

TP – Initiation à la Programmation

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

Corrigé de l'exercice N°01 :

1) Traduire l'algorithme en Programme C.

Algorithme	Programme C
Algorithme Matrice; Variables A : tableau [1..10, 1..10] de réel ; i, j, N : entier ; Début <i>// Entrées</i> 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 ; <i>// Traitement</i> Pour i ← 0 à N-1 faire A[i, i] ← 2 * A[i, i] ; FinPour ; <i>// Sorties</i> Écrire("La matrice A est : ") ; Pour i ← 0 à N-1 faire Pour j ← 0 à N-1 faire Écrire (A[i, j]) ; FinPour ; FinPour ; Fin.	#include <stdio.h> int main() { int N, i, j; float A[10][10]; <i>// Entrée de la taille de A</i> printf("Donner la taille de la matrice carrée A : "); scanf("%d", &N); <i>// Saisie de la matrice A</i> printf("Donner les composantes de la matrice A :\n"); for(i = 0; i < N; i++) { for(j = 0; j < N; j++) { printf("A[%d][%d] = ", i, j); scanf("%f", &A[i][j]); } } <i>// Traitement : doubler la diagonale principale</i> for(i = 0; i < N; i++) { A[i][i] = 2 * A[i][i]; } <i>// Affichage de la matrice</i> printf("\nLa matrice A est : \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:

$$N = 3 \text{ et } A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

```

main.c
1 #include <stdio.h>
2 int main() {
3     int N, i, j;
4     float A[10][10];
5     // Entrée de la taille de A
6     printf("Donner la taille de la matrice carree A : ");
7     scanf("%d", &N);
8     // Saisie de la matrice A
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        // Traitement : doubler la diagonale principale
15        for(i = 0; i < N; i++) {
16            A[i][i] = 2 * A[i][i];
17        }
18        // Affichage de la matrice
19        printf("\nLa matrice A est : \n");
20        for(i = 0; i < N; i++) {
21            for(j = 0; j < N; j++) {
22                printf("%.2f ", A[i][j]);
23            }
24            printf("\n");
25        }
26    }
27    return 0;
28 }

```

```

C:\Users\Maison\Desktop\lhklgljlgj\exxx\bin\D...
Donner la taille de la matrice carree A : 3
Donner les composantes de la matrice A :
1 2 3
4 5 6
7 8 9

La matrice A est :
2.00 2.00 3.00
4.00 10.00 6.00
7.00 8.00 18.00

Process returned 0 (0x0)   execution time : 12.849 s
Press any key to continue.

```

3) Dérouler le programme pour les valeurs suivantes : $N = 3$ et $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

Instructions	Variables				Affichage											
	N	i	j	A												
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 :											
for (i=0;i<=N;i++) { for (j=0;j<=N;j++) { scanf("%f", &A[i][j]); }}	3	1 2 3	1 2 3	<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														
for i=0 { A[i][i]=2*A[i][i] ; A[0][0]=2* A[0][0] A[0][0] = 2*1=2; }	3	0	/	<table><tr><td>2</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>	2	2	3	4	5	6	7	8	9	/		
2	2	3														
4	5	6														
7	8	9														
for i=1 { A[i][i]=2* A[i][i] ; A[1][1]=2* A[1][1]=2*5=10; }	3	1	/	<table><tr><td>2</td><td>2</td><td>3</td></tr><tr><td>4</td><td>10</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	2	2	3	4	10	6	7	8	9	/		
2	2	3														
4	10	6														
7	8	9														
for i=2 { A[i][i]=2* A[i][i] ; A[2][2]=2* A[2][2] A[2][2]=2*9=18; }	3	2	/	<table><tr><td>2</td><td>2</td><td>3</td></tr><tr><td>4</td><td>10</td><td>6</td></tr><tr><td>7</td><td>8</td><td>18</td></tr></table>	2	2	3	4	10	6	7	8	18	/		
2	2	3														
4	10	6														
7	8	18														
printf(" La matrice A est : \n");	3	/	/	//	La matrice A est :											
for(i=1;i<=N;i++) { for(j=1;j<=N;j++) { printf("%.2f",A[i][j]); } printf("\n"); }	3	0 1 2	0 1 2	<table><tr><td>2</td><td>2</td><td>3</td></tr><tr><td>4</td><td>10</td><td>6</td></tr><tr><td>7</td><td>8</td><td>18</td></tr></table>	2	2	3	4	10	6	7	8	18	2.00 4.00 7.00	2.00 10.00 8.00	3.00 6.00 18.00
2	2	3														
4	10	6														
7	8	18														

4) Déduire ce que fait le programme : Le programme permet de multiplier par 2 les éléments de la diagonale principale puis affiche la matrice A.

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 <stdio.h> int main() { int N, i, j; float A[10][10]; // Entrée de la taille de A printf("Donner la taille de la matrice carree A : "); scanf("%d", &N); // Saisie de la matrice A 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]); } // Traitement : doubler la diagonale principale } for(i = 0; i < N; i++) { A[i][i] = 2 * A[i][i]; } // Affichage de la matrice printf("\nLa matrice A est : \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() { int N, i, j; float A[10][10]; // Entrée de la taille de A printf("Donner la taille de la matrice carree A : "); scanf("%d", &N); // Saisie de la matrice A 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; } // Traitement : doubler la diagonale principale for(i = 0; i < N; i++) { A[i][i] = 2 * A[i][i]; } // Affichage de la matrice printf("\nLa matrice A est : \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() { int N, i, j; float A[10][10]; // Entrée de la taille de A printf("Donner la taille de la matrice carree A : "); scanf("%d", &N); // Saisie de la matrice A printf("Donner les composantes de la matrice A :\n"); for(i = 0; i < N; i++) {</pre>	<pre>#include <stdio.h> int main() { int N, i, j; float A[10][10]; // Entrée de la taille de A printf("Donner la taille de la matrice carree A : "); scanf("%d", &N); // Saisie de la matrice A printf("Donner les composantes de la matrice A :\n"); for(i = 0; i < N; i++) {</pre>

<pre> for(j =0; j <N; j++) { printf("A[%d][%d] = ", i, j); scanf("%f", &A[i][j]); } // Traitement : doubler la diagonale principale for(i = 0; i <N; i++) { A[i][i] = 2 * A[i][i]; } // Affichage de la matrice printf("\nLa matrice A est : \n"); for(i = 0; i <N; i++) { for(j = 0; j <N; j++) { printf("%.2f", A[i][j]); } printf("\n"); } return 0; } </pre>	<pre> for(j =0; j <N; j++) { printf("A[%d][%d] = ", i, j); scanf("%f", &A[i][j]); } // Traitement : doubler la diagonale principale i=0; do { j=0; do { A[i][i] = 2 * A[i][i]; j=j+1;} While(j<N); i=i+1 ;} While (j<N) ; // Affichage de la matrice printf("\nLa matrice A est : \n"); for(i =0; i <N; i++) { for(j =0; j <N; j++) { printf("%.2f", A[i][j]); } printf("\n"); } return 0; } </pre>
---	---

Corrigé de l'exercice N°02 : Algorithme/programme C

Algorithme	Programme C
Algorithme Somme_Lignes ; Variables A : tableau [1..10, 1..10] de réel ; S : tableau [1..10] de réel ; i, j, N, M : entier ; Début // Lecture des dimensions Écrire("Donner le nombre de lignes N et colonne M : ") ; Lire(N,M) ; // Saisie de la matrice 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 ; // Calcul des sommes par ligne Pour i ← 0 à N-1 faire S[i] ← 0 ; Pour j ← 0 à M-1 faire S[i] ← S[i] + A[i, j]; FinPour; FinPour ;	<pre> #include <stdio.h> int main() { int N, M, i, j; float A[10][10], S[10]; // Lecture des dimensions printf("Donner le nombre de lignes N et colonnes M : "); scanf("%d %d", &N, &M); // Saisie de la matrice 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]); } } // Calcul des sommes par ligne for(i = 0; i < N; i++) { S[i] = 0; for(j = 0; j < M; j++) { S[i] = S[i] + A[i][j]; } } } </pre>

// Affichage

```
Écrire("Somme de chaque ligne :")
Pour i ← 0 à N-1 faire
    Écrire("Ligne ", i, " : ", S[i]);
FinPour ;
Fin.
```

// Affichage

```
printf("\nSomme de chaque ligne :\n");
for(i = 0; i < N; i++) {
    printf("Ligne %d : %.2f\n", i + 1, S[i]);
}
return 0; }
```

Compilation et execution du programme

The screenshot shows a C program in a text editor and its execution in a command prompt. The C program calculates the sum of each row of a matrix A and prints the results. The command prompt shows the user inputting the dimensions N=2 and M=3, followed by the matrix elements. The output shows the sum for each line: Ligne 1 : 6.00 and Ligne 2 : 15.00.

```
main.c x
1 #include <stdio.h>
2 int main() {
3     int N, M, i, j;
4     float A[10][10], S[10];
5
6     // Lecture des dimensions
7     printf("Donner le nombre de lignes N et colonnes M : ");
8     scanf("%d %d", &N, &M);
9     // Saisie de la matrice
10    printf("Donner les composantes de la matrice A :\n");
11    for(i = 0; i < N; i++) {
12        for(j = 0; j < M; j++) {
13            scanf("%f", &A[i][j]);
14        }
15    }
16    // Calcul des sommes par ligne
17    for(i = 0; i < N; i++) {
18        S[i] = 0;
19        for(j = 0; j < M; j++) {
20            S[i] = S[i] + A[i][j];
21        }
22    }
23    // Affichage
24    printf("\nSomme de chaque ligne :\n");
25    for(i = 0; i < N; i++) {
26        printf("Ligne %d : %.2f\n", i + 1, S[i]);
27    }
28    return 0; }
```

```
C:\Users\Maison\Desktop\lhklkgjlgj\exa\bin\...
Donner le nombre de lignes N et colonnes M : 2 3
Donner les composantes de la matrice A :
1 2 3
4 5 6

Somme de chaque ligne :
Ligne 1 : 6.00
Ligne 2 : 15.00

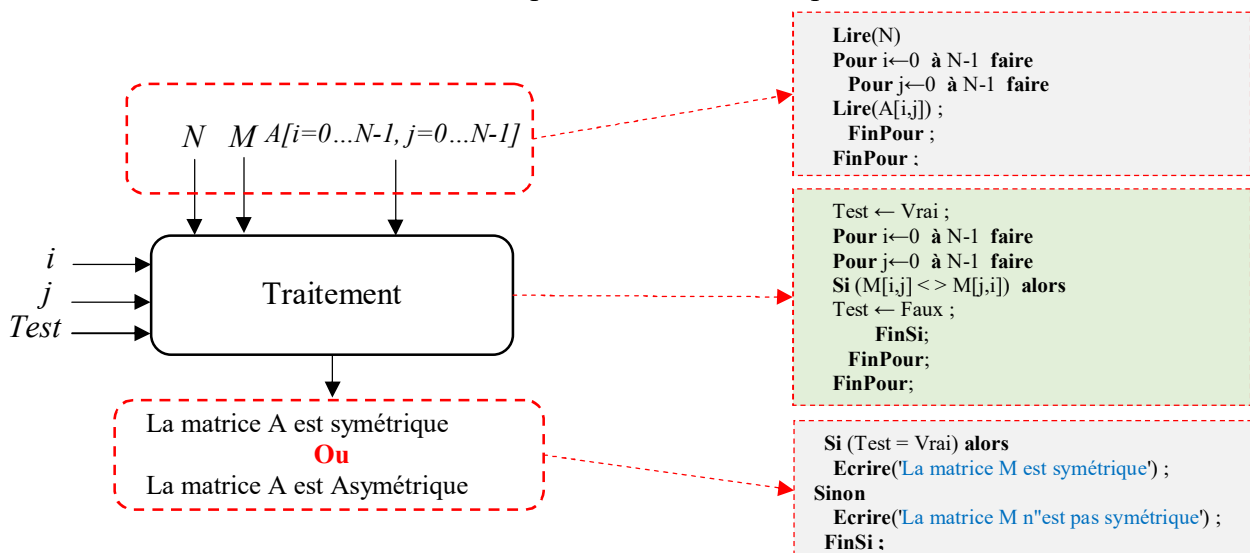
Process returned 0 (0x0)   execution time : 12.213 s
Press any key to continue.
```

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 sont :

- 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 on affecte 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
Algorithme 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←0 à N-1 faire Pour j←0 à 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.	<pre>#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; }</pre>

Compilation et exécution du code C

The image shows a C program being compiled and executed. The source code is in a file named 'main.c'. The program prompts the user to enter the dimension of a square matrix 'A' and then its components. It then checks if the matrix is symmetric by comparing elements A[i][j] and A[j][i]. If they are not equal, the matrix is asymmetric. The execution output shows that the matrix is asymmetric.

main.c

```
1 #include <stdio.h>
2 #include <stdbool.h>
3 int main()
4 {
5     float A[100][100];
6     int i, j, N;
7     bool Test;
8
9     // Entrées
10    printf("Donner la dimension de la matrice A :");
11    scanf("%d", &N);
12    printf("Donner les composantes de la matrice A : \n");
13    for(i=0; i<N; i++) {
14        for(j=0; j<N; j++) {
15            scanf("%f", &A[i][j]);
16        }
17    }
18
19    // Traitements
20    Test=true;
21    for(i=0; i<N; i++) {
22        for(j=0; j<N; j++) {
23            if (A[i][j] != A[j][i]) {
24                Test=false;
25            }
26        }
27    }
28
29    // Sorties
30    if (Test == true) {
31        printf("La matrice A est Symetrique ");
32    }
33    else {
34        printf("La matrice A est Asymetrique ");
35    }
36    return 0;
37 }
```

Asymétrique

```
Donner la dimension de la matrice A :3
Donner les composantes de la matrice A :
1 2 3
4 5 6
7 8 9
La matrice A est Asymetrique
Process returned 0 (0x0)   execution time : 9.776 s
Press any key to continue.
```

Symétrique

```
Donner la dimension de la matrice A :3
Donner les composantes de la matrice A :
1 2 3
2 4 5
3 5 6
La matrice A est Symetrique
Process returned 0 (0x0)   execution time : 21.620 s
Press any key to continue.
```