ME1 Computing - End of Term test

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

Python 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 [80]

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

Task A [10]

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

$$S = \sum_{j=-N}^{N+2} \left(\frac{1}{j!} \sum_{\substack{k=2 \\ k\ even}}^{10j} (-1)^j \frac{j^k}{k!} \right)$$

The function receives the value of N and returns as output the result S.

Compute the sum S eight times, taking N as each digit of your CID.
Plot the various values of S against the number of terms N.
Save your files frequently

Task B [35]

The file *Matrix.txt* contains 49 numerical values. Read in the content and organise the values into a mathematical matrix A, with dimensions 7 x 7, composing it row by row.

Write a script to form a matrix C, obtained by augmenting matrix A with an extra row and an extra column, positioned as first row and first column, respectively, as depicted in Figure 1.

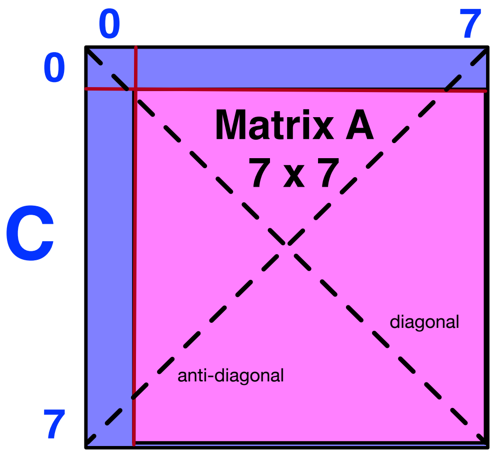

Figure 1

Each element C_{0j} of the new first row will contain the largest value of the respective column in A:

$$C_{0j} = largest\ value\ of\ A\ in\ column\ j-1, with\ j > 0$$

Each element C_{i0} of the new first column will contain the smallest value of the respective row in A:

$$C_{i0} = smallest\ value\ of\ A\ in\ row\ i-1, with\ i > 0$$

The first element C_{00} is set $C_{00} = 0$.

Implement the following operations:

1) Write a function, *Diagonals()*, to calculate the sum of all the elements in the diagonal of C and the sum of all the elements in the anti-diagonal of C. The function receives a matrix as input argument and returns a tuple with two values as output argument, namely the values of the two sums.

2) Write a script to compute the sum S of the determinants of all the minor matrices that can be obtained from matrix C, i.e.,

$$S = \sum_{r,c} |M_{rc}|$$

where M_{rc} is a minor of C, for all rows r and columns c of matrix C. (Re-use as much codes as possible from tutorials).

3) Subdivide matrix C in four sub-quadrants. Compose a new matrix D, of same size as C, where each sub-quadrant is obtained by transposing the homologous sub-quadrant of C (see Figure 2).

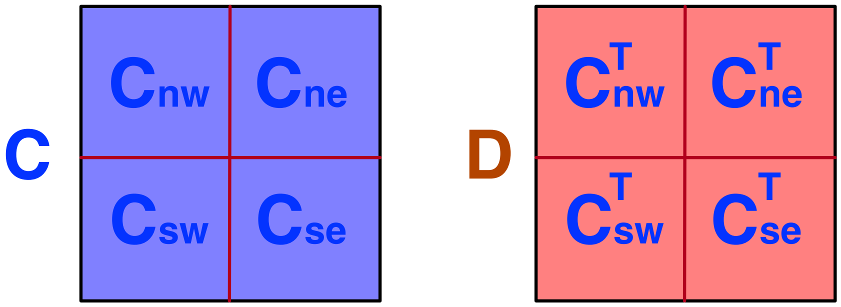

Figure 2

Save your files frequently

Task C [14]

In a videogame, the character Super Mario is trying to reach the winning line by moving with a zig-zag motion only. Super Mario is initially located at the starting position ($x = 0, y = 0$), and needs to reach the winning line located at $y = 100$, Figure 3. Super Mario moves with incremental steps. At each step, he moves firstly horizontally, by a distance dx, where dx is an integer random number between 1 and 10, and then vertically by a distance $dy = dx$.

Write a script to keep Super Mario moving, until he reaches/exceeds the winning line. Calculate the horizontal distance D walked through and how many steps Super Mario needed to win.

Plot the zig-zag path taken by Super Mario to reach the winning line.

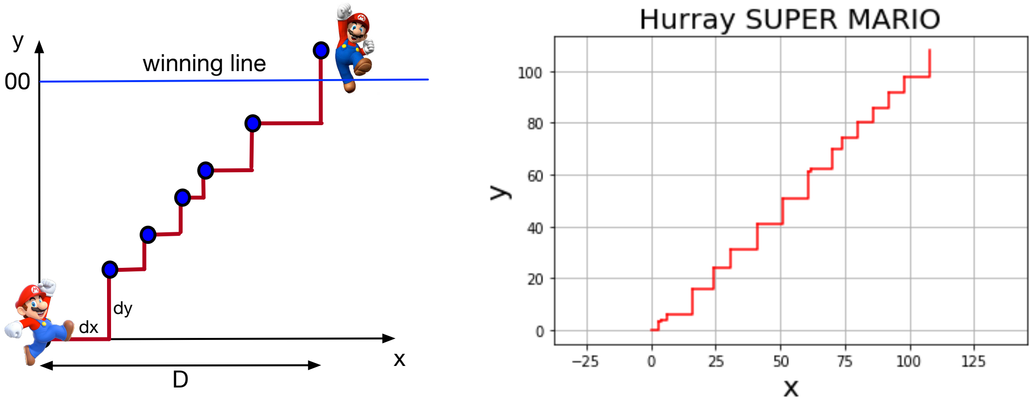

Figure 3

Task D [7]

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

$$y_1 = 0$$
$$y_2 = 1$$
$$y_n = y_{n-1} - (n+1)y_{n-2}$$

for any $n > 0$. The function receives the value of n only and returns the value of y_n only.

Print the first 20 values of the sequence.

Save your files frequently

Task E [9]

Define a class *Trigonometry*, with attributes x and $unit$. Attribute x represents a sequence of numbers. Attribute $unit$ is a Boolean variable with value *True* if numbers in x are given in radians and with value *False* if numbers in x are given in degree.

Write a method, *cos()*, to determine the cosine of an object of class *Trigonometry*, taking into consideration if the sequence of values are given in radians or degree.

Define an object xr of class *Trigonometry*, containing 100 values, equally spaced, in radians, in the range $[0{:}\,2\pi]$ and another object of the same class, xd, containing 100 values, equally spaced, in degree, in the range $[0{:}\,360]$.

Apply the method *cos()* to both objects and plot the results in two different plots.

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