

## Problem B. Cool Name

**Time limit** 1000 ms

**Mem limit** 262144 kB

After holding one team contest, boy Yura got very tired and wanted to change his life and move to Japan. In honor of such a change, Yura changed his name to something nice.

Fascinated by this idea he already thought up a name  $s$  consisting only of characters "\_" and "^". But there's a problem — Yura likes smiley faces "^\_" and "^ ^". Therefore any character of the name must be a part of at least one such smiley. Note that only the **consecutive** characters of the name can be a smiley face.

More formally, consider all occurrences of the strings "^\_" and "^ ^" in the string  $s$ . Then all such occurrences must cover the whole string  $s$ , possibly with intersections. For example, in the string "^ ^ \_ ^ ^ \_ ^" the characters at positions 3, 4, 9, 10 and 11 are not contained inside any smileys, and the other characters at positions 1, 2, 5, 6, 7 and 8 are contained inside smileys.

In one operation Yura can insert one of the characters "\_" and "^" into his name  $s$  (you can insert it at any position in the string). He asks you to tell him the minimum number of operations you need to do to make the name fit Yura's criteria.

### Input

Each test consists of multiple test cases. The first line contains a single integer  $t$  ( $1 \leq t \leq 100$ ) — the number of test cases. The description of test cases follows.

The first and only line of each test case contains a single string  $s$  ( $1 \leq |s| \leq 100$ ), consisting of characters "\_" and "^", — the name to change.

### Output

For each test case, output a single integer — the minimum number of characters you need to add to the name to make it fit for Yura. If you don't need to change anything in the name, print 0.

### Examples

Input	Output
<pre> 7 ^   _____ ^     ^   ^ ^ ^ ^   _____ ^   ^   ^ ^ ^ - ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^   -   ^ ^   -   - _____ - </pre>	<pre> 5 5 1 1 0 3 2 </pre>

### Note

In the first test case, you can get the following name by adding 5 characters:

$$\wedge \quad \wedge \quad \wedge \quad \wedge \quad \wedge \quad \wedge \quad \wedge$$

In the third test case, we can add one character "^" to the end of the name, then we get the name:

$$\begin{array}{c} \wedge \quad \Lambda \\ \text{—} \end{array}$$

In the fourth test case, we can add one character "^" to the end of the name, then we get the name:

 $\wedge \wedge$ 

In the fifth test case, all of the characters are already contained in smiley faces, so the answer is 0.

In the seventh test case, you can add one character "^" at the beginning of the name and one character "^" at the end of the name, then you get the name:

$$\Lambda \quad \Lambda$$