

Exercice 1*Correction de programme*

Le script ci-dessous doit répondre au cahier des charges suivant :

- récupérer dans une variable `n` un entier positif saisi par l'utilisateur ;
- calculer avec une fonction `sommecarre` la somme des carrés des entiers compris entre 1 et `n` et stocker cette valeur dans une variable `res` ;

Malheureusement ce script comporte plusieurs erreurs : ce sont des erreurs de syntaxe ou des erreurs qui se produiront lors de l'exécution.

Ecrire une version rectifiée de ce script.

```

1 def sommecarre(p):
2     s = 0
3     for k in range(1, p+1):
4         s += k**2
5     print(s)
6
7 n = input('Entrez un entier n > 0 : ')
8 if n < 0
9 print('On doit avoir n > 0 !!! ')
10 else:
11     res = sommecarre(n)

```

Script avec erreurs

Exercice 2*Corriger un programme*

1. La fonction Python ci-dessous doit répondre au cahier des charges suivant :

- prendre en paramètre un entier positif `n` de type `int` ;
- retourner une valeur, `None` si `n < 0` ou le terme de rang `n` de la suite (u_n) définie par :

$$\begin{cases} u_0 = 1 \\ u_{n+1} = u_n + n^2 - n + 1 \end{cases} \text{ pour tout entier } n \geq 0$$

Malheureusement le code proposé pour cette fonction comporte plusieurs erreurs : ce sont des erreurs de syntaxe ou des erreurs qui se produiront lors de l'exécution.

Ecrire une version rectifiée de cette fonction.

```

1 def suite(n):
2     if n < 0:
3         return None
4     for k in range(1, n+1):
5         u == u + k**2 - k + 1
6     return u

```

Une suite récurrente

2. La fonction Python ci-dessous doit répondre au cahier des charges suivant :

- prendre en paramètre un tableau d'entiers de type `list` ;
- retourner la valeur du maximum de ce tableau d'entiers ;

Malheureusement le code proposé pour cette fonction comporte plusieurs erreurs : ce sont des erreurs de syntaxe ou des erreurs qui se produiront lors de l'exécution.

Ecrire une version rectifiée de cette fonction.

```

1 def maximumtab(t):
2     for k in range(1, len(t)):
3         if t[k] > m:
4             m = t[k]
5     return m

```

Recherche du maximum dans un tableau d'entiers

Exercice 3

1. On veut écrire en Python une fonction `recherche_sequentielle` qui doit répondre au cahier des charges suivant :

- prendre en paramètre un tableau d'entiers `t` de type `list` et un entier `n` de type `int` ;
- retourne `True` si la valeur de `n` appartient à `t` et `False` sinon.

Compléter le corps de la fonction `recherche_sequentielle`.

```

1 def recherche_sequentielle(t, n):
2     k = 0
3     while k < len(t) and .....:
4         k += 1
5     return .....

```

Recherche séquentielle

2. On veut écrire un script qui doit répondre au cahier des charges suivant :

- récupérer dans une variable `m` un nombre réel positif saisi par l'utilisateur ;
- calculer avec une fonction `seuilfibo` le plus petit entier `n` tel que $f_n \geq m$ où f_n est le terme de rang `n` de la suite de Fibonacci définie par :

$$\begin{cases} f_0 = 0 & \text{et} & f_1 = 1 \\ f_n = f_{n-1} + f_{n-2} & \text{pour tout entier } n \geq 2 \end{cases}$$

Compléter le corps de la fonction `seuilfibo`.

```

1 def seuilfibo(s):
2     a, b = 0, 1
3     k = 0 #indice du terme dans la variable a
4     while a .....:
5         k = .....

```

```
6         a, b = .....
7     return k
8
9     m = int(input('Entrez un entier m >= 0 : '))
10    if m < 0:
11        print('On doit avoir m >= 0 !!! ')
12    else:
13        print("L'indice du premier terme de la suite de Fibonacci \
supérieur ou égal à ", m, "est : ", seuilfibo(m))
```

Algorithme de seuil et suite de Fibonacci

Exercice 4

Ecrire un programme

1. Ecrire une fonction de signature `inverser(chaine)` qui prend en paramètre une chaîne de caractère et qui retourne la chaîne obtenue en recopiant la chaîne initiale de droite à gauche.
2. Ecrire une fonction `somme(n)` qui prend en entrée un entier n et qui retourne valeur de la somme des puissances 5 des entiers consécutifs entre 1 et n

$$1 + 2^5 + \dots + n^5$$

Corrigé 1

```
1 def sommecarre(p):
2     s = 0
3     for k in range(1, p+1):
4         s += k**2
5     return s
6
7 n = int(input('Entrez un entier n > 0 : '))
8 if n < 0:
9     print('On doit avoir n > 0 !!! ')
10 else:
11     res = sommecarre(n)
12     print('Somme des carrés de 1 à %d = '%n, res)
13
14 # In [8]: (executing lines 7 to 18 of "ISN-DS2-Corrige.py")
15 # Entrez un entier n > 0 : 10
16 # Somme des carrés de 1 à 10 = 385
17 # In [9]: 10*(10+1)*(2*10+1)/6 #vérification avec la formule n*(n+1)*(2*n+1)
18 # Out[9]: 385.0
```

Corrigé 2

```
1 #Exo 1 code à rectifier
2
3 #Q1
4 def suite(n):
5     """Retourne le terme de rang n de la suite
6     u(0)=1 et u(n+1)= u(n) + n**2-n+1"""
7     u = 1
8     if n < 0:
9         return None
10    for k in range(1, n+1):
11        u = u + (k-1)**2 - (k-1) + 1
12    return u
13
14 # In [5]: [suite(i) for i in range(-1, 5)]
15 # Out[5]: [None, 1, 2, 3, 6, 13]
16
17 #Q2
18 def maximumtab(t):
19     """Retourne le maximum d'un tableau d'entiers"""
20     m = t[0]
21     for k in range(1, len(t)):
22         if t[k] > m:
```

```

23         m = t[k]
24     return m
25
26 # In [7]: from random import randint
27 #
28 # In [8]: t1 = [randint(1, 10) for i in range(10)]
29 #
30 # In [9]: t2 = [randint(1, 10) for i in range(10)]
31 #
32 # In [10]: t1, t2
33 # Out[10]: ([7, 4, 3, 8, 9, 6, 4, 4, 5, 9], [6, 9, 6, 1, 4, 6, 10, 6, 2, 9])
34 #
35 # In [11]: maximumtab(t1), maximumtab(t2)
36 # Out[11]: (9, 10)

```

Corrigé 3

```

1 def recherche_sequentielle(t, n):
2     k = 0
3     while k < len(t) and t[j] != n:
4         k += 1
5     return k == len(t)
6
7 #Q3
8
9 def fibo(n):
10     """Retourne le terme de rang n de la suite de Fibonacci"""
11     a, b = 0, 1
12     if n < 0:
13         return None
14     if n == 0:
15         return a
16     for k in range(2, n+1):
17         a, b = b, a + b
18     return b
19
20 def seuilfibo(s):
21     a, b = 0, 1
22     k = 0 #indice du terme dans la variable a
23     while a < s:
24         k += 1
25         a, b = b, a + b
26     return k
27
28 m = int(input('Entrez un entier m >= 0 : '))
29 if m < 0:
30     print('On doit avoir m >= 0 !!! ')
31 else:

```

```
32     print("L'indice du premier terme de la suite de Fibonacci \ supérieur ou
        égal à ", m, "est : ", seuilfibonacci(m))
33
34     """
35     In [26]: (executing lines 84 to 96 of "ISN-DS3-corrige.py")
36     Entrez un entier m >= 0 : 6
37     L'indice du premier terme de la suite de Fibonacci \ supérieur ou égal à 6
        est : 6
38
39     In [27]: (executing lines 84 to 107 of "ISN-DS3-corrige.py")
40     Entrez un entier m >= 0 : 6
41     L'indice du premier terme de la suite de Fibonacci \ supérieur ou égal à 6
        est : 6
42
43     In [28]: fibo(6)
44     Out[28]: 8
45
46     In [29]: [fibo(i) for i in range(-1, 7)]
47     Out[29]: [None, 0, 1, 1, 2, 3, 5, 8]
48     """
```

Corrigé 4

```
1 #Q1
2 def inverser(chaine):
3     """Retourne l'inverse d'une chaine de caractères"""
4     newchaine = ''
5     for c in chaine:
6         newchaine = c + newchaine
7     return newchaine
8
9 def inverser2(chaine):
10    """Retourne l'inverse d'une chaine de caractères"""
11    return chaine[::-1]
12
13 #Q2
14 def somme(n):
15    """Retourne la somme des puissances 5 des n premiers
16    entiers"""
17    s = 0
18    for k in range(1, n+1):
19        s += k**5
20    return s
21
22 # In [33]: somme(3)
23 # Out[33]: 276
24 #
25 # In [34]: sum(k**5 for k in range(1, 4))
26 # Out[34]: 276
```
