**C#.NET and F#.NET Functional Programming Assignment [50 pts]**  
**Due Date**: Feb. 25, 2015

# How to submit your solution

You must implement the four functions below in both C#.NET and F#.NET. The load test must also be executed for both C# and F# implementation. For C# you must write a load test method (use System.Diagnostics.Stopwatch) and for F# you may use the #time “on” directive in the FSI window.

Provide a source code file for each language-specific solution (one Hw2.cs file and one Hw2.fs file).

The **Hw2.cs** file must contain the implementation class of the provided interface IHw2.cs.

The **Hw2.fs** file must contain the implementation class of the provided interface type IHw2.fs.

# **Function 1 [10 pts]:** The sum of even and the sum of odd numbers of a list (5 pts for C#, 5 pts for F#)

Implement a function that calculates the sum of all even numbers and the sum of all odd numbers of a given list by returning a tuple of these two results. **Note:** In C#.NET you must use LINQ extension methods to implement the two sums. Do not use explicit FOR/WHILE loops nor recursion. In F#.NET you may use recursion (or any method you like).

|  |  |  |
| --- | --- | --- |
| **Function**: SumEvenAndOdd | **C#.NET** | **F#.NET** |
| **Input** | List<int> | int list |
| **Output** | Tuple<int,int> | int \* int |

**Example**: When the input list is [1,2,3,4,5], then the function returns the tuple (6,9), where 6=2+4 (sum of even numbers), and 9=1+3+5 (sum of odd numbers).

# **Function 2.1** **[10 pts]**: The sub-list of all numbers divisible by three using LINQ (C#) and List library functions (F#)

Implement a function that extracts from the input list all numbers divisible by three and returns a list of these numbers, preserving the ordering in the original list. **Note:** In C#.NET you must use LINQ extension methods (no recursion, no explicit for/while loops). In F#.NET you must use an inbuilt List function from the “List” library, but you may not reuse this implementation for problems 2.2 and 2.3.

|  |  |  |
| --- | --- | --- |
| **Function**: ListElementsDivisibleBy3 | **C#.NET** | **F#.NET** |
| **Input** | List<int> | int list |
| **Output** | List<int> | int list |

**Example:** For the input [1,2,-3,4,-5,6,7] the function returns the list [-3,6].

**Function 2.2 [10 pts]**: The sub-list of all numbers divisible by three using and ITERATIVE approach (for, foreach, etc.)

Implement the same function using FOR (or WHILE) loops. In F# you should consider using the “yield” keyword inside your for loop.

**Function**: ListElementsDivisibleBy3Iterative

# **Function 2.3 [10 pts]:** The sub-list of all numbers divisible by three using RECURSION

Implement the same function recursively.

**Function**: ListElementsDivisibleBy3Recursive

# **Load Test** Functions 2.1, 2.2, and 2.3 and determine which is fastest and which is slowest for the same input **[10 pts]**

Generate a large list of integers (see function Enumerable.Range in C# and [1 .. 100000] in F#) and test all three versions of the ListElementsDivisibleBy3 functions above. List these three functions in order of speed of execution, from the SLOWEST to the FASTEST. **Important:** In**cl**ude the timings (in milliseconds) and the size of the list you tested on (i.e. how many elements).

**Example:** For an input list of size 10,000, the following timings were found:

|  |  |
| --- | --- |
| ListElementsDivisibleBy3Iterative | 123 ms (SLOWEST) |
| ListElementsDivisibleBy3Recursive | 23 ms |
| ListElementsDivisibleBy3 | 3 ms (FASTEST) |