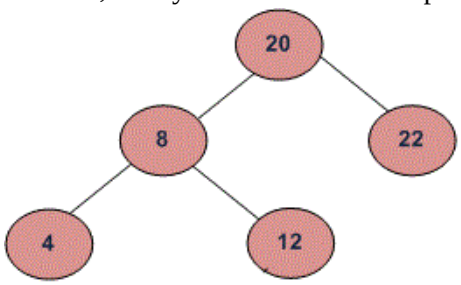
Indian Institute of technology, Guwahati
Department of Computer Science and Engineering
Data Structure Lab (CS210) Assignment: 4

Date: 20th August 2018. **Total Marks: 30**

1) (a) You are given the pre-order of a binary search tree (BST). You have to build the BST. The complexity of your algorithm should not be more than O(nh) where n is the number of nodes in the BST and h is the height of the BST. Your program should work for all test cases given as separate files. (Test the correctness of your program). **(10)**

(b) Given two values k1 and k2 (where k1 < k2) and a root pointer to a Binary Search Tree, print all the keys of tree in range k1 to k2. i.e. print all x such that $k1 \leq x \leq k2$ and x is a key of given BST. Print all the keys in increasing order. Use the tree constructed in question 1(a) to test your program. **(10)**

For example, if k1 = 10 and k2 = 22, then your function should print 12, 20 and 22.



Input:
The first line of input contains an integer N, where N is the number of elements and the second line of input contains N integers separated with spaces. In line 3, two range values k1 and k2 are given.

Output:
The keys of the BST with the range $k1 \leq x \leq k2$ in ascending order.

Input
6
10 5 1 7 40 50
6 43
Output
7 10 40

Important: Write a single main function for both problems. Create the BST with a function call and then call the second function to print the numbers in range k1 and k2. To ensure that your function for 1(a) is working fine, print the keys of the BST in in-order.

Evaluation Guidelines:

1. Full marks if both methods work for all test cases.
2. If code is not working but most of code is written, maximum 50% can be given based on TA's evaluation.
3. 20% marks will be deducted for each test not running.

4. At most 10% marks will be deducted for bad coding style. i.e., (1) code is not modular (2) code is not properly indented (3) code is not properly commented and (4) Variable and functions are not suitably named.

2) Our objective is to evolve an O(n) algorithm to construct BST from a given pre-order of a BST. We have given additional inputs - the **indexes of elements in the in-order** traversal of the BST. To help you out, you are also given a skeleton of the program as well. Make proper changes in the code wherever you find **/*TODO*/** to obtain the efficient implementation of binary search tree. The inputs are already defined in the program itself, which are **pre-order** and **indexes of elements in its in-order.** **(10)**

   Note: If you are not aware of vector in cpp, you can use array in c.

Evaluation Guidelines:

1. If the code is working, full marks.
2. 1 mark for each correct TODO replacement code.