CSE 1321L: Programming and Problem Solving I Lab

Assignment 8 – 100 points

Search and Sort Algorithms

Program 0 (30%): Why – just why? One of the least useful sorts that professors never like to talk about (because it's SOOO inefficient and never used in practice) has two phases: 1) it shuffles the numbers in the array and 2) checks to see if they are in ascending order. Why is this so bad? Because there are $n!$ ways to order an array with n elements. For example, if there are 100 elements in the array, n == 100 and there are 100*99*98*97*…*5*4*3*2*1 different ways to arrange that! Ridiculous? Yes. That's why you're going to code it up. How do you shuffle an array? The easiest way is to traverse each element in the array with a loop, then pick a random element/cell to swap it with. Think through how to determine if it's in order (hint: traverse).

For this assignment, you need to implement only (source code) a program that 1) initializes and array of 5 elements with random values between 1-50 (i.e. no user input), 2) shuffles the array, 3) prints the array, 4) determines if the elements are in ascending (or non-decreasing) order and 5) repeats steps 2-4 until step 4 is true. Below is an edited sample output, only showing the last few lines of output. Hint: stay sane and use functions. Also, since you need to be able to visualize what's going on, we'd recommend starting with printing the array to the screen. Note: the random number generator below

```
Sample run 1:
|34|44|28|33|29|
|33|28|29|34|44|
|34|29|44|33|28|
|33|44|28|34|29|
|34|28|29|33|44|
|33|29|44|34|28|
|34|44|28|33|29|
|33|28|29|34|44|
|34|29|44|33|28|
|33|44|28|34|29|
|34|28|29|33|44|
|33|29|44|34|28|
|34|44|28|33|29|
|33|28|29|34|44|
|34|29|44|33|28|
|33|44|28|34|29|
|34|28|29|33|44|
|33|29|44|34|28|
|34|44|28|33|29|
|33|28|29|34|44|
|34|29|44|33|28|
|33|29|28|44|34|
|44|29|34|28|33|
|28|29|33|34|44|
```

Program 1 (40%): Implement *only* (source code) a program to determine whether two two-dimensional arrays are equivalent or not. Two arrays are equivalent if they contain the same values in any order. The program main method defines two two-dimensional array of size 3-by-3 of type integer, and prompts the user to enter integer values to initialize the arrays.

The main method calls method `isEquivalent()` that takes two two-dimensional arrays of integer and returns true (boolean value) if the arrays contain the same values in any order, otherwise it returns false.

Hint: use a sort method from Lab 14 in developing the solution for this problem.

Document your code and properly label the input prompts and the outputs as shown below.

Sample run 1:

```
Array A:
    1   2   3
    4   5   6
    7   8   9

Array B:
    1   2   3
    6   5   4
    7   8   9

Judgment: The arrays are equivalent.
```

Sample run 2:

```
Array A:
    1   1   1
    1   5   6
    1   1   1

Array B:
    1   1   1
    1   6   6
    1   1   1

Judgment: The arrays are not equivalent.
```

Program 2 (30%): You sank my Wabbit! For this program, you're going to extend the "Find the Wabbit" game to 2 dimensions to create the game Battleship. For simplicity, imagine you have an 8x8 board of holes and two "ships" that have pegs on the bottom that go into those holes. When you play, you place your ships on your board and your opponent does as well. However, you can't see each other's boards. So, you call out coordinates – guessing where your opponent's ships are and vice versa. If you guess a correct coordinate, your opponent says "Hit". Otherwise, they say "Miss". For this program, you're going to play against the computer, even though you know where two ships are ☺ I hard-coded this example to the following, where S means "Ship":

```
| | | | | | | | |
| |S| | | | | | |
| |S| | | | | | |
| |S| | | | | | |
| |S| | | | | | |
| |S| | | | | | |
| | | | |S|S|S| |
| | | | | | | | |
```

Write the program that prints out the board, but does not show where the two ships are (similar to Find the Wabbit). For each round, print the board and ask the player for an X and a Y coordinate. If there's a hit, the 'S' should be replaced by a '!'. If it's a miss, the ' ' should be replaced by 'X'. The game is over when the number of hits == 8 (two ships of size 5 and 3). You must use at least two meaningful functions that take in a 2D array. Note, when printing, you may need to print [y][x] instead of [x][y].

Sample run:

```
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
Enter x: 0
Enter y: 4
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
|X| | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
Enter x: 1
Enter y: 1
| | | | | | | | |
| |!| | | | | | |
| | | | | | | | |
| | | | | | | | |
|X| | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
```

```
[CUT]

Enter x: 4
Enter y: 6
| | | | | | | |
| |!| | | |X| |
| |!|X| | | | |
| |!| | | | | |
|X|!| |X| | | |
| |!| | | |X| |
| |X| | |!|!|!|X|
| | | | | | | |
64m3 0v3r
```

Submission:

1. Review the assignment submission requirements and grading guidelines (see the FYE website).
2. Upload the source code file(s) to the assignment submission folder in D2L.
3. The files must be uploaded to D2L by the due date.