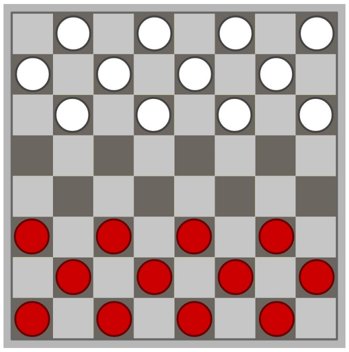
Checkers

Overview

- Implement this programming challenge in the programming language of your choice.
- Please tell us how much time was spent on the implementation.
- The objective of the exercise is to examine correctness, completeness, and readability of the implementation.
- Please provide a runnable/executable version beyond the source code (e.g. with a C implementation, provide a compiled executable).
- The program should be runnable from the command line and should take a single argument, which is the path (relative or absolute) to a `.txt` file containing the game moves.
- The program should write the output to the standard output.
- Four test cases are provided. The expected output for each of these test cases is provided in `expected output.txt`.
- Your solution will be tested against a large set of test cases.

Game Rules



- The game is played on a standard 8x8 board, as depicted above.
- White moves first.
- A piece can only move diagonally forward (towards the opposite side of the board).
- The program must print the winner's identity: `white`, `red`, or `tie`.
- If the provided moves sequence describes an illegal move, the program must print the relevant information in the following format: `line <line number> illegal move: <move>`. See the expected output file for an example.
- If the game did not end, the program must print `incomplete game`.
- Jumping the opponent's pieces is mandatory. If, on a player's turn, one of his or her pieces can jump over one of the other player's pieces, the only legal move at that state is to take the jump.
  - Jumping pieces is only done in the forward direction.
  - The jumped piece must be removed from the board.
  - If follow-up jumps are available, the player must keep jumping until all the follow-up jumps are completed.
  - If, at any state, more than one jump is available, the player may take any of the available jumps.

- To simplify, there are no Kings (or Queens for that matter). When a piece reaches the other side of the board, that is as far as it can go.
- In the game files, a move is given on each line as: `x0,y0,x1,y1` such that the initial position of the piece is at column `x0` column and row `y0`, and `x1,y1` is the destination position.
- `0,0` is the top left square. See above image for the initial board state.
- Obligatory multiple jumps are provided on multiple separate lines.
- The game ends when, at a player's turn, there are no legal moves.
- The winner is the player with the most remaining pieces on the board when the game ends.