Logan Leavitt

CS 202 1001

Project 1

The goal of this code is to sort a list of ten names by length and alphabetically. The list of names is read from a specified file and the sorted lists are outputted to the terminal and two output files specified by the user. The names should also be preceded by their original position in the list. My program begins by prompting the user for the names of the input file and output files, and opening them. If a file is unable to open, the program print that it was unable to do so, close any previously opened files and then halt execution. Names are then read from the input file to the names array. In order to maintain the original order of the names I initialized another array named “nameOrder” in which each element represented the index of a string. This way I could sort the names array without actually altering it. Since the program has to print the names to the terminal and a file at various times, I wrote a function which takes an ostream object and prints the list of names according to the format specified. With input and output in place, the last was to implement the sorting algorithm. For my sorting function I used the bubble sort algorithm. If I were to make any changes to my code I would most likely trade bubble sort for a more efficient algorithm, although speed isn’t an issue with only ten elements to sort. Because the program has to sort the names in two different ways, the sort function also takes a character which specifies the type of sort. While sorting, the function passes this character and two names to the compare function, which decides whether the elements should be swapped or kept in place. When two elements should be swapped, their respective indices in the nameOrder array are swapped instead of the strings themselves. Although I did write the myStringCopy function because it was a

requirement of the project, my implementation did not need to use it. If I was given more time and allowed to include libraries other than iostream and fstream, I would write a function which would take a varying number of files and close them all. This would compact the largest part of the main function, which was checking to see if a file did not open.