

TP Arbre non-binaire

L'objectif de ce TP est de créer un programme capable, à la saisie de chaque caractère d'un mot, de proposer des mots possibles sur la base d'un dictionnaire de mots anglais fourni.

Le fonctionnement global proposé est le suivant :

- Lancement du programme avec comme argument le fichier contenant les mots du dictionnaire (afin d'éviter les problèmes d'accents, on utilisera un dictionnaire de mots anglais),
- Demande de saisie d'un mot caractère par caractère. À chaque saisie, le programme proposera une liste de mots possibles. Le caractère # permet de passer à un nouveau mot.

Partie 1 : définition de la structure de données

Vous utiliserez une structure arborescente où :

- Chaque nœud (hormis la racine) est une lettre et comporte au plus autant de fils que de lettres dans l'alphabet,
- La racine ne contient aucune lettre, elle représente uniquement un point d'entrée dans l'arbre.

Représentez graphiquement la structure arborescente proposée et en donner la structure de données.

Vous ne passerez pas à la suite avant d'avoir fait valider cette partie.

Partie 2 : lecture du dictionnaire

Écrire le code permettant de lire un dictionnaire stocké dans un fichier et de construire l'arbre correspondant.

Pour le dictionnaire, vous supposerez que le fichier lu contient exactement un mot par ligne.

Afin de tester votre programme, écrivez une fonction `displayDict(...)` (les arguments seront à déterminer en fonction de votre structure de données) qui affichera la totalité des mots ainsi mis dans la structure.

En utilisant la redirection UNIX (symbole `>` sur la ligne de commande), la commande « `sort` » et la commande « `diff` » vous vérifierez que votre dictionnaire est bien conforme à l'original.

Partie 3 : réalisation du programme de saisie prédictive

Compléter votre programme afin qu'il soit conforme à la description donnée en introduction.