

## Exercices supplémentaires sur les tableaux

---

### Exercice 82 :

Écrire un programme permettant de remplir un tableau de 100 cases de façon à ce que la valeur contenue dans chaque case soit le numéro de cette case, et de l'afficher.

#### Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 100
    int t[n],i;
    for(i=0;i<n;i++){
        t[i]=i;
    }
    printf("Le tableau est:\n");
    for(i=0;i<n;i++){
        printf("%d | ",t[i]);
    }
    return 0;
}
```

### Exercice 83 :

Écrire un programme qui saisit un tableau de 10 entiers et qui détermine si tous les éléments sont positifs ou nuls.

#### Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 10
    int t[n],i;
    printf("Veuillez saisir le tableau:\n");
    for(i=0;i<n;i++){
        printf("Case %d: ",i);
        scanf("%d",&t[i]);
    }
    i=0;
    while(i<n && t[i]>=0){
        i=i+1;
    }
    if(i==n)printf("Tous les elements sont positifs ou nulls");
    else printf("Il y a au moins un element negatif");
    return 0;
}
```

### Exercice 84 :

Écrire un programme permettant de remplir un tableau de 20 cases de façon à ce que la case  $i$  contiendra la somme des valeurs de 0 à  $i$ , et de l'afficher.

**Exemple :** le tableau à obtenir est :

0	1	2	3	4	5	...	$\sum_{j=0}^i j$
0	1	3	6	10	15	...	

**Réponse :**

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 20
    int t[n],i,j;
    for(i=0;i<n;i++){
        t[i]=0;
        for(j=0;j<=i;j++){
            t[i]=t[i]+j;
        }
    }
    printf("Le tableau est:\n");
    for(i=0;i<n;i++){
        printf("%d | ",t[i]);
    }
    return 0;
}
```

**Exercice 85 :**

Ecrire un programme permettant de saisir un tableau de 10 éléments réels non nuls, de le normaliser, et d'afficher le tableau normalisé. La normalisation d'un tableau consiste à diviser tous les éléments du tableau par le plus grand élément de façon à n'avoir que des nombres compris entre 0 et 1.

**Exemple :** dans le tableau ci-dessous, la valeur maximale est 8.

Tableau initial										Tableau normalisé (les éléments sont divisés par 8)										
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	
3	1	8	1	2	8	0	5	4	4	→	0.375	0.125	1	0.125	0.25	1	0	0.625	0.5	0.5

**Réponse :**

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 10
    float t[n],max;int i;
    printf("Veuillez saisir le tableau:\n");
    for(i=0;i<n;i++){
        printf("Case %d: ",i);
        scanf("%f",&t[i]);
    }
    max=t[0];
    for(i=1;i<n;i++){
        if(t[i]>max)max=t[i];
    }
    for(i=0;i<n;i++){
        t[i]=t[i]/max;
    }
}
```

```

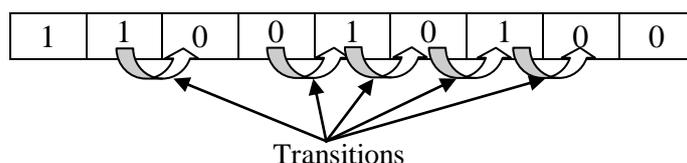
printf("Le tableau normalise est:\n");
for(i=0;i<n;i++){
    printf("%.3f | ",t[i]);
}
return 0;
}

```

### Exercice 86 :

Considérons un tableau de 20 valeurs binaires (0 ou 1) représentant les états d'un objet physique dans 100 instants différents. On appelle transition, le changement d'état entre deux instants successifs ; en d'autres termes le passage de 0 à 1 ou bien de 1 à 0. Ecrire un programme permettant de remplir le tableau et de calculer le nombre de transitions dans le tableau. Le programme doit s'assurer que le tableau n'est rempli que par des 0 et des 1.

**Exemple :** le nombre de transitions dans le tableau suivant est : 5



### Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 20
    int t[n],i,nbTrans;
    printf("Veuillez saisir le tableau:\n");
    for(i=0;i<n;i++){
        do{
            printf("Donner l'etat (0 ou 1) dans l'instant %d: ",i);
            scanf("%d",&t[i]);
        }while(t[i]!=0 && t[i]!=1);
    }
    nbTrans=0;
    for(i=0;i<n-1;i++){
        if(t[i]!=t[i+1])nbTrans=nbTrans+1;
    }
    printf("Le nombre de transitions est: %d",nbTrans);
    return 0;
}

```

### Exercice 87 :

Ecrire un algorithme/programme qui permet à un enseignant de saisir les notes d'examen d'un groupe de 20 étudiants, les stocke dans un tableau, calcule la moyenne du groupe, et qui détermine la note minimale dans le groupe, la note maximale, et le nombre de notes supérieures ou égales à la moyenne du groupe.

### Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 20
    float t[n],min,max,somme,moy;int i,nbSupMoy;
    printf("Veuillez saisir le tableau:\n");

```

```

for(i=0;i<n;i++){
    do{
        printf("Donner la note de l'etudiant %d: ",i+1);
        scanf("%f",&t[i]);
    }while(t[i]<0 || t[i]>20);
}
somme=t[0];min=t[0];max=t[0];
for(i=1;i<n;i++){
    somme=somme+t[i];
    if(t[i]<min)min=t[i];
    else if(t[i]>max)max=t[i];
}
moy=somme/n;
nbSuppMoy=0;
for(i=0;i<n;i++){
    if(t[i]>=moy)nbSuppMoy=nbSuppMoy+1;
}
printf("La moyenne du groupe est: %.2f\n",moy);
printf("La note minimale est: %.2f\n",min);
printf("La note maximale est: %.2f\n",max);
printf("Le nombre de notes >= la moyenne du groupe est: %d",nbSuppMoy);
return 0;
}

```

### Exercice 88 :

Ecrire un programme qui permet à l'utilisateur de remplir un tableau de 10 entiers, en insérant les nombres positifs saisis au début du tableau et les nombres négatifs à la fin du tableau. Ensuite, le programme doit afficher le tableau saisi.

#### Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 10
    int t[n],i,j,k,a;
    printf("Veuillez saisir le tableau:\n");
    j=0;k=n-1;
    for(i=0;i<n;i++){
        printf("Donner le nombre %d: ",i);
        scanf("%d",&a);
        if(a>=0){
            t[j]=a;
            j=j+1;
        }
        else{
            t[k]=a;
            k=k-1;
        }
    }
    printf("Le tableau saisi est:\n");
    for(i=0;i<n;i++){

```

```

        printf("%d | ",t[i]);
    }
    return 0;
}

```

### Exercice 89 :

Ecrire un programme qui permet de saisir un tableau de 12 réels et qui calcule le plus grand écart dans un tableau (l'écart est la valeur absolue de la différence de deux éléments consécutifs).

**Remarque :** n'utilisez pas la fonction prédéfinie `abs()` de la bibliothèque `math.h` pour calculer la valeur absolue.

**Exemple :** le plus grand écart dans le tableau ci-dessous est **12** qui égal à  $|-1 - 11|$

0	1	2	3	4	5	6	7	8	9	10	11
4	-2	-6	5	3	11	-1	4	-4	1	5	5

### Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 12
    float t[n],ecart,ecartMax;int i;
    printf("Veuillez saisir le tableau:\n");
    for(i=0;i<n;i++){
        printf("Case %d: ",i);
        scanf("%f",&t[i]);
    }
    ecartMax=t[1]-t[0];
    if(ecartMax<0)ecartMax=-ecartMax;
    for(i=1;i<n-1;i++){
        ecart=t[i+1]-t[i];
        if(ecart<0)ecart=-ecart;
        if(ecart>ecartMax)ecartMax=ecart;
    }
    printf("L'ecart max est: %.2f",ecartMax);
    return 0;
}

```

### Exercice 90 :

Écrire un programme qui demande à l'utilisateur de saisir 6 entiers non nuls stockés dans un tableau ainsi qu'un entier  $v$ . Le programme doit rechercher si  $v$  se trouve dans le tableau et doit supprimer la première occurrence de  $v$  en décalant d'une case vers la gauche les éléments suivants et en rajoutant 0 à la fin du tableau. Le programme doit ensuite afficher le tableau résultant.

**Exemple :** si la valeur à supprimer est  $v = 8$

0	1	2	3	4	5
3	1	8	1	2	8

→

0	1	2	3	4	5
3	1	1	2	8	0

### Réponse :

```

#include <stdio.h>
#include <stdlib.h>

```

```

int main(){
    #define n 6
    int t[n],i,j,v;
    printf("Veuillez saisir le tableau:\n");
    for(i=0;i<n;i++){
        printf("Case %d: ",i);
        scanf("%d",&t[i]);
    }
    printf("Donner le nombre a supprimer: ");
    scanf("%d",&v);
    i=0;
    while(i<n && t[i]!=v){
        i=i+1;
    }
    if(i==n)printf("La valeur %d n'existe pas dans le tableau",v);
    else{
        for(j=i;j<n-1;j++){
            t[j]=t[j+1];
        }
        t[n-1]=0;
        printf("Le tableau apres la suppression de %d est:\n",v);
        for(i=0;i<n;i++){
            printf("%d | ",t[i]);
        }
    }
    return 0;
}

```

### Exercice 91 :

Modifier le programme de l'exercice précédent pour qu'il supprime toutes les occurrences du nombre cherché et pas uniquement la première occurrence. Donc chaque fois où une occurrence du nombre cherché est trouvée dans le tableau, le programme doit décaler d'une case vers la gauche les éléments suivants et rajouter 0 à la fin du tableau.

### Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 6
    int t[n],i,j,v;
    printf("Veuillez saisir le tableau:\n");
    for(i=0;i<n;i++){
        printf("Case %d: ",i);
        scanf("%d",&t[i]);
    }
    printf("Donner le nombre a supprimer: ");
    scanf("%d",&v);
    i=0;
    while(i<n){
        if(t[i]==v){
            for(j=i;j<n-1;j++){
                t[j]=t[j+1];
            }
        }
        i++;
    }
    t[n-1]=0;
    printf("Le tableau apres la suppression de %d est:\n",v);
    for(i=0;i<n;i++){
        printf("%d | ",t[i]);
    }
}

```

```

        t[n-1]=0;
    }
    else i=i+1;
}
printf("Le tableau apres la suppression de %d est:\n",v);
for(i=0;i<n;i++){
    printf("%d | ",t[i]);
}
return 0;
}

```

### Exercice 92 :

Écrire un programme qui permet de saisir un tableau de 10 entiers supposés non nuls, et puis de l'éclater en deux tableaux : un contiendra les éléments positifs, et l'autre contiendra les éléments négatifs. Le programme doit ensuite afficher les deux tableaux résultants.

Comme on n'a aucune idée sur les valeurs qui vont être saisies par l'utilisateur, les deux tableaux résultants doivent être déclarés de taille égale à la taille du tableau de départ (10 cases), et les cases vides de ces deux tableaux seront remplies par des 0.

#### Exemple :

*Tableau initial*

0	1	2	3	4	5	6	7	8	9
4	-2	-6	5	3	11	-1	4	-4	1

*Tableau des valeurs positives*

0	1	2	3	4	5	6	7	8	9
4	5	3	11	4	1	0	0	0	0

*Tableau des valeurs négatives*

0	1	2	3	4	5	6	7	8	9
-2	-6	-1	-4	0	0	0	0	0	0

#### Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 10
    int t[n],tPos[n],tNeg[n],i,j,k;
    printf("Veuillez saisir le tableau:\n");
    for(i=0;i<n;i++){
        printf("Case %d: ",i);
        scanf("%d",&t[i]);
    }
    j=0;k=0;
    for(i=0;i<n;i++){
        if(t[i]>0){
            tPos[j]=t[i];
            j=j+1;
        }
        else if(t[i]<0){
            tNeg[k]=t[i];
            k=k+1;
        }
    }
    for(i=j;i<n;i++){
        tPos[i]=0;
    }
}

```

```

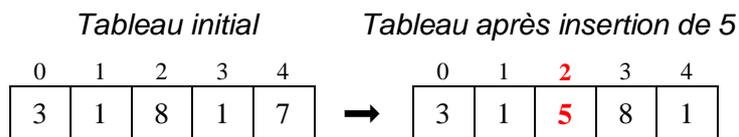
for(i=k;i<n;i++){
    tNeg[i]=0;
}
printf("Le tableau des nombres positifs est:\n");
for(i=0;i<n;i++){
    printf("%d | ",tPos[i]);
}
printf("\nLe tableau des nombres negatifs est:\n");
for(i=0;i<n;i++){
    printf("%d | ",tNeg[i]);
}
return 0;
}

```

### Exercice 93 :

Écrire un programme qui permet de saisir un tableau de 5 entiers et qui ensuite insère un nombre entier dans une case spécifiée dans le tableau. Le nombre à insérer  $nb$  et le numéro de la case dans laquelle le nombre va être inséré  $num$ , doivent être saisis par l'utilisateur. L'insertion se fait en décalant d'une case vers la droite les éléments à partir de la case numéro  $num$  et en stockant  $nb$  dans la case numéro  $num$ . De ce fait, le dernier élément du tableau initial sera enlevé.

**Exemple** : si la valeur à insérer est  $nb = 5$  dans la case numéro  $num=2$ , on obtient :



### Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 5
    int t[n],i,nb,num;
    printf("Veuillez saisir le tableau:\n");
    for(i=0;i<n;i++){
        printf("Case %d: ",i);
        scanf("%d",&t[i]);
    }
    printf("Donner le nombre a inserer: ");
    scanf("%d",&nb);
    do{
        printf("Dans quelle case voulez vous l'inserer? ");
        scanf("%d",&num);
    }while(num<0 || num>n-1);
    for(i=n-1;i>num;i--){
        t[i]=t[i-1];
    }
    t[num]=nb;
    printf("Le tableau apres l'insertion de %d est:\n",nb);
    for(i=0;i<n;i++){
        printf("%d | ",t[i]);
    }
    return 0;
}

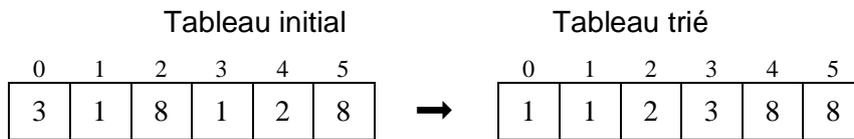
```

```
}
```

### Exercice 94 :

Écrire un programme qui permet de saisir un tableau de 6 entiers et de le trier ensuite en ordre croissant. Le programme doit ensuite afficher le tableau résultant.

**Exemple :**



**Réponse :**

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 6
    int t[n],i,j,a,min,posMin;
    printf("Veuillez saisir le tableau:\n");
    for(i=0;i<n;i++){
        printf("Case %d: ",i);
        scanf("%d",&t[i]);
    }
    for(i=0;i<n;i++){
        min=t[i];
        posMin=i;
        for(j=i+1;j<n;j++){
            if(t[j]<min){
                min=t[j];
                posMin=j;
            }
        }
        a=t[i];
        t[i]=t[posMin];
        t[posMin]=a;
    }
    printf("Le tableau triee est:\n");
    for(i=0;i<n;i++){
        printf("%d | ",t[i]);
    }
    return 0;
}
```

### Exercice 95 :

Écrire un programme permettant de remplir une matrice de 10 lignes et 10 colonnes de façon à ce que la valeur contenue dans chaque case soit le produit de son numéro de ligne et de son numéro de colonne, et de l'afficher.

**Exemple :** la matrice obtenue est:

	0	1	2	...	j	...
0	0	0	0	...	0	...
1	0	1	2	...	j	...
2	0	2	4	...	2j	...
...	...	...	...	...	...	...

$i$	0	$i$	$2i$	...	$i*j$	...
...	...	...	...	...	...	...

### Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main() {
#define n 10
    int m[n][n], i, j;
    for(i=0; i<n; i++) {
        for(j=0; j<n; j++) {
            m[i][j]=i*j;
        }
    }
    printf("La matrice est:\n");
    for(i=0; i<n; i++) {
        for(j=0; j<n; j++) {
            printf("%d\t", m[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

### Exercice 96 :

Ecrire un programme permettant de saisir une matrice de taille  $N \times N$  de réels, et qui calcule et affiche la somme de sa diagonale.

### Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main() {
#define n 4
    float m[n][n], s; int i, j;
    printf("Veuillez saisir la matrice:\n");
    for(i=0; i<n; i++) {
        for(j=0; j<n; j++) {
            printf("Case [%d,%d]: ", i, j);
            scanf("%f", &m[i][j]);
        }
    }
    s=0;
    for(i=0; i<n; i++) {
        s=s+m[i][i];
    }
    printf("La somme de la diagonale est: %.2f", s);
    return 0;
}
```

### Exercice 97 :

Considérant une matrice de 5 lignes et de 4 colonnes d'entiers. On veut calculer la somme d'éléments de chaque ligne et de chaque colonne et de les stocker pour d'éventuelles post-

traitement. Pour cela, on propose d'ajouter une ligne et une colonne (à la fin) qui vont stocker la somme d'éléments de chaque colonne et de chaque ligne respectivement. Ecrire un programme permettant de saisir les éléments de la matrice, et qui ensuite calcule la somme de valeurs par lignes et par colonnes et les affiche.

### Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    #define n 6
    #define m 5
    int t[n][m], i, j;
    printf("Veuillez saisir la matrice:\n");
    for(i=0; i<n-1; i++) {
        for(j=0; j<m-1; j++) {
            printf("Case [%d,%d]: ", i, j);
            scanf("%d", &t[i][j]);
        }
    }

    for(i=0; i<n-1; i++) {
        t[i][m-1]=0;
        for(j=0; j<m-1; j++) {
            t[i][m-1]=t[i][m-1]+t[i][j];
        }
    }
    for(j=0; j<m; j++) {
        t[n-1][j]=0;
        for(i=0; i<n-1; i++) {
            t[n-1][j]=t[n-1][j]+t[i][j];
        }
    }
    printf("La somme par ligne est:\n");
    for(i=0; i<n-1; i++) {
        printf("\t%d\n", t[i][m-1]);
    }
    printf("La somme par colonne est:\n");
    for(j=0; j<m-1; j++) {
        printf("%d\t", t[n-1][j]);
    }
    return 0;
}
```

### Exercice 98 :

Ecrire un programme permettant de saisir une matrice d'entiers de taille  $N \times N$ , et de tester si elle s'agit de la matrice identité ou pas. Dans la matrice identité, tous les éléments de sa diagonale sont égaux à 1 et tous les autres éléments sont égaux à 0.

### Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    #define n 4
```

```

int t[n][n],i,j,identite;
printf("Veuillez saisir la matrice:\n");
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        printf("Case [%d,%d]: ",i,j);
        scanf("%d",&t[i][j]);
    }
}
i=0;
identite=1;//variable logique qui prend 1 si la matrice est la matrice
identite et 0 sinon
while(i<n && identite==1){
    j=0;
    while(j<n && identite==1){
        if(i==j && t[i][j]!=1)identite=0;
        else if(i!=j && t[i][j]!=0)identite=0;
        j=j+1;
    }
    i=i+1;
}
if(identite==1)printf("La matrice saisie est la matrice identite");
else printf("La matrice saisie n'est pas la matrice identite");
return 0;
}

```

### Exercice 99 :

Ecrire un programme permettant de saisir une matrice symétrique d'entiers de taille  $N \times N$ , et de l'afficher. Dans une matrice symétrique, lorsque l'utilisateur saisit la valeur de la case  $[i, j]$ , cette valeur est stocké également dans la case  $[j, i]$  d'une façon automatique.

### Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 4
    int t[n][n],i,j;
    printf("Veuillez saisir la matrice:\n");
    for(i=0;i<n;i++){
        for(j=i;j<n;j++){
            printf("Case [%d,%d]: ",i,j);
            scanf("%d",&t[i][j]);
            t[j][i]=t[i][j];
        }
    }
    printf("La matrice saisie est:\n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            printf("%d\t",t[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

```
}
```

### Exercice 100 :

Ecrire un programme permettant de saisir une matrice d'entiers de taille  $N \times N$ , et de tester si elle est symétrique par rapport à son diagonale.

**Exemple :** La matrice suivante est symétrique

1	4	2	5
4	1	1	7
2	1	1	3
5	7	3	1

### Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 4
    int t[n][n],i,j,estSymetrique;
    printf("Veuillez saisir la matrice:\n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            printf("Case [%d,%d]: ",i,j);
            scanf("%d",&t[i][j]);
        }
    }
    estSymetrique=1;
    i=0;
    while(i<n && estSymetrique==1){
        j=0;
        while(j<n && estSymetrique==1){
            if(t[i][j]!=t[j][i])
                estSymetrique=0;
            else j=j+1;
        }
        i=i+1;
    }
    if(estSymetrique==1)printf("La matrice saisie est symetrique");
    else printf("La matrice saisie n'est pas symetrique");
    return 0;
}
```

### Exercice 101 :

Ecrire un programme permettant de saisir une matrice de 5 lignes et de 4 colonnes et de tester si elle est creuse ou pas. On dit qu'une matrice est creuse lorsque plus que deux tiers ( $2/3$ ) de ses éléments sont nuls.

**Exemple :** La matrice suivante est creuse car 14 de ses éléments sont nuls ( $14 > 20 \cdot 2/3$ ).

0	0	2	0
0	1	0	0
2	0	0	3
0	0	3	0
1	0	0	0

## Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    #define n 5
    #define m 4
    int t[n][m], i, j, nbZeros;
    printf("Veuillez saisir la matrice:\n");
    for(i=0; i<n; i++) {
        for(j=0; j<m; j++) {
            printf("Case [%d,%d]: ", i, j);
            scanf("%d", &t[i][j]);
        }
    }
    nbZeros=0;
    for(i=0; i<n; i++) {
        for(j=0; j<m; j++) {
            if(t[i][j]==0) nbZeros=nbZeros+1;
        }
    }
    if(nbZeros>((float)2/3)*n*m) printf("La matrice saisie est creuse");
    else printf("La matrice saisie n'est pas creuse");
    return 0;
}
```

## Exercice 102 :

Ecrire un programme permettant de saisir une matrice d'entiers de 4 lignes et de 4 colonnes, de l'alléger, et de l'afficher. Alléger une matrice revient à mettre à zéro (0) tous ses éléments sauf les éléments de la diagonale. Ce dernier contiendra la somme par colonne. Ainsi, la case  $[i, i]$  de la diagonale stockera la somme des éléments de la colonne  $i$ .

### Exemple :

Matrice initiale					Matrice allégée			
1	4	2	5	→	<b>11</b>	0	0	0
3	0	1	7		0	<b>7</b>	0	0
2	1	1	3		0	0	<b>8</b>	0
5	2	4	1		0	0	0	<b>16</b>

## Réponse :

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    #define n 4
    int t[n][n], i, j, s;
    printf("Veuillez saisir la matrice:\n");
    for(i=0; i<n; i++) {
        for(j=0; j<n; j++) {
            printf("Case [%d,%d]: ", i, j);
            scanf("%d", &t[i][j]);
        }
    }
}
```

```

for(j=0;j<n;j++){
    s=0;
    for(i=0;i<n;i++){
        s=s+t[i][j];
        if(i!=j)t[i][j]=0;
    }
    t[j][j]=s;
}
printf("La matrice allegee est:\n");
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        printf("%d\t",t[i][j]);
    }
    printf("\n");
}
return 0;
}

```

### Exercice 103 :

Ecrire un programme permettant de saisir une matrice d'entiers de 6 lignes et de 6 colonnes, et de trouver la plus longue séquence de 0 dans la matrice. La recherche de la plus longue séquence de 0 se fait selon les deux directions : horizontale et verticale. Le programme doit afficher la longueur de cette séquence, le numéro de ligne et de colonne de début, et le numéro de ligne et de colonne de fin.

**Exemple :** dans la matrice ci-dessous, la longueur de la plus longue séquence de 0 est 5. Elle commence de la case **[1,4]** et se termine dans la case **[5,4]**.

	0	1	2	3	4	5
0	5	0	0	2	17	0
1	2	4	3	0	0	0
2	1	12	0	0	0	3
3	1	3	0	5	0	11
4	7	0	0	9	0	20
5	0	0	0	8	0	6

### Réponse :

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    #define n 6
    int t[n][n],i,j,lon,lonMax,ligDeb,ligFin,colDeb,colFin,deb,fin;
    printf("Veuillez saisir la matrice:\n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            printf("Case [%d,%d]: ",i,j);
            scanf("%d",&t[i][j]);
        }
    }
    lonMax=0;ligDeb=0;ligFin=0;colDeb=0;colFin=0;
    for(i=0;i<n;i++){
        j=0;

```

```

while(j<n){
    while(j<n && t[i][j]!=0)
        j=j+1;
    if(j<n){
        deb=j;
        j=j+1;
        while(j<n && t[i][j]==0)
            j=j+1;
        fin=j-1;
        lon=fin-deb+1;
        if(lon>lonMax){
            lonMax=lon;
            ligDeb=i;
            ligFin=i;
            colDeb=deb;
            colFin=fin;
        }
    }
}
for(j=0;j<n;j++){
    i=0;
    while(i<n){
        while(i<n && t[i][j]!=0)
            i=i+1;
        if(i<n){
            deb=i;
            i=i+1;
            while(i<n && t[i][j]==0)
                i=i+1;
            fin=i-1;
            lon=fin-deb+1;
            if(lon>lonMax){
                lonMax=lon;
                colDeb=j;
                colFin=j;
                ligDeb=deb;
                ligFin=fin;
            }
        }
    }
}
if(lonMax==0)printf("Il n'y a pas de sequence de 0 dans la matrice");
else{
    printf("la longueur de la plus longue sequence de 0 est:
%d\n",lonMax);
    printf("Elle commence de la case [%d,%d] ",ligDeb,colDeb);
    printf("et se termine dans la case [%d,%d]",ligFin,colFin);
}
return 0;
}

```