Janaye Jackson

Professor Nguyen

CIS – 17A

12/13/2022

Course Project Documentation

**Problem & Solution**

Problem – encrypting and/or decrypting messages.

Solution – create a simulated Vigenere Cipher which encrypts message based on the shift keys position in the alphabet.

**Program Objective**

The objective of this project includes making a program that encrypts or decrypts user provided text. The encryption/decryption is done using a Vigenere Cipher. User input a message to be encrypted/decrypted and input the key that will encrypt or decrypt the message. The encrypted/decrypted message is then outputted to the screen. For this project the target audience is those who like to encrypt messages or need help decrypting a message.

**Implementation**

This program is implemented using a self-created class called Vigenere. In this class there are three functions, one to create shift key and the others two to encrypt/decrypt messages. The function to create the shift key determines the ascii value of each letter in the key, subtracts the base value of A or a to tell how many letters need to be shifted and adds that number to a vector until the vector has the same number of elements as the letters in the message. The functions to encrypt/decrypt the message take the value in the vector and use those as the shift. Each value corresponds to a specific letter in the message. In the encrypt function, the shift key value and letter value are added together and the base value for a or A is subtracted. That number is then divided by 26 and the remainder of that value is added to the value of a or A to get the new letter. That letter is then added to a string that returns the encrypted message. The decrypt function is very similar to this but the arithmetic before the division is performed is reversed, so addition is subtraction and subtraction is addition. The main code just ask the user to pick an option and enter their message and shift key based on their option.

**Limitations**

This program does function in its intended way but there are some things that would make it better. As of now I have not mastered the way in which to incorporate spaces in the users message and the shift key. If a space is inserted the program will not work as intended. I do have input validation for the menu choices provided to the user, but if they decide to input something other than a number the program will crash.

**Future Hope**

In the future I hope to improve this program into including multiple cipher encryption/decryptions. I also hope to improve this program to allow user to encrypt/decrypt more than one message at a time.

**Pseudocode**

**Main.cpp()**

int main()

Declare and initialize variables

Do{

Output menu

Input user choice

If(choice > 1 && < 3)

Choice is valid

Set valid to true

Else

Output and restart loop

}while(!valid)

If(choice == 1)

Output

User input

Output

User input

Create Vigenere object using user input

Output and call encrypt function of Vigenere class

Else – if(choice == 2)

Output

User input

Output

User input

Create Vigenere object using user input

Output and call decrypt function of Vigenere class

Else

User has chosen to exit, exit program

**Vigenere.cpp()**

Vigenere::Vigenere()

Constructor

Initialize and declare variables of this class

void Vigenere::shiftKey()

Initialize and declare variables

Increment for loop

Initialize variables

If(lower case letter)

Subtract a ascii value from letter

Add int value to vector

Else

Subtract A ascii value from letter

Add int value to vector

If(end of keyword reached)

Initialize variable to 0

String Vigenere::encrypt()

Declare and initialize variable

Call shiftKey()

Increment for loop

If(lower case letter)

Calculate ascii value based on lower case letters, make it a character, add it to string

Else

Calculate ascii value based on upper case letters, make it a character, add it to string

Return variable

String Vigenere::decrypt()

Declare and initialize variable

Call shiftKey()

Increment for loop

If(lower case letter)

Calculate ascii value based on lower case letters, make it a character, add it to string

Else

Calculate ascii value based on upper case letters, make it a character, add it to string

Return variable