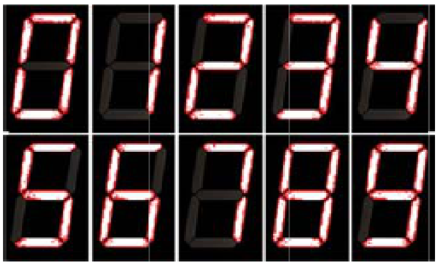
ACM-ICPC Live Archive

5049 Seven Segment Counter



A Seven Segment Display is generally a form for representing decimals numbers 0 to 9, using exactly 7 segments. You may have already seen such displays on price-list boards, like in exchange offices or fuel stations. Using three of these displays together, with a proper underlying electronic circuit, one can create a three digit display to show number between 0 and 999 (inclusive), without leading zero. Our man, Shelby, has already done this! Going further he has even made it to work as a time counter — As each second passes the number in display advances by one and when reaching 999, the next number is 0 (i.e. it works as a cyclic counter).

Shelby is a technician in a company. There is one of those three-digit seven segment display (working as a counter) hanging on Shelby's office wall. Shelby is asked to write the displaying number of the counter on a piece of paper, everyday at 15:00:00 that is the time he leaves the company. It has some security reasons and is requested by the big boss!

Unfortunately, yesterday Shelby forgot to write the number and he is too anxious now! As a matter of luck he has found some images taken by various surveillance cameras taken at 15:00:00 or later. But the problem is that, the state (either on or off) of some of the 21 segments (in three seven segments) are not visible in the photos clearly!

Given the photos, each containing information about those 21 segments (either visible and on, or visible and off, or not-visible in this certain photo), and the time the photos is taken, you are asked to help Shelby to find out the number of the counter at 15:00:00.

Input

The input consists of several test cases, each containing some photos (at least one and at most 20). Each photo is shown in 6 lines. The first line has a single non-negative integer, less than or equal to 61200, indicating the number of seconds passed since 15:00:00. Then, in the following 5 lines the photo is drawn.

If all the 21 segments of the photo are visible and on (value equals to 888), then the following characters will make the five lines containing space, dash (minus), and vertical bar characters.

```
 -   -   - 
| | | | | |
 -   -   - 
| | | | | |
 -   -   - 
```

Otherwise, for any segment (of these 21) that is visible but is off, a dot character (.) is written instead of '-' or '|'. For those segments which are not visible, an asterisk character (*) is put. You can assume that the digits 0 to 9 are supposed to be displayed like this in a clear photo:

```
 -               -     -         
| |     |       |     |   | |
                 -     -     -  
| |     |     |         |     |
 -               -     -       
```

```
 -     -     -     -     - 
|     |         |   | |   | |
 -     -             -     - 
  |   | |       |   | |     |
 -     -             -     - 
```

The test cases in the input are separated by a line containing a single '#' sign. The last line of the input contains a single '$' sign.

There are not any leading spaces but trailing spaces might occur. Thus, every line has at least (and not exactly) 9 characters in length. Remember the numbers in counter does not contain leading zeros. For example, the value 70 is shown as 7 on the second digit from the left and 0 on the third digit from the left. The leftmost digit is all-off (it is not 070).

Output

For every test case, if it is possible to find out the value of counter at 15:00:00, then find it and write it in the output. Otherwise, write a question mark (?) first, then a space followed by the number of different possible values the counter may have had at 15:00:00. Separate the question mark and the number by one blank character. If it is an impossible case and the photos are contradicting with each other, you should write '? 0' in the output.

Sample Input

```
1
 -  -  -
| || || |
 -  -  -
| || || |
 -  -  -
#
0
 *  *  -
* ** *| |
 *  *  -
* ** *. |
 *  *  -
1
 .  .  *
. |. |* *
 .  .  *
. |. |* *
 .  .  *
#
6120
 -  -  -
| || || *
 -  *  -
* || || |
 -  -  -
$
```

Sample Output

```
887
109
? 8
```