

# INTRODUCTION TO MATHEMATICAL SOFTWARE & PROGRAMMING

## LAB 4 PROBLEMS (MATLAB)

*Note: For each question make a script file in MATLAB.*

### QUESTION 1

- a) Define a vector  $X$  with entries ranging from 10 to 100 steps 10 ( $X=10:10:100$ )
- b) Make another vector  $Y$  such that the entries are squares of entries of  $X$ .
- c) Define a new variable called `ssX` and set it equal the *square root of the sum of squares of  $X$* .
- d) Use an appropriate message to print the value of `ssX` in 5 decimal places.

### QUESTION 2

- a) Create a vector  $V$  with 100 random integers between 0 and 1000.
- b) Using a `for` loop separate the even and odd integers of vector  $V$  and put them in two vectors  $U$  and  $W$ . *Hint: you can use `mod(a,b)` function in MATLAB.*
- c) Use two appropriate messages to print out the total of even and odd numbers randomly generated between 0 and 1000. For example: There are ... even numbers

### QUESTION 3

- a) Using input command ask the user to enter a positive integer,  $n > 5$
- b) Create an  $n$  by  $n$  matrix  $M$ , with random integer entries between 1000 and 2000.
- c) Use a **nested for loop** and MATLAB's built-in function `isprime()` to count the prime numbers generated randomly in your matrix  $M$ . Create a variable `primeCount` that stores the count of prime numbers. You should produce a message "there are ... prime numbers"
- d) Your script should produce a message "**there are no prime numbers**" in case that no prime numbers are generated.

### QUESTION 4

- a) Use a **nested for loop** to display the following matrix in MATLAB:

$$A(i,j) = \begin{cases} \frac{i}{i+j} & \text{if } i < j \\ ij & \text{if } i = j \\ \sqrt{i-j} & \text{if } i > j \end{cases} \text{ where } 1 \leq i \leq m \text{ and } 1 \leq j \leq n.$$

- b) You should prompt the user to input the size of the matrix ( $m,n$ ). You should ask the user to first enter the number of rows and then in a separate input message ask for the number of columns:  
"Enter the number of rows:"  
"Enter the number of columns:"
- c) Using a **while loop** ask the user if they want to try again?

## QUESTION 5

- a) Consider the following set of points on the plane:

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
x-coordinate	0	-1	1	3	-5
y-coordinate	-1	2	0	-3	4

- b) Write out a MATLAB code that creates a 5x5 proximity matrix for the points given above. The proximity matrix gives the Euclidean distance between each pair of points. Notice that the proximity matrix is symmetric and all the diagonal entries are 0. Recall that the Euclidean distance between two points  $P: (x_1, y_1)$  and  $Q: (x_2, y_2)$  is calculated via

$$\text{dist}(P, Q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

See the diagram below:

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
<b>A</b>	0=dist(A,A)	dist(A,B)	dist(A,C)	dist(A,D)	dist(A,E)
<b>B</b>	dist(B,A)	0	...	...	...
<b>C</b>	dist(C,A)	...	0	...	...
<b>D</b>	dist(D,A)	...	...	0	...
<b>E</b>	dist(E,A)	...	...	...	0

You can start your code by setting up the following row vectors:

```
xVals = [0,-1,1,3,-5];
```

```
yVals = [-1,2,0,-3,4];
```

## QUESTION 6

Write a MATLAB script that prompts the user to input a positive integer  $n > 2$  and then create an  $n$  by  $n$  matrix like below. For example if  $n=4$ , your script should create the following square matrix:

$$\begin{bmatrix} 4 & 5 & 0 & 0 \\ 3 & 4 & 5 & 0 \\ 0 & 3 & 4 & 5 \\ 0 & 0 & 3 & 4 \end{bmatrix}$$

You should use a nested loop. Also, using a **while loop**, ask the user if they want to try again?

## QUESTION 7

Write a MATLAB script that takes a positive integer from the user,  $n > 1$  and then computes  $n!$  You should use a **while loop** for computing the factorial.