Code Assignment 6
Arrays, Loops and Functions

ECE 131 – Programming Fundamentals

Instructor: M. Wolverton

Items Due

- **main.c**
- **numslider.h**
- **numslider.c**

Instructions

Complete the coding directive in your C language IDE of choice (Eclipse, NetBeans) or by command line compiling. Once you have verified that the program works as intended, login to the file server at *cec-code-lab.aps.edu* and create a folder with the assignment name (e.g. Code 6). Locate all .c and .h source files written by you (e.g. numslider.h, numslider.c, main.c) and upload them to into the assignment folder. Code will be downloaded and archived for grading on the assignment due date.

Number Slider Puzzle Overview

Create an interactive **Number Slider** game. The player is presented with a 2d grid of numbers with one open position. Nearby numbers can 'slide' in to the empty space to rearrange them. The objective is to put all numbers in order from 1 … max with 1 in the upper right.

Sample Program Outputs

```
  ------------------
 | 1  2  3  4  5  6 |
 | 7  8  9 10 11 12 |
 |13 14 15 22 16 18 |
 |19 20 27 21 17 24 |
 |25 26 33 28 23 30 |
 |31  . 32 34 29 35 |
  ------------------
 Move - u,d,l,r     Quit – q
u
```

```
  ------------------
 | 1  2  3  4  5  6 |
 | 7  8  9 10 11 12 |
 |13 14 15 22 16 18 |
 |19 20 27 21 17 24 |
 |25  . 33 28 23 30 |
 |31 26 32 34 29 35 |
  ------------------
```

```
  ------------------
 | 1  2  3  4  5  6 |
 | 7  8  9 10 11 12 |
 |13 14 15 16 17 18 |
 |19 20 21 22 23 24 |
 |25 26 27 28 29 30 |
 |31 32 33 34 35  . |
  ------------------
==========================
=         YOU WIN!!!     =
==========================
```

Required Features

- Your program must accept working up,down,left,right and quit commands somehow. Make a menu or something simlar to display the game controls clearly.
- Your program must display the puzzle state somehow after each move entry. You may do this with `printf` or `puts`
- Your program must detect when puzzle is in solved state, indicate that the player has won and exit.
- Your program should randomly shuffle the puzzle in such a way that it is guaranteed to be solvable.

Required Code Structure

- Your program should use a 2d array to store the primary game data.
- Create a library consisting of a header(.h) and source(.c) file to store game functions in.
- Your library should contain at least 5 functions which must be appropriately named and utilized.
- At least one of your library functions must use one or more parameter.
- At least two of your library functions must return a non-void type, and your code must make use of its value somehow.

Optional Extras (Extra Grade Credit Possible)

If most all other features and requirements are working, the following features can be added for extra credit. They must be well implemented for consideration in score.

- Difficulty settings for the board setup which change the amount of shuffling.
- Game grid size settings which allow different dimensions / number of tiles in the game.
- A game timer based score system.