Importing libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib as plt
from numpy import linalg as la
```

1.Create 5 matrices with five different dimensions(1-D,2-D,.....,5-D)

In [6]:

```
A=np.array([[2,3,4]])
print(A)
```

```
[[2 3 4]]
```

In [7]:

```
B=np.array([[1,2],[4,5]])
print(B)
```

```
[[1 2]
 [4 5]]
```

In [8]:

```
C=np.array([[4,5,6],[7,8,9],[7,8,5]])
print(C)
```

```
[[4 5 6]
 [7 8 9]
 [7 8 5]]
```

In [9]:

```
D=np.array([[1,2,3,4],[4,5,6,7],[7,8,9,0],[3,4,5,6]])
print(D)
```

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

In [11]:

```
E=np.array([[5,6,7,8,9],[9,0,7,6,8],[0,4,3,2,7],[7,6,8,6,4],[9,8,7,6,4]])
print(E)
```

```
[[5 6 7 8 9]
 [9 0 7 6 8]
 [0 4 3 2 7]
 [7 6 8 6 4]
 [9 8 7 6 4]]
```

2.Find determinate of 5 matrices and display your output

In [12]:

```
print(la.det(B))
```

```
-2.9999999999999996
```

In [13]:

```
print(la.det(C))
```

```
12.000000000000005
```

In [14]:

```
print(la.det(D))
```

```
4.123685520036316e-15
```

In [15]:

```
print(la.det(E))
```

```
3999.9999999999995
```

3.Find inverse of the above 5 matrices and display the output

In [17]:

```
print(la.inv(B))
```

```
[[-1.66666667  0.66666667]
 [ 1.33333333 -0.33333333]]
```

In [18]:

```
print(la.inv(C))
```

```
[[-2.66666667  1.91666667 -0.25      ]
 [ 2.33333333 -1.83333333  0.5       ]
 [-0.          0.25       -0.25      ]]
```

In [19]:

```
print(la.inv(D))
```

```
[[ 2.42501518e+15  4.85003037e+15 -4.78260870e-02 -7.27504555e+15]
 [-4.85003037e+15 -9.70006074e+15 -4.34782609e-03  1.45500911e+16]
 [ 2.42501518e+15  4.85003037e+15  1.52173913e-01 -7.27504555e+15]
 [-2.30769231e-02  3.53846154e-01 -1.00000000e-01 -2.30769231e-01]]
```

In [20]:

```
print(la.inv(E))
```

```
[[-0.084  0.094 -0.012 -0.188  0.21 ]
 [-0.01  -0.09   0.07  -0.07   0.15 ]
 [-0.188  0.008  0.116  0.484 -0.28 ]
 [ 0.372 -0.077 -0.304 -0.096 -0.055]
 [-0.02   0.07   0.14  -0.14   0.05 ]]
```

4.Find the rank,diagonaland trace of the 5 matrices

In [21]:

```
print(la.matrix_rank(A))
```

```
1
```

In [23]:

```
print(la.matrix_rank(B))
```

```
2
```

In [24]:

```
print(la.matrix_rank(C))
```

```
3
```

In [25]:

```
print(la.matrix_rank(D))
```

```
3
```

In [26]:

```
print(la.matrix_rank(E))
```

```
5
```

DIAGONAL

In [27]:

```
print(np.diag(A))
```

```
[2]
```

In [28]:

```
print(np.diag(B))
```

```
[1 5]
```

In [29]:

```
print(np.diag(C))
```

```
[4 8 5]
```

In [30]:

```
print(np.diag(D))
```

```
[1 5 9 6]
```

In [31]:

```
print(np.diag(E))
```

```
[5 0 3 6 4]
```

TRACE

In [32]:

```
print(np.trace(A))
```

```
2
```

In [33]:

```
print(np.trace(B))
```

```
6
```

In [34]:

```
print(np.trace(C))
```

```
17
```

In [35]:

```
print(np.trace(D))
```

```
21
```

In [36]:

```
print(np.trace(E))
```

```
18
```

6.Find eigen value and eigen vector for 5 matrices

eigen value

In [38]:

```
print(la.eig(B))
```

```
(array([-0.46410162,  6.46410162]), array([[-0.80689822, -0.34372377],
       [ 0.59069049, -0.9390708 ]]))
```

In [39]:

```
print(la.eig(C))
```

```
(array([19.89531295, -0.22595597, -2.66935698]), array([[-0.43511679, -0.7
3877911, -0.42586713],
       [-0.6919349 ,  0.67274299, -0.40365284],
       [-0.57610718, -0.04027783,  0.80975402]]))
```

In [40]:

```
print(la.eig(D))
```

```
(array([ 1.92574059e+01,  2.39346682e+00, -6.50872704e-01, -5.73119297e-1
5]), array([[-2.66925230e-01, -1.57663574e-01, -8.70521231e-01,
         4.08248290e-01],
       [-5.66762273e-01, -4.93898914e-01,  3.22769512e-01,
        -8.16496581e-01],
       [-6.24190450e-01,  7.65126912e-01,  3.63852330e-01,
         4.08248290e-01],
       [-4.66816592e-01, -3.81820467e-01, -7.49940688e-02,
         6.69116987e-16]]))
```

In [41]:

```
print(la.eig(E))
```

```
(array([29.02524063+0.j        , -5.87901187+1.10059186j,
       -5.87901187-1.10059186j,  0.36639155+1.92821621j,
        0.36639155-1.92821621j]), array([[-0.52401205+0.j        ,  0.3283
2675+0.16896396j,
         0.32832675-0.16896396j, -0.29570511+0.09636125j,
        -0.29570511-0.09636125j],
       [-0.45387874+0.j        ,  0.25930685-0.4115139j ,
         0.25930685+0.4115139j , -0.10183799+0.12948779j,
        -0.10183799-0.12948779j],
       [-0.2411905 +0.j        ,  0.35767066+0.25131617j,
         0.35767066-0.25131617j,  0.68450744+0.j        ,
         0.68450744-0.j        ],
       [-0.44977988+0.j        , -0.37671824-0.09586713j,
        -0.37671824+0.09586713j, -0.22143897-0.52033204j,
        -0.22143897+0.52033204j],
       [-0.5088523 +0.j        , -0.53373564+0.j        ,
        -0.53373564-0.j        , -0.13607067+0.26322732j,
        -0.13607067-0.26322732j]]))
```

EIGEN VECTOR

In [44]:

```
print(la.eigvals(B))
```

```
[-0.46410162  6.46410162]
```

In [45]:

```
print(la.eigvals(C))
```

```
[19.89531295 -0.22595597 -2.66935698]
```

In [46]:

```
print(la.eigvals(D))
```

```
[ 1.92574059e+01  2.39346682e+00 -6.50872704e-01 -5.73119297e-15]
```

In [47]:

```
print(la.eigvals(E))
```

```
[29.02524063+0.j         -5.87901187+1.10059186j -5.87901187-1.10059186j
  0.36639155+1.92821621j  0.36639155-1.92821621j]
```

In [ ]:

```

```