WORKSHEET 1

Student Name: Hardik Jhamb **UID:** 22BCS16504

Branch: BE-CSE **Section/Group:** 22BCS_NTPP-602-A

Semester: 6th **Date of Performance:**

Subject Name: AP LAB - II **Subject Code:** 22CSP-351

1. **Aim:** Remove duplicates from a sorted array
   Contains duplicate
   Two Sum

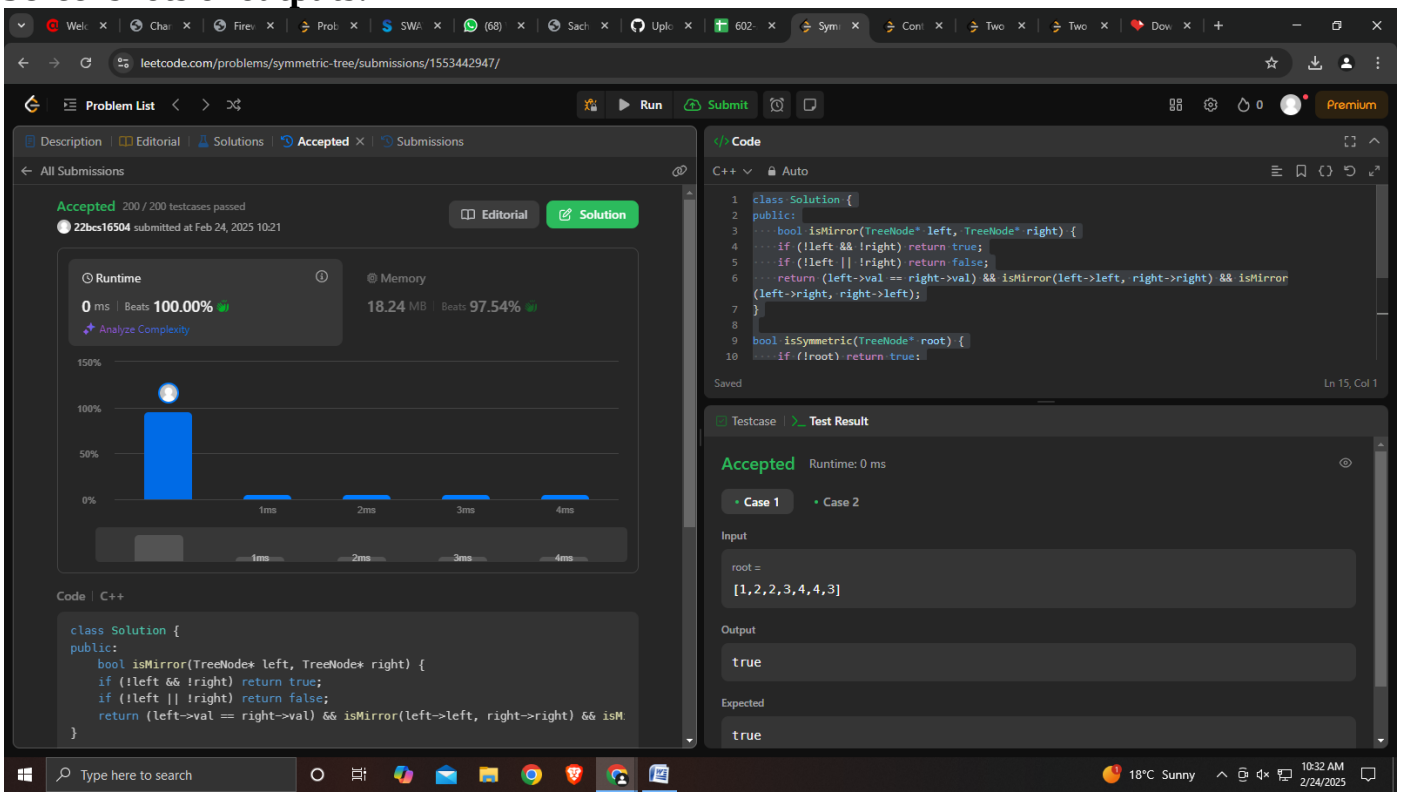
2. **Source Code:**

```
class Solution {
public:
    bool isMirror(TreeNode* left, TreeNode* right) {
    if (!left && !right) return true;
    if (!left || !right) return false;
    return (left->val == right->val) && isMirror(left->left, right->right) && isMirror(left-
>right, right->left);
}

bool isSymmetric(TreeNode* root) {
    if (!root) return true;
    return isMirror(root->left, root->right);
}

};
```

3. Screenshots of outputs:

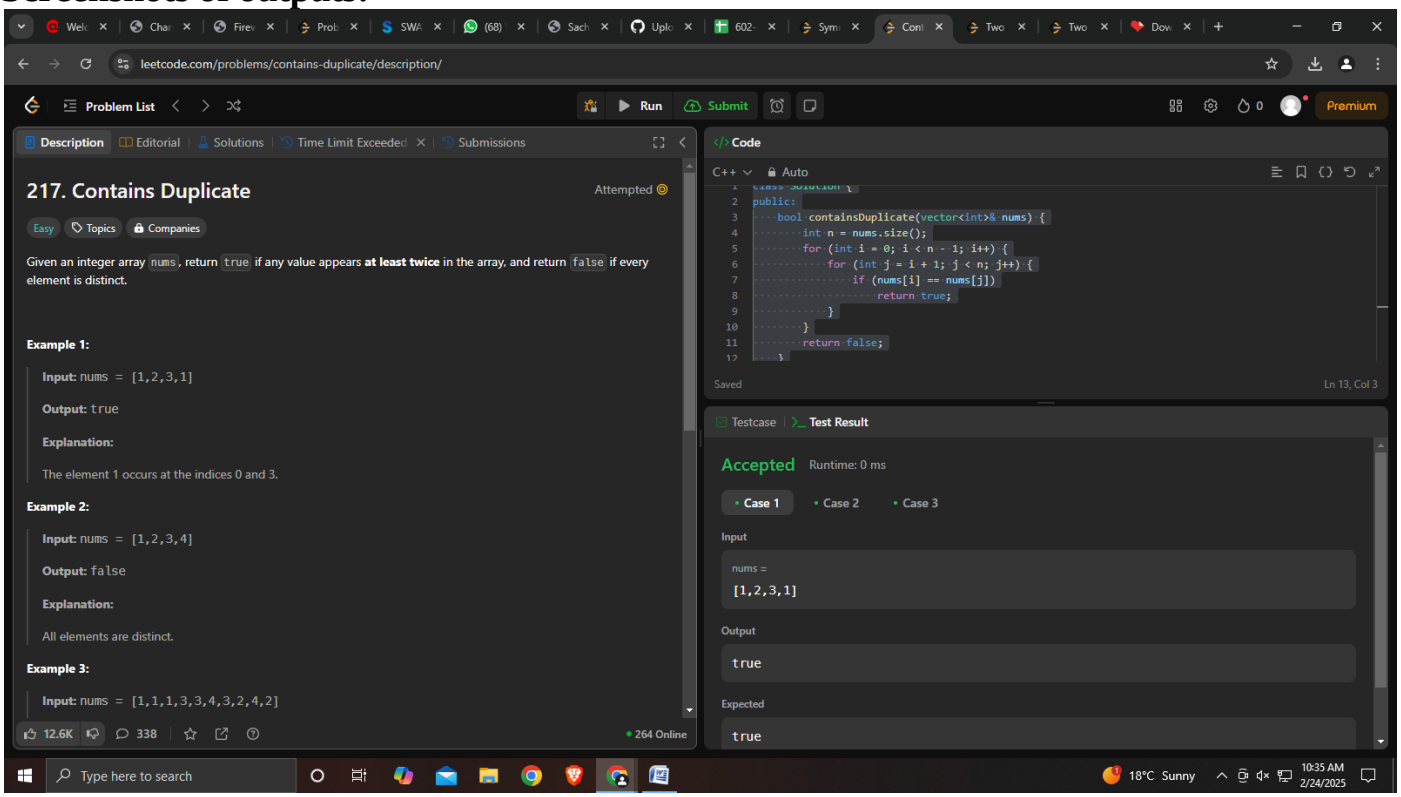


2. Aim: Given an integer array nums, return true if any value appears **at least twice** in the array, and return false if every element is distinct.

Source Code:

```
class Solution {
public:
    bool containsDuplicate(vector<int>& nums) {
        int n = nums.size();
        for (int i = 0; i < n - 1; i++) {
            for (int j = i + 1; j < n; j++) {
                if (nums[i] == nums[j])
                    return true;
            }
        }
        return false;
    }
};
```

Screenshots of outputs:



3. Given an array of integers nums and an integer target, return *indices of the two numbers such that they add up to target*.

You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice.

You can return the answer in any order

Source Code:

```
class Solution{
public:
vector<int> twoSum(vector<int>& nums, int target) {
 for (int i = 0; i < nums.size(); ++i) {
  for (int j = i + 1; j < nums.size(); ++j) {
 if (nums[i] + nums[j] == target) {
   return {i, j};
     }
   }
  }

    return {};
}
};
```

4. Screenshots of outputs: