Binary Tree Problems: Mirror and Symmetric Check

1. Mirror Binary Tree

```
// Mirror a Binary Tree
class Solution {
    // Function to convert a binary tree into its mirror tree.
    void mirror(Node node) {
        if (node == null) return;

        // Swap the left and right child
        Node temp = node.left;
        node.left = node.right;
        node.right = temp;

        // Recursively call mirror on left and right subtrees
        mirror(node.left);
        mirror(node.right);
    }
}
```

2. Check Symmetric Binary Tree

```
// Check if Binary Tree is Symmetric
class Solution {
    public boolean isSymmetric(Node root) {
        return root == null || isSymmetricHelp(root.left, root.right);
    }

    private boolean isSymmetricHelp(Node left, Node right) {
        if (left == null || right == null) {
            return left == right; // true if both are null
        }
        if (left.data != right.data) {
            return false; // values don't match
        }
        return isSymmetricHelp(left.left, right.right) &&
               isSymmetricHelp(left.right, right.left);
    }
}
```