Experiment No. - 4

Student Name: Vivek Kumar
Branch: BE-CSE(LEET)
Semester: 6th
Subject Name: Competitive coding - II

UID: 21BCS8129
Section/Group: 20BCS-ST-801/B
Date of Performance: 07/03/2023
Subject Code: 20CSP-351

1. **Aim/Overview of the practical:**

   Q.1 Longest Duplicate Substring.

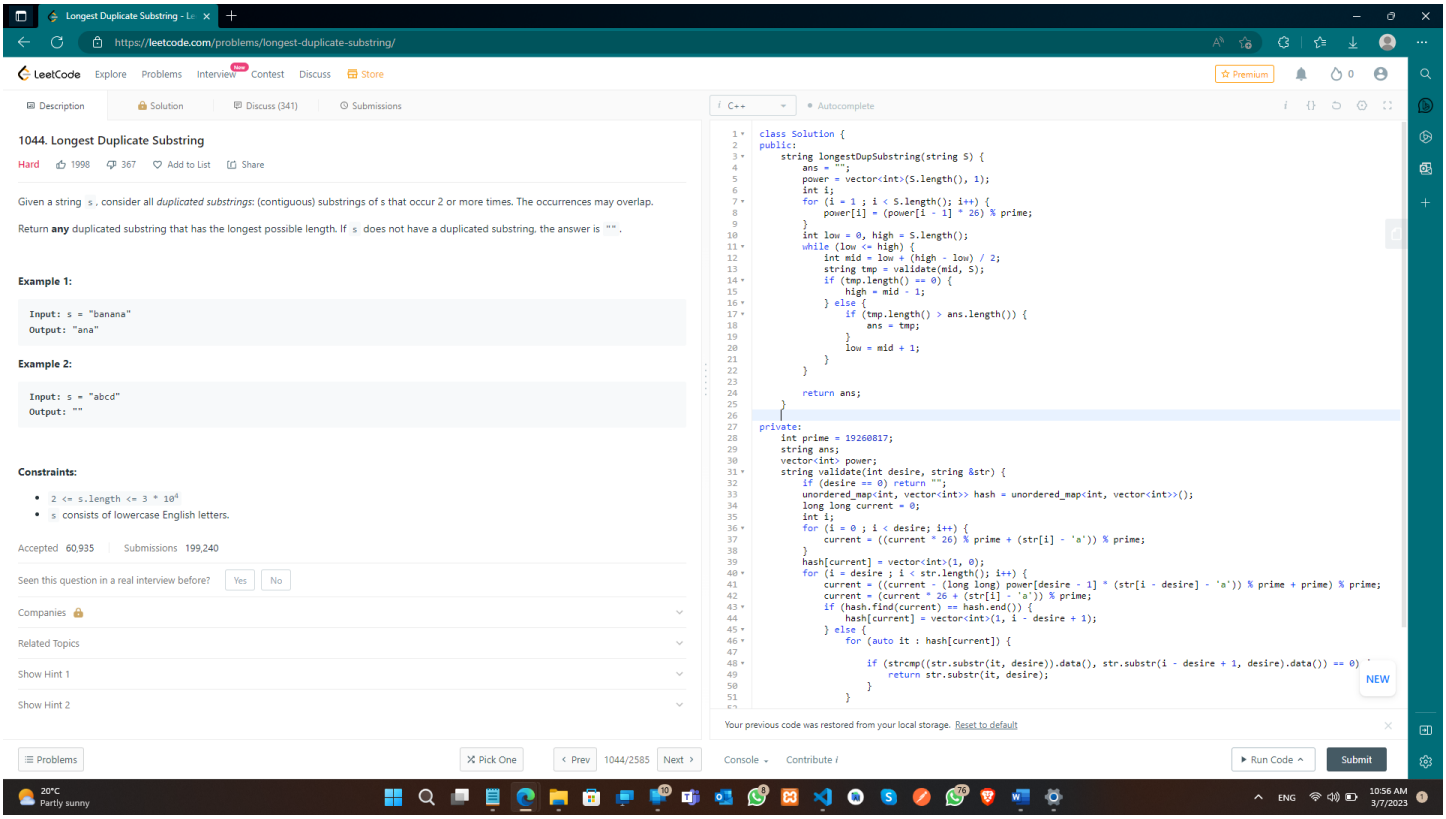
   https://leetcode.com/problems/longest-duplicate-substring/

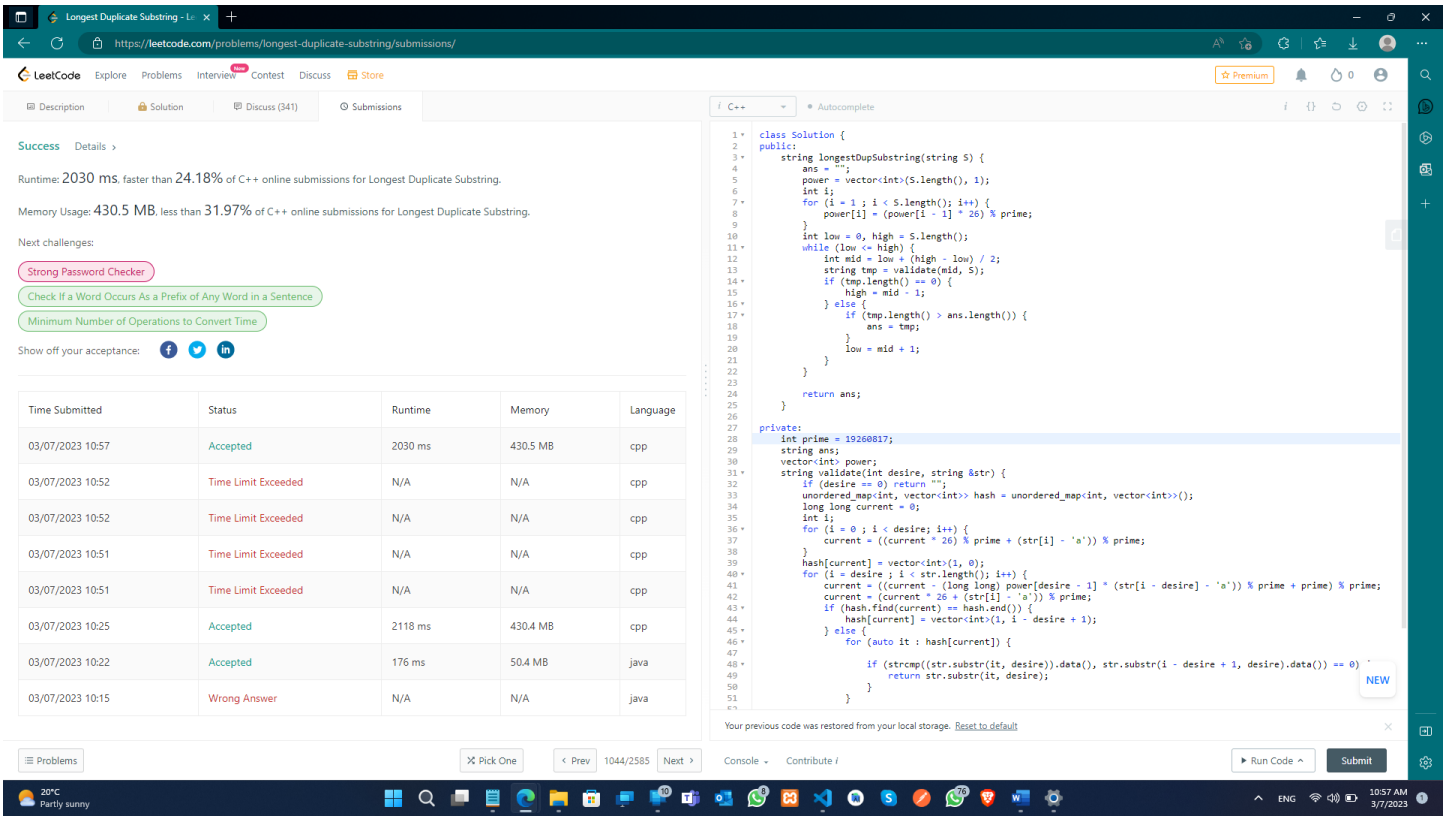
2. **Apparatus / Simulator Used:**
   - Windows 7 or above
   - Google Chrome
3. **Objective:**
   - To understand the concept of B Search
   - To implement the concept of Rabin Karp.
4. **Code:**

```
class Solution {
public:
    string longestDupSubstring(string S) {
        ans = "";
        power = vector<int>(S.length(), 1);
        int i;
        for (i = 1 ; i < S.length(); i++) {
            power[i] = (power[i - 1] * 26) % prime;
        }
        int low = 0, high = S.length();
        while (low <= high) {
            int mid = low + (high - low) / 2;
            string tmp = validate(mid, S);
            if (tmp.length() == 0) {
                high = mid - 1;
            } else {
                if (tmp.length() > ans.length()) {
                    ans = tmp;
                }
                low = mid + 1;
            }
        }

        return ans;
    }

private:
    int prime = 19260817;
    string ans;
```

```
        vector<int> power;
        string validate(int desire, string &str) {
            if (desire == 0) return "";
            unordered_map<int, vector<int>> hash = unordered_map<int, vector<int>>();
            long long current = 0;
            int i;
            for (i = 0 ; i < desire; i++) {
                current = ((current * 26) % prime + (str[i] - 'a')) % prime;
            }
            hash[current] = vector<int>(1, 0);
            for (i = desire ; i < str.length(); i++) {
                current = ((current - (long long) power[desire - 1] * (str[i - desire] - 'a')) % prime + prime) %
prime;
                current = (current * 26 + (str[i] - 'a')) % prime;
                if (hash.find(current) == hash.end()) {
                    hash[current] = vector<int>(1, i - desire + 1);
                } else {
                    for (auto it : hash[current]) {

                        if (strcmp((str.substr(it, desire)).data(), str.substr(i - desire + 1, desire).data()) == 0) {
                            return str.substr(it, desire);
                        }
                    }

                    hash[current].push_back(i - desire + 1);
                }
            }

            return "";
        }
    };
```

5. Result/Output/Writing Summary:





Submitted By: Vivek Kumar

1. **Aim/Overview of the practical:**

   Q.2 Missing Number
   https://leetcode.com/problems/missing-number/

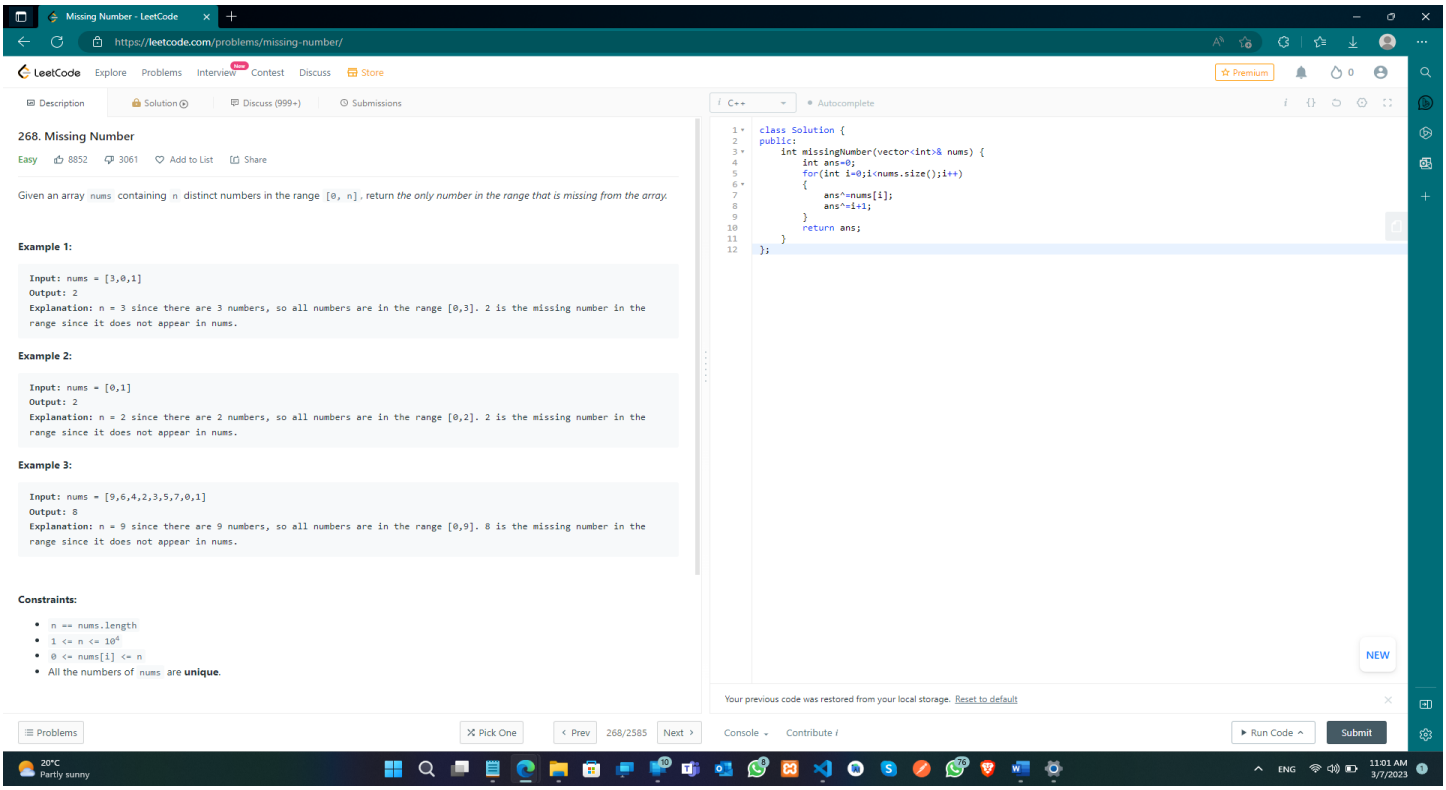
2. **Apparatus / Simulator Used:**
   - Windows 7 or above
   - Google Chrome

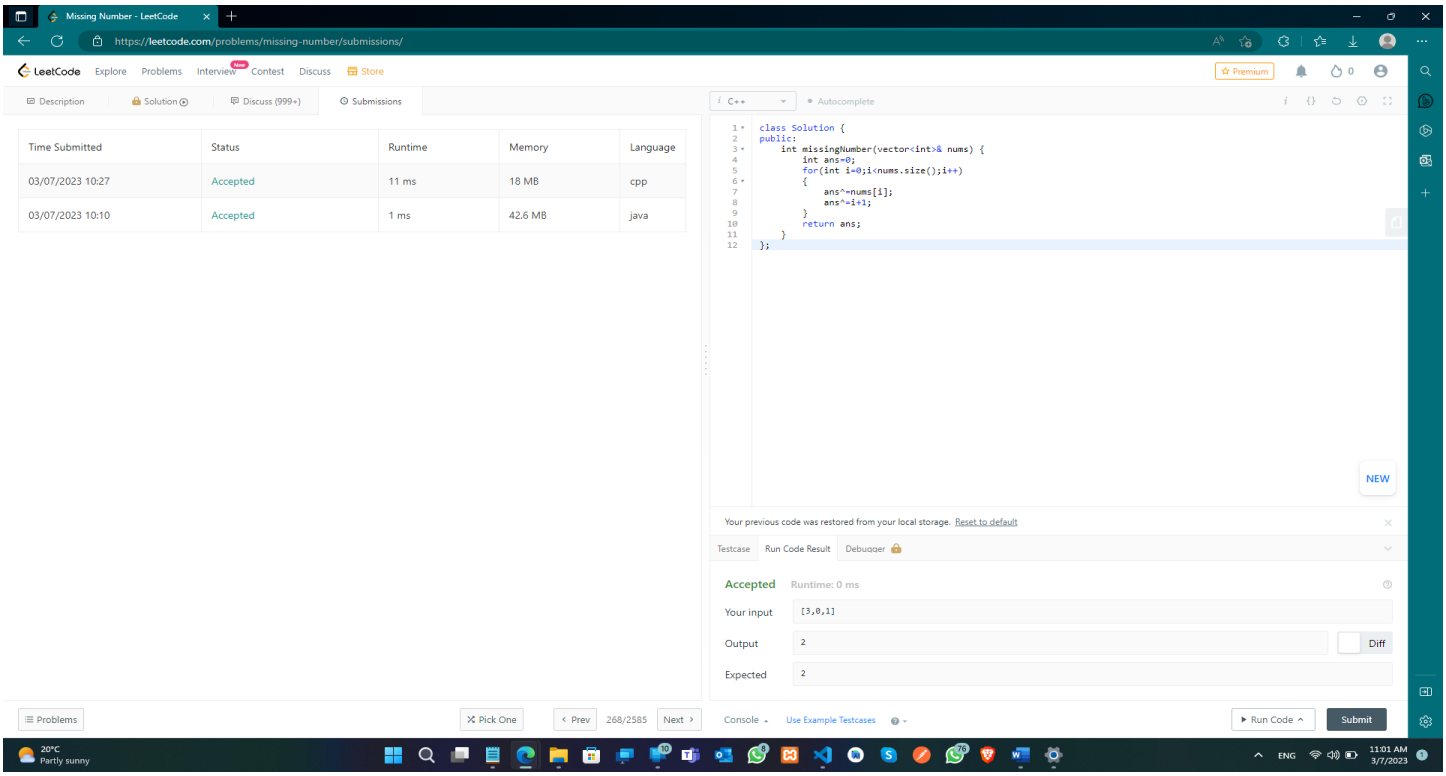
3. **Objective:**
   - To understand the concept of Looping.
   - To implement the concept of calculate the sum.

4. **Code:**

```
class Solution {
public:
    int missingNumber(vector<int>& nums) {
        int ans=0;
        for(int i=0;i<nums.size();i++)
        {
            ans^=nums[i];
            ans^=i+1;
        }
        return ans;
    }
};
```

5. Result/Output/Writing Summary:





Learning outcomes (What I have learnt):

- Learned the concept of cheapest flights within k stops.
- Learnt about Array in Vector and Its iteration.

Submitted By: Vivek Kumar