CS240, Fall 2021 – HW5

Due: midnight Dec 21, 2021

Instructions

1. Submit your code to the contest website (the link is on the class website)
2. Your submission will be automatically graded.
3. You can submit unlimited times and the highest grade among all submissions will be taken.
4. The input and output of your code must be stdin and stdout, respectively. The system's judge test consists of 10 pairs of input/output files. Your program will be tested with our input and the judge test will compare your output with our output. If they match for each test case, you earn 10 points. If they match for all 10 cases, you earn 100 points.
5. You should test your code locally and make sure it builds correctly before submitting.

Problem Description: Write a C program to do the following

Input (from stdin): A text consisting of two lines of words in the following format:
+ line 1: the first line of words, separated by **a white space**.
+ line 2: the second line of words separated by **a white space**.
+ note that the last line does not contain '\n'

Requirement:
+ use a binary search tree to store the words on the first line, each in a node.
+ the tree follows the order that the word at a node always appears before the word at its right child node and after the word at its left child node, according to the dictionary (alphabetical) order. You can use function strcmp(.) to compare strings.
+ after the tree is built, delete all the nodes storing the words on the second line. The lower case and upper case of a character are considered the same (e.g., "hi" and "Hi" and "HI" are considered identical).

Output (to stdout): two lines of text, all converted to lower case
+ line 1: the original words in the alphabet order, separated by single white space
+ line 2: the remaining words (after deletions), separated by single white space, in the order they appear original on line 1 of the input text

Hint: you should create a struct for each node of the tree with the following attributes:
+ the word that is stored
+ the position of the word in the original text input's line 1
+ the pointers for the left and right children of the node

Examples:

Input1

```
hello world 123
world
```

Output1

```
123 hello world
```

hello 123

Input2

Hello world 123
world

Output2

123 hello world
hello 123

Input3

123 abc def def 456 Def
123 def

Output3

123 456 abc def
abc 456

Input4

123 abc 456 def

Output4

123 456 abc def
123 abc 456 def

Input5

abc

Output5: