

# CHAPITRE : LES VARIABLES INDICEES

## Partie A : Les tableaux à une dimension (Vecteurs)

### I. DEFINITION :

Les tableaux représentent une structure de données permettant de stocker des données de même nature. On les représente souvent par un ensemble de cases contenant chacune une valeur.

Les tableaux peuvent avoir plusieurs dimensions dont les plus répandus sont les tableaux à une dimension (vecteurs) et les tableaux à deux dimensions (matrices).

	1 <sup>ère</sup> case	2 <sup>ème</sup> case	3 <sup>ème</sup> case	...	100 <sup>ème</sup> case
Moyenne	15.33	13.75	9.00	...	12.50

Un vecteur peut être vu comme une liste d'éléments arrangés dans des cases. Chaque case est référencée par un indice permettant d'identifier sa valeur.

**Exemple :** vecteur *Moyenne*

- Un tableau possède un nom (ici *Moyenne*) et un nombre d'éléments (de cases) qui représente sa taille (ici 100).
- Tous les éléments d'un tableau ont le même type (ici les éléments sont des réels).
- Pour désigner un élément, on indique le nom du vecteur suivi par son indice (son numéro) entre crochets : *Moyenne* [2] représente le 2<sup>ème</sup> élément du tableau *Moyenne* et vaut 13.75.

### II. DECLARATION DES TABLEAUX A UNE DIMENSION

La syntaxe de la déclaration d'une variable tableau est la suivante:

```
<Identificateur> [indiceMin.. indiceMax] : tableau de <type>
```

**Exemple :**

Moyenne [1..100] : tableau de réels

Tab [1..20] : tableau d'entiers

ElemT [1..200] : tableau de booléens

### III. MANIPULATION DES VECTEURS

Les tableaux se manipulent à travers leurs éléments représentés par le nom du tableau suivi d'un indice entre crochets.

L'**indice** d'un élément peut être:

- directement une valeur ex: Moyenne [10]
- une variable ex: Moyenne [i]
- une expression entière ex: Moyenne [2\*i+k] avec i et k de type entier

Quelque soit sa forme, **la valeur de l'indice** doit être :

- entière
- comprise entre les valeurs minimales et maximales déterminées à la déclaration du tableau.

Par exemple, avec le tableau Tab [1..50], il est impossible d'écrire Tab [0] et Tab [51]. Ces expressions font référence à des éléments qui n'existent pas.

### IV. LECTURE D'UN VECTEUR

Cette opération de lecture consiste en le remplissage des différentes cases qui constituent le tableau.

**Exemple :**

Ecrire un algorithme qui permet de lire un tableau de 20 éléments entiers.

```
Algorithme LectureTableau
Variables :
  Tab [1..20] : tableau d'entiers
  i : entier
Début
  |
  |   Pour i allant de 1 à 20 Faire
  |   |   Lire (Tab[i])
  |   Fin pour
Fin
```

## V. EDITION D'UN VECTEUR

L'édition d'un tableau consiste en l'affichage de ses éléments c'est-à-dire le contenu de ses différentes cases.

**Exemple :** Ecrire un algorithme qui permet d'afficher un tableau de 20 éléments entiers.

```
Algorithme LectureTableau
Variables :
  Tab [1..20] : tableau d'entiers
  i : entier
Début
  |
  |   Pour i allant de 1 à 20 Faire
  |   |   Ecrire (Tab[i])
  |   Fin pour
Fin
```

**Exercices :**

- Exercice1 :** Ecrire un algorithme qui calcule la moyenne des valeurs d'un tableau de 100 éléments entiers.
- Exercice2 :** Ecrire un algorithme qui calcule le nombre d'entiers positifs et le nombre d'entiers négatifs d'un tableau de 200 éléments.
- Exercice3 :** Ecrire un algorithme qui permet de trouver la valeur maximale d'un tableau de 20 éléments réels.
- Exercice4 :** Ecrire un algorithme qui permet de trouver si une valeur « X » appartient à un tableau de 100 éléments entiers.

## SOLUTION

**Algorithme** exercice1

**Variables :**

Tab [1..100] : tableau d'entiers

Som, i : entier

Moy : réel

**Début**

**Pour** i allant de 1 à 100 **Faire**

| Lire (Tab[i])

**Fin pour**

Som  $\leftarrow$  0

**Pour** i allant de 1 à 100 **Faire**

| Som  $\leftarrow$  Som+ Tab[i]

**Fin pour**

Moy  $\leftarrow$  Som/100

Ecrire ('Moyenne=', Moy)

**Fin**

### Algorithme exercice2

#### Variables :

Tab [1..200] : tableau d'entiers

Pos, Neg, i : entier

#### Début

**Pour** i allant de 1 à 200 **Faire**

    Lire (Tab[i])

**Fin pour**

    Pos  $\leftarrow$  0

    Neg  $\leftarrow$  0

**Pour** i allant de 1 à 200 **Faire**

**Si** (Tab[i] < 0) **Alors**

        Neg  $\leftarrow$  Neg+1

**Sinon**

        Pos  $\leftarrow$  Pos+1

**Fin si**

**Fin pour**

    Ecrire ('Le nombre de valeurs positives=', Pos)

    Ecrire ('Le nombre de valeurs négatives=', Neg)

#### Fin

### Algorithme exercice3

#### Variables :

Tab [1..20] : tableau de réels

i : entier

max : réel

#### Début

**Pour** i allant de 1 à 20 **Faire**

    Lire (Tab[i])

**Fin pour**

    max  $\leftarrow$  Tab [1]

**Pour** i allant de 2 à 20 **Faire**

**Si** (Tab[i] > max) **Alors**

        max  $\leftarrow$  Tab [i]

**Fin si**

**Fin pour**

    Ecrire ('Valeur max=', max)

#### Fin

**Algorithme** exercice4

**Constantes :**

N = 100

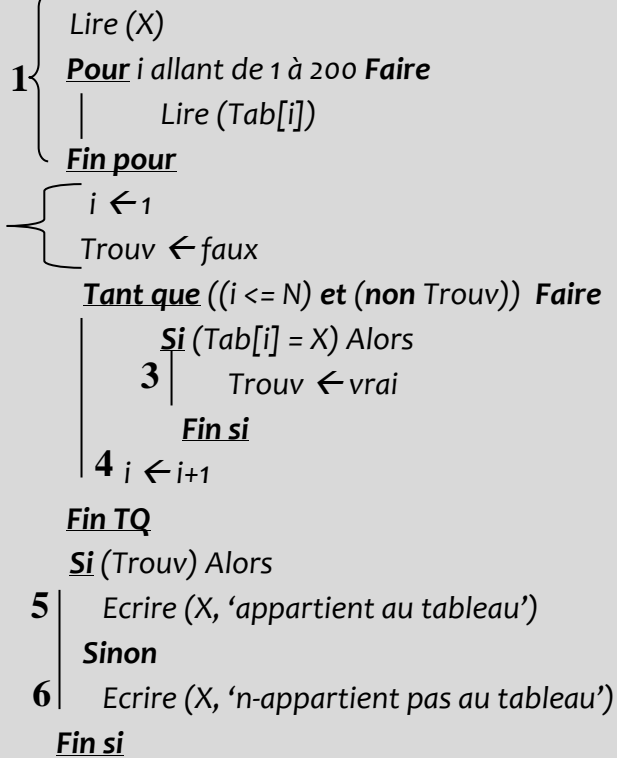
**Variables :**

Tab [1..N] : tableau d'entiers

X, i : entier

Trouv : booléen

**Début**



**Fin**

- Tracé d'exécution (N=5):

Etape	N	X	i	Trouv	Tab[1]	Tab[2]	Tab[3]	Tab[4]	Tab[5]	Ecran
1	5	9	/	/	2	7	9	12	3	/
2	5	9	1	Faux	2	7	9	12	3	/
4	5	9	2	Faux	2	7	9	12	3	/
4	5	9	3	Faux	2	7	9	12	3	/
3	5	9	3	Vrai	2	7	9	12	3	/
4	5	9	4	Vrai	2	7	9	12	3	/
5	5	9	4	Vrai	2	7	9	12	3	9 appartient au tableau

Etape	N	X	l	Trouv	Ta [1]	Tab[2]	Tab[3]	Tab[4]	Tab[5]	Ecran
1	5	8	/	/	2	7	12	9	3	/
2	5	8	1	faux	2	7	12	9	3	/
4	5	8	2	faux	2	7	12	9	3	/
4	5	8	3	faux	2	7	12	9	3	/
4	5	8	4	faux	2	7	12	9	3	/
4	5	8	5	faux	2	7	12	9	3	/
4	5	8	6	faux	2	7	12	9	3	/
6	5	8	6	faux	2	7	12	9	3	8 n'appartient pas au tableau

# Partie B : Les tableaux à deux dimensions (Matrices)

## I. DEFINITION :

Les matrices sont des tableaux à deux dimensions, c'est-à-dire une structure de données permettant de stocker des informations, de même nature, référencées par leurs numéros de ligne et de colonne.

## II. CARACTERISTIQUES DES MATRICES

Les matrices présentent les mêmes caractéristiques que les vecteurs. Les éléments d'une matrice sont arrangés dans des cases référencées par deux indices servant d'identifier sa valeur.

**Exemple :** matrice MAT

		1 <sup>ère</sup> colonne	2 <sup>ème</sup> colonne	3 <sup>ème</sup> colonne	...	100 <sup>ème</sup> colonne
MAT	1 <sup>ère</sup> ligne	5	2	0	...	7
	2 <sup>ème</sup> ligne	8	23	40	...	12
	...	...	...	...	...	...
	200 <sup>ème</sup> ligne	3	15	32	...	4

- Une matrice possède un nom (ici MAT) et un nombre d'éléments qui représente sa taille (ici 100\*200).
- Tous les éléments d'une matrice ont le même type (ici les éléments sont des entiers).
- Pour désigner un élément, on indique le nom de la matrice suivi par ses indices (indice ligne et indice colonne) entre crochets : MAT [2,3] représente la case située à l'intersection de la ligne « 2 » et la colonne « 3 » et vaut 23.

## III. DECLARATION DES MATRICES

La syntaxe de la déclaration d'une variable matrice est la suivante:

**<Identificateur> [LiMin.. LiMax, ColMin..ColMax] : Matrice de <type>**

**Exemple :**

MAT [1..100,1..200] : matrice d'entiers

Somme [1..20,1..30] : matrice de réels



#### IV. MANIPULATION DES MATRICES

Les éléments d'une matrice se manipulent en spécifiant le nom de la matrice suivi des deux indices ligne et colonne entre crochets.

Similairement aux vecteurs, Les deux indices ligne et colonne peuvent avoir plusieurs formes : valeur, variable ou expression. Quelque que soit cette forme, la valeur des deux indices doivent être de type entier et être dans l'intervalle des indices permis.

#### V. LECTURE D'UNE MATRICE

Cette opération de lecture consiste en le remplissage des différentes cases qui constituent la matrice.

**Exemple** : Ecrire un algorithme qui permet de lire une matrice de 20 lignes et 30 colonnes d'éléments entiers

```
Algorithme LectureMatrice
Variables :
  Mat [1..20, 1..30] : matrice d'entiers
  i, j : entier
Début
  |
  | Pour i allant de 1 à 20 Faire
  | | Pour j allant de 1 à 30 Faire
  | | | Lire (Mat [i, j])
  | | Fin pour
  | Fin pour
  |
Fin
```

#### VI. EDITION D'UNE MATRICE

L'édition d'une matrice consiste en l'affichage de ses éléments c'est-à-dire le contenu de ses différentes cases. **Exemple** : Ecrire un algorithme qui permet d'afficher une matrice de 20 lignes et 30 colonnes d'éléments entiers.

```
Algorithme éditionMatrice
Variables :
  Mat [1..20, 1..30] : matrice d'entiers
  i, j : entier
Début
  |
  | Pour i allant de 1 à 20 Faire
  | | Pour j allant de 1 à 30 Faire
  | | | écrire (Mat [i, j])
  | | Fin pour
  | Fin pour
  |
Fin
```

**Exercices :**

**Exercice1 :** Ecrire un algorithme qui lit deux matrices a et b et calcule leur somme.

**Exercice2 :** Ecrire un algorithme qui permet de trouver la valeur maximale pour chaque ligne d'une matrice de 20 lignes et 20 colonnes d'éléments réels.

**Exercice3** Ecrire un algorithme permettant de rechercher le plus grand élément d'une matrice et de retourner ses numéros de ligne et de colonne.