

Problem 2—Invisibility

Frodo needs to get to Mount Doom to drop off the ring. The problem is that between him and Mount Doom are orc archers who will shoot him if they see him. To avoid the orc archers, Frodo can wear the One Ring, which will make him invisible, but if he wears the ring too long, the ring will corrupt him so that he will lose interest in going to Mount Doom. You are to determine whether Frodo can make it to Mount Doom.

You will read in a series of grids. Each grid contains an “F” representing Frodo and a “D” representing Mount Doom. Walls are represented by “*”; the grid is always surrounded by walls, but there may be internal walls as well. There may be any number of orcs, represented by “O”. Orcs do not move. Orcs look horizontally and vertically but not diagonally. If they see Frodo, they will kill him. An orc cannot see Frodo if he is invisible or if there is a wall between them along the orc's line of sight. If Frodo exceeds the maximum usage of the ring, Frodo will not make it to Mount Doom. Frodo cannot cross walls, nor can Frodo cross an orc, invisible or not. Frodo's initial location and Mount Doom are guaranteed *not* to be in an orc's line of sight. Frodo can put on or take off the ring as many times as he likes. Frodo moves one square horizontally or vertically on each move. We think of him moving and donning or removing the ring as being simultaneous operations: at any given time, Frodo is at one specific grid location and the ring is either on or off.

INPUT SPECIFICATION. The input file will consist of a number of cases. Each case consists of two integers separated by one space and followed by <EOLN> representing the number of rows and columns (not exceeding 20) in the grid. The grid itself follows: the appropriate number of rows each containing the appropriate number of columns and each terminated by <EOLN>. Following the grid is an integer representing the maximum number of acceptable uses of the ring, followed by <EOLN>.

OUTPUT SPECIFICATION. The output cases should appear in the same relative order as the input cases. Each case consists of either “Case *c*: Frodo can get to Mount Doom in *d* moves.” (where *c* is the case number and *d* is the smallest number of moves Frodo needs to make) or “Case *c*: Frodo cannot get to Mount Doom.” Each case should be followed by two <EOLN>'s.

SAMPLE INPUT.

```
10 10<EOLN>
*****<EOLN>
*.....*<EOLN>
*.....*<EOLN>
*..O.....*<EOLN>
*.....*<EOLN>
*.....*<EOLN>
*.....*<EOLN>
*.....*<EOLN>
*..F..D...*<EOLN>
*.....*<EOLN>
*****<EOLN>
1<EOLN>

10 10<EOLN>
*****<EOLN>
*.....*<EOLN>
*.....*<EOLN>
*..O.....*<EOLN>
*.....*<EOLN>
*.....*<EOLN>
*.....*<EOLN>
*.....*<EOLN>
*..F..D...*<EOLN>
*.....*<EOLN>
*****<EOLN>
0<EOLN>
0 0<EOLN>
<EOF>
```

SAMPLE OUTPUT.

```
Case 1: Frodo can get to Mount Doom in 3 moves.<EOLN>
<EOLN>
Case 2: Frodo cannot get to Mount Doom.<EOLN>
<EOLN>
<EOF>
```