

**Dans cette partie, on va donner les programmes en C pour les exercices 1, 2, 3, et 04**

**Exercice 01:** Fonction en C qui retourne la partie entière d'un nombre réel positif en utilisant les soustractions successives.

```
#include <stdio.h>
float Entiere(float x)
{
    int y;
    y=0;
    while (x>=1)
    {
        x=x-1;
        y=y+1;
    }
    return y;
}
int main()
{
    float x, y;
    printf("\n Donner une valeur réelle positive = ");
    scanf("%f", &x);
    y= Entiere(x);
    printf("\n la partie entiere de %f est %f", x, y);
}
```

**Exercice 2** Ecrire une fonction, puis une procédure, (en C) qui calcule le PGCD de deux entiers strictement positifs.

```

/* PGCD est une fonction*/
#include <stdio.h>
int PGCD_f(int a, int b)
{
    while (a*b!=0)
    {
        if (a>b)
            a=a-b;
        else
            b=b-a;
    }
    if (a==0)
        return b;
    else
        return a;
}
int main()
{
    int x, y, t;
    printf("\n Donner deux valeurs entieres positive = ");
    scanf("%f", &x);
    scanf("%f", &y);
    t=PGCD_f(x, y);
    printf("\n le PGCD de %d et %d est %d", x, y, t);
    return 0;
}

```

```

/* PGCD est une procedure*/
#include <stdio.h>
void PGCD_proc(int a, int b)
{
    int x, y, t;
    x=a; y=b;
    while (a*b!=0)
    {
        if (a>b)
            a=a-b;
        else
            b=b-a;
    }
    if (a==0)
        t=b;
    else
        t=a;
    printf("\n le PGCD de %d et %d est %d", x, y, t);
}
int main()
{
    int x, y, t;
    printf("\n Donner deux valeurs entieres positive = ");
    scanf("%f", &x);
    scanf("%f", &y);
    PGCD_f(x, y);
    return 0;
}

```

**Exercice 3** Ecrire une fonction en C qui prend en entrée un nombre entier positif et retourne son image miroir.

```
/* miroir est une fonction*/
#include <stdio.h>
int miroir(int x)
{
    int a, y;
    y=0;
    while (x!=0)
    {
        a=x % 10;
        b=b*10 + a;
        x=x/10;
    }
    return y;
}
int main()
{
    int x, y, t;
    printf("\n Donner une valeur entiere positive = ");
    scanf("%f", &x);
    y = miroir (x);
    printf("\n Image miroir de %d = %d", x, y);
    return 0;
}
```

**Exercice 4** Ecrire une procédure qui affiche les n premiers nombres premiers: un nombre premier est toujours de la forme  $6 \times k \pm 1$ , où k est un entier  $> 1$ .

```
/* premier est une procedure*/
#include <stdio.h>
void premier(int n)
{
    int k, i, x, y;
    k=1;
    printf("\n les %d premiers nombres premiers sont ", n);
    for (i=1; i<=n/2; i++)
    {
        x=6*k-1;
        y=6*k+ 1;
        k=k+1;
        printf("\n %d   %d ", x, y);
    }
    if (n %2 !=0)
        printf("\n %d ", 6*k-1);
}
int main()
{
    int x, y, t;
    printf("\n Donner le nbr de valeurs premiers = ");
    scanf("%f", &n);
    premier (n);
    return 0;
}
```