**Pseudo code of TP**

// function for conversion to binary

Library of function

Function conversion (by ref pwd[1000])

Declare i,j,c as integer

for(I = 0; pwd[i] ! = '\0'; i++)

c = pwd[i];

for(j=7;j+1>0;j--)

if (c >= (1<<j)) then

c = c - (1<<j)

endif

output("1")

else

output("0")

output(" ")

Next

Next

//Encryption function

Declare encrypt as string[555]

Declare I,key as integer

Declare ch as string

output(“Enter key between 1 and 9 inclusive:”)

read(key)

//For lowercase characters

for(i = 0; text[i] != '\0'; ++i)

ch = text[i]

if (ch >= 'a' and ch <= 'z') then

ch = ch + key

else

if(ch > 'z') then

ch = ch - 'z' + 'a' - 1

endif

text[i] = ch

//For Uppercase characters

else if (ch >= 'A' and ch <= 'Z')

ch = ch + key

if ( ch > 'Z' ) then

ch = ch - 'Z' + 'A' - 1

text[i] = ch

endif

return key,text;

//Decryption function

function decrypt(byref text[555] as string)

declare x,key as integer

decalre ch as string

output ("Enter key between 1 and 9 inclusive:")

read(key)

//Lowercase characters

for(x = 0; text[x] != '\0'; ++x)

ch = text[x]

if (ch >= 'a' and ch <= 'z') then

ch = ch - key

else

if ( ch < 'a' ) then

ch = ch + 'z' - 'a' + 1

text[x] = ch;

//For Uppercase characters

else if (ch >= 'A' and ch <= 'Z')

ch = ch - key

if ( ch < 'A' ) then

ch = ch + 'Z' - 'A' + 1

text[x] = ch

endif

endif

Next

int main ()

declare text[500] as string

FILE\* fichier

//Open the file

fichier = fopen("TP1.txt","w")

output ("Entrer une phrase s'il vous plait:")

gets(text)

output(fichier,"Sentence: {0} ", &text)

declare i,length,hash as integer

declare pwd[555] as string

Output ("Enter the password in string:")

read(pwd)

Output("Password into binary is:");

conversion(pwd)

output("\n");

//XOR character by character

length= strlen(pwd)

for(i=0; i < length; i++)

hash= hash ^ pwd[i]

Next

//XOR letter and “decalage”

declare k, key, xor as integer

k= hash ^ key

//XOR cesar and k

declare atoi(const char \*str) as integer

xor = atoi(text) ^ k;

//Menu

declare choice[200] as integer

output("Choissisez encrypt or decrypt : ")

read(choice)

// Condition of choice

if ((strcmp(choice, "encrypt") == 0) or (strcmp(choice, "Encrypt") == 0)) then

encrypt(text)

output("Cesar encrypt : {0}" &text)

foutput(fichier,"Cesar encrypted : {0} \n", &text)

output("Encryptage XOR : {0} ", &xor)

output(xor)

output("XOR lettre et decalage (K): {0}", &k)

output("Le Hash : %d",hash)

foutput(fichier,"XOR est : {0}{1}", &xor,&xor)

// Condition of choice

if ((strcmp(choice, "decrypt") == 0) or (strcmp(choice, "Decrypt") == 0)) then

output("Enter text first in encrypt mode !")

endif

if ((strcmp(choice, "quit") == 0) or (strcmp(choice, "QUIT") == 0)) then

output("Program finished!”)

endif

output("Choissisez encrypt ou decrypt or quitter: ")

read(choice)

if ((strcmp(choice, "encrypt") == 0) or (strcmp(choice, "Encrypt") == 0)) then

output("Already encrypted !")

endif

if ((strcmp(choice, "decrypt") == 0) or (strcmp(choice, "Decrypt") == 0)) then

decrypt(text)

endif

foutput(fichier,"\nDecrypt Cesar : {0} \n", &text)

declare k1 , atoi(const char \*str) as integer

k1 = xor ^ atoi(text)

output("K est: %d",k1)

hash = k ^ key

length= strlen(pwd)

for(i=0; i < length; i++)

hash= hash ^ pwd[i];

Next

output("Le hash : {0}", &hash)

printf("Decrypt : {0} " ,&text)

if ((strcmp(choice, "quitter") == 0) or (strcmp(choice, "QUITTER") == 0)) then

output("Program Finished !")

Endif

fclose(fichier)

Date: 25/11/19