In an infinite binary tree:

- Each node has exactly two children – a left and a right child.
- If a node is labeled with the integer X, then its left child is labeled 2·X and its right child 2·X+1.
- The root of the tree is labeled 1.

A **walk** on the binary tree starts in the root. Each step in the walk is either a jump onto the left child, onto the right child, or pause for rest (stay in the same node).

A walk is described with a string of letters 'L', 'R' and 'P':

- 'L' represents a jump to the left child;
- 'R' represents a jump to the right child;
- 'P' represents a pause.

The **value** of the walk is the label of the node we end up on. For example, the value of the walk LR is 5, while the value of the walk RPP is 3.

A **set** of walks is described by a string of characters 'L', 'R', 'P' and '*'. Each '*' can be any of the three moves; the set of walks contains all walks matching the pattern.

For example, the set L*R contains the walks LLR, LRR and LPR. The set ** contains the walks LL, LR, LP, RL, RR, RP, PL, PR and PP.

Finally, the **value of a set** of walks is the sum of values of all walks in the set.

Calculate the value of the given set of walks.

INPUT

A string describing the set. Only characters 'L', 'R', 'P' and '*' will appear and there will be at most 10000 of them.

OUTPUT

Output the value of the set.

SCORING

In test data worth 30% points, there will be no characters '*'.

In test data worth 50% points, there will be at most three characters '*'.

EXAMPLES

input	**input**	**input**	**input**
P*P	L*R	**	LLLLLRRRRRLLLLLRRRRRLLLLLRRRRRLLLLL
output	**output**	**output**	**output**
6	25	33	35400942560