

Permutations of A in B

Problem Description

You are given two strings, A and B, of size N and M, respectively.

You have to find the **count of all permutations of A present in B as a substring**. You can assume a string will have only lowercase letters.

Problem Constraints

$1 \leq N < M \leq 10^5$

Input Format

Given two arguments, A and B of type String.

Output Format

Return a single integer, i.e., number of permutations of A present in B as a substring.

count = 0

A: a a b
 $\begin{bmatrix} 0 & 1 & 2 & 3 & \dots & 25 \\ 2 & 1 & 0 & 0 & \dots & 0 \end{bmatrix}$

B: $\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ a & b & a & a & b & c & b & a & b & a & a & c \end{matrix}$

Start	End	Remove	Add	State Of Array	Result
0	2	-	-	$\begin{bmatrix} 0 & 1 & 2 & \dots & 25 \\ 2 & 1 & 0 & \dots & 0 \end{bmatrix}$	✓ 1
1	3	0	3	$\begin{bmatrix} 0 & 1 & 2 & \dots & 25 \\ 2 & 1 & 0 & \dots & 0 \end{bmatrix}$	✓ 2
2	4	1	4	$\begin{bmatrix} 0 & 1 & 2 & \dots & 25 \\ 2 & 1 & 0 & \dots & 0 \end{bmatrix}$	✓ 3
3	5	2	5	$\begin{bmatrix} 0 & 1 & 2 & \dots & 25 \\ 1 & 1 & 1 & \dots & 0 \end{bmatrix}$	X 3
4	6	3	6	$\begin{bmatrix} 0 & 1 & 2 & \dots & 25 \\ 0 & 2 & 1 & \dots & 0 \end{bmatrix}$	X 3
5	7	4	7	$\begin{bmatrix} 0 & 1 & 2 & \dots & 25 \\ 1 & 1 & 1 & \dots & 0 \end{bmatrix}$	X 3
6	8	5	8	$\begin{bmatrix} 0 & 1 & 2 & \dots & 25 \\ 1 & 2 & 0 & \dots & 0 \end{bmatrix}$	X 3
7	9	6	9	$\begin{bmatrix} 0 & 1 & 2 & \dots & 25 \\ 2 & 1 & 0 & \dots & 0 \end{bmatrix}$	✓ 4
8	10	7	10	$\begin{bmatrix} 0 & 1 & 2 & \dots & 25 \\ 2 & 1 & 0 & \dots & 0 \end{bmatrix}$	✓ 5
9	11	8	11	$\begin{bmatrix} 0 & 1 & 2 & \dots & 25 \\ 2 & 0 & 1 & \dots & 0 \end{bmatrix}$	X 5

a a b c $\begin{bmatrix} 0 & 1 & 2 & \dots & 25 \\ 2 & 1 & 1 & \dots & 0 \end{bmatrix}$

Code:

b a a [0 1 2 25]

'a'

```
int isAnagram(String A, String B)
{
    int[] fA = new int[26];
    int[] fB = new int[26];
    for (i=0; i<A.length(); i++)
    { fA[A.charAt(i) - 'a']++; }
    for (i=0; i<B.length(); i++)
    { fB[B.charAt(i) - 'a']++; }
}
```

a → 0
b → 1
c → 2
i
z → 25

```
int count (String A, String B)
{
```

}

Longest Palindromic Substring

Problem Description

Given a string A of size N, find and return the **longest palindromic substring** in A.

Substring of string A is $A[i..j]$ where $0 \leq i \leq j < \text{len}(A)$

Palindrome string:

A string which reads the same backwards. More formally, A is palindrome if $\text{reverse}(A) = A$.

Incase of conflict, return the substring which occurs first (with the least starting index).

Problem Constraints

$1 \leq N \leq 6000$

Input Format

First and only argument is a string A.

Output Format

Return a string denoting the longest palindromic substring of string A.

ans:

length:
1 8

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
x b d y z z y d a n d y z z d p
↑ ↑

A.substring(4, 14)

1. Even length palindromic substring.

baab

zbaabz

aa

2. Odd length palindromic substring.

malayalam

pop

madam

3. Expansion from the middle.

0	1	2	3	4	5	6	7	8	9
x	b	d	y	z	z	y	d	b	x
	s↑							t↓	

9	10	11	12	13	14	15
x	d	y	z	y	d	p
	s↑				t↓	