Task 2

Algorithms

PROBLEM STATEMENT:

TASK 2(A): Longest Proper Prefix

Harsha is studying compiler design and come across the term 'parser'. After understanding what it is, he is super excited. He decides to build his own parser and designs one after programming for two days straight.

A parser is, in simple terms, a software component that processes text, and checks it's semantic correctness, or, if you prefer, if the text is properly built.

Today, however, he is concerned with an abstract instruction which is composed of the characters "<" and ">" , which he will use on the design of his language H++.

He is using it as an abstraction for generating XML code Tags in an easier fashion and he understood that, for an expression to be valid, a "<" symbol must always have a corresponding ">" character somewhere (not necessary immediately) after it. Moreover, each ">" symbol should correspond to exactly one "<" symbol.

So, for instance, the instructions:

<<>>

<>

<><> are all valid. While:

>>

><>< are not.

Given some expressions which represent some instructions to be analyzed by his parser, you should tell the length of the longest prefix of each of these expressions that are valid, or 0 if there's no such a prefix.

Input

Input will consist of an integer T denoting the number of test cases to follow.

Then, T strings follow, each on a single line, representing a possible expression in L++.

Output

For each expression you should output the length of the longest prefix that is valid or 0 if there's no such a prefix.

Constraints

$1 \leq T \leq 500$

$1 \leq$ The length of a single expression $\leq 10^6$

The total size all the input expressions is no more than $5*10^6$

SAMPLE

Input:

```
3
<<>>
<>>>
><><>
```

Output:

4

2

0

TASK 2(B): Max Element in Array

There were 2 friends Raj and Anoop. Both are exceptional problem solvers. One day anoop gave a question to Raj to solve. Since Raj is also having some project work to do, he asks you to solve on behalf of him.

According to question you are given with an array with size n of integers, where a[i]=i.

You will be given with q queries. In each query you have to add a value to each of the array element i between the given indices inclusive of both the indices. As a result, you have to return the maximum element of the resultant array.

Input:

Number of elements in the array, n

Number of queries to be performed, q

Constraints for Basic Mode:

$1<=n<=10^5$ and $1<=q<=100$

Constraints for Hacker Mode(1):

$1<=n<=10^5$ and $1<=q<=10^5$

Constraints for Hacker Mode(2):

$1<=n<=10^{18}$ and $1<=q<=10^5$

Additional Constraints:

Left(L) and Right(R) indices for query: 1<=L<=R<=n

Value to be added, v: 1<=v<=10^9

SAMPLE

Input:

5

3

1 4 10

2 2 20

3 5 25

Output:

39

GUIDELINES:

The best practice anyone could do is solving question. We have complied few questions. We want you to explore each topic and solve these questions while doing so. If you are stuck or have any doubts you can always ping your mentor. Happy Coding :)

EVALUATION METRICS:

- Plagiarism of any sort will result in elimination
- These tasks will be followed by personal interviews and if candidates fail to explain their concept and code, they will be eliminated
- Novel approaches will be rewarded
- Time Complexity and Space complexity for each solution will be considered for evaluation.

SUBMISSION:

Add the code files along with explanatory comments to a GitHub repository or a Google Drive directory and submit the link on the portal.