https://course.acciojob.com/idle?question=ac1af0fe-d5d6-4279-b47c-9787423ab256

- EASY
- Max Score: 30 Points

Binary Tree Postorder Traversal

Given the `root` of a binary tree, return *the postorder traversal of its nodes values*.

Note: You just need to implement the postorderTraversal() function. Input and output have been handled in the driver code.

Input Format

First line contains a string representing the tree as described below.

The values in the string are in the order of level order traversal of the tree where, numbers denote node values, and a character "N" denotes NULL child.

Output Format

Print the postorder traversal of the tree.

Example 1

Input

```
6 5 6 1 1 N 6
```

Output

```
1 1 5 6 6 6
```

Explanation

```
The given tree can be represented as:-
        6
       / \
```

```
      5   6
     / \   \
    1   1   6
The postorder traversal of the tree is :- 1 1 5 6 6 6
```

Example 2

Input

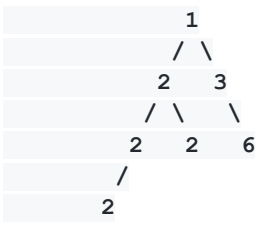
```
1 2 3 2 2 N 6 2 N N N N N N N
```

Output

```
2 2 2 2 6 3 1
```

Explanation

```
The given tree can be represented as: -
```



```
The postorder traversal of the tree is :- 2 2 2 2 6 3 1
```

Constraints

- **The number of nodes in the tree is in the range** `[0, 10^4]`.
- `-1000 <= Node.val <= 1000`
- **The depth of the tree will not exceed** `1000`.

Topic Tags

- **Trees**

My code

```
// n java
import java.util.*;
import java.lang.*;
import java.io.*;

class Node
{
    int data;
    Node next ,prev;

    Node(int data, Node next,Node prev)
    {
        this.data = data;
        this.next = next;
        this.prev = prev;
    }

    Node() {}
}
public class Main
{
  static Node insert(Node root,int n)
  {
    if(root==null)
    {
      root=new Node(n,null,null);
    return root;
```

```
    }
   else if(n< root.data)
      root.prev= insert( root.prev, n);
    else   if(n>root.data)
      root.next= insert( root.next, n);
    return root;
  }

static void postorder(Node root)
  {
    if(root !=null)
    {
      postorder(root.prev);
      postorder(root.next);
      System.out.print(root.data+" ");
    }
  }
        public static void main (String[] args) throws
java.lang.Exception
        {
                //your code here
      Scanner s=new Scanner(System.in);
      int n=s.nextInt();
      //int arr[]=new int[n];
      Node root=null;
      for(int i=0;i<n;i++)
      {
      int m=s.nextInt();
        root=insert( root, m);
      }
```

```
        postorder(root);
          }
}
```