Joshua Pollock

CS 200 Project 5

24 March 2017

Joshua Pollock

CS 200

Project 5 – Base64

24 March 2017

**Purpose:**

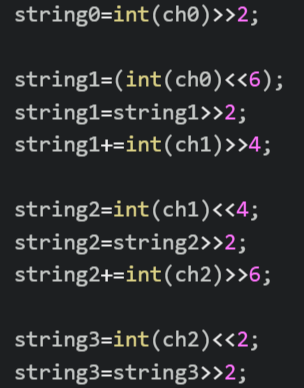
The purpose of this project is to create a program that will ask the user to input a string and then convert that string into base64 format. The decoding portion of this lab will be able to prompt the user to input a base64 string, and then will convert that string into regular format. This project will help to expand my skills with bit manipulation, and help further my ability with C/C++.

**Research:**

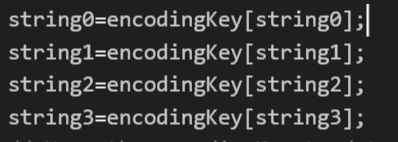
This project looks quite difficult. However, the project’s PDF file covered the project guidelines quite well. I did take the advice of the PDF and searched for the Wikipedia page for Base64(<https://en.wikipedia.org/wiki/Base64>). Looking for programs that accomplish this task, I found none. The closest thing I could find was on Github but did not function correctly(<https://gist.github.com/taxilian/b5991396932855cea6dd>).  
  
After looking at the PDF file and the Wikipedia page, I decided that I would begin programming using the given skeleton code.

**Program:**

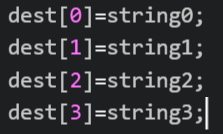
Beginning with the skelekton code, I found that it did not compile and run. By adding #include <cstring>, the errors changed to casting errors. I went through the skelekton code to correct these by casting unsigned characters into characters. After I got this to compile, I began work on the encode function. I began by setting up statements to check for empty bits (if ch1==’\0’). This is due to project constraints and these bits should be displayed as ‘=’. Next I cast the characters into ints and performed binary manipulation on them.



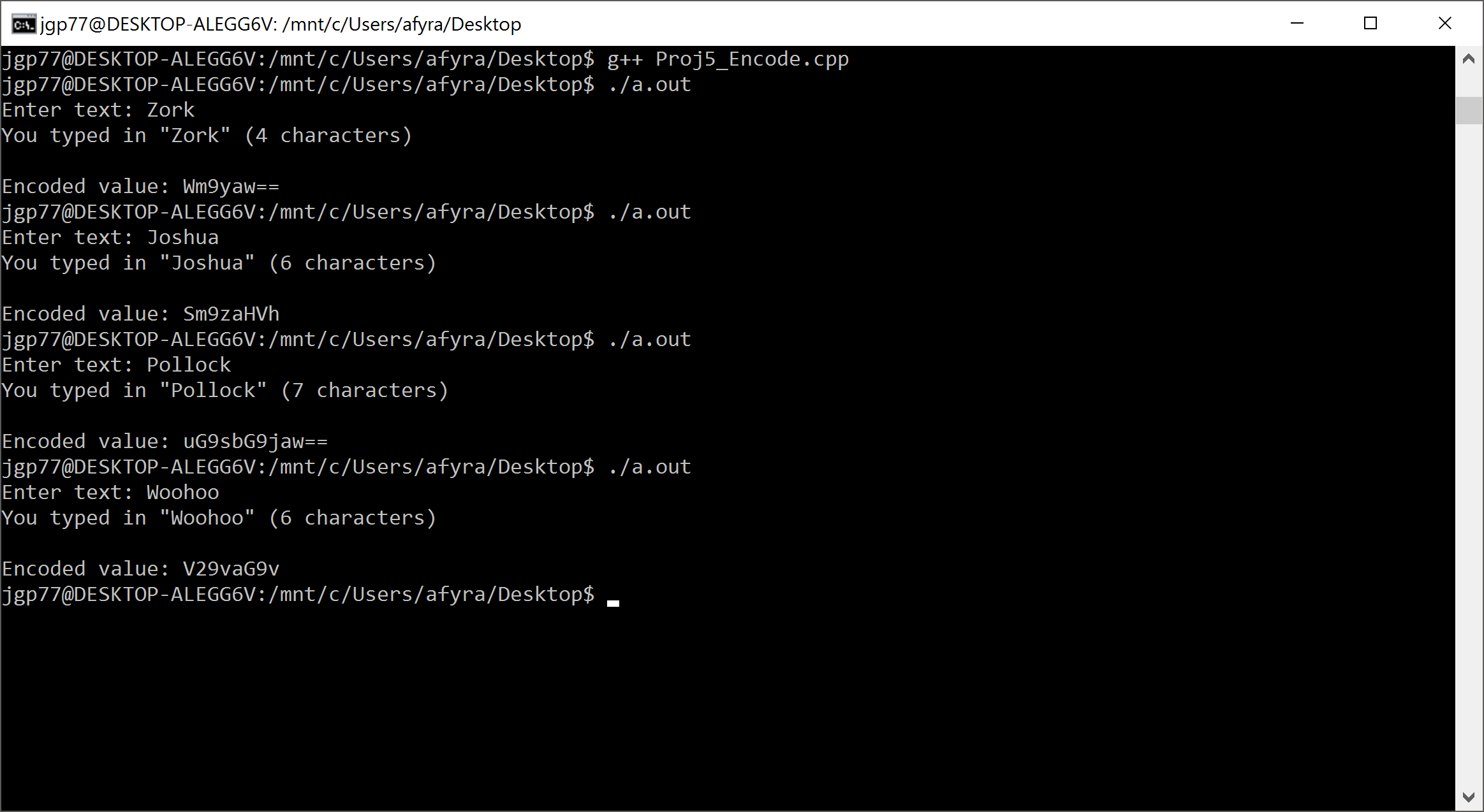
After performing this manipulation, I set some empty variables to the outputs translated using the encoding key.



If the input contains empty bits, depending on the amount of them, string2 and string3 may be set equal to ‘=’. Outside of the conditional statement I finished by using skelekton code to output these variables to the main function.



After this was complete, the encoding function of the project was finished. Below is the testing outputs of the program.



Like in the PDF, I inputted the string “Zork” and had matching outputs. I also tested some short strings just to make sure it worked with words other than Zork.

Sadly, I was unable to figure out how to create the decoding function properly for this project.

**Conclusion:**

This was quite a difficult project. I was unable to complete the decode portion of the project but successfully implemented an encoding section of the project. I am quite happy with how the encoding section turned out, and do not see room for improvement on it. To compile and execute my code, I used the Windows 10 terminal. I ran the command ‘g++ Proj5\_Encode.cpp’ and followed it up by using ‘./a.out’ to execute the code. Due to time constraints on this project I will be unable to turn in a working decode function. I do wish to complete the decode function in the next following days. Overall, this lab helped to improve my C++ coding. I liked troubleshooting the skeleton code, and finding out how to fix it.