

Create a .NET solution (either C# or F#) that will:

1) Calculate a single series of numbers given the following requirements:

a. The first number is calculated given the following function (which accepts a parameter 'x'):

$$((0.5 * x^2) + (30 * x) + 10) / 25$$

b. A growth rate for the series is calculated using the following function (which accepts parameters 'y' and the first number from a.):

$$(2\% \text{ of } y) / 25 / (\text{firstNumber})$$

c. The overall series is calculated using a function that accepts three parameters

firstNumber – the number from a.

growthRate – the number from b.

length - the length of the resulting series

The series should start with the first number; the subsequent numbers should be calculated as a product of:

$$\text{growthRate} * (\text{firstNumber}^{\text{index of the number being generated}})$$

d. The series should not contain any duplicates, be ordered from the lowest to the highest

e. The numbers in the series should be rounded to the nearest 0.25 (so 10.63 should be rounded to 10.75, 12.12 should be rounded to 12.00)

2) Select two 'special' numbers from the series:

a. **Number1** is the third largest number in the ordered series

b. **Number2** is chosen by first calculating the product of the following function

$$\text{approximateNumber} = y/z$$

where 'y' is a constant

and 'z' is an input of the function

then by selecting the closest number in the series to the approximateNumber. If two numbers are evenly apart from the approximateNumber the highest number is chosen.

Please provide unit tests to verify that your application is calculating correct results.

Please make sure your solution runs, if there are any special instructions to build, or run the solution please also let us know.

Please send us your solution either via Github, or using a file sharing service such as Dropbox – please do not send us the solution via email.

**Sample** - please remember that this is simply an example to illustrate the algorithm. The object of this exercise is to gauge your coding craftsmanship. Furthermore any application should work against a variety of input and edge cases.

Part 1)

- a. when  $x = 1$ , `firstNumber` = 1.62
- b. when  $y = 5062.5$ , `growthRate` = 2.5
- c, d when `length` = 5 the series is:

1.62  
4.05  
6.561  
10.62882  
17.2186884

e. rounded:

1.5  
4  
6.5  
10.75  
17.25

Part 2)

- a. 6.5 is special
- b. for  $y = 1000$  and  $z = 160$   
`approximateNumber` = 6.25, which makes 6.5 special.