



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment-1.4

Student Name: Aayush Gurung

Branch: CSE

Semester: 6

Subject Name: CC LAB

UID:20BCS5323

Section/Group:DM_607(A)

Date of Performance:31-03-2023

Subject Code: 20CSP-351

Aim:

To demonstrate the concept of Hashing

Objective

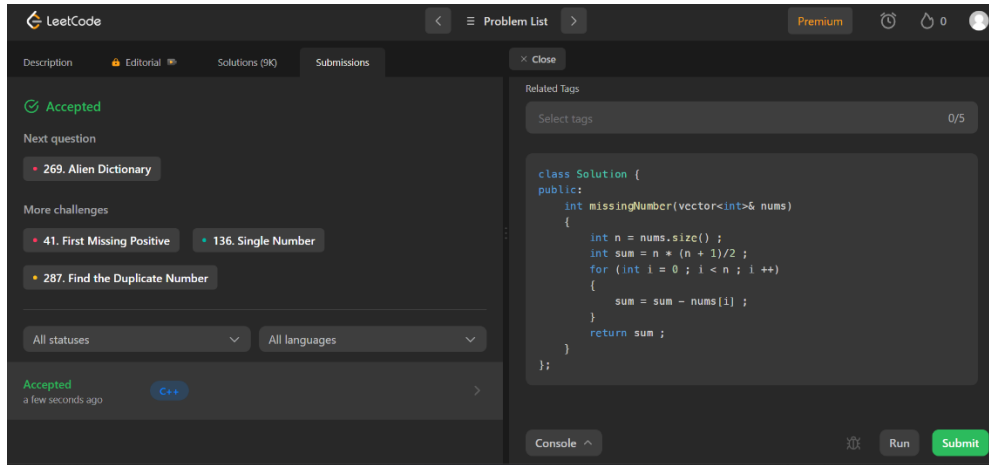
Problem 1:

Missing number

Code:

```
class Solution {
public:
    int missingNumber(vector<int>& nums)
    {
        int n = nums.size() ;
        int sum = n * (n + 1)/2 ;
        for (int i = 0 ; i < n ; i++)
        {
            sum = sum - nums[i] ;
        }
        return sum ;
    }
};
```

Output:



Problem 2:

Longest Duplicate substring

Code:

```
#define ull unsigned long long
```

```
class Solution {
```

```
public:
```

```
    string ans="";
```

```
    bool solve(int len, string &s, ull power){
```

```
        int start = 0, end = len;
```

```
        unordered_set<ull> st;
```

```
        ull curHash = 0;
```

```
        for(int i=0; i<len; ++i){
```

```
            curHash = (curHash*131 + (s[i]));
```

```
        }
```

```
        st.insert(curHash);
```

```
        for(int j=len; j<s.size(); ++j){
```

```
            curHash = ((curHash - power*(s[start])));
```

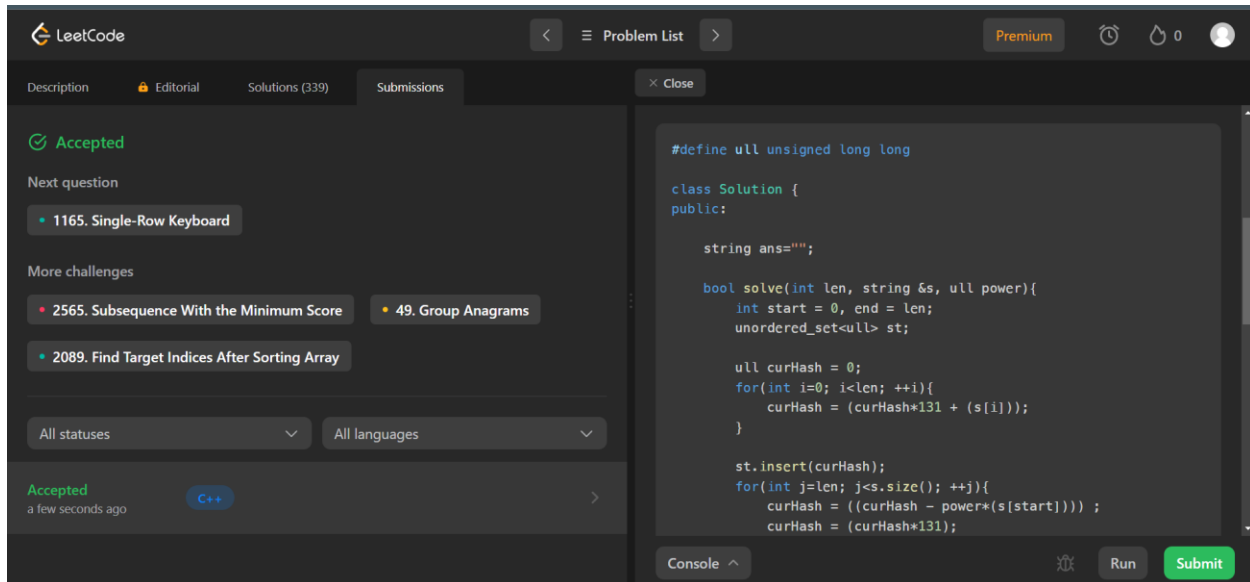
```
        curHash = (curHash*131);
        curHash = (curHash + (s[j]));
        start++;
        if(st.find(curHash) != st.end()){
            string curS = s.substr(start,len);
            if(curS.size()>ans.size()){
                ans = curS;
            }
            return true;
        }
        st.insert(curHash);
    }
    return false;
}

void binary(int l, int r, string &s, vector<ull>& power){
    if(l>r) return;
    int mid = l+(r-l)/2;
    if(solve(mid+1,s,power[mid])){
        l=mid+1;
    }else{
        r=mid-1;
    }
    binary(l,r,s,power);
}

string longestDupSubstring(string s) {
    int n = s.size();
```

```
vector<ull> power(n,1);  
for(int i=1;i<n;++i){  
    power[i]=(power[i-1]*131);  
}  
binary(0,n-1,s,power);  
return ans;  
}  
};
```

Output:



The screenshot shows the LeetCode interface for a submission. The left sidebar displays the 'Accepted' status, the problem name '1165. Single-Row Keyboard', and a list of 'More challenges' including '2565. Subsequence With the Minimum Score', '49. Group Anagrams', and '2089. Find Target Indices After Sorting Array'. The main area shows the C++ code for the solution, which defines a vector 'power' and a function 'binary' to calculate the result. The code is as follows:

```
#define ull unsigned long long  
  
class Solution {  
public:  
  
    string ans="";  
  
    bool solve(int len, string &s, ull power){  
        int start = 0, end = len;  
        unordered_set<ull> st;  
  
        ull curHash = 0;  
        for(int i=0; i<len; ++i){  
            curHash = (curHash*131 + (s[i]));  
        }  
  
        st.insert(curHash);  
        for(int j=len; j<s.size(); ++j){  
            curHash = ((curHash - power*(s[start])));  
            curHash = (curHash*131);  
        }  
    }  
};
```

At the bottom, there are buttons for 'Console', 'Run', and 'Submit'.