Discrete Time Signals and Systems

Elec 342

Section MK-X

Lab Experiment #1

Andre Hei Wang Law

4017 5600

TA: Ahmed Salameh

TA Email: ahmed.salameh@mail.concordia.ca

Performed on January 23, 2023

Due on February 6, 2023

“I certify that this submission is my original work and meets the Faculty’s Expectations of Originality.”

1) Objectives

The primary objective of the first experiment of the course Elec 342 is to present the students with an introduction to MATLAB. More precisely, we will go over operators, arrays, matrices as well as plots by working on questions that require the student to write scripts. This introductory experiment will build a foundation of using MATLAB which will greatly aid in following labs.

2) Theory

Operator:

-Assignment operator: =

-Division operator: /

-Power operator: ^

-Multiplication operator: *

-Addition operator: +

-Subtraction operator: -

-Colon operator: “colon_array_example = [1 : 10]” (result: 10 elements from 1 to 10)

-Scalar and Vectors operation: “A = A*2” and “A = A/2”

-Element-by-element multi/div/power: .*   ./   .*   respectively

One Dimensional Arrays (i.e. a vector):

-“A = [1, 2, 3]” is the same as “A = [1 2 3]”, this is a row vector of 1x3 array

-“B = [1; 2; 3]” is a column vector of 3x1 array

-The transpose operator is ‘

Matrices:

-“A = [1 2 3; 4 5 6]”, this is a 2x3 array

-In array multiplications, the number of columns in the first array must be equal to the number of rows in the second array. (i.e. 4x2 and 2x10 will yield 4x10 array)

Complex Numbers:

-“A = real_part + j*imaginary_part”

-We can use either “i” or “j”

Plotting:

-“X = [0, 1, 2, 3]” and “Y = 3*X”

-“plot(X, Y)”, a graphical plot

-“stem(X, Y)”, a variation plot that doesn’t connect the data points

-Label and title can be added with “xlabel(‘x-axis’)”, “ylabel(‘y-axis’)” and “title(‘graph’)”

3) Tasks, Results and Discussion

Task 1:

Prompt in the command window

```
>> A = 5
```

Result 1:

```
A =

        5

>>
```

Discussion 1:

The first task is for the student to analyse the format display result and the various windows making up the MATLAB application. Mainly, there is the command window, the workspace window, the command history as well as the current folder. In this case, the purpose was to observe the command window.

Task 2:

```
>> B = 5;
```

Result 2:

```
>>
```

Discussion 2:

Similarly in task 1, we assigned a value to a variable. However, because of the semicolon, the result won’t be displayed in the command window.

Task 3 and Results 3:

```
>> length = 5

length =

            5

>> width = 4

width =

            4

>> area = lenght * width

area =

            20

>> powers_of_two = 2 ^ 0

powers_of_two =

            1

>> powers_of_two = 2 ^ 1

powers_of_two =

            2

>> powers_of_two = 2 ^ 2

powers_of_two =

            4
```

```
>> powers_of_two = 2 ^ 3

powers_of_two =

        8

>> powers_of_two = 2 ^ 4

powers_of_two =

        16

>> divide_by_zero = 1 / 0

divide_by_zero =

        Inf
```

Discussion 3:

The purpose of the third task was to observe how MATLAB treats all variables as arrays such that they are stored in a 1x1 array. Then, students also performed simple arithmetic operations in which its symbols are as detailed in the Theory section of this report.

Task 4:

```
>> ans = 3 + 4 ^ 2 / 8
```

Result 4:

```
ans =

        5
```

Discussion 4:

The observation to be made here is that MATLAB respect the order of operations such that powers comes before Multiplication and Division in which also comes before Additions and Subtractions. It should be noted that parenthesis acts per normal such that they force precedence over the above-mentioned rule.

Task 5:

```
>> my_row_vector = [ 1, 2, 3 ]
```

Result 5:

```
my_row_vector =

        1       2       3
```

Discussion 5:

This section was a guide of vectors such that the above task represents a row vector. It should also be noted that the input [1,2,3] is the same as [1 2 3].

Task 6:

```
>> my_row_vector = [ 1; 2; 3 ]
```

Result 6:

```
my_row_vector =

        1

        2
        3
```

Discussion 6:

Similarly, to task 5, we worked on vectors. However, with a semicolon, the vector is a 3x1 array rather than a 1x3 array. An additional information to note is that in order to get a transpose of a vector, we simply have to put the transpose operator ‘. This means that a column vector can transpose into a row vector and vice versa.

Task 7:

```
>> colon_array_example = [ 1 : 10 ]
```

Result 7:

```
colon_array_example =

        1       2       3       4       5       6       7       8       9       10
```

Discussion 7:

From the results of the input [1 : 10] which yields the output as show above, it demonstrates that the colon operator is used to create an array within the mentioned range. In this case, it was from 1 to 10. In addition, there is also an alternative syntax of [min_range: increment: max_range] which dictates the amount the array increments to rather than the default increment of 1.

Task 8 and Results 8:

```
>> A = [ 1 2 3 ]

A =
        1       2       3

>> B = [ 4 5 6 ]

B =

        4       5       6

>> C = A + B

C =

        5       7       9
```

Discussion 8:

MATLAB follows the same logic as arithmetic additions when addition vectors. It processes the operation element-by-element. This works for addition, subtraction, multiplication, division and power. However, it should be noted that to perform element-by-element multiplications, division and power, its operators include a period. They look as follows:

```
.*

./

.^
```

Task 9:

```
>> mick = [ 1 2 3 ; 4 5 6 ]

mick =

        1       2       3

        4       5       6

>> keith = [ 7 8 ; 9 10 ; 11 12 ]
```

```
keith =

        7       8

        9       10

        11      12

>> stones = mick * keith
```

Result 9:

```
stones =

        58      64

        139     154
```

Discussion 9:

For this task, it should be discussed that to perform array multiplications, MATLAB follows a rule. Mainly, the size of the two array have to be of the (m x n) and (n x k) size which yield a final array of the size (m x k). This rule respects linear algebra and would not work otherwise.

Task 10:

```
>> help plot
```

Result 10:

```
plot Linear plot.
```

plot(X,Y) plots vector Y versus vector X. If X or Y is a matrix, then the vector is plotted versus the rows or columns of the matrix, whichever line up. If X is a scalar and Y is a vector, disconnected line objects are created and plotted as discrete points vertically at X.

plot(Y) plots the columns of Y versus their index.

If Y is complex, plot(Y) is equivalent to plot(real(Y),imag(Y)).

In all other uses of plot, the imaginary part is ignored.

Various line types, plot symbols and colors may be obtained with plot(X,Y,S) where S is a character string made from one element from any or all the following 3 columns:

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	Dashdot

Discussion 10:

For the last task, students try many ‘help’ command (i.e. *help delete*, *help subplot*, etc.). It should ne noted that the help command results can further be looked into directly from the official MATLAB website. This resource also provides examples and detailed explanation of the selected functions

4) Questions

Question 1:

Left Plot:

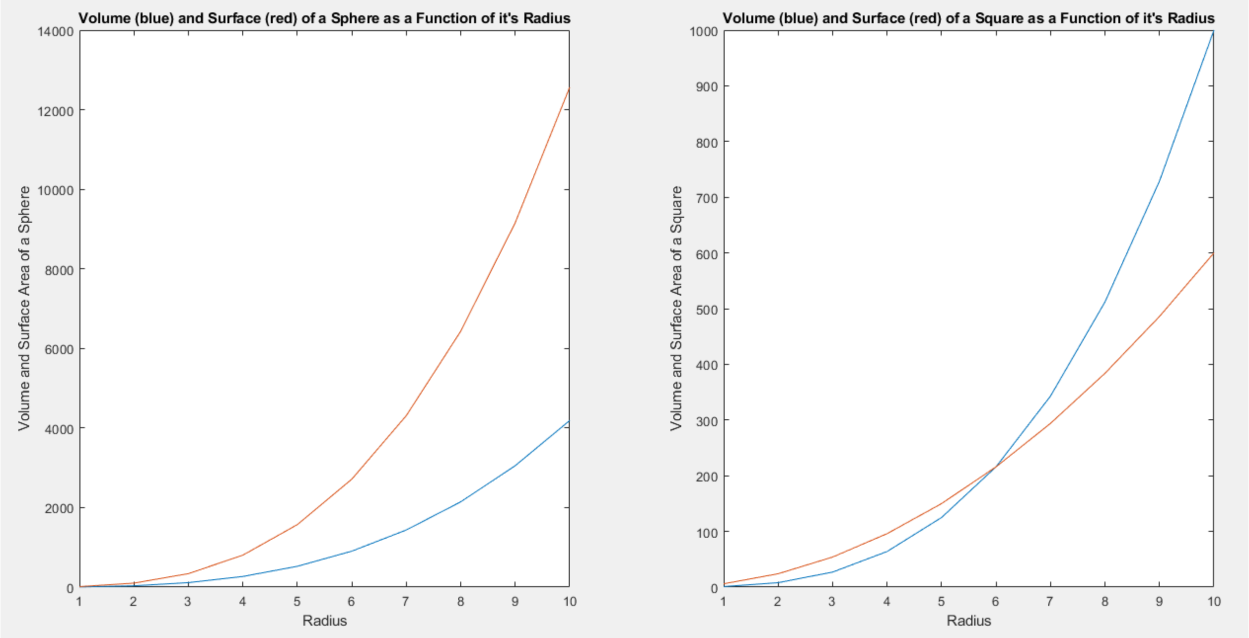
Blue: Volume of a sphere as a function of it’s radius from 1 to 10

Red: Surface area of a sphere as a function of the radius from 1 to 10

Right Plot:

Blue: Volume of a square as a function of it’s radius from 1 to 10

Red: Surface area of a square as a function of the radius from 1 to 10



The above plot is the result after running the code in the appendix question 1. It mainly uses the subplot and hold command. The former is responsible for the amount of plot and its position (left or right in this case), while the latter is responsible for overlaying the results over each and other rather than having the second drawing replace the first (this is represented through the red and blue lines). The raw results of running the code are as follows:

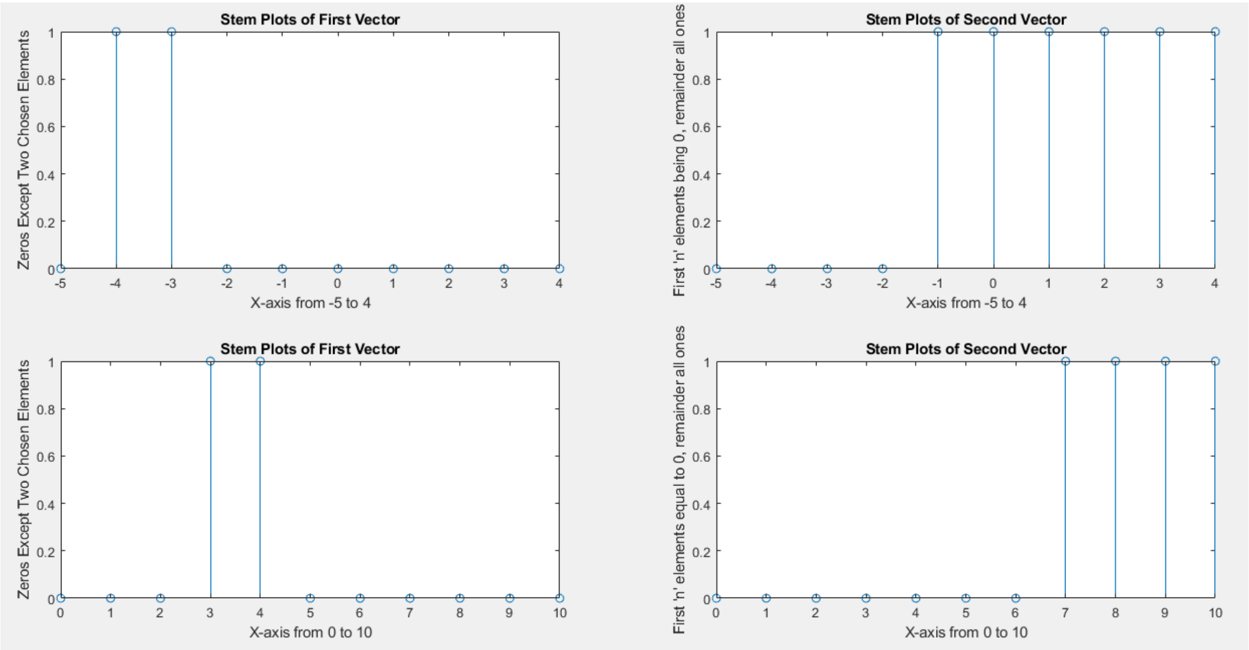
```
============================Start of Section============================

>> lab1_q1


r =


     1     2     3     4     5     6     7     8     9    10




vol =


   1.0e+03 *


  Columns 1 through 6


   0.004188790204786   0.033510321638291   0.113097335529233   0.268082573106329
0.523598775598299   0.904778684233860


  Columns 7 through 10


   1.436755040241732   2.144660584850632   3.053628059289279   4.188790204786391
```

```
sur =

   1.0e+04 *

  Columns 1 through 6

   0.001256637061436   0.010053096491487   0.033929200658770   0.080424771931899
0.157079632679490   0.271433605270158

  Columns 7 through 10

   0.431026512072520   0.643398175455190   0.916088417786784   1.256637061435917


r =

     1     2     3     4     5     6     7     8     9    10


vol2 =

           1           8          27          64         125         216         343         512         729        1000
```

```
sur2 =


     6    24    54    96   150   216   294   384   486   600
```

==============================End of Section==============================

Question 2:



By yet using again the subplot command, it was possible to create four subplots. The top left represents an array of all zeros except two elements. The two chosen elements to be ones are the 2nd element and 3rd element. This array is from -5 to +4. Similarly, the top right consist of first n elements being zeros, while the remaining elements being ones. In this case, it was arbitrary chosen that the first 4 elements are to be zeros. This also respect the array which is from -5 to 4. The bottom stem plots reflect its tom counterpart, except with different chosen elements and with the z values from 0 to 9, rather than from -5 to 4.

The command used to change the value of an element into a zero is zeros(n) in which ‘n’ represents the position in an array that needs to be changed. Conversely, to change a value into a one, the command is ones(n). To target all values and changing them into zeros, it is possible to assign the range from where to where it should be changed, i.e. vec1 = zeros(1, 10) which cam be found in the appendix.

(NOTICE: It has come to my attention that I seem to have mixed the order of the subplot. The top right should switch with the bottom left. This fix can be done by changing its corresponding subplot commands. In the case of the top right panel, rather than subplot(2, 2, 2), it should be subplot(2, 2, 3), and vice versa for the original bottom left panel.)

The raw results of running this program are as follows: (notice how the values of the vector changes from ones to zeros and from zeros to ones)

```
=============================Start of Section=============================
>> lab1_q2

x =

    -5    -4    -3    -2    -1     0     1     2     3     4



vec1 =

     0     0     0     0     0     0     0     0     0     0
```

```
vec1 =

     0     1     0     0     0     0     0     0     0     0


vec1 =

     0     1     1     0     0     0     0     0     0     0


x =

    -5    -4    -3    -2    -1     0     1     2     3     4


vec1 =

     1     1     1     1     1     1     1     1     1     1


n =
     4


vec1 =

     0     0     0     0     1     1     1     1     1     1


x =

     0     1     2     3     4     5     6     7     8     9    10


vec1 =

     0     0     0     0     0     0     0     0     0     0     0


vec1 =

     0     0     0     1     0     0     0     0     0     0     0


vec1 =
     0     0     0     1     1     0     0     0     0     0     0


x =

     0     1     2     3     4     5     6     7     8     9    10


vec1 =

     1     1     1     1     1     1     1     1     1     1     1


n =

     7


vec1 =

     0     0     0     0     0     0     0     1     1     1     1
```

=============================End of Section============================

Question 3:

Part i)

Given that a signal X[n] is said to be periodic (with period T) if X[n + T] = x[n], this section was a test to verify the effect of how floating points numbers are stored within a system. Given that the above rule is true, it should mathematically true that x[1] - x[1+1024], yielding 0. However, based on the results of our tests, the result of the subtraction y1 isn’t exactly equal to zero, rather it is 1.878800427557170e-05. This result tells us that the way a system stored a floating-point number isn’t exact, but very close as the above number is can be rounded to zero (y1 is a very small number very close to zero). The raw result of this section is as follows:

```
============================Start of Section============================

n1 =


     1




n2 =


        1025




y1 =


     1.878800427557170e-05
```

```
===============================End of Section=============================
```

Part ii)

In part 2, the constant ‘pi’ was changed to 3.14 which is a rounded down version of ‘pi’. The results obtained by the subtraction is 1.369587310406928e-05. Compared to part 1, the value of y2 is slightly smaller than the value of y1 which was equal to 1.878800427557170e-05. The raw result of this section is as follows:

```
==============================Start of Section==============================
n1 =


    1



n2 =


       1025



y2 =


    1.369587310406928e-05
===============================End of Section=============================
```

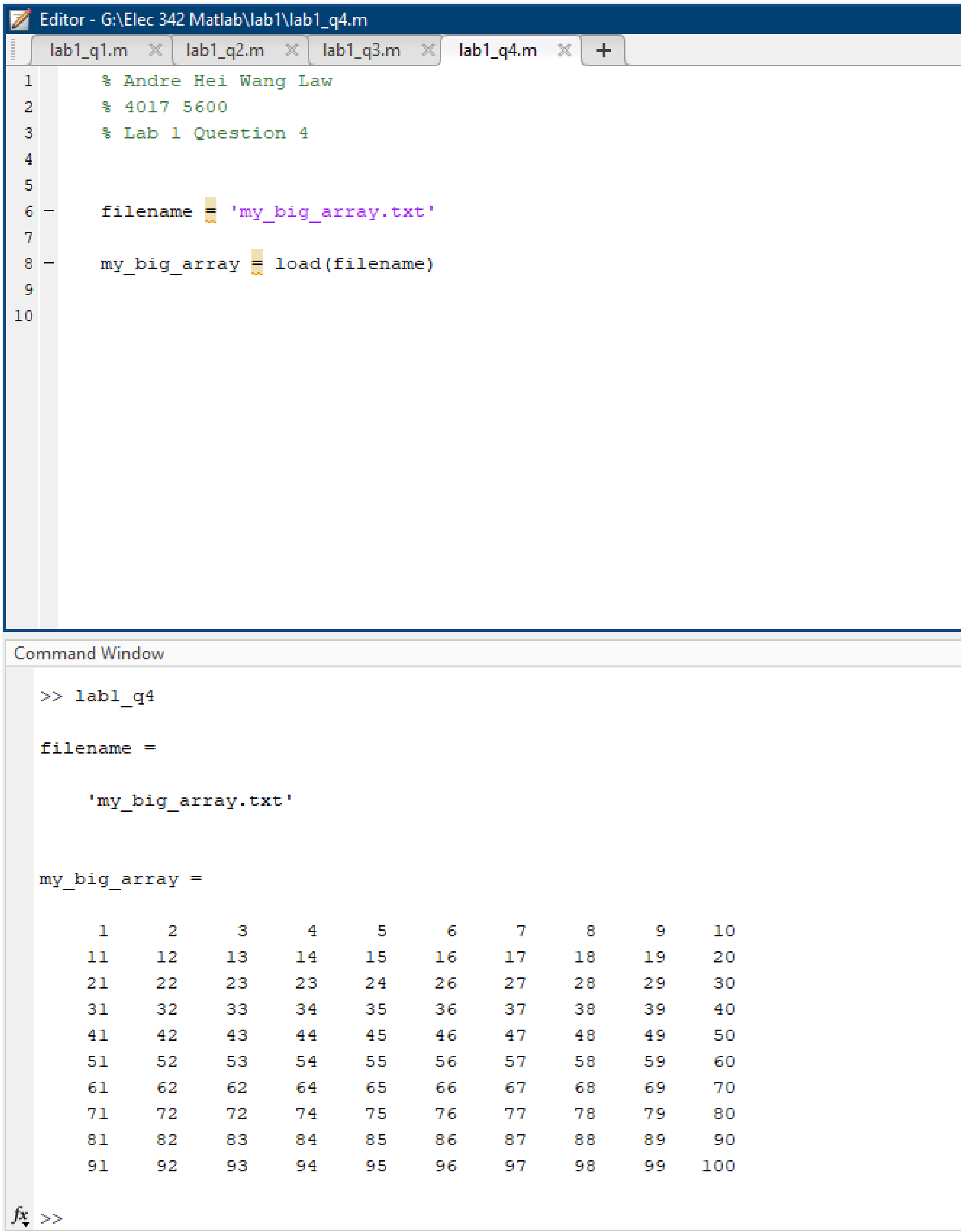
<u>Part iii)</u>

For the final part, it was asked to determine whether x1 and x2 where equal. We obtain our answer by subtracting xiii1 by xiii2 (look into the appendix) and we obtain a value of y3 equal to -4.440892098500626e-16 which is very small number close to zero. This value is even closer to zero than the values obtain in part i) and ii) since y3 is in the e-16 decimals. The raw result of this section is as follows:

```
=============================Start of Section=============================

niii =


     1




y3 =


    -4.440892098500626e-16

==============================End of Section============================
```

Question 4:

The last question's code is a very direct. Simply, use the load command to load variables from a file into the workspace, in this case it was an ascii text file of numbers ranging from 1 to 100. The directory the text file is in the same directory as lab1_q4.m, such that accessing it doesn't requires exiting and entering another path. This can be seen by the way the text file was called as is and put into the variable 'filename'. The raw results of this question can be seen as follows:



```
% Andre Hei Wang Law
% 4017 5600
% Lab 1 Question 4


filename = 'my_big_array.txt'

my_big_array = load(filename)


```

```
>> lab1_q4

filename =

    'my_big_array.txt'


my_big_array =

     1     2     3     4     5     6     7     8     9    10
    11    12    13    14    15    16    17    18    19    20
    21    22    23    23    24    26    27    28    29    30
    31    32    33    34    35    36    37    38    39    40
    41    42    43    44    45    46    47    48    49    50
    51    52    53    54    55    56    57    58    59    60
    61    62    62    64    65    66    67    68    69    70
    71    72    72    74    75    76    77    78    79    80
    81    82    83    84    85    86    87    88    89    90
    91    92    93    94    95    96    97    98    99   100

>>
```

5) Conclusions

In conclusion, the first experiment of Elec 342 allowed the students to get an introductory lesson about MATLAB and its many functionalities. By understanding the usage of the command window as well as the variables stored inside the workspace, students were able to grasp the structure and steps in which to take in order to assign values, create arrays and matrices. Then through answering four questions, students worked on command such as “subplot” and “hold”, “zeroes” and “ones”, “load”, etc. These questions also allowed students to familiarize themselves with the different operators, arrays, matrices and reviewed many other concepts. Overall, this first experiment introduced MATLAB programming to students in which will be useful for this course as well as other courses that utilises the MATLAB tool.

6) Appendix

Below is my source code written for question 1 to 4.

Question 1:

```
=============================Start of Section==============================
% Andre Hei Wang Law
% 4017 5600
% Lab 1 Question 1
% lab1_q1.m

clear
subplot(1, 2, 1)

r = 1 : 10                  % radius from 1 to 10
vol = (4/3).*pi.*(r.^3)     % volume of sphere
plot(r, vol)                % plot of volume vs radius

hold on
sur = 4.*pi.*(r.^3)         % surface area of a sphere
plot(r, sur)                % plot of surface vs radius
hold off
title("Volume (blue) and Surface (red) of a Sphere as a Function of it's Radius")
xlabel("Radius")
```

```
ylabel("Volume and Surface Area of a Sphere")


subplot(1, 2, 2)

r = 1 : 10                   % radius from 1 to 10

vol2 = r.^3                   % volume of square

plot(r, vol2)                 % plot of volume vs radius


hold on

sur2 = 6.*(r.^2)              % surface area of a sphere

plot(r, sur2)                 % plot of surface vs radius

hold off

title("Volume (blue) and Surface (red) of a Square as a Function of it's Radius")

xlabel("Radius")

ylabel("Volume and Surface Area of a Square")

==============================End of Section============================
```

Question 2:

=============================Start of Section=============================

```
% Andre Hei Wang Law
% 4017 5600
% Lab 1 Question 2
% lab1_q2.m
clear
subplot(2, 2, 1)
x = -5 : 4
vec1 = zeros(1,10)    %size of array
vec1(2) = 1           % 2nd element will be 1
vec1(3) = 1           % 3rd element will be 1
stem(x, vec1)
title("Stem Plots of First Vector")
xlabel("X-axis from -5 to 4")
ylabel("Zeros Except Two Chosen Elements")


subplot(2, 2, 2)
x = -5 : 4
vec1 = ones(1,10)     %size of array
n = 4                 % arbitrary number
vec1(1:n) = 0
stem(x, vec1)
```

```
title("Stem Plots of Second Vector")
xlabel("X-axis from -5 to 4")
ylabel("First 'n' elements being 0, remainder all ones")

subplot(2, 2, 3)
x = 0 : 10
vec1 = zeros(1,11)      % size of array
vec1(4) = 1             % 4nd element will be 1
vec1(5) = 1             % 5rd element will be 1
stem(x, vec1)
title("Stem Plots of First Vector")
xlabel("X-axis from 0 to 10")
ylabel("Zeros Except Two Chosen Elements")

subplot(2, 2, 4)
x = 0 : 10
vec1 = ones(1,11)       % size of array
n = 7                   % arbitrary number
vec1(1:n) = 0
stem(x, vec1)
title("Stem Plots of Second Vector")
xlabel("X-axis from 0 to 10")
ylabel("First 'n' elements equal to 0, remainder all ones")
```

Question 3:

=============================Start of Section=============================

```
% Andre Hei Wang Law
% 4017 5600
% Lab 1 Question 3
% lab1_q3.m

clear
% i) X[n] is periodic if X[n+T]=x[n]
n1 = 1
xi1 = cos((2.*pi)/n1);                  % n = 1

n2 = 1+1024
xi2 = cos((2.*pi)/n2);                % n = 1 + 1024

format long
y1 = xi1-xi2;
display(y1);

% ii) Same as (i), bu we use 3.14 rather than pi
n1 = 1
xii1 = cos((2.*3.14)/n1);                  % n = 1
```

```
n2 = 1+1024

xii2 = cos((2.*3.14)/n2);              % n = 1 + 1024



format long

y2 = xii1-xii2;

display(y2);



% iii) Determine if two signals are Equal

niii = 1

xiii1 = cos(pi./4.*niii+pi./3);



xiii2 = cos(9.*pi./4.*niii+pi./3);



format long

y3 = xiii1-xiii2;

display(y3);
```

=============================End of Section============================

Question 4: (part 1, MATLAB file)

=============================Start of Section=============================

```
% Andre Hei Wang Law

% 4017 5600

% Lab 1 Question 4

% lab1_q4.m


clear

filename = 'my_big_array.txt'


my_big_array = load(filename)
```

==============================End of Section============================

Question 4: (part 2, Text file, my_big_array.txt)

```
=============================Start of Section==============================

1 2 3 4 5 6 7 8 9 10

 11 12 13 14 15 16 17 18 19 20

 21 22 23 23 24 26 27 28 29 30

 31 32 33 34 35 36 37 38 39 40

 41 42 43 44 45 46 47 48 49 50

 51 52 53 54 55 56 57 58 59 60

 61 62 62 64 65 66 67 68 69 70

 71 72 72 74 75 76 77 78 79 80

 81 82 83 84 85 86 87 88 89 90

 91 92 93 94 95 96 97 98 99 100
==============================End of Section============================
```