

# Test 1 - M1 BioInformatique

22 octobre 2014 - Durée : 30 minutes

## Exercice 1

Écrivez un programme `somme_entiers_paires(n)` qui prends en paramètre un entier  $n$  et qui renvoie la somme des  $n$  premiers entiers paires.

Par exemple, `somme_entiers_paires(7)` calculera la somme  $0 + 2 + 4 + 6$ .

## Solution

```
1 def somme_entiers_paires(n):
2     res = 0
3     for i in range(n//2 + 1):
4         res = res + 2*i
5     return res
```

## Exercice 2

Écrivez un programme `suite(n)` qui prends en paramètre un entier  $n$  et qui renvoie la valeur de l'élément  $U_n$  défini par :

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

Par exemple, `suite(4)` renverra la valeur de  $U_4$  qui vaut 11.

(En effet,  $U_2 = U_1 + 2 * U_0 = 3$  et  $U_3 = U_2 + 2 * U_1 = 5$  et enfin  $U_4 = U_3 + 2 * U_2 = 11$ )

## Solution

```
1 def suite(n):
2     if n == 0:
3         return 1
4     if n == 1:
5         return 1
6     Un1 = 1
7     Un2 = 1
8     for i in range(2, n+1):
9         Un = Un1 + 2*Un2
10        Un2 = Un1
11        Un1 = Un
12    return Un
```

### Exercice 3

Voici un programme mystérieux :

```
1 def mystere(n , d):
2     r = n
3     q = 0
4     while r >= d :
5         q = q + 1
6         r = r - d
7     return [q, r]
```

Exécutez le programme pour les valeurs suivantes : ( $n = 10, d = 3$ ) et ( $n = 3, d = 5$ ). Vous présenterez l'exécution du programme à l'aide d'un tableau qui contient le contenu de l'espace mémoire de  $q$  et de  $r$  à la fin de chaque boucle while.

Que fait ce programme ?

#### Solution

Pour ( $n = 10, d = 3$ ),

boucle mémoire	q	r
0 (init)	0	10
1	1	7
2	2	4
3	3	1

Le programme renvoie [3, 1].

Pour ( $n = 3, d = 5$ ),

boucle mémoire	q	r
0 (init)	0	3

Le programme renvoie [0, 3].

Ce programme calcul le quotient et le reste de la division euclidienne de  $n$  par  $d$ .

### Exercice 4

Écrivez un programme `est_trie` qui prends en paramètre un tableau d'entiers et qui renvoie vraie si le tableau est trié et faux sinon.

Par exemple, `est_trie([1,4,4,7,9])` renvoie `True`, alors que `est_trie([1,4,9,4,7])` renvoie `False`.

#### Solution

```
1 def est_trie(T):
2     for i in range( len(T)-1 ):
3         if T[i] > T[i+1]:
4             return False
5     return True
```