

TD /TP N° : 2 Structures conditionnelles et répétitives

I Structure conditionnelles :

En programmation, on est souvent confronté à des situations où on a besoin de choisir entre deux ou plusieurs alternatives pour cela on utilise une structure conditionnelle

I.1 La forme simple

Algorithme
si (condition) alors
 instruction1
 instruction n
finsi

Fortran
if(condition) then
 instruction1
 instruction n
endif

I.2. La forme alternative

algorithme
si (condition) alors
 instruction1
 instruction2
sinon
 instruction3
 instruction 4
finsi

Fortran
if(condition) then
 instruction1
 instruction 2
else
 instruction3
 instruction 4
endif

I.3. forme imbriquée

algorithme
si (condition1) alors
 instruction1
 instruction2
sinon
si (condition2) alors
 instruction3
 instruction4
 sinon si (condition3) alors
 instruction5
 instruction6
 sinon
 instruction7
finsi
finsi
finsi

Fortran
if (condition1) then
 instruction1
 instruction 2
else if(condition2) then
 instruction3
 instruction 4

else if(condition3) then
 instruction5
 instruction 6
else
 instruction7
endif

Exercice1 :

Traduire l'exercice 3,5 de la fiche de td N2 en Fortran

Exercie2 : étant donné x un réel, écrire un programme en fortran77 qui permet de calculer y selon les cas suivants

$$Y = \begin{cases} X^3 & \text{si } X \leq 1.5 \\ X^2 & \text{si } 1.5 < X < 2 \\ X & \text{si } X \geq 2 \end{cases}$$

II. Structures Répétitives

Sans doute, ce sont les instructions les plus utilisées en programmation FORTRAN 77 comporte deux types de boucles.

II.1. La boucle DO ... WHILE

Algorithme :
Tant que(condition) faire
 instruction1
 instruction 2
Fintq

Fortran
do while (condition)
 Instruction1
 Instruction2
enddo

II.2. La boucle DO...ENDDO

Algorithme :
Pour compteur ← valI a valf **faire**
 instruction1
 instruction 2
finpr

Fortran
do compteur= valI, valf, pas
 Instruction1
 Instruction2
enddo

Exemple

Algorithme :
Pour i 1 ← a 5 **faire**
 S ← S+i
 V ← V*V*V
finpr

Fortran
do i= 1, 5
 S=S+i
 V=V**3
enddo

Remarque 1 : lorsque le pas est égale à 1, pas besoin de l'indiquer

Remarque 2 : les fonctions float et dble permettent de convertir explicitement des données d'un type en un autre type

```
program somme
implicit none
integer a,b
real c
read(*,*) a,b
c=a/b
write(*,*)c
end
```

```
program somme
implicit none
integer a,b
real c
read(*,*) a,b
c=float(a)/b
write(*,*)c
end
```

```
program somme
implicit none
real a,b
double precision c
read(*,*) a,b
c=a/b
write(*,*)c
end
```

```
program somme
implicit none
real a,b
double precision c
read(*,*) a,b
c=dble(a)/b
write(*,*)c
end
```

Exercice1 :

Traduire l'exercice 2(partie1) de la fiche de td N3 en Fortran 77

Exercice 2

Soit l'algorithme suivant

Algorithme somme

Var i,SP,SI,N : **Entier**

Début

Ecrire('donner la valeur de N')

Lire(N)

SI ← 0

SP ← 0

Pour i ← 1 à N **faire**

Si (i mod 2=0) **alors**

SP ← SP+i

Ecrire(SP)

Sinon

SI ← SI+i

Ecrire(SI)

Finsi

Finp

Fin

-Traduire l'algorithme suivant en Fortran 77

Exercie3 : Ecrire un programme en fortran 77 qui permet de calculer la somme suivante étant donnés x un reel et n un entier,

$$s = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots + \frac{x^{2n+1}}{2n+1}$$

TP/TD N°3 : Les tableaux

Les tableaux de façon générale, ont une dimension (appelés vecteurs), deux dimensions ou plus (appelés matrices). Les tableaux regroupent des objets du même type, dans la syntaxe de déclaration on doit spécifier le type, le nom et les dimensions maximales du tableau

1. Déclaration tableau a une dimension (vecteur)

Exemple

integer A(100) :A est un vecteur d'entier qui peut contenir 100 éléments

real B(50): B est un vecteur de réel qui peut contenir éléments

2.L'écriture et lecture d'un vecteur :

Pour la lecture et l'écriture d'un tableau à une dimension, nous avons besoin d'utiliser une boucle, tel que "i" va prendre les valeurs de 1 à Dimension(n).

2.1La lecture dans un vecteur

```
do i= 1 , n  
read(*,*) V(i)  
enddo
```

2.2L'écriture dans un vecteur

```
do i = 1 , n  
write(*,*) V(i)  
enddo
```

3. Déclaration tableau a deux dimension (matrice)

Integer M(20,10) :M est une matrice d'entier composée de 20 lignes et de10 colonnes

Real A(3,5) : A est une matrice de réel composée de 3 lignes et de5 colonnes

4. L'écriture et lecture dans une matrice :

Pour la lecture et l'écriture dans un tableau à deux dimension , nous avons besoin d'utiliser deux boucles tel que "i" va prendre les valeurs de 1 à n(nombre de lignes) et j va prendre les valeurs de 1 à m(nombre de colonnes)

4.1La lecture dans une matrice

```
do i = 1 , n  
do j=1,m  
read(*,*) M(i,j)  
enddo  
enddo
```

L'écriture dans une matrice

```
do I = 1 , n  
do j=1,m  
write (*,*) M(i,j)  
enddo  
enddo
```

Remarque 1

Pour Améliorer l'affichage d'un vecteur WRITE(*,*) 'le vecteur v('i,i')=' v(i)

Pour Améliorer l'affichage d'une matrice : WRITE(*,*) 'la matrice x('i,i','j,j')=' x(i,j)

Remarque 2 : fortran 77 offre une autre manière de lire et écrire un vecteur

Exemple : read(*,*)(V(i),i=1,n)
write(*,*) (V(i),i=1,n)

Remarque 3 : fortran 77 offre une autre manière de lire et écrire une matrice

exemple

```
READ(*,*) ((Z(i,j),j=1,m),i=1,n)  
WRITE(*,*) ((Z(i,j),j=1,m),i=1,n)
```

Exercice 1 :

Traduire l'exercice N 2 de la fiche de td3 en fortran 77

Exercice 2 :

Ecrire un programme qui calcul le produit scalaire de deux vecteurs de réel U et V de même dimension.

Exercice 3 :

Ecrire un programme fortran qui permet de saisir un tableau V contenant des entiers dont la taille est ≤ 20 , puis de calculer la somme des ses éléments pairs et impairs.

Exemple :

V					SP=14
7	-4	12	6	9	SI=16

Ecrire un programme fortran qui permet de calculer le double de la matrice x de type réel dont la taille maximale est de 30 lignes et 20 colonnes

Exemple

$$x = \begin{bmatrix} 2.5 & -5 \\ -4 & 1.2 \\ 3 & 1 \end{bmatrix} \quad x = \begin{bmatrix} 5 & -10 \\ -8 & 2.4 \\ 6 & 2 \end{bmatrix}$$

Exercice 4:

```

program matrice
  implicit none
  integer M(20,20), S, T(20), i, j, n
  write(*,*)'donner la valeur de n '
  read(*,*) n
  do i =1,n
    do j=1,n
      write(*,*)'donner les éléments de la matrice '
      read (*,*) M(i,j)
    enddo
  enddo
  do i =1,n
    S=0
    do j=1,n
      S=S+M(j,i)
    enddo
    T(i)=S
  enddo
  do i =1,n
    write(*,*) T(i)
  enddo
end

```

1-Dérouler le programme pour $M = \begin{bmatrix} 4 & -1 & 7 \\ -1 & 3 & 5 \\ -5 & 1 & 2 \end{bmatrix}$

2-Que calcule ce programme

3-Améliorer l'affichage de la dernière partie du programme