

# Rapport Projet Indexeur

# Sommaire

## I/ Partie Utilisateur

## II/ Partie Développeur

### I/ Partie Utilisateur

#### 1) Compiler le programme

Le programme compile en écrivant « make » dans le terminal à l'endroit du dossier.

```
mbpdebourennane:indexeur pcbourenn$ make
```



```
mbpdebourennane:indexeur pcbourenn$ make
gcc -Wall -Werror -o src/main.o -c src/main.c -I includes
gcc -Wall -Werror -o src/error.o -c src/error.c -I includes
gcc -Wall -Werror -o src/index.o -c src/index.c -I includes
gcc -Wall -Werror -o src/parsing.o -c src/parsing.c -I includes
gcc -Wall -Werror -o src/ft_base.o -c src/ft_base.c -I includes
gcc -Wall -Werror -o src/lst.o -c src/lst.c -I includes
gcc -Wall -Werror -o src/hachage.o -c src/hachage.c -I includes
gcc -Wall -Werror -o src/fonctionnalite.o -c src/fonctionnalite.c -I includes
gcc -Wall -Werror -o src/menu.o -c src/menu.c -I includes
gcc -Wall -Werror src/main.o src/error.o src/index.o src/parsing.o src/ft_base.o src/lst.o src/hachage.o src/fonctionnalite.o src/menu.o -o Index -I includes
```

#### 2) Utilisation de l'exécutable

Après la compilation on se retrouve donc avec un exécutable Indexeur

```
mbpdebourennane:indexeur pcbourenn$ ls
Gargantua.ascii      Index                Makefile             includes             leDroitALaParesse.ascii  src
```

Il suffit donc maintenant juste de lancer l'une des deux commandes suivantes dans le terminal :

```
./Index
```

- La commande ci-dessus va afficher un menu dans le terminal où il faudra renseigner un nom de fichier avec le suffixe .ascii, une question et si demandé un mot.

```
Entrer le nom de votre fichier (.ascii) :
Gargantua.ascii
Saisir la fonctionnalité parmi celles-ci dessous :
- p : affiche la suite des positions de votre mot dans le texte
- D : sauve dans un fichier la liste triée des mots du texte et leur position dans un
e ce fichier sera constituée d'un mot suivi de la liste de ses positions
- P : affiche les phrases du texte contenant votre mot
- l : affiche la liste triée des mots du texte
- d : affiche l'ensemble des mots du texte ayant votre mot pour préfixe.
- a : teste l'appartenance de votre mot au texte
a
Entrer un mot : par
par appartient au texte.
Voulez-vous recommencer ?(y/n)
NoN
```

- L'autre commande va quant à elle n'exécuter qu'une seule fonctionnalité demandé par l'utilisateur

```
mbpdebourennane:indexeur pcbourenn$ ./Index -a par Gargantua.ascii  
par appartient au texte.  
mbpdebourennane:indexeur pcbourenn$ █
```

### 3) Différentes options

Il y a au total 6 options dont 4 qui utilise un mot :

-a qui dit si le mot appartient au texte

-p qui affiche les position du mot dans le texte

```
mbpdebourennane:indexeur pcbourenn$ ./Index -p par Gargantua.ascii  
Voici les positions du mot par : 1924 2174 2886 3285 4719 5625 6094 6  
26267 26762 27873 27873 30127 30401 31137 31923 32359 32800 32800 336  
1 41751 42521 44110 44557 45353 45633 46154 46355 46355 46355 48086 4  
905 55302 55302 55302 56104 56433 56765 57072 57297 60505 61888 62281  
79935 80130 80422 80941 81299 81299 81695 82978 83243 83501 83501 8374
```

-P qui affiche les phrases où le mot apparaît dans le texte

```
mbpdebourennane:indexeur pcbourenn$ ./Index -P par Gargantua.ascii  
par ceste raison, quand le temps venu estoit que aulcun d' icelle abbaye, ou a la requeste de ses paren  
laquelle l' auroit prins pour son devot, et estoient ensemble mariez; et, si bien avoient vescu a thele  
moient ilz a la fin de leurs jours comme le premier de leurs nopces.  
s' il est permis de croyre fermement que par les corps qui sont au firmament humain esprit de soy puiss  
r avoir la congnoissance, tant que l' on juge en asseure discours des ans loingtains la destinee et cou  
e plus tost, en ce lieu ou nous sommes il sortira une maniere d' hommes las du repoz et faschez du sejo  
partialitez.  
s' il est permis de croyre fermement que par les corps qui sont au firmament humain esprit de soy puiss  
r avoir la congnoissance, tant que l' on juge en asseure discours des ans loingtains la destinee et cou  
e plus tost, en ce lieu ou nous sommes il sortira une maniere d' hommes las du repoz et faschez du sejo  
partialitez.  
lors se verra maint homme de valeur, par l' esguillon de jeunesse et chaleur et croire trop ce fervent  
et ne pourra nul laisser cest ouvrage, si une fois il y met le courage, qu' il n' ayt emply par noises  
reste, en apres ces accidens parfaictz, que les esleuz joyeusement refaictz soient de tous biens et de  
oient denuez.  
la lecture de cestuy monument parachevee, gargantua souspira profondement, et dist es assistans" ce n'  
heureux est celluy qui ne sera scandalize et qui tousjours tendra au but, au blanc que dieu, par son ch  
la lecture de cestuy monument parachevee, gargantua souspira profondement, et dist es assistans" ce n'  
heureux est celluy qui ne sera scandalize et qui tousjours tendra au but, au blanc que dieu, par son ch  
"le moyne dist" que pensez vous, en vostre entendement, estre par cest enigme designe et signifie?  
-par saint goderan( dist le moyne), telle n' est mon exposition; le stille est de merlin le prophete.
```

-d qui affiche tous les mots contenant le préfixe mot

```
mbpdebourennane:indexeur pcbourenn$ ./Index -d par Gargantua.ascii  
Voici les mots qui ont pour prefixe par :  
parents  
parabolis  
parlons  
parpaillos  
parler  
parlez
```

Et 2 autres qui n'utilise pas de mot :

-D qui enregistre la liste triée des mots avec leur position dans un .DICO

```
mbpdebourennane:indexeur pcbourenn$ ./Index -D Gargantua.ascii  
mbpdebourennane:indexeur pcbourenn$ ls  
Gargantua.ascii      Gargantua.ascii.DICO  Index
```

-l qui affiche la liste triée des mots

```
mbpdebourennane:indexeur pcbourenn$ ./Index -l Gargantua.ascii  
Les mots commençants par y  
y  
ycelle  
ycelles  
ycelluy  
yceulx  
ycy  
yeulx  
yeux  
yra  
yssir  
yssit  
yssoient  
yssoit  
yssu  
yssue  
yssues  
yvres  
  
Les mots commençants par z  
zaphran  
zelateur  
zencle  
zhtei
```

## II/ Partie Développeur

### 1) Les listes

Nous avons plusieurs fonctions pour les listes, une pour allouer une nouvelle cellule, une autre pour ajouter tout en triant par ordre croissant ou lexicographique et une autre pour libérer la mémoire utilisé par celles-ci. Vu que le tri est fait lors de l'ajout il n'y a pas besoin de faire un réel algorithme de trie, on ajoute l'élément au bon endroit directement, pour les positions on place les plus proches du début du fichier d'abord puis plus on s'éloigne de la tête de liste plus la position y est grande. Pour les mots, on a choisi de les placer par ordre lexicographique pour qu'elle soit trié par ordre alphabétique.

### 2) La table de hachage

Nous avons décidé de mettre un tableau dynamique pour la table de hachage pour ne pas être obligé de changer la valeur de N directement dans le fichier header à chaque fois que l'on passe d'un fichier conséquent à un tout petit. Il s'adapte donc au mieux, en comptant le nombre de mot différent dans le texte et en multipliant par 3 ce nombre pour faire un parfait compromis entre proportion de liste vide dans la table et l'écart-type des tailles des listes de la table, on a 2 fonction commenté dans le code qui calcule l'écart-type des longueurs des listes ainsi qu'une qui calcule un pourcentage des listes vide.

### 3) Le traitement

Après la lecture du fichier, la création des listes et la création de la table de hachage, on traite les information récupérer et selon les options données on sépare le traitement pour qu'il soit plus rapide en regroupant les options qui se ressemblent ou qui ont le même commencement tel que -a et -p. On enregistre dans un fichier .DICO tout simplement en parcourant notre liste de mots déjà triées.