```
package binarytree;

// Java program to check is binary tree is symmetric or not
class Node41
{
    int key;
    Node41 left, right;

    Node41(int item)
    {
        key = item;
        left = right = null;
    }
}

class BinaryTree65
{
    Node41 root;

    // returns true if trees with roots as root1 and root2 are mirror
    boolean isMirror(Node41 node1, Node41 node2)
    {
        // if both trees are empty, then they are mirror image
        if (node1 == null && node2 == null)
            return true;

        // For two trees to be mirror images, the following three
        // conditions must be true
        // 1 - Their root node's key must be same
        // 2 - left subtree of left tree and right subtree
        //   of right tree have to be mirror images
        // 3 - right subtree of left tree and left subtree
        //   of right tree have to be mirror images
        if (node1 != null && node2 != null && node1.key == node2.key)
            return (isMirror(node1.left, node2.right)
                    && isMirror(node1.right, node2.left));

        // if neither of the above conditions is true then
        // root1 and root2 are mirror images
        return false;
    }

    // returns true if the tree is symmetric i.e
    // mirror image of itself
    boolean isSymmetric(Node41 node)
    {
        // check if tree is mirror of itself
        return isMirror(root, root);
    }

    // Driver program
    public static void main(String args[])
    {
        BinaryTree65 tree = new BinaryTree65();
        tree.root = new Node41(1);
        tree.root.left = new Node41(2);
        tree.root.right = new Node41(2);
        tree.root.left.left = new Node41(3);
        tree.root.left.right = new Node41(4);
        tree.root.right.left = new Node41(4);
        tree.root.right.right = new Node41(3);
        boolean output = tree.isSymmetric(tree.root);
        if (output == true)
            System.out.println("1");
        else
            System.out.println("0");
```

```
    }
}

// This code has been contributed by Mayank Jaiswal
```