PROBLEM: Descended from other Latin square puzzle types, Skyscrapers is a Japanese creation. It received its first broad exposure at the 1st World Puzzle Championship in NYC in 1992 when publisher Sekai Bunka-sha presented a 20 page English edition of their Puzzler magazine to the competitors. In the USA it has been enhanced by Kevin Stone.

Each puzzle consists of an NxN grid with some clues along its sides. This program will use a 4 x 4 grid as shown below. The object is to place a skyscraper in each square, with a height from 1 to 4, so that no two skyscrapers in a row or column have the same height. In addition clues are given. The clues are the numbers outside of the grid. The number of visible skyscrapers, as viewed from the direction of each clue, is equal to the value of the clue.

```
        3       2       2       1

   4 |  1  |   2   |   3   |   4  | 1

   2 |  2  |   4   |   1   |   3  | 2

   1 |  4  |   3   |   2   |   1  | 4

   2 |  3  |   1   |   4   |   2  | 2

        2       3       1       3
```
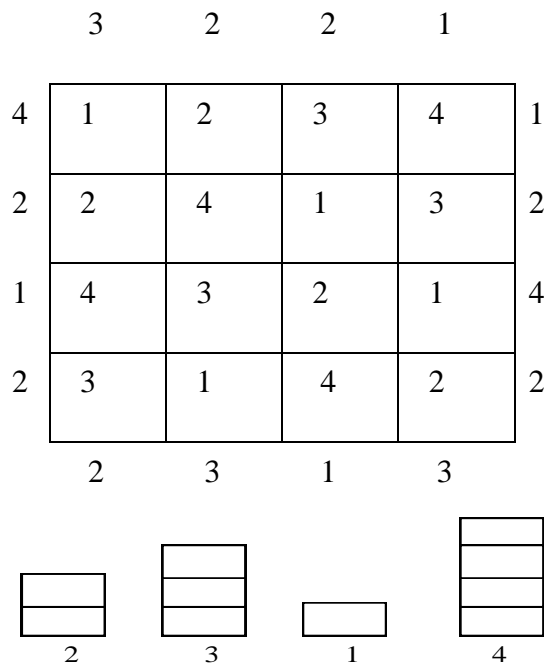
In the example above with the stacked rectangles representing the skyscrapers, looking up from the left you can see the #2, #3 and #4 skyscrapers. The #1 skyscraper is blocked by the number #3 skyscraper. Looking up from the right, the #4 skyscraper blocks all the others. Therefore the clue numbers would be 3 and 1.

INPUT: There will be 6 lines of input. The first line will contain 6 numeric character strings giving the data for the clues and the grid data for one row. The first and last strings will have 4 characters giving values for the top and bottom clues. The other strings will either be 2-character strings giving the clue numbers for that row or a 6-character string giving the clue numbers and the data for that row. Sample Input #1 gives the data for the table above with the data for the bottom row. The next 5 lines will each contain a 2-character numeric string giving a row-column number. In the grid above location 11 is a 1.

OUTPUT: For each row-column number print the correct grid value at that location.

SAMPLE INPUT	SAMPLE OUTPUT
1. 3221, 41, 22, 14, 231422, 2313	1. 4
2. 14	2. 1
3. 34	3. 1
4. 23	4. 3
5. 32	5. 2
6. 21	