Phillip Pham
CSE 310 Section 84794
Dr. Violet R. Syrotiuk

Project #1: Milestone Report

1.) **Describe any problems encountered in your implementation for this project milestone.**
   - The biggest problem I had with this project was deciding which data structure to use to store the strings. The reason that this was a problem, was that I figured out that I could not use a basic array due to the fact that C++ seemingly does not allow dynamic sizing of an array (array sizes must be defined at compile time) and it must use a constant integer for size. The solve for this was to use a vector data structure instead of an array, in which was risky because I did not know if it would be allowed on the project. It ended up being acceptable, and the solution ended up making the code design simpler since vectors are a dynamic data structure but are still very similar to basic arrays in function.
   - The next problem was remembering that in C++ you can use pass-by-reference so that you don't have to use auxiliary or "copied" vectors every time you process the strings, in which you can save resources and memory by doing so.
   - Realizing that some of the code I used such as "to_string()" function is only used by newer versions of the C++ compiler, so I had to edit my makefile accordingly to support my code, which ended up working on the ASU General server.

2.) **Describe any known bugs, and/or incomplete implementation in the project milestone.**
   - All of the required functionality for the milestone seems to have been met by my implemented code (judging off of the provided .pdf).
   - A bug that I did run into when implementing my "clusterize()" function was that every time it would read an empty line, it would crash my program. My solution to this was to add a condition that if the string that is passed in is just a new line character ("\n"), then it would just add that same new line character to the encoded output. This ended up working out well, as it helps format the output text and provides empty lines where it is also present in the input text.
   - I also came to the realization that "<input.txt" and ">output.txt" do not count as command line arguments that get stored in argv[]. This made me realize that I could just do "encode insertion" and then type whatever strings I want, type in Ctrl+Z and return, and then it will output the result to console if ">output.txt" is not specified in console. This was an amazing discovery for testing purposes. Not a bug, but a cool feature discovery!

3.) **Describe any significant interactions with anyone (peers or otherwise) that may have occurred.**
   - No code was shared with others at all, but there were some discussions in class/recitation about the data structure issue mentioned in my answer for question #1. There were a lot of other students who had the same issue that I was having, and then Nicole our TA mentioned that we could use vectors instead.

4.) **Cite any external books, and/or websites used or referenced**
   - http://www.cplusplus.com/reference/string/string/
   - http://www.cplusplus.com/reference/string/string/find/
   - http://www.cplusplus.com/reference/string/string/substr/
   - http://www.cplusplus.com/reference/vector/vector/
   - https://stackoverflow.com/questions/47612169/c-stop-asking-for-input-on-ctrl-d