            STDF
      ENDIF
      JJ=JJ+1
      GOTO 130
140 PARLET(J)=PL(JJ)
```

```
150 CONTINUE
```

```
900 FORMAT(A5,3X\)
```

```
DO 160 I=1,NP
      NOMSFICH(I)=' '
```

```
160 CONTINUE
      CALL QSMODE(ITMODE)
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QBORD(ITCOLOR)
```

```
c *** lecture de l'identification de la station
      WRITE(*,*)
      WRITE(*,*) 'Le fichier d'entre...'
      WRITE(*,*) ' '
      WRITE(*,*) '1) est un fichier original en provenance de Naquadat'
      WRITE(*,*) '2) a t manipul l'aide d'un diteur de texte '
      WRITE(*,*) ' '
      WRITE(*, '(A20\)' ) ' Votre choix : [1]'
      REF = '1'
      READ(*, '(A1)' )REF
      IREST = NP
      WRITE(*,*) ' '
      WRITE(*,*) ' Veuillez patienter un instant ...'
      WRITE(*,*) ' '
```

```
165 IF (REF.EQ.'1') THEN
      READ(99, '(A12,A24)' )STA1,DASTA
      ELSE
      READ(99, '(/,A12,A24)' )STA1,DASTA
      ENDIF
      STA2 = STA1
```

```
C
c *** s'il y a plus de 13 fichiers a traiter, il faudra y aller en GROUPES
C
```

```
IF (IREST.GT.13) THEN
      IREST = IREST - 13
      KF1 = 13
      ELSE
      KF1=IREST
      IREST = 0
      ENDIF
```

```
170 STA1 = STA2
      NOSTA=STA1(7:12)
      DO 180 I=1,NP
            NOMSFICH(I)(1:6)=NOSTA
            IF ((PARLET(I)(2:2)).EQ.' ') THEN
                  IBL=1
            ELSE
                  IF ((PARLET(I)(3:3)).EQ.' ') THEN
```

```

      ELSE          IBL=2
      ENDIF        IBL=3
    ENDIF
    NOMSFICH(I)(7:7)='.'
c *** on ajoute l'extension correspondant au type de parametre
    NOMSFICH(I)(8:8+IBL)=PARLET(I)
180  CONTINUE
    BACKSPACE(99)
    DO 190 I=1,KF1
190  OPEN(I,FILE=NOMSFICH(I),STATUS='NEW')
200  READ(99,'(A12,A24\)',END=240)STA2,DASTA
    IF (STA2.NE.STA1) THEN
      DO 210 I=1,KF1
        IF (NVAL(I).EQ.0) THEN
          CLOSE(I,STATUS='DELETE')
        ELSE
          WRITE(*,*) NOMSFICH(I),NVAL(I)
          NVAL(I)=0
          CLOSE(I)
        ENDIF
      CONTINUE
      WRITE(*,*) ' '
      GOTO 170
    ENDIF
    DO 215 I=1,NP-1
215  CONTINUE
      READ(99,'(F12.5,1X,\)') V(I)
    CONTINUE
      READ(99,'(F12.5)') V(NP)
220  CONTINUE
    DO 230 I=1,KF1
      WRITE(I,'(A36,F12.6)')DASTA,V(NP-IREST+I)
      IF (V(NP-IREST+I).GE.0) THEN
        NVAL(I)=NVAL(I)+1
      ENDIF
230  CONTINUE
      GOTO 200
    DO 250 I=1,KF1
      IF (NVAL(I).EQ.0) THEN
        CLOSE(I,STATUS='DELETE')
      ELSE
        WRITE(*,*) NOMSFICH(I),NVAL(I)
        NVAL(I)=0
        CLOSE(I)
      ENDIF
250  CONTINUE
    IF (IREST.GT.0) THEN
      REWIND(99)
      READ(99,'(/)')
      GOTO 165
    ENDIF
    CLOSE(99)
    RETURN
  END
```

SUBROUTINE DIV2

C *** -----

REAL DEB,FARAM(30),V(15),VAL
INTEGER AN,NVAL(15)
LOGICAL FASVIDE
CHARACTER*1 CAR,REP
CHARACTER*6 NOSTA
CHARACTER*3 PL(460),PARLET(30)
CHARACTER*5 CA(460)
CHARACTER*12 BID,B12,B1,B2
CHARACTER*12 STA1,STA2
CHARACTER*20 FDIV,NOMSFICH(15),NOMPAR
CHARACTER*24 DASTA

COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH/ IGMODE,IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,
\$ ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2
COMMON /BLOC1/ FDIV
COMMON /BLOC2/ PL,CA,PARLET

c *** lecture du nombre de parametres presents dans le fichier

READ(99,'(3X,I2,1X\)')NP

\$ IF (NP.GT.25) THEN
WRITE(*,*)' *** CE FICHIER CONTIENT UN TROP GRAND NOMBRE DE PAR
AMETRES ***'
STOP
ENDIF

c *** constitution des noms de fichiers a partir des codes lus dans le
c *** fichier source

DO150 J=1,NP

READ(99,900)NOMPAR

IF (NOMPAR(5:5).EQ.' ') THEN

c *** cas ou on doit changer de ligne pour continuer a lire le numero du
c *** parametre

100 DO 100 K=1,5
110 IF (NOMPAR(K:K).EQ.' ') THEN
IM=K-1
GOTO 110
ENDIF
CONTINUE
DO 120 L=1,5-IM
120 READ(99,'(A1\)')CAR
NOMPAR(IM+L:IM+L)=CAR
CONTINUE
READ(99,'(4X\)')
ENDIF
JJ=1

c *** on essaie de trouver le terme qui correspond au code

130 IF (NOMPAR.EQ.CA(JJ)) GOTO 140

IF (JJ.GE.460) THEN
WRITE(*,*)'CODE ANALYTIQUE INCONNU'
STOP
ENDIF

```
      JJ=JJ+1
      GOTO 130
140   PARLET(J)=PL(JJ)
150   CONTINUE
900   FORMAT(A5,3X\
      DO 160 I=1,NP
          NOMSFICH(I)=' '
160   CONTINUE
      CALL QSMODE(ITMODE)
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QBORD(ITCOLOR)
c *** lecture de l'identification de la station
      WRITE(*,*)
      WRITE(*,*) 'Le fichier d'entre...'
      WRITE(*,*) ' '
      WRITE(*,*) '1) est un fichier original en provenance de Naquadat'
      WRITE(*,*) '2) a t manipul l'aide d'un diteur de texte '
      WRITE(*,*) ' '
      WRITE(*,*(A20\)) ' Votre choix : [1]'
      REP = '1'
      READ(*,*(A1))REP
      IF (REP.EQ.'1') THEN
          READ(99,*(A12,A24))STA1,DASTA
      ELSE
          READ(99,*(/,A12,A24))STA1,DASTA
      ENDIF
      STA2 = STA1
      WRITE(*,*) ' '
      WRITE(*,*) '1) Possede des CR a la fin de chaque ligne de 80 cara.'
      WRITE(*,*) '2) Une ligne = un enregistrement '
      WRITE(*,*) ' '
      WRITE(*,*(A20\)) ' Votre choix : [2]'
      WRITE(*,*) ' '
      REP = '2'
      READ(*,*(A1))REP
      IF (REP.EQ.'1') THEN
          CR = 1
      ELSE
          CR = 0
      ENDIF
      WRITE(*,*) ' Veuillez patienter un instant ...'
      WRITE(*,*) ' '
c *** s'il y a plus de 13 fichiers a traiter, il faudra y aller en deux
c *** groupes
      IF (NP.LE.13) THEN
          KF1=NP
          KF2=0
      ELSE
          KF1=13
          KF2=NP-13
      ENDIF
170   STA1 = STA2
      NOSTA=STA1(7:12)
      DO 180 I=1,NP
          NOMSFICH(I)(1:6)=NOSTA
```



```
IF ((PARLET(I)(2:2)).EQ.' ') THEN
  IBL=1
ELSE
  IF ((PARLET(I)(3:3)).EQ.' ') THEN
    IBL=2
  ELSE
    IBL=3
  ENDIF
ENDIF
NOMSFICH(I)(7:7)='.'
c *** on ajoute l'extension correspondant au type de parametre
NOMSFICH(I)(8:8+IBL)=PARLET(I)
180 CONTINUE
DO 190 I=1,KF1
  OPEN(I,FILE=NOMSFICH(I),STATUS='NEW')
190 CONTINUE
  B12='
  BACKSPACE(99)
200 READ(99,'(A12,A24\)',END=240)STA2,DASTA
  IF (STA2.NE.STA1) THEN
    DO 210 I=1,KF1
      IF (NVAL(I).EQ.0) THEN
        CLOSE(I,STATUS='DELETE')
      ELSE
        WRITE(*,*) NOMSFICH(I),NVAL(I)
        NVAL(I)=0
        CLOSE(I)
      ENDIF
    CONTINUE
    WRITE(*,*)
    GOTO 170
  ENDIF
DO 220 I=1,KF1
  IF (CR.EQ.1) THEN
    IF (I.EQ.4) THEN
      B1=B12
      B2=B12
      READ(99,'(A5)')B1
      IF (NP.NE.4) THEN
        READ(99,'(A7,1X\)' )B2
      ELSE
        READ(99,'(A7,1X)')B2
      ENDIF
      BID=B12
      WRITE(BID,'(A5,A7)')B1,B2,B1
      READ(BID,'(F12.6)')V(4)
    ELSEIF (I.EQ.10) THEN
      B1=B12
      B2=B12
      READ(99,'(A7)')B1
      IF (NP.NE.10) THEN
        READ(99,'(A5,1X\)' )B2
      ELSE
        READ(99,'(A5,1X)')B2
      ENDIF
    ENDIF
  ENDIF
ENDIF
```

```

      BID=B12
      WRITE (BID, '(A7,A5)') B1,B2
      READ (BID, '(F12.6)') V(10)
    —ENDIF
  —ENDIF
  IF (I.EQ.KF1) THEN
    IF (NP.LE.15) THEN
      READ (99, '(F12.6)') V(I)
    —ELSEIF (NP.LE.21) THEN
      READ (99, '(F12.6,/)') V(I)
    —ELSE
      READ (99, '(F12.6,/,/)') V(I)
    —ENDIF
  —ELSE
    READ (99, '(F12.6,1X\')') V(I)
  —ENDIF
220 —CONTINUE
  DO 230 I=1,KF1
    WRITE (I, '(A36,F12.6)') DASTA,V(I)
    IF (V(I).GE.0) THEN
      NVAL(I)=NVAL(I)+1
    —ENDIF
230 —CONTINUE
    GOTD 200
240 —DO 250 I=1,KF1
    IF (NVAL(I).EQ.0) THEN
      CLOSE (I,STATUS='DELETE')
    —ELSE
      WRITE (*,*) NOMSFICH(I),NVAL(I)
      NVAL(I)=0
      CLOSE (I)
    —ENDIF
250 —CONTINUE
```

c *** traitement du deuxieme groupe de fichiers

```

  DO 260 I=1,KF2
    OPEN (I,FILE=NOMSFICH(I+13),STATUS='NEW')
260 —CONTINUE

  IF (KF2.EQ.0) THEN
    CLOSE (99)
    RETURN
  —ENDIF
  REWIND (99)
  IF (NP.GT.8.AND.NP.LE.17) THEN
    READ (99, '(/)')
  —ELSEIF (NP.GT.17) THEN
    READ (99, '(/,/)')
  —ENDIF

270  READ (99, '(A36,/)') ,END=300) DASTA
    READ (99, '(45X\')')

  DO 280 I=1,KF2
    IF (I.EQ.3) THEN
```

```
      B1=B12
      B2=B12
      READ(99,'(A9)')B1
      IF (KF2.NE.3) THEN
         READ(99,'(A3,1X\)' )B2
      ELSE
         READ(99,'(A3,1X)')B2
      ENDIF
      BID=B12
      WRITE(BID,'(A9,A3)')B1,B2
      READ(BID,'(F12.6)')V(3)
      ELSEIF (I.EQ.9) THEN
      B1=B12
      B2=B12
      READ(99,'(A11)')B1
      IF (KF2.NE.9) THEN
         READ(99,'(A1,1X\)' )B2
      ELSE
         READ(99,'(A1,1X)')B2
      ENDIF
      BID=B12
      WRITE(BID,'(A11,A1)')B1,B2
      READ(BID,'(F12.6)')V(9)
      ELSEIF (I.EQ.KF2) THEN
      READ(99,'(F12.6)')V(I)
      ELSE
      READ(99,'(F12.6,1X\)' )V(I)
      ENDIF
280 CONTINUE
      DO 290 I=1,KF2
         WRITE(I,'(A36,F12.6)')DASTA,V(I)
290 CONTINUE
      GOTO 270
300 DO 310 I=1,KF2
         CLOSE(I)
310 CONTINUE
      CLOSE(99)
      RETURN
      END
```

SUBROUTINE DIV3

C

```
REAL          PARAM(30)
INTEGER       IDKCOL(30)
CHARACTER*1   REP
CHARACTER*20  FDIV,NOMSFICH(30),FORM,FORM1,FORM2
```

```
COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH/ IGMODE,IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,
$ ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2
COMMON /BLOC1/ FDIV
```

```
75 WRITE(*,*) '
WRITE(*,*)' Entrez le nombre TOTAL de colonnes '
```

```
WRITE(*,'(A34\)' ) (INCLUANT LA DATE) (MAX = 30):  
READ(*,'(I2)',ERR=75)NCOL  
WRITE(*,*)  
80 CALL QSMODE(ITMODE)  
CALL QCLEAR(ITBACK,ITFORE)  
CALL QBORD(ITCOLOR)  
WRITE(*,*)  
WRITE(*,*) ' A l'entree:  
WRITE(*,*)  
WRITE(*,*) ' 1) La date est au debut de la ligne (3 premieres colo  
$ nnes)  
WRITE(*,*) ' 2) La date est a la fin de la ligne (3 dernieres colo  
$ nnes)  
WRITE(*,*) ' 3) Il n y a pas de date (donnees equidistantes)  
WRITE(*,*)  
WRITE(*,'(A20\)' ) ' Votre choix : '  
READ (*,*,ERR=80) IREP  
IF (IREP.EQ.3) THEN  
IDATE = -9  
ELSEIF (IREP.EQ.2) THEN  
IDATE = NCOL - 2  
ELSEIF (IREP.EQ.1) THEN  
IDATE = 1  
ELSE  
WRITE (*,*) ' Nombre illegal '  
GOTO 80  
ENDIF  
85 WRITE(*,*)  
WRITE(*,'(A49\)',ERR=85) ' Entrez la colonne du debit (0 = aucun  
$ debit) : '  
READ (*,*,ERR=85) ICOLDEBIT  
IF (ICOLDEBIT.LT.0.OR.ICOLDEBIT.GT.NCOL) GOTO 85  
IF (ICOLDEBIT.EQ.IDATE ) GOTO 85  
IF (ICOLDEBIT.EQ.IDATE+1) GOTO 85  
IF (ICOLDEBIT.EQ.IDATE+2) GOTO 85  
C  
C **** attention ici des format qui peuvent changer  
C  
C formats de la donnee a la sortie  
IF (ICOLDEBIT.NE.0) THEN  
IF (IDATE.EQ.0) THEN  
FORM = '(F12.6,1X,\)'  
ELSE  
FORM = '(F8.4,1X,\)'  
ENDIF  
90 CALL QSMODE(ITMODE)  
CALL QCLEAR(ITBACK,ITFORE)  
CALL QBORD(ITCOLOR)  
WRITE(*,*)  
WRITE(*,*) ' Il faut que les debits soit en Metre cube/Sec '  
WRITE(*,*)  
WRITE(*,*) ' Voulez-vous donc faire une conversion sur '  
WRITE(*,*) ' vos debits ? '  
WRITE(*,*)  
WRITE(*,*) ' EX : 1 litre 0.001 metre cube'
```

```
WRITE(*,*) '----- = ----- = 0.0002 '  
WRITE(*,*) '      min.           60 sec '  
WRITE(*,*) ' '  
WRITE(*,*) ' Donc DEBIT = DEBIT * 0.0002 '  
WRITE(*,*) ' '  
WRITE(*,*) ' Conversion sur les debits ? '  
WRITE(*,*) ' '  
WRITE(*,*) ' 1 = pas de conversion '  
WRITE(*,*) ' '  
WRITE(*,*(A18\)) ' DEBIT = DEBIT * '  
READ (*,*,ERR=90) CONVDEBIT  
IF (CONVDEBIT.LT.0) THEN  
    GOTO 90  
ENDIF  
WRITE(*,*) ' '  
ELSE  
    IF (IDATE.EQ.0) THEN  
        FORM = '(*) '  
    ELSE  
        FORM = '(FB,4) '  
    ENDIF  
ENDIF  
95 CALL QSMODE(ITMODE)  
CALL QCLEAR(ITBACK,ITFORE)  
CALL QBORD(ITCOLOR)  
WRITE(*,*) ' '  
WRITE(*,*) ' Il faut que les CHARGES soit en Kg/jour '  
WRITE(*,*) ' et que les CONCENTRATIONS soit en mg/Litre '  
WRITE(*,*) ' '  
WRITE(*,*) ' Voulez-vous donc faire une conversion sur '  
WRITE(*,*) ' vos parametres? '  
WRITE(*,*) ' '  
WRITE(*,*) ' EX : 1 gramme           0.001 Kgramme '  
WRITE(*,*) '      ----- = ----- = 0.0240 '  
WRITE(*,*) '      heure.           0.0417 jour '  
WRITE(*,*) ' '  
WRITE(*,*) ' Donc PARAMETRE = PARAMETRE * 0.024 '  
WRITE(*,*) ' '  
WRITE(*,*) ' Conversion sur les parametres ? '  
WRITE(*,*) ' '  
WRITE(*,*) ' 1 = pas de conversion '  
WRITE(*,*) ' '  
WRITE(*,*(A26\)) ' PARAMETRE = PARAMETRE * '  
READ (*,*,ERR=95) CONVPARAM  
IF (CONVPARAM.LE.0) THEN  
    GOTO 95  
ENDIF  
WRITE(*,*) ' '  
98 IF (IDATE.GT.0) THEN  
    CALL QSMODE(ITMODE)  
    CALL QCLEAR(ITBACK,ITFORE)  
    CALL QBORD(ITCOLOR)  
    WRITE(*,*) ' '  
    WRITE(*,*) ' A la sortie: '  
    WRITE(*,*) ' '
```

```
WRITE(*,*) ' 1) La date est le premier champ'  
WRITE(*,*) ' 2) Pas de date (donnees equidistantes)'  
WRITE(*,*) '  
WRITE(*, '(A20\)' ) ' Votre choix : '  
READ (*,*,ERR=98) IMPDAT  
IF (IMPDAT.NE.2.AND.IMPDAT.NE.1) GOTO 98  
ELSE  
    IMPDAT = 2  
ENDIF  
CALL QSMODE(ITMODE)  
CALL QCLEAR(ITBACK,ITFORE)  
CALL QBORD(ITCOLOR)  
NLEC=0  
DO 100 J=1,NCOL  
    WRITE(*,*) '  
    IF (J.EQ.ICOLDEBIT) THEN  
        WRITE(*,*) ' COLONNE DES DEBITS '  
    ELSEIF (J.EQ.IDATE ) THEN  
        WRITE(*,*) ' COLONNE DES ANNEES '  
    ELSEIF (J.EQ.IDATE+1) THEN  
        WRITE(*,*) ' COLONNE DES MOIS '  
    ELSEIF (J.EQ.IDATE+2) THEN  
        WRITE(*,*) ' COLONNE DES JOURS '  
    ELSE  
        WRITE(*, '(A17\)' ) ' VOULEZ-VOUS LA '  
        WRITE(*, '(I2\)' ) J  
        IF (J.EQ.1) THEN  
            WRITE(*, '(A3\)' ) 'ere '  
        ELSE  
            WRITE(*, '(A3\)' ) 'eme '  
        ENDIF  
        IF (REP.EQ.' ') THEN  
            REP = '0'  
            GOTO 210  
        ENDIF  
        WRITE(*, '(A22\)' ) ' COLONNE ? (O/N) [O] '  
        READ (*, '(A1)',ERR=200) REP  
        IF (REP.NE.'N'.AND.REP.NE.'n'.AND.REP.NE.'o'.AND.REP.NE.'O')  
            GOTO 200  
        IF (REP.EQ.'N'.OR.REP.EQ.'n') THEN  
            IOKCOL(J) = 0  
        ELSE  
            WRITE(*, '(A19\)' ) ' Entrez le nom du '  
            WRITE(*, '(A30\)' ) ' fichier pour cette colonne : '  
            READ(*, '(A20\)' ) NOMSFICH(J)  
            IOKCOL(J) = 1  
        ENDIF  
    ENDIF  
CONTINUE  
IFAIT = 0  
IFAIT1 = 0  
IFAIT2 = 0  
WRITE(*,*) '  
WRITE(*,*) ' LE TRAITEMENT EST EN COURS'
```

```
WRITE(*,*) ' '
WRITE(*,*) ' '
110 NBOPEN = 0
    IFAIT1 = IFAIT
120 IF (NBOPEN.LT.13.AND.IFAIT1.LT.NCOL) THEN
    IFAIT1 = IFAIT1 + 1
    IF (IOKCOL(IFAIT1).EQ.1) THEN
        NBOPEN = NBOPEN + 1
        OPEN(NBOPEN,FILE=NOMSFICH(IFAIT1),STATUS='NEW')
    ENDIF
    GOTO 120
ENDIF
130 READ(99,*,END=150)(PARAM(I),I=1,NCOL)
C *** format de la date si presente
910 FORMAT (12X,I2,1X,I2,1X,I2,16X,\)
C *** format du debit si present
920 FORMAT (F8.4)
    IFAIT2 = IFAIT
    NBOPEN = 0
135 IF (NBOPEN.LT.13.AND.IFAIT2.LT.NCOL) THEN
    IFAIT2 = IFAIT2 + 1
    IF (IOKCOL(IFAIT2).EQ.1) THEN
        NBOPEN = NBOPEN + 1
        IF (IMPDAT.EQ.1) THEN
            IAN = AINT(PARAM(IDATE))
            IMO = AINT(PARAM(IDATE+1))
            IJR = AINT(PARAM(IDATE+2))
            WRITE(NBOPEN,910) IAN,IMO,IJR
        ENDIF
        UNPARAM = PARAM(IFAIT2) * CONVPARAM
        WRITE(NBOPEN,FORM) UNPARAM
        IF (ICOLDEBIT.NE.0) THEN
            DEBIT = PARAM(ICOLDEBIT)*CONVDEBIT
            WRITE(NBOPEN,920) DEBIT
        ENDIF
    ENDIF
    GOTO 135
ENDIF
    GOTO 130
150 DO 160 I=1,NBOPEN
    CLOSE(I)
160 CONTINUE
IF (IFAIT1.LT.NCOL) THEN
    REWIND(99)
    IFAIT = IFAIT1
    GOTO 110
ELSE
    CLOSE(99)
ENDIF
CALL QBEEP
RETURN
END
```

CONF.FOR

ELSEIF, QBEEP, QSMODE, QBORD, QCLEAR, QDATE
MOD, QCOLOR, QPLOT, QSETUP, QINTERP, QTABL
QXAXIS, QYAXIS, QCMOV, QOVSCN, QPTABL, QRTOI
QLINE, QPTXTB, QPTXTC

AFFCOULEURS CONF.FOR 112 x4
112 CALL AFFCOULEURS
301 CALL AFFCOULEURS
374 CALL AFFCOULEURS

CONF.FOR 527
SUBROUTINE AFFCOULEURS

YFUN CONF.FOR 440
440 YARRAY(N)=YFUN(XX)

CONF.FOR 503
REAL FUNCTION YFUN(T)

ATAN, ABS, EXP, COS

#STORA GE:2

PROGRAM SETUP

```
C -----  
C *****  
C Titre : Configuration des modes texte et graphique  
C  
C Fonctions  
C  
C . Demander les choix de l'usager concernant les couleurs qu'il  
C . desire employer en mode texte  
C . Demander egalement ses choix concernant la resolution, les  
C . couleurs et les symboles a utiliser en mode graphique  
C . Produire un exemple a l'aide des choix recueillis  
C . Recommencer les demandes jusqu'a l'obtention de la satisfac-  
C . tion de l'usager, puis creer un fichier "setup.pc" pour regrou-  
C . per toutes ces informations. Ce fichier sera tour a tour lu par  
C . chacun des programmes d'analyse de serie de qualite de l'eau  
C . et permettra ainsi de configurer les modes texte et graphique  
C . selon les choix retenus.  
C  
C *****
```

```
REAL C(4,50),XARRAY(50),YARRAY(50)  
REAL XEX(10),YEX(10)  
EXTERNAL YFUN  
CHARACTER*1 TOUCHE  
INTEGER*4 ATTENTE  
DATA XEX/1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0/  
DATA YEX/11.0,23.0,29.0,39.0,54.0,61.0,68.0,79.0,91.0,100.0/
```

C *** affichage d'information preleminaire

```
5 FORMAT(15X,A50,14X)  
WRITE(*, '( /, / )')  
WRITE(*,5) '*****'  
WRITE(*,5) '  
WRITE(*,5) 'CE PROGRAMME CONFIGURE LES DIFFERENTES PARTIES DU '  
WRITE(*,5) 'LOGICIEL SELON LES CARACTERISTIQUES DE VOTRE ORDI-'  
WRITE(*,5) 'NATEUR PERSONNEL. '  
WRITE(*,5) '  
WRITE(*,5) 'POUR CONTROLER LES COULEURS, LE DRIVER "ANSI.SYS" '  
WRITE(*,5) 'NE DOIT PAS ETRE INSTALLE DANS LE FICHIER DE CON- '  
WRITE(*,5) 'FIGURATION COMME DEVICE. SI "ANSI.SYS" EST '  
WRITE(*,5) 'INSTALLE SEUL LE CURSEUR PRENDRA LA COULEUR '  
WRITE(*,5) 'CHOISIE. '  
WRITE(*,5) '  
WRITE(*,5) 'POUR OBTENIR DES GRAPHIQUES IMPRIMES, ON DOIT AVOIR '  
WRITE(*,5) 'EXECUTE LA COMMANDE "GRAPHICS" SOUS DOS '  
WRITE(*,5) '  
WRITE(*,5) 'LA SORTIE DE CE PROGRAMME CREE UN FICHIER SETUP.PC '  
WRITE(*,5) 'QUI DOIT ETRE PRESENT POUR PERMETTRE UNE EXECUTION '
```

```
WRITE(*,5) 'CORRECTE DES DIFFERENTS PROGRAMMES.  
WRITE(*, '(/,/))'  
WRITE(*,5) '*****'  
WRITE(*, '(/,/))'  
WRITE(*,5) ' ...TAPEZ <RETURN> POUR CONTINUER...  
READ(*,10) TOUCHE  
10 FORMAT(A1)  
20 WRITE(*, '(/,/,/,/,/))'
```

c *** configuration du mode texte

```
WRITE(*,5) ' CONFIGURATION DU MODE TEXTE  
WRITE(*,5) ' -----  
WRITE(*,5) '  
WRITE(*,*) ' '  
WRITE(*,5) 'EN MODE TEXTE,VOTRE SYSTEME SUPPORTE UN DES ECRANS  
WRITE(*,5) 'SUIVANTS :  
WRITE(*,5) '  
WRITE(*,5) '  
WRITE(*,5) ' 1) 80 X 25 NOIR ET BLANC,1 MONITEUR  
WRITE(*,5) '  
WRITE(*,5) ' 2) 80 X 25 COULEUR  
WRITE(*,5) '  
WRITE(*,5) ' 3) 80 X 25 NOIR ET BLANC,2 MONITEURS  
WRITE(*,5) '  
WRITE(*,5) '-----  
WRITE(*,5) '  
WRITE(*,5) '  
WRITE(*,*) ' '  
WRITE(*,15) 'ENTREZ LE NUMERO CORRESPONDANT A VOTRE ECRAN :
```


15

```
FORMAT(15X,A57\  
READ(*, '(I1)')IREP  
IF (IREP.EQ.1) THEN  
ITMODE=02  
ELSEIF (IREP.EQ.2) THEN  
ITMODE=03  
ELSEIF (IREP.EQ.3) THEN  
ITMODE=07  
ELSE  
WRITE(*,5) '  
WRITE(*,5) ' * * * * *  
WRITE(*,5) ' *** CHOIX INVALIDE.RECOMMENCEZ. ***  
WRITE(*,5) ' * * * * *  
WRITE(*,*) ' '
```

c *** on attend que l'utilisateur ait le temps de voir le message d'erreur

25

```
DO 25 ATTENTE=1,100000  
CONTINUE
```

c *** on fait entendre un son (beep) pour signaler l'erreur

```
CALL QBEEP  
GOTO 20
```

ENDIF

c *** a moins qu'il ne s'agisse du noir et blanc, on affiche alors
c *** le choix des couleurs

```
30   IF (IREP.NE.2) GOTO 100
    CALL AFFCOULEURS
40   WRITE(*,'(15X,A32\)' ) 'COULEUR DES LETTRES (0 A 15) : '
    READ(*,80)ITFORE
    IF ((ITFORE.LT.0).OR.(ITFORE.GT.15)) THEN
        WRITE(*,90)
        GOTO 40
    ENDIF
50   WRITE(*,'(15X,A26\)' ) 'COULEUR DU FOND (0 A 7) : '
    READ(*,80)ITBACK
    IF ((ITBACK.LT.0).OR.(ITBACK.GT.7)) THEN
        WRITE(*,90)
        GOTO 50
    ENDIF

c *** le cadre est la bordure qui contourne l'ecran
60   WRITE(*,'(15X,A27\)' ) 'COULEUR DU CADRE (0 A 15) : '
    READ(*,80)ITCOLOR
    IF ((ITCOLOR.LT.0).OR.(ITCOLOR.GT.15)) THEN
        WRITE(*,90)
        GOTO 60
    ENDIF
80   FORMAT(I2)
90   FORMAT('          * * * REPONSE INVALIDE.RECOMMENCEZ. * * *')

c *** essai du mode texte avec les couleurs specifiees
    CALL QSMODE(ITMODE)
    CALL QBORD(ITCOLOR)
    CALL QCLEAR(ITBACK,ITFORE)
    CALL QDATE(IAN,IMO,IJO)
95   FORMAT(11X, '                SESSION DU ',I2,'/',I2,'/',I4,10X)
    WRITE(*,'(/,/)')
    WRITE(*,95)IJO,IMO,IAN
    WRITE(*,'(/,/,/,/)')
    DO 97 I=1,10
        WRITE(*,*) '                * * * ESSAI DE COULEURS EN MODE TEXTE.*
#         * *
97   CONTINUE
    WRITE(*,'(/,/,/,/,/)')
98   WRITE(*,15) '                ETES-VOUS SATISFAIT ? (O/N)
#
    READ(*,10)TOUCHE
    IF (TOUCHE.NE.'N'.AND.TOUCHE.NE.'n'.AND.TOUCHE.NE.'o'.AND.
#   TOUCHE.NE.'O') THEN
        CALL QBEEP
        GOTO 98
    ENDIF

C *** si l'usager n'est pas satisfait,refaire la demande
    IF (TOUCHE.EQ.'N'.OR.TOUCHE.EQ.'n') GOTO 30
    GOTO 110

C *** donner les valeurs par default si on a choisi le noir et blanc
100  IF (IREP.EQ.3) CALL QBORD(1)
    ITFORE=07
```

```
ITBACK=00  
ITCOLOR=00  
CALL QSMODE(3)
```

```
c *** ouvrir le fichier destine a regrouper les specifications graphiques  
c *** Y inscrire les choix concernant le mode texte
```

```
110 OPEN(1,FILE='SETUP.PC',STATUS='NEW')  
WRITE(1,120)ITMODE,ITFORE,ITBACK,ITCOLOR  
120 FORMAT(4I2)
```

```
c *** configuration du mode graphique
```

```
WRITE(*,*)'  
130 WRITE(*,'(/,/))'  
WRITE(*,5)' CONFIGURATION DU MODE GRAPHIQUE  
WRITE(*,5)' -----  
WRITE(*,*)'  
WRITE(*,5)' EN MODE GRAPHIQUE,VOTRE INTERFACE VOUS PERMET UNE  
WRITE(*,5)' DES DEFINITIONS SUIVANTES :  
WRITE(*,5)'  
WRITE(*,5)' 1) 320 X 200 NOIR ET BLANC  
WRITE(*,5)'  
WRITE(*,5)' 2) 320 X 200 COULEUR(4)  
WRITE(*,5)'  
WRITE(*,5)' 3) 640 X 200 NOIR + UNE COULEUR  
WRITE(*,5)'  
WRITE(*,5)' 4) 320 X 200 COULEUR(16),EMULATION EGA  
WRITE(*,5)'  
WRITE(*,5)' 5) 640 X 200 COULEUR(16),EMULATION EGA  
WRITE(*,5)'  
WRITE(*,5)' 6) 640 X 350 COULEUR(4/16),EMULATION EGA  
WRITE(*,5)'  
WRITE(*,5)'  
WRITE(*,5)'  
WRITE(*,5)' -----  
WRITE(*,5)'  
WRITE(*,15)' ENTREZ LE NUMERO CORRESPONDANT :
```

\$

```
c *** lire la reponse de l'utilisateur  
READ(*,'(I1)')IREP
```

```
c *** selon la reponse fournie par l'utilisateur,definir les dimensions  
c *** de la fenetre graphique associee a la resolution choisie
```

```
IF (IREP.EQ.1) THEN  
    IGMODE=05  
    JRDW1=29  
    JROW2=180  
    JCOL1=50  
    JCOL2=290  
ELSEIF (IREP.EQ.2) THEN  
    IGMODE=04  
    JRDW1=30  
    JROW2=170  
    JCOL1=50  
    JCOL2=290  
ELSEIF (IREP.EQ.3) THEN
```

```

        IGMODE=06
        JROW1=37
        JROW2=180
        JCOL1=55
        JCOL2=600
ELSEIF (IREP.EQ.4) THEN
        IGMODE=13
        JROW1=30
        JROW2=170
        JCOL1=50
        JCOL2=290
ELSEIF (IREP.EQ.5) THEN
        IGMODE=14
        JROW1=30
        JROW2=180
        JCOL1=80
        JCOL2=600
ELSEIF (IREP.EQ.6) THEN
        IGMODE=16
        JROW1=50
        JROW2=310
        JCOL1=65
        JCOL2=600
ELSE
        WRITE(*,5)'
        WRITE(*,5)' * * * * *
        WRITE(*,5)' *** CHOIX INVALIDE.RECOMMENCEZ. ***
        WRITE(*,5)' * * * * *
        WRITE(*,*)'
C *** permettre a l'utilisateur de voir le message d'erreur (attente)
140      DO 140 ATTENTE=1,100000
        CONTINUE
        CALL QBEEP
        GOTO 130
ENDIF

C *** donner les valeurs par default pour les cas ou l'utilisateur ne doit
c *** pas tout preciser :
c *** couleur de fond
135      IGBACK=00
c *** numero de la palette
        IPAL=01
        IGCOLOR=03
c *** couleur pour tracer les lignes
        iline=03
c *** couleur du lettrage sur le graphique
        iletr=03
c *** couleur secondaire (auxiliaire pour tracer certaines courbes)
        isec=03
c *** couleur du cadre
        IGBORD=00
c *** couleur du symbole pour tracer la courbe
        KIRSYM=07
```

```
c *** choix du symbole pour tracer la courbe(caractere ascii)
WRITE(*, '(/,/)' )
145 WRITE(*, '(15X,A41)') 'CHOIX DU SYMBOLE POUR TRACER LA COURBE : '
WRITE(*,*) '
WRITE(*,5) '          1) *****
WRITE(*,5) '          2) -----
WRITE(*,5) '          3) .....
WRITE(*,*) '
WRITE(*, '(15X,A14\)' ) 'VOTRE CHOIX : '
READ(*, '(I1)') ICHOIX
IF ( ICHOIX.LE.0.OR.ICHOIX.GT.3) THEN
    WRITE(*,90)
    GOTO 145
ENDIF
```

```
c *** donner le numero ascii correspondant
IF ( ICHOIX.EQ.1) THEN
    ISYMBL=42
ELSEIF ( ICHOIX.EQ.2) THEN
    ISYMBL=45
ELSE
    ISYMBL=46
ENDIF
```

```
c *** offrir les couleurs disponibles
IF ( IREP.EQ.2) THEN
150 CALL AFFCOULEURS
    WRITE(*, '(15X,A27\)' ) 'COULEUR DE FOND (0 A 15) : '
    READ(*,80) IGBACK
```

```
c *** validation
IF (( IGBACK.LT.0).OR.( IGBACK.GT.15)) THEN
    WRITE(*,90)
    GOTO 150
ENDIF
```

```
c *** eviter qu'on choisisse la meme couleur pour le fond et pour le
c *** caractere (on oblige la palette opposee)
```

```
IF ( IGBACK.EQ.3.OR. IGBACK.EQ.5.OR. IGBACK.EQ.7) THEN
    ICHOIX=1
    GOTO 170
ENDIF
IF ( IGBACK.EQ.2.OR. IGBACK.EQ.4.OR. IGBACK.EQ.6) THEN
    ICHOIX=2
    GOTO 175
ENDIF
```

```
c *** choix de la palette
160 WRITE(*,*) '
WRITE(*,5) 'CHOIX DE LA PALETTE :
WRITE(*,5) '
WRITE(*,5) '1) VERT / ROUGE / JAUNE
WRITE(*,5) '2) AQUA / MAGENTA / BLANC
WRITE(*,5) '
WRITE(*, '(15X,A23\)' ) 'VOTRE CHOIX (1 OU 2) : '
READ(*, '(I1)') ICHOIX
IF (( ICHOIX.NE.1).AND.( ICHOIX.NE.2)) THEN
```

```
WRITE(*,90)
GOTO 160
ENDIF
170 IF (ICHOIX.EQ.1) THEN
c *** choisir la couleur du symbole de tracage selon la palette 00
WRITE(*,*)
WRITE(*,(15X,A19\))'COULEUR DU PIXEL : '
WRITE(*,*)
IPAL=00
WRITE(*,5)'1) VERT

WRITE(*,5)'2) ROUGE

WRITE(*,5)'3) JAUNE

WRITE(*,5)'
WRITE(*,(15X,A14\))'VOTRE CHOIX : '
READ(*,(I1))IGCOLOR
IF ((IGCOLOR.LT.1).OR.(IGCOLOR.GT.3)) THEN
WRITE(*,90)
GOTO 170
ENDIF
ENDIF
175 IF (ICHOIX.EQ.2) THEN
c *** choisir la couleur du symbole de tracage selon la palette 01
WRITE(*,*)
WRITE(*,(15X,A19\))'COULEUR DU PIXEL : '
WRITE(*,*)
IPAL=01
WRITE(*,5)'1) AQUA

WRITE(*,5)'2) MAGENTA

WRITE(*,5)'3) BLANC

WRITE(*,5)'
WRITE(*,(15X,A14\))'VOTRE CHOIX : '
READ(*,(I1))IGCDLOR
IF ((IGCDLOR.LT.1).OR.(IGCDLOR.GT.3)) THEN
WRITE(*,90)
GOTO 170
ENDIF
ENDIF
ILINE=MOD(IGCOLOR+1,3)+1
ILETR=MOD(IGCOLOR+2,3)+1
ISEC=MOD(IGCOLOR+3,3)+1
KIRSYM=IGCOLOR
ENDIF
IF (IREP.EQ.3) THEN
CALL AFFCOULEURS
WRITE(*,5)'LE FOND DE VOTRE ECRAN SERA DE COULEUR NOIRE.
WRITE(*,*)
180 WRITE(*,(15X,A42\))'CHOISISSEZ LA COULEUR DU PIXEL (0 A 15) :
*
READ(*,80)IGBACK
```

```

      IF ((IGBACK.LT.0).OR.(IGBACK.GT.15)) THEN
        WRITE(*,90)
        GOTO 180
      ENDIF
      KIRSYM=IGBACK
    ENDIF
  c *** offrir les couleurs disponibles
  200 IF (IREP.GE.4) THEN
      CALL AFFCOULEURS
      WRITE(*, '(15X,A30\)' ) 'COULEUR DE BORDURE (0 A 15) : '
      READ(*,80)IGBORD
      IF ((IGBORD.LT.0).OR.(IGBORD.GT.15)) THEN
        WRITE(*,90)
        GOTO 200
      ENDIF
  205 WRITE(*, '(15X,A30\)' ) 'COULEUR DES LIGNES (0 A 15) : '
      READ(*,80)ILINE
      IF ((ILINE.LT.0).OR.(ILINE.GT.15)) THEN
        WRITE(*,90)
        GOTO 205
      ENDIF
  207 WRITE(*, '(15X,A31\)' ) 'COULEUR DU LETTRAGE (0 A 15) : '
      READ(*,80)ILETR
      IF ((ILETR.LT.0).OR.(ILETR.GT.15)) THEN
        WRITE(*,90)
        GOTO 207
      ENDIF
  210 WRITE(*, '(15X,A30\)' ) 'COULEUR SECONDAIRE (0 A 15) : '
      READ(*,80)ISEC
      IF ((ISEC.LT.0).OR.(ISEC.GT.15)) THEN
        WRITE(*,90)
        GOTO 210
      ENDIF
  215 WRITE(*, '(15X,A29\)' ) 'COULEUR DU SYMBOLE (0 A 15): '
      READ(*,80)KIRSYM
      IF ((KIRSYM.LT.0).OR.(KIRSYM.GT.15)) THEN
        WRITE(*,90)
        GOTO 215
      ENDIF
    ENDIF

  c *** monter a l'usager l'effet de ses choix...
      WRITE(*, '(/,/,/,/,/,/,/,/,/,/,/,/)' )
      WRITE(*,5) '
      WRITE(*,*) '-----
      WRITE(*,5) '          *** TEST EN MODE GRAPHIQUE ***
      WRITE(*,5) '-----
      WRITE(*, '(/,/,/,/,/,/,/,/,/,/,/,/)' )
  218 DO 218 ATTENTE=1,70000
      CONTINUE
      CALL GSMODE(IGMODE)

  IF (IREP.LE.3) THEN
  C *** essai en mode CBA
  c *** tracage d'un graphique quelconque :
```



```
c *** definition des points
      NPT=50
      DO 220 N=1,50
          XARRAY(N)= -1.0 +2.0*(N-1)/(NPT-1)
          XX=XARRAY(N)
          YARRAY(N)=YFUN(XX)
220   CONTINUE

c *** tracage du graphique proprement dit
      CALL QCOLOR(IGBACK,IFAL)
      CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,-1.,1.0,-1.,1.0,0.,0.,0,
$       1.0,1.5)
      CALL QSETUP(2,IGCOLOR,ISYMBL,IGCOLOR)
      CALL QINTERP(NPT,XARRAY,YARRAY,C)
      CALL QTABL(0,NPT,XARRAY,YARRAY)
      CALL QXAXIS(-1.0,1.0,0.5,1,1,2)
      CALL QYAXIS(-1.0,1.0,0.5,1,1,2)
      CALL QCMOV(50,1)
      ELSE
c *** essai pour un mode d'emulation EGA
c *** tracage d'un graphique quelconque
      CALL GOVSDN(IGBORD)
      CALL QSETUP(0,ILINE,ISYMBL,KIRSYM)
      CALL QPTABL(1,1.0,0.0,10.0,0.0,100.0,1,10,XEX,YEX)
      CALL QRTOI(0.0,0.0,I1,J1)
      CALL QRTOI(10.0,100.0,I2,J2)
      CALL QRTOI(9.0,90.0,I3,J3)
      CALL QRTOI(1.0,10.0,I4,J4)
      CALL QLINE(I2,J1,I2,J2,ILINE)
      CALL QLINE(I1,J1,I2,J2,ILINE)
      CALL QLINE(I1,J2,I2,J2,ILINE)
      CALL QLINE(I3,J2,I1,J4,ISEC)
      CALL QLINE(I2,J3,I4,J1,ISEC)
      CALL QPTXTB(3,'QT',ILETR)
      CALL QPTXTC(3,'CT',ILETR)
      CALL QCMOV(0,1)
      ENDIF

c *** recommencer la demande des choix si l'usager n'est pas satisfait
c *** l'écriture du message se fait a un endroit different selon la
c *** resolution choisie
225  IF (IREP.EQ.1.OR.IREP.EQ.2.OR.IREP.EQ.4) THEN
      WRITE(*,*) '          ETES-VOUS SATISFAIT ? (O/N) '
      ELSE
      WRITE (*,*) '          ETES-VOUS SATISFAIT ? (O/N) '
      ENDIF
      READ(*,10)TOUCHE
      CALL QSMODE(ITMODE)
      CALL QBORD(ITCOLOR)
      CALL QCLEAR(ITBACK,ITFORE)
      IF (TOUCHE.NE.'N'.AND.TOUCHE.NE.'n'.AND.TOUCHE.NE.'o'.AND.
$ TOUCHE.NE.'O') THEN
          CALL QBEEP
```

```
      L GOTO 225  
      ENDIF  
      IF ((TOUCHE.EQ. 'D').OR. (TOUCHE.EQ. 'o')) GOTO 230  
      IF (IREP.EQ.1) GOTO 130  
      GOTO 135
```

```
c *** inscrire les choix retenus pour le mode graphique,dans le fichier  
c *** de sortie (setup.pc)  
230  WRITE(1, '(9I2,5I3)')IGMODE,IGBACK,IPAL,IGBORD,  
    #   IGCOLOR,ILINE,ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2  
      CLOSE(1)  
      END
```

REAL FUNCTION YFUN(T)

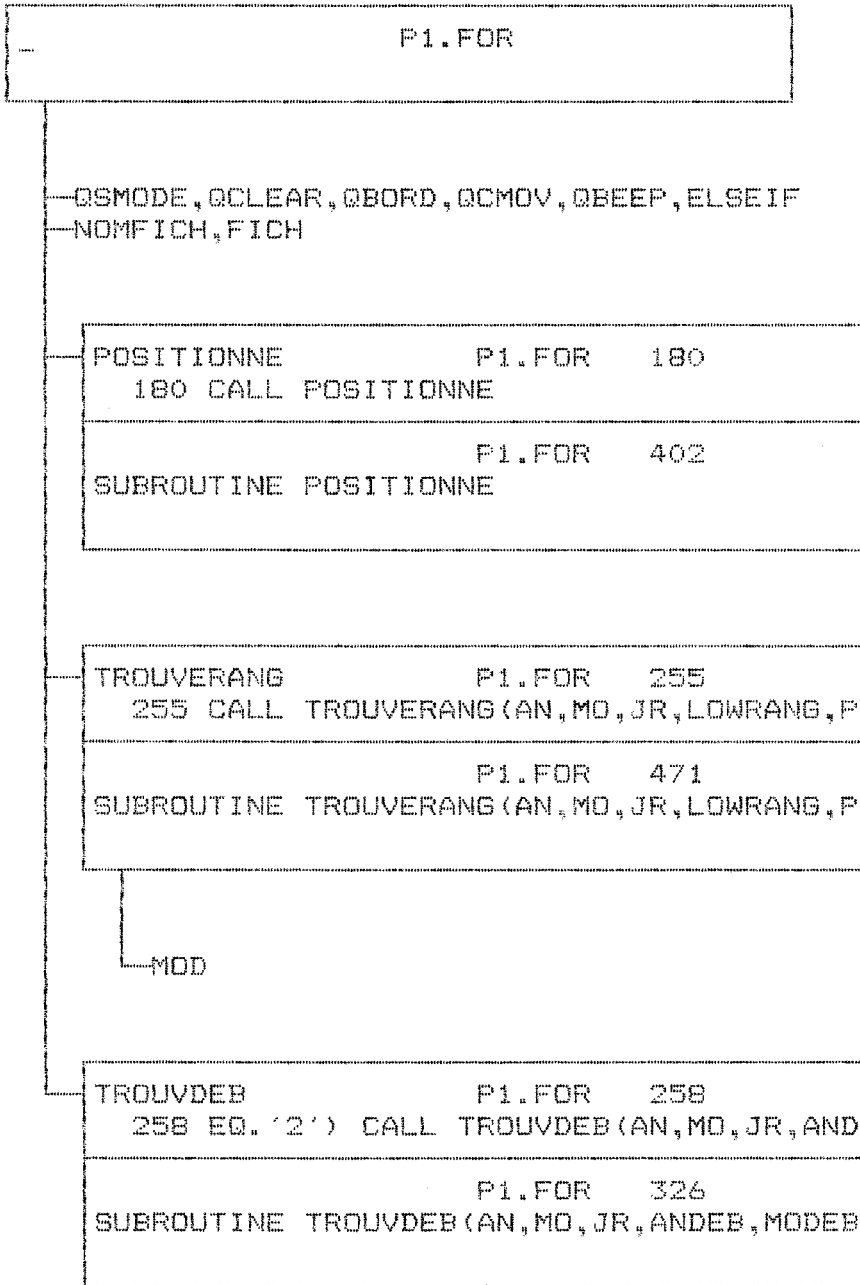
```
c -----  
c  
c Fonction :  
c  
c . Modifier la valeur d'entree T selon la fonction  
c           EXP(-xx)*COS(PI*2.5*T)  
c           ou xx est la valeur absolue de T  
c  
c Parametre :  
c  
c . T      Valeur a modifier (E/S)  
c -----
```

```
PI=4.0*ATAN(1.0)  
XX=ABS(T)  
YFUN= EXP(-XX)*COS(PI*2.5*T)  
RETURN  
END
```

SUBROUTINE AFFCOULEURS

```
c * * * * *  
c -----  
c  
c Fonctions:  
c  
c . Affiche l'ensemble des couleurs disponibles, accompagnees  
c   de leur numero respectif  
c . Demander le choix de l'utilisateur  
c -----  
5  WRITE(*, '(/,/,/,/,/,/,/,/,/,/,/)')  
    FORMAT(15X,A50,14X)  
    WRITE(*,5) 'VOICI LA LISTE DES COULEURS DISPONIBLES :'  
    WRITE(*,5) '-----'  
    WRITE(*,5) ' '
```

```
WRITE(*,5)'0) NOIR           8) NOIR PALE
WRITE(*,5)'1) BLEU           9) BLEU PALE
WRITE(*,5)'2) VERT          10) VERT PALE
WRITE(*,5)'3) AQUA          11) AQUA PALE
WRITE(*,5)'4) ROUGE         12) ROUGE PALE
WRITE(*,5)'5) MAGENTA       13) MAGENTA PALE
WRITE(*,5)'6) JAUNE-BRUN    14) JAUNE
WRITE(*,5)'7) BLANC         15) BLANC HAUTE INTENSITE
WRITE(*,5)'
WRITE(*,5)'ENTREZ LE NUMERO CORRESPONDANT A VOTRE CHOIX :
WRITE(*,5)'
WRITE(*,*)'
RETURN
END
```



#STORA GE:2

PROGRAM PARTIE1

```
c -----
c *****
c Titre: Preparation de la serie
c
c Fonctions :
c
c . Lecture du fichier d'entree
c . Localisation des debits s'ils existent
c . Elimination des dates ou le parametre etudie n'a pas
c   ete analyse
c . Ajout du nombre de jours depuis le premier prelevement de
c   la serie (en plus de la date)
c . Creation du second parametre de qualite (concentration ou
c   charge) a partir de la serie de debits si elle existe
c . Constitution d'un fichier de travail .TMP pour utilisation
c   ulterieure
c
c *****
c
c   PARAMETER (VALVIDE=-9)
c   INTEGER AN,TYP,ANDEB,DECDEB,ANBASE,DECBASE,SKIP,
c   ANFIN,ANGARDE,ANCOMP
c   INTEGER*4 RANG,PRERANG
c   CHARACTER*1 REP1,REP2,TYPD,TDEB
c   CHARACTER*12 FICHDEB,FICH,NOMFICH
c   REAL NBEG
c   LOGICAL OK,OK2,COMPILE,FIN1
c   COMMON /BLOC11/ FICHDEB
c   COMMON /BLOC12/LOW,MODEP,JODEP
c
c *** lecture des specifications graphiques
c   OPEN(5,FILE='SETUP.PC',STATUS='OLD')
c   READ(5,'(4I2)')ITMODE,ITFORE,ITBACK,ITCOLOR
c   CLOSE(5)
c   CALL QSMODE(ITMODE)
c   CALL QCLEAR(ITBACK,ITFORE)
c   CALL QBORD(ITCOLOR)
c   CALL QCMOV(0,23)
c
c *** entree des parametre au clavier
c   WRITE(*,10) ' ENTREZ LE NOM DU FICHIER DE DONNEES: '
c 10  FORMAT(/,/,A39\ )
c   NOMFICH='
c   READ(*,20) NOMFICH
c 20  FORMAT(A12)
c   OPEN(1,FILE=NOMFICH,STATUS='OLD')
c   CALL QCLEAR(ITBACK,ITFORE)
c   CALL QCMOV(0,23)
```

c *** demander les parametres necessaires a l'usager

```
30  WRITE(*,*)  
    WRITE(*,*) '    QUELLES DONNEES SONT PRESENTES DANS LE FICHER?'  
    WRITE(*,*) '  
    WRITE(*,*) '    1) LES CONCENTRATIONS'  
    WRITE(*,*) '    2) LES CHARGES'  
    WRITE(*,*) '  
    WRITE(*,40) 'VOTRE CHOIX : '  
40  FORMAT(A20\  
50  FORMAT (A1)  
    READ(*,50) TYPD  
    IF (TYPD.NE. '1'.AND.TYPD.NE. '2') THEN  
        WRITE(*,60)  
        CALL QBEEP  
        GOTO 30  
    ENDIF  
60  FORMAT(/, '    *** REPONSE INVALIDE.RECOMMENCEZ. *** '  
    CALL QCLEAR(ITBACK,ITFORE)  
    CALL QCMOV(0,23)
```

c *** les debits peuvent etre dans le meme fichier que les observations
c *** ou encore etre dans un fichier separe (debits journaliers).On peut
c *** aussi ne pas vouloir les utiliser.

```
70  WRITE(*,*) '  
    WRITE(*,*) '    CONCERNANT LES DEBITS: '  
    WRITE(*,*) '  
    WRITE(*,*) '    1) ILS APPARAISSENT DANS LE FICHER D'ENTREE '  
    WRITE(*,*) '    MENTIONNE PLUS HAUT '  
    WRITE(*,*) '  
    WRITE(*,*) '    2) ILS SERONT EXTRAITS D'UN FICHER DE DEBITS '  
    WRITE(*,*) '    JOURNALIERS '  
    WRITE(*,*) '  
    WRITE(*,*) '    3) ILS NE SERONT PAS UTILISES '  
    WRITE(*,*) '  
    WRITE(*,40) 'VOTRE CHOIX : '  
    READ(*,50) TDEB  
    IF (TDEB.NE. '1'.AND.TDEB.NE. '2'.AND.TDEB.NE. '3') THEN  
        WRITE(*,60)  
        CALL QBEEP  
        GOTO 70  
    ENDIF  
    IF (TDEB.EQ. '2') THEN  
        CALL QCLEAR(ITBACK,ITFORE)  
        CALL QCMOV(0,23)
```

c *** demander le nom du fichier de debits

```
80  WRITE(*,80) '    ENTREZ LE NOM DU FICHER DE DEBITS JOURNALIERS: '  
    FORMAT(/,/,A51\,/  
    READ(*,20) FICHDEB  
    OPEN(3,FILE=FICHDEB,STATUS='OLD')  
    ENDIF
```

c *** lire la date du 1er prelevement pour la conserver

```
85  READ(1,170,END=270)LOW,MODEP,JOSEP,VAL
```

```
c *** ignorer les donnees negatives
      IF (VAL.LT.0) GOTO 85
      REWIND(1)
```

```
c *** si les debits ne sont pas utilises,le type de donnees traitees
c *** sera celui present en entree (conc/charge)
```

```
      IF (TDEB.EQ.'3') THEN
          IF (TYPD.EQ.'1') THEN
              ITYP=0
          ELSE
              ITYP=1
          ENDIF
      ELSE
```

```
c *** sinon,on offre le choix a l'usager
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QCMDV(0,23)
```

90

```
      WRITE(*,*) '
      WRITE(*,*) '    QUELLES DONNEES DESIREZ-VOUS TRAITER '
      WRITE(*,*) '    AU COURS DE LA SESSION ? '
      WRITE(*,*) '
      WRITE(*,*) '    1) LES CONCENTRATIONS '
      WRITE(*,*) '    2) LES CHARGES '
      WRITE(*,*) '
      WRITE(*,40) 'VOTRE CHOIX : '
      READ(*,50)REP1
```

```
      IF (REP1.NE.'1'.AND.REP1.NE.'2') THEN
          WRITE(*,60)
          CALL QBEEP
          GOTO 90
      ENDIF
```

```
c *** la valeur de ityp sera conservee et refletera le choix de l'usager
```

```
      IF (REP1.EQ.'1') THEN
          ITYP=0
      ELSE
          ITYP=1
      ENDIF
  ENDIF
```

```
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QCMDV(0,23)
```

```
c *** les programmes subsequents tiendront compte du desir de l'usager
```

```
c *** a faire ou non imprimer ses graphiques, grace a la primitive
```

```
c *** graphique QPRSCN
```

```
100  WRITE(*,*) '
      WRITE(*,*) '    DESIREZ-VOUS FAIRE IMPRIMER CHACUN DES GRAPHIQUES '
      WRITE(*, '(A33)') '    SUR VOTRE IMPRIMANTE ? (O/N) '
      READ(*,50)REP2
```

```
      IF (REP2.EQ.'O'.OR.REP2.EQ.'o') THEN
          IMPRIM=1
      ELSEIF (REP2.EQ.'n'.OR.REP2.EQ.'N') THEN
          IMPRIM=0
      ELSE
          WRITE(*,60)
          CALL QBEEP
```

```
      GOTO 100
    ENDIF

c *** si on utilise pas de fichier de debits auxiliaire, le caractere blanc
c *** apparaîtra dans "ident.tmp" vis-a-vis le nom de ce fichier
    IF (TDEB.EQ.'1') THEN
        IDEB=1
        FICHDEB=' '
    ELSEIF (TDEB.EQ.'3') THEN
        IDEB=3
        FICHDEB=' '
    ELSE
c *** on utilise un fichier de debits
        IDEB=2
        CALL POSITIONNE
    ENDIF

c *** trouver la longueur du nom de fichier donne en entree
    DO 110 K=1,9
        IF (NOMFICH(K:K).EQ.' ' .OR. NOMFICH(K:K).EQ.' ') THEN
            LG=K-1
            GOTO 120
        ENDIF
110 CONTINUE
120 FICH='
    FICH(1:LG)=NOMFICH(1:LG)

c *** donner l'extension "tmp" au fichier de sortie
    FICH(LG+1:LG+4)='.TMP'
    OPEN(2,FILE=FICH,STATUS='NEW')

    OK=.FALSE.
    OK2=.FALSE.

c *** valeurs initiales bidons
    ANDEB=-99
    MODEB=-99
    RANG=-99
    NOBS=0
    REWIND(1)

c *** lecture de la 1ere observation et de la date correspondante
125 IF (TDEB.EQ.'1') THEN
        READ(1,160) ANCOMP, MOCOMP, JRCOMP, VALCOMP, DEBCOMP
    ELSE
        READ(1,170) ANCOMP, MOCOMP, JRCOMP, VALCOMP
    ENDIF

c *** ignorer les donnees negatives
    IF (VALCOMP.LT.0) GOTO 125
    FIN1=.FALSE.

c *** memoriser la date lue precedemment
130 AN=ANCOMP
    MO=MOCOMP
    JR=JRCOMP
```



```
VAL=VALCOMP  
IF (TDEB.EQ.'1') DEB=DEBCOMP  
TOT=VAL  
NBEG=1
```

c *** faire une copie des donnees pour le traitement du dernier prelevement

```
IF (VAL.GT.0) THEN  
    ANGARDE=AN  
    MOGARDE=MO  
    JRGARDE=JR  
ENDIF
```

```
140 IF (TDEB.EQ.'1') THEN  
    READ(1,160,END=220) ANCOMP,MOCOMP,JRCOMP,VALCOMP,DEBCOMP  
ELSE  
    READ(1,170,END=220) ANCOMP,MOCOMP,JRCOMP,VALCOMP  
ENDIF
```

c *** ignorer les donnees negatives
IF (VALCOMP.LT.0) GOTO 140

c *** s'il y a plusieurs observations pour une meme date, faire la
c *** moyenne des observations

```
IF (AN.EQ.ANCOMP.AND.MO.EQ.MOCOMP.AND.JR.EQ.JRCOMP.AND.VALCOMP.GE  
$ .0) THEN  
    TOT=TOT+VALCOMP  
    NBEG=NBEG+1  
    GOTO 140  
ENDIF
```

```
150 DONN=TOT/NBEG  
160 FORMAT(12X,I2,1X,I2,1X,I2,16X,F12.6,F12.6)  
170 FORMAT(12X,I2,1X,I2,1X,I2,16X,F12.6)  
190 NOBS=NOBS+1
```

c *** trouver le nombre de jours depuis le premier prelevement
CALL TROUVERANG(AN,MO,JR,LOWRANG,PRERANG,RANG,OK2)

c *** trouver le debit associe,si on fait usage du fichier de debits
IF (TDEB.EQ.'2') CALL TROUVDEB(AN,MO,JR,ANDEB,MODEB,JRDEB,OK,DEB)

c *** Calculer la valeur manquante (conc/charge) si necessaire
c *** Ecrire le tout dans le fichier de sortie

```
IF (TDEB.NE.'3') THEN  
    IF (TYPD.EQ.'2') GOTO 200
```

c *** calcul de la charge (si absente) en kg/J., avec les concentrations
c *** en mg/l

```
CONC=DONN  
CHARGE=DEB*CONC*86.4  
GOTO 210
```

c *** calcul de la concentration (si absente) en mg/l, avec les charges
c *** en kg/J.

```
200 CHARGE=DONN  
CONC=CHARGE/(DEB*86.4)
```

```
210      WRITE (2,230) AN,MO,JR,CONC,DEB,CHARGE,RANG
      ELSE
c *** mettre l'identificateur de valeur absente si le parametre de
c *** qualite est absent
      IF (TYPD.EQ.'1') THEN
          WRITE (2,230) AN,MO,JR,DONN,VALVIDE,VALVIDE,RANG
      ELSE
          WRITE (2,230) AN,MO,JR,VALVIDE,VALVIDE,DONN,RANG
      ENDIF
      IF (FIN1.EQV..FALSE.) GOTO 130
      GOTO 240
220      FIN1=.TRUE.

c *** traitement du dernier prelevement
      DONN=TOT/NBEG
      GOTO 190
230      FORMAT (I2,1X,I2,1X,I2,1X,E16.10,1X,E16.10,1X,E16.10,13X,I5)
      GOTO 220

C *** memoriser la derniere date
240      ANFIN=AN
      MOFIN=MO
      JOFIN=JR

c *** conservation des differents parametres recueillis
c *** ecriture dans "ident.tmp" qui sera ensuite relu et complete par
c *** les autres programmes

250      OPEN (4,FILE='IDENT.TMP',STATUS='NEW')
      WRITE (4,260) NOMFICH,LB,FICHDEB,IDEB,ITYP,IMPRIM,LOW,MODEP,JOSEP,
      * ANFIN,MOFIN,JOFIN,NOBS
260      FORMAT (A12,1X,I1/,A12,1X,I1/,I1,1X,I1/,I2,1X,I2,1X,I2,/,I2,1X,I2,
      * 1X,I2,/,I4)
      GOTO 280
270      WRITE (*,*) ' *** ERREUR: FICHER D'ENTREE VIDE *** '
      STOP
280      CLOSE (1)
      CLOSE (2)
      CLOSE (3)
      CLOSE (4)

c *** ouverture d'un fichier-bidon signalant que la partie p1 a ete
c *** completee sans erreur
      OPEN (99,FILE='P1.OK',STATUS='NEW')
      CLOSE (99)
      END
```

```
      SUBROUTINE TROUVDEB (AN,MO,JR,ANDEB,MODEB,DECDEB,OK,DEB)
c * * * * *
c * * * * *
```

C
C

c Fonction:

c
c . Trouve le debit journalier pour la date passee en argument
c . Le fichier de debits doit etre un fichier standard issu de
c la banque IPAR

c Parametres :

c . an,mo,jr Annee,mois et jour representant la date
c pour laquelle on desire avoir le debit (E)
c . andeb,modeb,decdeb Annee,mois et decade representant la position
c courante de lecture dans le fichier de debits
c Ces valeurs sont donc alterees a la sortie de
c la sous-routine (E/S)
c . ok Boolean dont la valeur est "true" uniquement
c si il s'agit de notre premiere recherche (E/S)
c . deb Valeur du debit recherche (S)

```
INTEGER*2 AN,ANDEB,DECDEB
LOGICAL OK
CHARACTER*12 FICHDEB
COMMON /BLOC11/ FICHDEB

IDECADE=((JR-1)/10)+1
IF (JR.EQ.31) IDECADE=3
IF ((AN.EQ.ANDEB).AND.(MO.EQ.MODEB).AND.(IDECADE.EQ.DECDEB)) THEN
c *** on continue a lire sur la meme ligne
      IPREPOS=IPDS
      IPOS=JR-((IDECADE-1)*10)
      IDIFF=IPOS-IPREPOS
      READ(3,'(1X)\')
      GOTO 40
ELSE
c *** on passe a la ligne suivante a moins qu'il ne s'agisse de
c *** la toute premiere ligne du fichier
      IF (OK.EQV..TRUE.) THEN
            READ(3,'(1X)\')
      ELSE
            OK=.TRUE.
      ENDIF
ENDIF
10 READ(3,20,END=60)ANDEB,MODEB,DECDEB
20 FORMAT(9X,I2,I2,I1\ )
   IF ((ANDEB.EQ.AN).AND.(MODEB.EQ.MO).AND.(DECDEB.EQ.IDECADE))
   $   GOTO 30
c *** lecture pour changer de ligne
   READ(3,'(1X)\')
   GOTO 10
30 IDIFF=JR-(IDECADE-1)*10
   IPOS=IDIFF

c *** avancer horizontalement dans le fichier
40 DO 50 I=1,IDIFF-1
```

```
50  READ(3, '(6X)\')
```

└── CONTINUE
 READ(3, '(F5.0)\')DEB
 GOTO 80

c *** erreur s'il n'y a pas de debit pour la date recherchee;
c *** Arrêt du programme.

```
60  WRITE(*,70)' AUCUNE VALEUR DE DEBIT POUR LE ',JR,'/',MO,'/',AN,  
$    ' DANS LE FICHIER ',FICHDEB  
70  FORMAT(A29,I2,A1,I2,A1,I2,A17,A50)  
    STOP  
80  RETURN  
    END
```

SUBROUTINE POSITIONNE

C * * * * *

c -----
c Fonctions:
c
c . Positionne le fichier de debits au debut de la premiere annee
c correspondant au 1er prelevement
c -----

```
COMMON /BLOC12/LOW,MODEP,JOEP  
INTEGER SKIP,ANBASE,DECBASE
```

c *** lire la date ou commence le fichier de debits journaliers
 READ(3,20,END=90)ANBASE,MOBASE,DECBASE
20 FORMAT(9X,I2,I2,I1\)

c *** cas ou le fichier commence au 1er janvier d'une annee
 IF ((MOBASE.EQ.1).AND.(DECBASE.EQ.1)) GOTO 40

c *** cas special ou le fichier commence a la fin decembre
 IF ((MOBASE.EQ.12).AND.(DECBASE.EQ.3)) THEN
 SKIP=LOW-1-ANBASE
 READ(3, '(1X)')
 GOTO 60
 ENDIF
 READ(3, '(1X)',END=90)

c *** continuer a lire jus qu'au positionnement au debut de la premiere
c *** annee rencontree debutant au 1er janvier

```
30  READ(3, '(11X,I2,I1\)',END=90)MOBASE,DECBASE  
    IF (MOBASE.EQ.12.AND.DECBASE.EQ.3) THEN  
        SKIP=LOW-1-ANBASE  
        READ(3, '(1X)',END=90)  
        GOTO 60  
    ELSE  
        READ(3, '(1X)',END=90)
```


LOGICAL OK2

DATA JPARM /31,28,31,30,31,30,31,31,30,31,30,31/

PRERANG=RANG

c *** calcul du rang absolu de la date

RANG=(AN*365)+(AN/4)+JR

DO 10 I=1,(MO-1)

RANG=RANG+JPARM(I)

10 CONTINUE

c *** ajustement si l'annee traitee est bissextile

IF ((MOD(AN,4).EQ.0).AND.(MO.LE.2)) RANG=RANG-1

IF (OK2.EQV..FALSE.) THEN

LOWRANG=RANG-1

OK2=.TRUE.

ENDIF

c *** transformer les rangs en rangs relatifs

RANG=RANG-LOWRANG

IF (OK2.EQV..TRUE.) THEN

IF (RANG.LT.PRERANG) GOTO 20

ELSE

c *** signaler que le premier rang a ete traite

OK2=.TRUE.

ENDIF

GOTO 30

c *** erreur si les dates ne sont pas dans l'ordre chronologique

c *** et sortie du programme

20 WRITE(*,*)' *** LES DATES NE SONT PAS DANS L'ORDRE CHRONOLOGIQUE

* ***'

WRITE(*,*)'

*** FIN DU PROGRAMME ***'

WRITE(*,*)'

STOP

30 RETURN

END



P2.FOR

FICH, NOMFICH, QSMODE, QCLEAR, QBORD, ELSEIF
QBEEP, TEXTMM

GRAPHI P2.FOR 56 x2
56 CALL GRAPHI(C, TD, RSAFE, XSAFE)
58 CALL GRAPHI(L, TD, RSAFE, XSAFE)

P2.FOR 501
SUBROUTINE GRAPHI(X, RANG, RSAFE, XSAFE)

TEXTCT, MIN, MAX, QSMODE, QCOLOR, QOVSCN
QPLOT, QXAXIS, QYAXIS, QGTXT, QPTXTC, QSETUP
QTABL, QCMOV, QBEEP

TICS P2.FOR 560 x2
560 CALL TICS(XMIN, XMAX, YSTS, YFINS, TIC
561 CALL TICS(0.0, RANG(NOBS), XSTS, XFIN

TICS.FOR 1
SUBROUTINE TICS(XST, XFIN, XSTS, XFINS, TIC,

ELSEIF, AINT, ABS

AJUSTE P2.FOR 563 x2
563 CALL AJUSTE(X, NOBS, AINTEY, TEXTEY, Y
566 CALL AJUSTE(RANG, NOBS, AINTEX, TEXTE

AJUSTE.FOR 1
SUBROUTINE AJUSTE(X, NOBS, AINTE, TEXTE, XST

ALOG10, EXP, CHAR, ABS, ELSEIF, MOD, TEXTE


```
COFFICH          P2.FOR    60
  60 BSGARD) CALL COFFICH(FICH,RSafe)
```

```
          P2.FOR    670
SUBROUTINE COFFICH(FICH,RSafe)
```

```
DOUBMASS        P2.FOR    126 x8
  126 CALL DOUBMASS(1,C,TD,TEXTC,TEXTT,T
  138 CALL DOUBMASS(1,L,TD,TEXTL,TEXTT,T
  150 CALL DOUBMASS(0,C,Q,TEXTC,TEXTQ,TE
```

```
          P2.FOR    248
SUBROUTINE DOUBMASS(IT,C,Q,TEXTC,TEXTQ,T
```

```
--QSMODE, QCOLOR, QOVSCN, QPLOT, QXAXIS
--QYAXIS, QBTXT, QPTXTC, QSETUP, QTABL, QPTXTA
--QRTOI, QLINE, QCMOV, QCHAR, QBEEP
```

```
TICS            P2.FOR    303 x2  p1
  303 CALL TICS(0.0, SQ(NOBS), XSTS, XFINS,
  304 CALL TICS(0.0, SC(NOBS), YSTS, YFINS,
```

```
AJUSTE          P2.FOR    306 x2  p1
  306 CALL AJUSTE(SQ, NOBS, AINTEX, TEXTEX,
  309 CALL AJUSTE(SC, NOBS, AINTEY, TEXTEY,
```

```
CUSUM           P2.FOR    174 x6
  174 CALL CUSUM(Q, TK, TEXTQ, TEXTT, TEXTMM
  186 CALL CUSUM(C, TK, TEXTC, TEXTT, TEXTMM
  198 CALL CUSUM(L, TK, TEXTL, TEXTT, TEXTMM
```

```
          P2.FOR    388
SUBROUTINE CUSUM(C, TC, TEXTC, TEXTT, TEXTMM
```

```
--QSMODE, QCOLOR, QOVSCN, QPLOT, QXAXIS
```

QYAXIS, QSETUP, QGTXT, QPTXTC, QPTXTA, QTABL
QRTOI, QLINE, QCMOV, QCHAR, QBEEP

TICS P2.FOR 451 x2 p1
451 CALL TICS(MINCU, MAXCU, YSTS, YFINS, T
452 CALL TICS(O.O, TCU(NOB), XSTS, XFINS,

AJUSTE P2.FOR 454 x2 p1
454 CALL AJUSTE(TCU, NOB, AINTEX, TEXTEX,
456 CALL AJUSTE(VCUSUM, NOB, AINTEY, TEXT

```
#STORA      GE:2
            PROGRAM PARTIE2
C           -----
C           INTEGER MAXI
C
C           MAXI EST LE NOMBRE MAXIMAL D'OBSERVATIONS POSSIBLES
C
C           PARAMETER (MAXI=1300)
C
C           REAL C(MAXI),Q(MAXI),L(MAXI),TD(MAXI),TK(MAXI),
#           TAMP1(MAXI),TAMP2(MAXI),RSAF(MAXI),XSAFE(MAXI)
C           INTEGER ANFIN,ANDEF
C           CHARACTER*60 TEXTMM
C           CHARACTER*21 TEXTDD,TEXTCU
C           CHARACTER*12 FICH,FICHDEB,NOMFICH
C           CHARACTER*15 TEXTC,TEXTL
C           CHARACTER*15 TEXTT,TEXTQ
C           CHARACTER*1 TOUCHE,CHOIX
C           COMMON /TEXTE /ITMODE,ITFORE,ITBACK,ITCOLOR
C           COMMON /GRAPH /IGMODE,IGBACK,IPAL,IGCOLOR,ILINE,ISEC,IGBORD
#           ,KIRSYM,ISYMBL,ILETR,JROW1,JROW2,JCOL1,JCOL2
C           COMMON /IDENT /NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEF,MODEP
#           ,JODEP,ANFIN,MOFIN,JOFIN,NOBS
C
C           LECTURE DU FICHIER SETUP.PC POUR CONFIGURATION
C
C           OPEN(2,FILE='SETUP.PC',STATUS='OLD')
C           READ(2,'(4I2/9I2,5I3)') ITMODE,ITFORE,ITBACK,ITCOLOR,IGMODE,
#           IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,ILETR,ISEC,KIRSYM,ISYMBL,
#           JROW1,JROW2,JCOL1,JCOL2
C           CLOSE(2)
C
C           LECTURE DU FICHIER IDENT.TMP
C
C           OPEN(3,FILE='IDENT.TMP',STATUS='OLD')
C           READ(3,2)NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEF,MODEP,JODEP,
#           ANFIN,MOFIN,JOFIN,NOBS
C           2  FORMAT(A12,1X,I1/,A12,1X,I1/,I1,1X,I1/,I2,1X,I2,1X,I2,/,I2,1X,I2,
#           1X,I2,/,I4)
C
C           FICH='
C           FICH(1:LG)=NOMFICH(1:LG)
C           FICH(LG+1:LG+4)='.TMP'
C           OPEN(1,FILE=FICH,STATUS='OLD')
C           TK(1)=0
C           I=0
C           15  I=I+1
C           READ(1,20,END=30)C(I),Q(I),L(I),TD(I)
C           RSAF(I)=TD(I)
C           TK(I+1)=TD(I)
C           20  FORMAT(9X,E16.10,1X,E16.10,1X,E16.10,13X,F5.0)
C           GOTO 15
C           30  NOBSGARD=NOBS
C           IF (ITYP.EQ.0) THEN
```

```

      CALL GRAPHI (C,TD,RSafe,XSafe)
ELSE
      CALL GRAPHI (L,TD,RSafe,XSafe)
ENDIF
IF (NOBS.NE.NOBSGARD) CALL COFFICH(FICH,RSafe)
TEXTC=' CONCEN mg/l '
TEXTQ=' DEBIT m3/sec '
TEXTL=' CHARGE Kg/jour '
TEXTT='TEMPS (# JOURS)'
TEXTDO='FONCTION DOUBLE-MASSE'
TEXTCU=' FONCTION CUSUM'

C
C CONFIGURATION EN MODE TEXTE
C
10 CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)

40 WRITE(*,*) ' ETUDES GRAPHIQUES'
WRITE(*, '(/)')
WRITE(*,*) ' [1] DOUBLE-MASSE : CONCENTRATION vs TEMPS'
WRITE(*,*) ' [2] DOUBLE-MASSE : CHARGE vs TEMPS'
WRITE(*,*) ' [3] DOUBLE-MASSE : CONCENTRATION vs DEBIT'
WRITE(*,*) ' [4] DOUBLE-MASSE : CHARGE VS DEBIT'
WRITE(*,*) ' [5] CUSUM : DEBIT vs TEMPS'
WRITE(*,*) ' [6] CUSUM : CONCENTRATION vs TEMPS'
WRITE(*,*) ' [7] CUSUM : CHARGE VS TEMPS'
WRITE(*,*) ' [8] TOUS LES GRAPHIQUES DISPONIBLES PARMi
* [1] A [7]'
WRITE(*,*) ' [9] SORTIE'
WRITE(*, '(/)')
WRITE(*, '(A34,\)') ' ENTREZ VOTRE CHOIX : (1 A 9) '
READ(*, '(A1)')CHOIX
IF (CHOIX.EQ.'1') THEN
      GOTO 50
ELSEIF (CHOIX.EQ.'2') THEN
      GOTO 55
ELSEIF (CHOIX.EQ.'3') THEN
      GOTO 60
ELSEIF (CHOIX.EQ.'4') THEN
      GOTO 65
ELSEIF (CHOIX.EQ.'5') THEN
      GOTO 70
ELSEIF (CHOIX.EQ.'6') THEN
      GOTO 75
ELSEIF (CHOIX.EQ.'7') THEN
      GOTO 80
ELSEIF (CHOIX.EQ.'8') THEN
      GOTO 85
ELSEIF (CHOIX.EQ.'9') THEN
      GOTO 99
ELSE
      CALL QBEEP
      CALL QBEEP
      GOTO 40

```

```

      ENDIF
42  FORMAT (/,'    *** LES CONCENTRATIONS NE SONT PAS DISPONIBLES ***')
44  FORMAT (/,'    *** LES CHARGES NE SONT PAS DISPONIBLES ***')
46  FORMAT (/,'    *** LES DEBITS NE SONT PAS DISPONIBLES ***')
48  FORMAT ('    *** CHOISISSEZ UNE AUTRE OPTION ***',/)
50  IF (IDEB.EQ.3.AND.ITYP.NE.0) THEN
      CALL QBEEP
      WRITE(*,42)
      WRITE(*,48)
      GOTO 40
      ENDIF
      TEXTMM(1:12)='FIGURE 2.2 : '
      TEXTMM(13:34)=TEXTDD
      TEXTMM(35:44)=' :C vs T, '
      TEXTMM(45:56)=NOMFICH
      CALL DOUBMASS(1,C,TD,TEXTC,TEXTT,TEXTMM,TAMP1,TAMP2)
      GOTO 10
55  IF (IDEB.EQ.3.AND.ITYP.NE.1) THEN
      CALL QBEEP
      WRITE(*,44)
      WRITE(*,48)
      GOTO 40
      ENDIF
      TEXTMM(1:12)='FIGURE 2.3 : '
      TEXTMM(13:34)=TEXTDD
      TEXTMM(35:44)=' :L vs T, '
      TEXTMM(45:56)=NOMFICH
      CALL DOUBMASS(1,L,TD,TEXTL,TEXTT,TEXTMM,TAMP1,TAMP2)
      GOTO 10
60  IF (IDEB.EQ.3) THEN
      CALL QBEEP
      WRITE(*,46)
      WRITE(*,48)
      GOTO 40
      ENDIF
      TEXTMM(1:12)='FIGURE 2.4 : '
      TEXTMM(13:34)=TEXTDD
      TEXTMM(35:44)=' :C vs Q, '
      TEXTMM(45:56)=NOMFICH
      CALL DOUBMASS(0,C,Q,TEXTC,TEXTQ,TEXTMM,TAMP1,TAMP2)
      GOTO 10
65  IF (IDEB.EQ.3) THEN
      CALL QBEEP
      WRITE(*,46)
      WRITE(*,48)
      GOTO 40
      ENDIF
      TEXTMM(1:12)='FIGURE 2.5 : '
      TEXTMM(13:34)=TEXTDD
      TEXTMM(35:44)=' :L vs Q, '
      TEXTMM(45:56)=NOMFICH
      CALL DOUBMASS(0,L,Q,TEXTL,TEXTQ,TEXTMM,TAMP1,TAMP2)
      GOTO 10
70  IF (IDEB.EQ.3) THEN
      CALL QBEEP

```

```
        WRITE (*,46)
        WRITE (*,48)
        GOTO 40
    ENDIF
    TEXTMM(1:12)='FIGURE 2.6 : '
    TEXTMM(13:34)=TEXTCU
    TEXTMM(35:44)=' :Q vs T, '
    TEXTMM(45:56)=NOMFICH
    CALL CUSUM(Q,TK,TEXTQ,TEXTT,TEXTMM,TAMP1,TAMP2)
    GOTO 10
75  IF (IDEB.EQ.3.AND.ITYP.NE.0) THEN
        CALL QBEEP
        WRITE (*,42)
        WRITE (*,48)
        GOTO 40
    ENDIF
    TEXTMM(1:12)='FIGURE 2.7 : '
    TEXTMM(13:34)=TEXTCU
    TEXTMM(35:44)=' :C vs T, '
    TEXTMM(45:56)=NOMFICH
    CALL CUSUM(C,TK,TEXTC,TEXTT,TEXTMM,TAMP1,TAMP2)
    GOTO 10
80  IF (IDEB.EQ.3.AND.ITYP.NE.1) THEN
        CALL QBEEP
        WRITE (*,44)
        WRITE (*,48)
        GOTO 40
    ENDIF
    TEXTMM(1:12)='FIGURE 2.8 : '
    TEXTMM(13:34)=TEXTCU
    TEXTMM(35:44)=' :L vs T, '
    TEXTMM(45:56)=NOMFICH
    CALL CUSUM(L,TK,TEXTL,TEXTT,TEXTMM,TAMP1,TAMP2)
    GOTO 10
85  IF (ITYP.EQ.0.OR.IDEB.NE.3) THEN
        TEXTMM(13:34)=TEXTDO
        TEXTMM(45:56)=NOMFICH
        TEXTMM(1:12)='FIGURE 2.2 : '
        TEXTMM(35:44)=' :C vs T, '
        CALL DOUBMASS(1,C,TD,TEXTC,TEXTT,TEXTMM,TAMP1,TAMP2)
    ENDIF
    IF (ITYP.EQ.1.OR.IDEB.NE.3) THEN
        TEXTMM(13:34)=TEXTDO
        TEXTMM(1:12)='FIGURE 2.3 : '
        TEXTMM(35:44)=' :L vs T, '
        CALL DOUBMASS(1,L,TD,TEXTL,TEXTT,TEXTMM,TAMP1,TAMP2)
    ENDIF
    IF (IDEB.NE.3) THEN
        TEXTMM(1:12)='FIGURE 2.4 : '
        TEXTMM(13:34)=TEXTDO
        TEXTMM(35:44)=' :C vs Q, '
        CALL DOUBMASS(0,C,Q,TEXTC,TEXTQ,TEXTMM,TAMP1,TAMP2)
        TEXTMM(1:12)='FIGURE 2.5 : '
        TEXTMM(13:34)=TEXTDO
        TEXTMM(33:44)=' :L vs Q, '
```

```
CALL DOUBMASS(O,L,Q,TEXTL,TEXTQ,TEXTMM,TAMP1,TAMP2)
TEXTMM(1:12)='FIGURE 2.6 : '
TEXTMM(13:34)=TEXTCU
TEXTMM(35:44)=' :Q vs T, '
CALL CUSUM(Q,TK,TEXTQ,TEXTT,TEXTMM,TAMP1,TAMP2)
ENDIF
IF (ITYP.EQ.0.OR.IDEB.NE.3) THEN
TEXTMM(1:12)='FIGURE 2.7 : '
TEXTMM(13:34)=TEXTCU
TEXTMM(35:44)=' :C vs T, '
CALL CUSUM(C,TK,TEXTC,TEXTT,TEXTMM,TAMP1,TAMP2)
ENDIF
IF (ITYP.EQ.1.OR.IDEB.NE.3) THEN
TEXTMM(1:12)='FIGURE 2.8 : '
TEXTMM(13:34)=TEXTCU
TEXTMM(35:44)=' :L vs T, '
CALL CUSUM(L,TK,TEXTL,TEXTT,TEXTMM,TAMP1,TAMP2)
ENDIF
CALL @SMODE(ITMODE)
CALL @CLEAR(0,7)
CALL @BORD(0)
CLOSE(1)
CLOSE(3)
99 OPEN(99,FILE='P2.OK',STATUS='NEW')
CLOSE(99)
END
```

SUBROUTINE DOUBMASS(IT,C,Q,TEXTC,TEXTQ,TEXTMM,SC,SQ)

Fonction:

. Tracage de la courbe double-masse pour les series originales

Parametres:

- . it Indique si on a le temps ou le debit en abscisse
- . c Vecteur contenant les valeurs e ordonnee
- . q Vecteur contenant les valeurs en abscisse
- . textc Texte a etre inscrit sur l'ordonnee
- . textq Texte a etre inscrit sur l'abscisse
- . textmm Titre du graphique
- . sc # Vecteur des sommes cumulees de "c"
- . sq # Vecteur des sommes cumulees de "q"

```
INTEGER ANDEF,ANFIN
REAL C(*),Q(*),SC(*),SQ(*)
CHARACTER*1 TOUCHE
CHARACTER*12 FICHDEB,NOMFICH
CHARACTER*10 TEXTEX,TEXTEY
COMMON /TEXTE /ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH /IGMODE,IGBACK,IFAL,IGCOLOR,ILINE,ISEC,IGBORD
*,KIRSYM,ISYMBL,ILETR,JROW1,JROW2,JCOL1,JCOL2
```

```
COMMON /IDENT /NOMFICH,LG,FICHDEB, IDEB, ITYP, IMPRIM, ANDEP, MDEP  
$, JODEP, ANFIN, MDFIN, JOFIN, NOBS
```

```
TEXTEX=' '  
TEXTY=' '  
SC(1)=C(1)  
SQ(1)=Q(1)
```

```
DO 10 I=1,NOBS-1  
    SC(I+1)=SC(I)+C(I+1)  
    IF (IT.EQ.1) THEN  
        SQ(I+1)=Q(I+1)  
    ELSE  
        SQ(I+1)=SQ(I)+Q(I+1)  
    ENDIF  
CONTINUE
```

C
C
C

CONFIGURATION EN MODE GRAPHIQUE

```
CALL QSMODE(IGMODE)  
IF (IGMODE.LT.10) THEN  
    CALL QCOLOR(IGBACK,IPAL)  
ELSE  
    CALL QOVSCN(IGBORD)  
ENDIF
```

C
C
C

CONSTRUCTION DU GRAPHE DE LA FONCTION DOUBLE-MASSES

```
CALL TICS(0.0,SQ(NOBS),XSTS,XFINS,TICX,MINORX,AINTEX)  
CALL TICS(0.0,SC(NOBS),YSTS,YFINS,TICY,MINORY,AINTEY)  
IF (AINTEX.GE.10.OR.AINTEY.LT.1) THEN  
    CALL AJUSTE(SQ,NOBS,AINTEX,TEXTEX,XSTS,XFINS,TICX)  
ENDIF  
IF (AINTEY.GE.10.OR.AINTEY.LT.1) THEN  
    CALL AJUSTE(SC,NOBS,AINTEY,TEXTY,YSTS,YFINS,TICY)  
ENDIF  
CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,XSTS,XFINS,YSTS,YFINS,  
$, XSTS,YSTS,0,1.0,1.5)  
CALL QXAXIS(XSTS,XFINS,TICX,MINORX,1,0)  
CALL QYAXIS(YSTS,YFINS,TICY,MINORY,1,0)  
CALL QGTXT(10,TEXTEX,ILETR,JCOL2-35,JROW1+10,0)  
CALL QPTXTC(10,TEXTY,ILETR)  
CALL QSETUP(0,ILINE,ISYMBL,KIRSYM)  
CALL QTABL(1,NOBS,SQ,SC)  
CALL QGTXT(56,TEXTMM,ILETR,JCOL1+60,JROW2+12,0)  
CALL QPTXTA(15,TEXTQ,ILETR)  
IF (IGMODE.EQ.16) THEN  
    INCR=85  
ELSE  
    INCR=55  
ENDIF  
CALL QGTXT(15,TEXTC,ILETR,0,(JROW2+JROW1)/2+INCR,-1)
```

C
C
C

TRACAGE DE LA DROITE

```
CALL QRTOI(0.0,0.0,I1,J1)
```



```
CALL QRTOI(SQ(NOBS),SC(NOBS),I2,J2)  
CALL QLINE(I1,J1,I2,J2,ILINE)
```

C
C
C

DETERMINATION DE LA PENTE

```
P=SC(NOBS)/SQ(NOBS)
```

C
C
C

CALCUL DES ECART-TYPES RELIES AUX RAILS

```
R1=0  
R2=0  
NA=0  
NB=0
```

```
DO 20 I=1,NOBS  
    PP=P*SQ(I)  
    IF (SC(I).GT.PP) GOTO 15  
    R2=R2+(SC(I)-PP)**2  
    NB=NB+1  
    GOTO 20  
15  R1=R1+(SC(I)-PP)**2  
    NA=NA+1
```

20 CONTINUE

```
IF (NA.LE.1) THEN  
    RA1=0.0  
ELSE  
    RA1=(R1/(NA-1))**0.5  
ENDIF  
IF (NB.LE.1) THEN  
    RA2=0.0  
ELSE  
    RA2=(R2/(NB-1))**0.5  
ENDIF
```

C
C
C

CONSTRUCTION DES RAILS

```
CTE1=1.96*RA1  
CTE2=1.96*RA2/P  
CTE3=SQ(NOBS)-(1.96*RA1/P)  
CTE4=SC(NOBS)-(1.96*RA2)  
CALL QRTOI(0.0,CTE1,I1,J1)  
CALL QRTOI(CTE2,0.0,I2,J2)  
CALL QRTOI(CTE3,SC(NOBS),I3,J3)  
CALL QRTOI(SQ(NOBS),CTE4,I4,J4)  
CALL QLINE(I1,J1,I3,J3,ISEC)  
CALL QLINE(I2,J2,I4,J4,ISEC)  
CALL QLINE(I4,J2,I4,J3,ILINE)  
CALL QLINE(I1,J3,I4,J3,ILINE)  
CALL QCMOV(1,1)  
IF (IMPRIM.EQ.1) CALL QPSCRN  
CALL QCHAR(27,ILETR,1)  
CALL QBEEP  
29 READ(*,30) TOUCHE  
30 FORMAT(A1)  
RETURN
```

END

SUBROUTINE CUSUM(C,TC,TEXTC,TEXTT,TEXTMM,TCU,VCUSUM)

Fonction:

. Tracage de la courbe CUSUM pour les series originales

Parametres:

. c Vecteur contenant les valeurs utilisees en ordonnee
. tc Vecteur contenant les valeurs utilisees en abscisse
. textc Texte a etre inscrit sur l'ordonnee
. textt Texte a etre inscrit sur l'abscisse
. textmm Titre du graphique
. tcu # Vecteur des valeurs de l'abscisse
. vcusum # Vecteur des valeurs de l'ordonnee

INTEGER ANFIN,ANDEP
CHARACTER*12 FICHDEB,NOMFICH
CHARACTER*10 TEXTEX,TEXTEY
REAL C(*),TC(*),TCU(*),VCUSUM(*),MAXCU,MINCU
COMMON /TEXTE /ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH /IGMODE,ISBACK,IPAL,IGCOLOR,ILINE,ISEC,IGBORD
\$,KIRSYM,ISYMBL,IETR,JROW1,JROW2,JCOL1,JCOL2
COMMON /IDENT /NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEP,MODEP
\$,JODEP,ANFIN,MOFIN,JOFIN,NOBS

TEXTEX=' '
TEXTEY=' '

CALCUL DE LA MOYENNE DE LA SERIE

CM=0.0
DO 10 I=1,NOBS
CM=CM+C(I)
CONTINUE
CM=CM/NOBS

CALCUL DE LA FONCTION CUSUM

NOB=NOBS+1
TCU(1)=0
TCU(2)=TC(1)
VCUSUM(1)=0
VCUSUM(2)=C(1)-CM
MAXCU=VCUSUM(2)
MINCU=VCUSUM(2)
DO 15 I=3,NOB
TCU(I)=TC(I-1)
VCUSUM(I)=VCUSUM(I-1)+C(I-1)-CM
IF (VCUSUM(I).GT.MAXCU) MAXCU=VCUSUM(I)
IF (VCUSUM(I).LT.MINCU) MINCU=VCUSUM(I)

15 CONTINUE

C
C
C

CONFIGURATION EN MODE GRAPHIQUE

CALL QSMODE (IGMODE)

IF (IGMODE.LT.10) THEN

CALL QCOLOR (IGBACK,IPAL)

ELSE

CALL QOVSCN (IGBORD)

ENDIF

CALL TICS (MINCU,MAXCU,YSTS,YFINS,TICY,MINORY,AINTEY)

CALL TICS (0.0,TCU (NOB),XSTS,XFINS,TICX,MINORX,AINTEX)

IF (AINTEX.GE.10.OR.AINTEX.LT.1) THEN

CALL AJUSTE (TCU,NOB,AINTEX,TEXTEX,XSTS,XFINS,TICX)

ENDIF

IF (AINTEY.GE.10.OR.AINTEY.LT.1) THEN

CALL AJUSTE (VCUSUM,NOB,AINTEY,TEXTTEY,YSTS,YFINS,TICY)

ENDIF

CALL QPLOT (JCOL1,JCOL2,JROW1,JROW2,XSTS,XFINS,YSTS,YFINS,

* XSTS,YSTS,0,1.0,1.5)

CALL QXAXIS (XSTS,XFINS,TICX,MINORX,1,0)

CALL QYAXIS (YSTS,YFINS,TICY,MINORY,1,0)

CALL QSETUP (0,ILINE,ISYMBL,KIRSYM)

CALL QGTXT (10,TEXTEX,ILETR,JCOL2-35,JROW1+10,0)

CALL QPTXTC (10,TEXTTEY,ILETR)

CALL QGTXT (56,TEXTMM,ILETR,JCOL1+60,JROW2+12,0)

CALL QPTXTA (15,TEXTT,ILETR)

IF (IGMODE.EQ.16) THEN

INCR=85

ELSE

INCR=55

ENDIF

CALL QGTXT (15,TEXTC,ILETR,0,INCR+(JROW2+JROW1)/2,-1)

C
C
C

CONSTRUCTION DU GRAPHE DE LA FONCTION CUSUM

CALL QTABL (1,NOB,TCU,VCUSUM)

IF (AINTEY.GE.10.OR.AINTEY.LT.1) THEN

MINCU=MINCU/AINTEY

MAXCU=MAXCU/AINTEY

ENDIF

CALL QRTDI (0.0,MINCU,IA,JA)

CALL QRTDI (TCU (NOB),MAXCU,IB,JB)

CALL QRTDI (0.0,0.0,IA,JC)

CALL QLINE (IA,JB,IB,JB,ILINE)

CALL QLINE (IB,JA,IB,JB,ILINE)

CALL QLINE (IA,JC,IB,JC,ILINE)

CALL QCMOV (1,1)

IF (IMPRIM.EQ.1) CALL QPSCRN

CALL QCHAR (27,ILETR,1)

CALL QBEEP

29 READ (*,30) TOUCHE

30 FORMAT (A1)

RETURN

END

SUBROUTINE GRAPHI(X,RANG,RSAFE,XSAFE)

C * * * * *

c *** trace les graphiques des charges ou des concentrations
c *** en fonction du temps
c *** offre la possibilite de retirer certaines valeurs aberrantes.
c *** La valeur de NOBS est modifiee si on enleve certaines valeurs

```
REAL X(*),RANG(*),RSAFE(*),XSAFE(*)
INTEGER ANDEF,ANFIN
CHARACTER*15 TEXTY
CHARACTER*12 FICHDEB,NOMFICH
CHARACTER*10 TEXTEX,TEXTEY
CHARACTER*50 TEXTCT
CHARACTER*38 TEXTLT
CHARACTER*20 F
LOGICAL TERMINE
COMMON /TEXTE /ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH /IGMODE,IGBACK,IPAL,IGCOLOR,ILINE,ISEC,IGBORD
* ,KIRSYM,ISYMBL,ILETR,JROW1,JROW2,JCOL1,JCOL2
COMMON /IDENT /NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEF,MODEP
* ,JODEP,ANFIN,MOFIN,JOFIN,NOBS
```

```
TEXTEX='
TEXTEY='
TEXTCT='
```

IF (ITYP.EQ.0) THEN

```
TEXTLT='FIGURE 2.1 : CONCENTRATION VS TEMPS,
TEXTCT(1:38)=TEXTLT
TEXTCT(39:50)=NOMFICH
TEXTY=' CONCEN mg/l
F='(9X,F12.6,51X,15)'
```

ELSE

```
TEXTLT='FIGURE 2.1 : CHARGE VS TEMPS,
TEXTCT(1:38)=TEXTLT
TEXTCT(39:50)=NOMFICH
TEXTY=' CHARGE Kg/jour
F='(35X,F20.5,17X,15)'
```

ENDIF

TERMINE=.FALSE.

NMINABER=0

NMAXABER=0

DO 2 I=1,NOBS

```
XSAFE(I)=X(I)
RSAFE(I)=RANG(I)
```

CONTINUE

XMIN=1E30

XMAX=-99.0

DO 10 I=1,NOBS

2
5

```

      XMIN=MIN(XMIN,X(I))
      XMAX=MAX(XMAX,X(I))
10  CONTINUE
      CALL QSMODE(IGMODE)
      IF(IGMODE.LT.10) THEN
          CALL QCOLOR(IGBACK,IPAL)
      ELSE
          CALL QOVSCN(IGBORD)
      ENDIF
      CALL TICS(XMIN,XMAX,YSTS,YFINS,TICY,MINORY,AINTEY)
      CALL TICS(0.0,RANG(NOBS),XSTS,XFINS,TICX,MINORX,AINTEX)
      IF (AINTEY.GE.10.OR.AINTEY.LT.1) THEN
          CALL AJUSTE(X,NOBS,AINTEY,TEXTEY,YSTS,YFINS,TICY)
      ENDIF
      IF (AINTEX.GE.10.OR.AINTEX.LT.1) THEN
          CALL AJUSTE(RANG,NOBS,AINTEX,TEXTEX,XSTS,XFINS,TICX)
      ENDIF
      CALL QPLOT(JCOL1,JCOL2,3*JROW1/4+JROW2/4,JROW2,XSTS,XFINS,YSTS,
*      YFINS,XSTS,YSTS,0,1.0,1.5)
      CALL QXAXIS(XSTS,XFINS,TICX,MINORX,1,0)
      CALL QYAXIS(YSTS,YFINS,TICY,MINORY,1,0)
      CALL QGTXT(50,TEXTCT,ILETR,JCOL1+60,JROW2+12,0)
      CALL QGTXT(10,TEXTEX,ILETR,JCOL2-35,3*JROW1/4+JROW2/4+10,0)
      CALL QPTXTC(10,TEXTEY,ILETR)
      INCR=0
      IF(IGMODE.EQ.16) INCR=15
      CALL QGTXT(15,'TEMPS (# JOURS)',ILETR,(JCOL1+JCOL2)/2-60,
*      3*JROW1/4+JROW2/4-(20+INCR),0)
      IF(IGMODE.EQ.16) THEN
          INCR=55
      ELSE
          INCR=35
      ENDIF
      CALL QGTXT(15,TEXTY,ILETR,0,3*JROW2/4+JROW1/4+INCR,-1)
      CALL QSETUP(0,ILINE,ISYMBL,KIRSYM)
      CALL QTABL(1,NOBS,RANG,X)
      DO 45 I=1,NOBS
          X(I)=XSAFE(I)
          RANG(I)=RSAFE(I)
45  CONTINUE
50  CALL QCMDV(5,4)
      IF (XMIN.LE.999999.AND.XMAX.LE.999999) THEN
          WRITE(*,55) 'MINIMUM: ',XMIN,' MAXIMUM: ',XMAX
      ELSE
          WRITE(*,56) 'MINIMUM: ',XMIN,' MAXIMUM: ',XMAX
      ENDIF
55  FORMAT(A9,F10.3,18X,A8,F10.3)
56  FORMAT(A9,E16.10,12X,A8,E16.10)
      WRITE(*,*) '
      WRITE(*,*) ' 1) RETIRER LA PLUS PETITE VALEUR      3) RETIRER LES DE
*      UX EXTREMES'
      WRITE(*,*) ' 2) RETIRER LA PLUS GRANDE VALEUR     4) LAISSER LES DO
*      NNEES TELLES QUELLES'
      WRITE(*,'(A30\)' ) ' -> CHOISISSEZ UNE OPTION : '
      IF(IMPRIM.EQ.1) CALL QPSCRN
```

```
C      CALL QCHAR(27,ILETR,1)
C      CALL QBEEP
C      READ(*,40) TOUCHE
40     FORMAT(A1)
      READ(*,'(A1)')CHOIX
      IF (CHOIX.NE.'1'.AND.CHOIX.NE.'2'.AND.CHOIX.NE.'3'.AND.CHOIX
*     .NE.'4') THEN
          CALL QBEEP
          GOTO 50
      ENDIF
      IF (CHOIX.EQ.'4') THEN
          TERMINE=.TRUE.
          GOTO 60
      ENDIF
      K=1
      IF (CHOIX.EQ.'3') THEN
          DO 65 I=1,NOBS
              IF (X(I).NE.XMIN.AND.X(I).NE.XMAX) THEN
                  X(K)=X(I)
                  XSAFE(K)=XSAFE(I)
                  RANG(K)=RANG(I)
                  RSAFE(K)=RSAFE(I)
                  K=K+1
              ELSE
                  NOBS=NOBS-1
                  IF (X(I).EQ.XMIN) THEN
                      NMINABER=NMINABER+1
                  ELSE
                      NMAXABER=NMAXABER+1
                  ENDIF
              ENDIF
          CONTINUE
65     ELSE
              IF (CHOIX.EQ.'1') THEN
                  COMPAR=XMIN
              ELSE
                  COMPAR=XMAX
              ENDIF
              DO 70 I=1,NOBS
                  IF (X(I).NE.COMPAR) THEN
                      X(K)=X(I)
                      XSAFE(K)=XSAFE(I)
                      RANG(K)=RANG(I)
                      RSAFE(K)=RSAFE(I)
                      K=K+1
                  ELSE
                      NOBS=NOBS-1
                      IF (COMPAR.EQ.XMIN) THEN
                          NMINABER=NMINABER+1
                      ELSE
                          NMAXABER=NMAXABER+1
                      ENDIF
                  ENDIF
              CONTINUE
70     ENDIF
      ENDIF
```

```
60 IF (TERMINE.EQV..FALSE.) GOTO 5
   REWIND(3)
   WRITE(3,90)NOMFICH,LG,FICHDEB, IDEB, ITYP, IMPRIM, ANDEP, MODEP, JODEP,
   * ANFIN, MOFIN, JOFIN, NOBS, NMINABER, NMAXABER
90  FORMAT (A12,1X,I1/,A12,1X,I1/,I1,1X,I1,/,I2,1X,I2,1X,I2,/,I2,1X,I2,
   * 1X,I2,/,I4,/,I2,1X,I2)
   RETURN
   END
```

SUBROUTINE COFFICH(FICH,RSAFE)

C * * * * *

c *** sous-routine appelee si on a enleve une ou plusieurs valeurs
c *** aberrantes.
c *** copie les valeurs conservees du fichier .TMP a un fichier tam-
c *** poraire.Detruit le fichier .TMP,puis recopie le fichier tampo-
c *** raire sous le meme nom .TMP.
c *** Fait la reecriture dans ident.tmp pour ecraser l'ancienne valeur
c *** de NOBS.

```
CHARACTER*12 FICH,FICHDEB,NOMFICH
CHARACTER*72 CHAINE
CHARACTER*80 LIGNE
INTEGER ANDEP,ANFIN
DIMENSION RSAFE(*)
COMMON /IDENT /NOMFICH,LG,FICHDEB, IDEB, ITYP, IMPRIM, ANDEP, MODEP
* ,JODEP,ANFIN,MOFIN,JOFIN,NOBS
```

```
REWIND(1)
OPEN(4,FILE='WZWZ.TMP',STATUS='NEW')
K=1
10  READ(1,20,END=30)CHAINE,IRANG
20  FORMAT(A72,I5)
   IF (RSAFE(K).EQ.IRANG) THEN
       WRITE(4,20)CHAINE,IRANG
       K=K+1
       GOTO 10
   ELSE
       GOTO 10
   ENDIF
30  CLOSE(1,STATUS='DELETE')
   REWIND(4)
   OPEN(1,FILE=FICH,STATUS='NEW')
   DO 40 I=1,NOBS
       READ(4,50,END=60)LIGNE
       WRITE(1,50)LIGNE
40  CONTINUE
50  FORMAT(A78)
60  CLOSE(4,STATUS='DELETE')
   RETURN
   END
```

```
$INCLU DE: 'TICS.FOR'
$INCLU DE: 'AJUSTE.FOR'
```

Thu 09-10-87 15:24:38

Pg 1

06-30-87 10:53:10 TICS.FOR

```
TICS          TICS.FOR      1
SUBROUTINE TICS(XST,XFIN,XSTS,XFINS,TIC,
```

```
ELSEIF,AINT,ABS
```


SUBROUTINE TICS(XST, XFIN, XSTS, XFINS, TIC, MINOR, AINTE)
REAL*8 INTER, INTER2

C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C

UTILITE : CETTE SOUS-ROUTINE DETERMINE LES ENDROITS OU SERONT
 SITUES LES TICS SUR LES AXES DES GRAPHIQUES

VARIABLES : XST VALEUR LA PLUS PETITE DES DONNEES SUR L'AXE
 EN QUESTION
 XFIN VALEUR LA PLUS GRANDE DES DONNEES SUR L'AXE
 EN QUESTION
 XSTS VARIABLE DE SORTIE DONNANT LA NOUVELLE VALEUR
 DE DEPART DE L'AXE
 XFINS VARIABLE DE SORTIE DONNANT LA NOUVELLE VALEUR
 DE LA FIN DE L'AXE
 TIC VARIABLE DE SORTIE DONNANT LA DISTANCE ENTRE
 CHAQUE TIC
 MINOR VARIABLE DE SORTIE, =0 IMPLIQUE QU'IL N'Y AURA
 PAS DE TICS SECONDAIRES. =1 IMPLIQUE QU'IL Y
 AURA DES TICS SECONDAIRES (UTILISE DANS LE CAS
 OU L'ON AURAIT EU PLUS DE 10 TICS).
 AINTE VARIABLE DE SORTIE DONNANT LE MULTIPLE DE 10 QUE
 L'ON DOIT MULTIPLIER A CHAQUE LABEL.

```

MINOR=0
INTER=1.0
ETEN=XFIN-XST
IF (ETEN.LE.0.0) THEN
  WRITE(*,*) ' *** ERREUR *** XFIN <= XST '
  WRITE(*,*) ' '
  WRITE(*,*) ' ETEN : ', ETEN
  WRITE(*,*) ' XST : ', XST
  WRITE(*,*) ' XFIN : ', XFIN
  WRITE(*,*) ' '
  STOP
ELSEIF (ETEN.GE.1.0) THEN
  4 ETENA=ETEN/INTER
  IF (ETENA.GT.10.0) THEN
    INTER=INTER*10
    GOTO 4
  ENDIF
  TIC=INTER/2
  INTER2 = INTER /10
  IF (ETENA.GT.2.0) THEN
    TIC=INTER
    INTER2 = INTER
  ELSEIF (ETENA.GT.5.0) THEN
    TIC=INTER*2
    INTER2 = INTER
  ENDIF
ELSE
  504 ETENA=ETEN*INTER
  IF (ETENA.LT.1.0) THEN
    INTER=INTER*10
    GOTO 504
  ENDIF

```

```
TIC=1/(INTER*2)
INTER2 = INTER *10
IF (ETENA.GT.2.0) THEN
    TIC=1/INTER
    INTER2 = INTER
ELSEIF (ETENA.GT.5.0) THEN
    TIC=2/INTER
    INTER2 = INTER
ENDIF
ENDIF
IF (XST.GE.0.0) THEN
    XSTS=AINTE(XST/TIC)*TIC
    ENT=AINTE((XFIN-XSTS)/TIC)
    ENT2=XFIN-XSTS/TIC
    IF (ENT.NE.ENT2)ENT=ENT+1
    XFINS=XSTS+(ENT*TIC)
ENDIF
IF (XFIN.LE.0) THEN
    ENT=AINTE(ABS(XST)/TIC)
    ENT2=ABS(XST)/TIC
    IF (ENT.NE.ENT2)ENT=ENT+1
    XSTS=-ENT*TIC
    ENT=AINTE((XFIN-XSTS)/TIC)
    ENT2=(XFIN-XSTS)/TIC
    IF (ENT.NE.ENT2)ENT=ENT+1
    XFINS=XSTS+(ENT*TIC)
ENDIF
IF (XST.LT.0.0.AND.XFIN.GT.0.0) THEN
    ENT=AINTE(ABS(XST)/TIC)
    ENT2=ABS(XST)/TIC
    IF (ENT.NE.ENT2)ENT=ENT+1
    XSTS=-ENT*TIC
    ENT=AINTE(XFIN/TIC)
    ENT2=XFIN/TIC
    IF (ENT.NE.ENT2)ENT=ENT+1
    XFINS=ENT*TIC
ENDIF
ETIC=(XFINS-XSTS)/TIC
IF (ETIC.GE.10.0) THEN
    MINOR=1
    TIC=2*TIC
ENDIF
IF (ETEN.GE.1.0) THEN
    AINTE=INTER2
ELSE
    AINTE=1/INTER2
ENDIF
999 RETURN
STOP
END
```

Thu 09-10-87 14:56:52

04-14-87 12:15:10 AJUSTE.FOR

Pg 1

```
AJUSTE          AJUSTE.FOR          1
SUBROUTINE AJUSTE(X,NOBS,AINTE,TEXTE,XST
```

```
ALOG10,EXP,CHAR,ABS,ELSEIF,MOD,TEXTE
```

```
      SUBROUTINE AJUSTE(X,NOBS,AINTE,TEXTE,XSTS,XFINS,TIC)
      REAL X(*)
      CHARACTER*3 EXP
      CHARACTER*10 TEXTE
      XSTS=XSTS/AINTE
      XFINS=XFINS/AINTE
      TIC=TIC/AINTE
10  DO 10 I=1,NOBS
      X(I)=X(I)/AINTE
      CONTINUE
      IN=ALOG10(AINTE)
      IF (IN.LT.0) THEN
      EXP(1:1)='- '
      EXP(2:2)='0 '
      EXP(3:3)=CHAR(ABS(IN)+48)
      GOTO 20
      ENDIF
      IF (IN.LT.10) THEN
      EXP(1:2)='+0 '
      EXP(3:3)=CHAR(IN+48)
      ELSEIF (IN.LT.20) THEN
      IN=MOD(IN,10)
      EXP(1:1)='+ '
      EXP(2:2)='1 '
      EXP(3:3)=CHAR(IN+48)
      ELSEIF (IN.LT.30) THEN
      IN=MOD(IN,10)
      EXP(1:1)='+ '
      EXP(2:2)='2 '
      EXP(3:3)=CHAR(IN+48)
      ENDIF
20  TEXTE(1:5)='X 10E '
      TEXTE(6:8)=EXP
      TEXTE(9:10)='  '
      RETURN
      END
```



P3.FOR

FICH, NOMFICH, NER, ELSEIF, QSMODE, QCLEAR
QBORD, QCMOV, QCHAR, QBEEP

GRAPHI P3.FOR 92
92 CALL GRAPHI(X,RANG)

P3.FOR 294
SUBROUTINE GRAPHI(X,RANG)

TEXTCT, MIN, MAX, QSMODE, QCOLOR, QOVSCN
QPLOT, QXAXIS, QYAXIS, QGTXT, QPTXTC, QPTXTA
QSETUP, QTABL, QCMOV, QCHAR, QBEEP

TICS P3.FOR 382 x2
382 CALL TICS(XMIN,XMAX,YSTS,YFINS,TIC
383 CALL TICS(0.0,RANG(N),XSTS,XFINS,T

TICS.FOR 1
SUBROUTINE TICS(XST,XFIN,XSTS,XFINS,TIC,

ELSEIF, AINT, ABS

AJUSTE P3.FOR 385 x2
385 CALL AJUSTE(X,N,AINTEY,TEXTEY,YSTS
388 CALL AJUSTE(RANG,N,AINTEX,TEXTEX,X

AJUSTE.FOR 1
SUBROUTINE AJUSTE(X,NOBS,AINTE,TEXTE,XST

ALOG10, EXP, CHAR, ABS, ELSEIF, MOD, TEXTE

FREQMENS P3.FOR 96
 96 20 CALL FREQMENS(DIMMAX,FM,TOTAN,D

P3.FOR 419
 SUBROUTINE FREQMENS(DIMMAX,FM,TOTAN,DEUX

MIN,R1,QSMODE,QCLEAR,QBORD,QCMOV,QCHAR
 QBEEP,ELSEIF,QDATE

MOYFICH P3.FOR 786
 786 CALL MOYFICH(FICH,ILACOL,EMOY,EQTY

P3.FOR 863
 SUBROUTINE MOYFICH(FICH,INUM,EMOY,EQTYPE

R1

COPIFICH P3.FOR 811
 811 CALL COPIFICH(FICH,IANDEB,IMODEB,I

P3.FOR 895
 SUBROUTINE COPIFICH(FICH,IAN1,IMD1,IAN2,

MOYMENS P3.FOR 172
 172 1 CALL MOYMENS(VECT,MOYC,MD,NINT,I

P3.FOR 1335
 SUBROUTINE MOYMENS(VECT,MOYC,MD,NINT,ISI

MIN,MAX

ANOVA P3.FOR 1423
 1423 CALL ANOVA(VECT,MOYC,ISIGNIF,MD,NI

P3.FOR 938
 SUBROUTINE ANOVA(VECTEUR,MOYC,ISIGNIF,MD

MOD, LOG10, QSMODE, QCLEAR, QBORD, QCMOV
 QCHAR, QBEEP

CHI2 P3.FOR 1055
 1055 CALL CHI2(BARTLETT, NINTA-1, ISIGNIF)

CHIDEUX.FOR 1
 SUBROUTINE CHI2(Z, IDL, ISIGNIF)

FLOAT

FISCH P3.FOR 1075
 1075 CALL FISCH(F, NINTA-1, NE, ISIGNIF)

FISCHER.FOR 1
 SUBROUTINE FISCH(Z, NDL1, NDL2, ISIGNIF)

GRAF P3.FOR 1424
 1424 CALL GRAF(VECT, YMIN, YMAX, MOYC, NBAN)

P3PLUS.FOR 233
 SUBROUTINE GRAF(VECTEUR, YMIN, YMAX, MOYC, N

TEXTMM, QSMODE, QCOLOR, QOVSON, QPLOT
 QXAXIS, QYAXIS, QGTXT, QPTXTA, MOD, QSETUP
 QTABL, QPTXTC, QCMOV, QCHAR, QBEEP

TICS P3PLUS.FOR 299 d1
 299 CALL TICS(YMIN, YMAX, YSTS, YFINS, TIC

AJUSTE P3PLUS.FOR 301 d1
 301 CALL AJUSTE(VECTEUR, NBAN*12, AINTE,


```

ANOVA8          P3.FOR  199
199 CALL ANOVA8(MD,INTER,NINT,ISAI8)

```

```

          P3.FOR  1128
SUBROUTINE ANOVA8(MD,INTER,NINT,ISAI8)

```

```

QBEEP, QSMODE, QCLEAR, QBORD, QCMOV, QCHAR
MOD

```

```

FISCH          P3.FOR  1245
1245 CALL FISCH(F,NR,NE,ISIGNIF)

```

```

          FISCHER.FOR  1
SUBROUTINE FISCH(Z,NDL1,NDL2,ISIGNIF)

```

```

REGRESSION     P3.FOR  270
270 CALL REGRESSION(X,Y,XCHAP,DEBI,VEC

```

```

          P3PLUS.FOR  1
SUBROUTINE REGRESSION(X,Y,XCHAP,DEBI,VEC

```

```

TEXTRE, ALOG, MAX, MIN, MSE, QSMODE, QCOLOR
QDVSCN, QPLOT, QXAXIS, QYAXIS, QSETUP, QSTXT
OPTXTA, OPTXTC, QTABL, QCMOV, QCHAR, QBEEP
EXP, ADR, SORT, QCLEAR, QBORD

```

```

TICS          P3PLUS.FOR  131 x2  p1
131 CALL TICS(XMIN,XMAX,XSTS,XFINS,TIC
132 CALL TICS(YMIN,YMAX,YSTS,YFINS,TIC

```

```

AJUSTE        P3PLUS.FOR  134 x2  p1
134 CALL AJUSTE(X,NOBS,AINTEX,TEXTEX,X
137 CALL AJUSTE(Y,NOBS,AINTEY,TEXTEY,Y

```

MOD, LOG10, QSMODE, QCLEAR, QBORD, QCMOV
 QCHAR, QBEEP

CHI2 P3.FOR 1055
 1055 CALL CHI2(BARTLETT, NINTA-1, ISIGNIF)

CHIDEUX.FOR 1
 SUBROUTINE CHI2(Z, IDL, ISIGNIF)

—FLOAT

FISCH P3.FOR 1075
 1075 CALL FISCH(F, NINTA-1, NE, ISIGNIF)

FISCHER.FOR 1
 SUBROUTINE FISCH(Z, NDL1, NDL2, ISIGNIF)

GRAF P3.FOR 1424
 1424 CALL GRAF(VECT, YMIN, YMAX, MOYC, NBAN)

P3PLUS.FOR 233
 SUBROUTINE GRAF(VECTEUR, YMIN, YMAX, MOYC, N

TEXTMM, QSMODE, QCOLOR, QOVSCN, QPLOT
 QXAXIS, QYAXIS, QGTXT, QPTXTA, MOD, QSETUP
 QTABL, QPTXTC, QCMOV, QCHAR, QBEEP

TICS P3PLUS.FOR 299 p1
 299 CALL TICS(YMIN, YMAX, YSTS, YFINS, TIC)

AJUSTE P3PLUS.FOR 301 p1
 301 CALL AJUSTE(VECTEUR, NBAN*12, AINTE,

ANOVA8 P3.FOR 199
199 CALL ANOVA8(MD,INTER,NINT,ISAI8)

 P3.FOR 1128
SUBROUTINE ANOVA8(MD,INTER,NINT,ISAI8)

—QBEEP, QSMODE, QCLEAR, QBORD, QCMOV, QCHAR
—MOD

FISCH P3.FOR 1245
1245 CALL FISCH(F,NR,NE,ISIGNIF)

 FISCHER.FOR 1
SUBROUTINE FISCH(Z,NDL1,NDL2,ISIGNIF)

REGRESSION P3.FOR 270
270 CALL REGRESSION(X,Y,XCHAP,DEBI,VEC

 P3PLUS.FOR 1
SUBROUTINE REGRESSION(X,Y,XCHAP,DEBI,VEC

—TEXTRE, ALDG, MAX, MIN, MSE, QSMODE, QCOLOR
—QOVSCN, QPLOT, QXAXIS, QYAXIS, QSETUP, QGTXT
—QPTXTA, QPTXTC, QTABL, QCMOV, QCHAR, QBEEP
—EXP, AOR, SORT, QCLEAR, QBORD

TICS P3PLUS.FOR 131 x2 p1
131 CALL TICS(XMIN,XMAX,XSTS,XFINS,TIC
132 CALL TICS(YMIN,YMAX,YSTS,YFINS,TIC

AJUSTE P3PLUS.FOR 134 x2 p1
134 CALL AJUSTE(X,NOBS,AINTEX,TEXTEX,X
137 CALL AJUSTE(Y,NOBS,AINTEY,TEXTEY,Y

TESTRHO P3PLUS.FOR 183
183 CALL TESTRHO(NDF,RHOCQ,1,ISIGN)

TESTRHO.FOR 1
SUBROUTINE TESTRHO(NDF,RHO,ISEUIL,ISIGN)

ELSEIF

\$large
\$noflo atcalls
\$stora ge:2

PROGRAM PARTIE3

c -----
c *****
c Titre : Caracterisation de la serie des mesures
c Fonctions :
c . Presentation de l'evolution temporelle (graphiques)
c . Etablissement d'un tableau de frequences mensuelles
c d'echantillonnage
c . Etude des saisonnalites sur une base mensuelle et
c graphique des moyennes mensuelles
c . Recherche d'une relation concentration-debit significative
c qui sera eventuellement utilisee par la suite
c
c NOTE: lors de la description des parametres de chaque sous-routine,
c les identificateurs dont la description se termine par le symbole #
c ne sont utilises qu'a l'interieur de cette sous-routine. Ils ne sont
c passes en parametre que parce qu'ils sont declares dans le programme
c principal. Cette facon de faire permet le dimensionnement des vecteurs
c a l'aide d'une constante, et permet egalement d'utiliser un minimum
c de structures.
c *****

INTEGER*2 DIMMAX,MAXI

c *** maxi est le nombre maximal d'observations possibles

PARAMETER (MAXI=1300)

c *** dimmax est le nombre d'annees maximal sur lequel s'etend l'analyse
c *** et sert a dimensionner la matrice des frequences mensuelles

PARAMETER (DIMMAX=32)

INTEGER*2 FM(DIMMAX,12),TOTAN(DIMMAX),NOBSERV,ANDEF,ANFIN

REAL X(MAXI),Y(MAXI),VECT(MAXI),XCHAP(MAXI),

VEC(MAXI),DEBI(MAXI),RANG(MAXI),MOYC(12)

CHARACTER*1 REP,ELIM

CHARACTER*12 FICH

CHARACTER*10 NMOIS(12)

CHARACTER*5 FMOIS(12)

CHARACTER*12 NOMFICH,FICHDEB

LOGICAL DEUXMOV,TRONC

COMMON /BLOC31/ NOBSERV,NBAN

COMMON /BLOC32/ ITRONC,IANTR,IDURTR

COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR

COMMON /GRAPH/ IGMODE,IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,

```
* ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2
COMMON /IDENT/ NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEF,
* MODEP,JODEF,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER
COMMON /FICH/ FICH
COMMON /FIN/ IFIN
DATA NMOIS/'JANVIER','FEVRIER','MARS','AVRIL','MAI','JUIN',
* 'JUILLET','AOUT','SEPTEMBRE','OCTOBRE','NOVEMBRE','DECEMBRE'/
DATA FMOIS/'(A7\)', '(A7\)', '(A4\)', '(A5\)', '(A3\)', '(A4\)',
* '(A7\)', '(A4\)', '(A9\)', '(A7\)', '(A8\)', '(A8\)' /
```

C *** lecture des specifications graphiques

```
OPEN(2,FILE='SETUP.PC',STATUS='OLD')
READ(2,'(4I2/9I2,5I3)')ITMODE,ITFORE,ITBACK,ITCOLOR,IGMODE,
* IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,ILETR,ISEC,KIRSYM,ISYMBL,
* JROW1,JROW2,JCOL1,JCOL2
CLOSE(2)
```

c *** lecture du fichier ident.tmp

```
OPEN(3,FILE='IDENT.TMP',STATUS='OLD')
READ(3,2)NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEF,MODEP,JODEF,
* ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER
2 FORMAT(A12,1X,I1/,A12,1X,I1./,I1,1X,I1./,I2,1X,I2,1X,I2./,I2,1X,
* I2,1X,I2./,I4./,I2,1X,I2)
3 FORMAT(I1,1X,I2,1X,I2,1X,I2)
```

c *** reconstitution du nom du fichier de travail

```
FICH='
FICH(1:LG)=NOMFICH(1:LG)
FICH(LG+1:LG+4)='.TMP'
OPEN(1,FILE=FICH,STATUS='OLD')
```

c *** graphique du parametre de qualite en fonction du temps

```
CALL GRAPHI(X,RANG)
```

c *** tableau des frequences mensuelles des prelevements

```
20 CALL FREQMENS(DIMMAX,FM,TOTAN,DEUXMOV)
IF (IFIN.EQ.1) GOTO 100
TRONC=.FALSE.
MD=1
NINT=12
```

c *** on offre la possibilite d'eliminer ccertains mois si au moins
c *** deux mois consecutifs ne contiennent par de prelevement
c *** (si c'est le cas, deuxmov="true")

```
4 IF (DEUXMOV.EQV..TRUE.) THEN
  WRITE(*,'(1X,A42\)' )'VOULEZ-VOUS ELIMINER CERTAINS MOIS (O/N/?)'
  READ(*,'(A1)')ELIM
  IF (ELIM.EQ.'O'.OR.ELIM.EQ.'o') THEN
```

c *** on demande de specifier les mois a conserver

```
WRITE(*,*) ' '
WRITE(*, '(A57\)' ) ' ENTREZ LE NUMERO DU PREMIER MOIS A
ELIMI
#   NER (1 A 12) : '
READ(*, '(I2)' ) IPR
WRITE(*, '(A57\)' ) ' ENTREZ LE NUMERO DU DERNIER MOIS A
#   ELIMI
   NER (1 A 12) : '
READ(*, '(I2)' ) IDE
IFREM=IDE+1
IF (IPREM .GE. 13) IPREM=IPREM-12
IDERN=IPR-1
IF (IDERN .LE. 0) IDERN=IDERN+12
MD=IPREM
NINT=IDERN+1-IPREM
IF (NINT .LT. 0) NINT=NINT+12
INTER=1
```

c *** on signale qu'on a effectivement tronque la serie

```
TRONC=. TRUE.
ELSEIF (ELIM.EQ. 'N'.OR.ELIM.EQ. 'n') THEN
MD=1
NINT=12
ELSEIF (ELIM.EQ. '?') THEN
CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
WRITE(*,7) '          TRONCATION DES MOIS ???
WRITE(*,*) ' '
WRITE(*,7) 'LA TRONCATION DES MOIS OFFERTE ICI, VISE A
WRITE(*,7) 'PERMETTRE A L'USAGER D'ELIMINER CERTAINS
WRITE(*,7) 'MOIS N'AYANT PAS ETE ECHANTILLONNES.
WRITE(*,*) ' '
WRITE(*,7) 'VOICI QUELQUES CONSEQUENCES DES CHOIX :
WRITE(*,7) '   OUI : IL NE SERA PLUS POSSIBLE DE REGROU-
WRITE(*,7) '   FER CERTAINS MOIS DANS L'ANOVA A-
WRITE(*,7) '   FIN DE FORMER DES SAISONS.
WRITE(*,7) '   ON PROPOSERA AUTOMATIQUEMENT DES
WRITE(*,7) '   SAISONS DE 1 MOIS SI ON REJETTE
WRITE(*,7) '   L'EGALITE DAS MOYENNES.
WRITE(*,7) '   NON : IL SERA POSSIBLE DE REGROUPER LES
WRITE(*,7) '   MOIS NON-ECHANTILLONNES DANS DES
WRITE(*,7) '   SAISONS.
CALL QCMOV(1,1)
CALL QCHAR(27,ITFORE,1)
CALL QBEEP
READ(*, '(A1)' ) TOUCHE
GOTO 20
ELSE
```

c *** l'usager a entre sa reponse de facon incorrecte

```
CALL QBEEP
CALL QBEEP
GOTO 4
ENDIF
ENDIF

7  FORMAT (20X,A42)
1  CALL MOYMENS(VECT,MOYC,MD,NINT,IS AIS)

c *** on conserve les valeurs de sasonnalite et subdivision de la
c *** serie tronquee
c *** s'il n'y a pas de sasonnalite detectee,on inscrit des zeros

IF (TRONC.EQV..TRUE.) THEN
  IF (IS AIS.EQ.0) THEN
    WRITE(3,3)2,MD,INTER,NINT
  ELSE
    WRITE(3,3)3,MD,INTER,NINT
  ENDIF
ELSEIF (IS AIS.EQ.0) THEN
  WRITE(3,3)0,0,0,0
  GOTO 25
ELSE
c *** on offre alors la possibilite de rediviser les saisons
5  WRITE(*,'(A52,\)')' VOULEZ-VOUS REGROUPER LES MOIS EN SAISONS
   $ (O/N/?)'
   READ(*,70)REP
   IF (REP.EQ.'0'.OR.REP.EQ.'o') THEN
c *** on teste l'egalite des moyennes mensuelles avec les nouvelles
c *** divisions specifiees
6  CALL ANOVAB(MD,INTER,NINT,IS AIS)
   ELSEIF (REP.EQ.'N'.OR.REP.EQ.'n') THEN
     WRITE(3,3)IS AIS,1,1,12
     GOTO 25
   ELSEIF (REP.EQ.'?') THEN
     FORMAT(15X,A48)
     CALL QSMODE(ITMODE)
     CALL QCLEAR(ITBACK,ITFORE)
     CALL QBORD(ITCOLOR)
     WRITE(*,6)' REGROUPEMENTS ???

     WRITE(*,*)' '
     WRITE(*,*)' '
     WRITE(*,6)' DES INTERVALLES NON-SAISONNIERS DE 1 A 21
       JOURS '
     WRITE(*,6)' ET DE 1 A 12 MOIS SERONT POSSIBLES PAR LA
       SUITE '
     WRITE(*,6)' DANS "P4". MAIS ON NE TIENDRA PAS COMPTE
       ALORS '
     WRITE(*,6)' DE LA DISCRIMINATION ENTRE LES MOIS QUE L'
```



```
      ANOVA  
      WRITE(*,6) 'A PERMIS DE DETECTER (SAISONNALITE).  
  
      WRITE(*,*) ' '  
      WRITE(*,6) 'LE BUT PRINCIPAL D'OFFRIR DES REGROUPEMENTS  
      EST '  
      WRITE(*,6) 'D'AVOIR LE PLUS PETIT NOMBRE DE SAISONS  
      POSSIBLE '  
      WRITE(*,6) 'DE MOYENNES SIGNIFICATIVEMENT DIFFERENTES.  
  
      CALL QCMOV(1,1)  
      CALL QCHAR(27,ITFORE,1)  
      CALL QBEEP  
      READ(*,'(A1)') TOUCHE  
      GOTO 1  
ELSE  
      CALL QBEEP  
      CALL QBEEP  
      GOTO 5  
ENDIF
```

c *** l'usager peut alors conserver sa division ou encore en specifier
c *** une nouvelle

```
8      FORMAT(A51)  
9      FORMAT(A44,\)  
10     IF (IS AIS.EQ. 1) THEN  
        WRITE(*,9) 'VOULEZ-VOUS SAUVER CE REGROUPEMENT (O/N) ?'  
        READ(*,70) REP  
        IF (REP.EQ. 'O'.OR.REP.EQ. 'o') THEN  
            WRITE(3,3) IS AIS,MD,INTER,NINT  
        ELSEIF (REP.EQ. 'N'.OR.REP.EQ. 'n') THEN  
            GOTO 5  
        ELSE  
            CALL QBEEP  
            CALL QBEEP  
            GOTO 10  
        ENDIF  
    ELSE  
        WRITE(*,8) 'VOULEZ-VOUS CONTINUER LA RECHERCHE DE  
        REGROUPEMENT '  
        WRITE(*,9) 'DE SAISONS DE MOYENNES DIFFERENTES (O/N) ?'  
        READ(*,70) REP  
        IF (REP.EQ. 'N'.OR.REP.EQ. 'n') THEN  
            WRITE(3,3) IS AIS,MD,INTER,NINT  
        ELSEIF (REP.EQ. 'O'.OR.REP.EQ. 'o') THEN  
            GOTO 5  
        ELSE  
            CALL QBEEP  
            CALL QBEEP  
            GOTO 10  
        ENDIF  
    ENDIF  
ENDIF
```

c *** recherche d'une relation concentration-debit significative
c *** celle-ci peut s'effectuer uniquement si l'on a choisi d'utiliser
c *** les debits
c *** on conserve les resultats de l'analyse dans "ident.tmp"

```
25  IF (IDEB.NE.3) THEN
      CALL REGRESSION(X,Y,XCHAP,DEBI,VEC)
      ELSE
30  WRITE (3,30)0,0.,0.
      FORMAT (I1,1X,F9.2,1X,F9.2)
      ENDIF
      WRITE (3,' (I1,1X,I2,1X,I2) ') ITRONC,IANTR,IDURTR
      CALL QSMODE(ITMODE)
      CALL QCLEAR(0,7)
      CALL QBORD(0)
70  FORMAT (A1)
```

c *** ouverture d'un fichier-bidon pour signaler que l'execution du
c *** programme p3 s'est effectuee normalement i.e. sans erreur provoquant
c *** l'arret imprevu du programme

```
      OPEN (99,FILE='P3.OK',STATUS='NEW')
      CLOSE (99)
      STOP
100  OPEN (99,FILE='STOP.P3',STATUS='NEW')
      CLOSE (99)
      STOP
      END
```

SUBROUTINE GRAPHI (X,RANG)

C * * * * *

c -----

c Fonction :

c . Trace le graphique des charges ou des concentrations
c en fonction du temps

c Parametres :

c . x Vecteur destine a contenir la serie de mesures #
c . rang Vecteur destine a contenir la suite de rangs relatifs
c associes aux dates de prelevements #

c -----

```
REAL X(*),RANG(*)
INTEGER ANDEF,ANFIN
CHARACTER*15 TEXTY
CHARACTER*10 TEXTEX,TEXTEY
CHARACTER*54 TEXTCT
CHARACTER*42 TEXTLT
CHARACTER*12 FICHDEB,NOMFICH
```

```
CHARACTER*20 F
COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH/ IGMODE,IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,
$ ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2
COMMON /IDENT/ NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEF,
$ MODEP,JODEF,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER

c *** positionnement du fichier d'entree (.tmp)

REWIND(1)
I=0
N=0
TEXTEX='
TEXTEY='
XMIN=1E30
XMAX=0.0

c *** selon le parametre etudie,on determine les titres et les
c *** formats a utiliser

TEXTCT='
IF (ITYP.EQ.0) THEN
TEXTLT='FIGURE 3.1 : CONCENTRATION VS TEMPS, '
TEXTCT(1:42)=TEXTLT
TEXTCT(39:50)=NOMFICH
TEXTY=' CONCEN mg/1 '
F='(9X,E16.10,47X,I5)'
ELSE
TEXTLT='FIGURE 3.2 : CHARGE VS TEMPS, '
TEXTCT(1:42)=TEXTLT
TEXTCT(39:50)=NOMFICH
TEXTY=' CHARGE kg/jour'
F='(43X,E16.10,13X,I5)'
ENDIF

c *** lecture du fichier de mesures

READ(1,F,END=30)DONN,IRANDEB
BACKSPACE(1)
5 READ(1,F,END=30)DONN,IRAN
I=I+1
X(I)=DONN

c *** on retient le maximum et le minimum rencontres

XMIN=MIN(XMIN,X(I))
XMAX=MAX(XMAX,X(I))
RANG(I)=IRAN-IRANDEB+1
GOTO 5
30 N=I

c *** configuration en mode graphique
CALL QSMODE(IGMODE)
```

```
IF (IGMODE.LT.10) THEN  
    CALL QCOLOR(IGBACK,IPAL)  
ELSE  
    CALL QOVSCN(IGBORD)  
ENDIF
```

c *** calcul de l'espacement des "tics"

```
CALL TICS(XMIN,XMAX,YSTS,YFINS,TICY,MINORY,AINTEY)  
CALL TICS(O.O,RANG(N),XSTS,XFINS,TICX,MINORX,AINTEX)  
IF (AINTEY.GE.10.OR.AINTEY.LT.1) THEN  
    CALL AJUSTE(X,N,AINTEY,TEXTEY,YSTS,YFINS,TICY)  
ENDIF  
IF (AINTEX.GE.10.OR.AINTEX.LT.1) THEN  
    CALL AJUSTE(RANG,N,AINTEX,TEXTEX,XSTS,XFINS,TICX)  
ENDIF
```

c *** tracage de la courbe

```
CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,XSTS,XFINS,YSTS,YFINS,XSTS,  
* YSTS,0,1.0,1.5)  
CALL QXAXIS(XSTS,XFINS,TICX,MINORX,1,0)  
CALL QYAXIS(YSTS,YFINS,TICY,MINORY,1,0)  
CALL QGTXT(50,TEXTCT,ILETR,JCOL1+60,JROW2+12,0)  
CALL QGTXT(10,TEXTEX,ILETR,JCOL2-35,JROW1+10,0)  
CALL QPTXTC(10,TEXTEY,ILETR)  
CALL QPTXTA(15,'TEMPS (# JOURS)',ILETR)  
IF (IGMODE.EQ.16) THEN  
    INCR=85  
ELSE  
    INCR=55  
ENDIF  
CALL QGTXT(15,TEXTY,ILETR,0,INCR+(JROW2+JROW1)/2,-1)  
CALL QSETUP(0,ILINE,ISYMBL,KIRSYM)  
CALL QTABL(1,N,RANG,X)  
CALL QCMOV(1,1)  
IF (IMPRIM.EQ.1) CALL QPSCRN  
CALL QCHAR(27,ITFORE,1)  
CALL QBEEP  
40 READ(*,40) TOUCHE  
    FORMAT(A1)  
    RETURN  
    END
```

```
SUBROUTINE FREQMENS(DIMMAX,FM,TOTAN,DEUXMOV)  
* * * * *
```

c
c -----
c Fonction :
c
c . Produit le tableau de frequences mensuelles des observations
c
c Parametres :
c

c . dimmax Nombre maximal d'annees sur lesquelles l'analyse
c s'etend.
c . fm Matrice destinee a contenir les frequences mensuelles
c Ses dimensions maximales sont dimmax X 12 #
c . totan Vecteur destine a recevoir les totaux annuels,expri-
c mes en termes de nombre de prelevements #
c . deuxmov Boolean qui vaut "true" en sortie si l'on doit offrir
c a l'usager la possibilite de tronquer sa serie
c (deux mois consecutifs ne contiennent alors aucune donnee)

c -----
\$ INTEGER*2 TOTMOIS(12),SOMX,TOTAL,NOBSERV,DIMMAX,FM,I,J,AN,
MO,LOW,HIGH,NBAN,TOTAN,ANDEP,ANFIN,IOKAN
INTEGER*4 SOMX2
REAL ECTYPE,MOYX,R1
LOGICAL DEUXMOV,DEUXANV
DIMENSION TOTAN(DIMMAX),FM(DIMMAX,12),IOKAN(50)
CHARACTER*1 TOUCHE
CHARACTER*12 FICHDEB,NOMFICH,FICH
CHARACTER*10 NMOIS(12)
CHARACTER*5 FMOIS(12)
COMMON /BLOC31/NOBSERV,NBAN
COMMON /BLOC32/ ITRONC,IANTR,IDURTR
COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR
\$ COMMON /IDENT/ NOMFICH,LG,FICHDEB,IDEB,ITYP,IMFRIM,ANDEP,
MODEP,JODEP,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER
COMMON /FICH/ FICH
COMMON /FIN/ IFIN
DATA NMOIS/'JANVIER','FEVRIER','MARS','AVRIL','MAI','JUIN',
\$ 'JUILLET','AOUT','SEPTEMBRE','OCTOBRE','NOVEMBRE','DECEMBRE'/
DATA FMOIS/'(A7\)', '(A7\)', '(A4\)', '(A5\)', '(A3\)', '(A4\)',
\$ '(A7\)', '(A4\)', '(A9\)', '(A7\)', '(A8\)', '(A8\)' /

C *** Positionnement du fichier de lecture (.tmp)

100 REWIND(1)
DEUXMOV=.FALSE.

c *** valeurs par default
IANTR=0
IDURTR=0
ITRONC=0
IOUT = 0

c *** initialisation de la matrice a zero

```

      DD 110 I=1,DIMMAX
      IOKAN(I) = 1
      DD 110 J=1,12
      FM(I,J)=0
110 CONTINUE
```

c *** lecture des donnees

```
      READ(1,900)AN,MO  
      NOBSERV=1  
900   FORMAT(I2,1X,I2)
```

c *** retenir l'annee la plus petite pour ajuster les indices en consequence

```
      LOW=AN
```

c *** on incremente l'element de la matrice qui correspond a l'annee et
c *** au mois lus

```
120   FM(AN-LOW+1,MO)=FM(AN-LOW+1,MO)+1  
      READ(1,900,END=130)AN,MO
```

c *** on incremente le nombre total d'observations

```
      NOBSERV=NOBSERV+1  
      GOTO 120
```

c *** memoriser l'annee la plus grande(la derniere)

```
130   HIGH=AN
```

c *** calcul du nombre d'annees effectif sur lequel s'etend l'analyse

```
      NBAN=HIGH-LOW+1  
      IANDEB = LOW  
      IANFIN = HIGH
```

c *** calcul des totaux annuels i.e. pour chaque annee, on aura le nombre
c *** de prelevements pris dans cette annee

```
      MINTOT=100  
140   DO 150 I=1,NBAN  
      TOTAL=0  
      DO 140 J=1,12  
      TOTAL=TOTAL+FM(I,J)  
      CONTINUE  
      TOTAN(I)=TOTAL  
      MINTOT=MIN(MINTOT,TOTAN(I))  
150   CONTINUE
```

c *** calcul des totaux mensuels i.e. pour chaque mois, on aura le nombre
c *** de prelevement pris dans ce mois,a travers les annees.

```
160   DO 180 J=1,12  
      TOTAL=0  
      DO 170 I=1,NBAN  
      IF(IOKAN(I).EQ.1)TOTAL=TOTAL+FM(I,J)  
      CONTINUE  
      TOTMOIS(J)=TOTAL  
180   CONTINUE
```

c *** calcul de la moyenne et de l'ecart-type associes aux totaux annuels

```
SOMX=0
SOMX2=0
R1 = 0.0
DO 190 I=1,NBAN
  IF (IOKAN(I).EQ.1) THEN
    SOMX=SOMX+TOTAN(I)
    SOMX2=SOMX2+(TOTAN(I)**2)
    R1 = R1 + 1
  ENDIF
190 CONTINUE
MOYX=SOMX/R1
ECTYPE=(1.0/R1*(SOMX2-R1*(MOYX**2)))*0.5

c *** impression de la matrice
ASSIGN 235 TO ISORTIE
200 CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
210 WRITE(IOUT,910)
910 FORMAT(/,/ ,/)
WRITE(IOUT,*) '
$ S OBSERVEES '
WRITE(IOUT,*) '
WRITE(IOUT,915)
915 FORMAT(15X,'J F M A M J J A S O N D TOTAL/A
$ N')
WRITE(IOUT,920)
920 FORMAT(15X,'
$ _ ,/)
NLIGNES=4
DO 230 I=1,NBAN
  IF (IOKAN(I).EQ.1) THEN
    WRITE(IOUT,930) I+LOW-1, (FM(I,J),J=1,12),TOTAN(I)
    NLIGNES=NLIGNES+1
  ENDIF
  IF (NLIGNES.EQ.21) THEN
    CALL QCMOV(1,1)
    CALL QCHAR(27,ITFORE,1)
    CALL QBEEP
    READ(*,1040) TOUCHE
  ENDIF
930 FORMAT(6X,'19',I2,4X,I2,2X,I2,2X,I2,2X,I2,2X,I2,2X,I2,2X,I2,
$ 2X,I2,2X,I2,2X,I2,2X,I2,2X,I2,5X,I3)
230 CONTINUE
WRITE(IOUT,920)
WRITE(IOUT,940) (TOTMOIS(I),I=1,12),SOMX
940 FORMAT(4X,'TOT/MOIS',1X,I3,1X,I3,1X,I3,1X,I3,1X,I3,1X,I3,1X,
$ I3,1X,I3,1X,I3,1X,I3,1X,I3,4X,I4)
WRITE(IOUT,950) MOYX
950 FORMAT(/,32X,'NOMBRE MOYEN D' OBSERVATION/AN : ',F7.3)
WRITE(IOUT,960) ECTYPE
960 FORMAT(20X,'ECART-TYPE DU NOMBRE DE MESURES ANNUELLES : ',F7.3,
$ /,/)
GOTO ISORTIE
235 CALL QCMOV(1,1)
```

```
CALL QCHAR(27,ITFORE,1)
CALL QBEEP
READ(*,1040) TOUCHE
```

c *** on propose d'eliminer des annees si le nombre minimal de valeurs
c *** pour une annee est plus petit que 4

```
IF (MINTOT.GT.4.) THEN
    ITRONC=9
    GOTO 380
ENDIF
```

c *** impression du menu

```
240 CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
WRITE(*,*) ' '
WRITE(*,*) ' Comme le nombre de valeur pour certaine(s) annee(s)'
WRITE(*,*) ' est petit vous avez la possibilite de limiter le'
WRITE(*,*) ' traitement '
WRITE(*,*) ' '
WRITE(*,*) ' 1) ELIMINER CERTAINES ANNEES INTERMEDIAIRES'
WRITE(*,*) ' '
WRITE(*,*) ' 2) ELIMINER CERTAINS MOIS AUX EXTREMITES'
WRITE(*,*) ' '
WRITE(*,*) ' 3) RETOUR AUX DECOMPTES AVANT ELIMINATION'
WRITE(*,*) ' '
WRITE(*,*) ' 4) VOIR LE TABLEAU COMPLET ACTUEL'
WRITE(*,*) ' '
WRITE(*,*) ' 5) DISTRIBUTION NON-REPRESENTATIVE:'
WRITE(*,*) ' SORTIE SANS TEST'
WRITE(*,*) ' '
WRITE(*,*) ' 6) SUITE NORMALE DU LOGICIEL'
WRITE(*,*) ' '
WRITE(*,*) ' ?) POUR AVOIR DE L' AIDE'
WRITE(*,*) ' '
WRITE(*,*(A28\)) ' Votre Choix : '
READ(*,1040) REP
```

C ****

```
IF (REP.EQ.'1'.AND.ITRONC.EQ.0) THEN
```

c *** demander quelles annees on doit tronquer. Les annees doivent etre
c *** consecutives

```
250 WRITE(*,*) ' '
WRITE(*,*(A43\)) ' ENTREZ LA PREMIERE ANNEE A TRONQUER : 19'
READ(*,*(I2)) IANTR
IF (IANTR.LT.LOW.OR.IANTR.GT.LOW+NBAN-1) THEN
c *** l'annee precisee depasse les bornes de la serie originale
WRITE(*,*) ' *** ANNEE INVALIDE ***'
CALL QBEEP
GOTO 250
ENDIF
WRITE(*,*(A41\)) ' ENTREZ LE NOMBRE D' ANNEES A TRONQUER : '
READ(*,*(I2)) IDURTR
IF (IDURTR+IANTR.GT.LOW+NBAN) THEN
WRITE(*,*) ' '
```



```
WRITE(*,*)' *** DUREE INVALIDE ***'  
WRITE(*,970)' VOTRE SERIE SE TERMINE A L'ANNEE '  
    LOW+NBAN-1  
WRITE(*,*)'  
CALL QBEEP  
GOTO 250  
ENDIF  
970  FORMAT(A35,I2)  
    ITRONC=1  
DO 260 I=IANTR-LOW+1,IANTR-LOW+IDURTR  
    IOKAN(I) = 0  
260  CONTINUE  
    GOTO 160  
C ****  
ELSEIF(REP.EQ.'2') THEN  
    CALL QSMODE(ITMODE)  
    CALL QCLEAR(ITBACK,ITFORE)  
    CALL QBORD(ITCOLOR)  
270  WRITE(*,*)'  
    WRITE(*,*)' ENTREZ LA DATE DU DEBUT DU TRAITEMENT '  
    WRITE(*,*)'  
    WRITE(*,'(A27\)' )'      LE MOIS (1-12)  : '  
    READ(*,*,ERR=270)IMODEB  
    IF (IMODEB.LE.0.OR.IMODEB.GT.12) THEN  
        CALL QBEEP  
        GOTO 270  
    ENDIF  
    WRITE(*,'(A27\)' )'      L'ANNEE           : 19'  
    READ(*,*,ERR=270)IANDEB  
c *** l'annee precisee depasse les bornes de la serie originale  
    IF (IANDEB.LT.LOW.OR.IANDEB.GT.LOW+NBAN-1) THEN  
        WRITE(*,*)'  
        WRITE(*,*)' *** ANNEE INVALIDE ***'  
        WRITE(*,*)'  
        CALL QBEEP  
        GOTO 270  
    ENDIF  
280  WRITE(*,*)'  
    WRITE(*,*)' ENTREZ LA DATE DE LA FIN DU TRAITEMENT '  
    WRITE(*,*)'  
    WRITE(*,'(A27\)' )'      LE MOIS (1-12)  : '  
    READ(*,*,ERR=280)IMOFIN  
    IF (IMOFIN.LE.0.OR.IMOFIN.GT.12) THEN  
        CALL QBEEP  
        GOTO 280  
    ENDIF  
    WRITE(*,'(A27\)' )'      L'ANNEE           : 19'  
    READ(*,*,ERR=280)IANFIN  
    IF (IANFIN.GT.LOW+NBAN-1.OR.IANFIN.LT.IANDEB) THEN  
        WRITE(*,*)'  
        WRITE(*,*)' *** ANNEE INVALIDE ***'  
        WRITE(*,*)'  
        CALL QBEEP  
        GOTO 270  
    ENDIF
```

```

290      DO 290 I=1,IANDEB-LOW
          IOKAN(I) = 0
          CONTINUE
300      DO 300 I=IANFIN-LOW+2,NBAN
          IOKAN(I) = 0
          CONTINUE
          GOTO 160
C ****
      ELSEIF (REP.EQ.'3') THEN
370      DO 370 I=1,NBAN
          IOKAN(I) = 1
          CONTINUE
          IANTR=0
          IDURTR=0
          ITRONC=0
          IANDEB=LOW
          IANFIN=HIGH
          GOTO 160
C ****
      ELSEIF (REP.EQ.'4') THEN
          GOTO 200
C ****
      ELSEIF (REP.EQ.'5') THEN
c *** valeurs par default
          IANTR=0
          IDURTR=0
          ITRONC=99
          IANDEB=LOW
          IANFIN=HIGH
310      OPEN (6,FILE='SYNTHESE.P6',STATUS='OLD')
          READ (6,980,END=320)LIGNE
          GOTO 310
980      FORMAT (A80)
320      BACKSPACE (6)
          CALL QCLEAR (ITBACK,ITFORE)
          CALL QCMOV (0,23)
          CALL QDATE (IAN,IMO,IJR)
c *** debut de l'impression de l'information
330      IOUT = 6
          WRITE (IOUT,*) ' '
          WRITE (IOUT,990) 'S Y N T H E S E           SESSION DU
          $           ',IJR
990      FORMAT (1X,A46,I2,1X\ )
          WRITE (IOUT,FMOIS (IMO))NMOIS (IMO)
          WRITE (IOUT,' (1X,I4) ') IAN
          WRITE (IOUT,*) ' '
          WRITE (IOUT,995) 'NOM DU FICHER : ',NOMFICH,'SERIE TRAITEE : '
995      FORMAT (1X,A18,A12,A17\ )
          IF (ITYP.EQ.0) THEN
              WRITE (IOUT,*) ' CONCENTRATIONS '
          ELSE
              WRITE (IOUT,*) ' CHARGES '
          ENDIF
          FORM1=' (F10.3) '
          FORM2=' (F10.3\ ) '
```

```
WRITE (IOUT,*) ' '
WRITE (IOUT,1010) 'SERIE ORIGINALE :      DATE DE LA PREMIERE OBSERV
1010 $   ATION : ',JODEP,'/',MODEP,'/',LOW
FORMAT (1X,A56,I2,A1,I2,A1,I2)
WRITE (IOUT,1010) 'DATE DE LA DERNIERE OBSERVATION : ',JOFIN,
1020 $   '/',MOFIN,'/',ANFIN
WRITE (IOUT,1020) 'NOMBRE D 'OBSERVATIONS      : ',NOBS
WRITE (IOUT,1020) 'NOMBRE DE MINIMA RETIRES   : ',NMINABER
WRITE (IOUT,1020) 'NOMBRE DE MAXIMA RETIRES   : ',NMAXABER
1020 $   FORMAT (1X,A56,I4)
IF (IOUT.NE.0) THEN
  IOUT = 0
  GOTO 330
ENDIF
DO 340 I=1,NBAN
  IOKAN(I) = 1
340 $ CONTINUE
C ***  impression du tableau du nombre de valeur
ASSIGN 360 TO ISORTIE
IOUT = 6
GOTO 210
360 $ ASSIGN 235 TO ISORTIE
CLOSE (1)
IF (ITYP.EQ.0) THEN
  ILACOL = 4
ELSE
  ILACOL = 6
ENDIF
CALL MOYFICH(FICH,ILACOL,EMOY,EQTYPE)
IOUT = 6
365 $ WRITE (IOUT,*) ' '
WRITE (IOUT,1025) ' MOYENNE DES DONNEES      : ',EMOY
1025 $ WRITE (IOUT,1025) ' ECART-TYPE DES DONNEES : ',EQTYPE
FORMAT (2X,A40,3(FB.3,2X))
WRITE (IOUT,*) ' '
WRITE (IOUT,*) '          PAS DE TRAITEMENT, ECHANTILLONNAGE TROP '
WRITE (IOUT,*) '          IRREGULIER OU DONNEES PEU REPRESENTATIVES '
WRITE (IOUT,*) ' '
IF (IOUT.NE.0) THEN
  IOUT = 0
  GOTO 365
ENDIF
IFIN = 1
CALL QCMOV(1,1)
CALL QCHAR(27,ITFORE,1)
CALL QBEEP
READ(*,1040) TOUCHE
RETURN
C **** ELSEIF (REP. EQ. '6') THEN
  WRITE (*,*) ' '
  IF (IANDEB.NE.LOW.OR.IANFIN.NE.HIGH) THEN
    CALL COPIFICH(FICH,IANDEB,IMODEB,IANFIN,IMOFIN)
    ANDEP = IANDEB
```

```

ANFIN = IANFIN
MODEP = IMODEB
MOFIN = IMOFIN
JODEP = 1
JOFIN = 1
ENDIF
C ****
ELSEIF (REP.EQ. '?') THEN
CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
WRITE(*,1030) ' TRONCATION DES ANNEES ???
WRITE(*,*) '
WRITE(*,1030) 'LA TRONCATION OFFERTE ICI, VISE A ELIMINER'
WRITE(*,1030) 'CERTAINES ANNEES QUI N'ONT PAS ETE SUFFI-
WRITE(*,1030) 'SAMMENNT ECHANTILLONNEES. LE CHOIX EST
WRITE(*,1030) 'LAISSE A L'USAGER (VOIR MANUEL POUR PLUS
WRITE(*,1030) 'DE RENSEIGNEMENTS).
WRITE(*,*) '
CALL QCMOV(1,1)
CALL QCHAR(27,ITFORE,1)
CALL QBEEP
READ(*,1040) TOUCHE
GOTO 200
C ****
ELSE
CALL QBEEP
GOTO 240
ENDIF
C ****
1030 FORMAT (20X,A42)
1040 FORMAT (A1)
NBZERO=0
```

c *** on met deuxmov a "true" si au moins deux mois consecutifs ne
c *** contiennent aucune donnee

```

380 DO 390 I=1,11
IF (TOTMOIS(I).LE.2.AND.TOTMOIS(I+1).LE.2) DEUXMOV=.TRUE.
390 CONTINUE
IF (ITRONC.EQ.99)GOTO 400
REWIND(3)
WRITE(3,2)NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEP,MODEP,JODEP,
# ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER
2 FORMAT (A12,1X,I1/,A12,1X,I1,/,I1,1X,I1,/,I2,1X,I2,1X,I2,/,I2,1X,
# I2,1X,I2,/,I4,/,I2,1X,I2)
400 RETURN
END
```

SUBROUTINE MOYFICH(FICH,INUM,EMOY,EGTYPE)

```

C***** * * * * *
C
C Calcule la moyenne et l'ecart-type de la INUMe colonnes de donnee
C du fichier FICH
```

C
C FICH nom du fichier (entree)
C INUM colonne de donnee a part de la date (entree)
C MOY la moyenne (sortie)
C EQTYPE ecart-type (sortie)
C
C*****

CHARACTER*12 FICH
REAL VAL(30)
REAL*8 SOM2

R1 = 0.0
SOM=0
SOM2=0
OPEN (4,FILE=FICH,STATUS='OLD')
10 READ(4,*,END=30) (VAL(I),I=1,INUM)
SOM=SOM+VAL(INUM)
SOM2=SOM2+(VAL(INUM)**2)
R1 = R1 + 1
GOTO 10
30 EMOY=SOM/R1
EQTYPE=(1.0/R1*(SOM2-R1*(EMOY**2)))*0.5
RETURN
END

SUBROUTINE COPIFICH(FICH,IAN1,IMD1,IAN2,IMD2)

C * * * * *

c *** sous-routine appelee si on a enleve une ou plusieurs valeurs
c *** aberrantes.
c *** copie les valeurs conservees du fichier .TMP a un fichier tam-
c *** poraire.Detruit le fichier .TMP,puis recopie le fichier tempo-
c *** raire sous le meme nom .TMP.
c *** Fait la reecriture dans ident.tmp pour ecraser l'ancienne valeur
c *** de NOBS.

CHARACTER*12 FICH,FICHDEB,NOMFICH
CHARACTER*72 CHAINE
CHARACTER*80 LIGNE
INTEGER ANDEF,ANFIN
COMMON /IDENT/ NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEF,
* MODEP,JOSEP,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER

REWIND(1)
NOBS = 0
OPEN(4,FILE='WZWZ.TMP',STATUS='NEW')
K=1
10 READ(1,20,END=30) IAN,IMD,CHAINE
20 FORMAT(I2,1X,I2,A72)
IF (IAN.LT.IAN1) GOTO 10
IF (IAN.EQ.IAN1.AND.IMD.LT.IMD1) GOTO 10
IF (IAN.EQ.IAN2.AND.IMD.GT.IMD2) GOTO 10
IF (IAN.GT.IAN2) GOTO 10

```
WRITE (4,20) IAN,IMD,CHaine
NOBS = NOBS +1
GOTO 10
30 CLOSE (1,STATUS='DELETE')
REWIND(4)
OPEN(1,FILE=FICH,STATUS='NEW')
DO 40 I=1,NOBS
    READ(4,50,END=60)LIGNE
    WRITE(1,50)LIGNE
40 CONTINUE
50 FORMAT(A78)
60 CLOSE(4,STATUS='DELETE')
RETURN
END
```

SUBROUTINE ANOVA(VECTEUR,MOYC,ISIGNIF,MD,NINT,NBAN)

Fonction:

. tester l'egalite des moyennes mensuelles, sur une base de
douze saisons par annee

Parametres:

. moyc Vecteur destine a contenir les moyennes mensuelles
. isignif Vaut "1" en sortie si on rejette l'egalite des moyennes

```
-----  
REAL VECTEUR(*),VEC(12),VEC2(12),MOYC(12),S2(12),MSTR,MSE,MST  
INTEGER*2 N(12),ANDEF,ANFIN  
CHARACTER*33 TEXTSIGN,TEXTB  
CHARACTER*12 FICHDEB,NOMFICH  
CHARACTER*20 FO  
CHARACTER*42 AVERTI  
COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR  
COMMON /GRAPH/ IGMODE,IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,  
* ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2  
COMMON /IDENT/ NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEF,  
* MODEP,JOSEP,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER
```

c *** positionnement du fichier d'entree (.tmp)

REWIND(1)

c *** initialisations a zero

```
NOB=0  
TOT=0  
TOT2=0  
SSTR=0  
DO 3 I=1,12  
    MOYC(I)=0.
```

```
      N(I)=0  
      VEC(I)=0.  
      VEC2(I)=0.  
3 CONTINUE
```

c *** utilisation des valeurs dans les mois appropriés
c pour les calculs de l'analyse de variance

```
      IDERN=MD+NINT  
      IF (IDERN.LE.13) THEN  
        DO 5 I=1,NBAN  
          DO 4 J=1,12  
            IF (J.GE.MD.AND.J.LT.IDERN) THEN  
              M=(I-1)*12+J  
              VEC(J)=VEC(J)+VECTEUR(M)  
              VEC2(J)=VEC2(J)+VECTEUR(M)**2  
              MOYC(J)=MOYC(J)+VECTEUR(M)  
              TOT=TOT+VECTEUR(M)  
              TOT2=TOT2+VECTEUR(M)**2  
              IF (VECTEUR(M).NE.0) N(J)=N(J)+1  
            ENDIF  
          CONTINUE  
        CONTINUE  
      ELSE  
4  
5
```

c *** la troncation amène le dernier mois à être plus grand que 12

c *** lecture des données

```
      IDERN=MOD(IDERN,12)  
      DO 7 I=1,NBAN  
        DO 6 J=1,12  
          IF (J.GE.MD.OR.J.LT.IDERN) THEN  
            M=(I-1)*12+J  
            VEC(J)=VEC(J)+VECTEUR(M)  
            VEC2(J)=VEC2(J)+VECTEUR(M)**2  
            MOYC(J)=MOYC(J)+VECTEUR(M)  
            TOT=TOT+VECTEUR(M)  
            TOT2=TOT2+VECTEUR(M)**2  
            IF (VECTEUR(M).NE.0) N(J)=N(J)+1  
          ENDIF  
        CONTINUE  
      CONTINUE  
6  
7
```

```
      ENDIF  
      NINTA=12  
      SP=0.  
      BARN=0.  
      BARD=0.
```

```
      DO 40 I=1,12  
        IF (N(I).EQ.0) THEN  
          NINTA=NINTA-1  
          MOYC(I)=0.  
          S2(I)=0.  
        ELSE
```

```
NOB=NOB+N(I)  
MOYC(I)=VEC(I)/N(I)  
IF(N(I).NE.1) THEN  
    S2(I)=(VEC2(I)/(N(I)-1))-((VEC(I)**2)/(N(I)*(N(I)  
    -1)))  
ENDIF  
SP=SP+S2(I)*(N(I)-1)  
IF(S2(I).NE.0.) THEN  
    BARN=BARN+LOG10(S2(I))*(N(I)-1)  
ENDIF  
IF(N(I).NE.1) THEN  
    BARD=BARD+1./(N(I)-1)  
ENDIF  
SSTR=SSTR+VEC(I)**2/N(I)  
ENDIF  
40 CONTINUE
```

c *** test sur l'egalite des variances

```
SP=SP/(NOB-NINTA)  
BARN=(NOB-NINTA)*LOG10(SP)-BARN  
BARD=1+((1./(3.*(NINTA-1)))*(BARD-(1./(NOB-NINTA))))  
BARTLETT=2.3026*BARN/BARD  
CALL CHI2(BARTLETT,NINTA-1,ISIGNIF)  
AVERTI='  
IF(ISIGNIF.EQ.1) THEN  
    AVERTI='EGALITE DES VARIANCES REJETEE, ATTENTION !'  
ENDIF
```

c *** calcul des statistiques pour la table ANOVA

```
SSTR=SSTR-TOT**2/NOB  
MSTR=SSTR/(NINTA-1)  
SST=TOT2-TOT**2/NOB  
NT=NOB-1  
MST=SST/NT  
SSE=SST-SSTR  
NE=NOB-NINTA  
MSE=SSE/NE  
F=MSTR/MSE
```

c *** verification de la signification du test

```
CALL FISCH(F,NINTA-1,NE,ISIGNIF)  
IF (ISIGNIF.EQ.0) THEN  
    TEXTSIGN='ON ACCEPTE L'EGALITE DES MOYENNES'  
    TEXTB='PAS DE SAISONNALITE'  
ELSE  
    TEXTSIGN='ON REJETTE L'EGALITE DES MOYENNES'  
    TEXTB='PRESENCE DE SAISONNALITE'  
ENDIF  
41 FORMAT (A8,I7,E15.5,E13.5,F9.3)  
42 FORMAT (A8,I7,E15.5,E13.5)
```

c *** remise en mode texte


```
CALL QSMODE(ITMODE)  
CALL QCLEAR(ITBACK,ITFORE)  
CALL QBORD(ITCOLOR)
```

c *** affichage du tableau de resultats

```
WRITE(*,*)'  
IF (ITYP.EQ.0)THEN  
    WRITE(*,*)'  
    WRITE(*,*)' TABLEAU 3.2 ANOVA POUR LES CONCENTRATIONS '  
ELSE  
    WRITE(*,*)'  
    WRITE(*,*)' TABLEAU 3.3 ANOVA POUR LES CHARGES '  
ENDIF  
WRITE(*, '(/,/))'  
WRITE(*,*)'  
WRITE(*,*)'  
WRITE(*,*)' TABLE ANOVA POUR L'EGALITE DES MOYENNES MENSUELLES '  
WRITE(*,*)'  
WRITE(*, '(/,/))'  
WRITE(*,*)' Source d.l. ss ms F '  
WRITE(*,*)'  
WRITE(*,41)' Mois ',NT-NE,SSTR,MSTR,F  
WRITE(*,42)' Erreur ',NE,SSE,MSE  
WRITE(*,*)'  
WRITE(*,42)' Total ',NT,SST,MST  
WRITE(*,*)'  
WRITE(*,*)'  
WRITE(*,*)'COMMENTAIRES: ',TEXTSIGN  
WRITE(*, '(15X,A33)')TEXTB  
WRITE(*, '(15X,A42)')AVERTI  
CALL QCMOV(1,1)  
CALL QCHAR(27,ITFORE,1)  
CALL QBEEP  
70 READ(*,70) TOUCHE  
FORMAT(A1)  
RETURN  
END
```

SUBROUTINE ANOVAB(MD,INTER,NINT,ISAIS)

```
C *****  
C -----  
C Fonction :  
C . teste l'egalite des moyennes, a l'aide de la division saison-  
C niere faite par l'usager  
C Parametres :  
C . md Mois de depart pour la division en saisons  
C . inter Intervalle pour la division en saisons  
C . nint Nombre resultant d'intervalles par annee
```

c -----

```
REAL VEC(12),VEC2(12),MOYC(12),MSTR,MSE,MST
INTEGER*2 N(12),ANDEP,ANFIN
CHARACTER*33 TEXTSIGN,TEXTB
CHARACTER*12 FICHDEB,NOMFICH
CHARACTER*1 REP
CHARACTER*30 FO
COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH/ IGMODE,IGBACK,IFAL,IGBORD,IGCOLOR,ILINE,
$ ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2
COMMON /BLOC31/ NOBSERV,NBAN
COMMON /IDENT/ NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEP,
$ MODEP,JOEP,ANFIN,MDFIN,JOFIN,NOBS,NMINABER,NMAXABER
```

c *** lecture au clavier des parametres de la division saisonniere

```
1 WRITE(*,'(A47)') ' DONNEZ LE NUMERO DU MOIS COMMENCANT LE PREMIER '
WRITE(*,'(A27)') ' INTERVALLE COMPLET (1 A 6) '
READ(*,*)MD
IF (MD.LT.1.OR.MD.GT.6) THEN
    CALL QBEEP
    CALL QBEEP
    GOTO 1
ENDIF
2 WRITE(*,'(A50)') ' DONNEZ LA DUREE DES SAISONS EN MOIS (2,3,4,6) '
$ :
READ(*,*)INTER
IF (INTER.NE.2.AND.INTER.NE.3.AND.INTER.NE.4.AND.INTER.NE.6.) THEN
    CALL QBEEP
    CALL QBEEP
    GOTO 2
ENDIF
```

c *** calcul du nombre d'intervalles par annee

```
NINT=12/INTER
```

c *** initialisations

```
K=0
TOT=0
TOT2=0
SSTR=0
DO 3 I=1,12
    MOYC(I)=0.
    N(I)=0
    VEC(I)=0.
    VEC2(I)=0.
3 CONTINUE
INIT=0
J=1
```

c *** determination du format de lecture a l'aide du parametre de qualite

c *** choisi pour l'analyse

```
IF (ITYP.EQ.0) THEN
    FO='(3X,I2,4X,E16.10) '
ELSE
    FO='(3X,I2,38X,E16.10) '
ENDIF
```

c *** positionnement et lecture du fichier d'entree(.tmp)

```
REWIND(1)
5 READ(1,FO,END=30)MOIS,DONN
```

c *** regroupement des valeurs dans les saisons appropriees

```
MO=(MOIS+INTER-MD)/INTER
IF (MO.EQ.0)MO=NINT
K=K+1
VEC(MO)=VEC(MO)+DONN
VEC2(MO)=VEC2(MO)+DONN**2
N(MO)=N(MO)+1
TOT2=TOT2+DONN**2
TOT=TOT+DONN
GOTO 5
30 NOB=K
NINTA=NINT
DO 40 I=1,NINT
    IF (N(I).EQ.0) THEN
        NINTA=NINTA-1
        MOYC(I)=0
    ELSE
        MOYC(I)=VEC(I)/N(I)
        SSTR=SSTR+VEC(I)**2/N(I)
    ENDIF
40 CONTINUE
```

c *** calcul des statistiques pour la table ANOVA

```
SSTR=SSTR-TOT**2/NOB
NR=NINTA-1
MSTR=SSTR/NR
SST=TOT2-TOT**2/NOB
NT=NOB-1
MST=SST/NT
SSE=SST-SSTR
NE=NOB-NINTA
MSE=SSE/NE
F=MSTR/MSE
CALL FISCH(F,NR,NE,ISIGNIF)
ISAIS=ISIGNIF
IF (ISIGNIF.EQ.0) THEN
    TEXTSIGN='ON ACCEPTE L'EGALITE DES MOYENNES'
    TEXTB='PAS DE SAISONNALITE'
ELSE
    TEXTSIGN='ON REJETTE L'EGALITE DES MOYENNES'
```

```
TEXTB='PRESENCE DE SAISONNALITE
ENDIF
41 FORMAT (A8,I7,E15.5,E13.5,F9.3)
42 FORMAT (A8,I7,E15.5,E13.5)

c *** remise en mode texte

CALL @SMODE(ITMODE)
CALL @CLEAR(ITBACK,ITFORE)
CALL @BORD(ITCOLOR)

c *** impression du tableau des resultats

WRITE(*,*) '
IF (ITYP.EQ.0) THEN
    WRITE(*,*) '
    WRITE(*,*) ' TABLEAU 3.2b ANOVA POUR LES CONCENTRATIONS '
ELSE
    WRITE(*,*) '
    WRITE(*,*) ' TABLEAU 3.3b ANOVA POUR LES CHARGES
ENDIF
WRITE(*, '(/,/) ')
WRITE(*,*) '
WRITE(*,*) '
WRITE(*,*) ' TABLE ANOVA POUR L'EGALITE DES MOYENNES MENSUELLES '
WRITE(*,*) '
WRITE(*, '(/,/) ')
WRITE(*,*) ' Source d.l. es ms F
WRITE(*,*) '
WRITE(*,41) ' Mois ',NR,SSTR,MSTR,F
WRITE(*,42) ' Erreur ',NE,SSE,MSE
WRITE(*,*) '
WRITE(*,42) ' Total ',NT,SST,MST
WRITE(*,*) '
WRITE(*,*) '
WRITE(*,*) ' COMMENTAIRES: ',TEXTSIGN
WRITE(*, '(15X,A33)')TEXTB
CALL @CMOV(1,1)
CALL @CHAR(27,ITFORE,1)
CALL @BEEP
READ(*,70) TOUCHE
70 FORMAT (A1)

c *** dans le cas ou on a rejete l'egalite des moennes,produire
c *** le tableau des moyennes saisonnieres

IF (ISIGNIF.EQ.1) THEN
    CALL @CLEAR(ITBACK,ITFORE)
    CALL @CMOV(0,23)
    WRITE(*,*) '
    WRITE(*,*) '
    IF (ITYP.EQ.0) THEN
        WRITE(*,*) ' TABLEAU 3.2c MOYENNES SAISONNIERES '
        WRITE(*,*) ' DES DONNEES ORIGINALES '
    ELSE
```

```
WRITE(*,*) ' TABLEAU 3.3c MOYENNES SAISONNIERES'  
WRITE(*,*) ' DES DONNEES ORIGINALES'  
ENDIF  
WRITE(*,*) '  
WRITE(*,*) '  
FO='  
FO='(7X,F10.3)'  
DO 72 I=1,NINT  
IF (MOYC(I).GT.999999) FO='(7X,E16.10)'  
CONTINUE  
DO 100 JJ=1,NINT  
MFIN=MOD(MD+JJ*INTER,12)-1  
IF (MFIN.EQ.-1) MFIN=11  
IF (MFIN.EQ.0) MFIN=12  
IF (INTER.EQ.1) THEN  
WRITE(*, '(2X,A5,1X,I2,5X\)' ) 'MOIS',MD+(JJ-1)*INTER  
ELSE  
WRITE(*,84) 'MOIS',MD+(JJ-1)*INTER,' A ',MFIN  
ENDIF  
WRITE(*,FO)MOYC(JJ)  
FORMAT(2X,A5,1X,I2,A3,I2\)  
CONTINUE  
WRITE(*, '(/,/,/)' )  
ENDIF  
RETURN  
END
```

SUBROUTINE MOYMENS (VECT,MOYC,MD,NINT,ISIGNIF)

c
c Fonction :
c
c . compile la moyenne mensuelle des prelevements pour chacun
c des douze mois
c . trace le graphique de ces moyennes
c
c Parametres:
c
c . vect Vecteur contenant les moyennes mensuelles pour chaque
c annee
c . moyc Vecteur contenant les moyennes mensuelles pour
c l'ensemble des annees
c
c
c

```
INTEGER*2 AN,HIGH,ANDEF,ANFIN  
REAL TOT,MOYC(12)  
DIMENSION VECT(*)  
CHARACTER*1 TOUCHE  
CHARACTER*12 FICHDEB,NOMFICH  
CHARACTER*21 F  
COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR  
COMMON /IDENT/ NOMFICH,L6,FICHDEB,IDEB,ITYP,IMPRIM,ANDEF,
```

```

$      MODEP, JODEF, ANFIN, MOFIN, JOFIN, NOBS, NMINABER, NMAXABER

      YMIN=1E38
      YMAX=-9.0
      IF (ITYP.EQ.0) THEN
         F='(I2,1X,I2,4X,E16.10) '
      ELSE
         F='(I2,1X,I2,38X,E16.10) '
      ENDIF

c *** positionnement du fichier d'entree (.tmp) et lecture des donnees

      REWIND(1)
10     READ(1,F,END=40) AN,MO,DONNEE
      LOW=AN

c *** trouver l'indice relatif du mois dans le vecteur

      IND1=(AN-LOW)*12+MO
      TOT=DONNEE
      NBVAL=1
30     READ(1,F,END=40) AN,MO,DONNEE
      IND2=(AN-LOW)*12+MO
      IF (IND1.EQ.IND2) THEN
c *** compiler pour obtenir une valeur moyenne
         TOT=TOT+DONNEE
         NBVAL=NBVAL+1
         GOTO 30
      ELSE
c *** inscrire la moyenne obtenue dans le vecteur
         VECT(IND1)=TOT/NBVAL
         IF (VECT(IND1).EQ.0) GOTO 39
c *** retenir les minimum et maximum trouves
39     YMIN=MIN(YMIN,VECT(IND1))
         YMAX=MAX(YMAX,VECT(IND1))
         TOT=DONNEE
         NBVAL=1
         IND1=IND2
      ENDIF
      GOTO 30

c *** traitement du dernier prelevement

40     VECT(IND2)=TOT/NBVAL
      IF (VECT(IND2).EQ.0) GOTO 41
41     YMIN=MIN(YMIN,VECT(IND2))
      YMAX=MAX(YMAX,VECT(IND2))

c *** on memorise la derniere annee
```

HIGH=AN
NBAN=HIGH-LOW+1

c *** tracage du graphique des moyennes mensuelles

CALL ANOVA(VECT,MOYC,ISIGNIF,MD,NINT,NBAN)
CALL GRAF(VECT,YMIN,YMAX,MOYC,NBAN)
RETURN
END

#INCLU DE: 'TICS.FOR'
#INCLU DE: 'AJUSTE.FOR'
#INCLU DE: 'FISCHER.FOR'
#INCLU DE: 'TESTRHO.FOR'
#INCLU DE: 'CHIDEUX.FOR'
#INCLU DE: 'P3PLUS.FOR'

```
REGRESSION      P3PLUS.FOR      1  
SUBROUTINE REGRESSION(X,Y,XCHAP,DEBI,VEC
```

```
—TEXTRE,ALOG,MAX,MIN,MSE,QSMODE,QCOLOR  
—QOVSCN,TICS,AJUSTE,QPLOT,QXAXIS,QYAXIS  
—QSETUP,QGTXT,QPTXTA,QPTXTC,QTABL,QCMOV  
—QCHAR,QBEEP,EXP,ADR,SQRT,TESTRHO,QCLEAR  
—QBORD
```

```
GRAF            P3PLUS.FOR     233  
SUBROUTINE GRAF(VECTEUR,YMIN,YMAX,MOYC,N
```

```
—TEXTMM,QSMODE,QCOLOR,QOVSCN,TICS,AJUSTE  
—QPLOT,QXAXIS,QYAXIS,QGTXT,QPTXTA,MOD  
—QSETUP,QTABL,QPTXTC,QCMOV,QCHAR,QBEEP
```


SUBROUTINE REGRESSION(X,Y,XCHAP,DEBI,VEC)

C * * * * *

C -----
C Fonction:

- C . Recherche d'une relation concentration-debit significative
- C par la technique de regression
- C . Trace le graphique des logs des concentrations en fonction
- C des logs des debits

C Parametres:

- C . x Vecteur destine a contenir les logs des debits
- C . y Vecteur destine a contenir les logs des concentrations
- C . xchap Vecteur destine a contenir les valeurs predites par
- C le modele
- C . debi Vecteur destine a contenir les debits
- C . vec Vecteur destine a recevoir les concentrations
- C . ideb Indique si les debits sont disponibles ou non
- C (valeur en provenance de "ident.tmp")

C -----
COMMON /BLOC31/ NOBSERV,NBAN
INTEGER*2 NOBSERV,ANDEP,ANFIN
REAL DEB,XCHAP(NOBSERV),MSE
REAL XMIN,XMAX,YMIN,YMAX,X(*),Y(*),DEBI(*),VEC(*),MOY,TOTD
CHARACTER*12 FICHDEB,NOMFICH
CHARACTER*10 TEXTEX,TEXTEY
CHARACTER*26 COMMENT
CHARACTER*28 F
CHARACTER*41 TEXTRE
COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH/ IGMODE,IGBACK,IFAL,IGBORD,IGCOLOR,ILINE,
* ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2
COMMON /IDENT/ NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEP,
* MODEP,JODEP,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER

REWIND(1)
IF (IDEB.EQ.3) THEN
c *** on ne peut pas effectuer la regression sans debit
GOTO 90
ENDIF
I=0
TEXTRE(1:33)='FIGURE 3.5 : Ln C VS Ln D,
TEXTRE(30:41)=NOMFICH
TEXTEX='
TEXTEY='
XMIN=1E38
YMIN=1E38
XMAX=0
YMAX=0

```
TOTD=0
10 I=I+1

c *** lecture des concentrations et des debits

15 READ (1,7,END=30)C,DEB
7 FORMAT (9X,E16.10,1X,E16.10)
IF (C.EQ.0..OR.DEB.EQ.0.) GOTO 25

c *** transformations en logs naturels
c *** memorisation du minimum et du maximum

Y(I)=ALOG(C)
VEC(I)=C
YMAX=MAX(YMAX,Y(I))
YMIN=MIN(YMIN,Y(I))
TOTD=TOTD+C
X(I)=ALOG(DEB)
DEBI(I)=DEB
XMAX=MAX(XMAX,X(I))
XMIN=MIN(XMIN,X(I))
GOTO 10
25 I=I-1
GOTO 10
30 NOBS=I-1
MOY=TOTD/NOBS
SOMPROD=0
SOMXI=0
SOMYI=0
SOMXI2=0
SOMYI2=0
99 CONTINUE

c *** calcul des moyennes et ecart-s-type associes

DO 50 I=1,NOBS
SOMPROD=SOMPROD+X(I)*Y(I)
SOMXI=SOMXI+X(I)
SOMYI=SOMYI+Y(I)
SOMXI2=SOMXI2+X(I)**2
SOMYI2=SOMYI2+Y(I)**2
50 CONTINUE
XBAR=SOMXI/NOBS
YBAR=SOMYI/NOBS
VARX=SOMXI2/NOBS-(XBAR**2)
VARY=SOMYI2/NOBS-(YBAR**2)

c *** calcul des estimateurs "a chapeau" et "b chapeau"

BCHAF=(SOMPROD-(XBAR*YBAR*NOBS))/(NOBS*VARX)
ACHAF=YBAR-BCHAF*XBAR
SSE=0
SSTR=0
DO 60 I=1,NOBS
YCHAPI=ACHAF+BCHAF*X(I)
```

```
        SSE=SSE+(YCHAPI-Y(I))**2  
        SSTR=SSTR+(YCHAPI-YBAR)**2  
60 CONTINUE  
        MSE=SSE/(NOBS-2)  
        VARB=MSE/(VARX*NOBS)  
        BSTAN=BCHAP/VARB**0.5  
        VARA=MSE*(1.0/NOBS+XBAR**2/(VARX*NOBS))  
        ASTAN=ACHAP/VARA**0.5
```

c *** mise en mode graphique

```
        CALL QSMODE(IGMODE)  
        IF(IGMODE.LT.10) THEN  
            CALL QCOLOR(IGBACK,IPAL)  
        ELSE  
            CALL QOVSCN(IGBORD)  
        ENDIF
```

c *** ajustement de l'espacement des "tics"

```
        CALL TICS(XMIN,XMAX,XSTS,XFINS,TICX,MINORX,AINTEX)  
        CALL TICS(YMIN,YMAX,YSTS,YFINS,TICY,MINDRY,AINTEY)  
        IF (AINTEX.GE.10.OR.AINTEY.LT.1) THEN  
            CALL AJUSTE(X,NOBS,AINTEX,TEXTEX,XSTS,XFINS,TICX)  
        ENDIF  
        IF (AINTEY.GE.10.OR.AINTEY.LT.1) THEN  
            CALL AJUSTE(Y,NOBS,AINTEY,TEXTEY,YSTS,YFINS,TICY)  
        ENDIF
```

c *** tracage de la courbe

```
        CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,XSTS,XFINS,YSTS,YFINS,  
# XSTS,YSTS,0,1.0,1.5)  
        CALL QXAXIS(XSTS,XFINS,TICX,MINORX,1,0)  
        CALL QYAXIS(YSTS,YFINS,TICY,MINDRY,1,0)  
        CALL QSETUP(0,ILINE,ISYMBL,KIRSYM)  
        CALL QGTXT(10,TEXTEX,ILETR,JCOL2-35,JROW1+10,0)  
        CALL QGTXT(41,TEXTRE,ILETR,JCOL1+60,JROW2+12,0)  
        IF (IGMODE.EQ.16) THEN  
            INCR=85  
        ELSE  
            INCR=55  
        ENDIF  
        CALL QGTXT(15,' CONCEN mg/l ',ILETR,0,(JROW2+JROW1)/2+incr,-1)  
        CALL QPTXTA(15,' DEBIT m3/sec ',ILETR)  
        CALL QPTXTC(10,TEXTEY,ILETR)  
        CALL QTABL(0,NOBS,X,Y)  
        IF(IMPRIM.EQ.1)CALL QPSCRN  
        CALL QCMOV(1,1)  
        CALL QCHAR(27,ITFORE,1)  
        CALL QBEEP  
        READ(*,70) TOUCHE  
70 FORMAT(A1)
```

c *** calcul du pourcentage de variance expliquée dans l'échelle

c *** originale (et non dans les log).

```
BOR=BCHAP
AOR=EXP(ACHAP)
VAR1=0
VAR2=0
DO 72 I=1,NOBS
    XCHAP(I)=AOR*(DEBI(I)**BOR)
    VAR1=VAR1+(MOY-XCHAP(I))**2
    VAR2=VAR2+(VEC(I)-MOY)**2
72 CONTINUE
VAREXP=VAR1*100/VAR2
```

c *** verification de la signification de la relation

```
RHOCC=SQRT(VAREXP/100)
NDF=NOBS-2
CALL TESTRHO(NDF,RHOCC,1,ISIGN)
75 FORMAT(I1,1X,F9.2,1X,F9.2)
```

c *** ecriture des estimateurs dans "ident.tmp"

```
IF(ISIGN.EQ.1) THEN
    ICQ=1
    WRITE(3,75) ICQ,AOR,BOR
    COMMENT='RELATION SIGNIFICATIVE'
ELSE
    WRITE(3,75) 0,0.,0.
    COMMENT='RELATION NON-SIGNIFICATIVE'
ENDIF
76 FORMAT(A37,1X,F9.2)
77 FORMAT(17X,A2,F9.2,10X,A2,F9.2)
```

c *** remise en mode texte

```
CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
```

c *** affichage du tableau des resultats

```
WRITE(*,*) '
WRITE(*,*) '
WRITE(*,*) '
WRITE(*,*) '
WRITE(*, '(/,/,/)' )
WRITE(*,76) '
WRITE(*,76) '
WRITE(*,76) '
WRITE(*,76) '
WRITE(*, '(/,/,/)' )
WRITE(*, '(5X,A39)' )
WRITE(*,*) '
WRITE(*,77) 'a=',ADR,'b=',BOR
WRITE(*,*) '
-----
TABLEAU 3.4 RESULTATS DE LA REGRESSION
-----
MOYENNE DES Ln(CONC) : ',YBAR
VARIANCE : ',VARY
MOYENNE DES Ln(DEB) : ',XBAR
VARIANCE : ',VARX
RELATION C=a Q**b'
```

```
WRITE(*,'(17X,A25,F6.2)') '% DE VARIANCE EXPLIQUEE: ',VAREXP  
WRITE(*,*) ' '  
WRITE(*,'(13X,A14,1X,A26)') 'COMMENTAIRES : ',COMMENT  
CALL QCMOV(1,1)  
CALL QCHAR(27,ITFORE,1)  
CALL QBEEP  
READ(*,80)TOUCHE  
80  FORMAT(A1)  
90  RETURN  
END
```

SUBROUTINE GRAF(VECTEUR,YMIN,YMAX,MOYC,NBAN)

```
C * * * * *  
C -----  
C Fonction :  
C . trace le graphique des moyennes mensuelles (on retrouvera les  
C mois de l'annee en abscisse)  
C Parametres :  
C . vecteur Vecteur contenant la serie des moyennes men-  
C mensuelles pour chaque annee  
C . ymin Minimum des donnees (concentration ou charge)  
C . ymax Maximum des donnees (concentration ou charge)  
C . moyc Vecteur contenant les moyennes mensuelles pour  
C l'ensemble des annees  
C . nban Nombre d'annee sur lequel s'etend l'analyse  
C -----
```

```
INTEGER*2 SYMBOLE,ANDEP,ANFIN  
REAL DONN(12),TEMPS(12),MOYC(12)  
DIMENSION VECTEUR(*)  
CHARACTER*1 TOUCHE  
CHARACTER*15 TEXTY  
CHARACTER*12 FICHDEB,NOMFICH  
CHARACTER*10 TEXTE  
CHARACTER*65 TEXTMM  
COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR  
COMMON /GRAPH/ IGMODE,IGBACK,IFAL,IGBORD,IGCOLOR,ILINE,  
* ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2  
COMMON /IDENT/ NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,ANDEP,  
* MODEP,JODEP,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER  
DATA TEMPS/1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0/
```

```
C *** determination des titres et formats,en fonction du parametre  
C *** de qualite choisi  
  
TEXTE=' '  
TEXTMM=' '
```

```
IF (ITYP.EQ.0) THEN
    TEXTMM(1:37)='FIGURE 3.3: MOY. MENSUELLES DES CONCE'
    TEXTMM(38:53)='NTRATIONS,'
    TEXTMM(50:61)=NOMFICH
    TEXTY=' CONCEN mg/l '
ELSE
    TEXTMM(1:37)='FIGURE 3.4: MOY. MENSUELLES DES CHARG'
    TEXTMM(38:53)='ES , '
    TEXTMM(50:61)=NOMFICH
    TEXTY=' CHARGE Kg/jour '
ENDIF
```

c *** mise en mode graphique

```
CALL @SMODE(IGMODE)
IF (IGMODE.LT.10) THEN
    CALL @COLOR(IGBACK,IPAL)
ELSE
    CALL @OVSCN(IGBORD)
ENDIF
```

c *** ajustement de l'espacement des "tics"

```
CALL TICS(YMIN,YMAX,YSTS,YFINS,TIC,MINOR,AINTE)
IF (AINTE.GE.10.OR.AINTE.LT.1) THEN
    CALL AJUSTE(VECTEUR,NBAN*12,AINTE,TEXTE,YSTS,YFINS,TIC)
ENDIF
```

c *** tracage du graphique

```
CALL @PLOT(JCOL1,JCOL2,JROW1,JROW2,0.0,12.0,YSTS,YFINS,0.0,
* YSTS,0,1.0,1.5)
CALL @XAXIS(0.0,12.0,1.0,0,1,0)
CALL @YAXIS(YSTS,YFINS,TIC,MINOR,1,0)
CALL @GTXT(61,TEXTMM,ILETR,JCOL1+45,JROW2+12,0)
IF (IGMODE.EQ.16) THEN
    INCR=85
ELSE
    INCR=55
ENDIF
CALL @GTXT(15,TEXTY,ILETR,0,(JROW2+JROW1)/2+INCR,-1)
CALL @PTXTA(4,'MOIS',ILETR)
ISYMB=MOD(ANDEP,10)+47
SYMBOLE=0
IF (AINTE.GE.10.OR.AINTE.LT.1) THEN
    DO 41 I=1,12
        MOYC(I)=MOYC(I)/AINTE
    CONTINUE
ENDIF
DO 45 I=1,NBAN
    DO 43 J=1,12
        M=(I-1)*12+J
        DONN(J)=VECTEUR(M)
        IF (DONN(J).EQ.0) THEN
            IF (MOYC(J).EQ.0) MOYC(J)=YSTS
```

41

```

                                DONN(J)=MOYC(J)
                                ENDIF
43  CONTINUE
                                SYMBOLE=SYMBOLE+1
c *** changement de la couleur pour ne pas avoir un symbole
c *** confondu avec le fond de l'ecran
                                IF (SYMBOLE.EQ.IGBACK) SYMBOLE=SYMBOLE+1
                                ISYMB=ISYMB+1
                                IF (ISYMB.GT.57) ISYMB=48
                                CALL QSETUP(0,ILINE,ISYMB,SYMBOLE)
                                CALL QTABL(0,12,TEMPS,DONN)
                                CALL QPTXTC(10,TEXTE,ILETR)
45  CONTINUE
                                CALL QSETUP(0,ILINE,42,15)
                                CALL QTABL(1,12,TEMPS,MOYC)
                                CALL QCMOV(1,1)
                                IF (AINTE.GE.10.OR.AINTE.LT.1) THEN
46  DO 46 I=1,12
                                        MOYC(I)=MOYC(I)*AINTE
                                        CONTINUE
                                ENDIF
                                IF (IMPRIM.EQ.1) CALL QPSCRN
                                CALL QCHAR(27,ITFORE,1)
                                CALL QBEEP
                                READ(*,70) TOUCHE
70  FORMAT(A1)
                                RETURN
                                END
```

ELSEIF

TESTRHO
TESTRHO.FOR 1
SUBROUTINE TESTRHO(NDF,RHO,ISEUIL,ISIGN)

THU 09-10-87 15:24:26
06-04-87 12:50:48 TESTRHO.FOR


```
      SUBROUTINE TESTRHO(NDF,RHO,ISEUIL,ISIGN)
      OPEN(15,FILE='TESTRHO.TAB',STATUS='OLD')
      ISIGN=0
10     READ(15,'(I3,2F5.3)',END=30)NDF1,VC5,VC1
      IF(NDF1.LT.NDF)GOTO 10
      IF(ISEUIL.EQ.5)THEN
         IF(RHO.GT.VC5)ISIGN=1
      ELSEIF(ISEUIL.EQ.1)THEN
         IF(RHO.GT.VC1)ISIGN=1
      ENDIF
      GOTO 99
30     IF(ISEUIL.EQ.5)THEN
         IF(RHO.GT.VC5)ISIGN=1
      ELSEIF(ISEUIL.EQ.1)THEN
         IF(RHO.GT.VC1)ISIGN=1
      ENDIF
99     CLOSE(15)
      RETURN
      END
```

Thu 09-10-87 15:04:51
06-04-87 12:45:38 FISCHER.FOR

Pg 1

```
FISCH          FISCHER.FOR      1  
SUBROUTINE FISCH(Z,NDL1,NDL2,ISIGNIF)
```

```
      SUBROUTINE FISCH(Z,NDL1,NDL2,ISIGNIF)
      REAL A(2,11),F1000(11)
      INTEGER DL2D(2)
      DATA F1000/3.84,2.99,2.60,2.37,2.21,2.10,2.01,1.94,1.88,1.83,1.79/
      ISIGNIF=0
      N1=NDL1
      IF (NDL2.GT.1000) THEN
         IF (Z.GT.F1000(N1)) ISIGNIF=1
         GOTO 99
      ENDIF
      OPEN(9,FILE='FISCHER.TAB',STATUS='OLD')
14     FORMAT(I4,F6.2,10F7.2)
15     READ(9,14)DL2D(2),(A(2,I),I=1,11)
      IF (DL2D(2).EQ.NDL2) THEN
         IF (Z.GT.A(2,N1)) ISIGNIF=1
         GOTO 99
      ENDIF
      IF (DL2D(2).GT.NDL2) THEN
         ETEN=DL2D(2)-DL2D(1)
         F=A(2,N1)*(NDL2-DL2D(1))/ETEN+A(1,N1)*(DL2D(2)-NDL2)/ETEN
         IF (Z.GT.F) ISIGNIF=1
         GOTO 99
      ELSE
         DL2D(1)=DL2D(2)
         DO 20 I=1,11
            A(1,I)=A(2,I)
         GOTO 15
      ENDIF
      CLOSE(9)
99     RETURN
      END
```

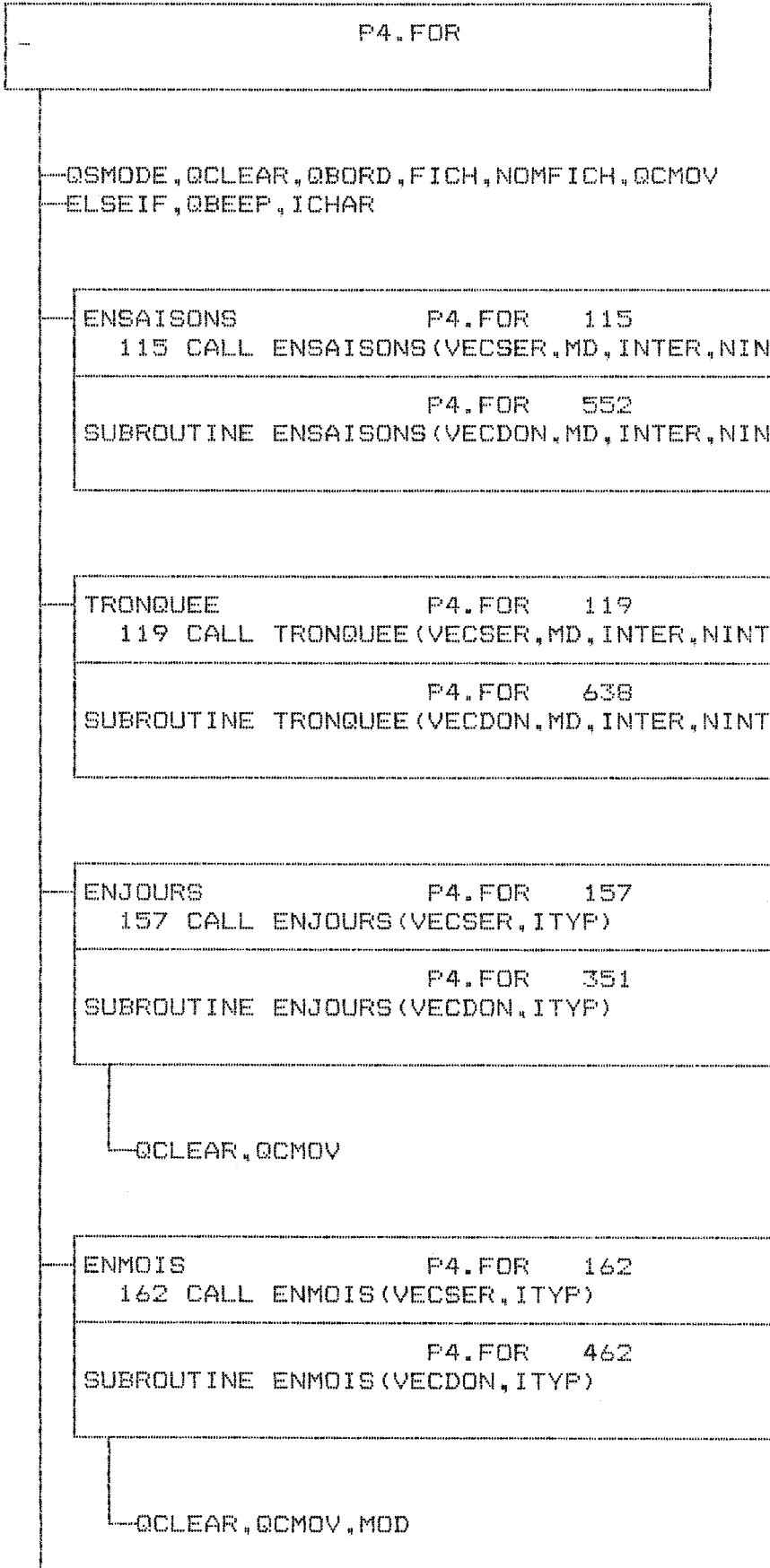
Thu 09-10-87 15:04:31
06-04-87 12:49:52 CHIDEUX.FOR

Pg 1

```
CHI2          CHIDEUX.FOR      1  
SUBROUTINE CHI2(Z,IDL,ISIGNIF)
```

└─FLOAT

```
      SUBROUTINE CHI2(Z,IDL,ISIGNIF)
      REAL DLC(2),C2(2)
      DL=FLOAT(IDL)
      ISIGNIF=0
      OPEN(99,FILE='CHIDEUX.TAB',STATUS='OLD')
15     READ(99,*)DLC(2),C2(2)
      IF(DL.EQ.DLC(2)) THEN
         IF(Z.GT.C2(2)) ISIGNIF=1
         GOTO 99
      ENDIF
      IF(DLC(2).GT.DL) THEN
         ETEN=DLC(2)-DLC(1)
         C=C2(2)*(DL-DLC(1))/ETEN+C2(1)*(DLC(2)-DL)/ETEN
         IF(Z.GT.C) ISIGNIF=1
         GOTO 99
      ELSE
         DLC(1)=DLC(2)
         C2(1)=C2(2)
         GOTO 15
      ENDIF
      CLOSE(99)
99     RETURN
      END
```



AFFICHE P4.FOR 171
171 CALL AFFICHE(VECSER,VALVIDE,COMPLE

P4.FOR 742
SUBROUTINE AFFICHE(VECSER,VALVIDE,COMPLE

MOD,OCLEAR,OCMOV,QCHAR,QBEEP

TROUVEX P4.FOR 773
773 CALL TROUVEX(VECSER,VALVIDE,N)

P4.FOR 1156
SUBROUTINE TROUVEX(VECSER,VALVIDE,N)

OTANN P4.FOR 213
213 CALL OTANN(VECSER)

P4.FOR 853
SUBROUTINE OTANN(VECSER)

METDMOY P4.FOR 220 x2
220 CALL METDMOY(VECSER,VALVIDE,TRANSF
298 CALL METDMOY(VECSER,VALVIDE,TRANSF

P4.FOR 883
SUBROUTINE METDMOY(VECSER,VALVIDE,TRANSF

MOD,OCMOV,QCHAR,QBEEP

INTERPOL P4.FOR 955 x2
955 CALL INTERPOL(MOYINT,TAMP,VALVIDE,
995 CALL INTERPOL(VECSER,TRANSF,VALVID

P4.FOR 1014
SUBROUTINE INTERPOL(VECSER,TRANSF,VALVID

```
TROUVEX          P4.FOR  1042
1042 CALL TROUVEX(VECSER,VALVIDE,LONG)
```

```
          P4.FOR  1156
SUBROUTINE TROUVEX(VECSER,VALVIDE,N)
```

```
TROUVEX          P4.FOR   975
975 CALL TROUVEX(VECSER,VALVIDE,N)
```

```
          P4.FOR  1156
SUBROUTINE TROUVEX(VECSER,VALVIDE,N)
```

```
TROUVEX          P4.FOR  233 x2
233 CALL TROUVEX(VECSER,VALVIDE,N)
310 CALL TROUVEX(VECSER,VALVIDE,N)
```

```
          P4.FOR  1156
SUBROUTINE TROUVEX(VECSER,VALVIDE,N)
```

```
INTERPOL        P4.FOR  301   p2
301 CALL INTERPOL(VECSER,TRANSF,VALVIDE)
```

```
CONCDEB         P4.FOR  304
304 CALL CONCDEB(VECSER,VALVIDE,ITYP)
```

```
          P4.FOR  1077
SUBROUTINE CONCDEB(VECSER,VALVIDE,ITYP)
```

```
POSITIONNE      P4.FOR  1118
1118 CALL POSITIONNE
```

```
          P4.FOR  1288
SUBROUTINE POSITIONNE
```


TROUVEX F4.FOR 1121
1121 CALL TROUVEX(VECSER,VALVIDE,N)

 F4.FOR 1156
SUBROUTINE TROUVEX(VECSER,VALVIDE,N)

TRANSDATE F4.FOR 1135
1135 CALL TRANSDATE(IDIV,IV,IBIN,LOW,NI

 TDATE.FOR 6
SUBROUTINE TRANSDATE(IDIV,IND,IBIN,LOW,N

ELSEIF

DATNORM TDATE.FOR 42
42 CALL DATNORM(IND,LOW,IBIN,NINTAN,I

 TDATE.FOR 54
SUBROUTINE DATNORM(IND,LOW,IBIN,NINTAN,I

MOD

DATSAIS TDATE.FOR 45
45 CALL DATSAIS(IND,LOW,MD,INTER,NINT

 TDATE.FOR 109
SUBROUTINE DATSAIS(IND,LOW,MD,INTER,NINT

MOD

DATTRONQ TDATE.FOR 48
48 CALL DATTRONQ(IND,LOW,MD,NINT,AN,M

 TDATE.FOR 154
SUBROUTINE DATTRONQ(IND,LOW,MD,NINT,AN,M

MOD

TROUVDEB	P4.FOR	1139
1139 CALL TROUVDEB(AN,MO,JR,ANDEB,MODEB		

	P4.FOR	1206
SUBROUTINE TROUVDEB(AN,MO,JR,ANDEB,MODEB		

AUTOCOR	P4.FOR	327
327 CALL AUTOCOR(VECSER,TAMP,VALVIDE,I		

	AUTO.FOR	6
SUBROUTINE AUTOCOR(VECSER,TAMP,VALVIDE,I		

QCMOV,QCHAR,QBEEP,ABS,ELSEIF,QCLEAR

AUTO	AUTO.FOR	59
59 CALL AUTO(TAMP,LS,7,R)		

	AUTO.FOR	123
Subroutine auto(a,n,l,r)		

#STORA GE:2

PROGRAM PARTIE4

```
c -----
c *****
c Titre: Serie de travail,creation et structuration
c Fonctions :
c . Permet a l'usager de choisir un intervalle de travail
c . Remplacement des valeurs absentes selon une des 3 methodes :
c   moyennes saisonnieres, interpolation, relation concentration-debit
c . Etude de persistance de la serie
c . Creation d'un nouveau fichier de travail contenant la serie
c   maintenant devenue equidistante
c
c Notation adoptee dans les sous-routines :
c
c Voici le sens des symboles places a la fin de la description de chaque
c parametre :
c
c (E) : Il s'agit d'un parametre d'entree
c (S) : Il s'agit d'un parametre de sortie
c (E/S) : Il s'agit d'un parametre necessaire en entree ET egalement
c         modifie en sortie.
c # : Il s'agit d'une structure de donnee qui n'est utilisee qu'a
c     l'interieur de cette sous-routine. Elle n'est passee en para-
c     metre que parce qu'elle est declaree dans le programme prin-
c     cipal. Cette facon de faire permet de pouvoir reutiliser la
c     structure a diverses fins dans divers sous-programmes. (per-
c     met une economie d'espace )
c
c *****
c
c INTEGER MAXI
c REAL VALVIDE
c PARAMETER (MAXI=1500)
c PARAMETER (VALVIDE=-9)
c CHARACTER*12 FICHDEB,FICH,NOMFICH
c CHARACTER*1 REP,CHOIX,METD
c INTEGER AN,LOW,INTERV,N,NINTAN,ANFIN,LS
c REAL VECSEB(MAXI),TRANSF(MAXI),TAMP(MAXI)
c LOGICAL COMPLETE,OKINTERP,TRONCAN,RETOUR
c COMMON /BLOC41/LOW,INTERV,N,NINTAN
c COMMON /BLOC42/ACQ,BCQ,CHOIX,FICHDEB
c COMMON /BLOC43/IPERS,RHO
c COMMON /BLOC44/ IEX1,IEX2,LS
c COMMON /BLOC45/ IDIV,IBIN,MD,INTER,NINT
c COMMON /BLOC46/ITBACK,ITFORE
c COMMON /BLOC47/ ITRONC,IANTR,IDURTR
c
c *** lecture des specifications graphiques
```

```
OPEN(3,FILE='SETUP.PC',STATUS='OLD')  
READ(3,'(4I2)')ITMODE,ITFORE,ITBACK,ITCOLOR  
CALL QSMODE(ITMODE)  
CALL QCLEAR(ITBACK,ITFORE)  
CALL QBORD(ITCOLOR)
```

c *** lecture du fichier "ident.tmp"

```
OPEN(4,FILE='IDENT.TMP',STATUS='OLD')  
READ(4,2)NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,LOW,MODEP,JOSEP,  
# ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER,ISAIS,MD,INTER,NINT,  
# ICQ,ACQ,BCQ,ITRONC,IANTR,IDURTR  
2 FORMAT(A12,1X,I1,/,A12,1X,I1/,I1,1X,I1,/,I2,1X,I2,1X,I2,/,I2,1X,  
# I2,1X,I2,/,I4,/,I2,1X,I2,/,I1,1X,I2,1X,I2,1X,I2,/,I1,1X,F9.2,1X,  
# F9.2/,I1,1X,I2,1X,I2)
```

```
FICH=''  
FICH(1:LG)=NOMFICH(1:LG)  
FICH(LG+1:LG+4)='.TMP'  
OPEN(1,FILE=FICH,STATUS='OLD')
```

```
11 CALL QCLEAR(ITBACK,ITFORE)  
CALL QCMOV(0,23)
```

c *** initialisation des vecteurs

```
DO 10 I=1,MAXI  
    VECSE(I)=VALVIDE  
10 CONTINUE
```

```
12 WRITE(*,*)'
```

c *** si on a detecte une saisonnalite precedemment (p3), on offre a
c *** l'usager la possibilite d'utiliser la division saisonniere deja
c *** specifiee

```
IF (ISAIS.NE.0) THEN  
    WRITE(*,*)' LA DIVISION SUIVANTE DE LA SERIE A ETE SUGGEREE'  
    WRITE(*,*)' PRECEDEMMENT :'  
    WRITE(*,*)'  
    WRITE(*,'(A23,I2)')' MOIS DE DEPART : ',MD  
    WRITE(*,'(A38,I2)')' NOMBRE DE MOIS PAR INTERVALLE : ',INTEP  
    WRITE(*,'(A39,I2)')' NOMBRE D'INTERVALLES PAR ANNEE : ',  
# NINT
```

c *** valeur bidon de depart

c *** "idiv" refletera le choix de division effectuee par l'usager

```
REP='X'  
WRITE(*,*)'  
WRITE(*,*)'  
WRITE(*,*)' SI VOUS REPONDEZ NON A LA PROCHAINE QUESTION,  
WRITE(*,*)' UN CHOIX D"INTERVALLES NON-SAISONNIERS VOUS  
WRITE(*,*)' SERA OFFERT  
WRITE(*,*)'  
WRITE(*,*)' DESIREZ-VOUS DIVISER VOTRE SERIE DE CETTE'  
WRITE(*,'(A20\)'')' MANIERE (O/N) ? '  
READ(*,'(A1)') REP  
IF (REP.EQ.'O'.OR.REP.EQ.'o') THEN
```

```

      IBIN=1
      IF (ISAIS.EQ.1) THEN
c *** diviser la serie a l'aide des saisons donnees
          IDIV=1
          CALL ENSAISONS(VECSER,MD,INTER,NINT,ITYP)
      ELSE
c *** diviser la serie en reprenant la troncation suggeree dans p3
          IDIV=2
          CALL TRONQUEE(VECSER,MD,INTER,NINT,ITYP)
      ENDIF
      CHOIX='2'
c *** l'usager devra choisir son propre intervalle de separation
      ELSEIF (REF.EQ.'N'.OR.REF.EQ.'n') THEN
          IDIV=0
      ELSE
          WRITE(*,45)
          CALL QBEEP
          GOTO 12
      ENDIF
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QCMOV(0,23)
  ENDIF
  IF (ISAIS.EQ.0.OR.REF.EQ.'N'.OR.REF.EQ.'n') THEN
40      IDIV=0
          WRITE(*,*) ' '
          CALL QCLEAR(ITBACK,ITFORE)
          CALL QCMOV(0,23)
c *** si on n'a ni choisi la division saisonniere,ni la troncation,
c *** on divise alors en specifiant l'intervalle
          WRITE(*,*) ' CHOISISSEZ L'UNITE POUR DIVISER LA SERIE : '
          WRITE(*,*) ' '
          WRITE(*,*) ' 1) DIVISION EN JOURS '
          WRITE(*,*) ' 2) DIVISION EN MOIS '
          WRITE(*,*) ' '
          WRITE(*, '(15X,A30,\)') '          VOTRE CHOIX: '
          READ(*, '(A1)') CHOIX
          IF (CHOIX.NE.'1'.AND.CHOIX.NE.'2') THEN
              WRITE(*,45)
              CALL QBEEP
              GOTO 40
          ENDIF
          IF (CHOIX.EQ.'1') THEN
c *** diviser la serie a l'aide d'un intervalle donne en jours
              IBIN=0
              CALL ENJOURS(VECSER,ITYP)
              GOTO 50
          ELSE
c *** diviser la serie a l'aide d'un intervalle donne en mois
              IBIN=1
              CALL ENMOIS(VECSER,ITYP)
              GOTO 50
          ENDIF
45      FORMAT(/, ' *** REPONSE INVALIDE.RECOMMENCEZ. ***')
```

```

      ENDIF
50    CALL QDCLEAR(ITBACK,ITFORE)
      CALL QCMDV(0,23)

c *** afficher le nombre de valeurs absentes par intervalle par annee
      CALL AFFICHE(VECSER,VALVIDE,COMPLETE,RETOUR)
      IF (RETOUR.EQV. .TRUE.) GOTO 11

c *** si on a tronque certaines annees au cours de p3,demander ce
c *** que l'usager entend faire avec ces annees

52    IF (ITRONC.NE.0.AND.ITRONC.NE.9) THEN
          CALL QDCLEAR(ITBACK,ITFORE)
          CALL QCMDV(0,23)
c *** recrire a l'ecran les annees deja identifiees dans p3
          WRITE(*,*) ' '
          IF (IDURTR.NE.1) THEN
              WRITE(*,56)'LES ANNEES ',IANTR,' A ',IANTR+IDURTR-1,
              ' ONT ETE IDENTIFIEES COMME CONTENANT PEU DE DONNEES'
          ELSE
              WRITE(*,57)'L'ANNEE ',IANTR,' A ETE IDENTIFIEE COMME
              CONTEN
              ANT PEU DE DONNEES'
          ENDIF
56      FORMAT(A12,1X,I2,A3,I2,A51)
57      FORMAT(A9,I2,A48)

c *** faire choisir une des actions suivantes :
          WRITE(*,*)
          WRITE(*,*) ' DESIREZ-VOUS: '
          WRITE(*,*) ' '
          WRITE(*,*) ' 1) ELIMINER CES ANNEES '
          WRITE(*,*) ' '
          WRITE(*,*) ' 2) REMPLACER LES DONNEES DE CES ANNEES PAR LEUR '
          WRITE(*,*) ' MOYENNE SAISONNIERE '
          WRITE(*,*) ' '
          WRITE(*,*) ' 3) CONSERVER CES ANNEES SANS CHANGEMENT '
          WRITE(*,*) ' '
          WRITE(*,'(A17\)' ) ' VOTRE CHOIX : '
          READ(*,'(A1)')REP
          IF (REP.NE.'1'.AND.REP.NE.'2'.AND.REP.NE.'3') THEN
              CALL QBEEP
              GOTO 52
          ENDIF
          IF (REP.EQ.'1') THEN

c *** retirer les donnees correspondant aux annees a tronquer
c *** reajuster la valeur de N
              CALL OTANN(VECSER)
              ITRONC=1
          ELSEIF (REP.EQ.'2') THEN

c *** replacer les valeurs sur ces annees par leur moyenne saisonniere
c *** respectivement.Laisser le reste du vecteur inchange
              ITRONC=2
              TRONCAN=.TRUE.
          ENDIF
      ENDIF
  
```

```
                CALL METDMOY(VECSER,VALVIDE,TRANSF,TRONCAN)
            ELSE
c *** on met l'indicateur de troncacion a trois;l'usager ne veut
c *** finalement pas traiter ces annees
                ITRONC=3
            ENDIF

C *** on reverifie si la serie est complete ou non
                NCOMP=0
                DO 54 I=1,N
                    IF (VECSER(I).NE.VALVIDE) NCOMP=NCOMP+1
54                CONTINUE
C *** on determine les indices des extremités non-absentes du vecteur
                CALL TROUVEX(VECSER,VALVIDE,N)
                NBEX=IEX2-IEX1+1
                IF (NCOMP.EQ.NBEX) COMPLETE=.TRUE.

c *** ecriture des nouveaux parametres (corriges) dans ident.tmp
                REWIND(4)
                WRITE(4,2)NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,LOW,MODEP,JODEP,
                ANFIN,MDFIN,JOFIN,NOBS,NMINABER,NMAXABER,ISAIS,MD,INTER,NINT,
                ICQ,ACQ,BCQ,ITRONC,IANTR,IDURTR
            ELSE
c *** on remet l'indicateur de troncacion a zero;l'usager veut
c *** finalement conserver ces annees
                ITRONC=0
            ENDIF

53          IF (COMPLETE.EQV..FALSE.) THEN

c *** remplacement des valeurs manquantes (si la serie n'est pas
c *** complete )

55                CALL QCLEAR(ITBACK,ITFORE)
                CALL QCMOV(0,23)
21                WRITE(*,*)'
                WRITE(*,*)'          QUELLE METHODE DESIREZ-VOUS EMPLOYER POUR'
                WRITE(*,*)'          REMPLACER LES VALEURS MANQUANTES?'
                WRITE(*,*)'
                WRITE(*,*)'          1) REMPLACEMENT PAR LA MOYENNE POUR'
                WRITE(*,*)'          L'INTERVALLE SUR LES ANNEES'
                WRITE(*,*)'
                WRITE(*,*)'          2) INTERPOLATION'
                WRITE(*,*)'
                WRITE(*,*)'          3) RELATION CONCENTRATION/DEBIT'
                WRITE(*,*)'
                WRITE(*,*(5X,A15\))'VOTRE CHOIX : '
                READ(*,*(A1))METD
                IF (METD.NE.'1'.AND.METD.NE.'2'.AND.METD.NE.'3') THEN
                    WRITE(*,45)
                    CALL QBEEP
                    GOTO 55
                ENDIF
                IF (METD.EQ.'3') THEN
                    IF (IDEB.EQ.3) THEN
```

```
WRITE(*,*)'  
WRITE(*,*)' *** DEBITS NON DISPONIBLES ***'  
WRITE(*,*)' *** CHOISISSEZ UNE AUTRE OPTION ***'  
GOTO 21  
ELSEIF (ICQ.EQ.0) THEN  
c *** la relation concentration-debit de p3 n'a pas ete significative  
c *** on ne peut donc pas utiliser cette methode  
WRITE(*,*)'  
WRITE(*,*)' *** PAS DE VALEURS DE a ET b  
DISPONIBLES ***'  
WRITE(*,*)' *** CHOISISSEZ UNE AUTRE  
OPTION ***'  
GOTO 21  
ENDIF  
ENDIF  
c *** memoriser la facon dont l'usager a decide de remplacer les  
c *** valeurs manquantes  
IBT=ICHAR(METD)-48  
c *** appeller la bonne sous-routine de remplacement, selon le choix  
c *** effectue et le parametre de qualite traite  
TRONCAN=.FALSE.  
IF (METD.EQ.'1') THEN  
c *** boucher les trous en les remplaçant par la moyenne saisonniere  
CALL METDMOY(VECSER,VALVIDE,TRANSF,TRONCAN)  
ELSEIF (METD.EQ.'2') THEN  
c *** boucher les trous par interpolation  
CALL INTERPOL(VECSER,TRANSF,VALVIDE,N)  
ELSEIF (METD.EQ.'3') THEN  
c *** boucher les trous a l'aide de la relation concentration-debit  
CALL CONCDEB(VECSER,VALVIDE,ITYP)  
ENDIF  
ELSE  
c *** cas ou la serie est complete  
c *** on appelle trouVEX pour mettre les bonnes valeurs  
c *** dans iex1,iex2 et ls  
CALL TROUVEX(VECSER,VALVIDE,N)  
IBT=0  
ENDIF  
c *** on determine le nom du prochain fichier de travail  
c *** on y inscrit la serie completee  
IF (ITYP.EQ.0) THEN  
FICH(LG+1;LG+4)='.TMC'  
OPEN(2,FILE=FICH,STATUS='NEW')  
ELSE  
FICH(LG+1;LG+4)='.TML'  
OPEN(2,FILE=FICH,STATUS='NEW')  
ENDIF  
WRITE(2,58)(VECSER(I),I=IEX1,IEX2)  
58 FORMAT(E16.10)  
c *** Etude de persistance  
CALL AUTOCOR(VECSER,TAMP,VALVIDE,IEX1,IEX2,LS)
```



```
c *** on inscrit dans "ident.tmp" les parametres utilises
WRITE(4,60)INTERV,IBIN,NINTAN,N
WRITE(4,'(I4,1X,I4,1X,I4,1X,I1,1X,I1)')IEX1,IEX2,LS,IDIV,IBT
WRITE(4,'(I1,1X,F5.2)')IPERS,RHO
60 FORMAT(I2,1X,I1,1X,I3,1X,I4)
CLOSE(1)
CLOSE(2)
CLOSE(3)
CLOSE(4)
WRITE(*,*)'
```

```
c *** ouverture d'un fichier-bidon pour signaler que l'execution du
c *** programme p4 s'est effectuee normalement i.e. sans erreur provoquant
c *** l'arret imprevu du programme
```

```
OPEN(99,FILE='P4.OK',STATUS='NEW')
CLOSE(99)
END
```

```
SUBROUTINE ENJOURS(VECDON,ITYP)
```

```
C *****
```

```
c -----
```

```
c Fonction:
```

```
c . Repartition des observations pour un intervalle donne en jours
c . Si plusieurs observations tombent dans le meme intervalle,on
c . inscrit alors la moyenne de ces observations
```

```
c Parametres:
```

```
c . vecdon Vecteur destine a contenir la serie equidistante
c (concentrations ou charges) (S)
c . ityp Issu de "ident.tmp",indique le parametre de qualite
c a analyser (conc ou charge) (E)
```

```
c -----
```

```
COMMON /BLOC41/LOW,INTERV,N,NINTAN
COMMON /BLOC44/ IEX1,IEX2,LS
COMMON /BLOC46/ITBACK,ITFORE
INTEGER AN,MO,JR,NBVAL,INTERV,LOW,IND1,IND2,NBJRS,NINTAN,N
INTEGER JPARM(12),LS
REAL DONN,TOTD
CHARACTER*30 F
DIMENSION VECDON(*)
DATA JPARM /31,28,31,30,31,30,31,31,30,31,30,31/
```

```
CALL @CLEAR(ITBACK,ITFORE)
CALL @CMOV(0,23)
```

```
3 WRITE(*,*)'
```

```
c *** celui-ci doit etre un nombre de mois qui divise 12 sans reste
2  WRITE(*,*) ' ENTREZ LA DUREE DE CHAQUE INTERVALLE '
  WRITE(*, '(A33,\)') ' DE TRAVAIL (1,2,3,4,6,12) : '
  READ(*, '(I3)') INTERV
  IF (MOD(12,INTERV).EQ.0) GOTO 5
  WRITE(*,*) ' '
  WRITE(*,*) ' *** REPONSE INVALIDE *** '
  GOTO 2

c *** calcul du nombre d'intervalles par annee
5  NINTAN=12/INTERV

c *** positionnement et lecture du fichier d'entree (.tmp)
  REWIND(1)
  IF (ITYP.EQ.0) THEN
    F='(I2,1X,I2,4X,E16.10)'
  ELSE
    F='(I2,1X,I2,38X,E16.10)'
  ENDIF
  READ(1,F,END=50) AN,MD,DONN

c *** trouver l'indice relatif i.e. la position associee a la date
c *** dans le vecteur
  IND1=(AN-LOW)*NINTAN+((MD-1)/INTERV+1)
  TOTD=DONN
  NBVAL=1
30  READ(1,F,END=40) AN,MD,DONN
  IND2=(AN-LOW)*NINTAN+((MD-1)/INTERV+1)
  IF (IND1.EQ.IND2) THEN
c *** les dates consecutives correspondent au meme intervalle
c *** compiler pour obtenir les valeurs moyennes
    TOTD=TOTD+DONN
    NBVAL=NBVAL+1
    GOTO 30
  ELSE
c *** calculer puis inscrire les moyennes pour l'intervalles
    VECDON(IND1)=TOTD/NBVAL
    TOTD=DONN
    NBVAL=1
    IND1=IND2
  ENDIF
  GOTO 30

c *** traitement du dernier intervalle
40  VECDON(IND2)=TOTD/NBVAL

c *** calcul de la longueur totale du vecteur (variable globale)
  N=(AN-LOW+1)*NINTAN
  GOTO 60

c *** signaler une erreur et sortir du programme
50  WRITE(*,*) ' *** ERREUR.FICHER D'ENTREE VIDE *** '
  STOP
60  RETURN
```

END

SUBROUTINE ENSAISONS(VECDON,MD,INTER,NINT,ITYP)

Fonction:

. Divise la serie selon les saisons determinees a la partie 3
si plusieurs observations tombent dans le meme intervalle,on
inscrit alors la moyenne de ces observations

Parametres:

. vecdon Vecteur destine a contenir la serie equidistante
(concentrations ou charges) (S)
. md Mois de depart utilise pour la division sasonniere (E)
. inter Intervalle utilise pour la division saisonniere (en
mois) (E)
. nint Nombre d'intervalles par annee (E)
. ityp Issu de "ident.tmp",indique le parametre de qualite
a analyser (conc ou charge) (E)

COMMON /BLOC41/LOW,INTERV,N,NINTAN
COMMON /BLOC44/ IEX1,IEX2,LS
INTEGER AN,MO,NBVAL,INTERV,LOW,IND1,IND2,NINTAN,N,LS
REAL DONN,TOTD
CHARACTER*30 F
DIMENSION VECDON(*)

*** positionnement et lecture du fichier d'entree (.tmp)

REWIND(1)

IF (ITYP.EQ.0) THEN
F='(I2,1X,I2,4X,E16.10)'
ELSE
F='(I2,1X,I2,38X,E16.10)'
ENDIF

10 READ(1,F,END=50) AN,MO,DONN

*** trouver l'indice relatif i.e. la position associee a la date dans
le vecteur

IND1=(AN-LOW)*NINT+((MO+INTER-MD)/INTER)

*** on ignore les observations avant le mois de depart donne

IF (IND1.EQ.0) GOTO 10

TOTD=DONN

NBVAL=1

30 READ(1,F,END=40) AN,MO,DONN

IND2=(AN-LOW)*NINT+((MO+INTER-MD)/INTER)

```
      IF (IND1.EQ.IND2) THEN
c ***les dates consecutives font reference au meme intervalle
c *** compiler pour obtenir les valeurs moyennes
          TOTD=TOTD+DONN
          NBVAL=NBVAL+1
          GOTO 30
      ELSE
c *** calculer puis inscrire les moyennes pour l'intervalle
          VECDON(IND1)=TOTD/NBVAL
          TOTD=DONN
          NBVAL=1
          IND1=IND2
      ENDIF
      GOTO 30

c *** traitement du dernier intervalle
40      VECDON(IND2)=TOTD/NBVAL

c *** calcul de la longueur totale du vecteur (variable globale)
          N=(AN-LOW+1)*NINT

c *** on place les valeurs d'intervalles dans les variables globales
c *** correspondantes; ainsi, interv et nintan contiendront les bonnes
c *** valeurs, peu importe la methode de division choisie

          INTERV=INTER
          NINTAN=NINT
          GOTO 60
50      WRITE(*,*)'*** ERREUR.FICHER D'ENTREE VIDE ***'
          STOP
60      RETURN
          END

      SUBROUTINE TRONQUEE(VECDON,MD,INTER,NINT,ITYP)
c * * * * *
c -----
c Fonction:
c
c . Divise la serie avec certains mois elimines a la partie 3
c   si plusieurs observations tombent dans le meme intervalle, on
c   inscrit alors la moyenne de ces observations
c
c Parametres:
c
c . vecdon   Vecteur destine a contenir la serie equidistante
c             (concentrations ou charges) (S)
c . md       Premier mois considere par la troncation (E)
c . inter    Intervalle specifie (en mois) (E)
c . nint     Nombre d'intervalles par annee (E)
c . ityp     Issu de "ident.tmp", indique le parametre de qualite
c             a analyser (conc ou charge) (E)
c
```

c

```
COMMON /BLOC41/LOW,INTERV,N,NINTAN  
COMMON /BLOC44/ IEX1,IEX2,LS  
INTEGER AN,MO,NBVAL,INTERV,LOW,IND1,IND2,NINTAN,N,LS  
REAL DONN,TOTD  
CHARACTER*30 F  
DIMENSION VECDON(*)
```

c *** positionnement et lecture du fichier d'entree (.tmp)

```
REWIND(1)  
IF (ITYP.EQ.0) THEN  
    F='(I2,1X,I2,4X,E16.10)'  
ELSE  
    F='(I2,1X,I2,38X,E16.10)'  
ENDIF  
10 READ(1,F,END=50) AN,MO,DONN
```

c *** ignorer les donnees qui tombent dans les mois tronques

```
IF (MD+NINT.LE.12) THEN  
    IF (MO.LT.MD.OR.MD.GT.MD+NINT) GOTO 10  
    IND1=(AN-LOW)*NINT+(MO-MD+1)  
ELSE  
    IF (MO.LT.MD.AND.MD.GT.(MD+NINT-13)) GOTO 10  
    IF (MO.GE.MD) THEN  
        IND1=(AN-LOW)*NINT+(MO+NINT-12)  
    ELSE  
        IND1=(AN-LOW)*NINT+MO  
    ENDIF  
ENDIF
```

c *** trouver l'indice relatif i.e. la position associee a la date dans

c *** le vecteur

```
TOTD=DONN  
NBVAL=1
```

```
30 READ(1,F,END=40) AN,MO,DONN
```

c *** ignorer les donnees qui tombent dans les mois tronques

```
IF (MD+NINT.LE.12) THEN  
    IF (MO.LT.MD.OR.MD.GT.MD+NINT) GOTO 30  
    IND2=(AN-LOW)*NINT+(MO-MD+1)  
ELSE  
    IF (MO.LT.MD.AND.MD.GT.(MD+NINT-13)) GOTO 30  
    IF (MO.GE.MD) THEN  
        IND2=(AN-LOW)*NINT+(MO+NINT-12)  
    ELSE  
        IND2=(AN-LOW)*NINT+MO  
    ENDIF  
ENDIF
```

```
IF (IND1.EQ.IND2) THEN
```

c *** deux dates consecutives font reference a un meme intervalle

c *** compiler pour obtenir les valeurs moyennes

```
TOTD=TOTD+DONN  
NBVAL=NBVAL+1
```

```

      GOTO 30
    ELSE
c *** calculer puis inscrire les moyennes pour l'intervalle
      VECDON(IND1)=TOTD/NBVAL
      TOTD=DONN
      NBVAL=1
      IND1=IND2
    ENDIF
    GOTO 30

c *** traitement du dernier intervalle
  40   VECDON(IND2)=TOTD/NBVAL

c *** calcul de la longueur totale du vecteur (variable globale)
      N=(AN-LOW+1)*NINT

c *** on place les valeurs d'intervalles dans les variables globales
c *** correspondantes; ainsi, interv et nintan contiendront les bonnes
c *** valeurs, peu importe la methode de division choisie

      INTERV=INTER
      NINTAN=NINT
      GOTO 60
  50   WRITE(*,*)'*** ERREUR.FICHER D'ENTREE VIDE ***'
      STOP
  60   RETURN
      END
```

SUBROUTINE AFFICHE(VECSER,VALVIDE,COMPLETE,RETOUR)

C * * * * *

C -----
C Fonction:
C
C . Affiche a l'ecran, pour chaque intervalle, le nombre de valeurs
C absentes pour cet intervalle a travers les annees.

C Parametres:
C
C . vecser Vecteur contenant la serie equidistante (E)
C . valvide Valeur representant les valeurs absentes (E)
C . complete Booleen qui vaut "true" en sortie si la serie ne
C comporte aucune valeur manquante (S)
C . retour Booleen qui prendra la valeur "true" si l'on desire
C prendre un intervalle different (S)
C -----

```

COMMON /BLOC41/LOW,INTERV,N,NINTAN
COMMON /BLOC44/IEX1,IEX2,LS
COMMON /BLOC46/ITBACK,ITFORE
INTEGER NINTAN,N,LOW,INTERV,ABSENTS(365),INT
DIMENSION VECSE(*)
CHARACTER*1 TOUCHE,REP
```

LOGICAL COMPLETE,RETOUR

COMPLETE=.TRUE.

c *** trouver le nombre de valeurs absentes aux extremités
CALL TROUVEX(VECSER,VALVIDE,N)

c *** initialisation
DO 10 I=1,NINTAN
 ABSENTS(I)=0
10 CONTINUE
 NBVTOT=0

c *** examiner chaque valeur du vecteur
DO 20 I=IEX1,IEX2
c *** incrementer le nombre de valeurs absentes vis-a-vis l'intervalle
c *** correspondant
 IF (VECSER(I).EQ.VALVIDE) THEN
 INT=MOD(I,NINTAN)
 IF (INT.EQ.0) INT=NINTAN
 ABSENTS(INT)=ABSENTS(INT)+1
 COMPLETE=.FALSE.
 ELSE
 NBVTOT=NBVTOT+1
 ENDIF
20 CONTINUE

c *** impression du tableau
CALL QDCLEAR(ITBACK,ITFDRE)
WRITE(*,*) 'TABLEAU 4.1 : NOMBRE DE VALEURS ABSENTES PAR INTER',
* 'VALLE'
CALL QCMOV(0,23)
WRITE(*,*)

c *** specifier le nombre possible de valeurs,i.e. le nombre d'annees
WRITE(*,25) 'NOMBRE D'ANNEES CONSIDEREES :',N/NINTAN
25 FORMAT(9X,A30,1X,I2)

c *** calcul du critere
XLS=LS
CRIT=XLS/NBVTOT

```
WRITE(*,*) '-----'
WRITE(*,*) '
WRITE(*,*) '      INTERVALLE      NOMBRE DE VALEURS      PROPORTION'
WRITE(*,*) '                        ABSENTES                ABSENTE %'
WRITE(*,*) '-----'
WRITE(*,*) '
DO 30 I=1,NINTAN
    XABS=ABSENTS(I)
    PROP=(XABS/(N/NINTAN))*100
    WRITE(*,40) I,ABSENTS(I),PROP
    FORMAT(8X,I3,16X,I2,14X,F5.1)
    IF ((MOD(I-15,25).EQ.0).OR.(I.EQ.NINTAN)) THEN
        IF (I.EQ.NINTAN) THEN
```

```

WRITE(*,*)' '
WRITE(*,35)' # TOTAL INTERVALLES / # INTERVALLES
OCCUPES
: ',CRIT
ENDIF
c *** permettre a l'utilisateur de visualiser l'ecran
CALL QCMOV(1,0)
CALL QCHAR(27,ITFORE,1)
CALL QBEEP
READ(*,'(A1)') TOUCHE
ENDIF
30 CONTINUE
35 FORMAT(A49,F4.2)
RETOUR=.FALSE.
IF (CRIT.GT.1.2) THEN
WRITE(*,*)' '
WRITE(*,*)' *** CREATION D'INFORMATION FICTIVE ***'
WRITE(*,'(A48\)' )'VOULEZ-VOUS CHOISIR UN AUTRE INTERVALLE (O/N)
?'
READ(*,'(A1)')REP
IF (REP.NE.'N'.AND.REP.NE.'n'.AND.REP.NE.'o'.AND.REP.NE.'O')
THEN
CALL QBEEP
GOTO 50
ENDIF
IF (REP.EQ.'O'.OR.REP.EQ.'o') RETOUR=.TRUE.
ENDIF
RETURN
END
```

```

SUBROUTINE DTANN(VECSER)
*****
```

```

COMMON /BLOC41/LOW,INTERV,N,NINTAN
COMMON /BLOC47/ ITRONC,IANTR,IDURTR
DIMENSION VECSER(*)
```

c Fonctions:
c
c . Retire de vecser les annees qui ont ete identifiees comme
c contenant peu de donnees
c . On retirera IDURTR annees, a partir de l'annee IANTR (varia-
c bles globales)
c . La valeur de la variable globale N est modifiee en consequence
c
c Parametres:
c
c . vecser Vecteur duquel on desire retirer les annees (E/S)
c
c -----

```

N=N-IDURTR*NINTAN
DO 10 I=(IANTR-LOW)*NINTAN+1,N
```



```
10  |      VEC SER ( I ) = VEC SER ( I + IDURTR * NINTAN )  
    | CONTINUE  
    | RETURN  
    | END
```

```
C      SUBROUTINE METDMOY ( VEC SER , VALVIDE , TRANSF , TRONCAN )  
C      * * * * *
```

```
C      -----  
C      Fonction:  
C  
C      . Calcule pour chaque saison la moyenne des valeurs obtenues  
C      sur les annees; substitue cette moyenne aux donnees manquantes  
C      dans le vecteur donne en entree (concentrations ou charges)  
C  
C      Parametres:  
C  
C      . vecser      Vecteur contenant la serie equidistante (E/S)  
C      . valvide     valeur attribuee aux valeurs absentes (E)  
C      . transf      Vecteur auxiliaire #  
C      . troncan     Boolean qui a la valeur "true" si on ne veut que  
C      remplacer les annees rtonquees par les moyennes  
C      saisonnieres (E)  
C      -----
```

```
C      COMMON /BLOC41/ LOW, INTERV, N, NINTAN  
C      COMMON /BLOC44/ IEX1, IEX2, LS  
C      COMMON /BLOC46/ ITBACK, ITFORE  
C      COMMON /BLOC47/ ITRONC, IANTR, IDURTR  
C      REAL TOT, NO, VAL, MOYINT (365), TAMP (365)  
C      CHARACTER*1 TOUCHE  
C      LOGICAL APPEL1, APPEL2, TRONCAN  
C      DIMENSION VEC SER (*), TRANSF (*)
```

```
C *** calcul des moyennes  
C      APPEL1 = .FALSE.  
C      J = 1  
C      5      K = 0  
C      TOT = 0  
C      NO = 0
```

```
C *** ind est l'indice courant de la valeur a observer dans le vecteur  
C *** on parcourt le vecteur en recherchant les valeurs relatives  
C *** a une meme saison (a intervalle regulier)
```

```
10      IND = J + NINTAN * K  
       IF (IND.GT.N) GOTO 20  
       VAL = VEC SER (IND)  
       IF (VAL.EQ.VALVIDE) GOTO 15
```

```
C *** faire la somme des valeurs pour cette saison  
C      TOT = TOT + VAL
```

```
      NO=NO+1
15     K=K+1
      GOTO 10
20     IF (NO.EQ.0) THEN
c *** il n'y a aucune observation pour cette saison. Il faudra par la
c *** suite interpoler
      MOYINT(J)=VALVIDE
      APPEL1=.TRUE.
      GOTO 25
    ENDIF

c *** determiner la moyenne pour la saison
      MOYINT(J)=TOT/NO
25     J=J+1
      IF (J.GT.NINTAN) GOTO 30
      GOTO 5
30     IF (APPEL1.EQV..TRUE.) THEN
c *** completer en interpolant pour chaque intervalle ou la moyenne
c *** n'a pu etre calculee (faute d'observations). Donner un message
c *** a l'usager.
      CALL INTERPOL(MOYINT,TAMP,VALVIDE,NINTAN)
      WRITE(*,*) '
      WRITE(*,*) '          *** ATTENTION ***
      WRITE(*,*) ' CERTAINES MOYENNES PEUVENT AVOIR ETE INTERPOLEES'
      WRITE(*,*) '
    ENDIF
      CONTINUE

c *** substitution des moyennes aux valeurs manquantes
c *** laisser les valeurs manquantes aux extremités du vecteur
      APPEL2=.FALSE.
      IF (TRONCAN.EQV..TRUE.) THEN
c *** on ne remplace que les valeurs tombant dans les années qu'on
c *** a préalablement suggere de tronquer
      J1=(IANTR-LOW)*NINTAN+1
      J2=(IANTR-LOW+IDURTR)*NINTAN
      DO 35 II=J1,J2
35     VECSER(II)=VALVIDE
      ELSE
c *** determiner les extremités non-absentes de la serie
      CALL TROUVEX(VECSER,VALVIDE,N)
      J1=IEX1
      J2=IEX2
    ENDIF
      DO 40 I=J1,J2
      IF (VECSER(I).EQ.VALVIDE) THEN
c *** remplacer la valeur manquante par la moyenne calculee
      IND=MOD(I,NINTAN)
      IF (IND.EQ.0) IND=NINTAN
      IF (MOYINT(IND).EQ.VALVIDE) THEN
c *** signaler qu'il faudra interpoler la serie resultante
      APPEL2=.TRUE.

```

```

c *** remplacer la valeur manquante par la moyenne saisonniere associee
      ELSE
      VECSER(I)=MOYINT(IND)
      ENDIF
    ENDIF
40 CONTINUE
c *** Interpoler la serie resultante
      IF ((APPEL2.EQV..TRUE.).AND.(TRONCAN.EQV..FALSE.)) THEN
      CALL INTERPOL(VECSER,TRANSF,VALVIDE,N)
      WRITE(*,*)'
      WRITE(*,*)'          *** ATTENTION ***'
      WRITE(*,*)' CERTAINES VALEURS DE LA SERIE RESULTANTE SONT'
      WRITE(*,*)' EGALEMENT SUSCEPTIBLES D'AVOIR ETE INTERPOLEES'
      WRITE(*,*)'
      ENDIF

c *** s'assurer que l'usager voit le message d'erreur s'il y en a eu un
      IF (APPEL1.EQV..TRUE..OR.APPEL2.EQV..TRUE.) THEN
      CALL QCMOV(1,1)
      CALL QCHAR(27,ITFORE,1)
      CALL QBEEP
      READ(*,'(A1)') TOUCHE
      ENDIF
RETURN
END
```

SUBROUTINE INTERPOL(VECSER,TRANSF,VALVIDE,LONG)

Fonction:

- . Remplacement des valeurs manquantes du vecteur donne en entree (concentrations ou charges) par les valeurs calculées par interpolation lineaire. Les valeurs manquantes au debut et a la fin du vecteur sont laissees intactes.
- . Les valeurs absentes aux extremités sont laissees intactes

Parametres:

- . vecser Vecteur contenant la serie equidistante (E/S)
- . transf Vecteur auxiliaire #
- . valvide Valeur attribuee aux valeurs absentes (E)
- . long Longueur du vecteur "vecser" (E)

```

COMMON /BLOC41/LOW,INTERV,N,NINTAN
COMMON /BLOC44/ IEX1,IEX2,LS
INTEGER LOW,INTERV,N,NINTAN,DISTG,DISTD,PREMVAL,L,LONG,LS
REAL VALGAUCHE,VALDROITE,VECSER(*),TRANSF(*)
```

c *** determiner les extremités non-absentes du vecteur

```
CALL TROUVEX(VECSER,VALVIDE,LONG)
DO 50 I=IEX1+1,IEX2-1
  IF (VECSER(I).NE.VALVIDE) THEN
    TRANSF(I)=VECSER(I)
  ELSE
c *** Interpoler a l'aide des deux valeurs voisines
    DISTG=0
    K=I
    K=K-1
    DISTG=DISTG+1
    IF (VECSER(K).EQ.VALVIDE) GOTO 10
c *** memoriser le premier voisin de gauche (1ere valeur qui precede)
    VALGAUCHE=VECSER(K)
    DISTD=0
    K=I
    K=K+1
    DISTD=DISTD+1
    IF (VECSER(K).EQ.VALVIDE) GOTO 20
c *** memoriser le premier voisin de droite (1ere valeur qui suit)
    VALDROITE=VECSER(K)
c *** faire une moyenne des deux valeurs voisines trouvees : cette moyenne
c *** sera ponderee par la distance ou se trouve le voisin en question
    TRANSF(I)=(DISTG*VALDROITE+DISTD*VALGAUCHE)/(DISTG+DISTD)
  ENDIF
50 CONTINUE

c *** transfert des valeurs calculees dans le vecteur original
55 DO 70 I=IEX1+1,IEX2-1
  VECSEK(I)=TRANSF(I)
70 CONTINUE
RETURN
END
```

SUBROUTINE CONCDEB(VECSER,VALVIDE,ITYP)

Fonction:

- Remplacement des valeurs manquantes des vecteurs donnees en entree (vecteurs des concentrations et des charges) par la valeur calculee avec les relations :
 $concentr = A \times (debit \times B)$
 $charge = concentr \times debit \times 86400$
ou A et B sont les estimateurs issus de la regression (partie 3). Les valeurs absentes aux extremités sont laissees intactes

Parametres:

- vecser Vecteur contenant la serie equidistante (E/S)
- valvide Valeur attribuee aux valeurs absentes (E)
- ityp Issu de "ident.tmp", indique le parametre de qualite dont on fait l'analyse (concentration ou charge) (E)

c

```
COMMON /BLOC41/LOW,INTERV,N,NINTAN  
COMMON /BLOC42/ACQ,BCQ,CHOIX,FICHDEB  
COMMON /BLOC44/ IEX1,IEX2,LS  
COMMON /BLOC45/ IDIV,IBIN,MD,INTER,NINT  
COMMON /BLOC47/ ITRONC,IANTR,IDURTR  
INTEGER AN,ANBASE,ANDEB,DECDEB,SKIP  
LOGICAL OK  
CHARACTER*12 FICHDEB  
CHARACTER*1 CHOIX  
DIMENSION VECSER(*)
```

```
WRITE(*,*) ' '
```

```
c *** ouverture du fichier de debits journaliers  
OPEN(5,FILE=FICHDEB,STATUS='OLD')
```

```
c *** positionnement du fichier de debits  
CALL POSITIONNE
```

```
c *** laisser les valeurs manquantes aux extremités du vecteur  
CALL TROUVEX(VECSER,VALVIDE,N)  
ANDEB=-99  
MODEB=-99  
DECDEB=-99  
OK=.FALSE.
```

```
DO 130 I=IEX1,IEX2  
  IF (VECSER(I).EQ.VALVIDE) THEN  
c *** trouver la date correspondant a la position vectorielle donnee  
c *** en argument  
    IF (ITRONC.EQ.1.AND.I.GT.(IANTR-LOW)*NINTAN) THEN  
      IV=I+(IANTR-LOW)*NINTAN  
    ELSE  
      IV=I  
    ENDIF  
    CALL TRANSDATE(IDIV,IV,IBIN,LOW,NINTAN,INTERV,MD,INTER,  
      NINT,AN,MO,JR)
```

```
c *** retracer le debit journalier qui correspond a la date calculee  
CALL TROUVDEB(AN,MO,JR,ANDEB,MODEB,DECDEB,OK,DEB)
```

```
c *** calcul de la donnee manquante a l'aide des equations mentionnees  
c *** plus haut
```

```
  IF (ITYP.EQ.0) THEN  
    VECSER(I)=ACQ*DEB**BCQ  
  ELSE  
    VECSER(I)=(ACQ*DEB**BCQ)*DEB*86400  
  ENDIF
```

```
130 CONTINUE  
CLOSE(5)  
RETURN
```

END

SUBROUTINE TROUVEX(VECSER,VALVIDE,N)

C * * * * *

C -----
C Fonction:
C
C . Determine les indices des extremités non-absentes du vecteur
C donne en parametre
C . Place la valeur de ces indices dans les variables globales
C iex1 et iex2.
C Place dans la variable globale "ls" la longueur totale contenue
C entre ces deux extremités
C
C Parametres:
C
C . vecser Vecteur dont on veut determiner les extremités (E)
C . valvide Valeur attribuee aux valeurs absentes (E)
C . n Longueur du vecteur a traiter (E)
C -----

COMMON /BLOC44/IEX1,IEX2,LS
DIMENSION VECSER(*)
INTEGER LS

 K=1
C *** trouver la premiere valeur non-absente
10 IF (VECSER(K).NE.VALVIDE) THEN
 IEX1=K
 GOTO 20
ELSE
 K=K+1
 GOTO 10
ENDIF
20 K=N
C *** trouver la derniere valeur non-absente
30 IF (VECSER(K).NE.VALVIDE) THEN
 IEX2=K
 GOTO 40
ELSE
 K=K-1
 GOTO 30
ENDIF
C *** determiner la longueur de la serie
40 LS=IEX2-IEX1+1
 RETURN
 END

SUBROUTINE TROUVDEB(AN,MO,JR,ANDEB,MODEB,DECDEB,OK,DEB)

c * * * * *

c -----

c Fonction:

c . Trouve le debit journalier pour la date passee en argument
c . Le fichier de debits doit etre un fichier standard issu de la
c banque IPAR

c Parametres :

c . an,mo,jr Annee,mois et jour representant la date
c pour laquelle on desire avoir le debit (E)
c . andeb,modeb,decdeb Annee,mois et decade representant la position
c courante de lecture dans le fichier de debits
c Ces valeurs sont donc alterees a la sortie de
c la sous-routine (E/S)
c . ok Booleen dont la valeur est "true" uniquement
c si il s'agit de notre premiere recherche (E/S)
c . deb Valeur du debit recherche (S)

c -----

INTEGER*2 AN,ANDEB,DECDEB
LOGICAL OK
CHARACTER*12 FICHDEB
COMMON /BLOC11/ FICHDEB

```
c *** calcul de la decade relative a la date
      IDECADE=((JR-1)/10)+1
      IF (JR.EQ.31) IDECADE=3
      IF ((AN.EQ.ANDEB).AND.(MO.EQ.MODEB).AND.(IDECADE.EQ.DECDEB)) THEN
C *** on continue a lire sur la meme ligne
          IPREPOS=IPOS
          IPOS=JR-((IDECADE-1)*10)
          IDIFF=IPOS-IPREPOS
          READ(5,'(1X)\')
          GOTO 40
      ELSE
C *** on passe a la ligne suivante a moins qu'il ne s'agisse de
c *** la toute premiere ligne du fichier
          IF (OK.EQV..TRUE.) THEN
              READ(5,'(1X)\')
          ELSE
c *** signaler que la iere ligne est maintenant traitee
              OK=.TRUE.
          ENDIF
      ENDIF
      10  READ(5,20,END=60)ANDEB,MODEB,DECDEB
      20  FORMAT(9X,I2,I2,I1\')
c *** cas ou on est deja positionne sur la bonne date
      IF ((ANDEB.EQ.AN).AND.(MODEB.EQ.MO).AND.(DECDEB.EQ.IDECADE))
      *   GOTO 30
```

```
C *** lecture pour changer de ligne
      READ(5,'(1X)')
      GOTO 10
30    IDIFF=JR-(IDECADE-1)*10
      IFOS=IDIFF

c *** avancer horizontalement dans le fichier
40    DO 50 I=1,IDIFF-1
      READ(5,'(6X)\')
50    CONTINUE
c *** lecture du debit correspondant a la date
      READ(5,'(F5.0)\')DEB
      GOTO 80

c *** erreur s'il n'y a pas de debit pour la date recherchee;
c *** Arrêt du programme.
60    WRITE(*,70)' AUCUNE VALEUR DE DEBIT POUR LE ',JR,'/',MO,'/',AN,
#     ' DANS LE FICHIER ',FICHDEB
70    FORMAT(A29,I2,A1,I2,A1,I2,A17,A50)
      STOP
80    RETURN
      END
```

SUBROUTINE POSITIONNE

```
C * * * * *
```

```
c -----
c Fonctions:
c
c . Positionne le fichier de debits au debut de la premiere annee
c   correspondant au 1er prelevement
c
c . Place dans les variables low,modep et jodep (globales) la date
c   du tout premier prelevement
c -----
```

```
COMMON /BLOC12/LOW,MODEP,JODEF
INTEGER SKIP,ANBASE,DECBASE
```

```
c *** lire la date ou commence le fichier de debits journaliers
      READ(5,20,END=90)ANBASE,MOBASE,DECBASE
20    FORMAT(9X,I2,I2,I1\ )
c *** Positionner le fichier de debits au 1er janvier d'une annee :
c *** cas ou le fichier debute au 1er janvier
      IF ((MOBASE.EQ.1).AND.(DECBASE.EQ.1)) GOTO 40
c *** cas special ou le fichier commence a la fin decembre
      IF ((MOBASE.EQ.12).AND.(DECBASE.EQ.3)) THEN
      SKIP=LOW-1-ANBASE
      READ(5,'(1X)')
      GOTO 60
```



```

  ENDIF
  READ(5,'(1X)',END=90)
30  READ(5,'(11X,I2,I1\)',END=90)MOBASE,DECBASE
    IF (MOBASE.EQ.12.AND.DECBASE.EQ.3) THEN
      SKIP=LOW-1-ANBASE
      READ(5,'(1X)',END=90)
      GOTO 60
    ELSE
      READ(5,'(1X)',END=90)
      GOTO 30
    ENDIF
40  SKIP=LOW-ANBASE
    REWIND(3)

```

c *** Dans le fichier de debits journaliers,sauter des blocs de lignes
c *** et se positionner au debut de l'annee qui correspond a l'annee de
c *** la premiere observation.
c *** Sauter un bloc = sauter une annee (12 x 3 decades = 36 lignes).

```

60  DO 80 I=1,SKIP
    READ(5,70)
70  FORMAT(/,/ /,/ /,/ /,/ /,/ /,/ /,/ /,/ /,/ /,/ /,/ /,/ /,/ /,/ /,/ /,
  * /,/ /,/ /,/ /)
80  CONTINUE
    GOTO 110

```

c *** erreur signalee si la fin du fichier est rencontree sans avoir
c *** trouve l'annee recherchee.Arret inconditionnel du programme.

```

90  WRITE(*,*)'ERREUR:FICHIER DE DEBITS INADEQUAT'
    STOP

```

c *** erreur si le fichier d'entree ne contient aucune observation

```

100 WRITE(*,*)' *** FICHIER D'ENTREE VIDE ***'
    STOP
110 RETURN
    END

```

```

$INCLU DE:'AUTO.FOR'
$INCLU DE:'TDATE.FOR'

```

Thu 09-10-87 15:07:05
06-04-87 14:01:26 P4PLUS.FOR

TROUVDEB	P4PLUS.FOR	1
SUBROUTINE TROUVDEB (AN,MO, JR, ANDEB, MODEB		

POSITIONNE	P4PLUS.FOR	83
SUBROUTINE POSITIONNE		

SUBROUTINE TROUVDEB(AN,MO,JR,ANDEB,MODEB,DECDEB,OK,DEB)

Fonction:

. Trouve le debit journalier pour la date passee en argument
. Le fichier de debits doit etre un fichier standard issu de la
banque IPAR

Parametres :

. an,mo,jr Annee,mois et jour representant la date
 pour laquelle on desire avoir le debit (E)
. andeb,modeb,decdeb Annee,mois et decade representant la position
 courante de lecture dans le fichier de debits
 Ces valeurs sont donc alterees a la sortie de
 la sous-routine (E/S)
. ok Booleen dont la valeur est "true" uniquement
 si il s'agit de notre premiere recherche (E/S)
. deb Valeur du debit recherche (S)

INTEGER*2 AN,ANDEB,DECDEB
LOGICAL OK
CHARACTER*12 FICHDEB
COMMON /BLOC11/ FICHDEB

```
c *** calcul de la decade relative a la date
      IDECADE=((JR-1)/10)+1
      IF (JR.EQ.31) IDECADE=3
      IF ((AN.EQ.ANDEB).AND.(MO.EQ.MODEB).AND.(IDECADE.EQ.DECDEB)) THEN
c *** on continue a lire sur la meme ligne
      IPREPOS=IPOS
      IPOS=JR-((IDECADE-1)*10)
      IDIFF=IPOS-IPREPOS
      READ(5,'(1X)\')
      GOTO 40
      ELSE
c *** on passe a la ligne suivante a moins qu'il ne sagisse de
c *** la toute premiere ligne du fichier
      IF (OK.EQV..TRUE.) THEN
          READ(5,'(1X)\')
      ELSE
c *** signaler que la 1ere ligne est maintenant traitee
          OK=.TRUE.
      ENDIF
      ENDIF
      10 READ(5,20,END=60)ANDEB,MODEB,DECDEB
      20 FORMAT(9X,I2,I2,I1\')
c *** cas ou on est deja positionne sur la bonne date
      IF ((ANDEB.EQ.AN).AND.(MODEB.EQ.MO).AND.(DECDEB.EQ.IDECADE))
```

```

$      GOTO 30

C *** lecture pour changer de ligne
      READ(5, '(1X)')
      GOTO 10
30     IDIFF=JR-(IDECADE-1)*10
      IPOS=IDIFF

C *** avancer horizontalement dans le fichier
40     DO 50 I=1, IDIFF-1
        READ(5, '(6X\)' )
50     CONTINUE
C *** lecture du debit correspondant a la date
      READ(5, '(F5.0\)' )DEB
      GOTO 80

C *** erreur s'il n'y a pas de debit pour la date recherchee;
C *** Arrêt du programme.
60     WRITE(*,70) ' AUCUNE VALEUR DE DEBIT POUR LE ',JR,'/',MO,'/',AN,
$       ' DANS LE FICHIER ',FICHDEB
70     FORMAT(A29,I2,A1,I2,A1,I2,A17,A50)
      STOP
80     RETURN
      END
```

SUBROUTINE POSITIONNE

```

C * * * * *
C -----
C Fonctions:
C
C . Positionne le fichier de debits au debut de la premiere annee
C   correspondant au 1er prelevement
C
C . Place dans les variables low,modep et jodep (globales) la date
C   du tout premier prelevement
C -----
```

```

COMMON /BLOC12/LOW,MODEP,JOEPE
INTEGER SKIP,ANBASE,DECBASE
```

```

C *** lire la date ou commence le fichier de debits journaliers
      READ(5,20,END=90)ANBASE,MOBASE,DECBASE
20     FORMAT(9X,I2,I2,I1\ )
C *** Positionner le fichier de debits au 1er janvier d'une annee :
C *** cas ou le fichier debute au 1er janvier
      IF ((MOBASE.EQ.1).AND.(DECBASE.EQ.1)) GOTO 40
C *** cas special ou le fichier commence a la fin decembre
      IF ((MOBASE.EQ.12).AND.(DECBASE.EQ.3)) THEN
        SKIP=LOW-1-ANBASE
        READ(5, '(1X)')
```

```
      GOTO 60  
    ENDIF  
    READ(5,'(1X)',END=90)  
30   READ(5,'(11X,I2,I1\)',END=90)MOBASE,DECBASE  
      IF (MOBASE.EQ.12.AND.DECBASE.EQ.3) THEN  
          SKIP=LOW-1-ANBASE  
          READ(5,'(1X)',END=90)  
          GOTO 60  
      ELSE  
          READ(5,'(1X)',END=90)  
          GOTO 30  
      ENDIF  
40   SKIP=LOW-ANBASE  
      REWIND(3)
```

c *** Dans le fichier de debits journaliers,sauter des blocs de lignes
c *** et se positionner au debut de l'annee qui correspond a l'annee de
c *** la premiere observation.
c *** Sauter un bloc = sauter une annee (12 x 3 decades = 36 lignes).

```
60   DO 80 I=1,SKIP  
      READ(5,70)  
70   FORMAT(/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,//,/,,  
$      /,/,/,/,/)/  
80   CONTINUE  
      GOTO 110
```

c *** erreur signalee si la fin du fichier est rencontree sans avoir
c *** trouve l'annee recherchee.Arret incondtionnel du programme.
90 WRITE(*,*)'ERREUR:FICHER DE DEBITS INADEQUAT'
 STOP

c *** erreur si le fichier d'entree ne contient aucune observation
100 WRITE(*,*)' *** FICHER D'ENTREE VIDE ***'
 STOP
110 RETURN
 END

```
c *** lecture de l'intervalle choisi
      WRITE(*,5)'ENTREZ LA DUREE DE CHAQUE INTERVALLE DE TRAVAIL (ENTRE
#      1 ET 21) : '
5      FORMAT(A67\ )
      READ(*,'(I3)') INTERV
      IF (INTERV.LT.1.OR.INTERV.GT.21) THEN
          WRITE(*,*)' '
          WRITE(*,*)' *** ENTREZ UN NOMBRE ENTRE 1 ET 21 ***'
          WRITE(*,*)' '
          GOTO 3
      ENDIF

c *** calcul du nombre d'intervalles par annee
      NINTAN=365/INTERV

c *** lecture du fichier d'entree (.tmp)
      REWIND(1)
      IF (ITYP.EQ.0) THEN
          F='(I2,1X,I2,1X,I2,1X,E16.10)'
      ELSE
          F='(I2,1X,I2,1X,I2,35X,E16.10)'
      ENDIF
10     READ(1,F,END=50) AN,MO,JR,DONN

c *** trouver l'indice relatif i.e. la position associee a la date
c *** dans le vecteur
      NBJRS=0
      DO 25 I=1,MO-1
          NBJRS=NBJRS+JPARM(I)
25     CONTINUE
      IND1=(AN-LOW)*NINTAN+((NBJRS+JR-1)/INTERV+1)

c *** ignorer les observations qui tombent dans le dernier intervalle
c *** si celui-ci est incomplet
      IF (IND1.GT.((AN-LOW+1)*NINTAN)) GOTO 10
      TOTD=DONN
      NBVAL=1
30     READ(1,F,END=40) AN,MO,JR,DONN
      NBJRS=0
      DO 35 I=1,MO-1
          NBJRS=NBJRS+JPARM(I)
35     CONTINUE
      IND2=(AN-LOW)*NINTAN+((NBJRS+JR-1)/INTERV+1)

c *** ignorer les observations qui tombent dans le dernier intervalle
c *** si celui-ci est incomplet
      IF (IND2.GT.((AN-LOW+1)*NINTAN)) GOTO 30
      IF (IND1.EQ.IND2) THEN
c *** les dates consecutives correspondent au meme intervalle
c *** compiler pour obtenir les valeurs moyennes
          TOTD=TOTD+DONN
          NBVAL=NBVAL+1
          GOTO 30
```

```
      ELSE
c *** calculer et inscrire les moyennes pour l'intervalle
      VECDON(IND1)=TOTD/NBVAL
      TOTD=DONN
      NBVAL=1
      IND1=IND2
      ENDIF
      GOTO 30

c *** traitement du dernier intervalle
40   VECDON(IND2)=TOTD/NBVAL

c *** calcul de la longueur totale du vecteur (variable globale)
      N=(AN-LOW+1)*NINTAN
      GOTO 60

c *** erreur signalee et sortie du programme si le fichier d'entree
c *** est vide
50   WRITE(*,*)'*** ERREUR.FICHIER D'ENTREE VIDE ***'
      STOP
60   RETURN
      END
```

SUBROUTINE ENMOIS(VECDON,ITYP)

Fonction:

. Repartition des observations pour un intervalle donne en mois
si plusieurs observations tombent dans le meme intervalle,on
inscrit alors la moyenne de ces observations

Parametres:

. vecdon Vecteur destine a contenir la serie equidistante
(concentrations ou charges) (S)
. ityp Issu de "ident.tmp",indique le parametre de qualite
a analyser (conc ou charge) (E)

```
COMMON /BLOC41/LOW,INTERV,N,NINTAN
COMMON /BLOC44/ IEX1,IEX2,LS
COMMON /BLOC46/ITBACK,ITFORE
INTEGER AN,MO,NBVAL,INTERV,LOW,IND1,IND2,NINTAN,N,LS
REAL DONN,TOTD
CHARACTER*30 F
DIMENSION VECDON(*)

CALL QCLEAR(ITBACK,ITFORE)
CALL QCMOV(0,23)
```

c *** lecture de l'intervalle choisi

```
TRANSDATE          TDATE.FOR      6  
SUBROUTINE TRANSDATE(IND,IBIN,LOW,N
```

ELSEIF

```
DATNORM           TDATE.FOR      42  
42 CALL DATNORM(IND,LOW,IBIN,NINTAN,I
```

```
                TDATE.FOR      54  
SUBROUTINE DATNORM(IND,LOW,IBIN,NINTAN,I
```

MOD

```
DATSAIS          TDATE.FOR      45  
45 CALL DATSAIS(IND,LOW,MD,INTER,NINT
```

```
                TDATE.FOR      109  
SUBROUTINE DATSAIS(IND,LOW,MD,INTER,NINT
```

MOD

```
DATTRONQ        TDATE.FOR      48  
48 CALL DATTRONQ(IND,LOW,MD,NINT,AN,M
```

```
                TDATE.FOR      154  
SUBROUTINE DATTRONQ(IND,LOW,MD,NINT,AN,M
```

MOD

c
c Ce module s'inscrit a la suite de P4.FOR, ce dernier y faisant
c reference a l'aide de la metacommande "INCLUDE".
c

SUBROUTINE TRANSDATE (IDIV,IND,IBIN,LOW,NINTAN,INTERV,MD,INTER,
\$ NINT,AN,MO,JR)

C * * * * *
c-----

c Fonction:

c . Aiguille vers la bonne sous-routine pour transformer un rang
c en date.Depend de la facon dont l'usager a decide de diviser sa
c serie.
c Le rang fournit doit etre l'indice vectoriel calcule a partir du
c 1er janvier de la 1ere annee.

c Parametres:

c Les parametres suivants sont issus du fichier "ident.tmp" :

c idiv Indique la methode choisie pour diviser la serie (E)
c ind Indice que l'on veut transformer en date (E)
c ibin Indique l' unite de division (jours/mois) (E)
c low Annee de depart de la serie (E)
c nintan Nombre d'intervalles par annee (E)
c Interv Intervalle (E)
c md Mois de depart,division saisonniere (E)
c inter Intervalle,division saisonniere (E)
c nint Nombre d'intervalles par annee,division saisonniere (E)
c an,mo,jr Date recherchee (annee,mois et jour) (S)

 INTEGER AN,MO,JR

c *** IF (IDIV.EQ.0) THEN
c | on a utilise la methode standard,division en jours/mois
c | CALL DATNORM(IND,LOW,IBIN,NINTAN,INTERV,AN,MO,JR)
c |-----ELSEIF (IDIV.EQ.1) THEN
c *** | on a effectue une division saisonniere
c | CALL DATSAIS(IND,LOW,MD,INTER,NINT,AN,MO,JR)
c |-----ELSE
c *** | on a tronquer certains mois
c | CALL DATTRONQ(IND,LOW,MD,NINT,AN,MO,JR)
c |-----ENDIF
c | RETURN
c | END

SUBROUTINE DATNORM (IND,LOW,IBIN,NINTAN,INTERV,AN,MO,JR)

C * * * * *


```
c -----  
c Fonction:  
c  
c . Convertit un rang donne (situe entre 1 et N, i.e. relatif au  
c 1er janvier de la premiere annee) en la date  
c correspondante,dans le cas ou l'usager a choisi de diviser sa  
c serie en saisons.Si l'intervalle comporte plusieurs mois ( cas  
c habituel ) on indique alors le milieu de l'intervalle.  
c  
c Parametres:  
c  
c S.V.P. Referer a la sous-routine "transdate"  
c -----
```

INTEGER AN,MD,JR

```
c *** retracer l'annee correspondante  
AN=(IND-1)/NINT+LOW  
IF (MD.EQ.1) THEN  
    MD=(MOD(IND-1,NINT)*INTER)+(INTER/2+1)  
ELSE  
    MD=(MOD(IND-1,NINT)*INTER+MD-1)+(INTER/2+1)  
    IF (MD.GT.12) THEN  
c *** cas ou l'intervalle donne est "a cheval" sur la fin de l'annee et  
c *** le debut de la suivante  
        AN=AN+1  
        MD=MOD(MD,12)  
    ENDIF  
ENDIF  
c *** determiner le jour dans le milieu de l'intervalle  
IF (MOD(INTER,2).EQ.0) THEN  
    JR=1  
ELSE  
    JR=15  
ENDIF  
RETURN  
END
```

SUBROUTINE DATTROND(IND,LOW,MD,NINT,AN,MO,JR)

C * * * * *

```
c -----  
c Fonction:  
c  
c . Convertit un rang donne (situe entre 1 et N, i.e. relatif au  
c 1er janvier de la premiere annee) en la date  
c correspondante,dans le cas ou l'usager a choisi de tronquer  
c certains mois de sa serie.On indique le milieu du mois trouve,  
c c'est-a-dire le jour 15.  
c
```

c Parametres:

c S.V.P. Referer a la sou-routine "transdate"

c -----

```

      INTEGER AN,MO,JR
c *** calcul de l'annee
      AN=(IND-1)/NINT+LOW
c *** calcul du mois
      MO=MOD(IND-1,NINT)+MD
      IF (MO.GT.12) THEN
c *** cas ou l'intervalle donne est "a cheval" sur la fin de l'annee et
c *** le debut de la suivante
          AN=AN+1
          MO=MOD(MO,12)
      ENDIF
      JR=15
      RETURN
      END
```

Thu 09-10-87 15:04:18
04-20-87 18:03:10 AUTO.FOR

Pg 1

```
AUTOCOR          AUTO.FOR      6  
SUBROUTINE AUTOCOR(VECSER,TAMP,VALVIDE,I
```

```
OCMOV,OCHAR,OBEEP,ABS,ELSEIF,OCLEAR
```

```
AUTO          AUTO.FOR      59  
59 CALL AUTO(TAMP,LS,7,R)
```

```
          AUTO.FOR      123  
Subroutine auto(a,n,l,r)
```

c
c Ce module s'inscrit a la suite de P4.FOR, ce dernier y faisant
c reference a l'aide de la metacommande "INCLUDE".
c

 SUBROUTINE AUTOCOR(VECSER,TAMP,VALVIDE,IEX1,IEX2,LS)

c * * * * *

c -----
c Fonction:

c . Calcul et presentation des coefficients d'autocorrelation
c et d'autocorrelation partielle

c Parametres:

c . vecser Vecteur contenant la serie equidistante (E)
c . tamp Vecteur auxiliaire #
c . valvide Valeur attribuee aux valeurs absentes (E)
c . iex1,iex2 Indices correspondant aux extremités non-absentes
c du vecteur (E)
c . ls Longueur de la serie (portion entre iex1 et iex2) (E)

c -----
REAL R(7),R2(6),ECT(6)
INTEGER ANFIN
CHARACTER*1 TOUCHE
CHARACTER*8 NOMFICH
CHARACTER*12 FICH,FICHDEB
CHARACTER*39 COMMEN
DIMENSION VEC SER(*),TAMP(*)
COMMON /BLOC41/LOW,INTERV,N,NINTAN
COMMON /BLOC43/IPERS,RHO
COMMON /BLOC46/ITBACK,ITFORE

c *** ne pas effectuer l'etude de persistance si l'on possede un
c *** nombre insuffisant de donnees

```
IF (LS.LE.7) THEN
  WRITE(*,*) ' '
  WRITE(*,*) ' *** AUCUNE ETUDE DE PERSISTANCE EFFECTUEE *** '
  WRITE(*,*) ' *** NOMBRE INSUFFISANT DE DONNEES *** '
  WRITE(*,*) ' '
  CALL QCMOV(1,1)
  CALL QCHAR(27,ITFORE,1)
  CALL QBEEP
  READ(*,'(A1)') TOUCHE
  GOTO 30
ENDIF
K=1
```

c *** transfert de la partie non-negative du vecteur

```
DO 20 I=IEX1,IEX2
  TAMP(K)=VECSER(I)
  K=K+1
```

20 —CONTINUE

```
c *** determiner les autocovariances de 1 a 6
      CALL AUTO(TAMP,LS,7,R)
      MARKOV=1
      RT=0.
      DO 10 I=1,6
         R2(I)=R(I+1)/R(1)
         ECT(I)=((1./LS)*(1+2*RT))**0.5
10     RT=RT+R2(I)**2
         PAUTO2=(R2(2)-R2(1)**2)/(1-R2(1)**2)

c *** determiner la valeur critique
      VALCRIT=1.96*ECT(1)
      IF (ABS(R2(1)).GT.VALCRIT) THEN
         RHO=R2(1)
         IPERS=1
      ELSE
         RHO=0.0
         IPERS=0
      ENDIF
      IF (ABS(PAUTO2).GT.VALCRIT) MARKOV=0

c *** determination du diagnostique a poser en fonction de la persistance
c *** et de sa nature
      IF (IPERS.EQ.1.AND.MARKOV.EQ.0) THEN
         COMMEN='PRESENCE DE PERSISTANCE NON-MARKOVienne'
      ELSEIF (IPERS.EQ.1.AND.MARKOV.EQ.1) THEN
         COMMEN='PRESENCE DE PERSISTANCE MARKOVienne'
      ELSE
         COMMEN='PAS DE PERSISTANCE SIGNIFICATIVE'
      ENDIF

c *** impression du tableau des resultats
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QCMDV(0,23)
      WRITE(*,*) 'TABLEAU 4.2 : ETUDE DE PERSISTANCE'
      WRITE(*, '(/,/)' )
      WRITE(*,*) '          AUTOCORRELATION'
      WRITE(*,*) ' '
      WRITE(*, '(9X,6I10)') 1,2,3,4,5,6
      WRITE(*,*) ' '
      WRITE(*, '(1X,AB,\)') 'COEFFIC.'
      WRITE(*, '(6F10.2)') R2(1),R2(2),R2(3),R2(4),R2(5),R2(6)
      WRITE(*, '(1X,AB,\)') 'EC.-TYPE'
      WRITE(*, '(6F10.2)') ECT(1),ECT(2),ECT(3),ECT(4),ECT(5),ECT(6)
      WRITE(*, '(/,/)' )
      WRITE(*,*) '          AUTOCORRELATION PARTIELLE'
      WRITE(*,*) ' '
      WRITE(*, '(9X,2I10)') 1,2
      WRITE(*,*) ' '
      WRITE(*, '(1X,AB,\)') 'COEFFIC.'
      WRITE(*, '(6F10.2)') R2(1),PAUTO2
      WRITE(*, '(1X,AB,\)') 'EC.-TYPE'
      WRITE(*, '(6F10.2)') ECT(1),ECT(1)
```

```
WRITE(*,*) ' '
WRITE(*, '(1X,A13,1X,A39)') 'COMMENTAIRE :',COMMENT
WRITE(*, '(/,/)' )
CALL QCMOV(1,1)
CALL QCHAR(27,ITFORE,1)
CALL QBEEP
READ(*, '(A1)')TOUCHE
RETURN
END
```

30

Subroutine auto(a,n,l,r)

c * * * * *

c Sous-routine issue du package "The Statistician", avec commentai-
c res traduits et adaptes.

c -----

c Fonction:

c . Trouver les autocovariances de la serie A pour les coefficients
c 1 a L-1

c Parametres:

c . a Vecteur d'entree de longueur n contenant la serie de temps
c pour laquelle on veut les autocovariances (E)
c . n Longueur du vecteur "a" (E)
c . l Autocovariances calculees pour les coefficients
c 0, 1, 2,.....,l-1. (S)
c . r Vecteur de sortie de longueur L contenant les autocovariances
c de la serie "a" (S)

c notes : La longueur de "r" est differente de la longueur de "a".
c "n" doit etre plus grand que "l". Sinon, "r(1)" est mis a zero
c (0) et un retour est fait au programme.

c Methode: Decrite dans Blackman and Tukey, "THE MEASUREMENT OF
c POWER SPECTRA", Dover Publications Inc.

c -----

Dimension a(1),r(1)

```
c *** Calcul de la moyenne de la serie de temps A
aver=0.0
if(n.gt.1)goto 100
r(1)=0.0
return
100 do 110 i=1,n
110 aver=aver+a(i)
fn=n
aver=aver/fn
```



```
c *** Calcul des Autocovariances
      do 130 j=1,1
          nj=n-j+1
          sum=0.0
          do 120 i=1,nj
              ij=i+j-1
              sum=sum+(a(i)-aver)*(a(ij)-aver)
              fnj=fnj+1
          120
      130 r(j)=sum/fnj
      Return
      End
```

P5.FOR

-FICH, NOMFICH, TEXTMM, TEXTMD, QSMODE
-QCLEAR, QBORD, QCMOV, QCHAR, QBEEP, CHAR
-FICHTEST, NOC

DOUBMASS P5.FOR 184
184 CALL DOUBMASS (VECTEUR, TEXTC, TEXTT,

P5.FOR 611
SUBROUTINE DOUBMASS (C, TEXTC, TEXTQ, TEXTMM

-QSMODE, QCOLOR, QOVSCN, QPLOT, QXAXIS
-QYAXIS, QGTXT, QPTXTC, QSETUP, QTABL, QPTXTA
-QRTDI, QLINE, QCMOV, QCHAR, QBEEP

TICS P5.FOR 659 x2
659 CALL TICS (0.0, SQ(LS), XSTS, XFINS, TI
660 CALL TICS (0.0, SC(LS), YSTS, YFINS, TI

TICS.FOR 1
SUBROUTINE TICS (XST, XFIN, XSTS, XFINS, TIC,

-ELSEIF, AINT, ABS

AJUSTE P5.FOR 662 x2
662 CALL AJUSTE (SQ, LS, AINTEX, TEXTEX, XS
665 CALL AJUSTE (SC, LS, AINTEY, TEXTEY, YS

AJUSTE.FOR 1
SUBROUTINE AJUSTE (X, NOBS, AINTE, TEXTE, XST

-ALOG10, EXP, CHAR, ABS, ELSEIF, MOD, TEXTE

```

CUSUM          P5.FOR  188
188 CALL CUSUM(VECTEUR,T,TEXTC,TEXTT,T

```

```

          P5.FOR  744
SUBROUTINE CUSUM(C,TC,TEXTC,TEXTT,TEXTMM

```

```

--QSMODE, QCOLOR, QOVSCN, QPLOT, QXAXIS
--QYAXIS, QSETUP, QGTXT, OPTXTC, OPTXTA, QTABL
--QRTOI, QLINE, QCMOV, QCHAR, QBEEP

```

```

TICS          P5.FOR  808 x2 p1
808 CALL TICS(MINCU,MAXCU,YSTS,YFINS,T
809 CALL TICS(O.O,TCU(NOBI),XSTS,XFINS,

```

```

AJUSTE       P5.FOR  911 x2 p1
811 CALL AJUSTE(TCU,NOB,AINTEX,TEXTEX,
815 CALL AJUSTE(VCUSUM,NOB,AINTEY,TEXT

```

```

TRANSFRANG   P5.FOR  324 x4
324 CALL TRANSFRANG(IDIV,AN,MO,JR,IBIN
352 CALL TRANSFRANG(IDIV,AN,MO,JR,IBIN
375 CALL TRANSFRANG(IDIV,AN,MO,JR,IBIN

```

```

          P5.FOR  857
SUBROUTINE TRANSFRANG(IDIV,AN,MO,JR,IBIN

```

```

--ELSEIF

```

```

RANGNORM     P5.FOR  895
895 CALL RANGNORM(AN,MO,JR,IBIN,LOW,IN

```

```

          P5.FOR  911
SUBROUTINE RANGNORM(AN,MO,JR,IBIN,LOW,IN

```


#STORA GE:2

PROGRAM PARTIES

c

c

Titre : Preparation aux tests

c

Fonctions :

c

- . Trace les graphiques double-masse et cusum
- . S'informe aupres de l'usager de la partie de serie a traiter, du desir pour une recherche d'une tendance monotone ou par saut
- . Cree un nouveau fichier de travail ne contenant que la portion de la serie que l'on souhaite traiter
- . Suggere le test approprie, a l'aide des caracteristiques de la serie de travail (tendance, saisonnalite et persistance)

c

Notation adoptee dans les sous-routines :

c

Voici le sens des symboles places a la fin de la description de chaque parametre :

c

- (E) : Il s'agit d'un parametre d'entree
- (S) : Il s'agit d'un parametre de sortie
- (E/S) : Il s'agit d'un parametre necessaire en entree ET egalement modifie en sortie.
- # : Il s'agit d'une structure de donnee qui n'est utilisee qu'a l'interieur de cette sous-routine. Elle n'est passee en parametre que parce qu'elle est declaree dans le programme principal. Cette facon de faire permet de pouvoir reutiliser la structure a diverses fins dans divers sous-programmes. (permet une economie d'espace)

c

```
INTEGER MAXI
REAL VALVIDE
PARAMETER (MAXI=1300)
PARAMETER (VALVIDE=-9)
INTEGER TEND, INTERV, IBIN, NINTAN, N, LDW, ETD, AN, MD, JR, NBJRS,
# JPARM(12), N1, N2, DR, ANFIN
CHARACTER*1 TOUCHE
CHARACTER*2 NDC
CHARACTER*6 FICHTEST
CHARACTER*15 TEXTC
CHARACTER*21 TEXTT
CHARACTER*12 FICHDEB, FICH, NOMFICH
CHARACTER*22 TEXTCU, TEXTDO
CHARACTER*30 NTEST
CHARACTER*56 TEXTMM, TEXTMD
REAL VECTEUR(MAXI), TAMP1(MAXI), TAMP2(MAXI)
COMMON /TEXTE /ITMODE, ITFORE, ITBACK, ITCOLOR
COMMON /GRAPH /IGMODE, IGBACK, IPAL, IGCOLOR, ILINE, ISEC, IGBORD
```

```
# ,KIRSYM, ISYMBL, ILETR, JROW1, JROW2, JCOL1, JCOL2
```

```
DATA JFARM/31,28,31,30,31,30,31,31,30,31,30,31/
```

```
c *** lecture des specifications graphiques
```

```
OPEN(4,FILE='SETUP.PC',STATUS='OLD')  
READ(4,'(4I2/9I2,5I3)') ITMODE,ITFORE,ITBACK,ITCOLOR,IGMODE,  
# IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,ILETR,ISEC,KIRSYM,ISYMBL,  
# JROW1,JROW2,JCOL1,JCOL2  
CLOSE(4)
```

```
c *** lecture du fichier ident.tmp
```

```
OPEN(3,FILE='IDENT.TMP',STATUS='OLD')  
READ(3,2)NOMFICH,L6,FICHDEB,IDEB,ITYP,IMPRIM,LOW,MODEP,JODEF  
# ,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER,ISAIS,MD,INTER,NINT,  
# ICG,ACQ,BCQ,ITRONC,IANTR,IDURTR,INTERV,IBIN,NINTAN,N,IEX1,IEX2,LS,  
# IDIV,IBT,IPERS,RHO  
2 FORMAT(A12,1X,I1,/,A12,1X,I1,/,I1,1X,I1,/,I2,1X,I2,1X,I2,/,I2,1X,  
# I2,1X,I2,/,I4,/,I2,1X,I2,/,I1,1X,I2,1X,I2,1X,I2,/,I1,1X,F9.2,1X,  
# F9.2,/,I1,1X,I2,1X,I2,/,I2,1X,I1,1X,I3,1X,I4,/,I4,1X,I4,1X,I4,1X,  
# I1,1X,I1,/,I1,1X,F5.2)
```

```
TEXTT='TEMPS (# INTERVALLES)'  
TEXTCU=' FONCTION CUSUM '  
TEXTDO='FONCTION DOUBLE-MASSE'
```

```
c *** reconstitution du nom du fichier de travail
```

```
FICH='  
FICH(1:L6)=NOMFICH(1:L6)  
IF (ITYP.EQ.0) THEN  
FICH(L6+1:L6+4)='.TMC'  
OPEN(1,FILE=FICH,STATUS='OLD')  
TEXTC=' CONCEN mg/l '  
TEXTMM(1:12)='FIGURE 5.3 : '  
TEXTMM(13:34)=TEXTCU  
TEXTMM(35:44)=' :C vs T, '  
TEXTMM(45:56)=NOMFICH  
TEXTMD(1:12)='FIGURE 5.1 : '  
ELSE  
FICH(L6+1:L6+4)='.TML'  
OPEN(1,FILE=FICH,STATUS='OLD')  
TEXTC=' CHARGE Kg/jour '  
TEXTMM(1:12)='FIGURE 5.4 : '  
TEXTMM(13:34)=TEXTCU  
TEXTMM(35:44)=' :L vs T, '  
TEXTMM(45:56)=NOMFICH  
TEXTMD(1:12)='FIGURE 5.2 : '  
ENDIF  
TEXTMD(13:34)=TEXTDO  
TEXTMD(35:44)=TEXTMM(35:44)
```

```
IF (DR.EQ.'1') THEN
c *** on offre les rangs (indices vectoriels) disponibles
c *** on demande les bornes desirées, exprimées en rangs
30 WRITE(*,*) '
WRITE(*, '(A46,I4)') ' VOTRE SERIE S'ECHELONNE DU RANG
1 AU
* RANG ',LS
WRITE(*,*) '
WRITE(*, '(A46\')') ' ENTREZ LE RANG DEBUTANT LA SOUS-
SERIE
*
:
READ(*,46)NTMP1
IF (NTMP1.LT.1.OR.NTMP1.GT.LS) THEN
WRITE(*,*) '
WRITE(*,*) ' *** RANG INVALIDE ***'
GOTO 30
ENDIF
N1=NTMP1
46 FORMAT(I4)
WRITE(*, '(A47\')') ' ENTREZ LE RANG TERMINANT LA SOUS-
SERIE
E :
READ(*,46)NTMP2
IF (NTMP2.LE.NTMP1.OR.NTMP2.GT.LS) THEN
WRITE(*,*) '
WRITE(*,*) ' *** RANGS INVALIDES.RECOMMENCEZ. **
*
GOTO 30
ENDIF
N2=NTMP2
WRITE(*,*) '
ELSE
c *** retracer les indices n1 et n2 a partir des dates specifiees par
c *** l'usager
c *** redonner les dates extremes a l'usager
c *** demander les dates bornant la portion souhaitee
IF (IBIN.EQ.0) THEN
c *** l'intervalles a ete donne en jours
WRITE(*,*) '
WRITE(*,48) 'VOTRE SERIE S'ECHELONNE ENTRE LE ',
JOSEP,
' ',MODEP,'/',LOW,' ET LE ',JOFIN,'/',MOFIN,'/',
ANFIN
48 FORMAT(A37,I2,A1,I2,A1,I2,A8,I2,A1,I2,A1,I2)
49 WRITE(*,*) '
WRITE(*,*) 'DONNEZ LA DATE DEBUTANT LA SOUS-SERIE
:
WRITE(*,*) '
WRITE(*, '(A8\')') 'JOUR :
```

```
ELSE
c *** l'intervalle a ete donne en mois
WRITE(*,48) ' VOTRE SERIE S'ECHELONNE ENTRE LE
',JODEP,
',/',MODEP,/',',LOW,' ET LE ',JOFIN,/',',MOFIN,/',',
ANFIN
75 WRITE(*,*) '
WRITE(*,*) 'DONNEZ LE MOIS ET L'ANNEE DEBUTANT
LA '
WRITE(*,*) 'SOUS-SERIE : '
WRITE(*,*) '
WRITE(*, '(AB\)' ) 'MOIS : '
READ(*,80)MO
WRITE(*, '(A11\)' ) 'ANNEE : 19'
80 READ(*,80)AN
FORMAT(I2,1X,I2)
IF (AN.LT.LOW.OR.AN.GT.ANFIN.OR.MO.LT.1.OR.MO.GT.
12) THEN
WRITE(*,*) '
WRITE(*,*) ' *** DATE INVALIDE.RECOMMENCEZ
. *** '
GOTO 75
ENDIF
ANTMP=AN
MOTMP=MO
CALL TRANSFRANG(IDIV,AN,MO,JR,IBIN,LOW,INTERV,
NINTAN,MD,
$, INTER,NINT,IND,IERR)
N1=IND-IEX1+1
WRITE(*,*) '
WRITE(*,*) 'DONNEZ LE MOIS ET L'ANNEE TERMINANT
LA '
WRITE(*,*) 'SOUS-SERIE : '
WRITE(*,*) '
WRITE(*, '(AB\)' ) 'MOIS : '
READ(*,80)MO
WRITE(*, '(A11\)' ) 'ANNEE : 19'
READ(*,80)AN
IF (AN.LT.ANTMP.OR.AN.GT.ANFIN.OR.MO.LT.1.OR.MO.
$, GT.12.OR.
$, (AN.EQ.ANTMP.AND.MO.LE.MOTMP)) THEN
WRITE(*,*) '
WRITE(*,*) ' *** DATE INVALIDE OU
$, INCOMPATIBLE AVEC LA
$, PREMIERE DATE ENTREE. *** '
WRITE(*,*) ' *** RECOMMENCEZ *** '
GOTO 75
ENDIF
CALL TRANSFRANG(IDIV,AN,MO,JR,IBIN,LOW,INTERV,
$, NINTAN,MD,
$, INTER,NINT,IND,IERR)
N2=IND-IEX1+1
ENDIF
ENDIF
ENDIF
```


50
c *** validation
\$
\$
\$
61
c *** validation
\$
\$
\$
\$

```
READ(*,50)JR
WRITE(*,'(A8\)' )'MOIS : '
READ(*,50)MO
WRITE(*,'(A11\)' )'ANNEE : 19'
READ(*,50)AN
FORMAT(I2)

IF (AN.LT.LOW.OR.AN.GT.ANFIN.OR.MO.LT.1.OR.MO.GT.
12.OR.J
R.LT.1.OR.JR.GT.31) THEN
WRITE(*,*)'
WRITE(*,*)' *** DATE INVALIDE.RECOMMENCEZ
. ***'
GOTO 49
ENDIF

c *** transformer la date en rang i.e. en indice vectorielle correspondant

CALL TRANSFRANG(IDIV,AN,MO,JR,IBIN,LOW,INTERV,
NINTAN,MD,
INTER,NINT,IND,IERR)

c *** ajuster la borne n1 a l'aide de la valeur iex1 (premier intervalle
c *** non-absent)

N1=IND-IEX1+1
ANTMP=AN
MOTMP=MO
JRTMP=JR
WRITE(*,*)'
WRITE(*,*)'DONNEZ LA DATE TERMINANT LA SOUS-
SERIE : '
WRITE(*,*)'
WRITE(*,'(A8\)' )'JOUR : '
READ(*,50)JR
WRITE(*,'(A8\)' )'MOIS : '
READ(*,50)MO
WRITE(*,'(A11\)' )'ANNEE : 19'
READ(*,50)AN

c *** validation

IF (AN.LT.ANTMP.OR.AN.GT.ANFIN.OR.MO.LT.1.OR.MO.
GT.12.OR
.JR.LT.1.OR.JR.GT.31.OR.(AN.EQ.ANTMP.AND.(MO.LT.
MOTMP.OR.(MO.EQ.MO
TMP.AND.JR.LT.JRTMP)))) THEN
WRITE(*,*)'
WRITE(*,*)' *** DATE INVALIDE OU
INCOMPATIBLE AVEC LA
PREMIERE DATE ENTREE. ***'
WRITE(*,*)' *** RECOMMENCEZ ***'
GOTO 49
ENDIF

CALL TRANSFRANG(IDIV,AN,MO,JR,IBIN,LOW,INTERV,
NINTAN,MD,
INTER,NINT,IND,IERR)
N2=IND-IEX1+1
```

c *** ouverture du fichier de sortie approprié
c *** transfert des variables globales ainsi que des données composant
c *** la sous-série

c *** on enlève le trou si on a tronqué l'année

```
IF (ITRONC.EQ.1) N2=N2-IDURTR*NINTAN  
90 WRITE(2, '(E16.10) ') (VECTEUR(I), I=N1, N2)
```

c *** sortie si on a tronqué la série

```
IF (ITRONC.EQ.1) GOTO 125  
CALL QCLEAR(ITBACK, ITFORE)  
CALL QCMOV(0, 23)
```

c *** afficher la structure de l'arbre de décision

```
WRITE(*,*) ' '
WRITE(*,*) ' ARBRE DE DECISION'
WRITE(*,*) ' -----'
WRITE(*,*) '
WRITE(*,*) ' TENDANCE PERSISTANCE SAISONNALITE TEST
# APPROPRIE'
WRITE(*,*) ' -----
#
WRITE(*,*) '
WRITE(*,*) ' Tendance --> PERSISTANCE - Pas de saison LETT
# ENMAIER/SPEARMAN'
WRITE(*,*) ' monotone markovienne'
WRITE(*,*) ' - Avec Saisons HIRS
# CH et SLACK'
WRITE(*,*) '
WRITE(*,*) ' Pas de - Pas de saison SPEA
# RMAN/KENDALL'
WRITE(*,*) ' PERSISTANCE'
WRITE(*,*) ' - Avec saisons KEND
# ALL SAISONNIER'
WRITE(*,*) '
WRITE(*,*) ' Tendance --> PERSISTANCE - Pas de saison LETT
# ENMAIER/MANN-WHITNEY'
WRITE(*,*) ' par saut markovienne'
WRITE(*,*) ' - Avec saisons MANN
# -WHITNEY SAIS./LETTEN.'
WRITE(*,*) '
WRITE(*,*) ' Pas de - Pas de saison MANN
# -WHITNEY'
WRITE(*,*) ' PERSISTANCE'
WRITE(*,*) ' - Avec saisons MANN
# -WHITNEY SAISONNIER'
WRITE(*,*) '
write(*,*) '
CALL QCMOV(1,1)
CALL QCHAR(27, ITFORE, 1)
CALL QBEEP
```

READ(*, '(a1)') TOUCHE

c *** diriger l'usager vers le test approprié

```
IF (TEND.EQ.'1') THEN
  IF (IPERS.EQ.1) THEN
    IF ((ISAIS.EQ.1.AND.IDIV.EQ.1).OR.ISAIS.EQ.3) THEN
      NTEST='HIRSCH ET SLACK'
      NO=8
    ELSE
      NTEST='LETTENMAIER/SPEARMAN'
      NO=6
    ENDIF
  ELSE
    IF ((ISAIS.EQ.1.AND.IDIV.EQ.1).OR.ISAIS.EQ.3) THEN
      NTEST='KENDALL SAISONNIER'
      NO=7
    ELSE
      NTEST='KENDALL'
      NO=5
    ENDIF
  ENDIF
ELSE
  IF (IPERS.EQ.1) THEN
    IF ((ISAIS.EQ.1.AND.IDIV.EQ.1).OR.ISAIS.EQ.3) THEN
      NTEST='MANN-WHITNEY SAIS/LETTENMAIER'
      NO=4
    ELSE
      NTEST='LETTENMAIER/MANN-WHITNEY'
      NO=2
    ENDIF
  ELSE
    IF ((ISAIS.EQ.1.AND.IDIV.EQ.1).OR.ISAIS.EQ.3) THEN
      NTEST='MANN-WHITNEY SAISONNIER'
      NO=3
    ELSE
      NTEST='MANN-WHITNEY'
      NO=1
    ENDIF
  ENDIF
ENDIF
```

NOSUGG=NO

CALL @CLEAR(ITBACK,ITFORE)

CALL @CMOV(0,23)

WRITE(*,*) ' TEST PROPOSE : ',NTEST

120

WRITE(*,*) ' '

WRITE(*,*) ' DESIREZ-VOUS : '

WRITE(*,*) ' '

WRITE(*,*) ' 1) UTILISER CE TEST '

WRITE(*,*) ' 2) UTILISER UN TEST DIFFERENT '

WRITE(*,*) ' '

WRITE(*,5) 'VOTRE CHOIX : '

READ(*,10)ITEST

IF (ITEST.NE.'1'.AND.ITEST.NE.'2') THEN

TEXTMD(45:56)=TEXTMM(45:56)

c *** lecture de la serie

READ(1, '(E16.10)') (VECTEUR(I), I=1, LS)

c *** ouverture du fichier de sortie

```
FICH(LB+3:LB+3)='P'  
IF (ITYP.EQ.0) THEN  
  OPEN(2, FILE=FICH, STATUS='NEW')  
ELSE  
  OPEN(2, FILE=FICH, STATUS='NEW')  
ENDIF
```

c *** mise en mode texte

```
CALL QSMODE(ITMODE)  
CALL QCLEAR(ITBACK, ITFORE)  
CALL QBORD(ITCOLOR)  
CALL QCMOV(0, 23)
```

c *** cas ou l'on a elimine certaines annees

c *** le test est choisi automatiquement; on ne fait pas les graphiques.

```
1  FORMAT(18X, A44)  
   IF (ITRONC.EQ.1) THEN  
     IF (IPERS.EQ.1) THEN  
       IF (ISAIS.EQ.1.AND.IDIV.EQ.1) THEN  
         NTEST='MANN-WHITNEY SAIS/LETTENMAIER'  
         WRITE(*, 1) '***TEST DE MANN-WHITNEY SAIS/LETTENMAIE  
           R ***'  
         WRITE(*, 1) '***          CHOISI AUTOMATIQUEMENT  
           ***'  
         WRITE(*, *) '  
         WRITE(*, 1) '*** AVEC SAUT A L"EMPLACEMENT DES  
           ANNEES ***'  
         WRITE(*, 1) '*** ELIMINEES  
           ***'  
         NO=4  
       ELSE  
         NTEST='LETTENMAIER/MANN-WHITNEY'  
         WRITE(*, 1) '*** TEST DE LETTENMAIER/MANN-WHITNEY  
           ***'  
         WRITE(*, 1) '***          CHOISI AUTOMATIQUEMENT  
           ***'  
         WRITE(*, *) '  
         WRITE(*, 1) '*** AVEC SAUT A L"EMPLACEMENT DES  
           ANNEES ***'  
         WRITE(*, 1) '*** ELIMINEES  
           ***'  
         NO=2  
       ENDIF  
     ELSE  
       IF (ISAIS.EQ.1.AND.IDIV.EQ.1) THEN  
         NTEST='MANN-WHITNEY SAISONNIER'
```

```
WRITE(*,1)'*** TEST DE MANN-WHITNEY SAISONNIER
***'
WRITE(*,1)'*** CHOISI AUTOMATIQUEMENT
***'
WRITE(*,*)'
WRITE(*,1)'*** AVEC SAUT A L"EMPLACEMENT DES
ANNEES ***'
WRITE(*,1)'*** ELIMINEES
***'
NO=3
ELSE
NTEST='MANN-WHITNEY'
WRITE(*,1)'*** TEST DE MANN-WHITNEY
***'
WRITE(*,1)'*** CHOISI AUTOMATIQUEMENT
***'
WRITE(*,*)'
WRITE(*,1)'*** AVEC SAUT A L"EMPLACEMENT DES
ANNEES ***'
WRITE(*,1)'*** ELIMINEES
***'
NO=1
ENDIF
ENDIF
NOSUGG=NO
ITEST='1'
c *** on s'en va offrir de couper la serie aux extremités
GOTO 42
ENDIF
c *** tracage du graphique de la fonction double-masse
CALL DOUBMASS(VECTEUR,TEXTC,TEXTT,TEXTMD,LS,TAMP1,TAMP2)
c *** tracage du graphique de la fonction cusum
CALL CUSUM(VECTEUR,T,TEXTC,TEXTT,TEXTMM,LS,TAMP1,TAMP2)
CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
CALL QCMOV(0,23)
c *** on demande le genre de tendance que l'on souhaite traiter
3 WRITE(*,*)'
WRITE(*,*)' SELON QUEL GENRE DE TENDANCE DESIREZ VOUS TRAITER'
WRITE(*,*)' LA SERIE ?'
WRITE(*,*)'
WRITE(*,*)' 1) TENDANCE MONOTONE'
WRITE(*,*)' 2) TENDANCE PAR SAUT'
WRITE(*,*)'
WRITE(*,5)'VOTRE CHOIX : '
5 FORMAT(5X,A14\)
```

```
        WRITE(*,20)  
        CALL QBEEP  
        CALL QBEEP  
        GOTO 120  
    _ENDIF
```

c *** ecrire les bornes choisies dans le fichier "ident.tmp"

```
125      WRITE(3, '(I4,1X,I4,1X,I4)') N1,N2,N2-N1+1  
        IF (ITEST.EQ.'1') THEN
```

c *** l'utilisateur desire utiliser le test suggere
c *** ouvrir un fichier-vidé portant le numero du test

```
        WRITE(3, '(I2,1X,I2)') NOSUGG,NO  
        IF (NO.LE.9) THEN  
            NOC=CHAR(NO+48)  
            FICHTEST(1:1)=NOC  
            FICHTEST(2:6)='.TST'  
        ELSE  
            FICHTEST(1:1)='1'  
            NOC=CHAR(NO+38)  
            FICHTEST(1:2)=NOC  
            FICHTEST(3:6)='.TST'  
        ENDIF  
    ELSE
```

c *** l'utilisateur desire utiliser un test different que celui qui a ete
c *** suggere

```
122      CALL @CLEAR(ITBACK,ITFORE)  
        CALL @CMOV(0,23)  
        NOC=''  
        WRITE(*,*)'  
        WRITE(*,*)'          TESTS DISPONIBLES'  
        WRITE(*,*)'  
        WRITE(*,*)'  
        WRITE(*,*)' [1] MANN-WHITNEY'  
        WRITE(*,*)' [2] MANN-WHITNEY/LETTENMAIER'  
        WRITE(*,*)' [3] MANN-WHITNEY SAISONNIER'  
        WRITE(*,*)' [4] MANN-WHITNEY SAIS/LETTENMAIER'  
        WRITE(*,*)' [5] KENDALL'  
        WRITE(*,*)' [6] SPEARMAN/LETTENMAIER'  
        WRITE(*,*)' [7] KENDALL SAISONNIER'  
        WRITE(*,*)' [8] HIRSCH ET SLACK'  
        WRITE(*,*)' [9] FOSTER ET STUART 1'  
        WRITE(*,*)' [10] FOSTER ET STUART 2'  
        WRITE(*,*)' [11] TAU DE KENDALL'  
        WRITE(*,*)' [12] SPEARMAN'  
        WRITE(*, '(/,/,/)' )  
        WRITE(*, '(A35\)' ) 'ENTREZ LE NUMERO DU TEST DESIRE : '  
        READ(*, '(A2)' ) NOC
```

c *** validation de l'entree
 IF (NOC.NE.'1'.AND.NOC.NE.'2'.AND.NOC.NE.'3'.AND.NOC.NE.'4'.AND

```
      READ(*,10)TEND
10     FORMAT (A1)
      IF (TEND.NE.'1'.AND.TEND.NE.'2') THEN
          WRITE(*,20)
          GOTO 3
      ENDIF
20     FORMAT(/,' *** REPONSE INVALIDE ***')
40     CALL QCLEAR(ITBACK,ITFORE)
      CALL QCMOV(0,23)
42     WRITE(*,*)' '
      WRITE(*,*)'     VOUS DESIREZ TRAITER : '
      WRITE(*,*)' '
      WRITE(*,*)'           1) TOUTE LA SERIE'
      WRITE(*,*)'           2) UNE PARTIE SEULEMENT'
      WRITE(*,*)' '
      WRITE(*,5)'VOTRE CHOIX : '
      READ(*,10)ETD
      IF (ETD.NE.'1'.AND.ETD.NE.'2') THEN
          WRITE(*,20)
          GOTO 42
      ENDIF
      IF (ETD.EQ.'1') THEN
c *** on veut traiter la serie au complet : les bornes choisies sont donc
c *** egales au extremités de la serie
          N1=1
          N2=LS
          GOTO 90
      ELSE
c *** l'usager prefere specifier des bornes
45         CALL QCLEAR(ITBACK,ITFORE)
          CALL QCMOV(0,23)
47         WRITE(*,*)' '
          WRITE(*,*)'     VOUS PREFEREZ DETERMINER LES EXTREMITES DE'
          WRITE(*,*)'     LA SOUS-SERIE EN SPECIFIANT : '
          WRITE(*,*)' '
          WRITE(*,*)'           1) LES RANGS AUX EXTREMITES'
          WRITE(*,*)'           2) LES DATES AUX EXTREMITES'
          WRITE(*,*)' '
          WRITE(*,5)'VOTRE CHOIX : '
          READ(*,10)DR
          IF (DR.NE.'1'.AND.DR.NE.'2') THEN
              WRITE(*,20)
              GOTO 47
          ENDIF
c *** n1 et n2 contiendront les valeurs des indices de la premiere et
c *** de la derniere valeur de la sous-serie (portion traitee) a partir
c *** de iex1 (i.e. du premier intervalle non-absent)
          CALL QCLEAR(ITBACK,ITFORE)
          CALL QCMOV(0,23)
```

```

$      .NOC.NE.'5'.AND.NOC.NE.'6'.AND.NOC.NE.'7'.AND.NOC.NE.'8'.AND.
$      NOC.
$      NE.'9'.AND.NOC.NE.'10'.AND.NOC.NE.'11'.AND.NOC.NE.'12') THEN
$          CALL QBEEP
$          GOTO 122
$      ENDIF
c *** interdire les tests saisonniers si on n'a qu'un intervalle par
c *** annee
$      IF (NINTAN.EQ.1.AND.(NOC.EQ.'3'.OR.NOC.EQ.'4'.OR.NOC.EQ.'7'))
$      THEN
$          WRITE(*,*)' '
$          WRITE(*,*)' *** UNE SEULE DONNEE PAR ANNEE ***'
$          WRITE(*,*)'*** CHOISISSEZ UN TEST NON-SAISONNIER ***'
$          GOTO 122
$      ENDIF
$      IF (NOC.EQ.'10'.OR.NOC.EQ.'11'.OR.NOC.EQ.'12') THEN
$          WRITE(3,'(I2,1X,A2)')NOSUBB,NOC
$      ELSE
$          WRITE(3,'(I2,2X,A2)')NOSUBB,NOC
$      ENDIF
c *** ouvrir un fichier-vidé portant le numero du test
$      IF (NOC(2:2).EQ.' ') THEN
$          FICHTEST(1:1)=NOC
$          FICHTEST(2:6)='.TST '
$      ELSE
$          FICHTEST(1:2)=NOC
$          FICHTEST(3:6)='.TST '
$      ENDIF
$      ENDIF
$      OPEN(4,FILE=FICHTEST,STATUS='NEW')
$      CLOSE(1)
$      CLOSE(2)
$      CLOSE(3)
$      CLOSE(4)
c *** ouvrir un fichier bidon indiquant que l'exécution de p5 s'est
c *** déroulée normalement
$      OPEN(99,FILE='P5.OK',STATUS='NEW')
$      CLOSE(99)
$      END

```

```

C      SUBROUTINE DOUBMASS(C,TEXTC,TEXTQ,TEXTMM,LS,SC,SO)
C      * * * * *
C      -----
C      Fonction:
C      . Tracage de la courbe double-masse pour la serie equidistante
C      Parametres:

```



```
c
c      . c          Vecteur contenant la serie a etre utilisee en
c          ordonnee
c      . textc      Texte a etre inscrit sur l'ordonnee
c      . textq      Texte a etre inscrit sur l'abscisse
c      . textmm     Titre du graphique
c      . ls         Longueur de la serie
c      . sc         # Vecteur des sommes cumulees de "c"
c      . sq         # Vecteur des sommes cumulees de "q"
c
```

```
-----
INTEGER ANFIN
REAL C(*),SC(*),SQ(*)
CHARACTER*1 TOUCHE
CHARACTER*10 TEXTEX,TEXTEY
CHARACTER*12 FICHDEB,NOMFICH
COMMON /TEXTE /ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH /IGMODE,IGBACK,IPAL,IGCOLOR,ILINE,ISEC,IGBORD
#      ,KIRSYM,ISYMBL,ILETR,JROW1,JROW2,JCOL1,JCOL2
```

```
TEXTEX='
TEXTEY='
SC(1)=C(1)
SQ(1)=1
```

```
DO 10 I=1,LS-1
      SC(I+1)=SC(I)+C(I+1)
      SQ(I+1)=I+1
10 CONTINUE
```

c *** configuration en mode graphique

```
CALL QSMODE(IGMODE)
IF (IGMODE.LT.10) THEN
      CALL QCOLOR(IGBACK,IPAL)
ELSE
      CALL QOVSCN(IGBORD)
ENDIF
```

c *** construction du graphe de la fonction double-masse

```
CALL TICS(0.0,SQ(LS),XSTS,XFINS,TICX,MINORX,AINTEX)
CALL TICS(0.0,SC(LS),YSTS,YFINS,TICY,MINORY,AINTEY)
IF (AINTEX.GE.10.OR.AINTEY.LT.1) THEN
      CALL AJUSTE(SQ,LS,AINTEX,TEXTEX,XSTS,XFINS,TICX)
ENDIF
IF (AINTEY.GE.10.OR.AINTEY.LT.1) THEN
      CALL AJUSTE(SC,LS,AINTEY,TEXTEY,YSTS,YFINS,TICY)
ENDIF
CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,XSTS,XFINS,YSTS,YFINS,
#      XSTS,YSTS,0,1.0,1.5)
CALL QXAXIS(XSTS,XFINS,TICX,MINORX,1,0)
CALL QYAXIS(YSTS,YFINS,TICY,MINORY,1,0)
CALL QGTXT(10,TEXTEX,ILETR,JCOL2-35,JROW1+10,0)
CALL QPTXTC(10,TEXTEY,ILETR)
CALL QSETUP(0,ILINE,ISYMBL,KIRSYM)
```

```
CALL QTABL(1,LS,SQ,SC)
CALL QGTXT(56,TEXTMM,ILETR,JCOL1+60,JROW2+12,0)
CALL QPTXTA(21,TEXTQ,ILETR)
IF (IGMODE.EQ.16) THEN
    INCR=85
ELSE
    INCR=55
ENDIF
CALL QGTXT(15,TEXTC,ILETR,0,(JROW2+JROW1)/2+INCR,-1)
```

c *** tracage de la droite

```
CALL QRTDI(0.0,0.0,I1,J1)
CALL QRTDI(SQ(LS),SC(LS),I2,J2)
CALL QLINE(I1,J1,I2,J2,ILINE)
```

c *** determination de la pente

```
P=SC(LS)/SQ(LS)
```

c *** calcul des ecartis-types relies aux rails

```
R1=0
R2=0
NA=0
NB=0
DO 20 I=1,LS
    PP=P*SQ(I)
    IF (SC(I).GT.PP) GOTO 15
    R2=R2+(SC(I)-PP)**2
    NB=NB+1
    GOTO 20
15    R1=R1+(SC(I)-PP)**2
    NA=NA+1
20    CONTINUE
    IF (NA.LE.1) THEN
        RA1=0.0
    ELSE
        RA1=(R1/(NA-1))**0.5
    ENDIF
    IF (NB.LE.1) THEN
        RA2=0.0
    ELSE
        RA2=(R2/(NB-1))**0.5
    ENDIF
```

c *** construction des rails

```
CTE1=1.96*RA1
CTE2=1.96*RA2/P
CTE3=SQ(LS)-(1.96*RA1/P)
CTE4=SC(LS)-(1.96*RA2)
CALL QRTDI(0.0,CTE1,I1,J1)
CALL QRTDI(CTE2,0.0,I2,J2)
CALL QRTDI(CTE3,SC(LS),I3,J3)
```

```
CALL QRTOI(SQ(LS),CTE4,I4,J4)
CALL QLINE(I1,J1,I3,J3,ISEC)
CALL QLINE(I2,J2,I4,J4,ISEC)
CALL QLINE(I4,J2,I4,J3,ILINE)
CALL QLINE(I1,J3,I4,J3,ILINE)
CALL QCMDV(1,1)
IF(IMPRIM.EQ.1)CALL QPSCRN
CALL QCHAR(27,ITFORE,1)
CALL QBEEP
29 READ(*,30) TOUCHE
30 FORMAT(A1)
RETURN
END
```

SUBROUTINE CUSUM(C,TC,TEXTC,TEXTT,TEXTMM,LS,TCU,VCUSUM)

Fonction:

. Tracage de la courbe CUSUM pour la serie equidistante

Parametres:

- . c Vecteur contenant les valeurs utilisees en ordonnee
- . tc Vecteur contenant les valeurs utilisees en abscisse
- . textc Texte a etre inscrit sur l'ordonnee
- . textt Texte a etre inscrit sur l'abscisse
- . textmm Titre du graphique
- . ls Longueur de la serie
- . tcu # Vecteur des valeurs de l'abscisse
- . vcusum # Vecteur des valeurs de l'ordonnee

```
INTEGER ANFIN
CHARACTER*10 TEXTEX,TEXTEY
CHARACTER*12 FICHDEB,NOMFICH
REAL C(*),TC(*),TCU(*),VCUSUM(*),MAXCU,MINCU
COMMON /TEXTE /ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH /IGMODE,IGBACK,IPAL,IGCOLOR,ILINE,ISEC,IGBORD
$ ,KIRSYM,ISYMBL,ILETR,JROW1,JROW2,JCOL1,JCOL2
```

TEXTEX=' '
TEXTEY=' '

c *** calcul de la moyenne de la serie

```
CM=0.0
MINCU=1E30
MAXCU=-1E30
DO 10 I=1,LS
  CM=CM+C(I)
CONTINUE
CM=CM/LS
```

c *** calcul de la fonction cusum

```
NOB=LS+1
TCU(1)=0
TCU(2)=1
VCUSUM(1)=0
VCUSUM(2)=C(1)-CM
MAXCU=VCUSUM(2)
MINCU=VCUSUM(2)
DO 15 I=3,NOB
    TCU(I)=I-1
    VCUSUM(I)=VCUSUM(I-1)+C(I-1)-CM
    IF (VCUSUM(I).GT.MAXCU) MAXCU=VCUSUM(I)
    IF (VCUSUM(I).LT.MINCU) MINCU=VCUSUM(I)
15 CONTINUE
```

c *** configuration en mode graphique

```
CALL QSMODE(IGMODE)
IF (IGMODE.LT.10) THEN
    CALL QCOLOR(IGBACK,IPAL)
ELSE
    CALL QOVSCN(IGBORD)
ENDIF
CALL TICS(MINCU,MAXCU,YSTS,YFINS,TICY,MINDRY,AINTEY)
CALL TICS(0.0,TCU(NO),XSTS,XFINS,TICX,MINDRX,AINTEX)
IF (AINTEX.GE.10.OR.AINTEX.LT.1) THEN
    CALL AJUSTE(TCU,NOB,AINTEX,TEXTEX,XSTS,XFINS,TICX)
ENDIF

IF (AINTEY.GE.10.OR.AINTEY.LT.1) THEN
    CALL AJUSTE(VCUSUM,NOB,AINTEY,TEXTXY,YSTS,YFINS,TICY)
ENDIF

CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,XSTS,XFINS,YSTS,YFINS,
# XSTS,YSTS,0,1.0,1.5)
CALL QXAXIS(XSTS,XFINS,TICX,MINDRX,1,0)
CALL QYAXIS(YSTS,YFINS,TICY,MINDRY,1,0)
CALL QSETUP(0,ILINE,ISYMBL,KIRSYM)
CALL QBTXT(10,TEXTEX,ILETR,JCOL2-35,JROW1+10,0)
CALL QPTXTC(10,TEXTXY,ILETR)
CALL QBTXT(56,TEXTMM,ILETR,JCOL1+60,JROW2+12,0)
CALL QPTXTA(21,TEXTT,ILETR)
IF (IGMODE.EQ.16) THEN
    INCR=85
ELSE
    INCR=55
ENDIF
CALL QBTXT(15,TEXTC,ILETR,0,(JROW2+JROW1)/2+INCR,-1)
```

c *** construction du graphe de la fonction cusum

```
CALL QTABL(1,NOB,TCU,VCUSUM)
IF (AINTEY.GE.10.OR.AINTEY.LT.1) THEN
    MINCU=MINCU/AINTEY
    MAXCU=MAXCU/AINTEY
```

---ENDIF

```
CALL ORTOI (O.O,MINCU,IA,JA)
CALL ORTOI (TCU(NO),MAXCU,IB,JB)
CALL ORTOI (O.O,O.O,IA,JC)
CALL OLINE (IA,JB,IB,JB,ILINE)
CALL OLINE (IB,JA,IB,JB,ILINE)
CALL OLINE (IA,JC,IB,JC,ILINE)
CALL QCMOV (1,1)
CALL QCHAR (27,ITFORE,1)
CALL QBEEP
29 READ (*,30)TOUCHE
30 FORMAT (A1)
99 CONTINUE
100 RETURN
END
```

```
SUBROUTINE TRANSFRANG (IDIV,AN,MO,JR,IBIN,LOW,INTERV,NINTAN,MD,
* INTER,NINT,IND,IERR)
```

C * * * * *

C Fonction:

C . Transforme une date donnees en argument en indice vectoriel
C correspondant (a partir du 1er janvier de la 1ere annee)
C Aiguille vers la bonne sous-routine selon la facon dont
C l'usager a decide de diviser sa serie

C Parametres :

C . idiv Indique la methode employee precedemment pour diviser
C la serie
C . an,mo,jr Date que l'on desire transformer en rang (annee,mois
C et jour)
C . ibin Indique si l'interval est en jour ou en mois
C . low Anne de depart de la serie
C . interv Intervalle choisi
C . nintan Nombre d'intervalles par annee
C . md mois de depart,cas de division saisonniere
C . inter Intervalle,cas de division saisonniere
C . nint Nombre d'intervalles par annee,cas de division saisonniere
C . ind Donne en sorte l'indice vectoriel correspondant a
C la date recherchee
C . ierr Code mis a "1" si la date tombe dans un intervalle
C non-considere par la methode employee

INTEGER AN,MO,JR

---IF (IDIV.EQ.0) THEN

C *** on a utilise la division standard en jours/mois

```
CALL RANGNORM(AN,MO,JR,IBIN,LOW,INTERV,NINTAN,IND,IERR)
ELSEIF (IDIV.EQ.1) THEN
c *** on a utilise la division saisonniere
CALL RANGSAIS(AN,MO,LOW,MD,INTER,NINT,IND,IERR)
ELSE
c *** on a utilise la troncation de certains mois
CALL RANGTRONQ(AN,MO,LOW,MD,INTER,NINT,IND,IERR)
ENDIF
RETURN
END
```

```
SUBROUTINE RANGNORM(AN,MO,JR,IBIN,LOW,INTERV,NINTAN,IND,IERR)
*****
```

c
c Fonction:
c
c . Convertit une date donnee en rang correspondant dans le vecteur
c debutant au 1er janvier de la premiere annee.Cas ou l'usager a
c choisi une division simple de sa serie en jours/mois
c
c Parametres:
c
c S.V.P. Referer a la sous-routine "transrang"
c
c notes: le code ierr est mis a 1 si la date tombe dans le dernier
c intervalle et que celui-ci est incomplet
c
c -----

```
INTEGER AN,MO,JR,NBJRS,JPARM(12)
DATA JPARM/31,28,31,30,31,30,31,31,30,31,30,31/

IERR=0
IF (IBIN.EQ.0) THEN
c *** l'intervalle est donne en jours
NBJRS=0
DO 10 I=1,MO-1
NBJRS=NBJRS+JPARM(I)
10 CONTINUE
c *** calcul de l'indice correspondant
IND=(AN-LOW)*NINTAN+((NBJRS+JR-1)/INTERV+1)
c *** si la date tombe dans le dernier intervalle et que ce dernier
c *** est incomplet, on ramene l'intervalle au precedent.
```

```

      IF (IND.GT.((AN-LOW+1)*NINTAN)) THEN
        IND=IND-1
        IERR=1
      ENDIF
    ELSE
c *** l'intervalle est donne en mois
      IND=(AN-LOW)*NINTAN+((MO-1)/INTERV+1)
    ENDIF
  RETURN
  END
```

```

SUBROUTINE RANGSAIS(AN,MO,LOW,MD,INTER,NINT,IND,IERR)
*****
```

Fonction:

. Convertit une date donnee en rang correspondant dans le vecteur
debutant au 1er janvier de la premiere annee.Cas ou l'usager a
choisi une division saisonniere de sa serie
On ne tient pas compte du jour dans le mois

Parametres:

S.V.P. Referer a la sous-routine "transrang"

note: le code ierr est mis a 1 si la date tombe avant le premier
intervalle

```

      INTEGER AN,MO
```

```

      IERR=0
```

```

c *** calcul de l'indice vectoriel correspondant
```

```

      IND=(AN-LOW)*NINT+((MO+INTER-MD)/INTER)
```

```

c *** si on tombe avant le premier intervalle, on ramene celui-ci au
c *** premier
```

```

      IF (IND.EQ.0) THEN
        IND=1
        IERR=1
      ENDIF
  RETURN
  END
```

```

SUBROUTINE RANGTRONQ(AN,MO,LOW,MD,INTER,NINT,IND,IERR)
```

c * * * * *

c -----

c Fonction:

c . Convertit une date donnee en rang correspondant dans le vecteur
c debutant au 1er janvier de la premiere annee. Cas ou l'usager a
c choisi de tronquer certains mois de sa serie
c On ne tient pas compte du jour dans le mois

c Parametres:

c S.V.P. Referer a la sous-routine "transrang"

c note : si la date tombe parmi les mois tronques, on ramene au pre-
c mier intervalle non-tronque qui suit, et on indique cette manoeuvre
c en mettant le code ierr a 1.

c -----

INTEGER AN,MO

IERR=0

IF (MD+NINT.LE.12) THEN

c *** les mois conserves tiennent dans une meme annee

IF (MO.LT.MD) THEN

c *** on tombe dans un mois tronque, au ddebut de l'annee

MO=MD
IERR=1

ELSEIF (MO.GT.MD+NINT) THEN

c *** on tombe dans un mois tronque, a la fin de l'annee

MO=MD+NINT
IERR=1

ENDIF

c *** calcul de l'indice vectoriel correspondant

IND=(AN-LOW)*NINT+(MO-MD+1)

ELSE

c *** les mois conserves sont "a cheval" sur la fin de l'annee et le
c *** debut de la suivante

IF (MO.LT.MD.AND.MO.GT.(MD+NINT-13)) THEN

c *** on tombe dans un mois tronque

MO=MD
IERR=1


```
      ENDIF  
c *** calcul de l'indice vectoriel correspondant  
      IF (MO.GE.MD) THEN  
          IND=(AN-LOW)*NINT+(MO+NINT-12)  
      ELSE  
          IND=(AN-LOW)*NINT+MO  
      ENDIF  
    ENDIF  
  RETURN  
END
```

```
$INCLU  DE: 'TICS.FOR'  
$INCLU  DE: 'AJUSTE.FOR'
```

FS.FOR

QSMODE, QCLEAR, QBORD, FICH, NOMFICH, FLOAT
SORT, LOG10, ELSEIF, LOG, CHAR

FOSTUART FS.FOR 65
65 CALL FOSTUART(VECDON, LONG, D, S)

FS.FOR 181
SUBROUTINE FOSTUART(VECDON, N, D, S)

NDTR FS.FOR 79 x2
79 CALL NDTR(FS1, P, DI)
133 CALL NDTR(FS2, P, DI)

NDTR.FOR 2
Subroutine ndtr(x, p, d)

abs, exp, t

#STORA GE:2

PROGRAM FOSTERSTUART

```
C -----
C *****
C Titre : Test de Foster & Stuart
C
C Fonction :
C
C . Teste l'indépendance des observation de la serie
C . Sortie : valeur de la statistique,valeur du test,niveau de
C   signification du test,commentaire sur la tendance observee
C *****
C
C   INTEGER ANFIN,D,S,MAXI,NCRIT(9)
C   REAL FCRIT(9)
C   PARAMETER (MAXI=1300)
C   CHARACTER*10 F
C   CHARACTER*12 FICH,FICHDEB,NOMFICH
C   CHARACTER*39 COMMEN
C   REAL VECDON(MAXI)
C   DATA NCRIT/10,25,50,75,100,150,200,300,1000/
C
C *** lecture des specifications graphiques
C
C   OPEN(4,FILE='SETUP.PC',STATUS='OLD')
C   READ(4,'(4I2)')ITMODE,ITFORE,ITBACK,ITCOLOR
C   CALL QSMODE(ITMODE)
C   CALL QCLEAR(ITBACK,ITFORE)
C   CALL QBORD(ITCOLOR)
C   CLOSE(4)
C
C *** lecture du fichier ident.tmp
C
C   OPEN(2,FILE='IDENT.TMP',STATUS='OLD')
C   READ(2,2)NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,LOW,MODEF,JODEF
C   ,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER,ISAIS,MD,INTER,NINT,
C   ,ICQ,ACQ,BCQ,ITRONC,IANTR,IDURTR,INTERV,IBIN,NINTAN,N,IEX1,IEX2,LS,
C   ,IDIV,IBT,IPERS,RHO,N1,N2,LONG,NOSUGG,NOTEST
C 2   FORMAT(A12,1X,I1,/,A12,1X,I1,/,I1,1X,I1,/,I2,1X,I2,1X,I2,/,I2,1X,
C   ,I2,1X,I2,/,I4,/,I2,1X,I2,/,I1,1X,I2,1X,I2,1X,I2,/,I1,1X,F9.2,1X,
C   ,F9.2,/,I1,1X,I2,1X,I2,/,I2,1X,I1,1X,I3,1X,I4,/,I4,1X,I4,1X,I4,1X,
C   ,I1,1X,I1,/,I1,1X,F5.2,/,I4,1X,I4,1X,I4,/,I2,1X,I2)
C
C *** ouverture du fichier de travail et lecture de la serie (sous-serie)
C *** a traiter
C
C   FICH='
C   FICH(1:LG)=NOMFICH(1:LG)
C   IF (ITYP.EQ.0) THEN
C     FICH(LG+1:LG+4)='.TPC'
```

```
ELSE  
    FICH(LG+1:LG+4)='.TPL'  
ENDIF  
5  FORMAT(E16.10)  
    OPEN(1,FILE=FICH,STATUS='OLD')  
    READ(1,5)(VECDON(I),I=1,LONG)
```

c *** calcul de la statistique

```
CALL FOSTUART(VECDON, LONG, D, S)  
IF (NOTEST.EQ.10) GOTO 100  
VLONG=FLOAT(LONG)  
IF (LONG.LT.130.AND.LONG.GE.8) THEN  
    SDIND=.48+(2.78*SQRT(LOG10(VLONG)))  
ELSEIF (LONG.GE.130) THEN  
    SDIND=.5+(2*(2*LOG(VLONG)-0.846))*0.5  
ELSE  
    WRITE(*,*) ' LA LONGUEUR DE LA SERIE EST TROP PETITE '  
    WRITE(*,*) ' POUR UNE UTILISATION VALABLE DE CE TEST '  
ENDIF  
FS1=D/SDIND
```

c *** determination du niveau de signification du test

```
CALL NDTR(FS1,P,DI)  
IF (P.LT.0.05) THEN  
    COMMEN='TENDANCE DECROISSANTE DETECTEE'  
    ITEND=1  
ELSEIF (P.GT.0.95) THEN  
    COMMEN='TENDANCE CROISSANTE DETECTEE'  
    ITEND=1  
ELSE  
    COMMEN='PAS DE TENDANCE DETECTEE'  
    ITEND=0  
ENDIF
```

c *** affichage des resultats

```
WRITE(*,*) '  
WRITE(*, '(/,/,/,35X,A9)') 'RESULTATS'  
WRITE(*, '(35X,A9)') '-----'  
WRITE(*, '(/,/,23X,A35)') '---> TEST DE FOSTER ET STUART 1 <---'  
WRITE(*,*) '  
WRITE(*, '(/,/,15X,A35\))' '          VALEUR DE LA STATISTIQUE : '  
WRITE(*, '(15)')D  
WRITE(*, '(/,15X,A35\))' '          VALEUR DU TEST : '  
WRITE(*, '(F6.2)')FS1  
WRITE(*, '(/,15X,A35\))' 'NIVEAU DE SIGNIFICATION DU TEST : '  
WRITE(*, '(F6.4)')P  
WRITE(*, '(/,/,/,8X,A13,1X,A39)') 'COMMENTAIRE : ',COMMEN  
WRITE(*, '(/,/)' )  
VD=D
```

c *** ecriture des resultats dans le fichier "ident.tmp"

```
WRITE(2, '(F10.2,1X,F8.4,1X,F7.4,1X,A39,1X,I4,1X,I1)')VD,
```

```
⌘ FS1,P,COMMEN,-9,ITEND  
TOUCHE=CHAR(27)  
WRITE(*,'(1X,A1,\)')TOUCHE  
READ(*,'(A1)')TOUCHE  
GOTO 999
```

c *** execution dans le cas de presence de persistance

```
100 OPEN (9,FILE='FS2.TAB',STATUS='OLD')  
DO 115 KK=1,11  
  READ(9,'(F3.2,9F6.2)')PERS,(FCRIT(I),I=1,9)  
  IF (PERS.GT.RHO) THEN  
    DO 110 I=1,9  
      IF (NCRIT(I).GT.N) THEN  
        SDIND=FCRIT(I)  
        GOTO 120  
      ENDIF  
    CONTINUE  
  ENDIF  
110 CONTINUE  
115 CONTINUE  
120 FS2=D/SDIND
```

c *** determination du niveau de signification du test

```
CALL NDTR(FS2,P,DI)  
IF (P.LT.0.05) THEN  
  COMMEN='TENDANCE DECCROISSANTE DETECTEE'  
  ITEND=1  
ELSEIF (P.GT.0.95) THEN  
  COMMEN='TENDANCE CROISSANTE DETECTEE'  
  ITEND=1  
ELSE  
  COMMEN='PAS DE TENDANCE DETECTEE'  
  ITEND=0  
ENDIF
```

c *** affichage des resultats

```
WRITE(*,*)'  
WRITE(*,'(/,/,/,35X,A9)')'RESULTATS'  
WRITE(*,'(35X,A9)')'-----'  
WRITE(*,'(/,/,23X,A35)')'---> TEST DE FOSTER ET STUART 2 <---'  
WRITE(*,*)'  
WRITE(*,'(/,/,15X,A35\)\)')'          VALEUR DE LA STATISTIQUE :'  
WRITE(*,'(I5)')D  
WRITE(*,'(/,15X,A35\)\)')'          VALEUR DU TEST :'  
WRITE(*,'(F6.2)')FS2  
WRITE(*,'(/,15X,A35\)\)')'NIVEAU DE SIGNIFICATION DU TEST :'  
WRITE(*,'(F6.4)')P  
WRITE(*,'(/,/,/,8X,A13,1X,A39)')'COMMENTAIRE :',COMMEN  
WRITE(*,'(/,/\)')  
VD=D
```

c *** ecriture des resultats dans le fichier "ident.tmp"

```
⌘ FS2,P,COMMEN,-9,ITEND  
WRITE(2,'(F10.2,1X,F8.4,1X,F7.4,1X,A39,1X,I4,1X,I1)')VD,  
⌘
```

```
TOUCHE=CHAR(27)
WRITE(*,'(1X,A1,\)')TOUCHE
READ(*,'(A1)')TOUCHE
CLOSE(1)
CLOSE(2)
```

```
c *** ouverture d'un fichier bidon pour signaler que l'execution du
c *** programme "fs" s'est effectuee normalement
```

```
OPEN(99,FILE='TST.OK',STATUS='NEW')
CLOSE(99)
999 END
```

```
C SUBROUTINE FOSTUART(VECDON,N,D,S)
C * * * * *
```

```
C -----
C Fonction:
```

```
C . calcul des statistiques D et S associees au test de Foster et
C Stuart.
```

```
C Parametres:
```

```
C . vecdon Vecteur contenant la serie a traiter
C . n Longueur de la serie
C . d,s Statistiques de sortie
C -----
```

```
INTEGER K,D,S
REAL GRAND,PETIT
DIMENSION VECDON(N)
```

```
D=0
S=0
```

```
c *** Recherche les maxima (de gauche a droite)
```

```
K=2
GRAND=VECDON(1)
10 IF (K.GT.N) GOTO 40
IF (VECDON(K).GT.GRAND) GOTO 20
GOTO 30
```

```
c *** incremente les statistiques
```

```
20 D=D+1
S=S+1
GRAND=VECDON(K)
30 K=K+1
GOTO 10
40 CONTINUE
```

c *** Cherche les minima (de gauche a droite)

```
      K=2
      PETIT=VECDON(1)
50    IF (K.GT.N) GOTO 80
      IF (VECDON(K).LT.PETIT) GOTO 60
      GOTO 70
60    D=D-1
      S=S+1
      PETIT=VECDON(K)
70    K=K+1
      GOTO 50
80    CONTINUE
```

c *** Cherche les maxima (de droite a gauche)

```
      K=N-1
      GRAND=VECDON(N)
90    IF (K.LT.1) GOTO 120
      IF (VECDON(K).GT.GRAND) GOTO 100
      GOTO 110
100   D=D-1
      S=S-1
      GRAND=VECDON(K)
110   K=K-1
      GOTO 90
120   CONTINUE
```

c *** Cherche les minima (de droite a gauche)

```
      K=N-1
      PETIT=VECDON(N)
130   IF (K.LT.1) GOTO 160
      IF (VECDON(K).LT.PETIT) GOTO 140
      GOTO 150
140   D=D+1
      S=S-1
      PETIT=VECDON(K)
150   K=K-1
      GOTO 130
160   CONTINUE
      RETURN
      END
```

*INCLU DE: 'NDTR.FOR'

KEN.FOR

QSMODE, QCLEAR, QBORD, FICH, NOMFICH, ELSEIF
CHAR, FLOAT, LONG

KRANK KEN.FOR 47
47 CALL KRANK(A,B,R, LONG, TAU, SD, Z, O)

KRANK.FOR 1
Subroutine krank(a,b,r,n,tau, sd, z, nr)

fn, sqrt

RANK KRANK.FOR 37 x2
37 5 CALL RANK(A,R,N)
38 CALL RANK(B,R(N+1),N)

RANK.FOR 1
Subroutine rank(a,r,n)

TIE KRANK.FOR 83 x2
83 CALL TIE(R,N,KT,TA)
84 CALL TIE(R(N+1),N,KT,TB)

TIE.FOR 1
Subroutine tie(r,n,kt,t)

ct

NDTR KEN.FOR 48
48 CALL NDTR(Z,P,D)

NDTR.FOR 2
Subroutine ndtr(x,p,d)

Labs,exp,t

#DEBUG

PROGRAM KENDALLSAISONNIER

C -----
C *****
C Titre : Test de Kendall saisonnier / Van Belle & Hughes / Hirsch
C & Slack
C
C Fonctions :
C
C . Teste l'indépendance entre deux séries synchrones contenant
C des saisonnalités (kendall saisonnier)
C . Enchaîne avec la recherche de tendances particulières à
C l'intérieur des saisons (Van Belle & Hughes)
C . Enchaîne avec le test de Hirsch & Slack, dans le cas où l'u-
C sager a choisi explicitement ce test
C
C . En sortie, fournit la valeur des statistiques, la valeur
C des tests, et le niveau de signification de ceux-ci
C
C *****

INTEGER DIMMAX,MAXI,DOUBMAXI
REAL VALVIDE
PARAMETER (DIMMAX=20)
PARAMETER (MAXI=1300)
PARAMETER (DOUBMAXI=2600)
PARAMETER (VALVIDE=-9)
INTEGER TYP,DL,ANFIN,NA(365),MANQ(365)
CHARACTER*1 TOUCHE
CHARACTER*3 COMSIG
CHARACTER*10 F
CHARACTER*12 FICH,FICHDEB,NOMFICH
CHARACTER*39 COMMEN,COMM
LOGICAL DROIT
REAL VECDON(MAXI),SOUSVEC(DIMMAX),A(DIMMAX),R(DOUBMAXI),
RHS(MAXI),RHB(MAXI),SOUSRG(60),SOUSRB(60)

c *** lecture des spécifications graphiques

OPEN(4,FILE='SETUP.PC',STATUS='OLD')
READ(4,'(4I2)')ITMODE,ITFORE,ITBACK,ITCOLOR
CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
CLOSE(4)

c *** lecture du fichier ident.tmp

OPEN(3,FILE='IDENT.TMP',STATUS='OLD')
READ(3,2)NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,LOW,MODEP,JOSEP,
,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER,ISAIS,MD,INTER,NINT,
ICQ,ACQ,BCQ,ITRONC,IANTR,IDURTR,INTERV,IBIN,NINTAN,N,IEX1,IEX2,LS,

```
# IDIV,IBT,IPERS,RHO,N1,N2,LONG,NOSUGG,NOTEST
2  FORMAT(A12,1X,I1,/,A12,1X,I1,/,I1,1X,I1,/,I2,1X,I2,1X,I2,/,I2,1X,
# I2,1X,I2,/,I4,/,/,I2,1X,I2,/,I1,1X,I2,1X,I2,1X,I2,/,I1,1X,F9.2,1X,
# F9.2,/,I1,1X,I2,1X,I2,/,I2,1X,I1,1X,I3,1X,I4,/,I4,1X,I4,1X,I4,1X,
# I1,1X,I1,/,I1,1X,F5.2,/,I4,1X,I4,1X,I4,/,I2,1X,I2)
FICH='
FICH(1:LG)=NOMFICH(1:LG)
IF (ITYP.EQ.0) THEN
    FICH(LG+1:LG+4)='.TPC'
ELSE
    FICH(LG+1:LG+4)='.TPL'
ENDIF
5  FORMAT(E16.10)
    OPEN(1,FILE=FICH,STATUS='OLD')
```

c *** completer le vecteur des observations aux extremités avec des
c *** valeurs absentes de facon a ce qu'on puisse ensuite retracer les
c *** saisons facilement

```
DO 40 I=1,(MOD(N1+IEX1-2,NINTAN))
    VECDON(I)=VALVIDE
40 CONTINUE
    IDEBUT=(MOD(N1+IEX1-2,NINTAN))+1
    DO 45 J=IDEBUT,IDEBUT+N2-N1
        READ(1,5)VECDON(J)
45 CONTINUE
    I1=IDEBUT+N2-N1+1
    IF (MOD(N2+IEX1-1,NINTAN).EQ.0) THEN
        I2=0
        LTOT=I1-1
    ELSE
        I2=IDEBUT+N2-N1+(NINTAN-MOD(N2+IEX1-1,NINTAN))
        LTOT=I2
    ENDIF
    DO 50 I=I1,I2
        VECDON(I)=VALVIDE
50 CONTINUE
    DO 60 I=1,(LTOT/NINTAN)
        A(I)=I
60 CONTINUE
    SOMS=0
    X2TOT=0
    VARTOTS=0
```

c *** calcul du nombre d'annees considerees

```
NBAN=LTOT/NINTAN
DO 65 I=1,NINTAN
    MANG(K)=0
65 CONTINUE
    DROIT=.TRUE.
    DO 90 K=1,NINTAN
```

c *** reconstitution de chaque sous-vecteur saisonnier a partir
c *** du vecteur complete

```

70
80
c *** manq(k)=1 si la donnee pour la premiere annee est absente
c *** "      =2 si la donnee pour la derniere annee est absente
c *** "      =3 si les deux extremes sont absentes
c *** "      =0 si complet

      IF (DROIT.EQV..TRUE.) THEN
        J=0
        I=1
        IF (VECDON(K+J*NINTAN).EQ.VALVIDE) THEN
          IF (J.EQ.0) THEN
            MANQ(K)=1
          ELSEIF (MANQ(K).EQ.0) THEN
            MANQ(K)=2
          ELSE
            MANQ(K)=3
          ENDIF
          J=J+1
          IF (J.LT.LTOT/NINTAN-1) THEN
            GOTO 80
          ELSE
            I=I-1
          ENDIF
        ELSE
          SOUSVEC(I)=VECDON(K+J*NINTAN)
          IF (J.LT.(LTOT/NINTAN-1)) THEN
            I=I+1
            J=J+1
            GOTO 80
          ENDIF
        ENDIF
        IF (I.LT.3) THEN
          DROIT=.FALSE.
          GOTO 98
        ENDIF
        IF (NOTEST.NE.8) GOTO 89
        NA(K)=I
        RI=I

c *** traitement du sous-vecteur saisonnier
          IF (DROIT.EQV..TRUE.) CALL RANK(SOUSVEC,SOUSRG,I)

c *** formation du vecteur des rangs utilise dans le test de Hirsh et Slack

82
      IF (MANQ(K).EQ.1) THEN
        DO 82 L=0,I-1
          SOUSRG(NBAN-L)=SOUSRG(NBAN-L-1)
          SOUSRB(NBAN-L)=SOUSRG(NBAN-L-1)
        CONTINUE
        SOUSRG(1)=(RI+1)/2.0
        SOUSRB(1)=0.
      ELSEIF (MANQ(K).EQ.2) THEN
        SOUSRG(NBAN)=(RI+1)/2.0
        SOUSRB(NBAN)=0.
  
```

```
84      ELSEIF (MANG(K).EQ.3) THEN
          DO 84 L=0,I-2
              SOUSRG(NBAN-L)=SOUSRG(NBAN-L-1)
              SOUSRB(NBAN-L)=SOUSRB(NBAN-L-1)
          CONTINUE
          SOUSRG(1)=(RI+1)/2.0
          SOUSRG(NBAN)=(RI+1)/2.0
          SOUSRB(1)=0.
          SOUSRB(NBAN)=0.
      ENDIF
      DO 86 L=1,NBAN
          RHS((L-1)*NINTAN+K)=SOUSRG(L)
          RHB((L-1)*NINTAN+K)=SOUSRB(L)
      CONTINUE
      IF (DROIT.EQV..TRUE.) CALL KRANK(A,SOUSVEC,R,I,TAU,SD,Z,
          0)
c *** calcul de la statistique
          S=(I*(I-1)/2)*TAU
          SDS=SD*(I*(I-1)/2)
          VARS=SDS**2
          VARTOTS=VARTOTS+VARS
          SOMS=SOMS+S
          X2TOT=X2TOT+(S/SDS)**2
      ENDIF
90  CONTINUE
      IF (NOTEST.EQ.8) GOTO 95
      IF (DROIT.EQV..FALSE.) GOTO 98
      VALKS=SOMS/(VARTOTS**0.5)
c *** determination du niveau de signification du test
      CALL NDTR(VALKS,P,D)
c *** impression des resultats pour kendall saisonnier
      IF (P.LT.0.05) THEN
          COMMEN='TENDANCE MOND'TONE DECROISSANTE DETECTEE'
          ITEND=1
          COMM='
      ELSEIF (P.GT.0.95) THEN
          COMMEN='TENDANCE MONOTONE CROISSANTE DETECTEE'
          ITEND=1
          COMM='
      ELSE
          COMMEN='PAS DE TENDANCE DETECTEE'
          COMM='ANALYSE DE VAN BELLE ET HUGHES SUI'T
          ITEND=0
      ENDIF
      WRITE(*,*) '
      WRITE(*, '(/,/,/,35X,A9)') 'RESULTATS'
      WRITE(*, '(35X,A9)') '-----'
      WRITE(*, '(/,/,23X,A34)') '---> TEST DE KENDALL SAISONNIER <---'
      WRITE(*,*) '

```

```
WRITE(*, '(/,/,15X,A35\)' ) ' VALEUR DE LA STATISTIQUE : '  
WRITE(*, '(F6.2)' )SOMS  
WRITE(*, '(/,15X,A35\)' ) ' VALEUR DU TEST : '  
WRITE(*, '(F6.2)' )VALKS  
WRITE(*, '(/,15X,A35\)' ) 'NIVEAU DE SIGNIFICATION DU TEST : '  
WRITE(*, '(F6.4)' )P  
WRITE(*, '(/,/,/,8X,A13,1X,A39)' ) 'COMMENTAIRE : ',COMMEN  
WRITE(*, '(22X,A39)' )COMM  
WRITE(*, '(/,/)' )  
* WRITE(3, '(F10.2,1X,FB.4,1X,F7.4,1X,A39,1X,I4,1X,I1)' )SOMS,  
VALKS,P,COMMEN,-9,ITEND  
TOUCHE=CHAR(27)  
WRITE(*, '(1X,A1,\)' )TOUCHE  
READ(*, '(A1)' )TOUCHE
```

c *** s'il y a tendance detectee :
c *** proceder au test de hirsh et slack si c'est le test choisi;sinon,
C *** terminer sans faire van Belle et Hughes

```
IF (ITEND.EQ.1) GOTO 97  
X2TEND=VALKS**2  
X2HOMO=X2TOT-X2TEND  
DL=NINTAN-1  
CALL CHI2(X2HOMO,DL,ISIGNIF)  
IF (ISIGNIF.EQ.0) THEN  
    COMMEN='PAS D'HETEROGENEITE DETECTEE'  
    COMSIG='NON'  
ELSE  
    COMMEN='NON-HOMOGENEITE DE TENDANCES'  
    COMSIG='OUI'  
ENDIF  
CALL QCLEAR(ITBACK,ITFORE)  
CALL QCMOV(0,23)  
WRITE(*,*) '  
WRITE(*, '(/,/,/,35X,A9)' ) 'RESULTATS '  
WRITE(*, '(35X,A9)' ) '-----'  
WRITE(*, '(/,/,23X,A35)' ) '---> TEST DE VAN BELLE ET HUGHES <---'  
WRITE(*,*) '  
WRITE(*, '(/,/,15X,A35\)' ) ' VALEUR DE LA STATISTIQUE : '  
WRITE(*, '(F6.2)' )X2HOMO  
WRITE(*, '(/,15X,A35\)' ) ' NOMBRE DE DEGRE DE LIBERTE : '  
WRITE(*, '(I5)' )DL  
WRITE(*, '(/,15X,A35\)' ) ' TEST SIGNIFICATIF ? : '  
WRITE(*, '(A3)' )COMSIG  
WRITE(*, '(/,/,/,8X,A13,1X,A39)' ) 'COMMENTAIRE : ',COMMEN  
WRITE(*, '(/,/)' )  
TOUCHE=CHAR(27)  
WRITE(*, '(1X,A1,\)' )TOUCHE  
READ(*, '(A1)' )TOUCHE  
95 IF (NOTEST.NE.8) GOTO 97  
IF (DROIT.EQV..FALSE.) GOTO 98  
SCOV1=0  
SCOV2=0  
SCOV3=0  
NBSAIS=NINTAN
```

```
DO 100 I=1,NBSAIS
  DO 110 J=I+1,NBSAIS
    SCOV1=SCOV1+(NBAN*(NA(I)+1)*(NA(J)+1))/3.
    DO 120 K=1,NBAN
      M=(K-1)*NBSAIS
      SCOV2=SCOV2+(RHS(M+I)*RHS(M+J))/3.
      DO 130 L=1,K
        MM=(L-1)*NBSAIS
        AD=((RHB(M+I)-RHB(MM+I))*(RHB(M+J)-RHB(MM+J)))
        CALL SIGNE(AD,IS)
        SCOV3=SCOV3+IS/3.
      CONTINUE
    CONTINUE
  CONTINUE
COV=SCOV3+4*SCOV2-SCOV1
VARHS=VARTOTS+2*COV
WRITE(*,*)COV,SCOV3,SCOV2,SCOV1,VARTOTS
HS=SOMS/(VARHS**0.5)
```

c *** determination du niveau de signification du test

```
CALL NDTR(HS,P,D)
```

c *** impression des resultats pour HIRSCH et SLACK

```
IF(P.LT.0.05) THEN
  COMMEN='TENDANCE MONOTONE DECROISSANTE DETECTEE'
  ITEND=1
ELSEIF(P.GT.0.95) THEN
  COMMEN='TENDANCE MONOTONE CROISSANTE DETECTEE'
  ITEND=1
ELSE
  COMMEN='PAS DE TENDANCE DETECTEE'
  ITEND=0
ENDIF
CALL QCMOV(0,23)
WRITE(*,*)
WRITE(*, '(/,/,/,35X,A9)') 'RESULTATS'
WRITE(*, '(35X,A9)') '-----'
WRITE(*, '(/,/,23X,A35)') '---> TEST DE HIRSCH ET SLACK <--'
WRITE(*,*)
WRITE(*, '(/,/,15X,A35\)\') 'VALEUR DE LA STATISTIQUE : '
WRITE(*, '(F8.2)') SOMS
WRITE(*, '(/,15X,A35\)\') 'VALEUR DU TEST : '
WRITE(*, '(F8.2)') HS
WRITE(*, '(/,15X,A35\)\') 'NIVEAU DE SIGNIFICATION : '
WRITE(*, '(F7.4)') P
WRITE(*, '(/,/,/,8X,A13,1X,A39)') 'COMMENTAIRE : ',COMMEN
WRITE(*, '(/,/)\')
WRITE(3, '(F10.2,1X,F8.4,1X,F7.4,1X,A39,1X,I4,1X,I1)') SOMS,
* HS,P,COMMEN,-9,ITEND
TOUCHE=CHAR(27)
WRITE(*, '(1X,A1,\)\') TOUCHE
```

READ(*, '(A1)')TOUCHE

c *** ouverture d'un fichier bidon pour signaler que l'executuion
c *** de "kens" s'est deroulee normalement

```
97 OPEN(99,FILE='TST.OK',STATUS='NEW')
   CLOSE(99)
   GOTO 99
98 WRITE(*,*)' '
   WRITE(*,*)' *** NOMBRE INSUFFISANT DE DONNEES ***'
   WRITE(*,*)' *** CE TEST N'EST PAS PERMIS ***'
   WRITE(*,*)' '
   READ(*, '(A1)')TOUCHE
99 CLOSE(3)
   END
```

SUBROUTINE CHI2(Z,NDL,ISIGNIF)

c * * * * *

```
REAL NDLC(2),C2(2)
ISIGNIF=0
OPEN(2,FILE='CHIDEUX.TAB',STATUS='OLD')
15 READ(2,*)NDLC(2),C2(2)
   IF(NDL.EQ.NDLC(2))THEN
     IF(Z.GT.C2(2))ISIGNIF=1
     GOTO 99
   ENDIF
   IF(NDLC(2).GT.NDL)THEN
     NETEN=NDLC(2)-NDLC(1)
     C=C2(2)*(NDL-NDLC(1))/NETEN+C2(1)*(NDLC(2)-NDL)/NETEN
     IF(Z.GT.C)ISIGNIF=1
     GOTO 99
   ELSE
     NDLC(1)=NDLC(2)
     C2(1)=C2(2)
     GOTO 15
   ENDIF
99 RETURN
   CLOSE(2)
   END
```

SUBROUTINE SIGNE(XNOMBRE,IS)

c * * * * *

c -----
c Fonction:

c . Retourne le signe (-1,0,1) du nombre fourni en parametre
c selon qu'il est < , = ou > a zero

c Parametres:

c

c . xnombre Nombre reel dont on desire connaitre le signe
c . is Valeur du signe, retournee en sortie

c

```
-----  
IF (XNOMBRE.LT.O.) THEN  
    IS=-1  
ELSEIF (XNOMBRE.GT.O.) THEN  
    IS=1  
ELSE  
    IS=0  
ENDIF  
RETURN  
END
```

```
$INCLU DE: 'KRANK.FOR'  
$INCLU DE: 'NDTR.FOR'  
$INCLU DE: 'RANK.FOR'  
$INCLU DE: 'TIE.FOR'
```

KENS.FOR

QSMODE, QCLEAR, QBORD, FICH, NOMFICH, MOD
ELSEIF, I, SD, CHAR, QCMOV, NBAN

RANK KENS.FOR 152
152 V..TRUE.) CALL RANK(SOUSVEC, SOUSRG
RANK.FOR 1
Subroutine rank(a,r,n)

KRANK KENS.FOR 180
180 ..TRUE.) CALL KRANK(A, SOUSVEC, R, I,
KRANK.FOR 1
Subroutine krank(a,b,r,n,tau,sd,z,nr)

fn, sqrt

RANK KRANK.FOR 37 x2
37 5 CALL RANK(A, R, N)
38 CALL RANK(B, R(N+1), N)
RANK.FOR 1
Subroutine rank(a,r,n)

TIE KRANK.FOR 83 x2
83 CALL TIE(R, N, KT, TA)
84 CALL TIE(R(N+1), N, KT, TB)
TIE.FOR 1
Subroutine tie(r,n,kt,t)

└ct

NDTR	KENS.FOR	198 x2
198 CALL NDTR(VALKS,P,D)		
297 CALL NDTR(HS,P,D)		

	NDTR.FOR	2
Subroutine ndtr(x,p,d)		

└abs,exp,t

CHI2	KENS.FOR	243
243 CALL CHI2(X2HOMO,DL,ISIGNIF)		

	KENS.FOR	347
SUBROUTINE CHI2(Z,NDL,ISIGNIF)		

SIGNE	KENS.FOR	284
284 CALL SIGNE(AD,IS)		

	KENS.FOR	373
SUBROUTINE SIGNE(XNOMBRE,IS)		

└ELSEIF

#STORA GE:2

PROGRAM KENDALL

```
c
-----
INTEGER MAXI,DOUBMAXI
PARAMETER (MAXI=1300)
PARAMETER (DOUBMAXI=2600)
INTEGER N1,N2,LONG,TYPS,ANFIN
REAL A(MAXI),B(MAXI),R(DOUBMAXI),VAL
CHARACTER*1 TOUCHE
CHARACTER*10 F
CHARACTER*12 FICH,FICHDEB,NOMFICH
CHARACTER*39 COMMEN
```

c *** lecture des specifications graphiques

```
OPEN(4,FILE='SETUP.PC',STATUS='OLD')
READ(4,'(4I2)')ITMODE,ITFORE,ITBACK,ITCOLOR
CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
CLOSE(4)
```

c *** lecture du fichier ident.tmp

```
OPEN(3,FILE='IDENT.TMP',STATUS='OLD')
READ(3,2)NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,LOW,MODEP,JODEP
$ ,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER,ISAIS,MD,INTER,NINT,
$ ICQ,ACQ,BCQ,ITRONC,IANTR,IDURTR,INTERV,IBIN,NINTAN,N,IEX1,IEX2,LS,
$ IDIV,IBT,IPERS,RHO,N1,N2,LONG,NOSUGG,NOTEST
2 FORMAT(A12,1X,I1,/,A12,1X,I1,/,I1,1X,I1,/,I2,1X,I2,1X,I2,/,I2,1X,
$ I2,1X,I2,/,I4,/,I2,1X,I2,/,I1,1X,I2,1X,I2,1X,I2,/,I1,1X,F9.2,1X,
$ F9.2,/,I1,1X,I2,1X,I2,/,I2,1X,I1,1X,I3,1X,I4,/,I4,1X,I4,1X,I4,1X,
$ I1,1X,I1,/,I1,1X,F5.2,/,I4,1X,I4,1X,I4,/,I2,1X,I2)
FICH=
FICH(1:LG)=NOMFICH(1:LG)
IF (ITYP.EQ.0) THEN
  FICH(LG+1:LG+4)='.TPC'
ELSE
  FICH(LG+1:LG+4)='.TPL'
ENDIF
5 FORMAT(E16.10)
OPEN(1,FILE=FICH,STATUS='OLD')
READ(1,5)(B(I),I=1,LONG)
DO 30 I=1,LONG
  A(I)=I
30 CONTINUE
CALL KRANK(A,B,R,LONG,TAU,SD,Z,0)
CALL NDTR(Z,P,D)
IF (P.LT.0.05) THEN
  COMMEN='TENDANCE MONOTONE DECROISSANTE DETECTEE'
  ITEND=1
ELSEIF (P.GT.0.95) THEN
  COMMEN='TENDANCE MONOTONE CROISSANTE DETECTEE'
  ITEND=1
ELSE
```

```
      COMMEN='PAS DE TENDANCE DETECTEE  
      ITEND=0  
ENDIF  
IF (NOTEST.EQ.5)GOTO 35  
WRITE(*,*) ' '  
WRITE(*, '(/,/,/,35X,A9)') 'RESULTATS '  
WRITE(*, '(35X,A9)') '-----'  
WRITE(*, '(/,/,25X,A30)') '---> TEST DU TAU DE KENDALL <--'  
WRITE(*,*) ' '  
WRITE(*, '(/,/,15X,A35\)'') '          VALEUR DE LA STATISTIQUE : '  
WRITE(*, '(F5.2)')TAU  
WRITE(*, '(/,15X,A35\)'') '          VALEUR DU TEST : '  
WRITE(*, '(F6.2)')Z  
WRITE(*, '(/,15X,A35\)'') 'NIVEAU DE SIGNIFICATION DU TEST : '  
WRITE(*, '(F6.4)')P  
WRITE(*, '(/,/,/,8X,A13,1X,A39)') 'COMMENTAIRE : ',COMMEN  
WRITE(*,*) ' '  
IF (LONG.LE.20) THEN  
    WRITE(*,33) 'ATTENTION : N INFERIEUR OU EGAL A 20  
    WRITE(*,33) '          VOIR TABLE POUR VALEUR EXACTE'  
    WRITE(*, '(19X,A2,I2)') 'N=',N  
    FORMAT(10X,A41)  
33 ELSEIF (LONG.GT.20.AND.LONG.LE.40) THEN  
    WRITE(*,33) 'LA VALEUR EXACTE DE LA VALEUR CRITIQUE  
    WRITE(*,33) 'EST DISPONIBLE EN ANNEXE  
ENDIF  
WRITE(*,*) ' '  
WRITE(3, '(F10.2,1X,F8.4,1X,F7.4,1X,A39,1X,I4,1X,I1)')TAU,Z,P,  
$  COMMEN,-9,ITEND  
TOUCHE=CHAR(27)  
WRITE(*, '(1X,A1,\)'')TOUCHE  
READ(*, '(A1)')TOUCHE  
GOTO 99  
35 VK=FLOAT((LONG*(LONG-1))/2)*TAU  
WRITE(*,*) ' '  
WRITE(*, '(/,/,/,35X,A9)') 'RESULTATS '  
WRITE(*, '(35X,A9)') '-----'  
WRITE(*, '(/,/,29X,A23)') '---> TEST DE KENDALL <--'  
WRITE(*,*) ' '  
WRITE(*, '(/,/,15X,A35\)'') '          VALEUR DE LA STATISTIQUE : '  
WRITE(*, '(F7.0)')VK  
WRITE(*, '(/,15X,A35\)'') '          VALEUR DU TEST : '  
WRITE(*, '(F6.2)')Z  
WRITE(*, '(/,15X,A35\)'') 'NIVEAU DE SIGNIFICATION DU TEST : '  
WRITE(*, '(F6.4)')P  
WRITE(*, '(/,/,/,8X,A13,1X,A39)') 'COMMENTAIRE : ',COMMEN  
WRITE(*, '(/,/)'')  
TOUCHE=CHAR(27)  
WRITE(*, '(1X,A1,\)'')TOUCHE  
READ(*, '(A1)')TOUCHE  
WRITE(3, '(F10.2,1X,F8.4,1X,F7.4,1X,A39,1X,I4,1X,I1)')VK,Z,P,  
$  COMMEN,-9,ITEND  
99  CLOSE(1)  
     CLOSE(3)  
     OPEN(99,FILE='TST.OK',STATUS='NEW')
```

07-14-87 12:29:48 KEN.FOR
Thu 09-10-87 13:29:43

Pg 3
of 3
111-117

CLOSE(99)
END

\$INCLU DE: 'KRANK.FOR'
\$INCLU DE: 'RANK.FOR'
\$INCLU DE: 'TIE.FOR'
\$INCLU DE: 'NDTR.FOR'

Thu 09-10-87 15:37:06

Pg 1

09-02-87 13:36:06 KRANK.FOR

```
krank          KRANK.FOR          1
Subroutine krank(a,b,r,n,tau,sd,z,nr)
```

fn,RANK,TIE,sqrt

Subroutine krank(a,b,r,n,tau,sd,z,nr)
Dimension a(1),b(1),r(1)

Test de correlation de rang de Kendall entre deux variables

Description des parametres:

a vecteur d'entree de "n" observations pour la 1ere variable
b vecteur d'entree de "n" observations pour la 2eme variable
r vecteur de sortie des rang des donnees de longueur "2 * N".
la plus petite observation = r(1)
n Nombre d'observations
tau Coefficient de correlation de Kendall (sortie)
sd Ecart-type (sortie)
z Approximation normale permettant de trouver le
niveau de signification (sortie)
nr Code (entree)
0 donnees non en rang dans "a" et "b"
1 donnees en rang dans "a" et "b"

Sous-routines requises : RANK, TIE

sd=0.0
z=0.0
fn=n
fn1=fn*(fn-1.0)

Determine si les donnees sont en rang

If(nr.eq.1)goto 10

met les donnees en rang

CALL RANK(A,R,N)
CALL RANK(B,R(N+1),N)
goto 40

Met les donnees ordonnees dans R

10 Do 20 i=1,n
r(i)=a(i)
20 r(i+n)=b(i)

Trie le vecteur R en sequence de la variable A

40 isort=0
do 50 i=2,n
45 if(r(i).ge.r(i-1))goto 50
isort=isort+1
rsave=r(i)
r(i)=r(i-1)
r(i-1)=rsave
50


```

      i2=i+n
      saver=r(i2)
      r(i2)=r(i2-1)
      r(i2-1)=saver
50  Continue
      if(isort.ne.0)goto 40
C
C   Calcule S pour la variable B. En commençant avec le premier rang,
C   on additionne 1 pour chaque rang plus grand a droite et on
C   soustraie 1 pour chaque rang plus petit a droite. On recommence
C   pour tous les rangs.
C
55  s=0.0
      nm=n-1
      do 60 i=1,nm
          j=n+i
          do 60 l=i,n
              k=n+1
              if(r(k)-r(j)) 56,60,57
56          s=s-1.0
57          goto 60
58          s=s+1.0
60  continue
C
C   Calcule, pour chaque variable, les index pour les rangs egaux
C
      kt=2
      CALL TIE(R,N,KT,TA)
      CALL TIE(R(N+1),N,KT,TB)
C
C   Calcul de TAU
C
      if(ta.ne.0.0 .or. tb.ne.0.0)goto 70
      tau=s/(0.5*fn1)
      goto 85
C
70  tau=s/((sqrt(0.5*fn1-ta))*(sqrt(0.5*fn1-tb)))
C
C   Calcul de l'ecart-type et de Z
C
85  sd=(sqrt((2.0*(fn+fn+5.0))/(9.0*fn1)))
      z=tau/sd
90  Return
      End
```

MW.FOR

QSMODE, QCLEAR, QBORD, FICH, NOMFICH, QCMOV
ELSEIF, LONG, NP1, NP2, CHAR, RHO

TRANSDATE MW.FOR 53 x3
53 CALL TRANSDATE (IDIV, N1+IEX1-1, IBIN
84 CALL TRANSDATE (IDIV, N1+IEX1-1, IBIN
86 CALL TRANSDATE (IDIV, N2+IEX1-1, IBIN

MW.FOR 493
SUBROUTINE TRANSDATE (IDIV, IND, IBIN, LOW, N

ELSEIF

DATNORM MW.FOR 504
504 CALL DATNORM (IND, LOW, IBIN, NINTAN, I

MW.FOR 284
SUBROUTINE DATNORM (IND, LOW, IBIN, NINTAN, I

MOD

DATSAIS MW.FOR 506
506 CALL DATSAIS (IND, LOW, MD, INTER, NINT

MW.FOR 329
SUBROUTINE DATSAIS (IND, LOW, MD, INTER, NINT

MOD

ELSEIF

ECHANGE MW.FOR 162
162 CALL ECHANGE (VECDON, TAMP, NP1, NP2)

MW.FOR 259
SUBROUTINE ECHANGE (VECDON, TAMP, NP1, NP2)

UTEST MW.FOR 165
165 CALL UTEST (VECDON, RANK, NP1, NP2, U, Z

UTEST.FOR 1
Subroutine utest (a, r, n1, n2, R2, z, ier)

sqrt, fn, fnx

RANK UTEST.FOR 26
26 CALL RANK (A, R, N)

RANK.FOR 1
Subroutine rank (a, r, n)

TIE UTEST.FOR 48
48 CALL TIE (R, N, KT, TS)

TIE.FOR 1
Subroutine tie (r, n, kt, t)

ct

NDTR MW.FOR 182 x2
182 CALL NDTR (Z, P, D)
223 CALL NDTR (Z, P, D)

NDTR.FOR 2
Subroutine ndtr (x, p, d)

DATTRONQ MW.FOR 508
508 CALL DATTRONQ(IND,LOW,MD,NINT,AN,M

MW.FOR 361
SUBROUTINE DATTRONQ(IND,LOW,MD,NINT,AN,M

MOD

TRANSFRANG MW.FOR 110 x2
110 CALL TRANSFRANG(IDIV,AN,MO,JR,IBIN
135 CALL TRANSFRANG(IDIV,AN,MO,JR,IBIN

MW.FOR 514
SUBROUTINE TRANSFRANG(IDIV,AN,MO,JR,IBIN

ELSEIF

RANGNORM MW.FOR 525
525 CALL RANGNORM(AN,MO,JR,IBIN,LOW,IN

MW.FOR 387
SUBROUTINE RANGNORM(AN,MO,JR,IBIN,LOW,IN

RANGSAIS MW.FOR 527
527 CALL RANGSAIS(AN,MO,LOW,MD,INTER,N

MW.FOR 422
SUBROUTINE RANGSAIS(AN,MO,LOW,MD,INTER,N

RANGSTRONQ MW.FOR 529
529 CALL RANGSTRONQ(AN,MO,LOW,MD,INTER,

MW.FOR 449
SUBROUTINE RANGSTRONQ(AN,MO,LOW,MD,INTER,

Labs,exp,t

\$stora ge:2
\$noflo atcalls

PROGRAM MANNWIT

```
c -----  
  
    INTEGER MAXI  
    PARAMETER (MAXI=1300)  
    REAL VECDON(MAXI),TAMP(MAXI),RANK(MAXI)  
    INTEGER AN,MO,JFARM(12),ANFIN,RG,AN1,AN2  
    INTEGER*4 U1,U2  
    CHARACTER*1 TOUCHE,CHOIX  
    CHARACTER*10 F  
    CHARACTER*12 FICH,FICHDEB,NOMFICH  
    CHARACTER*39 COMMEN  
    DATA JFARM/31,28,31,30,31,30,31,31,30,31,30,31/  
  
c *** lecture des specifications graphiques  
  
    OPEN(4,FILE='SETUP.PC',STATUS='OLD')  
    READ(4,'(4I2)')ITMODE,ITFORE,ITBACK,ITCOLOR  
    CALL QSMODE(ITMODE)  
    CALL QCLEAR(ITBACK,ITFORE)  
    CALL QBORD(ITCOLOR)  
    CLOSE(4)  
  
c *** lecture du fichier ident.tmp  
    OPEN(3,FILE='IDENT.TMP',STATUS='OLD')  
    READ(3,2)NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,LOW,MODEP,JOSEP  
    # ,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER,ISAIS,MD,INTER,NINT,  
    # ICQ,ACQ,BCQ,ITRONC,IANTR,IDURTR,INTERV,IBIN,NINTAN,N,IEX1,IEX2,LS,  
    # IDIV,IBT,IPERS,RHO,N1,N2,LONG,NOSUGG,NOTEST  
    2  FORMAT(A12,1X,I1,/,A12,1X,I1,/,I1,1X,I1,/,I2,1X,I2,1X,I2,/,I2,1X,  
    # I2,1X,I2,/,I4,/,I2,1X,I2,/,I1,1X,I2,1X,I2,1X,I2,/,I1,1X,F9.2,1X,  
    # F9.2,/,I1,1X,I2,1X,I2,/,I2,1X,I1,1X,I3,1X,I4,/,I4,1X,I4,1X,I4,1X,  
    # I1,1X,I1,/,I1,1X,F5.2,/,I4,1X,I4,1X,I4,/,I2,1X,I2)  
    FICH=''  
    FICH(1:LG)=NOMFICH(1:LG)  
    IF (ITYP.EQ.0) THEN  
    # FICH(LG+1:LG+4)='.TPC'  
    ELSE  
    # FICH(LG+1:LG+4)='.TPL'  
    ENDIF  
    OPEN(1,FILE=FICH,STATUS='OLD')  
    READ(1,5)(VECDON(I),I=1,LONG)  
    5  FORMAT(E16.10)  
    CALL QCLEAR(ITBACK,ITFORE)  
    CALL QCMOV(0,23)  
  
c *** dans le cas ou on a decide d'eliminer certaines annees de l'analyse:  
c *** determiner le saut automatiquement a la coupure  
    IF (ITRONC.EQ.1) THEN  
    # CALL TRANSDATE(IDIV,N1+IEX1-1,IBIN,LOW,NINTAN,INTERV,MD,INTER,  
    # NINT,AN1,MO1,JR1)  
    # NP1=(IANTR-AN1)*NINTAN
```

└─ENDIF GOTO 56

c *** sinon, demander le choix de l'usager

```
10  WRITE(*,*) ' '
    WRITE(*,*) 'DE QUELLE FACON DESIREZ-VOUS DIVISER LA SERIE ?'
    WRITE(*,*) ' '
    WRITE(*,*) '      1) EN DEUX PARTIES EGALES'
    WRITE(*,*) ' '
    WRITE(*,*) '      2) EN DEUX PARTIES DELIMITEES PAR UNE DATE'
    WRITE(*,*) '      SPECIFIQUE'
    WRITE(*,*) ' '
    WRITE(*,*) '      3) EN DEUX PARTIES DELIMITEES PAR UN RANG'
    WRITE(*,*) ' '
    WRITE(*,20) 'VOTRE CHOIX : '
20  FORMAT(A15\)
```

```
    READ(*, '(A1)') CHOIX
    IF (CHOIX.NE.'1'.AND.CHOIX.NE.'2'.AND.CHOIX.NE.'3') THEN
        WRITE(*,*) ' '
        WRITE(*,*) '*** REPONSE INVALIDE ***'
        GOTO 10
    └─ENDIF
```

```
    CALL QCLEAR(ITBACK,ITFORE)
    CALL QCMOV(0,23)
```

```
25  IF (CHOIX.EQ.'1') THEN
        NP1=LONG/2
    └─ELSEIF (CHOIX.EQ.'2') THEN
        WRITE(*,*) ' '
        CALL TRANSDATE(IDIV,N1+IEX1-1,IBIN,LOW,NINTAN,INTERV,MD,INTER,
        $      NINT,AN1,MO1,JR1)
        CALL TRANSDATE(IDIV,N2+IEX1-1,IBIN,LOW,NINTAN,INTERV,MD,INTER,
        $      NINT,AN2,MO2,JR2)
        WRITE(*,26) 'VOTRE SERIE S'ECHELONNE ENTRE LE ',JR1,'/',MO1,
        $      '/',AN1,' ET LE ',JR2,'/',MO2,'/',AN2
26  FORMAT(A37,I2,A1,I2,A1,I2,AB,I2,A1,I2,A1,I2)
```

```
27  IF (IBIN.EQ.0) THEN
        WRITE(*,*) ' '
        WRITE(*,*) 'DONNEZ LA DATE DEBUTANT LA SECONDE PARTIE'
        WRITE(*,*) 'DE LA SERIE : '
        WRITE(*,*) ' '
        WRITE(*, '(AB\)\') 'JOUR : '
        READ(*,30)JR
        WRITE(*, '(AB\)\') 'MOIS : '
        READ(*,30)MO
        WRITE(*, '(A11\)\') 'ANNEE : 19'
        READ(*,30)AN
        FORMAT(I2)
        IF (AN.LT.AN1.OR.AN.GT.AN2.OR.MD.LT.1.OR.MD.GT.12.OR.J
        $      R.LT.1.OR.JR.GT.31.OR.(AN.EQ.AN1.AND.MD.LT.MO1).OR.(AN.
        $      EQ.AN2.
        AND.MD.GT.MO2)) THEN
            WRITE(*,*) ' '
            WRITE(*,*) '*** DATE INVALIDE.RECOMMENCEZ. ***'
            GOTO 27
        └─ENDIF
```

```
CALL TRANSFRANG(IDIV,AN,MO,JR,IBIN,LOW,INTERV,NINTAN,MD,  
INTER,NINT,IND,IERR)  
NF1=IND-IEX1+1-N1  
IF (NF1.LE.0) THEN  
    WRITE(*,*)'  
    WRITE(*,*)' *** REPONSE INVALIDE.RECOMMENCEZ. *  
    **'  
    WRITE(*,*)'  
    GOTO 27  
ENDIF  
ELSE  
45    WRITE(*,*)'  
    WRITE(*,*)'DONNEZ L'ANNEE ET LE MOIS DEBUTANT LA  
    SECONDE'  
    WRITE(*,*)'PARTIE : '  
    WRITE(*,*)'  
    WRITE(*,*(A8\))'MOIS : '  
    READ(*,30)MO  
    WRITE(*,*(A11\))'ANNEE : 19'  
    READ(*,30)AN  
50    FORMAT(I2,1X,I2)  
    IF (AN.LT.AN1.OR.AN.GT.AN2.OR.MO.LT.1.OR.MO.GT.12.OR.((  
    AN.  
    EQ.AN1).AND.(MO.LT.MO1)).OR.((AN.EQ.AN2).AND.(MO.GT.  
    MO2))) THEN  
        WRITE(*,*)'  
        WRITE(*,*)' *** DATE INVALIDE.RECOMMENCEZ. ***'  
        GOTO 45  
    ENDIF  
    CALL TRANSFRANG(IDIV,AN,MO,JR,IBIN,LOW,INTERV,NINTAN,MD,  
    INTER,NINT,IND,IERR)  
    NF1=IND-IEX1+1-N1  
    IF (NF1.LE.0) THEN  
        WRITE(*,*)'  
        WRITE(*,*)' *** REPONSE INVALIDE.RECOMMENCEZ. *  
        **'  
        WRITE(*,*)'  
        GOTO 45  
    ENDIF  
ENDIF  
ELSE  
55    WRITE(*,*)'  
    WRITE(*,*(A34,I4,A9,I4))' VOTRE SERIE S'ECHELONNE DU RANG',  
    N1,' AU RANG ',N2  
    WRITE(*,*)'  
    WRITE(*,*(A46\))' DONNEZ LE RANG DEBUTANT LA SECONDE PARTIE  
    :  
    READ(*,*(I4))RG  
    IF (RG.LE.N1.OR.RG.GT.N2) THEN  
        WRITE(*,*)'  
        WRITE(*,*)' *** RANG INVALIDE ***'  
        GOTO 55  
    ENDIF  
    NP1=RG-N1  
ENDIF
```



```
56      NP2=LONG-NP1
      IF (NP1.GT.NP2) THEN
          CALL ECHANGE (VECDON,TAMP,NP1,NP2)
          IECH=1
      ENDIF
      CALL UTEST (VECDON,RANK,NP1,NP2,U,Z,IER)
      CALL QCLEAR (ITBACK,ITFORE)
      CALL QCMOV (0,23)
      IF (IECH.EQ.1) THEN
          U1=U
          U2=(LONG*(LONG+1)/2)-U1
          L=NP1
          NP1=NP2
          NP2=L
      ELSE
          U2=U
          U1=(LONG*(LONG+1)/2)-U2
      ENDIF
      EU1=NP1*(LONG+1)/2.
      IF (NOTEST.EQ.2) GOTO 60
      VARU1=NP1*NP2*(LONG+1)/12.
      Z=(U1-EU1)/VARU1**0.5
      CALL NDTR (Z,P,D)
      IF (P.LT.0.05) THEN
          COMMEN='TENDANCE PAR SAUT POSITIVE DETECTEE'
          ITEND=1
      ELSEIF (P.GT.0.95) THEN
          COMMEN='TENDANCE PAR SAUT NEGATIVE DETECTEE'
          ITEND=1
      ELSE
          COMMEN='PAS DE TENDANCE DETECTEE'
          ITEND=0
      ENDIF
      WRITE (*,*) ' '
      WRITE (*, '(/,/,/,35X,A9)') 'RESULTATS'
      WRITE (*, '(35X,A9)') '-----'
      WRITE (*, '(/,/,26X,A28)') '---> TEST DE MANN-WITHNEY <--'
      WRITE (*,*) ' '
      WRITE (*, '(/,/,15X,A35\))') '          VALEUR DE LA STATISTIQUE : '
      WRITE (*, '(I8)') U1
      WRITE (*, '(/,15X,A35\))') '          VALEUR DU TEST : '
      WRITE (*, '(F8.2)') Z
      WRITE (*, '(/,15X,A35\))') 'NIVEAU DE SIGNIFICATION DU TEST : '
      WRITE (*, '(F8.4)') P
      WRITE (*, '(/,/,/,8X,A13,1X,A39)') 'COMMENTAIRE : ',COMMEN
      WRITE (*,*) ' '
      IF (NP1.LE.20.AND.NP2.LE.20) THEN
          WRITE (*,57) 'ATTENTION : N1 ET N2 SONT INFERIEURS OU EGALX A 20'
          WRITE (*,57) '          VOIR TABLES POUR VALEURS EXACTES'
          WRITE (*, '(19X,2(A6,I4))') '    N1=',NP1,',    N2=',NP2
      ENDIF
57      FORMAT (10X,A50)
      WRITE (*,*) ' '
      VU=U1
      WRITE (3, '(F10.2,1X,F8.4,1X,F7.4,1X,A39,1X,I4,1X,I1)') VU,Z,P,
```

```

*   COMMEN,NP1,ITEND
    TOUCHE=CHAR(27)
    WRITE(*,'(1X,A1,\)')TOUCHE
    READ(*,'(A1)')TOUCHE
    GOTO 99
60  VI=1/((1+RHO)/(1-RHO)-(2*RHO*(1-RHO**LONG))/(LONG*(1-RHO)**2))
    VARU1=(NP1*NP2*(LONG+1))/(12.*VI)
    Z=(U1-EU1)/(VARU1**0.5)
    CALL NDTR(Z,P,D)
    IF (P.LT.0.05) THEN
        COMMEN='TENDANCE PAR SAUT POSITIVE DETECTEE'
        ITEND=1
    ELSEIF (P.GT.0.95) THEN
        COMMEN='TENDANCE PAR SAUT NEGATIVE DETECTEE'
        ITEND=1
    ELSE
        COMMEN='PAS DE TENDANCE DETECTEE'
        ITEND=0
    ENDIF
    WRITE(*,*)' '
    WRITE(*,'(/,/,/,35X,A9)')'RESULTATS'
    WRITE(*,'(35X,A9)')'-----'
    WRITE(*,'(/,/,20X,A39)')'---> TEST DE LETTENMAIER/MANN-WITHNEY <---'
    WRITE(*,*)' '
    WRITE(*,'(/,/,15X,A35\)')'          VALEUR DE LA STATISTIQUE : '
    WRITE(*,'(I8)')U1
    WRITE(*,'(/,15X,A35\)')'          VALEUR DU TEST : '
    WRITE(*,'(F7.2)')Z
    WRITE(*,'(/,15X,A35\)')'NIVEAU DE SIGNIFICATION DU TEST : '
    WRITE(*,'(F6.4)')P
    WRITE(*,'(/,/,/,8X,A13,1X,A39)')'COMMENTAIRE : ',COMMEN
    WRITE(*,'(/,/)')
    VU=U1
    WRITE(3,'(F10.2,1X,F8.4,1X,F7.4,1X,A39,1X,I4,1X,I1)')VU,Z,P,
*   COMMEN,NP1,ITEND
    TOUCHE=CHAR(27)
    WRITE(*,'(1X,A1,\)')TOUCHE
    READ(*,'(A1)')TOUCHE
99  CLOSE(1)
    CLOSE(3)
    OPEN(99,FILE='TST.OK',STATUS='NEW')
    CLOSE(99)
    END

```

SUBROUTINE ECHANGE(VECDON,TAMP,NP1,NP2)

C * * * * *

c *** Inverse les np1 premieres valeurs de vecdon avec les np2 suivantes
c *** Le resultat est replace dans vecdon

INTEGER NP1,NP2
REAL VECDON(*),TAMP(*)

```

DO 10 I=1,NP1
    TAMP(NP2+I)=VECDON(I)

```

```
10 CONTINUE
   DO 20 I=1,NP2
      TAMP(I)=VECDON(NP1+I)
20 CONTINUE
   DO 30 I=1,NP1+NP2
      VECDON(I)=TAMP(I)
30 CONTINUE
   I=NP2
   NP2=NP1
   NP1=I
   RETURN
   END
```

SUBROUTINE DATNORM(IND,LOW,IBIN,NINTAN,INTERV,AN,MO,JR)

C * * * * *

c *** convertit un rang donne (situe entre 1 et N, i.e. relatif au
c *** 1er janvier de la premiere annee) en la date
c *** correspondante,dans le cas ou l'usager a choisi une division
c *** simple en jours ou mois.Si l'intervalle comporte plusieurs mois
c *** jours,on indique alors le milieu de l'intervalle.

INTEGER JPARM(12),AN,MO,JR
DATA JPARM/31,28,31,30,31,30,31,31,30,31,30,31/

AN=(IND-1)/NINTAN+LOW

c *** retracer egalement le mois et le jour qui se situe au milieu
c *** de l'intervalle

```
IF (IBIN.EQ.0) THEN
c *** ajustement pour les annees bissextiles
   IF (MOD(AN,4).EQ.0) THEN
      JPARM(2)=29
   ELSE
      JPARM(2)=28
   ENDIF
   RG=(MOD(IND-1,NINTAN))*INTERV+INTERV/2+1
   K=1
10  RG=RG-JPARM(K)
   IF (RG.LE.0) GOTO 20
   K=K+1
   GOTO 10
20  MO=K
   JR=RG+JPARM(K)
ELSE
c *** l'intervalle est donne en mois :
   MO=MOD(IND-1,NINTAN)*INTERV+INTERV/2+1
c *** determiner le jour dans le milieu de l'intervalle
   IF (MOD(INTERV,2).EQ.0) THEN
      JR=1
   ELSE
      JR=15
   ENDIF
ENDIF
RETURN
```

END

SUBROUTINE DATSAIS(IND,LOW,MD,INTER,NINT,AN,MO,JR)

C *****

c *** convertit un rang donne (situe entre 1 et N, i.e. relatif au
c *** 1er janvier de la premiere annee) en la date
c *** correspondante,dans le cas ou l'usager a choisi de diviser sa
c *** serie en saisons.Si l'intervalle comporte plusieurs mois (cas
c *** habituel) on indique alors le milieu de l'intervalle.

INTEGER AN,MO,JR

AN=(IND-1)/NINT+LOW

IF (MD.EQ.1) THEN

MO=(MOD(IND-1,NINT)*INTER)+(INTER/2+1)

ELSE

MO=(MOD(IND-1,NINT)*INTER+MD-1)+(INTER/2+1)

IF (MO.GT.12) THEN

c *** cas ou l'intervalle donne est "a cheval" sur la fin de l'annee et
c *** le debut de la suivante

AN=AN+1

MO=MOD(MO,12)

ENDIF

ENDIF

c *** determiner le jour dans le milieu de l'intervalle

IF (MOD(INTER,2).EQ.0) THEN

JR=1

ELSE

JR=15

ENDIF

RETURN

END

SUBROUTINE DATTROND(IND,LOW,MD,NINT,AN,MO,JR)

C *****

c *** convertit un rang donne (situe entre 1 et N, i.e. relatif au
c *** 1er janvier de la premiere annee) en la date
c *** correspondante,dans le cas ou l'usager a choisi de tronquer
c *** certains mois de sa serie.On indique le milieu du mois trouve,
c *** c'est-a-dire le jour 15.

c *** pour l'instant,l'intervalle est fixe a 1 mois.

INTEGER AN,MO,JR

AN=(IND-1)/NINT+LOW

MO=MOD(IND-1,NINT)+MD

IF (MO.GT.12) THEN

c *** cas ou l'intervalle donne est "a cheval" sur la fin de l'annee et
c *** le debut de la suivante

AN=AN+1

```
      MO=MOD(MO,12)
    ENDIF
    JR=15
    RETURN
  END
```

```
      SUBROUTINE RANGNORM(AN,MO,JR,IBIN,LOW,INTERV,NINTAN,IND,IERR)
C      * * * * *
```

```
C *** convertit une date donnee en rang correspondant dans le vecteur
C *** debutant au 1er janvier de la premiere annee.Cas ou l'usager a
C *** choisi une division simple de sa serie en jours/mois.
```

```
C *** le code ierr est mis a 1 si la date tombe dans le dernier
C *** intervalle de l'annee et que celui-ci est incomplet
```

```
      INTEGER AN,MO,JR,NBJRS,JPARM(12)
      DATA JPARM/31,28,31,30,31,30,31,31,30,31,30,31/
```

```
      IERR=0
      IF (IBIN.EQ.0) THEN
C *** l'intervalle est donne en jours
        NBJRS=0
        DO 10 I=1,MO-1
          NBJRS=NBJRS+JPARM(I)
10      CONTINUE
        IND=(AN-LOW)*NINTAN+((NBJRS+JR-1)/INTERV+1)
C *** si la date tombe dans le dernier intervalle et que ce dernier
C *** est incomplet, on ramene l'intervalle au precedent.
        IF (IND.GT.((AN-LOW+1)*NINTAN)) THEN
          IND=IND-1
          IERR=1
        ENDIF
      ELSE
C *** l'intervalle est donne en mois
        IND=(AN-LOW)*NINTAN+((MO-1)/INTERV+1)
      ENDIF
      RETURN
    END
```

```
      SUBROUTINE RANGSAIS(AN,MO,LOW,MD,INTER,NINT,IND,IERR)
C      * * * * *
```

```
C *** convertit une date donnee en rang correspondant dans le vecteur
C *** debutant au 1er janvier de la premiere annee.Cas ou l'usager a
C *** choisi une division de sa serie en saisons.
C *** on ne tient pas compte du jour dans le mois
```

```
C *** le code ierr est mis a 1 si la date tombe avant le premier
C *** intervalle
```

```
      INTEGER AN,MO
```

```
      IERR=0
      IND=(AN-LOW)*NINT+((MO+INTER-MD)/INTER)
c *** si on tombe avant le premier intervalle, on ramene celui-ci au
c *** premier
      IF (IND.EQ.0) THEN
          IND=1
          IERR=1
      ENDIF
      RETURN
      END
```

```
      SUBROUTINE RANGTROND(AN,MO,LOW,MD,INTER,NINT,IND,IERR)
c * * * * *
c *** convertit une date donnee en rang correspondant dans le vecteur
c *** debutant au 1er janvier de la premiere annee.Cas ou l'usager a
c *** choisi de tronquer certains mois de sa serie.
c *** on ne tient pas compte du jour dans le mois
c *** si la date tombe parmi les mois tronques, on ramene au premier
c *** intervalle non-tronque qui suit, et on indique cette manoeuvre
c *** en mettant le code ierr a 1.
```

```
      INTEGER AN,MO
      IERR=0
      IF (MD+NINT.LE.12) THEN
c *** les mois conserves tiennent dans une meme annee
          IF (MO.LT.MD) THEN
c *** on tombe dans un mois tronque, au ddebut de l'annee
              MO=MD
              IERR=1
          ELSEIF (MO.GT.MD+NINT) THEN
c *** on tombe dans un mois tronque, a la fin de l'annee
              MO=MD+NINT
              IERR=1
          ENDIF
          IND=(AN-LOW)*NINT+(MO-MD+1)
      ELSE
c *** les mois conserves sont "a cheval" sur la fin de l'annee et le
c *** debut de la suivante
          IF (MO.LT.MD.AND.MD.GT.(MD+NINT-13)) THEN
c *** on tombe dans un mois tronque
              MO=MD
              IERR=1
          ENDIF
          IF (MO.GE.MD) THEN
              IND=(AN-LOW)*NINT+(MO+NINT-12)
          ELSE
              IND=(AN-LOW)*NINT+MO
          ENDIF
      ENDIF
```

RETURN
END

SUBROUTINE TRANSDATE (IDIV,IND,IBIN,LOW,NINTAN,INTERV,MD,INTER,
\$ NINT,AN,MO,JR)

C * * * * *

c *** aiguille vers la bonne sous-routine pour transformer un rang
c *** en date.Depend de la facon dont l'usager a decide de diviser sa
c *** serie

INTEGER AN,MO,JR

```
IF (IDIV.EQ.0) THEN
  CALL DATNORM(IND,LOW,IBIN,NINTAN,INTERV,AN,MO,JR)
ELSEIF (IDIV.EQ.1) THEN
  CALL DATSAIS(IND,LOW,MD,INTER,NINT,AN,MO,JR)
ELSE
  CALL DATTROND(IND,LOW,MD,NINT,AN,MO,JR)
ENDIF
RETURN
END
```

SUBROUTINE TRANSFRANG (IDIV,AN,MO,JR,IBIN,LOW,INTERV,NINTAN,MD,
\$ INTER,NINT,IND,IERR)

C * * * * *

c *** aiguille vers la bonne sous-routine pour transformer une date
c *** en rang.Depend de la facon dont l'usager a decide de diviser sa
c *** serie

INTEGER AN,MO,JR

```
IF (IDIV.EQ.0) THEN
  CALL RANGNORM(AN,MO,JR,IBIN,LOW,INTERV,NINTAN,IND,IERR)
ELSEIF (IDIV.EQ.1) THEN
  CALL RANGSAIS(AN,MO,LOW,MD,INTER,NINT,IND,IERR)
ELSE
  CALL RANGSTROND(AN,MO,LOW,MD,INTER,NINT,IND,IERR)
ENDIF
RETURN
END
```

\$INCLU DE: 'UTEST.FOR'
\$INCLU DE: 'RANK.FOR'
\$INCLU DE: 'TIE.FOR'
\$INCLU DE: 'NDTR.FOR'

MWS.FOR

QSMODE, QCLEAR, QBORD, FICH, NOMFICH, MOD
QCMOV, ELSEIF, QBEEP, RHO, N, I, NPP1, NPP2
CHAR

TRANSDATE MWS.FOR 72 x3
72 CALL TRANSDATE (IDIV, N1+IEX1-1, IBIN
101 CALL TRANSDATE (IDIV, N1+IEX1-1, IBIN
103 CALL TRANSDATE (IDIV, N2+IEX1-1, IBIN

TDATE.FOR 6
SUBROUTINE TRANSDATE (IDIV, IND, IBIN, LOW, N

ELSEIF

DATNORM TDATE.FOR 42
42 CALL DATNORM (IND, LOW, IBIN, NINTAN, I

TDATE.FOR 54
SUBROUTINE DATNORM (IND, LOW, IBIN, NINTAN, I

MOD

DATSAIS TDATE.FOR 45
45 CALL DATSAIS (IND, LOW, MD, INTER, NINT

TDATE.FOR 109
SUBROUTINE DATSAIS (IND, LOW, MD, INTER, NINT

MOD

DATTRONQ	TDATE.FOR	48
48 CALL DATTRONQ(IND,LOW,MD,NINT,AN,M		
TDATE.FOR 154		
SUBROUTINE DATTRONQ(IND,LOW,MD,NINT,AN,M		

MOD

ECHANGE	MWS.FOR	168
168 CALL ECHANGE(SOUSVEC,TAMP,NP1,NP2)		
MWS.FOR 268		
SUBROUTINE ECHANGE(VECDON,TAMP,NP1,NP2)		

UTEST	MWS.FOR	171
171 ..TRUE.) CALL UTEST(SOUSVEC,RANK,N		
UTEST.FOR 1		
Subroutine utest(a,r,n1,n2,R2,z,ier)		

sqrt,fn,fnx

RANK	UTEST.FOR	26
26 CALL RANK(A, R, N)		
RANK.FOR 1		
Subroutine rank(a,r,n)		

TIE	UTEST.FOR	48
48 CALL TIE(R, N, KT, TS)		
TIE.FOR 1		
Subroutine tie(r,n,kt,t)		

ct

NDTR	MWS.FOR	196 x2
196 CALL NDTR(Z,P,D)		
226 160 CALL NDTR(ZL,P,D)		

	NDTR.FOR	2
Subroutine ndtr(x,p,d)		

abs,exp,t

PROGRAM MANNWITNEYSAISONNIER

C -----
INTEGER DIMMAX,MAXI,DOUBMAXI
REAL VALVIDE
PARAMETER (DIMMAX=20)
PARAMETER (MAXI=1300)
PARAMETER (DOUBMAXI=2600)
PARAMETER (VALVIDE=-9)
INTEGER TYP5,DL,SIGNIF,ANFIN,ANSF,AN1,AN2
CHARACTER*1 CHOIX
CHARACTER*10 F
CHARACTER*12 FICH,FICHDEB,NOMFICH
CHARACTER*39 COMMEN
LOGICAL DROIT
REAL VECDON(MAXI),SOUSVEC(DIMMAX),SOMS,RANK(MAXI),
* X2TOT,X2TEND,X2HOMO,S,TAMP(MAXI)

c *** lecture des specifications graphiques

OPEN(4,FILE='SETUP.PC',STATUS='OLD')
READ(4,'(4I2)')ITMODE,ITFORE,ITBACK,ITCOLOR
CALL @SMODE(ITMODE)
CALL @CLEAR(ITBACK,ITFORE)
CALL @BORD(ITCOLOR)
CLOSE(4)

c *** lecture du fichier ident.tmp

OPEN(3,FILE='IDENT.TMP',STATUS='OLD')
READ(3,2)NOMFICH,L6,FICHDEB,IDEB,ITYP,IMPRIM,LOW,MODEP,JODEF
* ,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER,ISAIS,MD,INTER,NINT,
* ICQ,ACQ,BCQ,ITRONC,IANTR,IDURTR,INTERV,IBIN,NINTAN,N,IEX1,IEX2,LS,
* IDIV,IBT,IPERS,RHO,N1,N2,LONG,NOSUGG,NOTEST
2 FORMAT(A12,1X,I1,/,A12,1X,I1,/,I1,1X,I1,/,I2,1X,I2,1X,I2,/,I2,1X,
* I2,1X,I2,/,I4,/,I2,1X,I2,/,I1,1X,I2,1X,I2,1X,I2,/,I1,1X,F9.2,1X,
* F9.2,/,I1,1X,I2,1X,I2,/,I2,1X,I1,1X,I3,1X,I4,/,I4,1X,I4,1X,I4,1X,
* I1,1X,I1,/,I1,1X,F5.2,/,I4,1X,I4,1X,I4,/,I2,1X,I2)
FICH=''
FICH(1:L6)=NOMFICH(1:L6)
5 FORMAT(E16.10)
IF (ITYP.EQ.0) THEN
FICH(L6+1:L6+4)=' .TPC '
ELSE
FICH(L6+1:L6+4)=' .TPL '
ENDIF
OPEN(1,FILE=FICH,STATUS='OLD')
DO 40 I=1,(MOD(N1+IEX1-2,NINTAN))
VECDON(I)=VALVIDE
40 CONTINUE
IDEBUT=(MOD(N1+IEX1-2,NINTAN))+1
DO 45 J=IDEBUT,IDEBUT+N2-N1
READ(1,5)VECDON(J)
45 CONTINUE
I1=IDEBUT+N2-N1+1

```
IF (MOD(N2+IEX1-1,NINTAN).EQ.0) THEN
    I2=0
    LTOT=I1-1
ELSE
    I2=IDEBUT+N2-N1+(NINTAN-MOD(N2+IEX1-1,NINTAN))
    LTOT=I2
ENDIF
DO 50 I=I1,I2
    VECDON(I)=VALVIDE
50 CONTINUE
    CALL @CLEAR(ITBACK,ITFORE)
    CALL @CMOV(0,23)
```

C *** cas ou l'on a tronque certaines annee(s)

```
IF (ITRONC.EQ.1) THEN
    CALL TRANSDATE(IDIV,N1+IEX1-1,IBIN,LOW,NINTAN,INTERV,MD,INTER,
    NINT,AN1,MO1,JR1)
    NP1=IANTR-AN1
    GOTO 65
ENDIF
110 WRITE(*,*) '
WRITE(*,*) 'DE QUELLE FACON DESIREZ-VOUS DIVISER LES SERIES'
WRITE(*,*) 'SAISONNIERES ?'
WRITE(*,*) '
WRITE(*,*) ' 1) EN DEUX PARTIES EGALES'
WRITE(*,*) '
WRITE(*,*) ' 2) EN DEUX PARTIES DELIMITEES PAR UNE ANNEE'
WRITE(*,*) ' SPECIFIQUE'
WRITE(*,*) '
WRITE(*,*) ' 3) EN DEUX PARTIES DELIMITEES PAR UN RANG'
WRITE(*,*) '
WRITE(*,120) 'VOTRE CHOIX : '
120 FORMAT(A15\ )
    READ(*, '(A1)') CHOIX
    IF (CHOIX.NE.'1'.AND.CHOIX.NE.'2'.AND.CHOIX.NE.'3') THEN
        WRITE(*,*) '
        WRITE(*,*) '*** REPONSE INVALIDE ***'
        GOTO 110
    ENDIF
125 IF (CHOIX.EQ.'1') THEN
    NP1=(LTOT/NINTAN)/2
ELSEIF (CHOIX.EQ.'2') THEN
    CALL @CLEAR(ITBACK,ITFORE)
    CALL @CMOV(0,23)
    CALL TRANSDATE(IDIV,N1+IEX1-1,IBIN,LOW,NINTAN,INTERV,MD,INTER,
    NINT,AN1,MO1,JR1)
    CALL TRANSDATE(IDIV,N2+IEX1-1,IBIN,LOW,NINTAN,INTERV,MD,INTER,
    NINT,AN2,MO2,JR2)
    WRITE(*,126) 'VOTRE SERIE S'ECHELONNE ENTRE LE ',JR1,'/',MO1,
    '/',AN1,' ET LE ',JR2,'/',MO2,'/',AN2
    WRITE(*,126) '
126 FORMAT(A37,I2,A1,I2,A1,I2,A8,I2,A1,I2,A1,I2)
140 WRITE(*,*) '
    WRITE(*,*) 'DONNEZ L'ANNEE POUR DIVISER LES SERIES : '
    WRITE(*,*) '(ANNEE DEBUTANT LA 2eme PARTIE DES SERIES)'
```

```
WRITE(*,'(A11\)' )'ANNEE : 19'  
READ(*,'(I2)' )ANSF  
IF (ANSF.LE.AN1.OR.ANSF.GT.AN2) THEN  
    WRITE(*,*)'  
    WRITE(*,*)' *** ANNEE INVALIDE ***'  
    WRITE(*,*)'  
    GOTO 140  
ENDIF  
NP1=(ANSF-AN1)  
ELSE  
142 *  
    CALL QCLEAR(ITBACK,ITFORE)  
    CALL QCMOV(0,23)  
    WRITE(*,'(A33,I4,A9,I4)' )' VOTRE SERIE S'ECHELONNE DU RANG',  
        N1,' AU RANG ',N2  
    WRITE(*,*)'  
    WRITE(*,*)' L'ANNEE CORRESPONDANT AU RANG DONNE SERA UTILISEE'  
    WRITE(*,*)' POUR DIVISER CHAQUE SERIE.'  
    WRITE(*,*)'  
    WRITE(*,'(A19\)' )' DONNEZ LE RANG : '  
    READ(*,'(I4)' )IRG  
    IF (IRG.LE.N1.OR.IRG.GT.N2) THEN  
        WRITE(*,*)'  
        WRITE(*,*)' *** RANG INVALIDE ***'  
        GOTO 142  
    ENDIF  
    NP1=((IRG+IEX1-N1-2)/NINTAN)  
    IF (NP1.EQ.0) THEN  
        CALL QBEEP  
        GOTO 142  
    ENDIF  
ENDIF  
65  
SOMU=0  
VI=1/((1+RHO)/(1-RHO)-(2*RHO*(1-RHO**N))/(N*(1-RHO)**2))  
DROIT=.TRUE.  
DO 90 K=1,NINTAN  
70 80  
    IF (DROIT.EQV..TRUE.) THEN  
        J=0  
        I=1  
        IF (VECDON(K+J*NINTAN).EQ.VALVIDE) THEN  
            J=J+1  
            IF (J.LT.LTOT/NINTAN-1) THEN  
                GOTO 80  
            ELSE  
                I=I-1  
            ENDIF  
        ELSE  
            SOUSVEC(I)=VECDON(K+J*NINTAN)  
            IF (J.LT.(LTOT/NINTAN-1)) THEN  
                I=I+1  
                J=J+1  
                GOTO 80  
            ENDIF  
        ENDIF  
        NP2=I-NP1  
        IF (I.LT.6) DROIT=.FALSE.
```

```
c *** pour l'instant,np1 reste fixe
      IF (NP1.GT.NP2) THEN
          CALL ECHANGE(SOUSVEC,TAMP,NP1,NP2)
          IECH=1
      ENDIF
      IF (DROIT.EQV..TRUE.) CALL UTEST(SOUSVEC,RANK,NP1,NP2,U,
          Z,IER)
      IF (IECH.EQ.1) THEN
          U1=U
          U2=(I*(I+1)/2)-U1
          NPP1=NP2
          NPP2=NP1
          NP1=NP2
      ELSE
          U2=U
          U1=(I*(I+1)/2)-U2
          NPP1=NP1
          NPP2=NP2
      ENDIF
      SOMU=SOMU+U1
      EUS1=EUS1+NPP1*(I+1)/2
      VARUS1=VARUS1+NPP1*NPP2*(I+1)/12
      VARUSL1=VARUSL1+(NPP1*NPP2*(I+1)/(12.*VI))
      ENDIF
90 CONTINUE
      IF (DROIT.EQV..FALSE.) GOTO 95
      Z=(SOMU-EUS1)/VARUS1**0.5
      ZL=(SOMU-EUS1)/VARUSL1**.5
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QCMDV(0,23)
      IF (NOTEST.EQ.4) GOTO 160
      CALL NDTR(Z,P,D)
      IF (P.LT.0.05) THEN
          COMMEN='TENDANCE PAR SAUT POSITIVE DETECTEE'
          ITEND=1
      ELSEIF (P.GT.0.95) THEN
          COMMEN='TENDANCE PAR SAUT NEGATIVE DETECTEE'
          ITEND=1
      ELSE
          COMMEN='FAS DE TENDANCE DETECTEE'
          ITEND=0
      ENDIF
      WRITE(*,*) ' '
      WRITE(*, '(/,/,/,35X,A9)') 'RESULTATS'
      WRITE(*, '(35X,A9)') '-----'
      WRITE(*, '(/,/,21X,A3B)') '---> TEST DE MANN-WITHNEY SAISONNIER <---'
      WRITE(*,*) ' '
      WRITE(*, '(/,/,15X,A35\)'') '          VALEUR DE LA STATISTIQUE : '
      WRITE(*, '(F8.2)') SOMU
      WRITE(*, '(/,15X,A35\)'') '          VALEUR DU TEST : '
      WRITE(*, '(F6.2)') Z
      WRITE(*, '(/,15X,A35\)'') 'NIVEAU DE SIGNIFICATION DU TEST : '
      WRITE(*, '(F6.4)') P
      WRITE(*, '(/,/,/,8X,A12,1X,A39)') 'COMMENTAIRE : ',COMMEN
      WRITE(*, '(/,/)'')
```

```
WRITE(3, '(F10.2,1X,F8.4,1X,F7.4,1X,A39,1X,I4,1X,I1)')SOMU,Z,P,  
*   COMMEN,NP1,ITEND  
TOUCHE=CHAR(27)  
WRITE(*, '(1X,A1,\)')TOUCHE  
READ(*, '(A1)')TOUCHE  
GOTO 93  
160 CALL NDTR(ZL,P,D)  
IF(P.LT.0.05)THEN  
    COMMEN='TENDANCE PAR SAUT POSITIVE DETECTEE'  
    ITEND=1  
ELSEIF(P.GT.0.95)THEN  
    COMMEN='TENDANCE PAR SAUT NEGATIVE DETECTEE'  
    ITEND=1  
ELSE  
    COMMEN='PAS DE TENDANCE DETECTEE'  
    ITEND=0  
ENDIF  
WRITE(*,*)'  
WRITE(*, '(/,/,/,35X,A9)')'RESULTATS'  
WRITE(*, '(35X,A9)')'-----'  
WRITE(*, '(/,/,21X,A38)')'---> TEST DE MANN-WIT/LETT SAISONNIER <---'  
WRITE(*,*)'  
WRITE(*, '(/,/,15X,A35\))'          VALEUR DE LA STATISTIQUE : '  
WRITE(*, '(F8.2)')SOMU  
WRITE(*, '(/,15X,A35\))'          VALEUR DU TEST : '  
WRITE(*, '(F6.2)')ZL  
WRITE(*, '(/,15X,A35\))'NIVEAU DE SIGNIFICATION DU TEST : '  
WRITE(*, '(F6.4)')P  
WRITE(*, '(/,/,/,8X,A12,1X,A39)')'COMMENTAIRE : ',COMMEN  
WRITE(*, '(/,/))'  
WRITE(3, '(F10.2,1X,F8.4,1X,F7.4,1X,A39,1X,I4,1X,I1)')SOMU,ZL,P,  
*   COMMEN,NP1,ITEND  
TOUCHE=CHAR(27)  
WRITE(*, '(1X,A1,\)')TOUCHE  
READ(*, '(A1)')TOUCHE  
93  OPEN(99,FILE='TST.OK',STATUS='NEW')  
    CLOSE(99)  
    GOTO 99  
95  WRITE(*,*)'  
    WRITE(*,*)'          *** NOMBRE INSUFFISANT DE DONNEES ***'  
    WRITE(*,*)'          *** CE TEST N'EST PAS PERMIS ***'  
    WRITE(*,*)'  
    READ(*, '(A1)')TOUCHE  
99  CLOSE(3)  
    END
```

SUBROUTINE ECHANGE(VECDON,TAMP,NP1,NP2)

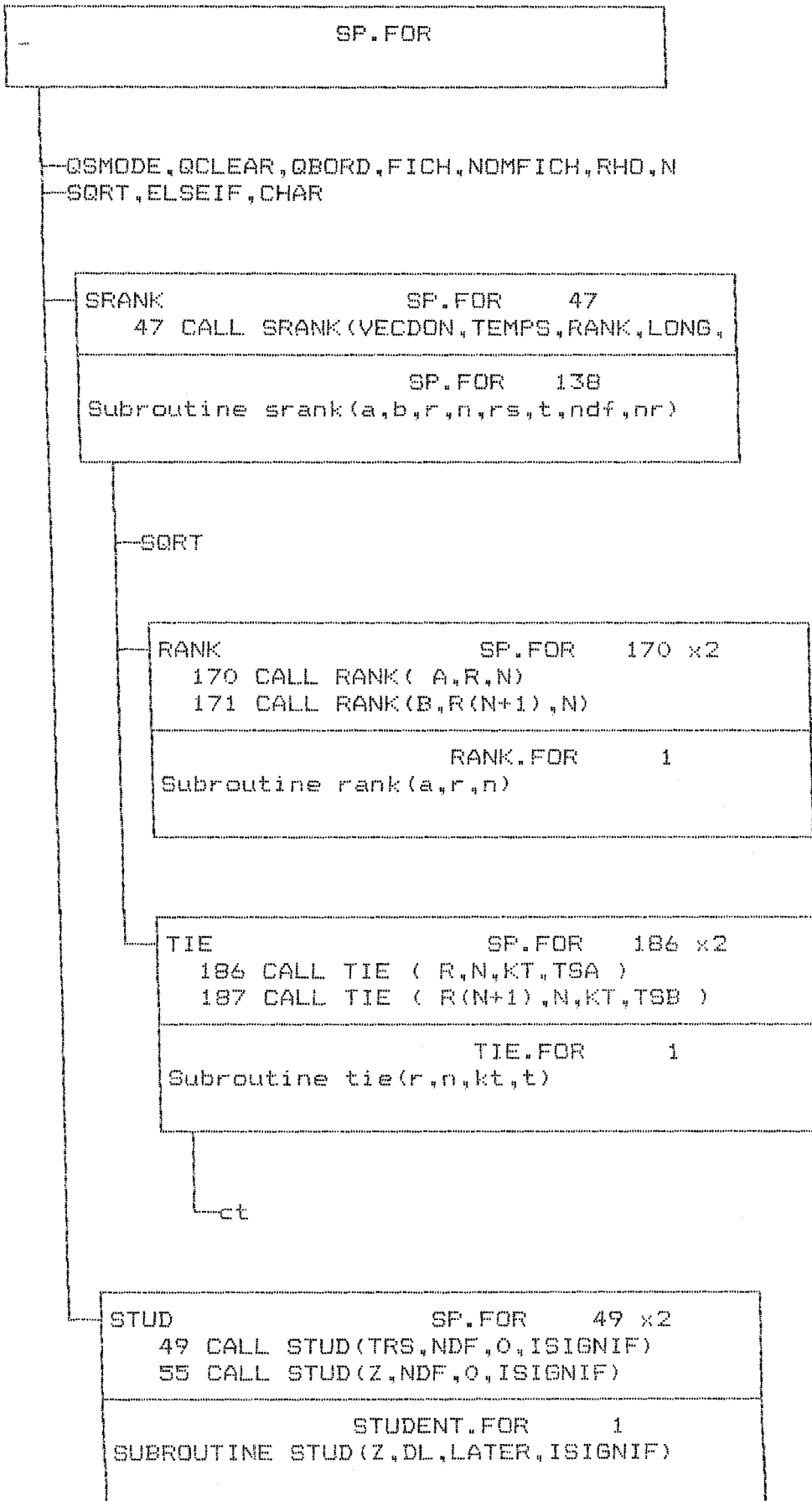
C * * * * *

c *** Inverse les np1 premieres valeurs de vecdon avec les np2 suivantes
c *** Le resultat est replace dans vecdon

INTEGER NP1,NP2

```
      REAL VECDON(*),TAMP(*)
      DO 10 I=1,NP1
         TAMP(NP2+I)=VECDON(I)
10     CONTINUE
      DO 20 I=1,NP2
         TAMP(I)=VECDON(NP1+I)
20     CONTINUE
      DO 30 I=1,NP1+NP2
         VECDON(I)=TAMP(I)
30     CONTINUE
c *** inter-echange des valeurs de np1 et np2
      NTAMP=NP1
      NP1=NP2
      NP2=NTAMP
      RETURN
      END

$INCLU      DE: 'UTEST.FOR'
$inclu     de: 'rank.for'
$inclu     de: 'tie.for'
$inclu     de: 'ndtr.for'
$INCLU     DE: 'TDATE.FOR'
```

L-ABS

#STORA GE:2
#NOFLO ATCALLS
PROGRAM SPEARMAN

c -----
INTEGER MAXI
PARAMETER (MAXI=1300)
REAL VECDON(MAXI),TEMPS(MAXI),RHO,VI,VN
INTEGER RANK(MAXI),LONG,NINTAN,N,LOW,TYPS,ANFIN
CHARACTER*1 TOUCHE
CHARACTER*3 COMSIG
CHARACTER*10 F
CHARACTER*12 FICH,FICHDEB,NOMFICH
CHARACTER*39 COMMEN

c *** lecture des specifications graphiques

```
OPEN(4,FILE='SETUP.PC',STATUS='OLD')
READ(4,'(4I2)')ITMODE,ITFORE,ITBACK,ITCOLOR
CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
CLOSE(4)
```

c *** lecture du fichier ident.tmp

```
OPEN(3,FILE='IDENT.TMP',STATUS='OLD')
READ(3,2)NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,LOW,MODEP,JODEF
# ,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER,ISAIS,MD,INTER,NINT,
# ICQ,ACQ,BCQ,ITRONC,IANTR,IDURTR,INTERV,IBIN,NINTAN,N,IEX1,IEX2,LS,
# IDIV,IBT,IPERS,RHO,N1,N2,LONG,NOSUGG,NOTEST
2 FORMAT(A12,1X,I1,/,A12,1X,I1,/,I1,1X,I1,/,I2,1X,I2,1X,I2,/,I2,1X,
# I2,1X,I2,/,I4,/,I2,1X,I2,/,I1,1X,I2,1X,I2,1X,I2,/,I1,1X,F9.2,1X,
# F9.2,/,I1,1X,I2,1X,I2,/,I2,1X,I1,1X,I3,1X,I4,/,I4,1X,I4,1X,I4,1X,
# I1,1X,I1,/,I1,1X,F5.2,/,I4,1X,I4,1X,I4,/,I2,1X,I2)
FICH=
FICH(1:LG)=NOMFICH(1:LG)
IF (ITYP.EQ.0) THEN
    FICH(LG+1:LG+4)='.TPC'
ELSE
    FICH(LG+1:LG+4)='.TPL'
ENDIF
5 FORMAT(E16.10)
OPEN(1,FILE=FICH,STATUS='OLD')
READ(1,5)(VECDON(I),I=1,LONG)
DO 30 I=1,LONG
    TEMPS(I)=I
30 CONTINUE
CALL SRANK(VECDON,TEMPS,RANK,LONG,RS,TRS,NDF,NR)
IF (NOTEST.EQ.12) THEN
    CALL STUD(TRS,NDF,0,ISIGNIF)
    GOTO 40
ENDIF
N=LONG
VI=1/((1+RHO)/(1-RHO)-(2*RHO*(1-RHO**N))/(N*(1-RHO)**2))
Z=RS*SQRT((N-2)/(1-RS**2)*VI)
CALL STUD(Z,NDF,0,ISIGNIF)
```

```
GOTO 45
40 IF (TRS.LT.0.0.AND.ISIGNIF.EQ.1) THEN
      COMMEN='TENDANCE MONOTONE DECROISSANTE DETECTEE'
      ITEND=1
      COMSIG='OUI'
ELSEIF (TRS.GT.0.0.AND.ISIGNIF.EQ.1) THEN
      COMMEN='TENDANCE MONOTONE CROISSANTE DETECTEE'
      ITEND=1
      COMSIG='OUI'
ELSE
      COMMEN='PAS DE TENDANCE DETECTEE'
      ITEND=0
      COMSIG='NON'
ENDIF
WRITE(*,*) ' '
WRITE(*, '(/,/,/,35X,A9)') 'RESULTATS'
WRITE(*, '(35X,A9)') '-----'
WRITE(*, '(/,/,28X,A24)') '---> TEST DE SPEARMAN <---'
WRITE(*,*) ' '
WRITE(*, '(/,/,15X,A35\)'') '          VALEUR DE LA STATISTIQUE : '
WRITE(*, '(F5.2)')RS
WRITE(*, '(/,15X,A35\)'') '          VALEUR DU TEST : '
WRITE(*, '(F6.2)')TRS
WRITE(*, '(15X,A35\)'') '          NOMBRE DE DEGRE DE LIBERTE : '
WRITE(*, '(I4)')NDF
WRITE(*, '(/,15X,A35\)'') '          TEST SIGNIFICATIF ? : '
WRITE(*, '(A3)')COMSIG
WRITE(*, '(/,/,/,8X,A13,1X,A39)') 'COMMENTAIRE : ',COMMEN
WRITE(*,*) ' '
IF (LONG.LE.11) THEN
      WRITE(*,42) 'ATTENTION : N INFERIEUR OU EGAL A 11'
      WRITE(*,42) '          VOIR TABLE POUR VALEUR EXACTE'
ENDIF
42 FORMAT(10X,A41)
WRITE(*,*) ' '
VISIGNIF=ISIGNIF
WRITE(3, '(F10.2,1X,FB.4,1X,F7.4,1X,A39,1X,I4,1X,I1)')RS,TRS,
* VISIGNIF,COMMEN,NDF,ITEND
TOUCHE=CHAR(27)
WRITE(*, '(1X,A1,\)'')TOUCHE
READ(*, '(A1)')TOUCHE
GOTO 99
45 IF (Z.LT.0.0.AND.ISIGNIF.EQ.1) THEN
      COMMEN='TENDANCE MONOTONE DECROISSANTE DETECTEE'
      ITEND=1
      COMSIG='OUI'
ELSEIF (Z.GT.0.0.AND.ISIGNIF.EQ.1) THEN
      COMMEN='TENDANCE MONOTONE CROISSANTE DETECTEE'
      ITEND=1
      COMSIG='OUI'
ELSE
      COMMEN='PAS DE TENDANCE DETECTEE'
      ITEND=0
      COMSIG='NON'
ENDIF
```

```
WRITE(*,*) ' '
WRITE(*, '(/,/,/,35X,A9)') 'RESULTATS'
WRITE(*, '(35X,A9)') '-----'
WRITE(*, '(/,/,22X,A36)') '---> TEST DE LETTENMAIER SPEARMAN <---'
WRITE(*,*) ' '
WRITE(*, '(/,/,15X,A35\))' '          VALEUR DE LA STATISTIQUE : '
WRITE(*, '(F5.2)')RS
WRITE(*, '(/,15X,A35\))' '          VALEUR DU TEST : '
WRITE(*, '(F6.2)')Z
WRITE(*, '(15X,A35\))' '          NOMBRE DE DEGRE DE LIBERTE : '
WRITE(*, '(I4)')NDF
WRITE(*, '(/,15X,A35\))' '          TEST SIGNIFICATIF ? : '
WRITE(*, '(A3)')COMSIG
WRITE(*, '(/,/,/,8X,A13,1X,A39)') 'COMMENTAIRE : ',COMMEN
WRITE(*, '(/,/)')
VISIGNIF=ISIGNIF
WRITE(3, '(F10.2,1X,FB.4,1X,F7.4,1X,A39,1X,I4,1X,I1)')RS,Z,
$ VISIGNIF,COMMEN,NDF,ITEND
TOUCHE=CHAR(27)
WRITE(*, '(1X,A1,\)')TOUCHE
READ(*, '(A1)')TOUCHE
99 CLOSE(1)
CLOSE(3)
OPEN(99,FILE='TST.OK',STATUS='NEW')
CLOSE(99)
END
```

C

```
Subroutine srank(a,b,r,n,rs,t,ndf,nr)
Dimension a(1),b(1),r(1)
```

C

Utilite: Test de rang de Spearman entre 2 variables

C

Description des parametres:

C

- C a vecteur d'entree de "n" observations pour la 1ere variable
- C b vecteur d'entree de "n" observations pour la 2eme variable
- C r vecteur de sortie des rangs de longueur 2*N.
- C n Nombre d'observations.
- C rs Coefficient de correlation de rang de Spearman (sortie)
- C t Test de signification de "rs" (sortie)
- C ndf Nombre de degres de liberte (sortie)
- C nr code d'entree:
 - C 0 donnees non en rang dans "a" et "b"
 - C 1 donnees en rang dans "a" et "b"

C

T est mis a zero (0) si N < 10.

C

Sous-routines requises: RANK, TIE

C

C

```
fltnm2=n-2
rn=n
fnnn=rn*rn*rn-rn
```

C

```
C Determine si les donnees sont en rang
C
      if(nr .eq. 1)goto 10
C
      CALL RANK( A,R,N)
      CALL RANK(B,R(N+1),N)
      goto 40
C
C remplit le vecteur R
C
10      do 20 i=1,n
          r(i)=a(i)
20      r(i+n)=b(i)
C
40      d=0.0
          do 50 i=1,n
              j=i+n
50      d=d+(r(i)-r(j))*(r(i)-r(j))
C
          kt=1
          CALL TIE ( R,N,KT,TSA )
          CALL TIE ( R(N+1),N,KT,TSB )
C
C Calcule le coefficient
C
          if(tsa .ne. 0.0 .or. tsb .ne. 0.0) goto 60
57      rs=1.0 - 6.0 * d / fnnn
          goto 70
60      x=fnnn/12.0-tsa
          y=x+tsa-tsb
          rs=(x+y-d)/(2.0*(sqrt(x*y)))
C
C Calcule T
C
          t=0.0
70      if(n .lt. 10)goto 80
75      t=rs*sqrt(flnm2/(1.0-rs*rs))
80      ndf=n-2
          Return
          End

#INCLU    DE: 'RANK.FOR'
#INCLU    DE: 'TIE.FOR'
#INCLU    DE: 'STUDENT.FOR'
```

Thu 09-10-87 15:22:52
09-10-87 14:41:16 RANK.FOR

```
rank          RANK.FOR      1  
Subroutine rank(a,r,n)
```

Subroutine rank(a,r,n)
Dimension a(1),r(1)

```
C
C   Utilite: Construit le vecteur des rangs d'un vecteur de donnees
C
C   Description des parametres:
C
C       a   vecteur d'entree de "n" valeurs (entree)
C       r   vecteur des rangs (sortie)
C       n   Nombre de valeurs (entree)
C
C*****
C
C   Initialisation
C
C       do 10 i=1,n
10      r(i)=0.0
C       do 100 i=1,n
C
C   Teste si le point a deja ete place
C
C       if(r(i).gt.0.0)goto 100
20      small=0.0
        equal=0.0
        x=a(i)
        do 50 j=1,n
          if(a(j)-x)30, 40, 50
C
C   Compte le nombre de donnees qui sont plus petites
C
30      small=small+1.0
        goto 50
C
C   Compte le nombre de donnees qui sont egales
C
40      equal=equal+1.0
        r(j)=-1.0
50      Continue
C
        if(equal .gt. 1.0)goto 70
60      r(i)=small+1.0
        goto 100
70      p=small + (equal + 1.0 ) * 0.5
        do 90 j=i,n
          if(r(j) .ne.-1.0) goto 90
80          r(j)=p
90      Continue
100     Continue
        Return
        End
```


Thu 09-10-87 15:23:59 STUDENT.FOR
06-04-87 12:49:26

STUD STUDENT.FOR 1
SUBROUTINE STUD(Z,DL,LATER,ISIGNIF)

ABS

```
      SUBROUTINE STUD(Z,DL,LATER,ISIGNIF)
      ISIGNIF=0
      OPEN(9,FILE='STUDENT.TAB',STATUS='OLD')
15     IF(DL2.EQ.30.)GOTO 20
      READ(9,*)DL2,VCU,VCB
      IF(DL2.NE.DL)GOTO 15
20     IF(LATER.EQ.0.AND.ABS(Z).GT.VCU)ISIGNIF=1
      IF(LATER.EQ.1.AND.ABS(Z).GT.VCB)ISIGNIF=1
      CLOSE(9)
      RETURN
      END
```

Thu 09-10-87 15:24:45
09-02-87 13:44:58 TIE.FOR

Pg 1

```
tie          TIE.FOR      1  
Subroutine tie(r,n,kt,t)
```

└ct

Subroutine tie(r,n,kt,t)
Dimension r(1)

```
C
C  Utilite : Calcul du facteur de correction
C
C  Description des parametres:
C
C  r  vecteur d'entree de grandeur "n" contenant les valeurs de 1 a N
C  n  nombre de rangs (entree)
C  kt code pour le calcul de correction
C  1  : equation 1
C  2  : equation 2
C  t  facteur de Correction (sortie)
C      equation 1      T=SUM(CT**3-CT)/12
C      equation 2      T=SUM(CT*(CT-1)/2)
C
C*****
C
C Initialisation
C
C      t=0.0
C      y=0.0
5      x=1.0e30
C      ind=0
C
C trouve le plus grand rang
C
C      do 30 i=1,n
C          if (r(i).le.y)goto 30
C          if (r(i).ge.x)goto 30
20          x=r(i)
C          ind=ind+1
30      Continue
C
C si tous les rang ont ete testes, Retour au programme
C
C      if(ind.le.0)Return
40      y=x
C      ct=0.0
C
C Compte les egalites
C
C      do 60 i=1,n
C          if(r(i).ne.x)goto 60
50          ct=ct+i.0
60      Continue
C
C Calcul le facteur de correction
C
C      if(ct .eq. 0.0)goto 5
70      if( kt .eq. 1)goto 80
75      t=t+ct*(ct-1.0)/2.0
C      goto 5
80      t=t+(ct*ct*ct-ct)/12.0
C      goto 5
```

09-02-87 13:44:58 TIE.FOR
Thu 09-10-87 14:27:13

tie

Pg 2
of 2
56-57

90 Return
 End

Thu 09-10-87 15:36:46
09-10-87 14:50:22 UTEST.FOR

Pg 1

```
utest          UTEST.FOR      1  
Subroutine utest(a,r,n1,n2,R2,z,ier)
```

└RANK,TIE,sqrt,fn,fnx

```
      Subroutine utest(a,r,n1,n2,R2,z,ier)
      Dimension a(1),r(1)
C
C  Utilite: Test de Mann-Whitney (U) pour 2 groupes independants d'une
C          meme population
C
C  Description des parametres:
C
C  a      Vecteur d'entree constitue de 2 groupes independants et
C          de grandeur "n1 + n2" ou n1 < n2
C  r      Vecteur de sortie contenant les rangs et
C          de grandeur "n1 + n2" ou n1 < n2
C  n1     Nombre d'elements dans le premier groupe
C  n2     Nombre d'elements dans le deuxieme groupe
C  u      Sert a tester l'homogeneite des 2 groupes (sortie)
C  z      Approximation normale permettant de trouver le
C          niveau de signification (output)
C  ier    0, si aucune erreur (sortie)
C
C  Sous-routines requises:  RANK, TIE
C
C*****
C
      ier=0
      n=n1+n2
      CALL RANK( A, R, N)
      z=0.0
C
C  Somme des rangs du premier groupe
C
      r2=0.0
      np=n1+1
      do 10 i=np,n
10    r2=r2+r(i)
C
C  Calcule de U
C
      u=r2
      fn1=n1
      fn2=n2
      fn=n
      fnx=fn1*fn2
      fnnn=fn*fn*fn-fn
C
C  Calcule l'ecart-type
C
40    kt=1
      CALL TIE(R, N, KT, TS)
      if( ts .eq. 0.0)goto 60
      if(ts .ne. fnnn)goto 52
      ier=1
      Return
52    s=sqrt((fnx/(fn*(fn-1.0)))*((fnnn/12.0)-ts))
      goto 70
60    s=sqrt(fnx*(fn+1.0)/12.0)
```

09-10-87 14:50:22 utest.for
Thu 09-10-87 14:50:49

utest

Pg 2
of 2
56-62

```
C
C Calcule de Z
C
70      z=(u-fnx*0.5)/s
80      Return
      End
```


P6.FOR

QSMODE, QCLEAR, QBORD, QCMOV, QDATE, RHO
LONG, ELSEIF, QCHAR, QBEEP, FICH, NOMFICH

TRANSDATE P6.FOR 241 x3
241 CALL TRANSDATE (IDIV, N1+IEX1-1, IBIN
245 CALL TRANSDATE (IDIV, N2+IEX1-1, IBI
248 CALL TRANSDATE (IDIV, N2+IEX1-1, IBIN

TDATE.FOR 6
SUBROUTINE TRANSDATE (IDIV, IND, IBIN, LOW, N

ELSEIF

DATNORM TDATE.FOR 42
42 CALL DATNORM (IND, LOW, IBIN, NINTAN, I

TDATE.FOR 54
SUBROUTINE DATNORM (IND, LOW, IBIN, NINTAN, I

MOD

DATSAIS TDATE.FOR 45
45 CALL DATSAIS (IND, LOW, MD, INTER, NINT

TDATE.FOR 109
SUBROUTINE DATSAIS (IND, LOW, MD, INTER, NINT

MOD

```

DATTRONQ          TDATE.FOR    48
48 CALL DATTRONQ(IND,LOW,MD,NINT,AN,M

```

```

          TDATE.FOR    154
SUBROUTINE DATTRONQ(IND,LOW,MD,NINT,AN,M

```

MOD

```

NOMTEST          P6.FOR    277 x2
277 CALL NOMTEST(NOSUBG,NOM1)
278 CALL NOMTEST(NOTEST,NOM2)

```

```

          P6PLUS.FOR    404
SUBROUTINE NOMTEST(NOTEST,NOM)

```

ELSEIF

```

STATGLOB          P6.FOR    341
341 CALL STATGLOB(VECSER,TAMP,LONG,N1,

```

```

          P6.FOR    438
SUBROUTINE STATGLOB(VECSER,TAMP,LONG,N1,

```

```

TRANSDATE          P6.FOR    477    p1
477 CALL TRANSDATE(IDIV,NF1+IEX1+N1-1,

```

```

STATSAIS          P6.FOR    347
347 CALL STATSAIS(VECSER,TAMP,LONG,N1,

```

```

          P6.FOR    525
SUBROUTINE STATSAIS(VECSER,TAMP,LONG,N1,

```

MOD, QCMOV, QCHAR, QBEEP, QCLEAR

```

GRAFAJUS          P6.FOR    357 x4
357 CALL GRAFAJUS(VECSE, TAMP, LONG, IVA
377 CALL GRAFAJUS(VECSE, TAMP, LONG, IVA
393 CALL GRAFAJUS(VECSE, TAMP, LONG, IVA

```

```

          P6PLUS.FOR    181
SUBROUTINE GRAFAJUS(X, T, LONG, NP1, ITY, N1,

```

```

MIN, MAX, FLOAT, QSMODE, QCOLOR, QOVSCN
QPLOT, QXAXIS, QYAXIS, QGTXT, QPTXTC, QPTXTA
QSETUP, QTABL, QRTOI, QLINE, ELSEIF, QCMOV
QCHAR, QBEEP, QCLEAR, QBORD

```

```

TICS              P6PLUS.FOR    267 x2
267 CALL TICS(XMIN, XMAX, YSTS, YFINS, TIC
268 CALL TICS(YMIN, YMAX, XSTS, XFINS, TIC

```

```

          TICS.FOR      1
SUBROUTINE TICS(XST, XFIN, XSTS, XFINS, TIC,

```

```

ELSEIF, AINT, ABS

```

```

AJUSTE           P6PLUS.FOR    271 x6
271 CALL AJUSTE(X, LONG, AINTEY, TEXTEY, Y
272 CALL AJUSTE(X2, NP2, AINTEY, O., O., O.
282 CALL AJUSTE(T, LONG, AINTEX, TEXTEX, X

```

```

          AJUSTE.FOR    1
SUBROUTINE AJUSTE(X, NOBS, AINTE, TEXTE, XST

```

```

ALOG10, EXP, CHAR, ABS, ELSEIF, MOD, TEXTE

```

```

STAMW            P6.FOR    367
367 CALL STAMW(IDIV, IVAR, IBIN, LOW, NINT

```

```

          P6.FOR    680
SUBROUTINE STAMW(IDIV, NP1, IBIN, LOW, NINTA

```

```

TRANSDATE          P6.FOR  721    p1
721 CALL TRANSDATE(IDIV,NP1+IEX1+N1-1,

```

```

STAMWSAIS          P6.FOR  385
385 CALL STAMWSAIS(AN1,N1,N2,IEX1,MD,I

```

```

          P6.FOR  812
SUBROUTINE STAMWSAIS(AN1,N1,N2,IEX1,MD,I

```

```

_QCMOV,_QCHAR,_QBEEP,_QCLEAR,MOD

```

```

MONOTONE          P6.FOR  400
400 CALL MONOTONE(VECSE,LONG,NINTAN)

```

```

          P6.FOR  1134
SUBROUTINE MONOTONE(X,N,NINTAN)

```

```

MONOSAIS          P6.FOR  416
416 CALL MONOSAIS(VECSE,TAMP,LONG,N1,

```

```

          P6PLUS.FOR  1
SUBROUTINE MONOSAIS(VECSE,TAMP,LONG,N1,

```

```

MOD,_QCLEAR,_QCMOV,_QCHAR,_QBEEP

```

\$STORA GE:2

PROGRAM PARTIE6

```
c -----
c
c *****
c Titre : Resume de la situation
c
c Fonctions :
c
c . Regrouper sous forme de tableaux l'ensemble de l'information
c   recueillie au cours du traitement de la serie et conservee
c   dans le fichier "ident.tmp"
c . Calculer differentes statistiques selon le test precedemment
c   utilise (moyennes,ecarts-types)
c
c *****
```

```
INTEGER ANFIN,MAXI
PARAMETER (MAXI=1300)
REAL VALVIDE
PARAMETER (VALVIDE=-9)
REAL VECSER(MAXI),TAMP(MAXI)
INTEGER AN,MO,JR,AN1,AN2
CHARACTER*1 TOUCHE
CHARACTER*3 COMSIG
CHARACTER*8 FORM1,FORM2
CHARACTER*12 FICHDEB,FICH,F,NOMFICH
CHARACTER*30 NOM1,NOM2
CHARACTER*39 COMMEN
CHARACTER*10 NMOIS(12)
CHARACTER*5 FMOIS(12)
CHARACTER*80 LIGNE
COMMON /IMPR/ IMPRIM,NOTEST,ITYP
COMMON /TRONC/ ITRONC,IDURTR,IANTR
COMMON /BLOC61/ FORM1,FORM2
COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH/ IGMODE,IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,
$ ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2
COMMON /SAUT/ SAV,ECTAVANT,SAP,ECTAPRES
COMMON /REGRE/Y1,YN,PENTE,SCM
COMMON /CONTINF/ VI
DATA NMOIS/'JANVIER','FEVRIER','MARS','AVRIL','MAI','JUIN',
$ 'JUILLET','AOUT','SEPTEMBRE','OCTOBRE','NOVEMBRE','DECEMBRE'/
DATA FMOIS/'(A7\)', '(A7\)', '(A4\)', '(A5\)', '(A3\)', '(A4\)',
$ '(A7\)', '(A4\)', '(A9\)', '(A7\)', '(A8\)', '(A8\)' /
```

c *** lecture des specifications graphiques

```
OPEN(1,FILE='SETUP.PC',STATUS='OLD')
READ(1,'(4I2/9I2,5I3)')ITMODE,ITFORE,ITBACK,ITCOLOR,IGMODE,
$ IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,ILETR,ISEC,KIRSYM,ISYMBL,
$ JROW1,JROW2,JCOL1,JCOL2
CLOSE(1)
```

c *** mise en mode texte

```
CALL QSMODE(ITMODE)
CALL QCLEAR(ITBACK,ITFORE)
CALL QBORD(ITCOLOR)
```

c *** lecture du fichier "ident.tmp"

```
OPEN(2,FILE='IDENT.TMP',STATUS='OLD')
READ(2,10)NOMFICH,LG,FICHDEB,IDEB,ITYP,IMPRIM,LOW,MODEP,JOEP
$ ,ANFIN,MOFIN,JOFIN,NOBS,NMINABER,NMAXABER,ISAI,MD,INTER,NINT,
$ ICQ,ACQ,BCQ,ITRONC,IANTR,IDURTR,INTERV,IBIN,NINTAN,N,IEX1,IEX2,
$ LS,IDIV,IBT,IPERS,RHO,N1,N2,LONG,NOSUGG,NOTEST,STAT,VALTEST,
$ SIGNIF,COMMEN,IVAR,ITEND
10 FORMAT(A12,1X,I1,/,A12,1X,I1,/,I1,1X,I1,/,I2,1X,I2,1X,I2,/,I2,1X,
$ I2,1X,I2,/,I4,/,I2,1X,I2,/,I1,1X,I2,1X,I2,1X,I2,/,I1,1X,F9.2,1X,
$ F9.2,/,I1,1X,I2,1X,I2,/,I2,1X,I1,1X,I3,1X,I4,/,I4,1X,I4,1X,I4,1X,
$ I1,1X,I1,/,I1,1X,F5.2,/,I4,1X,I4,1X,I4,/,I2,1X,I2,/,F10.2,1X,F8.4,
$ 1X,F7.4,1X,A39,1X,I4,1X,I1)
```

```
OPEN(3,FILE='SYNTHESE.P6',STATUS='OLD')
OPEN(4,FILE='TAMPON.P6',STATUS='NEW')
12 READ(3,13,END=11)LIGNE
WRITE(4,13)LIGNE
GOTO 12
13 FORMAT(A80)
11 CLOSE(3,STATUS='DELETE')
CALL QCLEAR(ITBACK,ITFORE)
CALL QCMOV(0,23)
CALL QDATE(IAN,IMO,IJR)
```

c *** calcul du contenu en information

```
VI=1/((1+RHO)/(1-RHO)-(2*RHO*(1-RHO**LONG))/(LONG*(1-RHO)**2))
```

c *** debut de l'impression de l'information

```
WRITE(*,*)' '
WRITE(4,*)' '
WRITE(*,14)'S Y N T H E S E' SESSION DU ',IJR
WRITE(4,14)'S Y N T H E S E' SESSION DU ',IJR
14 FORMAT(1X,A46,I2,1X\))
WRITE(*,FMOIS(IMO))NMOIS(IMO)
WRITE(*,'(1X,I4)')IAN
WRITE(*,*)' '
WRITE(*,15)'NOM DU FICHER : ',NOMFICH,'SERIE TRAITEE : '
WRITE(4,FMOIS(IMO))NMOIS(IMO)
WRITE(4,'(1X,I4)')IAN
WRITE(4,*)' '
WRITE(4,15)'NOM DU FICHER : ',NOMFICH,'SERIE TRAITEE : '
15 FORMAT(1X,A18,A12,A17\))
IF (ITYP.EQ.0) THEN
    WRITE(*,*)' CONCENTRATIONS'
    WRITE(4,*)' CONCENTRATIONS'
```

```
ELSE
    WRITE(*,*) ' CHARGES '
    WRITE(4,*) ' CHARGES '
ENDIF
FORM1='(F10.3)'
FORM2='(F10.3\)'
WRITE(*,*) ' '
WRITE(*,16) 'SERIE ORIGINALE :      DATE DE LA PREMIERE OBSERVATION :
$      ',JODEP,'/',MODEP,'/',LOW
WRITE(4,*) ' '
WRITE(4,16) 'SERIE ORIGINALE :      DATE DE LA PREMIERE OBSERVATION :
$      ',JODEP,'/',MODEP,'/',LOW
16  FORMAT(1X,A56,I2,A1,I2,A1,I2)
WRITE(*,16) '      DATE DE LA DERNIERE OBSERVATION :
$      ',JOFIN,'/',MOFIN,'/',ANFIN
WRITE(*,17) '      NOMBRE D'OBSERVATIONS
$      : ',NOBS
WRITE(*,17) '      NOMBRE DE MINIMA RETIRES      :
$      ',NMINABER
WRITE(*,17) '      NOMBRE DE MAXIMA RETIRES      :
$      ',NMAXABER
WRITE(4,16) '      DATE DE LA DERNIERE OBSERVATION :
$      ',JOFIN,'/',MOFIN,'/',ANFIN
WRITE(4,17) '      NOMBRE D'OBSERVATIONS
$      : ',NOBS
WRITE(4,17) '      NOMBRE DE MINIMA RETIRES      :
$      ',NMINABER
WRITE(4,17) '      NOMBRE DE MAXIMA RETIRES      :
$      ',NMAXABER
17  FORMAT(1X,A56,I4)

WRITE(*,*) ' '
WRITE(*,'(1X,A29\)' ) 'ANNEES A FAIBLE FREQUENCE : '
WRITE(4,*) ' '
WRITE(4,'(1X,A29\)' ) 'ANNEES A FAIBLE FREQUENCE : '
IF (ITRONC.EQ.1) THEN
$   WRITE(*,'(1X,A16,I2,A3,I2)' ) 'TRONCATION DE 19',IANTR,' A ',
$   IANTR+IDURTR-1
$   WRITE(4,'(1X,A16,I2,A3,I2)' ) 'TRONCATION DE 19',IANTR,' A ',
$   IANTR+IDURTR-1
ELSEIF (ITRONC.EQ.2) THEN
$   WRITE(*,'(1X,A27,I2,A3,I2)' ) 'MOYENNES SAISONNIERES DE 19',
$   IANTR,' A ',IANTR+IDURTR-1
$   WRITE(4,'(1X,A27,I2,A3,I2)' ) 'MOYENNES SAISONNIERES DE 19',
$   IANTR,' A ',IANTR+IDURTR-1
ELSEIF (ITRONC.EQ.3.OR.ITRONC.EQ.0) THEN
$   WRITE(*,'(1X,A40)' ) 'IDENTIFIEES ; TRAITEMENT SANS CHANGEMENT '
$   WRITE(4,'(1X,A40)' ) 'IDENTIFIEES ; TRAITEMENT SANS CHANGEMENT '
ELSE
$   WRITE(*,'(A6)' ) 'AUCUNE '
$   WRITE(4,'(A6)' ) 'AUCUNE '
ENDIF
WRITE(*,*) ' '
WRITE(4,*) ' '
IF (IBT.NE.0) THEN
```



```
        WRITE(*,'(1X,A43\)' ) 'METHODE EMPLOYEE POUR COMPLETER LA SERIE : '  
        WRITE(4,'(1X,A43\)' ) 'METHODE EMPLOYEE POUR COMPLETER LA SERIE : '  
        IF (IBT.EQ.1) THEN  
            WRITE(*,*) ' MOYENNES SAISONNIERES '  
            WRITE(4,*) ' MOYENNES SAISONNIERES '  
        ELSEIF (IBT.EQ.2) THEN  
            WRITE(*,*) ' INTERPOLATION '  
            WRITE(4,*) ' INTERPOLATION '  
        ELSE  
            WRITE(*,*) ' RELATION C/Q '  
            WRITE(4,*) ' RELATION C/Q '  
        ENDIF  
    ENDIF  
    WRITE(*,*) ' '  
    WRITE(4,*) ' '  
    IF (IDIV.EQ.0.AND.IBIN.EQ.1) MD=1  
    IF (IBIN.NE.0) THEN  
        * WRITE(*,18) 'INTERVALLE DE TRAVAIL :   MOIS DE DEPART  
        *   : ',MD  
        * WRITE(4,18) 'INTERVALLE DE TRAVAIL :   MOIS DE DEPART  
        *   : ',MD  
    ENDIF  
    IF (IBIN.EQ.0.AND.IDIV.EQ.0) THEN  
        * WRITE(*,18) '                               INTERVALLE (EN JOURS)  
        *   : ',INTERV  
        * WRITE(4,18) '                               INTERVALLE (EN JOURS)  
        *   : ',INTERV  
    ELSE  
        * WRITE(*,18) '                               INTERVALLE (EN MOIS)  
        *   : ',INTERV  
        * WRITE(4,18) '                               INTERVALLE (EN MOIS)  
        *   : ',INTERV  
    ENDIF  
    WRITE(*,18) '                               NOMBRE D'INTERVALLES PAR A  
    *   NNEE : ',NINTAN  
    WRITE(4,18) '                               NOMBRE D'INTERVALLES PAR A  
    *   NNEE : ',NINTAN  
18  FORMAT(1X,A60,I3)  
    WRITE(*,*) ' '  
    WRITE(*,'(1X,A15\)' ) 'PERSISTANCE : '  
    WRITE(4,*) ' '  
    WRITE(4,'(1X,A15\)' ) 'PERSISTANCE : '  
    IF (IPERS.EQ.0) THEN  
        WRITE(*,'(A7,9X\)' ) 'AUCUNE '  
        WRITE(4,'(A7,9X\)' ) 'AUCUNE '  
    ELSEIF (IPERS.EQ.1) THEN  
        WRITE(*,'(A11,5X\)' ) 'MARKOVIENNE '  
        WRITE(4,'(A11,5X\)' ) 'MARKOVIENNE '  
    ELSE  
        WRITE(*,'(A15,1X\)' ) 'NON-MARKOVIENNE '  
        WRITE(4,'(A15,1X\)' ) 'NON-MARKOVIENNE '  
    ENDIF  
    WRITE(*,'(1X,A15\)' ) 'SAISONNALITE : '  
    WRITE(4,'(1X,A15\)' ) 'SAISONNALITE : '  
    IF (ISAIS.EQ.0.OR.ISAIS.EQ.2) THEN
```

```
      WRITE(*,'(A5\)' ) 'NON '
      WRITE(4,'(A5\)' ) 'NON '
ELSE
      WRITE(*,'(A5\)' ) 'OUI '
      WRITE(4,'(A5\)' ) 'OUI '
ENDIF
WRITE(*,'(1X,A13\)' ) ' UTILISEE : '
WRITE(4,'(1X,A13\)' ) ' UTILISEE : '
IF (IDIV.EQ.1.OR.ISAIS.EQ.3) THEN
      WRITE(*,'(A3)' ) 'OUI '
      WRITE(4,'(A3)' ) 'OUI '
ELSE
      WRITE(*,'(A3)' ) 'NON '
      WRITE(4,'(A3)' ) 'NON '
ENDIF
```

C *** retracer les dates correspondant aux bornes specifiees dans p5
c *** (limites de la sous-serie a traiter)

```
      CALL TRANSDATE(IDIV,N1+IEX1-1,IBIN,LOW,NINTAN,INTERV,MD,
$      INTER,NINT,AN1,MO1,JR1)
      IF (ITRONC.EQ.1) THEN
$          N2T=N2+IDURTR*NINTAN
$          CALL TRANSDATE(IDIV,N2T+IEX1-1,IBIN,LOW,NINTAN,INTERV,MD,
$          INTER,NINT,AN2,MO2,JR2)
      ELSE
$          CALL TRANSDATE(IDIV,N2+IEX1-1,IBIN,LOW,NINTAN,INTERV,MD,
$          INTER,NINT,AN2,MO2,JR2)
$          N2T=N2
      ENDIF
      WRITE(*,*) ' '
      WRITE(*,19) 'SERIE DE TRAVAIL :      DATE DE DEPART      : ',JR1,'/',
$      MO1,'/',AN1,'      (RANG = ',N1,') '
      WRITE(*,19) '      DATE DE FIN      : ',JR2,'/',
$      MO2,'/',AN2,'      (RANG = ',N2T,') '
19  FORMAT(1X,A42,I2,A1,I2,A1,I2,A11,I4,A1)
      WRITE(*,'(A42,I4)' ) '      NOMBRE DE POINTS : ',
$      LONG
      WRITE(4,*) ' '
      WRITE(4,19) 'SERIE DE TRAVAIL :      DATE DE DEPART      : ',JR1,'/',
$      MO1,'/',AN1,'      (RANG = ',N1,') '
      WRITE(4,19) '      DATE DE FIN      : ',JR2,'/',
$      MO2,'/',AN2,'      (RANG = ',N2T,') '
      WRITE(4,'(A42,I4)' ) '      NOMBRE DE POINTS : ',
$      LONG
      CALL QCMOV(1,0)
      CALL QCHAR(27,ITFORE,1)
      CALL QBEEP
      READ(*,'(A1)' ) TOUCHE
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QCMOV(0,23)
      WRITE(*,*) ' '
```

c *** retracer les noms des test suggeres et utilises

```
CALL NOMTEST(NOSUGG,NOM1)
CALL NOMTEST(NOTEST,NOM2)
WRITE(*,23)'TEST SUGGERE : ',NOM1,
WRITE(*,23)'TEST UTILISE : ',NOM2
23  FORMAT(1X,A16,A30)
    WRITE(*,*)' '
    WRITE(*,20)'RESULTATS DU TEST UTILISE :   VALEUR DE LA STATISTIQUE
$   : ',STAT
20  FORMAT(1X,A58,F10.4)
    WRITE(*,20)'                               VALEUR DU TEST
$   : ',VALTEST
    WRITE(4,23)'TEST SUGGERE : ',NOM1,
    WRITE(4,23)'TEST UTILISE : ',NOM2
    WRITE(4,*)' '
    WRITE(4,20)'RESULTATS DU TEST UTILISE :   VALEUR DE LA STATISTIQUE
$   : ',STAT
    WRITE(4,20)'                               VALEUR DU TEST
$   : ',VALTEST
    IF (NOTEST.EQ.6.OR.NOTEST.EQ.12) THEN
        WRITE(*,21)'                               DEGRES DE
$         LIBERTE
        WRITE(*,21)'                               DEGRES DE
$         LIBERTE
        WRITE(*,21)'                               DEGRES DE
$         LIBERTE
        IF (ITEND.EQ.1) COMSIG='OUI'
        IF (ITEND.EQ.0) COMSIG='NON'
        WRITE(*,22)'                               TEST SIGNIFICATIF
$         ?
        WRITE(*,22)'                               TEST SIGNIFICATIF
$         ?
        WRITE(*,22)'                               TEST SIGNIFICATIF
$         ?
        WRITE(*,22)'                               TEST SIGNIFICATIF
$         ?
        GOTO 25
    ENDIF
21  FORMAT(1X,A58,I10)
22  FORMAT(1X,A58,A3)
    WRITE(*,20)'                               SIGNIFICATION DU TEST
$   : ',SIGNIF
    WRITE(4,20)'                               SIGNIFICATION DU TEST
$   : ',SIGNIF
25  WRITE(*,*)' '
    WRITE(*,24)'COMMENTAIRE : ',COMMEN
    WRITE(4,*)' '
    WRITE(4,24)'COMMENTAIRE : ',COMMEN
24  FORMAT(19X,A14,A39)
c *** ouverture du fichier de travail et lecture de la serie
26  FICH='
    FICH(1:LG)=NOMFICH(1:LG)
    IF (ITYP.EQ.0) THEN
        FICH(LG+1:LG+4)='.TPC'
```

```
ELSE
    FICH(LG+1:LG+4)='.TPL'
ENDIF
OPEN(5,FILE=FICH,STATUS='OLD')
READ(5,30)(VECSER(I),I=1,LONG)
30  FORMAT(E16.10)

c *** selon le test utilise ,la tendance detectee et la division de la
c *** serie, produire differentes statistiques :

IF (ITEND.EQ.0) THEN
c *** calcul de la moyenne et de l'ecart-type
$
    CALL STATGLOB(VECSER,TAMP,LONG,N1,IEX1,MD,INTER,NINT,
        VALVIDE,MAXI,NOTEST,AN1,IVAR,NINTAN,LOW)
    IF (IDIV.EQ.1.OR.ISAIS.EQ.3) THEN
c *** calcul des moyennes et ecarts-types pour chaque saison
$
        CALL STATSAIS(VECSER,TAMP,LONG,N1,IEX1,MD,INTER,NINT,
            VALVIDE,MAXI)
    ENDIF
c *** graphique de l'ajustement par la moyenne seulement
    CALL QCMOV(1,0)
    CALL QCHAR(27,ITFORE,1)
    CALL QBEEP
    READ(*,'(A1)') TOUCHE
    CALL GRAFAJUS(VECSER,TAMP,LONG,IVAR,1,N1,NINTAN,IEX1)
ELSE
c *** Une tendance a ete detectee
    IF (NOTEST.LE.2) THEN
c *** il s'agit d'un test de mann-whitney non-saisonnier
c *** donner les statistiques avant et apres le saut
$
        CALL STAMW(IDIV,IVAR,IBIN,LOW,NINTAN,INTERV,MD,INTER,
            NINT,N1,IEX1,LONG,VECSER)
c *** graphique de l'ajustement avec saut
    CALL QCMOV(1,0)
    CALL QCHAR(27,ITFORE,1)
    CALL QBEEP
    READ(*,'(A1)') TOUCHE
    CALL GRAFAJUS(VECSER,TAMP,LONG,IVAR,2,N1,NINTAN,IEX1)
    ENDIF
    IF (NOTEST.LE.4.AND.NOTEST.GT.2) THEN
```

```
c *** il s'agit d'un test de mw saisonnier
c *** donner les statistique avant et apres le saut,pour chaque saison
$
      CALL STAMWSAIS (AN1,N1,N2,IEX1,MD,INTERV,NINTAN,IDIV,IVAR,
        VECSER,TAMP,VALVIDE)

      CALL @CMOV(1,0)
      CALL @CHAR(27,ITFORE,1)
      CALL @BEEP
      READ(*,'(A1)') TOUCHE

      CALL GRAFAJUS (VECSER,TAMP,LONG,IVAR,2,N1,NINTAN,IEX1)
    -ENDIF

      IF (NOTEST.GT.4) THEN
c *** il s'agit d'un test pour detecter une tendance monotone
      CALL MONOTONE (VECSER,LONG,NINTAN)
c *** graphique de l'ajustement par la droite de regression
      CALL @CMOV(1,0)
      CALL @CHAR(27,ITFORE,1)
      CALL @BEEP
      READ(*,'(A1)') TOUCHE

      CALL GRAFAJUS (VECSER,TAMP,LONG,IVAR,3,N1,NINTAN,IEX1)

      IF (NOTEST.EQ.7.OR.NOTEST.EQ.8) THEN
c *** il s'agit d'un test pour tendance monotone avec saisonnalite
      CALL MONOSAIS (VECSER,TAMP,LONG,N1,IEX1,MD,INTERV,
        NINTAN,
        IDIV,VALVIDE,MAXI)
      -ENDIF
    -ENDIF
  -ENDIF
  OPEN (3,FILE='SYNTHESE.P6',STATUS='NEW')
  REWIND(4)
50  READ (4,13,END=60)LIGNE
    WRITE (3,13)LIGNE
    GOTO 50
60  CLOSE (3)
    CLOSE (4,STATUS='DELETE')
    CLOSE (5)

c *** ouverture d'un fichier bidon pour signaler que l'execution de p6
c *** s'est deroulee normalement

  OPEN (99,FILE='P6.OK',STATUS='NEW')
  CLOSE (99)
  END
```

```
      SUBROUTINE STATGLOB(VECSER,TAMP,LONG,N1,IEX1,MD,INTER,NINT,  
* VALVIDE,MAXI,NOTEST,AN1,NP1,NINTAN,LOW)
```

```
C *****
```

```
C -----
```

```
C Fonction:
```

```
C . Fournit la moyenne generale et l'ecart-type de l'ensemble de  
C la serie (sortie a l'ecran)
```

```
C Parametres:
```

```
C . vecser Vecteur contenant la serie  
C . long Longueur de la serie
```

```
C -----
```

```
      DIMENSION VECSER(*)  
      REAL ECTYP  
      INTEGER ANS,AN1  
      CHARACTER*8 FORM1,FORM2  
      COMMON /BLOC61/ FORM1,FORM2  
      COMMON /CONTINF/ VI
```

```
      SOM=0  
      SOM2=0
```

```
10 DO 10 I=1, LONG  
      SOM=SOM+VECSER(I)  
      SOM2=SOM2+(VECSER(I))**2
```

```
CONTINUE  
      VMOY=SOM/LONG  
      VARIANCE=SOM2/LONG-(VMOY**2)  
      ECTYP=VARIANCE**0.5  
      IF(NOTEST.GT.4)GOTO 12  
      IF(NOTEST.GT.2)GOTO 11
```

```
C *** retracer la date approximative de la separation
```

```
      WRITE(*,*) ' '  
      WRITE(4,*) ' '  
      CALL TRANSDATE(IDIV,NP1+IEX1+N1-1,IBIN,LOW,NINTAN,INTERV,MD,  
* INTER,NINT,ANS,MOS,JRS)  
      WRITE(*, '(A40,I2,A1,I2,A1,I2)') 'DATE APPROXIMATIVE DE LA SEPARATIO  
* N : ',JRS,'/',MOS,'/',ANS  
      WRITE(*, '(A37,I4)') 'RANG ASSOCIE : ',NP1+N1  
      WRITE(4, '(A40,I2,A1,I2,A1,I2)') 'DATE APPROXIMATIVE DE LA SEPARATIO  
* N : ',JRS,'/',MOS,'/',ANS  
      WRITE(4, '(A37,I4)') 'RANG ASSOCIE : ',NP1+N1  
      GOTO 12
```

```
C *** retracer l'annee de separation
```

```
11 IANSEP=AN1+NP1  
      WRITE(*,13) 'ANNEE DE SEPARATION : 19',IANSEP
```

```
13      WRITE(4,13)'ANNEE DE SEPARATION : 19',IANSEP
      FORMAT(A25,I2)

c *** impression des resultats
12      WRITE(*,*)' '
      WRITE(*,15)'MOYENNE GLOBALE : '
      WRITE(4,*)' '
      WRITE(4,15)'MOYENNE GLOBALE : '
15      FORMAT(A19,9X\ )
      IF (VMOY.LE.999999.AND.ECTYP.LE.999999) THEN
          WRITE(*,FORM1)VMOY
          WRITE(*,15)'ECART-TYPE      : '
          WRITE(*,FORM1)ECTYP
          EM=(ECTYP/(LONG**0.5))*VI**0.5
          WRITE(*,'(A28,F10.3)')'ECART-TYPE DE LA MOYENNE : ',EM
          WRITE(4,FORM1)VMOY
          WRITE(4,15)'ECART-TYPE      : '
          WRITE(4,FORM1)ECTYP
          WRITE(4,'(A28,F10.3)')'ECART-TYPE DE LA MOYENNE : ',EM
      ELSE
          WRITE(*,'(E16.10)')VMOY
          WRITE(*,15)'ECART-TYPE      : '
          WRITE(*,'(E16.10)')ECTYP
          EM=(ECTYP/(LONG**0.5))*VI**0.5
          WRITE(*,'(A28,E16.10)')'ECART-TYPE DE LA MOYENNE : ',EM
          WRITE(4,'(E16.10)')VMOY
          WRITE(4,15)'ECART-TYPE      : '
          WRITE(4,'(E16.10)')ECTYP
          WRITE(4,'(A28,E16.10)')'ECART-TYPE DE LA MOYENNE : ',EM
      ENDIF
      RETURN
      END
```

```
      SUBROUTINE STATSAIS(VECSER,TAMP,LONG,N1,IEX1,MD,INTER,NINT,
$      VALVIDE,MAXI)
```

```
C      * * * * *
```

```
C      -----
C      Fonction:
```

```
C      . Dans le cas ou l'usager a redivise les saisons pour sa serie,
C      . fournit la moyenne et l'ecart-type de chacune des saisons
```

```
C      Parametres:
```

```
C      . vecser   Vecteur contenant la serie
C      . tamp    Vecteur auxiliaire #
C      . long    Longueur de la serie
C      . n1      Premiere borne choisie par l'usager
C      . iex1    Indice de la premiere valeur non-absente de la serie
C      .         a partir du 1er janvier de la 1ere annee
C      . md      Mois de depart de la 1ere saison
C      . inter   Intervalle (duree de la saison)
C      . nint    Nombre d'intervalles par annee
```

c . valvide Valeur attribuee aux valeurs absentes
c . maxi Longueur maximale de la serie

c

```
-----  
DIMENSION VECSER(*),TAMP(*)  
REAL VAR(365),MOY(365)  
INTEGER*2 MAXI,NB(365)  
CHARACTER*8 FORM1,FORM2  
CHARACTER*1 TOUCHE  
COMMON /BLOC61/ FORM1,FORM2  
COMMON /TEXTE/ITMODE,ITFORE,ITBACK,ITCOLOR  
COMMON /CONTINF/ VI
```

c *** completer le vecteur des observations aux extremités avec des
c *** valeurs absentes de facon a ce qu'on puisse ensuite retracer les
c *** saisons facilement

```
DO 20 I=1,MAXI  
    TAMP(I)=VALVIDE  
20 CONTINUE  
    NESP=MOD((N1+IEX1-2),NINT)
```

c *** insertion des valeurs de la serie

```
DO 40 I=NESP+1,NESP+LONG  
    TAMP(I)=VECSER(I-NESP)  
40 CONTINUE  
    IF (MOD(NESP+LONG,NINT).EQ.0) THEN  
        LTOT=NESP+LONG  
    ELSE  
        LTOT=(NESP+LONG)+NINT-MOD((NESP+LONG),NINT)  
    ENDIF  
    NOMBRE=0  
    SOM=0
```

c *** calcul des statistiques pour chaque saison

```
DO 80 I=1,NINT  
    DO 60 J=0,LTOT/NINT-1  
        IF (TAMP(I+J*NINT).NE.VALVIDE) THEN  
            NOMBRE=NOMBRE+1  
            SOM=SOM+TAMP(I+J*NINT)  
            SOM2=SOM2+TAMP(I+J*NINT)**2  
        ENDIF  
    CONTINUE  
    IF (NOMBRE.NE.0) THEN  
        MOY(I)=SOM/NOMBRE  
        VAR(I)=SOM2/NOMBRE-(MOY(I)**2)  
        NB(I)=NOMBRE  
        NOMBRE=0  
        SOM=0  
        SOM2=0  
    ELSE
```



```
c *** cas ou il n'y a aucune valeur pour cette saison
      MOY(I)=VALVIDE
      NB(I)=0
      ENDIF
80 CONTINUE
   IF (NINT.GT.6) THEN
c *** on doit utiliser une nouvelle page d'ecran
      CALL QCMOV(1,0)
      CALL QCHAR(27,ITFORE,1)
      CALL QBEEP
      READ(*,'(A1)') TOUCHE
      CALL QCLEAR(ITBACK,ITFORE)
      CALL QCMOV(0,23)
   ENDIF

c *** impression des resultats

   WRITE(*,*) '
   WRITE(*,*) 'SAISON           MOYENNE      ECART-TYPE    EC-TYPE DE '
   WRITE(*,*) '
   WRITE(*,*) '-----          -----          -----
   WRITE(*,*) '
   WRITE(4,*) '
   WRITE(4,*) 'SAISON           MOYENNE      ECART-TYPE    EC-TYPE DE '
   WRITE(4,*) '
   WRITE(4,*) '-----          -----          -----
   WRITE(4,*) '
   DO 100 JJ=1,NINT
      MFIN=MOD(MD+JJ*INTER,12)-1
      IF (MFIN.EQ.-1) MFIN=11
      IF (MFIN.EQ.0) MFIN=12
      IF (INTER.EQ.1) THEN
         WRITE(*,'(A5,1X,I2,5X\)' ) 'MOIS',MD+(JJ-1)*INTER
         WRITE(4,'(A5,1X,I2,5X\)' ) 'MOIS',MD+(JJ-1)*INTER
      ELSE
         WRITE(*,84) 'MOIS',MD+(JJ-1)*INTER,' A ',MFIN
         WRITE(4,84) 'MOIS',MD+(JJ-1)*INTER,' A ',MFIN
      ENDIF
      IF (MOY(JJ).LE.999999.AND.VAR(JJ).LE.999999) THEN
         WRITE(*,'(3X\)' )
         WRITE(*,FORM2)MOY(JJ)
         WRITE(*,'(3X\)' )
         WRITE(*,FORM2)VAR(JJ)**0.5
         WRITE(*,'(3X\)' )
         EM=((VAR(JJ)**0.5)/(NB(JJ)**0.5))*VI**0.5
         WRITE(*,FORM1)EM
         WRITE(4,'(3X\)' )
         WRITE(4,FORM2)MOY(JJ)
         WRITE(4,'(3X\)' )
         WRITE(4,FORM2)VAR(JJ)**0.5
         WRITE(4,'(3X\)' )
```

```

      ELSE
      WRITE (4,FORM1)EM
      WRITE (*,'(3X\)' )
      WRITE (*,85)MOY(JJ)
      WRITE (*,'(3X\)' )
      WRITE (*,85)VAR(JJ)**0.5
      WRITE (*,'(3X\)' )
      EM=((VAR(JJ)**0.5)/(NB(JJ)**0.5))*VI**0.5
      WRITE (*,90)EM
      WRITE (4,'(3X\)' )
      WRITE (4,85)MOY(JJ)
      WRITE (4,'(3X\)' )
      WRITE (4,85)VAR(JJ)**0.5
      WRITE (4,'(3X\)' )
      WRITE (4,90)EM
    ENDIF
84   FORMAT (A5,1X,I2,A3,I2\ )
85   FORMAT (E16.10\ )
90   FORMAT (E16.10)
100  CONTINUE
      RETURN
      END
```

```

SUBROUTINE STAMW(IDIV,NP1,IBIN,LOW,NINTAN,INTERV,MD,INTER,
$ NINT,N1,IEX1,LONG,VECSE)
```

```

C *****
```

Fonction:

. Dans le cas ou l'usager a choisi un test de Mann et Withney non-saisonnier, fournit les moyennes et ecarts-type avant et apres le saut

Parametres:

- . idiv Indique quelle methode l'usager a employee pour diviser sa serie
- . ibin Indique si la division s'est faite en jours ou en mois
- . low Annee de depart de la serie originale
- . nintan Nombre d'intervalles par annee (cas standard)
- . Interv Intervalle choisi (cas standard)
- . md Mois de depart de la 1ere saison
- . inter Intervalle (duree de la saison)
- . nint Nombre d'intervalles par annee
- . n1 Premiere borne choisie par l'usager
- . iex1 Indice de la premiere valeur non-absente de la serie a partir du 1er janvier de la 1ere annee
- . long Longueur de la serie
- . vecser Vecteur contenant la serie

INTEGER ANS
CHARACTER*1 TOUCHE

```
DIMENSION VECSER(*)  
CHARACTER*8 FORM1,FORM2  
COMMON /BLOC61/ FORM1,FORM2  
COMMON /TEXTE/ITMODE,ITFORE,ITBACK,ITCOLOR  
COMMON /SAUT/ SAV,ECTAVANT,SAP,ECTAPRES  
COMMON /CONTINF/ VI
```

c *** retracer la date approximative du saut

```
WRITE(*,*) ' '  
CALL TRANSDATE(IDIV,NP1+IEX1+N1-1,IBIN,LOW,NINTAN,INTERV,MD,  
$ INTER,NINT,ANS,MOS,JRS)  
WRITE(*,'(A37,I2,A1,I2,A1,I2)') 'DATE APPROXIMATIVE DU SAUT  
$ : ',JRS,'/',MOS,'/',ANS  
WRITE(*,'(A37,I4)') 'RANG ASSOCIE : ',NP1+N1  
WRITE(*,*) ' '  
WRITE(4,*) ' '  
WRITE(4,'(A37,I2,A1,I2,A1,I2)') 'DATE APPROXIMATIVE DU SAUT  
$ : ',JRS,'/',MOS,'/',ANS  
WRITE(4,'(A37,I4)') 'RANG ASSOCIE : ',NP1+N1  
WRITE(4,*) ' '  
  
SAV=0  
SAP=0  
SAV2=0  
SAP2=0  
NP2=LONG-NP1
```

c *** calculer moyenne et ecart-type avant le saut

```
DO 40 I=1,NP1  
    SAV=SAV+VECSER(I)  
    SAV2=SAV2+(VECSER(I))**2  
40 CONTINUE
```

c *** calculer moyenne et ecart-type apres le saut

```
DO 60 I=NP1+1,LONG  
    SAP=SAP+VECSER(I)  
    SAP2=SAP2+(VECSER(I))**2  
60 CONTINUE  
WRITE(*,*) ' '  
ECTAVANT=(SAV2/NP1-(SAV/NP1)**2)**0.5  
WRITE(*,70) 'MOYENNE AVANT LE SAUT : '  
WRITE(4,*) ' '  
WRITE(4,70) 'MOYENNE AVANT LE SAUT : '  
IF (SAV/NP1.LE.999999.AND. ECTAVANT.LE.999999) THEN  
    WRITE(*,FORM1)SAV/NP1  
    WRITE(*,70) 'ECART-TYPE : '  
    WRITE(*,FORM1)ECTAVANT  
    EM=(ECTAVANT/(NP1**0.5))*VI**0.5  
    WRITE(*,'(A25,F10.3)') 'EC-TYPE DE LA MOYENNE : ',EM  
    WRITE(4,FORM1)SAV/NP1  
    WRITE(4,70) 'ECART-TYPE : '  
    WRITE(4,FORM1)ECTAVANT
```

```
ELSE WRITE(4,'(A25,F10.3)')'EC-TYPE DE LA MOYENNE : ',EM
WRITE(*,75)SAV/NP1
WRITE(*,70)'ECART-TYPE : '
WRITE(*,75)ECTAVANT
EM=(ECTAVANT/(NP1**0.5))*VI**0.5
WRITE(*,'(A25,E16.10)')'EC-TYPE DE LA MOYENNE : ',EM
WRITE(4,75)SAV/NP1
WRITE(4,70)'ECART-TYPE : '
WRITE(4,75)ECTAVANT
WRITE(4,'(A25,E16.10)')'EC-TYPE DE LA MOYENNE : ',EM
ENDIF
70 FORMAT(A25\ )
75 FORMAT(E16.10)
WRITE(*,*)' '
WRITE(*,*)' '
ECTAPRES=(SAP2/NP2-(SAP/NP2)**2)**0.5
WRITE(*,70)'MOYENNE APRES LE SAUT : '
WRITE(4,*)' '
WRITE(4,*)' '
WRITE(4,70)'MOYENNE APRES LE SAUT : '
IF (SAV/NP2.LE.999999.AND.ECTAPRES.LE.999999) THEN
WRITE(*,FORM1)SAP/NP2
WRITE(*,70)'ECART-TYPE : '
WRITE(*,FORM1)ECTAPRES
EM=(ECTAPRES/(NP2**0.5))*VI**0.5
WRITE(*,'(A25,F10.3)')'EC-TYPE DE LA MOYENNE : ',EM
WRITE(4,FORM1)SAP/NP2
WRITE(4,70)'ECART-TYPE : '
WRITE(4,FORM1)ECTAPRES
WRITE(4,'(A25,F10.3)')'EC-TYPE DE LA MOYENNE : ',EM
ELSE
WRITE(*,75)SAP/NP2
WRITE(*,70)'ECART-TYPE : '
WRITE(*,75)ECTAPRES
EM=(ECTAPRES/(NP2**0.5))*VI**0.5
WRITE(*,'(A25,E16.10)')'EC-TYPE DE LA MOYENNE : ',EM
WRITE(4,75)SAP/NP2
WRITE(4,70)'ECART-TYPE : '
WRITE(4,75)ECTAPRES
WRITE(4,'(A25,E16.10)')'EC-TYPE DE LA MOYENNE : ',EM
ENDIF
RETURN
END
```

SUBROUTINE STAMWSAIS(AN1,N1,N2,IEX1,MD,INTERV,NINTAN,IDIV,NP1,
\$ VECSEK,TAMP,VALVIDE)

C * * * * *

C
C -----
C Fonction:

C . Dans le cas ou l'usager a choisi un test de Mann et Withney
C saisonnier, fournit les moyennes et ecarts-type de chaque

```

c      saison ,avant et apres le saut
c
c      Parametres:
c
c      . an1      Annee correspondant a la 1ere borne
c      . n1,n2    Borne choisie par l'usager pour delimitier sa serie
c      . iex1     Indice de la premiere valeur non-absente de la serie
c      . md       Mois de depart de la 1ere saison
c      . interv   Intervalle
c      . nintan   Nombre d'intervalles par annee
c      . idiv     Indique quelle methode l'usager a employee pour
c                diviser sa serie
c      . np1      Longueur du premier segment
c      . vecser   Vecteur contenant la serie
c      . tamp     Vecteur auxiliaire #
c      . valvide  Valeur attribuee aux valeurs manquantes

```

```

-----
c
c      INTEGER AN1
c      CHARACTER*1 TOUCHE
c      CHARACTER*8 FORM1,FORM2
c      COMMON /BLOC61/ FORM1,FORM2
c      COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR
c      COMMON /SAUT/ SAV,ECTAVANT,SAP,ECTAPRES
c      COMMON /CONTINF/ VI
c      DIMENSION TAMP(*),VECSER(*)

```

```

c *** retracer l'annee choisie pour le saut

```

```

c      IANSEP=AN1+NF1
c      CALL QCMOV(1,0)
c      CALL QCHAR(27,ITFORE,1)
c      CALL QBEEP
c      READ(*,'(A1)') TOUCHE
c      CALL QCLEAR(ITBACK,ITFORE)
c      CALL QCMOV(0,23)
c      WRITE(*,5) 'ANNEE DE SEPARATION : 19',IANSEP
c      WRITE(4,5) 'ANNEE DE SEPARATION : 19',IANSEP
5  FORMAT (A25,I2)
c      WRITE(*,*) ' '
c      WRITE(4,*) ' '

```

```

c *** completer le vecteur des observations aux extremités avec des
c *** valeurs absentes de facon a ce qu'on puisse ensuite retracer les
c *** saisons facilement

```

```

c      DO 10 I=1,(MOD(N1+IEX1-2,NINTAN))
c          TAMP(I)=VALVIDE
10  CONTINUE
c      IDEBUT=(MOD(N1+IEX1-2,NINTAN))+1
c      DO 20 J=IDEBUT,IDEDEUT+N2-N1
c          TAMP(J)=VECSER(J-IDEDEUT+1)
20  CONTINUE
c      K1=IDEDEUT+N2-N1+1

```

```
IF (MOD(N2+IEX1-1,NINTAN).EQ.0) THEN  
    K2=0  
ELSE  
    K2=IDEBUT+N2-N1+(NINTAN-MOD(N2+IEX1-1,NINTAN))  
ENDIF  
DO 30 I=K1,K2  
    TAMP(I)=VALVIDE  
30 CONTINUE  
    LTOT=I-1
```

c *** si on a choisi la facon standard de diviser la serie,le mois de
c *** depart est considere=1

```
IF (IDIV.EQ.0) THEN  
    MDD=1  
ELSE  
    MDD=MD  
ENDIF
```

c *** traitement des moyennes saisonnieres et globale avant le saut

```
WRITE(*,*) 'AVANT LE SAUT :'  
WRITE(*,*) 'SAISON          MOYENNE      ECART-TYPE      EC-TYPE DE'  
WRITE(*,*) '          -----          -----          LA MOYENNE'  
WRITE(*,*) '          '          '          '          '  
WRITE(4,*) 'AVANT LE SAUT :'  
WRITE(4,*) 'SAISON          MOYENNE      ECART-TYPE      EC-TYPE DE'  
WRITE(4,*) '          -----          -----          LA MOYENNE'  
WRITE(4,*) '          '          '          '          '  
WRITE(4,*) '          '          '          '          '
```

```
SAV=0  
SAP=0  
SAVI=0  
S2=0  
SI2=0  
NB=0  
NBI=0
```

```
DO 50 I=1,NINTAN  
    DO 40 J=0,NP1-1  
        IF (TAMP(I+J*NINTAN).NE.VALVIDE) THEN  
            SAVI=SAVI+TAMP(I+J*NINTAN)  
            SAV=SAV+TAMP(I+J*NINTAN)  
            SAVI2=SAVI2+TAMP(I+J*NINTAN)**2  
            SAV2=SAV2+TAMP(I+J*NINTAN)**2  
            NBAVI=NBAVI+1  
            NBAV=NBAV+1  
        ENDIF  
    CONTINUE  
40  
45    FORMAT(A5,1X,I2,A3,I2,2X\  
    ECTYPI=(SAVI2/NBAVI-(SAVI/NBAVI)**2)**0.5  
    MFIN=MOD(MDD+I*INTERV,12)-1
```

c *** impression des resultats saisonniers

```
IF (MFIN.EQ.-1) MFIN=11
```

```
      IF (MFIN.EQ.0) MFIN=12
      IF (INTERV.EQ.1) THEN
        WRITE (*, '(A5,1X,I2,5X\)' ) 'MOIS',MDD+(I-1)*INTERV
        WRITE (4, '(A5,1X,I2,5X\)' ) 'MOIS',MDD+(I-1)*INTERV
      ELSE
        WRITE (*,45) 'MOIS',MDD+(I-1)*INTERV, ' A ',MFIN
        WRITE (4,45) 'MOIS',MDD+(I-1)*INTERV, ' A ',MFIN
      ENDIF
      IF (SAVI/NBAVI.LE.999999.AND.ECTYPI.LE.999999) THEN
        WRITE (*, '(3X\)' )
        WRITE (*,FORM2)SAVI/NBAVI
        WRITE (*, '(3X\)' )
        WRITE (*,FORM2)ECTYPI
        WRITE (*, '(3X\)' )
        EM=(ECTYPI/(NBAVI**0.5))*VI**0.5
        WRITE (*,FORM1)EM
        WRITE (4, '(3X\)' )
        WRITE (4,FORM2)SAVI/NBAVI
        WRITE (4, '(3X\)' )
        WRITE (4,FORM2)ECTYPI
        WRITE (4, '(3X\)' )
        WRITE (4,FORM1)EM
      ELSE
        WRITE (*, '(3X\)' )
        WRITE (*,56)SAVI/NBAVI
        WRITE (*, '(3X\)' )
        WRITE (*,56)ECTYPI
        WRITE (*, '(3X\)' )
        EM=(ECTYPI/(NBAVI**0.5))*VI**0.5
        WRITE (*,55)EM
        WRITE (4, '(3X\)' )
        WRITE (4,56)SAVI/NBAVI
        WRITE (4, '(3X\)' )
        WRITE (4,56)ECTYPI
        WRITE (4, '(3X\)' )
        WRITE (4,55)EM
      ENDIF
      SAVI=0
      SAVI2=0
      NBAVI=0
50 CONTINUE
55 FORMAT (E16.10)
56 FORMAT (E16.10\ )
   WRITE (*,*) ' '
   ECTYP=(SAV2/NBAV-(SAV/NBAV)**2)**0.5
```

c *** impression des statistiques globales

```
   WRITE (*, '(A32\)' ) 'MOYENNE GLOBALE AVANT SAUT : '
   WRITE (4,*) ' '
   WRITE (4, '(A32\)' ) 'MOYENNE GLOBALE AVANT SAUT : '
   IF (SAV/NBAV.LE.999999.AND.ECTYP.LE.999999) THEN
     WRITE (*,FORM1)SAV/NBAV
     SAVSAFE=SAV
     WRITE (*, '(A32\)' ) 'ECART-TYPE GLOBAL AVANT SAUT : '
```

```
WRITE(*,FORM1)ECTYP
WRITE(*,'(A32\)' )'EC-TYPE DE LA MOYENNE      : '
EM=(ECTYP/(NBAV**0.5))*VI**0.5
WRITE(*,FORM1)EM
WRITE(4,FORM1)SAV/NBAV
WRITE(4,'(A32\)' )'ECART-TYPE GLOBAL AVANT SAUT : '
WRITE(4,FORM1)ECTYP
WRITE(4,'(A32\)' )'EC-TYPE DE LA MOYENNE      : '
WRITE(4,FORM1)EM
ECTAVANT=ECTYP
ELSE
WRITE(*,55)SAV/NBAV
WRITE(*,'(A32\)' )'ECART-TYPE GLOBAL AVANT SAUT : '
WRITE(*,55)ECTYP
WRITE(*,'(A32\)' )'EC-TYPE DE LA MOYENNE      : '
EM=(ECTYP/(NBAV**0.5))*VI**0.5
WRITE(*,55)EM
WRITE(4,55)SAV/NBAV
WRITE(4,'(A32\)' )'ECART-TYPE GLOBAL AVANT SAUT : '
WRITE(4,55)ECTYP
WRITE(4,'(A32\)' )'EC-TYPE DE LA MOYENNE      : '
WRITE(4,55)EM
ENDIF
IF (NINTAN.GT.3) THEN
CALL QCMOV(1,1)
CALL QCHAR(27,ITFORE,1)
CALL QBEEP
READ(*,'(A1)' ) TOUCHE
CALL QCLEAR(ITBACK,ITFORE)
CALL QCMOV(1,23)
ENDIF
```

c *** traitement des moyennes saisonnieres et globale apres le saut

```
WRITE(*,*)'APRES LE SAUT : '
WRITE(*,*)'SAISON          MOYENNE      ECART-TYPE      EC-TYPE DE '
WRITE(*,*)'              -----      -----      LA MOYENNE '
WRITE(*,*)'              '              '              '
WRITE(4,*)'APRES LE SAUT : '
WRITE(4,*)'SAISON          MOYENNE      ECART-TYPE      EC-TYPE DE '
WRITE(4,*)'              -----      -----      LA MOYENNE '
WRITE(4,*)'              '              '              '
WRITE(4,*)'              '              '              '
SAV=0
SAVI=0
SAV2=0
SAVI2=0
NBAV=0
NBAVI=0
DO 70 I=1,NINTAN
  DO 60 J=NF1,(LTOT/NINTAN)-1
    IF (TAMP(I+J*NINTAN).NE.VALVIDE) THEN
      SAVI=SAVI+TAMP(I+J*NINTAN)
      SAV=SAV+TAMP(I+J*NINTAN)
```



```

        SAVI2=SAVI2+TAMP(I+J*NINTAN)**2
        SAV2=SAV2+TAMP(I+J*NINTAN)**2
        NBAVI=NBAVI+1
        NBAV=NBAV+1
    ENDIF
60 CONTINUE
    ECTYPI=(SAVI2/NBAVI-(SAVI/NBAVI)**2)**0.5
    MFIN=MOD(MDD+I*INTERV,12)-1
c *** Impression des resultats saisonniers
    IF (MFIN.EQ.-1) MFIN=11
    IF (MFIN.EQ.0) MFIN=12
    IF (INTERV.EQ.1) THEN
        WRITE(*,'(A5,1X,I2,5X\)' ) 'MOIS',MDD+(I-1)*INTERV
        WRITE(4,'(A5,1X,I2,5X\)' ) 'MOIS',MDD+(I-1)*INTERV
    ELSE
        WRITE(*,45) 'MOIS',MDD+(I-1)*INTERV,' A ',MFIN
        WRITE(4,45) 'MOIS',MDD+(I-1)*INTERV,' A ',MFIN
    ENDIF
    IF (SAVI/NBAVI.LE.999999.AND.ECTYPI.LE.999999) THEN
        WRITE(*,'(3X\)' )
        WRITE(*,FORM2)SAVI/NBAVI
        WRITE(*,'(3X\)' )
        WRITE(*,FORM2)ECTYPI
        WRITE(*,'(3X\)' )
        EM=(ECTYPI/(NBAVI**0.5))*VI**0.5
        WRITE(*,FORM1)EM
        WRITE(4,'(3X\)' )
        WRITE(4,FORM2)SAVI/NBAVI
        WRITE(4,'(3X\)' )
        WRITE(4,FORM2)ECTYPI
        WRITE(4,'(3X\)' )
        WRITE(4,FORM1)EM
    ELSE
        WRITE(*,'(3X\)' )
        WRITE(*,56)SAVI/NBAVI
        WRITE(*,'(3X\)' )
        WRITE(*,56)ECTYPI
        WRITE(*,'(3X\)' )
        EM=(ECTYPI/(NBAVI**0.5))*VI**0.5
        WRITE(*,55)EM
        WRITE(4,'(3X\)' )
        WRITE(4,56)SAVI/NBAVI
        WRITE(4,'(3X\)' )
        WRITE(4,56)ECTYPI
        WRITE(4,'(3X\)' )
        WRITE(4,55)EM
    ENDIF
    SAVI=0
    SAVI2=0
    NBAVI=0
70 CONTINUE
    WRITE(*,*) ' '
    ECTYP=(SAV2/NBAV-(SAV/NBAV)**2)**0.5
```

c *** impression des statistiques globales

```
WRITE(*,'(A32\)' ) 'MOYENNE GLOBALE APRES SAUT : '  
WRITE(4,*) ' '  
WRITE(4,'(A32\)' ) 'MOYENNE GLOBALE APRES SAUT : '  
IF (SAV/NBAV.LE.999999.AND.ECTYP.LE.999999) THEN  
    WRITE(*,FORM1)SAV/NBAV  
    WRITE(*,'(A32\)' ) 'ECART-TYPE GLOBAL APRES SAUT : '  
    WRITE(*,FORM1)ECTYP  
    WRITE(4,FORM1)SAV/NBAV  
    WRITE(4,'(A32\)' ) 'ECART-TYPE GLOBAL APRES SAUT : '  
    WRITE(4,FORM1)ECTYP  
    WRITE(*,'(A25\)' ) 'EC-TYPE DE LA MOYENNE : '  
    EM=(ECTYP/(NBAV**0.5))*VI**0.5  
    WRITE(*,FORM1)EM  
    WRITE(4,'(A25\)' ) 'EC-TYPE DE LA MOYENNE : '  
    WRITE(4,FORM1)EM  
ELSE  
    WRITE(*,55)SAV/NBAV  
    WRITE(*,'(A32\)' ) 'ECART-TYPE GLOBAL APRES SAUT : '  
    WRITE(*,55)ECTYP  
    WRITE(4,55)SAV/NBAV  
    WRITE(4,'(A32\)' ) 'ECART-TYPE GLOBAL APRES SAUT : '  
    WRITE(4,55)ECTYP  
    WRITE(*,'(A25\)' ) 'EC-TYPE DE LA MOYENNE : '  
    EM=(ECTYP/(NBAV**0.5))*VI**0.5  
    WRITE(*,55)EM  
    WRITE(4,'(A25\)' ) 'EC-TYPE DE LA MOYENNE : '  
    WRITE(4,55)EM  
    ECTAPRES=ECTYP  
ENDIF  
SAP=SAV  
SAV=SAVSAFE  
RETURN  
END
```

SUBROUTINE MONOTONE(X,N,NINTAN)

C * * * * *

c -----
c Fonction:
c
c . Calcul la droite de regression associee a la serie equidistantes
c . versus le temps (1,2,...,N)
c
c Parametres:
c . x Vecteur contenant la serie equidistante
c . n Longueur de la serie
c . nintan Nombre d'intervalles par annee
c -----

REAL X(*)

```
INTEGER*4 I  
CHARACTER*3 ON  
CHARACTER*8 FORM1,FORM2  
COMMON /BLOC61/ FORM1,FORM2  
COMMON /REGRE/Y1,YN,PENTE,SCM
```

c *** initialisations a zero

```
SOMPROD=0  
SOMXI=0  
SOMYI=0  
SOMXI2=0
```

c *** calcul des sommes et des sommes de carres

```
DO 10 I=1,N  
    SOMPROD=SOMPROD+X(I)*I  
    SOMYI=SOMYI+X(I)  
    SOMXI=SOMXI+I  
    SOMXI2=SOMXI2+I**2  
10 CONTINUE
```

c *** calcul des moyennes et des estimateurs de la regression

```
XBAR=SOMXI/N  
YBAR=SOMYI/N  
VARX=SOMXI2/N-((SOMXI/N)**2)  
BCHAP=(SOMPROD-(XBAR*YBAR*N))/(VARX*N)  
ACHAP=YBAR-BCHAP*XBAR  
Y1=ACHAP+BCHAP*I  
YN=ACHAP+BCHAP*N  
PENTE=BCHAP*NINTAN
```

c *** calcul de la somme des carres moyens pour cet ajustement

```
SCM=0  
DO 15 I=1,N  
    YCHAP=ACHAP+BCHAP*I  
    SCM=SCM+(YCHAP-X(I))**2  
15 CONTINUE  
SCM=SCM/N  
VARBCHAP=SCM/(VARX*N)  
VARPENTE=VARBCHAP*NINTAN**2
```

c *** impression des resultats

```
WRITE(*,'(/,/)' )  
WRITE(*,20)'PENTE DE LA TENDANCE : ',PENTE,'/AN      ECART-TYPE: '  
$   ,VARPENTE**0.5  
WRITE(4,'(/,/)' )  
WRITE(4,20)'PENTE DE LA TENDANCE : ',PENTE,'/AN      ECART-TYPE: '  
$   ,VARPENTE**0.5  
20  FORMAT (A24,F9.3,A19,F13.3)  
WRITE(*,*)'  
WRITE(*,'(A3B\)' )'NIVEAU INITIAL DE LA SERIE TRAITEE : '
```

```
WRITE(4,*) ' '
WRITE(4,'(A38\)' ) 'NIVEAU INITIAL DE LA SERIE TRAITEE : '
IF (Y1.LE.999999) THEN
    WRITE(*,FORM1)Y1
    WRITE(4,FORM1)Y1
ELSE
    WRITE(*,25)Y1
    WRITE(4,25)Y1
ENDIF
25 FORMAT(E16.10)
WRITE(*,*) ' '
WRITE(*,'(A38\)' ) 'NIVEAU FINAL DE LA SERIE TRAITEE : '
WRITE(4,*) ' '
WRITE(4,'(A38\)' ) 'NIVEAU FINAL DE LA SERIE TRAITEE : '
IF (YN.LE.999999) THEN
    WRITE(*,FORM1)YN
    WRITE(4,FORM1)YN
ELSE
    WRITE(*,25)YN
    WRITE(4,25)YN
ENDIF
RETURN
END
```

```
$INCLU DE: 'P6PLUS.FOR'
$INCLU DE: 'TDATE.FOR'
$INCLU DE: 'TICS.FOR'
$INCLU DE: 'AJUSTE.FOR'
```

```
GRAFAJUS          P6PLUS.FOR   181  
SUBROUTINE GRAFAJUS(X,T, LONG,NP1,ITY,N1,
```

```
—MIN,MAX,FLOAT,QSMODE,QCOLOR,QOVSCN,TICS  
—AJUSTE,QPLOT,QXAXIS,QYAXIS,QGTXT,QPTXTC  
—QPTXTA,QSETUP,QTABL,QRTOI,QLINE,ELSEIF  
—QCMOV,QCHAR,QBEEP,QCLEAR,QBORD
```

```
MONOSAIS          P6PLUS.FOR    1  
SUBROUTINE MONOSAIS(VECSER,TAMP, LONG,N1,
```

```
—MOD,QCLEAR,QCMOV,QCHAR,QBEEP
```

```
NOMTEST           P6PLUS.FOR   404  
SUBROUTINE NOMTEST(NOTEST,NOM)
```

```
—ELSEIF
```

```
          SUBROUTINE MONOSAIS(VECSER,TAMP,LONG,N1,IEX1,MD,INTERV,NINTAN,  
          $    IDIV,VALVIDE,MAXI)  
C      * * * * *
```

c Fonctions:
c
c . calcul des droites de regression pour chaque saison
c
c Parametres:
c
c . vecser Vecteur contenant la serie
c . tamp Vecteur auxiliaire #
c . long Longueur de la serie
c . n1 Premiere borne choisie par l'usager
c . iex1 Indice de la premiere valeur non-absente de la serie
c a partir du 1er janvier de la 1ere annee
c . md Mois de depart de la 1ere saison
c . interv Intervalle
c . nintan Nombre d'intervalles par annee
c . idiv Indique la methode employee pour diviser la serie
c . valvide Valeur attribuee aux valeurs absentes
c . maxi Longueur maximale de la serie

```
DIMENSION VECSER(*),TAMP(*)  
REAL PENTE(365),Y1(365),YN(365),SOUSVEC(100)  
INTEGER*2 MAXI  
CHARACTER*8 FORM1,FORM2  
CHARACTER*1 TOUCHE  
COMMON /BLOC61/ FORM1,FORM2  
COMMON /TEXTE/ITMODE,ITFORE,ITBACK,ITCOLOR
```

```
c *** completer le vecteur des observations aux extremités avec des  
c *** valeurs absentes de facon a ce qu'on puisse ensuite retracer les  
c *** saisons facilement
```

```
20 DO 20 I=1,MAXI  
    TAMP(I)=VALVIDE  
CONTINUE  
    NESP=MOD((N1+IEX1-2),NINTAN)  
40 DO 40 I=NESP+1,NESP+LONG  
    TAMP(I)=VECSER(I-NESP)  
CONTINUE  
    IF (MOD(NESP+LONG,NINTAN).EQ.0) THEN  
        LTOT=NESP+LONG  
    ELSE  
        LTOT=(NESP+LONG)+NINTAN-MOD((NESP+LONG),NINTAN)  
    ENDIF  
DO 80 I=1,NINTAN  
    J=0  
    DO 60 L=0,LTOT/NINTAN-1  
        IF (TAMP(I+L*NINTAN).NE.VALVIDE) THEN
```

```

        J=J+1
        SOUSVEC(J)=TAMP(I+L*NINTAN)
        ENDIF
60      CONTINUE

      IF (J.NE.0) THEN
c*** initialisations a zero

        SOMPROD=0
        SOMXI=0
        SOMYI=0
        SOMXI2=0
        MAND=0

        DO 65 K=1,J
          SOMPROD=SOMPROD+SOUSVEC(K)*K
          SOMYI=SOMYI+SOUSVEC(K)
          SOMXI=SOMXI+K
          SOMXI2=SOMXI2+K**2
65      CONTINUE
        XBAR=(SOMXI)/J
        YBAR=SOMYI/J
        VARX=SOMXI2/J-((SOMXI/J)**2)
        BCHAP=(SOMPROD-(XBAR*YBAR*J))/(VARX*J)
        ACHAP=YBAR-BCHAP*XBAR
        Y1(I)=ACHAP+BCHAP*I
        YN(I)=ACHAP+BCHAP*J

c *** calcul de la pente

        ELSE
          PENTE(I)=BCHAP
          PENTE(I)=VALVIDE
          Y1(I)=VALVIDE
          YN(I)=VALVIDE
        ENDIF
80      CONTINUE

c      CALL QCMOV(1,0)
c      CALL QCHAR(27,ITFORE,1)
c      CALL QBEEP
c      READ(*,'(A1)') TOUCHE
        CALL QCLEAR(ITBACK,ITFORE)
        CALL QCMOV(0,23)

c *** impression des statistiques

        WRITE(*,*) '
        WRITE(*,*) '
$      ALEUR
        WRITE(*,*) 'SAISON (BASE ANNUELLE)
$      INALE
        WRITE(*,*) '-----
$      -----
        WRITE(*,*) '
        PENTE
        VALEUR
        V
        INITIALE
        F
        -----

```

```
WRITE(4,*) ' '
WRITE(4,*) '
$   ALEUR
WRITE(4,*) 'SAISON      (BASE ANNUELLE)      INITIALE
$   INALE
WRITE(4,*) '-----
$   -----
WRITE(4,*) ' '

```

c *** si on a choisi la facon standard de diviser la serie,le mois de
c *** depart est considere=1

```
IF (IDIV.EQ.0) THEN
    MDD=1
ELSE
    MDD=MD
ENDIF

DO 100 JJ=1,NINTAN
    MFIN=MOD(MDD+JJ*INTERV,12)-1
    IF (MFIN.EQ.-1) MFIN=11
    IF (MFIN.EQ.0) MFIN=12
    IF (INTERV.EQ.1) THEN
        WRITE(*,89) 'MOIS',MDD+(JJ-1)*INTERV
        WRITE(4,89) 'MOIS',MDD+(JJ-1)*INTERV
    ELSE
        WRITE(*,90) 'MOIS',MDD+(JJ-1)*INTERV,' A ',MFIN
        WRITE(4,90) 'MOIS',MDD+(JJ-1)*INTERV,' A ',MFIN
    ENDIF
    IF (PENTE(JJ).LE.999999.AND.Y1(JJ).LE.999999.AND.YN(JJ).LE.
$ 999999) THEN
        WRITE(*,'(3X\)' )
        WRITE(*,FORM2)PENTE(JJ)
        WRITE(*,'(7X\)' )
        WRITE(*,FORM2)Y1(JJ)
        WRITE(*,'(7X\)' )
        WRITE(*,FORM1)YN(JJ)
        WRITE(4,'(3X\)' )
        WRITE(4,FORM2)PENTE(JJ)
        WRITE(4,'(7X\)' )
        WRITE(4,FORM2)Y1(JJ)
        WRITE(4,'(7X\)' )
        WRITE(4,FORM1)YN(JJ)
    ELSE
        WRITE(*,94)PENTE(JJ)
        WRITE(*,'(1X\)' )
        WRITE(*,94)Y1(JJ)
        WRITE(*,'(1X)' )
        WRITE(*,95)YN(JJ)
        WRITE(4,94)PENTE(JJ)
        WRITE(4,'(1X\)' )
        WRITE(4,94)Y1(JJ)
        WRITE(4,'(1X)' )
        WRITE(4,95)YN(JJ)
    ENDIF

```



```
100 CONTINUE
89  FORMAT (A5,1X,I2,5X\ )
90  FORMAT (A5,1X,I2,A3,I2,1X\ )
94  FORMAT (E16.10\ )
95  FORMAT (E16.10)
    CALL @CMOV(1,0)
    CALL @CHAR(27,ITFORE,1)
    CALL @BEEP
    READ(*,'(A1)') TOUCHE
    RETURN
    END
```

SUBROUTINE GRAFAJUS(X,T, LONG, NP1, ITY, N1, NINTAN, IEX1)

Fonction :

. Trace le graphique des charges ou des concentrations
en fonction du temps et presente la ou les droites
d'ajustement

Parametres :

- . x Vecteur contenant serie de mesures
- . t # Vecteur destine a contenir les valeurs en abscisse
- . long Nombre d'observations du vecteur x
- . np1 Longueur du premier segment dans le cas d'un saut
- . ity Indique le type de graphique a effectuer
1=avec moyenne totale de toute la serie
2=avec moyennes avant et apres le saut
3=avec droite de regression
- . n1 Premiere borne choisie par l'usager

```
REAL X(*),T(*),X2(1000),T2(1000)
INTEGER ANDEF,ANIN
CHARACTER*10 TEXTEX,TEXTEY
CHARACTER*12 FICHDEB,NOMFICH
CHARACTER*16 F
CHARACTER*19 F2
COMMON /IMPR/ IMPRIM,NOTEST,ITYP
COMMON /TRONC/ ITRONC,IDURTR,IANTR
COMMON /TEXTE/ ITMODE,ITFORE,ITBACK,ITCOLOR
COMMON /GRAPH/ IGMODE,IGBACK,IPAL,IGBORD,IGCOLOR,ILINE,
# ILETR,ISEC,KIRSYM,ISYMBL,JROW1,JROW2,JCOL1,JCOL2
COMMON /SAUT/ SAV,ECTAVANT,SAP,ECTAPRES
COMMON /REGRE/Y1,YN,PENTE,SCM
```

c *** on trouve le maximum et le minimum rencontres

```
XMIN=1E30
XMAX=-1E30
XMOY=0.
XECT=0.
IF (NOTEST.EQ.3.OR.NOTEST.EQ.4) THEN
  NP1=NP1*NINTAN-IEX1
ENDIF
NP2=LONG-NP1
GNP1=NP1+N1
TROU=0
IF (ITRONC.EQ.1) THEN
  TROU=IDURTR*NINTAN
ENDIF
LONGT=LONG+TROU+N1-1
DO 10 I=1, LONG
  XMOY=XMOY+X(I)
  XECT=XECT+X(I)**2
  XMIN=MIN(XMIN,X(I))
  XMAX=MAX(XMAX,X(I))
  T(I)=I+N1-1
  IF (ITRONC.EQ.1.AND.I.GT.NP1) THEN
    T2(I-NP1)=T(I)+TROU
    X2(I-NP1)=X(I)
  ENDIF
10 CONTINUE
YMIN=FLOAT(N1-1)
YMAX=FLOAT(LONGT)
XMOY=XMOY/LONG
XECT=XECT/LONG-XMOY**2
AVMOY=XMOY
APMOY=XMOY
IF (ITY.EQ.2) THEN
  AVMOY=SAV/NP1
  APMOY=SAP/NP2
ENDIF

c *** configuration en mode graphique
CALL QSMODE(IGMODE)
IF (IGMODE.LT.10) THEN
  CALL QCOLOR(IGBACK,IPAL)
ELSE
  CALL QOVSCN(IGBORD)
ENDIF

c *** calcul de l'espacement des "tics"
CALL TICS(XMIN,XMAX,YSTS,YFINS,TICY,MINORY,AINTEY)
CALL TICS(YMIN,YMAX,XSTS,XFINS,TICX,MINORX,AINTEX)
IF (AINTEY.GE.10.OR.AINTEY.LT.1) THEN
  CALL AJUSTE(X,LONG,AINTEY,TEXTY,YSTS,YFINS,TICY)
  CALL AJUSTE(X2,NP2,AINTEY,0.,0.,0.,0.)
  XMOY=XMOY/AINTEY
  XMIN=XMIN/AINTEY
  XMAX=XMAX/AINTEY
```

```
        Y1=Y1/AINTEY  
        YN=YN/AINTEY  
        AVMOY=AVMOY/AINTEY  
        APMOY=APMOY/AINTEY  
-ENDIF  
-IF (AINTEX.GE.10.OR.AINTEX.LT.1) THEN  
        CALL AJUSTE(T, LONG, AINTEX, TEXTEX, XSTS, XFINS, TICX)  
        CALL AJUSTE(T2, NP2, AINTEX, 0., 0., 0., 0.)  
        YMIN=YMIN/AINTEX  
        YMAX=YMAX/AINTEX  
        GNP1=GNP1/AINTEX  
        TROU=(TROU+1)/AINTEX  
-ENDIF
```

c *** tracage de la courbe

```
        CALL @PLOT(JCOL1, JCOL2, JROW1, JROW2, XSTS, XFINS, YSTS, YFINS, XSTS,  
# YSTS, 0, 1.0, 1.5)  
        CALL @XAXIS(XSTS, XFINS, TICX, MINDRX, 1, 0)  
        CALL @YAXIS(YSTS, YFINS, TICX, MINORY, 1, 0)  
        CALL @GTXT(10, TEXTEX, ILETR, JCOL2-35, JROW1+10, 0)  
        CALL @PTXT(10, TEXTEY, ILETR)  
        CALL @PTXTA(15, 'TEMPS (# INTER)', ILETR)  
-IF (IGMODE.EQ.16) THEN  
        INCR=85  
-ELSE  
        INCR=55  
-ENDIF  
-IF (ITYP.EQ.0) THEN  
        CALL @GTXT(15, 'CONCEN mg/1', ILETR, 0, (JROW1+JROW2)/2+INCR, -1)  
-ELSE  
        CALL @GTXT(15, 'CHARGE Kg/jour', ILETR, 0, (JROW1+JROW2)/2+INCR, -1)  
-ENDIF  
        CALL @SETUP(0, ILINE, ISYMBL, KIRSYM)  
-IF (ITRONC.EQ.1.AND.ITY.LE.2) THEN  
        GNP2=GNP1+TROU  
        CALL @TABL(1, NP1, T, X)  
        CALL @TABL(1, NP2, T2, X2)  
        CALL @RTOI(0.0, AVMOY, I1, J1)  
        CALL @RTOI(YMAX, APMOY, I2, J2)  
        CALL @RTOI(GNP1, AVMOY, I3, J3)  
        CALL @RTOI(GNP2, APMOY, I4, J4)  
        CALL @QLINE(I1, J1, I3, J3, ILINE)  
        CALL @QLINE(I4, J4, I2, J2, ILINE)  
        GOTO 30  
-ENDIF  
        CALL @TABL(1, LONG, T, X)  
-IF (ITY.EQ.1) THEN  
        CALL @RTOI(YMIN, XMOY, I1, J1)  
        CALL @RTOI(YMAX, XMOY, I2, J2)  
        CALL @QLINE(I1, J1, I2, J2, ILINE)  
-ELSEIF (ITY.EQ.2) THEN  
        CALL @RTOI(YMIN, AVMOY, I1, J1)  
        CALL @RTOI(YMAX, APMOY, I2, J2)  
        CALL @RTOI(GNP1, APMOY, I3, J2)
```

```
CALL @LINE(I1,J1,I3,J1,ILINE)  
CALL @LINE(I3,J2,I2,J2,ILINE)  
-ELSEIF (ITY.EQ.3) THEN  
CALL @RTOI(YMIN,Y1,I1,J1)  
CALL @RTOI(YMAX,YN,I2,J2)  
CALL @LINE(I1,J1,I2,J2,ILINE)  
-ENDIF
```

c *** calcul du critere d'ajustement

```
30 -IF (ITY.EQ.1) THEN  
SCMA=XECT**0.5  
-ELSEIF (ITY.EQ.2) THEN  
SCMA=(( (ECTAVANT**2)*NP1+(ECTAPRES**2)*NP2)/LONG)**0.5  
-ELSEIF (ITY.EQ.3) THEN  
SCMA=SCM**0.5  
-ENDIF
```

c *** ecriture du critere d'ajustement

```
CALL @CMOV(10,24)  
-IF (SCMA.LT.1E6) THEN  
F='(7X,A43,F11.3) '  
F2='(/,29X,A6,F11.3,\) '  
-ELSE  
F='(7X,A43,E16.10) '  
F2='(/,29X,A6,E16.10,\) '  
-ENDIF  
WRITE(*,F)'RMSE POUR L"INTERVALLE DE TRAVAIL CHOISI : ',SCMA  
-IF (NOTEST.LE.4) THEN  
WRITE(4,*)'ADEQUATION DU MODELE DE TENDANCE PAR SAUT AUX  
DONNEES'  
WRITE(4,*)'POUR L''INTERVALLE DE TRAVAIL CHOISI : '  
-ELSE  
WRITE(4,*)'ADEQUATION DU MODELE DE TENDANCE MONOTONE AUX  
DONNEES'  
WRITE(4,*)'POUR L''INTERVALLE DE TRAVAIL CHOISI : '  
-ENDIF  
WRITE(4,F2)'RMSE : ',SCMA  
-IF (ITYP.EQ.0) THEN  
WRITE(4,*)'mg/l '  
-ELSE  
WRITE(4,*)'kg/J '  
-ENDIF  
WRITE(2,'(E16.10)')SCMA  
CALL @CMOV(1,1)  
IF (IMPRIM.EQ.1) CALL @PSCRN  
CALL @CHAR(27,ITFORE,1)  
CALL @BEEP  
READ(*,40) TOUCHE  
40 FORMAT(A1)
```

c *** remise en mode texte

```
CALL @SMODE(ITMODE)
```

```
CALL QCLEAR(ITBACK,ITFORE)  
CALL QBORD(ITCOLOR)
```

c *** retour aux valeurs non-ajustees pour les graphiques

```
IF (AINTEY.GE.10.OR.AINTEY.LT.1) THEN  
    AINTEY=1/AINTEY  
    CALL AJUSTE(X, LONG, AINTEY, TEXTY, YSTS, YFINS, TICY)  
    CALL AJUSTE(X2, NF2, AINTEY, 0., 0., 0., 0.)  
    XMOY=XMOY/AINTEY  
    XMIN=XMIN/AINTEY  
    XMAX=XMAX/AINTEY  
    Y1=Y1/AINTEY  
    YN=YN/AINTEY  
    AVMOY=AVMOY/AINTEY  
    APMOY=APMOY/AINTEY  
ENDIF  
RETURN  
END
```

```
SUBROUTINE NOMTEST(NOTEST,NOM)
```

```
*****
```

c
c -----
c
c Fonction:

c . Retracer le nom du test correspondant au numero passe en
c parametre

c
c Parametres:

c . notest Numero du test (E)
c . nom Nom du test associe (S)

c
c -----
c
c CHARACTER*30 NOM

```
IF (NOTEST.EQ.1) THEN  
    NOM='MANN-WHITNEY'  
ELSEIF (NOTEST.EQ.2) THEN  
    NOM='MANN-WHITNEY/LETTENMAIER'  
ELSEIF (NOTEST.EQ.3) THEN  
    NOM='MANN-WHITNEY SAISONNIER'  
ELSEIF (NOTEST.EQ.4) THEN  
    NOM='MANN-WHITNEY SAIS/LETTENMAIER'  
ELSEIF (NOTEST.EQ.5) THEN  
    NOM='KENDALL'  
ELSEIF (NOTEST.EQ.6) THEN  
    NOM='SPEARMAN/LETTENMAIER'  
ELSEIF (NOTEST.EQ.7) THEN  
    NOM='KENDALL SAISONNIER'  
ELSEIF (NOTEST.EQ.8) THEN  
    NOM='HIRSCH ET SLACK'  
ELSEIF (NOTEST.EQ.9) THEN
```

```

|      NOM='FOSTER ET STUART 1'
|-----ELSEIF (NOTEST.EQ.10) THEN
|      NOM='FOSTER ET STUART 2'
|-----ELSEIF (NOTEST.EQ.11) THEN
|      NOM='TAU DE KENDALL'
|-----ELSE
|      NOM='SPEARMAN'
|-----ENDIF
|      RETURN
|      END
```