ME1 Computing, Consolidation 1: sessions 1-4

Learning outcomes:

- Recap and consolidate topics covered in Sessions 1-4

Before you start

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

Exercise 1: Counted loops: Series expansion

The function *sin(x)* can be evaluated with the Taylor expansion:

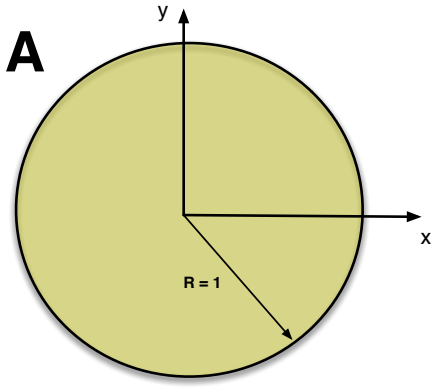
$$sin(x) = \sum_{\substack{i=1 \\ i\ odd}}^{N} (-1)^{int(\frac{i}{2})} \frac{x^i}{i!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \dots$$

Write a script to compute and plot *sin(x)* vs *x* in the range *x = [0 : 2π]* with step 0.01, for a given *N*.

Exercise 2: Counted loops and conditional flow: Ant movement tracing

Write a script to simulate and visualise the random motion of an ant within a closed circular domain.

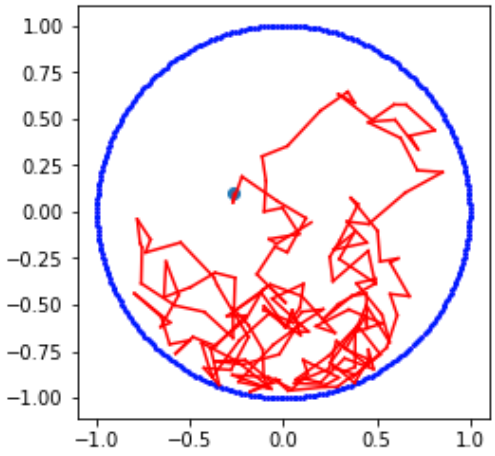
Initially generate **a** random point within the circular domain A and plot it.



Then move the point N times: at every step the ant must jump to a new position by a distance (*dx,dy*). The sizes *dx*, *dy* of the move must be independent random values between -0.2 and 0.2.

If the ant, after moving, leaves the domain, it must bounce back to the current position.

Plot the trace of the ant's moves.



Exercise 3: Lists: Dammit I'm mad!

Write a script to establish if a sequence of characters is palindrome.
A word or sentence (excluding any punctuation, upper/lower case and blank spaces) is palindrome if it is the same when read from left to right or right to left.
Input the word to be examined from the keyboard.

d	**a**	**m**	**m**	**i**	**t**	**i**	**m**	**m**	**a**	**d**

Once completed, you can test your script with these examples:

I did, did I?
Don't nod.
I, man, am regal; a German am I?
No mists or frost, Simon.
Did Hannah say as Hannah did?

Exercise 4: Lists: Anagrams

Given two strings, write a script to determine whether one string is the anagram of the other.

Exercise 5: Lists and Tuples

The budget of a company (in thousands pounds) for year 2018 is stored in the file *Budget.txt* with the following numerical format (column 1 of the table):

1	*this month*
4	*num. of expenses this month*
5.5	*expenses 1*
15.5	*expenses 2*
2.3	*expenses 3*
4.5	*expenses 4*
3	*num. of incomes this month*
2.6	*income 1*
4.8	*income 2*
1.0	*income 3*
2	*this month*
3	*num. of expenses this month*
2.5	*expenses 1*
	etc.

Write a script to read the file and organise the data in a list of tuples:

(month, total expenses, total incomes)

For every month compute the total savings and store them in the file *Savings.txt*.