

EXERCICE 1

Nom	Téléphone
Patrick	0612345678
Marie	0687654321
Hanae	0765432198
Piotr	0777666555

On considère les données suivantes :

1. Créer un dictionnaire **répertoire** qui associe à chaque nom son numéro de téléphone.
2. Afficher le téléphone de Piotr.
3. Afficher un booléen indiquant si "Fanny" est enregistrée dans le répertoire.
4. Modifier le numéro de Patrick pour qu'il se termine par un 9 et non un 8.
5. Ajouter "Raoul" dont le numéro est "0789898989"
6. Supprimer "Marie" du répertoire.
7. Afficher le répertoire sous la forme :

Nom : Numéro :

Nom : Numéro :

etc.

EXERCICE 2

Le Scrabble est un jeu de société où l'on doit former des mots avec un tirage aléatoire de lettres, chaque lettre valant un certain nombre de points.

1. écrire un Dictionnaire appelé **scrabble** qui, à chaque lettre associe sa valeur. Cherchera les valeurs sur Internet.
2. écrire alors une fonction **nbPoints** qui prend un mot en argument et qui renvoie le nombre de points de ce mot.

EXERCICE 3

1. Écrire une fonction **compteLettre** qui admet pour argument une chaîne de caractères et qui construit et renvoie un dictionnaire contenant le nombre d'occurrences de tous les caractères utilisés dans la chaîne. La clé est la lettre et la valeur correspondante est le nombre d'occurrence de cette lettre dans la chaîne.
2. Pour reconnaître la langue dans laquelle un texte est écrit, on utilise l'indice de coïncidence. Celui-ci est défini comme la somme des carrés des fréquences d'apparition de chaque lettre dans le texte.

Écrire une fonction **indiceCoïncidence** qui prend en argument un dictionnaire du type de celui de la question précédente et qui renvoie l'indice de coïncidence correspondant.

3. Chercher sur internet l'indice de coïncidence du français, de l'anglais, de l'espagnol, de l'italien et de l'allemand et créer un dictionnaire qui associe à chaque langue son indice.
4. Pour déterminer la langue du texte on regarde quel est l'indice le plus proche de celui obtenu avec le texte en question.

Écrire une fonction **langue** qui prend en argument un dictionnaire des indices de coïncidence et une valeur qui représente l'indice du texte évalué, et qui renvoie la langue dans laquelle le texte est écrit.

EXERCICE 4

1. Le code suivant permet de créer une liste de dictionnaires contenant des informations sur des célébrités :

```
import csv

with open('celebrities.csv') as csvfile:
    reader = csv.DictReader(csvfile)
    listePersonnes = []
    for row in reader:
        listePersonnes.append(row)
```

2. Créer une fonction `affiche` qui affiche la liste des dictionnaires en allant à la ligne pour chaque dictionnaire. Afficher alors la liste `listePersonnes`.
3. Écrire un code qui affiche uniquement les personnes qui ont moins de 50 ans.
4. Écrire un code qui affiche uniquement les personnes dont le prénom est avant le nom dans l'ordre alphabétique.
5. Écrire un code qui affiche uniquement les prénoms des personnes qui ont moins de 50 ans.
6. Écrire un code qui affiche uniquement les noms des femmes de moins de 40 ans.

Pour trier la liste suivant un critère on utilise la fonction `sort` en lui passant en argument un paramètre nommé `key` qui permet d'indiquer selon quel critère on trie. Par exemple par ordre alphabétique de nom, ou alors par âge...

Le code suivant permet de trier la liste `liste` selon l'âge.

```
liste.sort(key=lambda col : int(col['Age']))
```

7. Afficher la liste triée selon l'âge.
8. Trier puis afficher la liste par ordre alphabétique de prénoms.
9. Trier puis afficher la liste par ordre alphabétique de noms.
10. On dit qu'il y a un doublon si la liste contient au moins deux personnes ayant le même nom et le même prénom.

On souhaite rechercher les doublons de la liste. Pour cela on utilisera le fichier `celebriteDB.csv` qui contient des doublons.

- (a) Écrire une fonction `doublon` qui prend en argument une liste (supposée être une liste de dictionnaire comme `listePersonne`) et qui renvoie le "premier" couple (i, j) rencontré tel que $L[i] = L[j]$ avec $i \neq j$. S'il n'y a pas de doublon renvoyer l'entier 0.
- (b) En déduire une fonction `enleveDoublons` qui prend `L` en argument et qui enlève tous les doublons de la liste, puis renvoie la liste ainsi modifiée.
- (c) Afficher alors les deux listes, avant et après l'action de `enleveDoublons` pour vérifier que les doublons ont bien été retirés.