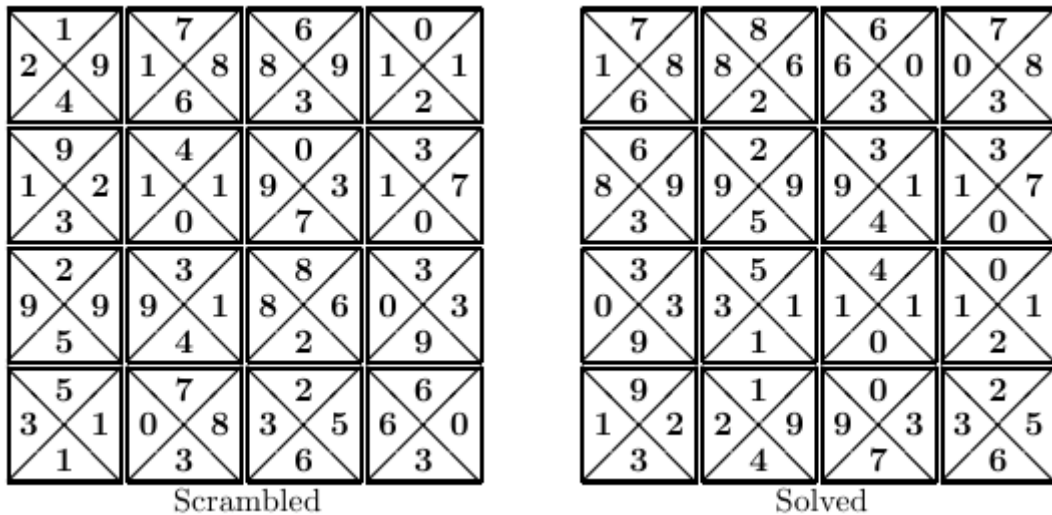


## 954 Tetravex Solver

In the so-called tetravex puzzle,  $MN$  square pieces are to be placed side-by-side inside an  $M \times N$  rectangle so that touching sides have the same number.

Given 16 different square pieces, arrange them in a  $4 \times 4$  square. Touching sides must have the same number. The following figure illustrates what is desired.

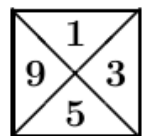


### Input

The input will contain several test cases, each of them as described below. Consecutive test cases are separated by a single blank line.

Each piece is encoded by four consecutive one digit numbers, each of which represented in the ASCII character set; 0 will be represented by `ascii(48)`, i.e., by '0' in the C language, 1 by `ascii(49)`, and so on, up to 9 (`ascii(57)`). The four digits represent, from left to right, the one digit numbers associated with the right, up, left, and down sides.

For example, the string "3195" (the double-quotation marks will not be present in the input) represents the square piece shown on the right.



This representation must also be used in the output. There must be at least one white space character between the piece representations. A white space character is either a space, i.e., `ascii(32)`, ' ' in the C language, or a new line character (also known as the line feed character), i.e., `ascii(10)`, '\n' in the C language. There may exist white space characters both before the description of the first piece and after the description of the last piece. If any other characters (i.e., not a digit nor a white space) are present then the input is not valid. The input is also not valid when there are repeated pieces. Two pieces are equal if and only if they have the same representation.

### Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

The first line of the output must contain a single digit, represented in the ASCII character set, immediately followed by the new line character. This digit must be

- 0, if the input is not valid, or

- 1, if the input is valid and there are no solutions, or
- 2, if the input is valid and there is only one solution, or
- 3, if the input is valid and there are two or more solutions.

If there is only one solution, the output must have 4 more lines, each of which containing the pieces of one row of the solution, top row first. Each of these 4 lines must contain the representation of the 4 pieces of the corresponding row, left piece first. There must exist a single space character between piece representations of the same row. Each line must be terminated by a new line character.

### Sample Input

```
13e2 6151 8161 1201 6152 8777
1651 1511 1394 7310 8726 1128
7151 5512 1512 7161
```

```
    9124
   8716 9683
  1012 2913 1410
3097 7310 9295 1394
 6882 3309 1531
   8703 5236
    0663
```

```
0512 9811 3333 1726 1755 0052
7614 6152 1987 1234 7771 1390
2185 2468 1103 7188
```

```
3298 6428 6833 8484 8924
9886 8679 4687 8929 2642 8999
9399 9784 6986 7477 7406
```

### Sample Output

```
0

2
8716 6882 0663 8703
9683 9295 1394 7310
3309 1531 1410 1012
2913 9124 3097 5236
```

```
1
```

```
3
```