

Spécialité NSI Tle
Contrôle d'informatique n° 1
Sujet A
Lundi 19 septembre 2022

Le barème est indicatif. Tous les programmes sont à écrire en Python.

A. Structures élémentaires (4 points)

1. Ecrire un programme qui demande à l'utilisateur de choisir deux entiers a et b et qui affiche « diviseur » si **b est un diviseur de a** et « pas diviseur » sinon.
2. Écrire un programme qui affiche tous les entiers multiples de 7 compris entre 14 à 45 compris (pas plus de quatre lignes de code).
3. On exécute le programme suivant :

```
n = 1
u = 2
while u < 50 :
    u = 2*u
    n = n + 1
print(n)
```

Que s'affiche-t-il à l'écran ?

4. On considère le code suivant :

```
nb = 21
while nb < 150 :
    print(nb)
    nb = nb + 8
```

Ecrire un programme qui produit exactement le même affichage à l'écran sans boucle `while` mais avec un boucle `for` (pas plus de deux lignes de code).

B. Fonctions (4 points)

5. On exécute le code suivant :

```
def maFct(a, b, c) :
    a = 5*a
    b = b + 3
    return (c - b)*a
m = 4
n = 3
p = 2
y = maFct(n, p, m)
print(y)
```

a. Que s'affiche-t-il à l'écran ?

b. Quelles sont les valeurs finales des variables m , n et p ?

6. Recopier et compléter le code de la fonction `sommeCubes` (n) qui renvoie la somme des cubes des entiers naturels inférieurs ou égaux à n ,

$$1^3 + 2^3 + \dots + n^3$$

où n est un entier positif.

```
def sommeCube(n) :
    """sommeCube : renvoie la somme des cubes des
entiers inférieurs ou égaux à n"""
    s = ...
    for i in range(...):
```

```
...
return ...
```

C. Tableaux et fonctions (8 points)

7. Écrire la fonction `maxi(T)` qui renvoie le plus grand élément d'un tableau de nombres supposé non vide.

8. a. Recopier et compléter le corps de la fonction `diviseursTab(n)` qui renvoie un tableau de tous les diviseurs de n compris entre 1 et $n - 1$

```
def diviseursTab(n) :
    """diviseursTab : renvoie le tableau des
    diviseurs de n compris entre 1 et n-1"""
    T = []
    for ... in range(1, n):
        if ... :
            T.append(k)
    return ...
```

b. Ecrire un *appel de la fonction* `diviseursTab` qui permette de récupérer dans un tableau nommé `tab` tous les diviseurs de 150 compris entre 1 et 149.

9. Écrire la fonction `sommeTab(T)` qui renvoie la somme des éléments du tableau de nombres `T` (et 0 si le tableau `T` est vide).

10. On suppose que les fonctions `diviseursTab` et `sommeTab` ont été correctement écrites. Utiliser ces deux fonctions pour écrire la fonction `parfait(n)` qui renvoie `True` si l'entier n est un nombre parfait et `False` sinon. On rappelle qu'un nombre parfait est un entier positif qui est la somme de tous ses diviseurs sauf lui-même.

D. Construction de tableau par compréhension et matrices (4 points)

11. On exécute le code suivant :

```
T1 = [5*n + 2 for n in range(5)]
print(T1)
```

Que s'affiche-t-il à l'écran ?

12. On exécute le code suivant :

```
T2 = [8*k - 2 for k in range(10) if k%3 == 0]
print(T2)
```

Que s'affiche-t-il à l'écran ?

13. On exécute le code suivant :

```
matrice = [[1, 2, 3, 4], [1, 4, 9, 16], [1, 8, 27,
64]]
x = mat[1][3]
```

Que contient alors la variable `x` ?

14. Ecrire une fonction `sommeLigne(mat, n)` qui prend en argument une matrice carrée (c'est-à-dire qui contient autant de lignes que de colonnes) non vide et qui renvoie la somme des éléments de la $n^{\text{ème}}$ ligne de la matrice (on rappelle que la *première* ligne correspond à l'indice 0).

Bonus : Ecrire une fonction `contientColonneNulle(mat)` qui prend en argument une matrice et qui renvoie `True` si au moins une des colonnes de la matrice ne contient que des zéros et `False` sinon.