**import** numpy **as** np

**import** pandas **as** pd

**from** numpy **import** linalg **as** la

In [13]:

a **=** np**.**array([1])

a

Out[13]:

array([1])

In [12]:

b **=** np**.**array([[1,2],[4,5]])

b

Out[12]:

array([[1, 2],

[4, 5]])

In [5]:

c **=** np**.**array([[1,2,3],[4,5,6],[2,4,6]])

c

Out[5]:

array([[1, 2, 3],

[4, 5, 6],

[2, 4, 6]])

In [11]:

d **=** np**.**array([[1,2,3,4],[4,5,6,7],[2,4,6,7],[7,8,9,10]])

d

Out[11]:

array([[ 1, 2, 3, 4],

[ 4, 5, 6, 7],

[ 2, 4, 6, 7],

[ 7, 8, 9, 10]])

In [9]:

e **=** np**.**array([[1,2,3,4,5],[4,5,6,7,8],[2,4,6,8,10],[7,8,9,10,11],[10,11,12,13,14]])

e

Out[9]:

array([[ 1, 2, 3, 4, 5],

[ 4, 5, 6, 7, 8],

[ 2, 4, 6, 8, 10],

[ 7, 8, 9, 10, 11],

[10, 11, 12, 13, 14]])

**Find determinants of 5 matrices and display your output**

In [18]:

print("M1")

print(la**.**det(b))

print("M2")

print(la**.**det(c))

print("M3")

print(la**.**det(d))

print("M4")

print(la**.**det(e))

print("M5")

print(la**.**det(b))

M1

-2.9999999999999996

M2

0.0

M3

-9.51619735392994e-16

M4

0.0

M5

-2.9999999999999996

**Find inverse of the above 5 matrices and display your output**

In [22]:

print("inverse of M1")

print(la**.**inv(b))

inverse of M1

[[-1.66666667 0.66666667]

[ 1.33333333 -0.33333333]]

**Find the rank, diagonal and trace of the 5 matrices**

In [28]:

print("rank")

print(la**.**matrix\_rank(a))

print(la**.**matrix\_rank(b))

print(la**.**matrix\_rank(c))

print(la**.**matrix\_rank(d))

print(la**.**matrix\_rank(b))

rank

1

2

2

3

2

In [31]:

print("diagonal")

print(np**.**diag(a))

print(np**.**diag(b))

print(np**.**diag(c))

print(np**.**diag(d))

print(np**.**diag(e))

diagonal

[[1]]

[1 5]

[1 5 6]

[ 1 5 6 10]

[ 1 5 6 10 14]

In [35]:

print("trace")

print(np**.**trace(c))

print(np**.**trace(b))

print(np**.**trace(c))

print(np**.**trace(d))

print(np**.**trace(e))

trace

12

6

12

22

36

**Find Eigen value and eigen vector for 5 matrices**

In [37]:

print("Eigen vector:")

print(la**.**eig(b))

print(la**.**eig(c))

print(la**.**eig(d))

print(la**.**eig(e))

print(la**.**eig(c))

Eigen vector:

(array([-0.46410162, 6.46410162]), array([[-0.80689822, -0.34372377],

[ 0.59069049, -0.9390708 ]]))

(array([ 1.17445626e+01, -4.29618684e-15, 2.55437353e-01]), array([[ 0.30674675, 0.40824829, 0.24714134],

[ 0.72768959, -0.81649658, -0.83343013],

[ 0.61349351, 0.40824829, 0.49428268]]))

(array([ 2.30145011e+01, -1.46070014e+00, 4.46199030e-01, -1.44289016e-15]), array([[-2.34681030e-01, -5.00108755e-01, 1.06208393e-01,

4.08248290e-01],

[-4.79134532e-01, 8.53231865e-02, 2.85856465e-01,

-8.16496581e-01],

[-4.37921534e-01, -5.41016400e-01, -8.30849797e-01,

4.08248290e-01],

[-7.23588034e-01, 6.70755128e-01, 4.65504537e-01,

-4.39114384e-16]]))

(array([ 3.92132034e+01, -3.21320344e+00, -3.58386232e-15, 7.77550482e-16,

-4.03832920e-17]), array([[-0.1881031 , 0.35583668, 0.24891244, 0.50206111, 0.04035691],

[-0.34771388, 0.05389925, -0.75177094, -0.3724284 , 0.07732766],

[-0.37620621, 0.71167337, 0.52679689, -0.19517453, 0.2207703 ],

[-0.50732465, -0.24803819, 0.2060693 , -0.50061017, -0.83495125],

[-0.66693543, -0.54997563, -0.23000768, 0.566152 , 0.49649637]]))

(array([ 1.17445626e+01, -4.29618684e-15, 2.55437353e-01]), array([[ 0.30674675, 0.40824829, 0.24714134],

[ 0.72768959, -0.81649658, -0.83343013],

[ 0.61349351, 0.40824829, 0.49428268]]))