Lab 01 – Double Matrix All The Way!

# Problem

We were issued the task of designing a Java program that prompted the user to input the size of a 2-D matrix. We then had to fill the newly created matrix with random integers, from 0 – 9. Once we successfully did this and printed the matrix to the screen, we then had to create a new matrix that was double in size of the previous, expand it with the numbers (in a 2x2 format) from our previous matrix, and print that out on the screen.

# Proposed Solution

When first reading the assignment, I immediately thought about the use of an array, greatly helped by a suggestion from class. It would need to be a multi-dimensional array to create a 2x2 look as suggested in the lab assignment. I thought of using some variation of a random number generator to create the data to fill the array, and performing simple arithmetic to create the second, larger array.

## Flow Chart

Use Math.random to generate #s

Create new double-sized array with #s from first

Output array to screen

Prompt user for array size

Initialize array[][] of size “input”

Use for loop to fill array with #s

Output array to screen

End program

# Tests and Results

I was successfully able to create the first matrix by using Math.double and then multiplying the result I obtained by 10 to create whole numbers. I had to use a force typecast to enforce whole numbers as opposed to floating point numbers with decimals.

Input: 3 Input 2 Input 1

Result: 7 3 9 Result : 8 1 Result : 9

8 1 3 4 7

4 8 2

I had significant issues with the second array, as described below.

# Problems Encountered

The first major problem I encountered was my unfamiliarity with arrays. I had to perform research and examine programs online to discover the proper implementation. After seeing several exams, I coded a working version of my first array. I had some output errors (improper alignment) until I converted the output to a String and used that for formatting.

I had a rough patch dealing with ArrayIndexOutofBounds exceptions until I realized I needed to have my for loops check for <x, rather than <=x as the final check would check for a nonexistent portion of the array.

My major problem that I encountered with this lab was the double sized array. I landed on a solution of creating a new array of size 2x the previous, and used a counter to layer the old numbers to the new array. I used the variables **ex** and **why** to keep track of my location in the new, larger array. The code I utilized was designed to take the value from (0,0) for instance, and then write that code to (0,1), (1,1), (1,0) and finally (1,1). After filling these coordinates with the values from (0,0), I sequenced the smaller array to (1,0) and repeated the process. I feel this code should work, but the formatting of the output is difficult to read and certainly not a finished product. I was unable to perfect the code for the second array before time at the end of the lab.

# Conclusions and Discussions

I feel that this lab was interesting, I got to utilize arrays much more than in my 145 class at Midlands Tech. I feel slightly behind some of the other students as I did not take CSCE145 at USC, and some of the concepts discussed in class that seem second nature to some students I have never dealt with. However, I feel I succeeded with the final result of this lab, and look forward to future assignments.