Engr 123 **February 23, 2015**
Hour Exam 2 Review

1. Write a console method which accepts an integer argument n and produces an n x n square of asterisks. That is if the method is called with:

```
int n = 4;
PrintSquare(n);
```

The following will be printed.

```
****
****
****
****
```

2. The following sequence uses a function called MinMax to find the minimum and maximum of an array. Write the function.

```
{int [] a = {1, 5, 2, 6, 8, 23, 45};
 int min, max;
 MinMax(a, out min, out max);
 Console.WriteLine("The min is {0}, the max is {1}", min, max);
}
```

3. An n x n magic square is a square array of n^2 distinct integers arranged such that the n numbers along any row, column, major diagonal, or minor diagonal have the same sum. For example, here is a 3 x 3 magic square in which all of the rows, columns, and both diagonals add to 15.

8	1	6
3	5	7
4	9	2

Write a method called CheckDiagonalSums which is started below. Your method should return true if the sum of the integers on both diagonals of its matrix parameter equals the second parameter. For example,

```
CheckDiagonalSums(m, 15);
```

Would return true for the matrix above since the sum of both diagonals is 15.

Complete the function CheckDiagonalSums below.

```
//precondition: m is a size x size matrix initialized with
distinct
//                positive integers.
//postcondition: returns true if the sum of each diagonal equals
sum,
//                returns false otherwise.
static bool CheckDiagonalSums(int [,]m, int sum)
  {
```

4. Write a method which will accept an an int array called a and will return the largest difference between any two adjacent elements. For example, the following sequence will print 4 since the largest difference is 7 – 3 = 4.

```
int[] a = {1, 1, 3, 7, 8, 9};
ConsoleWriteline(FindMaxDiff(a));
```

5. Given a function definition below. Write a new method which will evaluate this function beginning at x = 0 and continuing until the value of the function exceeds 1000. Increment x in steps of 0.1. Your method should return a double equal to the first value of x for which the function is greater than 1000.

```
private double F(double x)
  {double y;
   const double PI = 3.141592654;
   if(x < 0)
      y = -1;
   else
      y = x*x*x*Math.Sin(x*PI/180);
   return y;
  }
```