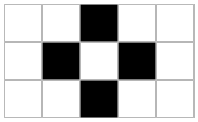
Zoom

Une image rectangulaire en noir et blanc peut être représentée par une liste de lignes qui sont des listes d'entiers valant 0 (pour le noir) et 1 (pour le blanc).
Par exemple l'image suivante, de dimensions 5×4



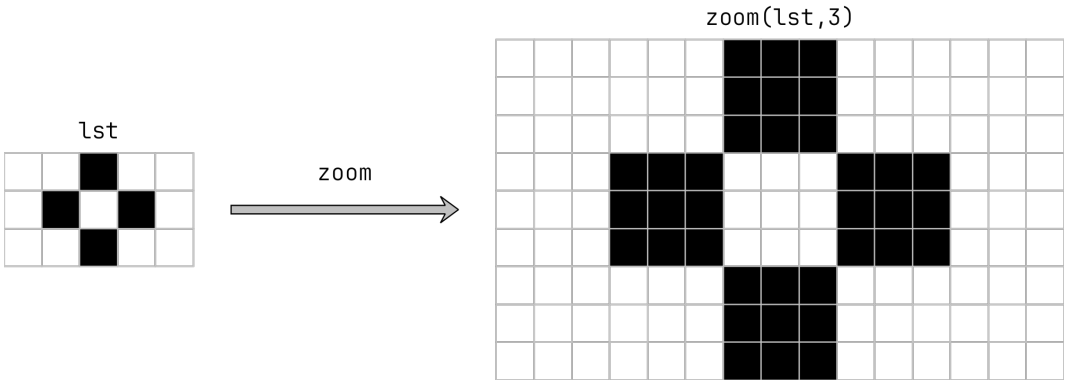
est représentée par la liste suivante :

```
[[0, 0, 1, 0, 0],
 [0, 1, 0, 1, 0],
 [0, 0, 1, 0, 0]]
```

On aimerait construire une fonction `zoom` qui

- en entrée prend une liste `lst` qui représente une image rectangulaire et un `int` strictement positif `k` ;
- renvoie une liste qui correspond à l'image représentée par `lst` grossie d'un facteur `k`.

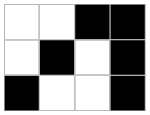
Ci-dessous figure un exemple d'utilisation de la fonction `zoom`



Étape 1

Question 1

Dessiner l'image obtenue en appliquant `zoom(lst,2)` avec une liste `lst` représentant l'image suivante :



Question 2

Si `lst` représente une image de n lignes par p colonnes, et que `lst2 = zoom(lst, k)`, quelle est la taille de

1. `lst2` ?
2. `lst2[0]` ?

Question 3

Compléter le pseudocode de la fonction `zoom_horiz` qui

```
fonction zoom(lst, k)

    variables
        résultat : liste
        valeur, compteur, i : entiers

    résultat ← liste vide
    n ← longueur(lst)
    pour i ...
        pour j ...
            ajouter ... à la fin de résultat
    renvoyer résultat
```

Question 4

Compléter le pseudocode de la fonction `zoom` que l'on veut coder

- en entrée prend une liste d'entiers `ligne` et un entier `k` ;
- renvoie une liste d'entiers dans laquelle chaque valeur de `ligne` est dupliquée `k` fois.

```
fonction zoom_horiz(ligne, k)

    variables
        résultat : liste
        valeur, compteur, i : entiers

    résultat ← liste vide
    p ← longueur(ligne)
    pour i ...
        pour j ...
```

```
                ajouter ... à la fin de résultat
        renvoyer résultat
```

Étape 2

Question 5

Ouvrir le fichier `zoom.py` et coder les fonctions manquantes.

Question 5

Coder la fonction `affiche` qui

- en entrée prend une liste `lst` qui représente une image ;
- ne renvoie rien mais affiche joliment l'image avec des `'.'` à la place des zéros et des `'*'` à la place des 1.

On rappelle que `print('*', end="")` affiche `'*'` sans retour à la ligne.