```
%%%%CONNECT FOUR%%%%
%%%%TEAM A SDP%%%%
clc
clear

%Display message that explains the rule for the user
fprintf('::::: Welcome to Connect Four :::::\n\n')
fprintf('This is a two player game where the objective is to be the
 first person to connect four markers in a row.\nEach player will
 select a column to drop their marker in, the game will switch players
 untill there is a winner or a tie.\n\n')

gameStart = input('Are you ready to start? Enter Y to begin: ','s');

%Lets the user read at their own place and proceed when ready
while (gameStart ~= 'Y')
    gameStart = input('Are you ready to start? Enter Y to begin:
 ','s')
end

%Constants and Variable declaration
ROW = 6;
COLUMN = 7;
board = zeros(ROW,COLUMN)
player = 1;
gameOver = false;

%while loop that iterates untill the game is over
while (gameOver == false)

    %checks to see if there is a tie
    gameDraw = c4_drawGame(board, ROW, COLUMN);
    if (gameDraw == true)
        fprintf('::::: DRAW :::::\n')
        break
    end

    %asks user for input then validates it
    fprintf('::::: Player %i :::::\n Enter a column number: ',
 player);
    playerMove = input("");
    playerMove = c4_inputChecker(playerMove);


    %checks to see if the users input is in a valid column (ie, column
 isnt
    %already filled) , else, reprompts user, updates board with users
    %selection
    if (c4_columnValid(board, playerMove) == true)
        row = c4_rowFinder(board, playerMove, ROW, COLUMN);
        board = c4_updateBoard(board, row, playerMove, player)
    else
```

```
        while(c4_columnValid(board, playerMove) == false)
            fprintf('Invalid input. Please re-enter column number: ')
            playerMove = input("");
        end
        row = c4_rowFinder(board, playerMove, ROW, COLUMN)
        board = c4_updateBoard(board, row, playerMove, player)
    end


    %checks if anyone has won yet
    gameOver = c4_winnerWinner(board, player, ROW, COLUMN);
    if(gameOver == true)
        fprintf('::::: GAME OVER :::::\nPlayer %i wins!\n',player)
    end

    %%switches the player
    if (player == 1)
        player = 2;
    else
        player = 1;
    end
end
```

```
::::: Welcome to Connect Four :::::

This is a two player game where the objective is to be the first
 person to connect four markers in a row.
Each player will select a column to drop their marker in, the game
 will switch players untill there is a winner or a tie.


Error using input
Cannot call INPUT from EVALC.

Error in c4_main (line 10)
gameStart = input('Are you ready to start? Enter Y to begin: ','s');
```

Published with MATLAB® R2019b

```
function draw = c4_drawGame(board, ROW, COLUMN)

%c4_drawGame checks to see if the game ended as a tie, b/c connect 4
%alternates players and the values stored are either 0,1,2, a
 completly
%filled ""draw"" board is equal to 63 and thus checks if this number
 is
%ever reached, returns a boolean value

    sum = 0;
    for (i = 1:COLUMN)
        for (k = 1:ROW)
            sum = sum + board(k, i);
        end
    end

    if (sum == 63)
        draw = true;
    else
        draw = false;
    end

end
```

```
Not enough input arguments.

Error in c4_drawGame (line 9)
    for (i = 1:COLUMN)
```

Published with MATLAB® R2019b

```
function columnNum = c4_inputChecker(columnNum)

%c4_inputChecker validates users input to make sure it is within the
 range
%of 1-7 and is an integer, returns the columnNum after reprompting the
 user
%to input a valid number

    while ((columnNum < 1) || (columnNum > 7) || (columnNum ~=
 floor(columnNum)))
        columnNum = input('Invalid input. Please re-enter your column
 number: ');
    end
end
```

```
Not enough input arguments.

Error in c4_inputChecker (line 7)
    while ((columnNum < 1) || (columnNum > 7) || (columnNum ~=
 floor(columnNum)))
```

Published with MATLAB® R2019b

```
function  valid = c4_columnValid(board, playerMove)

%c4_columnValid checks to see if the users inputted column number isnt
%completely full, returns a boolean value

    if(board(1, playerMove) == 0)
        valid = true;
    else
        valid = false;
    end
end
```

```
Not enough input arguments.

Error in c4_columnValid (line 6)
    if(board(1, playerMove) == 0)
```

Published with MATLAB® R2019b

```
function row = c4_rowFinder(board, playerMove, ROW, COLUMN)

%c4_rowFinder finds the next available row in the user specified
 column
%starting from the bottom down, returns row number

    for (i = 1:ROW)
        if (board(COLUMN - i, playerMove) == 0)
            row = COLUMN - i;
            break;
        end
    end
end
```

```
Not enough input arguments.

Error in c4_rowFinder (line 6)
    for (i = 1:ROW)
```

Published with MATLAB® R2019b

```
function board = c4_updateBoard(board, ROW, playerMove, player)

%c4_updateBoard updates board with the now valid user selection and
%displays their playerID in the matrix, returns and displays a matrix
 of
%values acting as the gameboard

    board(ROW, playerMove) = player;
end
```

```
Not enough input arguments.

Error in c4_updateBoard (line 7)
    board(ROW, playerMove) = player;
```

Published with MATLAB® R2019b

```
function gameOver = c4_winnerWinner(board, player, ROW, COLUMN)

%c4_gameOver checks to see if there are any connects fours by
%systematically checking the board horizontally, vertically, and
%diaganolly, returns a boolean value

    gameOver = false;

%HORIZONTAL WIN CONDITIONS
    for (i = 1:(COLUMN-3))
        for (k = 1:ROW)
            if ((board(k,i) == player) && (board(k, i+1) == player) &&
 (board(k, i+2) == player) && (board(k,i+3) == player))
                gameOver = true;
                break;
            end
        end
    end

%VERTICAL WIN CONDITIONS
    for (i = 1:COLUMN)
        for (k = 1:(ROW-3))
            if ((board(k,i) == player) && (board(k+1, i) == player) &&
 (board(k+2, i) == player) && (board(k+3, i) == player))
                gameOver = true;
                break;
            end
        end
    end

%DIAGANOL WIN CONDITIONS
    for (i = 1:ROW-3)
        for (k = 4:COLUMN)
            if ((board(i,k) == player) && (board(i+1, k-1) == player)
 && (board(i+2, k-2) == player) && (board(i+3, k-3) == player))
                gameOver = true;
                break;
            end
        end
    end
%DIANGONAL WIN CONDITIONS (cont.)
    for (i = 1:ROW-3)
        for (k = 1:COLUMN-3)
            if ((board(i,k) == player) && (board(i+1, k+1) == player)
 && (board(i+2, k+2) == player) && (board(i+3, k+3) == player))
                gameOver = true;
                break;
            end
        end
    end
end

Not enough input arguments.
```

```
Error in c4_winnerWinner (line 10)
    for (i = 1:(COLUMN-3))
```

Published with MATLAB® R2019b