

You are given a bracket sequence B . The bracket sequence may contain 4 types of brackets:

1. Round brackets (or)
2. Curly brackets { or }
3. Square brackets [or]
4. Angle brackets < or >

For each position in the bracket sequence B , you need to output the length of the longest balanced contiguous bracket sequence starting from (and including) that particular position.

Formally, a bracket sequence T is balanced if

- T is empty
- T is (P) , $[P]$, $\{P\}$, $\langle P \rangle$ where P is a balanced bracket sequence
- T is PQ where P and Q are balanced bracket sequences.

For example, for $B = (\langle \rangle) \langle \rangle$, the answer is '4 2 0 0 0 0'.

Input

First line of the input will contain a single positive integer T ($T \leq 10$) denoting the number of test cases. Hence T cases follow. Each case is a single line of non-empty bracket sequence, containing only 8 types of characters '(', ')', '{', '}', '[', ']', '<', '>'. Each of these bracket sequences will not contain more than 10^5 characters.

If it helps, most of the judge data is produced randomly. First a random bracket sequence (not necessarily balanced) of certain length is generated. Say it is X . Then we change X by replacing some substring of it with a random balanced bracket sequence several times.

Output

For each test case, output case number (no trailing space after 'Case x :'), followed by the answers in separate line. There is **NO** empty line between cases. For details, please see the sample.

Sample Input

```
5
()
<>
(<>)<>
()()
{[[]}
```

Sample Output

```
Case 1:
2
0
Case 2:
2
0
Case 3:
4
2
0
0
0
0
Case 4:
4
0
2
0
Case 5:
0
0
0
0
```



chris.com