CIS 343 – Structure of Programming Languages
Winter 2026, Programming Assignment #2

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

Purpose

In C programming, context is of the utmost importance. For instance, sometimes we may be able to view a char** as a multi-dimensional array, and other times we may only be able to view it as a pointer to a pointer. Context affects how we can then use this data structure. This assignment is intended to help you begin to understand the effects of context in C programming, particularly as it relates to the use of pointers.

History

Conway's Game of Life (https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life) is a cellular automaton created by John Conway, a British mathematician. The "game" is really not a game in the classical sense - there are no win or loss conditions. Instead, the "player" sets the beginning state of cells before allowing the game to evolve to new states through subsequent generations. The game is played on a grid. Each cell of the grid has two states, either off or on, corresponding to dead or alive. The rules are:

- A live cell with less than two live neighbors dies
- A live cell with two or three live neighbors lives
- A live cell with more than three neighbors dies
- A dead cell with three live neighbors becomes live

A "generation" consists of these rules being applied to every cell of the current board state, to create a new board.

Assignment

Your job is to create the Game of Life in C. The initial state of the game will be read in from a text file. The file must adhere to the following convention:

- First number is the length of the grid
- Second number is the width of the grid
- The remaining values are only 0s or 1s and correspond to a dead or live cell. There are `length x width` number of values after the first two values.

The game can be played from command line using the command below. The file name (containing the initial state of the grid) must be passed to the program as a command-line parameter.

The driver program (`driver.c`) is already implemented for you. You can run the program with this command:

```
$ ./driver datafile
```

Once the program has started, the board from the data file will be read. If that is not possible the program should print a message and exit. If the file is valid, the current state of the board will be displayed followed by a menu with the following options:

- Iterate once – will automatically iterate one generation.
- Iterate multiple – will ask the user how many generations and iterate each of those, displaying each iteration.
- Write current state – will ask for a file name and write the current board state to a file. Errors will be reported. If the file cannot be written the program will recover and continue.
- Quit – will gracefully exit, freeing any memory allocated.

Project Files

You are provided with the following files. Keep these files in a folder designated for this project.

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

- **`life.c`**: This file contains the implementation of the functions declared in the header file `life.h`. The function **`toString()`** is already completed for you and do not make any changes to this function. Your job 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 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.**

- **`beacon.gol`**, **`blinker.gol`**, **`glider.gol`**, **`toad.gol`**, and **`tub.gol`**: data files that can be used to play/test the game. **DO NOT MODIFY THESE FILES**.

Compiling and Running the Program on EOS machines

Use the following command to compile the files:

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

Run the driver program as follows providing the data file as command-line argument. Here is an example:

```
$ ./driver blinker.gol
<<<<< Welcome to the Game of Life >>>>>
```

```
 .  .  .  .  .
 .  .  X  .  .
 .  .  X  .  .
 .  .  X  .  .
 .  .  .  .  .

Press n (or return) for next generation, i to iterate, l to load grid, s to
save grid, or q to quit >>

 .  .  .  .  .
 .  .  .  .  .
 .  X  X  X  .
 .  .  .  .  .
 .  .  .  .  .

Press n (or return) for next generation, i to iterate, l to load grid, s to
save grid, or q to quit >> q
```

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 life.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 **`life.c`** file.
2. Upload only **`life.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 only provided so you can debug, 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 any files 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**.