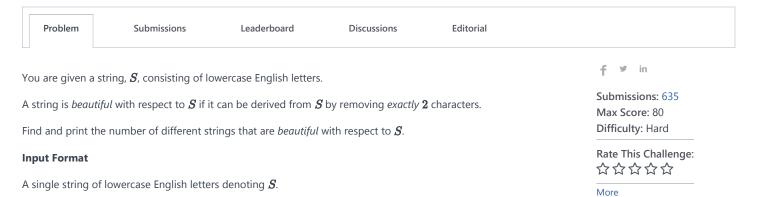


Beautiful Strings





Constraints

- $3 \le |S| \le 10^6$
- $3 \le |S| \le 20$ holds for test cases worth at least 15% of the problem's score.
- $3 \le |S| \le 2000$ 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 ${f 2}$ characters from ${f S}:{f ab,bb,ba,ab,ba,aa,aa,and}$ ${f bb}$.

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) 

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

```
8/29/2017
     8
                 Scanner scan = new Scanner(System.in);
     9
                String str = scan.next();
    10
                 long indivitual = 0;
    11
    12
                 long duplicates = 0;
    13
                 long pattern = 0;
    14
    15
                 char lastChar = '\0';
    16
                 char secondLastChar = '\0';
    17
    18
                 long duplicateLen = 0;
    19
                 for(int i = 0 ; i < str.length() ; i++){</pre>
    20 ▼
    21
    22 🔻
                     if(lastChar == str.charAt(i)){
    23
    24 ▼
                         if(duplicateLen == 0){
    25
                             duplicates++;
    26
                         }
    27 ▼
                         else{
                             duplicateLen++;
    28
                         }
    29
    30
    31
                     else{
    32 🔻
    33
                         indivitual++;
    34
                         duplicateLen = 0;
    35
    36
    37 ▼
                     if(secondLastChar == str.charAt(i)){
    38
                         pattern++;
    39
    40
    41
                     secondLastChar = lastChar;
    42
                     lastChar = str.charAt(i);
    43
                }
    44
    45
                 long output = ((indivitual * (indivitual - 1)) / 2) + (duplicates) - (pattern);
    46
    47
                System.out.println(output);
    48
    49
            }
    50
        }
                                                                                                                         Line: 1 Col: 1
```

Test against custom input **1** Upload Code as File

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