README

Partners: Claudia Pan (cp728)

Yelin Shin (ys21)

pointersorter description:

Our code takes the input string argv[1] and has a char pointer that holds a copy of the input string. If there is no input, the program ends and returns "No input error." Using a while loop, we loop through the string based on the length of the string. Checking each index, we determine if it is an alphabetical letter or not. The first determined alphabet letter gets set as the first index, and as we continue to traverse through each index, the next time we hit a non-alphabetical letter, that index-1 is the last index. When we hit the non-alphabetical, we will know the first index, last index and length of the first word. At this time we create a new node that will contain the word. These nodes will be linked together by a link list. We only add words to the link list. This process continues until the loop is done. Once the loop is done, we check if the input even had an alphabetical input, if not we end print "Error: there are no words" and terminate the program. If not, we call a bubble sort method that uses bubble sort to sort the link list in alphabetical order. Bubble sort is implemented by a do while loop to compare  two linked list next to each other and swap if they are not in the correct order. This process continues until every two compared link list is in the correct order. We compare each word using strcmp. After that, we just traverse through the link list by starting at the head and print the words out.

Features: Linklist; each node stores a char of words and can point to the next node

Bubble sort; switching order of linklist by node->next manipulation.