

MATH 2221 A Mathematics Laboratory II

Lab Assignment 7

Name: _____

Student ID.: _____

In this assignment, you are asked to run **MATLAB** demos to see **MATLAB** at work. The color version of this assignment can be found in your own **H:** drive.

Instructions

1. Start **MATLAB**, until you see a window with the **MATLAB** prompt "**>>**". This window is called the **Command Window**.
2. After you started have **MATLAB**, you will automatically be in the directory **H:**. Please enter "**diary on**" after the **MATLAB** prompt **>>** only once to record all your work in **H:\diary**. No marks will be given if no diary is found.
3. Enter "demo" after the prompt **>>**. You will see a new window with many things to play with. This is the **Demo Window**.
4. In the **Demo Window**, try to locate figures or problems similar to those in the exercises below. Then locate the commands that generate these figures or problems. Try them in the **Command Window**. Just enter (or cut and paste) the commands after **>>** to see what happens.
5. You should write your results on the lab sheet provided, and save the figures in the **H:** drive, in your personal drive.
6. Please read and sign the following declaration before handing in your assignment. Otherwise, no marks will be given.

I declare that the assignment here submitted is original except for source material explicitly acknowledged. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained on the website <http://www.cuhk.edu.hk/policy/academichonesty/>

Signature

Date

1 (20 marks)		3 (20 marks)	
2 (15 marks)		4 (45 marks)	
		Total	/100 pts

Please read the following carefully:

General Guidelines for Lab Assignment Submission.

- Please sign and date the statement of Academic Honesty.
- Please go to the class and lab indicated by your registered course code via the CUSIS system. If you go to a different lab than the one you are registered for, you will not receive credit for the assignment even if you completed it.
- Write your COMPLETE name and student ID number legibly on the cover sheet (otherwise we will not take any responsibility for your lab). Please write your answers using a black or blue pen, NOT any other color or a pencil.
- Write your solutions on a double-sided printout of this pdf file. Try to fit your answers inside the available space.
- The use of computers/cellular phones/graphing calculators/iPads will NOT be permitted during tests and lab assignments. Please do not use our lab computer to recharge your cellular phone battery.
- In order to make it fair for all students, during the labs and tests, if you touch/press any icons on your cellular phone, our TA will check your phone to determine whether or not you are exchanging messages with another student. If you are found cheating (in the tests or in the lab or on homework assignments), you will automatically get an F grade in this course and your act will be reported to the Department for necessary disciplinary actions.

Exercises

1. (20 marks) Write a script named `lab_7_q1.m` to find the size m, n of an $m \times n$ matrix A and compute the determinant if it is a square matrix.

Instructions:

- (a) Firstly, input the matrix by showing “Input a matrix A: ”;
- (b) Find the size of A using a suitable built-in function.
- (c) If A is a square matrix, display the result with

```
“This is an n-by-n square matrix. ”;
```
- (d) Compute the determinant of square matrix by a built-in function and display the result with “The determinant of this matrix is d”;
- (e) If A is not a square matrix, just display “This is an m-by-n matrix. ” without computing the determinant;
- (f) Be careful about the space between words.

Save your script `lab_7_q1.m` in the directory H: .

Sample Output:

```
>> lab_7_q1
Input a matrix A: [1,2;2,3]
This is an 2-by-2 square matrix.
The determinant of this matrix is -1
```

Solution of lab_7_q1.m

```
% q1

clear;
A = input('Input a matrix A:');
s = size(A);
if s(1) == s(2)
    d = det(A);
    disp(['This is an ' num2str(s(1)) '-by-' num2str(s(2)) ...
        ' square matrix.']);
    disp(['The determinant of this matrix is ' num2str(d)]);
else
    disp(['This is an ' num2str(s(1)) '-by-' num2str(s(2)) ...
        ' matrix.'])
end
```

2. (15 marks) Let

$$C_n(x) = \sum_{k=0}^n \frac{(-1)^k x^{2k}}{(2k)!}, \text{ for all } x$$

- (a) Write a script named `lab_7_q2.m` to compute $C_n(x)$ for any given natural number n and real x .

Instructions:

- i. Input n by showing “Input n: ”;
- ii. Input x by showing “Input x: ”;
- iii. Compute and display “Cn = ”.

Save your script `lab_7_q2.m` in the directory H: .

Hint: You may use the built-in function `factorial(z)` to compute $(z)!$.

Sample Output

```
>> lab_7_q2
Input n: 2
Input x: 1
Cn = 0.54167
```

Solution of lab_7_q2.m

```
% q2

clear

n = input('Input n: ');
x = input('Input x: ');
Cn = 0;
for k = 0 : n
    Cn = Cn + (-1)^k * x^(2*k) / factorial(2*k);
end
disp(['Cn = ' num2str(Cn)]);
```

- (b) i. Compute $C_{100}(\pi)$ and compare with $\cos(\pi)$. Do these two values agree?
- ii. Try some other x with large enough n , do you think $C_n(x)$ converges for all real x as $n \rightarrow \infty$? If so, what should be the limit function? (just write down the conclusion)

```
i) -1, Yes, two values agree;
ii) Yes. the limit function is cos(x)
```

3. (20 marks) Recall the built-in function `rand` generates uniformly distributed random number over $[0, 1]$. One would expect the number of random numbers which are less than 0.7 to get closer and closer to 70% of the total number as more and more random numbers are generated.

Generate a few thousand random numbers, say 5000 and use if loop to count how many are less than 0.7. If this is repeated a few times with a new set of random numbers each time, one should never get the same answer each time because the numbers are random.

Instructions:

- Input the total number of random numbers by showing "N: ";
- $a = 0$ (a is the number of random numbers larger than or equal to 0.7) and $b = 0$ (b is the number of random numbers less than 0.7);
- Repeat N times as the following:
Replace r by `rand` (generate a new random number in each loop),
if r is larger than or equal to 0.7, replace a by $a + 1$, else, replace b by $b + 1$;
- Display b by showing "The number of random numbers less than 0.7 is ".

Save your script `lab_7_q3.m` in the directory H: .

Sample Output

```
>> lab_7_q3
N: 1000
The number of random numbers less than 0.7 is 724
```

Solution of lab_7_q3.m

```
% q3
clear;

N_iter = input('N: ');
a = 0;
b = 0;
for n = 1 : N_iter
    r = rand;
    if r >= 0.7
        a = a + 1;
    else
        b = b + 1;
    end
end
disp(['The number of random numbers less than 0.7 is '...
    num2str(b)])
```

4. (45 marks)

- (a) (15 marks) The greatest common divisor (g.c.d.) of two integers a and b is the largest positive integer that divides both a and b without leaving a remainder. Write a script named `lab_7_q4a.m` to calculate the g.c.d. of two integers a and b . The program first prompts the user to input the integer a by showing “a = ”, then prompts the user to input the integer b by showing “b = ”. Do **NOT** use the built-in function `gcd(a,b)` or `lcm(a,b)` to complete this question.

Save your script `lab_7_q4a.m` in the directory H: .

Hint: You may use the Euclidean algorithm, which reads as follows:

- if $a < b$ then swap a and b ;
- $\text{gcd} := b$;
- while remainder of a divided by $b \neq 0$, do
 - $\text{gcd} :=$ remainder of a divided by b
 - $a := b$;
 - $b := \text{gcd}$.

Run the program by typing `lab_7_q4a` in command window. Find out the g.c.d. of two integer 2574 and 253. Write down the result you get.

```
the g.c.d. of two integer 2574 and 253 is  
  
gcd = 11
```

Solution for lab_7_q4a.m

```
% Solving The greatest common divisor (g.c.d.)  
% of two integers a and b  
close all; clear all;  
  
a = input('a = ');  
b = input('b = ');  
if (a < b)  
    c = a;  
    a = b;  
    b = c;  
end  
c = b;  
while (mod(a, b) ~= 0)  
    c = mod(a, b);  
    a = b;  
    b = c;  
end  
  
gcd = c
```

- (b) (15 marks) Use g.c.d. to compute the lowest common multiple (l.c.m.). The l.c.m. is the lowest positive integer that is divisible by both a and b . Write a script named `lab_7_q4b.m` based on `lab_7_q4a.m` and save it. Then compute the l.c.m. of two integer 2574 and 253 and write down the result. Do **NOT** use the built-in function `gcd(a,b)` or `lcm(a,b)` to complete this question.

```
the l.c.m. of two integer 2574 and 253 is

lcm = 59202
```

Solution for `lab_7_q4b.m`

```
% Solving the lowest common multiple (l.c.m.)
% of two integers a and b

close all; clear all;

a = input('a = ');
b = input('b = ');
d = a*b;
if (a < b)
    c = a;
    a = b;
    b = c;
end
c = b;
while (mod(a, b) ~= 0)
    c = mod(a, b);
    a = b;
    b = c;
end

lcm = d/c
```

- (c) (15 marks) The greatest common divisor (g.c.d.) of n integers k_1, k_2, \dots, k_n is the largest positive integer that divides k_1, k_2, \dots, k_n without leaving a remainder. Write a script named `lab_7_q4c.m` to calculate the g.c.d. of n integers k_1, k_2, \dots, k_n . The program first prompts the user to input an integer sequence $[k_1, k_2, \dots, k_n]$ by showing

```
Input all the numbers in array k = [k1 k2 ... kn],
k =
```

The program then shows the g.c.d.. Do **NOT** use the built-in function `gcd(a,b)` to complete this question. You may use the script in 4(a) and extend it to finish this question.

Save your scrip `lab_7_q4c.m`.

The following shows a sample output.

Sample Output:

```
Input all the numbers in array k = [k1 k2 ... kn]
k = [20 12 24]

gcd =

    4
```

Run the program by typing `lab_7_q4c` in the command window. Find out the g.c.d. of three integers 207, 161 and 253. Write down the result you get.

```
gcd =

    23
```

Solution of lab_7_q4c.m

```
% Solving the greatest common divisor (g.c.d.)
%of n integers [k_1, k_2, ... k_n]

close all; clear all;

disp('Input all the numbers in array k = [k1 k2, ... kn]');
k = input('k = ');
c = k(1); i = 1;
while i ~= size(k,2)
    i = i+1;
    a = c;
    b = k(i);
    if (a < b)
        c = a;
        a = b;
        b = c;
    end
    c = b;
    while (mod(a,b) ~=0)
        c = mod(a,b);
        a = b;
        b = c;
    end
end
gcd = c
```