Name:	Muhammad Attiq
Reg. No:	FA23 – BCE – 060
Section:	BCE – 4A
Course:	Signals and Systems
Instructor:	Dr. Bilal Qasim
Lab Report:	1
Course Code:	CPE223

LAB 1

Introduction to MATLAB: Basic Commands and Array Manipulation

Task 1

Are the following true or false? Assume A is a generic n×n matrix. Please provide a proper reasoning for your answer.

(a) A^(-1) equals 1/A

(b) A.^(-1) equals 1./A

```
>> A = [ 1 2 3; 4 5 6; 7 8 9 ]

A =

     1     2     3
     4     5     6
     7     8     9

>> A^(-1)
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  1.541976e-18.

ans =

   1.0e+16 *

   -0.4504    0.9007   -0.4504
    0.9007   -1.8014    0.9007
   -0.4504    0.9007   -0.4504

>> 1 / A
Error using  /
Matrix dimensions must agree.

>> 1./A

ans =

    1.0000    0.5000    0.3333
    0.2500    0.2000    0.1667
    0.1429    0.1250    0.1111

>> A.^(-1)

ans =

    1.0000    0.5000    0.3333
    0.2500    0.2000    0.1667
    0.1429    0.1250    0.1111
```

Answer

- Thus Statement (a) is False and Statement (b) is True because MATLAB only performs element wise operations as compared to the A^(-1) and 1/A
- In statement (a) it is calculating inverse by A^(-1) and 1 / A gives error because of not same dimensions.
- In statement (b) A.^(-1) is exponentiation of each element in matrix A and 1./A applies that dividing 1 / each element of matrix A.

Task 2

Vector Generation

(a) Generate the following vectors: A = [1 0 4 5 3 9 0 2] a = [4 5 0 2 0 0 7 1]

Note: Be aware that Matlab are case sensitive. Vector A and a have different values.

```
>> A = [1 0 4 5 3 9 0 2]

A =

     1     0     4     5     3     9     0     2

>> a = [4 5 0 2 0 0 7 1]

a =

     4     5     0     2     0     0     7     1
```

(b) Generate the following vectors: B = [A a] C = [a, A]

```
>> B = [A a]

B =

     1     0     4     5     3     9     0     2     4     5     0     2     0     0     7     1

>> C = [a, A]

C =

     4     5     0     2     0     0     7     1     1     0     4     5     3     9     0     2
```

(c) Generate the following vectors using function zeros and ones:

D = [0 0 0 . . . 0] with fifty 0's. E = [1 1 1 . . . 1] with a hundred 1's.

```
>> D = zeros(1, 50)

D =

  Columns 1 through 21

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0

  Columns 22 through 42

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0

  Columns 43 through 50

     0     0     0     0     0     0     0     0

>> E = ones(1, 100)

E =

  Columns 1 through 21

     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 22 through 42

     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 43 through 63

     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 64 through 84

     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 85 through 100

     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
```

(d) Generate the following vectors using the colon operator

F = [1 2 3 4 . . . 30]

G = [25 22 19 16 13 10 7 4 1]

H = [0 0.2 0.4 0.6 . . . 2.0

```
>> F = 1:1:30

F =

  Columns 1 through 21

     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20    21

  Columns 22 through 30

    22    23    24    25    26    27    28    29    30

>> G = 25:-3:1

G =

    25    22    19    16    13    10     7     4     1
```

```
>> H = 0.0:0.2:2.0

H =

         0    0.2000    0.4000    0.6000    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
```

Task 3

Operate with the vectors

V1 = [1 2 3 4 5 6 7 8 9 0]

V2 = [0.3 1.2 0.5 2.1 0.1 0.4 3.6 4.2 1.7 0.9]

V3 = [4 4 4 4 3 3 2 2 2 1]

(a) Calculate, respectively, the sum of all the elements in vectors V1, V2, and V3.

```
>> V1 = [1 2 3 4 5 6 7 8 9 0]

V1 =

     1     2     3     4     5     6     7     8     9     0

>> sum(V1)

ans =

    45

>> V2 = [0.3 1.2 0.5 2.1 0.1 0.4 3.6 4.2 1.7 0.9]

V2 =

    0.3000    1.2000    0.5000    2.1000    0.1000    0.4000    3.6000    4.2000    1.7000    0.9000

>> sum(V2)

ans =

   15.0000

>> V3 = [4 4 4 4 3 3 2 2 2 1]

V3 =

     4     4     4     4     3     3     2     2     2     1

>> sum(V3)

ans =

    29
```

(b) How to get the value of the fifth element of each vector?

```
>> V1(5)

ans =

     5

>> V2(5)

ans =

    0.1000

>> V3(5)

ans =

     3
```

What happens if we execute the command V1(0) and V1(11)?

```
>> V1(0)
Subscript indices must either be real positive integers or logicals.

>> V1(11)
Index exceeds matrix dimensions.
```

Remember if a vector has N elements, their subscripts are from 1 to N.

(c) Generate a new vector V4 from V2, which is composed of the first five elements of V2.

Generate a new vector V5 from V2, which is composed of the last five elements of V2.

```
>> V4 = V2(1:5)

V4 =

    0.3000    1.2000    0.5000    2.1000    0.1000

>> V5 = V2(6:10)

V5 =

    0.4000    3.6000    4.2000    1.7000    0.9000
```

(d) Derive a new vector V6 from V2, with its 6th element omitted. Derive a new vector V7 from V2, with its 7th element changed to 1.4. Derive a new vector V8 from V2, whose elements are the 1st, 3rd, 5th, 7th, and 9th elements of V2.

```
>> V6 = V2

V6 =

    0.3000    1.2000    0.5000    2.1000    0.1000    0.4000    3.6000    4.2000    1.7000    0.9000

>> V6(6) = []

V6 =

    0.3000    1.2000    0.5000    2.1000    0.1000    3.6000    4.2000    1.7000    0.9000

>> V7 = V2

V7 =

    0.3000    1.2000    0.5000    2.1000    0.1000    0.4000    3.6000    4.2000    1.7000    0.9000

>> V7(7) = 1.4

V7 =

    0.3000    1.2000    0.5000    2.1000    0.1000    0.4000    1.4000    4.2000    1.7000    0.9000

>> V8 = [V2(1:2:9)]

V8 =

    0.3000    0.5000    0.1000    3.6000    1.7000
```

(e) What are the results of

- 9-V1

```
>> 9 - V1

ans =

     8     7     6     5     4     3     2     1     0     9
```

- V1*5

```
>> V1*5

ans =

     5    10    15    20    25    30    35    40    45     0
```

- V1+V2

```
>> V1 + V2

ans =

    1.3000    3.2000    3.5000    6.1000    5.1000    6.4000   10.6000   12.2000   10.7000    0.9000
```

- V1-V3

```
>> V1 - V3

ans =

    -3    -2    -1     0     2     3     5     6     7    -1
```

- V1.*V2

```
>> V1.*V2

ans =

    0.3000    2.4000    1.5000    8.4000    0.5000    2.4000   25.2000   33.6000   15.3000         0
```

- V1*V2

```
>> V1*V2
Error using  *
Inner matrix dimensions must agree.
```

- V1.^2

```
>> V1.^V2

ans =

   1.0e+03 *

    0.0010    0.0023    0.0017    0.0184    0.0012    0.0020    1.1024    6.2084    0.0419         0
```

- V1.^V3

```
>> V1.^V3

ans =

     1    16    81   256   125   216    49    64    81     0
```

- V1^V3

```
>> V1^V3
Error using  ^
Inputs must be a scalar and a square matrix.
To compute elementwise POWER, use POWER (.^) instead.
```

Task 4

Suppose p is a row vector such that p=[4 2 3 1].

What does this line do? Please provide a detailed answer stepwise [length(p)-1:-1:0] .* p

```
>> p=[4 2 3 1]

p =

     4     2     3     1

>> length(p) - 1

ans =

     3

>> [length(p) - 1:-1:0]

ans =

     3     2     1     0

>> [length(p) - 1:-1:0].*p

ans =

    12     4     3     0
```

Answer

- length (p) gives 4 after subtracting 1 it becomes 3.
- we are making a vector which starts from 3 decrementing values up to 0.
- This vector is then element wise multiplied with the original p vector.

Task 5

Suppose A is any matrix. What does this statement do? Please provide a reasonable reason. A(1:size(A,1)+1:end)

```
>> A = [ 1 2 3; 4 5 6; 7 8 9]

A =

     1     2     3
     4     5     6
     7     8     9

>> size(A, 1)

ans =

     3

>> A(1:size(A,1)+1:end)

ans =

     1     5     9
```

Answer

- size(A, 1) gives the number of rows because 1 indicates the number of rows and A is the input matrix
- A(1: size(A, 1)+1:end) this statement indicates that starting from index 1 take 4 steps and grab the element which in this case take the element 5 and then it takes 4 steps and took the element 9 of the matrix

Task 6

Try to avoid using unnecessary brackets in an expression. Can you spot the errors in the following expression? (Test your corrected version with MATLAB.) (2(3+4)/(5*(6+1))ˆ2

```
>> 2*(3+4)/(5*(6+1))^2

ans =

    0.0114
```

Task 7

Set up a vector n with elements 1, 2, 3, 4, 5.

Use MATLAB array operations on it to set up the following four vectors, each with five elements:

(a) 2, 4, 6, 8, 10

(b) 1/2, 1, 3/2, 2, 5/2

(c) 1, 1/2, 1/3, 1/4, 1/5

```
>> A = [ 2, 4, 6, 8, 10 ]

A =

     2     4     6     8    10

>> A./4

ans =

    0.5000    1.0000    1.5000    2.0000    2.5000

>> A = 1./(A./2)

A =

    1.0000    0.5000    0.3333    0.2500    0.2000
```

Task 8

Suppose vectors a and b are defined as follows:

a = [2 –1 5 0];

b = [3 2 –1 4];

Evaluate by hand the vector c in the following statements.

Check your answers with MATLAB.

(a) c = a – b;

(b) c = b + a – 3;

(c) c = 2 * a + a .^ b;

(d) c = b ./ a;

(e) c = b . a;

(f) c = a .^b;

(g) c = 2.^b+a;

(h) c = 2*b/3.*a;

(i) c = b*2.*a;

```
>> a = [2 -1 5 0]

a =

     2    -1     5     0

>> b = [3 2 -1 4]

b =

     3     2    -1     4

>> c = a - b

c =

    -1    -3     6    -4

>> c = b + a - 3

c =

     2    -2     1     1

>> c = 2 * a + a .^ b

c =

   12.0000   -1.0000   10.2000         0

>> c = b ./ a

c =

    1.5000   -2.0000   -0.2000       Inf

>> c = b . a
Attempt to reference field of non-structure array.
```

```
>> c = a .^ b

c =

    8.0000    1.0000    0.2000         0

>> c = 2.^b+a

c =

   10.0000    3.0000    5.5000   16.0000

>> c = 2*b/3.*a

c =

    4.0000   -1.3333   -3.3333         0


>> c = b*2.*a

c =

    12    -4   -10     0
```

Task 9

Make a vector v=[1 2 3 4 5 6 7 8 9 10], develop an algorithm such that the first element of the vector is multiplied by length(v), second element by length(v)-1and similarly the last element i.e. 10 is multiplied by length(v)-9.

The final vector should be f=[10 18 24 28 30 30 28 24 18 10]. The algorithm devised should only use the length of vector v to achieve vector f.

```
>> v = [1 2 3 4 5 6 7 8 9 10]

v =

     1     2     3     4     5     6     7     8     9    10

>> n = length(v)

n =

    10

>> f = v .* (n - (0:n-1))

f =

    10    18    24    28    30    30    28    24    18    10
```

Task 10

(a) Make a matrix M1 which consists of two rows and three columns and all the entries in the matrix are ones.

(b) Make a vector V1 consisting of three ones.

(c) Make a 3x3 matrix M2 in which the diagonal entries are all fives.

(d) Now make a matrix M3 from M1, M2 and V1 which look like the matrix given below

$$M3 = \begin{bmatrix} 1 & 1 & 1 & 5 & 0 & 0 \\ 1 & 1 & 1 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

```
>> M1 = ones(2,3)

M1 =

     1     1     1
     1     1     1

>> V1 = ones(1, 3)

V1 =

     1     1     1

>> M2 = 5 * eye(3)

M2 =

     5     0     0
     0     5     0
     0     0     5
```

```
>> M1(3, :) = V1.*0

M1 =

     1     1     1
     1     1     1
     0     0     0

>> M3 = [M1 M2]

M3 =

     1     1     1     5     0     0
     1     1     1     0     5     0
     0     0     0     0     0     5
```

(e) Now use the referencing element concept to make three vectors V2, V3 and V4 such that V2 consists of first row of M3, V3 consists of second row of M3 and V4 consists of third row of M3.

(f) Now alter the fourth entry of vectors V2, fifth entry of V3 and sixth entry of V4 to 1.4 and make a new vector M4 which looks like the matrix given below.

$$M4 = \begin{bmatrix} 1 & 1 & 1 & 1.4 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1.4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.4 \end{bmatrix}$$

```
>> V2 = M3(1,:)

V2 =

     1     1     1     5     0     0

>> V3 = M3(2, :)

V3 =

     1     1     1     0     5     0

>> V4 = M3(3, :)

V4 =

     0     0     0     0     0     5



>> V2(4) = 1.4

V2 =

    1.0000    1.0000    1.0000    1.4000         0         0

>> V3(5) = 1.4

V3 =

    1.0000    1.0000    1.0000         0    1.4000         0

>> V4(6) = 1.4

V4 =

         0         0         0         0         0    1.4000

>> M4 = [ V2; V3; V4]

M4 =

    1.0000    1.0000    1.0000    1.4000         0         0
    1.0000    1.0000    1.0000         0    1.4000         0
         0         0         0         0         0    1.4000
```

Post-lab Task

Critical Analysis / Conclusion:

In this lab, we learned basic MATLAB commands for working with matrices and vectors. We explored how to perform different operations like adding, multiplying, and changing elements in a matrix. Understanding the difference between matrix operations and element-wise operations is important for writing correct code. Overall, this lab helped build a good foundation for more complex topics in signals and systems.

Lab Assessment		
Pre Lab	**/5**	**/25**
Performance	**/5**	
Results	**/5**	
Viva	**/5**	
Critical Analysis	**/5**	
Instructor Signature and Comments		