Take Input Level Wise of Binary Tree 1

Take Input Level Wise of Binary Tree and then print in the inorder format

Input Format

-

Constraints

-

Output Format

Inorder traversal of Binary tree

Sample Input 0

```
1 2 3 4 5 6 7 -1 -1 -1 -1 -1 -1 -1 -1
```

Sample Output 0

```
4 2 5 1 6 3 7
```

Java 7

```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 disp(node root) {
        if (root == null) {
            return;
        }

        Stack<node> stack = new Stack<>();
        node current = root;

        while (current != null || !stack.isEmpty()) {
```

```
            while (current != null) {
                stack.push(current);
                current = current.prev;
            }

            current = stack.pop();
            System.out.print(current.data + " ");

            current = current.next;
        }
    }

    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);
            e.prev=nn;
            q.add(nn);}
            val=sc.nextInt();
            if(val!=-1){
            nn=new node(val);
            e.next=nn;
            q.add(nn);}
        }
        disp(root);
    }
}
```

Line: 1 Col: 1

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