Name: RUID:

Daniel Fraser   174005456

Peter Laskai   157002815

String Sorter

Objective: Break a string into substrings that only contain upper and lower case letters, then sorting them in ascending order.

Unique Features: A linked list paired with an unique string function that breaks and sorts each string as it is added into the linked list.

Overview: Using a self defined stringSplitter function, pointersorter.c first takes a string and splits it into tokens consisting of alphabetic characters either lowercase, uppercase, or a combination of both. It does this by first finding an alphabetical character and continues down the string until it hits a non-alphabetical character (like the null terminator). While processing, the program checks for errors to make sure the program can store the strings in order to pass to the linked list.

      The new string arguments are then sent into addNode() along with the head node in order to select which node the string comes before. It first checks if head is empty, otherwise it starts comparing it against the linked list using stringCompare() which returns an int (-1 if the new string is after the current string, 0 if they are the same length and same characters, 1 if the new string comes before the current string). It compares the string with the strings inside the linked list by first getting the length of the shortest string. Afterward it compares the strings up n bytes. If the n bytes are the same, we choose the comparison based on the shortest string of the two.

      After it finds where to put the new string, it malloc()'s space for the new node and the string inside of it, and checks for errors if any arise in the process. After every string is added, it prints the list starting at the head and goes down the line until it hits the end of the list. Afterward it clears all memory that was malloced, both the string inside the node and the node itself. If any error occurs during anytime of this process, the program prints a message based on the error and main() returns the error code (-1 if malloc() failed to get memory for a string (char*), and -2 - if malloc() failed to get memory for a node). The program still clears any memory that has been allocated already.