Experiment 01

'MATRICES'

--> //Forming a Row Matrix//

```
--> P=[1 10 7]
 P  =
   1.   10.   7.
```

--> //Forming a Column Matrix//

```
--> Q=[-4;
 > 5;
 > 2]
 Q  =
  -4.
   5.
   2.
```

-->//Forming any desired Matrix//

```
--> R=[1 2 3;
 > 5 6 7]
 R  =
   1.   2.   3.
   5.   6.   7.
```

--> //Forming Random Matrix of any order//

```
--> A=testmatrix('magi',3)
```

//here order of matrix is 3//

```
 A  =
   8.   1.   6.
   3.   5.   7.
```

```
  4.  9.  2.
```

--> //Numbers of Row and Column//

```
--> size(A)
 ans  =
  4.  4.
```

--> //Total numbers of elements in Matrix//

```
--> length(A)
 ans  =
  16.
```

--> //Sum of elements of Row or Column//

```
--> sum(A,'r')
 ans  =
  15.  15.  15.

--> sum(A,'c')
 ans  =
  15.
  15.
  15.

--> sum(A)
 ans  =  45.
```

--> //Product of elements of Matrix//

```
--> prod(A,'c')
 ans  =
  48.
  105.
  72.
```

```
--> prod(A,'c')
 ans  =
  48.
  105.
  72.

--> prod(A)
 ans  =
  362880.
```

--> //Minimum or Maximum values within Matrix//

```
--> min(A,'r')
 ans  =
 3.   1.   2.

--> max(A,'r')
 ans  =
  8.   9.   7.

--> min(A,'c')
 ans  =
  1.
  3.
  2.

--> max(A,'r')
 ans  =
  8.   9.   7.

--> max(A,'c')
 ans  =
  8.
  7.
```

9.

--> min(A)
ans =
1.

--> max(A)
ans =
9.

--> //Singularity of Matrix//
--> clean(det(A))~=0
ans =
T
--> clean(det(A))
ans =
-360.

--> //Determinant of Matrix A//
--> det(A)
ans =
-360.

--> //Diagonal elements os Matrix A//
--> diag(A)
ans =
8.
5.
2.

--> //Transpose of Matrix A//
--> A'
ans =
8. 3. 4.

1. 5. 9.
6. 7. 2

--> //inverse of Matrix A//

--> inv(A)
ans =
0.1472222 -0.1444444 0.0638889
-0.0611111 0.0222222 0.1055556
-0.0194444 0.1888889 -0.1027778

--> //Adjoint of Matrix A//

--> det(A)*inv(A)
ans =
-53. 52. -23.
22. -8. -38.
7. -68. 37.

--> //Trace of Matrix A//

--> trace(A)
ans =
15.

--> //Eigenvalues and Eigenvectors of Matrix A//

--> [V,E]=spec(A) **//here E=eigenvalues, V=eigenvectors//**
E =
15. 0. 0.
0. 4.8989795 0.
0. 0. -4.898979
V =
-0.5773503 -0.8130525 -0.341648
-0.5773503 0.4714045 -0.4714045

-0.5773503 0.341648 0.8130525

--> //Rank of Matrix A//

--> rank(A)

ans =

3.

--> //Operations on elements of Matrix//

--> zeros(A) **//All element Zero//**

ans =

0. 0. 0.

0. 0. 0.

0. 0. 0.

--> ones(A) **//All elements one//**

ans =

1. 1. 1.

1. 1. 1.

1. 1. 1.

--> rand(A) **//random elements//**

ans =

0.2113249 0.3303271 0.8497452

0.7560439 0.6653811 0.685731

0.0002211 0.6283918 0.8782165

--> eye(A) **//Identity Matrix//**

ans =

1. 0. 0.

0. 1. 0.

0. 0. 1.

--> //COLON Operator//

--> //To view any Row or Column//

```
--> A(1,2)
 ans  =
   1.

--> A(2,3)
 ans  =
   7.
```

--> //To view any row or column of matrix//

```
--> A(:,1)
 ans  =
   8.
   3.
   4.

--> A(3,:)
 ans  =
  4.   9.   2.
```

--> //Changing Row or Column//

```
--> A(:,3)=[-19 8 -7]
 A  =
   8.   1.  -19.
   3.   5.   8.
   4.   9.  -7.

--> A(2,:)=[12 11 -6]
 A  =
   8.   1.   -19.
   12.  11.  -6.
   4.   9.   -7.
```

--> //Deleting any Row or Column//

```
--> A(:,3)=[]
 A =
  8.   1.
  12.  11.
  4.   9.

--> A(1,:)=[]
 A =
  12.  11.
  4.   9.
```

--> //Adding any Row or Column to Matrix//

```
--> A(3,:)=[10 -5]
 A =
  12.  11.
  4.   9.
  10.  -5.

--> A(:,3)=[-9 8 5]
 A =
  12.  11.  -9.
  4.   9.   8.
  10.  -5.   5.
```

--> //Normally distribute random numbers in Matrix//

```
--> A=rand(2,3,'narmal')
 A =
 -0.7335813   0.8915736  -1.3925211
  0.1034169   1.2429915   0.2044185
```

--> //Operator 'gsort'=pin out the matrix element in decresing order and return them column by column//

```
--> A=[1 -5 6 10;
 > 5 -4 7 11;
 > 30 12 -1 9;
 > 11 -11 6 7]

 A =
  1.  -5.   6.  10.
  5.  -4.   7.  11.
  30.  12. -1.  9.
  11. -11.  6.  7.


--> gsort(A)
 ans =
  30.  10.  6. -1.
  12.  9.   6. -4.
  11.  7.   5. -5.
  11.  7.   1. -11.
```