

## Travaux Pratiques N°6

### Les tableaux à deux dimensions

#### Objectifs

- Manipuler des tableaux à deux dimensions
- Déclaration de tableaux à deux dimensions
- Initialisation de tableaux à deux dimensions

#### I. Déclaration

##### Syntaxe:

<type simple> <Nom tableau> [<dimlig>] [<dimcol>] ;

##### Exemples :

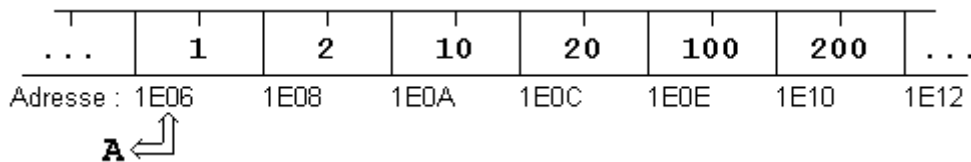
```
int A [10][10] ;
float B [2][10] ;
char C [10][20] ;
```

#### II. Mémorisation

Comme pour les tableaux à une dimension, le nom d'un tableau est le représentant de *l'adresse du premier élément* du tableau (c.-à-d. l'adresse de la première **ligne** du tableau). Les composantes d'un tableau à deux dimensions sont stockées ligne par ligne dans la mémoire.

##### Exemples :

```
int A [3][2] = {{1,2},{10,20},{100,200}};
```



```
int B[3][4] = {{1,2,3,4},{10,20,30,40},{100,200,300,400}};
```

Réservation de  $3 \times 4 \times 2 = 24$  octets

##### Remarque :

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 B [ ] [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  $3 \times 10 \times 2 = 60$  octets

#### III. Accès aux composantes d'une matrice

Considérons un tableau A de dimensions **L**(lignes)et **C**(colonnes).

- ❖ Les indices du tableau varient de **0** à **L-1**, respectivement de **0** à **C-1**.
- ❖ la composante de la N<sup>ième</sup> ligne et M<sup>ième</sup> colonne est notée:

**A[N-1][M-1]**

#### IV. Chargement et affichage des éléments d'une matrice

```

void main(){
    int M [50][100];
    int NL, LC ;
    int i, j;
    do{
        printf ("donner le nombre de lignes \n") ;
        scanf("%d", &NL);
    } while (NL<= 0 || NL>50) ;
    do{
        printf ("donner le nombre de colonnes \n") ;
        scanf("%d", &NC);
    } while (NC<= 0 || NC>100) ;

    /* remplissage des éléments */
    for (i=0; i<NL; i++)
        for (j=0; j<NC; j++) {
            printf ("M[%d][%d]:", i, j);
            scanf ("%d",&M[i][j]);
        }
    }
    /* affichage des éléments */
    for (i=0; i<NL; i++) {
        for (j=0; j<NC; j++)
            printf("%7d", M[i][j]);
        printf("\n"); /* Retour à la ligne */
    }
}

```

#### VI. Travail demandé

##### Exercice 1

Ecrire un programme qui lit les dimensions L et C d'un tableau T 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 2

Ecrire un programme qui lit les dimensions L et C d'un tableau T 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 chaque ligne et de chaque colonne en n'utilisant qu'une variable d'aide pour la somme.

##### Exercice 3

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.

**Exemple:**

```

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

```