Kripash Shrestha
CS 202 Spring 2017
Project 1 documentation
1/31/2017

The purpose of the program is to use the functions copyNames, find_smallest_string, find_string_length and my_string_compare_and_alphabetize to get the first 10 names from an input file (the user must enter the name of the input file), print the gathered list of names to the terminal, arrange the list of 10 names alphabetically, print the alphabetically arranged list of 10 names to the terminal and then print is to the output file (the user will enter the name of the output file.)

My design process/thought for this was to use the proper file stream taught in lecture and shown on the PowerPoints. I laid down a skeleton and focused on the getting the list of 10 names from the file at the very beginning. After that I printed the list of names to the terminal to make sure that everything was working fine and properly. After I was on par with where I was supposed to be, I transferred the list of names to a second 2-Dimensional array. I then asked the user for an output file name. After that was working as intended, I focused on printing the second array of names to the terminal. Then I printed the list from the 2-Dimnensional array to the output file.

I wanted to tackle the easy problems first so that I could work on bubble sorting the array to alphabetically sort the array. I first used my functions to find the string length and the minimum string length of two compared arrays so I know how many loops to run to test the columns of each row. I tried using many nested for loops to do this but in the end I managed to do it by going through every single row first and then check each and every single column until they matched my logic. After matching my logic, I matched the arrays properly and swap them. If it ever came to a point where the names were similar to identical, I would use a comparison of string lengths (from my function) to determine which one would swap properly with the other. For example, Dan and Daniel, the first has 3 characters so the loop would run 3 times and realize that the two are similar because you are comparing the first 3 characters of both words, which are Dan and Dan. After that is determined, then it will determine which has the longer string length and which has the lower one and then it will swap the arrays alphabetically like so.

After I figured out the entire algorithm process, I divided my program into functions and properly called them and arranged everything. **My indentation may be a little weird/ awkward but that is how I learned to indent my code.**

The main problems that I encountered when working on my algorithm was with bubble sorting and getting the loop counter correct for each and every single nested for loop that I had. I used my string length and smallest string length function to solve this issue and then after that was solved, and then I used the swap function to swap the array row locations. This was my main problem while trying to figure out the algorithm to solve this issue. One other minor issue that I had was to figure out how to do all of this without the string library. So in response to that, I just made my own string copy, string length, string compare, and such functions. I used the null terminator that I put in each array row to solve these issues.

If I could make a change if I was given more time, it would be that I would add a function for the file i/o stream. Instead of just doing it each time I have to open and close a file, I would create a function and use that to make it more smooth and less tedious.