**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | | **01-07-2020** | | | | **Name:** | **RACHANA K N** | |
| **Sem & Sec** | | **6th -B** | | | | **USN:** | **4AL17CS070** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | |  | | | | | |
| **Max. Marks** | | | **-** | **Score** | | | **-** | |
| **Pre-placement Training Summary** | | | | | | | | |
| **Topic** | **---** | | | | | | | |
| **Faculty** | -- | | | | **Duration** | | | **--** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement**: | | | | | | | | |
| **Status: Solved** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | **DAILY\_STATUS** | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

**PROGRAMS**

Write a program to find given two trees are mirror or not.

class Node

{

int data;

Node left, right;

public Node(int data)

{

this.data = data;

left = right = null;

}

}

public class BinaryTree

{

Node a, b;

boolean areMirror(Node a, Node b)

{

if (a == null && b == null)

return true;

if (a == null || b == null)

return false;

return a.data == b.data

&& areMirror(a.left, b.right)

&& areMirror(a.right, b.left);

}

public static void main(String[] args)

{

BinaryTree tree = new BinaryTree();

Node a = new Node(1);

Node b = new Node(1);

a.left = new Node(2);

a.right = new Node(3);

a.left.left = new Node(4);

a.left.right = new Node(5);

b.left = new Node(3);

b.right = new Node(2);

b.right.left = new Node(5);

b.right.right = new Node(4);

if (tree.areMirror(a, b) == true)

System.out.println("Yes");

else

System.out.println("No");

}

}

