XOR Tree

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

You are given a binary with N nodes. For every node X, print the XOR of the values of all the nodes in the subtree of X including the node X itself in postorder form.

Input

1<= N <=10^5
1<= Value of a node <=10^9
You are given a binary tree in **level order** form where -1 represents null node.
The first line contains the root node.
The second line contains the left child and right child of the root.
And so on..

The following code snippet can be used to build the tree in java: (the Node class consists of the fields- Node left, Node right and int data)

```
public static Node takeInput(){
Scanner sc = new Scanner(System.in);
Queue<Node> queue = new LinkedList<Node>();
int rootData = sc.nextInt();
if(rootData == -1){
return null;
}
Node root = new Node(rootData);
queue.add(root);


while(!queue.isEmpty()){
Node front = queue.poll();
int leftChild = sc.nextInt();
if(leftChild!=-1){
Node left = new Node(leftChild);
front.left =  left;
queue.add(left);
}
int rightChild = sc.nextInt();
if(rightChild!=-1){
Node right = new Node(rightChild);
front.right = right;
queue.add(right);
}
}

return root;
}
```

Output

Print the XOR value for each node in the postorder form.

Examples

standard input	standard output

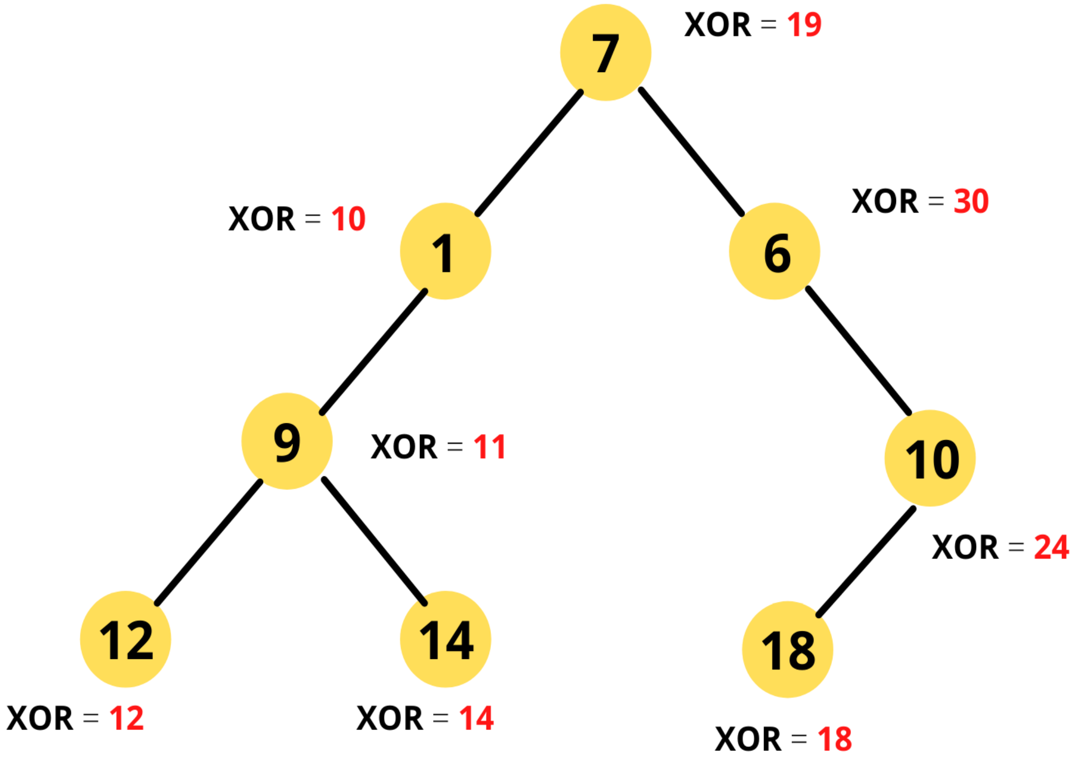
1
2 3
4 5 6 7
-1 -1 -1 -1 -1 -1 -1 -1
	4 5 3 6 7 2 0

7
1 6
9 -1 -1 10
12 14 18 -1
-1 -1 -1 -1 -1 -1
	12 14 11 10 18 24 30 19

Note

Consider Example 2

The tree formed is as follows:



Output in postorder form: **12 14 11 10 18 24 30 19**