CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	JAVA	Session	Strings	Question Information	Level 1 Challenge 67

Problem

Question description

Walter was reading bed time stories to his son and teaching him alphabets at the same time and he though to give him a little problem to sharpen his mind also so he gave him a string S. He has to perform moves on the string s, to make it a good string.

In one move, he can delete a character from string s.

A good string, is a string in which the number of occurrences of every character is at most 1.

Walter's son has to find out the minimum number of moves required to make the given string s a good string.

Can you help Walter's son?

Constraints:

$1 \leq |S| \leq 10^5$

Input Format:

A single line input which contains a string S

Output Format:

Print the minimum number of moves required to make string S, a good string.

Test Cases

Logical Test Cases

Test Case 1

INPUT (STDIN)

```
plausible
```

EXPECTED OUTPUT

```
1
```

Test Case 2

INPUT (STDIN)

```
tantalizingly
```

EXPECTED OUTPUT

```
5
```

Mandatory Test Cases

Test Case 1

KEYWORD

```
s=sc.next();
```

Test Case 2

KEYWORD

```
char[]
str=s.toCharArray();
```

Test Case 3

KEYWORD

```
Arrays.sort(str);
```

Test Case 4

KEYWORD

```
for(int j=0;j<str.length-
1;j++)
```

Complexity Test Cases

Test Case 1	Test Case 2	Test Case 3
CYCLOMATIC COMPLEXITY	TOKEN COUNT	NLOC
3	150	25

Code Editor

You have already solved this challenge ! Though you can run the code with different logic !

Code Editor | JAVA SE 1.8 | Light Theme

```java
import java.util.*;
public class Class332241010280 {
     public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);
     String s;
     s=sc.next();
     char[] str=s.toCharArray();
     Arrays.sort(str);

     int moves = 0;
     for(int j=0;j<str.length-1;j++) {
         if (str[j] == str[j+ 1]) {
             moves++;
         }
     }
     System.out.println(moves);
    }
}
```

Custom Input (stdin) | T1 | T2

Type Here

Output | MATCH T1 | MATCH T2

Empty

Complexity Analysis

Test Case Status