Checkers Programming Exercise

Please implement a simplified checkers game.

Implementation notes

- Please implement the exercise with a programming language that you are comfortable with, and tell us how much time you spent on the exercise. The purpose of this exercise is to test correctness, readability and effectiveness of implementation, taking into account the time spent.
- Please attach with the code an executable version (for example, compiled C code).
- The program should accept a text file that describes the sequence of checkers moves, and will print the winner: "first", "second" or "tie".
- If the files includes an illegal move, or if the game was not completed, the program should report that.
- This exercise comes with 4 example input files. The expected output appears in the file "expected output.txt".
- Please provide a command-line program, which takes as an argument the input file name, and prints one line of output: "first", "second", "tie" or "line X illegal move: <line>" (See the file expected_output.txt for examples).

Simplified checkers rules

- The board size is 8x8 squares.
- Pieces can only move diagonally forward.
- The white player is the first to play.
- If there is a possible capture move, a capture must be made. This includes multiple-capture sequences. Captures are forward only.
- Nothing special happens to a piece that reaches the final row.
- (0, 0) is the bottom-right white square. The initial state is shown in the picture below.
- In the input text file, a move is described like this: x0, y0, x1, y1.
  x0 is the source column, y0 is the source row. x1 and y1 are the target. A multiple capture appears as multiple lines.
- The game ends when there are no possible moves for the player that should play now.
- The winner is the player who has more pieces left at the end of the game.