TP04

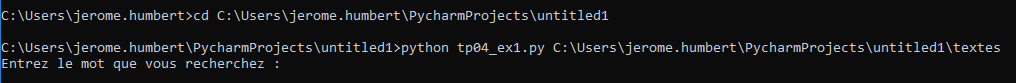
**Consignes impératives !**

Les affichages (sorties) et noms de fonctions doivent respecter exactement l’énoncé. Si une fonction vous est fournie, il ne faut pas modifier son nom ! Si elles ne sont pas fournies mais explicitées dans l’énoncé, créez-les **avec le même nom, et les paramètres dans le même ordre** ! Tout changement de nom ne sera simplement pas pris en compte pas mon script de correction et donc le point ne sera pas accordé. Vous pouvez toujours créer des fonctions supplémentaires.

De plus, le programme doit être exécutable via le terminal de Pycharm en spécifiant le chemin d’accès au répertoire contenant les documents textes. Par exemple :



Ou via la console :



L’objectif de ce TP est de vous faire pratiquer l’utilisation des dictionnaires et affiner vos connaissances des fonctions et des boucles.

1. **Distance des documents**

Pour cet exercice, il vous est fourni quatre fichiers textes et il vous est demandé, au moyen de dictionnaires, de stocker chaque mot d’un texte ainsi que sa fréquence d’apparition dans celui-ci.

Exemple :

« To be or not to be »

Fréquence de « be » vaut 2

Fréquence de « or » vaut 1

Fréquence de « to » vaut 2

Fréquence de « not » vaut 1

Comme vous pouvez le remarquer, il convient de mettre chaque mot en minuscule. De plus, **il ne faut prendre que les caractères utiles** (lettres et lettres avec accents).

Grâce à ces fréquences de mots, nous pourrons calculer la distance du document par rapport à un texte (mot ou phrase) donné par l’utilisateur (Par distance de document nous entendons le document qui se rapproche le plus des données entrées en terme de contenu).

Le calcul de la distance se fait comme ceci :

somme(freq\_mot\_trouvé\_1, freq\_mot\_trouvé\_2, . . .)

nb\_mots\_document

Dans l’exemple précédent :

Si notre document contient « To be or not to be »

Et que l’utilisateur entre « Say hello to Jim or Bryan »

somme(2, 1)

6

Car seuls « to » et « or » sont trouvés dans notre document et ont respectivement des fréquences de 2 et 1. Le nombre de mots de notre document est 6.

La proximité du document contenant « to be or not to be » par rapport au texte « Say hello to Jim or Bryan » est de 0.5

Le programme doit **redemander** une autre recherche tant que l’utilisateur d’appuie pas sur la touche « Enter ».

Vous serez également évalués sur votre découpage en fonction (il faut que celui-ci soit pertinent).

**Exemples de sorties :**

(L’input est trop grand pour être visible mais c’est le texte1.txt entier)

