Programming for Data Science (11/07/2023)

0% of the points are assigned to quality of documentation and/or comments to solutions.
Solutions must include tests of executions of the developed functions.

Name files as "<your matricola>_<firstname>_<lastname>_ex2.py" for Exercise 2, and "<your matricola>_<firstname>_<lastname>_ex3.c" for the third exercise.

Upload the TWO files in a folder
(named with your student number and your last name) at the following URL: Upload here
(access GDrive using your university credentials)

.Exercise 1. (Math, on paper)

Let A be an nxn matrix with integer entries. We say that matrix A is **congruent to the identity matrix modulo 5**, written **A ≡ I (mod 5)** if each entry of the matrix A, when taken modulo 5, is congruent to the corresponding entry in the identity matrix of size n x n.

Formally, A ≡ I (mod 5) if and only A and I have the same dimension and $\forall i, j.$ $A_{ij} \equiv I_{ij}$ (mod 5)

Example 1: Consider matrix A = $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, we have A ≡ I_{2x2} (mod 5) since A is already the identity matrix 2x2.

Example 2: Consider matrix B= $\begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix}$, we have B ≡ I_{2x2} (mod 5) since B is 2x2 and for each entry B_{ij} ≡ I_{ij} (mod 5)

Example 3: Consider matrix C = $\begin{bmatrix} 11 & 10 \\ 20 & 16 \end{bmatrix}$, we have C ≡ I_{2x2} (mod 5)

Example 4: Consider matrix D = $\begin{bmatrix} 11 & 10 \\ 20 & 19 \end{bmatrix}$, we do **not** have D ≡ I_{2x2} (mod 5) since D ≡ $\begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$ (mod 5) which is not the identity.

Now, let A, B, and C be n x n matrices with integer entries and I is I_{nxn}. Determine which of the following statements are true and which are false:
a) If A ≡ I (mod 5) and B ≡ I (mod 5), then A + B ≡ I (mod 5).
b) If A ≡ I (mod 5), then A^2 ≡ I (mod 5), where A^2 represents the matrix product A * A.
c) If A ≡ I (mod 5), then A^T ≡ I (mod 5), where A^T represents the transpose of matrix A.

For each statement, justify whether it is true or false based on the given definition of matrix congruence modulo 5. If the statement is true, provide a brief explanation. If the statement is false, provide a counterexample or a brief explanation why it does not hold.

Exercise 2. (Python) Define in Python the following functions:

- **read_square_matrix(n)**: it iteratively reads nxn elements from the user and return them into a numpy matrix of integer
- **is_congruent(m, mod)**: it implements the logic defined in exercise 1, with m being the matrix and mod the parametric modulo. It takes a matrix in input and the modulo value, and return True iff the matrix is congruent to the identity matrix modulo mod, False otherwise.

- Invoke the **is_congruent** function over the matrix m with mod=5, and printing its output.
- For each statement in Exercise 1, by exploiting the numpy operations for computing the transpose and the matrix product, and by properly invoking the **is_congruent** function:
    - If the statement is True, write a matrix A (and B if needed) returning True
    - If the statement is False, write a matrix A (and B if needed) returns False

Exercise 3. (C) Write a C program that implements a stack of integers using dynamic memory allocation. The stack uses an array that doubles in size when it is full, and halves its space when the occupancy falls below the 20%. The array doubling and halving ensure efficient memory usage as the stack grows and shrinks. Additionally, implement the following methods on top of the stack:

- **push**: Add an integer element to the top of the stack
- **pop**: Remove and return the element at the top of the stack
- **top**: Return the element at the top of the stack without removing it
- **is_empty**: Return whether the stack is empty or not (1 if empty, 0 if not empty)
- **size**: Return the number of elements currently in the stack

Within the main, create a stack and test the implementation of each method.