Joseph Velasquez

CiS-7 Project documentation

Introduction:

The purpose of this project was to create a program that implemented Vigenere Cipher in C++ using Discrete Structures. This program asks the user to input a key and a string that will be encrypted and decrypted back to the original string.

Team Members:

Joseph Velasquez

Project Information:

* What problems are you solving in this project?

This project is trying to solve the implementation of Vigenere Cipher in C++

* What solutions are you implementing in the project?

By using strings and character arrays I was able to create a program that takes an inputted string and a key and uses them to encrypt and decrypt the strings.

Encryption is handled by the equation Ei = (Pi + Ki) mod 26 and Decryption is handled by the equation Di = (Ei - Ki + 26) mod 26. However, it is not as simple as passing the strings into these equations. In order for this to work we have to iterate through each character.

* Provide explanation of calculations and algorithm implementation.

Encryption:

Ei = (Pi + Ki) mod 26

Actual code:

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

//added A to offset to ASCII [ 65-90 | A-Z ]

Decryption:

Di = (Ei - Ki + 26) mod 26

Actual code:

output += (c - k[j] + 26) % 26 + 'A';

//added A to offset to ASCII [ 65-90 | A-Z ]

* What is the program objectives? Explain how your program is interacting with the user and its purpose.

The objective is to allow the user to encrypt data by inputting a “key” and a string. The program takes these inputs and passes them into the corresponding functions which results in the encrypted and decrypted strings.

* How are discrete structures implemented in the C++ program?

Discrete structures are implemented in the use of the equations used for encrypting and decrypting strings.

* What are the limitations of the program?

This program is limited by the number of strings the user can input. Essentially this is a one and done program. This means that it will only take one key and string run through the program and end. This can easily be fixed by adding switch statements along with do/while loops. There are also restraints based on the data types being used however it is not likely to affect the program.

* Provide recommendation on improving the limitations of the program.

One recommendation on improving the limitations of the program would be to add switch statements along with do/while loops in order to allow the user to input more than one string and key. This would allow the user to encrypt and decrypt more than one string without having to rerun the program.

Pseudocode:

//Input/Output Library:

#include <iostream>

#include <string>

using namespace std;

Function Prototypes:

string Vig(string);

string encryption(string,string);

string decryption(string,string);

Start of program:

Declare variables:

string KEY;

string ori;

Display on console:

Input the key to be used to encrypt/decrypt

Key must be capitalized

User input:

cin>>KEY;

Display on console:

Input word or phrase to encrypt/decrypt:

User input:

cin>>ori;

cout<<endl;

Call functions:

string Vig(KEY);

string encrypt = encryption(ori,KEY);

string decrypt = decryption(encrypt,KEY);

Display on console:

Original Message: original string

Encrypted Message: encrypted string

Decrypted Message: decrypt string

End of program

Functions:

Function handles KEY:

string Vig(string k) {

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

if (k[i] >= 'A' && k[i] <= 'Z')

//conditional statement checking if a character is between “A” - “Z”

k += k[i];

else if (k[i] >= 'a' && k[i] <= 'z')

//conditional statement checking if a character is between “a” - “z”

k += k[i] + 'A' - 'a';

}//End of for loop iterating through characters in string

Return k;

Function returns string k used to hold user input for KEY

}

Function handles encryption:

string encryption(string t, string k) { //passing inputted string & key

string output; //stores new string to be displayed

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

For loop iterating through original inputted string

char c = t[i]; //creating character array

//this section of the code takes care of the equation Ei = (Pi + Ki) mod 26 which is used for encryption however because of the way characters are stored by the computer we need to determine if the string is lower, upper case, or a special character.

As a result, we needed to add three conditional statements to determine what characters are in the string

This is placed inside the for loop that iterates through the characters in the sting that was passed into the function

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

//conditional statement checking if a character is between “a” - “z”

c += 'A' - 'a';

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

//added A to offset to ASCII [ 65-90 | A-Z ]

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

}

else if (c >= 'A' && c <= 'Z'){

c += 'A' - 'A';

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

//added A to offset to ASCII [ 65-90 | A-Z ]

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

}

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

c=c;

output += c ;

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

}

}//End of for loop iterating through characters in string

Function returns output string

//returns encrypted string that is displayed on console

}

Function handles decryption:

string decryption(string t, string k) { //passing encrypted string & key

string output; //stores new string to be displayed

//this section of the code takes care of the equation Di = (Ei - Ki + 26) mod 26 which is used for decryption however because of the way characters are stored by the computer we need to determine if the string is lower, upper case, or a special character.

As a result, we needed to add three conditional statements to determine what characters are in the string

This is placed inside the for loop that iterates through the characters in the sting that was passed into the function

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

char c = t[i]; //creating character array

if (c >= 'A' && c <= 'Z'){

c += 'A' - 'A';

output += (c - k[j] + 26) % 26 + 'A';

//added A to offset to ASCII [ 65-90 | A-Z ]

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

}

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

c=c;

output += c;

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

}

}

return output; //returns decrypted string that is displayed on console

}