

Experiment No. - 2

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: 21/02/2023
Subject Code: 20CSP-351

1. Aim/Overview of the practical:

Q.1 Find the Index of the First Occurance in a String.

<https://leetcode.com/problems/find-the-index-of-the-first-occurrence-in-a-string/>

2. Apparatus / Simulator Used:

- Windows 7 or above
- Google Chrome

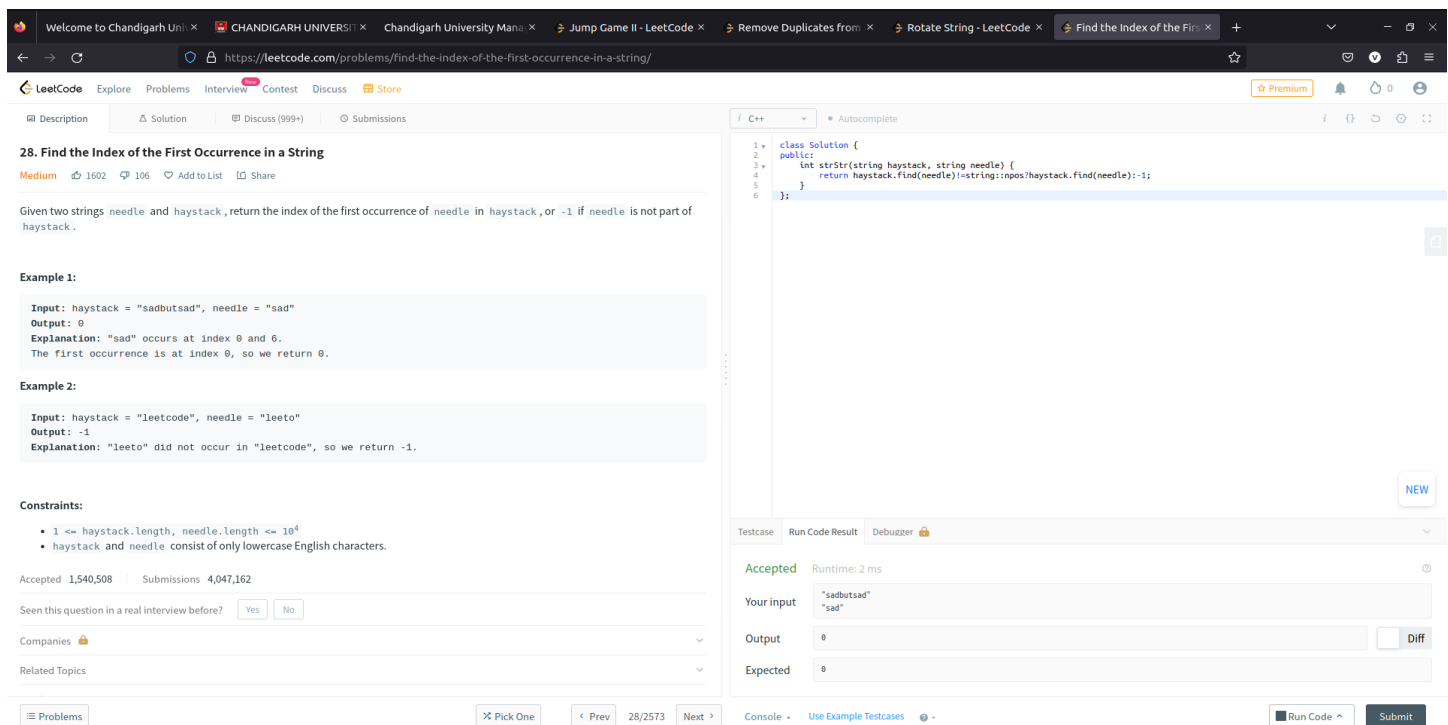
3. Objective:

- To understand the concept of String
- To implement the concept of Occurance Count.

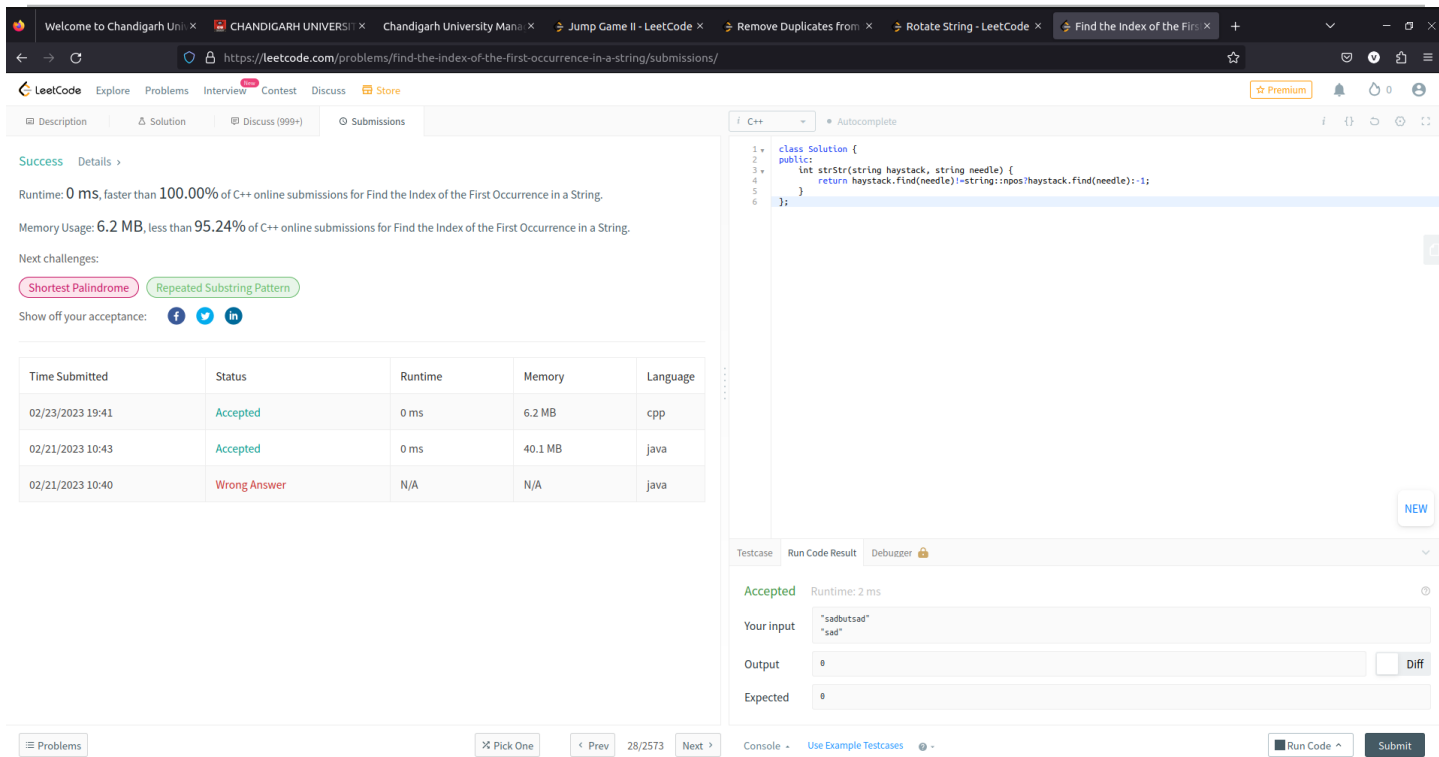
4. Code:

```
class Solution {
public:
    int strStr(string haystack, string needle) {
        return haystack.find(needle)!=string::npos?haystack.find(needle):-1;
    }
};
```

5. Result/Output/Writing Summary:



The screenshot shows the LeetCode interface for the problem "Find the Index of the First Occurrence in a String". The problem is categorized as Medium. The description states: "Given two strings needle and haystack, return the index of the first occurrence of needle in haystack, or -1 if needle is not part of haystack." Example 1 shows input haystack = "sadbutsad", needle = "sad", output = 0. Example 2 shows input haystack = "leetcode", needle = "leeto", output = -1. The constraints are: 1 ≤ haystack.length, needle.length ≤ 10⁴; haystack and needle consist of only lowercase English characters. The solution code in C++ is shown in the editor, and the test results show it is "Accepted" with a runtime of 2 ms. The input "sadbutsad" and "sad" yields an output of 0, which matches the expected output.



Success Details

Runtime: 0 ms, faster than 100.00% of C++ online submissions for Find the Index of the First Occurrence in a String.

Memory Usage: 6.2 MB, less than 95.24% of C++ online submissions for Find the Index of the First Occurrence in a String.

Next challenges: [Shortest Palindrome](#) [Repeated Substring Pattern](#)

Show off your acceptance: [f](#) [t](#) [i](#)

Time Submitted	Status	Runtime	Memory	Language
02/23/2023 19:41	Accepted	0 ms	6.2 MB	cpp
02/21/2023 10:43	Accepted	0 ms	40.1 MB	java
02/21/2023 10:40	Wrong Answer	N/A	N/A	java

```

1 class Solution {
2 public:
3     int strStr(string haystack, string needle) {
4         return haystack.find(needle) != string::npos ? haystack.find(needle) : -1;
5     }
6 };

```

Testcase Run Code Result Debuzzer

Accepted Runtime: 2 ms

Your input: "sadbutsad", "sad"

Output: 0

Expected: 0

Run Code Submit

1. Aim/Overview of the practical:

Q.2 Rotate String

<https://leetcode.com/problems/rotate-string/>

2. Apparatus / Simulator Used:

- Windows 7 or above
- Google Chrome

3. Objective:

- To understand the concept of Rotation
- To implement the concept of String.

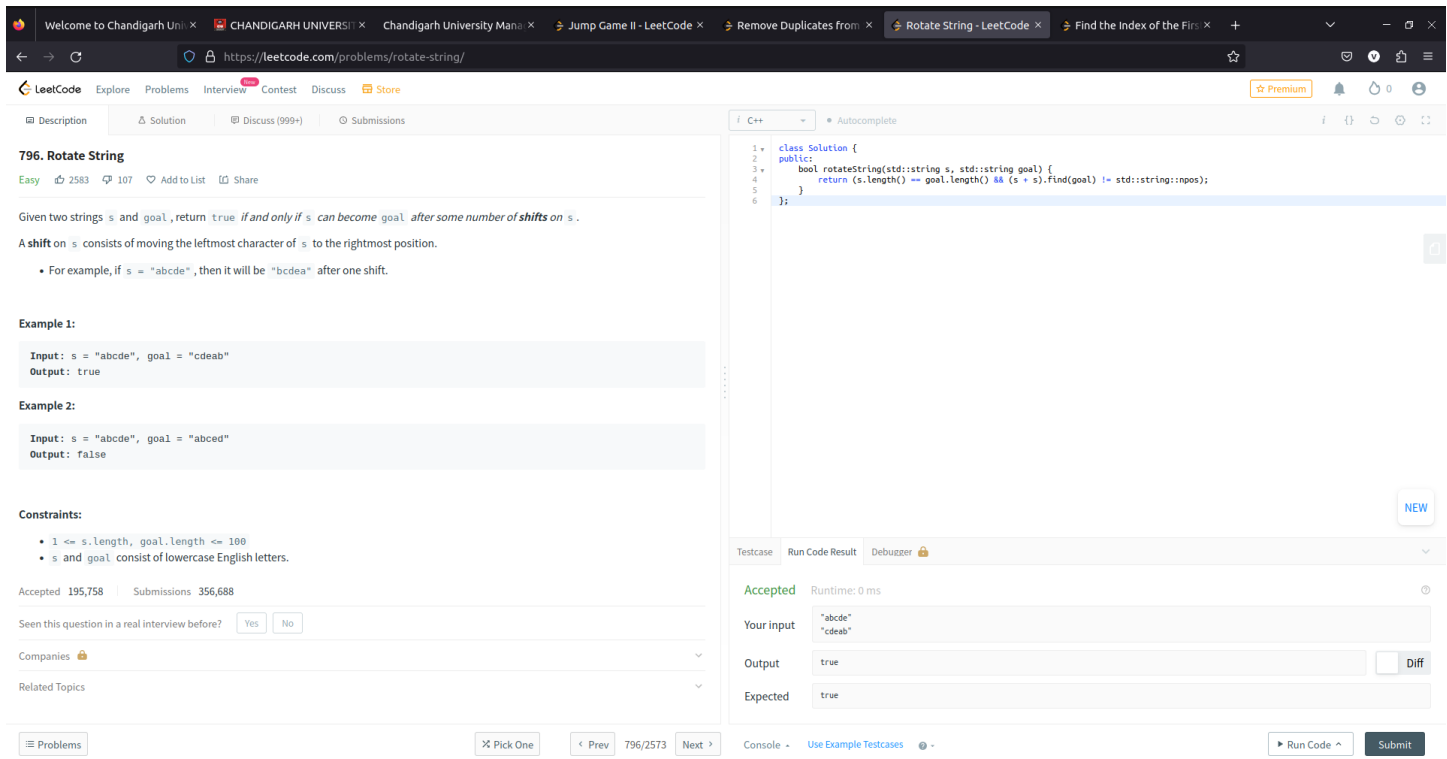
4. Code:

```

class Solution {
public:
    bool rotateString(std::string s, std::string goal) {
        return (s.length() == goal.length() && (s + s).find(goal) != std::string::npos);
    }
};

```

5. Result/Output/Writing Summary:



796. Rotate String

Easy 2583 107 Add to List Share

Given two strings *s* and *goal*, return *true* if and only if *s* can become *goal* after some number of *shifts* on *s*.

A *shift* on *s* consists of moving the leftmost character of *s* to the rightmost position.

- For example, if *s* = "abcde", then it will be "bcdea" after one shift.

Example 1:

Input: *s* = "abcde", *goal* = "cdeab"

Output: *true*

Example 2:

Input: *s* = "abcde", *goal* = "abcd"


Output: *false*


Constraints:

- 1 ≤ *s*.length, *goal*.length ≤ 100
- s* and *goal* consist of lowercase English letters.


Accepted 195,758 Submissions 356,688

Seen this question in a real interview before? ☐ Yes ☐ No

Companies 

Related Topics 

```
1 class Solution {
2 public:
3     bool rotateString(std::string s, std::string goal) {
4         return (s.length() == goal.length() && (s + s).find(goal) != std::string::npos);
5     }
6 }
```



Testcase Run Code Result Debuzzer 

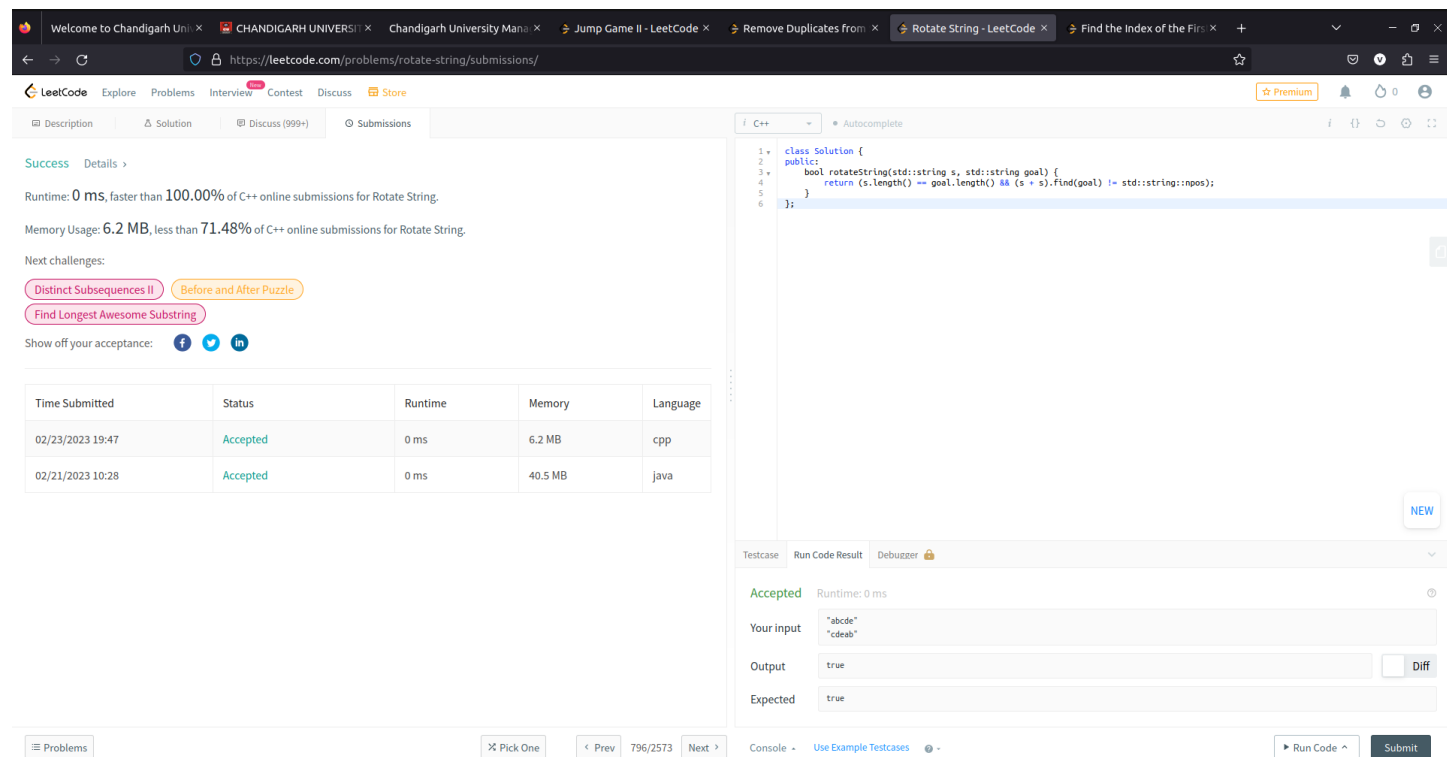
Accepted Runtime: 0 ms

Your input "abcde" "cdeab"

Output true

Expected true

Console  Use Example Testcases 



Success Details >




Runtime: 0 ms, faster than 100.00% of C++ online submissions for Rotate String.

Memory Usage: 6.2 MB, less than 71.48% of C++ online submissions for Rotate String.

Next challenges:


[Distinct Subsequences II](#) [Before and After Puzzle](#)

[Find Longest Awesome Substring](#)

Show off your acceptance:   

Time Submitted	Status	Runtime	Memory	Language
02/23/2023 19:47	Accepted	0 ms	6.2 MB	cpp
02/21/2023 10:28	Accepted	0 ms	40.5 MB	java

```
1 class Solution {
2 public:
3     bool rotateString(std::string s, std::string goal) {
4         return (s.length() == goal.length() && (s + s).find(goal) != std::string::npos);
5     }
6 }
```



Testcase Run Code Result Debuzzer 

Accepted Runtime: 0 ms

Your input "abcde" "cdeab"

Output true

Expected true

Console  Use Example Testcases 

Learning outcomes (What I have learnt):

- Learned the concept of String.
- Learnt about Array in Occurance and Rotation.

Evaluation Grid (To be created per the faculty's SOP and Assessment guidelines):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Worksheet completion including writing learning objectives/Outcomes. (To be submitted at the end of the day).		
2.	Post-Lab Quiz Result.		
3.	Student Engagement in Simulation/Demonstration/Performance and Controls/Pre-Lab Questions.		
	Signature of Faculty (with Date):	Total Marks Obtained:	