

TD 5 (feuille d'exo 4) - Boucles, listes

Informatique
PCSI - Lycée Thiers

Exercice 1.

Enoncé

Correction

Exercice 2.

Enoncé

Correction

Exercice 3.

Enoncé

Correction

Exercice 4. Sujet 0 d'oral (E3A)

Enoncé

Corrigé

Exercice 1.

Exercice 1.

1. Ecrire une fonction $\text{pgcd}(a,b)$ qui calcule le PGCD de deux entiers positifs a et b à l'aide de l'**algorithme d'Euclide** :

```
Donnés deux entiers positifs  $a$  et  $b$  :  
TANT QUE  $b \neq 0$  :  
    Changer  $a$  et  $b$  respectivement par  $b$  et  $a$  modulo  $b$   
RETOURNER  $a$ 
```

2. Ecrire la fonction $\text{irreductible}(a,b)$ qui prend en paramètre 2 entiers $(a,b) \in \mathbb{Z} \times \mathbb{Z}^*$ et qui renvoie le couple $c,d \in \mathbb{Z} \times \mathbb{Z}^*$ où $\frac{c}{d}$ est une fraction irréductible égale à $\frac{a}{b}$.
3. Ecrire la fonction $\text{ppcm}(a,b)$ qui retourne le plus petit commun multiple de deux entiers a et b .

Exercice 1. Corrigé

1. (Remarque : marche aussi pour des entiers relatifs.)

```
def pgcd(a,b):  
    while b != 0:  
        a, b = b, a% b  
    return a
```

2.

```
def irreductible(a,b):  
    d = pgcd(a,b)  
    return a//d, b//d
```

3. On utilise : $\text{pgcd}(a, b) \times \text{ppcm}(a, b) = a \times b$.

```
def ppcm(a,b):  
    return a*b//pgcd(a,b)
```

Exercice 2.

Exercice 2.

1. Ecrire une fonction `premier(n)` prenant en paramètre un entier positif `n` et qui renvoie le booléen `True` ou `False` selon si `n` est un nombre premier ou pas. (Indication : Pour déterminer si un nombre n est premier il suffit de vérifier qu'il n'est divisible par aucun entier compris entre 2 et $\sqrt{n}$.)
2. En déduire une fonction `Premiers(N)` qui retourne la liste des nombres premiers dans $[0, N]$.

Exercice 2. Corrigé

1. Un entier naturel n est premier ssi $n \geq 2$ et n n'a aucun diviseur entre 2 et $n-1$ ssi $n \geq 2$ et n n'a aucun diviseur entre 2 et $\sqrt{n}$.

```
def premier(n):  
    if n == 0 or n == 1:  
        return False  
    for d in range(2, int(n**0.5)+1):  
        if n % d == 0:  
            return False  
    return True
```

Exercice 2. Corrigé

2.

```
def Premiers(N):  
    L = []  
    for n in range(N+1):  
        if premier(n):  
            L.append(n)  
    return L
```

ou à l'aide d'une compréhension de liste :

```
def Premiers(N):  
    return [ n for n in range(N+1) if premier(n) ]
```

Exercice 3.

Exercice 3. Écrire une fonction `fibonacci(n)` qui prend en paramètre un entier n et qui retourne la liste des termes $u_0, \dots, u_n$ de la suite de Fibonacci :

$$u_0 = 0, \quad u_1 = 1, \quad \forall n \in \mathbb{N}, u_{n+2} = u_n + u_{n+1}$$

Exercice 3. Corrigé

```
def Fibonacci(N):  
    L = [0, 1]  
    for k in range(N-1):  
        L.append(L[-1]+L[-2])  
    return L
```

Exercice 4. Sujet 0 d'oral (E3A)

Exercice 4.

1. On considère le code Python de la fonction `d` suivante :

```
def d(n):  
    L=[1]  
    for nombre in range(2,n+1):  
        if n%nombre==0:  
            L.append(nombre)  
    return L
```

Quel est le résultat de l'appel `d(4)` ? Puis de l'appel `d(10)` ?

Que fait la fonction `d` ?

2. Un *diviseur non-trivial* d'un entier `n` est un diviseur de `n` différent de 1 et de `n`. Écrire une fonction `DNT`, d'argument `n`, renvoyant la liste des diviseurs non-triviaux de l'entier `n`.
3. Écrire une fonction `sommeCarresDNT`, d'argument `n`, renvoyant la somme des carrés des diviseurs non-triviaux de l'entier `n`.
4. Écrire la suite des instructions permettant d'afficher tous les nombres entiers inférieurs à 1000 et égaux à la somme des carrés de leurs diviseurs non-triviaux. Que peut-on conjecturer ?

Exercice 4 : Corrigé

1) L'appel de `d(4)` retourne la liste `[1,2,4]`. L'appel de `d(10)` retourne `[1,2,5,10]`. Plus généralement, la fonction `d` retourne pour un paramètre `n` entier positif la liste des diviseurs de `n`.

2) Il suffit de modifier le code de la fonction `d` :

```
def DNT(n):  
    L = []  
    for nombre in range(2,n):  
        if n % nombre == 0:  
            L.append(nombre)  
    return L
```

3)

```
def sommeCarresDNT(n):  
    L = DNT(n)  
    S = 0  
    for x in L:  
        S += x**2  
    return S
```

Exercice 4 : Corrigé

4)

```
L = []  
for n in range(1001):  
    if n == sommeCarresDNT(n):  
        L.append(n)  
print(L)
```

Ce qui produit :

[0, 4, 9, 25, 49, 121, 169, 289, 361, 529, 841, 961]

On conjecture :

"Les nombres égaux à la somme des carrés de leurs diviseurs non-triviaux sont les carrés des nombres premiers"