

 Article marked as read.

Check if Strings are Rotations

Given a string s1 and a string s2, write a snippet to say whether s2 is a rotation of s1? (eg given s1 = ABCD and s2 = CDAB, return true, given s1 = ABCD, and s2 = ACBD , return false)



Algorithm: areRotations(str1, str2)

1. Create a temp string and store concatenation of str1 to str1 in temp.

```
temp = str1.str1
```

2. If str2 is a substring of temp then str1 and str2 are rotations of each other.

Example:

```
str1 = "ABACD"
```

```
str2 = "CDABA"
```

```
temp = str1.str1 = "ABACDABACD"
```

Since str2 is a substring of temp, str1 and str2 are rotations of each other.

C++

Java

```
// Java program to check if two given strings are rotations of  
// each other
```

```
class StringRotation
```

```
{
```

```
    /* Function checks if passed strings (str1 and str2)  
    are rotations of each other */
```

Track Progress

43 of 67 Complete. (65%)

// There lengths must be same and str2 must be



✓ Article marked as read.

}



Articles

// Driver method

public static void main (String[] args)



Videos

{

String str1 = "AACD";

String str2 = "ACDA";



Problems

if (areRotations(str1, str2))

System.out.println("Strings are rotations of each other");

else



Quiz

System.out.printf("Strings are not rotations of each other");

}

}

// This code is contributed by munjal



Output

Strings are rotations of each other

Method 2(Using STL):

Algorithm :

1. If the size of both the strings is not equal, then it can never be possible.
2. Push the original string into a queue **q1**.
3. Push the string to be checked inside another queue **q2**.



4. Keep popping **q2**'s and pushing it back into it till the number of such operations are less than the size of the string.

Track Progress

43 of 67 Complete. (65%)

Dash



Article marked as read.



Articles



Videos



Problems



Quiz



```
import java.util.*;

class GFG{
static boolean check_rotation(String s, String goal)
{
    if (s.length() != goal.length())
        ;
    Queue<Character> q1 = new LinkedList<>();
    for (int i = 0; i < s.length(); i++) {
        q1.add(s.charAt(i));
    }
    Queue<Character> q2 = new LinkedList<>();
    for (int i = 0; i < goal.length(); i++) {
        q2.add(goal.charAt(i));
    }
    int k = goal.length();
    while (k>0) {
        k--;
        char ch = q2.peek();
        q2.remove();
        q2.add(ch);
        if (q2.equals(q1))
            return true;
    }
    return false;
}

public static void main(String[] args)
{
    String s1 = "ABCD";
    String s2 = "CDAB";
    if (check_rotation(s1, s2))
        System.out.print(s2+ " is a rotated form of " + s1

```

Menu



Track Progress

43 of 67 Complete. (65%)

System.out.print(s2+ " is not a rotated form of " + s1



✓ Article marked as read.

```

System.out.print(s3+ " is a rotated form of " + s1
                +"\n");
else
    System.out.print(s3+ " is not a rotated form of " + s1
                    +"\n");
}
}

```



// This code is contributed by gauravrajput1



Output

```

CDAB is a rotated form of ABCD
ACBD is not a rotated form of ABCD

```

Time Complexity: Time complexity of this problem depends on the implementation of strstr function.

If the implementation of strstr is done using KMP matcher then complexity of the above program is $O(n_1 + n_2)$ where n_1 and n_2 are lengths of strings. KMP matcher takes $O(n)$ time to find a substring in a string of length n where length of substring is assumed to be smaller than the string.

Method 3:

Algorithm:

1. Find all the positions of first character of original string in the string to be checked.

2. For every position found, consider it to be the starting index of the string to be checked.

3. Beginning from the new starting index, compare both strings and check whether

Track Progress

43 of 67 Complete. (65%)

(Suppose original string to be **s1**, string to be checked be **s2**, **n** is length of strings and **j**

 All

✓ Article marked as read.

4. Repeat 3rd step for all positions found.

C++

Java

 Articles

 Videos

 Problems

 Quiz

```
/*package whatever //do not write package name here */

import java.io.*;
import java.util.*;

class GFG
{
    // java program to check if two strings are rotation of each other or no
    static boolean checkString(String s1, String s2, int indexFound, int Siz
    {
        for(int i=0;i<Size;i++)
        {

            //check whether the character is equal or not
            if(s1.charAt(i) != s2.charAt((indexFound+i)%Size))return false;

            // %Size keeps (indexFound+i) in bounds,
            // since it ensures it's value is always less than Size
        }

        return true;
    }

    // Driver code
    public static void main(String args[])
    {
        String s1="abcd";
        String s2="cdab";
```

Menu

Track Progress

43 of 67 Complete. (65%)

System.out.println("s2 is not a rotation on s1");



Article marked as read.

ArrayList<Integer>indexes = new ArrayList<Integer>(); //store oc



int Size = s1.length();

Articles

char firstChar = s1.charAt(0);



Videos

```
for(int i=0;i<Size;i++)
{
    if(s2.charAt(i)==firstChar)
    {
        indexes.add(i);
    }
}
```



Problems



Quiz

Get 90% Refund

Courses

Tutorials

Jobs

Practice

Contests



boolean isRotation=false;

// check if the strings are rotation of each other for every oc
for(int idx: indexes)

```
{
    isRotation = checkString(s1, s2, idx, Size);
```

```
    if(isRotation)
        break;
}
```

```
if(isRotation)System.out.println("s2 is rotation of s1");
else System.out.println("s2 is not a rotation of s1");
```

```
}
```

```
}
```

```
}
```

Menu



Track Progress

43 of 67 Complete. (65%)

Dash



✔ Article marked as read.

Output

s2 is rotation of s1



Articles



Videos

Time Complexity:

Time Complexity will be $n \times n$ in the worst case, where n is the length of the string.



Problems

Marked as Read



Quiz

🚩 Report An Issue

If you are facing any issue on this page. Please let us know.



Menu

Track Progress

43 of 67 Complete. (65%)