

INTRODUCTION TO MATHEMATICAL SOFTWARE & PROGRAMMING

LAB 5 PROBLEMS (MATLAB)

QUESTION 1

- Write a MATLAB function `[q,r]=intDiv(a,b)` which returns the quotient and remainder of dividing two positive integers. For example `[q,r]=intDiv(15,6)` should return `q=2, r=3`.
- Recall Euclid's algorithm for finding the greatest common divisor of two positive integers. Write a MATLAB function called `d=Euclid(a,b)` which returns the greatest common divisor (gcd) of two positive integers. You should call the function `intDiv` from above inside this function. **HINT: use a while loop!**

QUESTION 2

- Write a MATLAB function called `[m,n]=matSize(X)` such that it takes a matrix `X` and returns its size: `m` (# of rows), `n` (# of cols). Obviously, you should come up with an algorithm of your own and you should NOT use MATLAB's `size` or `length` function.
HINT: use matrix row/column operations you learnt in LEC1. A MATLAB Boolean function `isempty` might be useful!
- Write out a main script in which you prompt the user to enter any matrix and then call your function `matSize` from above to determine the size of the user-input matrix. Make up a message to display the size: Your matrix size is 5x10. Also, use a while loop to ask the user, whether they want to repeat this process: Would you like to repeat? [Y/N]

QUESTION 3

Write a MATLAB function that takes a square matrix ($n \times n, n \geq 3$) and decomposes it into three matrices: upper-triangular + diagonal + lower-triangular. Your function should return these matrices as its output. Your function definition header should be: `[U,D,L]=splitMat(X)`, where `X` is the input square matrix and `U` is upper-triangular matrix, `D` is diagonal and `L` is lower-triangular. For example if

$$X = \begin{pmatrix} 1 & 3 & 7 \\ 5 & 2 & -1 \\ 9 & 8 & 4 \end{pmatrix}$$

Then `splitMat(X)` will return:

$$U = \begin{pmatrix} 0 & 3 & 7 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}, D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{pmatrix}, L = \begin{pmatrix} 0 & 0 & 0 \\ 5 & 0 & 0 \\ 9 & 8 & 0 \end{pmatrix}$$

You should devise a nested loop in your function for producing `U`, `D` and `L`.

You then make a separate m-file called `splitter.m` that prompts the user to enter a square matrix. Call your `splitMat(X)` function in order to decompose the user-input matrix. Use `sprintf` and `disp` to print the matrices U,D and L with messages like the following (assuming that X is as per the example above):

The Upper-triangular Matrix is:

```
0    3    7
0    0   -1
0    0    0
```

The Diagonal Matrix is:

```
1    0    0
0    2    0
0    0    4
```

The Lower-triangular Matrix is:

```
0    0    0
5    0    0
9    8    0
```

NOTE: the above example is just a model illustration. The output varies depending on the size of the input matrix.