

TP– Initiation à la Programmation

Série de TP N°3 – Tableaux à deux dimensions - Matrices

Exercice N°01:

Soit l'algorithme suivant:

Algorithme Exercice1;

Variables

A, B : tableau[0..99,0..99] de réel;

N,M,i, j : entier;

Début

//Entrées

Écrire("Donner la taille N et M de la matrice A: ");

Lire(N,M);

Écrire("Donner les composantes de la matrice A: ");

Pour i←0 à N-1 **faire**

Pour j←0 à M-1 **faire**

 Lire(A[i, j]);

FinPour

FinPour

//Traitement

Pour i←0 à N-1 **faire**

Pour j←0 à M-1 **faire**

 B[i,j] ← A[i,j]*A[i,j] ;

FinPour ;

FinPour ;

//Sorties

Écrire("Affichage de la matrice B: ");

Pour i←0 à N-1 **faire**

Pour j←0 à M-1 **faire**

 Écrire (B[i, j]);

FinPour

FinPour

Fin.

Questions :

1. Traduire l'algorithme en Programme C.

2. Compiler et exécuter le programme pour les valeurs de N, Met A suivantes:

$$N= M = 2 \text{ et } A = \begin{bmatrix} 3 & 8 \\ -4 & 7 \end{bmatrix}$$

3- Dérouler le programme pour les valeurs de N, M et A ci-dessus ?

4- Dédurre ce que fait le programme?

Exercice N°02 :

Écrire un programme en langage C qui permet de calculer un vecteur Y, résultat du produit d'une matrice A par un vecteur X d'entiers.

Rappel : Pour chaque indice i, on a :

$Y[i] = \sum A[i, j] * X[j]$, c'est-à-dire que le produit matrice-vecteur consiste à multiplier chaque ligne de la matrice A par le vecteur X, puis à additionner les produits pour obtenir un élément du vecteur Y comme résultat.

Exercice N°03 :

Soit **T** une matrice de type entier de taille $N \times M$.

Écrire un algorithme/programme C qui permet de rechercher le **plus grand élément** dans la matrice **T** ainsi que sa position (**ligne et colonne**).

TP – Initiation à la Programmation

Série de TP N°3 – Exercices supplémentaires sur les Matrices

Exercice Sup01: Somme, Moyenne et Produit des éléments d'une matrice

Soit une matrice A réelle d'ordre $N \times M$.

1. Écrire un algorithme/programme **en C** qui calcule la somme et la moyenne des éléments de la matrice A .
2. Écrire un algorithme/programme **en C** qui permet de calculer la somme de chaque ligne et le produit de chaque colonne.

Exercice Sup02: Produit de deux matrices

Soit A et B deux matrices carrées d'ordre N et M .

Écrire un algorithme/programme **en C** qui permet de calculer le produit de A et B .

Exercice Sup03: La recherche d'une valeur dans une matrice

Soit A une matrice de type réel de taille $N \times M$.

Écrire un algorithme/programme **en C** qui permet de rechercher si une valeur réelle X existe ou non dans la matrice A . Dans le cas où X existe dans A , on affiche aussi sa position (numéro de ligne et de colonne).

Exercice Sup04: Le nombre d'éléments positives, négatives et nuls dans une matrice

Soit A une matrice réelle d'ordre $N \times M$.

Écrire un algorithme/programme **en C** qui permet de rechercher le nombre d'éléments positives, négatives et nuls de la matrice A .

Exercice Sup05: Rotation d'une Matrice

Écrire un programme **en C** qui permet de lire une matrice carrée A et d'effectuer une rotation de 90° degrés dans le sens horaire et d'afficher la matrice B après la rotation.

Exercice Sup06: Conversion d'une matrice en un tableau unidimensionnel

Écrire un programme **en C** qui transforme une matrice M de taille $n \times m$ en un tableau T unidimensionnel de taille $n * m$. Demande à l'utilisateur de saisir la matrice, puis affichez le tableau résultant.

Exemple: Pour $n=3$, $m=3$ et $M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, on affiche $T = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$ de $n * m = 3 * 3 = 9$ éléments