

## **Rapport projet Algorithmique 2 :**

### **Objectifs :**

Dans ce projet d'algorithmique nous devons utiliser le langage C pour générer un texte aléatoire à partir d'un texte donné, ainsi d'afficher le tableau des successeurs de chaque mots du texte.

### **Modules du projets :**

Module hashtable : qui contient hashtable.c et hashtable.h qui contiennent des fonctions pour gérer les tables de hachage.

Module holdall : qui contient holdall.h et holdall.c qui gère le holdall ( four tout ) qui va contenir l'adresse de nos élément pour les quelle on va allouer de la place dans nos listes.

Module list : qui contient list.h et list .c qui gère nos différentes listes qui s'attacheraient à notre table de hachage

Module main : dans lequel y a le main.c où on met en marche l'algorithme donné en utilisant toute les fonctions créer dans les différents modules et le makefile où on gère l'exécution de notre main

Makefile : qui gère la compression en tar.gz

Module Textes : où on a à notre disposition des fichiers textes à utiliser pour tester notre programme.

### **Structures des données :**

J'ai utilisé cette structure: parce que le head va nous aider pour tout ce qui est insertion des éléments et length la longueur de ma liste pour la génération aléatoire

```
struct list {  
    clist *head;  
    size_t length;  
};
```

et pour les cellules de ma liste j'ai utilisé la structure suivante :

```
struct clist {  
    void *value;  
    clist *next;  
};
```

### **Description des modules :**

Tout mes modules je les ai récupéré des programmes tp de monsieur Hancart par contre le module list était fait par mes soins où j'ai mis le list.h où y a toute mes fonctions et la déclaration de ma structure, que j'ai définie dans le list.c

```
list *list_empty(void); //qui va me créer une liste vide  
bool list_is_empty(const list *s); // qui teste si ma liste est vide  
size_t list_length(const list *s); //qui retourne la taille de ma liste
```

```

int list_insert_head(list *s, void *ptr);//qui fait une insertion en tête des éléments à ma liste
void list_dispose(list **sptr);// qui libère les éléments déjà utilisé par la récupération de leur adresse
dans le holdall
void *list_n_elem(list *s, size_t idx);//qui crée une liste de n éléments pour les clés
int list_compar(list *s1, list *s2);//la fonction de comparaison des listes à utiliser pour la fonction
hahstable_empty

```

### Résultats du teste :

Après le teste avec time valgrind et le text des misérables de victors hugo en utilisant la formule :  
« time valgrind ./main < ../textes/lesmiserables.txt »

```

Tome I -- Fantine Livre premier -- Paris etudie dans son jardin. Pres de la cour
, si l'identite de la poupee et ce que tu as, trois que me disent sans cesse aut
our du cou, lui tombait jusqu'aux pieds. Il revint sur ses plates-bandes fletrie
s. Il pleuvait par instants, si fort et s'indigner bien vite leur tete sous leur
maitre-autel pour la regarder et ne sait qui, un voleur, il avait fait place au
pensionnat et servaient a y tomber, etendit avec sa robe couleur de gens se cro
ient au-dessus de laquelle etait Jean Valjean. Marius ne lui etait apparu

```

ça affiche la table de hachage et le texte généré aléatoirement ;

### Problème rencontré :

De se rappeler qu'il faut tester toute les fonctions où y a une allocations ou une dés-allocation ainsi que le teste de dépassement de capacité ainsi que de se retrouver dans ton projet si tu passes une nuit loin de luit et il te faut reprendre tout depuis le début et de redessiner des schémas pour s'y retrouver .