COP2221 – Intermediate C++ Programming
Module #6 Assignment Two
10 points

This assignment refers to **Learning Outcome #2: Create and utilize arrays to store lists of related data**

Programming Problem

There is nothing particularly creative about this problem – it's purely to practice working with 2-dimensional arrays. Create and initialize a 2-dimensional array (see Program 7-32 for the needed syntax). Use this data to populate your array:

4	24	62	91	101
77	0	45	33	6
56	7	1	27	211
1	4	0	22	50

Your program will have the following functions:

- ***getTotal***. This function should accept a 2-dimensional array as its argument and return the total of all values stored in the array
- ***getAverage***. This function should accept a 2-dimensional array as its argument and return the average of all of the array values
- ***getRowTotal***. This function should accept a 2-dimensional array as its first argument and an integer as its second argument. The second argument should be the subscript of a row in the array. The function should return the total of the values in the specified row.
- ***getColumnTotal***. This function should accept a 2-dimensional array as its first argument and an integer as its second argument. The second argument should be the subscript of a column in the array. The function should return the total of the values in the specified column.
- ***getHighestInRow.*** This function should accept a 2-dimensional array as its first argument and an integer as its second argument. The second argument should be the subscript of a row in the array. The function should return the highest value in the specified row of the array.
- ***getLowestInRow.*** This function should accept a 2-dimensional array as its first argument and an integer as its second argument. The second argument should be the subscript of a row in the array. The function should return the lowest value in the specified row of the array.

(see next page)

Demonstrate each of the functions in the program (call each from function main( )). No welcome or good-bye screen is needed. This program runs on its own to test your functions.

Grading Rubric

10 points – well documented, compiles and executes perfectly

8 points – lack of documentation, compiles and executes with a minor (or two) error(s)

6 points - lack of documentation, compiles and executes with a minor (or two) error(s), not all needed tasks are correct or included (missing some functionality)

5 and fewer points – doesn’t compile, lack of functionality to solve the programming problem, poor programming style

0 points – no submission