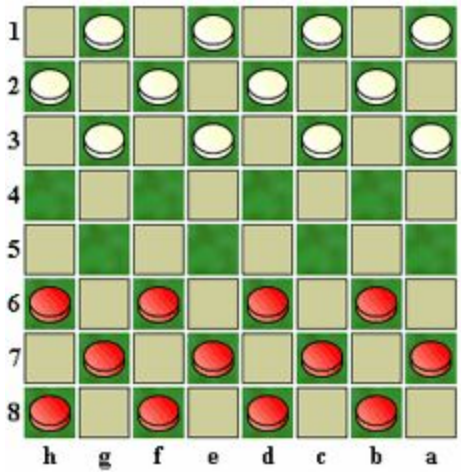
For this exercise, you will be given 3 hours to implement a slightly-simplified version of the game Checkers. You may use any language you want, in any programming environment. You have full access to the internet, excepting things related to existing implementations of the game. Email your solution to dev@everlaw.com.

We do not require you to implement kings or to check that the game has ended. Instead, please direct your focus on implementing the core logic in a robust, clean way. In particular, please be aware of the lesser known rule that if you can jump, then you must jump (more on that below).

Here are the rules of the game:

- It is played by two players seated on opposite sides of an 8x8 board of alternating light and dark squares.

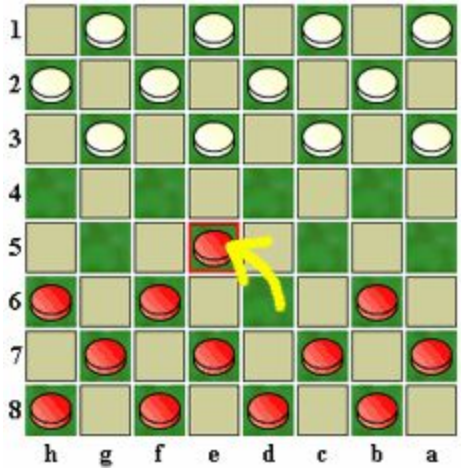
- A player's pieces are placed on the first three rows (counting from the row closest to each player) on alternating squares. Here is an example starting board:



- It is a turn-based game, and the red player begins play.

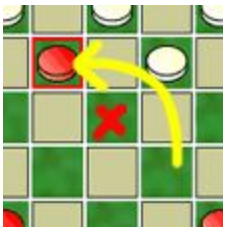
- On your turn, you may perform a **move** or a number of **jump**s. Both actions result in one piece moving diagonally a number of spaces forward (the direction away from the player). Thus, the game is only played on the dark-colored squares.

    a. A **move** is defined by moving a piece diagonally forward one space left or right. A **move** is valid only if the ending space is unoccupied. Here is an example of a valid **move**:



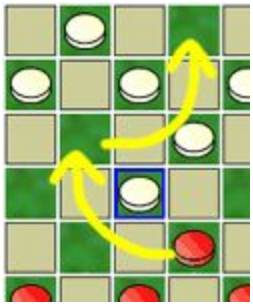
b. A **jump** is defined by moving a piece diagonally forward two spaces in the same direction (diagonally left or diagonally right). A **jump** is valid only if the adjacent space is occupied by a piece owned by the other player, and if the ending space is unoccupied. Here is an example of a valid jump:

Here is an example of an *invalid* jump:



After a **jump**, the other player's piece (or pieces) that were in the occupied space(s) are removed from the board.

c. In one turn, a *single* piece may perform multiple **jump**s. For each **jump**, the piece must move two spaces in the same direction, but a series of **jump**s can move in different directions (the first jump is to the left, while the second is to the right); all **jump**s are still required to move forward. Here is an example of performing multiple **jump**s:



d. On your turn, if any of your pieces can **jump**, you must do a **jump** that turn. Furthermore, after you **jump** with a piece, *you must continue to **jump** with that piece until that piece cannot perform a **jump***. In the event that multiple pieces can **jump**, you may choose which piece to move, and only that piece will move this turn. If at any point the player has multiple **jump** options, they may choose which **jump** to execute, even if one of the options involves fewer **jump**s than the other option.

Photos courtesy of http://www.itsyourturn.com/t_helptopic2030.html