

Rotations and Reflections

Many games, tricks and puzzles depend on determining whether two patterns on a rectangular grid are the "same" or not. For instance, the 96 different ways of arranging 8 queens safely on a chessboard can be shown to consist of rotations and/or reflections of only 12 basic patterns.

Write a program that will read in pairs of patterns and determine whether there is a simple transformation that will convert one into the other. Because symmetrical patterns bear many relationships to each other, the transformations must be checked in a specific order. The possible transformations (in order) are:

Preservation	The patterns are identical
90 degree rotation	The pattern was rotated clockwise by 90 degrees
180 degree rotation	The pattern was rotated clockwise by 180 degrees
270 degree rotation	The pattern was rotated clockwise by 270 degrees
Reflection	The pattern was reflected about the horizontal axis (effectively by a mirror held at the top of the pattern)
Combination	A reflection (as above), followed by one of the above rotations
Improper	The patterns do not match under any of the above transformations

Input

Input will consist of a series of pairs of patterns. Each set will consist of a line containing a single integer N ($2 \leq N \leq 10$) giving the size of the patterns, followed by N lines. Each line will consist of N dots or 'x's (specifying a line of the original pattern), a space, and another set of N dots and 'x's (specifying a line of the transformed pattern). The file will be terminated by a line consisting of a single zero (0).

Output

Output will consist of a series of lines, one for each pattern pair in the input. Each line will consist of one of the following: 'Preserved', 'Rotated through m degrees' (where m is one of 90, 180 or 270), 'Reflected', 'Reflected and rotated through m degrees', 'Improper'.

Sample Input

```

5
x...x  ....x
.x...  ....x.
...x.  .x...
..x.x  ..x..
.....x xx...x
2
x.  xx
x.  xx
4
..x.  ....x
xx...  ....
....  xx..
...x  ..x.
4
.x..  ..x.
.x.x  x...
....  ..xx
..x.  ....
0

```

Sample Output

```

Rotated through 90 degrees
Improper
Reflected
Reflected and rotated through 270 degrees

```