Umar Kagzi

Question 6 Whitepaper

6. Use a stack to pass in a string and reverse each word in the string.

Question 6 was a simple yet enjoyable homework question. The task at hand was to parse in a text file with text and reverse the individual words of the file without reversing the letters. I was able to complete this question using a stack and a few handy built-in C++ cstdlib functions.

The heart of the program is in the reverse function I created, which encompasses all processes of the program in one single function. Since the question asked to read in text from a file, I used the ifstream library to read in a file I created called jokes.txt. This file included the "Chuck Norris Joke" that I wanted to reverse.

After opening the text file, I wanted to convert the text into a string to make readability easier for the stack. I used the "istreambuf_iterator" built in std::string function to read in the file's text and store the text as a string. This built in function reduced a lot of overhead and made it easy for me to continue on with the program.

After this step, I created a stack using the std::stack library and created the main part of the function: the traversing for loop. This for loop would traverse throughout the string we created and look at the individual index values of the string.

If the index was anything besides a "space" (character, number, etc), it would push the value to the stack. If a space were to be found at the index, the program would print out the values of the stack one by one by popping the printed value off the stack (think LIFO, the last value would now be printed first, essentially doing all the reversing work for us). After the stack

was found to be empty, the program would continue on to the next word until the final word was reached.

To run the program, the user simply needs to call the reverse function (assuming the "jokes.txt" file is present) and every word in the file would be reversed using the evaluating method I mentioned.