10651   Pebble Solitaire

Pebble solitaire is an interesting game. This is a game where you are given a board with an arrangement of small cavities, initially all but one occupied by a pebble each. The aim of the game is to remove as many pebbles as possible from the board. Pebbles disappear from the board as a result of a move. A move is possible if there is a straight line of three adjacent cavities, let us call them A, B, and C, with B in the middle, where A is vacant, but B and C each contain a pebble. The move constitutes of moving the pebble from C to A, and removing the pebble in B from the board. You may continue to make moves until no more moves are possible.

In this problem, we look at a simple variant of this game, namely a board with twelve cavities located along a line. In the beginning of each game, some of the cavities are occupied by pebbles. Your mission is to find a sequence of moves such that as few pebbles as possible are left on the board.
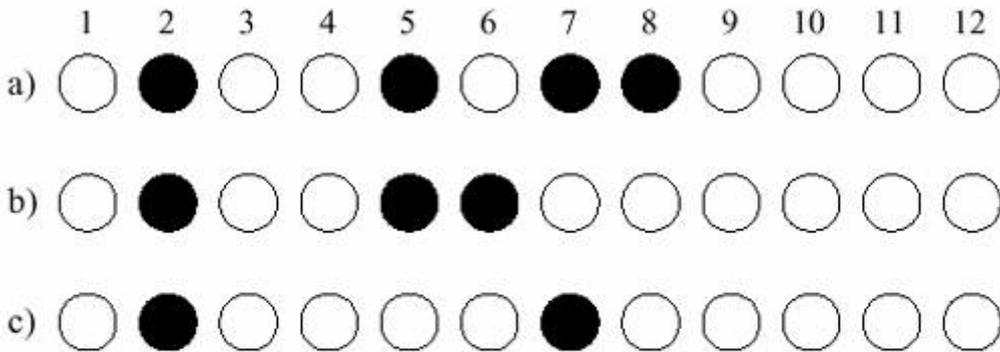

Fig 1. In a) there are two possible moves, namely $8 \rightarrow 6$, or $7 \rightarrow 9$. In b) the result of the $8 \rightarrow 6$ move is depicted, and again there are two possible moves, $5 \rightarrow 7$, or $6 \rightarrow 4$. Making the first of these results in c), from which there are no further moves.

Input

The input begins with a positive integer n on a line of its own. Thereafter n different games follow. Each game consists of one line of input with exactly twelve characters, describing the twelve cavities of the board in order. Each character is either '-' or 'o' (The fifteenth character of English alphabet in lowercase). A '-' (minus) character denotes an empty cavity, whereas a 'o' character denotes a cavity with a pebble in it. As you will find in the sample that there may be inputs where no moves is possible.

Output

For each of the n games in the input, output the minimum number of pebbles left on the board possible to obtain as a result of moves, on a row of its own.

Sample Input

```
5
---oo-------
-o--o-oo----
-o----ooo---
```

```
oooooooooooo
oooooooooo-o
```

Sample Output

```
1
2
3
12
1
```