

Corrigé sujet 34 - Année : 2022

Sujet 34 - 20222



1. Exercice 1

Bug

Il y a une faute de frappe dans la variable `alphabet` de l'énoncé (une virgule en trop dans la valeur `'o,'`)

```
1 alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',
2             'j', 'k', 'l', 'm', 'n', 'o,', 'p', 'q', 'r',
3             's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
4
5 def occurrence_max(ch):
6     occurrence = [0]*26
7     for caractere in ch:
8         if caractere in alphabet:
9             index_caractere = alphabet.index(caractere)
10            occurrence[index_caractere] += 1
11    indice_max_occurrence = 0
12    max_occurrence = occurrence[0]
13    for i in range(0, len(occurrence)):
14        if occurrence[i] > max_occurrence:
15            max_occurrence = occurrence[i]
16            indice_max_occurrence = i
17    return alphabet[indice_max_occurrence]
```

Commentaires

1. Le problème du nombre d'occurrence maximal d'un caractère dans un texte peut aussi se résoudre en utilisant un dictionnaire dont les clés sont les caractères et les valeurs les nombres d'occurrences.
2. On utilise dans la correction proposée ici, la méthode `index` qui renvoie l'indice de la première apparition d'un élément dans une liste. On peut aussi rechercher cet indice en effectuant un parcourt de la liste.

2. Exercice 2

```
1 def nbLig(image):
2     '''renvoie le nombre de lignes de l'image'''
3     return len(image) #(1)
4
5 def nbCol(image):
6     '''renvoie la largeur de l'image'''
7     return len(image[0]) #(2)
8
9 def negatif(image):
10    '''renvoie le negatif de l'image sous la forme
11        d'une liste de listes'''
12    L = [[0 for k in range(nbCol(image))] for i in range(nbLig(image))]
13    # on cree une image de 0 aux memes dimensions que le parametre image
14    for i in range(len(image)):
15        for j in range(nbCol(image)): #(3)
16            L[i][j] = 255-image[i][j] #(4)
17    return L
18
19 def binaire(image, seuil):
20    '''renvoie une image binarisee de l'image sous la forme
21        d'une liste de listes contenant des 0 si la valeur
22        du pixel est strictement inferieure au seuil
23        et 1 sinon'''
24    L = [[0 for k in range(nbCol(image))] for i in range(nbLig(image))] # on cree une image de 0 aux memes
25    dimensions que le parametre image
26    for i in range(len(image)):
27        for j in range(nbCol(image)):
28            if image[i][j] < seuil : #(5)
29                L[i][j] = 0
30            else:
31                L[i][j] = 1
32    return L
```

1. Comme indiqué dans l'énoncé, une image est une listes de listes (toutes de même longueur).la hauteur de l'image est le nombres de listes
2. La largeur de l'image est la longueur d'une sous-liste (elles ont toute la même longueur).
3. On utilise la fonction `nbCol` définie plus haut
4. Se référer à l'énoncé : la somme du pixel et de son négatif doit faire 255.
5. On applique l'algorithme donné en commentaire : le pixel est mis à 0 s'il est inférieur au seuil et à 1 sinon.

Attention

1. On peut regretter que le programme définisse les fonctions `nbLig` et `nbCol` afin de récupérer les dimensions d'une image mais ne les utilise pas pour parcourir l'image (voir ligne 14 et 25)