**EASY 1 [EXAMPLE 3]**

Input: s = "luffy is still joyboy"

Output: 6

Explanation: The last word is "joyboy" with length 6.

public class length {

public static int lengthOfWord(String s) {

s = s.trim();

int spaceIndex = s.IndexOf(' ');

int length = s.length() - spaceIndex - 1;

return length;

}

public static void main(String[] args) {

String input = "luffy is still joyboy";

int result = lengthOfWord(input);

System.out.println("Length of the word: " + result);

}

}

**MEDIUM 1[EXAMPLE 2]**

class TreeNode {

int val;

TreeNode left, right;

public TreeNode(int val) {

this.val = val;

this.left = this.right = null;

}

}

public class LowestCommonAncestor {

public TreeNode lowestCommonAncestor(TreeNode root, TreeNode p, TreeNode q) {

if (root == null || root == p || root == q) {

return root;

}

TreeNode leftLCA = lowestCommonAncestor(root.left, p, q);

TreeNode rightLCA = lowestCommonAncestor(root.right, p, q);

if (leftLCA != null && rightLCA != null) {

return root;

}

return (leftLCA != null) ? leftLCA : rightLCA;

}

public static void main(String[] args) {

TreeNode root = new TreeNode(6);

root.left = new TreeNode(2);

root.right = new TreeNode(8);

root.left.left = new TreeNode(0);

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

root.right.left = new TreeNode(7);

root.right.right = new TreeNode(9);

root.left.right.left = new TreeNode(3);

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

TreeNode p = new TreeNode(2);

TreeNode q = new TreeNode(4);

LowestCommonAncestor lcaFinder = new LowestCommonAncestor();

TreeNode result = lcaFinder.lowestCommonAncestor(root, p, q);

System.out.println("Lowest Common Ancestor: " + result.val);

}

}

**HARD 2[EXAMPLE 2]**

public class ReverseConcatenate {

public static String reverseConcatenate(String s) {

// Check if the string is empty

if (s.isEmpty()) {

return s;

}

StringBuilder reversed = new StringBuilder(s).reverse();

return reversed.toString() + s;

}

public static void main(String[] args) {

String input = "abcd";

String result = reverseConcatenate(input);

System.out.println("Output: " + result);

}

}