

# Enonce



On travaille sur des dessins en noir et blanc obtenu à partir de pixels noirs et blancs : La figure « cœur » ci-dessus va servir d'exemple. On la représente par une grille de nombres, c'est-à-dire par une liste composée de sous-listes de même 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 et répétées `k` fois.

Soit le code ci-dessous :

Script Python

```

1  coeur = [[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], \
2      [0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0], \
3      [0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0], \
4      [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0], \
5      [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0], \
6      [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0], \
7      [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0], \
8      [0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0], \
9      [0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0], \
10     [0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0], \
11     [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0], \
12     [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]] \
13 
14 def affiche(dessin):
15     ''' affichage d'une grille : les 1 sont représentés par
16         des " * ", les 0 par deux espaces "   " '''
17     for ligne in dessin:
18         for col in ligne:
19             if col == 1:
20                 print(" *", end="")
21             else:
22                 print("   ", end="")
23         print()
24 
25 
26 def zoomListe(liste_depart,k):
27     '''renvoie une liste contenant k fois chaque
28     élément de liste_depart'''
29     liste_zoom = ...
30     for elt in ... :
31         for i in range(k):
32             ...

```

```
33         return liste_zoom
34
35 def zoomDessin(grille,k):
36     '''renvoie une grille où les lignes sont zoomées k fois
37     ET répétées k fois'''
38     grille_zoom=[]
39     for elt in grille:
40         liste_zoom = ...
41         for i in range(k):
42             .... .append(....)
43     return grille_zoom
```

## Résultats à obtenir :

## Script Python

```
>>> affiche(oeuvre)
```



## Script Python

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

Texte

The image shows a grid of 16 columns and 16 rows, where each cell contains an asterisk (\*). The grid is composed of four identical quadrants, each rotated 90 degrees clockwise relative to its neighbors. The top-left quadrant contains the following pattern:

*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*

The other three quadrants follow a similar pattern but are rotated 90 degrees clockwise.