

23 NSI 03

```
In [ ]: 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, 0, 1, 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]]
```

```
In [ ]: 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()
```

```
In [ ]: 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
```

```
In [ ]: def zoomDessin(grille, k):
    '''renvoie une grille ou 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
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js