

Bot Friends

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

Little Cyan Fish has n robots friends (referred to as bots) positioned on a number line. The i -th bot is located at position $i + 0.5$ for $1 \leq i \leq n$. Additionally, there are $n + 1$ holes on the number line, with the i -th hole at position i for $1 \leq i \leq n + 1$.

Initially, all bots are inactive. The Little Cyan Fish activates the bots sequentially in any order. Upon activation, a bot moves either to its left or right, following these rules based on its type:

- Bots of type “<” can only move left.
- Bots of type “>” can only move right.
- Bots of type “?” can choose to move either left or right.

A bot continues to move in its chosen direction until it falls into an unoccupied hole, at which point it stops and occupies that hole. If a bot reaches the position 0 or the position $n + 2$, it disappears.

The objective is to avoid having bad bots, defined as follows: If a bot at position $i + 0.5$ falls into the hole at position i or $i + 1$, it is considered bad. Otherwise, it is considered good.

The challenge for the Little Cyan Fish is to determine the direction for each bot of type ? and the activating order such that the number of good bots is maximized. Help Little Cyan Fish to find an optimal to assign the directions and activate the bots, and find the maximum number of good bots.

Input

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \leq T \leq 10^5$), indicating the number of test cases.

For each test case, the first line of the input contains a string of length n ($1 \leq n \leq 5000$). Each character of the string is either “<”, “>” or “?”, indicating the type of the i -th bot.

It is guaranteed that the sum of n^2 over all test cases does not exceed 5×10^7 .

Output

For each test case, output a single line contains a single integer, indicating the answer.

Example

standard input	standard output
10	2
?>?	2
>?<	3
?<?<?	4
?><?<	5
???????	8
>?<?<?<?<	7
?><????><><	8
??>>><><??	5
<>>?>>?>?>	6
<?<>>??<?>	