CSE 4304-Data Structures Lab. Winter 2022-23
Batch: CSE 21
Date: 25 September, 2023,
Target Group: All
Topic: Binary Search Tree (BST)

Instructions:

- Regardless you finish the tasks in the lab, you have to submit the solutions in the Google Classroom. In case I forget to upload the tasks there, CR should contact me. The deadline will always be at 11.59 PM of the day in which the lab has taken place.
- Task naming format: fullID_T01L02_2A.c/cpp
- If you find any issues in the problem description/test cases, comment in the google classroom.
- If you find any test case that is tricky that I didn't include but others might forget to handle, please comment! I'll be happy to add.
- Use appropriate comments in your code. This will help you to easily recall the solution in the future.
- Obtained marks will vary based on the efficiency of the solution.
- Do not use <bits/stdc++.h> library.
- Modified sections will be marked with BLUE color.

Group	Tasks
2A	1 2 3 8
2B	1 2 3 4
1A	
1B	
Assignment	1 2 3 4

Task 1

Disneyland has built its airport. The airport has only one runway, which results in heavy traffic. So the authority has decided to create a ***'Runway reservation system'*** for their only runway, which will take the reservation of any transport desired to use the runway.

Before making the entry, the system checks for reservations within the three-minute range of any existing reservation(s). For example, if there is a reservation in the k^{th} minute, it won't take any reservation in k-1, k-2, k-3, k+1, k+2, $(k+3)^{th}$ minutes.

Your task is to help them build the system using Binary Search Trees(BST). (Take reservations until the user gives '-1' as input.)
For every reservation, print the existing reservations in a sorted manner.

Sample Input	**Sample Output**
50 75 53 25 60 29 45 42 28 10 -1	50 50 75 50 75 (Reservation failed) 25 50 75 25 50 60 75 25 29 50 60 75 25 29 45 50 60 75 25 29 45 50 60 75 (Reservation failed) 25 29 45 50 60 75 (Reservation failed) 10 25 29 45 50 60 75

Note:

- **Do not use any recursive function in this lab.**
- Utilize the **insertion**, **inorder-traversal** function of BST.

Task 2:

'Runway reservation system' has a new requirement. They want to introduce a feature that will allow any transport owner to make a new query, which will allow any transport owner to give a timestamp as input. The system will tell '*How many reservations are in the system before it*?'.

One of their employees proposed a solution that traverses the tree in an In-order fashion and then finds the timestamps that are less than the query. They are not happy with this ***O(n)*** solution. They want you to solve this problem in ***O(height)*** time.
Your task is to fulfill their requirement.

The first line of input will give you the number of queries.
Each query gives you the timestamp of a specific reservation. Your task is to find the number for reservation before that timestamp.

Sample input	**Sample output**
(current reservations) 50 75 25 29 45 60 10 -1	
5 45 75 50 10 29	 3 6 4 0 2

Explanation:

45 has 3 before it (10 25 29)
75 has 6 before it (10 25 29 45 50 60)
50 has 4 before it (10 25 29 45)

[Hint: You might need to use a new attribute for each node called 'subtree-size'. Start traversing from the root and try to use that subtree-sizes wisely. It will lead you to a solution of O(height).]

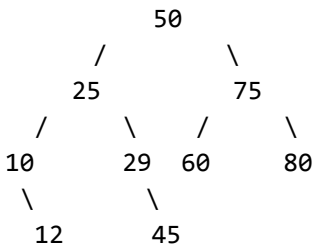
Task 3

Suppose a set of numbers are stored using a Binary Search Tree(BST). Two numbers **x** and **y** are given as input(x<y). Your task is to find the maximum number that will be encountered from the path going from x to y.

[**O(n)** solution will not be accepted]

Sample input	**Sample output**
(current values) 50 75 25 29 45 60 10 80 -1	
8 10 50 25 45 60 80 25 60 12 25 10 60 50 60 75 80	 50 45 80 75 25 75 75 80

Explanation:



Now if the value of (x,y) = (10,50) the path to reach from 10...50 contains the nodes with values {10,25,50}. We need to pick the maximum value amongst them, which is 50.

If (x,y) = (25,60) the nodes encountered in the path from 25...60 are {25, 50, 75, 60}. The maximum value amongst them is 75.

Task 4

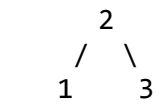
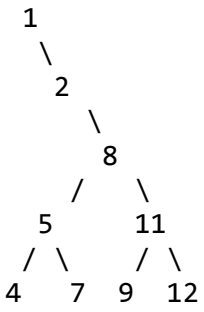
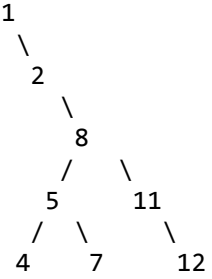
'Runway reservation system' has given its final requirement. Now they want your help to introduce a new feature that will show the updated list of reservations after a certain transport has used the runway(that transport will be taken out of the reservations set).

The first line of input will give you the number of queries.
Each query gives you the timestamp of transport which has used the runway. Your job is to **remove** the reservation and show the updated set using level-order traversal.

Sample input	**Sample output (level-order traversal)**
(current reservations) 50 75 25 29 45 60 90 10 80 100 84 88 -1	
6 29 25 75 45 50 80	50 25 75 10 29 60 90 45 80 100 88 ***(initial)*** 50 25 75 10 45 60 90 80 100 84 88 50 45 75 10 60 90 80 100 84 88 50 45 80 10 60 90 84 100 88 50 10 80 60 90 84 100 88 60 10 80 90 84 100 88 60 10 84 90 88 100

Task 8:

Given a Binary Search Tree and a node value X. Delete the node with the given value X from the BST. If no node with value x exists, then do not make any change.

Sample Input	**Sample output**	**Explanation**
 **3** **2 1** **2 3** X = 12	1 2 3	In the given input there is no node with value 12 , so the tree will remain the same.
 **9** **1 2** **2 8** **8 5** **8 11** **5 4** **5 7** **11 9** **11 12** X = 9	1 2 4 5 7 8 11 12	In the given input tree after deleting 9 will be 

Note: You don't need to follow the sample input verbatim. Just take the nodes as input so that the tree looks like the tree given in the sample input.

Note 2: The inputs now have the total number of nodes n in the first line. The lines that follow contains all the edges of the tree $u - v$, where u is the parent and v is the child.