ME1 Computing - End of Term test

CID number:	0								
	1st	2nd	3rd	4th	5th	6th	7th	8th	

Imported libraries allowed: *random, maths, matplotlib.pyplot, Numpy*

STATE YOUR CID in a comment at the beginning of every file

Save each task into a different file named TaskA, TaskB, etc.

You can submit Jupyter files (.ipynb), Python files (.py) or text files (.txt).

Marks are out of [75]

Remember to comment appropriately all your scripts. Comments are marked too! [5]

Task A [10]

Write a function, *Series()*, to calculate the mathematical series:

$$S = \sum_{j=-N}^{N} \left(\frac{1}{100 - a^2} \sum_{\substack{k=-j \\ k \neq 0}}^{j-1} (-1)^j \frac{\sin(ak)}{k^a} \right)$$

where a is the 4th digit of your CID.
The function receives the value of N and returns as output the result S.

Compute the sum S for integer values of N in the range [1…20].
Plot the various values of S against the number of terms N.
Save your files frequently

Task B [35]

The file *Matrix.txt* contains 360,000 numerical values. Read in the content and organise the values into a mathematical matrix *A*, with dimensions 600 x 600.

Consider the matrix A as if subdivided into many (60 x 60) smaller sub-matrices of dimension 10 x 10 each, as depicted in Figure A. Write a script to implement the following operations:

1) For every sub-matrix, determine the average of all elements, and store the results into a matrix *AV*, of dimension 60 x 60.
2) For every sub-matrix, compute the determinant, and store the results into a matrix *Det,* of dimension 60 x 60.
3) For every sub-matrix with a non-zero determinant compute its transpose and substitute the original sub-matrix in A with the transpose.

For the three operations, try to use the minimum overall amount of loops.

4) Save the matrix *AV* into a file named *Averages.txt*, one element of the matrix per line of file, and the matrix *Det* into a file named *Determinants.txt*, one element of the matrix per line of file.

5) Plot the final matrix A as an image.

To compute the determinant and the transpose, you can use the Numpy built in functions np.linalg.det and np.transpose, respectively, if you wish, or you can re-use any functions from the tutorials.

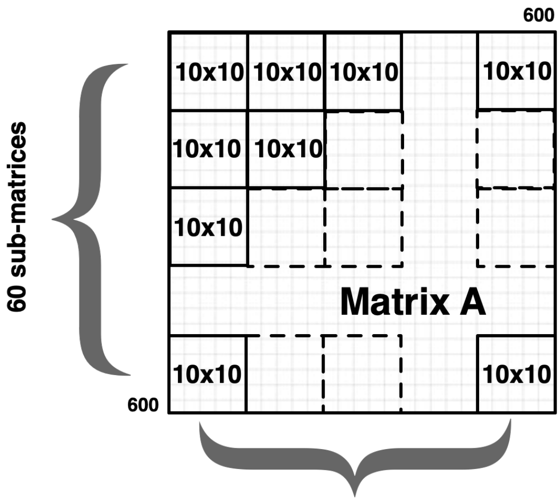

Figure A

Save your files frequently

Task C [15]

An ant is located within a pipe, with shape and dimensions as specified in Figure C. The pipe contains two circular obstacles centred at (-2,1) and (-1,2), respectively, both with radius 0.4.
The spatial domain is already set up for plotting in file *TaskC.txt*.
The ant is initially located at position (-3,0) and moves in small steps.
At every step the ant jumps to a new position by a small distance. The lengths, *dx*, *dy* of the motions are independent random values between *-0.3* and *0.3*.
If, after a step, the ant hits one of the walls (or the circular obstacles) or goes beyond them, it is bounced back to the last position.
Write a script to simulate the movement of the ant, until it escapes through the exit. If the ant does not manage to escape after 1000 moves (including those when it hits a wall or an obstacle and bounces back) the script should stop and declare with a printed message that "The ant is exhausted".
Plot the trace of the ant's steps.

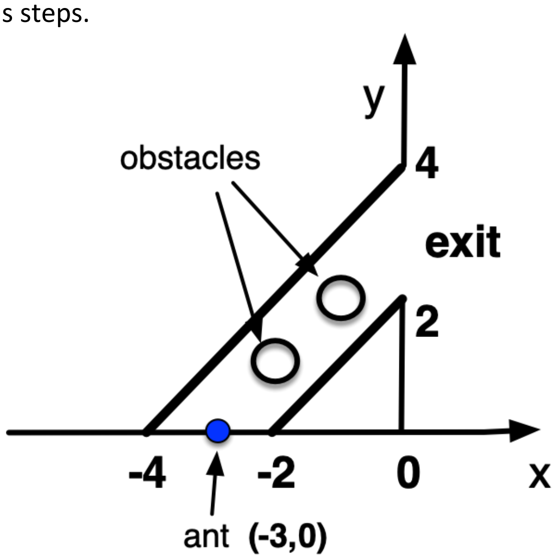

Figure C

Task D [5]

Write a **RECURSIVE** function, *ExpSeq*(), to calculate the sequence:

$$y_n = (10 - a)^n$$

for any $n > 0$, where a is the 4th digit of your CID. The function receives the value of n only and returns the value of y_n only.
Print the first 10 values of the sequence.
Save your files frequently

Task E [5]

Amend the lines 3rd and 4th only, of the script below, in order to plot the following graph, Figure E:

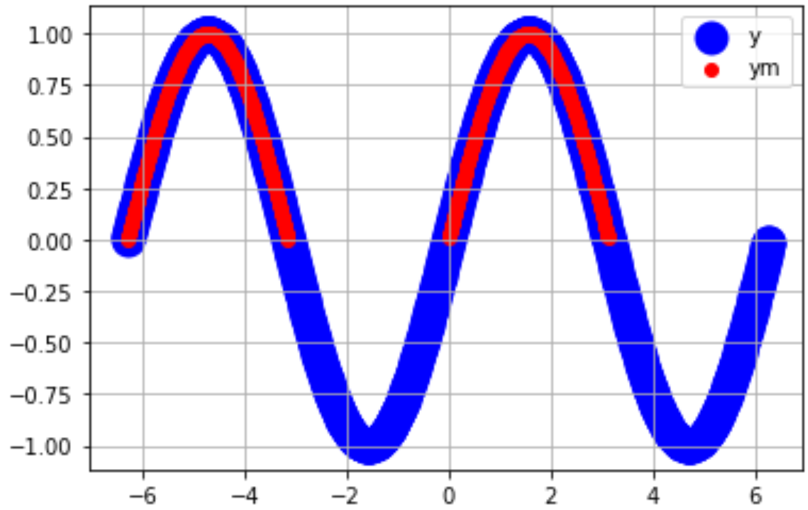

Figure E

```
x = np.arange(-2*np.pi,2*np.pi,0.01)
y = np.sin(x)
xm = x[]
ym = y[]
pl.scatter(x,y,Linewidth=10,c='b')
pl.scatter(xm,ym,Linewidth=1,c='r')
pl.grid()
pl.legend(['y','ym'])
```

The script is also available on Blackboard in file *TaskE.txt*.

Upload files TaskA, TaskB, TaskC, TaskD, TaskE into Blackboard and submit.