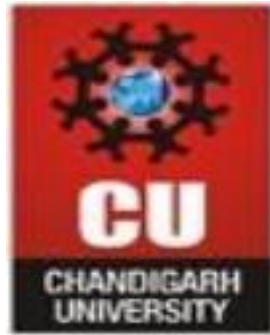


# **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



## **UNIVERSITY INSTITUTE OF ENGINEERING Department of Computer Science & Engineering**

**Subject Name:**

**Subject Code: 20CSP351**

**Submitted to:**

Er. Vipasha Sharma

**Submitted by:**

Name: Kanishk Soni

UID: 20BCS9398

Section: 20BCS\_DM\_708

Group: B

# **DEPARTMENT OF**

# **COMPUTER SCIENCE & ENGINEERING**

## **INDEX**

<b>Ex. No</b>	<b>List of Experiments</b>	<b>Conduct (MM: 12)</b>	<b>Viva (MM: 10)</b>	<b>Record (MM: 8)</b>	<b>Total (MM: 30)</b>	<b>Remarks/Signature</b>
1.1	Arrays, Queues, Stack and Linked List					
1.2	String-Matching algorithms					
1.3						
2.1						
2.2						
2.3						
2.4						
3.1						
3.2						
3.3						

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## Experiment-1.2

**Student Name:** Kanishk Soni

**Branch:** BE-CSE

**Semester:** 6<sup>th</sup>

**Subject Name:** Competitive Coding-II

**UID:** 20BCS9398

**Section/Group:** 20BCS-DM\_708B

**Date of Performance:** 23-02-2023

**Subject Code:** 20CSP-351

**AIM:** To demonstrate the concept of String-Matching algorithms

**Problem1:** Rotate string

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

**Program Code:**

```
class Solution {
    public boolean rotateString(String s, String goal) {

        if(s.length()==goal.length()){
            String str = s+s;
            int j=0;
            int k=goal.length();
            for(int i=0; i<goal.length(); i++){
                String sub = str.substring(j,k);
                if(sub.equals(goal))
                    return true;
                j++;
                k++;
            }
        }
        else{
            return false;
        }
        return false;
    }
}
```

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
}  
}
```

Output:

The screenshot shows the LeetCode interface for problem 797, "All Paths From Source to Target". The submission is marked as "Accepted" and was submitted by user "taabish19" on Feb 22, 2023, at 11:05. The submission details show a runtime of 1 ms, memory usage of 42.2 MB, and it beats 38.44% of other submissions. The code is written in Java and is as follows:

```
class Solution {  
    public boolean rotateString(String s, String goal) {  
        if(s.length()==goal.length()){  
            String str = s+s;  
            int j=0;  
            int k=goal.length();  
            for(int i=0; i<goal.length(); i++){  
                String sub = str.substring(i,k);  
                if(sub.equals(goal)) return true;  
            }  
        }  
        return false;  
    }  
}
```

Problem2: Repeated string match

<https://leetcode.com/problems/repeated-string-match/>

Program Code:

```
class Solution {  
    public int repeatedStringMatch(String a, String b) {  
        String copy = "";  
        int count=0;  
        while(copy.length()<b.length()){  
            copy+=a;  
            count++;  
        }  
        if(copy.indexOf(b)>=0)  
            return count;  
        if((copy+a).indexOf(b)>=0)  
            return count+1;  
        return -1;  
    }  
}
```

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
        return ++count;

        return -1;
    }
}
```

Output:

The screenshot shows the LeetCode interface for the problem "Repeated Substring Pattern" (459). The submission is by user "taabish19" and is marked as "Accepted". The submission details show a runtime of 364 ms, beating 16.81% of other submissions, and a memory usage of 113.8 MB, beating 5.19% of other submissions. The code is written in Java and implements a solution using a while loop to check for repeated patterns in the input string.

LeetCode

Problem List

Premium

0

Description Discussion (9) Solutions (735) Submissions

Accepted

Next question

687. Longest Univalue Path

More challenges

459. Repeated Substring Pattern

All statuses

All languages

Accepted

a few seconds ago

Java

Accepted

Feb 20, 2023

Java

taabish19

Feb 22, 2023 11:06

Details

+ Solution

Java

Runtime 364 ms

Beats 16.81%

Memory 113.8 MB

Beats 5.19%

Click the distribution chart to view more details

Notes

Write your notes here

Related Tags

Select tags

0/5

```
class Solution {
    public int repeatedStringMatch(String a, String b) {

        String copy = "";
        int count=0;
        while(copy.length()<b.length()){
            copy+=a;
            count++;
        }
    }
}
```

Console

Run

Submit