

## Program: 10

Find number of times a string occurs as a subsequence in given string.

### Input:

```
string a = "GeeksforGeeks"  
string b = "Gks"
```

Output: 4

### Explanation:

The four strings are - (Check characters marked in **bold**)

**G**eeksforGeeks

Geeksfor**G**eeks

**G**eeksforGeeks

Geeksfor**G**eeks

The idea is to process all characters of both strings one by one starting from either from left or right side. Let us traverse from right corner, there are two possibilities for every pair of character being traversed.

m: Length of str1 (first string)

n: Length of str2 (second string)

If last characters of two strings are same,

1. We consider last characters and get count for remaining strings. So we recur for lengths m-1 and n-1.
2. We can ignore last character of first string and recurse for lengths m-1 and n.

else

If last characters are not same,

We ignore last character of first string and  
recurse for lengths m-1 and n.

### Source Code:

```
# Recursive function to find the  
# number of times the second string  
# occurs in the first string
```

```
def count(a, b, m, n):
```

```
    # If both first and second string  
    # is empty, or if second string  
    # is empty, return 1  
    if ((m == 0 and n == 0) or n == 0):  
        return 1
```

```
    # If only first string is empty  
    # and second string is not empty,  
    # return 0  
    if (m == 0):  
        return 0
```

```
    # If last characters are same  
    # Recur for remaining strings by  
    # 1. considering last characters
```

```

# of both strings

# 2. ignoring last character

# of first string
if (a[m - 1] == b[n - 1]):
    return (count(a, b, m - 1, n - 1) +
            count(a, b, m - 1, n))
else:

    # If last characters are different,
    # ignore last char of first string
    # and recur for remaining string
    return count(a, b, m - 1, n)

# Driver code
a = "GeeksforGeeks"
b = "Gks"

print(count(a, b, len(a), len(b)))

```

**Output:**

4