

**ME1 Computing- End of Term test**

Name:	CID number:
-------	-------------

**Before you start**

In your H drive create a folder *H:\ME1MCP\EndTest* and work within it.

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

**Section 1: Short questions** (to be solved in the given space below)

1. Write a function, *LastLine*, that receives a matrix **A**, as a list of lists, and returns a single list with the values of the last line of **A**. (Max three lines of code).

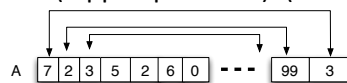
.....  
 .....  
 .....[4]

2. Find and correct the two mistakes in the following recursive function, such that it implements correctly the sum:  $S = \sum_{n=2}^N 3^{n-2}$

```
def Sum(N):
    if N == 2:
        S = 0
    else:
        S = 3**(N-2) + Sum(N-2)
    return S
```

[5]

3. Write a function, *Flipping*, in the space below, that receives a list of items, and returns the same list in reverse order (flipped positions). (Max 6 lines of codes)



.....  
 .....  
 .....  
 .....  
 .....  
 .....[5]

**Section 2: Writing scripts**

You will need to submit these scripts through BlackBoard.

*Comment appropriately all your scripts. Comments are marked too!*

[5]

1. Write a function, *AAT*, that receives a matrix **A** and returns the matrix **C**:

$$C = AA^T$$

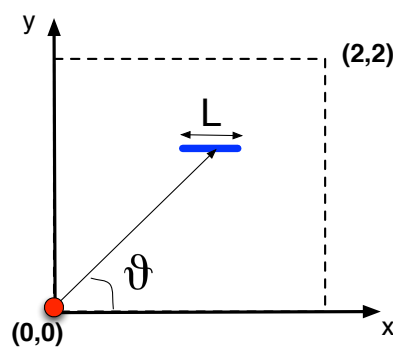
where  $A^T$  is the transpose of matrix **A**.

[16]

2. In the game illustrated in the figure below, a point bullet is shot from the origin (0,0) towards a one-dimensional horizontal brick of length  $L = 0.2$ . The brick is positioned randomly anywhere within the square window delimited by the corners (0,0) and (2,2).

Write a script, *Brick*, that generates the horizontal brick, with random allocation within the window, and determines the range of shooting angles,  $(\theta_{\min}, \theta_{\max})$ , in degree, for which the bullet will hit the brick directly, in a straight line (no gravity or bouncing need to be considered). Plot the line trajectory (and the brick) when the angle is in the middle of the range.

[Useful inverse trigonometric functions, in *Math* library, could be *atan()*, *asin()*, *acos()* ].



[18]

3. The file *FTSE100.txt* contains information about the best 100 stock exchange market companies. For every company the following data are stored sequentially, one per line:

Name
Share price (in \$)
Number of shares allocated

Write a script, *LSE*, to analyse the data as following:

- (a) Organise all the information into a list of tuples, one tuple per company.
- (b) Sort the list by the share price, in descending order.
- (c) Compute the overall value, in billion dollars, of all the FTSE companies. Each single company is worth the share price times the number of shares allocated.

[17]