



# 2005 South Central USA Regional Programming Contest



## *Pegs*

### Introduction:

Peg games may be played on a variety of boards, but the goal is always the same, to end the game with as few pegs on the board as possible. This is accomplished by making a sequence of moves, where one peg 'jumps' an adjacent peg to the empty space on the opposite side. Pegs which are jumped by other pegs are immediately removed.

Given the starting configuration of a peg board, your program should determine the number of pegs that will remain on the board if the player makes the best possible sequence of moves.

One standard peg board is a 5x5 cross (shown below with periods indicating empty peg holes and pound signs indicating non-playable space). Figure A shows an empty board. Figure B shows a board with five pegs, indicated by lower-case letter *o*s. and Figure C shows the result of an optimal sequence of moves from Figure B's starting position. Note that this ends with 1 peg, the best possible result for any game. Also note that the optimal sequence of moves used to generate Figure C is not unique; there are at least two sets of moves that leave only one peg.

#...#	#.o.#	#...#
.....	..o..	.....
.....	oo.o.	...o.
.....	.....	.....
#...#	#...#	#...#
Figure A	Figure B	Figure C

Note that not all of the peg boards given to you will be the same shape! They will all be 5x5, and will all have at least one peg, at least one empty cell, and at least four non-playable spaces, but the layout may differ drastically from the above.

### Input:

Input to this problem will begin with a line containing a single integer  $n$  indicating the number of boards that require analysis. The following  $5n$  lines contain the boards. Each board has a representation like that in the figure above, with a period (.) representing an empty space, lower case *o* representing a peg, and # representing a non-playable space.

### Output:

For each board in the input, print the message "The best case ends with  $Y$  pegs." where  $Y$  is replaced

with the number of pegs remaining after an optimal set of jumps.

### Sample Input:

```
3
#.o.#
..o..
oo.o.
.....
#...#
#...#
o.o.o
....o
...o.
#o..#
#..##
.o..#
ooo.o
.o.o.
#..o#
```

### Sample Output:

```
The best case ends with 1 pegs.
The best case ends with 4 pegs.
The best case ends with 2 pegs.
```

The statements and opinions included in these pages are those of *Hosts of the South Central USA Regional Programming Contest* only. Any statements and opinions included in these pages are not those of *Louisiana State University* or the *LSU* Board of Supervisors.

© 1999, 2000, 2001, 2002, 2003, 2004, 2005 Isaac Traxler