

ME1 Computing- Session 9: Consolidation

Learning outcomes:

- Recap and consolidate topics covered in Sessions 1-8

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

Before you start

In your H drive create a folder `H:\ME1MCP\Session9` and work within it.

Task A: Trace of a matrix

Write a function, *Trace*, that receives a square matrix and returns its trace, as defined in Maths lectures.

Task B: Matrix-matrix multiplication

Write a function, *MatMat*, that receives two matrices, **A** and **B**, of size $M \times N$ and $N \times P$, and returns the product of the two matrices, $\mathbf{C} = \mathbf{AB}$, as defined in Maths lectures.

The function should return the value 0 if the sizes of the two matrices are incompatible for the product.

Task C: Anagrams

Write a Boolean function, *IsAnagram*, that receives two strings and determines whether one string is the anagram of the other.

Task D: Experimental analysis

The performance of a boiling system has been monitored with daily measurements of its water temperature.

Measurements were performed throughout ten different days. Each day, measurements were taken at every hour, for a total of 24 data per day.

The data measured are stored sequentially, one per line, in the file *Temperatures.txt*.

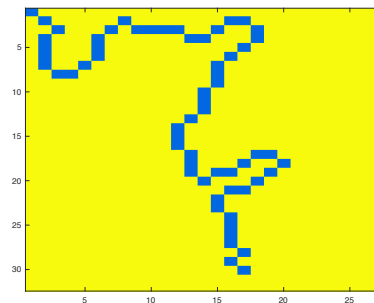
Compute the average temperature, T_{av} , of the water for each day, and plot the result in a graph with T_{av} vs *day*.

Compute also the maximum and minimum temperatures overall the testing period.

Bonus Tasks:

Task E: The Pirates treasure map: path searching

A matrix *Map* contains the path to a hidden treasure. Stones of the path are represented by the value 0 in the matrix and are laid at interior positions, never touching the boundaries of the domain. The hunt starts from position [0][0]. Every successive step is unambiguous, as there is only one possible adjacent forward move, from any current position.



Write a script to follow the secret path and find the treasure.

Files *Map1.txt* ... *Map4.txt* contain different map exemplars for you to test the script. [Matrices are stored in the same format as in Session 8, i.e.: Line 1: M rows; Line 2: N columns; Line 3 and subsequent ones: single entries of the matrix.]

Task F: The Towers of Hanoi

This task is quite challenging and above expectations, but presented here if you wish to challenge yourself. If done with a recursive logic, the code will require very few lines: do not attempt it iteratively.

The Towers of Hanoi is a game made of three pegs and a number of disks of increasing size.

The objective of the game is to start with the disks stacked in the left peg and to move the entire stack to the right peg, obeying the following rules:

1. only one disk at a time can be moved;
2. no larger disc can be placed on top of a smaller disk.

Write a recursive function to generate the sequence of moves, for a given number N of disks.

