TP 2 : Arbre binaire de recherche

travail à réaliser sur 2 séances (6h)

Les fichiers (pas d'exe, code source uniquement : .c, .h) sont à déposer sur Moodle à la fin de la séance dans la zone de dépôt de votre groupe. Les .zip sont acceptés mais pas les autres types de compressions (pas de .rar ou .tar par exemple).

Vous pouvez travailler en binôme si vous le souhaitez, dans ce cas n'oubliez pas d'indiquer vos 2 noms (vos2noms.zip).

L'objectif est de lister les mots qui apparaissent dans des SMS de type spam. On dispose sur Moodle du fichier "SMS-SpamCollection.txt" qui contient un ensemble de SMS (réels et en anglais) qui sont des spams.

1. Création de l'arbre binaire de recherche pour stocker les mots utilisés et leur occurrence (le nombre de fois où le mot apparaît dans les SMS) :
   (a) Déclarer les types Chaine de 52 caractères (il y a des mots de cette longueur dans les SMS) ; Mot qui est une structure contenant un mot de type Chaine, son occurrence (un entier), les pointeurs sur fils gauche et droit ; et ABR un pointeur de Mot.
   (b) Ecrire une fonction permettant de créer un nouveau Mot contenant un nouveau mot lu.
   (c) Ecrire une fonction qui permet d'ajouter un nouveau mot dans l'arbre s'il n'y est pas encore, ou d'incrémenter son occurrence s'il y est déjà. Les mots sont stockés selon l'ordre lexicographique (l'ordre du dictionnaire).
   (d) Ecrire une fonction permettant d'afficher par ordre lexicographique les mots stockés dans l'arbre ainsi que leur occurrence.
   (e) Tester dans une fonction main, avec quelques mots, en répétant plusieurs fois les mêmes mots pour mettre en évidence le nombre d'occurences.
2. Récupération des données : écrire une fonction permettant de lire les mots dans le fichier et les ajouter à l'arbre à l'aide des fonctions ci-dessus. Il est recommandé de tester d'abord avec un extrait (10 lignes) du fichier seulement.
3. Ce qui nous intéresse, c'est de voir les mots fréquents :
   (a) Ecrire une fonction permettant de rechercher et d'afficher les mots qui apparaissent plus de n fois avec leur occurrence.
   (b) Bonus : améliorer les résultats en créant une fonction *motInutile* qui renvoie vrai si le mot lu est un des mots suivants : "spam", "of", "to", "a" , "an", "at", "the", "and", "in", "is", "it", "on", "2". Utiliser cette fonction pour ne tenir compte que des mots utiles lors de la lecture du fichier.