

## Tableaux à deux dimensions (Matrices)

### 1. Introduction

L'informatique nous offre la possibilité de déclarer des tableaux dans lesquels les valeurs ne sont pas repérées par une seule coordonnée, mais par deux.

### 2. Déclaration

Un tableau à deux dimensions (matrice) se déclare ainsi :

**Nom\_tableau : Tableau [1..nombre\_lignes , 1.. nombre\_colonnes] de Type de données ;**

Où :

Nom\_tableau : le nom de la matrice.

Nombre\_lignes : le nombre de lignes de la matrice.

Nombre\_colonnes : le nombre de colonnes de la matrice.

Type de données : le type de données des éléments de la matrice.

#### Exemple :

Mat :tableau[1..4,1..5] de entier ;

Cette déclaration veut dire qu'on demande à l'ordinateur de réserver un espace de mémoire pour 4 x 5 entiers, et quand on aura besoin de l'une de ces valeurs, on la repèrera par deux indices. Le tableau Mat peut être assimilé à une matrice constituée de 4 lignes et 5 colonnes et qui peut être représenté comme suit :

	1	2	3	4	5
1					
2					
3					
4					

### 3. Accès aux éléments d'une matrice

Pour accéder à un élément d'une matrice, on a besoin de spécifier deux indices. Le 1<sup>er</sup> correspond au numéro de la ligne et le 2<sup>ème</sup> correspond au numéro de la colonne. Ainsi, Mat[1][3] représente l'élément situé à la première ligne et la troisième colonne.

### 4. Remplissage d'une matrice

Pour remplir l'élément situé dans la i<sup>ème</sup> ligne et la j<sup>ème</sup> colonne d'une matrice « **Mat** » par une valeur saisie par l'utilisateur, on utilise l'instruction lire comme suit : **Lire (Mat[i][j]) ;**

Pour remplir une matrice, nous avons besoins deux boucles. La première permet de parcourir les lignes et la deuxième est associée aux colonnes.

Ainsi, pour remplir une matrice Mat composée de 4 lignes et 5 colonnes par des valeurs saisies par l'utilisateur, on utilise l'algorithme suivant :

<b>La boucle POUR</b>	<b>La boucle TANTQUE</b>	<b>La boucle REPETER</b>
Algorithme remplir_matrice ; Mat : tableau [1..4, 1..5] de entier ; i, j : entier ; début pour i de 1 à 4 pas 1 faire pour j de 1 à 5 pas 1 faire lire(mat[i][j]) ; fin faire fin faire fin .	Algorithme remplir_matrice ; Mat : tableau [1..4, 1..5] de entier ; i, j : entier ; début i ← 1 ; tant que (i ≤ 4) faire j ← 1 ; tant que (j ≤ 5) faire lire (mat[i][j]) ; j ← j + 1 ; fin faire i ← i + 1 ; fin faire fin .	Algorithme remplir_matrice ; Mat : tableau [1..4, 1..5] de entier ; i, j : entier ; début i ← 1 ; Répéter j ← 1 ; Répéter lire (mat[i][j]) ; j ← j + 1 ; jusqu'à (j > 5); i ← i + 1; jusqu'à (i > 4); fin .

### 5. Affichage d'un tableau à deux dimensions

Pour afficher l'élément situé dans la i<sup>ème</sup> ligne et la j<sup>ème</sup> colonne d'une matrice « **Mat** », on utilise l'instruction écrire comme suit : **écrire (Mat[i][j])** ;

Ainsi, pour afficher le contenu d'une matrice Mat composée de 4 lignes et 5 colonnes, on utilise l'algorithme suivant :

<b>La boucle POUR</b>	<b>La boucle TANTQUE</b>	<b>La boucle REPETER</b>
Algorithme afficher_matrice ; Mat : tableau [1..4, 1..5] de entier ; i, j : entier ; début pour i de 1 à 4 pas 1 faire pour j de 1 à 5 pas 1 faire écrire(mat[i][j]) ; fin faire fin faire fin .	Algorithme afficher_matrice ; Mat : tableau [1..4, 1..5] de entier ; i, j : entier ; début i ← 1 ; tant que (i ≤ 4) faire j ← 1 ; tant que (j ≤ 5) faire écrire (mat[i][j]) ; j ← j + 1 ; fin faire i ← i + 1 ; fin faire fin .	Algorithme afficher_matrice ; Mat : tableau [1..4, 1..5] de entier ; i, j : entier ; début i ← 1 ; Répéter j ← 1 ; Répéter écrire (mat[i][j]) ; j ← j + 1 ; jusqu'à (j > 5); i ← i + 1; jusqu'à (i > 4); fin .

### Exercices d'application

- 1- Ecrire un algorithme qui remplit une matrice d'entiers de 5 lignes et 5 colonnes. L'algorithme doit aussi afficher la somme des éléments de la diagonale.
- 2- Ecrire un algorithme qui remplit une matrice d'entiers de 5 lignes et 3 colonnes et affiche la somme des éléments de chaque ligne.
- 3- Ecrire un algorithme qui remplit une matrice d'entiers de 5 lignes et 3 colonnes et affiche la matrice transposée.