

tp_l8dn003_20_02

February 5, 2020

0.0.1 TP n°2 : recherche des anagrammes

1. On part d'un fichier texte qui est chargé en mémoire sous la forme d'une liste de lignes.
2. A partir de ce fichier, il est d'abord demandé de construire un *lexique* : la liste de tous les mots apparaissant dans le fichier. On peut utiliser la fonction `split()` ou essayer d'améliorer le découpage.
3. Ecrire une fonction booléenne `anagramme(x,y)` qui détermine si les deux mots `x` et `y` sont des anagrammes l'un de l'autre. On peut utiliser la combinaison de fonctions `sorted(list(x))`.
4. Au moyen d'un double parcours du lexique, afficher toutes les paires de mots qui sont des anagrammes.
5. Construire un dictionnaire indiquant, pour chaque mot du lexique, la liste de ses anagrammes

```
[7]: def charge_fichier():
      lgns = []
      with open("demo-file-utf8.txt", "r", encoding="utf8") as f:
          for ligne in f:
              lgns.append(ligne.strip())
      return lgns
```