Problem 3—Light Cycles

written by Sir Michael R. Kowalczyk, Seeker of the Grail

The red team is having a light cycle competition among themselves to decide who will be the light cycle jockey to face off against Tron: light cycles are digitized motorcycles within the video game universe that leave solid immovable walls of light called "jet walls" in their wake.

The competition occurs as a series of discrete time steps in a discrete rectangular game grid. At the beginning of each step, each light cyclist observes the current state of the immediate surrounding squares and uses this information to decide whether to advance one square in the same direction, to turn left and advance one square, or to turn right and advance one square. Once all players have made their decisions, they simultaneously perform their chosen maneuvers, each leaving a jet wall to occupy the square they had just left. Any light cycle which collides with a wall, jet wall, or another cycle is subject to deresolution (i.e. complete annihilation) before the next time step: it is removed from play, along with ALL jet walls that were produced by that cycle. (It does not matter whether a jet wall is parallel or perpendicular to the light cycle. If a light cycle enters a square with any jet wall whatsoever, including its own, it is derezzed.)

Red 1 starts out facing the right side of the game grid.
Red 2 starts out facing the left side of the game grid.
Red 3 starts out facing the bottom side of the game grid.

Red 1 and 2 use the following decision procedure on each turn:

*Move forward if there is currently a blank square there,
*otherwise if there is currently a blank square immediately to the right, turn 90° right and advance,
*otherwise turn 90° left and advance.

Red 3 uses the following decision procedure on each turn:

*Move forward if there is currently a blank square there,
*otherwise if there is currently a blank square immediately to the left turn 90°left and advance,
*otherwise turn 90° right and advance.

INPUT SPECIFICATION. You will be given a set of input cases. Each case consists of two positive integers, separated by one space, giving the width and height of the game grid, respectively. This is followed by a visual depiction of the game grid with walls represented by #, empty squares represented by spaces, and each cycle represented by its number. Each of the three cycles is included exactly once in each game grid, and each game grid has walls around the edges, preventing the cycles from escaping. "0 0<**EOLN**>" will follow the last case.

OUTPUT SPECIFICATION. For each case, give the number of the surviving light cycle, followed by <**EOLN**>. If all light cycles were derezzed, then give 0 followed by <**EOLN**>.

SAMPLE INPUT.

```
5·4<EOLN>
#####<EOLN>
#·3·#<EOLN>
#1·2#<EOLN>
#####<EOLN>
9·6<EOLN>
#########<EOLN>
##······#<EOLN>
#·#··3··#<EOLN>
#·#····##<EOLN>
#·1··2··#<EOLN>
#########<EOLN>
0·0<EOLN>
<EOF>
```

SAMPLE OUTPUT.

```
0<EOLN>
1<EOLN>
<EOF>
```