Faculty of Arts & Sciences
Department of Computer Science
CMPS 241 – Systems Programming
Fall 2025/26 – Course Project

## 1 Guidelines and Grading Policy

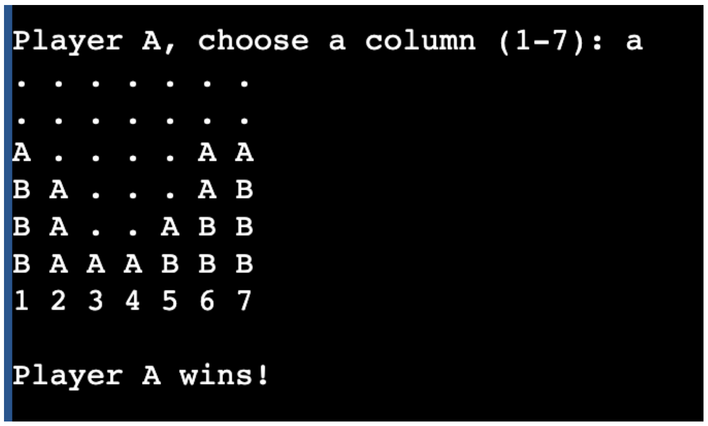
A program that does not compile will receive a zero. Make sure to upload your solution to your GitHub repository and maintain proper version control practices (commit your work regularly and meaningfully). You are required to include testing strategy with test cases validating your program's correctness. You are also required to debug your code using GDB and Valgrind. A report explaining your strategy is required, and you'll also need to present your project orally. The report, presentation, GitHub repository, specifications, and testing strategy account for 50% of the grade, with the remaining 50% for program correctness.

This project will be divided into four sprints. Each sprint has specific requirements and deliverables.

## 2 Connect 4 - Game Description

Connect 4 is a strategy game played by children and adults in 1974. It quickly became a top-rated game due to its easy rules and various strategies players may use to win. Connect 4 is an excellent game to work your children's brain cells.

Connect 4 is a turn-based 2-player game played on a 7x6 board. Each player places their checkers from the lowest row until the highest one. The goal is to align four checkers horizontally, vertically, or diagonally.



```
Player A, choose a column (1-7): a
. . . . . . .
. . . . . . .
A . . . . A A
B A . . . A B
B A . . A B B
B A A A B B B
1 2 3 4 5 6 7

Player A wins!
```

You can find the rules and try the game on the following link:

https://papergames.io/en/connect4

## 3 Useful strategies and tactics to win at Connect 4 online

### 3.1 At the start of the game

Try to place a maximum of checkers in the middle column to maximize the possibilities of aligning 4 checkers in the following turns. This tactic prevents your opponent from doing the same while using the column. https://youtu.be/17w0W9M53uc

### 3.2 The double line

A straightforward way of trapping your opponent at Connect 4 is to create a winning double line. The idea is to have two lines that lack one checker to have a line of 4. Try to compel them to make a move that would lead to you being able to align a line of 4 at the next turn. https://youtu.be/G7YHiN39qS8

### 3.3 How to beat a novice

When opening the game try to align three checkers on the lowest line. If your opponent is not paying attention, you can choose between reacting on the left or right. In whichever case, you win the game! https://youtu.be/ZZh6Tb_3fbE

### 3.4 Blocking columns

The idea is to block columns so that whenever your opponent places a checker or pieces of discs on a row or a column, you would win at the next turn. The goal is to lead your opponent into unwanted moves, which enable you the freedom of developing traps for the next steps. https://youtu.be/pht2Bf1WkOA

## 4 Sprint 1: Two-Player Game (Due October 4th)

### 4.1 Description

Implement a two-player version of Connect four (Players A and B), where both players interact through a console-based interface. Players take turns adding checkers and their initials will appear inside it (A or B). The game continues until all checkers are used or when one player connects four checkers horizontally, vertically or diagonally.

### 4.2 Requirements

- Display a 6x7 grid.
- Allow two players to take turns placing their checkers
- Update the board after each move
- Check if a move causes four checkers to be connected
- Declare the winner if there's any
- Run the game on a small Linux distribution without a graphical user interface (GUI), such as TinyCore Linux or Alpine Linux.
- The Linux OS will run inside a Virtual Machine (VM), and your game should launch automatically as the first thing displayed after starting the VM.

**You can find at the end of this file screenshots for sprint 1.**

## 5 Sprint 2: Easy-Level Bot (Due October 18th)

### 5.1 Description

In this sprint, implement a simple bot that plays against a human player. The bot could make random moves, but ensure that it doesn't choose invalid move. The human player should still be able to play with the same rules as Sprint 1.

### 5.2 Requirements

- All previous requirements
- Modify the game to add an option so that the player can play against a bot and choose the difficulty level.
- A simple idea is to program the bot to make random valid moves. However, it's up to you to decide whatever strategy you want the bot to follow.
- Ensure the bot only makes valid moves.
- Display the updated board after every move.
- Declare the winner at the end of the game.

## 6 Sprint 3: Medium-Level Bot (Due November 1st)

### 6.1 Description

In this sprint, you will improve the bot. You can follow any strategy you want. Again, it's up to you to propose whatever strategy you want.

### 6.2 Requirements

- All previous requirements
- Calculate the complexity of the strategy

## 7 Sprint 4: Hard-Level Bot (Due November 15th)

### 7.1 Description

For the final sprint, the bot should use an advanced strategy. You are free to research and choose a strategy. The strategy should significantly improve the bot's performance.

### 7.2 Requirements

- Implement a sophisticated strategy for the hard-level bot.
- Continue calculating the complexity of the advanced strategy.
- Submit the final report and a PowerPoint presentation. During the last week of the semester you will do a presentation and a demo
- All team members are expected to know all the details of the project

## 8 Bonus Points

Up to 5% extra credit will be awarded for implementing multithreading in your project to improve execution speed.

## 9 Tournament: Bot Showdown

A **knockout tournament** between the students' bot implementations will be held at the end of the semester. Participation is optional but highly encouraged. Bonus points will be awarded to the top 2 groups, which will be added to their final course grade.

### 9.1 Bonus Points

- 1st place: +3% to final grade
- 2nd place: +2% to final grade
- 3rd place: +1% to final grade

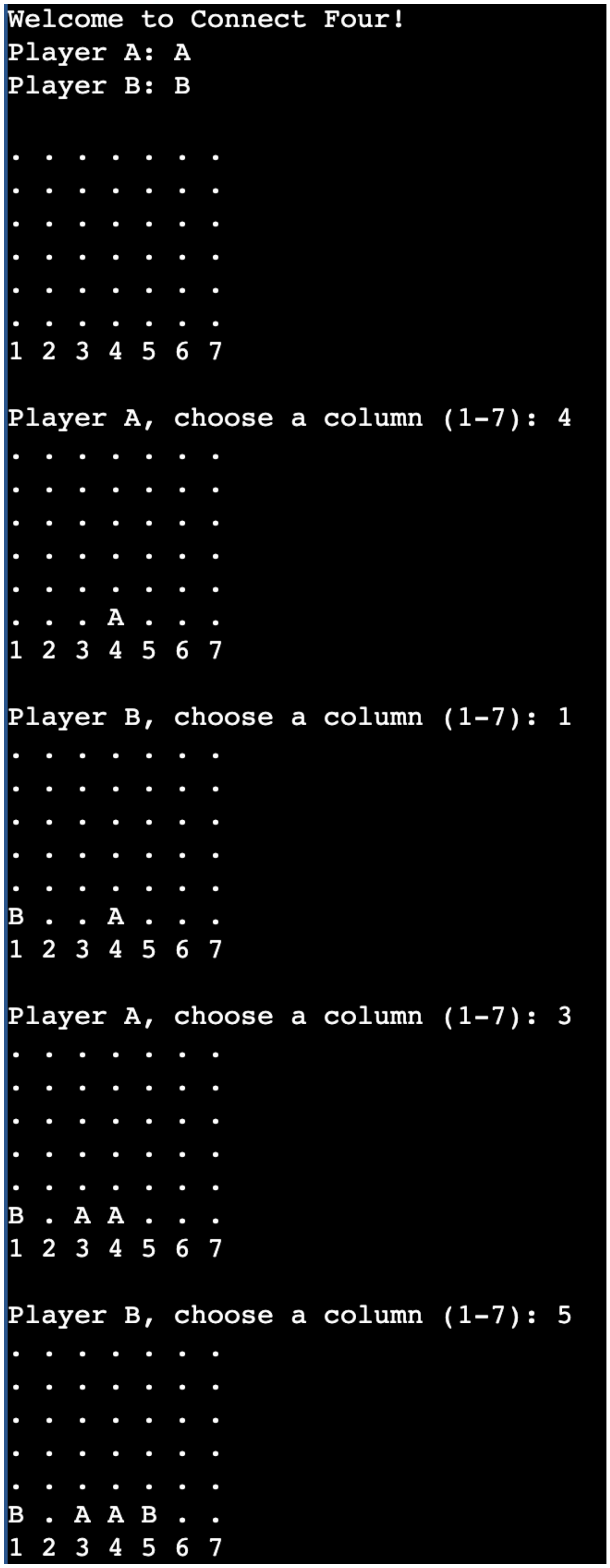
### 9.2 Tournament Rules

1. **Pairing**: Players (or groups) will be randomly matched to compete.
2. **Games**: Each pair will play two games, allowing both players to be the starting player once.
3. **Winner Determination**:
    - If one player wins both games, they proceed to the next round.
    - If each player wins one game, the tie will be broken based on the number of moves.

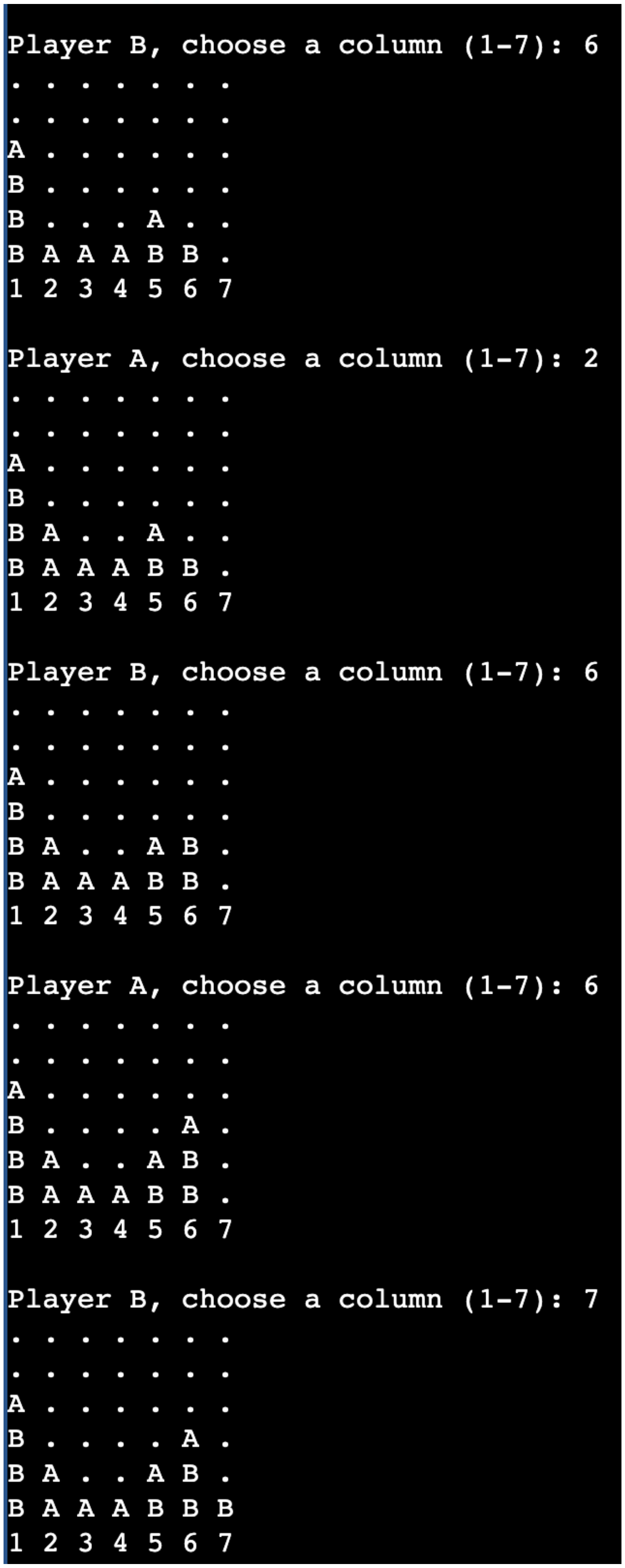
This tournament is a fun way to apply and showcase the strategies developed during the project. It's a great opportunity to see how your bot stacks up against others and gain recognition for a well-implemented algorithm!

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

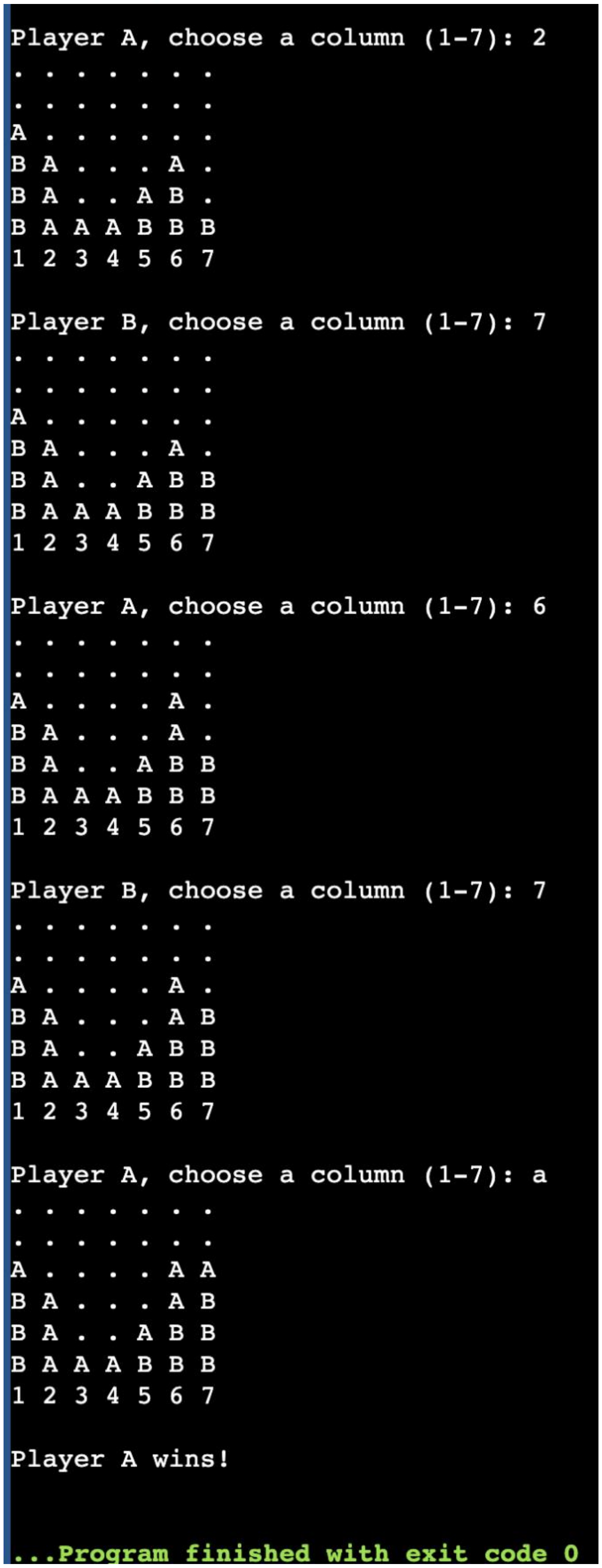
**Below are some screenshots for the first sprint:**



```
Welcome to Connect Four!
Player A: A
Player B: B

. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
1 2 3 4 5 6 7

Player A, choose a column (1-7): 4
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . A . . .
1 2 3 4 5 6 7

Player B, choose a column (1-7): 1
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
B . . A . . .
1 2 3 4 5 6 7

Player A, choose a column (1-7): 3
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
B . A A . . .
1 2 3 4 5 6 7

Player B, choose a column (1-7): 5
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
B . A A B . .
1 2 3 4 5 6 7
```



```
Player B, choose a column (1-7): 6
. . . . . . .
. . . . . . .
A . . . . . .
B . . . . . .
B . . . A . .
B A A A B B .
1 2 3 4 5 6 7

Player A, choose a column (1-7): 2
. . . . . . .
. . . . . . .
A . . . . . .
B . . . . . .
B A . . A . .
B A A A B B .
1 2 3 4 5 6 7

Player B, choose a column (1-7): 6
. . . . . . .
. . . . . . .
A . . . . . .
B . . . . . .
B A . . A B .
B A A A B B .
1 2 3 4 5 6 7

Player A, choose a column (1-7): 6
. . . . . . .
. . . . . . .
A . . . . . .
B . . . . A .
B A . . A B .
B A A A B B .
1 2 3 4 5 6 7

Player B, choose a column (1-7): 7
. . . . . . .
. . . . . . .
A . . . . . .
B . . . . A .
B A . . A B .
B A A A B B B
1 2 3 4 5 6 7
```



```
Player A, choose a column (1-7): 2
. . . . . . .
. . . . . . .
A . . . . . .
B A . . . A .
B A . . A B .
B A A A B B B
1 2 3 4 5 6 7

Player B, choose a column (1-7): 7
. . . . . . .
. . . . . . .
A . . . . . .
B A . . . A .
B A . . A B B
B A A A B B B
1 2 3 4 5 6 7

Player A, choose a column (1-7): 6
. . . . . . .
. . . . . . .
A . . . . A .
B A . . . A .
B A . . A B B
B A A A B B B
1 2 3 4 5 6 7

Player B, choose a column (1-7): 7
. . . . . . .
. . . . . . .
A . . . . A .
B A . . . A B
B A . . A B B
B A A A B B B
1 2 3 4 5 6 7

Player A, choose a column (1-7): a
. . . . . . .
. . . . . . .
A . . . . A A
B A . . . A B
B A . . A B B
B A A A B B B
1 2 3 4 5 6 7

Player A wins!


...Program finished with exit code 0
```