

Ecrire un programme C qui lit un entier n , puis n autres entiers positifs dans un tableau, l'affiche puis calcul la somme, le max, et le min de ses éléments.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int t[100];
    int i,n,som,max,min;
    printf("Donnez le nombre des elements du tableau:\n");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Donnez l'element %d:\n",i+1);
        scanf("%d",&t[i]);
    }
    som=0;
    min=t[0];
    max=t[0];
    for(i=0;i<n;i++)
    {
        printf("%d ",t[i]);
        som+=t[i];
        if(t[i]>max) max=t[i];
        if(t[i]<min) min=t[i];
    }
    printf("\n");
    printf("La somme est: %d\n",som);
    printf("Le max est: %d\n",max);
    printf("Le min est: %d\n",min);
    system("pause");
    return 0;
}
```

Ecrire un programme C permettant de Comparer deux tableaux T1 et T2 de N entiers (remplis par l'utilisateur). Deux tableaux sont égaux si tous les éléments se trouvant dans le même indice sont égaux.

```
#include<stdio.h>
main()
{
    int n, i, c, T1[30], T2[30];

    printf("Entrer la taille des deux tableaux: ");
    scanf("%d",&n);

    printf("\nRemplissage du tableau 1: \n");
    for(i=0; i<n; i++){
        printf("Entrer l'element %d :", i+1);
        scanf("%d",&T1[i]);
    }
    printf("\nRemplissage du tableau 2: \n");
    for(i=0; i<n; i++){
        printf("Entrer l'element %d :", i+1);
        scanf("%d",&T2[i]);
    }

    c = 1;
    for(i=0; i<n; i++){
        if(T1[i] != T2[i])
            c = 0;
    }
    if(c == 1)
        printf("\nLes deux tableaux sont identiques");
}
```

```

        else
            printf("\nLes deux tableaux ne sont pas
            identiques");
    }

```

Créer des programmes en C qui permettent de chercher une valeur X (entrée au clavier) dans un tableau d'entiers avec les possibilités suivantes:

1) Déterminer si l'entier X appartient au tableau T.

```

#include<stdio.h>
main()
{
    int n, i, T[30], x, p[30], c;
    printf("Entrer la taille du tableau: ");
    scanf("%d",&n);
    printf("\nRemplissage du tableau: \n");
    for(i=0; i<n; i++){
        printf("Entrer l'element %d :", i+1);
        scanf("%d",&T[i]);
    }
    printf("\nEntrer la valeur a chercher: ");
    scanf("%d",&x);

    c = 0;
    for(i=0; i<n; i++){
        if(T[i] == x){
            p[c] = i;
            c = c + 1;
        }
    }

    if(c == 0)
        printf("\nLa valeur %d n'existe pas dans le tableau.", x);
    else{
        printf("\nLes positions de la valeur %d sont:\n", x);
        for(i=0; i<c; i++){
            printf("%d\n",p[i] + 1);
        }
    }
}

```

1) Élaborer une matrice à deux dimensions, avec des composants entiers, par initialisation au moment de la déclaration.

```

#include<stdio.h>
void main()
{
    int a[4][3]={
        {2,-9,0},
        {7,11,-8},
        {2,2,-20},
        {12,7,90}
    };

    //l'operation d'affichage de la matrice avec 4 lignes et 3 colonnes
    for(int i=0; i<4; i++)
    {
        for(int j=0; j<3; j++)
            printf("%5d", a[i][j]);
        printf("\n");
    }
}

```

2) Ecrire une routine pour lire une matrice à deux dimensions avec des composants entiers, et l'afficher après.

```
#include<stdio.h>
void main()
{
    int a[10][10]; /* matrice à deux dimensions de maximum 10 lignes si 10 colonnes
pour memorer */
    int m,n;      /* nombre reel (à la lecture) des rangées et des colonnes de la
matrice */
    int i,j;
    //introduire les dimensions de la matrice
    printf("Entrer le nombre des lignes: ");
    scanf("%d", &m);
    printf("Entrer le nombre des colonnes: ");
    scanf("%d", &n);
    //Entrer les elements de la matrice
    printf("Entrer les elements de la matrice: \n");
    for(i=0; i<m; i++)
        for(j=0; j<n; j++)
        {
            printf("\tElem [%d,%d]= ", i+1, j+1);
            scanf("%d", &a[i][j]);
        }
    //affichage des elements de la matrice
    for(i=0; i<m; i++)
    {
        for(j=0; j<n; j++)
            printf("%5d", a[i][j]);
        printf("\n");
    }
    getchar();
    int k;
    scanf("%d", k);
}
```

écrire un programme qui demande à l'utilisateur d'entrer deux matrices et après il calcule la multiplication de ces deux matrices.

Ex :

Multiplication

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \times \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} = \begin{pmatrix} 1*1 + 2*3 + 3*5 & 1*2 + 2*4 + 3*6 \\ 4*1 + 5*3 + 6*5 & 4*2 + 5*4 + 6*6 \end{pmatrix}$$

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

int main(){
    int mat1[10][10], mat2[10][10], mul[10][10];
    int n,m,i,j,k;

    printf("Entrez le nombre de lignes et de colonnes:\n");
    scanf("%d%d", &n, &m);

    printf("Entrez les éléments de la première matrice\n");
```

```

for(i = 0; i < n; i++)
{
    for(j = 0; j < m; j++)
    {
        scanf("%d",&mat1[i][j]);
    }
}

printf("Entrez les éléments de la deuxième matrice\n");
for(i = 0; i < n; i++)
{
    for(j = 0; j < m; j++)
    {
        scanf("%d",&mat2[i][j]);
    }
}

printf("Multiplication du matrice = \n");
for(i = 0; i < n; i++)
{
    for(j = 0; j < m; j++)
    {
        mul[i][j]=0;
        for(k = 0; k < m; k++)
        {
            mul[i][j] += mat1[i][k] * mat2[k][j];
        }
    }
}
//Afficher le résultat
for(i = 0; i < n; i++)
{
    for(j = 0; j < m; j++)
    {
        printf("%d\t",mul[i][j]);
    }
    printf("\n");
}
return 0;
}

```