ME1 Computing- Session 7: Recursive functions and Matrices

Learning outcomes:

- Explore how to code functions in recursive form
- Being able to define and generate matrices
- Being able to compute basic matrix operations

Please provide feedback at: www.menti.com with code 63 53 57

Before you start

In your H drive create a folder *H:\ME1MCP\Session7* and work within it.

Task 1: Recursive function: Fibonacci sequence

The Fibonacci set is a mathematical sequence used to represent physical events such as spiral galaxies, hurricanes and the evolution of prolific species (like rabbits).

The Fibonacci sequence is generated, starting from the seeds 1 and 1, by adding up the two previous numbers of the sequence, i.e.

1	1	**2**	**3**	**5**	**8**	**13**	**21**	…
1st	2rd	3h	4h	5th	6th	7th	8th	

$$F(n) = F(n-1) + F(n-2)$$

Write a **recursive** function, *Fibonacci,* to compute the n-th Fibonacci number. Then write a script to make use of the function, to generate the first N numbers of the sequence.

Answer Question 1

Task B: Defining matrices

Write the following tasks into a new script:

1. Create a matrix A with values: $\begin{bmatrix} 24 & 67 & 81 \\ 10 & 3 & 28 \\ 63 & 55 & 17 \end{bmatrix}$
2. Swap 67 with 17. Then Swap 81 with 63. Sum 24 with 28 and replace 3 with the result.
3. Create a matrix B with values: $\begin{bmatrix} 3 & 5 & 7 \\ 2 & 4 & 6 \end{bmatrix}$

Answer Question 2

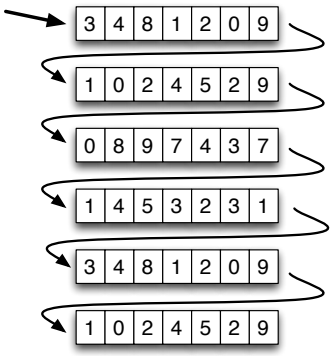
Task C: Patterns: the Scottish flag

Write a script to generate a *N x N* matrix (the size N must be inputted from the keyboard), with the following pattern:

$$\begin{pmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \end{pmatrix}$$

Task D: Matrix operations: sum of two matrices

The files *MatA.tx*t and *MatB.txt* contain the values of two matrices of size 60x60. Entries of the matrices are stored in the file one value per line, sequentially as:



Read in the numerical values from the two files and form the two matrices A and B accordingly.

[*Suggestion:* instead of writing twice the lines of code for reading the files and forming the matrix, write and use a function *ReadMatrix* that receives the name of the file to be read and the size of the expected matrix. The function *ReadMatrix* will then return the matrix in the expected form].

Answer Quiz 3

Compute a third matrix C, as C = A + B.

Answer Quiz 4