Nick Alvarez

CS202.1101

Project 1

The purpose of this program was to work with file input and output in a new language, as well as working with arrays and functions. A user specified input file with names will be read in. The list of names will be printed, with numbers preceding them. These numbers will remain with the specified name throughout the rest of the execution. Next, the names will be sorted by length. Finally, the names will be listed in alphabetical order. In addition, whatever is outputted to the terminal will be outputted to a user specified text file.

In designing this program, I had some framework already laid down. From CS135, I had built custom string library functions, so I was able to bring those in to this program. However, those functions were designed with pointers in mind, so that had to be adjusted. The main function was not difficult to create, as the assignment very clearly laid out what was expected, part-by-part. While finishing the program, most of the remaining time was spent on perfecting the sort and alphabetize functions, both requiring some visualizations so as to better understand what was happening when the array elements were swapped.

A glaring issue faced was in writing the code and compiling it. Using g++ in the Macintosh terminal yielded multiple errors. However, using g++ on the Debian machine worked fine. The biggest issue with the project was the prohibition of pointer usage, considering how useful they are in iterating through arrays. Finally, the third troubling issue that took some time to fix was the sort functions. Originally, a name would be missing from the length and alphabetical arrays. Changing the for loop yielded correct results with the alphabetical array, but the last two elements of the length array would not be sorted. The issue was simple. The variable used to iterate was not reset to zero between the length and alphabetical loops.

Given more time, implementing most of the program into respective functions would be a priority. The structure of the code looks jumbled, despite it working nonetheless. Also, the compare function has a lot of useless lines that weren’t relevant to the project. However, it still worked, so that is acceptable. Tidiness would be the goal if there was more time.