

## Guidelines and Grading Policy

A program that does not compile will receive a **zero**. For every exercise, make sure you upload your solutions to your GitHub repository and maintain proper version control practices (commit your work regularly and meaningfully). You are also required to include the specifications of each method, in addition to the testing strategy and test-cases used to validate your program's correctness. You are also expected to submit a report explaining your strategy, and to present your project orally. The report, presentation, GitHub repository, specifications, and testing skills will account for 50% of the grade. The remaining 50% is for the program's correctness. Not every program that runs will be considered correct. In phase two, your program should perform better than random. It should perform the winning move if such a move exists, and a blocking move to block the opponent from winning if such a move exists.

## Connect Four - Game Description

Connect Four is a two-player board game in which the players take turns in dropping tokens into a vertical grid with the objective of getting four of one's own tokens into a line. The game was published by Milton Bradley in 1974; a non-proprietary version is known as "The Captain's Mistress". See [http://en.wikipedia.org/wiki/Connect\\_Four](http://en.wikipedia.org/wiki/Connect_Four) for more information. In the game, there are two colors of tokens. One player plays yellow, the other red. The following depicts a game configurations that may occur in a concrete game:

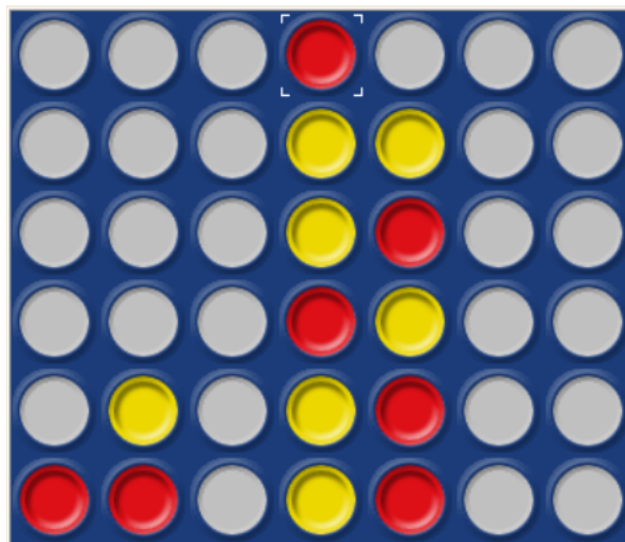


Figure 1: A Connect Four game board.

If we enumerate the columns with 0..6, we see that if yellow plays 2 and red plays 2, red will have won, because red has managed to get four pieces into a line. This line is diagonal, but it could also be horizontal or vertical.



## Rules

1. The game board has seven columns and six rows.
2. There are 21 red and 21 yellow tokens.
3. One player plays with red tokens, the other with yellow tokens.
4. The tokens are inserted at the top of a column, and they will fall down and land on the ground (if the column was empty) or on top of a previously inserted token.
5. Red starts.
6. Red and yellow take turns.
7. One can only insert tokens in one of the seven columns.
8. One cannot insert a token into a column that is full.
9. A line consists of several tokens, either in vertical, horizontal, or diagonal form, which contain only tokens of the same color.
10. A player wins if they manage to form a line of four tokens of their color.
11. The game ends if one of the players wins.
12. There will be no ties. In case of a tie on the board, the player that took less over-all time wins.

## Implementation

Write a C program that implements one player taking part in a Connect Four game. We will provide a driver to simulate the game between two players. Your player can expect to be called with an input representing the current state of the game board. The board will be represented as a two-dimensional array of integers. The integer values correspond to the following:

- 0 - The cell is empty.
- 1 - The cell has a red token.
- 2 - The cell has a yellow token.

The player should examine the initial board state to determine whether they will be playing red or yellow. The player returns the board with one extra token inserted with their color value.

## Phase One - Due October 7

Implement a two-player version of the Connect Four game. Write a program that:

1. Starts by asking the two players for their name (single name, no spaces).
2. Proceeds by displaying the initial game board state (Figure 2).



Figure 2: Initial gameboard configuration as displayed by the program. The program should then print an updated game board with the new inserted tokens after every successful move executed.

3. Tosses a fair coin to determine which player will start, and announces the result.
4. Takes turns asking the players for their move. A move is input as a single number representing the column at which the player would like to insert their current token.
5. If the move is not allowed, the program should inform the player along with the corresponding reason the move is not allowed, then ask the same player for another move (until the player enters a legal move).
6. The program should print the updated game board after every move.
7. If a player wins, the game stops and the program announces the winner.
8. In case of a tie on the game board, the program announces the player that took less time selecting legal moves (including time wasted selecting illegal ones) as the winner (and announces that they won by speed).

## Phase Two - Due November 18

Implement a bot that plays the Connect Four game. Write a program that runs similarly to Phase 1 but where one of the players is the bot you implemented. In other words, the program asks for a single player's name, then proceeds similarly to phase one and accepts moves submitted by the player and others generated by the bot. The bot accepts the current game board's state as a two-dimensional integer array and should return the move it took as an integer representing the column at which it inserted its token. You may use any strategy to implement your bot, as long as you can clearly explain it in your oral presentation.



Faculty of Arts & Sciences  
Department of Computer Science  
CMPS 270 – Software Construction  
Fall 2022 – Course Project

## **Bonus (Extra 10% of the project's grade)**

Modify your program to offer different difficulty levels (Easy, Medium, Hard) that launch different bots with clear differences in terms of the strategies used. All levels should perform better than random.

## **Battle Royale**

A knockout tournament will be held at the end of the semester. The top 3 groups will receive 5%, 3%, and 2% bonus points respectively. The bonus points will be added to the final grade.

**In addition to your GitHub submission, add your programs to one folder: `group-name.connect4.zip` (or `.rar`) and submit it to moodle.**