

```
def nbLig(image):
    '''renvoie le nombre de lignes de l'image'''
    return len(image)

def nbCol(image):
    '''renvoie la largeur de l'image'''
    return len(image[0])

def negatif(image):
    '''renvoie le négatif de l'image sous la forme
    d'une liste de listes'''
    L = [[0 for k in range(nbCol(image))] for i in range(nbLig(image))] # on cree un
e image de 0 aux memes dimensions que le parametre image
    for i in range(len(image)):
        for j in range(nbCol(image)):
            L[i][j] = 255 - image[i][j]
    return L

def binaire(image, seuil):
    '''renvoie une image binarisée de l'image sous la forme
    d'une liste de listes contenant des 0 si la valeur
    du pixel est strictement inférieure au seuil
    et 1 sinon'''
    L = [[0 for k in range(nbCol(image))] for i in range(nbLig(image))] # on crée un
e image de 0 aux mêmes dimensions que le paramètre image
    for i in range(len(image)):
        for j in range(nbCol(image)):
            if image[i][j] < seuil :
                L[i][j] = 0
            else:
                L[i][j] = 1
    return L

img=[[20, 34, 254, 145, 6], [23, 124, 287, 225, 69], [197, 174, 207, 25, 87], [255,
0, 24, 197, 189]]
assert nbLig(img) == 4
assert nbCol(img) == 5
assert negatif(img) == [[235, 221, 1, 110, 249], [232, 131, -32, 30, 186], [58, 81,
48, 230, 168], [0, 255, 231, 58, 66]]
assert binaire(img,120) == [[0, 0, 1, 1, 0], [0, 1, 1, 1, 0], [1, 1, 1, 0, 0], [1, 0
, 0, 1, 1]]
```