DA Paradox

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Dwight loves *Scranton*'s branch employees especially *Michael Scott*. So, when *Scranton*'s branch absorbed *Stamford*'s branch, a lot of new employees started working in *Scranton*'s branch including *Andy*.

Andy is trying to get close to *Michael* to boost his career, but *Dwight* knew *Andy*'s real intentions, so he decided to tell *Michael*.

Michael being a fun boss, decided to give them a challenge to keep them busy, so he can keep calling *Jan Levinson ~~Gould~~*. The challenge goes as follows.

Given a string S consisting of lowercase Latin letters, in one move you can perform one of the following operations:

- Change any vowel into a consonant letter.
- Change any consonant letter into a vowel.

A consonant is any letter which is not a vowel.

Vowels are: a, e, i, o, u.

What is the minimum number of moves needed to fix the given string, so that **NO** two vowels are adjacent to each other and **NO** two constant letters are adjacent to each other as well?

You — being a good friend to *Dwight* — decided to help him beat *Andy* in the challenge.

Input

The only line of input contains a string S ($1 \leq |S| \leq 10^5$) consisting of lowercase Latin letters where $|S|$ is the size of the string.

Output

Print a single integer, the answer to the problem.

Examples

standard input	standard output
dwight	2
knb	1