

```
# Code écrit par Morgane Farez L1 CMI Informatique USMB
### FICHIERS ET FONCTIONS TESTS ###
```

```
### BIBLIOTHEQUES ###
```

```
import csv
import timeit
import matplotlib.pyplot as plt
#from Hash1 import *
from Hash2 import *
#from Hash3 import *
```

```
### FONCTIONS ###
```

```
def import_LesMiserables() :
    """
    Entree:
        None
    Sortie:
        tab: tableau de chaines de caracteres
    """
    tab = []
    with open ("LesMiserables.txt", newline="", encoding='utf-8') as csvfile :
        spamreader = csv.reader (csvfile , delimiter=',')
        for row in spamreader :
            if row != []:
                tab.append(row)
    return tab
```

```
def import_etudiants() :
    """
    Entree:
        None
    Sortie:
        tab: tableau de chaines de caracteres
    """
    tab = []
    with open ("liste-etu.txt ", newline="") as csvfile :
        spamreader = csv.reader (csvfile , delimiter=',')
        for row in spamreader :
            tab.append(row[0])
    return tab
```

```
def import_baudelaire() :
    """
    Entree:
        None
    Sortie:
```

```

    tab: tableau de chaines de caracteres
"""
tab = []
with open ("baudelaire.txt", newline =",", encoding='utf-8') as csvfile :
    spamreader = csv.reader (csvfile , delimiter =',')
    for row in spamreader :
        if row != []:
            tab.append(row)
return tab

def compteLettresDico(tableau):
    """
    creer un dictionnaire, ajoute et compte les apparitions de tous les
    caracteres rencontres dans un fichier comme LesMiserables.txt
    Parameters
    -----
    tableau : tableau de tableaux de chaines de caracteres
    Returns
    -----
    dicoLettres : dictionnaire python
    """
    dicoLettres = dict()
    for ligne in tableau:
        if len(ligne) > 0:
            for mot in ligne:
                for lettre in mot:
                    if not lettre in dicoLettres:
                        dicoLettres[lettre] = 1
                    else:
                        dicoLettres[lettre] += 1
    return dicoLettres

def compteLettresTable(tableau):
    """
    creer une table de hachage ajoute et compte les apparitions de tous les
    caracteres rencontres dans un fichier comme LesMiserables.txt
    Parameters
    -----
    tableau : tableau de tableaux de chaines de caracteres
    Returns
    -----
    table : table de hachage
    renvoie l'adresse memoire de la table de hachage. Ajouter '.table' pour
    visualiser la table ou faire print(str(compteLettresTable(tableau)))
    """
    table = HashTable(1)
    elt = 0 #compteur du nombre d'éléments inseres

```

```

for ligne in tableau:
    if len(ligne) > 0:
        for mot in ligne:
            for lettre in mot:
                valeur = table.recherche_valeur(lettre)
                if valeur == None or valeur == -1:
                    table.ajouter_valeur(lettre, 1)
                    elt += 1
                else:
                    table.ajouter_valeur(lettre, valeur + 1)

                #invariant de boucle: verifie que le nombre d'elements inséré est cohérent
                #assert elt == table.nb_elements, lettre

# creation du nuage de points
plt.scatter(tab_nb_elements, collision, color = 'k')
plt.ylabel('Collisions')
plt.ylim(0, 25)
plt.xlabel("Nombre d'éléments")
return table

```

```

def compteMotsTable(tableau, longueur_chaine):
    """
    creer une table de hachage ajoute et compte les apparitions de chaines de
    caracteres, de taille longueur_chaine, rencontres dans le fichier LesMiserables.txt
    Parameters
    -----
    tableau : tableau de tableaux de chaines de caracteres
    longueur_chaine : entier
    Returns
    -----
    table : table de hachage
    renvoie l'adresse memoire de la table de hachage. Ajouter '.table' pour
    visualiser la table ou faire print(str(compteMotsTable(tableau, longueur_chaine))).
    """
    table = HashTable(1)
    for ligne in tableau:
        if len(ligne) > 0:
            for elements in ligne:
                for n in range(len(elements)-longueur_chaine):
                    mot = elements[n:n+longueur_chaine]
                    valeur = table.recherche_valeur(mot)
                    if valeur == None:
                        table.ajouter_valeur(mot, 1)
                    else:
                        table.ajouter_valeur(mot, valeur + 1)

```

```

    # creation du nuage de points
plt.scatter(tab_nb_elements, collision, color = 'k')
plt.ylabel('Collisions')
plt.xlabel("Nombre d'éléments")
return table

def gdNbCollision(collision): #test à effectuer avec le tableau tab_liste_collisions
'''
renvoie la plus grande chaine de collisions
Parameters
-----
collision : tableau d'entiers
'''
return max(collision)

### VARIABLES ###
tableau_etu = import_etudiants()
tableau_livre = import_LesMiserables()
tableau_baudelaire = import_baudelaire()

### CHRONOMETRE ###
starttime = timeit.default_timer()
tab = compteLettresTable(tableau_livre)
#tab = compteMotsTable(tableau_livre, 2)

```