

## 6979 Known Notation

Do you know reverse Polish notation (RPN)? It is a known notation in the area of mathematics and computer science. It is also known as postfix notation since every operator in an expression follows all of its operands. Bob is a student in Marjar University. He is learning RPN recent days.

To clarify the syntax of RPN for those who haven't learnt it before, we will offer some examples here. For instance, to add 3 and 4, one would write "3 4 +" rather than "3 + 4". If there are multiple operations, the operator is given immediately after its second operand. The arithmetic expression written "3 - 4 + 5" in conventional notation would be written "3 4 - 5 +" in RPN: 4 is first subtracted from 3, and then 5 added to it. Another infix expression "5 + ((1 + 2) × 4) - 3" can be written down like this in RPN: "5 1 2 + 4 × + 3 -". An advantage of RPN is that it obviates the need for parentheses that are required by infix.

In this problem, we will use the asterisk "\*" as the only operator and digits from "1" to "9" (without "0") as components of operands.

You are given an expression in reverse Polish notation. Unfortunately, all space characters are missing. That means the expression are concatenated into several long numeric sequence which are separated by asterisks. So you cannot distinguish the numbers from the given string.

Your task is to check whether the given string can represent a valid RPN expression. If the given string cannot represent any valid RPN, please find out the minimal number of operations to make it valid. There are two types of operation to adjust the given string:

1. Insert. You can insert a non-zero digit or an asterisk anywhere. For example, if you insert a "1" at the beginning of "2\*3\*4", the string becomes "12\*3\*4".
2. Swap. You can swap any two characters in the string. For example, if you swap the last two characters of "12\*3\*4", the string becomes "12\*34\*".

The strings "2\*3\*4" and "12\*3\*4" cannot represent any valid RPN, but the string "12\*34\*" can represent a valid RPN which is "1 2 \* 3 4 \*".

### Input

There are multiple test cases. The first line of input contains an integer  $T$  indicating the number of test cases. For each test case:

There is a non-empty string consists of asterisks and non-zero digits. The length of the string will not exceed 1000.

### Output

For each test case, output the minimal number of operations to make the given string able to represent a valid RPN.

### Sample Input

```
3
1*1
11*234**
*
```

**Sample Output**

1  
0  
2