

TP - Programmation

Corrigé de la Série de TP N°2 – Tableaux à deux dimensions - Matrices

Exercice N°01 :

Soit l'algorithme suivant :

```

Algorithme Exercice1;
Variables A: tableau[1..10, 1..10] de réel;
            i, j, N : entier;  Z : réel ;

Début
    // Entrées
    Écrire("Donner la taille de la matrice carrée A : ") ;
    Lire (N);
    Écrire("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
    Pour i←0 à N-1 faire
        Z ← A[i, i] ;
        A[i, i] ← A[i, N-i+1];
        A[i, N-i+1] ← Z;
    FinPour
    // Sorties
    Écrire("Affichage de la matrice A : ");
    Pour i←0 à N-1 faire
        Pour j←0 à N-1 faire
            Écrire (A[i, j]) ;
        FinPour
    FinPour
Fin.
    
```

Questions :

- Traduire l'algorithme en Programme C.
- Compiler et exécuter le programme pour les valeurs de N et A suivantes:

$$N = 3 \text{ et } A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$
- Dérouler le programme pour les valeurs de N et A ci-dessus ?
- Déduire ce que fait le programme ?
- Ré-écrire le programme en remplaçant la boucle **Pour** par la boucle **Tantque** dans la partie des entrées.
- Ré-écrire le programme en remplaçant la boucle **Pour** par la boucle **Répéter** dans la partie de traitement.

Corrigé de l'exercice N°01 :

1) Traduire l'algorithme en Programme C.

Algorithme	Programme C
Algorithme Exercice1; Variables A: tableau[1..10, 1..10] de réel; i, j, N : entier; Z : réel ; Début // Entrées Écrire("Donner la taille de la matrice carrée A : ") ; Lire (N); Écrire("Donner les composantes de la matrice A : ");	<pre> #include <stdio.h> int main() { float A[10][10]; int i,j,N; float Z; // Entrées printf("Donner la taille de la matrice carrée A : "); scanf("%d", &N); printf("Donner les composantes de la matrice A : \n"); </pre>

Pour i←0 à N-1 faire Pour j←0 à N-1 faire Lire (A[i, j]) ; FinPour FinPour // Traitement Pour i←0 à N-1 faire Z ← A[i, i] ; A[i, i] ← A[i, N-i-1]; A[i, N-i-1] ← Z; FinPour // Sorties Écrire("Affichage de la matrice A : "); Pour i←0 à N-1 faire Pour j←0 à N-1 faire Écrire (A[i, j]) ; FinPour FinPour Fin.	for (i=0; i<N; i++) { for (j=0; j<N; j++) { scanf("%f", &A[i][j]); } } // Traitements for (i=0; i<N; i++) { Z=A[i][i] ; A[i][i] = A[i][N-i-1]; A[i][N-i-1] = Z; } // Sorties printf("Affichage de la matrice A : \n"); for(i=0; i<N; i++) { for(j=0; j<N; j++) { printf("%.2f ",A[i][j]); } printf("\n"); } return 0; }
--	---

2) Compiler et exécuter le programme pour les valeurs de N et A suivantes:

N = 3 et A =

1

2

3

4

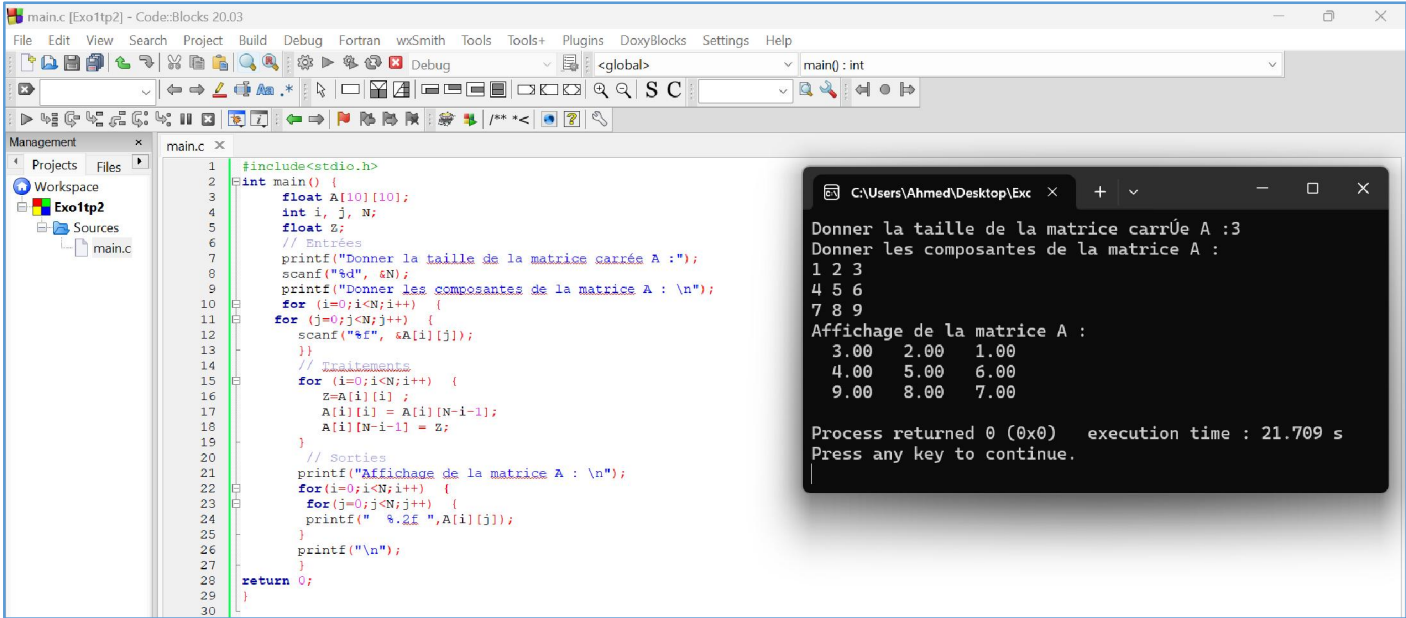
5

6

7

8

9



3) Dérouler le programme pour les valeurs suivantes : N = 3 et A =

1

2

3

4

5

6

7

8

9

Instructions	Variables					Affichage
	N	i	j	A	Z	
printf("Donner la taille de la matrice carrée A :");	/	/	/	/	/	Donner la taille de la matrice carrée A
scanf("%d", &N);	3	/	/	/	/	/

printf("Donner les composantes de la matrice A : \n");	3	/	/	/	/	Donner les composantes de la matrice A :																		
<pre>for (i=0; i<N; i++) { for (j=0; j<N; j++) { scanf("%f", &A[i][j]); } }</pre>	3	0 1 2	0 1 2	<table><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	1	2	3	4	5	6	7	8	9	/	/									
1	2	3																						
4	5	6																						
7	8	9																						
<pre>for i=0 { Z=A[i][i] ; Z=A[0][0] ; Z=3 A[i][i] = A[i][N-i-1]; A[0][0] = A[0][2] A[i][N-i-1] = Z; A[0][2] =3 }</pre>	3	0	/	<table><tr><td>3</td><td>2</td><td>1</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	3	2	1	4	5	6	7	8	9	1	/									
3	2	1																						
4	5	6																						
7	8	9																						
<pre>for i=1 { Z=A[i][i] ; Z=A[1][1] ; Z=5 A[i][i] = A[i][N-i-1]; A[1][1] = A[1][1] A[i][N-i-1] = Z; A[1][1] =5 }</pre>	3	1	/	<table><tr><td>3</td><td>2</td><td>1</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	3	2	1	4	5	6	7	8	9	5	/									
3	2	1																						
4	5	6																						
7	8	9																						
<pre>for i=2 { Z=A[i][i] ; Z=A[2][2] ; Z=9 A[i][i] = A[i][N-i-1]; A[2][2] = A[2][0] A[i][N-i-1] = Z; A[2][0] =9 }</pre>	3	2	/	<table><tr><td>3</td><td>2</td><td>1</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>9</td><td>8</td><td>7</td></tr></table>	3	2	1	4	5	6	9	8	7	9	/									
3	2	1																						
4	5	6																						
9	8	7																						
printf("Affichage de la matrice A : \n");	3	/	/			Affichage de la matrice A :																		
<pre>for(i=0; i<N; i++) { for(j=0; j<N; j++) { printf("%.2f ", A[i][j]); } printf("\n"); }</pre>	3	0 1 2	0 1 2	<table><tr><td>3</td><td>2</td><td>1</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>9</td><td>8</td><td>7</td></tr></table>	3	2	1	4	5	6	9	8	7	9	<table><tr><td>3.00</td><td>2.00</td><td>1.00</td></tr><tr><td>4.00</td><td>5.00</td><td>6.00</td></tr><tr><td>9.00</td><td>8.00</td><td>7.00</td></tr></table>	3.00	2.00	1.00	4.00	5.00	6.00	9.00	8.00	7.00
3	2	1																						
4	5	6																						
9	8	7																						
3.00	2.00	1.00																						
4.00	5.00	6.00																						
9.00	8.00	7.00																						

4) D  duire ce que fait le programme ?

Le programme permet de faire une **permutation** entre la **diagonale principale** et la **diagonale secondaire** (ou anti-diagonale).

5) R   crire le programme en rempla  ant la boucle *Pour* par la boucle *Tantque* dans la partie des entr  es.

Programme C (avec la boucle For)	Programme C (avec la boucle While)
<pre>#include <stdio.h> int main() { float A[10][10]; int i,j,N; float Z; // Entr��es printf("Donner la taille de la matrice carr��e A : "); scanf("%d", &N); printf("Donner les composantes de la matrice A : \n"); for (i=0; i<N; i++) { for (j=0; j<N; j++) { scanf("%f ", &A[i][j]); } } // Traitements</pre>	<pre>#include <stdio.h> int main() { float A[10][10]; int i,j,N; float Z; // Entr��es printf("Donner la taille de la matrice carr��e A : "); scanf("%d", &N); printf("Donner les composantes de la matrice A : \n"); i=0; while(i<N) { j=0; while(j<N) { scanf("%f ", &A[i][j]); j=j+1; } i=i+1; }</pre>

<pre> for (i=0; i<N; i++) { Z=A[i][i] ; A[i][i] = A[i][N-i-1]; A[i][N-i-1] = Z; } // Sorties printf("Affichage de la matrice A : \n"); for(i=0; i<N; i++) { for(j=0; j<N; j++) { printf("%.2f ",A[i][j]); } printf("\n"); } return 0; } </pre>	<pre> i=i+1; } // Traitements for (i=0; i<N; i++) { Z=A[i][i] ; A[i][i] = A[i][N-i-1]; A[i][N-i-1] = Z; } // Sorties printf("Affichage de la matrice A : \n"); for(i=0; i<N; i++) { for(j=0; j<N; j++) { printf("%.2f ",A[i][j]); } printf("\n"); } return 0; } </pre>
---	---

6) Ré-écrire le programme en remplaçant la boucle Pour par la boucle Répéter dans la partie de traitement.

Programme C (avec la boucle For)	Programme C (avec la boucle Do-While)
<pre> #include <stdio.h> int main() { float A[10][10]; int i,j,N; float Z; // Entrées printf("Donner la taille de la matrice carrée A : "); scanf("%d", &N); printf("Donner les composantes de la matrice A : \n"); for (i=0; i<N; i++) { for (j=0; j<N; j++) { scanf("%f", &A[i][j]); } } // Traitements for (i=0; i<N; i++) { Z=A[i][i] ; A[i][i] = A[i][N-i-1]; A[i][N-i-1] = Z; } // Sorties printf("Affichage de la matrice A : \n"); for(i=0; i<N; i++) { for(j=0; j<N; j++) { printf("%.2f ",A[i][j]); } printf("\n"); } return 0; } </pre>	<pre> #include <stdio.h> int main() { float A[10][10]; int i,j,N; float Z; // Entrées printf("Donner la taille de la matrice carrée A : "); scanf("%d", &N); printf("Donner les composantes de la matrice A : \n"); for (i=0; i<N; i++) { for (j=0; j<N; j++) { scanf("%f", &A[i][j]); } } // Traitements i=0; do { Z=A[i][i] ; A[i][i] = A[i][N-i-1]; A[i][N-i-1] = Z; i=i+1; } while(i<N); // Sorties printf("Affichage de la matrice A : \n"); for(i=0;i<N;i++) { for(j=0;j<N;j++) { printf("%.2f ",A[i][j]); } printf("\n"); } return 0; } </pre>

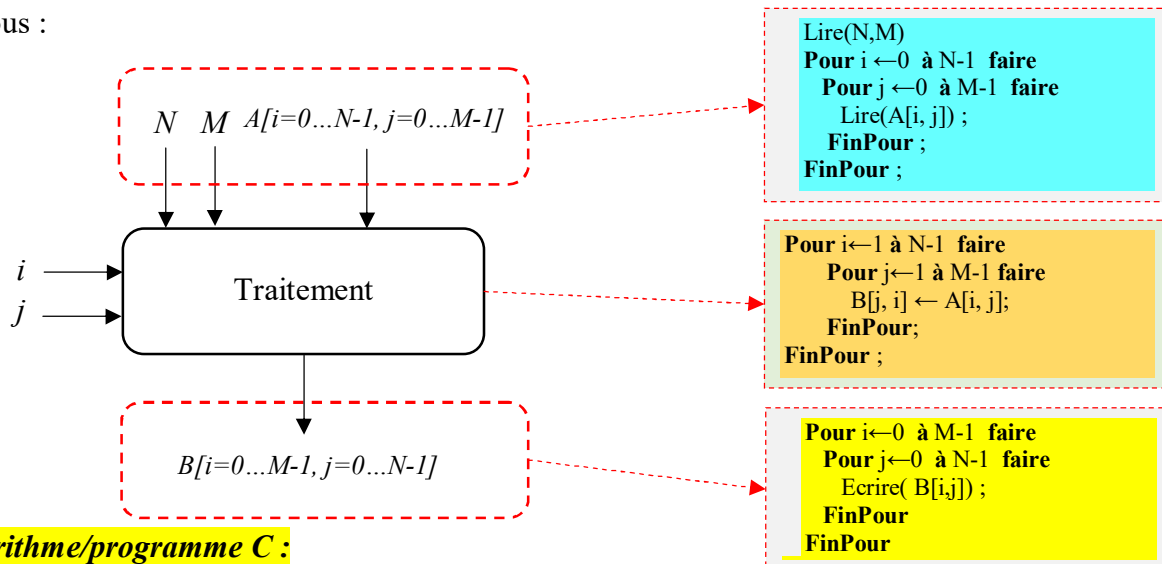
Exercice N°02 :

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

Corrigé de l'exercice N°02 :

Le transposé d'une matrice A d'ordre N×M est une matrice B d'ordre M×N. Chaque ligne de A devient une colonne de B (ou chaque colonne de A devient une ligne de B). Ainsi, chaque case B[i, j] correspond à la case A[j,i], où i=0,...,N-1 et j=0,...,M-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
Algorithme Transposee ; Variables A, B : Tableau [0..99, 0..99] de réel ; i,j,N,M : Entier ; Début // Entrées Ecrire("Donner le nombre des lignes et des colonnes de A:") ; Lire(N,M) ; Ecrire("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 ; // Traitements Pour i←0 à N-1 faire Pour j←0 à M-1 faire B[j, i]← A[i, j] ; FinPour ; FinPour ; // Sorties Ecrire("La matrice B Transposée de A est:") ; Pour i←0 à M-1 faire Pour j←0 à N-1 faire Ecrire(B[i, j]) ; FinPour ; FinPour ; Fin.	<pre> #include<stdio.h> 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", &N, &M); printf("Donner les composantes de la matrice A : \n"); for(i=0; i<N; i++) { for(j=0; j<M; j++) { //printf("A[%d, %d]= ", i, j); scanf("%f", &A[i][j]); } } // Traitements for(i=0; i<N; i++) { for(j=0; j<M; j++) { B[j][i]=A[i][j]; } } // Sorties printf("La matrice B transposée de la matrice A est : \n"); for(i=0; i<M; i++) { for(j=0; j<N; j++) { printf("%.2f ", B[i][j]); } printf("\n"); } return 0; } </pre> Une autre méthode : for(i=0; i<M; i++) for(j=0; j<N; j++) B[i][j]=A[j][i];

```

1  #include<stdio.h>
2  int main() {
3      float A[100][100], B[100][100];
4      int i,j,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 }

```

MyC V1.20.5 (Exécution) C:\Users\Ahmed\Des...

Donner la dimension de la matrice A : 3 4
 Donner les composantes de la matrice A :
 1 2 3 5
 7 5 6 9
 3 9 1 2

La matrice B transposée de la matrice A est :
 1.00 7.00 3.00
 2.00 5.00 9.00
 3.00 6.00 1.00
 5.00 9.00 2.00

Exercice N°03 :

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.

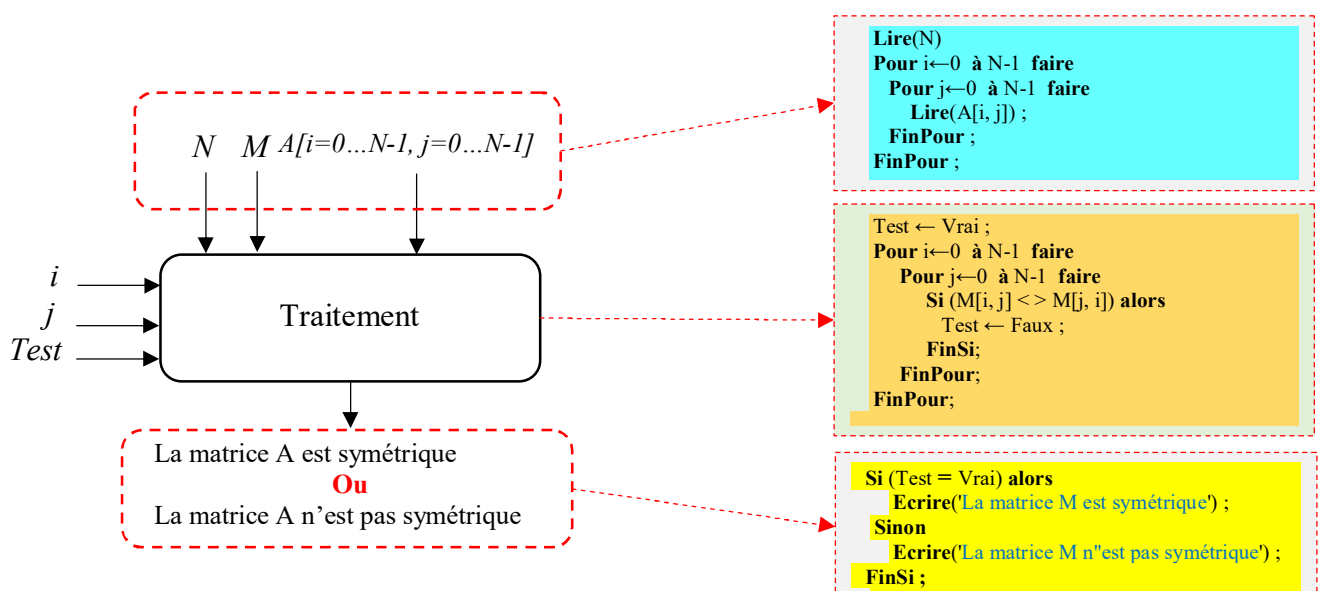
Corrigé de l'exercice N°03 :

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

Les étapes à suivre :

- D'abord supposé **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 :



Algorithmme/programme C :

Algorithmme	Programme C
Algorithmme Matrice_Symetrique; Variables A : Tableau [0..99,0..99] de réel; N,i,j : entier ; Test : booléen ; Début // Entrées Ecrire('Donner la dimension de la matrice carrée A :'); Lire(N) ; Ecrire('Donner les composantes de la matrice A : '); Pour 0←1 à N-1 faire Pour j←0 à N-1 faire Lire(A[i, j]) ; FinPour ; FinPour ; // Traitements Test← Vrai ; Pour i←0à N-1 faire Pour j←0à N-1 faire Si (A[i,j] <> A[j,i]) alors Test ← Faux ; FinSi ; FinPour ; FinPour ; // Sorties Si (Test = Vrai) alors Ecrire('La matrice A est Symétrique') Sinon Ecrire('La matrice A est Asymétrique') ; FinSi ; Fin.	#include <stdio.h> #include <stdbool.h> int main() { float A[100][100]; int i,j,N; bool test; // Entrées printf("Donner la dimension de la matrice A :"); scanf("%d", &N); printf("Donner les composantes de la matrice A : \n"); for(i=0; i<N; i++) { for(j=0; j<N; j++) { scanf("%f", &A[i][j]); } } // Traitements Test=true; for(i=0; i<N; i++) { for(j=0; j<N; j++) { if (A[i][j] != A[j][i]) { Test=false; } } } // Sorties if (Test == true) { printf("La matrice A est Symétrique "); } else { printf("La matrice A est Asymétrique "); } return 0; }

Compilation et exécution du code C

```

#include <stdio.h>
#include <stdbool.h>
int main()
{
    float A[100][100];
    int i,j,N;
    bool Test;

    // Entrées
    printf("Donner la dimension de la matrice A : ");
    scanf("%d", &N);
    printf("Donner les composantes de la matrice A : \n");
    for(i=0; i<N; i++) {
        for(j=0; j<N; j++) {
            scanf("%f", &A[i][j]); }
    }

    // Traitements
    Test=true;
    for(i=0; i<N; i++) {
        for(j=0; j<N; j++) {
            if (A[i][j] != A[j][i]) {
                Test=false;
            }
        }
    }

    // Sorties
    if (Test == true) {
        printf("La matrice A est Symétrique "); }
    else {
        printf("La matrice A est Asymétrique "); }
    return 0;
}

```

```

C:\Users\Maison\Desktop\lhkldgljgljgl\matrice_
Donner la dimension de la matrice A : 3
Donner les composantes de la matrice A :
1 2 3
2 5 6
3 6 8
La matrice A est Symétrique
Process returned 0 (0x0) execution time : 30.009 s
Press any key to continue.

```

Exercice N°04 :

Ecrire un programme en C qui demande à l'utilisateur de remplir un tableau bidirectionnel ($N \times M$) de type *caractères* puis recherche si un caractère spécifique présent dans le tableau. Si le caractère est trouvé le programme affiche sa position (*ligne et colonne*), sinon il affiche un message "*le caractère n'existe pas dans le tableau*".

Corrigé de l'exercice 04:

Programme en C

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    int N, M, i, j;
    char T[100][100];
    char caractere;
    bool tr;

    // Demander à l'utilisateur les dimensions du tableau
    printf("Entrez le nombre de lignes (N) et le nombre de colonne (M) : ");
    scanf("%d%d", &N, &M);

    // Demander à l'utilisateur de remplir le tableau
    printf("Entrez les elements du tableau (un caractère par case) : \n");
    for( i = 0; i < N; i++) {
        for( j = 0; j < M; j++) {
            scanf(" %c", &T[i][j]);
        }
    }

    // Demander le caractère à rechercher
    printf("Entrez le caractere a rechercher : ");
    scanf(" %c", &caractere);

    // Rechercher le caractère dans le tableau
    tr = false; // initialisation de la variable trouvé tr =false
    for( i = 0; i < N; i++) {
        for( j = 0; j < M; j++) {
            if(T[i][j] == caractere) {
                printf("Le caractere ' %c ' a ete trouve a la position : ligne %d, colonne %d \n",
                    caractere, i, j);
                tr = true; // Marquer que le caractère a été trouvé
            }
        }
    }

    // Si le caractère n'a pas été trouvé, afficher un message
    if(!tr) {
        printf("le caractere '%c' n'existe pas dans le tableau.\n", caractere);
    }
    return 0;
}
```

Compilation et exécution du programme C

main.c X

```
1 #include <stdio.h>
2 #include <stdbool.h>
3
4 int main() {
5     int N, M;
6     char T[100][100];
7     char caractere;
8     int i, j;
9     bool tr;
10
11     // Demander à l'utilisateur les dimensions du tableau
12     printf("Entrez le nombre de lignes (N) : ");
13     scanf("%d", &N);
14     printf("Entrez le nombre de colonnes (M) : ");
15     scanf("%d", &M);
16
17     // Demander à l'utilisateur de remplir le tableau
18     printf("Entrez les elements du tableau (un caractère par case) :\n");
19     for(i = 0; i < N; i++) {
20         for(j = 0; j < M; j++) {
21             scanf(" %c", &T[i][j]);
22         }
23     }
24
25     // Demander le caractère à rechercher
26     printf("Entrez le caractere a rechercher : ");
27     scanf(" %c", &caractere);
28
29     // Rechercher le caractère dans le tableau
30     tr = false;
31     for(i = 0; i < N; i++) {
32         for(j = 0; j < M; j++) {
33             if(T[i][j] == caractere) {
34                 printf("Le caractere '%c' a ete trouve a la position : ligne %d, colonne %d\n", caractere, i, j);
35                 tr = true; // Marquer que le caractère a été trouvé
36             }
37         }
38     }
39
40     // Si le caractère n'a pas été trouvé, afficher un message
41     if(!tr) {
42         printf("le caractere '%c' n'existe pas dans le tableau.\n", caractere);
43     }
44
45     return 0;
46 }
47
48 }
```

C:\Users\Maison\Desktop\lhlklgljgljgl\caracters\bin\Debug\ca... — □ X

Entrez le nombre de lignes (N) : 5
Entrez le nombre de colonnes (M) : 5
Entrez les elements du tableau (un caractPre par case) :
A R r t i
: l ; m ù
S 1 & ; :
! ' () =
+ " z W b
Entrez le caractere a rechercher : &
Le caractere '&' a ete trouve a la position : ligne 2, colonne 2