

Structures séquentielles

Exercice n°1 : création de listes

- Écrire une fonction `liste_inv` qui, étant donné un entier naturel non nul n , renvoie la liste des réels de forme $\frac{1}{k}$ pour $k \in \llbracket 1; n \rrbracket$.
- Écrire une fonction `liste_harm` qui, étant donné un entier naturel non nul n , renvoie la liste des réels de la forme $\sum_{j=1}^k \frac{1}{j}$ pour $k \in \llbracket 1; n \rrbracket$.

Exercice n°2 : opération sur une liste

- Écrire une fonction `somme` qui, étant donnée une liste L de nombre entiers ou flottants, renvoie la somme des nombres apparaissant dans L .
- Écrire une fonction `moyenne` qui, étant donnée une liste L de nombre entiers ou flottants, renvoie la moyenne des nombres apparaissant dans L .
- Écrire une fonction `médiane` qui, étant donnée une liste L de nombre entiers ou flottants, renvoie la médiane des nombres apparaissant dans L .
- Écrire une fonction `indice_maximum` qui, étant donnée une liste L de nombre entiers ou flottants, renvoie le maximum de ces nombres avec son indice dans la liste. Si plusieurs éléments sont égaux au maximum, la fonction renvoie l'indice le plus petit parmi les indices de ces éléments.
- Écrire une fonction `indices_minimum` qui, étant donnée une liste L de nombre entiers ou flottants, renvoie le minimum de ces nombres avec la liste des indices de tous les éléments égaux au minimum.
- Écrire une fonction `décalage_droite` qui, étant donnée une liste L de nombre entiers ou flottants, renvoie la même liste dans laquelle tous les contenus ont été décalés d'un cran vers la droite (avec la convention que le dernier terme passe en première position).

Exercice n°3 : opération sur une chaîne de caractère et dictionnaires

- Écrire une fonction `longueur` qui prend en argument une chaîne de caractères et renvoie longueur de celle-ci (il est interdit d'utiliser la fonction `len()`)
- Écrire une fonction `recherche_e` qui prend en argument une chaîne de caractères, qui peut représenter un mot ou un texte, et qui renvoie le nombre de 'e' contenus dans la chaîne.
- Écrire une fonction `stat` qui prend en argument un texte (type `str`) et renvoie un dictionnaire dont les clés sont les différentes lettres du texte et les valeurs le nombre d'occurrences de chaque lettre. On suppose le texte écrit en lettres capitales non accentuées. On ne comptabilise pas les autres caractères (espaces, ponctuation).

Exercice n°4 : intérêt des dictionnaires

La question est de mesurer l'intérêt d'une représentation par un dictionnaire plutôt que par une liste de listes. Nous allons donc comparer le temps de recherche d'un élément.

- Construire une liste dont les éléments sont de la forme $[i, i]$ pour i allant de 0 à $10^6 - 1$. Mélanger cette liste avec la fonction `shuffle` du module `random`. Il suffit d'importer la fonction et d'écrire `shuffle(liste)`. Créer alors le dictionnaire correspondant qui contient les couples clé:valeur de la forme $i:i$ à l'aide de la fonction `dict`.
- Écrire une fonction `recherche1` qui prend en paramètres une liste de listes et une variable `k` et renvoie le deuxième élément de la sous-liste dont le premier élément a la valeur de `k`.
- Écrire une fonction `recherche2` qui prend en paramètres un dictionnaire et une variable `k` et renvoie la valeur correspondant à la clé `k`.
- Tester les fonctions de recherche sur une liste et le dictionnaire en utilisant pour le paramètre `k` les valeurs de 0 à 49. Pour les mesures, utiliser la fonction `time` du module `time`. Le mode d'utilisation est :

```

From time import time
top=time()
#ici le programme à exécuter
print(time() - top)

```