

Problem 6—Pac Man

Namco first came out with “Pac Man” in 1980. It consisted of a square game grid, one Pac Man, and 4 ghosts (Blinky, Pinky, Inky, and Clyde). At any point in time, each ghost had a “target” square in mind, and would try to move towards this target square (by giving each ghost different rules for choosing the target square, the original writer of Pac Man was able to give the ghosts different personalities). You will implement a simplified version of these movement rules.

Ghosts are always moving and never reverse direction. Thus if a ghost is adjacent to two walls, there is only one possible place to go. When a ghost reaches a 3 or 4-way intersection, it will choose from the 2 or 3 available adjacent squares; whichever minimizes the Euclidean distance to the target square (as measured from the center of each square). In the case of a tie for shortest distance, the tie is broken among the shortest distances by giving preference to moving up, followed by left, and then down (thus ghosts never go right in a tie for shortest).

INPUT SPECIFICATION.

You will be given a set of input cases. Each case starts with 2 positive integers, separated by one space; these are the width and height of the game grid, respectively. Pinky the ghost is represented by a P, and her previous location is represented by a p. The target square is represented by an uppercase X. There is no wraparound and the input maze will not have any dead ends. “Up” when viewing the input file corresponds to “up” in the movement behavior described above. “0 0<EOLN>” will follow the last case.

OUTPUT SPECIFICATION.

For each case, you are to output the state of the game one time unit later, followed by an extra <EOLN>. Only indicate the walls, spaces, and Pinky's new location. Do not mark the target square and do not mark Pinky's previous square.

SAMPLE INPUT

```
5 5<EOLN>
*****<EOLN>
*X.*.*<EOLN>
*p*.*.*<EOLN>
*p*.*.*<EOLN>
*****<EOLN>
5 5<EOLN>
*****<EOLN>
*X.*.*<EOLN>
*p*.*.*<EOLN>
*p*.*.*<EOLN>
*****<EOLN>
8 7<EOLN>
*****<EOLN>
*****.X*.*<EOLN>
*****.*.*<EOLN>
*...P.*.*<EOLN>
*...p*.*.*<EOLN>
*...*.*.*<EOLN>
*****<EOLN>
0 0<EOLN>
<EOF>
```

SAMPLE OUTPUT

```
*****<EOLN>
*...*.*<EOLN>
*...*.*<EOLN>
*.P.*.*<EOLN>
*****<EOLN>
<EOLN>
*****<EOLN>
*p*.*.*<EOLN>
*...*.*<EOLN>
*...*.*<EOLN>
*****<EOLN>
<EOLN>
*****<EOLN>
*****.*.*<EOLN>
*****P.*.*<EOLN>
*...*.*.*<EOLN>
*...*.*.*<EOLN>
*...*.*.*<EOLN>
*****<EOLN>
<EOLN>
<EOF>
```