
Total as in questions and total bs in exclamations

X95873_en

The input of this exercise consists of several non-empty sentences made of a's and b's and ending in a punctuation sign, either . or ? or !. Each sentence will be in a different line to make the presentation clearer. For instance, this would be a possible input:

```
bbb?  
aabba .  
b .  
bbbba?  
abbaa?  
aaabbb!  
babbabb .  
bbabb!  
aa!
```

Sentences ending in . are called statements.

Sentences ending in ? are called questions.

Sentences ending in ! are called exclamations.

Implement a program that reads this input and outputs the total number of a's appearing in questions, and the total number of b's appearing in exclamations.

In the previous example, the output would be 4 and 7.

Input

The input contains an arbitrary number of lines. Each one has one or more characters a or b ending with a punctuation sign . or ? or !.

Output

Two integers must be written in one line and separated by a space: the total number of a's appearing in questions, and the total number of b's appearing in exclamations.

Sample input 1

```
bbb?  
aabba .  
b .  
bbbba?  
abbaa?  
aaabbb!  
babbabb .  
bbabb!  
aa!
```

Sample output 1

```
4 7
```

Sample input 2

```
abbbbbaabbababb?  
a?  
ba .  
aaabbbbbaaa!
```

```
bababbbbabaab .  
abaabaaab .  
a .  
babbbabababa!  
abaa!
```

```
abbabaaa?  
aaabbaaa!  
babaaab!  
bbbababbbbb!  
bbabbababb!  
babbbaaabababa?  
bbbbaaababaa?  
ababbbbaa!  
abb.  
ab!  
bbaabbab!  
bbbbaa?  
aaabb!  
abababa!  
abbaa.  
aab.  
a!  
aabbaaababb?  
a!  
a?  
abaaaaa.
```

Sample output 2

```
35 51
```

Observation

Massive storage methods are forbidden, not even a simple `string`. The input has to be read character by character. In particular, using `getline` or similar is forbidden, and the reason is obvious: in normal conditions, sentences would not be separated by newlines. In fact, the input is separated by newlines only for presentation reasons. If you keep reading characters using `cin >> c`, it is irrelevant if newlines happen in the input, since reading with `cin` skips them entirely.

Grading up to 10 points:

- Slow solution: 5 points.
- Fast solution: 10 points.

We understand as fast solution one which is correct, has linear cost and passes the public and private tests. We understand as slow solution one which is not fast, but it is correct and capable of passing the public tests.

Problem information

Author : PRO1

Generation : 2023-11-01 13:35:12

© [Jutge.org](https://jutge.org), 2006–2023.

<https://jutge.org>