Dear Candidate,

below are the instructions of the test.

Remember. If you have any doubts or questions, I encourage you to contact me as soon as possible at claudio.corti@hondaracing.co.jp so that you don't waste valuable time.

Overview

The test consists in creating a solution based on .NET Framework 4.8.
The solution should be uploaded to a repository (public or private) at one of the providers of your choice between GitHub and GitLab. In case you opt for a private repository, this should be shared with the user @hrcHonda (same username for both providers).
You are given 48 hours to carry it out and push your final version on the repository.

The solution contains:

1) WPF project

Initially composed of a single view, you already know that it will become a large and complex project and so you understand that it must be set up accordingly (MVVM, no code-behind, IOC...)

The view will contain a ComboBox, a DataGrid, two empty output labels, and two buttons.

ComboBox

- Allows 3 options to be selected:
  - 4x4
  - 5x5
  - 36x36

DataGrid

- At runtime it takes the size selected by the ComboBox
- Initial cell values are a random integer from 1 to 9
- Allows selection of an area of the available cells (1)
- Allows copy/paste of the selected cells by using shortcuts: “Ctrl+C”, “Ctrl+V” (1)
- When edited, applies the new value to the entire selected cells (2)
- Validates that the numbers entered are integers < 10 (3)

Buttons

When buttons 1 and 2 are clicked, the values on the DataGrid are sent to two separate methods, exposed by a WCF service (see later). The outputs returned by the invoked methods should be shown concatenated to output labels 1 and 2.

2) WCF project

The WCF project exposes two methods to clients:

- CalcDeterminant: receives a matrix (2D array) and returns the determinant. (4)

- FilterAndOrderValues: receives a matrix, enumerate the values, discards odds, eliminates duplicates, sorts them in ascending order, and returns them as a string to the client. (4)

3) Test project

Create two uinit-tests for the two methods exposed by the WCF service. Any unit-testing framework can be used.

4) Class library COM visible project

optional

Create a class library that connects to the WCF service and wraps its two methods.
Make the output dll “COM visible”, so that any VBA program can consume, indirectly, the WCF services.

By making this project you no longer need to handle the WCF connection in other clients, so you can use this dll as a dependency in the WPF project.

5) Excel file

optional

If you have created the COM visible dll, you can now test it in the Excel attached file.
Call the two WCF methods wrapped in the dll with the provided values, and print the output in the related cells.
No need to replicate other behaviors of the WPF app

Constraints

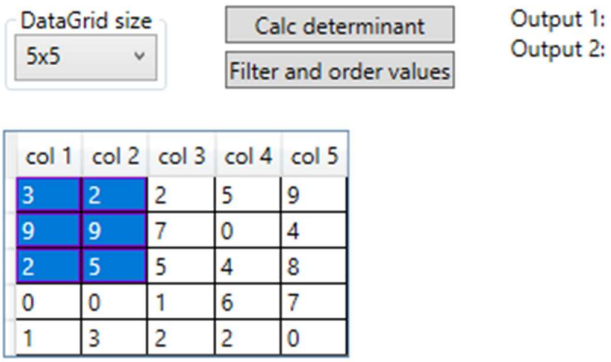
- .NET Framework 4.8
- C# language
- Namespaces, variables, classes, comments and any other keyword should be in English
- Use standard WPF components (DataGrid and others)

Key Aspects

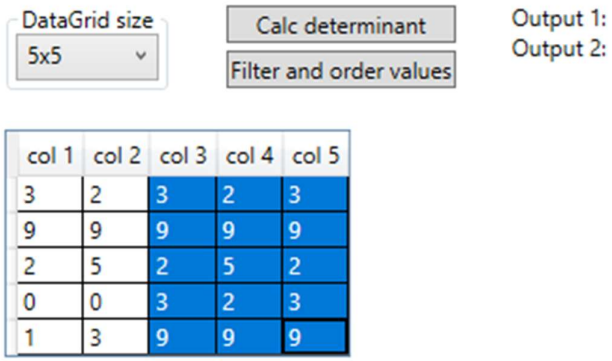
- Clean and readable code
- Reasonably fast at runtime: Avoid unnecessary bottlenecks.
- WPF project is not a PoC and it should soon scale to a mid/large-size application.
- Exception handling: WPF and WCF should never crash. Class library should throw useful error messages.
- Care to the user interface is not important, although appreciated.

Notes

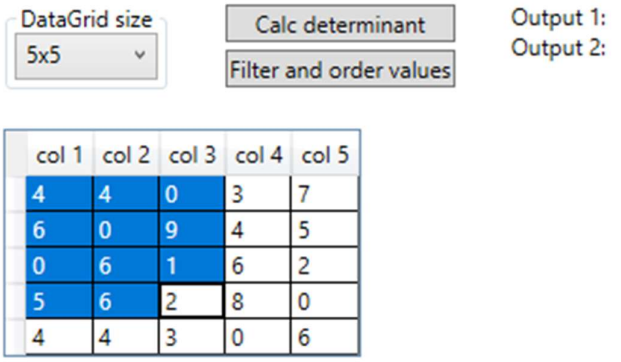
(1) Selecting an area of the DataGrid and copying the selected cells (purple border).

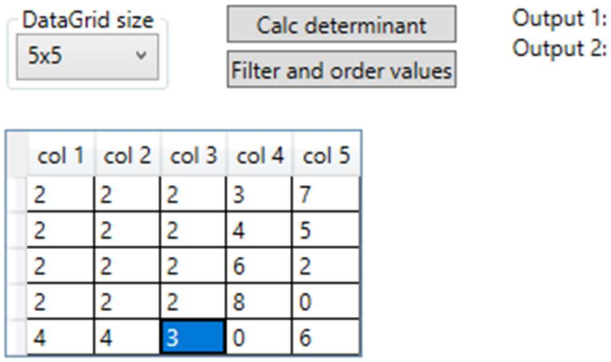


The previously copied area is pasted into a different, and larger, area of the DataGrid.

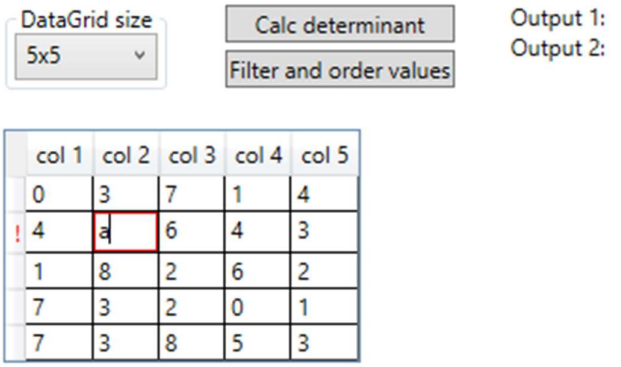


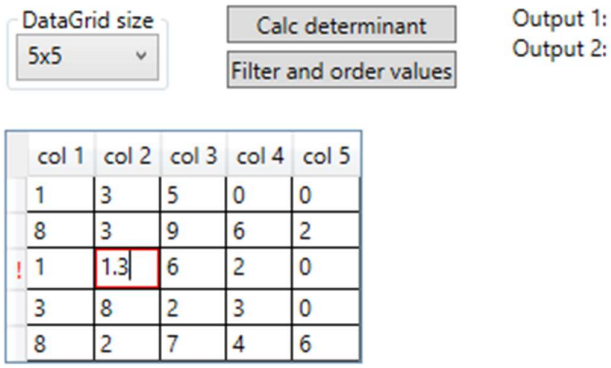
(2) When editing, the new value is applied to all selected cells





(3) Validates that the numbers entered are integers < 10





(4) Output 1 and Output 2