

## Warmup

This is the first assignment for Part II, which will familiarise you with the automarker and with our programming languages, style, and tools.

The assignment covers 4 course marks and additionally offers 3 possible bonus marks:

- Base part, C#: 4 marks
- Bonus part, F#: 1 mark
- Bonus part, Node.js: 1 mark
- Bonus part, Python 3: 1 mark

The C# part allocates 2 marks for completing all automarker tests and 2 marks for using the indicated functional style (partial marks possible for programs that do not meet the functional criteria).

All bonus marks require both the completion of all automarker tests and using the required functional style (no marks will be awarded for partial fulfilment of these tasks).

The functional style required here mandates the use of **higher-order functions**, while avoiding any explicit classical control constructs from the **for/while** family. Note that comprehensions, although also good style, are also prohibited here, to ensure that we focus on higher-order functions.

Specific requirements:

- C#: Use a .NET Core 3.1 csproj-ect, and LINQ functions such as `Select()`.
- F#: Use a .NET Core 3.1 fsproj-ect, and functions of the standard core modules, such as `Seq.map` or `Array.map`.
- Node.js: Use Node.js 12.X and, optionally, package `sys`.
- Python 3: Use Python 3.X and, optionally, functions from modules `sys`, `functools` and `operator`.

You can start using the provided skeletal frames. Please ask us if you want to use something that is not listed here, as it may not be available on automarker.

**Problem:** Start with a text containing integer numbers in free format. Divide by two the even numbers. Remove duplicates. Print the sum.

The assignment requires the standard IO for the automarker.

**Input:** The input text file is given on **stdin** - Console.In in C# - by command-line redirection (<).

**Output:** Write on **stdout** - Console.Out in C#. In our case, write one single line, containing just the required number (no spaces).

Exceptions should be caught and their message should be printed on **stdout** (in lieu of the sum), prefixed by three stars and one space (\*\*\*)

A few examples will help clarify the specs.

### Example 1:

Test input:

```
1 2 3 4 5 6 7 8 9
 1 2 3 4 5 6 7 8 9
```

Input numbers:

```
1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9,
```

Even numbers halved:

```
1, 1, 3, 2, 5, 3, 7, 4, 9, 1, 1, 3, 2, 5, 3, 7, 4, 9,
```

Duplicates removed:

```
1, 3, 2, 5, 7, 4, 9,
```

Expected output:

```
31
```



**Submission:**

One single source file, to the **automarker** (<https://www.automarker.cs.auckland.ac.nz/student.php>):

- C# base part (required): Your C# source file, after changing its extension from .cs to .dcs.
- F# bonus: Your F# source file, after changing its extension from .fs to .fcs.
- Node.js bonus: Your .js files.
- Python 3 bonus: Your .py file.

**Notes for .NET Core:**

- The current automarker requires non-standard file extensions for .NET Core (.dcs, .dfs), to differentiate these from files that require the old .NET Framework (.cs, .fs).
- The required project files will be automatically provided by automarker, so don't worry about these.