A CME (Correct Mathematical Expression) was defined by the following rules:

- $z$ is a CME;
- if $X$ is a CME, then so is $(X)$;
- if $X$ and $Y$ are both CMEs, then so is $(X+Y)$;
- there are no more CMEs than those produced by the three rules above.

This set of rules produces terms like

$$(z) \quad (z+z) \quad (z+(z+z)) \quad (((z))) \quad ...$$

Unfortunately, the job to produce the CMEs was given to a half-crazy computer (a HAL’s cousin) that sometimes flipped the parentheses, from ‘)’ to ‘(’ and viceversa, thus producing terms like

$$(z) \quad (z+z ( \quad (z+(z+z ( \quad ))) ) z ((( \quad ...$$

We call these terms ACMEs (Almost Correct Mathematical Expressions). You are asked to write a program such that, given an ACME, computes the minimum number of parentheses that must be flipped to get a CME.

**Input**

The input has several non-empty strings consisting of at most $10^4$ characters chosen from \{‘z’, ‘(’, ‘)’, ‘+’\}.

**Output**

For each string of the input, tell if it is a CME, an ACME, or rubbish. In the second case, compute the minimum number of flips to convert the string into a CME.

**Sample input**

<table>
<thead>
<tr>
<th>$z$</th>
<th>(z+(z+z())</th>
<th>+z</th>
</tr>
</thead>
</table>

**Sample output**

| $z$ : this is a CME | (z+(z+z()) : 1 flip(s) | +z : this is rubbish |