

Série 3

Exercice 1

1. Ecrire un programme qui teste si un nombre entier est premier ou non.
2. Modifier le programme pour qu'il affiche le nombre de nombres premiers inférieurs à 10000.

Solution:

1)

```
#include<stdio.h>
#include<math.h>
int i,x;
main()
{
    printf("entrer un nombre \t") ;
    scanf("%d",&x);
    if(x==1)
        printf("%d est premier\n",x);
    if(x==2)
        printf("%d est premier\n",x);
    i=2;
    while(i<=sqrt(x) && x%i!=0)
        i++;
    if(x%i==0)
        printf("%d est non premier\n",x);
    else
        printf("%d est premier\n",x);
}
```

```

    }

b)

#include<stdio.h>

#include<math.h>

int i,n,nomb;

main()
{
    nomb=2;
    for(n=3;n<=10000;n++)
    {
        i=2;
        while(i<=sqrt(n) && n%i!=0)
            i++;
        if(n%i!=0)
            nomb=nomb+n;
    }
    printf("le nombre de nombres premiers <10000 est %d\n",nomb);
}

```

Exercice 2

Calculez le N -ième terme U_N de la suite de FIBONACCI qui est donnée par la relation de récurrence:

$$\begin{aligned}
 U_1 &= 1 \\
 U_2 &= 1 \\
 U_N &= U_{N-1} + U_{N-2} \quad (\text{pour } N > 2)
 \end{aligned}$$

Solution:

```

#include<stdio.h>
int n,U,V,temp,i;
main()
{
    printf("entrer la valeur de n \t");
    scanf("%d",&n);
}

```

```

U=1;
V=1;
for(i=3;i<=n;i++)
{
    temp=V;
    V=V+U;
    U=temp;
}
printf("le %d eme terme de la serie est  %d \n",n,V);
}

```

Exercice 3

Ecrire un programme qui détermine la plus grande et la plus petite valeur dans un tableau d'entiers. Afficher ensuite la valeur et la position du maximum et du minimum. Si le tableau contient plusieurs maxima ou minima, le programme retiendra la position du premier maximum ou minimum rencontré.

Solution

```

#include<stdio.h>

int T[100];

int n,pos_max,pos_min,i;

main()
{
    printf("entrer le nombre d'elemets n \t") ;

    scanf("%d",&n);

    for(i=0;i<n;i++)

        {

            printf("entrer T[%d] \n",i) ;

            scanf("%d",&T[i]);

        }

    pos_max=0;

    pos_min=0;

    for(i=1;i<n;i++)

```

```

{
    if(T[i]< T[pos_min])
        pos_min=i;

    if(T[i]>T[pos_max])
        pos_max=i;
}

printf("la plus grande valeur du tableau est %d et sa position est
%d\n",T[pos_max],pos_max+1);

printf("la plus petite valeur du tableau est %d et sa position est %d\n",T[pos_min],pos_min+1);

}

```

Exercice 4

Ecrire un programme qui permet de calculer la matrice C produit de deux matrices A(N,M) et B(M,L) .

$$C=A*B$$

Solution:

```

#include<stdio.h>
int A[20][20],B[20][20],C[20][20];
int n,m,l,i,j,k;
main()
{
    printf("entrer le nombre de lignes de la matrice A \t") ;
    scanf("%d",&n);
    printf("entrer le nombre de collones de A de lignes de la matrice B \t") ;
    scanf("%d",&m);
    printf("entrer le nombre de collones de la matrice B \t") ;
    scanf("%d",&l);
    for(i=0;i<n;i++)
    for(j=0;j<m;j++)
    {
        printf("entrer A[%d][%d]\n",i,j) ;
        scanf("%d",&A[i][j]);
    }
    for(i=0;i<m;i++)
    for(j=0;j<l;j++)
    {
        printf("entrer B[%d][%d]\n",i,j) ;
    }
}

```

```

        scanf("%d",&B[i][j]);
    }
    for(i=0;i<n;i++)
    for(j=0;j<l;j++)
    {
        C[i][j]=0;
        for(k=0;k<m;k++)
            C[i][j]+=A[i][k]*B[k][j];
    }
    for(i=0;i<n;i++)
    {
        for(j=0;j<l;j++)
            printf("%d\t",C[i][j]);
        printf("\n");
    }
}

```