

Computer Science 320SC – (2016)

Assignment 4 (programming)

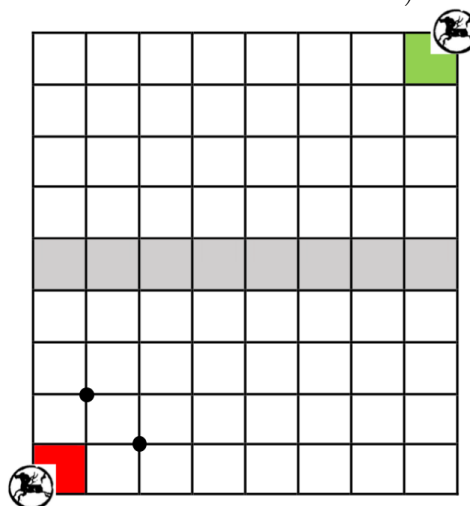
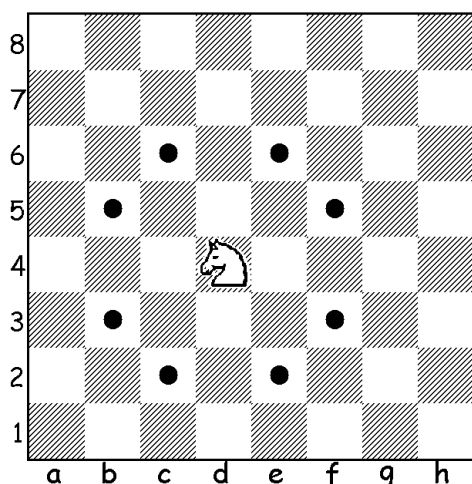
Due: Sunday, September 25th (9pm)

Knight's Shortest Path

Limit: 5s

This problem is taken from last year's New Zealand Programming Contest and will test your understanding of dynamic programming.

Master Li likes to play various types of chess and was thinking about how efficient the knight piece moves between positions on a chess board. It is known that the knight travels (in one move) two cells in one direction then one cell in the other axis as shown in the international 8×8 chessboard shown below on the left. There are also various flavors of chess, including the Chinese version (xiangqi) shown on the right, where the knight is placed on the intersections of lines (equivalent to a 10×9 cell-based board).



With various sizes of boards, Master Li wants to know the minimum number of knight moves to travel from opposite corners of a board (e.g. go from cell 1a to cell 8h). In addition, if there is more than one possible path he wants to know the total number of different minimum paths. Can you help him calculate this information?

Input

The input consists of a series of up to 100 test cases. Each test case is a line consisting of two integers $1 \leq r \leq 200$ and $1 \leq c \leq 200$, denoting the number of cells in vertical direction (rows) and horizontal direction (columns) for the chessboard, respectively. The input is terminated by a test case that does not have a valid knight path that reaches the opposite corner and no reported results is required for this case.

Output

For each test case with a valid knight path, output a single line showing the smallest number of knight moves and a count of the number of distinct paths of that shortest distance between two opposite corners.

Sample input and output

Input	Output
1 1	0 1
4 4	2 2
4 5	3 1
2 4	

Submission

For this assignment name your source code `knightE.ext` and `knightH.ext` where `ext` denotes one of { `java`, `cpp`, `py` } that indicates java/c++/python language. Please use just one source file per problem. Here the suffix `E` denotes 'E'asy (test data) and `H` denotes 'H'arder (test data). Three marks is allocated to `knightE.ext` and two marks are allocated for `knightH.ext`.