**CSE220 Lab quiz05 (Binary Tree)**

**SET A:**

Function is\_identical(p, q):

// If both nodes are null, they are identical

if (p == null && q==null) {

return true;

}

// If one is null but the other isn't, they are not identical

if (p == null or q == null) {

return false;

}

// Check if current nodes have the same value

// and recursively check left and right subtrees

return (p.element == q.element &&

is\_identical(p.left, q.left) &&

is\_identical(p.right, q.right))

| **Criteria** | **Marks** | **Description** |
| --- | --- | --- |
| Base case: both null | 2 | Returns True when both nodes are None. |
| Base case: one null | 2 | Returns False when only one of the nodes is None. |
| Value comparison | 3 | Correctly compares the element values of the current nodes. |
| Recursive subtree check | 4 | Recursively checks both left and right subtrees. |
| Logical structure and flow | 2 | Conditions are ordered and applied clearly and correctly. |
| Syntax and clarity | 2 | Uses proper syntax, readable naming, and consistent style. |

**SET B:**

// Function to check if a tree is symmetric

Function is\_symmetric(root):

// A tree with no root is symmetric

if (root == null) {

return true;

}

// Use helper function to compare left and right subtrees

return is\_mirror(root.left, root.right)

// Helper function to check if two subtrees are mirror images

Function is\_mirror(t1, t2):

// If both nodes are null, they are mirrors

if (t1 == null && t2 == null) {

return true;

}

// If one is null but the other isn't, they are not mirrors

if (t1 == null or t2 == null) {

return false;

}

// Check if current nodes have the same value

// and their children are mirrors in opposite direction

return (t1.element == t2.element &&

is\_mirror(t1.left, t2.right) &&

is\_mirror(t1.right, t2.left))

| **Criteria** | **Marks** | **Description** |
| --- | --- | --- |
| Base case handling | 2 | Checks if root is None and returns True. |
| Use of helper function | 2 | Calls a separate helper (e.g., is\_mirror) to compare subtrees. |
| Correct recursive logic | 4 | Properly compares node values and mirror positions recursively. |
| Null checks in helper | 2 | Returns True if both nodes are None, False if only one is. |
| Logical structure and flow | 2 | Follows clear, correct order of conditions and recursion. |
| Syntax and clarity | 3 | Uses correct syntax, readable code, and consistent style. |