

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

DM fait par : Prashath SIVAYANAMA et Yann ROUX-DAUGROIS

## Tables des matières :

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

### I – Présentation et objectif du projet

Le rapport traite sur la seconde partie du projet d'algo. des arbres, l'objectif est donc de créer un code capable de donner une proposition de correction de mot à partir des mots issus d'un fichier contenant une liste de mots mal orthographiés (texte.txt). Cette correction de mot provient d'un fichier contenant des mots bien orthographiés (dictionnaire.dico).

La proposition de mot est possible grâce à la fonction « Levenshtein » qui calcule la distance de Levenshtein entre deux chaînes de caractères.

### II – Moyen mis en œuvre au choix du développement

Nous avons gardé la même structure que pour la première fois, les fonctions sont les mêmes

la seule différence est que nous avons créé deux fonctions en plus < [DistanceDeLevenshtein\(char \\* chaine1, char \\* chaine2\)](#) > cette fonction, nous est donnée sur le lien qui nous mène vers wikipedia, nous

avons aussi <Liste force\_brute(char \*mot, FILE \*texte)> qui est le main donnée dans le sujet à l'étape 2 cette fonction nous renvoie une liste de mot qui peuvent être les corrections du mot mis en paramètre .

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

Nous avons intégré Levenshtein.h au main2.c.

## III – Conclusion

En ce qui concerne la répartition des tâches, nous sommes retrouvés à plusieurs reprises ensemble dans des salles de TP 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 et nous avons beaucoup échangé sur discord afin d'améliorer le code nous

En soit ce DM nous a permis de consolider nos connaissances en ce qui concerne les arbres et à mieux gérer les erreurs qui peuvent être commises dans la mémoire durant la compilation du code .