

**Spécialité NSI Terminale**  
**Correction du contrôle n° 1**  
Lundi 19 septembre 2022

**A Structures élémentaires**

1. On peut écrire le programme suivant :

```
a = int(input("Choisissez le nombre entier a : "))
b = int(input("Choisissez le nombre entier b : "))
if a%b == 0 :
    print("diviseur")
else :
    print("pas diviseur")
```

2. On peut écrire le programme suivant :

```
for k in range(14, 46, 7) :
    print(k)
```

3. Il s'affiche :

6

car  $2^6 = 64 > 50$  mais  $2^5 = 32 < 50$

4. On peut écrire le programme suivant :

```
for nb in range(21, 150, 8) :
    print(nb)
```

**B Fonctions**

5. a. Il s'affiche :

-15

En effet, lors de l'appel de la fonction `maFct`, les arguments `a`, `b` et `c` prennent respectivement les valeurs 3, 2 et 4.

b. Les valeurs finales des variables sont `m : 4 ; n : 3 ; p : 2`. Elles n'ont pas été modifiées par l'appel de la fonction `maFct`.

6. On peut compléter le code de la façon suivante :

```
def sommeCube(n) :
    s = 0
    for i in range(1, n+1):
        s = s + i**3
    return s
```

**C Tableaux et fonctions**

7. On peut écrire au choix l'un des deux programmes suivants :

```
def maxi(T) :
    maxi = T[0]
    for el in T :
        if el > maxi :
            maxi = el
    return maxi
```

```
def maxi(T) :
    maxi = T[0]
    for i in range(1, len(T)):
        if T[i] > maxi :
            maxi = T[i]
    return maxi
```

8. a. On doit compléter le code de la façon suivante :

```
def diviseursTab(n) :
    T = []
    for k in range(1, n):
        if n%k == 0 :
            T.append(k)
    return T
```

b. On doit écrire l'instruction suivante :

```
tab = diviseursTab(150)
```

9. On peut écrire la fonction suivante :

```
def sommeTab(T) :
    s = 0
    for e in T:
        s = s + e
    return s
```

10. On peut écrire la fonction suivante :

```
def parfait(n) :
    tabDiv = diviseursTab(n)
    sommeDiv = sommeTab(tabDiv)
    return n == sommeDiv
```

## **D Construction de tableau par compréhension et matrices**

11. Il s'affiche :

```
[2, 7, 12, 17, 22]
```

12. Il s'affiche :

```
[-2, 22, 46, 70]
```

13. La variable  $x$  contient 16 (le 4<sup>ème</sup> élément du 2<sup>ème</sup> sous-tableau).

14. On peut écrire la fonction suivante :

```
def sommeLigne(mat, n) :
    s = 0
    lg = len(mat)
    for j in range(lg):
        s = s + mat[n-1][j]
    return s
```

**Bonus :** On peut écrire la fonction suivante :

```
def contientColonneNulle(mat) :
    lg = len(mat)
    for j in range(lg):
        colNulle = True
        for i in range(lg) :
            if mat[i][j] != 0 :
                colNulle = False
                break
        if colNulle :
            return True
    return False
```