CSC345_Lab 03: "Processing 2D Arrays"

Points: **100** points

Objective:

- To design and implement functions to process 2DArrays.

Instructions:

- Be sure to document your code (add comments on top of each function).
- In the comments add your name, date, course, homework number, and statement of problem.
- Once you are done, upload your final solution through Blackboard.
- No need for input validation.
- Write a project called **`Arrays2D.c`**

Steps:

- It is up to you to choose what the return type should be. Be creative!
- Functions should not have pointers implementations.
- **`Arrays2D.c`** has the following functions

Part 1 [75 points]: (2-Dimensional Array Functions)

1. **[15 points]** Write a function called `max` that returns the maximum value in the 2d array.
2. **[15 points]** Write a function called `rowSum` returns the sum of the elements in Row x of the 2d array.
3. **[15 points]** Write a function called `columnSum` returns the sum of the elements in Column x of the 2d array.
4. **[15 points]** Write a function called `isSquare` that checks if the array is square (i.e. every row has the same length as the 2d array itself).
5. **[15 points]** Write a function called `displayOutputs` that displays the 2 dim-array elements.

Part 2 [25 points]: (Testing main) `Arrays2D`

- First declare a 2-dim array. How to do that? You need to read the number of rows and the number of columns from the user, and then it reads a corresponding entries to that size. E.g., if a user enters 3 for the number of rows, and enters 3 for the number of columns, then we declare an array of 9 and then read 9 entries and store them in the array. (Remember the Run-time Array Length!).
- Make calls to all functions in part 01 to match the sample demo below.
- Make sure you display the same exact messages.
- Use blank lines to separate outputs and make then more readable.
    - Look how I printed out the output format below.

Sample run:

```
Let's create a 2Dim array!

      How many rows? 2
      How many columns? 3

      enter [0][0]: 11
      enter [0][1]: 22
      enter [0][2]: 33
      enter [1][0]: 44
      enter [1][1]: 55
      enter [1][2]: 66

Sum of row 1 = 66
Sum of row 2 = 165

Sum of column 1 = 55
Sum of column 2 = 77
Sum of column 3 = 99

This is not a square array.

Here is your 2Dim array:
[11, 22, 33]
[44, 55, 66]
```

```
Let's create a 2Dim array!

      How many rows? 3
      How many columns? 3

      enter [0][0]: 10
      enter [0][1]: 20
      enter [0][2]: 30
      enter [1][0]: 40
      enter [1][1]: 50
      enter [1][2]: 60
      enter [2][0]: 70
      enter [2][1]: 80
      enter [2][2]: 90

Sum of row 1 = 60
Sum of row 2 = 150
Sum of row 3 = 240

Sum of column 1 = 120
Sum of column 2 = 150
Sum of column 3 = 180

This is a square array.

Here is your 2Dim array:
[10, 20, 30]
[40, 50, 60]
[70, 80, 90]
```