Christopher Eichstedt

CS-202/Spring 2016

Project 1 Documentation

I was given the task to create a program that could read in a file, take the list of names which were separated by newline characters. Print the unsorted list and then to sort them alphabetically into a character array, print the newly sorted list and save them into a new user named file. The whole process was supposed to be a test our ability to code using standard I/O in C++ and to act as a review of CS135. Using the C++ I/O code we were given in class, I set out to make my program with a straightforward design. I initially built it without a query for a file name and had it boot a file called, “unsorted”. And then automatically save it into another one called, “sorted.” This was later changed to reflect the minimum requirements for project 1. I knew that I had to build my own functions that would act similar to string copy and to print out the character arrays. Although, with this in mind, it wasn’t as easy I had expected. Mostly in part because I had forgotten a necessary function.

The biggest challenge I had, was with sorting and string compare. Originally I had tried to contain my string compare in a simple “if” statement, and tried to contrast between the first and second element of my character array. All while using a swap sort, that would take the greatest value character row and store it into a temporary character array. Then put the lower value where the highest was and finally put the temp value into where the highest previously was. After some refinement, I got the swap concept down but was still getting an exact copy of the unsorted list. I decided to take a break until I could figure out what the issue was stemming

from and then during Professor LaTourrette's lecture, she reminded me to check individual functions prior to trying to attempting the whole program. I sought out guidance at the ECC and had found that I had overcompensated certain parts with unnecessary secondary character arrays and nested loops. I proceeded to clean up my work and found that I was running into less issues but still couldn't get my program to sort correctly.It was then that I had conversed with my peers, and found that I was forgetting a major component that was listed in the project document, a string compare function. It was with this addition, that my program began to function correctly. And with some minor alterations, primarily adding a null terminator at the end of my while loop, within my string compare functions. I was able to get all of the listed names, in correct alphabetical order.

With that said, the only major changes I had wished to make was to of asked for some guidance earlier, instead of brute forcing my ideas. I learned that you can gain better perspective with more than one person tackling a problem. And that it is okay to ask for help, when you find yourself incapable of finding a solution directly.