

## Problem 1—Mastermind

Mastermind is a game that zombies enjoy playing because it makes them think of Master Brains. One player chooses four colored pegs (red, orange, yellow, green, blue, violet), arranges them in a row, and hides them from the other player. The first player is allowed to repeat one or more colors if s/he chooses. The second player then has to guess the pattern. Each guess made by the second player is graded by the first player using black and white pegs. Each black peg represents a correct color in a correct position. Each white peg represents a correct color in an incorrect position. Each peg in the guess (as well as each peg in the answer) results in at most one corresponding grading peg. For example, if the correct answer is BYBV and the second player guesses VGBG, the grade would be one black peg and one white peg: there is a blue in the right position yielding a black peg; there is a V in the wrong position yielding a white peg. Since the B already earned a black peg, it does *not* yield a white peg for the other blue in the answer. The two G's in the guess do not earn pegs at all since there are no G's in the correct answer. Note that grading is symmetric: Pattern A graded against B would yield the same number of black and white pegs as pattern B graded against A.

Given a list of guesses and grades, you are to compute how many possible patterns could be correct based on the information at hand. If there is only one valid answer, you are to print that answer.

**INPUT SPECIFICATION.** You will be given a set of input cases, each of which will begin with an unsigned positive decimal integer  $n$  followed by **<EOLN>**, representing the number of guesses. The next  $n$  lines will each consist of guesses and grades. There will be a four-byte guess (using the letters R, O, Y, G, B, V) followed by the number of black pegs in the grade followed by the number of white pegs in the grade. There will be one space separating the guess and the black grade and one space separating the black and white grades. Each line will be terminated by **<EOLN>**. An extra **<EOLN>** will follow each input case. The last input case will be followed by **"0<EOLN>"**.

**OUTPUT SPECIFICATION.** The output cases should appear in the same order as the input cases. Each output case will be of the form "Case  $c$ : There are  $n$  solutions." (where  $c$  is the number of the input case and  $n$ , is the number of solutions) followed by **<EOLN>** unless there is exactly one solution, in which case the output will be of the form "Case  $c$ : There is 1 solution:  $S$ ." (where  $S$  is the four-byte solution) followed by **<EOLN>**

### **SAMPLE INPUT.**

```
1<EOLN>
RRRR·0·1<EOLN>
<EOLN>
4<EOLN>
RRRR·3·0<EOLN>
BBBB·0·1<EOLN>
BGBB·0·1<EOLN>
BBBG·0·1<EOLN>
<EOLN>
0<EOLN>
<EOF>
```

### **SAMPLE OUTPUT.**

```
Case·1:···There·are·0·solutions·.<EOLN>
Case·2:···There·is·1·solution:··RRGR·.<EOLN>
<EOF>
```