

# Projet Indexeur

## Table des matières

Partie utilisateur.....	2
Partie développeur.....	3

# Projet Indexeur

## Partie utilisateur

- **Ce que fait le projet**

- Le projet a pour but d'accéder rapidement aux mots du texte ainsi que les phrases qui les contiennent.

- **La compilation**

- Pour lancer toutes les fonctions il faut saisir la commande «gcc Indexeur.c -o Index» suivis de «Index nom\_fichier.txt» (le nom\_fichier fournis est Fichier.txt)  
l'utilisateur aura donc un aperçu des résultats de chaque fonction.

- **L'utilisattion**

- La commande «Index Fichier.txt -a mot» aura pour but de tester si le mot saisie par l'utilisateur est bien présent dans le fichier texte.
- La commande «Index Fichier.txt -p mot» affiche toutes les positions de mot dans le texte.
- La commande «Index Fichier.txt -P mot» affiche toutes les phrases du texte contenant mot.
- La commande «Index Fichier.txt -l» affiche la liste triée des mots du texte.
- La commande «Index Fichier.txt -d mot» affiche l'ensemble des mots du texte ayant pour préfixe mot.
- La commande «Index Fichier.txt -D» enregistre dans un fichier .DICO la liste trié des mots du texte ainsi que leurs positions.

# Projet Indexeur

## Partie développeur

- **Structure du programme**

- Tout d'abord on insère les mots dans une liste chaîné ainsi que leurs positions dans le texte, grâce à la fonction insert.  
Dans cette fonction on réalise une insertion en fin de liste afin d'obtenir les mots dans le même ordre que le texte idem pour la position.  
On utilise plusieurs fonctions afin d'allouer une cellule pour les mots du texte ajoutés à la liste.  
On obtient donc une liste chaîné contenant les mots du texte (1 mot par cellule) ainsi qu'une liste contenant les positions des mots du texte.
- Par la suite chaque option(appartenance de mot dans le texte, position du mot dans le texte...) sera représenté par une fonction afin de faciliter leurs appels en fonction de ce que l'utilisateur entre en argument lors de la compilation.

- **Les fonctions prédéfinis utilisés**

- La fonction strlen() a été notamment utilisé dans les fonctions d'allocation d'une cellule pour un mot afin de connaître la taille des cellules à allouer.
- La fonction strcmp() qui compare 2 chaînes de caractère nous a été utile dans beaucoup de fonctions tel que l'appartenance d'un mot au texte, le préfixe des mots...  
Nous aurions aussi pu l'utilisé pour trié la liste chaîné de mot.

- **Les fonctions**

- insert(...) : prend en paramètre **la liste chaîné de mot** et la **liste de position des mots**.  
Elle ajoute un mot dans la liste de mot et la position dans la liste de position lorsque celle-ci est formé. Chaque mot est donc ajouté l'un après l'autre, l'ordre des mots par rapport au texte est conservé.
- appartenance(...) : prend en paramètre **la liste chaîné de mot** et le **mot** à chercher.  
La fonction parcourt toute la liste de mot et renvoie les mots égales au mot en paramètre.

# Projet Indexeur

## Partie développeur

- affichePosMot(...) : prend en paramètre **la liste des positions** des mots en plus des paramètres de appartenance.  
La fonction part du même principe que appartenance, il y a juste l'affichage qui change.
- Save(...) : prend en paramètre **la liste chaîné de mot** et la **liste des positions des mots**.  
Cette fonction a pour but de sauvegarder les mots du texte suivis de leurs positions dans un autre fichier «fichier.DICO».
- afficheMotPrefixe(...) : prend en paramètre **la liste chaîné de mot** et le **mot** à chercher.  
Cette fonction affiche tous les mots ayant pour préfixe celui donné par l'utilisateur en argument.  
Elle parcourt la liste de mot et compare le préfixe donné a chaque mot du texte(on compare pour le mot seulement le même nombre de caractère que possède le préfixe)  
**ex** : préfixe = 'Bon' alors on ne comparement que les 3 premières lettres de chaque mot du texte.
- AffichePhrase(...) : prend en paramètre 2 liste chaîné qui sont identiques et le mot a rechercher, par la suite il parcourt la liste pour récupérer la position du mot afin de récupérer tout les mots qui ont la meme position en octet dans le texte , les rassemble pour ainsi formé la phrase contenant le mot.
- main(...) : Nous avons fait un switch afin d'exécuter les fonctions nécessaires en fonction des arguments entré par l'utilisateur.  
De plus nous avons fait en sorte que le fichier donnée soit modifier par un filtre afin de faciliter la recherche des mots.(filtre ascii donnée et modifier)  
S'il n'y a pas d'arguments lors de la commande sur le terminal le programme effectue les démonstrations des fonctions sinon par rapport a l'argument il effectue une option.