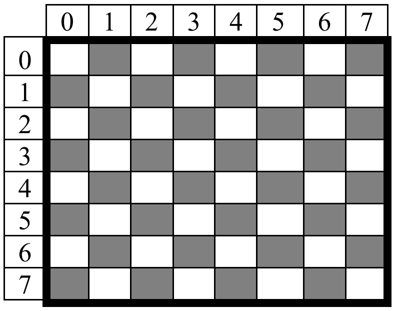
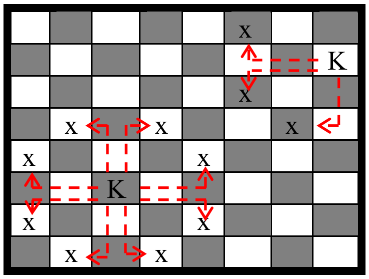
Knight Attack Minimum

Consider a chessboard that is represented by a two-dimensional 8x8 array a[][] in which a[0][0] represents the board's upper left corner.

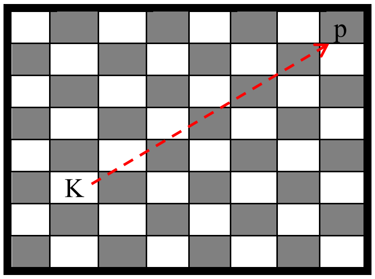


In chess, a knight moves in an L-shape, meaning, it moves two squares vertically followed by one square horizontally, *or*, two squares horizontally followed by one square vertically.



We place one knight and one pawn on a chessboard and stipulate *the pawn <u>never</u> moves*.

What is the minimum number of moves the knight must make to capture the pawn?



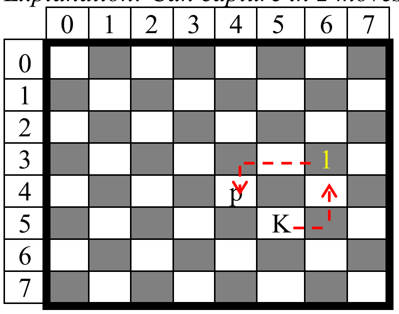
Minimum moves of the knight?

Write a function
`int captured(int knight_r, int knight_c, int pawn_r, int pawn_c)`
where
`knight_r,knight_c` is the knight's staring location
`pawn_r,pawn_c` is the pawn's staring location
and returns the minimum number of moves for the knight to capture the pawn if
(0 <= `knight_r,knight_c,pawn_r,pawn_c` <= 7) and the pieces are not placed on the same square,
otherwise returns -1.

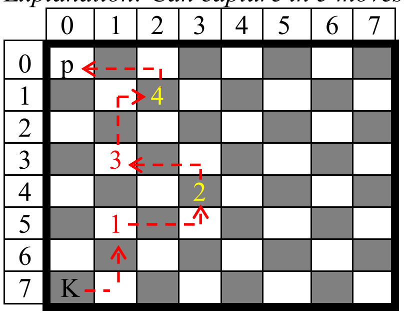
File you must submit: `soln_func.cc`

Examples:

knight_r=5,knight_c=5,   pawn_r=4,pawn_c=4
Returns: 2
Explanation: Can capture in 2 moves.



knight_r=7,knight_c=0,   pawn_r=0,pawn_c=0
Returns: 5
Explanation: Can capture in 5 moves.


0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
p
4
3
2
1
K