

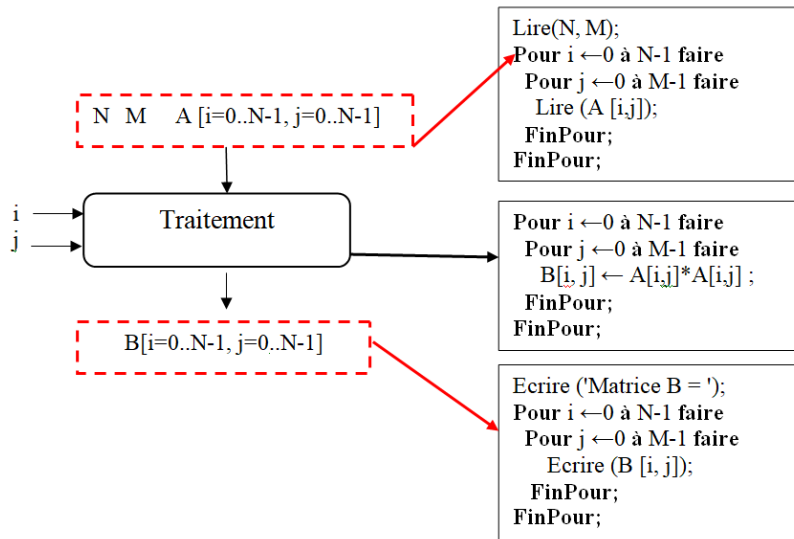
TP– Initiation à la Programmation

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

Exercice N°01:

Les variables d'entrée, variables de sortie et la partie traitement sont présentées dans le schéma suivant :

1. Traduire l'algorithme en Programme C.

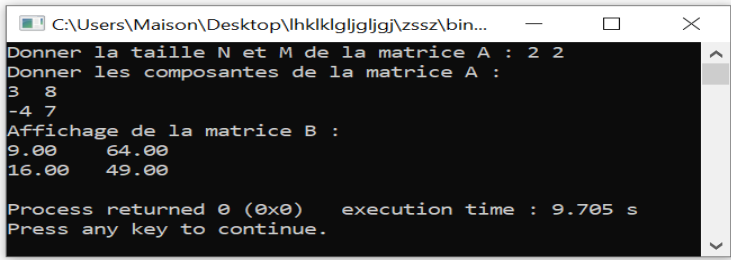


| Algorithme | Programme en C |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Algorithme Exercice1; Variabes A , B : tableau[0..99][0..99] deréel; i, j, N,M : entier; Début //Entrées Écrire("Donnerla taille Net 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]);</p> | <pre> #include <stdio.h> int main(){ int N, M,i, j; float A[100][100], B[100][100]; /* Entrées */ printf("Donner la taille N et M de la matrice A : "); scanf("%d%d", &N, &M); printf("Donner les composantes de la matrice A :\n"); for(i=0;i< N;i++){ for(j =0; j < M;j++){ scanf("%f", &A[i][j]); } } /* Traitement */ for(i=0;i< N;i++){ for(j =0; j < M;j++){ B[i][j]= A[i][j]* A[i][j]; } } /* Sorties */ printf("Affichage de la matrice B :\n"); for(i=0;i< N;i++){ for(j =0; j < M;j++){ printf("%.2f\t", B[i][j]); // \t insère une tabulation qui permet d'aligner les colonnes lors de l'affichage de la matrice. } } </pre> |

| | |
|-------------------------------------------------|---------------------------------------------------------|
| FinPour FinPour Fin. | <pre>printf("\n"); // saut de ligne } return 0; }</pre> |
|-------------------------------------------------|---------------------------------------------------------|

2. Compiler et exécuter le programme pour les valeurs de N, M et A suivantes : $N = 2, M = 2$ et $A = \begin{bmatrix} 3 & 8 \\ -4 & 7 \end{bmatrix}$

```
*main.c x
1 #include <stdio.h>
2 int main() {
3     int N, M, i, j;
4     float A[100][100], B[100][100];
5     /* Entrées */
6     printf("Donner la taille N et M de la matrice A : ");
7     scanf("%d%d", &N, &M);
8
9     printf("Donner les composantes de la matrice A :\n");
10    for(i=0; i< N; i++){
11        for(j =0; j < M; j++){
12            scanf("%f", &A[i][j]);
13        }
14    }
15    /* Traitement */
16    for(i=0; i< N; i++){
17        for(j =0; j < M; j++){
18            B[i][j]= A[i][j]* A[i][j];
19        }
20    }
21    /* Sorties */
22    printf("Affichage de la matrice B :\n");
23    for(i=0; i< N; i++){
24        for(j =0; j < M; j++){
25            printf("%.2f\t", B[i][j]);
26            // \t insère une tabulation qui permet d'aligner les colonnes lors de l'affichage de la matrice.
27        }
28        printf("\n"); // saut de ligne
29    }
30    return 0;}
31
```



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

| Instructions | Variables | | | | | | Affichages |
|--------------------------------------------------------------------------------------------------------------|-----------|---|--------|--------|-------------------------------------------------|---------------------------------------------------|----------------------------------------------------------|
| | N | M | i | j | A | B | |
| Écrire("Donner la taille Net M de la matrice A : "); | / | / | / | / | / | / | Donner la taille Net M de la matrice A : |
| Lire (N,M); | 2 | 2 | / | / | / | / | / |
| Écrire("Donner les composantes de la matrice A : "); | 2 | 2 | / | / | / | / | 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]); | 2 | 2 | 0 1 | 0 1 | $\begin{bmatrix} 3 & 8 \\ -4 & 7 \end{bmatrix}$ | / | / |
| Pour i←0 Pour j←0 $B[i,j] \leftarrow A[i,j]*A[i,j]$; $B[0,0] \leftarrow A[0,0]*A[0,0] = 3 *3 = 9$ | 2 | 2 | 0 | 0 | $\begin{bmatrix} 3 & 8 \\ -4 & 7 \end{bmatrix}$ | $\begin{bmatrix} 9 & \\ & \end{bmatrix}$ | / |
| Pour i←0 Pour j←1 $B[i,j] \leftarrow A[i,j]*A[i,j]$; $B[0,1] \leftarrow A[0,1]*A[0,1] = 8 *8 = 64$ | 2 | 2 | 0 | 1 | $\begin{bmatrix} 3 & 8 \\ -4 & 7 \end{bmatrix}$ | $\begin{bmatrix} 9 & 64 \\ & \end{bmatrix}$ | / |
| Pour i←1 Pour j←0 $B[i,j] \leftarrow A[i,j]*A[i,j]$; $B[1,0] \leftarrow A[1,0]*A[1,0] = -4 *-4 = 16$ | 2 | 2 | 1 | 0 | $\begin{bmatrix} 3 & 8 \\ -4 & 7 \end{bmatrix}$ | $\begin{bmatrix} 9 & 64 \\ 16 & \end{bmatrix}$ | / |
| Pour i←1 Pour j←1 $B[i,j] \leftarrow A[i,j]*A[i,j]$; $B[1,1] \leftarrow A[1,1]*A[1,1] = 7 *7 = 49$ | 2 | 2 | 1 | 1 | $\begin{bmatrix} 3 & 8 \\ -4 & 7 \end{bmatrix}$ | $\begin{bmatrix} 9 & 64 \\ 16 & 49 \end{bmatrix}$ | / |
| Écrire("Affichage de la matrice B : "); Pour i←0 à N-1 faire Pour j←0 à M-1 faire Écrire (B[i, j]); | 2 | 2 | 1 | 1 | $\begin{bmatrix} 3 & 8 \\ -4 & 7 \end{bmatrix}$ | $\begin{bmatrix} 9 & 64 \\ 16 & 49 \end{bmatrix}$ | Affichage de la matrice B : 9.00 64.00 16.00 49.00 |

4. Dédire ce que fait le programme ?

Le programme crée une matrice B où chaque élément est le carré de l'élément correspondant dans la matrice A.

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.à.d 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.

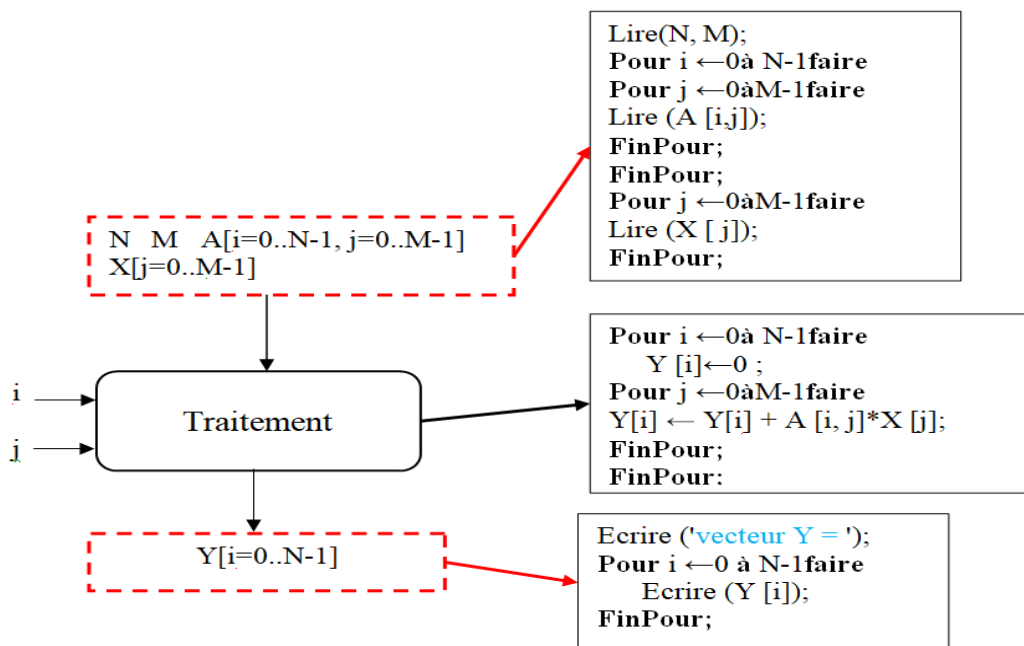
N.B : Pour effectuer le produit matrice-vecteur il faut que : le nombre de colonne de la matrice = le nombre de composantes du vecteur

Solution :

Les étapes à suivre :

- Lire les dimensions **N** et **M** de la matrice **A**.
- Lire les composantes de la matrice **A** à l'aide de deux boucles **for**.
- Lire les éléments du vecteur **X** à l'aide de la boucle **for**.
- Pour chaque ligne i de la matrice **A** :
 - Initialiser **Y[i]** à 0.
 - Multiplier chaque élément de la ligne $A[i, j]$ par $X[j]$ et additionner les produits.
- Afficher le vecteur résultat **Y**.

Les variables d'entrée, variables de sortie et la partie traitement sont présentées dans le schéma suivant :



Programme en C

```
#include <stdio.h>

int main(){
int N, M, i, j;
int A[100][100], X[100], Y[100];

/* Entrées */
printf("Donner la taille N et M de la matrice A : ");
scanf("%d%d", &N, &M);

printf("Donner les elements de la matrice A : \n");
for(i=0; i< N; i++){
for(j =0; j < M; j++){
scanf("%d", &A[i][j]);
}
}
```

```

}

printf("Donner les elements du vecteur X :\n");
for(j =0; j < M;j++){
    scanf("%d",&X[j]);
}

/* Traitement : */
for(i=0;i< N;i++){
    Y[i]=0;
    for(j =0; j < M;j++){
        Y[i]+= A[i][j]* X[j]; /* attention : X[j] et pas X[i] */
    }
}

/* Sorties */
printf("Vecteur Y =[ ");
for(i=0;i< N;i++){
    printf("%d ", Y[i]);
}
printf("]");

return 0;
}

```

Compiler et exécuter le programme pour : $N = 2, M = 3, A = \begin{bmatrix} 1 & 2 & 7 \\ 9 & 6 & 5 \end{bmatrix}$ et $X = [4 \ 3 \ 7]$

The image shows a code editor window with a C program named 'main.c' and a terminal window showing its execution. The code in the editor is as follows:

```

main.c x
1 #include <stdio.h>
2 int main(){
3     int N, M, i, j;
4     int A[100][100], X[100], Y[100];
5     /* Entrées */
6     printf("Donner la taille N et M de la matrice A : ");
7     scanf("%d%d",&N,&M);
8
9     printf("Donner les elements de la matrice A :\n");
10    for(i=0;i< N;i++){
11        for(j =0; j < M;j++){
12            scanf("%d",&A[i][j]);
13        }
14    }
15    printf("Donner les elements du vecteur X :\n");
16    for(j =0; j < M;j++){
17        scanf("%d",&X[j]);
18    }
19    /* Traitement : */
20    for(i=0;i< N;i++){
21        Y[i]=0;
22        for(j =0; j < M;j++){
23            Y[i]+= A[i][j]* X[j]; /* attention : X[j] et pas X[i] */
24        }
25    }
26    /* Sorties */
27    printf("Vecteur Y =[ ");
28    for(i=0;i< N;i++){
29        printf("%d ", Y[i]);
30    }printf("]");
31    return 0;}
32

```

The terminal window shows the following output:

```

C:\Users\Maison\Desktop\lhkklgljgjljg\zssz\bin\Deb...
Donner la taille N et M de la matrice A : 2 3
Donner les elements de la matrice A :
1 2 7
9 6 5
Donner les elements du vecteur X :
4 3 7
Vecteur Y =[ 59 89 ]
Process returned 0 (0x0) execution time : 25.478 s
Press any key to continue.

```

Exercice N°03 :

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

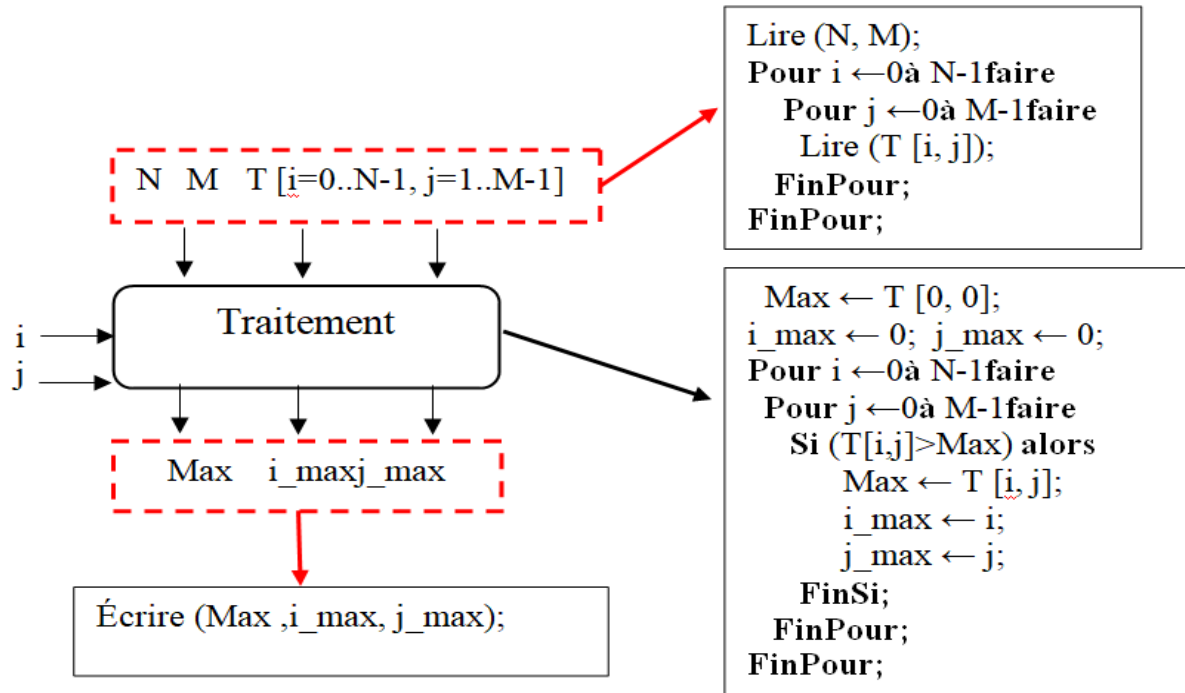
Ecrire un algorithme/programme C qui permet de rechercher le **plus grand élément** dans la matrice T ainsi que sa position (*ligne et colonne*).

Solution :

Les étapes à suivre :

- Initialiser la valeur maximale **Max** à **T[0,0]** et les indices **i_max** et **j_max** à **0**.
- Parcourir la matrice avec deux boucles **for** pour comparer chaque élément de la matrice **T** avec **Max**.
- Si un élément plus grand est trouvé, mettre à jour la variable **Max** ainsi que les indices **i_max** et **j_max**.
- Afficher la composante maximale de la matrice **T** et ses coordonnées.

Les variables d'entrée, variables de sortie et la partie traitement sont présentées dans le schéma suivant :



| Algorithme | Programme en C |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>Algorithme Exercice2; Variables T : Tableau [0..99, 0..99] d'entier; i, j, i_max, j_max, N, M: entier; Max : entier; Début /*-*- Entrées *-* Ecrire ('Donner le nombre des lignes N et des colonnes M de la matrice T : '); Lire (N, M); Ecrire ('Donner les composantes de la matrice T : '); Pour i ← 0 à N-1 faire Pour j ← 0 à M-1 faire Lire (T [i, j]); FinPour; FinPour; /*-*- Traitement *-* Max ← T [0, 0]; i_max ← 0; j_max ← 0; Pour i ← 0 à N-1 faire Pour j ← 0 à M-1 faire</pre> | <pre>#include <stdio.h> int main() { int T[100][100]; int i, j, i_max, j_max, N, M; int Max; /*-*- Entrées *-* printf("Donner le nombre des lignes N et des colonnes M de la matrice T : "); scanf("%d %d", &N, &M); printf("Donner les composantes de la matrice T :\n"); for (i = 0; i < N; i++) { for (j = 0; j < M; j++) { scanf("%d", &T[i][j]); } } /*-*- Traitement *-* Max = T[0][0]; i_max = 0; j_max = 0; for (i = 0; i < N; i++) {</pre> |

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> Si (Max < T[i,j]) alors Max ← T [i, j]; i_max ← i; j_max ← j; FinSi; FinPour; FinPour; <i>/*-*-* Sorties *-** */</i> Ecrire ('Max = ', Max , ', situe à la ligne ', i_max, ' et à la colonne ', j_max); Fin. </pre> | <pre> for (j = 0; j < M; j++) { if (Max < T[i][j]) { Max = T[i][j]; i_max = i; j_max = j; } } <i>/*-*-* Sorties *-** */</i> printf("Max = %d, situe a la ligne %d et a la colonne %d\n", Max, i_max, j_max); return 0; } </pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Compiler et exécuter le programme pour : $N = 2$, $M = 3$ et $T = \begin{bmatrix} 8 & -7 & 20 \\ 5 & 0 & -3 \end{bmatrix}$

The image shows a C program in a text editor and its execution in a terminal window. The code in the editor is as follows:

```

main.c X
1 #include <stdio.h>
2 int main()
3 {
4     int T[100][100];
5     int i, j, i_max, j_max, N, M;
6     int Max;
7     /*-*-** Entrées *-** */
8     printf("Donner le nombre des lignes N et des colonnes M de la matrice T : ");
9     scanf("%d %d", &N, &M);
10    printf("Donner les composantes de la matrice T :\n");
11    for (i = 0; i < N; i++) {
12        for (j = 0; j < M; j++) {
13            scanf("%d", &T[i][j]);
14        }
15    }
16    /*-*-** Traitement *-** */
17    Max = T[0][0];
18    i_max = 0;
19    j_max = 0;
20    for (i = 0; i < N; i++) {
21        for (j = 0; j < M; j++) {
22            if (Max < T[i][j]) {
23                Max = T[i][j];
24                i_max = i;
25                j_max = j;
26            }
27        }
28    }
29    /*-*-** Sorties *-** */
30    printf("Max = %d, situe a la ligne %d et a la colonne %d\n", Max, i_max, j_max);
31    return 0;
32 }
33

```

The terminal window shows the following output:

```

C:\Users\Maison\Desktop\lhkklgjljg\zssz\bin\Debug\zssz.exe
Donner le nombre des lignes N et des colonnes M de la matrice T : 2 3
Donner les composantes de la matrice T :
8 -7 20
5 0 -3
Max = 20, situe a la ligne 0 et a la colonne 2

Process returned 0 (0x0)   execution time : 50.768 s
Press any key to continue.

```