



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Alger 1 – BENYOUCEF Benkhedda

Faculté des Sciences

Département Mathématiques-Informatique

(Enseignant : Mr Sadou)

Rapport

Thème

Indexer un fichier texte

Réalisé par :

Laieb Meriem

Section : B

Groupe : 1

Année universitaire : 2022/2023

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
```

```
/* Nom : Laieb
   Prénom : Meriem
   Section : B
   Groupe : 1 */
```

```
/* le but est de calculer la taille de tout les mots du fichier f et puis les insérer dans un
   abr d'une manière de mettre
```

```
le premier mot comme une racine et comparer sa taille avec la taille du mot suivant */
```

```
// declarer une liste pour mettre les mots du fichier
```

```
typedef struct maillon{
    char mot[50];
    struct maillon *suiv;
};
```

```
/* créer une liste chaînée pour mettre le nombre de la phrase ou il se trouve le mot */
```

```
typedef struct plist{
    int occ;
```

```
        struct plist *suiv;  
    }plist;
```

```
/* déclaration d'un arbre tel que la partie info de l'arbre contient un champ pour les  
mots et          elle pointe sur une liste chaînée qui va contenir le numero des  
phrases */
```

```
typedef struct Arbre{  
    char info[50];  
    struct plist h;  
    struct Arbre * SAG;  
    struct Arbre * SAD;  
} Arbre;
```

```
typedef struct Arbre *abr;
```

```
// une procedure pour mettre le contenu du fichier dans la liste chaînée  
void remplir(struct maillon **L,char car[50]){  
    struct maillon *nouv=malloc(sizeof(struct maillon));  
    struct maillon *der;  
    strcpy(nouv->mot,car);  
    nouv->suiv=NULL;  
    if (*L==NULL){
```

```

        *L=nouv;
    }else{
        struct maillon *der=*L;
        while (der->suiv != NULL){
            der=der->suiv;
        }
        der->suiv = nouv;
    }
}

```

// une fonction pour calculer la taille de chaque mot du fichier pour insérer ces mots dans l'arbre binaire de recherche

```

int size(struct maillon *L,int n,char m[50]){
    while (L!=NULL){
        n=strlen(m);
    }
    return(n);
}

```

//remplir l'abr

```

abr ajouter(struct abr *R,char w[50]){
    int cpt,cpt2;
    struct maillon *L;
    cpt=size(L,cpt,w);
    if(R==NULL){
        abr * R=(abr *)malloc(sizeof(abr));
    }
}

```

```

        R->info=w;
        R->SAG=NULL;
        R->SAD=NULL;
    }
    else{
        cpt2=size(L,cpt2,R->info)
        if(cpt<cpt2){
            R->SAG = ajouter(R->SAG,);
        }else{
            R->SAD = ajouter(R->SAD,);
        }
    }
    return R;
}

int main(){
    struct maillon *list=NULL;
    FILE *f;

    // création du fichier f
    f=fopen("projet_asd.txt","w");
    if (f==NULL){
        printf("echec d'ouverture");
        return (1);
    }

    fputs("Le but du projet d'asd est d'indexer un fichier .\nLe projet doit etre remis
le 5 janvier .\nLe projet d'asd sera compter comme un test .",f);

    fclose(f);

```

```

f=fopen("projet_asd.txt","r");
if (f=NULL){
    printf("erreur de l'ouverture");
    return (1);
}
char c;
int taille;
struct *Tree;
while ((c=fgetc(f)) != EOF){
    while(c!=' ' && c!='.' && c!='\n'){
        c=fgetc(f);
    }
    remplir(list,c);
    taille = size(list,taille,c);
    *Tree=ajouter(Tree,c);
}
system("pause");
}

```