

TP 2 : Les fonctions

Objectifs du TP

- Maitriser la déclaration des fonctions Python
- Comprendre les fonctions récursives

Exercice 1 :

1. Ecrire une fonction « **somme_div(n)** » qui retourne la somme de tous les diviseurs d'un nombre passé en paramètre.

```
def somme_div( n ) :  
    s = 0  
    for i in range(1, n+1):  
        if n%i==0:  
            s += i  
    return s
```

Un nombre entier est parfait s'il est égal à la somme de ses diviseurs (sauf lui-même).

Ex : 6 = 1 + 2 + 3 est parfait.

2. Ecrire une fonction « **parfait (m)** » qui teste si un nombre passé en paramètre **m** est parfait et qui retourne **True** s'il l'est et **False** sinon.

```
def parfait( m ):  
    return somme_div(m) - m == m
```

3. Ecrire une fonction « **premier (x)** » qui teste si un nombre passé en paramètre **x** est premier et qui retourne **True** s'il l'est et **False** sinon.

```
def premier(x):  
    return somme_div(x) == 1+x
```

Exercice 2 :

Implémentez une fonction **fibonacci(n)** qui renvoie le n^{ème} terme de la suite de Fibonacci en utilisant à la fois une approche **itérative** et une approche **récursive**.

$$\begin{cases} U_0 = 0 \\ U_1 = 1 \\ U_n = U_{n-2} + U_{n-1} \end{cases}$$

```
# version itérative  
def fibo_it(n):  
    if n==0 or n==1 :  
        return n
```

```

u0,u1 = 0,1
for i in range(2, n+1):
    u = u0 + u1
    u0 = u1
    u1 = u
return u

# version réursive
def fibo(n):
    if n==0 or n==1:
        return n
    return fibo(n-1) + fibo(n-2)

```

Exercice 3 :

Implémentez l'algorithme d'Euclide dans une fonction **pgcd(a,b)** pour trouver le plus grand commun diviseur (PGCD) de deux nombres entiers a et b.

```

def pgcd(a,b) :
    if b==0:
        return abs(a)
    return pgcd(b, a%b)

```