Connect Four

# Project Profile

The goal of this project is to implement the game [Connect Four](https://en.wikipedia.org/wiki/Connect_Four) using the C programming language. The game will be interactive and played between a human user and an AI (artificial intelligence). The focus of this project is primarily on exercising an introductory understanding of the C programming language including basic data types, looping and conditional constructs, arrays, iteration, basic I/O, formatted output, and functions.

## Connect Four

Traditionally, the connect four game is a two-player connection game in which the players first choose a color and then take turns dropping one colored disc from the top into a seven-column, six-row vertically suspended grid. The pieces fall straight down, occupying the lowest available space within the column. The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of four of one's own discs. Please see the [Wikipedia entry](https://en.wikipedia.org/wiki/Connect_Four) for more details. In your implementation you will use X and O characters in place of the colors.

## Game Board

The game board is a 7-column by 6-row suspended grid. The rows are labeled 1 - 6 and the columns are labeled A - G. This is an example of an empty game board:

A B C D E F G

6 . . . . . . .

5 . . . . . . .

4 . . . . . . .

3 . . . . . . .

2 . . . . . . .

1 . . . . . . .

The rows are labeled from the bottom to top starting with 1, and the columns are labeled from left to right starting with A. Here is an example state of the game where play begins with the human player using the X discs (X always plays first):

A B C D E F G

6 . . . . . . .

5 . . . . . . .

4 . . O . . . .

3 . . O X . . .

2 . O X O X . .

1 . O X X O X .

The record of moves can easily be recorded as [D,E,C,B,F,B,C,C,E,D,D,C] where the first entry is an X move and then alternates between O and X on each subsequent entry.

## Game Play

Your implementation will prompt the user before game play to determine who goes first and then proceed to alternate between the human player and the AI. When it is the human’s turn the game will prompt the player for which column she would like to drop in a disc. For example, using the above game state, if the human player chose column B the resulting game state would be:

A B C D E F G

6 . . . . . . .

5 . . . . . . .

4 . . O . . . .

3 . **X** O X . . .

2 . O X O X . .

1 . O X X O X .

The next move would be decided by the AI player, followed by prompting the human player again. Game play would then proceed in that fashion. Here is an example of the game playing out to completion:

|  |  |  |  |
| --- | --- | --- | --- |
| A B C D E F G  6 . . . . . . .  5 . . . . . . .  4 . . O . . . .  3 . X O X . . .  2 . O X O X . .  1 . O X X O X . | A B C D E F G  6 . . . . . . .  5 . . . . . . .  4 . . O . . . .  3 . X O X . . .  2 . O X O X . .  1 **O** O X X O X . | A B C D E F G  6 . . . . . . .  5 . . . . . . .  4 . . O . . . .  3 . X O X . . .  2 . O X O X . .  1 O O X X O X **X** | A B C D E F G  6 . . . . . . .  5 . . . . . . .  4 . . O O . . .  3 . X O X . . .  2 . O X O X . .  1 O O X X O X X |

In this example, the AI player got *connect four* in a diagonal with discs 1A, 2B, 3C, 4D and won the game.

# Requirements

Your primary objective is to use the C programming language to design and implement the connect four game that operates according to the game play described in the previous section. You are required to design and implement the appropriate data structures and corresponding algorithms that will enable a human player to play against an AI player. The AI design is entirely up to you, however, it should be at least as smart as a 5 year old playing the game. That is, the AI should make some obvious choices such as blocking as a defensive move or trying to build a connection (horizontal, vertical, or diagonal) as an offensive move. If you have taken a course in Artificial Intelligence you are welcome to make your AI more sophisticated (e.g., [Minimax](https://en.wikipedia.org/wiki/Minimax)), however, this is not required. Furthermore, you are constrained to using only the C constructs that we have covered up. In particular:

|  |  |
| --- | --- |
| Allowable C Language Constructs | |
| * printf * basic C data types and variables (int, float, double, char, \_Byte/byte) * storage sizes and ranges * type specifiers * arithmetic expressions * for, while, do loop, if statement, switch, conditional operator * aligning output * scanf | * 1D array and initialization * const * multi-dim arrays * variable length arrays * array length bounds, iteration, length * functions, arguments, locals, returns * prototype declaration * functions and arrays, mutability * global variables * automatic/static variables |

In addition, your implementation must meet the following specific requirements:

1. An array must be used to represent the game board as described above.
2. One or more functions must be used as part of the implementation.
3. Arrays must be passed to functions as arguments.
4. Iteration must be used to traverse the game board.
5. Your implementation must be able to determine end of game: win, lose, or draw.
6. An AI must exist in your game as described above.
7. You should minimize the use of global variables - no global variables is the best.
8. Your implementation should allow either the human or AI to go first.
9. Your implementation must show the state of the game, modeled after our examples above, after each move (both human and AI) and display the final game state, who won or who drew, along with the recorded moves in the form mentioned in the previous section. All output should be properly formatted and aligned.

# Deliverables:

You must submit two files:

* **cf.c** - this is your implementation of the connect four game.
* **README.txt** - this is a text file containing a brief 1 paragraph overview of your work highlighting the important parts of your implementation. The goal should make it easy and obvious for a person to find the important rubric items.