

TP - Programmation

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

**Exercice N°01 :** Algorithme → Programme C

Soit l'algorithme suivant :

**Algorithme** Matrice ;

**Variables**

A : Tableau [0..99, 0..99] de réel;

i, j, N : entier;

S, M : réel;

**Début**

// Entrées

Ecrire('Donner la taille de la matrice carrée A :');

Lire(N);

Ecrire('Donner les composantes de la matrice A :');

**Pour** i ← 0 à N-1 **faire**

**Pour** j ← 0 à N-1 **faire**

        Lire(A[i,j]);

**FinPour**;

**FinPour**;

// Traitement

S ← 0;

**Pour** i ← 0 à N-1 **faire**

    S ← S+A[i,i] ;

**FinPour**;

M ← S/N ;

// Sorties

Ecrire('S=', S, 'M=', M);

**Fin.**

**Questions :**

1- Traduire l'algorithme en Programme C.

2- Compiler et exécuter le programme pour :

N = 3 et

$$A = \begin{bmatrix} 2 & 4 & -4 \\ 4 & 8 & 1 \\ 3.5 & 9 & 4 \end{bmatrix}$$

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

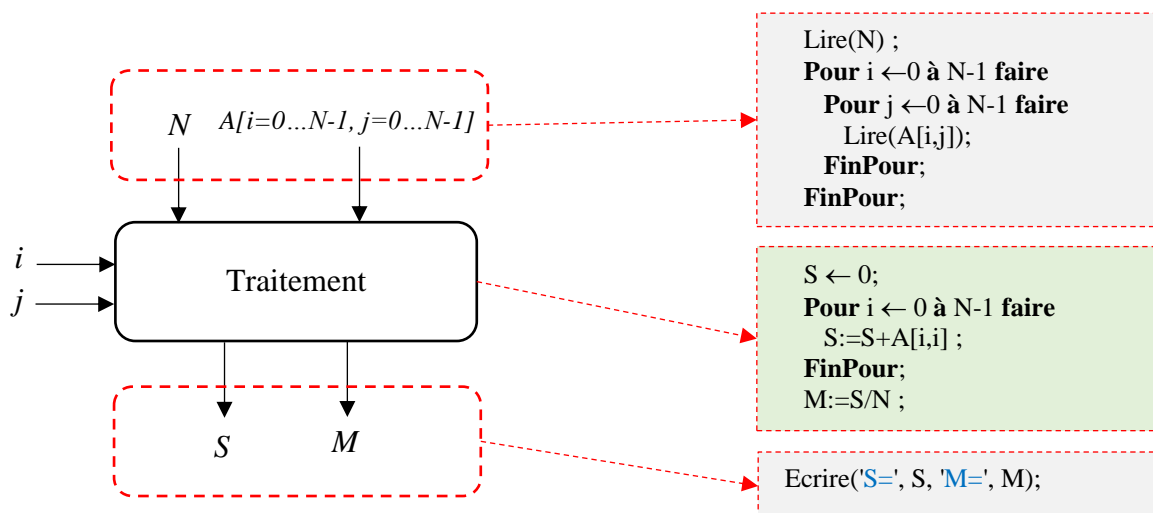
4- Déduire ce que fait le programme ?

5- Ré-écrire le programme en remplaçant la boucle *Pour* par la boucle *Tantque* dans la partie des entrées.

6- Ré-écrire le programme en remplaçant la boucle *Pour* par la boucle *Répéter* dans la partie des entrées.

**Solution :**

Les variables d'entrée, variable de sortie et la partie traitement sont présentées dans le schéma ci-dessous :



**Remarque :**

Les variables  $i$  et  $j$  sont des variables de traitement ou intermédiaires, utilisées pour parcourir la matrice A.

**1 - Algorithme/programme C :**

| Algorithme  | Programme C   |
|---|---|
| <p><b>Algorithme</b> Matrice ;</p> <p><b>Variables</b></p> <p>A : Tableau [0..99, 0..99] de réel;</p> <p>i, j, N : entier;</p> <p>S, M : réel;</p> <p><b>Début</b></p> <p><i>// Entrées</i></p> <p>Ecrire('Donner la taille de la matrice carrée A :');</p> <p>Lire(N);</p> <p>Ecrire('Donner les composantes de la matrice A :');</p> <p><b>Pour</b> i ← 0 à N-1 <b>faire</b></p> <p>    <b>Pour</b> j ← 0 à N-1 <b>faire</b></p> <p>        Lire(A[i,j]);</p> <p>    <b>FinPour</b>;</p> <p><b>FinPour</b>;</p> <p><i>// Traitements</i></p> <p>S ← 0;</p> <p><b>Pour</b> i ← 0 à N-1 <b>faire</b></p> <p>    S ← S+A[i,i] ;</p> <p><b>FinPour</b>;</p> <p>M ← S/N ;</p> <p><i>// Sorties</i></p> <p>Ecrire('S= ', S, ' M=', M);</p> <p><b>Fin.</b></p> | <pre>#include&lt;stdio.h&gt; int main() {     float A[100][100];     int i, j, N;     float S, M;      // Entrées     printf("Donner la taille de la matrice carrée A :");     scanf("%d", &amp;N);     printf("Donner les composantes de la matrice A : \n");     for (i=0;i&lt;N;i++) {         for (j=0;j&lt;N;j++) {             scanf("%f", &amp;A[i][j]);         }     }      // Traitements     S=0;     for (i=0;i&lt;N;i++) {         S=S+A[i][i];     }     M=S/N;      // Sorties     printf(" S = %.2f, M= %.2f", S,M);      return 0; }</pre> |

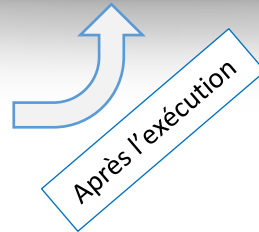
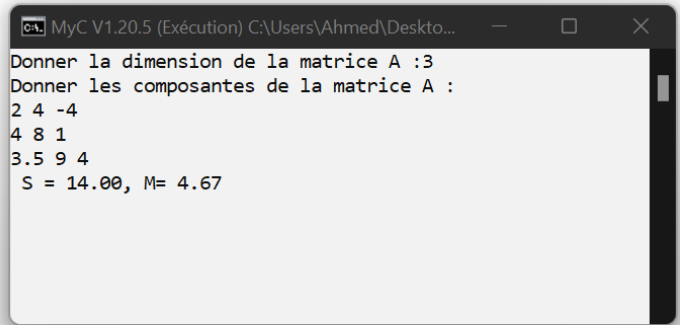
**2 - Compiler et exécuter le programme pour : N = 3 et**

$$A = \begin{bmatrix} 2 & 4 & -4 \\ 4 & 8 & 1 \\ 3.5 & 9 & 4 \end{bmatrix}$$

```

1 #include<stdio.h>
2 int main() {
3     float A[100][100];
4     int i,j,N;
5     float S,M;
6     /* Entrées */
7     printf("Donner la dimension de la matrice A :");
8     scanf("%d", &N);
9     printf("Donner les composantes de la matrice A : \n");
10    for (i=0;i<N;i++) {
11        for (j=0;j<N;j++) {
12            scanf("%f", &A[i][j]);
13        }
14    }
15    /* Traitements */
16    S=0;
17    for (i=0;i<N;i++) {
18        S=S+A[i][i];
19    }
20    M=S/N;
21    /* Sorties */
22    printf(" S = %.2f, M= %.2f", S,M);
23    return 0;
24 }

```



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

| Instructions  | Variables |             |             |  |    |      | Affichage                                |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
|---|-----------|-------------|-------------|--|----|------|--|-----|-----|---|---|----|-----|---|---|---|-----|-----|---|---|---|---|----|
|   | N         | i           | j           | A  | S  | M    |  |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| printf("Donner la dimension de la matrice A :");                        | /         | /           | /           | /  | /  | /    | Donner la dimension de la matrice A :    |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| scanf("%d", &N);  | 3         | /           | /           | /  | /  | /    |  |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| printf("Donner les composantes de la matrice A : \n");                  | 3         | /           | /           | /  | /  | /    | Donner les composantes de la matrice A : |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| for(i=0;i<N;i++){<br>for (j=0;j<N;j++){<br>scanf("%f", &A[i][j]);<br>}} | 3         | 0<br>1<br>2 | 1<br>2<br>3 | <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td></td> <td>j=0</td> <td>j=1</td> <td>j=2</td> </tr> <tr> <td>i=0</td> <td>2</td> <td>4</td> <td>-4</td> </tr> <tr> <td>i=1</td> <td>4</td> <td>8</td> <td>1</td> </tr> <tr> <td>i=2</td> <td>3.5</td> <td>9</td> <td>4</td> </tr> </table> |    | j=0  | j=1                                      | j=2 | i=0 | 2 | 4 | -4 | i=1 | 4 | 8 | 1 | i=2 | 3.5 | 9 | 4 | / | / | // |
|   | j=0       | j=1         | j=2         |  |    |      |  |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| i=0   | 2         | 4           | -4          |  |    |      |  |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| i=1   | 4         | 8           | 1           |  |    |      |  |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| i=2   | 3.5       | 9           | 4           |  |    |      |  |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| S=0;  | 3         | /           | /           | //   | 0  | /    | //                                       |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| <b>For</b> i=0<br>S=S+A[i,i];<br>S=0+A[0,0];<br>S=0+2= <b>2</b> ;       | 3         | 0           |             | //   | 2  | /    | //                                       |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| <b>For</b> i=1<br>S=S+A[i,i];<br>S=2+A[1,1];<br>S=2+8= <b>10</b> ;      | 3         | 1           | /           | //   | 10 | /    | //                                       |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| <b>For</b> i=2<br>S=S+A[i,i];<br>S=10+A[2,2];<br>S=10+4= <b>14</b> ;    | 3         | 2           | /           | //   | 14 | /    | //                                       |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| M=S/N;<br>M=14/3= <b>4.67</b> ;   | 3         |             |             | //   | 14 | 4.67 | //                                       |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |
| printf(" S = %.2f, M= %.2f", S,M);                                      | 3         |             |             | //   | 14 | 4.67 | <b>S=14.00 M=4.67</b>                    |     |     |   |   |    |     |   |   |   |     |     |   |   |   |   |    |

### 4 - Dédire ce que fait le programme ?

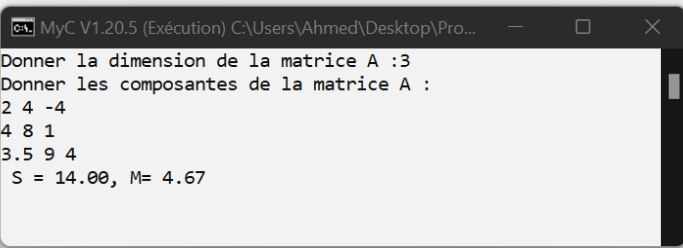
Le programme calcule la somme des éléments de la diagonale de la matrice A et leur moyenne.

## 5 - Ré-écrire le programme en remplaçant la boucle *Pour* par la boucle *Tantque* dans la partie des entrées.

| Programme C (avec la boucle For)  | Programme C (avec la boucle While)  |
|---|---|
| <pre>#include&lt;stdio.h&gt; int main() {     float A[100][100];     int i, j, N;     float S, M;     // Entrées     printf("Donner la dimension de la matrice A :");     scanf("%d", &amp;N);     printf("Donner les composantes de la matrice A : \n");     for (i=0;i&lt;N;i++) {         for (j=0;j&lt;N;j++) {             scanf("%f", &amp;A[i][j]);         }     }     // Traitements     S=0;     for (i=0;i&lt;N;i++)     {         S=S+A[i][i];     }     M=S/N;      // Sorties      printf(" S = %.2f, M= %.2f", S,M); return 0; }</pre> | <pre>#include&lt;stdio.h&gt; int main() {     float A[100][100];     int i, j, N;     float S, M;     // Entrées     printf("Donner la dimension de la matrice A :");     scanf("%d", &amp;N);     printf("Donner les composantes de la matrice A : \n");     i=0;     while(i&lt;N) {         j=0;         while(j&lt;N) {             scanf("%f", &amp;A[i][j]);             j=j+1;         }         i=i+1;     }     // Traitements     S=0;     for (i=0;i&lt;N;i++) {         S=S+A[i][i];     }     M=S/N;     // Sorties     printf(" S = %.2f, M= %.2f", S,M); return 0; }</pre> |

```

1 #include<stdio.h>
2 int main() {
3     float A[100][100];
4     int i,j,N;
5     float S,M;
6     /* Entrées */
7     printf("Donner la dimension de la matrice A :");
8     scanf("%d", &N);
9     printf("Donner les composantes de la matrice A : \n");
10    i=0;
11    while(i<N) {
12        j=0;
13        while(j<N) {
14            scanf("%f", &A[i][j]);
15            j=j+1;
16        }
17        i=i+1;
18    }
19    /* Traitements */
20    S=0;
21    for (i=0;i<N;i++) {
22        S=S+A[i][i];
23    }
24    M=S/N;
25    /* Sorties */
26    printf(" S = %.2f, M= %.2f", S,M);
27    return 0;
28 }
```



Après l'exécution

## 6 - Ré-écrire le programme en remplaçant la boucle *Pour* par la boucle *Répéter* dans la partie des entrées.

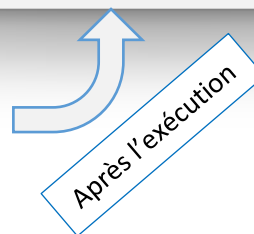
| Programme C (avec la boucle For)  | Programme C (avec la boucle Repeat)   |
|---|---|
| <pre>#include&lt;stdio.h&gt; int main() {     float A[100][100];     int i, j, N;     float S, M;     // Entrées     printf("Donner la dimension de la matrice A :");     scanf("%d", &amp;N);     printf("Donner les composantes de la matrice A : \n");     for (i=0;i&lt;N;i++) {         for (j=0;j&lt;N;j++) {             scanf("%f", &amp;A[i][j]);         }     }     // Traitements     S=0;     for (i=0;i&lt;N;i++)     {         S=S+A[i][i];     }     M=S/N;      // Sorties     printf(" S = %.2f, M= %.2f", S,M);      return 0; }</pre> | <pre>#include&lt;stdio.h&gt; int main() {     float A[100][100];     int i, j, N;     float S, M;     // Entrées     printf("Donner la dimension de la matrice A :");     scanf("%d", &amp;N);     printf("Donner les composantes de la matrice A : \n");     i=0;     do {         j=0;         do {             scanf("%f", &amp;A[i][j]);             j=j+1;         } while(j&lt;N);         i=i+1;     } while(i&lt;N);     // Traitements     S=0;     for (i=0;i&lt;N;i++) {         S=S+A[i][i];     }     M=S/N;      // Sorties     printf(" S = %.2f, M= %.2f", S,M);      return 0; }</pre> |

```

1  #include<stdio.h>
2  int main() {
3      float A[100][100];
4      int i,j,N;
5      float S,M;
6      /* Entrées */
7      printf("Donner la dimension de la matrice A :");
8      scanf("%d", &N);
9      printf("Donner les composantes de la matrice A : \n");
10     i=0;
11     do {
12         j=0;
13         do {
14             scanf("%f", &A[i][j]);
15             j=j+1;
16         } while(j<N);
17         i=i+1;
18     } while(i<N);
19     /* Traitements */
20     S=0;
21     for (i=0;i<N;i++) {
22         S=S+A[i][i];
23     }
24     M=S/N;
25     /* Sorties */
26     printf(" S = %.2f, M= %.2f", S,M);
27     return 0;
28 }
```

```

MyC V1.20.5 (Exécution) C:\Users\Ahmed\Desktop\Program...
Donner la dimension de la matrice A : 3
Donner les composantes de la matrice A :
2 4 -4
2 8 1
3.5 9 4
S = 14.00, M= 4.67
```



## Exercice N°02 : Transposée d'une matrice

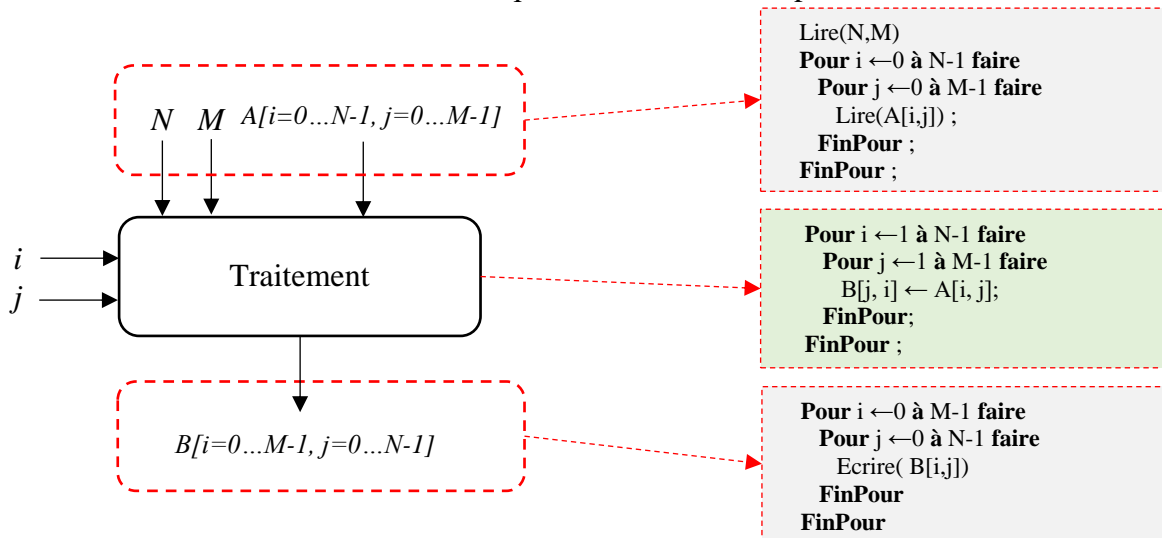
Ecrire un algorithme/programme C qui permet de calculer la matrice B transposée d'une matrice réelle A d'ordre  $N \times M$ .

### Solution :

Le transposé d'une matrice A d'ordre  $N \times M$  est une matrice B d'ordre  $M \times N$ .

Chaque ligne de A devient une colonne de B (ou chaque colonne de A devient une ligne pour B). Chaque case  $B[i, j]$  correspond à la case  $A[j, i]$  tel que :  $i=0, \dots, M-1$  et  $j=0, \dots, N-1$ .

Les variables d'entrée, variable de sortie et la partie traitement sont présentées dans le schéma ci-dessous :



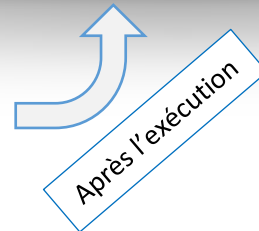
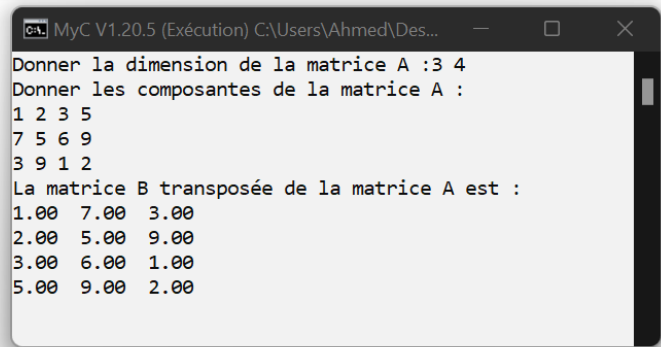
### Algorithme/programme C :

| Algorithme  | Programme C  |
|---|--|
| <p><b>Algorithme</b> Transposee ;</p> <p><b>Variables</b><br/> A, B : Tableau [0..99, 0..99] de réel ;<br/> i, j, N, M : Entier ;</p> <p><b>Début</b><br/> // Entrées<br/> Ecrire('Donner le nombre des lignes et des colonnes de A:') ;<br/> Lire(N,M) ;<br/> Ecrire('Donner les composantes de la matrice A :') ;<br/> <b>Pour</b> i ← 0 à N-1 faire<br/>   <b>Pour</b> j ← 0 à M-1 faire<br/>     Lire(A[i,j]) ;<br/>   <b>FinPour</b> ;<br/> <b>FinPour</b> ;<br/> // Traitements<br/> <b>Pour</b> i ← 0 à N-1 faire<br/>   <b>Pour</b> j ← 0 à M-1 faire<br/>     B[j, i] ← A[i, j] ;<br/>   <b>FinPour</b> ;<br/> <b>FinPour</b> ;<br/> // Sorties<br/> Ecrire('La matrice B Transposée de A est :') ;<br/> <b>Pour</b> i ← 1 à M faire<br/>   <b>Pour</b> j ← 1 à N faire<br/>     Ecrire(B[i,j]) ;<br/>   <b>FinPour</b> ;<br/> <b>FinPour</b> ;<br/> <b>Fin.</b></p> | <pre>#include&lt;stdio.h&gt; int main() {     float A[100][100], B[100][100];     int i, j, N, M;     // Entrées     printf("Donner la dimension de la matrice A :");     scanf("%d %d", &amp;N,&amp;M);     printf("Donner les composantes de la matrice A : \n");     for(i=0;i&lt;N;i++) {         for(j=0;j&lt;M;j++) {             scanf("%f",&amp;A[i][j]);         }     }     // Traitements     for(i=0;i&lt;N;i++) {         for(j=0;j&lt;M;j++) {             B[j][i]=A[i][j];         }     }     // Sorties     printf("La matrice B transposée de la matrice A est : \n");     for(i=0;i&lt;M;i++) {         for(j=0;j&lt;N;j++) {             printf("%.2f",B[i][j]);         }         printf("\n");     }     return 0; }</pre> |

```

1 #include<stdio.h>
2 int main() {
3     float A[100][100], B[100][100];
4     int ij,N,M;
5     /* Entrées */
6     printf("Donner la dimension de la matrice A :");
7     scanf("%d %d", &N,&M);
8     printf("Donner les composantes de la matrice A : \n");
9     for(i=0;i<N;i++) {
10         for(j=0;j<M;j++) {
11             scanf("%f",&A[i][j]);
12         }
13     }
14     /* Traitements */
15     for(i=0;i<N;i++) {
16         for(j=0;j<M;j++) {
17             B[j][i]=A[i][j];
18         }
19     }
20     /* Sorties */
21     printf("La matrice B transposée de la matrice A est : \n");
22     for(i=0;i<M;i++) {
23         for(j=0;j<N;j++) {
24             printf("%.2f ",B[i][j]);
25         }
26         printf("\n");
27     }
28     return 0;
29 }

```



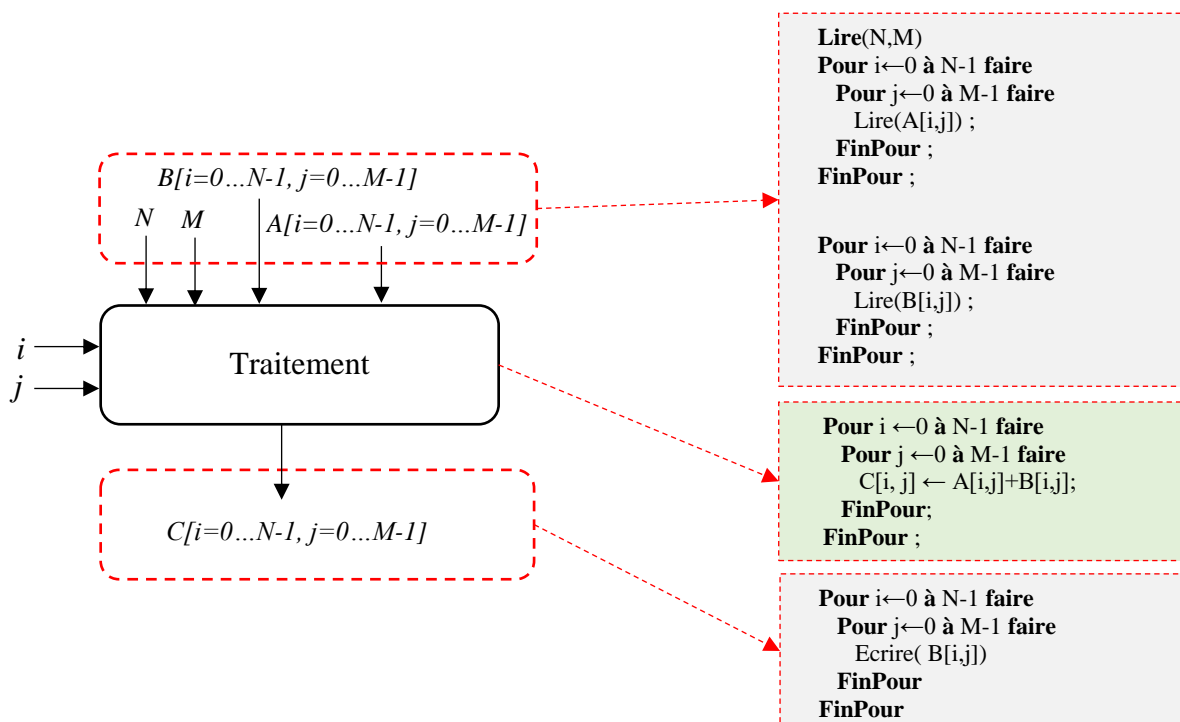
**Exercice N°03 : Somme de deux matrices**

Ecrire un algorithme/programme C qui permet de réaliser la somme de deux matrices réelles A et B d'ordre N x M.

**Solution :**

Pour pouvoir réaliser la somme de deux matrices réelles, A et B, une condition nécessaire doit être vérifiée : A et B doivent être de même taille. Ainsi, si A est d'ordre NxM, alors B est aussi d'ordre NxM. Par conséquent, la matrice C, la somme des deux matrices, est aussi d'ordre NxM.

Les variables d'entrée, variable de sortie et la partie traitement sont présentées dans le schéma ci-dessous :



## Algorithme/programme C :

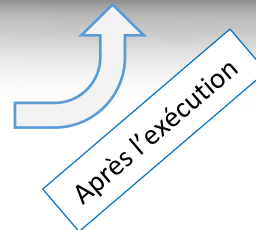
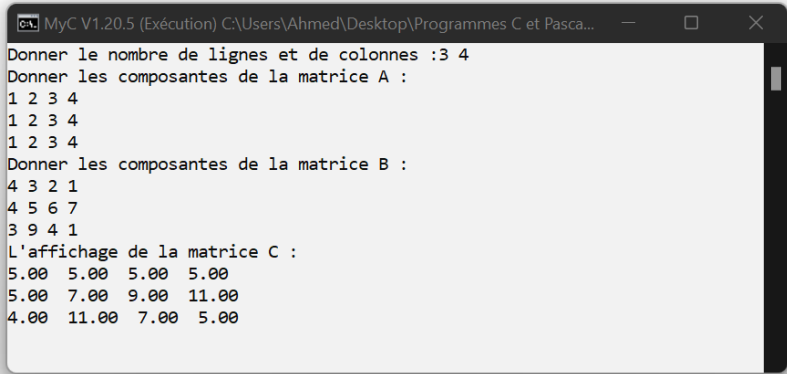
| Algorithme  | Programme C   |
|---|---|
| <p><b>Algorithme</b> Somme_deux_matrices ;</p> <p><b>Variables</b><br/> A, B, C : Tableau [0..99, 0..99] de réel ;<br/> i, j, N, M : Entier ;</p> <p><b>Début</b></p> <p>// Entrées<br/> Ecrire("Donner le nombre des lignes et des colonnes :") ;<br/> Lire(N,M) ;<br/> Ecrire("Donner les composantes de la matrice A :") ;<br/> <b>Pour</b> i←0 à N-1 <b>faire</b><br/>     <b>Pour</b> j←0 à M-1 <b>faire</b><br/>         Lire(A[i,j]) ;<br/>     <b>FinPour</b> ;<br/> <b>FinPour</b> ;</p> <p>Ecrire("Donner les composantes de la matrice B :") ;<br/> <b>Pour</b> i←0 à N-1 <b>faire</b><br/>     <b>Pour</b> j←0 à M-1 <b>faire</b><br/>         Lire(B[i,j]) ;<br/>     <b>FinPour</b> ;<br/> <b>FinPour</b> ;</p> <p>// Traitements<br/> <b>Pour</b> i←0 à N-1 <b>faire</b><br/>     <b>Pour</b> j←0 à M-1 <b>faire</b><br/>         C[i, j]← A[i,j]+B[i,j] ;<br/>     <b>FinPour</b> ;<br/> <b>FinPour</b> ;</p> <p>// Sorties<br/> Ecrire(" L'affichage de la matrice C :") ;<br/> <b>Pour</b> i←0 à N-1 <b>faire</b><br/>     <b>Pour</b> j←0 à M-1 <b>faire</b><br/>         Ecrire( C[i,j]) ;<br/>     <b>FinPour</b> ;<br/> <b>FinPour</b> ;</p> <p><b>Fin.</b></p> | <pre>#include&lt;stdio.h&gt; int main() {     float A[100][100], B[100][100], C[100][100];     int i, j, N, M;     // Entrées     printf("Donner le nombre de lignes et de colonnes :");     scanf("%d %d", &amp;N,&amp;M);     printf("Donner les composantes de la matrice A : \n");     for(i=0;i&lt;N;i++) {         for(j=0;j&lt;M;j++) {             scanf("%f",&amp;A[i][j]);         }     }     printf("Donner les composantes de la matrice B : \n");     for(i=0;i&lt;N;i++) {         for(j=0;j&lt;M;j++) {             scanf("%f",&amp;B[i][j]);         }     }     // Traitements     for(i=0;i&lt;N;i++) {         for(j=0;j&lt;M;j++) {             C[i][j]=A[i][j]+B[i][j];         }     }     // Sorties     printf("L'affichage de la matrice C : \n");     for(i=0;i&lt;N;i++) {         for(j=0;j&lt;M;j++) {             printf("%.2f ",C[i][j]);         }         printf("\n");     }     return 0; }</pre> |



```

1 #include<stdio.h>
2 int main() {
3     float A[100][100], B[100][100], C[100][100];
4     int i,j,N,M;
5     /* Entrées */
6     printf("Donner le nombre de lignes et de colonnes :");
7     scanf("%d %d", &N,&M);
8     printf("Donner les composantes de la matrice A : \n");
9     for(i=0;i<N;i++) {
10        for(j=0;j<M;j++) {
11            scanf("%f",&A[i][j]);
12        }
13    }
14    printf("Donner les composantes de la matrice B : \n");
15    for(i=0;i<N;i++) {
16        for(j=0;j<M;j++) {
17            scanf("%f",&B[i][j]);
18        }
19    }
20    /* Traitements */
21    for(i=0;i<N;i++) {
22        for(j=0;j<M;j++) {
23            C[i][j]=A[i][j]+B[i][j];
24        }
25    }
26    /* Sorties */
27    printf("L'affichage de la matrice C : \n");
28    for(i=0;i<N;i++) {
29        for(j=0;j<M;j++) {
30            printf("%.2f ",C[i][j]);
31        }
32        printf("\n");
33    }
34    return 0;
35 }

```



### Exercice N°04 : Matrice symétrique

Soit A une matrice carrée de taille N x N et de type réel.

Ecrire un programme C qui permet de vérifier si la matrice A est symétrique.

Rappel : Une matrice A est symétrique si  $A[i, j] = A[j, i]$  pour tout  $i$  et  $j$ .

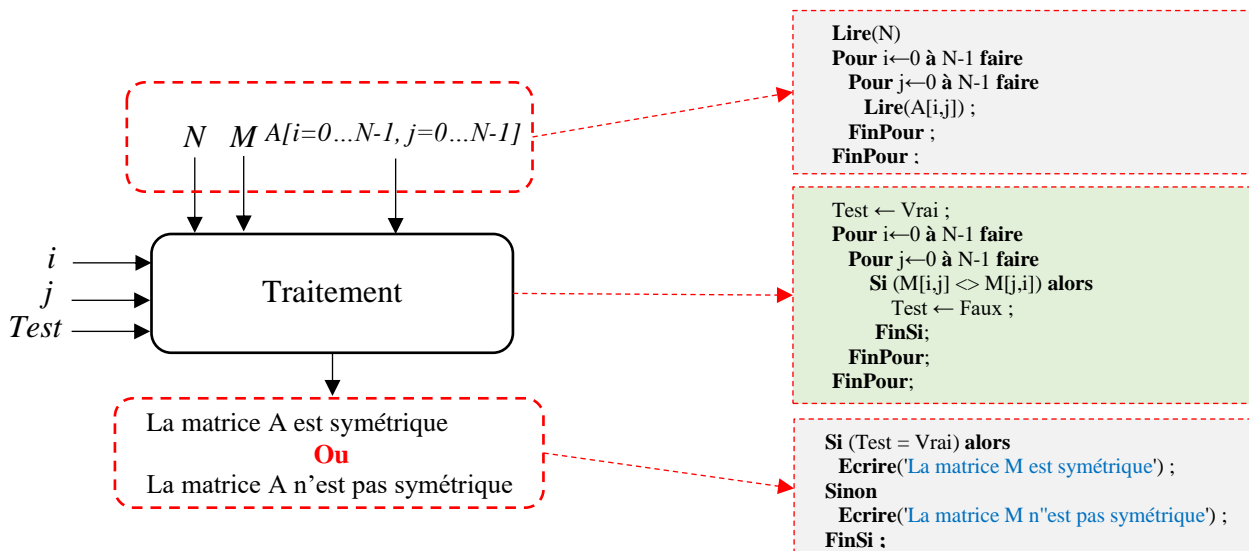
#### Solution :

Rappel : A est symétrique si  $A[i, j] = A[j, i]$  pour tout  $i$  et  $j$ .

Les étapes à suivre :

- D'abord supposer que A est symétrique (Test = True)
- Ensuite, comparer chaque case  $A[j, i]$  avec la case  $A[i, j]$ .
- Si elles sont différentes alors affecter la valeur False à la variable Test.
- A la fin, il suffit de voir la valeur de Test pour savoir si la matrice A est symétrique ou non.

Les variables d'entrée, variable de sortie et la partie traitement sont présentées dans le schéma ci-dessous :



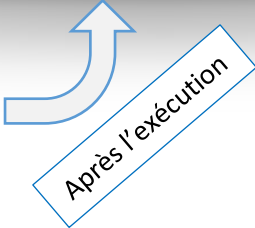
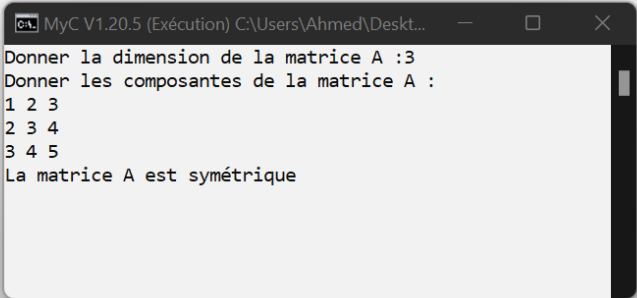
## Algorithme/programme C :

| Algorithme   | Programme C   |
|--|---|
| <p><b>Algorithme</b> Matrice_Symetrique;</p> <p><b>Variables</b><br/> A : Tableau [0..9,0..9] de réel;<br/> N, i, j : entier ;<br/> Test : booléen ;</p> <p><b>Début</b><br/> // Entrées<br/> Ecrire('Donner la dimension de la matrice carrée A :');<br/> Lire(N) ;<br/> Ecrire('Donner les composantes de la matrice A : ');<br/> <b>Pour</b> 0←1 à N-1 <b>faire</b><br/>     <b>Pour</b> j←0 à N-1 <b>faire</b><br/>         Lire(A[i, j]) ;<br/>     <b>FinPour</b>;<br/> <b>FinPour</b>;</p> <p>// Traitements<br/> Test ← Vrai ;</p> <p><b>Pour</b> i←0 à N-1 <b>faire</b><br/>     <b>Pour</b> j←0 à N-1 <b>faire</b><br/>         <b>Si</b> (A[i,j] &lt;&gt; A[j,i]) <b>alors</b><br/>             Test ← Faux ;<br/>         <b>FinSi</b>;<br/>     <b>FinPour</b>;<br/> <b>FinPour</b>;</p> <p>// Sorties<br/> <b>Si</b> (Test = Vrai) <b>alors</b><br/>     Ecrire('La matrice A est symétrique')<br/> <b>Sinon</b><br/>     Ecrire('La matrice A n'est pas symétrique') ;<br/> <b>FinSi</b> ;</p> <p><b>Fin.</b></p> | <pre>#include&lt;stdio.h&gt; int main() {     float A[100][100];     int i, j, N;     int Test;      // Entrées     printf("Donner la dimension de la matrice A :");     scanf("%d", &amp;N);     printf("Donner les composantes de la matrice A : \n");     for(i=0;i&lt;N;i++) {         for(j=0;j&lt;N;j++) {             scanf("%f",&amp;A[i][j]); }         }      // Traitements     Test=1;     for(i=0;i&lt;N;i++) {         for(j=0;j&lt;N;j++) {             if (A[i][j]!=A[j][i]) {                 Test=0;             }         }     }      // Sorties     if(Test==1) {         printf("La matrice A est symétrique "); }     else {         printf("La matrice A n'est pas symétrique "); }     return 0; }</pre> |

```

1 #include<stdio.h>
2 int main() {
3     float A[100][100];
4     int i,j,N;
5     int Test;
6     /* Entrées */
7     printf("Donner la dimension de la matrice A :");
8     scanf("%d", &N);
9     printf("Donner les composantes de la matrice A : \n");
10    for(i=0;i<N;i++) {
11        for(j=0;j<N;j++) { scanf("%f",&A[i][j]); }
12    }
13    /* Traitements */
14    Test=1;
15    for(i=0;i<N;i++) {
16        for(j=0;j<N;j++) {
17            if (A[i][j]!=A[j][i]) {
18                Test=0;
19            } } }
20    /* Sorties */
21    if(Test==1) { printf("La matrice A est symétrique "); }
22    else { printf("La matrice A n'est pas symétrique "); }
23    return 0;
24 }

```



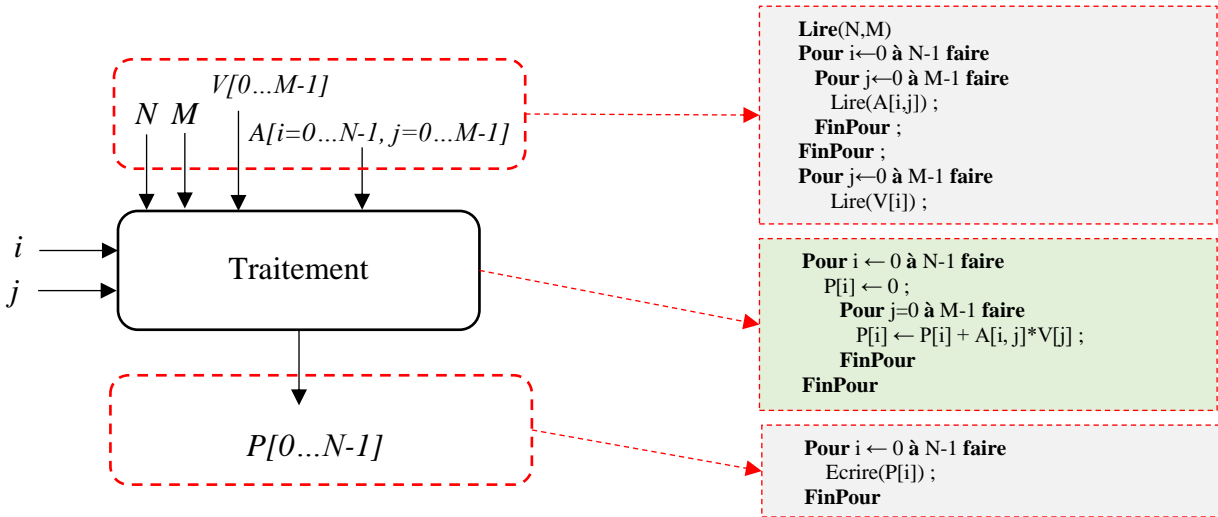
**Exercice N°05 : Produit d'une matrice par un vecteur**

Soit A une matrice de type réel et d'ordre N x M.

Ecrire un algorithme/programme C qui permet de calculer le produit de la matrice A par un vecteur V de type réel et de taille M.

**Solution :**

Les variables d'entrée, variable de sortie et la partie traitement sont présentées dans le schéma ci-dessous :



## Algorithme/programme C :

| Algorithme  | Programme C   |
|---|---|
| <p><b>Algorithme</b> Produit-Matrice_Vecteur ;</p> <p><b>Variables</b><br/> A : Tableau [0..99, 0..99] de réel ;<br/> V, P : Tableau [0..99] de réel ;<br/> N, M, i, j : entier ;</p> <p><b>Début</b><br/> // Entrées<br/> Ecrire("Donner la dimension de la matrice A :");<br/> Lire (N, M);<br/> Ecrire("Donner les composantes de la matrice A : ");<br/> <b>pour</b> i ← 0 à N-1 <b>faire</b><br/>     <b>Pour</b> j ← 0 à M-1 <b>faire</b><br/>         Lire(A[i, j]) ;<br/>     <b>FinPour</b> ;<br/> <b>FinPour</b> ;<br/> Ecrire("Donner les composantes du vecteur V : ");<br/> <b>pour</b> i ← 0 à M-1 <b>faire</b><br/>     Lire(V[i]) ;<br/> <b>FinPour</b> ;<br/> // Traitement<br/> <b>pour</b> i ← 0 à N-1 <b>faire</b><br/>     P[i] ← 0 ;<br/>     <b>Pour</b> j=0 à M-1 <b>faire</b><br/>         P[i] ← P[i] + A[i, j]*V[j] ;<br/>     <b>FinPour</b> ;<br/> <b>FinPour</b> ;<br/> // Sortie<br/> Ecrire("Le résultat de produit :");<br/> <b>pour</b> i ← 0 à N-1 <b>faire</b><br/>     Ecrire(P[i]) ;<br/> <b>FinPour</b><br/> <b>Fin.</b></p> | <pre>#include&lt;stdio.h&gt; int main() {     float A[100][100];     float V[100], P[100];     int N, M, i, j;      // Entrées     printf("Donner la dimension de la matrice A : ");     scanf("%d %d", &amp;N, &amp;M);     printf("Donner les composantes de la matrice A : \n");     for (i=0;i&lt;N;i++)         for (j=0;j&lt;M;j++)         {             scanf("%f", &amp;A[i][j]);         }     printf("Donner les composantes du vecteur V : \n");     for (i=0;i&lt;M;i++) {         scanf("%f", &amp;V[i]);     }     // Traitement     for (i=0;i&lt;N;i++) {         P[i]=0;         for (j=0;j&lt;M;j++) {             P[i]=P[i]+A[i][j]*V[j];         }     }     // Sortie     printf("Le résultat de produit : \n ");     for (i=0;i&lt;N;i++)         printf("%.2f ", P[i]);     return 0; }</pre> |

```

1 #include<stdio.h>
2 int main() {
3     float A[100][100]; float V[100], P[100];
4     int N,M,i,j;
5     /* Entrées */
6     printf("Donner la dimension de la matrice A : ");
7     scanf("%d %d", &N, &M);
8     printf("Donner les composantes de la matrice A : \n");
9     for (i=0;i<N;i++)
10        for (j=0;j<M;j++) {
11            scanf("%f", &A[i][j]);
12        }
13    printf("Donner les composantes du vecteur V : \n");
14    for (i=0;i<M;i++) {
15        scanf("%f", &V[i]);
16    }
17    /* Traitement */
18    for (i=0;i<N;i++) {
19        P[i]=0;
20        for (j=0;j<M;j++) {
21            P[i]=P[i]+A[i][j]*V[j];
22        }
23    }
24    /* Sortie */
25    printf("Le résultat de produit : \n");
26    for (i=0;i<N;i++)
27        printf("%.2f ", P[i]);
28    return 0;
29 }

```

```

MyC V1.20.5 (Exécutio  x  +  v  -  □  ×
1 2 3 4
5 4 3 6
8 7 4 1
Donner les composantes du vecteur V :
3 5 7 1
Le résultat de produit :
38.00 62.00 88.00

```

