

Sujet 26

Exercice 1

```
# Avec les méthodes des listes Python
def occurrence_max(chaine):
    alphabet=['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',
    occurrence = [0]*26 # tableau avec 26 zéros
    for c in chaine:
        if c in alphabet:
            i = alphabet.index(c)
            occurrence[i] += 1
    k = occurrence.index(max(occurrence)) # position d'un maximum
    return alphabet[k]

# Sans les méthodes des listes Python
def occurrence_max(chaine):
    alphabet=['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',
    occurrence = [0]*26 # tableau avec 26 zéros
    for c in chaine:
        for i in range(len(alphabet)):
            if c == alphabet[i]:
                occurrence[i] += 1
    max = 0
    for i in range(len(occurrence)):
        if occurrence[i] > max:
            max = occurrence[i]
            k = i
    return alphabet[k]
```

Exercice 2

```
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'''
    # on crée une image de 0 aux mêmes dimensions que le paramètre image
    L = [[0 for k in range(nbCol(image))] for i in range(nbLig(image))]
    for i in range(len(image)):
        for j in range(len(image[i])):
            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'''
    # on crée une image de 0 aux mêmes dimensions que le paramètre image
    L = [[0 for k in range(nbCol(image))] for i in range(nbLig(image))]
    for i in range(len(image)):
        for j in range(len(image[i])):
            if image[i][j] < seuil : # Attention erreur dans le sujet !
                L[i][j] = 0
            else:
                L[i][j] = 1
    return L

```

Le dernier exemple est également faux !