

README for Assignment0:

Xiao Yan (xy184), Satita Vittayaareekul (sv439)

-To compile: gcc -Wall -Werror -fsanitize=address stringsorter.c -o stringsorter

-To run: ./stringsorter "input_string"

Note: to compile, you can also simply using the Makefile (to compile: make)

Results:

It will print out the extracted and sorted words from the input string, any non-alphabet character will be taken as a separator.

Main Function and Sorting Part:

-In this stringsorter assignment, we choose to use **binary search tree** to sort all the string.

-The best case of BST from accessing, searching, insertion and deletion is $O(\log(n))$. The worst case of BST from accessing, searching, insertion and deletion is $O(n)$.

-stringsorter.c is consisted by three parts, BSTsorting method, BSTdisplay method, and the main method.

-In the **main method**, we first check that if the input string is in the right form such as only one input string, the input should not be an empty or empty string. Then we start to iterate through the input string and record the length of it until we meet a separator. When we run into a separator, we create an array based on the length we recorded before and copy the word we found into the array. Run sorting method with the length of the word and the word. We run the sorting method again if the last part of the input string is not a separator. At the end, we display the whole tree.

-In the **sorting method**, we first create a new node for the incoming word and copy the array into the newnode based on the input length and array. If the tree is not empty, we start iterating through the whole tree until we reach the bottom of tree which is where we are going to insert the newnode. After we find the spot, then we insert the newnode depends on if it is the right child or left child of the previous node.

-Finally, we **display** the whole tree with recursion method.

Thanks for grading!