

CS112-B-Assignment 2

Deadline 23rd February 2023

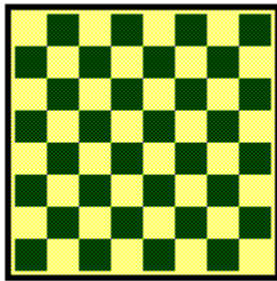
Hint: Start working early

Group size: Max. two students

Problem Statement:

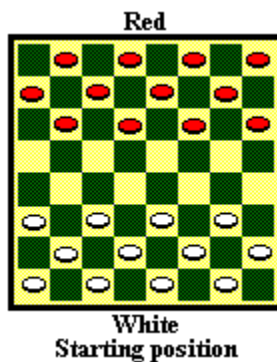
In this assignment you have to develop the game of checkers.

Checkers is a board game played between two players, who alternate moves. The player who cannot move, because he has no pieces, or because all of his pieces are blocked, loses the game. Players can resign or agree to draws.




The board is square, with sixty-four smaller squares, arranged in an 8x8 grid. The smaller squares are alternately light and dark colored (green and buff in tournaments), in the famous "checker-board" pattern. The game of checkers is played on the **dark** (black or green) squares. Each player has a dark square on his far left and a light square on his far right. The double-corner is the distinctive pair of dark squares in the near right corner.

The pieces are Red and White, and are called Black and White in most books. In some modern publications, they are called Red and White. Sets bought in stores may be other colors. Black and Red pieces are still called Black (or Red) and White, so that you can read the books. The pieces are of cylindrical shape, much wider than they are tall (see diagram). Tournament pieces are smooth, and have no designs (crowns or concentric circles) on them. The pieces are placed on the **dark** squares of the board.



The starting position is with each player having twelve pieces, on the twelve dark squares closest to his edge of the board. Notice that in checker diagrams, the pieces are usually placed on the light colored squares, for readability. On a real board they are on the dark squares.

Moving: A piece which is not a king  can move one square, diagonally, forward, as in the diagram at the right. A king can move one square diagonally, forward or backward. A piece (piece or king) can only move to a vacant square. A move can also consist of one or more jumps (next paragraph).



Jumping: You capture an opponent's piece (piece or king) by jumping over it, diagonally, to the adjacent vacant square beyond it. The three squares must be lined up (diagonally adjacent) as in the diagram at the left: your jumping piece (piece or king), opponent's piece (piece or king), empty square. A king can jump diagonally, forward or backward. A piece which is not a king, can only jump diagonally forward. You can make a multiple jump (see the diagram on the right), with one piece only, by



jumping to empty square to empty square. In a multiple jump, the jumping piece or king can change directions, jumping first in one direction and then in another direction. You can only jump one piece with any given jump, but you can jump several pieces with a move of several jumps. You remove the jumped pieces from the board. You cannot jump your own piece. You cannot jump the same piece twice, in the same move. If you can jump, you must. And, a multiple jump must be completed; you cannot stop part way through a multiple jump. If you have a choice of jumps, you can choose among them, regardless of whether some of them are multiple, or not. A piece, whether it is a king or not, can jump a king.

Kinging: When a piece reaches the last row (the King Row), it becomes a King. A second checker is placed on top of that one, by the opponent. A piece that has just kinged, cannot continue jumping pieces, until the next move.

Red moves first. The players take turns moving. You can make only one move per turn. You must move. If you cannot move, you lose. Players normally choose colors at random, and then alternate colors in subsequent games

Constraints:

- Must use classes
- Each piece on the board must be an instance of a Class called Piece
- The class Piece may contain attributes
 - Player
 - LocationX
 - LocationY
 - IsKing
 -
- The class Piece may contain functions.
 - TakeMove
 - ValidateMove
 -(any additional function as you may need)
- Display the game board as a 2D array.
- Check all the valid and invalid moves

Assignment Submission

Name your file as <yourRegNo1_ yourRegNo2_HW1>.cpp. For example, 2022121_2022111_HW3.cpp