C# general tasks

Set 3

Note: *for all these exercises, ignore input validation unless otherwise directed. Assume the user enters a value in the format that the program expects. For example, if the program expects the user to enter a number, don't worry about validating if the input is a number or not. When testing your program, simply enter a number.*

Task 1

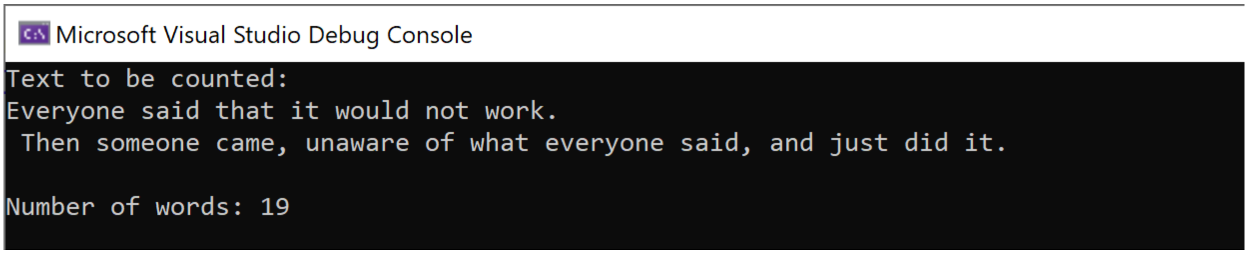
Count the words in a text and print out the text as well as the number of words.

Example string:

```
"Everyone said that it would not work.\n Then someone came, unaware of what everyone said, and just did it."
```

TIP: Look into the `Split()` method for strings for this task:

Example of output:



```
Text to be counted:
Everyone said that it would not work.
 Then someone came, unaware of what everyone said, and just did it.

Number of words: 19
```

Extra: This week we learned about files and reading from files. If you want to, see if you manage to count all words from a text file as well.

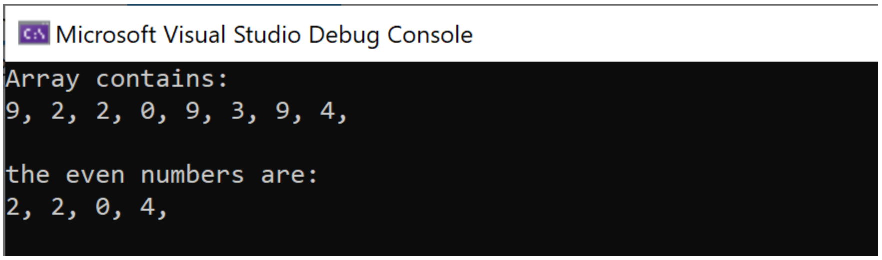
Task 2

Create a method, ExtractEvenNumbersFromArray, that takes an array of integer numbers. The method should return a list that only contains the even numbers from the array. To do this, you will need some way of checking if a number is odd or even, for example through the use of the modulus ('%') operator.

Test the method in Main. For example, try the following:

```
int[] numberArray = {9, 2, 2, 0, 9, 3, 9, 4};
Console.WriteLine("Array contains:");
foreach (int element in numberArray) {
    Console.Write(element + ", ");
}
Console.WriteLine("\n");

List<int> evenNumberList = ExtractEvenNumbersFromArray(numberArray);

Console.WriteLine("the even numbers are:");
foreach (int element in evenNumberList) {
    Console.Write(element + ", ");
}
Console.WriteLine();
```

Here is the result of the example Main, above:



```
Array contains:
9, 2, 2, 0, 9, 3, 9, 4,

the even numbers are:
2, 2, 0, 4,
```

Extra: There is a more elegant way to write all contents an arrays and lists, than the foreach loops used above. (We learned a bit about this last week). If you want to, improve on the example code above: Use the more elegant version!

Task 3

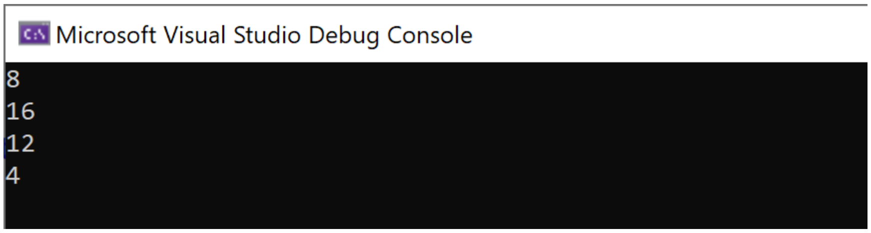
Implement 4 "calculator" methods for the basic math operations: addition, subtraction, multiplication, and division. The methods should each take 2 integers as parameters and give the result of the mathematic operation as their return value.

(Note: When dividing, C# will automatically drop decimals. This is OK.)

Test the calculator methods in Main. For example, try the following:

```
int result = 0;

result = Add(result, 8);
Console.WriteLine(result);

result = Multiply(result, 2);
Console.WriteLine(result);

result = Subtract(result, 4);
Console.WriteLine(result);

result = Divide(result, 3);
Console.WriteLine(result);
```

Here is the result of the example Main, above:



```
8
16
12
4
```

Extra: Instead of summarizing fixed values as in the example above, let the user enter valueA and valueB. Also, feel free to add more advanced math operations (as extra methods). :-)