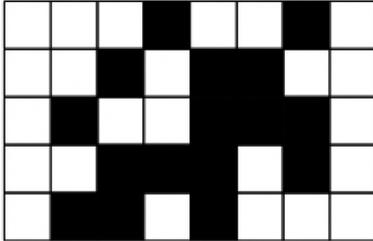


Exercice 5 - Partie A :

1. a.



1. b. Le pixel situé juste au dessus à gauche est à la ligne 3 et à la colonne 0. Celui situé juste au dessus à droite est à la ligne 3 et à la colonne 2.

1. c. Le pixel situé juste au dessus à gauche est à la ligne **li-1** et à la colonne **co-1**. Celui situé juste au dessus à droite est à la ligne **li-1** et à la colonne **co+1**.

2. a. Pour que `image[li][co]` prenne la valeur 1 il faut que les valeurs de `image[li-1][co-1]` et `image[li-1][co+1]` soient différentes.

2. b.

```
def remplir_ligne(image, li):
    image[li][0] = 0
    image[li][7] = 0
    for co in range(1,7):
        if image[li-1][co-1] != image[li-1][co+1]:
            image[li][co] = 1
```

2. c.

```
def remplir(image):
    for li in range(1, len(image)):
        remplir_ligne(image, li)
```

Exercice 5 - Partie B :

1. a. $00101100_{(2)} = 2^5 + 2^3 + 2^2 = 32 + 8 + 4 = 44_{(10)}$

1. b.

```
def conversion2_10(tab):
    nb_bit = len(tab)
    somme = 0
    for i in range(nb_bit):
        somme += tab[i]*2**(nb_bit-1-i)
    return somme
```

1.c. $78_{(10)} = 64 + 8 + 4 + 2 = 2^6 + 2^3 + 2^2 + 2^1 = 0100\ 1110_{(2)}$

Le tableau correspondant à l'entier 78 est donc : [0, 1, 0, 0, 1, 1, 1, 0]

2. a. Les pixels à gauche et à droite sont forcément blancs donc la plus grande valeur représentable est $0111\ 1110_{(2)} = 126_{(10)}$. Le premier bit à droite est toujours nul donc la valeur représentée est obligatoirement paire.

En conclusion : n est forcément un entier pair inférieur ou égal à 126.

2. b.

```
def generer(n, k):
    tab = [None for i in range(k)]
    image = [[0 for j in range(8)] for i in range(k+1)]
    image[0] = conversion10_2(n) # Écriture de la première ligne
    remplir(image)             # Remplissage des autres lignes
    for li in range(1, k+1):
        tab[li-1] = conversion2_10(image[li])
    return tab
```