A certain puzzle consists of four tiles, as shown in the first figure below. To solve this puzzle, the tiles must be rotated (but not moved) so that wherever two tiles share a border, either both tiles have a pipe at that border, or neither does. The second figure below shows one possible solution. Note that the presence or absence of a pipe at exterior edges doesn’t matter.

You are to write a program that takes as input a description of such a puzzle, and outputs a solution if one exists, or if not, a message indicating this. An input consists of four sequences of four digits, each of which is either 0 or 1. Each sequence represents a tile, starting with the upper left, then the upper right, lower left, and finally lower right. The digits in a sequence indicate the presence or absence of a pipe on an edge, beginning with the top edge and working clockwise. A 1 indicates the presence of a pipe, and a 0 indicates the absence of one. Thus, the first figure below is represented by 1011, 1100, 0000, 0101. An output describing a solution consists of four numbers, each in the range 0-3. These numbers each give a number of clockwise turns applied to one of the tiles. These numbers are given in the order upper left, followed by upper right, lower left, and finally lower right. Thus, the solution shown below is described by 1, 2, 0, 1 (or 1, 2, 1, 3, etc. — the number of rotations of the lower left tile is irrelevant, and 1 rotation of the lower left tile is equivalent to 3 rotations).

Example 1:

Enter upper left: 1011
Enter upper right: 1100
Enter lower left: 0000
Enter lower right: 0101
The following rotations work:
Upper left: 1
Upper right: 2
Lower left: 0
Lower right: 1

Note: Other solutions to the above example exist.

Example 2:

Enter upper left: 1011
Enter upper right: 0000
Enter lower left: 0000
Enter lower right: 0011
There is no solution.