

ISN_DM1_2020_Corrige

September 14, 2019

1 ISN Lycée du Parc 2019/2020 Corrigé du DM n°1

1.1 Exercice 1

Question 1 Écrire un programme Python qui affiche les entiers pairs compris entre 0 et 100 inclus dans l'ordre croissant.

```
In [ ]: for n in range(0, 101, 2):  
        print(n)
```

Question 2 Écrire un programme Python qui affiche les entiers pairs compris entre 0 et 100 inclus dans l'ordre décroissant.

```
In [ ]: for n in range(100, -1, -2):  
        print(n)
```

Question 3 Écrire un programme Python qui calcule dans une variable *s* la somme des puissances 4 des entiers compris entre 1 et 100 inclus.

```
In [5]: s = 0  
        for n in range(1, 101):  
            s = s + n ** 4
```

```
In [6]: s
```

```
Out[6]: 2050333330
```

Question 4 Si on liste tous les entiers naturels inférieurs à 20 qui sont multiples de 5 ou de 7, on obtient 5, 7, 10, 14 et 15. La somme de ces nombres est 51. Écrire un programme Python qui détermine la somme de tous les multiples de 5 ou de 7 inférieurs à 42455.

```
In [13]: s = 0  
         for n in range(1, 42456):  
             if n % 5 == 0 or n % 7 == 0:  
                 s = s + n  
         print(s)
```

```
283259760
```

1.2 Exercice 2

Question 1 Parmi les fonctions ci-dessous déterminer celles qui retournent le terme de rang n de la suite définie par

$$\begin{cases} u_0 = 734 \\ \forall n \in \mathbb{N}, \quad u_{n+1} = \frac{2}{3}u_n + 100 \end{cases}$$

In [15]: *#NON*

```
def suiteU1(n):  
    u = 734  
    for k in range(n):  
        u = 2 / 3 * u + 100  
    return u
```

In [16]: *#Oui*

```
def suiteU2(n):  
    u = 734  
    for k in range(n):  
        u = 2 / 3 * u + 100  
    return u
```

In [17]: *#OUI*

```
def suiteU3(n):  
    u = 734  
    for k in range(1, n + 1):  
        u = 2 / 3 * u  
        u = u + 100  
    return u
```

In [18]: *#NON*

```
def suiteU4(n):  
    u = 734  
    for k in range(1, n + 1):  
        u = u + 100  
        u = 2 / 3 * u  
    return u
```

Question 2 Écrire une fonction `suiteV(n)` pour qu'elle retourne le terme de rang n de la suite définie par

$$\begin{cases} v_0 = 10 \\ \forall n \in \mathbb{N}, \quad v_{n+1} = \frac{2}{3}v_n + n + 1 \end{cases}$$

In [25]: *#Version 1*

```
def suiteV(n):  
    v = 10  
    for k in range(n):  
        v = 2 / 3 * v + k + 1  
    return v
```

```

#Version 2
def suiteV1(n):
    v = 10
    for k in range(1, n+1):
        v = 2 / 3 * v + k
    return v

```

1.3 Exercice 3

Question 1 Fonction mystere

```

In [27]: def mystere(n):
        s = 0
        print("n=",n,"s=",s)
        while n > 0:
            s = s + n % 10
            n = n // 10
            print("n=",n,"s=",s)
        return s

```

```

In [28]: mystere(734)

```

```

n= 734 s= 0
n= 73 s= 4
n= 7 s= 7
n= 0 s= 14

```

```

Out[28]: 14

```

La fonction `mystere(n)` retourne la somme des chiffres de l'entier naturel `n`

Question 2 Fonction miroir

```

In [31]: def miroir(n):
        m = 0
        while n > 0:
            m = m * 10 + n % 10
            n = n // 10
        return m

```

```

In [32]: miroir(734)

```

```

Out[32]: 437

```

Écrire un programme qui détermine le grand grand entier naturel `n` inférieur à 10 millions ayant la propriété : $\text{miroir}(n) = 4 \times n$

```

In [35]: miroirmax = 0
        for n in range(1, 10**7):
            if miroir(n) == 4 * n:
                miroirmax = n
        print(miroirmax)

```

2199978

```
In [36]: miroir(2199978) == 4 * 2199978
```

```
Out[36]: True
```

Question 3 Suite de Fibonacci

```
In [37]: def fibo(n):  
    u = 1  
    v = 1  
    for k in range(2, n + 1):  
        w = u  
        u = v  
        v = w + v  
    return u
```

```
In [41]: #10 premiers termes de la suite de Fibonacci  
    [fibo(n) for n in range(1, 11)]
```

```
Out[41]: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

```
In [52]: def sommeImpairFibo(m):  
    """Somme des termes impairs de la suite de Fibonacci dont  
    la valeur ne dépasse pas m"""  
    u = 1  
    v = 1  
    s = 1  
    while v < m:  
        w = u  
        u = v  
        v = w + v  
        if u % 2 == 1:  
            s = s + u  
    return s
```

```
In [53]: sommeImpairFibo(4000000)
```

```
Out[53]: 4613732
```