

Atelier : Les Tableaux

Objectifs :

Apprendre la programmation avec les tableaux à une et à deux dimensions.

Exercice 1:

Ecrire un programme qui lit la dimension **N** d'un tableau **T** du type **int** (dimension maximale: 50 composantes), remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

- Calculer et afficher ensuite la somme des éléments du tableau.
- Calculer et afficher la somme des éléments positifs et la somme des éléments négatifs du tableau.

Exercice 2:

Ecrire un programme qui lit la dimension **N** d'un tableau **T** du type **int** (dimension maximale: 50 composantes), remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Copiez ensuite toutes les composantes strictement positives dans un deuxième tableau **TP** et toutes les valeurs strictement négatives dans un troisième tableau **TN**. Afficher les tableaux **TP** et **TN**.

Exercice 3 :

Ecrire un programme qui calcule le produit scalaire de deux vecteurs d'entiers **U** et **V** (de même dimension).

$$(3 \ 2 \ -4) * (2 \ -3 \ 5) = 3*2 + 2*(-3) + (-4)*5 = -20$$

Exercice 4 :

Etant donné un tableau d'entiers, écrire un programme qui détermine le nombre d'éléments strictement supérieurs à la valeur moyenne du tableau.

Exercice 5:

Ecrire un programme qui détermine la plus grande et la plus petite valeur dans un tableau d'entiers **A**. Afficher ensuite la valeur et la position du maximum et du minimum. Si le tableau contient plusieurs maxima ou minima, le programme retiendra la position du premier maximum ou minimum rencontré.

Exercice 6:

Ecrire un programme qui permet de vérifier si un tableau d'entiers **T** saisi au clavier est symétrique ou non.

Remarque : Un tableau est dit symétrique s'il est lit de la même façon droite-gauche ou gauche-droite.

Exemple :

T : 5 0 0 0 5 est symétrique

T : 2 3 4 8 2 n'est pas symétrique

Exercice 7:

Ecrire un programme qui lit les dimensions **L** et **C** d'un tableau **M** à deux dimensions du type **int** (dimensions maximales: 50 lignes et 50 colonnes). Remplir le tableau par des valeurs entrées au clavier et afficher le tableau ainsi que la somme de tous ses éléments.

Exercice 8:

Ecrire un programme qui transfère un tableau **M** à deux dimensions **L** et **C** (dimensions maximales: 10 lignes et 10 colonnes) dans un tableau **V** à une dimension **L*C**.

```
a b c d
e f g h ==> ( a b c d e f g h i j k l )
i j k l
```

Exercice 9:

Ecrire un programme C qui lit la dimension **N** d'une matrice d'entier **M** carrée (au maximum 10*10) et de saisir ensuite les éléments de cette matrice et un entier **x** et afficher à la fin le nombre d'occurrence de **x** dans **M**.

Exemple: N=3

```
M = | 1 2 3 | Pour x=2, le programme affiche :
     | 4 5 2 | Nombre d'occurrence de x dans M = 3
     | 2 8 9 |
```

Exercice 10:

Ecrire un programme C qui permet d'afficher la transposée d'une matrice d'entiers **M** saisi au clavier.

Exemple : Soit **M** une matrice d'entiers et **M1** sa transposée.

$$M = \begin{pmatrix} 5 & 3 & 8 \\ 4 & 2 & 1 \\ 0 & 3 & 7 \end{pmatrix} \quad M1 = \begin{pmatrix} 5 & 4 & 0 \\ 3 & 2 & 3 \\ 8 & 1 & 7 \end{pmatrix}$$

Annexe : LES TABLEAUX

1. Les tableaux à une dimension

a. Déclaration de tableaux en C

TypeSimple NomTableau[<Dimension>;

Exemples : int A[25]; float B[100]; int C[10];

b. Initialisation et réservation automatique

- *Initialisation*

Lors de la déclaration d'un tableau, on peut initialiser les composantes du tableau, en indiquant la liste des valeurs respectives entre accolades.

Exemples :

```
int A[5] = {10, 20, 30, 40, 50};
int C[10] = {1, 0, 0, 1, 1, 1, 0, 1, 0, 1};
```

Il faut évidemment veiller à ce que le nombre de valeurs dans la liste corresponde à la dimension du tableau. Si la liste ne contient pas assez de valeurs pour toutes les composantes, les composantes restantes sont initialisées par zéro.

- *Réservation automatique*

Si la dimension n'est pas indiquée explicitement lors de l'initialisation, alors l'ordinateur réserve automatiquement le nombre d'octets nécessaires.

Exemples

```
int A[] = {10, 20, 30, 40, 50};
```

==> réservation de **5*sizeof(int)** octets (dans notre cas: 20 octets)

c. Accès aux composantes

- l'accès au premier élément du tableau se fait par **T[0]**(T est le nom de tableau).
- l'accès au dernier élément du tableau se fait par **T[N-1]**

2. Les tableaux à deux dimensions

a. Déclaration de tableaux à deux dimensions en C

TypeSimple NomTabl [DimLigne][DimCol];

Exemples : int A[10][10];

```
float B[2][20];
```

b. Initialisation et réservation automatique

- *Initialisation*

Lors de la déclaration d'un tableau, on peut initialiser les composantes du tableau, en indiquant la liste des valeurs respectives entre accolades. A l'intérieur de la liste, les composantes de chaque ligne du tableau sont encore une fois comprises entre accolades. Pour améliorer la lisibilité des programmes, on peut indiquer les composantes dans plusieurs lignes.

Exemples :

```
int A[3][10] = {{ 0,10,20,30,40,50,60,70,80,90},
               {10,11,12,13,14,15,16,17,18,19},
               { 1,12,23,34,45,56,67,78,89,90}};
```

- *Réservation automatique*

Si le nombre de **lignes L** n'est pas indiqué explicitement lors de l'initialisation, l'ordinateur réserve automatiquement le nombre d'octets nécessaires.

```
int A[][10] = {{ 0,10,20,30,40,50,60,70,80,90},
               {10,11,12,13,14,15,16,17,18,19},
               { 1,12,23,34,45,56,67,78,89,90}};
```

Réservation de $3 * 10 * 4 = 120$ octets

c. Accès aux composantes

NomTableau[Ligne][Colonne]

Considérons un tableau A de dimensions L et C.

- les indices du tableau varient de **0** à **L-1**, respectivement de **0** à **C-1**.
- la composante de la N^{ième} ligne et M^{ième} colonne est notée: **A[N-1][M-1]**