

---

# ANAGRAMMES

## UE4 Base de données avancées - Mini-projet

---

### EXERCICE 1

---

Le script est constitué de 2 parties.

Dans la 1<sup>re</sup> partie, on range tous les mots du dictionnaires dans une liste `stocker`. Dans une cette liste, on prends les anagrammes d'un mot que l'on range dans la liste `K` un mot et dans la liste `V` ses anagrammes. Puis, on prends une nouvelle liste `newT` où les mots sélectionnés sans absent et qui sera utilisé pour choisir les anagrammes. On applique ceci pour chaque mot. À la fin, on obtient deux listes : `K` où se trouve des mots et `V` les anagrammes de ces mots.

Dans la 2<sup>e</sup> partie, on range dans un *data frame* avec le mot témoin en 1<sup>re</sup> colonne avec ses anagrammes en 2<sup>e</sup> colonne. On présente dans un second *data frame* le nombre d'ensemble ayant *i* anagramme.

L'algorithme prends 1mn pour traité seulement 0,67% du dictionnaire. Pour traiter la totalité du dictionnaire, l'algorithme prends 2h48.

On conclut que l'algorithme est très lent et qu'il est nécessaire de l'améliorer.

### EXERCICE 2

---

Pour améliorer la vitesse d'exécution, la première solution était d'utiliser le plus fréquemment possible des fonctions Python.

1<sup>er</sup> : Dans une liste `stocker` où se trouve tous les mots du dictionnaire, on range les lettres de chaque mot dans l'ordre alphabétique que l'on range dans une liste `B`. Pour chaque mot, on créer une liste booléenne que l'on range dans une liste `C`. La liste `C` contient des doublons de liste, on créer une liste `D` où chaque liste est unique. Dans une nouvelle liste `E`, on trouve l'index des `True` pour chaque liste. Enfin, grâce à la liste `E`, on révèle les mots à partir des index pour chaque liste que l'on range dans la liste `F`. La vitesse de ce script est encore très lent.

Une autre piste pour améliorer la vitesse d'exécution, réduire l'utilisation de boucle et la création de liste; penser à utiliser les dictionnaires et recourir plus fréquemment aux fonctions Python.

2<sup>e</sup> : Dans une liste `stocker` où se trouve tous les mots du dictionnaire, on range les lettres de chaque mot dans l'ordre alphabétique que l'on range dans une liste `B`. Ces mots sont rangés dans un dictionnaires `dico` en tant que clé et ont comme valeur une liste vide. Avec la fonction `enumerate`, on range les anagrammes dans la liste correspondant proprement au mot. La vitesse de ce script est très rapide : seulement 0,51 s.