

Enoncé

On considère une image en 256 niveaux de gris que l'on représente par une grille de nombres, c'est-à-dire une liste composée de sous-listes toutes de longueurs identiques.

La largeur de l'image est donc la longueur d'une sous-liste et la hauteur de l'image est le nombre de sous-listes.

Chaque sous-liste représente une ligne de l'image et chaque élément des sous-listes est un entier compris entre 0 et 255, représentant l'intensité lumineuse du pixel.

Le négatif d'une image est l'image constituée des pixels x_n tels que $x_n + x_i = 255$ où x_i est le pixel correspondant de l'image initiale.

Compléter le programme suivant :

 Script Python

```

1 def nbLig(image):
2     '''renvoie le nombre de lignes de l'image'''
3     return ...
4
5 def nbCol(image):
6     '''renvoie la largeur de l'image'''
7     return ...
8
9 def negatif(image):
10    '''renvoie le negatif de l'image sous la forme
11       d'une liste de listes'''
12
13    # on cree une image de 0 aux memes dimensions que le parametre image
14    L = [[0 for k in range(nbCol(image))] for i in range(nbLig(image))]
15
16    for i in range(nbLig(image)):
17        for j in range(...):
18            L[i][j] = ...
19    return L
20
21 def binaire(image, seuil):
22    '''renvoie une image binarisee de l'image sous la forme
23       d'une liste de listes contenant des 0 si la valeur
24       du pixel est strictement inferieure au seuil
25       et 1 sinon'''
26
27    # on cree une image de 0 aux memes dimensions que le parametre image
28    L = [[0 for k in range(nbCol(image))] for i in range(nbLig(image))]
29
30    for i in range(nbLig(image)):
31        for j in range(...):
32            if image[i][j] < ... :
33                L[i][j] = ...
34            else:
35                L[i][j] = ...
36    return L

```

Exemples : **Script Python**

```
>>> img=[[20, 34, 254, 145, 6], [23, 124, 237, 225, 69], [197, 174,  
207, 25, 87], [255, 0, 24, 197, 189]]  
>>> nbLig(img)  
4  
>>> nbCol(img)  
5  
>>> negatif(img)  
[[235, 221, 1, 110, 249], [232, 131, 18, 30, 186], [58, 81, 48, 230,  
168], [0, 255, 231, 58, 66]]  
>>> binaire(img,120)  
[[0, 0, 1, 1, 0], [0, 1, 1, 1, 0], [1, 1, 1, 0, 0], [1, 0, 0, 1, 1]]
```