**Daniel Williamson**

**Project #5**

**3/11/16**

**CS 200**

Project Overview

Purpose:

The purpose of this project was to have a user input a message and use base64 encoding to encode said message and then decode back to the original message.

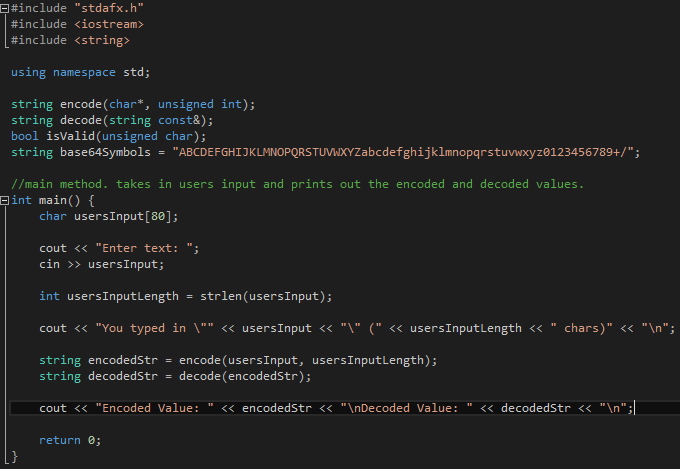
Approach:

To begin this project, I immediately went to the internet to gain a better understanding of how to program in C++. Also, how base64 encoding works. Once I read up, I began to plan how the program should run. I decided to make a decode, encode and main methods. The decode method was to do just that, decode the now encoded message back to its original method. The encode method was to do just that, encode the message the user inputted. And lastly the main method to ask the user for input and print out the encoded and decoded message. As I went through I ended up making another method, isValid, to verify if the encoded values were in fact valid to decode.

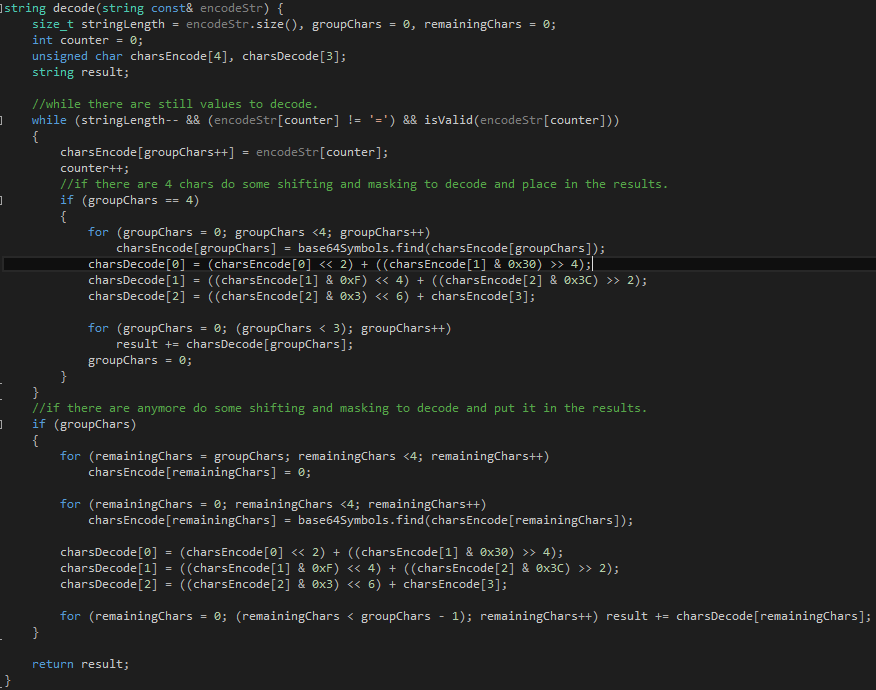
Results

These are my results:

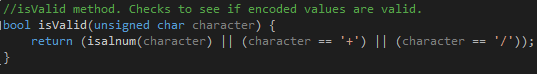
This is my main method, my includes, namespace and declaring of methods. The main method takes in a user’s input and calls the encode and decode methods to do their thing.



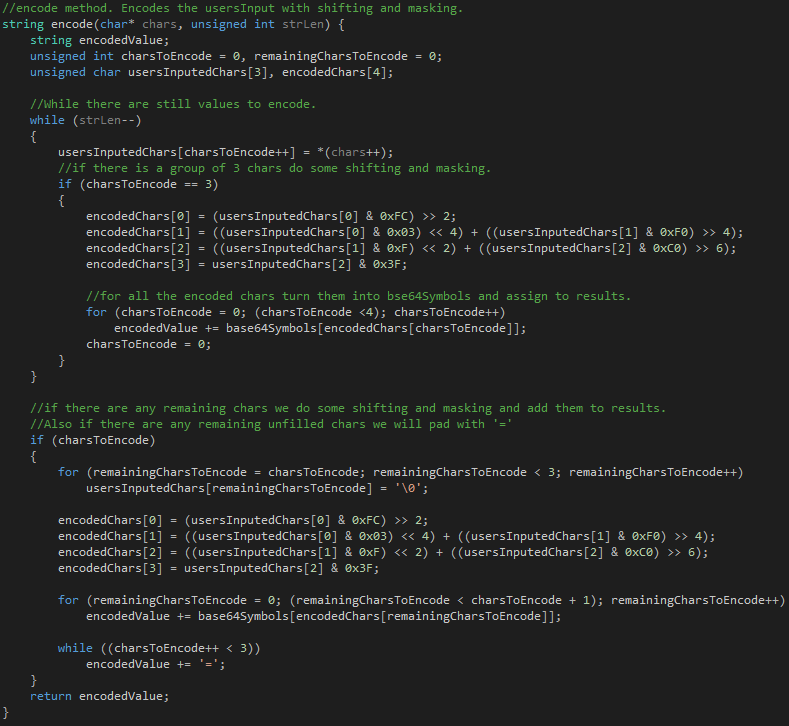
This is my decode method. It decodes the encoded message the user input, back into what the user inputted. It does this by shifting each char and masking them.



This is my isValid method. It determines if the encoded values are valid to decode.

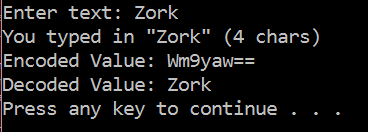


This is my encode method. It takes in the users input and uses base64 encoding to encode it, by shifting and masking each char.



Testing:

Knowing that the base64 encode for Zork is Wm9yaw== from your pdf file posted on BBlearn. We can clearly see that the output from my program is correctly displaying the encode for Zork and also the decode.



Conclusion

In conclusion this was an extremely difficult project. I relied heavily on the internet for how base64 encoding worked. However, because of that I am now more proficient in both, programming in C++ and how base64 encoding works. The hardest challenge I faced was doing the correct shifting and masking for the encoding. Once I figured that out, the decoding was simply going backwards, in a way.