
Devoir maison n°1 - Version 1 - A rendre sur papier le jeudi 5 novembre

Consignes : Ne pas utiliser les fonctions intégrées (*max*, *min*, *sorted*, etc.).

Vous pouvez utiliser *len*, sauf dans l'exercice 3.

Exercice 1 : Écrire une fonction `positions(L,elt)` qui retourne la liste de toutes les positions d'un élément donné dans une liste.

Exercice 2 : Écrire une fonction `est_inclu(L1,L2)` qui détermine si tous les éléments de `L1` sont aussi des éléments de `L2`. Cette fonction devra retourner un booléen. On suppose que chacune de deux listes ne contient pas de doublons.

Exercice 3 : Écrire une fonction `longueur(L)` qui détermine la longueur d'une liste `L` sans utiliser `len`.

Exercice 4 : Écrire une fonction `retourne(chaine)` qui inverse une chaîne de caractères sans utiliser de fonctions intégrées. La chaîne de départ passée en argument ne devra pas être modifiée.

Exercice 5 : Écrire une fonction qui renvoie le plus grand nombre impair d'une liste d'entiers. Si aucun nombre impair n'existe, la fonction renvoie `None`.

Exercice 6 : Écrire une fonction `filtrer_dictionnaire(d, seuil)` qui prend un dictionnaire de valeurs numériques et un nombre `seuil`, et retourne un nouveau dictionnaire ne conservant que les paires clé-valeur dont la valeur est supérieure ou égale au seuil.

Exercice 7 : Écrire une fonction `fusion_dictionnaires(d1, d2)` qui prend deux dictionnaires dont les valeurs sont des entiers et renvoie un nouveau dictionnaire fusionné. Si une même clé est présente dans les deux dictionnaires, sa valeur dans le résultat doit être la somme des deux valeurs.

Exercice 8 :

1. Écrire une fonction `occurrences(L)` qui à partir d'une liste retourne un dictionnaire qui a pour clés les éléments distincts de la liste et pour valeur le nombre d'occurrences de chaque élément dans la liste.
2. Écrire une fonction `clef_valeur_max(D)` qui renvoie la clé associée à la valeur maximale d'un dictionnaire (le dictionnaire passé en argument est supposé non vide, avec des valeurs qui sont toutes de type entier).
3. Écrire une fonction `phrase_to_liste(chaine)` qui prend une phrase (sous forme d'une chaîne de caractères) et renvoie la liste de tous les mots de cette phrase. La liste renvoyée est donc une liste de chaîne de caractères. On suppose que les mots sont tous séparés par le caractère "espace", et tous écrits en minuscules.
4. Écrire une fonction qui prend une phrase (sous forme d'une chaîne de caractères) et renvoie le mot le plus fréquent. On suppose que les mots sont tous séparés par le caractère "espace", et tous écrits en minuscules.

Cette fonction devra impérativement utiliser les trois fonctions précédemment créées dans cet exercice.