Rapport du projet de programmation avancé 2018-2019

Réalisation d'un correcteur orthographique

Réalisé par :

RAOUAK Haroun
TD2-TP3-IMA3

Encadré par :

DEQUIDT Jérémie
RUDAMETKIN Walter
FORGET Julien
ARANEGA Vincent

Objectif

L’objectif de ce projet est de réaliser un programme en langage C qui permet de détecter dans un texte en anglais tous les mots mal orthographiés. Pour cela on utilisera un dictionnaire qui sera construit à partir d’un texte ou d’un ensemble de mots de référence.

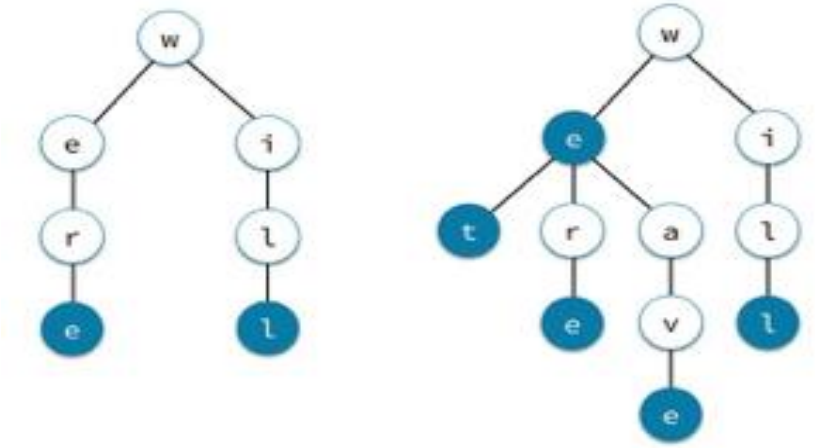
Structure de données

Le dictionnaire qu’on a choisi est une structure basée sur un tableau de 26 arbres préfixes, chaque arbre correspond à une lettre de l’alphabet latin, où laquelle des mots ayant des préfixes communs sont factorisés.

Chaque nœud de l’arbre est un entier équivalent à la lettre en code ASCII, qui peut être terminale ou pas. Aussi, il possède un pointeur vers les 28 nœuds suivants dont 26 alphabets, un pour l’apostrophe et un pour marquer la fin du mot.

Une telle structure arborescente a pour effet de minimiser l’espace mémoire nécessaire au stockage du dictionnaire tout en fournissant un temps de recherche bas.

Considérons les mots : were et will. L’arbre correspondant est affiché ci-dessous. Si on ajoute les mots we, wet et weave l’arbre est affiché à droite, les nœuds colorés représentant des lettres terminales.



Structuration de programme

1. *creer_et_initialiser_le_noeud*

Créer le nœud en allouant dynamiquement la mémoire avec le *malloc,* puis insérer Un entier qui correspond à la lettre et initialiser tous les 28 fils avec NULL.

2. *remplissage*

Elle parcoure le mot lettre par lettre. Si le nœud est vide, elle insère la lettre dans l'arbre en utilisant la fonction *creer_et_initialiser_le_noeud,* sinon, elle passe à la lettre suivante.

*S*i la lettre est une apostrophe, elle l'insère dans la case N°26.

Une fois arrivé à la lettre terminale, on marque la fin du mot en initialisant le fils N°27 par un zéro.

3. *lecture*

Elle lit le fichier texte du dictionnaire mot par mot et remplit chaque mot dans son arbre correspondant jusqu'à la fin du fichier.

4. *insertion_dictionnaire*

Elle permet d'insérer les mots dans les arbres en appelant la fonction lecture 26 fois, pour compléter tous les arbres avec le dictionnaire du référence.

5. *initialiser_dictionnaire*

Elle initialise le dictionnaire de 26 arbres préfixes avec des NULL, pour pouvoir le peupler avec les mots.

6. *existe*

C'est une fonction booléenne qui renvoie 1 si le mot existe dans le dictionnaire et 0 sinon.

7. *corriger_texte*

Elle ouvre le texte à corriger et vérifie orthographiquement les mots à l'aide de la fonction existe, si un mot n'est pas reconnu par le dictionnaire, elle l'affiche dans la fenêtre du Terminal.

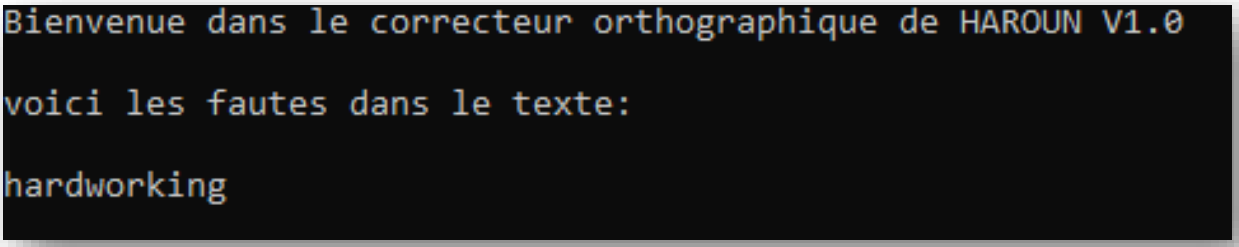
Exemple

Prenons le texte suivant :

« In the old days ants and cicadas were friends They were very different The ants were hardworking but the cicadas were lazy

In the summer the ant families were very busy They knew that in the winter they would have to stay in their anthill They wanted to have enough food for the whole winter »

Et voici le résultat :



Limites de la solution

Apparemment le programme bogue pour un texte accentué et ponctué, aussi pour un dictionnaire de référence plein de mots avec accents.

Donc il faut concevoir un algorithme qui permet de convertir les caractères accentués en leur équivalent non-accentué, et qui enlève les signe de ponctuations et les caractères spéciaux.

Ainsi, il ne libère pas la mémoire allouée comme prévu, d’où les fuites de mémoire.

Conclusion

La réalisation du projet de programmation avancée nous a permis de consolider nos compétences en programmation et de développer notre autonomie et notre savoir-faire.

C’était l'occasion pour mettre en pratique tout ce qu’on a appris en théorie et en ajoutant toutes les connaissances qu'on a eu depuis le début d’année.

On est satisfait, car la vérification des fautes marche correctement et surtout on a pris du temps et du plaisir pour le réaliser malgré les problèmes du débogage qu’on a rencontré pendant la conception et la réalisation du projet.