

# Postorder

Given a binary tree which has  $N$  nodes, write a program to read the tree data, then **print all nodes of the tree in the post order (left, right, root)**

## Input

The first line contains the number of nodes of a tree,  $N$  ( $0 < N \leq 10^5$ ). The nodes are numbered from 1 to  $N$ . The root of the tree is 1.

The  $i$ th line of the next  $N$  lines contains two integers left and right ( $\text{left}, \text{right} \leq N$ ), respectively, the left child and the right child of the  $i$ th node. If left is less than or equal to 0, node  $i$  has no children on the left. If right is less than or equal to 0, node  $i$  has no right child.

*\* The tree is guaranteed to exist*

## Output

**All nodes of the tree in the post order, separated by spaces**

## Example

**Input:**

```
4
2 3
0 4
0 0
0 0
```

**Output:**

```
4 2 3 1
```

*Gợi ý*

```
class EIUEASPOST
{
    static void Main(string[] args)
    {
        var nNode = NextInt();
        var nodes = ReadTree(nNode);
        // You code here
    }

    static void PrintPostOrder(/*Parameters*/)
    {
    }

    static Node[] ReadTree(int nNode)
    {
        Node[] nodes = new Node[nNode];
        for (var i = 0; i < nNode; i++)
        {
            nodes[i] = new Node(i + 1);
        }
        for (var i = 0; i < nNode; i++)
```

```
{
    var leftIndex = NextInt();
    nodes[j].Left = leftIndex > 0 ? nodes[leftIndex - 1] : null;
    var rightIndex = NextInt();
    nodes[j].Right = rightIndex > 0 ? nodes[rightIndex - 1] : null;
}
return nodes;
}
```

```
class Node
{
    public int Id;
    public Node Left;
    public Node Right;

    public Node(int id)
    {
        Id = id;
    }
}
}
```