CIS 343 – Structure of Programming Languages
Winter 2026, Programming Assignment #1

Othello Game in C Language
Due Date: Monday, February 16, 2026

Project Objectives

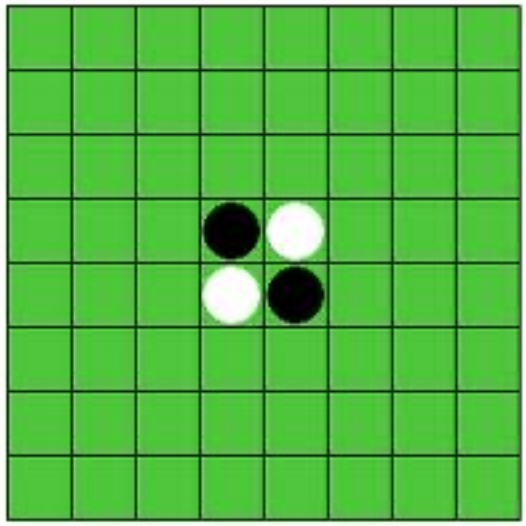
- Input and output
- Command-line arguments
- Two-dimensional arrays
- Functions
- Unit testing

Problem Specification

Write a C program to play a game of Othello (also called Reversi). The objective and the rules of Ohello game can be found on Wikipedia here: http://en.wikipedia.org/wiki/Reversi:

You can play this game online here: http://www.mathsisfun.com/games/reversi.html

The program starts the game with the following board configuration:



- Players take alternate turns.
- A player is only allowed to make valid moves. A valid move is one where at least one disc of the opponent is reversed.
- If a player cannot make a valid move, then that player loses his/her turn.
- The game ends when neither player can move or when the board is full.
- The player with the most pieces (discs) on the board at the end wins the game.

Project Files

You are provided with the following files. Keep these files in a folder created for this project (say Project1 folder).

- **`othello.h`:** This header file contains constant definitions and function declarations used in the rest of the program. **DO NOT MODIFY THIS FILE**.

- **`othello.c`:** This file contains the implementation of the functions declared in the header file `othello.h`. The function **`toString()`** is already completed for you and do not make any changes to this function. Your task is to implement the rest of the functions in this file. You are free to add other helper functions in this file. DO NOT make any changes to function parameters and return type. Any changes to function signatures will break the code in `driver.c` and `tests.c` files.

- **`driver.c`:** This file contains the main driver code to play the game with command-line interface (CLI). **DO NOT MODIFY THIS FILE.**

- **`CuTest.h`** and **`CuTest.c`**: Files for CuTest tool – a unit testing library for the C language. **DO NOT MODIFY THESE FILES.**

- **`tests.c`**: This file contains unit tests for testing your implementation of the game. **DO NOT MODIFY THIS FILE.**

Compiling and Running the Program on EOS machines

Use the following command to compile the files:

```
$ gcc -Wall -std=c99 -o driver driver.c othello.c
```

Run the program as follows (Player 1 to start the game using black disc):

```
$ ./driver 8 1 B
<<<<< Welcome to the game of Othello >>>>>
Player 1: B   Player 2: W
Player 1 starts the game...

  0 1 2 3 4 5 6 7
0 - - - - - - - -
1 - - - - - - - -
2 - - - - - - - -
3 - - - B W - - -
4 - - - W B - - -
5 - - - - - - - -
6 - - - - - - - -
7 - - - - - - - -
```

```
Turn> Player 1 (B) - Enter location to place your disc (row col): 3 5

  0 1 2 3 4 5 6 7
0 - - - - - - - -
1 - - - - - - - -
2 - - - - - - - -
3 - - - B B B - -
4 - - - W B - - -
5 - - - - - - - -
6 - - - - - - - -
7 - - - - - - - -

The game continues...
```

Compiling and Running the Unit Tests on EOS machines

Use the following command to compile the files:

```
$ gcc -Wall -std=c99 -o tests tests.c CuTest.c othello.c
```

Run the unit tests with the following command:

```
$ ./tests
```

Project Deliverables (VERY IMPORTANT)

1. Make sure you name(s) appear in the comment section at the top in the **`othello.c`** file.

2. Upload only **`othello.c`** file on Blackboard by midnight on due date.

3. DO NOT upload the remaining files. If you upload these files, they will be discarded.

4. Your project will be graded based on how many unit tests in **`tests.c`** file pass. The driver code (`driver.c`) is provided so you can debug, run/play, and test your implementation of the game.

5. The submission time on Blackboard will be used as the official submission date/time.

6. Due to possible portability issues with compiling and running C programs across different platforms, make sure your program compiles and runs on EOS machines before submitting **`othello.c`** file on Blackboard. I will compile your program and run unit tests on your program on EOS machines. If your program does not run on EOS machines, it will not be graded and points will be assigned accordingly.

7. **Late penalty (10% per day) applies after Monday, February 16th.**