Single Ton

Problem Submissions Leaderboard Discussions

For a given Binary Tree of type integer, print all the nodes without any siblings.

Input Format

The first and the only line of input will contain the node data, all separated by a single space. Since -1 is used as an indication whether the left or right node data exist for root, it will not be a part of the node data.

Constraints

1 <= N <= 10^5 Where N is the total number of nodes in the binary tree. Time Limit: 1 second

Output Format

The only line of output prints the node data in a top to down fashion with reference to the root. Node data in the right subtree will be printed first and then the left subtree. A single space will separate them all.

Sample Input 0

```
5 6 10 2 3 -1 -1 -1 -1 -1 9 -1 -1
```

Sample Output 0

```
9
```



```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

class node{
    int data;
    node prev;
    node next;
```

```
    node(int d){
        data=d;
        prev=null;
        next=null;
    }
}

public class Solution {


    public static void ns(node root) {
        if (root == null) {
            return;
        }

        if(root.prev==null && root.next!=null)
            System.out.print(root.next.data+" ");
        if(root.next==null&& root.prev!=null)
            System.out.print(root.prev.data+ " ");
        ns(root.next);
        ns(root.prev);
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        Scanner sc=new Scanner(System.in);
        Queue<node> q=new LinkedList<>();
        int val=sc.nextInt();
        if(val==-1) return;
        node nn=new node(val);
        node root=nn;
        q.add(nn);
        while(!q.isEmpty()){
            node e=q.poll();
            val=sc.nextInt();
            if(val!=-1){
            nn=new node(val);
```

```
48              e.prev=nn;
49              q.add(nn);}
50              val=sc.nextInt();
51              if(val!=-1){
52              nn=new node(val);
53              e.next=nn;
54              q.add(nn);}
55          }
56          ns(root);
57      }
58  }
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code