



PROGRAMMATION PROCEDURALE - Année 2019/2020

CLASSE CYCLE D'INGENIEUR ING1G04

INTRODUCTION AU C :

TP8

Made by :
Lilian Naretto

```
1  #include "tp8.h"
2  /*!\author Lilian Naretto <narettolil@eisti.eu>
3  \date 30 novembre 2019
4  \file tp8.c
5  \brief tp8
6  \version 0.1 premier jet*/
7
8  void error(){
9      printf("Mauvais agrument\n");
10     printf("liste des arguments :\n");
11     printf("Cesar\n");
12     printf("Vigenere\n");
13     printf("Scytale\n");
14 }
15
16 void error2(){
17     printf("mot incorrecte\n");
18 }
19
20 void cesar(){
21     int dec;
22     char* mot=NULL;
23     char* mot2;
24     int taille;
25     size_t taille2;
26     printf("donne un mot : ");
27     taille=getline(&mot,&taille2,stdin);
28     printf("donne le decalage : ");
29     scanf("%d",&dec);
30     mot2=malloc((taille+1)*sizeof(char));
31     for (int i = 0; i < taille; i++)
32     {
33         if (mot[i]>=65 && mot[i]<=90)
34         {
35             mot2[i]=mot[i]-65;
36             mot2[i]=mot2[i]+dec;
37             mot2[i]=mot2[i]%26;
38             mot2[i]=mot2[i]+65;
39         }
40         else if (mot[i]>=97 && mot[i]<=122)
41         {
42             mot2[i]=mot[i]-97;
43             mot2[i]=mot2[i]+dec;
44             mot2[i]=mot2[i]%26;
45             mot2[i]=mot2[i]+97;
46         }
47         else
48         {
49             mot2[i]=mot[i];
```

```
49         mot2[i]=mot[i];
50     }
51
52     }
53     mot2[taille]='\0';
54     printf("mot codé : %s",mot2);
55     printf("\n");
56 }
57
58 void vigenere(){
59     char* mot=NULL;
60     char* mot2;
61     char* mot3=NULL;
62     char* mot4;
63     int taille,taille3;
64     size_t taille2;
65     size_t taille4;
66     printf("donne un mot : ");
67     taille=getline(&mot,&taille2,stdin);
68     printf("donne la clé : ");
69     taille3=getline(&mot3,&taille4,stdin);
70     mot2=malloc((taille+1)*sizeof(char));
71     mot4=malloc((taille+1)*sizeof(char));
72     for (int k=0,i=0;k<taille-1; k++,i++)
73     {
74         mot4[k]=mot3[(i%(taille3-1))];
75     }
76     mot4[taille-1]='\0';
77     for (int i = 0; i < taille; i++)
78     {
79         if (mot[i]>=65 && mot[i]<=90)
80         {
81             mot2[i]=mot[i]-65;
82             mot2[i]=mot2[i]+(tolower(mot4[i])-97);
83             mot2[i]=mot2[i]%26;
84             mot2[i]=mot2[i]+65;
85         }
86         else if (mot[i]>=97 && mot[i]<=122)
87         {
88             mot2[i]=mot[i]-97;
89             mot2[i]=mot2[i]+(tolower(mot4[i])-97);
90             mot2[i]=mot2[i]%26;
91             mot2[i]=mot2[i]+97;
92         }
93     }
94     else
95     {
96
```

```
96     {
97         mot2[i]=mot[i];
98     }
99 }
100 mot2[taille]='\0';
101 printf("mot codé : %s",mot2);
102 printf("\n");
103
104
105 }
106
107 void scytale(){
108     char* mot=NULL;
109     size_t taille;
110     char** tab;
111     int taille2,taille3;
112     int accum=0;
113     printf("donne un mot : ");
114     taille2=getline(&mot,&taille,stdin);
115     taille3=1;
116     while (pow(taille3,2)<taille2-1){
117         taille3++;
118     }
119     tab=malloc((taille3)*sizeof(char*));
120     for (int i=0;i<taille3;i++){
121         tab[i]=malloc(taille3*sizeof(char));
122     }
123     for (int i=0;i<taille3;i++){
124         for(int v=0;v<taille3;v++){
125             tab[i][v]=mot[accum];
126             accum++;
127         }
128     }
129     printf("mot codé : ");
130     for (int i=0;i<taille3;i++){
131         for(int v=0;v<taille3;v++){
132             printf("%c",tab[v][i]);
133         }
134     }
135     printf("\n");
136 }
137
138 /*! \fn int main(int argc, char** argv)
139 \param argc nombre d'arguments en entrée
140 \param argv valeur des arguments en entrée
141 \brief lance le jeu*/
142 int main(int argc, char** argv){
143     if (argc==1 || argc>2)
```

```
137
138  /*! \fn int main(int argc, char** argv)
139  \param argc nombre d'arguments en entrée
140  \param argv valeur des arguments en entrée
141  \brief lance le jeu*/
142  int main(int argc, char** argv){
143      if (argc==1 || argc>2)
144      {
145          error();
146      }
147      else
148      {
149          if (strcmp(argv[1],"Vigenere")==0 || strcmp(argv[1],"vigenere")==0)
150          {
151              vigenere();
152          }
153          else if (strcmp(argv[1],"Cesar")==0 || strcmp(argv[1],"cesar")==0)
154          {
155              cesar();
156          }
157          else if (strcmp(argv[1],"Scytale")==0 || strcmp(argv[1],"scytale")==0)
158          {
159              scytale();
160          }
161          else
162          {
163              error();
164          }
165      }
166  }
167
168
169
170 }
171
```