import java.util.\*;

public class BinaryTreeSerialization {

// Definition for a binary tree node.

static class TreeNode {

int val;

TreeNode left;

TreeNode right;

TreeNode(int val) {

this.val = val;

this.left = null;

this.right = null;

}

}

// Serialize the binary tree to a string

public static String serialize(TreeNode root) {

StringBuilder sb = new StringBuilder();

serializeHelper(root, sb);

return sb.toString();

}

// Helper function to serialize using pre-order traversal

private static void serializeHelper(TreeNode node, StringBuilder sb) {

if (node == null) {

sb.append("null,");

return;

}

sb.append(node.val).append(",");

serializeHelper(node.left, sb);

serializeHelper(node.right, sb);

}

// Deserialize the string back to a binary tree

public static TreeNode deserialize(String data) {

String[] nodes = data.split(",");

Queue<String> queue = new LinkedList<>(Arrays.asList(nodes));

return deserializeHelper(queue);

}

// Helper function to rebuild the tree recursively from the queue

private static TreeNode deserializeHelper(Queue<String> queue) {

String val = queue.poll();

if (val.equals("null")) {

return null;

}

TreeNode node = new TreeNode(Integer.parseInt(val));

node.left = deserializeHelper(queue);

node.right = deserializeHelper(queue);

return node;

}

// Helper function to print the tree in pre-order (for testing)

private static void printPreOrder(TreeNode node) {

if (node == null) {

System.out.print("null ");

return;

}

System.out.print(node.val + " ");

printPreOrder(node.left);

printPreOrder(node.right);

}

public static void main(String[] args) {

// Create a simple binary tree

TreeNode root = new TreeNode(1);

root.left = new TreeNode(2);

root.right = new TreeNode(3);

root.left.left = new TreeNode(4);

root.left.right = new TreeNode(5);

// Serialize the tree

String serializedTree = serialize(root);

System.out.println("Serialized Tree: " + serializedTree);

// Deserialize the tree

TreeNode deserializedRoot = deserialize(serializedTree);

System.out.print("Deserialized Tree (Pre-order): ");

printPreOrder(deserializedRoot);

}

}