

Dans cette partie, on va donner les solutions d'exercices 1, 2, 3, et 04

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

```
Algorithme partie_entiere ;  
  Var a, b: réels;  
  Fonction Entiere(x: réel) :réel ;  
    Var y:réel;  
    Debut  
    y←0;  
    Tant que x ≥ 1 Faire  
      x←x-1;  
      y←y+1;  
    FinTQ  
    Entiere ← y;  
    Fin  
  Debut  
  Ecrire("Donner une valeur réelle positive ");  
  lire (a);  
  b← Entiere (a);  
  Ecrire ("La partie entiere de ", a, " est ",b);  
  Fin
```

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

Algorithme PGCD_Cal

Var a, b, c: entiers

Fonction PGCD(a: entier, b:entier) : entier ;

Debut

Tant que a*b≠ 0 **Faire**

Si a>b alors

a←a-b;

sinon

b←b-a;

FinTQ

Si a = 0 alors

PGCD ←b;

Sinon

PGCD ←a;

FinSi

Fin

Debut

Ecrire(" donner deux valeurs entieres positives");

Lire (a, b);

c← PGCD(a, b);

Ecrire (" le pgcd de ", a, " et ", b, " est ", c);

Fin

Algorithme PGCD_Cal

Var a, b, c: entiers

Proceure PGCD(a: entier, b:entier) : entier ;

Var x, y, c: entiers;

Debut

x←a; y←b;

Tant que a*b≠ 0 **Faire**

Si a>b alors

a←a-b;

sinon

b←b-a;

FinTQ

Si a = 0 alors

c ←b;

Sinon

c←a;

FinSi

Ecrire (" le pgcd de ", x, " et ", y, " est ", c);

Fin

Debut

Ecrire(" donner deux valeurs entieres positives");

Lire (a, b);

PGCD(a, b);

Fin

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

```
Algorithme image_invesre;
Var a, b: réels;
Procédure Miroir(x: entier) :entier;
Var y:réel;
  Debut
  y←0;
  Tant que x≠0 Faire
    a←x mod 10;
    Ecrire (a) ;
    x←x div 10;
  FinTQ
Fin

Debut
  Ecrire("Donner une valeur entiere positive ");
  lire (a);
  b← Miroir (a);
Fin
```

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 .

```

Algorithme nombre_premier;
  Var n: entier;
Procédure Premier(n: entier)
  Var y, i, k: entiers;
  Debut
    k ← 1;
    Pour i ← 1 jusqu'à n/2 Faire
      x ← 6*k-1;
      y ← 6*k+1;
      k ← k+1;
      écrire (x, y);
    FinPour
    si n mod 2 <> 0 alors
      écrire (6*k-1)
    finsi
  Fin
Debut
  Ecrire("Donner le nbr des entiers premiers ");
  lire (n);
  Premier (n);
Fin

```