VG101 Lab Worksheet

Lab 1

Instructor: Dr. Yifei ZHU
TA: Hangrui CAO
TA: Qinhang WU
TA: Muchen XU

Code Quality Requirement

- Ensure proper indentation
- Make naming meaningful
- Include necessary comments
- Split the code over functions
- Test the code as much as you can

Objectives

This lab aims at making you familiar with basic MATLAB programming. Specific contents include:

1. Naïve number operation
2. Naïve string operation
3. Basic matrix manipulation
4. MATLAB Documentation searching
5. Simple application of MATLAB programming to solve a problem.

Exercises in this lab will **not** count towards the grade. However, **we expect you to try your best** to go through the procedure of analyzing problems (especially the *Basic Syntax* part) and **implementing** them in MATLAB.

Tips

In this lab exercise, relevant functions may include but not limit to:

- `randi()`
- `mod()`
- `lcm()` / `gcd()`
- `transpose()`
- `floor()` / `ceil()`
- `num2str()`
- `int64()`

Remember that in MATLAB, you can always type `help <FunctionName>` in the command line to view the explanation of the usage of that function, or type `doc <FunctionName>` to view the documentation of it.

Remember to **clear the variables in the workspace** and the output in the command window as well with the function `clearvars, clc;`.

Basic Syntax

Write code to implement the following operations:

Number Operation

1. Create a random number `x` between 1 and 10.
2. Create a 1x10 matrix `r` composed of ten uniformly distributed integers from (10-50).
3. Transform `x` into the string type.

String Combination

1. Define two string type `s1` and `s2` to represent 'Happy' 'VG101.'
2. Combine `s1` and `s2` to a string `s3`. (Hint: `strcat`)

Matrix Manipulation

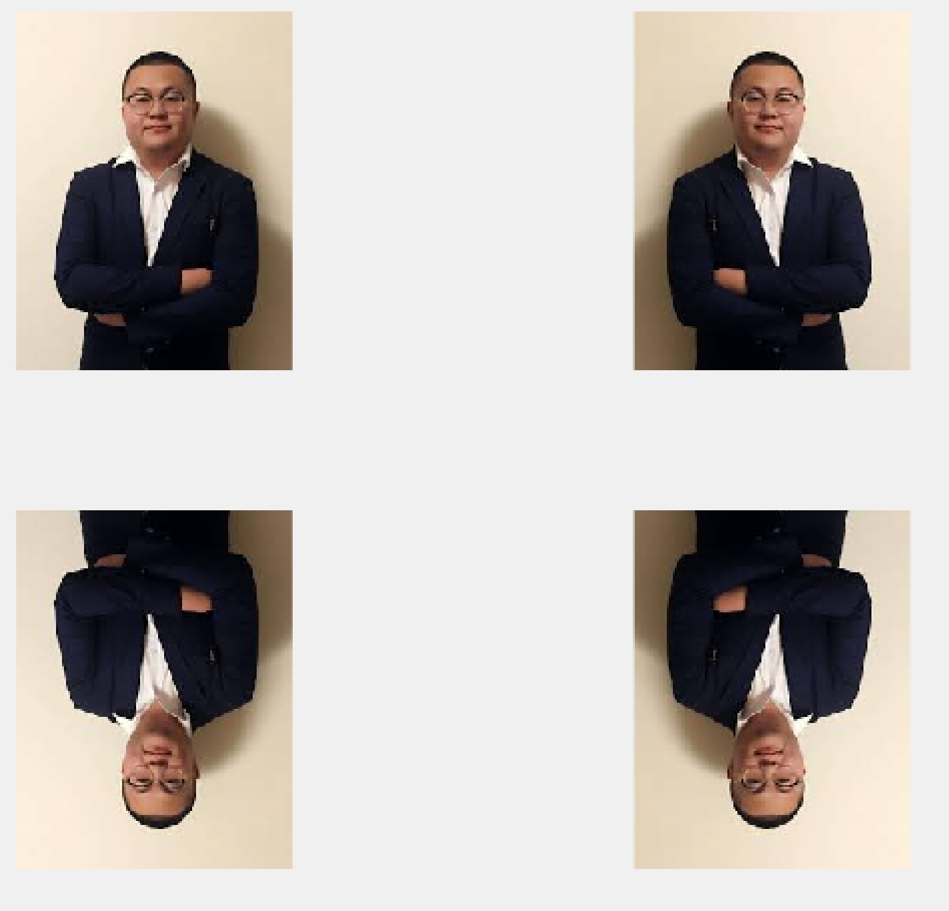
1. Initialize two matrices of size *2*5* with random integers between 1 and 5. Assign them to the variable `X` and `Y`.
2. Print the sum of the elements in `X`.
3. Concatenate `X` and `Y` to form a matrix of size...
   - *4*5* with `X` on the "above".
   - *2*10* with `X` on the "right".
4. Compute the sum and the **element-wise** product of two matrices.
5. Raise each element of `X` to the power specified by the corresponding element in `Y`。
6. Find the *least common multiple* of corresponding element in `X` and `Y`.
7. Compute product of X and the transpose of Y.
8. Find the remainder of elements in `X` divided by the corresponding elements in `Y`.
9. Replace the first 3 elements in the first line of `X` by the last 3 elements in the second line of `Y`.

Learn To Search for documentations

1. MATLAB is a world a matrix and it deals images with matrix too. In this exercise, we are going to flip one photo for 90, 180 and 270 degrees, and show the four photos in one plot.

Hints

1. Use`imread` to import one image/png/tif photo file to MATLAB, you can use one matrix `image_a` to store it.
2. Use functions of `fliplr`,`flipud` and `fliplr` to operate on matrix `image_a`;
3. Use `imshow` and `subplot` function to show the four photos in one Figure. Sample output is shown below:



2. Some interesting functions: Use documentation/help to search for functions: `vibes`, `teapotdemo`, `makevase`, `fifteen`, `earthmap` and `soma`.

Simple Algorithms - Linear Congruential Generator Cracker

Read the material and consider the problem. The last three problems may

Problem Description

One day Scarlet wants to generate a sequence of random integers, exactly, 7 random integers, for the use of making decision on 7 multiple-choice questions. She forgets the builtin-function `randi` and consults you to help her implement one.

As it happens, you learn a random number generating method called *Linear Congruential Generator*. In this method, numbers are generated by the formula

$$X_n = (aX_{n-1} + c) \mod m,$$

where a, c, m, X_0 are positive integers that you can determine, and mod stands for mathematical *modulo*. Parameters satisfy that $0 \leq a, c, X_0 < m$.

After writing the program, you just write down $X_1 \dots X_7$ for Scarlet's use, but forget the parameter like a, c, m and X_0. Now the problems come.

Problems

1. Given $X_0 = 114, a = 514, m = 1919, c = 810$, find the $X_1 \cdots X_7$.
2. Given $X_1 = 346, X_2 = 1042, a = 3, m = 1919$, find c.

3. Given $X_1 = 863, X_2 = 674, X_3 = 964, m = 2333$, find a and c.
4. Given $X_1 = 115666, X_2 = 123882, X_3 = 207144, X_4 = 117227, X_5 = 41605, X_6 = 34225, X_7 = 39639$, find a, m, c.
5. Can you find X_0 and X_8 for the problem 4?

Hints

1. MATLAB always provide the non-negative remainder by `mod()`.
2. $X_n = (aX_{n-1} + c) \mod m$ implies $c \equiv X_n - aX_{n-1} \pmod m$ and also $c = X_n - aX_{n-1} + km$, where k is an integer.
3. Try to take the transform $T_n = X_n - X_{n-1}$ to temporarily eliminate c by differing two equations.
4. MATLAB builtin-function `[g,u,v] = gcd(a,b)` will return the *Bezout coefficients* u and v, which satisfies $au + bv = g$ and g is the great common divisor of integers a, b. **Make good use of this equation!**
5. You can validate that whether $T_n^2 - T_{n-1}T_{n+1}$ is always the multiple of m.
6. For four random positive integers, the probability that no common factor will divide them all is 0.9829. That means, if you have four integers with a common divisor g, g is likely to equal to their greatest common divisor.
7. Algorithm is not nothing but coding. Involve and believe in your math skills!