1. Create an integer array and fill it with 1000 random ints that range from 1 to 6. Calculate and print the following: number of 1's, 2's, .. 6's, and the average value.

2. Create a int array and fill it with the first 50 Fibonacci number numbers. (The first Fibonacci number is 0 and the second one is 1. The remaining numbers are always the sum of the previous 2.) Prompt the user for an int from 0 to 50 and print the Fibonacci number stored at that location in the array.

3. Create an array of 100 random ints which range in value from 0 to 1000. Calculate and print the minimum and maximum values stored in the array.

4. Create two arrays of doubles called x and y which have the following data: x = {0, 2.2, 3.4, -9.5, 10.3, 7.5, 2.1}, y = {0, -2.3, -8.5, -2.1, 4.4, 6.5, 0.9}. Treat values in the x and y vectors which have the same index as representing a point on a two dimensional grid. Use the distance formula to find the two points which are the farthest apart.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

5. The Sieve of Eratosthenes is an ancient simple method of finding prime numbers. To use the sieve to find all of the prime number between 1 and say 1000, we make an array of all integers from 1 to 1000. Begin with the number 2 and delete all multiples of 2. Next take 3 and delete all of the higher multiples of 3. Continue eliminating multiples until you get to the square root of 1000. The numbers left in the array will be the primes from 1 to 1000. Write a program to do the Sieve of Eratosthenes and print the resulting prime numbers.

6. Write a method which accepts the following array as a parameter and returns the array with all of the negative values set to their absolute value.
   ```
   int [] data = {1, -4, -2, 34, 52, -61, -33, -90, 12, 54};
   ```

7. Write a method called Rotate which shifts all the values in an array argument to the left 1 place with the value at index 0 becoming the last value in the array. For example, if the array is defined by `int [] data = { 4, 5, 6}`, your method should return the array with the data `{5, 6, 4}`

8. Write a method which accepts to int arrays of the same length and returns the first array modified so that it contains the sum, by element, of the two arrays. For example if the two arrays at `{4, 5, 6}` and `{1, 2, 3}`, the first array will be returned as `{5, 7, 9}`

9. Write a method which returns the integer sum of the odd elements in an int array passed as an argument.

10. Write a method which accepts a double array as an argument and returns the standard deviation of the values in the array. The standard deviation is defined as:

$$\sigma = \frac{\sqrt{\sum_{i=0}^{n}(a_i - \mu)^2}}{n-1}$$ where σ is the standard deviation, a_i are the elements of the array, μ is the mean or average values of the array, and n is the total number of elements in the array.

1. Write a method in C# which accepts a two-dimensional integer array as an argument along with two integer arguments which give the number of rows and columns in the array. Your method should return the average of the elements in the array. The following statement is a sample calling statement for your method.

```
int a[,] = {{9, 4, 3, 2], {4, 23, 32, 45}, {6, 31, 14, 26}};
Console.WriteLine(AddMethod(a, 3, 4);
```

2. The following statement creates a 1-dimensional array and initializes its values. Write a complete method named *FindFirstNegative(a, out n)* where a is the array and n is an int. The method checks the array to find the first negative number. It returns that number in the parameter n. Set n = -1 if there are no negative numbers.

```
int[] a = {0, 1, 2, 3, -5, 7, 9, 0, -11};
```

3. Write a method (started below) which will accept and array aData and an integer index. The method will return the difference between the number at the index and the number at index 0. For example, if aData = {14, 2, 13, 12, 3, 1} and n = 2 your program would return -1 since 13 - 14 = -1.

```
static int LessThanTen(int [] aData, int n)
```