C# Developer questionnaire

Purpose
The purpose of this evaluation is to measure your proficiency in C#/ .NET, evaluate your design approach and see how you structure your code.

Prerequisites
Your solution should try and target .NET Framework 4 or higher using C#. We use Visual Studio 2015.

Where applicable submit the following:
1. Source Code
2. Design View
3. Implementation View
4. Deployment View (Use fictitious servers)
5. Data View
6. Readme.txt describing any special steps to follow to build and run the application.

Where applicable apply the following conditions:
1. Do not over engineer (This is exceptionally important to comply)
2. Apply KISS
3. All communication should be transactional

Good luck

Test 1

Pascal's Triangle

At the tip of Pascal's Triangle is the number 1, which makes up the zero'th row. The first row (1 & 1) contains two 1's, both formed by adding the two numbers above them to the left and the right, in this case 1 and 0 (all numbers outside the Triangle are 0's). Do the same to create the 2nd row: 0+1=1; 1+1=2; 1+0=1. And the third: 0+1=1; 1+2=3; 2+1=3; 1+0=1

```
            1
          1   1
        1   2   1
      1   3   3   1
    1   4   6   4   1
  1   5  10  10   5   1
```
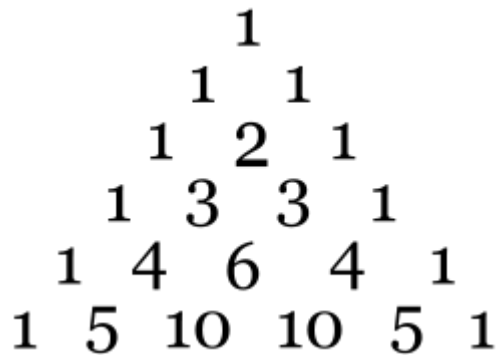
Write a C# console application that calls a **recursive function** that calculates the value of a row and column.

Example of console interface

Row : 4

Column : 2

Answer: 6

Test 2

Compound interest

The formula for annual compound interest is A = P (1 + r/n) ^ nt:

Where:

A = the future value of the investment/loan, including interest
P = the principal investment amount (the initial deposit or loan amount)
r = the annual interest rate (decimal)
n = the number of times that interest is compounded per year
t = the number of years the money is invested or borrowed for

Example:

Annual Compound Interest Formula:

A = P(1+r/n)^nt

If an amount of $5,000 is deposited into a savings account at an annual interest rate of 5%, compounded monthly, the value of the investment after 10 years can be calculated as follows...

P = 5000. r = 5/100 = 0.05 (decimal). n = 12. t = 10.

If we plug those figures into the formula, we get:

A = 5000 (1 + 0.05 / 12) ^ 12(10) = 8235.05.

So, the investment balance after 10 years is $8,235.05

Write a function to calculate the compound interest based on the above.
Write a unit test to test you function

Submit all code.

Test 3

Tic Tac Toe

Create an Asp.net web site project to play Tic Tac Toe against the computer.

Requirements:

Front End

- Use Angular JS to manage UI and front end calculations, example: if a block is already occupied then notify the user.
- You may use any style sheets you wish to style the UI (We use bootstrap).
- The user should first select if they wish to play X or 0 or set to random.
  X always starts.
- The user is then shown the game board and can drag their game piece (either X or O) onto the game board to an open square.
- Using Angular's $http, post the current board layout to the web api. The web api should receive that layout, calculate (as per Server Side functions below) and then send the board layout back and update UI.
- Communicate at the end of a game the result (Win, Lose or Draw), update overall win/lose/draw stats, and ask to play new game.
- Display overall stats if user says no to a new game.

Server side

The web api should have at minimum the following functions:

- Function that takes in current board state and return a new board state with a valid computer move.
- Function that updates Win Lose and Draw stats. Stats should be saved to text file.
- Function that reads text file and returns that stats on request.