M&G Analyst/[Developer Assessment Exercise](http://isconfluence.mandg.co.uk:8080/display/RT/Developer+Assessment+Exercise+-+Candidate+Details)

Introduction

The objective of this exercise is to test SEDOL checker implemented by C# developer based on requirements provided by Data Management team. The following excerpt from Wikipedia (<http://en.wikipedia.org/wiki/SEDOL>) provides some background information around SEDOL validation:

SEDOL stands for Stock Exchange Daily Official List, a list of security identifiers used in the United Kingdom and Ireland for clearing purposes.

SEDOLs are seven characters in length, consisting of two parts: a six-place alphanumeric code and a trailing check digit. The check digit for a SEDOL is chosen to make the total weighted sum of all seven characters a multiple of 10. The check digit is computed using a weighted sum of the first six characters. Letters have the value of 9 plus their alphabet position, such that B = 11 and Z = 35. While vowels are never used in SEDOLs, they are not ignored when computing this weighted sum (e.g. H = 17 and J = 19, even though I is not used), simplifying code to compute this sum. The resulting string of numbers is then multiplied by the weighting factor as follows:

First:  1; Second: 3; Third: 1; Fourth: 7; Fifth: 3; Sixth: 9; Seventh: 1 (the check digit)

The character values are multiplied by the weights. The check digit is chosen to make the total sum, including the check digit, a multiple of 10, which can be calculated from the weighted sum of the first six characters as (10 − (weighted sum modulo 10)) modulo 10.

**Example: Prudential SEDOL 0709954**

The checksum can be calculated by multiplying the first six digits by their weightings:

(0×1, 7×3, 0×1, 9×7, 9×3, 5×9) = (0, 21, 0, 63, 27, 45)

Then summing up the results:

0 + 21 + 0 + 63 + 27 + 45 = 156

The check digit is then calculated by:

[10 − (156 modulo 10)] modulo 10 = (10 − 6) modulo 10 = 4 modulo 10 = 4

For further details, please refer to the following sources:

<http://www.answers.com/topic/sedol>

<http://en.wikipedia.org/wiki/SEDOL>

Your Task

Implement SEDOL validation logic by using the following interface:

|  |
| --- |
| public interface ISedolValidator  {      ISedolValidationResult ValidateSedol(string input);  }  public interface ISedolValidationResult  {      string InputString { get; }      bool IsValidSedol { get; }      bool IsUserDefined { get; }      string ValidationDetails { get; }  } |

For a supplied input string, the validator is expected to return an instance of ISedolValidationResult. The following table shows the various inputs along with the expected outputs.

|  | **ISedolValidationResult** | | | |
| --- | --- | --- | --- | --- |
| **Scenario** | **InputString examples** | **IsValidSedol** | **IsUserDefined** | **ValidationDetails** |
| **Null, empty string or string other than 7 characters long.** | Null  “”  12  123456789 | False | False | Input string was not 7-characters long. |
| **7-characters long input with incorrect checksum digit** | 1234567 | False | False | Checksum digit does not agree with the rest of the input. |
| **Valid SEDOL that is not defined by the end user (leading character is not 9).** | 0709954  B0YBKJ7  B0Ybkj7 | True | False | Null |
| **Valid End User Defined SEDOL (leading character is 9).** | 9123458  9aBcDe1 | True | True | Null |

**Provided files**

|  |  |
| --- | --- |
| **Filename** | **Description** |
| ISedolValidator.cs | Interface of SEDOL validator. |
| IValidationResult.cs | Interface of validation result. |

**Expected solution format and restrictions**

* You **may not** collaborate with others or post your solution on a discussion board.
* Implement in .NET C# (preferably 4.0).
* Supply zipped Visual Studio 2010 or 2012 solution. Alternatively, provide only source code files in a zip archive.
* In order to avoid problems with firewalls/antivirus checker, please do **not include** any binary executable files (delete *bin*, *obj* folders and Nuget packages – keep only Nuget config files so that packages can be restored if needed).
* The solution will be executed in a test engine against test scenarios described above – in order to ensure smooth execution please implement the validator with *parameterless constructor* so that it can be launched as follows:

var results = tester.TestSedolValidator(new YourImplementedSedolValidator());

**Good luck!**