Damian Diaz
N00821283

CSC 217 JA Fall 2017
Project #1 : Vowels

Project #1 Analysis

Programming Environment
OS: macOS High Sierra
Editor: Atom by GitHub
Compiler: gcc in Terminal

Testing
To ensure my program worked as it should, I continuously experimented with different types of inputs. My project is programmed to receive any input whether typed manually or copy and pasted in the terminal, and continue to run until it reads a EOF or an empty line. So I tested multiple lines such as, “xxaxxexixxoxxxuxx” and full paragraphs like the sample input provided in the project outline. I also tried mixing capital and lowercase vowels like “aEiOu”. Only when my program passed of all these tests was I convinced it was functioning properly.

Modularity
Since I never coded in C before, I can admit that my use of modularity fell to the wayside in the early stages of this project. However, I was quick to realize when my functions were doing too many things or when the code was repeated too much. I decided to plan out my code as shown below,

main (int) - this function controls the whole program and contains a single while loop that runs the loop until the “checkLine” function returns a length less than 1. Once the while loop finishes, it prints the output char array to the terminal.

checkLine (int) - this function works almost as a parser that handles one line at a time. The function runs a while loop and once the line is complete, sends that line to the “checkVowels” function. If checkVowels returns 1 it calls the “saveLine” function which saves the line to the output array. The function ends by returning the length of the line.

checkVowels (int) - This function accepts a char array and runs through each char calling “isVowel” on each one. If isVowel returns 1, it runs through the algorithm to keep track and count the order of the vowels. If the count reaches 5, then the function found all 5 vowels in order and returns 1.

isVowel (int) - This function compares a passed char to the array of vowels declared in an array. If char is a vowel then it returns 1.

saveLine (void) - This function accepts a char array and saves that line to the output.

Although its hard to judge myself on my use of modularity in a language I have just learned. I strongly feel my program is very modular and meets the requirements to attain maximum amount of points.

Resources and Challenges
On this project my main resource was the text “The C Programming Language. Kernighan, Brian W. Dennis M. Ritchie". My program has implemented many skills and logic from lessons taught in the textbook, such as “getLine” function, searching for new lines by comparing characters to “\n” and the use of the EOF value in ending my loops.

My biggest challenge in this project was my “checkLine” function. Specifically making sure my lines were properly being read and checked. A major bug I kept having was my while loop would continue to run even after my input was blank. But by reading the textbooks example of “getline”, and correctly checking for ‘\n’ and EOF I was able to make a solution that worked with my project.