

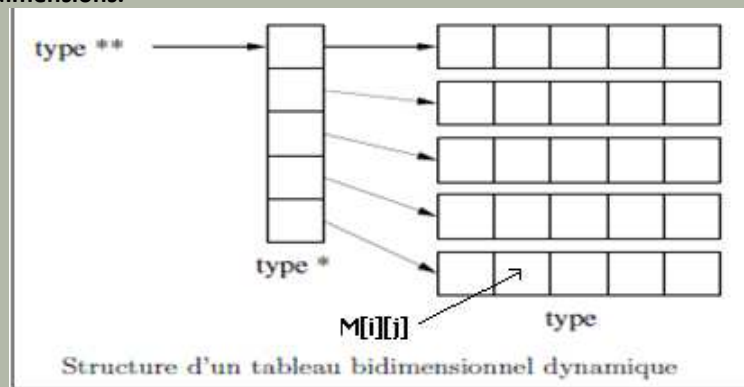
## SÉRIE N° 8 : ALLOCATION DYNAMIQUE DE LA MÉMOIRE(SUITE)

### Allocation de tableaux multidimensionnels (Matrice)

En C, l'allocation dynamique d'un tableau multidimensionnel se fait en deux étapes :

- ✓ allocation d'un tableau de pointeurs dont la taille correspond au nombre de lignes souhaité
- ✓ allocation de chacune des lignes en utilisant le tableau de pointeurs précédent pour stocker leur adresse

Ensuite, l'accès au tableau se fait comme d'habitude en C, en utilisant l'opérateur [ ] pour préciser l'indice dans chacune des dimensions.



### Problème : Matrice creuse

Une matrice creuse est une matrice contenant plus de 50% de valeurs nulles. La représentation d'une matrice par un tableau à 2 dimensions, conduit à une occupation inutile de l'espace mémoire par des 0.

Dans ce problème, on travaille sur d'autres moyens de représentation des matrices creuses.

Soit une matrice d'entiers M de taille NxN déclarée et initialisée dans la fonction main().

N est une constante globale.

#### Exemple : matrice M de taille 5x5 :

5	4	0	0	1
0	5	0	0	0
0	1	0	7	0
0	0	0	0	0
0	0	3	0	0

**Proposition 1 :**

La proposition 1 consiste à représenter la matrice  $M$  par deux tableaux  $T_v$  et  $T_n$ . Le tableau  $T_v$  contient les couples : valeur non nulle de  $M$  et son numéro de colonne. Le tableau  $T_n$  contient le nombre de valeurs non nulles de  $M$  par ligne.

**Exemple : D'après l'exemple précédent**

<b>Tv est :</b>	<table border="1"><tr><td>5</td><td>4</td><td>1</td><td>5</td><td>1</td><td>7</td><td>3</td></tr><tr><td>0</td><td>1</td><td>4</td><td>1</td><td>1</td><td>3</td><td>2</td></tr></table>	5	4	1	5	1	7	3	0	1	4	1	1	3	2	<b>et Tn est :</b>	<table border="1"><tr><td>3</td><td>1</td><td>2</td><td>0</td><td>1</td></tr></table>	3	1	2	0	1
5	4	1	5	1	7	3																
0	1	4	1	1	3	2																
3	1	2	0	1																		

1. Définir le type du tableau de structure **Tv**.
2. Sans programmer, que pouvez-vous dire sur la dimension du tableau  $T_v$  et du tableau  $T_n$  ?

On cherche à transformer  $M$  en  $T_v$  et  $T_n$ .

3. Ecrire une fonction **nombreV(M)** qui donne le nombre de valeurs non nulles de  $M$ .
4. Ecrire la fonction **remplirTv(M)** qui remplit et retourne le tableau **Tv** obtenu à partir de  $M$ .
5. Ecrire la fonction **remplirTn(M)** qui remplit et retourne le tableau **Tn** obtenu à partir de  $M$ .

On suppose que la taille de **Tv** est **H** (entier déclaré globale)

On veut exploiter la matrice  $M$  à travers sa représentation équivalente **Tv** et **Tn**.

Répondre aux questions suivantes à partir de **Tv** et **Tn** au lieu de  $M$ .

6. Ecrire la fonction **sommeC1(Tv, Tn, j)** qui retourne la somme de la colonne  $j$ .
7. Ecrire la fonction **sommeL1(Tv, Tn, i)** qui retourne la somme de la ligne  $i$ .
8. Ecrire la fonction **valeur1(Tv, Tn, i, j)** qui donne la valeur de  $M[i][j]$ .

On veut afficher les valeurs de la ligne  $i$  de  $M$ . Si  $i = 2$  alors on affiche 0 1 0 7 0

9. Ecrire la fonction **afficheL1(Tv, Tn, i)** qui affiche les valeurs de la ligne  $i$  de  $M$ .

**Proposition 2 :** Représentation par une matrice de structure dynamique (**SÉRIE N° 9**).