

BACCALAUREAT

SESSION 2022

Épreuve de l'enseignement de spécialité

NUMERIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°10

DUREE 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 (4 points)

L'occurrence d'un caractère dans une phrase est le nombre de fois où ce caractère est présent.

Exemples :

- l'occurrence du caractère 'o' dans 'bonjour' est 2 ;
- l'occurrence du caractère 'b' dans 'Bébé' est 1 ;
- l'occurrence du caractère 'B' dans 'Bébé' est 1 ;
- l'occurrence 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'ayant pas d'importance).

Écrire une fonction `occurrence_lettres` avec `phrase` comme paramètre une variable de type `str`. Cette fonction doit renvoyer un dictionnaire de type constitué des occurrences des caractères présents dans la phrase.

EXERCICE 2 (4 points)

La fonction `fusion` prend deux listes `L1`, `L2` d'entiers triées par ordre croissant et les fusionne en une liste triée `L12` qu'elle renvoie.

Le code Python de la fonction est

```
def fusion(L1, L2):
    n1 = len(L1)
    n2 = len(L2)
    L12 = [0]*(n1+n2)
    i1 = 0
    i2 = 0
    i = 0
    while i1 < n1 and ... :
        if L1[i1] < L2[i2]:
            L12[i] = ...
            i1 = ...
        else:
            L12[i] = L2[i2]
```

```
        i2 = ...
    i += 1
while i1 < n1:
    L12[i] = ...
    i1 = i1 + 1
    i = ...
while i2 < n2:
    L12[i] = ...
    i2 = i2 + 1
    i = ...
return L12
```

Compléter le code.

Exemple :

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

```
def occurrence_lettres(phrase):
    occurrences = {}
    for lettre in phrase:
        if lettre in occurrences:
            occurrences[lettre] += 1
        else:
            occurrences[lettre] = 1
    return occurrences

assert occurrence_lettres('Hello world !') == {'H': 1, 'e': 1, 'l': 3, 'o': 2, ' ': 2,
, 'w': 1, 'r': 1, 'd': 1, '!': 1}
```

```
def fusion(L1, L2):
    n1 = len(L1)
    n2 = len(L2)
    L12 = [0]*(n1+n2)
    i1 = 0
    i2 = 0
    i = 0
    while i1 < n1 and i2 < n2 :
        if L1[i1] < L2[i2]:
            L12[i] = L1[i1]
            i1 = i1 + 1
        else:
            L12[i] = L2[i2]
            i2 = i2 + 1
        i += 1
    while i1 < n1:
        L12[i] = L1[i1]
        i1 = i1 + 1
        i = i + 1
    while i2 < n2:
        L12[i] = L2[i2]
        i2 = i2 + 1
        i = i + 1
    return L12

print(fusion([1,6,10],[0,7,8,9]))
```