import java.util.LinkedList;

import java.util.Queue;

import java.util.Stack;

public class BinaryTree {

private TreeNode root;

private class TreeNode {

private TreeNode left;

private TreeNode right;

private int data; // Can be any generic type

public TreeNode(int data) {

this.data = data;

}

}

public void levelOrder() {

if(root == null) {

return;

}

Queue<TreeNode> queue = new LinkedList<>();

queue.offer(root);

while(!queue.isEmpty()) {

TreeNode temp = queue.poll();

System.out.print(temp.data + " ");

if(temp.left != null) {

queue.offer(temp.left);

}

if(temp.right != null) {

queue.offer(temp.right);

}

}

}

public void createBinaryTree() {

TreeNode first = new TreeNode(1);

TreeNode second = new TreeNode(2);

TreeNode third = new TreeNode(3);

TreeNode fourth = new TreeNode(4);

TreeNode fifth = new TreeNode(5);

TreeNode sixth = new TreeNode(6);

TreeNode seventh = new TreeNode(7);

root = first;

first.left = second;

first.right = third;

second.left = fourth;

second.right = fifth;

third.left = sixth;

third.right = seventh;

}

public static void main(String[] args) {

BinaryTree bt = new BinaryTree();

bt.createBinaryTree();

bt.levelOrder();

}

}