**R L’s Security Suite**

**Varun Krishna S**

XII A2

**INDEX**

Acknowledgements………………………………………………….i

Bona fide certificate.....................................................................ii

1. Introduction……………………………………………..…1
2. Abstract…………………………………………………….2
3. Source code……………………………………….………6
4. Output………………………………………………………30
5. Conclusion…………………………………………….…..34
6. Bibliography ……………………………………………...35

**Acknowledgements**

I thank my computer teacher, Mrs. Allpious R who remained patient with me throughout the development of this application and supported me in ways that I can’t explain in words.

I also thank the Principal, who gave unconditional support to make this happen.

Finally I thank my parents and all my friends who stood by my side through thick and thin.

**Introduction**

R L Security Suite, as the name suggests is a security suite which provides the user the option to encrypt or decrypt messages in four different ways:

* Steganography\*
* XOR
* Caesar
* Vigenere

R L Security Suite can be run on any computer without the prerequisite of a C++ compiler.

\*Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video. However in this program only text can be concealed in an image file.

**Abstract**

This application is built using C++ only.

**XOR Encryption**

As everything on a computer is stored as binary data, in the form of bytes (8 bits, or individual 1's or 0's)  
  
Binary data can easily be "encrypted" with a "key" based on a little Boolean operation called an xor, or exclusive or.   
when we xor a single bit (a 1 or 0) with another bit:  
  
if 1 bit is true, and 1 bit is false, it returns true, otherwise it returns false;  
so:  
  
1 xor 1 = 0  
1 xor 0 = 1  
0 xor 1 = 1  
0 xor 0 = 0  
  
now one reason this is useful is if we take the new bit and xor it with the same key (the second bit) the result will always be the first bit.   
so:  
  
1 xor 1 xor 1 = 1;  
1 xor 0 xor 0 = 1;  
  
  
Of course on a single bit, it wouldn't do much good, but when you get into higher levels of memory it's a fairly simple and effective encryption method.  
  
Now when I say everything is stored as binary data that also means strings, a single char is stored as either ascii or Unicode values.

**Caesar Cipher**

It is one of the simplest encryption technique in which each character in plain text is replaced by a character some fixed number of positions down to it.

For example, if key is 3 then we have to replace character by another character that is 3 positions down to it. Like A will be replaced by D, C will be replaced by F and so on.

**Vigenere Cipher**

It is kind of polyalphabetic substitution method. It is used for encryption of alphabetic text. For encryption and decryption Vigenere Cipher Table is used in which alphabets from A to Z are written in 26 rows.

Vigenere Cipher Encryption

**Message Text:** THECRAZYPROGRAMMER

**Key:**HELLO

Here we have to obtain a new key by repeating the given key till its length become equal to original message length.

**New Generated Key:** HELLOHELLOHELLOHEL

For encryption take first letter of message and new key i.e. T and H. Take the alphabet in Vigenere Cipher Table where T row and H column coincides i.e. A.

Repeat the same process for all remaining alphabets in message text. Finally the encrypted message text is:

**Encrypted Message:** ALPNFHDJAFVKCLATIC

The algorithm can be expressed in algebraic form as given below. The cipher text can be generated by below equation.

**Ei= (Pi+ Ki) mod 26**

Here P is plain text and K is key.

Vigenere Cipher Decryption

**Encrypted Message:** ALPNFHDJAFVKCLATIC

**Key:**HELLO

**New Generated Key:** HELLOHELLOHELLOHEL

Take first alphabet of encrypted message and generated key i.e. A and H. Analyze Vigenere Cipher Table, look for alphabet A in column H, the corresponding row will be the first alphabet of original message i.e. T.

Repeat the same process for all the alphabets in encrypted message.

**Original Message:** THECRAZYPROGRAMMER

Above process can be represented in algebraic form by following equation.

**Pi= (Ei– Ki + 26) mod 26**

The above algebraic equations have been used in the program.

**Steganography**

Steganography works by replacing bits of useless or unused data in regular computer files with bits of our important data. In our case, our data will be the plain text that we need to hide, and the unused data is the least significant bits (LSBs) in the image pixels.

The least significant bit (LSB) is the bit that when flipped from 0 to 1 or from 1 to 0, then no significant change will occur on the total value. It's the bit on the rightmost, that when flipped, the value will be only affected by 1 to be 100 instead of 101. This means that the image will not be significantly affected when we reserve this bit for our purpose. Where the most significant bit (MSB) is bit on the leftmost, that when flipped, the value will be affected by 128 (1x27) to be 229 instead of 101.

When we try to hide our data in an image (as in our case), then we need enough budget of LSBs to hide our data in. These bits are located in the image pixels. Since each pixel has three elements (R, G, and B that represent the Red, Green, and Blue elements of the pixel consecutively, assuming non-transparent image), each of these elements can have a value between 0 and 255. Now, assume that the image was 300 pixels width by 400 pixels height, then we'll have 300 x 400 x 3 = 360000 LSBs. And as each character can be represented by 8 bits, then that image can hide 360000 / 8 = 45000 characters!

**FLTK**(the graphics interface library used in the project)

The Fast Light **Toolkit** is a cross-platform Graphical User Interface library written in C++.This GUI was used because Bjarne Stroustrup recommended it in his book ‘Programming Principles and Practice Using C++’.

**Source code**

#include <FL/Fl.H>

#include<iostream>

#include <FL/Fl\_Button.H>

#include <FL/Fl\_Window.H>

#include <FL/Fl\_Input.H>

#include <FL/Fl\_Output.H>

#include<FL\Fl\_JPEG\_Image.H>

#include<FL\Fl\_Image.H>

#include <FL/Fl\_Box.H>

#include<FL\Fl\_Shared\_Image.H>

#include<FL\fl\_ask.H>

#include <cstdlib>

#include<string>

#include<Windows.h>

#include<vector>

#include<time.h>

#include<fstream>

#include<cstring>

#include<FL\fl\_show\_colormap.H>

#include<FL\Fl\_Text\_Display.H>

#include <FL/Fl\_BMP\_Image.H>

#include<stdio.h>

#include<FL\Fl\_Double\_Window.H>

#include<FL\Fl\_Scroll.H>

using namespace std;

string name;

void creds(Fl\_Widget\* o, void\* );

void steg\_op(Fl\_Widget\* , void\* );

void steg\_de(Fl\_Widget\*, void\*);

void steg\_win(Fl\_Widget\* , void\*);

void steg(Fl\_Widget\* , void\*);

void bmp\_win(Fl\_Widget\* , void\*);

void vigde\_win(Fl\_Widget\*, void\*);

void vignere2(Fl\_Widget\* , void\* );

void vig\_de(Fl\_Widget\* , void\*);

void vig\_e(Fl\_Widget\* , void\*);

void vig\_win(Fl\_Widget\* , void\* );

void cd\_window(Fl\_Widget\*, void\*);

void copy\_cb3(Fl\_Widget\*, void\*);

void copy\_cb2(Fl\_Widget\* , void\*);

void ceasar2(Fl\_Widget\*, void\*);

void c\_window(Fl\_Widget\*, void\*);

void copy\_cb(Fl\_Widget\*, void\*);

void close\_cb(Fl\_Widget\*, void\*);

void make\_window(Fl\_Widget\*, void\*);

void menu\_window(Fl\_Widget\*, void\*);

void close1(void \*);

string encrypt(string);

void fl\_color(uchar r,

uchar g,

uchar b

);

//#include<bits/stdc++.h>

#include "bitmap\_image.hpp"

using namespace std;

struct col

{

unsigned char red;

unsigned char green;

unsigned char blue;

}m;

vector <col> c;

vector <char> s;

void encode(char a, char b, char c)

{

m.red = a;

m.green = b;

m.blue = c;

}

string decode\_caller(string x)

{

//cout << "please enter the name of the file from which you want to retrive the message" << endl;

cin >> x;

bitmap\_image image((x + ".bmp").c\_str());

x.erase();

int height = image.height();

int width = image.width();

bitmap\_image cool(width, height);

cool.clear();

for (int j = 0; j<height; j++)

{

for (int i = 0; i<width; i++)

{

rgb\_t colour;

image.get\_pixel(i, j, colour);

m.red = colour.red;

m.blue = colour.blue;

m.green = colour.green;

cool.set\_pixel(i, j, m.red, m.green, m.blue);

x.push\_back(m.red);

x.push\_back(m.green);

x.push\_back(m.blue);

}

}

cool.save\_image("x.bmp");

char message[200];

for (int i = 0; i<x.size(); i++)

{

if (x.at(i) == '#')

break;

else

message[i]= x.at(i);

}return message;

}

void encode\_caller( string ta,string x)

{

char \*exe = &ta[0u];

//char exe[] = ta.c\_str;

int q = 0;

char ch;

//cout << "please enter the required message and terminate it with #" << endl;

while (exe)

{

ch = exe[q];

if (ch != '#')

s.push\_back(ch);

else

{

s.push\_back(ch);

break;

}

q++;

}

int t = s.size();

int o = 0;

cout << "please enter the name of the file on which you want to hide the message" << endl;

cin >> x;

if (!x.size() % 3)

x += '#';

if (!x.size() % 3)

x += '#';

bitmap\_image image((x + ".bmp").c\_str());

int height = image.height();

int width = image.width();

bitmap\_image cool(width, height);

for (int j = 0; j<height; j++)

{

for (int i = 0; i<width; (i++))

{

if (o<t)

{

rgb\_t colour;

image.get\_pixel(i, j, colour);

encode(s.at(o), s.at(o + 1), s.at(o + 2));

cool.set\_pixel(i, j, m.red, m.green, m.blue);

o += 3;

}

else

{

rgb\_t colour;

image.get\_pixel(i, j, colour);

cool.set\_pixel(i, j, colour);

}

}

}

cool.save\_image("cool1.bmp");

}

class encrypt

{

public:

string a;

string b;

int s;

void get\_encrypt()

{

cout << "enter a";

cout << "enter b";

cout << "enter s";

cin >> a;

cin >> b;

cin >> s;

}

string XOR\_encrypt()

{

int k = 0;

for (int i = 0; i<a.length(); i++)

{

a[i] = a[i] ^ b[0];

//k=(++k<b.length()?k:0);

}

return a;

}

string ceaser\_encrypt()

{

for (int i = 0; i<a.length(); i++)

{

if (isupper(a[i]))

a[i] = char(int(a[i] + s - 65) % 26 + 65);

else

a[i] = char(int(a[i] + s - 97) % 26 + 97);

}

return a;

}

string vignere\_encrypt()

{

string m;

for (int i = 0, j = 0; i < a.length(); ++i)

{

char c = a[i];

if (c >= 'a' && c <= 'z')

c += 'A' - 'a';

else if (c < 'A' || c > 'Z')

continue;

m += (c + b[j] - 2 \* 'A') % 26 + 'A';

j = (j + 1) % b.length();

}

return m;

}

}grt;

class decrypt

{

public:

string a;

string b;

int s;

void get\_decrypt()

{

cout << "enter a";

cout << "enter b";

cout << "enter s";

cin >> a;

cin >> b;

cin >> s;

}

string XOR\_decrypt()

{

int k = 0;

for (int i = 0; i<a.length(); i++)

{

a[i] = a[i] ^ b[k];

}

return a;

}

string ceaser\_decrypt()

{

for (int i = 0; i<a.length(); i++)

{

if (isupper(a[i]))

a[i] = char(int(a[i] - s - 65) % 26 + 65);

else

a[i] = char(int(a[i] - s - 97) % 26 + 97);

}

return a;

}

string vignere\_decrypt()

{

string out;

for (int i = 0, j = 0; i < a.length(); ++i)

{

char c = a[i];

if (c >= 'a' && c <= 'z')

c += 'A' - 'a';

else if (c < 'A' || c > 'Z')

continue;

out += (c - b[j] + 26) % 26 + 'A';

j = (j + 1) % b.length();

}

return out;

}

}gtr;

int main() {

Fl\_Window\* pwin = new Fl\_Window(1280, 720, "pic");

pwin->begin();

Fl\_JPEG\_Image\* im = new Fl\_JPEG\_Image("cavaman1.jpg");

double w = im->w();

double h = im->h();

double d = im->d();

Fl\_Box\* box = new Fl\_Box(10, 10, 1280 - 20, 720 - 20);

box->image(im);

pwin->add(box);

Fl\_Button\* cloose = new Fl\_Button(1000, 600, 100, 30, "START");

cloose->callback(menu\_window, pwin);

pwin->show();

return Fl::run();

}

void creds(Fl\_Widget\* o, void\* v)

{

Fl\_Double\_Window\* pwin = new Fl\_Double\_Window(1280, 720, "pic");

pwin->begin();

Fl\_JPEG\_Image\* im = new Fl\_JPEG\_Image("credits.jpg");

double w = im->w();

double h = im->h();

double d = im->d();

Fl\_Box\* box = new Fl\_Box(0, 0, im->w(), im->h());

box->image(im);

pwin->add(box);

pwin->end();

pwin->show();

}

void menu\_window(Fl\_Widget\* o, void\* v)

{

close1(v);

Fl\_Window\* win1 = new Fl\_Window(1280, 720, "menu");

win1->color(fl\_rgb\_color(30, 30, 30));

win1->label("security suite");

Fl\_Output \*out = new Fl\_Output(740, 70,0 , 0, "Security Suite");

out->labelsize(34);

out->labelcolor(fl\_rgb\_color(255,140,3));

out->labelfont(FL\_HELVETICA);

Fl\_Output \*outt = new Fl\_Output(800, 72, 0, 0, "BETA");

outt->labelsize(20);

outt->labelcolor(fl\_rgb\_color(255, 140, 3));

outt->labelfont(FL\_HELVETICA);

Fl\_Button\* xor = new Fl\_Button(580, 200, 140, 30, "XOR Encryption");

xor->labelcolor(fl\_rgb\_color(255, 140, 3));

xor->color(fl\_rgb\_color(62, 62, 66));

xor->callback(make\_window,win1);

Fl\_Button\* ceasar = new Fl\_Button(580, 350, 140, 30, "Ceasar Encryption");

ceasar->labelcolor(fl\_rgb\_color(255, 140, 3));

ceasar->color(fl\_rgb\_color(62, 62, 66));

ceasar->callback(ceasar2,win1);

Fl\_Button\* V1 = new Fl\_Button(580, 400, 140, 30, "Vignere Encryption");

V1->labelcolor(fl\_rgb\_color(255, 140, 3));

V1->color(fl\_rgb\_color(62, 62, 66));

V1->callback(vignere2,win1);

Fl\_Button\* bmp = new Fl\_Button(580, 275, 140, 30, "Steganography");

bmp->labelcolor(fl\_rgb\_color(255, 140, 3));

bmp->color(fl\_rgb\_color(62, 62, 66));

bmp->callback(steg\_op,win1);

Fl\_Button\* close = new Fl\_Button(580, 600, 70, 30, "&Quit");

close->callback(close\_cb);

close->color(fl\_rgb\_color(62, 62, 66));

close->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Button\* cred = new Fl\_Button(580, 500, 140, 30, "Credits");

cred->callback(creds);

cred->color(fl\_rgb\_color(62, 62, 66));

cred->labelcolor(fl\_rgb\_color(255, 140, 3));

win1->show();

}

void make\_window(Fl\_Widget\* o, void\* v ) {

close1(v);

Fl\_Window\* win = new Fl\_Window(1280, 720, "SECURITY SUITE");

win->color(fl\_rgb\_color(30,30,30));

win->begin();

Fl\_Button\* copy = new Fl\_Button(550, 300, 180, 30, "ENCRYPT/DECRYPT");

copy->color(fl\_rgb\_color(62, 62, 66));

copy->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Button\* close = new Fl\_Button(580, 600, 70, 30, "&QUIT");

close->color(fl\_rgb\_color(62, 62, 66));

close->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp = new Fl\_Input(250, 150, 200, 30, "Enter Text");

inp->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp->color(fl\_rgb\_color(62, 62, 66));

inp->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Output\* out = new Fl\_Output(500, 400, 250, 100, "Output");

out->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

out->color(fl\_rgb\_color(62, 62, 66));

out->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp2 = new Fl\_Input(800, 150, 140, 30, "Enter Key ");

inp2->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp2->color(fl\_rgb\_color(62, 62, 66));

inp2->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Output \*out1 = new Fl\_Output(740, 70, 0, 0, "XOR Encryption");

out1->labelsize(34);

out1->labelcolor(fl\_rgb\_color(255, 140, 3));

out1->labelfont(FL\_HELVETICA);

Fl\_Button\* xor = new Fl\_Button(1050, 30, 140, 30, "Main Menu");

xor->labelcolor(fl\_rgb\_color(255, 140, 3));

xor->color(fl\_rgb\_color(62, 62, 66));

xor->callback(menu\_window, win);

win->show();

win->end();

copy->callback(copy\_cb);

close->callback(close\_cb);

}

void ceasar2(Fl\_Widget\* o, void\* v)

{

close1(v);

Fl\_Window\* win = new Fl\_Window(1280, 720, "Security Suite");

win->color(fl\_rgb\_color(30, 30, 30));

Fl\_Output \*out1 = new Fl\_Output(740, 70, 0, 0, "Ceasar Encryption");

out1->labelsize(34);

out1->labelcolor(fl\_rgb\_color(255, 140, 3));

out1->labelfont(FL\_HELVETICA);

Fl\_Button\* copy = new Fl\_Button(550, 300, 180, 30, "ENCRYPT");

copy->color(fl\_rgb\_color(62, 62, 66));

copy->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Button\* copy2 = new Fl\_Button(550, 400, 180, 30, "DECRYPT");

copy2->color(fl\_rgb\_color(62, 62, 66));

copy2->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Button\* xor = new Fl\_Button(1050, 30, 140, 30, "Main Menu");

xor->labelcolor(fl\_rgb\_color(255, 140, 3));

xor->color(fl\_rgb\_color(62, 62, 66));

xor->callback(menu\_window, win);

win->show();

copy->callback(c\_window);

copy2->callback(cd\_window);

}

void c\_window(Fl\_Widget\* o, void\* v) {

Fl\_Window\* win = new Fl\_Window(1280, 720, "SECURITY SUITE");

win->color(fl\_rgb\_color(30, 30, 30));

win->begin();

Fl\_Button\* copy = new Fl\_Button(550, 350, 180, 30, "ENCRYPT");

copy->color(fl\_rgb\_color(62, 62, 66));

copy->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Button\* close = new Fl\_Button(580, 600, 70, 30, "&QUIT");

close->color(fl\_rgb\_color(62, 62, 66));

close->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp = new Fl\_Input(550, 150, 200, 30, "Enter Text");

inp->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp->color(fl\_rgb\_color(62, 62, 66));

inp->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Output\* out = new Fl\_Output(520, 450, 250, 100, "Output");

out->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

out->color(fl\_rgb\_color(62, 62, 66));

out->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp2 = new Fl\_Input(550, 250, 200, 30, "Enter Key");

inp2->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp2->color(fl\_rgb\_color(62, 62, 66));

inp2->labelcolor(fl\_rgb\_color(255, 140, 3));

copy->callback(copy\_cb2);

close->callback(close\_cb);

Fl\_Output \*out1 = new Fl\_Output(780, 70, 0, 0, "Ceasar Encryption");

out1->labelsize(34);

out1->labelcolor(fl\_rgb\_color(255, 140, 3));

out1->labelfont(FL\_HELVETICA);

win->show();

}

void vignere2(Fl\_Widget\* o, void\* v)

{

close1(v);

Fl\_Window\* win = new Fl\_Window(1280, 720, "Security Suite");

win->color(fl\_rgb\_color(30, 30, 30));

Fl\_Output \*out1 = new Fl\_Output(740, 70, 0, 0, "Vignere Encryption");

out1->labelsize(34);

out1->labelcolor(fl\_rgb\_color(255, 140, 3));

out1->labelfont(FL\_HELVETICA);

Fl\_Button\* copy = new Fl\_Button(550, 300, 180, 30, "ENCRYPT");

copy->color(fl\_rgb\_color(62, 62, 66));

copy->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Button\* copy2 = new Fl\_Button(550, 400, 180, 30, "DECRYPT");

copy2->color(fl\_rgb\_color(62, 62, 66));

copy2->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Button\* xor = new Fl\_Button(1050, 30, 140, 30, "Main Menu");

xor->labelcolor(fl\_rgb\_color(255, 140, 3));

xor->color(fl\_rgb\_color(62, 62, 66));

xor->callback(menu\_window, win);

win->show();

copy->callback(vig\_win);

copy2->callback(vigde\_win);

}

void steg\_op(Fl\_Widget\* o, void\* v)

{

close1(v);

Fl\_Window\* win = new Fl\_Window(1280, 720, "Security Suite");

ShellExecuteA(NULL, "open", "steg.exe", NULL, NULL, SW\_SHOWDEFAULT);

Fl\_Button\* xor = new Fl\_Button(1050, 30, 140, 30, "Main Menu");

xor->labelcolor(fl\_rgb\_color(255, 140, 3));

xor->color(fl\_rgb\_color(62, 62, 66));

xor->callback(menu\_window, win);

win->show();

}

void vig\_win(Fl\_Widget\* o, void\* v) {

Fl\_Window\* win = new Fl\_Window(1280, 720, "SECURITY SUITE");

win->color(fl\_rgb\_color(30, 30, 30));

win->begin();

Fl\_Button\* copy = new Fl\_Button(230, 350, 180, 30, "ENCRYPT");

copy->color(fl\_rgb\_color(62, 62, 66));

copy->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Button\* close = new Fl\_Button(580, 600, 70, 30, "&QUIT");

close->color(fl\_rgb\_color(62, 62, 66));

close->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp = new Fl\_Input(550, 150, 200, 30, "Enter Text");

inp->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp->color(fl\_rgb\_color(62, 62, 66));

inp->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Output\* out = new Fl\_Output(520, 450, 250, 100, "Output");

out->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

out->color(fl\_rgb\_color(62, 62, 66));

out->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp2 = new Fl\_Input(550, 250, 200, 30, "Enter Key");

inp2->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp2->color(fl\_rgb\_color(62, 62, 66));

inp2->labelcolor(fl\_rgb\_color(255, 140, 3));

copy->callback(vig\_e);

close->callback(close\_cb);

Fl\_Output \*out1 = new Fl\_Output(780, 70, 0, 0, "Vignere Encryption");

out1->labelsize(34);

out1->labelcolor(fl\_rgb\_color(255, 140, 3));

out1->labelfont(FL\_HELVETICA);

win->show();

}

void vigde\_win(Fl\_Widget\* o, void\* v) {

Fl\_Window\* win = new Fl\_Window(1280, 720, "SECURITY SUITE");

win->color(fl\_rgb\_color(30, 30, 30));

win->begin();

Fl\_Button\* copyd = new Fl\_Button(800, 350, 180, 30, "DECRYPT");

copyd->color(fl\_rgb\_color(62, 62, 66));

copyd->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Button\* close = new Fl\_Button(580, 600, 70, 30, "&QUIT");

close->color(fl\_rgb\_color(62, 62, 66));

close->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp = new Fl\_Input(550, 150, 200, 30, "Enter Text");

inp->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp->color(fl\_rgb\_color(62, 62, 66));

inp->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Output\* out = new Fl\_Output(520, 450, 250, 100, "Output");

out->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

out->color(fl\_rgb\_color(62, 62, 66));

out->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp2 = new Fl\_Input(550, 250, 200, 30, "Enter Key");

inp2->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp2->color(fl\_rgb\_color(62, 62, 66));

inp2->labelcolor(fl\_rgb\_color(255, 140, 3));

copyd->callback(vig\_de);

close->callback(close\_cb);

Fl\_Output \*out1 = new Fl\_Output(780, 70, 0, 0, "Vignere Encryption");

out1->labelsize(34);

out1->labelcolor(fl\_rgb\_color(255, 140, 3));

out1->labelfont(FL\_HELVETICA);

win->show();

}

void cd\_window(Fl\_Widget\* o, void\* v)

{

Fl\_Window\* win = new Fl\_Window(1280, 720, "SECURITY SUITE");

win->color(fl\_rgb\_color(30, 30, 30));

win->begin();

Fl\_Button\* copy = new Fl\_Button(550, 350, 180, 30, "DECRYPT");

copy->color(fl\_rgb\_color(62, 62, 66));

copy->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Button\* close = new Fl\_Button(580, 600, 70, 30, "&QUIT");

close->color(fl\_rgb\_color(62, 62, 66));

close->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp = new Fl\_Input(550, 150, 200, 30, "Enter Text");

inp->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp->color(fl\_rgb\_color(62, 62, 66));

inp->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Output\* out = new Fl\_Output(520, 450, 250, 100, "Output");

out->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

out->color(fl\_rgb\_color(62, 62, 66));

out->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp2 = new Fl\_Input(550, 250, 200, 30, "Enter Key");

inp2->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp2->color(fl\_rgb\_color(62, 62, 66));

inp2->labelcolor(fl\_rgb\_color(255, 140, 3));

copy->callback(copy\_cb3);

close->callback(close\_cb);

Fl\_Output \*out1 = new Fl\_Output(780, 70, 0, 0, "Ceasar Decryption");

out1->labelsize(34);

out1->labelcolor(fl\_rgb\_color(255, 140, 3));

out1->labelfont(FL\_HELVETICA);

win->show();

}

void copy\_cb(Fl\_Widget\* o, void\*)

{

string name1;

Fl\_Button\* b = (Fl\_Button\*)o;

Fl\_Input\* iw = (Fl\_Input\*)b->parent()->child(2);

Fl\_Output\* ow = (Fl\_Output\*)b->parent()->child(3);

Fl\_Input\* iCC = (Fl\_Input\*)b->parent()->child(4);

grt.a = iw->value();

grt.b = iCC->value();

name1 = grt.XOR\_encrypt();

ow->value(name1.c\_str());

}

void copy\_cb2(Fl\_Widget\* o, void\* )

{

string name2;

Fl\_Button\* b = (Fl\_Button\*)o;

Fl\_Input\* iw = (Fl\_Input\*)b->parent()->child(2);

Fl\_Output\* ow = (Fl\_Output\*)b->parent()->child(3);

Fl\_Input\* iCC = (Fl\_Input\*)b->parent()->child(4);

grt.a = iw->value();

grt.b = iCC->value();

grt.s = 1;

name2 = grt.ceaser\_encrypt();

ow->value(name2.c\_str());

}

void copy\_cb3(Fl\_Widget\* o, void\*)

{

string name2;

Fl\_Button\* b = (Fl\_Button\*)o;

Fl\_Input\* iw = (Fl\_Input\*)b->parent()->child(2);

Fl\_Output\* ow = (Fl\_Output\*)b->parent()->child(3);

Fl\_Input\* iCC = (Fl\_Input\*)b->parent()->child(4);

gtr.a = iw->value();

gtr.b = iCC->value();

gtr.s = 1;

name2 = gtr.ceaser\_decrypt();

ow->value(name2.c\_str());

}

void vig\_e(Fl\_Widget\* o, void\*)

{

string name2;

Fl\_Button\* b = (Fl\_Button\*)o;

Fl\_Input\* iw = (Fl\_Input\*)b->parent()->child(2);

Fl\_Output\* ow = (Fl\_Output\*)b->parent()->child(3);

Fl\_Input\* iCC = (Fl\_Input\*)b->parent()->child(4);

grt.a = iw->value();

grt.b = iCC->value();

grt.s = 1;

name2 = grt.vignere\_encrypt();

ow->value(name2.c\_str());

}

void vig\_de(Fl\_Widget\* o, void\*)

{

string name2;

Fl\_Button\* b = (Fl\_Button\*)o;

Fl\_Input\* iw = (Fl\_Input\*)b->parent()->child(2);

Fl\_Output\* ow = (Fl\_Output\*)b->parent()->child(3);

Fl\_Input\* iCC = (Fl\_Input\*)b->parent()->child(4);

gtr.a = iw->value();

gtr.b = iCC->value();

gtr.s = 1;

name2 = gtr.vignere\_decrypt();

ow->value(name2.c\_str());

}

void bmp\_win(Fl\_Widget\* o, void\*)

{

string a, b;

Fl\_Window\* win = new Fl\_Window(1280, 720, "SECURITY SUITE");

win->color(fl\_rgb\_color(30, 30, 30));

win->begin();

Fl\_Button\* stag = new Fl\_Button(550, 350, 180, 30, "Encrypt");

stag->color(fl\_rgb\_color(62, 62, 66));

stag->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp = new Fl\_Input(550, 150, 200, 30, "Enter Text");

inp->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp->color(fl\_rgb\_color(62, 62, 66));

inp->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp2 = new Fl\_Input(550, 250, 200, 30, "Enter Key");

inp2->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp2->color(fl\_rgb\_color(62, 62, 66));

inp2->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Output \*out1 = new Fl\_Output(780, 70, 0, 0, "Steganography Encryption");

out1->labelsize(34);

out1->labelcolor(fl\_rgb\_color(255, 140, 3));

out1->labelfont(FL\_HELVETICA);

win->show();

stag->callback(steg);

}

void steg(Fl\_Widget\* o, void\*)

{

string aa,ba;

Fl\_Button\* b = (Fl\_Button\*)o;

Fl\_Input\* iw = (Fl\_Input\*)b->parent()->child(1);

Fl\_Input\* iCC = (Fl\_Input\*)b->parent()->child(2);

aa = iw->value();

ba = iCC->value();

encode\_caller(aa, ba);

}

void steg\_win(Fl\_Widget\* o, void\*)

{

string a;

Fl\_Window\* win = new Fl\_Window(1280, 720, "SECURITY SUITE");

win->color(fl\_rgb\_color(30, 30, 30));

win->begin();

Fl\_Button\* stag = new Fl\_Button(550, 350, 180, 30, "Decrypt");

stag->color(fl\_rgb\_color(62, 62, 66));

stag->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Input\* inp = new Fl\_Input(550, 150, 200, 30, "Enter Text");

inp->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

inp->color(fl\_rgb\_color(62, 62, 66));

inp->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Output\* out = new Fl\_Output(500, 400, 250, 100, "Output");

out->align(FL\_ALIGN\_TOP | FL\_ALIGN\_CENTER);

out->color(fl\_rgb\_color(62, 62, 66));

out->labelcolor(fl\_rgb\_color(255, 140, 3));

Fl\_Output \*out1 = new Fl\_Output(810, 70, 0, 0, "Steganography Decryption");

out1->labelsize(34);

out1->labelcolor(fl\_rgb\_color(255, 140, 3));

out1->labelfont(FL\_HELVETICA);

win->show();

stag->callback(steg\_de);

}

void steg\_de(Fl\_Widget\* o, void\*)

{

string a,c;

Fl\_Button\* b = (Fl\_Button\*)o;

Fl\_Input\* iw = (Fl\_Input\*)b->parent()->child(1);

Fl\_Output\* ow = (Fl\_Output\*)b->parent()->child(2);

a = iw->value();

c= decode\_caller(a);

ow->value(c.c\_str());

}

void close1(void \* v)

{

Fl\_Window\* b = (Fl\_Window\*)v;

b->hide();

}

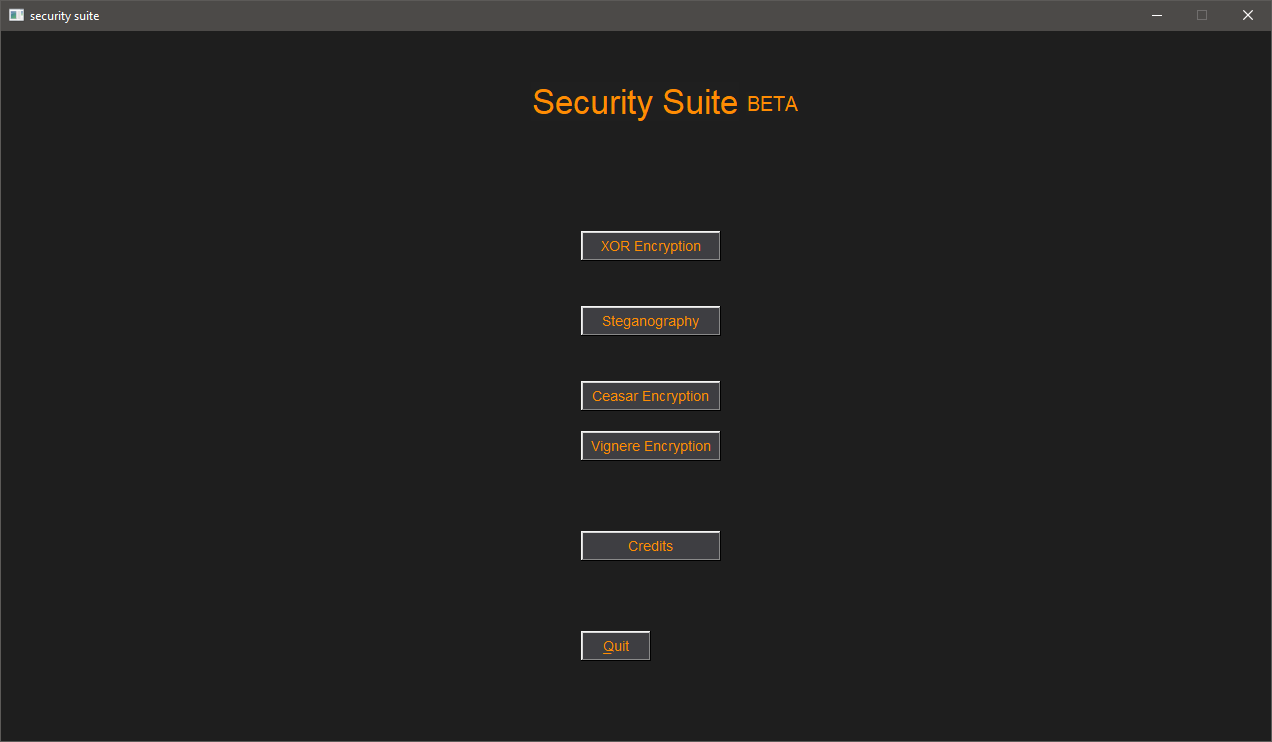
void close\_cb(Fl\_Widget\* o, void\*) {

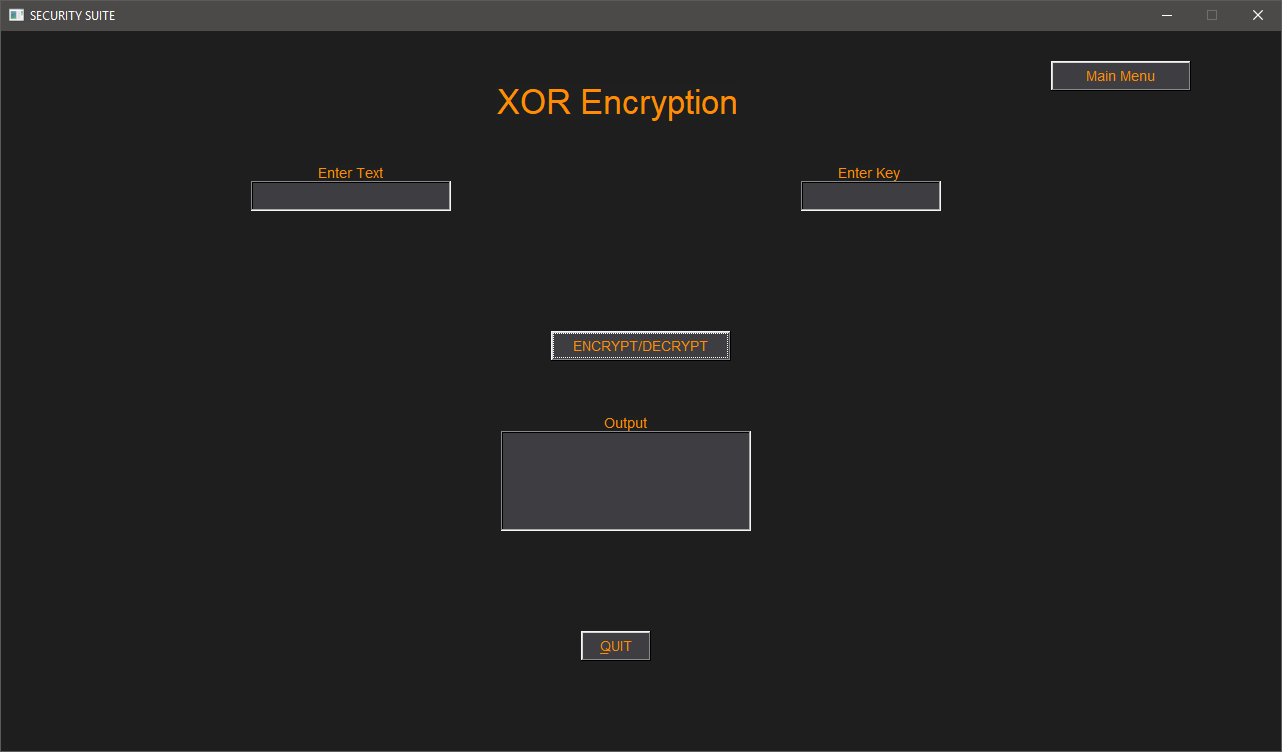
exit(0);

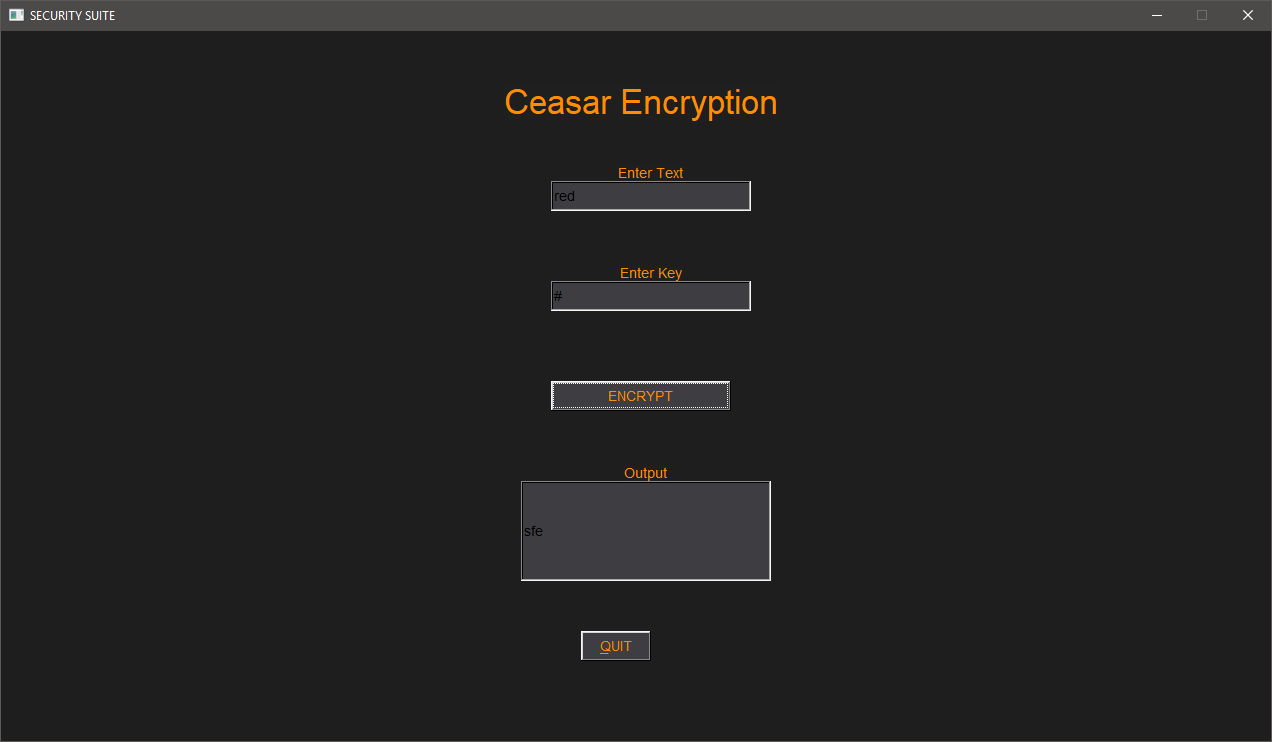
}

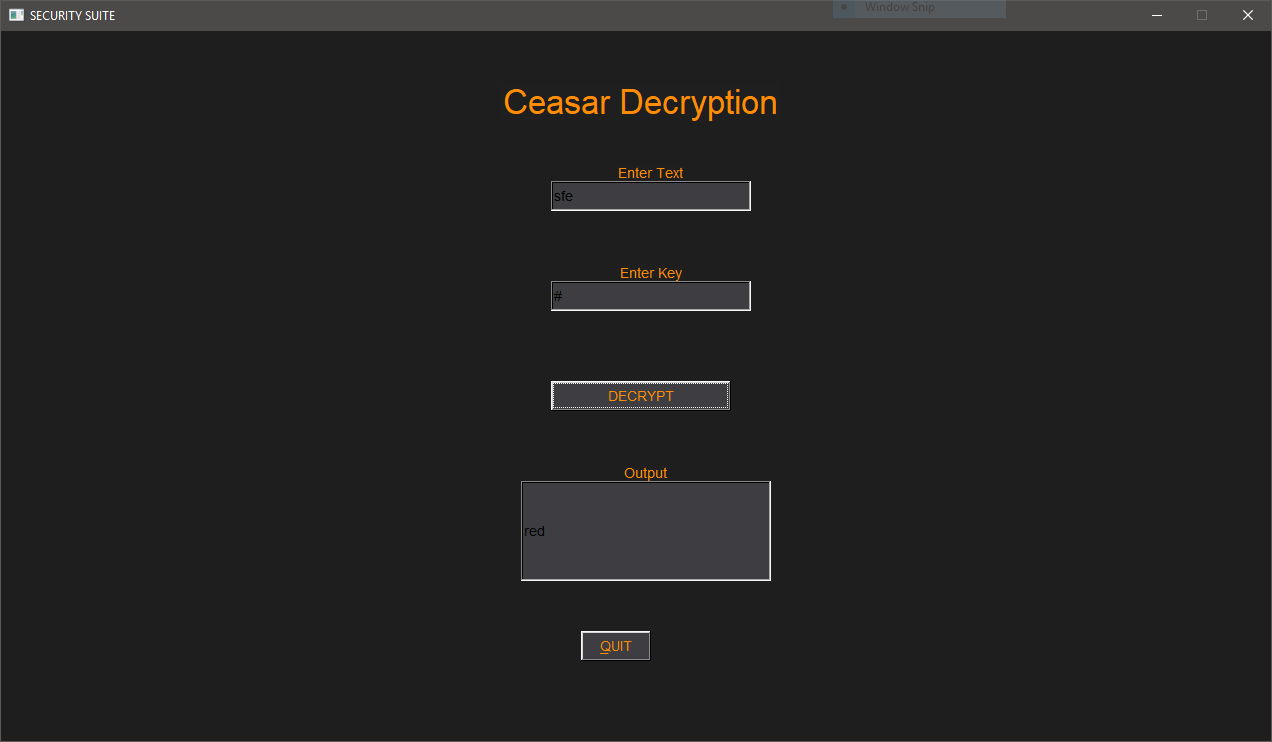
**Output**

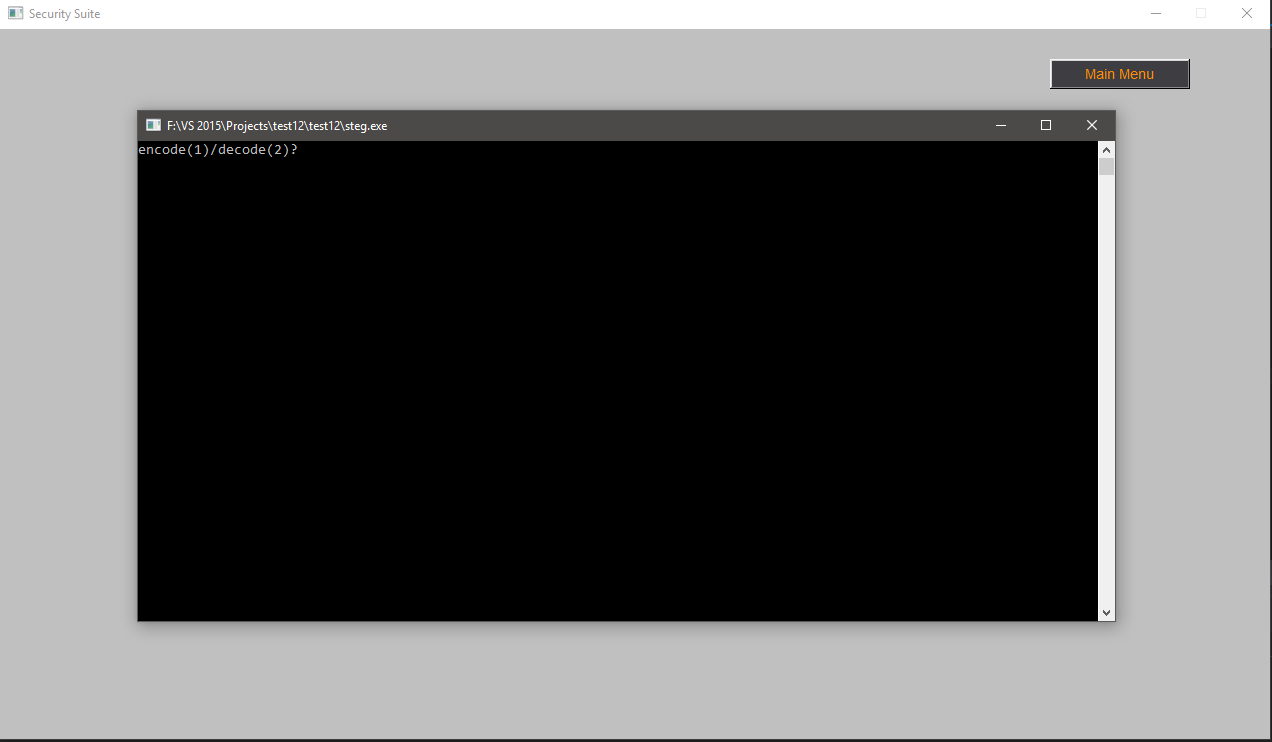


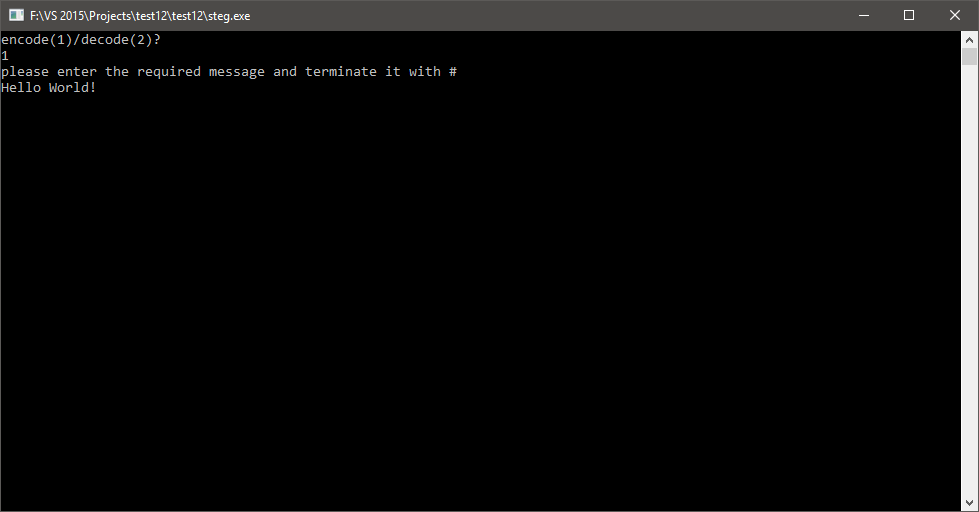












Before concealing After concealing

**Conclusion**

Possible areas of improvement:

The program is still in beta. More encryption functions can be added in the future ,and currently, the steganography function opens in a separate console window it can be integrated with the GUI.

**Bibliography**

http://www.fltk.org

[www.tutorualspoint.com](http://www.tutorualspoint.com)

[www.stackoverflow.com](http://www.stackoverflow.com)

Computer science with c++ by Sumita Arora

Programming Principles and Practice Using C++ by Bjarne Stroustrup