Problem Description

Given a binary search tree, write a function to find the node with the kth smallest value in it and return its value.
Note: You may assume that k is always valid, 1 ≤ k ≤ BST's total elements.

Input format

Line1 to X: Details of the binary tree structure (refer section below for the format)
Last line contains a single integer k.

Output format

Print the value of the kth smallest element.

Constraints

1 <= Number of nodes (N) <= 105
1 <= Value of nodes <= 109
1 <= k <= N

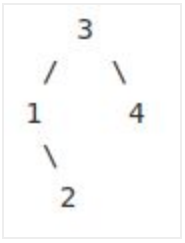
Sample Input 1

```
4
3 1 4 2
1 2 3
2 -1 4
3 -1 -1
4 -1 -1
1
```

Sample Output 1

```
1
```

Explanation 1



1 is the 1st smallest element in this BST

Instructions to create custom input for a Binary Tree

In order to specify a binary tree that can be used as custom input to the problem, you'll need to follow this convention.

- Line 1: Number of nodes in the Binary Tree (N)
- Line 2: N space separated node values. The position of the Nodes on this line will be used to refer to them in the below lines, starting from 1.
- Line 3 to N+2: Lines specifying the child nodes for each of the N nodes

Format of each line (space separated): Parent_node Left_child_node Right_child_node

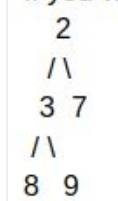
```
* Parent_node - Node at this Position on Line 2 is the Node to which we are assigning the Left and Right child here

* Left_child_node - Node at this position on Line 2 is the left child. Specify -1 if there is no Left child.

* Right_child_node - Node at this position on Line 2 is the right child. Specify -1 if there is no Right child.
```

Example1

If you want to create a Tree that looks like this:



Your input would be:

5	→ Number of Nodes
2 3 7 8 9	→ Node values
1 2 3	→ Node 1(value 2) and its child nodes (left child value 3 and right child value 7)
2 4 5	→ Node 2(value 3) and its child nodes (left child value 8 and right child value 9)
3 -1 -1	→ Node 3(value 7) and its child nodes (left and right child are Null i.e. Leaf Node)
4 -1 -1	→ Node 4(value 8) and its child nodes (left and right child are Null i.e. Leaf Node)
5 -1 -1	→ Node 5(value 9) and its child nodes (left and right child are Null i.e. Leaf Node)