2.6- Matrices Linear Algebra:

The mathematical operations defined on matrices are the subject of *linear algebra*. Let:

```
A =
   16 3 2 13
    5 10 11 8
    9  6 7 12
    4 15 14 1
```

provides several examples that give a taste of MATLAB matrix operations. You’ve already seen the matrix transpose, A'. Adding a matrix to its transpose produces a *symmetric* matrix.

```
>> A + A'
ans =

32 8 11 17
8 20 17 23
11 17 14 26
17 23 26 2
```

The multiplication symbol, *, denotes the *matrix* multiplication involving inner products between rows and columns. Multiplying the transpose of a matrix by the original matrix also produces a symmetric matrix.

```
>> A'*A
ans =

378 212 206 360
212 370 368 206
206 368 370 212
360 206 212 378
```

The determinant of this particular matrix happens to be zero, indicating that the matrix is *singular*.

```
>> d = det(A)
d =
   0
```

Since the matrix is singular, it does not have an inverse. If you try to compute the inverse with

```
>> X = inv(A)
```

you will get a warning message "warning: Matrix is close to singular or badly scaled".

```
>> e = eig(A)

e =
34.0000
8.0000
0.0000
-8.0000
```

One of the eigen values is zero, which is another consequence of singularity.

2.7- Generating Matrices:

MATLAB provides four functions that generate basic matrices of size (R×C):

Zeros(R,C)	All the elements of the matrix are zeros.
Ones(R,C)	All the elements of the matrix are ones.
Rand(R,C)	Uniformly distributed random elements.
Randn(R,C)	Normally distributed random elements.

and here are some examples.

```
>> Z = zeros(2,4)

Z =
   0 0 0 0
   0 0 0 0

>> F = 5*ones(3,3)

F =
   5 5 5
   5 5 5
   5 5 5
>> N = fix(10*rand(1,10))

N =
   4 9 4 4 8 5 2 6 8 0
```

```
>> R = randn(4,4)
R =
    1.0668   0.2944   -0.6918   -1.4410
    0.0593  -1.3362    0.8580    0.5711
   -0.0956   0.7143    1.2540   -0.3999
   -0.8323   1.6236   -1.5937    0.6900
```

3- Arrays

Informally, the terms *matrix* and *array* are often used interchangeably. More precisely, a *matrix* is a two-dimensional numeric array that represents a *linear transformation*.

3.1- Array operators:

Arithmetic operations on arrays are done element-by-element. This means that addition and subtraction are the same for arrays and matrices, but that multiplicative operations are different. MATLAB uses a dot, or decimal point, as part of the notation for multiplicative array operations. The list of operators includes:

+	Addition
-	Subtraction
.*	Element-by-element multiplication
./	Element-by-element division
.\	Element-by-element left division
.^	Element-by-element power
.'	Unconjugated array transpose

As an example: enter the following statements at the command line

```
>> a = [2  4  8];
>> b = [4  2  2];
>> a . * b       <ENTER>
ans =
      8   8  16
>> a . / b       <ENTER>
ans =
```

```
        0.5   2   4
>> a . ^ b
```
<ENTER>

```
ans =
        16   16   64
```

A common application of element-by-element multiplication is in finding the *scalar product* (also called the *dot product*) of two vectors **a** and **b**, which is defined as:

$$\mathbf{a}\,.\,\mathbf{b} = \sum_{\mathbf{i}} \mathbf{a_i b_i}$$

and in MATLAB can be represented as:

```
>> Sum ( a .* b )
ans =
        32
```

3.2- <u>Array tables:</u>

Array operations are useful for building tables. Suppose n is the column vector

```
>> n = (0:8)';
```

Then

```
>> pows = [n n.^2 2.^n]
```

builds a table of squares and powers of two.

```
pows =
         0 0 1
         1 1 2
         2 4 4
         3 9 8
         4 16 16
         5 25 32
         6 36 64
         7 49 128
         8 64 256
```

The elementary math functions operate on arrays element by element. So format short g

```
>> x = (1:0.1:2)';
>> logs = [x log10(x)]
```

builds a table of logarithms.

```
logs =
        1.0   0
        1.1   0.04139
        1.2   0.07918
        1.3   0.11394
        1.4   0.14613
        1.5   0.17609
        1.6   0.20412
        1.7   0.23045
        1.8   0.25527
        1.9   0.27875
        2.0   0.30103
```

Exercises:

1. Let C be any 4×4 matrix. Write some statements to find :

a) Sum of each columns.
b) Sum of each rows.
c) Sum of the main diagonal elements.
d) Sum of the anti diagonal elements.
e) Sum of the third row.

2. Set up any 3×3 matrix D. Write some statements to convert D into a row vector X contains:

a) The odd elements of D.
b) The even elements of D.
c) The first and the last elements of D.
d) The three last elements of D.

3. If A and B are 2×2 matrices. Find a matrix C such that:

a) $\mathbf{C = A^T + B}$

b) $\mathbf{C} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{A+1} & \mathbf{B+2} \end{bmatrix}$

c) $\mathbf{C = AB / (A+B)}$

d) $\mathbf{C = A - A^{-1}B}$

4. If A is 4×4 normally distributed random matrix and I is 4×4 identity matrix. Proof that:

a) $\mathbf{A^{-1} A = I}$
b) A is not a singular matrix (use two different solutions).
c) $\mathbf{|A| - |I|\,|A| = 0}$
d) $\mathbf{A^{-1} A = AA^{-1}}$

5. Solve the equations below :

$$\mathbf{2x - y + z = 4}$$
$$\mathbf{x + y + z = 3}$$
$$\mathbf{3x - y - z = 1}$$

Hint : The solution of the equation AX=B is: $X=A^{-1}B$. Where A is the variables coefficient matrix, X is the variables column vector and B is the constants column vector.

6. If X and Y are two row vectors. Use the array operations to find:

a) $\sum_{i=1}^{10} X_i Y_i$ **b)** $\sum_{i=1}^{10} X_i^{Y_i}$ **c)** $\sum_{i=1}^{10} 4X_i^3 + \sum_{i=1}^{10} 5Y_i^2$ **d)** $\sum_{i=1}^{10} [6(\frac{X_i}{Y_i}) - 2(X_i^{Y_i})]$

7. Build the following table by using arrays where the table below converts the power to its value in decibels (dB) according to the relation:

$$\mathbf{G[dB] = 10\log(P)}$$

Where the function log is the logarithm to base 10.

P	G[dB]
2	
1	
0.5	
0.1	
10^{-3}	