Données en tables

1. Commencer par récupérer les fichiers Prenoms1.csv et Prenoms2.csv.
   Le premier fichier contient les prénoms d'enfants nés à Strasbourg en 2008.
   Le second fichier contient ceux d'enfants nés à Rennes en 2007.
   Ces données proviennent d'un fichier obtenu sur https://www.data.gouv.fr/.
2. Créer ensuite un fichier python dans le même répertoire que les deux fichiers.
   On commence par y importer le module csv, avec lequel on ouvre le fichier Prenoms1.csv.
   Copier les lignes suivantes dans le fichier :

```
import csv
with open('Prenoms1.csv') as csvfile1:
    data1 = list(csv.reader(csvfile1))
```

3. On va tout d'abord reprendre ces données pour obtenir des p-uplets et utiliser un type int pour les entiers.
   Ajouter pour cela le code suivant dans le fichier :

```
champs = tuple(data1[0])

# La liste des champs de données est donc :
print(champs)

indice={champs[i]:i for i in range(len(champs))}
s =indice['Sexe']
# s est l'indice du champ 'Sexe' de la table dans les données
p =indice['Prénom']
# p est l'indice du champ 'Prénom' de la table dans les données
nb=indice['Nombre']
# nb l'indice du champ 'Nombre' de la table dans les données

table1 = []
for ligne in data1[1:]:
    ligne[nb]=int(ligne[nb])
    table1.append(tuple(ligne))

# Ainsi, la première ligne des données de la table :
print(table1[0])
```

   On utilisera maintenant la variable table1 pour parcourir les données.
4. Ajouter et exécuter un code permettant d'afficher la liste des prénoms qui ont été donnés au moins 30 fois.
5. Les données dans la table sont triées d'abord par sexe, puis par nombre de naissances.
   Pour la trier seulement par prénom, on peut utiliser la fonction `sorted`, avec le paramètre `key` de la manière suivante :

```
table_triee=sorted(table,key=lambda ligne:ligne[p])

# si on veut trier par odre décroissant :
table_triee_d=sorted(table,key=lambda ligne:ligne[p],reverse=True)
```

En utilisant le fait que la liste `table_triee` est triée par prénom, vérifier s'il y a ou non un prénom apparaissant deux fois (il existe des prénoms utilisés à la fois pour les hommes et les femmes).

On pourra définir une fonction qui renvoit `True` s'il y a un double, `False` sinon.

6. (a) Définir une nouvelle variable `table_triee_nb` contenant le contenu de `table1` trié selon la colonne du champ 'Nombre', par ordre décroissant.
   (b) Retrouver alors la liste des prénoms donnés au moins 30 fois, en utilisant cette fois le fait que la liste `table_triee_nb` soit triée par ordre décroissant de nombres.
   On doit ici voir les prénoms masculins intercalés avec les prénoms féminins.
7. Nous allons maintenant fusionner les données du fichier Prenoms1.csv avec celles du fichier Prenoms2.csv.
   (a) Ouvrir et transformer de même manière que pour le premier les données du deuxième fichier en les mettant dans une variable table2 (suivre la même méthode en transformant en entiers les données numériques).
   On admet (mais on peut le vérifier) que les données ont les mêmes champs, dans le même ordre, avec les mêmes domaines de valeurs (c'est à dire les mêmes types de données).
   (b) Se contenter de concaténer les deux tables comme elles sont pour l'instant n'aurait pas de sens : on va préférer rajouter deux champs, 'Ville' et 'Année'. On a indiqué au tout début que le premier fichier concerne Strasbourg en 2008 et que le second concerne Rennes en 2007.
   Créer alors une variable table contenant l'ensemble des données des deux tables auxquelles sont ajoutées, pour chaque ligne, les deux données des champs supplémentaires.
   (c) Compléter également la variable champs avec les deux nouveaux champs.
   (d) On va maintenant exporter le tout dans un fichier csv nommé Prenoms.csv. Pour ce faire, copier et exécuter le code suivant :

```
data=[champs]+table
with open('Prenoms.csv', 'w') as csvfile:
    ecriture = csv.writer(csvfile)
    for ligne in data:
        ecriture.writerow(ligne)
```

   Vérifier alors à l'aide d'un tableur ou d'un simple éditeur de texte, le contenu du fichier.

Remarque La fusion de tables n'est en fait pas quelque chose de si simple. Il y a différentes manières possibles de fusionner des tables.
L'une d'elles est de **concaténer** simplement, comme on l'a fait, lorsque les champs sont les mêmes, dans le même ordre et ont les mêmes domaines de valeur.
L'autre est de faire une **jointure**, dans le cas où les tables se complètent l'une l'autre, avec des champs communs mais surtout des champs différents. Par exemple on considère une table dont les champs sont 'Pays' et 'Capitale', et un autre table dont les champs sont 'Pays' et 'Langue'. Alors on peut fusionner les tables avec une jointure sur le 'Pays'. Dans le cas où un Pays n'existe que dans une des tables, on peut choisir de ne pas le mettre dans la jointure, ou de mettre une valeur None pour le champ manquant. Ces choses-là seront vues plus en détail en terminale.

8. Question subsidiaire : la variable table contenant toutes nos données, écrire du code Python permettant d'obtenir le nombre total de « Léane »