**Problem - 1**

1. **Aim of the Experiment :**

Given a string s, find the first non-repeating character in it and return its index. If it does not exist, return -1.

1. **Objective of the Experiment :**

To find the first non-repeating character in it and return its index. If it does not exist, return -1.

1. **Problem Description:**

A string s is given and we have to find the first non repeating character in the string and then return index of that character. If the character does not exist then return -1.

We will use the map data structure and map each character with their occurrences. Then we will traverse the given string and check the value for each character in the map and if the corresponding occurrence is 1 then that character will be the required answer.

1. **Algorithm:**

Step 1: Define unordered\_map<char,int> mp.

Step 2: Initialize int n = s.length().

Step 3: Traverse the whole string and set mp[s[i]]++.

Step 4: Traverse the string and compare the occurrences of each character.

if (mp[s[i]] == 1) {

return i;

}

Step 5: If no character have occurrence equal to 1, then return -1.

1. **Code :**

class Solution {

public:

int firstUniqChar(string s) {

int index=-1;

int n=s.size();

map<char,int> mpp;

for(int i=0;i<n;i++){

mpp[s[i]]++;

}

for(int i=0;i<n;i++){

if(mpp[s[i]]>1){

continue;

}else{

return i;

}

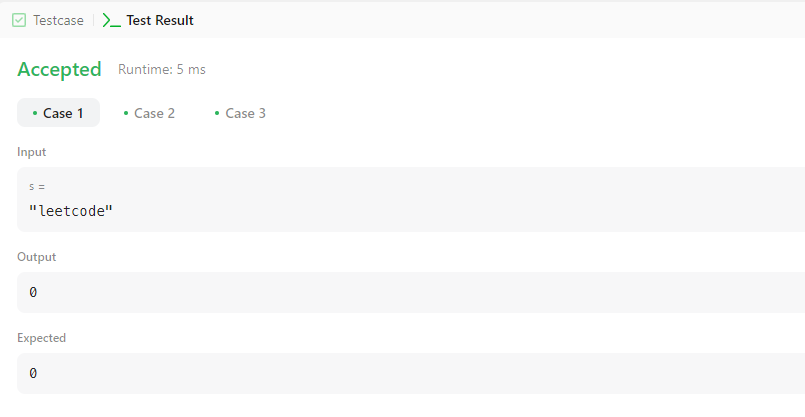
}

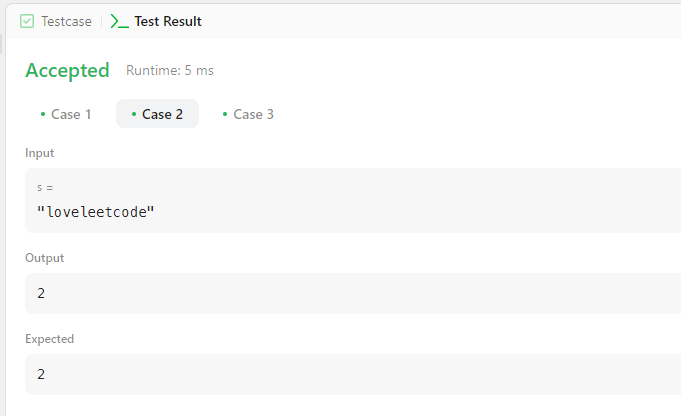
return index;

}

};

1. **Output:**

****

****

**Learning outcomes (What I have learnt):**

1. **Learned how to use the map() in C++.**