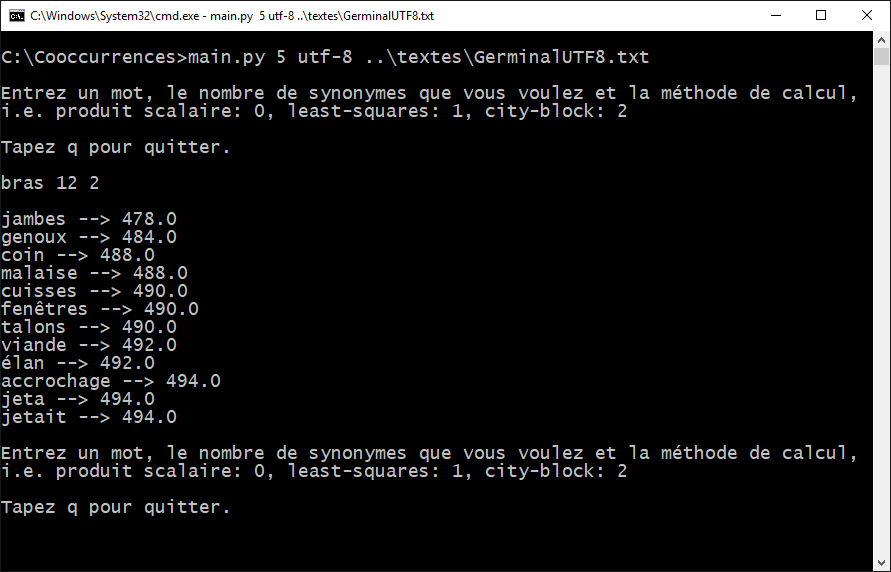
# Travail pratique 1

## Synonymes

Pour le TP1, vous devez créer un « dictionnaire de synonymes ». Votre application devra être en mesure de fournir une liste des n meilleurs synonymes pour un mot que l’utilisateur lui fournira. Vous DEVEZ utiliser la méthode décrite dans le document Cooccurrences.pptx.

#### Type d’application

Ligne de commande, voir l’image suivante:



#### Arguments (respectez l’ordre!)

1. Taille de la fenêtre de cooccurrences.
2. Encodage du texte (ceux que je vous ai fournis sont en utf-8)
3. Chemin du texte à utiliser pour l’entraînement (corpus d’entraînement).

#### Déroulement

Après avoir lancé l’application, l’utilisateur sera invité à fournir un mot pour lequel il veut des synonymes. L’application lui en imprimera. Il devra taper « q » pour quitter, sinon l’application redemande un mot. L’utilisateur doit fournir le mot, le nombre de résultats à afficher et un nombre qui représente la méthode de score à utiliser :

* 0 : produit scalaire
* 1 : least-squares
* 2 : city-block

Ces informations doivent être fournies dans l’ordre décrit et séparées d’une espace.

Vous devez afficher le mot et le score des n meilleurs résultats, en ordre décroissant du meilleur au moins bon, un par ligne. N’oubliez pas de retirer les stop-words de la liste de résultats !

#### Code

Quelques consignes au niveau du code :

* Créez au moins deux classes dans deux fichiers distincts
  + Une pour l’entraînement/généralisation
  + Une pour la recherche/prédiction
* Créez un main() dans un autre fichier distinct.
  + Le fichier peut contenir d’autres fonctions accessoires.
  + La fonction main ne doit pas être trop longue : 20 lignes ou moins.
  + Respectez le paradigme « if \_\_name\_\_ == ‘\_\_main\_\_’ : »

#### Équipes

Vous devez former des équipes de deux et utiliser GIT pour communiquer votre code entre vous.

#### À remettre (livrables)

Seuls les fichiers source (.gitignore…), à remettre via GIT. Votre dépôt GIT devra être nommé NomPrénom1\_NomPrénom2, afin de représenter les co-équipiers/équipières. Votre dépôt DOIT avoir la structure suivante :

* C62
  + TP1
  + TP2
  + TP3

Le TP1 doit donc, évidemment se trouver dans le répertoire TP1. De plus, le travail devra se trouver sur la branche master. Vous devez créer un tag nommé *TP1* pour votre remise. C’est ceci qui sera corrigé. N’oubliez pas que les tags exigent un push explicite.

Astuce :

Entraînez et testez votre système sur de très, très petits corpus au début (une dizaine de mots) et imprimez les états internes pour voir si tout fonctionne.