For this exercise, you will be given 3 hours to implement the game of EverChess. You may use any language you want, in any programming environment. You have full access to the internet, except for things directly related to existing implementations of chess.

What we're looking for

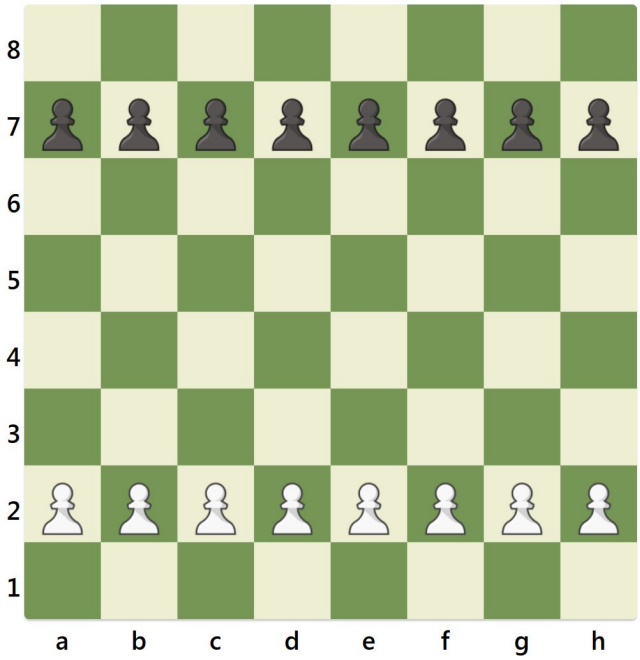
- Clean, readable, well-planned-out code for the game logic
- Some system of input-output (command line is fine, anything else is also fine) that allows the game to be played and the users to see the board state
- A game that functions as specified in the rules below
- If you're running low on time, a plan for implementing the rest of your solution

What we're NOT looking for

- We don’t expect a pretty user interface. The bare minimum of playability is fine, such as a command line interface using text to represent the board and pieces.
- We don’t expect optimal performance. Clean, well-organized code is more important than speed.
- You don’t need to use a complex development environment or complex libraries. Only use things that you're comfortable with and that are necessary.
- We don’t expect excessive documentation. You should provide as much or as little documentation as you're comfortable with, but clean code should be fairly self-explanatory, and spending too much time documenting instead of coding is a liability.
- You don’t have to use a particular programming language. We prefer for you to use any language that you’re familiar with.

Overview and setup

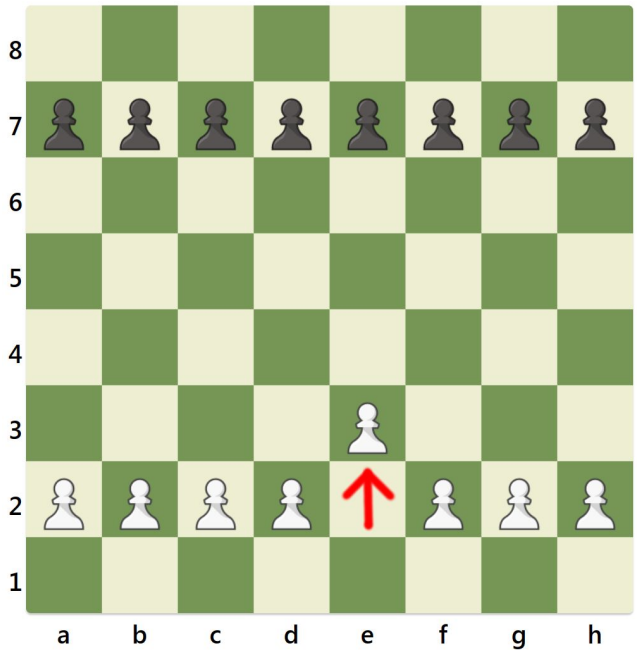
- EverChess is played by two players (white and black) on an 8 x 8 board.
- Each player starts with a row of eight pawns in the row second-closest to them, like so:



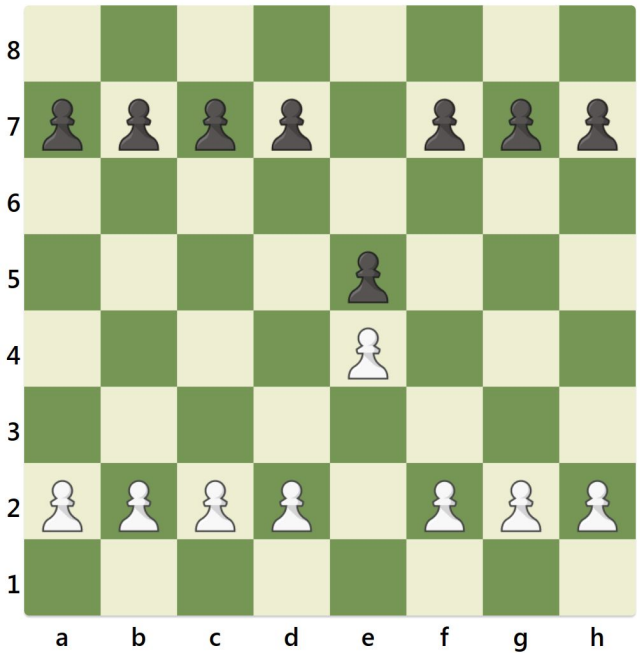
- Play alternates, with white going first. (More on movement and play below.)
- The first player to get a pawn to the opposite side of the board (row 8 for white, row 1 for black) wins.

Movement

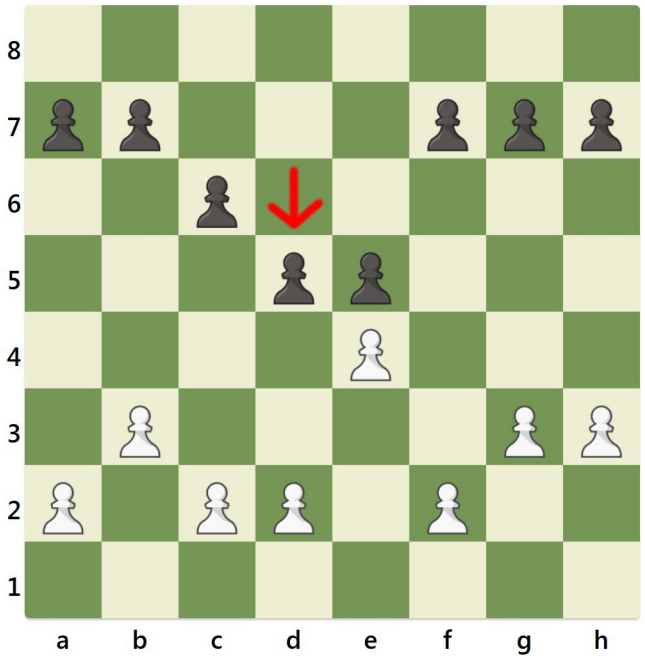
- A pawn can move forward exactly one square, provided there is no other pawn occupying that space. White's first turn, for example, could be to move the pawn in column "e" forward one square to row 3:



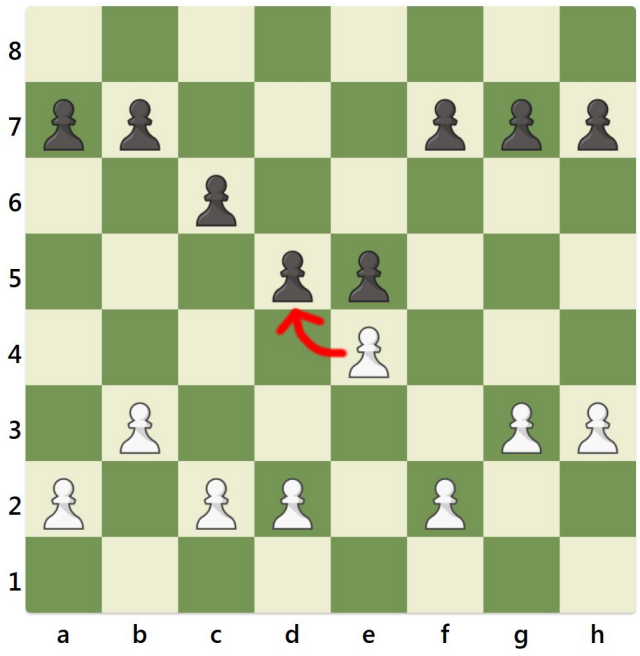
- If, after two full turns, both players have moved their pawns in column "e" forward two spaces, those pawns would not be able to move anymore because they block each other's path:

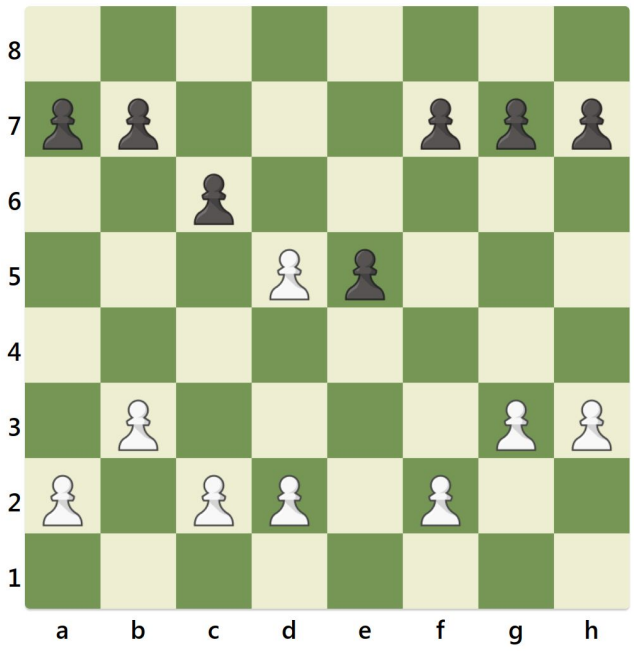


- Pawns can capture an opponent's pawn by moving diagonally into a square occupied by the opponent. The opponent's pawn is then removed from the board. This is the only time a pawn can move diagonally. Suppose that on turn five, black moves his pawn in column "d" forward to row 5:



Now white's pawn at e4 can capture black's pawn by moving diagonally into location d5:





Play

- If a player has a capturing move available, the player **must** capture the opponent's pawn. In the above scenario, white **must** capture black's pawn at location d5.

- When a player captures an opponent's pawn, that player gets to **immediately move again with a different pawn**. In the scenario above, after capturing black's pawn, white immediately moves any pawn other than the one at d5.
- If, after a capture, the player still has another capture available (with a different pawn), then similarly a capture must be made and the player gets to immediately move again **with a pawn that hasn't yet moved that turn** (i.e., the player can't go back and move the first pawn that was already moved).
- If a player gets a pawn to the other end of the board, that player immediately wins.
- If a player has no move available, the game immediately ends and the player loses.

Summary

- White moves first.
- Pawns can move forward one space to an empty location, or diagonally one space to capture an opponent's pawn at that location.
- A player must capture an opponent's pawn if a capturing move is available.
- On any capturing move, the player immediately moves again with another pawn that has not yet moved that turn.
- The first player to get a pawn to the opposite end of the board wins, unless a player is left without any move, in which case the other player immediately wins.