

# Corrigé sujet 24 - Année : 2023

[Sujet 24 - 2022 ↴](#)

## 1. Exercice 1



```

1 def nbr_occurrences(chaine):
2     occ = {}
3     for caractere in chaine:
4         if caractere in occ:
5             occ[caractere] += 1
6         else:
7             occ[caractere]=1
8     return occ

```

### Commentaire

Bon exercice pour revoir l'utilisation des dictionnaires.

## 2. Exercice 2



- Le code présent sur le `pdf` est buggé : on trouve `lst2` comme paramètre mais `L2` dans le corps de la fonction.



```

1 def fusion(lst1,lst2):
2     n1 = len(lst1)
3     n2 = len(lst2)
4     lst12 = [0] * (n1+n2)
5     i1 = 0
6     i2 = 0
7     i = 0
8     while i1 < n1 and i2 < n2 : # (1)
9         if lst1[i1] < lst2[i2]:
10             lst12[i] = lst1[i1] # (2)
11             i1 = i1 + 1

```

```
12     else:
13         lst12[i] = lst2[i2]
14         i2 = i2 + 1
15     i += 1
16     while i1 < n1:
17         lst12[i] = lst1[i1] #(3)
18         i1 = i1 + 1
19         i = i + 1
20     while i2 < n2:
21         lst12[i] = lst2[i2]
22         i2 = i2 + 1
23         i = i + 1
24 return lst12
```

1. `i1` est l'indice de parcours de `L1` (de longueur `n1`) `i2` est l'indice de parcours de `L2` (de longueur `n2`)
2. On se trouve dans le cas où le plus petit élément se trouve dans `L1`, c'est donc lui qui est ajouté à liste fusionnée `L12`.
3. On a atteint la fin de l'une des listes, il reste donc à ajouter les éléments restants de l'autre liste.

### ⚠️ Attention

- Le sujet est difficile, il faut d'abord comprendre l'algorithme utilisé (qui n'est pas expliqué)
- Le même exercice figure en exercice 1 du [sujet 7](#)