

Assignment 0

In this assignment we were tasked with creating an extensible program that could read a string and sort the words in the string. The length of the string, number of words and length of the words is unknown so dynamic data structures must be used to solve this problem. To prevent file limit errors, we used console arguments to determine the amount of inputs the user sends into the program. The *int argc* tells how many arguments were passed into the program. The character array *argv* tells us what the command line arguments were. The first element (*argv[0]*) is always the call to the program itself i.e the *./program_name* part of the command. To make sure no more than a single argument is given; a quick check of *argc* is done as seen in the code. The string to be sorted itself is stored in the next argument (*argv[1]*). To help tokenize the string into its subsequent “words”, we used the *strtok* function to move through the string. The delimiter for this *strtok* function is by default white space so we implemented a function called *makeSpaces* which turns all nonletter characters into spaces allowing us to use *strtok* easily. For the sorting itself a modified version of insertion sort is used. The sorting algorithm finds the smallest word in the linked list for each iteration and after a word has been determined to be the smallest; the logical array named “used” is updated. The smallest node is then inserted into the answer linked list at the very end. This makes sure that the words are increasing alphabetically. This finding the smallest node operation is performed until all the elements in the “used” array are 1.

This program has several features such as graceful error handling and excellent modularity. Several of the functions that we made were re used in different situations. This ease of reusability makes the program easily understood.