# Travaux dirigés 2

Tutorat IMAC

*Lexique des mots d'un fichier*

Dans toute la suite, on appelle mot une suite de caractères ne contenant pas d'espace, de passage à la ligne ou de tabulation, c'est à dire une chaine de caractères lue avec le format `%s` par `scanf`.On pourra considérer que les mots font moins de 500 caractères et utiliser le format `%500s` pour lire un mot dans un tableau de 501 char.
On veut écrire un programme qui construit la suite triée des mots d'un fichier et crée un nouveau fichier contenant la suite de ces mots et de leur nombre d'occurences.
Le fichier de sortie contient un mot par ligne et les mots apparaissent en ordre lexicographique.
La référence du fichier dans lequel on lit les mots sera fournie sur la ligne de commande, la référence du fichier dans lequel on écrit sera celle du fichier dans lequel on lit, suffixée par ".cpt". Ainsi si l'exécutable a pour nom `Compte`, l'éxécution de la ligne `Compte /home/zip/Exemple` entrainera la création du fichier
`/home/zip/Exemple.cpt`.
Si le contenu du fichier `/home/zip/Exemple` est :

```
exemple d'informatique tres tres facile.
```

celui du fichier `/home/zip/Examen.cpt` sera :

```
d'informatique  1
exemple 1
facile. 1
tres 2
```

Le nombre de mots étant inconnu avant traitement on utilisera une structure allouée dynamiquement :

```
typedef struct{
int occurence;
char *mot;
}Mot;

typedef struct{
int taille;
int nombre;
Mot* Dico;
}Lexique;
```

où `taille` est le nombre d'emplacements réservés à partir de l'adresse `Dico` et `nombre` est le nombre de `Mots` présents. Lorsque tous les emplacements sont utilisés, on doublera la taille du lexique. Le lexique sera initialisé avec un taille de 100.
Les mots seront ajoutés en ordre lexical.

1. Décrire l'organisation de votre programme :
   décomposition en fonctions et rôle de chaque fonction
2. Ecrire le programme.
3. Ecrire un programme permettant de générer un fichier contenant un nombre de mots choisit par l'utilisateur. Les mots seront générés aléatoirement avec les fonctions

```
 #include <time.h>
#include <stdlib.h>

      long int random(void);

      void srandom(unsigned int seed);


      time_t time(time_t *t);
```

Le programme recevra ses arguments sur la ligne de commande. Si l'executable a pour nom Genere, la commande `Genere 5321 test` entraînera la création d'un fichier Test contenat 5321 mots.