

Ceil from BST

Problem Statement

[Suggest Edit](#)

Ninja is given a binary search tree and an integer. Now he is given a particular key in the tree and returns its ceil value. Can you help Ninja solve the problem?

Note:

Ceil of an integer is the closest integer greater than or equal to a given number.

For example:

`arr[] = {1, 2, 5, 7, 8, 9}, key = 3.`

The closest integer greater than 3 in the given array is 5. So, its ceil value in the given array is 5.

Detailed explanation (Input/output format, Notes, Images)



Input Format:

The first line of input contains a single integer T , representing the number of test cases.

The first line of each test case contains elements in the level order form. The line consists of values of nodes separated by a single space. In case a node is null, we take -1 in its place.

The second line of each test case contains integer X , denoting the key value.

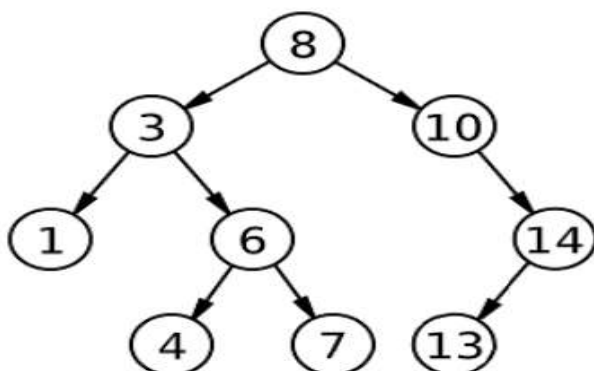
Output Format :

The first and only line of each test case in the output contains ceil of integer X from given BST.

Note:

You are not required to print the expected output; it has already been taken care of. Just implement the function.

Example



```
for the above tree
X=2
ceil =3
X=7
ceil =8
X=12
ceil =13
```

Constraints:

```
1 <= T <= 10
1 <= N <= 10^5
0 <= node data <= 10^9
1 <= X <= 10^9

Time limit: 1 second
```

Sample Input 1:

```
2
8 5 10 2 6 -1 -1 -1 -1 -1 7 -1 -1
4
8 5 10 2 6 -1 -1 -1 -1 -1 7 -1 -1
7
```

Sample Output 1:

```
5
7
```

Explanation For Sample Output 1:

In the first test case, we traverse the tree starting from the root node which has a value of 8. Now the value of its left child is 5 and its right child is 10. Key-value 4 is less than the left child so we traverse the left subtree. Now we reach a node with value 5 and then again repeat the above process till we reach a null node. Finally, we return 5 as our answer.

In the second test case, we traverse the tree starting from the root node which has a value of 8. Now the value of its left child is 5 and its right child is 10. Key-value 7 is less than the right child and more than the left child, so we traverse the right subtree. Now we reach a node with value 10 and then again follow the same procedure and reach a node with value 7 and stop there and return 7 as our final answer.

Sample Input 2:

2

55 25 82 13 34 67 86 6 21 28 47 61 70 84 92 1 10 17 24 26 29 45 54 56 65 68 81 83 85 91

96 -1

34

84 19 97 0 56 96 99 -1 8 50 83 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

50

Sample Output 2:

34

50