Beautiful Strings

by YuryBandarchuk

Problem Submissions Leaderboard Discussions Editorial

Submissions: 635
Max Score: 80
Difficulty: Hard

Rate This Challenge:

More

You are given a string, S, consisting of lowercase English letters.

A string is *beautiful* with respect to S if it can be derived from S by removing *exactly* 2 characters.

Find and print the number of different strings that are *beautiful* with respect to S.

Input Format

A single string of lowercase English letters denoting S.

Constraints

- $3 \leq |S| \leq 10^6$
- $3 \leq |S| \leq 20$ holds for test cases worth at least 15% of the problem's score.
- $3 \leq |S| \leq 2000$ holds for test cases worth at least 30% of the problem's score.

Output Format

Print the number of different strings that are *beautiful* with respect to S.

Sample Input

```
abba
```

Sample Output

```
4
```

Explanation

$S = \{abba\}$

The following strings can be derived by removing 2 characters from S: $ab, bb, ba, ab, ba, aa,$ and bb.

This gives us our set of *unique* beautiful strings, $B = \{ab, ba, aa, bb\}$. As $|B| = 4$, we print 4.

Current Buffer (saved locally, editable) Java 8

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7
```

```
        Scanner scan = new Scanner(System.in);
        String str = scan.next();

        long indivitual = 0;
        long duplicates = 0;
        long pattern = 0;

        char lastChar = '\0';
        char secondLastChar = '\0';

        long duplicateLen = 0;

        for(int i = 0 ; i < str.length() ; i++){

            if(lastChar == str.charAt(i)){

                if(duplicateLen == 0){
                    duplicates++;
                }
                else{
                    duplicateLen++;
                }

            }
            else{
                indivitual++;
                duplicateLen = 0;
            }

            if(secondLastChar == str.charAt(i)){
                pattern++;
            }

            secondLastChar = lastChar;
            lastChar = str.charAt(i);
        }

        long output = ((indivitual * (indivitual - 1)) / 2) + (duplicates) - (pattern);

        System.out.println(output);

    }
}
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature