

BACCALAURÉAT

SESSION 2024

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°34

DURÉE DE L'ÉPREUVE : 1 heure

**Le sujet comporte 3 pages numérotées de 1 / 3 à 3 / 3
Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

Le candidat doit traiter les 2 exercices.

EXERCICE 1 (10 points)

Le nombre d'occurrences d'un caractère dans une chaîne de caractère est le nombre d'apparitions de ce caractère dans la chaîne.

Exemples :

- le nombre d'occurrences du caractère 'o' dans 'bonjour' est 2 ;
- le nombre d'occurrences du caractère 'b' dans 'Bébé' est 1 ;
- le nombre d'occurrences du caractère 'B' dans 'Bébé' est 1 ;
- le nombre d'occurrences du caractère ' ' dans 'Hello world !' est 2.

On cherche les occurrences des caractères dans une phrase. On souhaite stocker ces occurrences dans un dictionnaire dont les clefs seraient les caractères de la phrase et les valeurs l'occurrence de ces caractères.

Par exemple : avec la phrase 'Hello world !' le dictionnaire est le suivant :

```
{'H': 1, 'e': 1, 'l': 3, 'o': 2, ' ': 2, 'w': 1, 'r': 1, 'd': 1, '!': 1}
```

L'ordre des clefs n'a pas d'importance.

Écrire une fonction `nbr_occurrences` prenant comme paramètre une chaîne de caractères `chaîne` et renvoyant le dictionnaire des nombres d'occurrences des caractères de cette chaîne.

EXERCICE 2 (10 points)

La fonction `fusion` prend deux tableaux `tab1`, `tab2` (type `list`) d'entiers triés par ordre croissant et les fusionne en un tableau trié `tab12` qu'elle renvoie.

Compléter le code de la fonction `fusion` ci-dessous.

```
def fusion(tab1,tab2):
    '''Fusionne deux tableaux triés et renvoie
    le nouveau tableau trié.'''
    n1 = len(tab1)
    n2 = len(tab2)
    tab12 = [0] * (n1 + n2)
    i1 = 0
    i2 = 0
    i = 0
    while i1 < n1 and ...:
        if tab1[i1] < tab2[i2]:
            tab12[i] = ...
            i1 = ...
        else:
            tab12[i] = tab2[i2]
            i2 = ...
        i += 1
    while i1 < n1:
        tab12[i] = ...
        i1 = i1 + 1
        i = ...
    while i2 < n2:
        tab12[i] = ...
        i2 = i2 + 1
        i = ...
    return tab12
```

Exemple :

```
>>> fusion([1,2,3],[ ])
[1, 2, 3]
>>> fusion([ ], [ ])
[ ]
>>> fusion([1, 6, 10],[0, 7, 8, 9])
[0, 1, 6, 7, 8, 9, 10]
```