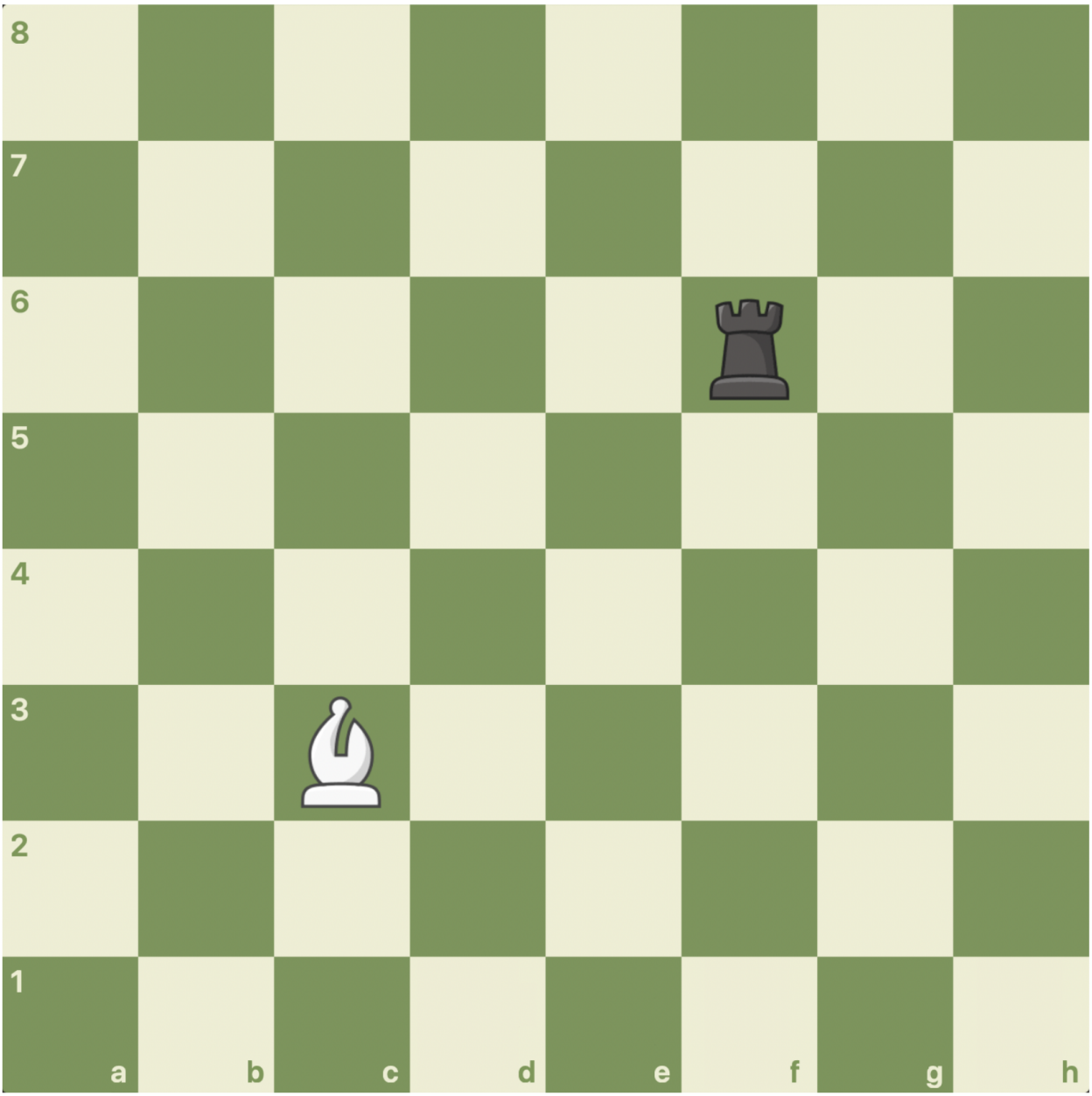
Rock vs Bishop



We are looking at a 'special' game of chess where each player only has one piece. Player with white pieces only has a bishop while player with black pieces only has a rook.

- In chess, the columns (called Files) are denoted as alphabets(a-h) and rows (called Ranks) are denoted as numbers (1-8).
- Any given square is a combination of file and rank. In the above figure the position of bishop is c3 while that of the rook is f6.

- Bishop only moves diagonally from its position, in all 4 directions (top-right, top-left, bottom-right, bottom-left).
- Rook moves either vertically (up and down) or horizontally (left and right) from its given position.
- Both rook and bishop can capture one another only if the other piece falls in the valid motion path.

Set Up

In the object-oriented language of your choosing

1. Create classes/objects for the two pieces
2. Create an appropriate representation of board state
3. Given a board state, write code to determine whether the rook can capture the bishop.
4. Write code to determine whether the bishop can capture the rook.

The Problem

We decide to move the black rook and play for its survival. The move happens as follows:

1. Toss a coin, if it's heads, the rook moves up. If it's tails, the rook moves to the right.
2. Roll 2 dice (6 sided). The sum of numbers on the face up side of both the dice will be the number of squares the rook moves.
3. If the rook reaches the right most column on the board, it emerges again from the left most column.
4. If the rook reaches the top most row, it emerges again from the bottom most row.

- Move the rook as described above for 15 rounds. If it manages to survive from being captured by the bishop, the player with the rook wins. Else the player with bishop wins.
- The starting position for rook is h1 square and bishop remains stationary on c3.

Write code to determine which player won, given the above constraints. Make sure to record (or print) the result of coin toss, dice and rook's position after every move.