

738 A Logical Problem

The input file for this program contains several logic circuit diagrams composed of zero or more dual-input AND and/or OR gates (with possible inversions of input and/or output values), one or more inputs, and a single output. The goal of this problem is to write a program which will determine what the output should be for a given circuit diagram and for a specified set of input values.

The logic circuits are drawn using printable ASCII characters and are allowed to be a maximum of 100 by 100 characters in size. The AND and OR gates are represented by the following arrangements of ASCII characters, respectively

```

      :\
      : )
      :/
  
```

```

      :\
      : >
      :/
  
```

Note that the only difference is that the AND gate has a right parenthesis character in the rightmost column of the center row, while the OR gate has a greater-than symbol in that same position. The gates in the data file will always be oriented as shown (i.e. they will not be flipped, rotated, or otherwise modified in any way from the arrangements shown above).

The “circuit” paths used to connect the gates are represented using the dash character (ASCII code 45 decimal) for horizontal paths, and the vertical bar character (ASCII code 124 decimal) for vertical paths. A path always travels in a straight line (either horizontally or vertically) unless altered by a junction, represented by a plus character (ASCII code 43 decimal), at which point it may make a 90 degree turn only. No two junctions may be adjacent to one another in horizontal or vertical direction, and no two paths may cross at any point.

The following circuits illustrate some legal and illegal circuit paths

```

----+           |
--+|  These two paths are both legal
  ||
  |
----+  Illegal, no two junctions may
----+  be adjacent to one another
  |    in a horizontal or vertical direction
  |
----+-----  Illegal, no 2 paths
  |           may cross
  
```

The two inputs for a specific logic gate always approach horizontally from the left. One input is adjacent to the leftmost character of the first row of the gate, and the other input is adjacent to the leftmost character of the third row. The output for a gate always leaves horizontally to the right and is adjacent to the rightmost character of the second row of the gate. The positions of these inputs and output locations are indicated by the dash characters (‘-’) in the diagram below:

```

-:\
 : )-
-:/
  
```

It is possible for either of the two inputs or the output to be inverted. That is, if the input or output is a ‘1’, it is changed to a ‘0’, or if it was to be a ‘0’, it is changed to a ‘1’. To indicate inversion, a

lowercase ‘o’ character (ASCII code 111 decimal) is placed in the appropriate input or output position adjacent to the gate. If an input or output is inverted, it will always be preceded (in the case of an input) or followed (in the case of an output) by at least a single dash character. As an example, consider the diagram below which shows an AND gate with its top input and its output inverted:

```
-o:\
 : )o-
-:/
```

As previously stated, there are one or more inputs and a single output in the complete logic circuit. Each input is indicated by one of the capital letters ‘A’ through ‘Z’, and the output is indicated by a question mark character. So, a simple circuit consisting only of a single OR gate with two inputs labeled ‘A’ and ‘B’ could be represented as shown here:

```
A-:\
 : >-?
B-:/
```

Input

You may assume that the input data obeys all of the rules outlined above (ie. there are no illegal circuits). The bottom end of a logic diagram is indicated by line containing only a single asterisk in the first column. After this are several lines each of which indicate the state of the inputs in the previous logic diagram. Each line is a string of 26 ‘0’ and/or ‘1’ characters, with the first position representing the state of input ‘A’, the second position representing the state of input ‘B’, etc. Note that input values which are not actually used in the circuit may simply be ignored. The list of input states is terminated by a line containing only a single asterisk character in the first column.

Following the asterisk which terminates the list of input states is another circuit diagram followed by an asterisk and a list of input states terminated by an asterisk, which is then followed by another circuit diagram and another list of input states, and so on until the end of the file. The file will always contain at least one circuit and one set of inputs for that circuit.

Output

For each logic circuit diagram, the program is to report the value of the output (0 or 1), one value per line, for each input state in the list which follows the circuit. The list of outputs for a given circuit should be followed by a single blank line to separate it from the lists for subsequent circuits.

Sample Input

```
A-:\
 : )-?
B-:/
*
00000000000000000000000000000000
10000000000000000000000000000000
01000000000000000000000000000000
11000000000000000000000000000000
*
+-A
|
+-:\
```

```

      : >o-:\
+-:/      : )-?
|+-----o:/
B-+|
C-+
*
00000000000000000000000000000000
11100000000000000000000000000000
*
A-:\
      : )-?
A-:/
*
00000000000000000000000000000000
10000000000000000000000000000000
*

```

Sample Output

```

0
0
0
1

1
0

0
1

```