

RAPPORT DE MINI PROJET

Deuxième Année Informatique
Anne universitaire 2024-2023
Module ASD 3
Group 02

Réalise par :

- BABAUSMAIL HADJ KACEM

Sous la supervision du professeur :

- BADAOU IATIKA

Le contenu du rapport :

- Introduction
- Analyse
- Algorithme
- Programme
- Conclusion

Introduction :

Les bibliothèques sont considérées comme l'une des sources d'information les plus importantes pour les chercheurs, les étudiants et les personnes intéressées par divers sujets. elles contiennent de nombreux dictionnaires, qui sont des références importantes pour comprendre et maîtriser le vocabulaire des différentes langues.

Analyse :

Les variables globales :

```
struct dictionnaires N[5];  
int i,j;  
char mot[30],imot [30];
```

- **Dictionnaire** : c'est un type d'enregistrement, il contient le mot et sa traduction (Fr / En).

```
struct dictionnaires  
{  
    char fr [20];  
    char en [20];  
};
```

- **mot et imot** : de type chaîne de caractère, sont les variables qui contiendront les mots de dictionnaire.
- **i et j** : de type entier, sont les variables compteur qui sont utilisées pour parcourir le tableau.

```
int choix;
```

- **Choix** : de type entier, pour le choix sur le menu de programme

Les fonctions utilisées :

- **Fonction 1** : fonction qui convertit tous les éléments d'une chaîne en minuscules, en utilisant la méthode de code ascii, qui l'intervalle des majuscules [65,90] et les minuscules [97,122], donc la différence entre les deux ($a=97$) - ($A=65$) = 32
Exemple : $66+32=98 \rightarrow$ en code ascii est (66=B) (98=b).

```
void minuscule (){
    int i = 0;
    while (mot[i] != mot[(strlen(mot) +1)])
    {
        if (mot[i] >= 'A' && mot[i] <= 'Z')
        {
            mot[i] =mot[i]+ 32;
        }
        i++;
    }
}
```

- **Fonction 2** : qui remplit le tableau, et divise le mot en deux parties français et anglais,
Exemple : salut hello

FR	EN
salut	hello

```
void remplissage (struct dictionnaires N[5]){
    printf("\n-----INSUALISATION LES MOTS POUR LE DICTIONNAIR-----\n");
    printf("-----<ATTENTION AVANT ENTER UN MOT !! >-----\n");
    printf("\n- Premier mot en français ( espace ) deuxième mot leur
traduction en anglais \n");
    printf("- |EXAMPLE| --> ( | salut | ('espace') | hello | ) \n\n");
    for ( i = 0; i <5; i++)
    {
        printf("le (%d) mot : ",i+1);
        gets(mot);
        minuscule ();
        j=0;
        while ((j < strlen(mot)) && (mot[j] != ' '))
        {
            j++;
        }
        strncpy(((N[i]).fr), mot, j);
        strncpy(((N[i]).en), (mot + j+1 ), strlen(mot));
    }
}
```

- **Fonction 3 et 4 :** Qui tri le tableau en (Fr / En).

Exemple : tri seulement le champ français ou anglais

FR		FR		FR		FR		FR	
	EN		EN		EN		EN		EN

- **Fonction 5 :** qui traduit le mot, on utilis  la m thode de recherche s quentielle, on sait que chaque mot dans le tableau est ce forme deux champ (Fr / En) donc quand la fonction trouver le mot dans le tableaux la traduction est juste a cote de l'autre.

Exemple :

FR			EN	FR		EN
ciel	sky			filles	girl	

```
void Traduiremot (struct dictionnaires N[5]){
    bool trouve;
    trouve=false;
    i=0;
    printf("\n----- < dictionnaires > -----\\n");
    printf ("donner un mot pour traduer :");
    scanf("%s",imot);
    while ((!trouve) && (i<5))
    {
        if (((strcmp(imot,(N[i].fr)))==0 )|| ((strcmp(imot,(N[i].en)))==0))
        {
            trouve =true;
        } i++;
    }
    if ((trouve) && ((strcmp(imot,N[i-1].fr))==0))
    {
        printf("\\nle mot : %s en anglais est : %s\\n\\n\\n",imot,N[i-1].en);
    }else if ((trouve) && ((strcmp(imot,N[i-1].en))==0))
    {
        printf("\\nle mot : %s en francais est : %s\\n\\n\\n",imot,N[i-1].fr);
    } else
    {
        printf("Le mot n'existe pas");
    }
}
```

- **Fonction 6 et 7 : Qui afficher le tableau en (Fr / En).**
Exemple : afficher seulement le champ français ou anglais

FR		FR		FR		FR		FR	

	EN		EN		EN		EN		EN

Programme :

```

-----INSUALISATION LES MOTS POUR LE DICTIONNAIRE-----
<ATTENTION AVANT ENTER UN MOT !! >

- Premier mot en français ( espace ) deuxième mot leur traduction en anglais
- |EXAMPLE| --> ( | salut | ('espace') | hello | )

warning: this program uses gets(), which is unsafe.
le (1) mot : bonjour HELLO
le (2) mot : MERCI thanks
le (3) mot : belle beautiful
le (4) mot : bibliotheque library
le (5) mot : ecole school

----- Menu de choix -----
1- Pour afficher les mots en français tri [ 1 ]
2- Pour afficher les mots en anglais tri [ 2 ]
3- Pour traduire un mots [ 3 ]
- Mon choix est : 1

-----les mots en français-----
< tri >
- belle
- bibliotheque
- bonjour
- ecole
- merci
o macbook@MBP-de-Macbook totural %

```

Afficher uniquement le champ FR

FR	EN
"mot"	

convertir le mot en minuscules

Les mots sont classés par ordre alphabétique

```

-----INSUALISATION LES MOTS POUR LE DICTIONNAIRE-----
<ATTENTION AVANT ENTER UN MOT !! >

- Premier mot en français ( espace ) deuxième mot leur traduction en anglais
- |EXAMPLE| --> ( | salut | ('espace') | hello | )

warning: this program uses gets(), which is unsafe.
le (1) mot : ciel sky
le (2) mot : dormir sleep
le (3) mot : FILLE GIRL
le (4) mot : bonjour hello
le (5) mot : arbre TREE

----- Menu de choix -----
1- Pour afficher les mots en français tri [ 1 ]
2- Pour afficher les mots en anglais tri [ 2 ]
3- Pour traduire un mots [ 3 ]
- Mon choix est : 2

-----les mots en anglais-----
< tri >
- girl
- hello
- sky
- sleep
- tree
o macbook@MBP-de-Macbook totural %

```

Afficher uniquement le champ EN

FR	EN
	"mot"

convertir le mot en minuscules

Les mots sont classés par ordre alphabétique

-----INSUALISATION LES MOTS POUR LE DICTIONNAIRE-----
-----<ATTENTION AVANT ENTER UN MOT !! >-----

- Premier mot en français (espace) deuxième mot leur traduction en anglais
- |EXAMPLE| --> (| salut | ('espace') | hello |)

warning: this program uses gets(), which is unsafe.

le (1) mot : bonjour hello
le (2) mot : arbre tree
le (3) mot : ciel sky
le (4) mot : dormir sleep
le (5) mot : fille girl

Traduire un mot français en anglais

----- Menu de choix -----

1- Pour afficher les mots en français tri [1]
2- Pour afficher les mots en anglais tri [2]
3- Pour traduire un mots [3]
- Mon choix est : 3

----- < dictionnaires > -----

donner un mot pour traduer :ciel

le mot : ciel en anglais est : sky

ciel	sky
------	-----

traduction

○ macbook@MBP-de-Macbook totural %

-----INSUALISATION LES MOTS POUR LE DICTIONNAIRE-----
-----<ATTENTION AVANT ENTER UN MOT !! >-----

- Premier mot en français (espace) deuxième mot leur traduction en anglais
- |EXAMPLE| --> (| salut | ('espace') | hello |)

warning: this program uses gets(), which is unsafe.

le (1) mot : bonjour hello
le (2) mot : arbre tree
le (3) mot : ciel sky
le (4) mot : dormir sleep
le (5) mot : fille girl

Traduire un mot anglais en français

----- Menu de choix -----

1- Pour afficher les mots en français tri [1]
2- Pour afficher les mots en anglais tri [2]
3- Pour traduire un mots [3]
- Mon choix est : 3

----- < dictionnaires > -----

donner un mot pour traduer :girl

le mot : girl en francais est : fille

fille	gir
-------	-----

traduction

○ macbook@MBP-de-Macbook totural %

Algorithme : (programmation par langage C)

```
2  #include<stdlib.h>
3  #include<stdbool.h>
4  #include<string.h>
5  struct dictionnaires
6  {
7      char fr [20];
8      char en [20];
9  };
10
11  struct dictionnaires N[5];
12  int i,j;
13  char mot[30],imot [30];
14  void minuscule (){
15      int i = 0;
16      while (mot[i] != mot[(strlen(mot) +1)])
17      {
18          if (mot[i] >= 'A' && mot[i] <= 'Z')
19          {
20              mot[i] =mot[i]+ 32;
21              i++;
22          }
23      }
24  }
25  void remplissage (struct dictionnaires N[5]){
26      printf("\n-----INSUALISATION LES MOTS POUR LE DICTIONNAIRE-----\n");
27      printf("-----<ATTENTION AVANT ENTER UN MOT !! >-----\n");
28      printf("\n- Premier mot en français ( espace ) deuxième mot leur traduction en anglais \n");
29      printf("- |EXAMPLE| --> ( | salut | ('espace') | hello | ) \n\n");
30      for ( i = 0; i <5; i++)
31      {
32          printf("le (%d) mot : ",i+1);
33          gets(mot);
34          minuscule ();
35          j=0;
36          while ((j < strlen(mot)) && (mot[j] != ' '))
37          {
38              j++;
39          }
40          strncpy((N[i].fr), mot, j);
41          strncpy((N[i].en), (mot + j+1 ), strlen(mot));
42      }
43  }
```

```
42  void TriselectionFR(struct dictionnaires N[5]) {
43      int x;
44      struct dictionnaires temp;
45      for (i = 0; i < 4; i++) {
46          x = i;
47          for (j = i + 1; j < 5; j++) {
48              if (strcmp(N[j].fr, N[x].fr) < 0) {
49                  x = j;
50              }
51          }
52          temp = N[i];
53          N[i] = N[x];
54          N[x] = temp;
55      }
56  }
57  void TriselectionEN(struct dictionnaires N[5]) {
58      int x;
59      struct dictionnaires temp;
60      for (i = 0; i < 4; i++) {
61          x = i;
62          for (int j = i + 1; j < 5; j++) {
63              if (strcmp(N[j].en, N[x].en) < 0) {
64                  x = j;
65              }
66          }
67          temp = N[i];
68          N[i] = N[x];
69          N[x] = temp;
70      }
71  }
```

```

72 void Traduiremot (struct dictionnaires N[5]){
73     bool trouve;
74     trouve=false;
75     i=0;
76     printf("\n----- < dictionnaires > ----- \n");
77     printf ("donner un mot pour traduer :");
78     scanf("%s",imot);
79     while ((!trouve) && (i<5))
80     {
81         if (((strcmp(imot,(N[i].fr))==0) || ((strcmp(imot,(N[i].en))==0))
82         {
83             trouve =true;
84             i++;
85         }
86         if ((trouve) && ((strcmp(imot,N[i-1].fr))==0))
87         {
88             printf("\nle mot : %s en anglais est : %s\n\n",imot,N[i-1].en);
89             }else if ((trouve) && ((strcmp(imot,N[i-1].en))==0))
90             {
91                 printf("\nle mot : %s en francais est : %s\n\n",imot,N[i-1].fr);
92             } else
93             {
94                 printf("Le mot n'existe pas");
95             }
96         }
97     void affichageFR (struct dictionnaires N[5]){
98         printf("\n-----les mots en français----- \n");
99         printf("                < tri >                ");
100         for ( i = 0; i <5; i++)
101         {
102             printf("\n- %s ",(N[i].fr));
103         }
104     }
105     void affichageEN (struct dictionnaires N[5]){
106         printf("\n-----les mots en anglais----- \n");
107         printf("                < tri >                ");
108         for ( i = 0; i <5; i++)
109         {
110             printf("\n- %s ",(N[i].en));
111         }

```

```

113 int main() {
114     int choix;
115     remplissage (N);
116     printf("\n----- Menu de choix ----- \n");
117     printf("\n1- Pour afficher les mots en français tri [ 1 ]");
118     printf("\n2- Pour afficher les mots en anglais tri [ 2 ]");
119     printf("\n3- Pour traduire un mots [ 3 ]");
120     printf("\n - Mon choix est : ");
121     scanf("%d",&choix);
122     switch (choix)
123     {
124     case 1 :
125         TriselectionFR(N);
126         affichageFR(N);
127         break;
128     case 2:
129         TriselectionEN(N);
130         affichageEN(N);
131         break;
132     case 3:
133         Traduiremot(N);
134         break;
135     }
136     return 0;
137 }
138
139

```


Conclusion :

A la fin l'importance du stockage et de l'organisation des données dans la programmation est bénéfique pour l'utilisateur.

le stockage et l'organisation des données contribuent à améliorer les performances des applications, en réduisant le temps et les efforts pour accéder et traiter les données, donc le stockage et l'organisation des données dans la programmation sont une question importante qui affecte l'expérience de l'utilisateur.