Assignment 2 Due: 1.4.2022 (29.8.1443 AH)

Q1)

Write a python program that prints the first N approximations of π using the three series given below. Also print the percent relative error of each approximation.

$$\sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} = \frac{\pi}{4}, \qquad \sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}, \qquad \sum_{k=1}^{\infty} \frac{1}{k^4} = \frac{\pi^4}{90}$$

Bonus: Plot the results of the three series in one Chart (using Excel), use suitable N in your chart (say 20 or 25).

Try to use Excel features to produce the best Chart (Chart Title, Axes Titles, Legends, Colors,…).

We will post the best three Charts on Teams.

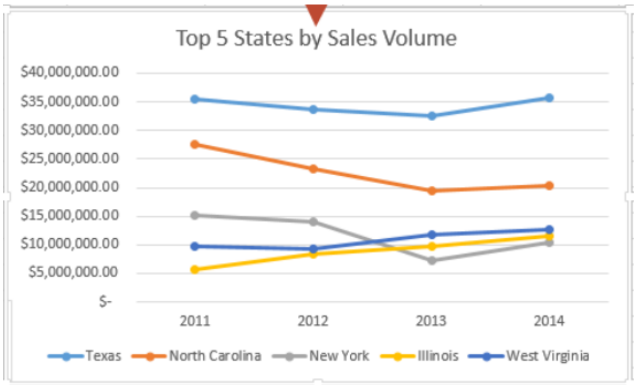
Remark:

You may like to save results directly to a text file. You may use this demo. Program:

```
import math
f = open("myfile.txt", "w")
for i in range (10):
    f.write(str(math.sqrt(i))+'\n')
f.close( )
```

Submit the Python Program (and the Bonus: Excel file).

This is just a demo of a similar Excel Chart:



Q2) Find the largest interval in which $\sqrt{5}$ approximations must lie so the relative error is at most 0.01 for any of these approximations.

Q3) A company has four owners: (W, X, Y and Z) with certain number of shares for each of them.

Assume we computed the shares percentages and got:

Percentage of shares of W: 17.567%

Percentage of shares of X: 65.654%

Percentage of shares of Y: 6.679%

Percentage of shares of Z: 10.1%

Total: 100.000%

If we round these percentages to the nearest whole number, we get:

W: 18%

X: 66%

Y: 7%

Z: 10%

Total: 101%

What is the cause of the error in this situation?

Write a python program that reads a list of real numbers and prints the percentage (as whole numbers) of each of them (with respect to their total) such that the printed percentages are of sum 100% by subtracting 1 (or adding 1) to one of the percentages (applicable only if they don't sum up to 100%) which results in the minimum error.

Hint: Notice that the **absolute error** is more suitable to be applied here. Notice also that the program **needs not** include any error formula computations since you may come up with a 'simple' algorithm that guarantees the minimum error.

Example: In the above example: the 18% will be subtracted by 1 (to be 17%).