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

In [2]:

```
import numpy as np
import pandas as pd
from numpy import linalg as li
```

In [3]:

```
a = np.array([1])
print(a)
```

```
[1]
```

In [5]:

```
b = np.array([[1,2,3,4,5],[1,2,3,4,5]])
print(b)
```

```
[[1 2 3 4 5]
 [1 2 3 4 5]]
```

In [6]:

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

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

In [7]:

```
d = np.array([[1,2,3,4,5],[1,2,3,4,5],[9,8,7,6,5],[11,12,13,14,15]])
print(d)
```

```
[[ 1  2  3  4  5]
 [ 1  2  3  4  5]
 [ 9  8  7  6  5]
 [11 12 13 14 15]]
```

In [9]:

```
e = np.array([[1,2,3,4,5],[1,2,3,4,5],[9,8,7,6,5],[11,12,13,14,15],[22,23,24,25,26
print(e)
```

```
[[ 1  2  3  4  5]
 [ 1  2  3  4  5]
 [ 9  8  7  6  5]
 [11 12 13 14 15]
 [22 23 24 25 26]]
```

2. Find determinants of 5 matrices and display your output

In [13]:

```
e = np.array([[1,2,3],[1,2,55],[9,8,7]])
s = np.array([[11,12],[44,55]])
f = np.array([[10,20,30,40],[15,24,30,74],[19,58,7,6],[110,127,135,140]])
g = np.array([[1,2,3,4,5],[1,2,3,4,88],[99,88,77,66,55],[1,1,13,14,5],[2,3,24,5,26
h = np.array([[19,20,36,4,55],[61,2,83,4,45],[9,18,7,16,5],[11,1,13,14,5],[22,23,2
print(li.det(e))
print(li.det(s))
print(li.det(f))
print(li.det(g))
print(li.det(h))
```

```
520.0000000000005
77.00000000000001
-3154500.000000001
-2373799.999999997
5573699.999999998
```

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

In [14]:

```
print(li.inv(e))
```

```
[[-0.81923077  0.01923077  0.2       ]
 [ 0.93846154 -0.03846154 -0.1       ]
 [-0.01923077  0.01923077 -0.        ]]
```

In [15]:

```
print(li.inv(s))
```

```
[[ 0.71428571 -0.15584416]
 [-0.57142857  0.14285714]]
```

In [16]:

```
print(li.inv(f))
```

```
[[-0.09414297  0.01911555 -0.0134855   0.01737201]
 [ 0.02389602 -0.0042796   0.02291964 -0.00554763]
 [ 0.07328832 -0.03965763 -0.00983357  0.00044381]
 [-0.01837851  0.02710414 -0.00071327 -0.00190204]]
```

In [17]:

```
print(li.inv(g))
```

```
[[-8.07771084e-01  3.89249305e-02  1.71678322e-02  1.53846154e-01
  -4.23076923e-02]
 [ 9.27108434e-01 -4.63392030e-02 -7.34265734e-03 -2.30769231e-01
   3.84615385e-02]
 [-4.29518072e-02 -1.20481928e-02 -4.54545455e-04  1.66533454e-18
   5.00000000e-02]
 [ 3.56626506e-02  7.41427247e-03 -2.79720280e-04  7.69230769e-02
  -4.61538462e-02]
 [-1.20481928e-02  1.20481928e-02  0.00000000e+00  0.00000000e+00
   0.00000000e+00]]
```

In [18]:

```
print(li.inv(h))
```

```
[[-0.14726304  0.05515815  0.03089241 -0.20157167  0.10784578]
 [ 0.02196028 -0.00145146  0.06015932 -0.03719253 -0.02095556]
 [ 0.11545293 -0.02830077 -0.00994671  0.1598579  -0.09709888]
 [ 0.01243339 -0.01740675 -0.01030195  0.09413854 -0.0053286 ]
 [-0.01540449  0.00126325 -0.02528823 -0.0283223   0.03430755]]
```

4. Find the rank, diagonal and trace of the 5 matrices

In [19]:

```
print(li.matrix_rank(e))
print(np.diag(e))
print(np.trace(e))
```

```
3
[1 2 7]
10
```

In [20]:

```
print(li.matrix_rank(s))
print(np.diag(s))
print(np.trace(s))
```

```
2
[11 55]
66
```

In [21]:

```
print(li.matrix_rank(f))
print(np.diag(f))
print(np.trace(f))
```

```
4
[ 10  24   7 140]
181
```

In [22]:

```
print(li.matrix_rank(g))
print(np.diag(g))
print(np.trace(g))
```

```
5
[ 1  2 77 14 26]
120
```

In [23]:

```
print(li.matrix_rank(h))
print(np.diag(h))
print(np.trace(h))
```

```
5
[19  2  7 14 60]
102
```

5. Find Eigen value and eigen vector for 5 matrices

In [26]:

```
o,p = li.eig(e)
print(p)
print(o)
```

```
[[-0.11636192 -0.64726532  0.05745268]
 [-0.90411499  0.76156857 -0.9494539 ]
 [-0.41113988 -0.03257175  0.30860409]]
[ 27.13956234  -1.20222196 -15.93734038]
```

In [27]:

```
q,r = li.eig(s)
print(q)
print(r)
```

```
[ 1.18805256 64.81194744]
[[-0.77415438 -0.2176527 ]
 [ 0.63299684 -0.97602628]]
```

In [31]:

```
t,u = li.eig(f)
print(t)
print(u)
```

```
[234.47783301 +0.j        -20.02171928+24.5058729j
 -20.02171928-24.5058729j -13.43439445 +0.j       ]
[[ 0.20969896+0.j          0.24296   +0.05322741j  0.24296   -0.05322741j
  -0.66108854+0.j        ]
 [ 0.35111503+0.j          0.25453475-0.37303637j  0.25453475+0.37303637j
  -0.02670509+0.j        ]
 [ 0.13085958+0.j         -0.76929659+0.j         -0.76929659-0.j
   0.73461408+0.j        ]
 [ 0.90311702+0.j          0.23474355+0.2954174j   0.23474355-0.2954174j
  -0.15030278+0.j        ]]
```

In [30]:

```
z,x = li.eig(g)
print(z)
print(x)
```

```
[123.54040599 +0.j          -7.24816806+36.14973678j
  -7.24816806-36.14973678j  -1.1660903  +0.j
  12.12202043 +0.j        ]
[[ 0.04037009+0.j          0.0172162 +0.00706531j  0.0172162 -0.00706531j
  -0.65520756+0.j         -0.1111123 +0.j        ]
 [ 0.20462711+0.j          0.66630227+0.j          0.66630227-0.j
   0.7546798 +0.j          0.61475867+0.j        ]
 [ 0.93869574+0.j         -0.42792497-0.50894678j -0.42792497+0.50894678j
  -0.02107009+0.j          0.03214454+0.j        ]
 [ 0.12479846+0.j         -0.04455306+0.1685823j  -0.04455306-0.1685823j
   0.01803223+0.j         -0.77295146+0.j        ]
 [ 0.24448649+0.j         -0.05360573+0.28331929j -0.05360573-0.28331929j
  -0.0198079 +0.j          0.10601232+0.j        ]]
```

In [32]:

```
u,i = li.eig(h)
print(u)
print(i)
```

```
[118.67120873+0.j        -36.9327678 +0.j         -6.61099252+0.j
  13.4362758 +3.4392845j  13.4362758 -3.4392845j]
[[ 0.50580035+0.j         -0.05406624+0.j          0.80247827+0.j
  -0.43483366+0.1554564j  -0.43483366-0.1554564j ]
 [ 0.61536847+0.j         -0.90299471+0.j          0.09952535+0.j
   0.21117122+0.24022774j  0.21117122-0.24022774j]
 [ 0.18087573+0.j          0.39714274+0.j         -0.58495063+0.j
   0.55580996+0.j          0.55580996-0.j        ]
 [ 0.10856221+0.j         -0.08467847+0.j         -0.05864811+0.j
   0.36590341-0.20210938j  0.36590341+0.20210938j]
 [ 0.56655615+0.j          0.12955534+0.j         -0.02272649+0.j
  -0.43293747-0.11557365j -0.43293747+0.11557365j]]
```

In [ ]:

```
```