**Symmetric Tree:-**

Given a Binary Tree. Check whether it is Symmetric or not, i.e. whether the binary tree is a **Mirror image of itself** or not.

**Example 1:**

**Input:**

5

/ \

1 1

/ \

2 2

**Output:** True

**Explanation:** Tree is mirror image of

itself i.e. tree is symmetric

**Example 2:**

**Input:**

5

/ \

10 10

/ \ \

20 20 30

**Output:** False

**Your Task:**  
You don't need to read input or print anything. Your task is to complete the function **isMirror()** which takes the root of the Binary Tree as its input and returns True if the given Binary Tree is a same as the Mirror image of itself. Else, it returns False.

**Expected Time Complexity:** O(N).  
**Expected Auxiliary Space:** O(Height of the Tree).

**Constraints:**  
1<=Number of nodes<=100