

Algorithmique et structures de données 1

TP 4

Exercice 1. Ecrire un programme C qui permet de chercher un caractère dans une chaîne et compter toutes ces occurrences.

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

int main ()
{
    char chaine[100], c;
    int i, nbocc=0;
    printf("Entrer une chaîne de caractères : \n");
    gets(chaine);
    printf("Entrez le caractère à rechercher : ");
    c=getchar();

    while(chaine[i] != '\0') // '\0' est le dernier élément (caractère nul)
    {
        if(chaine[i]==c) {
            nbocc++;
        }
        i++;
    }
    printf("Le caractère '%c' apparaît %d fois \n", c, nbocc);
}
```

Exercice 2. Ecrire un programme C qui permet de compter le nombre total de voyelles ou de consonnes dans une chaîne.

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int main ()
{
    char chaine[100];
    int i, l, v, c;
    printf("Entrer une chaîne de caractères : \n");
    gets(chaine);

    v = 0;
    c = 0;
    l = strlen(chaine);

    for(i=0; i<l; i++)
    {
        if(chaine[i] == 'a' || chaine[i]=='e' || chaine[i]=='i' || chaine[i]=='o' || chaine[i]=='u'
           || chaine[i]=='A' || chaine[i]=='E' || chaine[i]=='I' || chaine[i]=='O' || chaine[i]=='U')
        {
            v++;
        }
        else if((chaine[i]>='a' && chaine[i]<='z') || (chaine[i]>='A' && chaine[i]<='Z'))
        {
            c++;
        }
    }
    printf("\nLe nombre total de voyelles dans la chaîne est : %d\n", v);
    printf("Le nombre total de consonnes dans la chaîne est : %d\n\n", c);
}

```

Exercice 3. Ecrire un programme C pour vérifier si une chaîne est palindrome ou non. Par Exemple: coloc, été, gag, radar, pop, elle ou kayak sont toutes des chaînes de caractères palindromes.

```

#include <stdio.h>
#include <string.h>

int main(){
    char chaine[100];
    int i, n, palindrome = 0;

    printf("Entrez une chaine de caractère: ");
    gets(chaine);
    n = strlen(chaine);

    for(i=0;i < n ;i++){
        if(chaine[i] != chaine[n-i-1]){
            palindrome = 1;
            break;
        }
    }

    if (palindrome) {
        printf("%s n'est pas un palindrome", chaine);
    }
    else {
        printf("%s est un palindrome", chaine);
    }
    return 0;
}

```

Exercice 4. Ecrire un programme C pour lire les éléments d'un tableau et calculer la somme et la moyenne des éléments du tableau.

```

#include<stdio.h>
#include<stdlib.h>
int main()
{
    int i, n, t[100], s=0;
    float m;

    /* Taille d'entrée du tableau */
    printf("Donnez la taille du tableau:\n");
    scanf("%d",&n);

    /* Éléments d'entrée dans le tableau */
    printf("Entrez %d element(s):\n",n);

    for(i=0;i<n;i++){
        scanf("%d",&t[i]);
        s += t[i];
    }
    m = (float)s/n;
    printf("La somme = %d\n", s);
    printf("La moyenne = %f", m);
}

```

Exercice 5. Ecrire un programme C pour trouver le plus petit et le plus grand nombre avec leurs positions dans un tableau.

```

#include<stdio.h>
int main()
{
    int t[100], n, i, Min, Min_Position, Max, Max_Position;

    printf("\nVeuillez entrer la taille du tableau : ");
    scanf("%d",&n);

    printf("\nVeuillez entrer %d éléments d'un tableau : \n", n);
    for(i=0; i<n; i++)
    {
        scanf("%d",&t[i]);
    }

    Min = t[0];
    Max = t[0];

    for(i=1; i<n; i++)
    {
        if(Min > t[i])
        {
            Min = t[i];
            Min_Position = i;
        }
        if(Max < t[i])
        {
            Max=t[i];
            Max_Position = i;
        }
    }

    printf("\n Le plus petit élément du tableau = %d", Min);
    printf("\n Position d'index du plus petit élément = %d", Min_Position);

    printf("\n Le plus grand élément du tableau = %d", Max);
    printf("\n Position d'index de l'élément le plus grand = %d", Max_Position);
    return 0;
}

```

Exercice 6. Ecrire un programme C pour entrer des éléments dans un tableau et trier les éléments du tableau dans l'ordre croissant (ou décroissant).

```

#include <stdio.h>
#define taille_max 100 // Taille maximale du tableau
int main()
{
    int t[taille_max];
    int n;
    int i, j, temp;

    /* Taille d'entrée du tableau */
    printf("Entrez la taille du tableau : ");
    scanf("%d", &n);

```

```

/* Éléments d'entrée dans le tableau */
printf("Entrer des éléments dans le tableau : \n");
for(i=0; i<n; i++)
{
    scanf("%d", &t[i]);
}

for(i=0; i<n; i++)
{
    /*
    * Place l'élément actuellement sélectionné t[i]
    * à sa place correcte.
    */
    for(j=i+1; j<n; j++)
    {
        /*
        * Swap (échanger) si l'élément de tableau actuellement sélectionné
        * n'est pas à sa position correcte.
        */
        if(t[i] > t[j])
        {
            temp = t[i];
            t[i] = t[j];
            t[j] = temp;
        }
    }
}

/* Afficher le tableau trié */
printf("\nÉléments du tableau dans l'ordre croissant : \n");
for(i=0; i<n; i++)
{
    printf("%d\t", t[i]);
}

```

```
return 0;
}
```

Exercice 7. Ecrire un programme C pour entrer des éléments dans un tableau et placer les éléments pairs et impairs dans des tableaux séparés.

```
#include <stdio.h>

void main()
{
    int t1[100], t2[100], t3[100];
    int i,j=0,k=0,n;

    printf("\n\nSeparez les nombres entiers pairs et impairs dans des tableaux separes :\n");
    printf("-----\n");

    printf("Entrez le nombre d'elements à stocker dans le tableau :");
    scanf("%d",&n);

    printf("Entrez %d elements dans le tableau :\n",n);
    for(i=0;i<n;i++)
    {
        printf("element - %d : ",i);
        scanf("%d",&t1[i]);
    }

    for(i=0;i<n;i++)
    {
        if (t1[i]%2 == 0)
        {
            t2[j] = t1[i];
            j++;
        }
        else
        {
```

```

        t3[k] = t1[i];
        k++;
    }
}

printf("\nLes elements pairs sont : \n");
for(i=0;i<j;i++)
{
    printf("%d ", t2[i]);
}

printf("\nLes elements impairs sont : \n");
for(i=0;i<k;i++)
{
    printf("%d ", t3[i]);
}
printf("\n\n");
}

```

Exercice 8. Ecrire un programme C qui permet de faire l'addition de deux matrices de même taille.

```

#include <stdio.h>

void main()
{
    int a[50][50],b[50][50],c[50][50],i,j,n;

    printf("\n\nAddition de deux Matrices : \n");
    printf("-----\n");
    printf("Entrez la taille de la matrice carree : ");
    scanf("%d", &n);

    /* Valeurs stockées dans la matrice*/
    printf("Éléments d'entrée dans la première matrice : \n");
    for(i=0;i<n;i++)
    {

```

```

    for(j=0;j<n;j++)
    {
        printf("element - [%d],[%d] : ",i,j);
        scanf("%d",&a[i][j]);
    }
}

printf("Éléments d'entrée dans la deuxième matrice :\n");
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        printf("element - [%d],[%d] : ",i,j);
        scanf("%d",&b[i][j]);
    }
}

printf("\nLa première matrice est :\n");
for(i=0;i<n;i++)
{
    printf("\n");
    for(j=0;j<n;j++)
        printf("%d\t",a[i][j]);
}

printf("\nLa deuxième matrice est :\n");
for(i=0;i<n;i++)
{
    printf("\n");
    for(j=0;j<n;j++)
        printf("%d\t",b[i][j]);
}

/* calculer la somme de la matrice */
for(i=0;i<n;i++)
    for(j=0;j<n;j++)
        c[i][j]=a[i][j]+b[i][j];

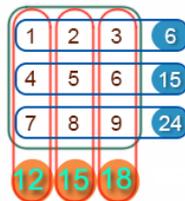
```

```

printf("\nL'addition de deux matrices est : \n");
for(i=0;i<n;i++){
    printf("\n");
    for(j=0;j<n;j++){
        printf("%d\t",c[i][j]);
    }
    printf("\n\n");
}

```

Exercice 9. Ecrire un programme C qui permet de lire les éléments d'une matrice et calculer la somme des éléments de chaque ligne et de chaque colonne de la matrice. Par exemple :



```

#include <stdio.h>
void main ()
{
    int a[100][100];
    int i, j, m, n, sum = 0;

    printf("Entrez la taille de la matrice : \n");
    scanf("%d %d", &m, &n);

    printf("Entrez des elements dans la matrice\n");
    for (i = 0; i < m; ++i)
    {
        for (j = 0; j < n; ++j)
        {
            scanf("%d", &a[i][j]);
        }
    }

    /* Calculer la somme des éléments de chaque ligne de la matrice */
    for (i = 0; i < m; ++i)

```

```

{
    for (j = 0; j < n; ++j)
    {
        sum = sum + a[i][j] ;
    }
    printf("La somme de la ligne %d est = %d\n", i, sum);
    sum = 0;
}

/* Calculer la somme des éléments de chaque colonne de la matrice */
    sum = 0;
for (j = 0; j < n; ++j)
{
    for (i = 0; i < m; ++i)
    {
        sum = sum + a[i][j];
    }
    printf("La somme de la colonne %d est = %d\n", j, sum);
    sum = 0;
}
}

```

Exercice 10. Ecrire un programme C pour la multiplication de deux matrices. La multiplication de deux matrices est définie comme suite :

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}$$

```

#include <stdio.h>
int main() {
    int i,j,k,al,ac,bl,bc;
    printf("Donner la taille (AL * AC) de la matrice A : ");
    scanf("%d%d",&al,&ac);
    int A[al][ac];
    printf("Entrer les éléments de la matrice A : ");
    for(i=0;i<al;i++){

```

```

        for(j=0;j<ac;j++){
            printf("A[%d][%d] = ", i, j);
            scanf("%d", &A[i][j]);
        }
    }

    printf("Donner la taille (BL * BC) de la matrice B : ");
    scanf("%d%d",&bl,&bc);
    int B[bl][bc];
    printf("Entrer les éléments de la matrice B : ");
    for(i=0;i<bl;i++){
        for(j=0;j<bc;j++){
            printf("B[%d][%d] = ", i, j);
            scanf("%d", &B[i][j]);
        }
    }

    if(ac==bl){
        int C[al][bc];
        printf("La matrice C : \n");
        for(i=0;i<al;i++){
            for(j=0;j<bc;j++){
                C[i][j]=0;
                for(k=0;k<ac;k++){
                    C[i][j] = C[i][j] + A[i][k]*B[k][j];
                }
                printf("%d\t",C[i][j]);
            }
            printf("\n");
        }
    }
    else
        printf("Impossible de faire la multiplication entre les deux matrices. ");
}

```

Exercice 11. Ecrire un programme en C pour trouver la transposée d'une matrice donnée. La transposition d'une matrice A est définie comme la conversion de toutes les lignes en colonnes et toutes les colonnes en lignes.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

```
#include <stdio.h>

void main()
{
    static int a[100][100];
    int i, j, m, n;

    printf("Entrez la taille de la matrice : \n");
    // Saisie d'éléments dans la matrice par l'utilisateur
    scanf("%d %d", &m, &n);
    printf("Entrez des elements dans la matrice : \n");
    for (i = 0; i < m; ++i)
    {
        for (j = 0; j < n; ++j)
        {
            scanf("%d", &a[i][j]);
        }
    }
    //Affichage de la matrice originale
    printf("La matrice donnée est : \n");
    for (i = 0; i < m; ++i)
    {
        for (j = 0; j < n; ++j)
        {
            printf(" %d", a[i][j]);
        }
        printf("\n");
    }
    //Affichage de la transposition de matrice
    printf("La transposée de la matrice est : \n");
    for (j = 0; j < n; ++j)
    {
        for (i = 0; i < m; ++i)
        {
            printf(" %d", a[i][j]);
        }
        printf("\n");
    }
}
```

```
#include <stdio.h>
int main() {
    int i,j,n,m;
```

```

printf("Donner la taille (n * n) de la matrice A : ");
scanf("%d%d",&n, &m);
float A[n][m],At[m][n];
printf("Entrer les éléments de la matrice A : ");
for(i=0;i<n;i++){
    for(j=0;j<m;j++){
        printf("A[%d][%d] = ", i, j);
        scanf("%f", &A[i][j]);
        At[j][i] = A[i][j];
    }
}

printf("Le transposé de la matrice A est : \n");
for(i=0;i<m;i++){
    for(j=0;j<n;j++){
        printf("%f\t",At[i][j]);
    }
    printf("\n");
}
}

```