EGR 125 Due date: _____
Introduction to Engineering Methods (C++)
File: N125H5B

Homework Assignment #5 – Arrays

Reading Assignment:
Read Chapter 7 in C++ for Engineers & Scientists, 3rd Edition, by Bronson

Problem Assignment:

1. (36 points – 12 points per part) Write a C++ main program that calls each function listed below. Load values into each array using a list to test each function. Print out results to show original array contents and contents after using the functions. All printing should be done in the main programs – do not print from the functions. Turn in copies of the program and all results.
   Output: The output should clearly show original array contents and the result. For example, the output for part A might look as follows:

   Contents of array A before swap: 1 2 3 4 5 6
   Contents of array B before swap: 7 8 9 10 11 12
   Contents of array A after swap: 7 8 9 10 11 12
   Contents of array B after swap: 1 2 3 4 5 6

   A) Include a function to find the length of a vector stored in an array where the length of an N-dimensional vector L is: Length = $\sqrt{(L_0)^2 + (L_1)^2 + (L_2)^2 + \cdots + (L_{N-1})^2}$
      Form of function call: Result = Length(A, Size)
   B) Include a function to swap the contents of two equal size arrays named A and B.
      Form of function call: Swap(A, B, Size)
   C) Include a function to reverse the contents of an array. For example, an array that originally contained the elements 1,2,3,4 would contain 4,3,2,1 after the function call.
      Form of function call: Reverse(A, Size)
2. (3 pts) Complete Exercise 7.3, Problem 2 in the text.
3. (11 pts) Complete Exercise 7.4, Problem 6 in the text.
4. (11 pts) Repeat Exercist 7.4, Problem 6 above where the three 1D arrays are replaced by a single 2D array.
5. (11 pts) Write a C++ program that reads and unspecified number of temperatures in Celcius from the keyboard (prompt the user to enter them one at a time). Add each value to a vector created using the STL with the ***push_back*** function. Instruct the user to enter an invalid temperature (<-273) to indicate that they are done entering values. Use STL functions and algorithms 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.

6. (12 points) Determine the output for each part below by hand (do not compile the programs).

// Array Homework: Problem 6A
#include <iostream>
using namespace std;
int main(void)
{ int const ArraySize=10;
int Sum = 0, A[ArraySize] = {10,20,30,40,50};
double Avg;
for (int j = 0; j < ArraySize; j++) Sum += A[j];
Avg = Sum/ArraySize;
cout << "Avg = " << Avg << endl;
system("pause");
return 0;
}
	Avg = _____

// Array Homework: Problem 6B
#include <iostream>
using namespace std;
int main(void)
{ int const ArraySize=10;
int Sum = 0, A[ArraySize];
double Avg;
for (int j = 0; j < ArraySize; j++) A[j] = j*j;
for (int k = 0; k < ArraySize; k += 2) Sum += A[k];
cout << "Sum = " << Sum << endl;
system("pause");
return 0;
}
	Sum = _____

// Array Homework: Problem 6C
#include <iostream>
using namespace std;
int main(void)
{ int const ArraySize=10;
int Sum = 0, A[ArraySize];
double Avg;
for (int j = 0; j < ArraySize; j++) A[j] = j*j-j;
for (int k = 0; k < ArraySize/2; k++) Sum += A[k];
cout << "Sum = " << Sum << endl;
system("pause");
return 0;
}
	Sum = _____

```
// Array Homework: Problem 6D
#include <iostream>
using namespace std;
int main(void)
{   int const ArraySize=10;
   int Sum = 0, A[ArraySize] = {2,4,6,8,10,12,14,16,18,20};
   int B[ArraySize] = {1,3,5,7,9,11,13,15,17,19},C[ArraySize];
   double Avg;
   for (int j = 0; j < ArraySize; j++) C[j] = A[j]-B[j];
   for (int k = 0; k < ArraySize; k++) Sum += C[k];
   cout << "Sum = " << Sum << endl;
   system("pause");
    return 0; }
```

Sum = _____

```
// Array Homework: Problem 6E
#include <iostream>
using namespace std;
int main(void)
{   int const ArraySize=10;
   int Sum = 0, A[ArraySize], B[ArraySize], C[ArraySize];
   double Avg;
   for (int j = 0; j < ArraySize; j++)
   {   A[j] = j+4;
      B[j] = 2*j;   }
   for (int k = 0; k < ArraySize; k++)
   {   if (A[k]>B[k]) C[k]=A[k];
      else C[k] = B[k];
      Sum += C[k];   }
   cout << "Sum = " << Sum << endl;
   system("pause");
    return 0; }
```

Sum = _____

```
// Array Homework: Problem 6F
#include <iostream>
using namespace std;
int main(void)
{   int const ArraySize=10;
   int Sum = 0, A[ArraySize], B[ArraySize], C[ArraySize];
   double Avg;
   for (int j = 0; j < ArraySize; j++)
   {   A[j] = j;
      B[j] = ArraySize - j;   }
   for (int k = 0; k < ArraySize; k++)
   {   if (A[k]>B[k]) C[k]=0;
      else C[k] = 1;
      Sum += C[k];  }
   cout << "Sum = " << Sum << endl;
   system("pause");
    return 0; }
```

Sum = _____

7. (16 points) Determine the output for each part below by hand (do not compile the programs).

Part A	const int R = 3, C = 4;
	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 = 3, C = 4;
	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 = 3, C = 4;
	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 = 3, C = 4;
	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 = 3, C = 4;
	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 = 3, C = 4;
	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 = 3, C = 4;
	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 = 3, C = 4;
	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; }	