Épisode III : Les listes (encore) et les dictionnaires

EXERCICE 1

Écrire une fonction `supprimeDoublons(liste)` qui prend comme argument une liste et renvoie la liste sans doublons.

EXERCICE 2

Écrire une fonction `echanger(L,i,j)` qui prend comme arguments une liste et deux indices et qui échange les deux éléments correspondants. Par exemple, echanger ([1, 2, 5, 8], 0, 3) devra retourner [8, 2, 5, 1]. On supposera que les deux indices donnés sont valides (dans le cas contraire, l'appel de cette fonction engendrera une erreur).

EXERCICE 3

En utilisant la fonction echanger, écrire une fonction `inverser(L)` qui prend comme argument une liste et renvoie la liste inversée (le 1er élément est devenu le dernier, le 2ème l'avant-dernier, etc.). Attention, il est possible que la liste à inverser soit vide.

EXERCICE 4

Écrire une fonction `mediane(L)` qui prend comme argument une liste non vide de nombres et renvoie la valeur médiane de cette liste. On utilisera l'instruction sorted(L) qui permet de trier la liste L par ordre croissant.

EXERCICE 5

Écrire une fonction `moinsDeSix(L)` qui prend comme argument une liste de mots (chaînes de caractères) et renvoie deux listes : la première contient les mots de moins de six lettres, la seconde les mots de six lettres ou plus (si m est une chaîne de caractères, len(m) renvoie le nombre de caractères de la chaîne m). Là encore, il est possible que la liste argument soit vide.

EXERCICE 6

On considère le dictionnaire suivant dont les clés sont les noms des élèves et les valeurs des clés sont les moyennes générales obtenues à la fin de l'année universitaire :

```
etudiants = {"etudiant_1" : 13 , "etudiant_2" : 17 , "etudiant_3" : 9 ,
"etudiant_4" : 15 , "etudiant_5" : 8 , "etudiant_6" : 14 , "etudiant_7" : 16 ,
"etudiant_8" : 12 , "etudiant_9" : 13 , "etudiant_10" : 15 , "etudiant_11" : 14 ,
"etudiant_12" : 9 , "etudiant_13" : 10 , "etudiant_14" : 12 , "etudiant_15" : 13 ,
"etudiant_16" : 7 , "etudiant_17" : 12 , "etudiant_18" : 15 , "etudiant_19" : 9 ,
"etudiant_20" : 17}
```

1. Écrire un programme Python qui partitionne ce dictionnaire en deux sous dictionnaires :
   (a) **etudiantAdmis** dont les clés sont les étudiants admis et les valeurs des clés sont les moyennes obtenues (moyennes supérieures ou égales à 10).
   (b) **etudiantNonAdmis** dont les clés sont les étudiants non admis et les valeurs des clés sont les moyennes obtenues (moyennes inférieures strictement à 10).
2. Écrire une fonction `moyenneDesNotes(etudiants)` qui renvoie la moyenne des notes des 20 étudiants.

EXERCICE 7

Écrire une fonction `premieresOccurences(L)` qui renvoie un dictionnaire dont les clés sont les éléments de la liste L, et les valeurs sont les premières occurrences de chaque élément.

EXERCICE 8

Écrire une fonction `dernieresOccurences(L)` qui renvoie un dictionnaire dont les clés sont les éléments de la liste L, et les valeurs sont les dernières occurrences de chaque élément.

EXERCICE 9

1. Écrire une fonction nommée `compteLesCaracteres()` qui accepte une chaîne de caractères et qui renvoie l'occurrence des caractères contenus dans la chaîne sous forme de dictionnaire.
2. Écrire une fonction nommée `compteMotsLigne()` qui accepte une chaîne de caractères et qui renvoie l'occurrence des mots contenus dans la chaîne sous forme de dictionnaire.

EXERCICE 10

On dispose d'un dictionnaire dont les clefs sont le nom de personnages, et les valeurs des tuples contenant la force (int), l'intelligence (int) et la description (str) de ce personnage. Voici un extrait de ce dictionnaire :

```
superHero={
    'Spiderman' : (5, 5, 'araignée à quatre pattes'),
    'Hulk':(4,7,"Grand homme vert"),
    'Agent 13':(2,6,'agent 13'),
    'H':(3, 7, 'Expert en graphes')
}
```

1. Pour accéder à la force de Superman, on écrit forceDeSuperman = superHero['Superman'][0]. Dans le même esprit, compléter le code suivant :

```
intelligenceDeHulk = ...
descriptionDeDrax = ...
(forceDeGamora , intelligenceDeGamora , descriptionDeGamora) = ...
```

2. Écrire une fonction intelligenceMoyenne qui prend en paramètre un tel dictionnaire et qui renvoie la valeur moyenne de l'intelligence de tous les personnages.
3. Écrire une fonction kikelplusfort qui prend en paramètre un tel dictionnaire et qui renvoie le nom du personnage le plus fort.
4. Écrire une fonction combienDeCretins qui prend en paramètre un tel dictionnaire et qui renvoie le nombre de personnages dont l'intelligence est strictement inférieure à la moyenne.