

Programming Project Report

Name: Jan Bobda

Date: 10/02/2018

Academic Integrity Statement: I pledge that I have neither given nor received unauthorized help on this programming assignment.

Problem Statement:

The goal of this programming assignment was to complete two tasks. The first task was to implement the Caesar Cipher encryption algorithm. The second task was to revise that algorithm to change the shift after every character iteration based on a mathematical formula that we were supposed to create. The user should enter their choice of classic caesar encryption/decryption or revised caesar cipher encryption/decryption, then choose their shift value, and then enter the string that they would like encrypted/decrypted. The program would then print out the encrypted/decrypted string and reloop through the program. The project required that we made sure that the user entered a proper menu choice, that the initial shift value is between 1 and 25, and that the program properly handled characters outside of the alphabet. According to the sample output provided on Dr.Gauch's website, the program should shift a given string to the left by the shift amount. So the string "HELLO MOM PLEASE SEND MONEY" should encrypt to "EBIIL JLJ MIBXPB PBKA JLKBV."

Design:

One design decision that I had to make was how to handle the menu and menu error checking. To address this, I created a menu that loops through and offers the user one of four options, classic caesar encryption/decryption or revised caesar cipher encryption/decryption. If the user enters an invalid menu option, the program will tell the user that the input is incorrect and prompt them again until they input a proper value. The program then asks for a shift value, if the shift value is less than 1 or greater than 25, then the program will re-prompt the user to input a valid shift input until the user does so. Lastly, if the user enters characters outside of the alphabet, it will simply ignore the character and will only convert the characters that are in the alphabet. To implement the Caesar cipher/decipher algorithm I used a recursive approach where you pass in the initial index value of the function. The function accesses the index of the string that is input, then shifts the value based on the shift, and then makes a recursive call with the next index passed to the recursive function call. The revised caesar cipher/decipher works very similarly but rather alters the shift value every recursive call according to the mathematical function that I created that is seeded by the original shift value entered.

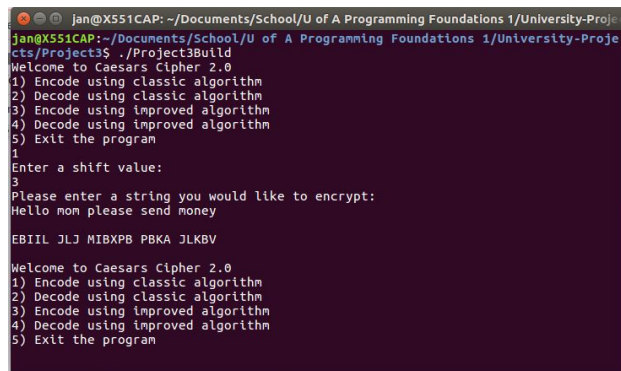
Implementation:

I started with an empty Cpp file and incrementally added functionality to it. I did not start with any sample code and started the program by implementing a simple iterative approach to the caesar cipher program, and shortly afterwards I added the caesar de-cipher function. Afterwards I added the menu for the program and added error checking. After I had a functioning program, I decided to challenge myself by implementing caesar cipher/decipher recursively. After I completed this, I added the revised caesar cipher/decipher and added it to the menu in the program. This program took roughly a week because I spaced out the programming across several days, adding each function incrementally.

Testing:

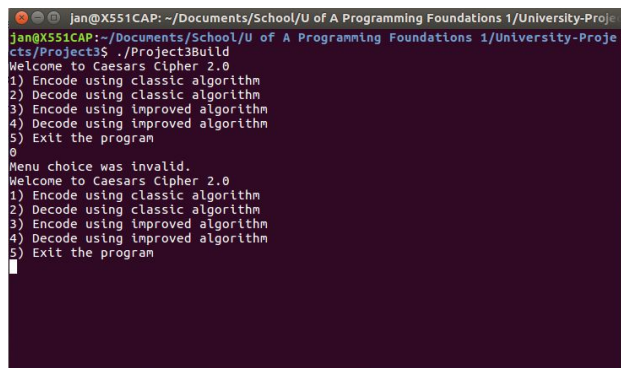
For the testing of the program I tested edge cases, normal cases, and erroneous inputs to see how the program would handle them. Here are some test cases that I tried for testing.

Normal Case Test



```
jan@XS51CAP: ~/Documents/School/U of A Programming Foundations 1/University-Proje
jan@XS51CAP:~/Documents/School/U of A Programming Foundations 1/University-Proje
cts/Project3$ ./Project3Build
Welcome to Caesars Cipher 2.0
1) Encode using classic algorithm
2) Decode using classic algorithm
3) Encode using improved algorithm
4) Decode using improved algorithm
5) Exit the program
1
Enter a shift value:
3
Please enter a string you would like to encrypt:
Hello mom please send money
EBIIL JIJ MIBXPB PBKA JLBV
Welcome to Caesars Cipher 2.0
1) Encode using classic algorithm
2) Decode using classic algorithm
3) Encode using improved algorithm
4) Decode using improved algorithm
5) Exit the program
```

Invalid Menu Choice Test



```
jan@XS51CAP: ~/Documents/School/U of A Programming Foundations 1/University-Proje
jan@XS51CAP:~/Documents/School/U of A Programming Foundations 1/University-Proje
cts/Project3$ ./Project3Build
Welcome to Caesars Cipher 2.0
1) Encode using classic algorithm
2) Decode using classic algorithm
3) Encode using improved algorithm
4) Decode using improved algorithm
5) Exit the program
0
Menu choice was invalid.
Welcome to caesars cipher 2.0
1) Encode using classic algorithm
2) Decode using classic algorithm
3) Encode using improved algorithm
4) Decode using improved algorithm
5) Exit the program
```

```
Jan@X551CAP: ~/Documents/School/U of A Programming Foundations 1/University-Proje
Jan@X551CAP: ~/Documents/School/U of A Programming Foundations 1/University-Proje
cs/Project5 - /Project5Build
Welcome to Caesars Cipher 2.0
1) Encode using classic algorithm
2) Decode using classic algorithm
3) Encode using improved algorithm
4) Decode using improved algorithm
5) Exit the program
1
Enter a shift value:
-5
Enter a shift value:
0
Enter a shift value:
26
Enter a shift value:
2
Please enter a string you would like to encrypt:
Hello
FCJJM

Welcome to Caesars Cipher 2.0
1) Encode using classic algorithm
```

```
Jan@XSS1CAP: ~/Documents/School/U of A Programming Foundations 1/University-Proje
Jan@XSS1CAP:~/Documents/School/U of A Programming Foundations 1/University-Proje
cs/Projects1. ./Project3Build
Welcome to Caesars Cipher 2.0

1) Encode using classic algorithm
2) Decode using classic algorithm
3) Encode using improved algorithm
4) Decode using improved algorithm
5) Exit the program
1
Enter a shift value:
3
Please enter a string you would like to encrypt:
[[[????????]]]
[[[????????]]]

Welcome to Caesars Cipher 2.0

1) Encode using classic algorithm
2) Decode using classic algorithm
3) Encode using improved algorithm
4) Decode using improved algorithm
5) Exit the program
1
```

Conclusions:

The programming project ended up being a success. The Program successfully takes an input string, and a shift, and then applies the Caesar cipher algorithm, or the revised Caesar cipher algorithm correctly to the input string and is also able to decrypt a string encrypted by that method. Something that I would do differently would be to find a more elegant solution to the problem, and make sure that I have more thorough testing. The project took one week to complete because I spread the work out and really put the program together piece by piece.