CS 232 Introduction to C and Unix

Extra Credits (Refactoring HW3 Program)

(Due on March 24, 2022, 11:59pm)

The goal of this extra-credit assignment is to test your ability to use pointers and memory management in C.

Collaboration policy:

Similar to homework 3, you are to work alone. If you get stuck on part of the assignment, you are encouraged to come by office hours. If you cannot make office hours, just send me an email and we can find a time to meet. I'm here to help you, but I do not know how to help if you don't ask me.

Description (Game of Fifteen):

In homework 3, you have implemented the program for the game of fifteen. In this assignment, you will change the implementation and make the code better:

- The original implement is using “board” and “d” as global variables. In this assignment, you need to remove these global variables and define them in the main function.
- Moreover, instead of using the array for board, you will define “int ** board” and dynamically allocate the memory (in heap by malloc) to board based on the user provided value for d from the command-line argument. Remember to free the allocated memory before exiting the program, and use valgrind to check it.
- You will need to change the function definitions for some functions since now “board” and “d” are not global variables. They needs to be passed through function input arguments. But you should be able to keep the original function implementation from homework 3.

Under the homework directory, please create a new subdirectory “extra_credit” and copy all files from “hw3” (e.g., Makefile, 3x3.txt, 4x4.txt, and fifteen.c that you have implemented for homework 3) to this directory. Then modify/refactor fifteen.c code.

Compiling and Testing:

Under the Diamond Linux server, to compile the code, please use the Makefile. Specifically, run the command:

$ make

to compile your code “fifteen.c”. Indeed, “make” command looks into the Makefile to run:

gcc -g -std=c11 -Wall -o fifteen fifteen.c -lm

The Makefile is a very convenient way to compile multiple files, which we will discuss later.

To test your new implementation of fifteen, you can certainly try playing it. (Know that you can force your program to quit by hitting ctrl-c.) Be sure that you (and we) cannot crash your program, as by providing bogus tile numbers. Moreover, you can automate execution of this game. In fact, files 3x3.txt and 4x4.txt provide the winning sequences of moves for a 3 × 3 board and a 4 × 4 board, respectively. To test your program with those inputs, execute the below:

$ ./fifteen 3 < 3x3.txt
$ ./fifteen 4 < 4x4.txt

Submission

After testing the program, push the code fifteen.c to your GitHub homework repository under the directory “extra_credit”.

If you have any questions, please let the instructor know.

Grading rubric:

- Code defines “int ** board” and “int d” in main function and is without any global variables. Code can be complied and run without error. Code is without memory leakage.

Total: 2 points towards the final grade.

Acknowledgement:
This assignment comes from CS50 by David Malan.