# Problem 1—Pegs

Tyrion Lannister had the rare privilege of visiting a Westerosian Cracker Barrel not long ago. Not terribly impressed with the Farmer's Breakfast, he did take a liking to the peg game. The peg game consists of fifteen holes arranged in an equilateral triangular shape, in rows of one, two, three, four, and five. There are fourteen pegs initially placed in the holes so that only one hole is empty. The point of the game is to "jump" over a peg with another peg into an empty hole, removing the jumped peg from the board in the process (very similar to a jump in checkers in which the jumped checker is removed from the board). Jumps have to occur in a direction parallel to the sides of the triangle. Tyrion will win if he can reduce the number of pegs to one as a result of jumps.

You are to solve this game for Tyrion for various sized triangles and for various initial empty holes.

**INPUT SPECIFICATION.** Each input case contains three integers separated by one space and followed by **<EOLN>**. These three integers are, in order, the number of holes along each side of the triangle, the row number of the initial hole (numbered from zero from the top), and the column number of the initial hole (numbered from zero from the left). The last data case is followed by "0**<EOLN>**". This 0 is not to be processed; it just signals the end of input.

**OUTPUT SPECIFICATION.** The output cases are to be processed in the same order as the input cases. Each case begins with "Case *c***<EOLN>**" where *c* is the case number. If the game cannot be won from the given input configuration, "UNSOLVABLE**<EOLN><EOLN>**" is to be printed. If the game can be solved, then every configuration from the starting configuration to the final configuration is to be printed, using the example below, with pegs represented by the vertical stile and holes represented by the capital O. Note that an extra **<EOLN>** is to be printed after each configuration.

There may well be more than one way to solve the game, so given more than one possible winning move from any configuration, choose the peg with the lower row number over the peg with the higher. If two pegs from the same row can be a valid next move leading to a win, choose the one with the lower column number over the one with the higher. If the same peg can be moved in more than one direction to win, a south move is favored over a north move. North is favored over east, east is favored over west, west is favored over southeast, southeast is favored over northwest.

## SAMPLE INPUT.

```
4·1·0<EOLN>
3·1·1<EOLN>
0<EOLN>
<EOF>
```

## SAMPLE OUTPUT.

```
Case·1<EOLN>
|<EOLN>
O|<EOLN>
|||<EOLN>
||||<EOLN>
<EOLN>
|<EOLN>
||<EOLN>
O||<EOLN>
O|||<EOLN>
<EOLN>
O<EOLN>
O|<EOLN>
|||<EOLN>
O|||<EOLN>
<EOLN>
|<EOLN>
OO<EOLN>
||O<EOLN>
O|||<EOLN>
<EOLN>
|<EOLN>
OO<EOLN>
OO|<EOLN>
O|||<EOLN>
<EOLN>
|<EOLN>
O|<EOLN>
OOO<EOLN>
O||O<EOLN>
<EOLN>
O<EOLN>
OO<EOLN>
OO|<EOLN>
O||O<EOLN>
<EOLN>
O<EOLN>
OO<EOLN>
OO|<EOLN>
OOO|<EOLN>
<EOLN>
O<EOLN>
O|<EOLN>
OOO<EOLN>
OOOO<EOLN>
<EOLN>
Case·2<EOLN>
UNSOLVABLE<EOLN>
<EOLN>
<EOF>
```