

UNIVERSITE PIERRE ET MARIE CURIE

Licence de Chimie UE-209

INTRODUCTION A LA PROGRAMMATION FORTRAN

Plan :

1 : Introduction

2 : Structure d'un programme Fortran

3 : Aiguillages

4 : Processus itératifs

5 : Variables indicées

6 : Sous-programmes

Contrôle des connaissances :

CC N°1 (10 min - 5pts) pendant le deuxième cours

CC N°2 (10 min - 5pts) pendant le troisième cours

Examen (1h - 30pts) pendant le quatrième cours

Objectifs

L'objectif de cette UE est d'approfondir les connaissances des étudiants dans les objets et concepts mathématiques de base utilisés dans les différents domaines de la chimie. L'enseignement est donc axé sur les besoins des différents enseignements dispensés en L2 et L3 de la mention chimie, dont seront tirées les principales applications. Les concepts sont illustrés et complétés par des enseignements d'informatique qui vont familiariser les étudiants avec des notions de Linux et de Fortran. Ces notions d'informatique permettront ensuite aux étudiants lors de leurs études (L3 et Master) d'aborder l'informatique appliquée aux sciences chimiques.

Conventions utilisées

Le texte écrit `comme ceci` est un message affiché sur l'écran ou un texte à taper au clavier

Le texte écrit «*comme ceci*» est un texte que vous devez adapter selon le contexte.

la touche ENTER (entrée) est indiquée par le signe ↵

le caractère espace est indiqué par le signe ^

- COURS 5-

VARIABLES INDICEES

V.1) Déclarations.

Il est fréquent d'avoir à désigner sous un même nom de variable des quantités de même nature qui diffèrent par un système d'indice (séries, calcul matriciel,...). On peut définir un tableau de variables. La définition se fait lors des déclarations par :

```
^^^^^INTEGER^V(40)
```

```
^^^^^REAL^W(20,10)
```

qui signifient que V est un tableau unidimensionnel (vecteur) de 40 valeurs et W un tableau bidimensionnel (matrice) de 200 valeurs (20 lignes, 10 colonnes). Les mémoires correspondantes sont alors réservées. Dans le tableau, les valeurs sont rangées sur une seule ligne; le premier indice varie le premier, le second (qualifié pour un tableau bidimensionnel d'indice de colonne) ensuite et ainsi de suite dans le cas de dimension supérieure.

La déclaration a pour fonction

- de réserver la place nécessaire aux tableaux
- de déclarer la structure: variable d'un ou de plusieurs indices

Si l'indiciage débute par une valeur différente de 1, il est possible de le spécifier. Les valeurs extrêmes des indices sont séparées par deux points.

```
^^^^^REAL^V(20:60), W(10:30,10:20)
```

Cette définition implique une réservation de place excédentaire qui ne se justifie que dans des cas particuliers (adaptation à un format d'un tableau de taille différente reçu ou transmis)

Si vous oubliez la déclaration d'une variable indicée, il n'y aura pas de message à la compilation : Le compilateur va prendre votre variable avec des parenthèses pour une fonction que vous lui fournirez à l'édition de lien. Il y aura un message à l'édition de lien disant que cette variable est une référence externe non résolue.

V.2) Saisie des variables indicées.

Dans les instructions fortran les variables devront apparaître avec leurs indices: V(4) sera le quatrième élément du tableau V, W(3,7) sera le quarante septième élément du tableau W (troisième ligne, septième colonne). Dans le tableau, les indices peuvent être des variables entières à condition que la valeur soit définie lors de l'exécution, ou des expressions arithmétiques du premier degré où n'intervient pas le symbole quotient.

N'oubliez pas d'initialiser vos tableaux si vous ne voulez pas redéfinir pas toutes les valeurs ou si vous procédez par incrémentation. Chaque variable indicée doit avoir une valeur attribuée avant d'être utilisée. Cela peut être fait par attribution A(I)= individuellement ou dans une boucle ou sous forme de DATA.

```
^^^^^DATA (A(I), I=1, 10) / 3*0, 2*5, 4*10, 12 /
```

Enfin les données peuvent être lues:

```
^^^^^READ (*, *) (A(I), I=1, N)
```

```
^^^^^READ (*, *) ((A(I, J), J=1, N1), I=1, N2)
```

Quand l'indice maximum déclaré est une variable issue d'un calcul, il est utile de placer des tests sur cette valeur de façon à vérifier qu'elle ne dépasse pas les limites permises.

V.3) Présentation des variables indicées.

Utiliser une boucle explicite pour l'indice des lignes (I) et une boucle implicite pour celui des colonnes (J) :

- en lecture

```
^^^^^DO I=1, N
```

```
^^^^^^^^^READ (*, *) (A(I, J), J=1, N)
```

```
^^^^^ENDDO
```

- en écriture

```
^^^^^DO I=1, N
```

```
^^^^^^^^^WRITE (*, *) (A(I, J), J=1, N)
```

```
^^^^^ENDDO
```

V.4) L'instruction PARAMETER.

Si aucune instruction exécutable ne peut précéder une déclaration, cela implique que les valeurs des dimensions soient constantes. Cela n'est pas forcément vrai dans le cas de sous-programmes, si la réservation de place a été faite dans un programme principal utilisant éventuellement des adressages particuliers (allocation dynamique de mémoire). Cette solution n'est pas simple.

L'instruction PARAMETER est une instruction simple que l'on peut placer avant une déclaration.

```
^^^^^INTEGER N, M
```

```
^^^^^PARAMETER (N=10, M=N*N)
```

```
^^^^^REAL V(N), W(N2)
```

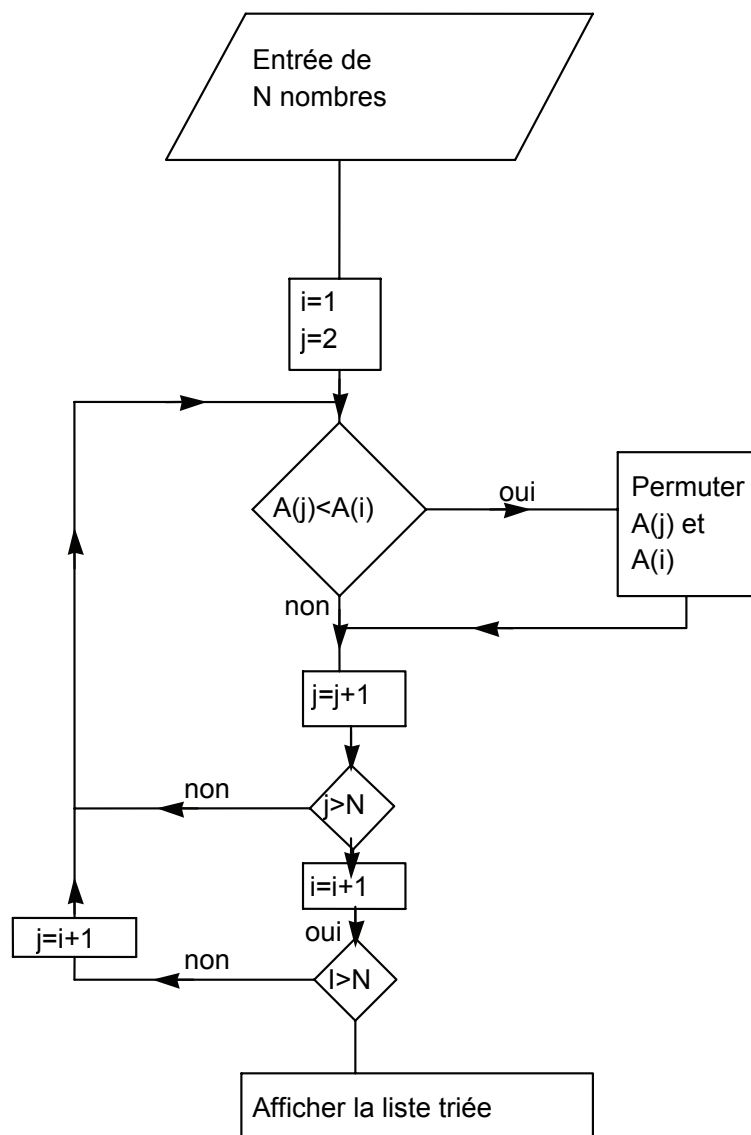
Les valeurs ainsi fixées ne peuvent plus être modifiées. L'avantage cependant est qu'en cas de modification, il n'y ait qu'une seule valeur à remplacer.

Les nombres N et M (et les expressions correspondantes) doivent être des entiers.
 La déclaration de A comme entier doit précéder la déclaration du paramètre.
 La valeur du paramètre peut être transmise en argument dans les sous-programmes.

Exercices:

V-1) Ecrire les programmes donnant la somme de deux matrices, le produit de deux matrices, le déterminant d'une matrice. Soigner la présentation des matrices. Ce programme ne doit pas être réalisé en conversationnel mais impérativement utiliser des fichiers de données. Utiliser les commandes unix : `EXE <DONNEES` pour faire lire dans le fichier `DONNEES` les valeurs des matrices A et B .

V-2) Tri à bulles Réaliser le programme de tri dont l'organigramme est ci-dessous. Le premier élément est comparé à tous les autres. Si l'ordre n'est pas respecté, on inverse les nombres de sorte que le premier soit le plus petit. On fait de même pour le second, puis les suivants jusqu'à ce que l'ordre soit satisfaisant.



V-3) Calculer le produit scalaire de deux vecteurs et déterminer l'angle que font ces deux vecteurs.

V-4) Quelle sera la programmation la plus rapide?

$\mathcal{A}$	$\mathcal{B}$
DO I=1,1000	DO J=1,100
READ(*,*) (A(I,J),J=1,100)	READ(*,*) (A(I,J),I=1,1000)
ENDDO	ENDDO