الجمهورية الجزائرية الديمقراطية الشعبية

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ IBN-KHALDOUN DE TIARET

FACULTÉ DES SCIENCES APPLIQUÉES

LABORATOIRE DE RECHERCHE DES TECHNOLOGIES INDUSTRIELLES

DÉPARTEMENT DE GÉNIE MÉCANIQUE

Programmer en FORTRAN

(100 programmes exécutables en langage Fortran)

Préparé par Dr Aissat Sahraoui

Avant-propos

On trouve le langage FORTRAN " FORmula TRANslator " (traducteur d'équations) sous différents systèmes d'exploitation comme MS-DOS (ordinateurs compatibles IBM PC), UNIX (mini ordinateurs et gros systèmes), ainsi que sous WINDOWS.

Sous WINDOWS, le Fortran offre aux étudiants une puissance de calcul considérable, que l'on ne trouve que sur des stations de travail, et à un prix beaucoup plus élevé.

Un programme Fortran nécessite trois types de fichiers pour son élaboration :

- Les fichiers source (extension .FOR, .F90 sous MS-DOS ou WINDOWS, .f sous UNIX).
- Les fichiers objet (extension .OBJ sous MS-DOS, .o sous UNIX).
- Le fichier exécutable (extension .EXE sous MS-DOS ou WINDOWS, définie par l'utilisateur sous UNIX).

Le présent ouvrage est une approche par la pratique du langage Fortran. Cet ouvrage est un recueil de plus de 100 programmes exécutables en Fortran 77 et 90. Il s'adresse aux adeptes du fortran et en particulier aux étudiants qui s'intéressent à la programmation informatique.

La compréhension des programmes présentés dans cet ouvrage nécessite : une connaissance appropriée des problèmes posés et une connaissance des spécificités du langage Fortran.

Un rappel de quelques instructions du fortran, des fonctions numériques standards utilisées en Fortran, des opérateurs de comparaison et des opérateurs logiques sont donnés au début de ce travail.

Dr Aissat Sahraoui
Département de Génie Mécanique
Faculté des Sciences Appliquées
Université Ibn Khaldoun – Tiaret
Email : sh_aissat@yahoo.fr

SOMMAIRE

Page

1) Quelques instructions du fortran **1**

2) Fonctions numériques standards **2**

3) Opérateurs de comparaison **3**

4) Opérateurs logiques **3**

5) Quelques instructions relatives aux tableaux et aux matrices **3**

6) Programmes **4**

Programme 01 : Table de multiplication de 2 à 12 **4**

Programme 02 : Calcul de la quantité d'essence **4**

Programme 03 : Calcul du salaire hebdomadaire **4**

Programme 04 : Calcul des indemnités (its) **4**

Programme 05 : Réalisation d'une facture d'achats **5**

Programme 06 : Montant et reste de la monnaie **6**

Programme 07 : Saluer en fonction de la langue d'un pays **6**

Programme 08 : Jours de la semaine et programme de la journée 7

Programme 09 : Calcul de la distance entre deux points **7**

Programme 10 : Calcul de l'hypoténuse d'un triangle à angle droit **7**

Programme 11 : Calcul de l'aire d'un triangle (instruction **function**) 8

Programme 12 : Surface et périmètre d'un cercle et d'un rectangle **8**

Programme 13 : Carré magique **9**

Programme 14 : Incrémentation **10**

Programme 15 : Conversion degrés en radians **10**

Programme 16 : Table de conversion degrés en radians **11**

Programme 17 : Conversion heures, minutes, secondes == total secondes **11**

Programme 18 : Conversion du temps en secondes en heures, minutes et secondes **11**

Programme 19 : Conversion degrés Fahrenheit en degrés Celsius **12**

Programme 20 : Conversion des duretés (HB, HRC et HV) **12**

Programme 21 : Calcul du PGCD et du PPCM de deux entiers m et n **14**

Programme 22 : Conditions de coupe (tournage, fraisage, …etc) **14**

Programme 23 : Calcul de la valeur de la pression autour d'une aile **15**

Programme 24 : Calcul de $S= \sum(a_i^2+b_i^2)/a_i * b_i$ **16**

Programme 25 : Calcul de $\begin{cases} y=1/(1+x) & \text{si } x \leq 0 \\ y=-(x^2+1)/100 & \text{si } 0 < x \leq 5 \\ y=15/4x & \text{si } 5 < x \leq 30 \\ y=1/8 & \text{si } x > 30 \end{cases}$ **16**

Programme 26 : Calcul de $Y=\begin{cases} 2x^2+3 & \text{si } x < 0 \\ 3-4x & \text{si } 0 \leq x \leq 2 \\ 3-2x^2 & \text{si } x > 2 \end{cases}$ (x=1, 10, 0.1) **17**

Programme 27 : Températures inferieures à 20°C, supérieures à 40°C et leurs moyennes **17**

Programme 28 : Somme des entiers impairs **18**

Programme 29 : Somme des entiers pairs **18**

Programme 30 : Produit de n entiers **18**

Programme 31 : Produit de n entiers impairs **19**

Programme 32 : Somme, différence, produit et quotient de deux réels (avec opérande) **19**

Programme 33 : Somme des carrés $S=1^2+2^2+3^2+\cdots\cdots\cdots\cdots 20^2$ **19**

Programme 34 : Somme $S=1+\frac{1}{2}+\frac{1}{3}+\cdots\cdots\cdots\cdots+\frac{1}{n}$ n entier **20**

Programme 35 : Somme $S=\frac{1}{2}+\frac{1}{4}+\frac{1}{6}+\frac{1}{8}+\cdots\cdots\cdots\cdots+\frac{1}{n}$ n entier **20**

Programme 36 : Calcul itératif de la factorielle d'un entier naturel n **20**

Programme 37 : Somme de la factorielle d'un entier $S=\sum_{i=1}^{n} i!$ **20**

Programme 38 : Somme $S=1+\frac{1}{2!}+\frac{1}{3!}+\cdots\cdots\cdots\cdots+\frac{1}{n!}$ **21**

Programme 39 : Somme $S=\frac{2}{1!}+\frac{3}{2!}+\frac{4}{3!}+\cdots\cdots\cdots\cdots+\frac{n}{(n-1)!}$ **21**

Programme 40 : Somme $S=\frac{2!}{4!}+\frac{4!}{6!}+\frac{6!}{8!}+\cdots\cdots\cdots\cdots+\frac{(2n)!}{(2n+2)!}$ **22**

Programme 41 : Quotient $Q=\frac{1!\times 2!\times 3!\times\cdots\cdots\times n!}{1!+2!+3!+\cdots\cdots+n!}$ **22**

Programme 42 : Calcul de la moyenne arithmétique (**subroutine**) **23**

Programme 43 : Calcul de la variance et de l'écart type **23**

Programme 44 : Fonctions trigonométriques **24**

Programme 45 : Méthode des rectangles **24**

Programme 46 : Comparaison de deux nombres complexes Z1 et Z2 **25**

Programme 47 : Conjugué d'un nombre complexe Z **25**

Programme 48 : Module d'un nombre complexe $|Z|$ **25**

Programme 49 : Produit d'un réel α $(\alpha \in R)$ par un nombre complexe Z $(Z \in C)$ **26**

Programme 50 : Somme et différence de deux nombres complexes **26**

Programme 51 : Produit de deux nombres complexes **27**
Programme 52 : Quotient de deux nombres complexes **27**
Programme 53 : Division de deux nombres complexes et affichage sous forme a+ib et expo **28**
Programme 54 : Somme de deux polynômes $C(x)=A(x)+B(x)$ **29**
Programme 55 : Produit de deux polynômes $C(x)=A(x)*B(x)$ **29**
Programme 56 : Lecture des éléments d'un tableau (vecteur) **30**
Programme 57 : Lecture des éléments d'une matrice **30**
Programme 58 : Plus grande valeur d'un tableau (instruction **MAX**) **30**
Programme 59 : Plus petite valeur d'un tableau (instruction **MIN**) **31**
Programme 60 : Lecture des éléments d'un tableau et leur décompte par signe **31**
Programme 61 : Somme des éléments impairs d'un tableau **31**
Programme 62 : Tri des éléments d'un vecteur dans le sens croissant et décroissant (triabulle) **32**
Programme 63 : Notes des stagiaires (nombre de notes saisies, calcul de la moyenne) **32**
Programme 64 : Classement des notes par module, calcul de la moyenne et nombre d'étudiants ayant réussis l'examen du module **33**
Programme 65 : Moyenne des éléments positifs et négatifs d'un tableau **34**
Programme 66 : Notes à partir d'une liste d'étudiants et observation **34**
Programme 67 : Plus grand élément d'un tableau et choix de sa position **35**
Programme 68 : Recherche des prénoms d'un tableau **35**
Programme 69 : Somme des éléments d'une matrice (tableau à deux dimensions) **36**
Programme 70 : Triangle de Pascal **36**
Programme 71 : Table de Pythagore **37**
Programme 72 : Produit scalaire de deux tableaux **37**
Programme 73 : Produit scalaire de deux vecteurs **38**
Programme 74 : Produit vectoriel de deux vecteurs **38**
Programme 75 : Produit mixte de trois vecteurs **39**
Programme 76 : Déterminant d'ordre 2 **39**
Programme 77 : Déterminant d'ordre 3 **40**
Programme 78 : Déterminant d'ordre N **40**
Programme 79 : Somme et différence de deux matrices **41**
Programme 80 : Produit de deux matrices **42**
Programme 81 : Produit de deux matrices == utilisation de l'instruction **matmul** **43**
Programme 82 : Transposée d'une matrice **44**
Programme 83 : Transposée d'une matrice == utilisation de l'instruction **transpose** **44**

Programme 84 : Inversion d'une matrice carrée par pivot **45**
Programme 85 : Inversion d'une matrice par DOOLITTLE **45**
Programme 86 : Résolution de l'équation du premier degré $Ax+B=0$ **47**
Programme 87 : Résolution de l'équation du second ordre $ax^2+bx+c \ (a \neq 0)$ **47**
Programme 88 : Calcul de la racine carrée d'un entier par l'expression $y=((x+a)/x)/2$ **48**
Programme 89 : Approximation de la racine carrée de 2 $(\sqrt{2})$ par la méthode de Newton **48**
Programme 90 : Suite de Fibonacci **48**
Programme 91 : Calcul de l'intégrale d'une fonction par la méthode des trapèzes **49**
Programme 92 : Résolution d'un système de deux équations à deux inconnues (CRAMER) **50**
Programme 93 : Élimination de GAUSS (système d'équations linéaires) **50**
Programme 94 : Résolution d'un système d'équations Linéaires par CHOLESKY **52**
Programme 95 : Résolution d'un système d'équations Linéaires par CHOLESKY_Bis **54**
Programme 96 : Résolution d'un système d'équations Ax=B par DOOLITTLE (factorisation LU) **56**
Programme 97 : Réponse d'un système à un ddl en vibrations libres amorties **57**
Programme 98 : Fréquences propres et modes propres des lignes d'arbres non ramifies libres-libres **58**
Programme 99 : Fréquences propres et modes propres des lignes d'arbres non ramifies encastrées-libres **60**
Programme 100 : Fréquences propres et modes propres des lignes d'arbres non ramifies encastrées-encastrées **61**
Autres programmes **63**
Bibliographie **80**

1) Quelques instructions du fortran

ENTRÉES-SORTIES

READ : Lecture

WRITE : Écriture

PRINT : Écriture (F90)

COMMENTAIRES

C, c, * ou **!** : En colonne 1 sont des commentaires (F77).

Le caractère **!** : Rencontré sur une ligne indique que ce qui suit est un commentaire (F90).

DÉCLARATIONS

INTEGER : Déclaration pour variables entières

REAL : Déclaration de type réel

DOUBLE PRECISION : Déclaration des variables de type réel double précision

LOGICAL : Déclaration des variables de type logique

COMPLEX : Déclaration des variables de type complexe

DIMENSION : Déclaration pour tableaux

DATA : Initialisation de variables

PARAMETER : Donne un nom à une constante

PROGRAMMES ET SOUS-PROGRAMMES

PROGRAM : Début de programme (F90)

SUBROUTINE : nom d'un sous-programme

CALL : Appelle et exécute un sous-programme

RETURN : Retour de sous-programme ou fonction

FICHIERS

CLOSE : Fermeture de fichier

OPEN : Ouverture de fichier

STRUCTURES ALTERNATIVES

IF : Structure alternative SI

ELSE : Sinon de la structure SI... ALORS... SINON

END IF : Fin construction alternée SI.

BOUCLES

DOWHILE : Boucle tant que (DOWHILE ……END DO)

DO : Boucle pour ou répéter ……. jusqu'à (DO …….END DO)

CONTINUE : Fin de la boucle (F77)

ENDDO : Fin de la boucle (F90)

GOTO : Branchement à un endroit particulier du programme

EXIT : Sortie anticipée d'une boucle

CYCLE : Rompre le déroulement normal d'une boucle

2) Fonctions numériques standards

ABS (x)	Renvoie la valeur absolue de x réel
IABS (x)	Renvoie la valeur absolue de x entier
SQRT (x)	Renvoie la valeur de la racine carrée de x
REAL (x)	Renvoie le résultat de la conversion de la valeur de x en réel.
AIMAG (z)	Renvoie la partie imaginaire d'un complexe z
MOD (x1, x2)	Renvoie le reste de la division de x1par x2 différent de 0
MAX (xi, ...)	Renvoie la valeur maximale de deux arguments ou plus.
MIN (xi, ...)	Renvoie la valeur minimale de deux arguments ou plus.
EXP (x)	Renvoie la valeur de l'exponentielle de x
LOG (x)	Renvoie la valeur du logarithme népérien (naturel)
LOG10 (x	Renvoie la valeur du logarithme décimal (à base 10)
SIN (x)	Renvoie le sinus de x, exprimé en radians
COS (x)	Renvoie le cosinus de x, exprimé en radians
TAN (x)	Renvoie la tangente de x, exprimé en radians
ASIN (x)	Renvoie la valeur de arc sin(x) exprimée en radians
ACOS (x)	Renvoie la valeur de arc cos(x) exprimée en radians
ATAN (x)	Renvoie la valeur de arc tg(x) exprimée en radians
SINH (x)	Renvoie la valeur de sh(x), exprimé en radians
COSH (x)	Renvoie la valeur de ch(x), exprimé en radians
TANH (x)	Renvoie la valeur de th(x), exprimé en radians

3) Les opérateurs de comparaison

FORTRAN 77	Signification	**FORTRAN 90**
.GT.	Plus grand que	>
.GE.	Plus grand ou égal que	> =
.EQ.	égal	= =
.NE.	Différent que	**!** =
.LE.	Inferieur ou égal à	< =
.LT.	Inférieur à	<

4) Les opérateurs logiques

FORTRAN 77 (90)	Signification
.OR.	OU
.AND.	ET
.XOR.	OU exclusif
.NOT.	NON

5) Quelques instructions relatives aux tableaux et aux matrices

MAXVAL (array)	Fournit la plus grande valeur du tableau array. Le résultat est du même type que les éléments du tableau array
MINVAL (array)	Fournit la plus petite valeur du tableau array
PRODUCT (array)	Fournit le produit des éléments du tableau array
SUM (array)	Fournit la somme des éléments du tableau array
DOTPRODUCT (vecta, vectb)	Fournit le produit scalaire de deux vecteurs vecta et vectb ceci est équivalent à SUM (vecta*vectb)
MATMUL(mata, matb)	Fournit une matrice contenant le produit matriciel de deux matrices mata et matb
TRANSPOSE (mata)	Matrice contenant la transposée de mata

6) Programmes

PROGRAMME N°1

```
PROGRAM  TABLESDEMULTIPLICATION
DO  I  =  2,  12
PRINT  *,'TABLE  DE  MULTIPLICATION  ----->',  I
PRINT*
DO  J  =  1,  12
PRINT  *,I,'x',J,'=',I*J
ENDDO
ENDDO
PRINT*
END
```

PROGRAMME N°2

```
PROGRAM  ESSENCE
INTEGER  STOPS,PLEIN
PRINT  *,'LA  DISTANCE  PARCOURUE  EN  KM'
READ  *,  KM
PRINT  *,'LE  NOMBRE  DES  ARRETS'
  READ  *,  STOPS
  PRINT  *,'QUANTITE  DE  DEPART'
  READ  *,  PLEIN
QUTL  =  40*STOPS  +  PLEIN
!  CALCULE  L'ESSENCE  UTILISEE  ET  LA  CONVERTIT  EN  REAL
  KPL  =  KM/QUTL  +  0.5
!  0.5  EST  AJOUTE  POUR  S'ASSURER  QUE  LE  RESULTAT  EST  ARRONDI
PRINT  *
PRINT  *,  'CONSOMMATION  MOYENNE  EN  Km/l  --->',KPL,'    ','LITRES'
END
```

PROGRAMME N°3

```
PROGRAM  SALAIREHEBDO
PRINT  *,  'DONNER  LE  NOMBRE  HEURES  SEMAINE'
READ  *,  HRS
PRINT  *,  'DONNER  LE  TAUX  STANDARD  AU  POSTE  EN  D.A.'
READ  *,  TAUX
IF  (HRS.LE.40)  THEN
SALAIRE  =  HRS*TAUX
ELSE  IF  (HRS.LE.50)  THEN
SALAIRE  =  40.0*TAUX  +  (HRS-40.0)*TAUX*1.5
ELSE
SALAIRE  =  40.0*TAUX  +  10.0*TAUX*1.5  +  (HRS-50.0)*TAUX*2.0
END  IF
PRINT  *,'LE  SALAIRE  HEBDO  EST',SALAIRE,'      ','DA'
END
```

PROGRAMME N°4

```
PROGRAM  INDEMNITES
REAL  ITS,S
CHARACTER*15  SITUATION
PRINT*,'DONNER  LE  NOMBRE  DE  SALARIES'
READ*,  N
```

```
I=1
DO WHILE(I<=N)
PRINT*,'DONNER ITS EN DINARS'
READ*, ITS
PRINT*,'DONNER LE SALAIRE S EN DINARS'
READ*, S
IF(S<1000) GOTO 1
PRINT*,'LA SITUATION DU SALARIE'
READ*, SITUATION
IF(SITUATION=='CELIBATAIRE') THEN
ITS=ITS*1.4
ELSE
IF(SITUATION=='MARIE') THEN
ITS=ITS*1.8
ENDIF
ENDIF
1 PRINT*,'ITS=',ITS
I=I+1
ENDDO
END
```

PROGRAMME N°5

```
PROGRAM FACTURE
CHARACTER(20) ARTICLE
INTEGER QUANTITE
REAL MONTANT
DIMENSION ARTICLE(50),QUANTITE(50),PRIXUNITAIRE(50),MONTANT(50)
TOTAL=0.
PRINT*,'LE NOMBRE D''ARTICLES'
READ*,N
PRINT*,'DONNER CES ARTICLES'
DO I=1,N
PRINT*,'ARTICLE',I,'';READ*,ARTICLE (I)
PRINT*,'DONNER SA QUANTITE' ; READ*,QUANTITE(I)
PRINT*,'DONNER SON PRIX UNITAIRE EN DA' ; READ*,PRIXUNITAIRE(I)
MONTANT(I)=PRIXUNITAIRE(I)*QUANTITE(I)
TOTAL=TOTAL+MONTANT(I)
ENDDO
TVA=TOTAL*0.17
TTC=TOTAL+TVA
PRINT 1,'NOM :', ' .....'
PRINT 1,'PRENOM : ','.....'
1 FORMAT(2X,A8,2X,A30,/,20X,A8,2X,A30)
PRINT 2,'FACTURE'
2 FORMAT(25X,A8,/,26X,7(1H=),/,2X,77(1H*))
PRINT 3,'ARTICLE','DESIGNATION','QUANTITE','PRIX UNITAIRE','MONTANT'
3FORMAT(2X,1(1H*),A8,1(1H*),3X,1(1H*),A12,1(1H*),7X,1(1H*),A9,1(1H*),3X,1(1H*),A14,1(1H*),3X,1(1H*),A8,
1(1H*),/,2X,77(1H*))
DO I=1,N
PRINT 4,'ARTICLE',I,ARTICLE(I),QUANTITE(I),PRIXUNITAIRE(I),MONTANT(I)
4FORMAT(2X,1(1H*),A7,I2,1(1H*),4X,A16,6X,1(1H*),I3,1(1H*),10X,1(1H*),F8.2,1(1H*),6X,1(1H*),F8.2,1(1H*),/,2
X,77(1H*))
ENDDO
PRINT 5,'TOTAL : ', TOTAL,'TVA 17 % : ',TVA,'TTC :',TTC
5 FORMAT(56X,A8,2X,F12.2,/,56X,A10,F12.2,/,55X,A6,5X,F12.2)
END
```

PROGRAMME N°6

```
PROGRAM MONTANT_RESTE
INTEGER E,SOMDUE,M,RESTE,NB10D,NB5D,NB2D
E=1
SOMDUE=0
DO WHILE (E.NE.0)
PRINT*,'ENTREZ LE MONTANT : '
READ*, E
SOMDUE=SOMDUE+E
END DO
PRINT*,'VOUS DEVEZ :', SOMDUE, ' DINARS'
PRINT*,'MONTANT VERSE :'
READ*,M
RESTE=M-SOMDUE
NB10D=0
DO WHILE (RESTE>=10)
NB10D=NB10D+1
RESTE=RESTE-10
END DO
NB5D=0
IF(RESTE>=5) THEN
NB5D=1
RESTE=RESTE-5
ENDIF
NB2D=0
IF(RESTE>=2) THEN
NB2D=1
RESTE=RESTE-2
ENDIF
PRINT*,'------> RENDU DE LA MONNAIE : <------'
PRINT*,'PIECES DE 10 DINARS :', NB10D
PRINT*,'PIECES DE 5 DINARS : ', NB5D
PRINT*,'PIECES DE 2 DINARS : ', NB2D
PRINT*,'PIECES DE 1 DINARS : ', RESTE
END
```

PROGRAMME N°7

```
PROGRAM LANGUE     ! Saluer en fonction de la langue du pays
CHARACTER (LEN=30) :: PAYS
READ*, PAYS
SELECT CASE ( PAYS )
CASE ('FRANCE','QUEBEC','SUISSE','BELGIQUE')
PRINT* , 'BONJOUR'
CASE ('ROYAUME-UNI','USA')
PRINT*, 'HELLO'
CASE ('ALGÉRIE','MAROC','KAS','EGYPTE')
PRINT*, 'SALEM'
CASE DEFAULT
PRINT*,'LANGUE PAS DISPONIBLE'
END SELECT
END
```

PROGRAMME N°8

```
PROGRAM  JOURS_SEMAINE
DIMENSION  JOUR(10)
CHARACTER*10  JOUR
PRINT*,'DONNER  LE  NOMBRE  DE  JOURS  DE  LA  SEMAINE'
READ*,  N
DO  I=1,N
PRINT*,  'DONNER  LE',I,'EME  JOUR  DE  LA  SEMAINE'
READ*,  JOUR(I)
IF(JOUR(I)=='SAMEDI')  THEN
1  PRINT*,'****  PAS  DE  COURS  ****'
ELSE
IF(JOUR(I)=='DIMANCHE')  THEN
PRINT*,'****  COURS  PG  INFORMATIQUE  LE  MATIN  ****'
ELSE
IF(JOUR(I)=='MARDI')  THEN
PRINT*,'****  TP  PG  INFORMATIQUE  LE  SOIR  ****'
ELSE
IF(JOUR(I)=='VENDREDI')GOTO  1
ENDIF
ENDIF
ENDIF
ENDDO
END
```

PROGRAMME N°9

```
PROGRAM DISTANCE_POINTS   ! Calcul  de  la  distance  entre  2  points  a  et  b
PRINT*,'DONNER  LES  COORDONNÉES  DU  POINT  A'
PRINT*
PRINT*,'XA=';READ*,XA
PRINT*,'YA=';READ*,YA
PRINT*,'DONNER  LES  COORDONNÉES  DU  POINT  B'
PRINT*
PRINT*,'XB=';READ*,XB
PRINT*,'YB=';READ*,YB
PRINT*
! CALCUL  DE  LA  DISTANCE  ENTRE  A  ET  B
D=SQRT((XB-XA)**2+(YB-YA)**2)
PRINT*
PRINT*,'DISTANCE  ENTRE  LES  POINTS  A  ET  B  EST:',D
PRINT*
STOP
END
```

PROGRAMME N°10

```
PROGRAM  HYPOTENUSE  ! Calcul  l'hypotenuse  d'un  triangle  à  angle  droit
PRINT  *,  'DONNER  LES  COTES  X  ET  Y'
READ  *,  X,Y
PRINT  *,  HYPOT  (X,Y)
END
REAL  FUNCTION  HYPOT(X,Y)
REAL  X,Y
```

```
IF(X .LE. 0.0 .OR. Y .LE. 0.0) THEN
PRINT *,'ATTENTION : VALEURS IMPOSSIBLES'
HYPOT = 0.0
RETURN
END IF
HYPOT = SQRT(X**2 + Y**2)
END
```

PROGRAMME N°11

```
PROGRAM TRIANGLE  ! Calcul de aire d'un triangle : utilisation de l'instruction FONCTION
! les formules pour l'aire d'un triangle de côtés a, b, et c sont :
!s = (a + b + c)/2   et aire = sqrt(s.(s-a).(s-b).(s-c))
PRINT *,'Entrer les longueurs des 3 cotes :'
READ *,  COTEA, COTEB, COTEC
PRINT *
PRINT *,'Aire est : ', AIRE3 (COTEA,COTEB,COTEC)
END
FUNCTION AIRE3(A, B, C)
! Calcul aire du triangle de côtés a, b, et c
S = (A + B + C) / 2.0
AIRE3 = SQRT(S * (S-A) * (S-B) * (S-C))
END
```

PROGRAMME N°12

```
PROGRAM SURFACE_PERIMETRE_CERCLE_RECTANGLE
! Calcul de la surface est du périmètre d'un cercle et d'un réctangle
REAL, PARAMETER :: PI=3.1415926
REAL DIAMETRE,RAYON
REAL LONG, LARG
PRINT*, 'DONNER LE DIAMÈTRE DU CERCLE EN MM'
READ*, DIAMETRE
RAYON=DIAMETRE/2
SC=PI*RAYON**2
PC=PI*DIAMETRE
PRINT*
PRINT*, 'LA SURFACE DU CERCLE EN MM² EST : ', SC
PRINT*
PRINT*, 'LE PÉRIMÈTRE DU CERCLE EN MM EST : ', PC
PRINT*
PRINT*, 'DONNER LA LONGUEUR DU RECTANGLE EN MM'
READ*, LONG
PRINT*, 'DONNER LA LARGEUR DU RECTANGLE EN MM'
READ*, LARG
! CALCUL DE LA SURFACE D'UN RECTANGLE
SR=LONG*LARG
! CALCUL DU PÉRIMÈTRE D'UN RÉCTANGLE
PR=(LONG+LARG)*2
PRINT*
PRINT*, 'LA SURFACE DU RÉCTANGLE EN MM² EST : ', SR
PRINT*
PRINT*, 'LE PÉRIMÈTRE DU RECTANGLE EN MM EST : ', PR
STOP
END
```

PROGRAMME N°13

```
PROGRAM  CARRE_MAGIQUE
IMPLICIT  NONE
INTEGER  N,A(100,100)
CALL  CHOIXORDRE(N)
CALL  INITIALISATION(N,A)
CALL  CALCUL(N,A)
CALL  AFFICHAGE(N,A)
STOP
END

! SOUS  PROGRAMME  "CHOIX  ET  VERIFICATION  DE  N"
SUBROUTINE  CHOIXORDRE(N)
IMPLICIT  NONE
INTEGER  N
WRITE(6,*)'!!!  CARRE  MAGIQUE  !!!'
WRITE(6,*)'  '
1  WRITE(6,*)'CHOISIR  UN  ENTIER  IMPAIR  COMPRIS  ENTRE  1  ET  99  :'
READ(5,*)N
IF(N.LT.1)  THEN
GOTO  1
ELSE  IF(N.GT.99)  THEN
GOTO  1
ENDIF
RETURN
END

! SOUS  PROGRAME  "INITIALISATION"
SUBROUTINE  INITIALISATION(N,A)
IMPLICIT  NONE
INTEGER  I,J,N,A(N,N)
DO  I=1,N
DO  J=1,N
A(I,J)=0
END  DO
END  DO
I=(N+3)/2
J=(N+1)/2
A(I,J)=1
RETURN
END

! SOUS  PROGRAMME  "CALCUL  DU  CARRE  MAGIQUE"
SUBROUTINE  CALCUL(N,A)
IMPLICIT  NONE
INTEGER  I,J,K,L,IT,N,A(N,N)
I=(N+3)/2
J=(N+1)/2
DO  IT=2,N**2
K=I+1
IF(K.GT.N)  THEN
K=1
ENDIF
I=K
L=J+1
IF(L.GT.N)  THEN
```

```
L=1
ENDIF
J=L
2  IF(A(I,J).EQ.0)  THEN
A(I,J)=IT
ELSE
I=I+1
IF(I.GT.N)  THEN
I=1
ENDIF
J=J-1
IF(J.EQ.0)  THEN
J=N
ENDIF
GOTO  2
ENDIF
END  DO
RETURN
END

! SOUS  PROGRAME  "AFFICHAGE  DU  CARRE  MAGIQUE"
SUBROUTINE  AFFICHAGE(N,A)
IMPLICIT  NONE
INTEGER  N,I,J,A(N,N)
DO  I=1,N
WRITE(6,*)(A(I,J),J=1,N)
END  DO
RETURN
END
```

PROGRAMME N°14

```
PROGRAM  INCREMENTATION
DIMENSION  DELF(15)
DEL0=.005
PASD=.005
DO  Z=1,15
DELF(Z)=DEL0+PASD
DEL0=DELF(Z)
PRINT  *,  'DELF(',Z,')',DELF(Z)
ENDDO
END
```

PROGRAMME N°15

```
PROGRAM  CONVDEGENRAD
INTEGER  DEGRE
CONFAC  =  3.141593/180.0  !  FACTEUR  DE  CONVERSION  DES  DEGRES  EN  RADIANS
PRINT  *,'              ','DEGRES','       ','                    ','RADIAN'
PRINT  *
DO  DEGRE=0,  360,  10
RADIAN  =  DEGRE*CONFAC
PRINT  *,DEGRE,'                    ',RADIAN
ENDDO
END
```

PROGRAMME N°16

```
PROGRAM ANGLES
IMPLICIT NONE
REAL RADIAN,CONFAC
INTEGER DEGRE,PAGE,OUT,N
DATA OUT/1/
DATA PAGE/0/
OPEN (1,FILE='PAGES.DAT')
CONFAC = 3.141593/180.0
! FACTEUR DE CONVERSION DES DEGRES EN RADIANS
DO 10, DEGRE = 1, 360
N = DEGRE-1
IF (N/40*40 .EQ. N) THEN
! IMPRIMER EN-TETE DE LA PAGE ET NUMERO DE LA PAGE
PAGE = PAGE+1
WRITE(OUT,100) PAGE
ELSE IF (N/10*10 .EQ.N) THEN
WRITE(OUT,110)
END IF
RADIAN = DEGRE*CONFAC
WRITE(OUT,120)DEGRE,RADIAN
10 CONTINUE
100 FORMAT (1H1//1X,'TABLE DE CONVERSION DES DEGRES EN
RADIANS',30x,'PAGE',I2//1X,'DEGRES RADIANS'/)
110 FORMAT (1X)
120 FORMAT(1X,I5,T10,F7.5)
END
```

PROGRAMME N°17

```
PROGRAM TOTALSECS ! TSECS converti heures, minutes, secondes en un total de secondes
INTEGER HEURES
PRINT *,'DONNER LE NOMBRE HEURES, MINUTES ET SECONDES'
PRINT*
read *, HEURES, MINS, SECS
PRINT *, 'LE TOTAL EN SECONDES EST :', TSECS(HEURES, MINS, SECS)
END
FUNCTION TSECS(HEURES, MINS, SECS)
INTEGER HEURES
TSECS = ((HEURES * 60) + MINS) * 60 + SECS
END
```

PROGRAMME N°18

```
PROGRAM HEURESMINSEC ! Le programme HMS convertis le TEMPS en secondes, en heures,
minutes et secondes
PRINT *,'DONNER LE TEMPS EN SECONDES'
READ *, TEMPS
CALL HMS(TEMPS, NHRS, MINS, SECS)
PRINT *, NHRS, MINS, SECS
END
SUBROUTINE HMS(TEMPS, NHEURES, MINS, SECS)
NHEURES = INT(TEMPS / 3600.0)
SECS = TEMPS - 3600.0 * NHEURES
MINS = INT(SECS / 60.0)
SECS = SECS - 60*MINS
END
```

PROGRAMME N°19

```
PROGRAM FAHRENCELSIUS ! Conversion de la température des degrés Fahrenheit en degrés Celsius
PRINT*,'ENTRER LA TEMPERATURE EN DEGRE FAHRENHEIT'
READ*, FAREN
CELSIUS=(FAREN-32)/1.8
PRINT*,'LA TEMPERATUE EN DEGRE CELSIUS EST :',CELSIUS
STOP
END
```

PROGRAMME N°20

```
PROGRAM    CONVERSION_DES_DURETES
PRINT*,'DONNER LE TYPE DE CONVERSION 1, 2, 3, OU 4'
PRINT*
PRINT*,'1:HV30-->HB','    ','2:HB-->HV30','    ','3:HV30-->HRC','    ','4:HRC-->HV30'
READ *, N
IF(N>4) GOTO 6
IF (N==1) GOTO 1   !CONVERSION DE LA DURETÉ VICKERS EN BRINELL
IF (N==2) GOTO 2   !CONVERSION DE LA DURETÉ BRINELL VICKERS
IF (N==3) GOTO 3   !CONVERSION DE LA DURETÉ VICKERS EN ROCKWELL
IF (N==4) GOTO 4   !CONVERSION DE LA DURETÉ ROCKWELL EN VICKERS
1 CALL HVHB
GOTO 5
2 CALL HBHV
GOTO 5
3 CALL HVHRC
GOTO 5
4 CALL HRCHV
PRINT*
GOTO 5
6 PRINT*,'PAS DE PROGRAMME DE CONVERSION'
PRINT*
5 END
SUBROUTINE HVHB
PRINT *, 'DONNER LA VALEUR DE HV30';READ *,HV30
IF(HV30<80) THEN
PRINT*,'*************************'
PRINT*,'* CONVERSION IMPOSSIBLE *'
PRINT*,'*************************'
ELSE IF(HV30>470) THEN
PRINT*,'*************************'
PRINT*,'* CONVERSION IMPOSSIBLE *'
PRINT*,'*************************'
ELSE
HB=0.953*HV30
PRINT*,'---------------------------------------'
PRINT*, HV30,'HV30','  ','=','  ',HB,'HB'
PRINT*,'---------------------------------------'
ENDIF
RETURN
END
```

```
SUBROUTINE  HBHV
PRINT  *,'DONNER  LA  VALEUR  DE  HB';READ  *,HB
IF(HB<76)  THEN
PRINT*,'*************************'
PRINT*,'*  CONVERSION  IMPOSSIBLE  *'
PRINT*,'*************************'
ELSE  IF(HB>447)  THEN
PRINT*,'*************************'
PRINT*,'*  CONVERSION  IMPOSSIBLE  *'
PRINT*,'*************************'
ELSE
HV30=1.04932*HB
PRINT*,'---------------------------------------'
PRINT*,     HB,'HB','  ','=','  ',HV30,'HV30'
PRINT*,'---------------------------------------'
ENDIF
RETURN
END
SUBROUTINE  HVHRC
PRINT  *,  'DONNER  LA  VALEUR  DE  HV30';READ  *,HV30
IF(HV30<300)  THEN
PRINT*,'*************************'
PRINT*,'*  CONVERSION  IMPOSSIBLE  *'
PRINT*,'*************************'
ELSE  IF(HV30>880)  THEN
PRINT*,'*************************'
PRINT*,'*  CONVERSION  IMPOSSIBLE  *'
PRINT*,'*************************'
ELSE
HRC=0.0915*HV30
PRINT*,'---------------------------------------'
PRINT*,     HV30,'HV30','  ','=','  ',HRC,'HRC'
PRINT*,'---------------------------------------'
ENDIF
RETURN
END
SUBROUTINE  HRCHV
PRINT  *,'DONNER  LA  VALEUR  DE  HRC';READ  *,HRC
IF(HRC<29)  THEN
PRINT*,'*************************'
PRINT*,'*  CONVERSION  IMPOSSIBLE  *'
PRINT*,'*************************'
ELSE  IF(HRC>67)  THEN
PRINT*,'*************************'
PRINT*,'*  CONVERSION  IMPOSSIBLE  *'
PRINT*,'*************************'
ELSE
HV30=10.929*HRC
PRINT*,'---------------------------------------'
PRINT*,  HRC,'HRC','  ','=','  ',HV30,'HV30'
PRINT*,'---------------------------------------'
ENDIF
RETURN
END
```

PROGRAMME N°21

```
PROGRAM PGCD_PPCM          ! Calcul du pgcd et du ppcm de 2 entiers m et n
INTEGER M, N, I, J, R
PRINT *, 'SAISISSEZ DEUX ENTIERS M ET N :'
READ *, M, N
I = MAX(M,N)
J = MIN(M,N)
10        R = MOD(I,J)
IF (R.EQ.0) GOTO 20
I = J
J = R
GOTO 10
20        PRINT *, 'PGCD =', J, '    PPCM =', M*N/J
END
```

PROGRAMME N°22

```
PROGRAM CONDITIONS_COUPE
! DÉCLARATIONS
CHARACTER*15 MATERIAU
CHARACTER*15 OUTIL
CHARACTER*15 OPERATION
REAL N
PARAMETER (PI=3.14159)
! PARAMÈTRES D'ENTRÉE
PRINT*,'DONNER LE DIAMETRE DE LA PIECE OU DE LA FRAISE [MM]'
READ*, D
PRINT*,'DONNER LE MATERIAU DE LA PIECE A USINER'
READ*,MATERIAU
PRINT*,'DONNER OUTIL UTILISE'
READ*,OUTIL
PRINT*,'DONNER OPERATION USINAGE'
READ*, OPERATION
! APPEL DES SOUS PROGRAMMES
IF(MATERIAU=='ACIER'.AND.OUTIL=='ARS') GOTO 1
IF(MATERIAU=='ALUMINIUM'.AND.OUTIL=='ARS') GOTO 2
IF(MATERIAU=='ACIER'.AND.OUTIL=='CARBURE') GOTO 3
IF(MATERIAU=='ALUMINIUM'.AND.OUTIL=='CARBURE') GOTO 4
1 CALL ACIER_ARS(OPERATION)
GOTO 5
2 CALL ALU_ARS(OPERATION)
GOTO 5
3 CALL ACIER_CARBURE(OPERATION)
GOTO 5
4 CALL ALU_CARBURE(OPERATION)
! LECTURE DE VC ET CALCUL DE N
5 PRINT*,'DONNER LA VALEUR DE LA VITESSE DE COUPE VC EN M/MIN'
READ*, VC
N=(1000*VC)/(PI*D)
PRINT*,'LA FRÉQUENCE DE ROTATION DE LA BROCHE N',N,'[TR/MIN]'
END
! LES SOUS PROGRAMMES
SUBROUTINE ACIER_ARS(OPERATION)
CHARACTER*15 OPERATION
IF (OPERATION=='EBAUCHE') THEN
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 10<VC<30 M/MIN'
```

```
ELSE
IF(OPERATION=='FINITION')THEN
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 20<VC<50 M/MIN'
ENDIF
ENDIF
RETURN
END
SUBROUTINE ALU_ARS(OPERATION)
CHARACTER*15 OPERATION
IF(OPERATION=='EBAUCHE') THEN
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 25<VC<45 M/MIN'
ELSE
IF(OPERATION=='FINITION') THEN
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 30<VC<60 M/MIN'
ENDIF
ENDIF
RETURN
END
SUBROUTINE ACIER_CARBURE(OPERATION)
CHARACTER*15 OPERATION
IF(OPERATION=='EBAUCHE') THEN
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 60<VC<150 M/MIN'
ELSE
IF(OPERATION=='FINITION') THEN
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 120<VC<220 M/MIN'
ENDIF
ENDIF
RETURN
END
SUBROUTINE ALU_CARBURE(OPERATION)
CHARACTER*15 OPERATION
IF(OPERATION=='EBAUCHE') THEN
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 100<VC<180 M/MIN'
ELSE
IF(OPERATION=='FINITION') THEN
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 140<VC<260 M/MIN'
ENDIF
ENDIF
RETURN
END
```

PROGRAMME N°23

```
PROGRAM COEFFICIENT_PRESSION_CP
! Calcul de la valeur du coefficient de pression autour d'une aile selon quatre paramètres
! théta : l'angle d'attaque du profil d'aile
! Minf : le nombre de Mach à l'infini
! Tinf : la température en Kelvin à l'infini
! v : la vitesse de l'air autour de l'aile
REAL, PARAMETER :: GAMMA=1.4 , R=256
REAL MINFINI
PRINT*,'ENTRER ANGLE ATTAQUE DU PROFIL AILE'; READ*,THETA
PRINT*
PRINT*,'ENTRER LA VALEUR DE LA TEMPÉRATURE EN KELVIN'; READ*, TINFINI
PRINT*,'ENTRER LA VALEUR DE LA VITESSE DE AIR AUTOUR DE AILE'; READ*,V
```

```
! CALCUL DU NOMBRE DE MACH À L'INFINI
DENOM=SQRT(GAMMA*R*TINFINI)
MINFINI=V/DENOM
PRINT*
PRINT*,'LE NOMBRE MACH MINFINI =',MINFINI
IF (MINFINI>1) THEN
DENOM1=SQRT((MINFINI)**2-1)
CP=(2*THETA)/DENOM1
PRINT*
PRINT*,'LE COEFFICIENT DE PRESSION EST CP =',CP
ELSE
PRINT*
PRINT*,'LE COEFFICIENT DE PRESSION CP=0'
ENDIF
PRINT*
END
```

PROGRAMME N°24

```
PROGRAM CALCUL_FORMULE          ! Calcul de somme ((ai*ai)+(bi*bi))/ai*bi
DIMENSION A(10), B(10)
SUM=0
PRINT*, 'DONNER LES DIMENSIONS DES TABLEAUX'; READ*, N
PRINT*, 'DONNER LES ELEMENTS DU TABLEAU A'
DO   I=1,N
PRINT*, 'A(',I,')='; READ*, A(I)
ENDDO
PRINT*, 'DONNER LES ELEMENTS DU TABLEAU B'
DO   I=1,N
PRINT*, 'B(',I,')=' ; READ*, B(I)
ENDDO
DO   I=1,N
S=A(I)**2 + B(I)**2
P=A(I)*B(I)
SUM=SUM+(S/P)
ENDDO
PRINT*, 'LA VALEUR DE LA SOMME EST : ', SUM
END
```

PROGRAMME N°25

```
PROGRAM FORMULE_1
PRINT*, 'DONNER LA VALEUR DE X'; READ*, X
IF(X<=0)THEN
Y=1/(1+X)
PRINT*, 'LA VALEUR DE Y EST ', Y
ELSE
IF(X>0 .AND. X<=5)THEN
Y=-(X**2+1)/100
PRINT*, 'LA VALEUR DE Y EST ',Y
ELSE
IF(X>5 .AND. X<=30)THEN
Y=15/(4*X)
PRINT*, 'LA VALEUR DE Y EST ',Y
ELSE
```

```
Y=1./8.
PRINT*, 'LA VALEUR DE Y EST ',Y
ENDIF
END IF
ENDIF
END
```

PROGRAMME N°26

```
PROGRAM FORMULE_CALCULE_2
4 PRINT*, 'DONNER LA VALEUR DE X'; READ*, X
IF (X<-1) THEN
PRINT*,'VALEUR MINIMALE EST -1'; GOTO 4
PRINT*
ELSE
ENDIF
IF(X<0)THEN
DO A=X,10,0.1
Y=2*A**2+3
PRINT 1, A,Y
1 FORMAT (5X,'POUR X=',F8.2,6X,'LA VALEUR DE Y =',F8.2)
ENDDO
ELSE
IF(X>=0 .AND. X<=2)THEN
DO A=X,10,0.1
Y=3-4*A
PRINT 2, A,Y
2 FORMAT (5X,'POUR X=',F8.2,6X,'LA VALEUR DE Y =',F8.2)
ENDDO
ELSE
DO A=X,10,0.1
Y=3-2*A**2
PRINT 3, A,Y
3 FORMAT (5X,'POUR X=',F8.2,6X,'LA VALEUR DE Y =',F8.2)
ENDDO
END IF
ENDIF
STOP
END
```

PROGRAMME N°27

```
PROGRAM TINF_TSUP
! Calcul du nombre de températures < à 20°C &
! du nombre de températures > à 40°C, ainsi que leurs moyennes
DIMENSION T(50)
PRINT*, 'DONNER LA DIMENSION DU TABLEAU DES TEMPÉRATURES'; READ*, N
PRINT*, 'DONNER LES TEMPÉRATURES EN DEGRE CELCIUS'
SUM = 0.0
SUM1 = 0.0
DO I=1,N
PRINT*, 'T(',I,')='
READ*, T(I)
IF(T(I)<20) THEN
NTINF=NTINF+1
```

```
SUM = SUM + T(I)
ELSE
IF(T(I)>40) THEN
NTSUP=NTSUP+1
SUM1 = SUM1 + T(I)
ENDIF
ENDIF
ENDDO
PRINT*
PRINT*, 'LE NOMBRE DE TEMPÉRATURES INFERIEURS 20 DEG  --->',NTINF
PRINT*
VMTINF= SUM/NTINF
PRINT *,'LA MOYENNE DES TEMPERATURES INFERIEURS 20 DEG =', VMTINF
PRINT*
PRINT*, 'LE NOMBRE DE TEMPÉRATURES SUPERIEURS 40 DEG  --->',NTSUP
PRINT*
VMTSUP = SUM1 / NTSUP
PRINT *,'VALEUR MOYENNE DES TEMPERATURES SUPERIEURS 40 DEG =', VMTSUP
PRINT*
END
```

PROGRAMME N°28

```
PROGRAM SOMME_ENTIERS_IMPAIRS ! Calcul de s = 1 + 3 + 5 + 7 + 9 + 11 + 13 + 15
+ 17 + 19
INTEGER :: I,S
S=0.
DO I=1,19,2
S=S+I
END DO
PRINT*,'LA SOMME DES ENTIERS IMPAIRS =',S
END
```

PROGRAMME N°29

```
PROGRAM SOMME_ENTIERS_PAIRS  ! Calcul de la somme des entiers pairs
PRINT*,'DONNER LE NOMBRE DES ENTIERS N'
READ*, N
S=0
DO I=0,N,2
S=S+I
END DO
PRINT*,' LA SOMME DES ENTIERS EST : ', S
END
```

PROGRAMME N°30

```
PROGRAM PRODUIT_ENTIERS  ! Calcul du produit de N entiers
PRINT*,'DONNER LE NOMBRE DES ENTIERS N'
READ*, N
P=1
DO I=1, N
P=P*I
END DO
PRINT*,' LE PRODUIT DES N ENTIERS EST : ', P
END
```

PROGRAMME N°31

```
PROGRAM PRODUIT_ENTIERS_IMPAIRS  ! Calcul du produit de N entiers impairs
PRINT*,'DONNER LE NOMBRE DES ENTIERS N'
READ*, N
P=1
DO I=1,N,2
P=P*I
END DO
PRINT*,' LE PRODUIT DES N ENTIERS EST : ', P
END
```

PROGRAMME N°32

```
PROGRAM OPERATIONS
REAL A, B
CHARACTER*2 OPERANDE
PRINT*,'DONNER LE PREMIER REEL A'
READ*, A
PRINT*,'DONNER LE DEUXIEME REEL B'
READ*, B
PRINT*,'DONNER L"OPERANDE +, -, X, :'
5 READ*, OPERANDE
IF(OPERANDE=='+') GOTO 1
IF(OPERANDE=='-') GOTO 2
IF(OPERANDE=='X') GOTO 3
IF(OPERANDE==':') GOTO 4
1 PRINT*,'L"OPERATION EST:A+B';S=A+B;PRINT*,'LA SOMME A+B = ', S
PRINT*,'CHOISISSEZ UN AUTRE OPERANDE'
GOTO 5
2 PRINT*,'L"OPERATION EST:A-B';D=A-B;PRINT*,'LA DIFFÉRENCE A-B = ', D
PRINT*,'CHOISISSEZ UN AUTRE OPERANDE'
GOTO 5
3 PRINT*,'L"OPERATION EST:A*B';P=A*B;PRINT*,'LE PRODUIT A*B = ', P
PRINT*,'CHOISISSEZ UN AUTRE OPERANDE'
GOTO 5
4 IF(B==0) THEN
PRINT*,'DENOMINATEUR NUL, QUOTIENT IMPOSSIBLE'
ELSE
PRINT*,'L"OPERATION EST:A/B';Q=A/B;PRINT*,'LE QUOTIENT A/B = ', Q
ENDIF
END
```

PROGRAMME N°33

```
PROGRAM SOMME_ENTIERS_CARRES  ! Calcul de la somme des entiers élevés au carré
S=1^2+2^2+3^2+4^2+ ..... +20^2
S=0
PRINT*, 'DONNER LA VALEUR FINALE N'
READ*, N
!CALCUL DE LA SOMME DES CARRÉS
DO I=1,N
S=S+(I)**2
END DO
PRINT*, 'LA VALEUR DE LA SOMME S EST : ', S
END
```

PROGRAMME N°34

```
PROGRAM SOMME_ENTIERS  ! Calcul de la somme S=1+1/2+1/3+ ..... +1/n
REAL I
S=0.
PRINT*, 'DONNER LA VALEUR FINALE N'
READ*, N
! CALCUL DE LA SOMME 1/N
DO I=1,N
S=S+1/I
END DO
PRINT*, 'LA VALEUR DE LA SOMME S EST : ', S
STOP
END
```

PROGRAMME N°35

```
PROGRAM SOMME_ENTIERS ! Calcul de la somme S=1/2+1/4+1/6+ ..... +1/n
REAL I
S=0.
PRINT*, 'DONNER LA VALEUR FINALE N'
READ*, N
!CALCUL DE LA SOMME 1/N
DO I=2,N,2
S=S+1/I
END DO
PRINT*, 'LA VALEUR DE LA SOMME S EST : ', S
STOP
END
```

PROGRAMME N°36

```
PROGRAM FACTOR     ! Calcul iteratif de la factorielle d'un entier naturel n
IMPLICIT NONE
INTEGER I, N, FACT
 10     PRINT*, 'ENTREZ UN ENTIER NATUREL :'
READ*, N
IF (N.LT.0) THEN
PRINT*, 'VALEUR INCORRECTE'
GOTO 10
ENDIF
FACT = 1
DO I=1, N
FACT = FACT * I
ENDDO
PRINT*, N, '! = ', FACT
END
```

PROGRAMME N°37

```
PROGRAM SOMME_FACTOR ! Calcul de la somme de la factorielle d'un entier naturel n
IMPLICIT NONE
INTEGER I, N, FACT,S
 10     PRINT*, 'ENTREZ UN ENTIER NATUREL :'
READ*, N
IF (N.LT.0) THEN
PRINT*, 'VALEUR INCORRECTE'
```

```
GOTO  10
ENDIF
S=0
FACT  =  1
DO  I=1,  N
FACT  =  FACT  *  I
S=S+FACT
ENDDO
PRINT*,  'S=  ',  S
END
```

PROGRAMME N°38

```
PROGRAM  SOMME_UN_SUR_FACTOR_N    !  Calcul  de  la  somme  1/1!+1/2!+1/3!+.....+1/n!
IMPLICIT  NONE
REAL  I,FACT,S
INTEGER  N
  10      PRINT*,  'ENTREZ  UN  ENTIER  NATUREL  :'
READ*,  N
IF  (N.LT.0)  THEN
PRINT*,  'VALEUR  INCORRECTE'
GOTO  10
ENDIF
S=0
FACT  =  1
DO  I=1,  N
FACT  =  FACT  *  I
S=S+1/FACT
ENDDO
PRINT*
PRINT*,  'S=  ',  S
END
```

PROGRAMME N°39

```
PROGRAM  SOMME_N_SUR_FACTOR_N_1    !  Calcul  de  la  somme  2/1!+3/2!+4/3!+.....+n/(n-1)!
IMPLICIT  NONE
REAL  I,FACT,S
INTEGER  N
  10      PRINT*,  'ENTREZ  UN  ENTIER  NATUREL  :'
READ*,  N
IF  (N.LT.0)  THEN
PRINT*,  'VALEUR  INCORRECTE'
GOTO  10
ENDIF
S=0
FACT  =  1
DO  I=2,  N
FACT  =  FACT  *  (I-1)
S=S+I/FACT
ENDDO
PRINT*
PRINT*,  'S=  ',  S
END
```

PROGRAMME N°40

```
PROGRAM SOMME_FACTOR_2N_SUR_FACTOR_2N_1 ! Calcul de la somme 2!/4!+4!/6!+6!/8!+.....+2n!/(2n+2)!
IMPLICIT NONE
REAL I,FACT,S,FACT1
INTEGER N
 10    PRINT*, 'ENTREZ UN ENTIER NATUREL :'
READ*, N
IF (N.LT.0) THEN
PRINT*, 'VALEUR INCORRECTE'
GOTO 10
ENDIF
S=0.
DO I=1,N
FACT = 2*I+1
FACT1=2*I+2
S=S+1/(FACT*FACT1)
ENDDO
PRINT*
PRINT*, 'S= ', S
END
```

PROGRAMME N°41

```
PROGRAM SOMME_QUOTIENT_FACTOR ! Calcul de la somme (1!x2!x3!.....n!)/1!+2!+3!+.....n!
IMPLICIT NONE
REAL I,FACT,S,P,Q
INTEGER N
 10    PRINT*, 'ENTREZ UN ENTIER NATUREL :'
READ*, N
IF (N.LT.0) THEN
PRINT*, 'VALEUR INCORRECTE'
GOTO 10
ENDIF
S=0.
P=1.
FACT = 1
DO I=1,N
FACT = FACT*I
P=P*FACT
PRINT*,'P=',P
S=S+FACT
PRINT*,'S=',S
Q=P/S
ENDDO
PRINT*
PRINT*, 'Q= ', Q
END
```

PROGRAMME N°42

```
PROGRAM STATS
! Calcul de la moyenne arithmétique des valeurs x1, x2, x3,....,xN
! Utilisation de l'instruction SUBROUTINE
! La moyenne est donnée par M = sum (xi) / N
REAL X(1000)
! lire le nombre de valeurs NPTS
PRINT *,'DONNER LE NOMBBRE DE VALEURS A CALCULER LEUR MOYENNE'
READ *, NPTS
PRINT *,'DONNER CES VALEURS'
READ *, (X(I), I = 1,NPTS)
CALL MOY(X, NPTS, AVG)
PRINT *,'MOYENNE =', AVG
END
! subroutine : sous pg pour le calcul de la moyenne
SUBROUTINE MOY(X, NPTS, AVG)
REAL X(1000)
SUM = 0.0
DO 15, I = 1,NPTS
SUM = SUM + X(I)
15 CONTINUE
AVG = SUM / NPTS
END
```

PROGRAMME N°43

```
PROGRAM ECART_TYPE  ! Calcul de l'ecart type sigma=sqrt(1/n(somme(x(i)-moy)**2))
DIMENSION X(50)
REAL MOY1,MOY
MOY1=0
S=0
PRINT*, 'DONNER LA DIMENSION DU TABLEAU X -->N'
READ*, N
! LECTURE DES ÉLÉMENTS DU TABLEAU X
DO I=1,N
PRINT*,'X(',I,')='
READ*, X(I)
ENDDO
! CALCUL DE LA MOYENNE ARITHMITIQUE
DO I=1,N
MOY1=MOY1+X(I)
ENDDO
MOY=MOY1/N
PRINT*,'MOYENNE=', MOY
! CALCUL DE LA SOMME DES ÉCARTS AU CARRÉ (X(I)-MOY)**2
DO I=1,N
S=S+(X(I)-MOY)**2
ENDDO
PRINT*,'SOMME DES ECARTS AU CARRE =', S
! CALCUL DE L'ÉCART TYPE
SIGMA=SQRT(S/N)
PRINT*
PRINT*,'ECART TYPE SIGMA=', SIGMA
PRINT*
END
```

PROGRAMME N°44

```
PROGRAM FONCTIONS_TRIGONOMETRIQUES
EXTERNAL COT
INTRINSIC SIN,COS,TAN
PRINT*
PRINT*,'DONNER L"ANGLE EN DEGRE'
READ*, DEGRE ! LE POINT REMPLACE LA VIRGULE
CONFAC = 3.141593/180.0 ! FACTEUR DE CONVERSION DES DEGRES EN RADIANS
ANGLE= DEGRE*CONFAC
CALL TRIG(ANGLE,SIN,SINE)
PRINT*,'LE SINUS DE L"ANGLE EST:',SINE
PRINT*
CALL TRIG(ANGLE,COS,COSINE)
PRINT*,'LE COSINUS DE L"ANGLE EST:',COSINE
PRINT*
CALL TRIG(ANGLE,TAN,TANGT)
PRINT*,'LA TANGENTE DE L"ANGLE EST:',TANGT
PRINT*
CALL TRIG(ANGLE,COT,COTAN)
PRINT*,'LA COTANGENTE DE L"ANGLE EST:',COTAN
END
SUBROUTINE TRIG(X,F,Y) !X ANGLE EN RAD, F FONCTION TRIGONOMÉTRIQUE,
Y = F(X)       ! Y EST LA FONCTION TRIGONOMÈTRIQUE DE X
RETURN
END
FUNCTION COT(X)
COT=COTAN(X)
RETURN
END
```

PROGRAMME N°45

```
PROGRAM METHODE_RECTANGLES
IMPLICIT NONE
REAL :: S,S1,S2,X,F,FA,FB !A:BORNE_INF, B:BORNE_SUP
INTEGER :: I,NP
PRINT*, 'RENTRER UN NOMBRE POSITIF'
READ*, X
PRINT*, 'RENTRER LE NOMBRE DE POINTS'
READ*, NP
S=0.
DO I=0,NP
CALL MAFONCTION(I*X/NP,F)   ! X/NP LE PAS
IF(I==0) THEN
FA=F
ELSE
IF(I==NP) THEN
FB=F
ENDIF
ENDIF
S=S+F
ENDDO
S1=S*(X/NP)
PRINT*,' LA VALEUR DE L"INTEGRALE EST EGALE :', S1 ! AIRE SUR LA COURBE F(X)
S2=S1-(X/NP*(FA+FB))   ! AIRE SUR LA COURBE F(X)
```

```
PRINT*,'L"AIRE DE LA COURBE F(X) EST :',S2,'<AIRE<',S1
END
! SOUS PROGRAMME DE CALCUL DE LA FONCTION F(X) CONTINUE, POSITIVE ET CROISSANTE
SUBROUTINE MAFONCTION(X,F)
IMPLICIT NONE
REAL :: X,F,PI
PI=4*ATAN(1.)
F=EXP(-X**2/2)/SQRT(2*PI)
RETURN
END
```

PROGRAMME N°46

```
PROGRAM COMPARAISON_DEUX_COMPLEXES
COMPLEX Z1,Z2
LOGICAL(1) CR
PRINT*,'DONNER LE PREMIER NOMBRE COMPLEXE'
PRINT*,'Z1(RE,IM)='; READ*, Z1
PRINT*,'DONNER LE DEUXIEME NOMBRE COMPLEXE'
PRINT*,'Z2(RE,IM)='; READ*, Z2
IF(REAL(Z1)==REAL(Z2).AND.IMAG(Z1)==IMAG(Z2)) THEN
CR=.TRUE.
PRINT*,'LE RESULTAT EST :',CR
ELSE
CR=.FALSE.
PRINT*,'LE RESULTAT EST :',CR
ENDIF
END
```

PROGRAMME N°47

```
PROGRAM CONJUGUE_UN_NOMBRE_COMPLEXE
COMPLEX Z,Z1   !Z1 CONJUGUE DE Z
PRINT*,'DONNER LE NOMBRE COMPLEXE Z'
PRINT*,'Z(RE,IM)='; READ*, Z
PRINT*
PRINT*,'LE NOMBRE COMPLEXE Z EST :'
PRINT*
PRINT 1,REAL(Z),IMAG(Z)
1 FORMAT(2X,'Z=',F8.2,'+I',F8.2)
!CALCUL DU CONJUGE
Z1=CONJG(Z)
PRINT*
PRINT*,'SON CONJUGE Z1 EST :'
PRINT*
PRINT 2,REAL(Z1),IMAG(Z1)
2 FORMAT(2X,'Z1=',F8.2,'+I',F8.2)
END
```

PROGRAMME N°48

```
PROGRAM MODULE_UN_NOMBRE_COMPLEXE
COMPLEX Z
REAL MOD
PRINT*,'DONNER LE NOMBRE COMPLEXE Z'
PRINT*,'Z(RE,IM)='; READ*, Z
```

```
PRINT*,'LE NOMBRE COMPLEXE Z EST :'
PRINT*
PRINT 1,REAL (Z),IMAG(Z)
1 FORMAT (2X,'Z=',F8.2,'+I',F8.2)
! CALCUL DU MODULE DE Z
A= REAL (Z)
B=IMAG(Z)
PRINT*
MOD=SQRT(A**2+B**2)
PRINT*,'SON MODULE EST :',MOD
END
```

PROGRAMME N°49

```
PROGRAM PRODUIT_REEL_PAR_UN_NOMBRE_COMPLEXE
COMPLEX Z,Z1
REAL ALPHA
PRINT*,'DONNER LE NOMBRE RÉEL ALPHA'
READ*, ALPHA
PRINT*,'DONNER LE NOMBRE COMPLEXE Z'
PRINT*,'Z (RE,IM)='; READ*, Z
PRINT*
PRINT*,'LE NOMBRE COMPLEXE Z EST :'
PRINT*
PRINT 1,REAL (Z),IMAG(Z)
1 FORMAT (2X,'Z=',F8.2,'+I',F8.2)
! CALCUL DU PRODUIT ALPHA*REAL(Z) ET ALPHA*IMAG(Z)
A=ALPHA*REAL (Z)
B=ALPHA*IMAG (Z)
Z1=CMPLX(A,B)
PRINT*
PRINT*,'LE NOMBRE COMPLEXE Z1 EST :'
PRINT*
PRINT 2,REAL(Z1),IMAG(Z1)
2 FORMAT (2X,'Z1=',F8.2,'+I',F8.2)
END
```

PROGRAMME N°50

```
PROGRAM SOMME_DIFFERENCE_DEUX_NOMBRES_COMPLEXES
COMPLEX Z1,Z2,Z3,Z4
PRINT*,'DONNER LE NOMBRE COMPLEXE Z1'
PRINT*,'Z1(RE,IM)='; READ*, Z1
PRINT*
PRINT*,'LE NOMBRE COMPLEXE Z1 EST :'
PRINT*
PRINT 1,REAL(Z1),IMAG(Z1)
1 FORMAT (2X,'Z1=',F8.2,'+I',F8.2)
PRINT*,'DONNER LE NOMBRE COMPLEXE Z2'
PRINT*,'Z2(RE,IM)='; READ*, Z2
PRINT*
PRINT*,'LE NOMBRE COMPLEXE Z2 EST :'
PRINT*
PRINT 2,REAL(Z2),IMAG(Z2)
2 FORMAT (2X,'Z2=',F8.2,'+I',F8.2)
! SOMME DES PARTIES RÉELLES ET IMMAGINAIRES
A=REAL (Z1)+REAL(Z2)
```

```
B=IMAG(Z1)+IMAG(Z2)
Z3=CMPLX(A,B)
PRINT*
PRINT*,'LA SOMME DES DEUX COMPLEXES Z3=Z1+Z2 EST :'
PRINT*
PRINT 3,REAL (Z3),IMAG(Z3)
3 FORMAT (2X,'Z3=',F8.2,'+I',F8.2)
! DIFFERENCE DES PARTIES RÉELLES ET IMMAGINAIRES
C=REAL(Z1)-REAL(Z2)
D=IMAG(Z1)-IMAG(Z2)
Z4=CMPLX(C,D)
PRINT*
PRINT*,'LA DIFFERENCE DES DEUX COMPLEXES Z4=Z1-Z2 EST :'
PRINT*
PRINT 4,REAL(Z4),IMAG(Z4)
4 FORMAT (2X,'Z4=',F8.2,'+I',F8.2)
END
```

PROGRAMME N°51

```
PROGRAM PRODUIT_DEUX_NOMBRES_COMPLEXES
COMPLEX Z1,Z2,Z3
PRINT*,'DONNER LE NOMBRE COMPLEXE Z1'
PRINT*,'Z1(RE,IM)='; READ*, Z1
PRINT*
PRINT*,'LE NOMBRE COMPLEXE Z1 EST :'
PRINT*
PRINT 1,REAL (Z1),IMAG(Z1)
1 FORMAT (2X,'Z1=',F8.2,'+I',F8.2)
PRINT*,'DONNER LE NOMBRE COMPLEXE Z2'
PRINT*,'Z2(RE,IM)='; READ*, Z2
PRINT*
PRINT*,'LE NOMBRE COMPLEXE Z2 EST :'
PRINT*
PRINT 2,REAL (Z2),IMAG(Z2)
2 FORMAT (2X,'Z2=',F8.2,'+I',F8.2)
! PRODUIT DES PARTIES RÉELLES ET IMMAGINAIRES
A=(REAL(Z1)*REAL(Z2))-(IMAG(Z1)*IMAG(Z2))
B=(REAL(Z1)*IMAG(Z2))+(IMAG(Z1)*REAL(Z2))
Z3=CMPLX(A,B)
PRINT*
PRINT*,'LE PRODUIT DES DEUX COMPLEXES Z3=Z1*Z2 EST :'
PRINT*
PRINT 3,REAL(Z3),IMAG(Z3)
3 FORMAT (2X,'Z3=',F8.2,'+I',F8.2)
END
```

PROGRAMME N°52

```
PROGRAM QUOTIENT_DEUX_NOMBRES_COMPLEXES
COMPLEX Z1,Z2,Z3 !Z4
REAL DENO
PRINT*,'DONNER LE NOMBRE COMPLEXE Z1'
PRINT*,'Z1(RE,IM)='; READ*, Z1
PRINT*
```

```
PRINT*,'LE NOMBRE COMPLEXE Z1 EST :'
PRINT*
PRINT 1,REAL(Z1),IMAG(Z1)
1 FORMAT (2X,'Z1=',F8.2,'+I',F8.2)
PRINT*,'DONNER LE NOMBRE COMPLEXE Z2'
PRINT*,'Z2(RE,IM)='; READ*, Z2
PRINT*
PRINT*,'LE NOMBRE COMPLEXE Z2 EST :'
PRINT*
PRINT 2,REAL (Z2),IMAG(Z2)
2 FORMAT (2X,'Z2=',F8.2,'+I',F8.2)
!CALCUL DU DÉNOMENATEUR
DENO=REAL(Z2)**2+IMAG(Z2)**2
! CALCUL DU DÉNOMINATEUR AUTREMENT
! Z4=CONJG(Z2)
! DENO=Z2*Z4
!PRODUIT DES PARTIES RÉELLES ET IMMAGINAIRES DE Z3
A=(REAL(Z1)*REAL(Z2))+(IMAG(Z1)*IMAG(Z2))
B=(IMAG(Z1)*REAL(Z2))-(REAL(Z1)*IMAG(Z2))
Z3=CMPLX(A,B)
PRINT*
PRINT*,'LE QUOTIENT DES DEUX COMPLEXES Z3=Z1/Z2 EST :'
PRINT*
PRINT 3,REAL(Z3),DENO,IMAG(Z3),DENO
3 FORMAT (2X,'Z3=',F8.2,'/',F6.2,'+I',F8.2,'/',F6.2)
END
```

PROGRAMME N°53

```
PROGRAM DIVCOMPLEXE ! Division de deux complexes cnum et cden
IMPLICIT NONE
COMPLEX :: CNUM, CDEN, CRES
REAL :: X, Y, R, THETA
!  SAISIE DES DEUX COMPLEXES
PRINT*, 'PARTIE REELLE ET IMAGINAIRE DU COMPLEXE NUMERATEUR :'
READ*, X, Y
CNUM = CMPLX(X, Y)
PRINT*, 'PARTIE REELLE ET IMAGINAIRE DU COMPLEXE DENOMINATEUR :'
READ*, X, Y
CDEN = CMPLX(X, Y)
IF (ABS(CDEN) == 0.) THEN
PRINT*, 'DIVISION IMPOSSIBLE'
ELSE
CRES = CNUM / CDEN
 X=REAL(CRES); Y=AIMAG(CRES);     ! PARTIE REELLE ET IMAGINAIRE DE CRES
R=ABS(CRES)     ! CALCUL DU MODULE
THETA = ATAN2(Y, X)     ! CALCUL DE L'ARGUMENT
PRINT*, 'AFFICHAGE DU COMPLEXE SOUS LA FORME A+IB :'
WRITE(6, 1000) X, Y
PRINT*, 'AFFICHAGE DU COMPLEXE SOUS FORME EXPONENTIELLE :'
WRITE (6, 1001) R, THETA
END IF
1000 FORMAT (F10.6, ' + I *', F10.6)
1001 FORMAT (F10.6, ' * EXP(I *', F10.6, ')')
END
```

PROGRAMME N°54

```
PROGRAM SOMME_POLYNOMES_NON_NULS
DIMENSION A(50),B(50),C(50)
PRINT*, 'DONNER LE NOMBRE DE TERMES DU POLYNOME A(X)' ! N+1 TERMES
READ*, NP
PRINT*, 'DONNER LE NOMBRE DE TERMES DU POLYNOME B(X)'  ! M+1 TERMES
READ*, MP
PRINT*,'DONNER LES COEFFICIENTS DU POLYNOME A(X)'
DO J=1, NP
PRINT*, 'A(',J,')=' ; READ*,A(J)
ENDDO
PRINT*,'DONNER LES COEFFICIENTS DU POLYNOME B(X)'
DO J=1, MP
PRINT*,'B(',J,')='; READ*,B(J)
ENDDO
I=NP-MP
PRINT*
PRINT*,'LES COEFFICIENTS DU POLYNOME SOMME C(X) SONT :'
PRINT*
DO K=1,NP
IF(K<=I) THEN
C(K)=A(K)
PRINT*,'C(',K,')=',C(K)
ELSE
C(K)=A(K)+B(K-I)
PRINT*,'C(',K,')=',C(K)
ENDIF
ENDDO
STOP
END
```

PROGRAMME N°55

```
PROGRAM PRODUIT_POLYNOMES_NON_NULS
DIMENSION A(50),B(50),C(50)
PRINT*, 'DONNER LE NOMBRE DE TERMES DU POLYNOME A(X)' !N+1 TERMES
READ*, NP
PRINT*, 'DONNER LE NOMBRE DE TERMES DU POLYNOME B(X)'  !M+1 TERMES
READ*, MP
PRINT*,'DONNER LES COEFFICIENTS DU POLYNOME A(X)'
DO J=1,NP
PRINT*, 'A(',J,')=' ; READ*,A(J)
ENDDO
PRINT*,'DONNER LES COEFFICIENTS DU POLYNOME B(X)'
DO J=1,MP
PRINT*,'B(',J,')='; READ*,B(J)
ENDDO
LP=NP+MP-1 !NOMBRE DE TERMES DU POLYNOME C(X) !L+1 TERMES
PRINT*
PRINT*,'LES COEFFICIENTS DU POLYNOME PRODUIT C(X) SONT :'
PRINT*
DO K=1,LP
IF (K<=NP) THEN
J=K
ELSE
J=NP
```

```
ENDIF
IF (K<=MP) THEN
I0=1
ELSE
I0=K-MP+1
ENDIF
C(K)=0.0
DO I=I0,J
C(K)=C(K)+A(I)*B(K-I+1)
ENDDO
PRINT*,'C(',K,')=',C(K)
ENDDO
END
```

PROGRAMME N°56

```
PROGRAM LECTURE_ELEMENTS_VECTEUR  ! Lecture des éléments d'un tableau de dimension N
--> vecteur
REAL, DIMENSION(50) :: T
PRINT*, 'DONNER LA DIMENSION DU TABLEAU'; READ*, N
PRINT*, 'DONNER LES ELEMENTS DU TABLEAU'
DO 1 I=1,N
PRINT*, 'T(',I,')='
READ*, T(I)
1 CONTINUE
END
```

PROGRAMME N°57

```
PROGRAM LECTURE_ELEMENTS_MATRICE  ! Lecture des éléments d'un Tableau de dimension
NxM --> matrice
DIMENSION T(50,50)
PRINT*, 'DONNER LE NOMBRE DE LIGNES'; READ*, N
PRINT*, 'DONNER LE NOMBRE DE COLONNES'; READ*, M
DO 1 I=1,N
DO 2 J=1,M
PRINT*, 'T(',I,',',J,')='
READ*, T(I,J)
2 CONTINUE
1 CONTINUE
END
```

PROGRAMME N°58

```
PROGRAM MAXIMUM  ! plus grande valeur d'un tableau  T de N éléments
DIMENSION T(100)
PRINT *, 'DONNER LE NOMBRE DES ELEMENTS N DU TABLEAU T'
READ *, N
PRINT *, 'DONNER CES ELEMENTS'
READ *, (T(I),I=1,N)
TOP = T(1)
DO 25,I = 2,N
TOP = MAX(T(I), TOP)
25 CONTINUE
PRINT *, TOP
END
```

PROGRAMME N°59

```
PROGRAM MINIMUM  ! plus petite valeur d'un tableau  T de N éléments
DIMENSION T(100)
PRINT *, 'DONNER LE NOMBRE DES ELEMENTS N DU TABLEAU T'
READ *, N
PRINT *
PRINT *, 'DONNER CES ELEMENTS'
READ *, (T(I),I=1,N)
TOP = T(1)
DO 25,I = 2,N
TOP = MIN(T(I), TOP)
25 CONTINUE
PRINT *, TOP
END
```

PROGRAMME N°60

```
PROGRAM DECOMPTE_VECTEUR  ! Lecture des éléments d'un tableau de dimension N --> vecteur et leur décompte par signe
DIMENSION T(50)
PRINT*, 'DONNER LA DIMENSION DU TABLEAU'; READ*, N
PRINT*, 'DONNER LES ELEMENTS DU TABLEAU'
DO 1 I=1,N
PRINT*, 'T(',I,')='
READ*, T(I)
IF(T(I)<0) THEN
NN=NN+1
ELSE
IF(T(I)>0) THEN
NP=NP+1
ELSE
N0=N0+1
ENDIF
ENDIF
1 CONTINUE
PRINT*
PRINT*, 'LE NOMBRE DE VALEURS NEGATIFS DU TABLEAU --->',NN
PRINT*
PRINT*, 'LE NOMBRE DE VALEURS POSITIFS DU TABLEAU --->',NP
PRINT*
PRINT*, 'LE NOMBRE DE VALEURS NULLES DU TABLEAU --->',N0
PRINT*
STOP
END
```

PROGRAMME N°61

```
PROGRAM SOMME_ELEMENTS_IMPAIRS_TABLEAU  !somme des éléments impairs d'un tableau T(i)
IMPLICIT NONE
INTEGER :: Q
INTEGER :: N,L,S
INTEGER, DIMENSION (100) :: T !PAR EXEMPLE
! LECTURE DE LA DIMENSION DU TABLEAU
PRINT*,'DONNER LES DIMENSIONS DU TABLEAU'; READ*, N
```

```
!LECTURE DES ÉLÉMENTS DU TABLEAU
PRINT*, 'DONNER LES ELEMENTS DU TABLEAU'
DO L=1,N
PRINT*, 'T(',L,')';READ*,T(L)
END DO
S=0.
DO L = 1,N
  Q=MOD(T(L),2) !Q=LE RESTE DE LA DIVISION DES ÉLÉMENTS DU TABLEAU SUR 2
  IF(Q==0) CYCLE ! ÉLIMINER TOUTS LES ÉLÉMENTS PAIRS DU TABLEAU
S=S+T(L)
END DO
PRINT *, 'S EST LA SOMME DES ELEMENTS IMPAIRS DU TABLEAU T:',S
STOP
END
```

PROGRAMME N°62

```
PROGRAM TRIABULLE ! Tri des éléments d'un vecteur dans le sens croissant et décroissant
IMPLICIT NONE
INTEGER, PARAMETER :: CROISSANT=1, DECROISSANT=2, N=10
REAL, DIMENSION(N) :: TAB
REAL                :: TEMP
LOGICAL             :: TRI_TERMINE, EXPR1, EXPR2
INTEGER             :: SENS, I, J
! LECTURE DES ÉLÉMENTS DU VECTEUR TAB.
PRINT*,'DONNER LES ELEMENTS DU VECTEUR A TRIE'
READ*,(TAB(J),J=1,N)
DO SENS=CROISSANT, DECROISSANT          ! SENS DU TRI
DO TRI_TERMINE = .TRUE.                                    ! TRI
DO I=2,N
EXPR1 = SENS == CROISSANT  .AND. TAB(I-1) > TAB(I)
EXPR2 = SENS == DECROISSANT .AND. TAB(I-1) < TAB(I)
IF (EXPR1 .OR. EXPR2) THEN
TRI_TERMINE = .FALSE.
TEMP = TAB(I-1); TAB(I-1) = TAB(I); TAB(I) = TEMP
END IF
END DO
IF (TRI_TERMINE) EXIT
END DO
! IMPRESSION DU VECTEUR TRIÉ.
IF (SENS == CROISSANT)   PRINT*, ' TRI CROISSANT '
PRINT*,'   '
IF (SENS == DECROISSANT) PRINT*, ' TRI DECROISSANT '
PRINT*, '   '
PRINT*, TAB
END DO
END PROGRAM TRIABULLE
```

PROGRAMME N°63

```
PROGRAM MOYNOTE
REAL TOTAL,NOTE,MOYENNE
INTEGER COUNTEUR
TOTAL = 0.
COUNTEUR = 0
```

```
PRINT *,'ENTREZ VOS NOTES UNE PAR UNE '
PRINT*
READ *, NOTE
!La lecture des données est stoper par la note fictive 999
10      IF (NOTE.NE.999) THEN
COUNTEUR = COUNTEUR+1
TOTAL = TOTAL+NOTE
READ *, NOTE
GOTO 10
END IF
IF (COUNTEUR.GT.0) THEN
MOYENNE = TOTAL/COUNTEUR
PRINT *, COUNTEUR,' ','NOTES ENTREES.'
PRINT*
PRINT *, 'MOYENNE DES NOTES : ', MOYENNE
ELSE
PRINT *, 'AUCUNE NOTE ENTREE'
END IF
END
```

PROGRAMME N°64

```
PROGRAM CLASSEXAM
REAL NOTE(5),TOTAL(5),MOYNOTE
NTEGER COUNTEUR(5)
DATA COUNTEUR/5*0/,TOTAL/5*0/
!Une note négative est attribuée aux étudiants n'ayant pas pris part de l'examen d'un module quelconque
PRINT*,'Donner les 5 premières notes'
READ *,(NOTE(I),I=1,5)
!La lecture des données est stoper par la note fictive 999
10      IF (NOTE(1).NE.999) THEN
DO 20, I=1,5
IF (NOTE(I).GE.0) THEN
COUNTEUR(I) = COUNTEUR(I)+1
TOTAL(I) = TOTAL(I)+NOTE(I)
END IF
20      CONTINUE
PRINT*,'Donner la suite des notes'
READ *,(NOTE(I),I=1,5)
GOTO 10
END IF
DO 30, I=1,5
IF (COUNTEUR(I).GT.0) THEN
MOYNOTE = TOTAL(I)/COUNTEUR(I)
PRINT *,COUNTEUR(I),' ','Etudiants ayant pris part de l',' ','examen du module :',I
PRINT*
PRINT *,'LA MOYENNE DES NOTES EST : ', MOYNOTE
PRINT*
ELSE
PRINT *,'Aucun Etudiant na pris part de l',' ','examen du module : ',I
PRINT*
END IF
30 CONTINUE
END
```

PROGRAMME N°65

```
PROGRAM VMEP_VMEN
! Calcul de la moyenne des éléments positifs d'un tableau ----> VMEP
! et Calcul de la moyenne des éléments negatifs d'un tableau ----> VMEPN
DIMENSION T(50)
PRINT*, 'DONNER LA DIMENSION DU TABLEAU'; READ*, N
PRINT*, 'DONNER LES ELEMENTS DU TABLEAU'
NP=0.
NN=0.
SUM = 0.0
SUM1 = 0.0
DO I=1,N
PRINT*, 'T(',I,')=' ; READ*, T(I)
IF(T(I)>0) THEN
NP=NP+1
SUM = SUM + T(I)
ELSE
IF(T(I)<0) THEN
NN=NN+1
SUM1 = SUM1 + T(I)
ENDIF
ENDIF
ENDDO
PRINT*
PRINT*, 'LE NOMBRE DES ELEMENTS POSITIFS DU TABLEAU --->',NP
VMEP = SUM / NP
PRINT *,'VALEUR MOYENNE DES ELEMENTS POSITIFS =', VMEP
PRINT*, 'LE NOMBRE DE VALEURS NEGATIFS DU TABLEAU --->',NN
VMEN = SUM1 / NN
PRINT *,'VALEUR MOYENNE DES ELEMENTS NEGATIFS =', VMEN
END
```

PROGRAMME N°66

```
PROGRAM MOYENNE_OBSERVATION
DIMENSION NOM(50)
REAL N1,N2,N3,MOY
CHARACTER*10 OBSERVATION
CHARACTER*15 NOM
OPEN(2,FILE='LISTE',STATUS='OLD') !FICHIER LISTE DES ÉTUDIANTS
OPEN (1,FILE='RESULT') !FICHIER DES RÉSULTATS
PRINT*,'DONNER LE NOMBRE D''ETUDIANTS'
READ*, N
WRITE(1,10) 'NOM','N1','N2','N3','MOYENNE','RESULTAT'
10 FORMAT(/,1X,A3,5X,A2,7X,A5,7X,A5,5X,A8,5X,A9,/)
DO I=1, N
READ (2,*) NOM(I)
PRINT*,'DONNER LES NOTES DE L''ETUDIANT :', NOM(I)
PRINT*,'N1=';READ*, N1
PRINT*,'N2=';READ*, N2
PRINT*,'N3=';READ*, N3
MOY=(N1+N2+N3)/3
IF(MOY<10) THEN
OBSERVATION='AJOURNE(E)'
ELSE
OBSERVATION='ADMIS(E)'
```

```
ENDIF
WRITE (1,12) NOM(I),N1,N2,N3,MOY,OBSERVATION
12 FORMAT(A8,4X,F5.2,6X,F5.2,6X,F5.2,4X,F5.2,6X,A10)
ENDDO
END
```

PROGRAMME N°67

```
PROGRAM PGTAB_PGITAB
DIMENSION V(50)
PG=0
PRINT*,'ENTREZ LA DIMENSION DU TABLEAU V'
READ*, N
PRINT*,'ENTREZ LES ELEMENTS DU TABLEAU V'
DO I=1,N
READ*,V(I)
IF(V(I)>PG) THEN
PG=V(I)
IPG=I
ENDIF
ENDDO
PRINT*,'LE NOMBRE LE PLUS GRAND ETAIT : ', PG
PRINT*,'IL A ETE SAISI EN POSITION NUMERO ', IPG
END
```

PROGRAMME N°68

```
PROGRAM TABLEAU_PRENOMS
CHARACTER*15 PRENOM
DIMENSION PRENOM(10)
OPEN(1,FILE='RES',STATUS='REPLACE')
PRINT*,'DONNER LE NOMBRE DE PRENOMS'
READ*, N
PRINT*,'DONNER CES PRENOMS'
DO I=1,N
READ*, PRENOM (I)
IF (PRENOM(I)(1:1)=='F') THEN
NF=NF+1
ELSE
IF(PRENOM(I)(1:1)=='H') THEN
NH=NH+1
ENDIF
ENDIF
ENDDO
IF(NF==0.AND.NH==0) THEN
WRITE(1,*)'LE TAB NE CONTIENT PAS DE PRENOM COMMENÇANT PAR F ET H'
ELSE
WRITE(1,*)'LE NOMBRE DE PRENOMS COMMENÇANT PAR F EST:', NF
WRITE(1,*)'LES PRENOMS SONT:'
DO I=1,N
IF(PRENOM(I)(1:1)=='F') THEN
WRITE(1,*) PRENOM (I)
ENDIF
ENDDO
WRITE(1,*)'LE NOMBRE DE PRENOMS COMMENÇANT PAR H EST', NH
WRITE(1,*)'LES PRENOMS SONT:'
```

```
DO I=1,N
IF(PRENOM(I)(1:1)=='H') THEN
WRITE(1,*) PRENOM (I)
ENDIF
ENDDO
END IF
END
```

PROGRAMME N°69

```
PROGRAM SOMME_ELEMENTS_MATRICE  ! Programme : calcul de la somme des elements d'un tableau n*m
DIMENSION TABL(100,100)
PRINT *, 'DONNER LES DIMENSIONS DU TABLEAU'
READ *, M, N
PRINT *, 'DONNER LES ELEMENTS DU TABLEAU'
DO 10,L = 1,N
DO 15,K = 1,M
PRINT *, 'TABL(',K,',',L,')'
READ *, TABL(K,L)
15  CONTINUE
10  CONTINUE
SUM = 0.0
DO 20,L = 1,N
DO 25,K = 1,M
SUM = SUM + TABL(K,L)
25  CONTINUE
20  CONTINUE
PRINT *, 'SUM===>',SUM
END
```

PROGRAMME N°70

```
PROGRAM TRIANGLE_PASCAL  !triangle de Pascal
IMPLICIT NONE
INTEGER :: N
INTEGER :: I,J
INTEGER, DIMENSION(100,100) :: A
!LECTURE DES DIMENSIONS DE LA MATRICE A
PRINT*,'DONNER LES DIMENSIONS DE LA MATRICE A'
READ*, N
! LES ÉLÉMENTS DE LA PREMIÈRE COLONNE ET DE LA DIAGONALE PRINCIPALE = 1
DO I=1,N
A(I,I)=1
A(I,1)=1
END DO
DO J=2,N-1
DO I=J+1,N
A(I,J)=A(I-1,J)+A(I-1,J-1)
ENDDO
ENDDO
PRINT*,'LE TRIANGLE DE PASCAL'
PRINT 1, ((A(I,J), J=1,N),I=1,N)
1 FORMAT(T10,10I4.0,/) !L'INDICE DEVANT I EST LA DIMENSION DE A
STOP
END
```

PROGRAMME N°71

```
PROGRAM TABLE_PYTHAGORE
IMPLICIT NONE
INTEGER :: N
INTEGER :: I,J
INTEGER, DIMENSION(100,100) :: C
! LECTURE DE LA DIMENSION DE LA TABLE DE PYTHAGORE
PRINT*,'DONNER LES DIMENSIONS DE LA TABLE'
READ*, N
! LES ÉLÉMENTS DE LA TABLE C(I,J)=I*J
DO I=1,N
DO J=1,N
C(I,J)=I*J
ENDDO
ENDDO
PRINT*,'LA TABLE DE PYTHAGORE EST'
PRINT 1, ((C(I,J), J=1,N),I=1,N)
1 FORMAT(T10,10I3,/)  !L'INDICE DEVANT I EST LA DIMENSION N
STOP
END
```

PROGRAMME N°72

```
PROGRAM PROD
REAL A(100), B(100), C(100)
PS=0.
PRINT *, 'DONNER LES DIMENSIONS DE A ET B'
READ *, M
DO 1 I=1,M
PRINT *,'A(',I,')='; READ *, A(I)
1 CONTINUE
DO 2 I=1,M
PRINT *,'B(',I,')='
2 READ *, B(I)
CALL DOT(M,A, B, C)
DO 3 I=1,M
PRINT *,'C(',I,')=',C(I)
3 PS=PS+C(I)
PRINT*, 'LE PRODUIT SCALAIRE PS =', PS
END

SUBROUTINE DOT (M,X, Y, Z)
! Calcul le produit scalaire des tableaux X et Y de m éléments
! le produit est calculé dans un tableau Z de même ordre
REAL X(100), Y(100), Z(100)
DO 15, I = 1,M
Z(I) = X(I) * Y(I)
15 CONTINUE
END
```

PROGRAMME N°73

```
PROGRAM PRODUITSCALAIRE  ! Calcul du produit scalaire de deux vecteurs de dimension N
DIMENSION U(50),V(50)
SCALAIRE=0.
PRINT*,'DONNER LES DIMENSIONS DES VECTEURS U ET V'
PRINT*
PRINT*,'N=';READ *,N
PRINT*
PRINT*,'DONNER LES COMPOSANTES DU VECTEUR U'
PRINT*
DO 10 I=1,N
PRINT*,'U(',I,')=';READ *, U(I)
10 CONTINUE
PRINT*,'DONNER LES COMPOSANTES DU VECTEUR V'
PRINT*
DO 11 I=1,N
PRINT*,'V(',I,')='; READ *,V(I)
11 CONTINUE
PRINT*
DO 12 I=1,N
SCALAIRE=SCALAIRE +U(I)*V(I)
12 CONTINUE
PRINT*
PRINT*,'LA VALEUR DU PRODUIT SCALAIRE EST :',SCALAIRE
IF (SCALAIRE==0.) THEN
PRINT*
PRINT*, '------> LES 2 VECTEURS SONT ORTHOGONAUX <------'
ENDIF
PRINT*
END
```

PROGRAMME N°74

```
PROGRAM PRODUITVECTORIEL  ! Calcul du produit vectoriel de deux vecteurs
DIMENSION U(3),V(3)
PRINT*,'DONNER LES COMPOSANTES DU VECTEUR U'
PRINT*
DO 10 I=1,3
PRINT*,'U(',I,')=';READ *, U(I)
10 CONTINUE
UX=U(1);UY=U(2);UZ=U(3) !IDENTIFICATION DES COMPOSANTES DE U
PRINT*,'DONNER LES COMPOSANTES DU VECTEUR V'
PRINT*
DO 11 I=1,3
PRINT*,'V(',I,')='; READ *,V(I)
11 CONTINUE
VX=V(1);VY=V(2);VZ=V(3) !IDENTIFICATION DES COMPOSANTES DE V
PRINT*
WX=(UY*VZ)-(UZ*VY)
WY=-((UX*VZ)-(UZ*VX))
WZ=(UX*VY)-(UY*VX)
PRINT*
PRINT*,'LES COMPOSANTES DU VECTEUR W=U X V SONT:'
PRINT*
PRINT*, 'WX=',WX
PRINT*,'WY=',WY
```

```
PRINT*,'WZ=',WZ
PRINT*
IF(WX==0.) THEN ; IF(WY==0.) THEN ; IF(WZ==0.) THEN
PRINT*,'------> LES 2 VECTEURS SONT COLINEAIRES <------'
PRINT*
ENDIF
ENDIF
ENDIF
STOP
END
```

PROGRAMME N°75

```
PROGRAM PRODUITMIXTE  ! Calcul du produit mixte de trois vecteurs
DIMENSION U(3), V(3), W(3)
PRINT*,'DONNER LES COMPOSANTES DU VECTEUR U'
PRINT*
DO 10 I=1,3
PRINT*,'U(',I,')=';READ *, U(I)
10 CONTINUE
UX=U(1);UY=U(2);UZ=U(3) !IDENTIFICATION DES COMPOSANTES DE U
PRINT*,'DONNER LES COMPOSANTES DU VECTEUR V'
PRINT*
DO 11 I=1,3
PRINT*,'V(',I,')='; READ *,V(I)
11 CONTINUE
VX=V(1);VY=V(2);VZ=V(3) !IDENTIFICATION DES COMPOSANTES DE V
PRINT*
PRINT*,'DONNER LES COMPOSANTES DU VECTEUR W'
PRINT*
DO 12 I=1,3
PRINT*,'W(',I,')='; READ *,W(I)
12 CONTINUE
WX=W(1);WY=W(2);WZ=W(3) !IDENTIFICATION DES COMPOSANTES DE W
PRINT*
SCALAIRE=UX*(VY*WZ-VZ*WY)-UY*(VX*WZ-VZ*WX)+UZ*(VX*WY-VY*WX)
PRINT*
PRINT*,'LA VALEUR DU PRODUIT MIXTE EST :',SCALAIRE
IF (SCALAIRE==0.) THEN
PRINT*
PRINT*, '------> LES 3 VECTEURS SONT COPLANAIRES <------'
ENDIF
PRINT*
STOP
END
```

PROGRAMME N°76

```
PROGRAM DETERMINANT_ORDRE_2_MATRICE_CARREE
DIMENSION A(5,5)
WRITE(*,100)
100 FORMAT(1X,'DONNEZ LA DIMENSION DE LA MATRICE A',/,36(1H-),/)
PRINT*, 'N=';READ*, N
WRITE(*,200)
200 FORMAT(1X,'DONNEZ LES ELEMENTS DE LA MATRICE CAREE A',/,42(1H-),/)
DO I=1,N
DO J=1,N
```

```
PRINT*,  'A(',I,',',J,')=';  READ*,  A(I,J)
ENDDO
ENDDO
WRITE(*,300)
300  FORMAT(1X,'LE  DETERMINANT  DE  LA  MATRICE  A  :  ',/,33(1H-),/)
X=A(1,1)*A(2,2)
Y=A(1,2)*A(2,1)
DELTA=X-Y
PRINT*,  'DETLA=',  DELTA
STOP
END
```

PROGRAMME N°77

```
PROGRAM  DETERMINANT_ORDRE_3_MATRICE_CARREE
DIMENSION  A(5,5)
WRITE(*,100)
100  FORMAT(1X,'DONNEZ  LA  DIMENSION  DE  LA  MATRICE  A',/,36(1H-),/)
PRINT*,  'N=';READ*,  N
WRITE(*,200)
200  FORMAT(1X,'DONNEZ  LES  ELEMENTS  DE  LA  MATRICE  CAREE  A',/,42(1H-),/)
DO  I=1,N
DO  J=1,N
PRINT*,  'A(',I,',',J,')=';  READ*,  A(I,J)
ENDDO
ENDDO
WRITE(*,300)
300  FORMAT(1X,'LE  DETERMINANT  DE  LA  MATRICE  A  :  ',/,33(1H-),/)
X=A(1,1)*A(2,2)*A(3,3)+A(2,1)*A(3,2)*A(1,3)+A(1,2)*A(2,3)*A(3,1)
Y=A(2,2)*A(1,3)*A(3,1)+A(1,1)*A(3,2)*A(2,3)+A(3,3)*A(2,1)*A(1,2)
DELTA=X-Y
PRINT*,  'DETLA=',  DELTA
STOP
END
```

PROGRAMME N°78

```
PROGRAM  DETERMINANT_ORDRE_N_MATRICE_CARREE
DIMENSION  A(100,100)
WRITE(*,100)
100  FORMAT(1X,'DONNEZ  LA  DIMENSION  DE  LA  MATRICE  A',/,36(1H-),/)
PRINT*,  'N=';READ*,  N
WRITE(*,200)
200  FORMAT(1X,'DONNEZ  LES  ELEMENTS  DE  LA  MATRICE  CAREE  A',/,42(1H-),/)
DO  I=1,N
DO  J=1,N
PRINT*,  'A(',I,',',J,')=';  READ*,  A(I,J)
ENDDO
ENDDO
B=1
P=1
18  I=P
L=P
K=P
4  J=P
IF(A(K,L)==0)  GOTO  1
X=A(K,L)
B=B*A(K,L)
```

```
2     A(I,J)=A(I,J)/X
IF(J==N)  GOTO  1
J=J+1
GOTO  2
1     IF(I==N)  GOTO  3
I=I+1
K=K+1
GOTO  4
3  I=P
J=P
IF(A(I,J)==1)  GOTO  5
7  A(N+1,J)=  A(I,J)
IF(J==N)  GOTO  6
J=J+1
GOTO  7
6  I=P+1
J=P
13     IF(A(I,J)==0)  GOTO  8
10  A(P,J)=  A(I,J)
A(I,J)=  A(N+1,J)
IF(J==N)  GOTO  9
J=J+1
GOTO  10
9  B=-B
IF(I==N)  GOTO  11
GOTO  5
8  IF(I==N)  GOTO  12
I=I+1
GOTO  13
12  D=0
GOTO  14
5  I=I+1
J=P
IF(A(I,J)==1)  GOTO  15
GOTO  16
15     A(I,J)=A(I,J)-A(P,J)
IF(J==N)  GOTO            16
J=J+1
GOTO  15
16     IF(I==N)  GOTO  11
GOTO  5
11  IF(P==N-1)  GOTO  17
P=P+1
GOTO  18
17     D=B*A(N,N)
14  WRITE(*,300)
300  FORMAT(1X,'LE  DETERMINANT  DE  LA  MATRICE  A  :  ',/,33(1H-),/)  ;  PRINT*,'D=',D
END
```

PROGRAMME N°79

```
PROGRAM  SOMME_MATRICE    !somme  de  deux  matrices  A+B=C
IMPLICIT  NONE
INTEGER  ::  N,M
INTEGER  ::  I,J
REAL,  DIMENSION(100,100)  ::  A
REAL,  DIMENSION(100,100)  ::  B
```

```
REAL, DIMENSION(100,100) :: C
! LECTURE DES DIMENSIONS DES MATRICES A ET B
PRINT*,'DONNER LES DIMENSIONS DES MATRICES A ET B'
READ*, N,M
! LECTURE DES ÉLÉMENTS DES MATRICES A ET B
DO I=1,N
DO J=1,M
PRINT*, 'A(',I,',',J,')='
READ*, A(I,J)
END DO
END DO
DO I=1,N
DO J=1,M
PRINT*, 'B(',I,',',J,')='
READ*, B(I,J)
END DO
END DO
! SOMME DE MATRICE
DO I=1,N
DO J=1,M
C(I,J) = A(I,J) + B(I,J)
END DO
END DO
! IMPRESSION DE LA MATRICE C
DO I=1,N
DO J=1,M
PRINT*,'C(',I,',',J,')=',C(I,J)
END DO
END DO
STOP
END
```

PROGRAMME N°80

```
PROGRAM PRODUIT_MATRICE  ! produit de deux matrices A*B=C
IMPLICIT NONE
INTEGER :: N,M,P
INTEGER :: I,J,K
REAL, DIMENSION(100,100) :: A
REAL, DIMENSION(100,100) :: B
REAL, DIMENSION(100,100) :: C
! LECTURE DES DIMENSIONS DES MATRICES A ET B
PRINT*,'DONNER LES DIMENSIONS DES MATRICES A ET B'
READ*, N,M,P
! LECTURE DES ÉLÉMENTS DES MATRICES A ET B
DO I=1,N
DO K=1,M
PRINT*, 'A(',I,',',K,')='
READ*, A(I,K)
END DO
END DO
DO K=1,M
DO J=1,P
PRINT*, 'B(',K,',',J,')='
READ*, B(K,J)
END DO
END DO
```

```
! PRODUIT DE MATRICE
DO I=1,N
DO J=1,P
C(I,J) = 0.
DO K=1,M
C(I,J) = C(I,J) + A(I,K) * B(K,J)
END DO
END DO
END DO
! IMPRESSION DE LA MATRICE C
DO I=1,N
DO J=1,P
PRINT*,'C(',I,',',J,')=',C(I,J)
END DO
END DO
STOP
END
```

PROGRAMME N°81

```
PROGRAM PRODUIT_MATRICE_BIS  !produit de deux matrices A*B=C en utilisant l'instruction matmul
IMPLICIT NONE
INTEGER :: N,M,P
INTEGER :: I,J,K
REAL, DIMENSION(100,100) :: A
REAL, DIMENSION(100,100) :: B
REAL, DIMENSION(100,100) :: C
!LECTURE DES DIMENSIONS DES MATRICES A ET B
PRINT*,'DONNER LES DIMENSIONS DES MATRICES A ET B'
READ*, N,M,P
! LECTURE DES ÉLÉMENTS DES MATRICES A ET B
DO I=1,N
DO K=1,M
PRINT*, 'A(',I,',',K,')='
READ*, A(I,K)
END DO
END DO
DO K=1,M
DO J=1,P
PRINT*, 'B(',K,',',J,')='
READ*, B(K,J)
END DO
END DO
! PRODUIT DE MATRICE
C=MATMUL(A,B)
! IMPRESSION DE LA MATRICE C
DO I=1,N
DO J=1,P
PRINT*,'C(',I,',',J,')=',C(I,J)
END DO
END DO
STOP
END
```

PROGRAMME N°82

```
PROGRAM TRANSPOSITION_MATRICE   ! Transposition d'une matrice A
IMPLICIT NONE
INTEGER :: N,M
INTEGER :: I,J
REAL, DIMENSION(100,100) :: A !PAR EXEMPLE
REAL, DIMENSION(100,100) :: B !PAR EXEMPLE
! LECTURE DES DIMENSIONS DE LA MATRICE A
PRINT*,'DONNER LES DIMENSIONS DE LA MATRICE A' ; READ*, N,M
! LECTURE DES ÉLÉMENTS DE LA MATRICE A
DO I=1,N
DO J=1,M
PRINT*, 'A(',I,',',J,')='; READ*, A(I,J)
END DO
END DO
!TRANSPOSITION DE LA MATRICE A
DO I=1,N
DO J=1,M
B(J,I)=A(I,J)
ENDDO
ENDDO
! IMPRESSION DE LA MATRICE TRANSPOSÉE
DO J=1,M
DO I=1,N
PRINT*,'B(',J,',',I,')=',B(J,I)
END DO
END DO
END
```

PROGRAMME N°83

```
PROGRAM TRANSPOSITION_MATRICE_BIS   ! Utilisant l'instruction transpose
IMPLICIT NONE
INTEGER :: N,M
INTEGER :: I,J
REAL, DIMENSION(100,100) :: A
REAL, DIMENSION(100,100) :: B
! LECTURE DES DIMENSIONS DE LA MATRICE A
PRINT*,'DONNER LES DIMENSIONS DE LA MATRICE A'
READ*, N,M
! LECTURE DES ÉLÉMENTS DE LA MATRICE A
DO I=1,N
DO J=1,M
PRINT*, 'A(',I,',',J,')='
READ*, A(I,J)
END DO
END DO
! TRANSPOSITION DE LA MATRICE A
B=TRANSPOSE(A)
! IMPRESSION DE LA MATRICE TRANSPOSÉE
DO J=1,M
DO I=1,N
PRINT*,'B(',J,',',I,')=',B(J,I)
END DO
END DO
END
```

PROGRAMME N°84

```
PROGRAM  INVERSION_MATRICE_CARREE_N
DIMENSION  A(20,20)
WRITE(*,100)
100  FORMAT(1X,'DONNEZ  LA  DIMENSION  DE  LA  MATRICE  A',/,36(1H-),/)
PRINT*,  'N=';READ*,  N
WRITE(*,200)
200  FORMAT(1X,'DONNEZ  LES  ELEMENTS  DE  LA  MATRICE  A',/,36(1H-),/)
DO  I=1,N
DO  J=1,N
PRINT*,  'A(',I,',',J,')=';  READ*,  A(I,J)
ENDDO
ENDDO
WRITE(*,300)
300  FORMAT(1X,'LES  ELEMENTS  DE  LA  MATRICE  INVERSE  :',/,36(1H-),/)
CALL  INV(A,N)
DO  I=1,N
DO  J=1,N
PRINT*,  'A(',I,',',J,')=',A(I,J)
ENDDO
ENDDO
END
SUBROUTINE  INV(A,NN)
DIMENSION  A(20,20)
DO  I=1,NN
X=A(I,I)
A(I,I)=1
DO  J=1,NN
A(I,J)=A(I,J)/X
ENDDO
DO  K=1,NN
IF(K.EQ.I)  GOTO  5
X=A(K,I)
A(K,I)=0
DO  L=1,NN
A(K,L)=A(K,L)-X*A(I,L)
ENDDO
5  ENDDO
ENDDO
RETURN
END
```

PROGRAMME N°85

```
PROGRAM  INVERSION_PAR_DOOLITTLE
REAL  A(100,100),C(100,100)
REAL  B(100),X(100)
PRINT  *,  'DONNER  ORDRE  DU  SYSTEME';  READ  *,  N
PRINT  *,  'DONNER  LES  ELEMENTS  DE  LA  MATRICE  A'
DO  I=1,N
DO  J=1,N
PRINT  *,'A(',I,',',J,')=';  READ  *,  A(I,J)
ENDDO
ENDDO
PRINT  *,  'DONNER  LES  ELEMENTS  DU  VECTEUR  B'
```

```
DO  I=1,N
PRINT  *,'B(',I,')=';  READ  *,  B(I)
ENDDO
CALL  INVERSE(A,B,N,C,X)
PRINT*,'LA  MATRICE  INVERSE  C=A**(-1)'
DO  I=1,N
DO  J=1,N
PRINT  *,'C(',I,J,')=',C(I,J)
ENDDO
ENDDO
END
SUBROUTINE  INVERSE(A,B,N,C,X)
REAL  A(100,100),C(100,100)
REAL  L(100,100),U(100,100),B(100),Y(100),X(100)
!  INITIATION  DES  MATRICES  L,  U  ET  B
K=0.0
L=0.0
B=0.0
!  CALCUL  DES  COEFFICIENTS  DE  LA  MATRICE  L
DO  K=1,N-1
DO  I=K+1,N
COEFF=A(I,K)/A(K,K)
L(I,K)=COEFF
DO  J=K+1,N
A(I,J)=A(I,J)-COEFF*A(K,J)
ENDDO
ENDDO
ENDDO
!  ELEMENTS  DIAGONAUX  DE  LA  MATRICE  L
DO  I=1,N
L(I,I)=1.0
ENDDO
!  DETERMINATION  DES  ELEMENTS  DE  LA  MATRICE  U
DO  J=1,N
DO  I=1,J
U(I,J)=A(I,J)
ENDDO
ENDDO
!  RESOUDRE  LY=B  PAR  SUBSTITUTION  AVANT
DO  K=1,N
B(K)=1.0
Y(1)=B(1)
DO  I=2,N
Y(I)=B(I)
DO  J=1,I-1
Y(I)=Y(I)-L(I,J)*Y(J)
ENDDO
ENDDO
!  RESOUDRE  UX=Y  PAR  SUBSTITUTION  ARRIERE
X(N)=Y(N)/U(N,N)
DO  I=N-1,1,-1
X(I)=Y(I)
DO  J=N,I+1,-1
X(I)=X(I)-U(I,J)*X(J)
ENDDO
```

```
X(I)=X(I)/U(I,I)
ENDDO
! REMPLACER LA SOLUTION X(N) DANS LA COLONNE K DE C
DO I=1,N
C(I,K)=X(I)
ENDDO
B(K)=0.0
ENDDO
RETURN
END
```

PROGRAMME N°86

```
PROGRAM EQUATION_PREMIER_ORDRE  ! Résolution de l'équation du premier degré ax+b=0
IMPLICIT NONE
REAL A, B, X
! LECTURE DES COEFFICIENTS A, B, ET C
PRINT*, 'DONNER A, B'
READ*, A,B
! A DOIT ÊTRE NON NUL.
IF (A.NE.0) THEN
X = -B/A
! IMPRESSION DE LA SOLUTION
PRINT *,'X = ', X
ELSE
STOP 'A DOIT ÊTRE NON NUL, SYSTÈME IMPOSSIBLE'
END IF
END
```

PROGRAMME N°87

```
PROGRAM EQUATION_SECONDE_ORDRE  ! Calcul des racines du trinôme a*(x*x)+b*x+c
IMPLICIT NONE
REAL, PARAMETER :: EPSILON = 1E-6
REAL A, B, C
REAL DELTA, R_DELTA, X1, X2
! LECTURE DES COEFFICIENTS A, B, ET C
PRINT*, 'DONNER A, B, C'
READ*, A,B,C
! A DOIT ÊTRE NON NUL.
IF ( A > -EPSILON .AND. A < EPSILON ) &
STOP 'A DOIT ÊTRE NON NUL'
! CALCUL DU DÉTERMINANT.
DELTA = B*B - 4*A*C
! CAS DU DÉTERMINANT NÉGATIF.
IF( DELTA < -EPSILON ) STOP 'PAS DE RACINE RÉELLE'
! CAS DU DÉTERMINANT NUL.
IF ( DELTA > -EPSILON .AND. DELTA < EPSILON ) THEN
X1 = -B/(2*A); X2 = X1
ELSE ! CAS DU DÉTERMINANT POSITIF.
R_DELTA = SQRT( DELTA )
X1 = (-B - R_DELTA)/(2*A); X2 = (-B + R_DELTA)/(2*A)
END IF
! IMPRESSION DES RACINES.
PRINT *,'X1 = ', X1 , 'X2 = ', X2
END
```

PROGRAMME N°88

```
PROGRAM  RACINE_ENTIER_POSITIF
REAL  X,Y
INTEGER  A
X=1
PRINT*,  'DONNER  LA  VALEUR  DE  A'
READ  *,  A
10     Y=(X+A/X)/2
EPSILON=ABS((Y-X)/X)
DO  WHILE  (EPSILON>1E-6)
X=Y
GOTO  10
ENDDO
PRINT*,'LA  RACINE  CARREE  DE',A,'=',  Y
END
```

PROGRAMME N°89

```
PROGRAM  RACINE_2_NEWTON
IMPLICIT  NONE
!  DÉCLARATION  DES  VARIABLES
INTEGER  ::  N
REAL  ::  U
!  INITIALISATION
U=1.0
!BOUCLE
DO  N=1,10
U=U/2.0+1.0/U
END  DO
!  AFFICHAGE
PRINT*,'APPROXIMATION  DE  RACINE  DE  2  PAR  NEWTON  :',U
END
```

PROGRAMME N°90

```
PROGRAM  SUITE_FIBONACCI
REAL  NBOR,U
INTEGER  R
DIMENSION  U(500)
!  Valeur  de  U(0)  et  U(1)  ?'
U(0)=0
U(1)=1
1  PRINT  3
3  FORMAT(2X,'DONNER  LE  RANG  MAXIMUM  R  :  ',$)
READ*,  R
IF  (R<=1)  THEN
PRINT*,'DESOLE!!  LE  RANG  MAXIMUM  R  DOIT  ETRE  >  1';  GOTO  1
else
print*
PRINT*,  'LA  SUITE  DE  FIBONACCI  EST  :'
PRINT  *,'U(',0,')=',U(0)
print*
PRINT  *,'U(',1,')=',U(1)
print*
DO  n=2,R
U(n)=U(n-2)+U(n-1)
```

```
PRINT  2,'U(',n,')=',U(n-2),'+',U(n-1),'=',U(n)
2  FORMAT(A3,I3,A2,F10.0,A1,F10.0,A1,F10.0)
PRINT*,'LE  NOMBRE  D''OR  CONVERGE  VERS  :'
NBOR=u(n)/u(n-1)
print  4,  'NBOR=',NBOR
4  format  (A7,F8.4)
print*
END  DO
ENDIF
STOP
END
```

PROGRAMME N°91

```
PROGRAM  INTEGRALE_TRAPEZES
IMPLICIT  NONE
INTEGER  N
REAL  *4  A,B
REAL  *4  TRAPEZE
REAL  *4  FUNC
EXTERNAL  TRAPEZE
EXTERNAL  FUNC
!  LECTURE  DES  DONNÉES
PRINT*,'DONNER  LA  BORNE  INF  A_INFERIEURE'
READ*,  A
PRINT*,'DONNER  LA  BORNE  SUP  B_SUPERIEURE'
READ*,  B
PRINT*,'DONNER  LE  NOMBRE  DE  PAS  NECESSAIRE  ENTIER  >=1'
READ*,  N
!  AFFICHAGE  DE  LA  VALEUR  DE  L'INTÉGRALE
PRINT*,'LA  VALEUR  DE  INTEGRALE  DE  LA  FONCTION  EST  :  ',  TRAPEZE(A,B,N,FUNC)
END  !FIN  DU  PROGRAMME  PRINCIPAL
!  FONCTION  A  INTÉGRER
REAL  FUNCTION  FUNC(X)
IMPLICIT  NONE
REAL  *4  X
FUNC=1.0/(1.0+X*X)
!  FUNC=EXP(-X*X/2)
RETURN
END  !FIN  DE  LA  FONCTION  A  INTEGRER
!  MÉTHODE  DES  TRAPÈZES
FUNCTION  TRAPEZE(A,B,N,FUNC)
IMPLICIT  NONE
REAL  *4  TRAPEZE,A,B,FUNC
REAL  *4  SUM,DELTA
INTEGER  N
INTEGER  I
EXTERNAL  FUNC
SUM=0.0
DELTA=(B-A)/REAL(N)
DO  I=1,N-1
SUM=SUM+FUNC(A+DELTA*I)
ENDDO
TRAPEZE=DELTA*(SUM+0.5*(FUNC(A)+FUNC(B)))
RETURN
END  !FIN  DE  LA  MÉTHODE  DES  TRAPÈZES
```

PROGRAMME N°92

```
PROGRAM CRAMER  !méthode de Cramer
IMPLICIT NONE
REAL U1,U2
REAL V1,V2
REAL W1,W2
REAL DELTA, DELTA_X, DELTA_Y
REAL X,Y
! LECTURE DES COEFFICIENTS U1, V1, W1, U2, V2, W2
PRINT*,'DONNER U1,V1 ET W1'
READ*, U1,V1,W1
PRINT*,'DONNER U2, V2 ET W2'
READ*, U2,V2,W2
! CALCUL DU DÉTERMINANT PRINCIPAL
DELTA = U1*V2 - U2*V1
IF ( DELTA==0) THEN
PRINT *, "LE SYSTÈME N'A PAS DE SOLUTIONS"
STOP
END IF
! CALCUL DU DÉTERMINANT EN X
DELTA_X = W1*V2 - W2*V1
! CALCUL DU DÉTERMINANT EN Y
DELTA_Y = U1*W2 - U2*W1
! CALCUL DES SOLUTIONS
X = DELTA_X/DELTA
Y = DELTA_Y/DELTA
! IMPRESSION DES SOLUTIONS
PRINT *, 'X = ', X, ', Y = ', Y
STOP
END
```

PROGRAMME N°93

```
PROGRAM ELIMINATION_GAUSS
REAL A(100,100),A1(100,100)
REAL B(100),X(100)
PRINT *, 'DONNER ORDRE DU SYSTEME'; READ *, N
PRINT*
PRINT *, 'DONNER LES ELEMENTS DE LA MATRICE A'
DO I=1,N
DO J=1,N
PRINT *,'A(',I,',',J,')='; READ *, A(I,J)
ENDDO
ENDDO
PRINT *, 'DONNER LES ELEMENTS DU VECTEUR B'
DO I=1,N
PRINT *,'B(',I,')='; READ *, B(I)
ENDDO
CALL GAUSS(A,B,N,A1,X)
PRINT*
PRINT*,'ELEMENTS DE LA MATRICE A1 TRIANGULAIRE SUPERIEURE'
DO I=1,N
DO J=1,N
PRINT *,'A1(',I,',',J,')=',A1(I,J)
ENDDO
ENDDO
```

```
PRINT*
PRINT*,'LA SOLUTION DU SYSTEME AX=B'
DO I=1,N
PRINT *,'X(',I,')=',X(I)
ENDDO
PRINT*
PRINT*,'LE DETERMINANT DE LA MATRICE A'
DETA=A1(1,1)
DO I=2,N
DETA=DETA*A1(I,I)
ENDDO
PRINT *,'DET(A)=',DETA
STOP
END
SUBROUTINE GAUSS(A,B,N,A1,X)
REAL A(100,100),A1(100,100)
REAL L(100,100),B(100),X(100)
! CALCUL DES COEFFICIENTS DE LA MATRICE L
DO K=1,N-1
! TEST : ON TESTE SI LE PIVOT EST NUL
IF(A(K,K)==0.0)THEN
PRINT*,'PIVOT NUL:PERMUTER VOS LIGNES'
ENDIF
DO I=K+1,N
COEFF=A(I,K)/A(K,K)
L(I,K)=COEFF
B(I)=B(I)-L(I,K)*B(K)
DO J=K+1,N
A(I,J)=A(I,J)-COEFF*A(K,J)
ENDDO
ENDDO
ENDDO
! DETERMINATION DES ELEMENTS DE LA MATRICE A1 TRIANGULAIRE SUPERIEURE
DO J=1,N
DO I=1,J
A1(I,J)=A(I,J)
ENDDO
ENDDO
! RESOUDRE A1X=B
X(N)=B(N)/A1(N,N)
DO I=N-1,1,-1
X(I)=B(I)
DO J=N,I+1,-1
X(I)=X(I)-A1(I,J)*X(J)
ENDDO
X(I)=X(I)/A1(I,I)
ENDDO
RETURN
END
```

PROGRAMME N°94

```
PROGRAM CHOLESKY
REAL A(100,100),L(100,100),Lt(100,100),B(100),X(100),Y(100)
PRINT *, 'DONNER ORDRE DU SYSTEME'; READ *, N
PRINT *, 'DONNER LES ELEMENTS DE LA MATRICE A'
DO I=1,N
DO J=1,N
PRINT *,'A(',I,',',J,')='; READ *, A(I,J)
ENDDO
ENDDO
PRINT*
! TEST : ON TESTE SI LA MATRICE EST SYMETRIQUE
DO I=1,N
DO J=1,N
IF (A (J,I)==A(I,J))THEN
GOTO 1
ELSE
PRINT*,'IMPOSSIBLE DE RESOUDRE LE SYSTEME'
PRINT*,'LA METHODE DE CHOLESKY APPLIQUABLE QUE SUR LES MATRICES SYMETRIQUES'
ENDIF
GOTO 2
1 ENDDO
ENDDO
PRINT *, 'DONNER LES ELEMENTS DU VECTEUR B'
DO I=1,N
PRINT *,'B(',I,')='; READ *, B(I)
ENDDO
CALL CHOL(A,B,N,L,Lt,X,Y)
! AFFICHAGE DE LA MATRICE L
PRINT*, 'ELEMENTS DE LA MATRICE L'
DO I=1,N
DO J=1,N
PRINT *,'L(',I,',',J,')=',L(I,J)
ENDDO
ENDDO
! AFFICHAGE DE LA MATRICE Lt
PRINT*,'ELEMENTS DE LA MATRICE Lt'
DO I=1, N
DO J=1, N
PRINT *,'Lt(',I,',',J,')=',Lt(I,J)
ENDDO
ENDDO
! AFFICHAGE DE LA SOLUTION DU SYSTEME
PRINT*,'LA SOLUTION DU SYSTEME LY=B'
DO I=1,N
PRINT *,'Y(',I,')=',Y(I)
ENDDO
PRINT*
PRINT*,'LA SOLUTION DU SYSTEME AX=B'
DO I=1,N
PRINT *,'X(',I,')=',X(I)
ENDDO
! DEDUCTION DU DETERMINANT DE LA MATRICE A
PRINT*,'LE DETERMINANT DE LA MATRICE A'
```

```
DETA=L(1,1)
DO  I=2,N
DETA=DETA*L(I,I)
ENDDO
PRINT*,'DET(A)=',  DETA**2
2  STOP
END
SUBROUTINE  CHOL(A,B,N,L,Lt,X,Y)
REAL  A(100,100)
REAL  L(100,100),Lt(100,100),B(100),Y(100),X(100)
! INITIALITATION DES MATRICES L ET U
L=0.0
Lt=0.0
! ELEMENTS DE LA MATRICE L
DO  I=1,N-1
DO  K=1,I-1
COEFF=COEFF+(L(I,K)*L(I,K))
ENDDO
L(I,I)=  SQRT(A(I,I)-COEFF)
DO  J=I+1,N
DO  K=1,I-1
COEFF1=COEFF1+(L(J,K)*L(I,K))
ENDDO
L(J,I)=(A(I,J)-COEFF1)/L(I,I)
ENDDO
ENDDO
DO  J=1,N-1
COEFF2=COEFF2+(L(N,J)*L(N,J))
ENDDO
L(N,N)=SQRT(A(N,N)-COEFF2)
!DETERMINATION DES ELEMENTS DE LA MATRICE Lt
DO  I=1,N
DO  J=1,N
Lt(J,I)=L(I,J)
ENDDO
ENDDO
!RESOUDRE LY=B PAR SUBSTITUTION AVANT
Y(1)=B(1)/L(1,1)
DO  I=2,N
Y(I)=B(I)
DO  J=1,I-1
Y(I)=Y(I)-L(I,J)*Y(J)
ENDDO
Y(I)=Y(I)/L(I,I)
ENDDO
!RESOUDRE LtX=Y PAR SUBSTITUTION ARRIERE
X(N)=Y(N)/Lt(N,N)
DO  I=N-1,1,-1
X(I)=Y(I)
DO  J=N,I+1,-1
X(I)=X(I)-Lt(I,J)*X(J)
ENDDO
X(I)=X(I)/Lt(I,I)
ENDDO
RETURN
END
```

PROGRAMME N°95

```
PROGRAM CHOLESKY_BIS
REAL A(100,100),L(100,100),Lt(100,100),B(100),X(100),Y(100)
PRINT *, 'DONNER ORDRE DU SYSTEME'; READ *, N
PRINT*
PRINT *, 'DONNER LES ELEMENTS DE LA MATRICE A'
DO I=1,N
DO J=1,N
PRINT *,'A(',I,',',J,')='; READ *, A(I,J)
ENDDO
ENDDO
!TEST : ON TESTE SI LA MATRICE EST SYMETRIQUE
DO I=1,N
DO J=1,N
IF(A(J,I)==A(I,J))THEN
GOTO 1
ELSE
PRINT*,'IMPOSSIBLE DE RESOUDRE LE SYSTEME'
PRINT*,'LA METHODE DE CHOLESKY ===> APPLIQUABLE QUE SUR LES MATRICES
SYMETRIQUES'
ENDIF
GOTO 2
1 ENDDO
ENDDO
PRINT *, 'DONNER LES ELEMENTS DU VECTEUR B'
DO I=1,N
PRINT *,'B(',I,')='; READ *, B(I)
ENDDO
CALL CHOL(A,B,N,L,Lt,X,Y)
PRINT*
!AFFICHAGE DE LA MATRICE L
PRINT*, 'ELEMENTS DE LA MATRICE L'
DO I=1,N
DO J=1, N
PRINT *,'L(',I,',',J,')=',L(I,J)
ENDDO
ENDDO
!AFFICHAGE DE LA MATRICE Lt
PRINT*,'ELEMENTS DE LA MATRICE Lt'
DO I=1,N
DO J=1,N
PRINT *,'Lt(',I,',',J,')=',Lt(I,J)
ENDDO
ENDDO
!AFFICHAGE DE LA SOLUTION DU SYSTEME
PRINT*,'LA SOLUTION DU SYSTEME LY=B'
DO I=1,N
PRINT *,'Y(',I,')=',Y(I)
ENDDO
PRINT*
PRINT*,'LA SOLUTION DU SYSTEME AX=B'
DO I=1,N
PRINT *,'X(',I,')=',X(I)
ENDDO
```

```
!DEDUCTION DU DETERMINANT DE LA MATRICE A
PRINT*,'LE DETERMINANT DE LA MATRICE A'
DETA=L(1,1)
DO I=2,N
DETA=DETA*L(I,I)
ENDDO
PRINT*,'DET(A)=', DETA**2
2 STOP
END
SUBROUTINE CHOL(A,B,N,L,Lt,X,Y)
REAL A(100,100)
REAL L(100,100),Lt(100,100),B(100),Y(100),X(100)
!ELEMENTS DE LA MATRICE L
L(1,1)=SQRT(A(1,1))
!ELEMENTS HORS DIAGONAUX DE LA MATRICE L
DO I=2,N
DO J=1,I-1
DO K=1,J-1
COEFF=COEFF+(L(I,K)*L(J,K))
ENDDO
L(I,J)=(A(I,J)-COEFF)/L(J,J)
ENDDO
!ELEMENTS DIAGONAUX DE LA MATRICE L
DO K=1,I-1
COEFF1=COEFF1+(L(I,K)*L(I,K))
ENDDO
L(I,I)=SQRT(A(I,I)-COEFF1)
COEFF1=0.0
ENDDO
!DETERMINATION DES ELEMENTS DE LA MATRICE Lt
DO I=1,N
DO J=1,N
Lt(J,I)=L(I,J)
ENDDO
ENDDO
!RESOUDRE LY=B PAR SUBSTITUTION AVANT
Y(1)=B(1)/L(1,1)
DO I=2,N
Y(I)=B(I)
DO J=1,I-1
Y(I)=Y(I)-L(I,J)*Y(J)
ENDDO
Y(I)=Y(I)/L(I,I)
ENDDO
!RESOUDRE LtX=Y PAR SUBSTITUTION ARRIERE
X(N)=Y(N)/Lt(N,N)
DO I=N-1,1,-1
X(I)=Y(I)
DO J=N,I+1,-1
X(I)=X(I)-Lt(I,J)*X(J)
ENDDO
X(I)=X(I)/Lt(I,I)
ENDDO
RETURN
END
```

PROGRAMME N°96

```
PROGRAM DOOLITTLE_FACTORISATION_LU
REAL A(100,100)
REAL L(100,100),U(100,100),B(100),X(100)
PRINT *, 'DONNER ORDRE DU SYSTEME'; READ *, N
PRINT*
PRINT *, 'DONNER LES ELEMENTS DE LA MATRICE A'
DO I=1,N
DO J=1,N
PRINT *,'A(',I,',',J,')='; READ *, A(I,J)
ENDDO
ENDDO
PRINT*
PRINT *, 'DONNER LES ELEMENTS DU VECTEUR B'
DO I=1,N
PRINT *,'B(',I,')='; READ *, B(I)
ENDDO
CALL FAC_LU_X(A,B,N,L,U,X)
PRINT*, 'ELEMENTS DE LA MATRICE L'
DO I=1,N
DO J=1,N
PRINT *,'L(',I,',',J,')=',L(I,J)
ENDDO
ENDDO
PRINT*,'ELEMENTS DE LA MATRICE U'
DO I=1,N
DO J=1,N
PRINT *,'U(',I,',',J,')=',U(I,J)
ENDDO
ENDDO
PRINT*,'LA SOLUTION DU SYSTEME AX=B'
DO I=1,N
PRINT *,'X(',I,')=',X(I)
ENDDO
END
SUBROUTINE FAC_LU_X(A,B,N,L,U,X)
REAL A(100,100)
REAL L(100,100),U(100,100),B(100),Y(100),X(100)
!INITIALITATION DES MATRICES L ET U
L=0.0
U=0.0
!CALCUL DES COEFFICIENTS DE LA MATRICE L
DO K=1,N-1
DO I=K+1,N
COEFF=A(I,K)/A(K,K)
L(I,K)=COEFF
DO J=K+1,N
A(I,J)=A(I,J)-COEFF*A(K,J)
ENDDO
ENDDO
ENDDO
!ELEMENTS DIAGONAUX DE LA MATRICE L
DO I=1,N
L(I,I)=1.0
ENDDO
```

```
!DETERMINATION DES ELEMENTS DE LA MATRICE U
DO J=1,N
DO I=1,J
U(I,J)=A(I,J)
ENDDO
ENDDO
!RESOUDRE LY=B PAR SUBSTITUTION AVANT
Y(1)=B(1)
DO I=2,N
Y(I)=B(I)
DO J=1,I-1
Y(I)=Y(I)-L(I,J)*Y(J)
ENDDO
ENDDO
!RESOUDRE UX=Y PAR SUBSTITUTION ARRIERE
X(N)=Y(N)/U(N,N)
DO I=N-1,1,-1
X(I)=Y(I)
DO J=N,I+1,-1
X(I)=X(I)-U(I,J)*X(J)
ENDDO
X(I)=X(I)/U(I,I)
ENDDO
RETURN
END
```

PROGRAMME N°97

```
PROGRAM VIBRATIONS_LIBRES_AMORTIES
! Programme qui calcule la réponse libre d'un système masse-ressort-amortisseur amorti à 1 d.d.l.
REAL M,K
PRINT*, 'DONNEZ S.V.P.'
PRINT*, 'LA MASSE DU SYSTÈME M=';READ*, M
PRINT*, 'LE COEF D"AMORTISSEMENT DU SYSTÈME C=';READ*, C
PRINT*, 'LA RAIDEUR DU SYSTÈME K='; READ*, K
PRINT*, 'LA CONDITION INITIALE EN DÉPLACEMENT X0=';READ*, X0
PRINT*, 'LA CONDITION INITIALE EN VITESSE V0=';READ*, V0
PRINT*, 'LE TEMPS D"OBSERVATION TF=';READ*, TF
OPEN(1,FILE='RÉPONSE.DAT') !FICHIER DES RÉSULTATS
W=SQRT(K/M) !PULSATION NATURELLE
Z=C/(2*M*W) !CALCUL DU RAPPORT D'AMORTISSEMENT
WRITE(1,*) 'LA PULSATION NATURELLE DU SYSTÈME EN RAD/S W=',W
WRITE(1,*) 'LE RAPOPORT D AMORTISSEENT EST Z=',Z
IF(Z<1) THEN
WD=W*SQRT(1-Z**2) !PULSATION NATURELLE AMORTIE
WRITE(1,*) 'LA PULSATION NATURELLE AMORTIE EST WD=',WD
A=SQRT(((V0+Z*W*X0)**2+(X0*WD)**2)/WD**2)!AMPLITUDE DE L'EXPONENTIELLE
PHI=ATAN2(X0*WD,V0+Z*W*X0) !PHASE
DO T=0,TF,0.1
X=A*EXP(-Z*W*T)*SIN(WD*T+PHI) !RÉPONSE SOUS AMORTIE
WRITE(1,1) T,X
1 FORMAT(1X,'T=',F6.3,12X,'X=',F6.3)
ENDDO
ELSE IF(Z==1) THEN
A1=X0
A2=V0+W*X0
```

```
WRITE(1,*)  'A1=',A1
WRITE(1,*)  'A2=',A2
DO  T=0,TF,0.1
X=(A1+A2*T)*EXP(-W*T)  !RÉPONSE  CRITIQUE
WRITE(1,2)  T,X
2  FORMAT(1X,'T=',F6.3,12X,'X=',F6.3)
ENDDO
ELSE
A1=(V0+(Z+SQRT(Z**2-1))*W*X0)/(2*W*SQRT(Z**2-1))
A2=(-V0-(Z-SQRT(Z**2-1))*W*X0)/(2*W*SQRT(Z**2-1))
WRITE(1,*)  'A1=',A1
WRITE(1,*)  'A2=',A2
DO  T=0,TF,0.1
X=EXP(-Z*W*T)*(A1*EXP(W*SQRT(Z**2-1)*T)+A2*EXP(-W*SQRT(Z**2-1)*T))!RÉPONSE
!SUR  AMORTIE
WRITE(1,3)  T,X
3  FORMAT(1X,'T=',F6.3,12X,'X=',F6.3)
ENDDO
ENDIF
END
```

PROGRAMME N°98

```
PROGRAM RUTHIS     ! Calcul des fréquences propres et modes propres. Cas libre-libre.
REAL,  PARAMETER::  EPSI=1.0E-05,  PI=3.14159
DIMENSION    INERT(50),RAID(50),C(50),D(50),ALPHA(50),OM(50),TM(50),X(50,50)
REAL  INERT
OPEN(1,FILE='DONNE.DAT',STATUS='OLD')
OPEN(2,FILE='RES.DAT')
!  LECTURE  DES  DONNÉES
PRINT*,  'DONNER  LE  NOMBRE  DE  DISQUES  N'
READ*,  N
K=N-1
!  LECTURE  DES  INERTIES  DES  ARBRES  À  PARTIR  DU  FICHIER  DONNE.DAT
DO  I=1,N
READ(1,*)  INERT(I)
END  DO
!  LECTURE  DES  RIGIDITÉS  DES  ARBRES  À  PARTIR  DU  FICHIER  DONNE.DAT
DO  I=1,K
READ  (1,*)  RAID(I)
END  DO
!  CALCUL  DES  ÉLÉMENTS  C  ET  ALPHA  DE  LA  MATRICE  C°
C(1)=RAID(1)/INERT(1)
D(1)=C(1)
DO  I=2,K
J=I-1
C(I)=(RAID(J)+RAID(I))/INERT(I)
D(I)=C(I)
ALPHA(I)=(RAID(J))**2/(INERT(I)*INERT(J))
ENDDO
C(N)=RAID(K)/INERT(N)
D(N)=C(N)
ALPHA(N)=RAID(K)**2/(INERT(N)*INERT(K))
!  CALCUL  DES  ÉLÉMENTS  C  ET  ALPHA  DE  LA  MATRICE  C
DO  NB=1,100
A=C(1)
```

```
DO  I=1,K
B=ALPHA(I+1)/A
C(I)=A+B
A=C(I+1)-B
ALPHA(I+1)=A*B
ENDDO
C(N)=A
! TEST DE CONVERGENCE ET D'ARRÊT
S1=0.
S2=0.
DO  L=1,N
S1=S1+ABS(C(L)-D(L))
S2=S2+C(L)
D(L)=C(L)
ENDDO
Q=S1/S2
IF((Q-EPSI).LE.0)  THEN
PRINT*,  'TEST  ARRET  A  :',NB,'ITERATIONS'
PRINT*
GOTO  1
ENDIF
ENDDO
!CALCUL  ET  AFFICHAGE  DES  FRÉQUENCES  PRORES
1  DO  J=1,N
OM(J)=SQRT(ABS(C(J)))
TM(J)=(OM(J)*30)/PI
ENDDO
!CALCUL  ET  AFFICHAGE  DES  DÉFORMÉES  MODALES
DO  I=1,N
J=N-I
X(1,I)=1
X(2,I)=1-(C(J)*(INERT(1)/RAID(1)))
DO  L=2,K
PRINT*,'INERT(',L,')',INERT(L),'RAID(',L,')',RAID(L)
X(L+1,I)=(INERT(L)/RAID(L))*(((((RAID(L-1)+RAID(L))/INERT(L))-C(J))*X(L,I))-(RAID(L-1)*X(L-1,I))/INERT(L))
ENDDO
ENDDO
DO  I=1,N
II=N-I
PRINT*
WRITE(2,*)  'MODE',I  ;  WRITE(2,*)'OMEGA(',I,')=',OM(II),'','RAD/S'
PRINT*
WRITE(2,3)  'DISQUE','AMPLITUDE  RELATIVE'
3  FORMAT(1X,A8,6X,A20)
PRINT*
DO  J=1,N
WRITE(2,4)  J,X(J,I)
4  FORMAT(4X,I3,10X,F15.5)
ENDDO
ENDDO
END
```

PROGRAMME N°99

```
PROGRAM RUTHIS  ! Calcul des fréquences propres et modes propres. Cas encastré-libre REAL, PARAMETER:: EPSI=1.0E-05, PI=3.14159
DIMENSION INERT(50),RAID(50),C(50),D(50),ALPHA(50),OM(50),TM(50),X(50,50)
REAL INERT
OPEN(1,FILE='DONNEE1.DAT',STATUS='OLD')
OPEN(2,FILE='RES.DAT')
! LECTURE DES DONNÉES
PRINT*, 'DONNER LE NOMBRE DE DISQUES N'
READ*, N
! LECTURE DES INERTIES DES ARBRES À PARTIR DU FICHIER DONNEE1.DAT
DO I=1,N
READ(1,*) INERT(I)
END DO
! LECTURE DES RIGIDITÉS DES ARBRES À PARTIR DU FICHIER DONNEE1.DAT
DO I=1,N
READ(1,*) RAID(I)
END DO
! CALCUL DES ÉLÉMENTS C ET ALPHA DE LA MATRICE C°
DO I=1,N-1
J=I+1
C(I)=(RAID(J)+RAID(I))/INERT(I)
D(I)=C(I)
ALPHA(J)=(RAID(J))**2/(INERT(I)*INERT(J))
END DO
C(N)=RAID(N)/INERT(N)
D(N)=C(N)
ALPHA(N)=RAID(N)**2/(INERT(N)*INERT(N-1))
! CALCUL DES ÉLÉMENTS C ET ALPHA DE LA MATRICE C
DO NB=1,100
A=C(1)
DO I=1,N-1
B=ALPHA(I+1)/A
C(I)=A+B
A=C(I+1)-B
ALPHA(I+1)=A*B
ENDDO
C(N)=A
! TEST DE CONVERGENCE ET D'ARRÊT
S1=0.
S2=0.
DO L=1,N
S1=S1+ABS(C(L)-D(L))
S2=S2+C(L)
D(L)=C(L)
ENDDO
Q=S1/S2
IF((Q-EPSI).LE.0) THEN
PRINT*, 'TEST ARRET A :',NB,'ITERATIONS'
PRINT*
GOTO 1
ENDIF
ENDDO
```

```
! CALCUL ET AFFICHAGE DES FRÉQUENCES PRORES
1 DO J=1,N
OM(J)=SQRT(ABS(C(J)))
TM(J)=(OM(J)*30)/PI
ENDDO
! CALCUL ET AFFICHAGE DES DÉFORMÉES MODALES
DO I=1,3
J=N+1-I
X(1,I)=1
X(2,I)=1+(RAID(1)/RAID(2))-(C(J)*(INERT(1)/RAID(2)))
DO L=2,N-1
X(L+1,I)=(INERT(L)/RAID(L+1))*(((((RAID(L)+RAID(L+1))/INERT(L))-C(J))*X(L,I))-(RAID(L)*X(L-1,I))/INERT(L))
ENDDO
ENDDO
DO I=1,3
II=N+1-I
WRITE(2,*) 'MODE',I ; WRITE(2,*)'OMEGA(',I,')=',OM(II),",'RAD/S'
PRINT*
WRITE(2,3) 'DISQUE','AMPLITUDE RELATIVE'
3 FORMAT(1X,A8,6X,A20)
PRINT*
DO J=1,N
WRITE(2,4) J,X(J,I)
4 FORMAT(4X,I3,10X,F12.5)
ENDDO
ENDDO
END
```

PROGRAMME N°100

```
PROGRAM RUTHIS ! Calcul des fréquences propres et modes propres. Cas encastré- encastré
REAL, PARAMETER:: EPSI=1.0E-05, PI=3.14159
DIMENSION INERT(50),RAID(50),C(50),D(50),ALPHA(50),OM(50),TM(50),X(50,50)
REAL INERT
OPEN(1,FILE='DONNEE2.DAT',STATUS='OLD')
OPEN(2,FILE='RES.DAT')
! LECTURE DES DONNÉES
PRINT*, 'DONNER LE NOMBRE DE DISQUES N'
READ*, N
K=N+1
! LECTURE DES INERTIES DES ARBRES À PARTIR DU FICHIER DONNEE2.DAT
DO I=1,N
READ(1,*) INERT(I)
END DO
! LECTURE DES RIGIDITÉS DES ARBRES À PARTIR DU FICHIER DONNEE2.DAT
DO I=1,K
READ(1,*) RAID(I)
END DO
! CALCUL DES ÉLÉMENTS C ET ALPHA DE LA MATRICE C°
DO I=1,N-1
J=I+1
C(I)=(RAID(J)+RAID(I))/INERT(I)
D(I)=C(I)
ALPHA(J)=(RAID(J))**2/(INERT(I)*INERT(J))
ENDDO
```

```
C(N)=(RAID(N)+RAID(K))/INERT(N)
D(N)=C(N)
! CALCUL DES ÉLÉMENTS C ET ALPHA DE LA MATRICE C
DO NB=1,100
A=C(1)
DO I=1,N-1
B=ALPHA(I+1)/A
C(I)=A+B
A=C(I+1)-B
ALPHA(I+1)=A*B
ENDDO
C(N)=A
! TEST DE CONVERGENCE ET D'ARRÊT
S1=0.
S2=0.
DO L=1,N
S1=S1+ABS(C(L)-D(L))
S2=S2+C(L)
D(L)=C(L)
ENDDO
Q=S1/S2
IF((Q-EPSI).LE.0) THEN
PRINT*, 'TEST ARRET A :',NB,'ITERATIONS'
PRINT*
GOTO 1
ENDIF
ENDDO
! CALCUL ET AFFICHAGE DES FRÉQUENCES PRORES
1 DO J=1,N
OM(J)=SQRT(ABS(C(J)))
TM(J)=(OM(J)*30)/PI
ENDDO
! CALCUL ET AFFICHAGE DES DÉFORMÉES MODALES
DO I=1,3
J=K-I
X(1,I)=1
X(2,I)=1+(RAID(1)/RAID(2))-(C(J)*(INERT(1)/RAID(2)))
DO L=2,N-1
X(L+1,I)=(INERT(L)/RAID(L+1))*(((((RAID(L)+RAID(L+1))/INERT(L))-C(J))*X(L,I))-(RAID(L)*X(L-1,I))/INERT(L))
ENDDO
ENDDO
DO I=1,3
II=K-I
WRITE(2,*) 'MODE',I ; WRITE(2,*)'OMEGA(',I,')=',OM(II),",'RAD/S'
PRINT*
WRITE(2,3) 'DISQUE','AMPLITUDE RELATIVE'
3 FORMAT(1X,A8,6X,A20)
PRINT*
DO J=1,N
WRITE(2,4) J,X(J,I)
4 FORMAT(4X,I3,10X,F12.5)
ENDDO
ENDDO
END
```

Autres programmes

PROGRAMME N°P1

```
PROGRAM ABSENCES
REAL NOTE
PARAMETER (ABS1=14)
PRINT*,'DONNER LE NOMBRE DES ABSENCES ABS'
READ*,ABS
IF(ABS>=5.AND.ABS<7) THEN
NOTE=ABS1-2*ABS
PRINT*,'NOTE ABSENCES EST',NOTE
ELSE
IF(ABS>=7) THEN
NOTE=0.
PRINT*,'NOTE ABSENCES EST',NOTE
ELSE
NOTE=ABS1-ABS
PRINT*,'NOTE ABSENCES EST',NOTE
END IF
END IF
END
```

PROGRAMME N°P2

```
PROGRAM CLASSIFICATION_ANGLE_ALPHA
REAL ALPHA
PRINT*, 'DONNER ANGLE ALPHA EN DEGRE'
READ *, ALPHA
IF(ALPHA>=0.AND.ALPHA<90)THEN
PRINT*,'ALPHA EST INCLUS DANS LE PREMIER INTERVALLE'
ELSE
IF(ALPHA>=90.AND.ALPHA<180)THEN
PRINT*,'ALPHA EST INCLUS DANS LE SECOND INTERVALLE'
ELSE
IF(ALPHA>=180.AND.ALPHA<270)THEN
PRINT*,'ALPHA EST INCLUS DANS LE TROISIEME INTERVALLE'
ELSE
PRINT*,'ALPHA EST INCLUS DANS LE QUATRIEME INTERVALLE'
ENDIF
ENDIF
ENDIF
END
```

PROGRAMME N°P3

```
PROGRAM CONDITIONS_COUPE_1
CHARACTER*15 MATERIAU
CHARACTER*15 OUTIL
CHARACTER*15 OPERATION
PARAMETER (PI=3.14159)
PRINT*,'DONNER LE DIAMETRE DE LA PIECE OU DE LA FRAISE [mm]'
READ*, D
PRINT*,'DONNER LE MATERIAU DE LA PIECE A USINER'
READ*,MATERIAU
PRINT*,'DONNER OUTIL UTILISE'
READ*,OUTIL
```

```
PRINT*,'DONNER OPERATION USINAGE'
READ*,OPERATION
IF(MATERIAU=='ACIER'.AND.OUTIL=='ARS') THEN
IF(OPERATION=='EBAUCHE') GOTO 1
IF(OPERATION=='FINITION') GOTO 3
ELSE
IF(MATERIAU=='ALUMINIUM'.AND.OUTIL=='ARS') THEN
IF(OPERATION=='EBAUCHE') GOTO 2
IF(OPERATION=='FINITION') GOTO 4
ELSE
IF(MATERIAU=='ACIER'.AND.OUTIL=='CARBURE') THEN
IF(OPERATION=='EBAUCHE') GOTO 11
IF(OPERATION=='FINITION') GOTO 13
ELSE
IF(MATERIAU=='ALUMINIUM'.AND.OUTIL=='CARBURE') THEN
IF(OPERATION=='EBAUCHE') GOTO 12
IF(OPERATION=='FINITION') GOTO 14
ENDIF
ENDIF
ENDIF
ENDIF
1 PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 10<VC<30 m/min'
GOTO 5
2 PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 25<VC<45 m/min'
GOTO 5
3 PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 20<VC<50 m/min'
GOTO 5
4 PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 30<VC<60 m/min'
GOTO 5
11 PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 60<VC<150 m/min'
GOTO 5
12 PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 100<VC<180 m/min'
GOTO 5
13 PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 120<VC<220 m/min'
GOTO 5
14 PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 140<VC<260 m/min'
PRINT*
5 PRINT*,'DONNER LA VALEUR DE LA VITESSE DE COUPE VC EN m/min'
READ*, VC
N=(1000*VC)/(PI*D)
PRINT*,'LA FREQUENCE DE ROTATION DE LA BROCHE N',N,'[tr/min]'
END
```

PROGRAMME N°P4

```
PROGRAM CONDITIONS_COUPE_2
! DECLARATIONS
CHARACTER(10) MATERIAU
CHARACTER*8 OUTIL
CHARACTER*8 OPERATION
PARAMETER (PI=3.14159)
!PARAMETRES D'ENTREE
PRINT*,'DONNER LE DIAMETRE DE LA PIECE OU DE LA FRAISE [mm]'
READ*, D
PRINT*,'DONNER LE MATERIAU DE LA PIECE A USINER'
READ*,MATERIAU
PRINT*,'DONNER OUTIL UTILISE'
```

```
READ*,OUTIL
PRINT*,'DONNER OPERATION USINAGE'
READ*,OPERATION
!APPEL DES SOUS PROGRAMMES
IF(MATERIAU=='ACIER'.AND.OUTIL=='ARS') GOTO 1
IF(MATERIAU=='ALUMINIUM'.AND.OUTIL=='ARS') GOTO 2
IF(MATERIAU=='ACIER'.AND.OUTIL=='CARBURE') GOTO 3
IF(MATERIAU=='ALUMINIUM'.AND.OUTIL=='CARBURE') GOTO 4
1 CALL ACIER_ARS(OPERATION,D,PI)
GOTO 5
2 CALL ALU_ARS(OPERATION,D,PI)
GOTO 5
3 CALL ACIER_CARBURE(OPERATION,D,PI)
GOTO 5
4 CALL ALU_CARBURE(OPERATION,D,PI)
5 END

! LES SOUS PROGRAMMES

SUBROUTINE ACIER_ARS(OPERATION,D,PI)
DIMENSION VC(50),N(50)
CHARACTER*8 OPERATION
REAL N
OPEN(1,FILE='PLAGE')
DO I=1,24
READ(1,*) VC(I)
ENDDO
IF(OPERATION=='EBAUCHE') THEN
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 10<VC<30 m/min '
DO I=1,3
N(I)=(1000*VC(I))/(PI*D)
PRINT*,'VC(',I,')=',VC(I),' [M/MIN] ===> ','N(',I,')=',N(I),' [tr/min]'
ENDDO
ELSE
IF(OPERATION=='FINITION')THEN
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 20<VC<50 m/min'
DO I=4,6
N(I)=(1000*VC(I))/(PI*D)
PRINT*,'VC(',I,')=',VC(I),' [M/MIN] ===> ','N(',I,')=',N(I),' [tr/min]'
ENDDO
ENDIF
ENDIF
RETURN
END

SUBROUTINE ALU_ARS(OPERATION,D,PI)
DIMENSION VC(50),N(50)
CHARACTER*8 OPERATION
REAL N
OPEN(1,FILE='PLAGE')
DO I=1,24
READ(1,*) VC(I)
ENDDO
IF(OPERATION=='EBAUCHE') THEN
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 25<VC<45 m/min'
DO I=7,9
N(I)=(1000*VC(I))/(PI*D)
```

```
PRINT*,'VC(',I,')=',VC(I),'  [M/MIN]  ===>  ','N(',I,')=',N(I),'  [tr/min]'
ENDDO

ELSE
IF(OPERATION=='FINITION')  THEN
PRINT*,'LA  PLAGE  DE  LA  VITESSE  DE  COUPE  EST  :  30<VC<60  m/min'
DO  I=10,12
N(I)=(1000*VC(I))/(PI*D)
PRINT*,'VC(',I,')=',VC(I),'  [M/MIN]  ===>  ','N(',I,')=',N(I),'  [tr/min]'
ENDDO
ENDIF
ENDIF
RETURN
END

SUBROUTINE  ACIER_CARBURE(OPERATION,D,PI)
DIMENSION  VC(50),N(50)
CHARACTER*8  OPERATION
REAL  N
OPEN(1,FILE='PLAGE')
DO  I=1,24
READ(1,*)  VC(I)
ENDDO

IF(OPERATION=='EBAUCHE')  THEN
PRINT*,'LA  PLAGE  DE  LA  VITESSE  DE  COUPE  EST  :  60<VC<150  m/min'
DO  I=13,15
N(I)=(1000*VC(I))/(PI*D)
PRINT*,'VC(',I,')=',VC(I),'  [M/MIN]  ===>  ','N(',I,')=',N(I),'  [tr/min]'
ENDDO
ELSE
IF(OPERATION=='FINITION')  THEN
PRINT*,'LA  PLAGE  DE  LA  VITESSE  DE  COUPE  EST  :  120<VC<220  m/min'
DO  I=16,18
N(I)=(1000*VC(I))/(PI*D)
PRINT*,'VC(',I,')=',VC(I),'  [M/MIN]  ===>  ','N(',I,')=',N(I),'  [tr/min]'
ENDDO
ENDIF
ENDIF
RETURN
END

SUBROUTINE  ALU_CARBURE(OPERATION,D,PI)
DIMENSION  VC(50),N(50)
CHARACTER*8  OPERATION
REAL  N
OPEN(1,FILE='PLAGE')
DO  I=1,24
READ(1,*)  VC(I)
ENDDO

IF(OPERATION=='EBAUCHE')  THEN
PRINT*,'LA  PLAGE  DE  LA  VITESSE  DE  COUPE  EST  :  100<VC<180  m/min'
DO  I=19,21
N(I)=(1000*VC(I))/(PI*D)
PRINT*,'VC(',I,')=',VC(I),'  [M/MIN]  ===>  ','N(',I,')=',N(I),'  [tr/min]'
ENDDO
```

```
ELSE
IF(OPERATION=='FINITION') THEN
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 140<VC<260 m/min'
DO I=22,24
N(I)=(1000*VC(I))/(PI*D)
PRINT*,'VC(',I,')=',VC(I),' [m/min] ===> ','N(',I,')=',N(I),' [tr/min]'
ENDDO
ENDIF
ENDIF
RETURN
END
```

PROGRAMME N°P5

```
PROGRAM CONDITIONS_COUPE_3
! DECLARATIONS
CHARACTER(10) MATERIAU
CHARACTER*8 OUTIL
CHARACTER*8 OPERATION
PARAMETER (PI=3.14159)
!PARAMETRES D'ENTREE
PRINT*,'DONNER LE DIAMETRE DE LA PIECE OU DE LA FRAISE [mm]'
READ*, D
PRINT*,'DONNER LE MATERIAU DE LA PIECE A USINER'
READ*,MATERIAU
PRINT*,'DONNER OUTIL UTILISE'
READ*,OUTIL
PRINT*,'DONNER OPERATION USINAGE'
READ*,OPERATION
!APPEL DES SOUS PROGRAMMES
IF(MATERIAU=='ACIER'.AND.OUTIL=='ARS') GOTO 1
IF(MATERIAU=='ALUMINIUM'.AND.OUTIL=='ARS') GOTO 2
IF(MATERIAU=='ACIER'.AND.OUTIL=='CARBURE') GOTO 3
IF(MATERIAU=='ALUMINIUM'.AND.OUTIL=='CARBURE') GOTO 4
1 CALL ACIER_ARS(OPERATION,D,PI)
GOTO 5
2 CALL ALU_ARS(OPERATION,D,PI)
GOTO 5
3 CALL ACIER_CARBURE(OPERATION,D,PI)
GOTO 5
4 CALL ALU_CARBURE(OPERATION,D,PI)
5 END

! LES SOUS PROGRAMMES

SUBROUTINE ACIER_ARS(OPERATION,D,PI)
CHARACTER*8 OPERATION
REAL N
OPEN(2,FILE='RESULT1')
IF(OPERATION=='EBAUCHE') THEN
WRITE(2,*)'LA PLAGE DE LA VITESSE DE COUPE EST : 10<VC<30 m/min'
DO VC=10,30,5
N=(1000*VC)/(PI*D)
WRITE(2,*)'VC=',VC,' [m/min] ===> ','N=',N,' [tr/min]'
ENDDO
ELSE
IF(OPERATION=='FINITION')THEN
```

```
PRINT*,'LA PLAGE DE LA VITESSE DE COUPE EST : 20<VC<50 m/min'
DO VC=20,50,5
N=(1000*VC)/(PI*D)
WRITE(2,*)'VC=',VC,' [m/min] ===> ','N=',N,' [tr/min]'
ENDDO
ENDIF
ENDIF
RETURN
END

SUBROUTINE ALU_ARS(OPERATION,D,PI)
CHARACTER*8 OPERATION
REAL N
OPEN(2,FILE='RESULT1')
IF(OPERATION=='EBAUCHE') THEN
WRITE(2,*)'LA PLAGE DE LA VITESSE DE COUPE EST : 25<VC<45 m/min '
DO VC=25,45,5
N=(1000*VC)/(PI*D)
WRITE(2,*)'VC=',VC,' [m/min] ===> ','N=',N,' [tr/min]'
ENDDO
ELSE
IF(OPERATION=='FINITION') THEN
WRITE(2,*)'LA PLAGE DE LA VITESSE DE COUPE EST : 30<VC<60 m/min'
DO VC=30,60,5
N=(1000*VC)/(PI*D)
WRITE(2,*)'VC=',VC,' [m/min] ===> ','N=',N,' [tr/min]'
ENDDO
ENDIF
ENDIF
RETURN
END

SUBROUTINE ACIER_CARBURE(OPERATION,D,PI)
CHARACTER*8 OPERATION
REAL N
OPEN(2,FILE='RESULT1')
IF(OPERATION=='EBAUCHE') THEN
WRITE(2,*)'LA PLAGE DE LA VITESSE DE COUPE EST : 60<VC<150 m/min'
DO VC=60,150,5
N=(1000*VC)/(PI*D)
WRITE(2,*)'VC=',VC,' [m/min] ===> ','N=',N,' [tr/min]'
ENDDO
ELSE
IF(OPERATION=='FINITION') THEN
WRITE(2,*)'LA PLAGE DE LA VITESSE DE COUPE EST : 120<VC<220 m/min'
DO VC=120,220,5
N=(1000*VC)/(PI*D)
WRITE(2,*)'VC=',VC,' [m/min] ===> ','N=',N,' [tr/min]'
ENDDO
ENDIF
ENDIF
RETURN
END

SUBROUTINE ALU_CARBURE(OPERATION,D,PI)
CHARACTER*8 OPERATION
REAL N
```

```
OPEN(2,FILE='RESULT1')
IF(OPERATION=='EBAUCHE') THEN
WRITE(2,*)'LA PLAGE DE LA VITESSE DE COUPE EST : 100<VC<180 m/min'
DO VC=100,180,5
N=(1000*VC)/(PI*D)
WRITE(2,*)'VC=',VC,' [m/min] ===> ','N=',N,' [tr/min]'
ENDDO
ELSE
IF(OPERATION=='FINITION') THEN
WRITE(2,*)'LA PLAGE DE LA VITESSE DE COUPE EST : 140<VC<260 m/min'
DO VC=140,260,5
N=(1000*VC)/(PI*D)
WRITE(2,*)'VC=',VC,' [m/min] ===> ','N=',N,' [tr/min]'
ENDDO
ENDIF
ENDIF
RETURN
END
```

PROGRAMME N°P6

```
PROGRAM DEMANDE_NOMBRE
INTEGER N
PRINT*,'DONNER N'
DO WHILE (N<25.OR.N>50)
READ*,N
IF(N<25) THEN
PRINT*,'PLUS GRAND'
ELSE
IF(N>50) THEN
PRINT*,'PLUS PETIT'
END IF
ENDIF
ENDDO
PRINT*,'NOMBRE INCLU DANS 25<N<50',N
END
```

PROGRAMME N°P7

```
PROGRAM SCHTROUMPF
DIMENSION T1(50),T2(50)
SCH=0.
PRINT*,'ENTREZ LA DIMENSION DU TABLEAU T1'
READ*, N
PRINT*,'ENTREZ LES ELEMENTS DU TABLEAU T1'
DO I=1,N
READ*,T1(I)
ENDDO
PRINT*,'ENTREZ LA DIMENSION DU TABLEAU T2'
READ*, M
PRINT*,'ENTREZ LES ELEMENTS DU TABLEAU T2'
DO I=1,M
READ*,T2(I)
ENDDO
DO J=1,M
DO I=1,N
SCH=SCH+T2(J)*T1(I)
```

```
ENDDO
ENDDO
PRINT*,'LE SCHTROUMPF SERA :',SCH
END
```

PROGRAMME N°P8

```
PROGRAM MATRICE
INTEGER MAXLIG,MAXCOL
PARAMETER(MAXLIG=5,MAXCOL=3)
REAL M(MAXLIG,MAXCOL)
INTEGER IL,IC
DO IL=1,MAXLIG
DO IC=1,MAXCOL
M(IL,IC)=IL+(IC/10.0)
ENDDO
ENDDO
CALL AFFMAT(M,MAXLIG,MAXCOL)
END

SUBROUTINE AFFMAT(M,DIML,DIMC)
IMPLICIT NONE
INTEGER DIML,DIMC
REAL M(DIML,DIMC)
INTEGER IL,IC
DO IL=1,DIML
PRINT *,('M(',IL,IC,')=',M(IL,IC),IC=1,DIMC)
ENDDO
END
```

PROGRAMME N°P9

```
PROGRAM MOYENNE_OBSERVATION
DIMENSION NOM(50)
REAL N1,N2,N3,MOY
CHARACTER*10 OBSERVATION
CHARACTER*15 NOM
OPEN(2,FILE='LISTE',STATUS='OLD') !FICHIER LISTE DES ETUDIANTS
OPEN (1,FILE='RESULT') !FICHIER DES RESULTATS
PRINT*,'DONNER LE NOMBRE D''ETUDIANTS'
READ*, N
WRITE(1,10) 'NOM','N1','N2','N3','MOYENNE','RESULTAT'
10 FORMAT(/,1X,A3,5X,A2,7X,A5,7X,A5,5X,A8,5X,A9,/)
DO I=1,N
READ(2,*) NOM(I)
PRINT*,'DONNER LES NOTES DE L''ETUDIANT :',NOM(I)
PRINT*,'N1=';READ*, N1
PRINT*,'N2=';READ*, N2
PRINT*,'N3=';READ*, N3
MOY=(N1+N2+N3)/3
IF(MOY<10) THEN
OBSERVATION='AJOURNE(E)'
ELSE
OBSERVATION='ADMIS(E)'
ENDIF
WRITE(1,12) NOM(I),N1,N2,N3,MOY,OBSERVATION
12 FORMAT(A8,4X,F5.2,6X,F5.2,6X,F5.2,4X,F5.2,6X,A10)
```

```
ENDDO
END
```

PROGRAMME N°P10

```
PROGRAM TRINOME
IMPLICIT NONE
REAL, PARAMETER :: EPSILON = 1E-6
REAL A, B, C
REAL DELTA, R_DELTA, X1, X2
! VALORISATION DES COEFFICIENTS.
A = 3.; B = 7.; C = -11.
! A DOIT ETRE NON NUL.
IF ( A > -EPSILON .AND. A < EPSILON ) &
STOP "A DOIT ETRE NON NUL."
! CALCUL DU DETERMINANT.
DELTA = B*B - 4*A*C
! CAS DU DETERMINANT NEGATIF.
IF( DELTA < -EPSILON ) STOP "PAS DE RACINE REELLE."
! CAS DU DETERMINANT NUL.
IF ( DELTA > -EPSILON .AND. DELTA < EPSILON ) THEN
X1 = -B/(2*A); X2 = X1
ELSE
! CAS DU DETERMINANT POSITIF.
R_DELTA = SQRT( DELTA )
X1 = (-B - R_DELTA)/(2*A); X2 = (-B + R_DELTA)/(2*A)
END IF
! IMPRESSION DES RACINES.
PRINT *,"X1 = ", X1, ", X2 = ", X2
END PROGRAM TRINOME
```

PROGRAMME N°P11

```
PROGRAM MOYENNE_CLASSE
REAL SOM,MOY
DIMENSION T(50)
SOM=0.
NBSUP=0.
PRINT*,'ENTREZ LE NOMBRE DE NOTES A SAISIR : '
READ*,NB
PRINT*,'ENTREZ CES NOTES : '
DO I=1,NB
READ*, T(I)
ENDDO
DO I=1,NB
SOM=SOM + T(I)
ENDDO
MOY=SOM/NB
PRINT*,'LA MOYENNE DE LA CLASSE EST :',MOY
DO I=1,NB
IF(T(I)>MOY)THEN
NBSUP=NBSUP+1
ENDIF
ENDDO
PRINT*, NBSUP,' ','ELEVES DEPASSENT LA MOYENNE DE LA CLASSE'
END
```

PROGRAMME N°P12

```
PROGRAM  MOYENNE
REAL  K,N1,N2,M
CHARACTER*2  REPONSE
PRINT*,'DONNER  LE  NOMBRE  D"ETUDIANTS'
READ*,K
DO  I=1,K
PRINT*,'DONNER  LES  NOTES  DE  L"ETUDIANT',I
READ*,  N1,N2
M=(N1+N2)/2
PRINT*,'LA  MOYENNE  EST  EGALE  :',M
PRINT*,'VOULEZ  VOUS  RECOMMENCER  [O/N]'
READ*,  REPONSE
IF(REPONSE=='N')  GOTO  1
ENDDO
1  END
```

PROGRAMME N°P13

```
PROGRAM  PROG
IMPLICIT  NONE
REAL,  DIMENSION(5)  ::  TAB
REAL  ::  SOMME
INTEGER  ::  I
TAB=(/  (I*10,I=1,5)  /)
PRINT  *,  TAB
CALL  SP1S(SOMME,TAB)
PRINT  *,SOMME
CALL  SP2S(SOMME)
PRINT  *,SOMME
END  PROGRAM  PROG
SUBROUTINE  SP1S(SOM,TAB)
IMPLICIT  NONE
REAL,  DIMENSION(5)  ::  TAB
REAL  ::  SOM
INTEGER  ::  I
SOM=0.
DO  I=1,5
SOM=SOM+TAB(I)
ENDDO
END  SUBROUTINE  SP1S
SUBROUTINE  SP2S(X)
IMPLICIT  NONE
REAL  ::  X
X=-X
END  SUBROUTINE  SP2S
```

PROGRAMME N°P14

```
PROGRAM  PG_PGI
INTEGER  PG,  IPG
N=1
I=0
PG=0
DO  WHILE  (N.NE.0)
PRINT*,'  ENTREZ  UN  NOMBRE  :'
```

```
READ*, N
I=I+1
IF(I==1.OR.N>PG) THEN
PG=N
IPG=I
ENDIF
ENDDO
PRINT*,'LE NOMBRE LE PLUS GRAND ETAIT : ', PG
PRINT*,'IL A ETE SAISI EN POSITION NUMERO ', IPG
END
```

PROGRAMME N°P15

```
PROGRAM TRIABULLE
IMPLICIT NONE
INTEGER, PARAMETER :: CROISSANT=1, DECROISSANT=2, N=10
REAL, DIMENSION(N) :: TAB
REAL :: TEMP
LOGICAL :: TRI_TERMINE, EXPR1, EXPR2
INTEGER :: SENS, I
VALORISATION DU VECTEUR.
DATA TAB/0.76, 0.38, 0.42, 0.91, 0.25, &
            0.13, 0.52, 0.69, 0.76, 0.98/
DO SENS=CROISSANT, DECROISSANT                    ! SENS DU TRI
DO TRI_TERMINE = .TRUE.                           ! TRI
DO I=2,N
EXPR1 = SENS == CROISSANT     .AND. TAB(I-1) > TAB(I)
EXPR2 = SENS == DECROISSANT .AND. TAB(I-1) < TAB(I)
IF (EXPR1 .OR. EXPR2) THEN
TRI_TERMINE = .FALSE.
TEMP = TAB(I-1); TAB(I-1) = TAB(I); TAB(I) = TEMP
END IF
END DO
IF (TRI_TERMINE) EXIT
END DO
! IMPRESSION DU VECTEUR TRIE.
IF (SENS == CROISSANT)     PRINT*, "TRI CROISSANT "
IF (SENS == DECROISSANT) PRINT*, "TRI DECROISSANT "
PRINT*, TAB
END DO
END PROGRAM TRIABULLE
```

PROGRAMME N°P16

```
PROGRAM REPRESENTATION_POLYNOMES
DIMENSION A(50),P(50)
PRINT*,'DONNER LE DERGRE DU POLYNOME'
READ*, N
PRINT*,'LE NOMBRE DES COEFFICIENTS DU POLYNOME EST --->', N+1
!LECTURE DES COEFFICIENTS
PRINT*,'DONNER CES COEFFICIENTS'
DO I=N,0,-1
PRINT*,'A(',I,')'; READ*,A(I)
ENDDO
!AFFICHAGE DES MONÔMES
PRINT*,'P(X)=',(A(I),'.','X**',",I,",'+', I=N,0,-1)
CALL POLY(X,N,P)
```

```
DO I=N,0,-1
Y=Y+A(I)*P(I)
ENDDO
PRINT*,'P(',X,')=',Y
END
! CALCUL DES MONOMES
SUBROUTINE POLY(X,N,P)
DIMENSION P(50)
PRINT*,'DONNER LA VALEUR DE X'
READ*,X
DO I=N,0,-1
P(I)=X**I
ENDDO
RETURN
END
```

PROGRAMME N°P17

```
PROGRAM AFFICHAGE_PRENOMS
CHARACTER*2 LETTRE
PRINT*,'DONNER LA PREMIERE LETTRE EN MAJUSCULE'
READ*,LETTRE
IF(LETTRE=='F')THEN
PRINT*,'FATIMA'
ELSE
IF(LETTRE=='B')THEN
PRINT*,'BRAHIM'
ELSE
IF(LETTRE=='N')THEN
PRINT*,'NACERA'
ELSE
IF(LETTRE=='K')THEN
PRINT*,'KARIM'
ELSE
PRINT*,'LE PRENOM EST INEXIXTANT'
ENDIF
ENDIF
ENDIF
ENDIF
END
```

PROGRAMME N°P18

```
PROGRAM CALCUL_PANNES_Cas_1
REAL L2,L1,L,IX,IY
OPEN(4,FILE='DB.DON')
OPEN(3,FILE='PROJET.DAT')
READ(4,*) CP,CS,L1,L,Z,Q,E
WRITE(3,*) 'CHARGES PERMAMENTES CP [DAN/M**2] =',CP
WRITE(3,*) 'SURCHARGES CS [DAN/M**2] =',CS
WRITE(3,*) 'PORTEE DE LA PANNE L1 [M] =',L1
WRITE(3,*) 'ENTRAXE L [M] =',L
WRITE(3,*) 'PENTE AU VERSANT Z [%] =',Z
WRITE(3,*) 'LIMITE ELASTIQUE Q [DAN/MM**2]=',Q
WRITE(3,*) 'MODULE DE YOUNG E [DAN/CM**2]=',E
P=((CP*4/3.)+(CS*3/2.))*L
Z=ATAND(Z)
```

```
PY=P*COS(Z)
PX=P*SIN(Z)
PN=(CP+CS)*L
PXX=PN*SIN(Z)
PYY=PN*COS(Z)
L2=L1*100
! PANNE SUR APPUIS SIMPLES SANS LIERNES
OPEN(1,FILE='IN.DON')
DO N=1,20
READ(1,*) N,IPN,IX,IY,WX,WY
QMAX=(PX*L1**2)/(8.*WY)+(PY*L1**2)/(8.*WX)
IF(QMAX.LE.Q) GOTO 431
ENDDO
431     FX=(0.013*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200) GOTO 72
DO M=N+1,20
READ(1,*) M,IPN,IX,IY,WX,WY
FX=(0.013*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200) GOTO 72
ENDDO
72 WRITE(3,1)
1 FORMAT(3X,'PANNES SUR 2 APPUIS SIMPLES SANS
LIERNES',/,3X,'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@',/)
WRITE(3,50)IPN
50 FORMAT(3X,'IPN FINAL ===>',1X,I3,/)
CLOSE (1)
OPEN(2,FILE='IE.DON')
DO N=1,18
READ(2,*) N,IPE,IX,IY,WX,WY
QMAX=(PX*L1**2)/(8.*WY)+(PY*L1**2)/(8.*WX)
IF(QMAX.LE.Q) GOTO 331
ENDDO
331     FX=(0.013*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200) GOTO 71
DO M=N+1,18
READ(2,*) M,IPE,IX,IY,WX,WY
FX=(0.013*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200) GOTO 71
ENDDO
71      WRITE(3,51) IPE
51      FORMAT(3X,'IPE FINAL ===>',1X,I3,//)
CLOSE (2)
! PANNE SUR 3 APPUIS
OPEN(1,FILE='IN.DON')
DO N=1,20
READ(1,*) N,IPN,IX,IY,WX,WY
QMAX=(PX*L1**2)/(8.*WY)+(PY*L1**2)/(8.*WX)
IF(QMAX.LE.Q)GOTO 16
ENDDO
```

```
16        FX=(0.0052*PXX*L2**4)/(100*E*IY)
FY=(0.0052*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200)  GOTO  19
DO  M=N+1,20
READ(1,*)  M,IPN,IX,IY,WX,WY
FX=(0.0052*PXX*L2**4)/(100*E*IY)
FY=(0.0052*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200)  GOTO  19
ENDDO
19        WRITE(3,2)
2  FORMAT(3X,'PANNES  SUR  3  APPUIS  SIMPLES  SANS
LIERNES',/,3X,'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@',/)
          WRITE(3,52)IPN
52  FORMAT(3X,'IPN  FINAL  ===>',1X,I3,/)
CLOSE  (1)
OPEN(2,FILE='IE.DON')
DO  N=1,18
READ(2,*)  N,IPE,IX,IY,WX,WY
QMAX=(PX*L1**2)/(8.*WY)+(PY*L1**2)/(8.*WX)
IF(QMAX.LE.Q)GOTO  160
ENDDO
160    FX=(0.0052*PXX*L2**4)/(100*E*IY)
FY=(0.0052*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200)  GOTO  21
DO  M=N+1,18
READ(2,*)  M,IPE,IX,IY,WX,WY
FX=(0.0052*PXX*L2**4)/(100*E*IY)
FY=(0.0052*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200)  GOTO  21
ENDDO
21        WRITE(3,53)IPE
53        FORMAT(3X,'IPE  FINAL  ===>',1X,I3,//)
CLOSE  (2)
STOP
END
```

PROGRAMME N°P19

```
PROGRAM  CALCUL_PANNES_Cas_2
REAL  L2,L1,L,IX,IY
OPEN(4,FILE='DB.DON')
OPEN(3,FILE='PROJET.DAT')
READ(4,*)  CP,CS,L1,L,Z,Q,E
WRITE(3,*)  'CHARGES  PERMAMENTES  CP  [DAN/M**2]  =',CP
WRITE(3,*)  'SURCHARGES                CS  [DAN/M**2]  =',CS
WRITE(3,*)  'PORTEE  DE  LA  PANNE    L1  [M]            =',L1
WRITE(3,*)  'ENTRAXE                        L    [M]            =',L
WRITE(3,*)  'PENTE  AU  VERSANT       Z    [%]            =',Z
WRITE(3,*)  'LIMITE  ELASTIQUE        Q    [DAN/MM**2]=',Q
WRITE(3,*)  'MODULE  DE  YOUNG          E    [DAN/CM**2]=',E
P=((CP*4/3.)+(CS*3/2.))*L
Z=ATAND(Z)
PY=P*COS(Z)
```

```
PX=P*SIN(Z)
PN=(CP+CS)*L
PXX=PN*SIN(Z)
PYY=PN*COS(Z)
L2=L1*100
! PANNE SUR 2 APPUIS AVEC LIERNES A MI-PORTEE
OPEN(1,FILE='IN.DON')
DO N=1,20
READ(1,*) N,IPN,IX,IY,WX,WY
QMAX=(PX*L1**2)/(32.*WY)+(PY*L1**2)/(8.*WX)
IF(QMAX.LE.Q) GOTO 36
ENDDO
36    FX=(0.00065*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200) GOTO 38
DO M=N+1,20
READ(1,*) M,IPN,IX,IY,WX,WY
FX=(0.00065*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200) GOTO 38
ENDDO
38    WRITE(3,3)
3     FORMAT(3X,'PANNES SUR 2 APPUIS AVEC LIERNES A MI-
PORTEE',/,3X,'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@',/)
WRITE(3,54)IPN
54 FORMAT(3X,'IPN FINAL ===>',1X,I3,/)
CLOSE (1)
OPEN(2,FILE='IE.DON')
DO N=1,18
READ(2,*) N,IPE,IX,IY,WX,WY
QMAX=(PX*L1**2)/(32.*WY)+(PY*L1**2)/(8.*WX)
IF(QMAX.LE.Q) GOTO 360
ENDDO
360   FX=(0.00065*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200) GOTO 76
DO M=N+1,18
READ (2,*) M,IPE,IX,IY,WX,WY
FX=(0.00065*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200) GOTO 76
ENDDO
76    WRITE(3,55)IPE
55    FORMAT(3X,'IPE FINAL ===>',1X,I3,//)
CLOSE (2)
STOP
END
```

PROGRAMME N°P20

```
PROGRAM  CALCUL_PANNES_Cas_3
REAL  L2,L1,L,IX,IY
OPEN(4,FILE='DB.DON')
OPEN(3,FILE='PROJET.DAT')
READ(4,*)  CP,CS,L1,L,Z,Q,E
WRITE(3,*)  'CHARGES  PERMAMENTES  CP  [DAN/M**2]  =',CP
WRITE(3,*)  'SURCHARGES                    CS  [DAN/M**2]  =',CS
WRITE(3,*)  'PORTEE  DE  LA  PANNE    L1  [M]            =',L1
WRITE(3,*)  'ENTRAXE                          L    [M]            =',L
WRITE(3,*)  'PENTE  AU  VERSANT        Z    [%]            =',Z
WRITE(3,*)  'LIMITE  ELASTIQUE        Q    [DAN/MM**2]=',Q
WRITE(3,*)  'MODULE  DE  YOUNG          E    [DAN/CM**2]=',E
P=((CP*4/3.)+(CS*3/2.))*L
Z=ATAND(Z)
PY=P*COS(Z)
PX=P*SIN(Z)
PN=(CP+CS)*L
PXX=PN*SIN(Z)
PYY=PN*COS(Z)
L2=L1*100
!   PANNES  SUR  2  APPUIS  SIMPLES  AVEC  LIERNES  AU  1/3  DE  LA  PORTEE
OPEN(1,FILE='IN.DON')
DO  N=1,20
READ(1,*)  N,IPN,IX,IY,WX,WY
QMAX=(0.0027*PX*L1**2)/(WY)+(PY*L1**2)/(8.*WX)
IF(QMAX.LE.Q)GOTO  61
ENDDO
61      FX=(0.00025*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200)  GOTO  63
DO  M=N+1,20
READ(1,*)  M,IPN,IX,IY,WX,WY
FX=(0.00025*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200)  GOTO  63
ENDDO
63        WRITE(3,4)
4   FORMAT(3X,'PANNES  SUR  2  APPUIS  AVEC  LIERNES  AU  1/3  DE  LA
PORTEE',/,3X,'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@',/)
WRITE(3,14)
14  FORMAT(3X,'CAS  1  :  A  MI-LONGUEUR',/,3X,'@@@@@@@@@@@@@@@@@@@@@@'  ,/)
WRITE(3,56)  IPN
56  FORMAT(3X,'IPN  FINAL  ===>',1X,I3,/)
CLOSE  (1)
OPEN(2,FILE='IE.DON')
DO  N=1,18
READ(2,*)  N,IPE,IX,IY,WX,WY
QMAX=(0.0027*PX*L1**2)/(WY)+(PY*L1**2)/(8.*WX)
IF(QMAX.LE.Q)GOTO  48
ENDDO
48  FX=(0.00025*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
```

```
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200)  GOTO  78
DO  M=N+1,18
READ  (2,*)  M,IPE,IX,IY,WX,WY
FX=(0.00025*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200)  GOTO  78
ENDDO
78        WRITE(3,57)IPE
57        FORMAT(3X,'IPE  FINAL  ===>',1X,I3,//)
CLOSE  (2)
OPEN(1,FILE='IN.DON')
DO  N=1,20
READ(1,*)  N,IPN,IX,IY,WX,WY
QMAX=(PY*L1**2)/(9.*WX)+(0.012*PX*L1**2)/(WY)
IF(QMAX.LE.Q)GOTO  81
ENDDO
81      FX=(0.00025*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200)  GOTO  83
DO  M=N+1,20
READ  (1,*)  M,IPN,IX,IY,WX,WY
FX=(0.00025*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200)  GOTO  83
ENDDO
83      WRITE(3,84)
84      FORMAT(3X,'CAS  2  :  AU  DROIT  DU
LIERNE',/,3X,'@@@@@@@@@@@@@@@@@@@@@@@@@@@@'  ,/)
          WRITE(3,85)  IPN
85  FORMAT(3X,'IPN  FINAL  ===>',1X,I3,/)
CLOSE  (1)
OPEN(2,FILE='IE.DON')
DO  N=1,18
READ(2,*)  N,IPE,IX,IY,WX,WY
QMAX=(PY*L1**2)/(9.*WX)+(0.012*PX*L1**2)/(WY)
IF(QMAX.LE.Q)GOTO  88
ENDDO
88      FX=(0.00025*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200)  GOTO  90
DO  M=N+1,18
READ(2,*)  M,IPE,IX,IY,WX,WY
FX=(0.00025*PXX*L2**4)/(100*E*IY)
FY=(0.013*PYY*L2**4)/(100*E*IX)
F=SQRT(FX**2+FY**2)
IF(F.LE.L2/200)  GOTO  90
ENDDO
90        WRITE(3,91)IPE
91        FORMAT(3X,'IPE  FINAL  ===>',1X,I3,//)
CLOSE  (2)
STOP
END
```

Bibliographie

1. L. Baghdali, l'enseignement du langage algorithmique et fortran 77, office des publications universitaires, 1996.
2. J. J. Hunsinger, le langage fortran, https : //fortran.developpez.com/cours/hunsinger-cours-complet/, 2007.
3. R. Khima, algorithmes mathématiques et langage basic, office des publications universitaires, 1988.
4. L. Baghdali & S. A. Laribi, langage de programmation pascal, entreprise nationale du livre, ENAL, 1994.
5. J. L. Jardrin, analyse algorithmiques et programmes en pascal, Dunod, 1989.
6. L. Mieussens, cours de fortran 90, 2011.
7. Alexandre Mayer, Cours de programmation : Fortran 90, http://perso.fundp.ac.be/~amayer.
8. D. Lefebvre, Introduction au FORTRAN, http : //www.tangentex.com/IntroFortranCN.htm, 2006.
9. P. Corde & A. Fouilloux, Cours langage Fortran, Institut du Développement et des Ressources en Informatique Scientifique, 2006.
10. M. Dobrijevic, introduction au langage de programmation fortran 90. Application au calcul scientifique, 2002.
11. P. Gentil & D. Martin, Fortran 77, 1994.