

Ce devoir contient deux exercices indépendants. Indiquez votre parcours et le numéro de votre groupe, ainsi que votre nom et prénom

- en tête du fichier que vous remettrez si vous rédigez vos réponses sur ordinateur ;
- en tête de chacune des feuilles si vous composez sur papier.

Sauf mention explicite du contraire, il n'est pas attendu de documentation complète pour les fonctions demandées mais uniquement les **conditions d'utilisation (CU)**. En revanche, toute autre fonction non explicitement demandée, mais dont vous estimez avoir besoin, doit être documentée.

En cas de besoin, vous pouvez poser toute question relative à ce sujet par courrier électronique à l'adresse `sos-ap1@univ-lille1.fr`

Exercice 1 *Questions diverses*

Dans cet exercice, il n'est pas demandé de réaliser des fonctions. Vous répondrez à chacune des questions en donnant une séquence d'instructions permettant de réaliser ce qui est demandé (le plus souvent une seule instruction suffit).

Question 1 Étant donnée une chaîne de caractères `s`, construisez la liste des chaînes extraites de `s` en découpant `s` selon le caractère `' ; '`.

Question 2 Étant donnée une liste de chaînes de caractères `l_chaines`, construisez la chaîne obtenue en concaténant toutes les chaînes de cette liste.

Question 3 Étant donné un entier `n`, construisez une liste de tous les entiers impairs compris entre 0 et `n` exclus.

Question 4 Étant donnée une liste d'entiers `l_entiers`, construisez une liste de tous les entiers de cette liste divisibles par 3.

Question 5 Étant donnée une liste `liste`, construisez une liste de tous les couples (x, x) avec `x` éléments de `liste`. Par exemple, avec la liste `[1, 2, 3]` on doit obtenir la liste

`[(1, 1), (2, 2), (3, 3)]`.

Question 6 Étant donnée une liste `liste`, construisez une liste de tous les couples (x, y) , `x` et `y` étant deux éléments consécutifs de `liste`. Par exemple, avec la liste `[1, 2, 3]` on doit obtenir la liste

`[(1, 2), (2, 3)]`.

Question 7 Étant donnée une liste `liste`, construisez la liste de tous les couples (x, y) , `x` et `y` étant deux éléments quelconques de `liste`. Par exemple, avec la liste `[1, 2, 3]` on doit obtenir la liste

`[(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)]`.

Supposons donnée une liste d'informations sur les pays du monde dans laquelle ces informations sont présentées sous la forme (sigle, nom, superficie, population, continent) dont voici le début :

```
liste_infos_pays = [('AD', 'Andorra', 468.0, 84000, 'EU'),
                   ('AE', 'United Arab Emirates', 82880.0, 4975593, 'AS'),
                   ('AF', 'Afghanistan', 647500.0, 29121286, 'AS'),
                   ...
                   ]
```

Question 8 Construisez un dictionnaire dont les clés sont les sigles des pays de la liste `liste_infos_pays` et les valeurs associées sont les tuples (nom, superficie, population, continent) correspondant. Vous nommerez `dico_infos_pays` ce dictionnaire.

Question 9 Construisez l'ensemble des noms de pays figurant dans le dictionnaire `dico_infos_pays`.

Question 10 Calculez la population totale obtenue en cumulant les populations de tous les pays de ce dictionnaire.

Question 11 Calculez le nombre de pays situés sur le continent Asie ('AS').

Exercice 2 Analyse de textes

Le thème de cet exercice est l'analyse statistique d'occurrences de caractères dans un texte.

Question 1 Sans utiliser la méthode `count`, réalisez une fonction `nbre_occurrences(car, texte, deb, fin)` qui renvoie le nombre d'occurrences du caractère `car` dans la chaîne `texte` entre les indices `deb` inclus et `fin` exclu. Vous donnerez une spécification complète de cette fonction sous forme d'une docstring. Dans les exemples qui suivent, on utilisera la variable `phrase` dont la valeur est la chaîne de 37 caractères encadrée ci-dessous :

```
Timoleon est un homme politique grec.
```

```
>>> nbre_occurrences('o', phrase, 0, len(phrase))
4
>>> nbre_occurrences('o', phrase, 4, 18)
2
>>> nbre_occurrences('o', phrase, 3, 3)
0
```

Question 2 Donnez le nombre de comparaisons effectuées entre le caractère `car` et les caractères du texte. Vous exprimerez ce nombre en fonction de la longueur $\ell = \text{fin} - \text{deb}$ de l'extrait de texte examiné.

On souhaite déterminer la fréquence d'occurrences dans une fenêtre de texte, la fenêtre ayant une certaine longueur, et une position de début. La longueur d'une fenêtre est ici le nombre de caractères de la sous-chaîne observée par la fenêtre. La figure 1 montre une fenêtre sur notre texte exemple débutant à l'indice 4 et de longueur 25. Dans cette fenêtre il y a 3 occurrences du caractère `o` et la fréquence de ce caractère dans la fenêtre est donc $\frac{3}{25} = 0.12 = 12.0\%$.

```
Timoleon est un homme politique grec.
```

FIGURE 1 – Une fenêtre de longueur 25 débutant à l'indice 4 sur un texte

Question 3 Donnez une expression PYTHON utilisant la fonction `nbre_occurrences` qui calcule la fréquence d'un caractère `car` dans une fenêtre de longueur ℓ débutant à l'indice `debut` dans un texte `texte`.

On veut réaliser une fonction qui donne les fréquences d'un caractère donné dans toutes les fenêtres d'une longueur donnée d'un texte. La spécification de cette fonction est donnée ci-dessous,

```
def frequences(car, texte, longueur):
    """
    :param car: (str) le caractère étudié
    :param texte: (str) le texte dans lequel l'étude se fait
    :param longueur: (int) longueur des fenêtres consécutives
    :return: (list) liste des fréquences de car dans les fenêtres consécutives du texte.
    """
```

et en voici un exemple d'utilisation avec la variable `phrase` :

```
>>> frequences('o', phrase, 25)
[0.16, 0.16, 0.16, 0.16, 0.12, 0.12, 0.12, 0.08, 0.08, 0.08, 0.08, 0.08, 0.08]
```

Question 4 Exprimez en fonction de la longueur n du texte et de la longueur ℓ de la fenêtre la longueur de la liste des fréquences que doit produire la fonction `frequencies`, en réfléchissant bien aux contraintes d'utilisation.

Question 5 Donnez une première réalisation de cette fonction en utilisant la fonction `nbre_occurrences`.

Question 6 Donnez en fonction de la longueur n du texte et de la longueur ℓ de la fenêtre, le nombre de comparaisons entre le caractère `car` et un caractère du texte.

Question 7 Un linguiste fait une étude sur les symboles de ponctuation dans le roman *Les Misérables* de Victor Hugo. Il fait une étude sur des fenêtres de longueur 10^5 caractères. Sachant que ce roman comprend environ $3,5 \times 10^6$ caractères, et que l'ordinateur qu'il utilise effectue 10^5 comparaisons de caractères à la seconde, calculez le temps que met votre programme pour faire l'analyse.

Il s'agit maintenant de réaliser une seconde version plus performante pour analyser les fréquences d'un caractère dans des fenêtres consécutives (i.e. d'indices de début consécutifs) de même longueurs.

Cette version s'appuie sur la différence existant entre les nombres d'occurrences d'un caractère dans deux fenêtres consécutives, comme le montre la figure 2.

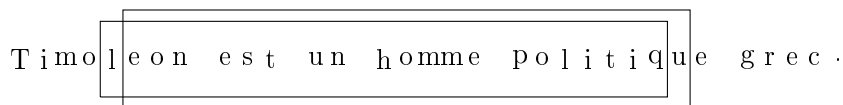


FIGURE 2 – Deux fenêtres consécutives de longueur 25

Question 8 Quelle(s) différence(s) peut-il exister entre les nombres d'occurrences d'un caractère `car` dans deux fenêtres consécutives de même longueur? Décrivez précisément les différents cas.

Question 9 En tenant compte des différences établies dans la question précédente, réalisez une deuxième version de la fonction `frequencies` que vous nommerez `frequencies2`.

Question 10 Quel est le coût en nombre de comparaisons de caractères de cette nouvelle version? Vous l'exprimerez en fonction de la longueur n du texte et de la longueur ℓ des fenêtres consécutives.

En fait l'analyse statistique ne porte pas sur un seul caractère, mais sur plusieurs (par exemple, les symboles de ponctuation). Le premier paramètre de la fonction `frequencies` n'est plus un simple caractère, mais une séquence de caractères. Et la valeur renvoyée par la fonction n'est plus une simple liste de fréquences, mais un dictionnaire dont les clés sont les caractères étudiés et les valeurs associées les listes de fréquences observées.

```
>>> freqs = frequencies3(('o', 'e'), phrase, 25)
>>> freqs['o']
[0.16, 0.16, 0.16, 0.16, 0.12, 0.12, 0.12, 0.08, 0.08, 0.08, 0.08, 0.08, 0.08]
>>> freqs['e']
[0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12]
```

Question 11 Réalisez cette nouvelle version de la fonction `frequencies`, que vous nommerez `frequencies3`, en vous efforçant de limiter au maximum les comparaisons de caractères.