



Ministère de l'enseignement supérieur et de la recherche scientifique
Université Djillali BOUNAAMA - Khemis Miliana (UDBKM)
Faculté des Sciences et de la Technologie
Département de Mathématiques et d'Informatique



Chapitre 5

Les Types Structurés: 1 : Les tableaux

MI-L1-UEF121 : Algorithmiques et Structures de Données I

Noureddine AZZOUZA

n.azzouza@univ-dbkm.dz

Plan du Cours

1. Problématique et Introduction

2. Tableaux à une dimension: Vecteurs

2.1 Définitions

2.2 Déclaration et manipulation

2.3 Algorithmes sur les tableaux

3. Tableaux à deux dimensions : Matrices

3.1 Définitions

3.2 Déclaration et manipulation

Problématique

Problématique

Calculer la moyenne des d'étudiants, puis d'afficher le classement final?

- ✓ **Proposition 1** : Utiliser une structure répétitive qui permettra de lire, calculer et afficher la moyenne de chaque étudiant.
 - ✓ **Problème 1** : on peut pas classer les étudiants car on est besoin de toutes les moyennes déjà calculées.
- ✓ **Proposition 2** : Utiliser autant de variables que des étudiants.
 - ✓ **Problème 1** : difficile à faire et à gérer : on écrira autant de blocs d'instructions que de nombre d'étudiants.
 - ✓ **Problème 2** : il se peut qu'on sait pas à l'avance le nombre d'étudiants exacte.



Introduction

- ✓ Pour remédier à ces problèmes, on doit utiliser une **structure de données** capable de **rassembler** toutes ces variables en **une seule**.
- ✓ Une telle structure est appelée **Tableau**. C'est un type structuré.
- ✓ Les types présentés jusqu'à maintenant sont des types simples. Il existe d'autres types dits structurés (complexes).
- ✓ Un type structuré est tout type défini à base d'autres types.



Définition

- ✓ Un tableau est une **structure de données** permettant de regrouper sous un même nom un ensemble d'éléments de même type.
- ✓ Un tableau est caractérisé par :
 - son **nom** : un identificateur unique
 - ses **éléments** : désigne les **cases** du tableaux. Ces éléments ont le même type qui peut être **simple** ou **structuré**.
 - sa **taille** : le nombre d'éléments que le tableau peut contenir. Définit lors de la déclaration par une **borne inférieure** et **borne supérieure**.

(il faut pas confondre la **taille maximale** déclaré avec la **taille effective** réellement utilisée dans l'algorithme)



Représentation

- ✓ Un tableau est constitué d'un nombre –fixé au déclaration– de cases **contigües** situé en mémoire centrale.
- ✓ Le tableau est repéré par l'adresse de son **1^{er}** élément
- ✓ L'**accès** à un élément du tableau se fait à l'aide de son **indice**.
- ✓ **Indice** d'un élément : son **index**, sa **position** dans le tableau

T

10.25	14.75	9.5	12.00
1	2	3	4	5	6	7	8	9	10

Adresse @	Mémoire (Valeur)	Variables	
0			
1			
2			
14788	10.25	1	T
14789	14.75	2	
14790	9.5	3	
14791	.	.	
14792	12.00	10	
63997			
63998			
63999			

Tableaux 1D

Les Vecteurs



Définition

- ✓ Un tableau **à une dimension** ou **vecteur** est une manière de ranger des éléments ou des valeurs de même type.
- ✓ Il **regroupe** ces éléments dans une structure fixe et permet d'accéder à chaque élément par l'intermédiaire de son rang ou **indice**.

A

4	12	8	-4	0	-7	15	2	-24	17
1	2	3	4	5	6	7	8	9	10



Déclaration

Syntaxe

```
Var nom_tableau : Tableau [borne_inf .. borne_up] Type_elements;
```

ou bien

```
Type nom_tableau = Tableau [borne_inf .. borne_up] Type_elements;
```

✓ Exemples :

```
Type Tab_Moy = Tableau [1..20] de réel;  
Tab_Nom = Tableau [1..100] de chaine;
```

```
Var Moyennes : Tab_Moy ;  
nom_etudiant, nom_enseignants : Tab_Nom ;  
Notes : Tableau [1..100] de réel;  
Coef : Tableau [1..20] d'entier;
```



Manipulation

- ✓ **L'accès** à un élément du tableau s'effectue en précisant le *nom du tableau* suivi de la valeur de l'*indice* entre crochets.

Syntaxe : lecture / consultation

```
nom_variable ← nom_tableau [indice_element];
```

✓ Exemples :

```
N1 ← Notes[3];  
coef_math ← Coef[i];  
moy_a ← Coef[x1+x2 MOD a];  
note_alg ← Notes[x]*Coef[y];
```

Syntaxe : écriture / modification

```
nom_tableau [indice_element] ← expression
```

✓ Exemples :

```
Notes[5] ← N2;  
Coef[j] ← coef_phy  
Coef[x1*x2 DIV b] ← moy_b
```

Tableaux à 1D : Les Vecteurs

Tableaux 1D : Vecteurs

Déclaration

PASCAL

Syntaxe:

VAR nom_var= **Array**[1.. Long_max] **of**
type ;

Exemples

```
program Exemple;

var
  t : Array[1..5] of Integer;
  somme, i : Integer;

begin
  somme := 0;
  for i:= 1 to 5 do
    Readln(t[i]);

  for i:= 1 to 5 do
    somme := somme + t[i];

  Writeln('la somme = ', somme);

end.
```

C

Syntaxe:

type nom_var[long_max]

```
#include <stdio.h>

int main (){

  int t[5];
  int somme = 0;
  int i;

  for (i = 0; i <= 5; ++i)
    scanf("%d\n", t[i]);

  for (i = 0; i <= 5; ++i)
    somme += t[i];

  printf("%d\n", somme);

  return 0;
}
```

Algorithmes

sur les Vecteurs



Exemple 1 : Remplir un tableau

Remplir un tableau

Soit T un tableau de N valeur entière avec $N < 100$. Ecrire un algorithme qui remplit le tableau T .

Analyse :

Remplir un tableau \rightarrow Lire de valeur et le sauvegarder dans un tableau

Lire un tableau \rightarrow lire le éléments de tableau

$T[i]$: le $i^{\text{ème}}$ élément dans le tableau (case $N^{\circ} i$)

Lire ($T[i]$) : lire une valeur –à partir de clavier- et la sauvegarder dans la case numéro « i » dans le tableau T

```
Algorithme tab_remplir;  
Type Tab = Tableau[1..100] de entier;  
Var T : Tab ;  
    i : entier;  
Début  
    Ecrire ("Donner le nombre d'élément de T");  
    Lire (N);  
  
    Pour i de 1 à N Faire  
        Lire(T[i]);  
    Fpour  
Fin.
```

Exemple 2 : Afficher un tableau

Remplir un tableau

Soit T un tableau de N valeur entière avec $N < 100$. Ecrire un algorithme qui affiche le contenu du tableau T.

Analyse :

Afficher un tableau → afficher le contenu du tableau

Afficher le contenu d'un tableau → afficher la valeur de chacune des cases de tableau

Ecrire (T[i]) : afficher –sur l'écran- la valeur de la case numéro « i » dans le tableau T

Algorithme tab_affiche;

*Var T : Tableau[1..100] de entier;
i : entier;*

Début

*Ecrire ("Donner le nombre d'élément de T");
Lire (N);*

*Pour i de 1 à N Faire
Ecrire(T[i]);*

Fpour

Fin.

Exemple 3 : La somme des éléments d'un tab

Somme des éléments

Soit T un tableau de N valeur entière avec $N < 100$. Ecrire un algorithme qui calcule et affiche la somme des éléments du tableau.

Analyse :

Déclarer une variable Somme « S » et l'initialiser à zéro

Parcourir tous les éléments du tableau
à chaque fois, on ajoute la valeur de la case courante à la somme « S »

```
Algorithme tab_somme;  
Type Tab = Tableau[1..100] de entier;  
Var T : Tab;  
    S, i : entier;  
Début  
    Ecrire ("Donner le nombre d'élément de T");  
    Lire (N);  
    S ← 0;  
    Pour i de 1 à N Faire  
        S ← S + T[i];  
    Fpour  
    Ecrire ("La somme = ", S);  
Fin.
```

Algorithmes

de Trie



Tableaux 2D

Les Matrices



Définition

- ✓ Un tableau à deux dimensions appelé également **matrice** est une structure de données permettant d'organiser des informations de même type en **lignes** et en **colonnes**.
- ✓ La **taille d'une matrice** est son **nombre de lignes** et son **nombre de colonnes**.
- ✓ **Exemples** : La matrice « A » est un tableau de 4 lignes et 5 colonnes

A	1	2	3	4	5
1	4	12	8	-4	0
2	15	-2	-7	9	5
3	3	23	39	-10	-1
4	-14	18	5	2	-19

Déclaration

Syntaxe

```
Var nom_tableau : Tableau [1.. nb_lignes, 1 .. nb_colonnes] Type_elements;  
ou bien
```

```
Type nom_tableau = Tableau [1.. nb_lignes, 1 .. nb_colonnes] Type_elements;
```

✓ Exemples :

```
Type Tab_Moy = Tableau [1..20, 1..50] de réel;  
Tab_Nom = Tableau [1..100, 1..100] de chaine;
```

```
Var Moyennes : Tab_Moy ;  
nom_etudiant, nom_enseignants : Tab_Nom ;  
Notes : Tableau [1..100, 1..100] de réel;  
Coef : Tableau [1..20, 1..20] d'entier;
```



Manipulation

- ✓ **L'accès** à un élément du tableau s'effectue en précisant le **nom du tableau** suivi de la valeur de l'**indice** de la ligne et l'**indice** de la colonne entre crochets (séparés par ,).

Syntaxe : lecture / consultation

```
nom_variable ← nom_tableau [indice_lign, indice_col];
```

✓ Exemples :

```
N1 ← Notes[3,4];  
coef_math ← Coef[i,j];  
moy_a ← Coef[x1+x2, b MOD a];  
note_alg ← Notes[x1,y1]*Coef[x2,y2];
```

Syntaxe : écriture / modification

```
nom_tableau [indice_lign, indice_col] ← expression;
```

✓ Exemples :

```
Notes[5,2] ← N2;  
Coef[j,k] ← coef_phy  
Coef[x1*x2, a DIV b] ← moy_b
```

Tableaux à 2D : Les Matrices

Tableaux 2D : Matrices

Déclaration

PASCAL

Syntaxe:

VAR nom_var= **Array**[1.. Max_ligne, Max_col] **of** type ;

Exemples

```
program Exemple;
var
  m : Array[1..5,1..10] of Integer;
  i, j : Integer;
begin
  somme := 0;
  for i:= 1 to 5 do
    for j:= 1 to 10 do
      Readln(m[i,j]);
    ...
  end.
```

C

Syntaxe:

type nom_var[Max_ligne][Max_col]

```
#include <stdio.h>

int main (){

  int m[5][10];
  int i,j;

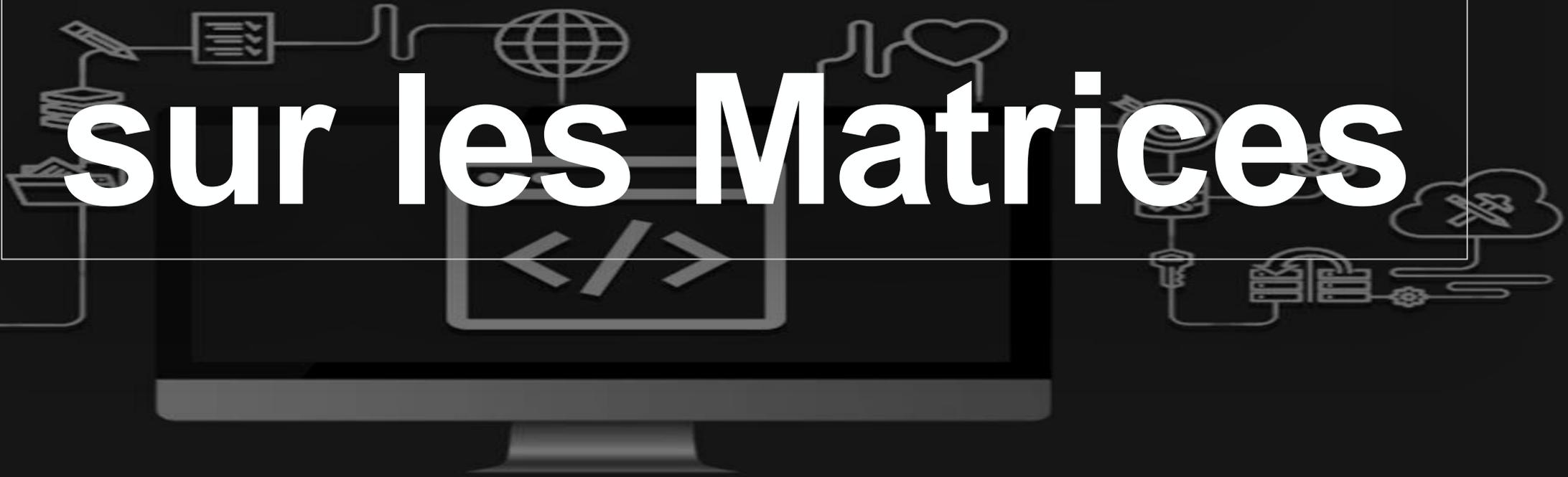
  for (i = 0; i <= 5; ++i)
    for (j = 0; j <= 10; ++j)
      scanf("%d\n", m[i][j]);

  ...

  return 0;
}
```

Algorithmes

sur les Matrices



Exemple 1 : Remplir une matrice

Remplir un tableau

Soit A un tableau à deux dimensions de N valeur entière avec $N < 100$. Ecrire un algorithme qui remplit le tableau T.

Analyse :

Remplir un tableau → Lire de valeur et le sauvegarder dans un tableau

Lire un tableau → lire le éléments de tableau

T[i] : le $i^{\text{ème}}$ élément dans le tableau (case N° i)

Lire (T[i]) : lire une valeur –à partir de clavier- et la sauvegarder dans la case numéro « i » dans le tableau T

```
Algorithme tab_remplir;  
Type Tab = Tableau[1..100] de entier;  
Var T : Tab ;  
    i : entier;  
Début  
    Ecrire ("Donner le nombre d'élément de T");  
    Lire (N);  
  
    Pour i de 1 à N Faire  
        Lire(T[i]);  
    Fpour  
Fin.
```

Exemple 2 : Afficher une matrice

Remplir un tableau

Soit T un tableau de N valeur entière avec $N < 100$. Ecrire un algorithme qui affiche le contenu du tableau T .

Analyse :

Afficher un tableau \rightarrow afficher le contenu du tableau

Afficher le contenu d'un tableau \rightarrow afficher la valeur de chacune des cases de tableau

Ecrire ($T[i]$) : afficher –sur l'écran- la valeur de la case numéro « i » dans le tableau T

Algorithme $tab_affiche;$

Var $T : Tableau[1..100]$ de entier;
 $i : entier;$

Début

$Ecrire ("Donner le nombre d'élément de T");$
 $Lire (N);$

Pour i de 1 à N Faire
 $Ecrire(T[i]);$

Fpour

Fin.

Chapitre 5

Les Types Structurés: 1 : Les tableaux

MI-L1-UEF121 : Algorithmiques et Structures de Données I

Noureddine AZZOUZA

n.azzouza@univ-dbkm.dz