

# DM - Correcteur orthographique – Algo. des arbres : Rapport (partie III)

DM fait par : Prashath SIVAYANAMA et Yann ROUX-DAUGROIS

## Tables des matières :

- I – Présentation et objectifs du Projet
- II – Moyen mis en œuvre et choix de développement
- III – Conclusion

### I – Présentation et objectif du projet

Le rapport traite sur la troisième et dernière partie du projet d'algo des arbres, le sujet est de créer un programme en langage C capable de donner une proposition de correction de mot à partir des mots issue d'un fichier contenant une liste de mots mal orthographiés en prenant en référence un fichier contenant des mots bien orthographié (dico\_2.dico,dico\_1.dico).

La proposition de mot est possible grâce à l'implémentation d'un module «ArbreBK.h », ce module introduit des notions d'arbreBK utile et pratique pour la recherche de mot mal orthographié, de plus la fonction permettant l'insertion dans l'arbreBK utilise une fonction précédemment codé (partie 2 du rendu) pour calculer la distance de Levenshtein entre deux chaînes de caractère.

### II – Moyen mis en œuvre et choix de développement

En ce qui concerne les structures nous avons conservé les mêmes que pour le premiers rendu, c'est-à-dire, une structure pour les listes chaînée (Listes.c et Listes.h) et une structure pour les Arbre Ternaire de Recherche (ATR.c ATR.h) et enfin nous avons ajouté une nouvelle structure de données pour les ArbresBK (ArbreBK.c et ArbreBK.h) .

Pour cette partie III, nous y avons intégré deux fichiers au makefile «ArbreBK.c» et «ArbreBK.h».

En ce qui concerne les nouveautés,nous avons créé un exécutable «correcteur\_2» produit par le Make .

Le fichier «ArbreBK.c» a comme fonction :

La fonction <ArbreBK creer\_BK\_vide(char \* mot)> permet d'allouer en mémoire un arbre BK et lui attribue le mot en paramètre .

La fonction `< int inserer_dans_ArbreBK(ArbreBK * A, char * mot)>` permet d'insérer un mot dans l'arbre par rapport à sa distance de Levenshtein avec sa racine.

La fonction `<int est_dans_ArbreBK(ArbreBK A, char * mot)>` permet de savoir si un mot est présent dans l'arbre .

La fonction `< void liberer_ArbreBK(ArbreBK * A)>` permet de libérer de la mémoire l'ArbreBK.

La fonction `<ArbreBK creer_ArbreBK(FILE * dico)>` permet de créer un arbre en utilisant les mots présents dans le dico et en utilisant la fonction `< int inserer_dans_ArbreBK(ArbreBK * A, char * mot)>`.

La fonction `<Liste cherche_corrections_ArbreBK(ArbreBK A, char * mot, Liste *corrections, int *seuil)>` permet de trouver les mots de correction possible grâce à la distance de Levenshtein .

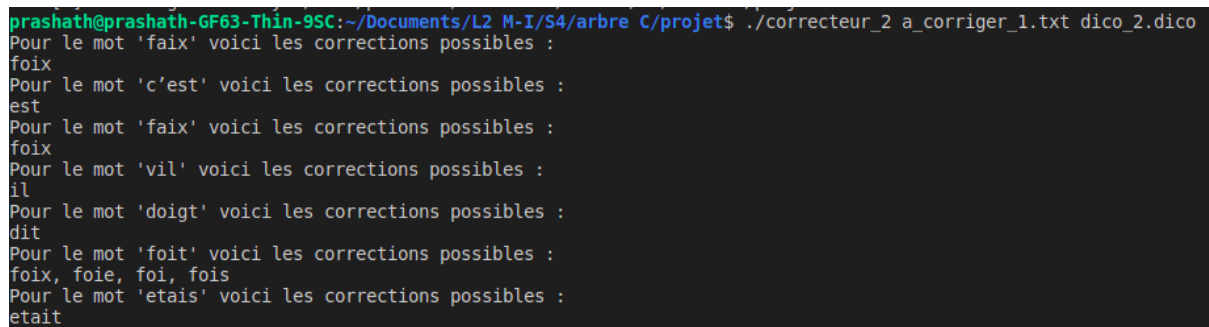
La fonction `<Liste rechercher_dans_ArbreBK(ArbreBK A, char * mot)>` initialise la fonction ci-dessus.

Les fonctions `<void ecrireDebut(FILE *fichier)>`, `<void ecrireArbreBK(FILE *fichier, ArbreBK A)>`, `<void ecrireFin(FILE *fichier)>` et `<void dessine(FILE *fichier, ArbreBK A)>` vont permettre d'écrire dans un fichier dot l'arbre BK nommée `<visualise.dot>` .

La fonction `<void creePDF (char * dot, char * pdf, ArbreBK a)>` vont transformer le fichier dot pour créer un fichier PDF nommé `<visualise.pdf>` .

La fonction `<void afficher_ArbreBK(ArbreBK A)>` va afficher le PDF qui contient l'arbre BK avec tous ces mots .

Voici une image de ce que le programme fait :



```
prashath@prashath-GF63-Thin-9SC:~/Documents/L2 M-I/S4/arbre C/projet$ ./correcteur_2 a_corriger_1.txt dico_2.dico
Pour le mot 'faix' voici les corrections possibles :
foix
Pour le mot 'c'est' voici les corrections possibles :
est
Pour le mot 'faix' voici les corrections possibles :
foix
Pour le mot 'vil' voici les corrections possibles :
il
Pour le mot 'doigt' voici les corrections possibles :
dit
Pour le mot 'foit' voici les corrections possibles :
foix, foie, foi, fois
Pour le mot 'etais' voici les corrections possibles :
etait
```

### III – Conclusion

En ce qui concerne la répartition des tâches, nous sommes réparties les fonctions et nous avons échangé avec discord à plusieurs reprises pour avancer sur le Projet et lorsqu'on rencontrait des problèmes nous essayons de les corriger chacun de notre côté et de s'échanger nos codes sur discord . En soit ce DM nous a permis de consolider nos connaissances en ce qui concerne les arbres BK et à mieux gérer les erreurs qui peuvent être commises dans la mémoire durant la compilation du code .

