

Les tableaux à deux dimensions

1. Écrire une fonction *void lire_matrice(int n, int m, int M[n][m])* qui permet de remplir une matrice *M* d'entiers de *n* lignes, *m* colonnes.
2. Écrire une fonction *void ecrire_matrice(int n, int m, int M[n][m])* qui permet d'afficher une matrice *M* d'entiers de *n* lignes, *m* colonnes.
3. Donner la fonction rotation permettant d'effectuer la transformation d'une matrice carrée d'ordre *n* de la manière suivante :

Exemple :

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \text{ devient } \begin{pmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{pmatrix}$$

4. Écrire une fonction *int produit(int n, int m, int p, int q, int A[n][m], int B[p][q], int C[n][q])* qui permet de calculer la matrice *C* résultat du produit matriciel de *A* de dimensions (n, m) et *B* de dimensions (p, q) .

Le produit matriciel de $A(n, m)$ et $B(p, q)$ ne s'effectue que si $m = p$. Le résultat est une matrice *C* de dimensions (n, q) telle que, $C_{ij} = \sum_{k=1..m} A_{ik} \times B_{kj}$ pour $i = 1..n$ et $j = 1..q$.

NB : La fonction doit retourner 0 lorsque les dimensions des deux matrices sont incompatibles, 1 sinon.

5. Écrire un programme permettant de chercher dans une matrice donnée les éléments qui sont à la fois un maximum sur leur ligne et un minimum sur leur colonne. Ces éléments sont appelés des points-cols. Afficher les positions et les valeurs de tous les points-cols trouvés.

Exemple :

$$\begin{pmatrix} 10 & 18 & 2 & 9 \\ 2 & 16 & 5 & 4 \\ 9 & 13 & 8 & 7 \\ 9 & 22 & 1 & 3 \\ 100 & 92 & 7 & 5 \end{pmatrix}$$

Le programme affichera : point-col 13 : (2,1)