

## Chapitre 3: Les variables indicées

### I. Les vecteurs (tableaux à une dimension)

#### 1. Structure de données

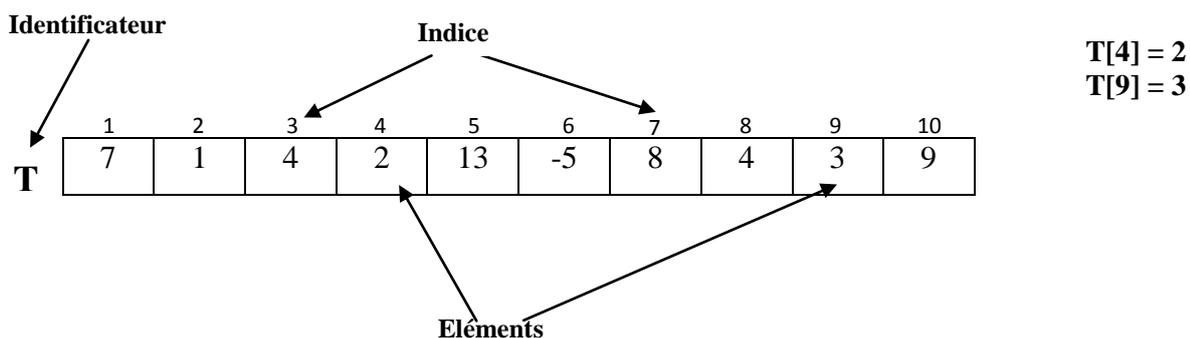
- Les types de bases (entier, réel, car, chaîne et booléen) correspondent à des variables qui ne peuvent stocker qu'une seule valeur simple.
- Une *structure de données* ou bien un *type structuré* correspond à des variables *composées* d'un nombre de données élémentaires reliées d'une certaine façon.
  - ✓ **Structure de données homogène:** une structure de données est dite *homogène* si tous les éléments de la structure (les données élémentaires) sont de *même type*.
  - ✓ **Structure de données hétérogène:** une structure de données est dite *hétérogène* si les éléments de la structure (les données élémentaires) sont de *types différents*.

#### 2. Tableaux à une dimension (vecteurs)

Un *tableau à une dimension* ou bien un *vecteur* est une structure de données homogène formée de cellules contiguës, *identifiée par un nom* et a un *accès direct*.

- L'**accès direct** signifie que nous pouvons obtenir directement le contenu d'une cellule (il n'est pas nécessaire de connaître le contenu des cellules précédentes).
- L'accès à un élément du tableau (obtenir le contenu d'une cellule) se fait en utilisant *un indice*, qui représente la *position (le rang)* de cet élément dans le tableau.

**Exemple:** un tableau **T** contenant dix (10) éléments de type entier.



- ✓ Le *quatrième* élément du tableau est égal à **2** et le *neuvième* élément est égal à **3**.

#### 3. Déclaration d'un vecteur

```
Var <identificateur> : tableau [intervalle] de <type>
```

*Ou bien*

```
Var <identificateur> : tableau [taille] de <type>
```

- ✓ <identificateur> est le nom du tableau;
- ✓ <intervalle> est l'intervalle entre l'indice du premier élément et l'indice du dernier élément;
- ✓ <taille> est la taille (nombre d'éléments) du tableau;
- ✓ <type> est le type des éléments du tableau.

**Exemple :** déclaration du tableau précédent.

```
Var T : tableau [1..10] de Entier
```

*Ou bien*

```
Var T : tableau [10] de Entier
```

**Remarques:**

- 1) Le nombre d'éléments d'un tableau est fixé dans la déclaration et ne peut pas changer.
- 2) On peut déclarer le tableau précédent de la manière suivante:

```
Type Tab = tableau [1..10] de Entier
Var T : Tab
```

#### 4. Opérations sur les éléments d'un vecteur

a) **Accès à un élément du tableau:** il est désigné par:

```
<identificateur>[indice de l'élément]
```

**Ex.** T[4] correspond au quatrième élément du tableau T.

T[4] = 2 (le quatrième élément du tableau T est égale à 2).

b) **Lecture et écriture d'un élément du tableau:**

- Lire (T[3])  $\Leftrightarrow$  lire (saisir) le 3<sup>ème</sup> élément du tableau T;
- Ecrire (T[3])  $\Leftrightarrow$  écrire (afficher) le 3<sup>ème</sup> élément du tableau T;
- Lire (T[i])  $\Leftrightarrow$  lire (saisir) le i<sup>ème</sup> élément du tableau T;
- Ecrire (T[i])  $\Leftrightarrow$  écrire (afficher) le i<sup>ème</sup> élément du tableau T.

c) **Affectation:**

- T[i] ← valeur      **Exemple** T[5] ← 20
- T[i] ← variable    **Exemple** T[5] ← x
- T[i] ← expression **Exemple** T[5] ← (x+y) \* 2

d) **Opérations de calcul:**

Exemple:      T[i] ← T[j] \* 2

e) **Faire des comparaisons:**

Exemple: **Si** (T[i] > 0) **Alors** ...

**5. Opérations sur la totalité d'un vecteur**

Pour faire des opérations sur la **totalité** du vecteur (sur tous les éléments) on utilise des **boucles**.

➤ *Lecture (saisie) du vecteur T*

```
Pour i=1 à 10 Faire
  Lire (T[i])
FinPour
```

➤ *Ecriture (affichage) du vecteur T*

```
Pour i=1 à 10 Faire
  Ecrire (T[i])
FinPour
```

➤ *Tri d'un vecteur*➤ *Calcul de la moyenne d'un vecteur*➤ *Fusion de deux vecteurs*➤ *... etc.***II. Les matrices (tableaux à deux dimensions)**

Un tableau à *deux dimensions* ou bien une *matrice* est un tableau dont *chaque élément est un vecteur* (tableau de tableaux). Un tableau à deux dimensions est organisé sous forme de **lignes** et de **colonnes**.

**Exemple:** soit la matrice M suivante, qui représente les notes de cinq (05) étudiants dans quatre (04) modules.

		Modules			
		1	2	3	4
M	1	12	11.5	13	15
	2	11	10	14	11
	3	14	<b>13</b>	12	11
	4	18	14,5	15	9
	5	13	17	10	16

$M[3,2] = 13$   
 ↗ Ligne 3      ↘ Colonne 2

- L'accès à un élément d'une matrice se fait en utilisant *deux indices*, le premier représente le *numéro de la ligne* et le deuxième représente le *numéro de la colonne* de cet élément.

## Déclaration d'une matrice

```
Var <identificateur> : tableau [nbr_lignes,nbr_colonnes] de <type>
```

- ✓ <identificateur> est le nom de la matrice;
- ✓ <nbr\_lignes> est le nombre de lignes *ou bien l'intervalle entre l'indice de la première ligne et l'indice de la dernière ligne*;
- ✓ <nbr\_colonnes> est le nombre de colonnes *ou bien l'intervalle entre l'indice de la première colonne et l'indice de la dernière colonne*;
- ✓ <type> est le type des éléments de la matrice.

**Exemple :** déclaration de la matrice **M** précédente.

```
Var M : tableau [1..5,1..4] de Réel
```

*Ou bien*

```
Var T : tableau [5,4] de Réel
```

### Remarques:

- 1) Le nombre de lignes et le nombre de colonnes d'une matrice sont fixés dans la déclaration et ne peuvent pas changer.
- 2) On peut déclarer la matrice précédente de la manière suivante:

```
Type Mat = tableau [1..5,1..4] de Entier  
Var M : Mat
```

## Accès à un élément de la matrice

L'accès à un élément de la matrice se fait selon la syntaxe suivante:

```
<identificateur>[num_ligne , num_colonne]
```

- ✓ num\_ligne : numéro de la ligne contenant l'élément;
- ✓ num\_colonne : numéro de la colonne contenant l'élément.

**Ex.**  $M[3, 2]$  représente la 2<sup>ème</sup> note du 3<sup>ème</sup> étudiant.

$$M[3, 2] = 13$$

## Opérations sur les matrices

Pour faire des opérations sur tous les éléments de la matrice, on utilise deux boucles imbriquées.

**Exemple 1:** lecture (saisie) d'une matrice  $M$  ayant  $N$  lignes et  $C$  colonnes.

```
Pour i=1 à N Faire
|
|   Pour j=1 à C Faire
|   |   Lire (M[i,j])
|   |   FinPour
|   FinPour
FinPour
```

**Exemple 2:** écriture (affichage) d'une matrice  $M$  ayant  $N$  lignes et  $C$  colonnes.

```
Pour i=1 à N Faire
|
|   Pour j=1 à C Faire
|   |   Ecrire (M[i,j])
|   |   FinPour
|   FinPour
FinPour
```

### Exercices:

1. Donner un algorithme qui permet de:

- Remplir deux tableaux  $A$  et  $B$  de taille  $N$  ( $N=100$ ) par des nombres réels;
- Construire et afficher le tableau  $C=A+B$ .

2. Donner un algorithme qui permet de:

- Remplir un tableau  $T$  de  $N$  éléments ( $N=100$ );
- Calculer et afficher le nombre d'occurrences d'un élément donné ( $e$ ) dans le tableau  $T$ .

3. Ecrire un algorithme qui permet de:

- Remplir une matrice  $M$  de taille  $L \times C$  ( $L = 10$  et  $C = 20$ ) par des nombres entiers;
- Calculer et afficher le nombre des éléments pairs et le nombre des éléments impairs.

### Solutions:

```

Algorithme exo1
Const N = 100
Type Tab = Tableau [1..N] de Réel
Var A,B,C:Tab
    i:Entier

Début
  Pour i=1 à N Faire
    Lire (A[i])
  FinPour
  Pour i=1 à N Faire
    Lire (B[i])
  FinPour
  Pour i=1 à N Faire
    C[i] ← A[i]+B[i]
    Ecrire(C[i])
  FinPour
Fin

```

} *Lecture de A*

} *Lecture de B*

} *Calcul et Affichage de C*

```

Algorithme exo2
Const N = 100
Type Tab = Tableau [1..N] de Réel
Var T:Tab
    e:Réel
    i,nb_occ:Entier

Début
  Pour i=1 à N Faire
    Lire (T[i])
  FinPour
  Lire (e)
  nb_occ ← 0
  Pour i=1 à N Faire
    Si (T[i]=e) Alors
      nb_occ ← nb_occ+1
    FinSi
  FinPour
  Ecrire (nb_occ)
Fin

```

```

Algorithme exo3
Const L = 10
    C = 20
Type Mat= Tableau[1..L,1..C] de Entier
Var M:Mat
    i,j,nb_pair,nb_impair:Entier

Début
  Pour i=1 à L Faire
    Pour j=1 à C Faire
      Lire (M[i,j])
    FinPour
  FinPour
  nb_pair ← 0
  nb_impair ← 0
  Pour i=1 à L Faire
    Pour j=1 à C Faire
      Si (M[i,j] mod 2 = 0) Alors
        nb_pair ← nb_pair+1
      Sinon
        nb_impair ← nb_impair+1
      FinSi
    FinPour
  FinPour
  Ecrire (nb_pair)
  Ecrire (nb_impair)
Fin

```