***TP2***

# *Manipulation*

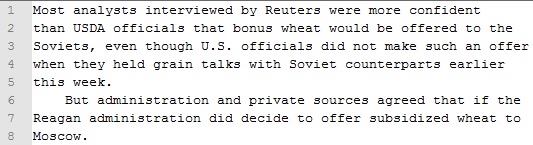
Ecrivez un programme java permettant de réaliser les fonctions de manipulation d’un corpus à savoir :

* Créer un nouveau corpus.
* Mise à jour (Ajouter, supprimer et consulter un document) d’un corpus.
* Supprimer un corpus.
* Rechercher un mot dans un corpus.
* L’interface graphique de ce programme ressemble à celui de la Figure 4.1

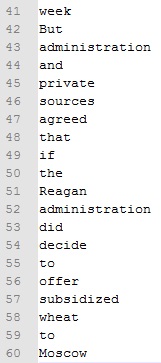
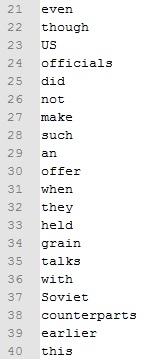
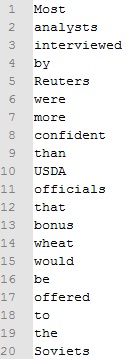


**Figure 1 :** *Interface principale pour la manipulation d’un corpus.*

# *Manipulation2*

Ecrivez un programme java qui permet de générer à partir d’un fichier texte en entrée un autre fichier en sortie qui est composé de tous les termes du fichier en entrée. Le programme doit donner le nombre de mots du fichier en sortie. Un exemple de fichier en entrée et sortie est présenté dans les 5.1 et 5.2.

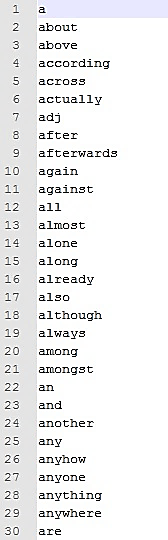
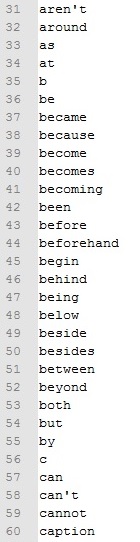
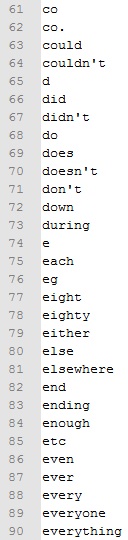
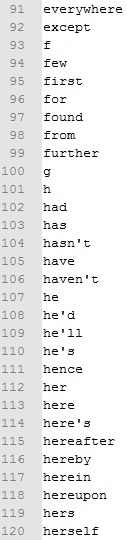
**Figure 2** : *Exemple de fichier en entrée.*



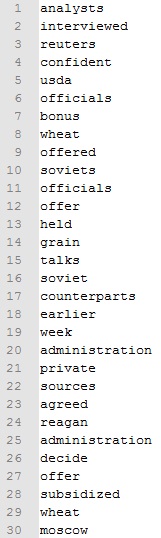
**Figure 3 :** *Exemple de fichier en sortie.*

# *Manipulation3*

Ecrivez un programme java qui permet d’éliminer les mots vides d’un texte se trouvant dans un fichier en entrée, et de générer un fichier ne contenant que la liste des mots non vides en sortie. Le programme doit afficher le nombre de mots du fichier en sortie. Un exemple de fichier généré en sortie, après l’élimination des mots vides du texte se trouvant dans le fichier en entrée de la figure 4, est présenté dans la figure 5.

**Figure 4 :** *Exemple de fichier des mots vides.*



**Figure 5 :** *Exemple de fichier en sortie (sans les mots vides).*

# *Manipulation*

Ecrivez un programme java qui permet à partir d’un corpus de :

* Réaliser l’analyse lexicale.
* Supprimer les mots vides.
* Calculer pour chaque terme son TF et IDF.
* Générer un nouveau fichier contenant tous les termes avec leurs TF/IDF.

Le programme est divisé en trois parties, la première pour calculer TF, la deuxième pour calculer IDF et la troisième pour calculer TF/IDF.