National Institute of Technology Calicut
Department of Computer Science and Engineering
Third Semester B. Tech.(CSE)
CS2092D Programming Laboratory
Modification Question for Assignment-7 (18.11.2021)

Instructions: For the two questions given below, write the design in the shared doc. Upload your design (of both questions, Q1& Q2) as a **single** .pdf file in the Eduserver on or before 2:45 pm in the link provided for *submitting the design of the Modification question.* After submitting the design, implement your design using *C Language* and show the output of your programs to the evaluator for the test cases given for the Modification question in Eduserver. In any case, you should submit your C Programs of both questions, Q1& Q2, as separate files on or before 03:45 PM in the corresponding links provided in Eduserver for *submitting the C Program for the Modification question.* In case of clarifications, your evaluator will help you.

Marks	Q1 (Design + Implementation)	1 + 4
	Q2 (Design + Implementation)	3.5 + 1.5
Submission Deadlines	Q1& Q2 Design	02:45 PM
	Q1 & Q2 Implementation	03:45 PM
Naming Conventions	For Design	<ROLLNO>_<FIRSTNAME>_ASSGN7_MOD.pdf
	For Implementation	<ROLLNO>_<FIRSTNAME>_ASSGN7_MOD_1.c <ROLLNO>_<FIRSTNAME>_ASSGN6_MOD_2.c

The marks for the implementation will be based on the results for the test cases. The evaluator will be conducting a viva for a maximum of 5-10 minutes.

QUESTIONS

Write pseudocodes/algorithms for the following Binary Search Tree (BST) operations and implement them. You can reuse the functions submitted for Assignment 7 and you need not write those functions in the pseudocode.

1. **Combine1(T1, k, T2):** Combines the two BSTs T1 and T2 along with another key value k to create a new BST consisting of all keys in T1 and T2 and the key k. It is assumed that each key in T1 is smaller than k and each key in T2 is greater than k. Both T1 and T2 should be empty after the operation. The running time of this function should be O(1).

   Input format:

   - First line of the input contains the keys in T1 separated by space.
   - Second line of the input contains the keys in T2 separated by space.
   - Third line of the input contains the key k.
   - If a line is blank, then the corresponding tree is empty.

   Output Format:

   - The output is the inorder traversal of the newly constructed BST.

   Sample Input 1:
   1 2 3 4 5
   //T2 is empty
   6

   Sample Output 1:
   1 2 3 4 5 6

Sample Input 2:
//T1 is empty
7 8 9 10
6

Sample Output 2:
6 7 8 9 10

Sample Input 3:
//T1 is empty
//T2 is empty
6
Sample Output 3:
6

2. **Combine2(T1, T2):** Combines the two BSTs T1 and T2 and creates a new BST consisting of all keys in T1 and T2. It is assumed that all keys in T1 are smaller than all keys in T2. There should be an invocation of Combine1() from Combine2().Both T1 and T2 should be empty after the operation. The running time of this function should be O(height(T1)).

   Input format:

   - First line of the input contains the keys in T1 separated by space.
   - Second line of the input contains the keys in T2 separated by space.
   - If a line is blank, then the corresponding tree is empty.

   Output format:

   - The output is the inorder traversal of the newly constructed BST.

   Sample Input1:
   1 2 3 4 5
   7 8 9 10

   Sample Output1:
   1 2 3 4 5 7 8 9 10

   Sample Input2:
   1 2 3 4 5
   //T2 is empty

   Sample Output2:
   1 2 3 4 5

   Sample Input 3:
   //T1 is empty
   7 8 9 10

   Sample Output 3:
   7 8 9 10