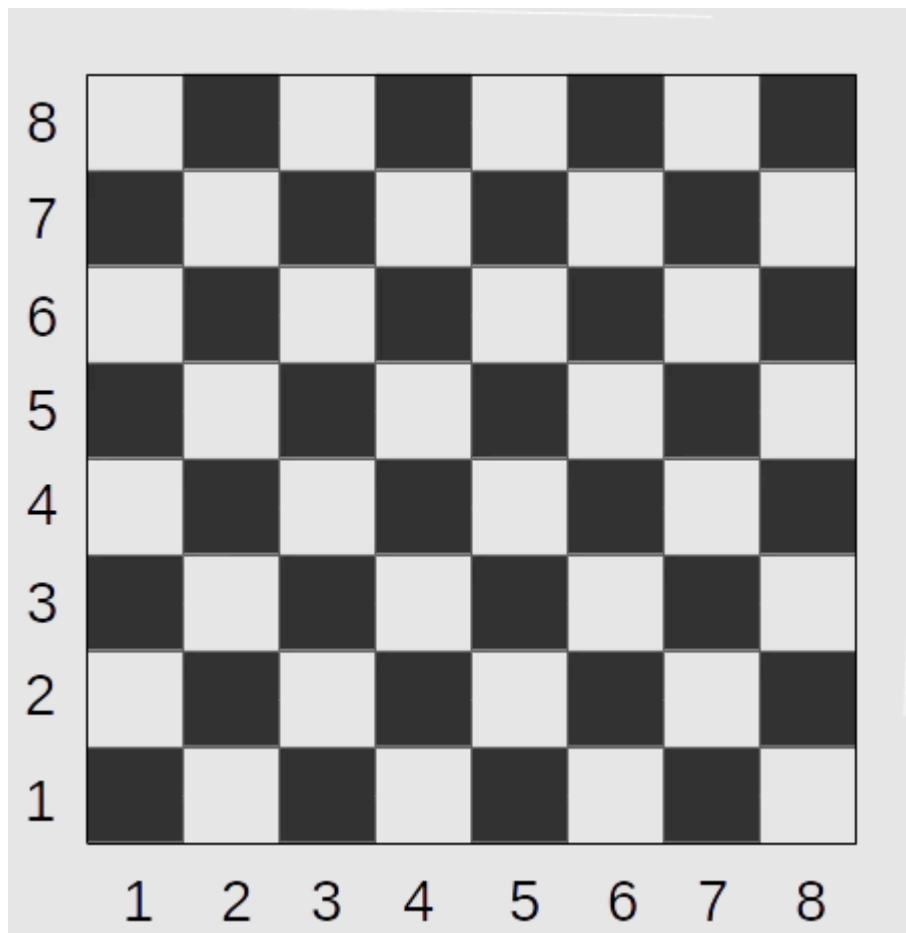# Challenge

Long live the king

Let's imagine a chessboard, the squares on which are marked with coordinates. The coordinates are numbers between 1 and 8 inclusively. The first number indicates a column, the second one indicates a row.

The chess king can stand on any square and can move one step horizontally, vertically, or diagonally in any direction within the board.

The input will contain the coordinates on which the king is located. You should figure out and print how many moves the figure can make: for example, from the position (1, 8), the king can make only 3 moves (right, down, diagonally).

Here's the steps you can follow:

1. Start by reading the coordinates of the king from the input.

2. To determine the number of possible moves for the king, consider all possible directions: horizontally (left and right), vertically (up and down), and diagonally (up-left, up-right, down-left, down-right).

3. Iterate through each possible direction and check if the king can move to the corresponding square. Make sure to verify that the new position is within the bounds of the chessboard (coordinates between 1 and 8 inclusively).

4. Keep track of the count of valid moves.

5. Finally, print out the total count of valid moves.

Sample Input 1:

```
In [ ]:  3
         2
```

Sample Output 1:

```
In [ ]:  8
```

First approach

```
In [18]:  col = int(input())
          row = int(input())
          input_tuple = (col, row)

          # Only 3 moves - King on a corn of the chessboard
          three_moves = [(1, 1), (1, 8), (8, 8), (8, 1)]

          # Only 5 moves - King on a border but not in the corn of the chessboard
          five_moves = [
              (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 1), (2, 8), (3, 1), (3,
              (4, 1), (4, 8), (5, 1), (5, 8), (6, 2), (6, 8), (7, 1), (7, 8), (8, 2), (8,
              (8, 4), (8, 5), (8, 6), (8, 7)
          ]

          # King on a corn of the chessboard
          if input_tuple in three_moves:
              print(3)
          #  King on a border but not in the corn of the chessboard
          elif input_tuple in five_moves:
              print(5)
          # King in the center of the chessboard
          else:
              print(8)
```

8

Last Approach

```
In [24]:  col = int(input())
          row = int(input())

          # King on a corn of the chessboard
          if (col == 1 or col == 8) and (row == 1 or row == 8):
              print(3)
          # King on a border but not in the corn of the chessboard
          elif col == 1 or col == 8 or row == 1 or row == 8:
```

```
    print(5)
# King in the center of the chessboard
else:
    print(8)
```

3