

Université Abdelmalek Essaâdi
Ecole nationale des sciences appliquées d'Al Hoceima
(ENSAH)

Algorithmique

Préparé et présenté par
Mr. Ouazzani Chahdi

Année universitaire: 2018/2019

1

04-Jan-19

Les tableaux

2

1-Présentation

- ❖ On veut élaborer un algorithme *Moyenne_Notes* qui permet de calculer la moyenne d'une série de notes saisie au clavier. Il doit aussi afficher cette série en sa totalité.
- ❖ On commencera par l'opération la plus simple, c'est le calcul de la moyenne, pour cela :
 - Tout d'abord, on doit demander à l'utilisateur la taille de la série;
 - On invite l'utilisateur à saisir les notes de la série et au fur et à mesure, on calcule la somme.
 - On calcule la moyenne.
 - Et finalement, on affiche le résultat à l'écran.
- ❖ On commence par un premier algorithme *Moyenne* :

3

```

Algorithme Moyenne
Variable note, som, moy : Réel
           N, i : Entier
Début
Ecrire("Donnez la taille de la série :")
Lire(N)
Som ← 0
Pour i allant de 1 Jusqu'à N Faire
  Ecrire("Donnez la note ", i)
  Lire(note)
  Som ← som + note
FinPour
moy ← som/N
Ecrire("La moyenne est : ", moy)
Fin
  
```

4

- ❖ Maintenant, si on veut en plus du calcul de la moyenne, afficher les notes de la série à l'écran, on doit procéder comment ?
- ❖ Pour que ça soit réalisable, on doit tout d'abord enregistrer les notes saisies.
- ❖ Et pour les enregistrer, on aura besoin de N variables, par exemple, si N = 4, alors on aura besoin des variables *note1*, *note2*, *note3*, *note4*.
- ❖ Le problème, c'est que le nombre N n'est pas connu à l'avance, alors on ne peut pas savoir à l'avance le nombre de variable à manipuler.
- ❖ Et si on suppose que N est connu à l'avance, par exemple 30, alors on aura besoin des variables *note1*, *note2*, ..., *note30*.
- ❖ Et dans ce cas là, le traitement pour une note sera réécrit 30 fois dans le même algorithme.

5

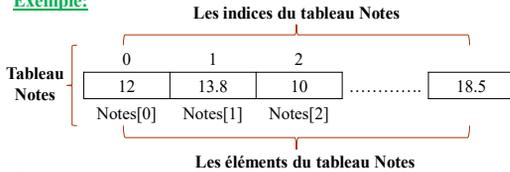
- ❖ La solution est de trouver une **structure** qui nous permet d'enregistrer cette série de notes sans connaître à l'avance le nombre N.
- ❖ De tel sorte que le **traitement lié à une note** sera **écrit une seule fois** dans l'algorithme et par la suite **appliqué à toutes les notes de la série**.
- ❖ Cette structure consiste à **rassembler** toutes les notes de la série **en une seule variable**, par exemple *Notes*.
- ❖ Et par la suite, chaque note sera récupérée par un **indice** *i*, par exemple *Notes[0]*, *Notes[1]*, *Notes[2]*,....
- ❖ Cette structure n'est rien autre qu'un **Tableau**.
- ❖ En algorithmique il existe deux types de tableaux les plus utilisés :
 - Tableaux à une seule dimension;
 - Tableaux à deux dimensions.

6

2-Définition

- ❖ Un tableau est une **structure de données** regroupant un ensemble de variables de **même type**, désignées par un **même nom** et distinguées les unes des autres par leurs numéros appelés **indices**.
- ❖ Cette structure permet d'effectuer un **même traitement** sur des données de même types.

Exemple:



7

3-Tableau à une seule dimension

- ❖ En algorithmique, un tableau d'une seule dimension sert à représenter un **vecteur de données de même type**.

3.1- Déclaration

Syntaxe: **Tableau** NomTableau[Taille] : **Type**

- **NomTableau** : le nom du tableau
- **Taille** : un entier naturel représentant le nombre d'éléments du tableau
- **Type** : le type des éléments du tableau

Exemple:

- **Tableau T[10]** : Entier : tableau de 10 éléments de type Entier;
- **Tableau tab[5]** : Caractère : Tableau de 5 éléments de type Caractère;

- ❖ La déclaration d'un tableau à une seule dimension est placée dans le **bloc de déclaration** d'un algorithme.

8

3.2- Accès aux éléments

- ❖ Pour accéder à l'élément de l'indice i dans un tableau T , on utilise la syntaxe $T[i]$.
- ❖ Les indices d'un tableau de N éléments varient de 0 à $N-1$.

Exemple:

- On considère un tableau T de 5 éléments de type entier :

0	1	2	3	4
4	7	3	8	11

- Ce tableau contient 5 éléments qui sont : 4, 7, 3, 8 et 11;
- Ses indices commencent de 0, il sont 0, 1, 2, 3 et 4;
- Le premier élément du tableau est $T[0]$;
- Le dernier élément du tableau est $T[4]$;
- $T[1] = 7$ et $T[3] = 8$;
- 4 se trouve à l'indice 0 et 3 se trouve à l'indice 2.

9

3.3- Saisie et parcours des éléments

- ❖ On considère un tableau T de N éléments de type X .
- ❖ Pour la saisie des éléments, on utilise la boucle suivante :

```
Pour i allant de 0 Jusqu'à N-1 Faire
    Lire(T[i])
FinPour
```

- ❖ Pour l'affichage des éléments, on utilise la boucle suivante :

```
Pour i allant de 0 Jusqu'à N-1 Faire
    Ecrire(T[i], " ")
FinPour
```

Exemple:

- Écrivez un algorithme *Affichage_Tableau* qui lit 10 valeurs entières au clavier et puis les affiche à l'écran.

10

Algorithme Affichage_Tableau

Variable i : Entier

Tableau $T[10]$: Entier

Début

```
Pour i allant de 0 Jusqu'à 9 Faire
    Ecrire("Donnez la valeur ", i+1)
    Lire(T[i])
FinPour
```

```
Ecrire("Les valeurs que vous avez saisies
sont : ")
```

```
Pour i allant de 0 Jusqu'à N-1 Faire
    Ecrire(T[i], " ")
FinPour
```

Fin

11

- ❖ Écrivons l'algorithme *Moyenne_Notes* :

Algorithme Moyenne_Notes

Variable som, moy : Réel

N, i : Entier

Tableau $Notes[50]$: Entier

Début

```
Ecrire("Donnez N (N <= 50) :")
```

```
Lire(N)
```

```
Som ← 0
```

```
Pour i allant de 0 Jusqu'à N-1 Faire
```

```
    Ecrire("Donnez la note ", i+1)
```

```
    Lire(Notes[i])
```

```
    Som ← som + Notes[i]
```

```
FinPour
```

```
moy ← som/N
```

```
Ecrire("La moyenne est : ", moy)
```

```
Ecrire("Les notes que vous avez saisies sont :")
```

```
Pour i allant de 0 Jusqu'à N-1 Faire
```

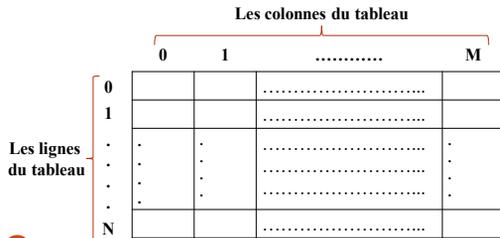
```
    Ecrire(Notes[i], " ")
FinPour
```

Fin

12

4-Tableau à deux dimensions

- ❖ En algorithmique, un tableau à deux dimensions sert à représenter une matrice de données de même type.
- ❖ Il se compose d'un nombre de lignes **N** et de colonnes **M**.



13

4.1- Déclaration

Syntaxe : `Tableau NomTableau [N,M] : Type`

- **NomTableau** : le nom du tableau
- **N** : un entier naturel représentant le nombre de lignes;
- **M** : un entier naturel représentant le nombre de colonnes;
- **Type** : le type des éléments du tableau.

Exemple:

- Tableau T[3, 4] : Entier : tableau de type Entier de 3 lignes et de 4 colonnes;
- Tableau tab[2, 5] : Caractère : Tableau de type Caractère, de 2 lignes et de 5 colonnes;

- ❖ La déclaration d'un tableau à deux dimensions est placée dans le bloc de déclaration d'un algorithme.

14

4.2- Accès aux éléments

- ❖ Pour accéder à l'élément de l'indice (i,j) dans un tableau T, on utilise la syntaxe `T[i, j]`.
- ❖ Les indices d'un tableau à deux dimensions de N lignes et de M colonnes varient de (0,0) à (N-1, M-1).

Exemple:

- On considère un tableau T de type Entier, de 3 lignes et de 4 colonnes.
- Ses indices sont (0,0), (0,1), ..., (2,3)
- Le premier élément du tableau est T[0,0]
- Le dernier élément est T[2, 3]
- 27 se trouve à l'indice (1,1).

	0	1	2	3
0	2	9	0	10
1	7	27	6	18
2	1	8	3	13

15

4.3- Saisie et parcours des éléments

- ❖ On considère un tableau T de type X, de N lignes et de M colonnes.

- ❖ Pour la saisie et l'affichage des éléments on utilise une imbrication de deux boucles *Pour...Jusqu'à...* :

Saisie :

```
Pour i allant de 0 Jusqu'à N-1 Faire
  Pour j allant de 0 Jusqu'à M-1 Faire
    Lire(T[i, j])
  FinPour
FinPour
```

Affichage:

```
Pour i allant de 0 Jusqu'à N-1 Faire
  Pour j allant de 0 Jusqu'à M-1 Faire
    Ecrire(T[i, j])
  FinPour
FinPour
```

16

Exemple:

Ecrivez un algorithme *Affichage_Matrice* qui remplit une matrice de 10 lignes et de 8 colonnes par des valeurs entières saisies au clavier, puis les affiche à l'écran.

Algorithme Affichage_Matrice

Variable i, j : Entier

Tableau T[10, 8] : Entier

Début

```
Pour i allant de 0 Jusqu'à 9 Faire
  Pour j allant de 0 Jusqu'à 7 Faire
    Ecrire("Donnez la valeur (" , i , " , " , j , " ) ")
    Lire(T[i, j])
  FinPour
FinPour
Pour i allant de 0 Jusqu'à 9 Faire
  Ecrire("Les éléments de la ligne " , i + 1)
  Pour j allant de 0 Jusqu'à 7 Faire
    Ecrire(T[i, j] , " ")
  FinPour
FinPour
Fin
```

17