Experiment 4.1

Student Name: Sahil Kaundal
UID: 21BCS8197
Branch: BE CSE (Lateral Entry)
Section/Group: 616/A
Semester: 6th
Date of Performance: 20/03/2023
Subject Name: CC-2 Lab
Subject Code: 20CSP-351

1. Aim/Overview of the practical:

Missing Number

Given an array nums containing n distinct numbers in the range [0, n], return *the only number in the range that is missing from the array.*

https://leetcode.com/problems/missing-number/

2. Apparatus / Simulator Used:

- Windows 7 or above
- Google Chrome

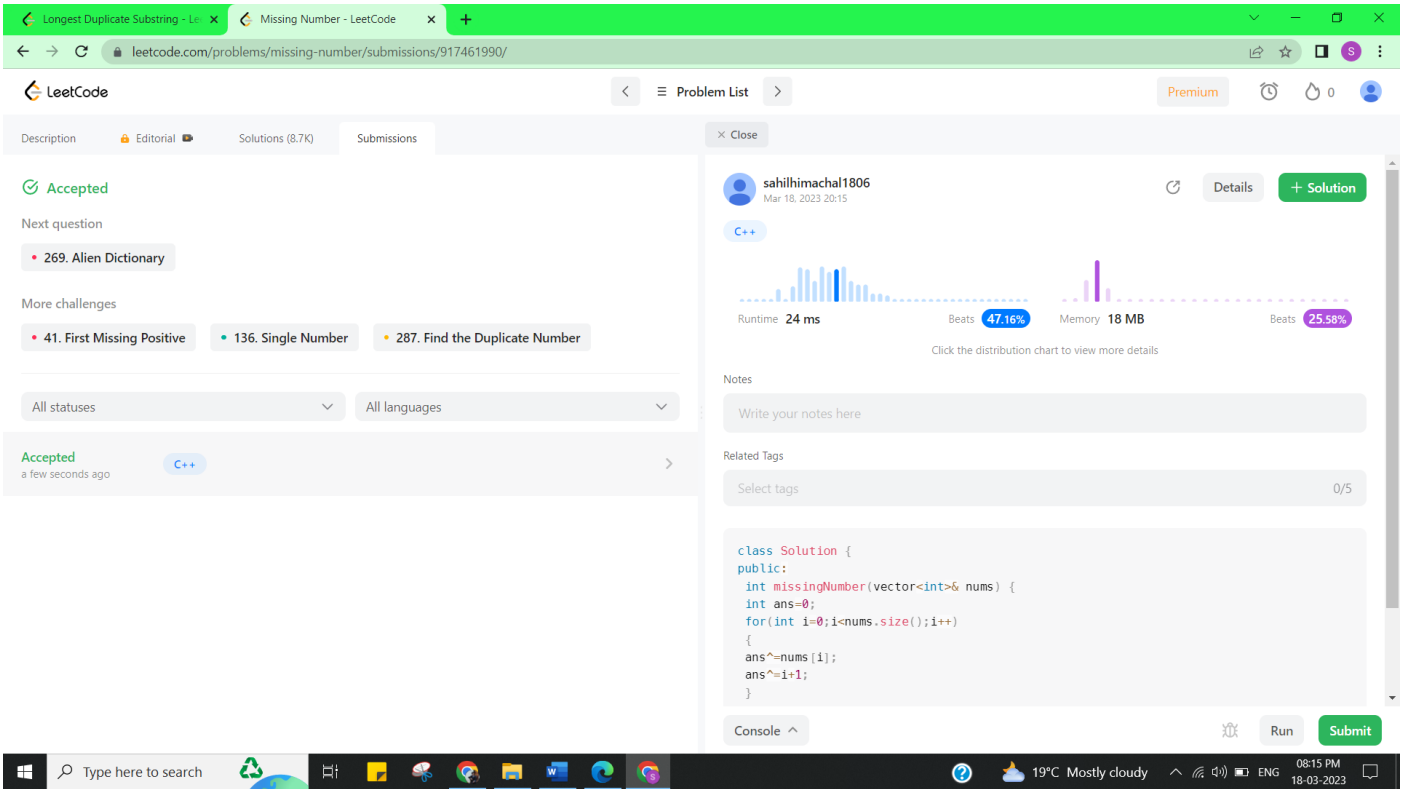
3. Objective:

- 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;
 }
};
```

4. Result/Output/Writing Summary:



Experiment 4.2

1. Aim/Overview of the practical:

Longest Duplicate Substring

Given a string s, consider all *duplicated substrings*: (contiguous) substrings of s that occur 2 or more times. The occurrences may overlap.

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

2. Apparatus / Simulator Used:

- Windows 7 or above
- Google Chrome

3. Objective:

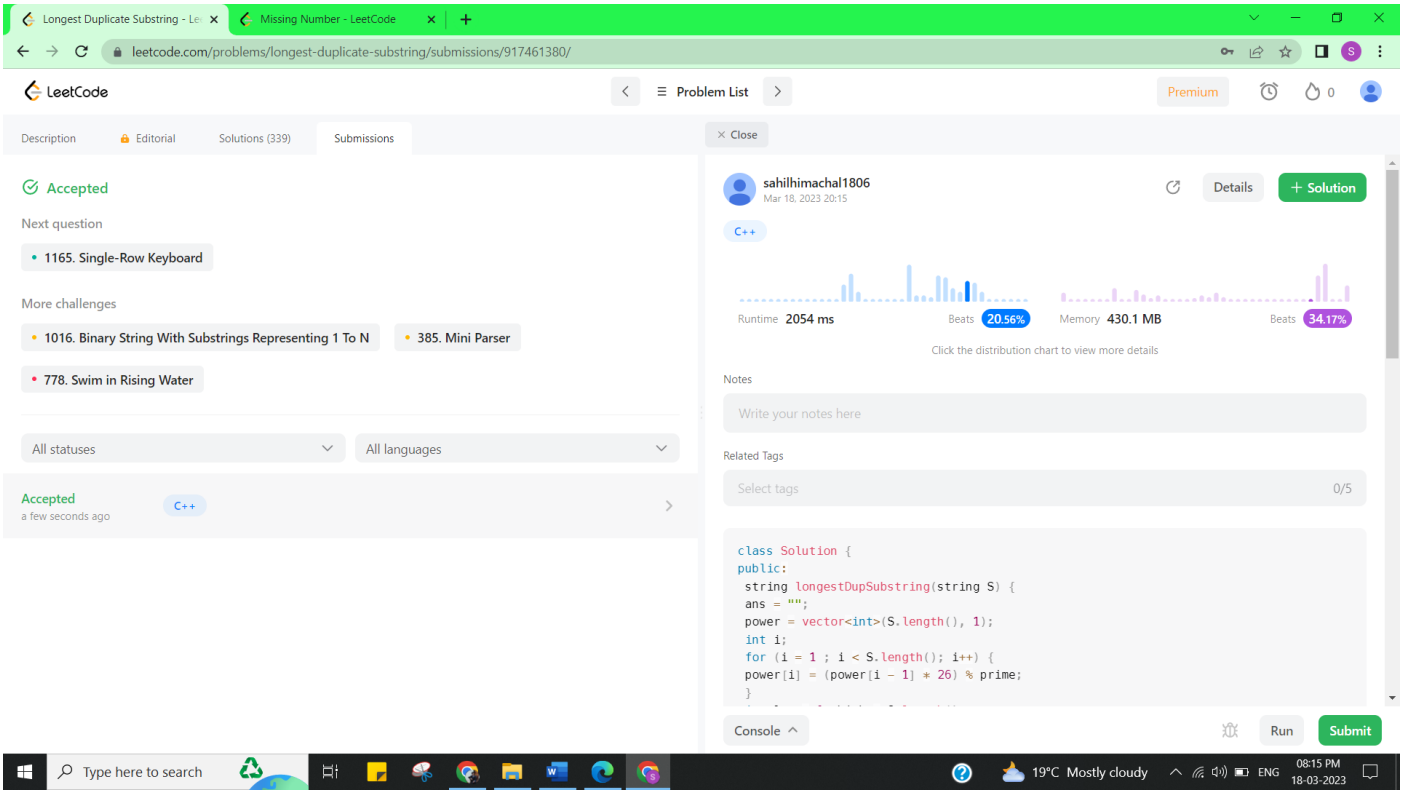
- To understand the concept of B Search
- To implement the concept of Rabin Karp.

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:



Learning outcomes (What I have learnt):

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

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			