

# **BACCALAURÉAT**

**SESSION 2023**

---

**Épreuve de l'enseignement de spécialité**

## **NUMÉRIQUE et SCIENCES INFORMATIQUES**

**Partie pratique**

**Classe Terminale de la voie générale**

---

**Sujet n°03**

---

**DURÉE DE L'ÉPREUVE : 1 heure**

**Le sujet comporte 4 pages numérotées de 1 / 4 à 4 / 4  
Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

*Le candidat doit traiter les 2 exercices.*

## EXERCICE 1 (4 points)

Dans cet exercice, les nombres sont des entiers ou des flottants.

Écrire une fonction `moyenne` renvoyant la moyenne pondérée d'une liste non vide, passée en paramètre, de tuples à deux éléments de la forme `(valeur, coefficient)` où `valeur` et `coefficient` sont des nombres positifs ou nuls. Si la somme des coefficients est nulle, la fonction renvoie `None`, si la somme des coefficients est non nulle, la fonction renvoie, sous forme de flottant, la moyenne des valeurs affectées de leur coefficient.

Exemples :

```
>>> moyenne([(8, 2), (12, 0), (13.5, 1), (5, 0.5)])
9.142857142857142
>>> moyenne([(3, 0), (5, 0)])
None
```

Dans le premier exemple la moyenne est calculée par la formule :

$$\frac{8 \times 2 + 12 \times 0 + 13,5 \times 1 + 5 \times 0,5}{2 + 0 + 1 + 0,5}$$

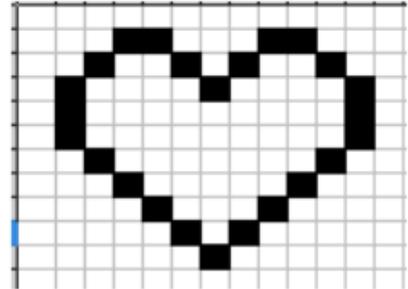
## EXERCICE 2 (4 points)

On travaille sur des dessins en noir et blanc obtenus à partir de pixels noirs et blancs :

La figure « cœur » ci-contre va servir d'exemple.

On la représente par une grille de nombres, c'est-à-dire par une liste composée de sous-listes de mêmes longueurs.

Chaque sous-liste représentera donc une ligne du dessin.



Dans le code ci-dessous, la fonction `affiche` permet d'afficher le dessin. Les pixels noirs (1 dans la grille) seront représentés par le caractère " \*" et les blancs (0 dans la grille) par deux espaces.

La fonction `zoomListe` prend en argument une liste `liste_depart` et un entier `k`. Elle renvoie une liste où chaque élément de `liste_depart` est dupliqué `k` fois.

La fonction `zoomDessin` prend en argument la grille `dessin` et renvoie une grille où toutes les lignes de `dessin` sont zoomées `k` fois (c'est-à-dire, on applique à chaque ligne la fonction `zoomListe` avec comme second paramètre `k`) et répétées `k` fois.

Compléter le code ci-dessous :

```
coeur = [[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
         [0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0],
         [0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0],
         [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0],
         [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
         [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
         [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
         [0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0],
         [0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
```

```
def affiche(dessin):
    ''' affichage d'une grille : les 1 sont représentés par
        des " *" , les 0 par deux espaces "  ".
        La valeur "" donnée au paramètre end permet de ne pas avoir
        de saut de ligne.'''
    for ligne in dessin:
        for col in ligne:
            if col == 1:
                print(" *", end= "")
            else:
                print("  ", end= "")
        print()
```

```

def zoomListe(liste_depart, k):
    '''renvoie une liste contenant k fois chaque
        élément de liste_depart'''
    liste_zoom = ...
    for elt in ... :
        for i in range(k):
            ...
    return liste_zoom

def zoomDessin(grille, k):
    '''renvoie une grille où les lignes sont zoomées k fois
        ET répétées k fois'''
    grille_zoom = []
    for elt in grille:
        liste_zoom = ...
        for i in range(k):
            ... .append(...)
    return grille_zoom

```

Résultats à obtenir :

```
>>> affiche(coeur)
```



```
>>> affiche(zoomDessin(coeur, 3))
```

