Assignment 3
Implementation of Connect 4 game

Connect Four (or Four-in-a-line) is a two-player strategy game played on a 7-column by 6-row board. Each player has a color and drops successively a piece of his color in one column; the piece falls down to the lowest empty cell of the column. The first player to make an alignment of four pieces of his color, whether horizontally, vertically, diagonally, or anti-diagonally, wins. If the board is filled without any player achieving an alignment, it's a draw game.

Part 1: Game formulation

Propose a formulation of the Connect Four game.

1. To model a state, create a class called "ConnectFourBoard" where you will define the following attributes and functions:

- *board:* A matrix representing the game board. Initially, all the cells of the matrix are set to zero. Then, each time a player takes a turn, one cell is set to the player's piece (1 or 2).
- *drawBoard():* Displays the game board.
- *getPossibleMoves():* Returns the possible moves (i.e., the possible positions for the piece).
- *makeMove(row, col, piece):* The function to drop the piece in the desired position (row, col).
- *win(piece):* Checks if the player with the piece '*piece*' has won the game.
- *gameOver():* Checks if one of the players has won or if it's a draw with no possible moves remaining.
- *heuristicEval(piece):* The heuristic function to evaluate a state.

2. The class "Play" where you will define:

- The function *humanTurn()* allowing the user to take their turn.
- The function *computerTurn()* allowing the computer to take its turn. The choice of the move to be played by the computer is made using the **MinimaxAlphaBetaPruning** algorithm.
- The *MinimaxAlphaBetaPruning()* search function.

Part 2: Connect Four Web Application

Develop a web application for the Connect Four game, featuring two distinct settings: Human vs. AI bot and AI bot 1 vs. AI bot 2. The implementation of the AI component will be in Python, while the user interface can be implemented using JavaScript.

1. Human vs. AI bot:
In this setting, the AI bot is programmed to employ the minimax alpha beta pruning algorithm, enhanced with a heuristic, to effectively navigate and solve the game.

2. AI bot 1 vs. AI bot 2:
For this setting, both AI bots have the option to utilize the minimax alpha beta pruning algorithm with distinct heuristics. Alternatively, one bot may employ the minimax alpha beta pruning algorithm, while the other utilizes a different adversarial search algorithm, such as Monte Carlo.