EGR 125 - Introduction to Engineering Methods (C++) Due date: ___________________
File: N125-Ch8L_and_Ch12L

Chapter 8 and Chapter 12 Homework:
Multidimensional Arrays and the Vector Class

Reading Assignment:

Read Chapters 8 and 12 in Introduction to Programming with C++, 3rd Edition by Liang

Problem Assignment:

1. (10 pts) Work the following ***Quick Check Exercises*** in the textbook. As usual, include the instructions and the given code with the answers.
   - ***Quick Check Exercise*** 8.4 on p. 313
   - ***Quick Check Exercise*** 8.5 on p. 314
2. (22 pts) ***Function with matrix argument***: Programming Exercise 8.1 on page 326 (prompt user to enter the elements in a matrix and call a function to find the sum of each column).
   - Include a printout of the main program and the function.
   - Include printouts for the test case in the textbook as well as one additional test case.
3. (22 pts) ***Matrix of temperatures***: Write a C++ program to repeat problem 4 from the Chapter 7 homework as follows:
   - Use a single 2D array with 3 columns and 19 rows for the temperature values, where the Farenheit values are stored in column 0, the Celcius values are stored in column 1, and the Kelvin values are stored in column 2. This is illustrated below:

32		
42		
52		
62		
72		
82		
92		
102		
112		
122		
132		
142		
152		
162		
172		
182		
192		
202		
212		

   - Use a loop or a list to assign the Farenheit values to column 0.
   - Use a loop to calculate and assign the Celcius and Kelvin values to columns 1 and 2.
   - Display the temperatures in three columns with a table heading.

5. (22 pts) ***Vector Class***: Write a C++ program that reads an unspecified number of temperatures in Celcius from the keyboard (prompt the user to enter them one at a time). The values should be added to a vector using the ***vector class*** with the ***push_back*** function. Instruct the user to enter an invalid temperature (<-273) to indicate that they are done entering values. Use functions and algorithms in the **<*algorithm*>** and **<*numeric*>** libraries to find the max, min, sum, and count (number of values) and display the results. Also sort the temperatures in increasing order and display the values. Print the program along with an example that includes at least 10 valid temperatures.

5. (24 points) Determine the output for each part below by hand (do not compile the programs).

Part A	`const int R = 4, C = 3;`
	`double A[R][C]={1,2,3,4,5,6,7,8,9,10,11,12};`
	`for (int i = 0; i < R; i++)`
	`{   for (int j = 0; j < C; j++)`
	`        cout << setw(3) << A[i][j];`
	`    cout << endl;  }`	

Part B	`const int R = 4, C = 3;`
	`double A[R][C]={1,2,3,4,5,6,7,8,9,10,11,12};`
	`for (int i = 1; i < R; i++)`
	`{   for (int j = 1; j < C; j++)`
	`        cout << setw(3) << A[i][j];`
	`    cout << endl;  }`	

Part C	`const int R = 4, C = 3;`
	`double A[R][C]={1,2,3,4,5,6,7,8,9,10,11,12};`
	`for (int j = 0; j < C; j++)`
	`{   for (int i  = 0; i < R; i++)`
	`        cout << setw(3) << A[i][j];`
	`    cout << endl;  }`	

Part D	`const int R = 4, C = 3;`
	`double A[R][C]={1,2,3,4,5,6,7,8,9,10,11,12};`
	`for (int i = R-1; i >= 0; i--)`
	`{   for (int j = C-1; j >= 0; j--)`
	`        cout << setw(3) << A[i][j];`
	`    cout << endl;  }`	

Part E	`const int R = 4, C = 3;`
	`double A[R][C]={1,2,3,4,5,6,7,8,9,10,11,12};`
	`for (int i = 0; i < R; i++)`
	`{   for (int j = 0; j < i+1; j++)`
	`        cout << setw(3) << A[i][j];`
	`    cout << endl;  }`	

Part F	`const int R = 4, C = 3;`
	`double A[R][C]={1,2,3,4,5,6,7,8,9,10,11,12};`
	`for (int i = 0; i < R; i++)`
	`{   for (int j = 0; j < C; j++)`
	`        if(i>=j)cout<<setw(3)<<A[i][j];`
	`    cout << endl;  }`	

Part
G
	const int R = 4, C = 3;
	double A[R][C]={1,2,3,4,5,6,7,8,9,10,11,12};
	for (int i = R-1; i >= 0; i--)
	{ for (int j = 0; j < C; j++)
	          cout << setw(3) << A[i][j];
	     cout << endl; }	

Part
H
	const int R = 4, C = 3;
	double A[R][C]={1,2,3,4,5,6,7,8,9,10,11,12};
	for (int i = 0; i < R; i++)
	{ for (int j = C-1; j >=0; j--)
	          cout << setw(3) << A[i][j];
	     cout << endl; }	