МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»

Отчет по лабораторной работе №3
«Исследование методов работы с матрицами и векторами с помощью библиотеки NumPy»

по дисциплине «Технологии распознавания образов»

Выполнил студент группы ПИЖ-б-о-20-1
Симоненко А.С. « » _____ 2022г.
Подпись студента _____
Работа защищена « »_____2022г.
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь, 2022 г.

Примеры

In [1]:
```
import numpy as np
```

In [2]:
```
v_hor_np = np.array([1, 2])
print(v_hor_np )
```

```
[1 2]
```

In [4]:
```
v_hor_zeros_v1 = np.zeros((5,))
print(v_hor_zeros_v1 )
```

```
[0. 0. 0. 0. 0.]
```

In [5]:
```
v_hor_zeros_v2 = np.zeros((1, 5))
print(v_hor_zeros_v2 )
```

```
[[0. 0. 0. 0. 0.]]
```

In [6]:
```
 v_hor_one_v1 = np.ones((5,))
 print(v_hor_one_v1)
```

```
[1. 1. 1. 1. 1.]
```

In [7]:
```
v_hor_one_v2 = np.ones((1, 5))
print(v_hor_one_v2)
```

```
[[1. 1. 1. 1. 1.]]
```

In [8]:
```
v_vert_np = np.array([[1], [2]])
print(v_vert_np)
```

```
[[1]
 [2]]
```

In [9]:
```
v_vert_zeros = np.zeros((5, 1))
print(v_vert_zeros)
```

```
[[0.]
 [0.]
 [0.]
 [0.]
 [0.]]
```

In [10]:
```
v_vert_ones = np.ones((5, 1))
print(v_vert_ones)
```

```
[[1.]
 [1.]
 [1.]
 [1.]
 [1.]]
```

In [11]:
```
m_sqr_arr = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
print(m_sqr_arr)
```

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

In [12]:
```
m_sqr = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
m_sqr_arr = np.array(m_sqr)
print(m_sqr_arr)
```

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

In [13]:
```
m_sqr_mx = np.matrix([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
print(m_sqr_mx)
```

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

In [14]:
```
m_sqr_mx = np.matrix('1 2 3; 4 5 6; 7 8 9')
print(m_sqr_mx)
```

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

In [15]:
```
m_diag = [[1, 0, 0], [0, 5, 0], [0, 0, 9]]
m_diag_np = np.matrix(m_diag)
print(m_diag_np)
```

```
[[1 0 0]
 [0 5 0]
 [0 0 9]]
```

In [16]:
```
m_sqr_mx = np.matrix('1 2 3; 4 5 6; 7 8 9')
```

In [17]:
```
diag = np.diag(m_sqr_mx)
print(diag)
```

```
[1 5 9]
```

In [18]:
```
m_diag_np = np.diag(np.diag(m_sqr_mx))
print(m_diag_np)
```

```
[[1 0 0]
 [0 5 0]
 [0 0 9]]
```

In [20]:
```
m_e = [[1, 0, 0], [0, 1, 0], [0, 0, 1]]
m_e_np = np.matrix(m_e)
print(m_e_np)
```

```
[[1 0 0]
 [0 1 0]
 [0 0 1]]
```

In [21]:
```
m_eye = np.eye(3)
print(m_eye)
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

In [22]:
```
m_idnt = np.identity(3)
print(m_idnt)
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

In [23]:
```
m_zeros = np.zeros((3, 3))
print(m_zeros)
```

```
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
```

In [24]:
```
m_mx = np.matrix('1 2 3; 4 5 6')
print(m_mx)
```

```
[[1 2 3]
 [4 5 6]]
```

In [25]:
```
m_var = np.zeros((2, 5))
print(m_var)
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```

In [26]:
```
A = np.matrix('1 2 3; 4 5 6')
print(A)
```

```
[[1 2 3]
 [4 5 6]]
```

In [27]:
```
A_t = A.transpose()
print(A_t)
```

```
[[1 4]
 [2 5]
 [3 6]]
```

In [28]:
```
print(A.T)
```

```
[[1 4]
 [2 5]
 [3 6]]
```

In [29]:
```
A = np.matrix('1 2 3; 4 5 6')
print(A)
```

```
[[1 2 3]
 [4 5 6]]
```

In [30]:
```
R = (A.T).T
print(R)
```

```
[[1 2 3]
 [4 5 6]]
```

In [31]:
```
A = np.matrix('1 2 3; 4 5 6')
B = np.matrix('7 8 9; 0 7 5')
L = (A + B).T
R = A.T + B.T
print(L)
```

```
[[ 8  4]
 [10 12]
 [12 11]]
```

In [32]:
```
print(R)
```

```
[[ 8  4]
 [10 12]
 [12 11]]
```

In [33]:
```
A = np.matrix('1 2; 3 4')
B = np.matrix('5 6; 7 8')
L = (A.dot(B)).T
R = (B.T).dot(A.T)
print(L)
```

```
[[19 43]
 [22 50]]
```

In [34]:
```
print(R)
```

```
[[19 43]
 [22 50]]
```

In [35]:
```
A = np.matrix('1 2 3; 4 5 6')
k = 3
L = (k * A).T
R = k * (A.T)
print(L)
```

```
[[ 3 12]
 [ 6 15]
 [ 9 18]]
```

In [36]:
```
print(R)
```

```
[[ 3 12]
 [ 6 15]
 [ 9 18]]
```

In [38]:
```
A = np.matrix('1 2; 3 4')
A_det = np.linalg.det(A)
A_T_det = np.linalg.det(A.T)
print(format(A_det, '.9g'))
```

```
-2
```

In [39]:
```
print(format(A_T_det, '.9g'))
```

```
-2
```

In [40]:
```
A = np.matrix('1 2 3; 4 5 6')
C = 3 * A
print(C)
```

```
[[ 3  6  9]
 [12 15 18]]
```

In [41]:
```
A = np.matrix('1 2; 3 4')
L = 1 * A
R = A
print(L)
```

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

In [42]:
```
print(R)
```

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

In [43]:
```
A = np.matrix('1 2; 3 4')
Z = np.matrix('0 0; 0 0')
L = 0 * A
R = Z
print(L)
```

```
[[0 0]
 [0 0]]
```

In [44]:
```
print(R)
```

```
[[0 0]
 [0 0]]
```

In [45]:
```
A = np.matrix('1 2; 3 4')
p = 2
q = 3
L = (p + q) * A
R = p * A + q * A
print(L)
```

```
[[ 5 10]
 [15 20]]
```

In [46]:
```
print(R)
```

```
[[ 5 10]
 [15 20]]
```

In [47]:
```
A = np.matrix('1 2; 3 4')
p = 2
q = 3
L = (p * q) * A
R = p * (q * A)
print(L)
```

```
[[ 6 12]
 [18 24]]
```

In [48]:
```
print(R)
```

```
[[ 6 12]
 [18 24]]
```

In [49]:
```
A = np.matrix('1 2; 3 4')
B = np.matrix('5 6; 7 8')
k = 3
L = k * (A + B)
R = k * A + k * B
print(L)
```

```
[[18 24]
 [30 36]]
```

In [50]:
```
print(R)
```

```
[[18 24]
 [30 36]]
```

In [51]:
```
A = np.matrix('1 6 3; 8 2 7')
B = np.matrix('8 1 5; 6 9 12')
C = A + B
print(C)
```

```
[[ 9  7  8]
 [14 11 19]]
```

In [52]:
```
A = np.matrix('1 2; 3 4')
B = np.matrix('5 6; 7 8')
L = A + B
R = B + A
print(L)
```

```
[[ 6  8]
 [10 12]]
```

In [53]:
```
print(R)
```

```
[[ 6  8]
 [10 12]]
```

In [54]:
```
A = np.matrix('1 2; 3 4')
B = np.matrix('5 6; 7 8')
C = np.matrix('1 7; 9 3')
L = A + (B + C)
R = (A + B) + C
print(L)
```

```
[[ 7 15]
 [19 15]]
```

In [55]:
```
print(R)
```

```
[[ 7 15]
 [19 15]]
```

In [56]:
```
A = np.matrix('1 2; 3 4')
Z = np.matrix('0 0; 0 0')
L = A + (-1)*A
print(L)
```

```
[[0 0]
 [0 0]]
```

In [57]:
```
print(Z)
```

```
[[0 0]
 [0 0]]
```

In [58]:
```
A = np.matrix('1 2 3; 4 5 6')
B = np.matrix('7 8; 9 1; 2 3')
C = A.dot(B)
print(C)
```

```
[[31 19]
 [85 55]]
```

In [59]:
```
A = np.matrix('1 2; 3 4')
B = np.matrix('5 6; 7 8')
C = np.matrix('2 4; 7 8')
L = A.dot(B.dot(C))
R = (A.dot(B)).dot(C)
print(L)
```

```
[[192 252]
 [436 572]]
```

In [60]:
```
 print(R)
```

```
[[192 252]
 [436 572]]
```

In [61]:
```
A = np.matrix('1 2; 3 4')
B = np.matrix('5 6; 7 8')
C = np.matrix('2 4; 7 8')
L = A.dot(B + C)
R = A.dot(B) + A.dot(C)
print(L)
```

```
[[35 42]
 [77 94]]
```

In [62]:
```
print(R)
```

```
[[35 42]
 [77 94]]
```

In [63]:
```
A = np.matrix('1 2; 3 4')
B = np.matrix('5 6; 7 8')
L = A.dot(B)
R = B.dot(A)
print(L)
```

```
[[19 22]
 [43 50]]
```

In [64]:
```
print(R)
```

```
[[23 34]
 [31 46]]
```

In [65]:
```
A = np.matrix('1 2; 3 4')
E = np.matrix('1 0; 0 1')
L = E.dot(A)
R = A.dot(E)
print(L)
```

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

In [66]:
```
print(R)
```

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

In [67]:
```
print(A)
```

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

In [68]:
```
A = np.matrix('1 2; 3 4')
Z = np.matrix('0 0; 0 0')
L = Z.dot(A)
R = A.dot(Z)
print(L)
```

```
[[0 0]
 [0 0]]
```

In [69]:
```
print(R)
```

```
[[0 0]
 [0 0]]
```

In [70]:
```
print(Z)
```

```
[[0 0]
 [0 0]]
```

In [71]:
```
A = np.matrix('-4 -1 2; 10 4 -1; 8 3 1')
print(A)
```

```
[[-4 -1  2]
 [10  4 -1]
 [ 8  3  1]]
```

In [72]:
```
np.linalg.det(A)
```

Out[72]:
```
-14.000000000000009
```

In [73]:
```
A = np.matrix('-4 -1 2; 10 4 -1; 8 3 1')
print(A)
```

```
[[-4 -1  2]
 [10  4 -1]
 [ 8  3  1]]
```

In [74]:
```
print(A.T)
```

```
[[-4 10  8]
 [-1  4  3]
 [ 2 -1  1]]
```

In [75]:
```
det_A = round(np.linalg.det(A), 3)
det_A_t = round(np.linalg.det(A.T), 3)
print(det_A)
```

```
-14.0
```

In [76]:
```
print(det_A_t)
```

```
-14.0
```

In [77]:
```
A = np.matrix('-4 -1 2; 0 0 0; 8 3 1')
print(A)
```

```
[[-4 -1  2]
 [ 0  0  0]
 [ 8  3  1]]
```

In [78]:
```
np.linalg.det(A)
```

Out[78]:
```
0.0
```

In [79]:
```
A = np.matrix('-4 -1 2; 10 4 -1; 8 3 1')
print(A)
```

```
[[-4 -1  2]
 [10  4 -1]
 [ 8  3  1]]
```

In [80]:
```
B = np.matrix('10 4 -1; -4 -1 2; 8 3 1')
print(B)
```

```
[[10  4 -1]
 [-4 -1  2]
 [ 8  3  1]]
```

In [81]:
```
round(np.linalg.det(A), 3)
```

Out[81]:
```
-14.0
```

In [82]:
```
round(np.linalg.det(B), 3)
```

Out[82]:
```
14.0
```

In [83]:
```
A = np.matrix('-4 -1 2; -4 -1 2; 8 3 1')
print(A)
```

```
[[-4 -1  2]
 [-4 -1  2]
 [ 8  3  1]]
```

In [84]:
```
np.linalg.det(A)
```

Out[84]:
```
0.0
```

In [85]:
```
A = np.matrix('-4 -1 2; 10 4 -1; 8 3 1')
print(A)
```

```
[[-4 -1  2]
 [10  4 -1]
 [ 8  3  1]]
```

In [86]:
```
k = 2
B = A.copy()
B[2, :] = k * B[2, :]
print(B)
```

```
[[-4 -1  2]
 [10  4 -1]
 [16  6  2]]
```

In [87]:
```
det_A = round(np.linalg.det(A), 3)
det_B = round(np.linalg.det(B), 3)
det_A * k
```

Out[87]:
```
-28.0
```

In [88]:
```
det_B
```

Out[88]:
```
-28.0
```

In [89]:
```
A = np.matrix('-4 -1 2; -4 -1 2; 8 3 1')
B = np.matrix('-4 -1 2; 8 3 2; 8 3 1')
C = A.copy()
C[1, :] += B[1, :]
print(C)
```

```
[[-4 -1  2]
 [ 4  2  4]
 [ 8  3  1]]
```

In [90]:
```
print(A)
```

```
[[-4 -1  2]
 [-4 -1  2]
 [ 8  3  1]]
```

In [91]:
```
print(B)
```

```
[[-4 -1  2]
 [ 8  3  2]
 [ 8  3  1]]
```

In [92]:
```
round(np.linalg.det(C), 3)
```

Out[92]:
```
4.0
```

In [93]:
```
round(np.linalg.det(A), 3) + round(np.linalg.det(B), 3)
```

In [93]:
```
round(np.linalg.det(A), 3) + round(np.linalg.det(B), 3)
```

Out[93]:
```
4.0
```

In [94]:
```
A = np.matrix('-4 -1 2; 10 4 -1; 8 3 1')
k = 2
B = A.copy()
B[1, :] = B[1, :] + k * B[0, :]
print(A)
```

```
[[-4 -1  2]
 [10  4 -1]
 [ 8  3  1]]
```

In [95]:
```
print(B)
```

```
[[-4 -1  2]
 [ 2  2  3]
 [ 8  3  1]]
```

In [96]:
```
round(np.linalg.det(A), 3)
```

Out[96]:
```
-14.0
```

In [97]:
```
round(np.linalg.det(B), 3)
```

Out[97]:
```
-14.0
```

In [102]:
```
A = np.matrix('-4 -1 2; 10 4 -1; 8 3 1')
print(A)
k = 2
A[1, :] = A[0, :] + k * A[2, :]
print(A)
round(np.linalg.det(A), 3)
```

```
[[-4 -1  2]
 [10  4 -1]
 [ 8  3  1]]
[[-4 -1  2]
 [12  5  4]
 [ 8  3  1]]
```

Out[102]:
```
0.0
```

In [103]:
```
A = np.matrix('1 -3; 2 5')
A_inv = np.linalg.inv(A)
print(A_inv)
```

```
[[ 0.45454545  0.27272727]
 [-0.18181818  0.09090909]]
```

In [105]:
```
A = np.matrix('1. -3.; 2. 5.')
A_inv = np.linalg.inv(A)
A_inv_inv = np.linalg.inv(A_inv)
print(A)
print(A_inv_inv)
```

```
[[ 1. -3.]
 [ 2.  5.]]
[[ 1. -3.]
 [ 2.  5.]]
```

In [106]:
```
A = np.matrix('1. -3.; 2. 5.')
L = np.linalg.inv(A.T)
R = (np.linalg.inv(A)).T
print(L)
print(R)
```

```
[[ 0.45454545 -0.18181818]
 [ 0.27272727  0.09090909]]
[[ 0.45454545 -0.18181818]
 [ 0.27272727  0.09090909]]
```

In [107]:
```
A = np.matrix('1. -3.; 2. 5.')
B = np.matrix('7. 6.; 1. 8.')
L = np.linalg.inv(A.dot(B))
R = np.linalg.inv(B).dot(np.linalg.inv(A))
print(L)
print(R)
```

```
[[ 0.09454545  0.03272727]
 [-0.03454545  0.00727273]]
[[ 0.09454545  0.03272727]
 [-0.03454545  0.00727273]]
```

In [108]:
```
m_eye = np.eye(4)
print(m_eye)
```

```
[[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]
```

In [109]:
```
rank = np.linalg.matrix_rank(m_eye)
print(rank)
```

```
4
```

In [110]:
```
m_eye[3][3] = 0
print(m_eye)
rank = np.linalg.matrix_rank(m_eye)
print(rank)
```

```
[[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 0.]]
3
```

Задача

In [10]:
```
import numpy as np
```

In [13]:
```
v_hor_np = np.array([8, 9])
print(v_hor_np )
```

```
[8 9]
```

In [14]:
```
v_hor_zeros_v1 = np.zeros((10,))
print(v_hor_zeros_v1 )
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

In [107]:
```
v_hor_zeros_v2 = np.zeros((3, 5))
print(v_hor_zeros_v2 )
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```

In [108]:
```
 v_hor_one_v1 = np.ones((10,))
 print(v_hor_one_v1)
```

```
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

In [17]:
```
v_hor_one_v2 = np.ones((2, 5))
print(v_hor_one_v2)
```

```
[[1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]]
```

In [18]:
```
v_vert_np = np.array([[5], [2]])
print(v_vert_np)
```

```
[[5]
 [2]]
```

In [18]:
```
v_vert_np = np.array([[5], [2]])
print(v_vert_np)
```

```
[[5]
 [2]]
```

In [20]:
```
v_vert_zeros = np.zeros((5, 3))
print(v_vert_zeros)
```

```
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
```

In [21]:
```
v_vert_ones = np.ones((5, 4))
print(v_vert_ones)
```

```
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
```

In [22]:
```
m_sqr_arr = np.array([[2, 3, 4], [5, 6, 7], [8, 9, 10]])
print(m_sqr_arr)
```

```
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]
```

In [23]:
```
m_sqr = [[2, 3, 4], [5, 6, 7], [8, 9, 10]]
m_sqr_arr = np.array(m_sqr)
print(m_sqr_arr)
```

```
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]
```

In [24]:
```
m_sqr_mx = np.matrix([[2, 3, 4], [5, 6, 7], [8, 9, 10]])
print(m_sqr_mx)
```

```
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]
```

In [14]:
```
m_sqr_mx = np.matrix('1 2 3; 4 5 6; 7 8 9')
print(m_sqr_mx)
```

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

In [15]:
```
m_diag = [[1, 0, 0], [0, 5, 0], [0, 0, 9]]
m_diag_np = np.matrix(m_diag)
print(m_diag_np)
```

```
[[1 0 0]
 [0 5 0]
 [0 0 9]]
```

In [26]:
```
m_sqr_mx = np.matrix('2 3 4; 5 6 7; 8 9 10')
print(m_sqr_mx)
```

```
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]
```

In [17]:
```
diag = np.diag(m_sqr_mx)
print(diag)
```

```
[1 5 9]
```

In [18]:
```
m_diag_np = np.diag(np.diag(m_sqr_mx))
print(m_diag_np)
```

```
[[1 0 0]
 [0 5 0]
 [0 0 9]]
```

In [27]:
```
m_e = [[7, 0, 0], [0, 8, 0], [0, 0, 9]]
m_e_np = np.matrix(m_e)
print(m_e_np)
```

```
[[7 0 0]
 [0 8 0]
 [0 0 9]]
```

In [30]:
```
m_eye = np.eye(5)
print(m_eye)
```

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

In [31]:
```
m_idnt = np.identity(5)
print(m_idnt)
```

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

In [32]:
```
m_zeros = np.zeros((4, 4))
print(m_zeros)
```

```
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

In [24]:
```
m_mx = np.matrix('1 2 3; 4 5 6')
print(m_mx)
```

```
[[1 2 3]
 [4 5 6]]
```

In [33]:
```
m_var = np.zeros((3, 6))
print(m_var)
```

```
[[0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]]
```

In [26]:
```
A = np.matrix('1 2 3; 4 5 6')
print(A)
```

```
[[1 2 3]
 [4 5 6]]
```

In [27]:
```
A_t = A.transpose()
print(A_t)
```

```
[[1 4]
 [2 5]
 [3 6]]
```

In [28]:
```
print(A.T)
```

```
[[1 4]
 [2 5]
 [3 6]]
```

In [34]:
```
A = np.matrix('2 3 4; 5 6 7')
print(A)
```

```
[[2 3 4]
 [5 6 7]]
```

In [30]:
```
R = (A.T).T
print(R)
```

```
[[1 2 3]
 [4 5 6]]
```

In [35]:
```
A = np.matrix('3 4 5; 6 7 8')
B = np.matrix('5 6 7; 8 9 0')
L = (A + B).T
R = A.T + B.T
print(L)
```

```
[[ 8 14]
 [10 16]
 [12  8]]
```

In [36]:
```
print(R)
```

```
[[ 8 14]
 [10 16]
 [12  8]]
```

In [37]:
```
A = np.matrix('4 2; 4 4')
B = np.matrix('7 6; 7 8')
L = (A.dot(B)).T
R = (B.T).dot(A.T)
print(L)
```

```
[[42 56]
 [40 56]]
```

In [34]:
```
print(R)
```

```
[[19 43]
 [22 50]]
```

In [38]:
```
A = np.matrix('7 5 3; 4 9 6')
k = 3
L = (k * A).T
R = k * (A.T)
print(L)
```

```
[[21 12]
 [15 27]
 [ 9 18]]
```

In [39]:
```
print(R)
```

```
[[21 12]
 [15 27]
 [ 9 18]]
```

In [40]:
```
A = np.matrix('1 9; 3 9')
A_det = np.linalg.det(A)
A_T_det = np.linalg.det(A.T)
print(format(A_det, '.9g'))
```

```
-18
```

In [41]:
```
print(format(A_T_det, '.9g'))
```

```
-18
```

In [42]:
```
A = np.matrix('9 2 9; 9 5 6')
C = 3 * A
print(C)
```

```
[[27  6 27]
 [27 15 18]]
```

In [43]:
```
A = np.matrix('1 9; 3 9')
L = 1 * A
R = A
print(L)
```

```
[[1 9]
 [3 9]]
```

In [44]:
```
print(R)
```

```
[[1 9]
 [3 9]]
```

In [45]:
```
A = np.matrix('9 9; 9 9')
Z = np.matrix('0 0; 0 0')
L = 0 * A
R = Z
print(L)
```

```
[[0 0]
 [0 0]]
```

In [46]:
```
print(R)
```

```
[[0 0]
 [0 0]]
```

In [47]:
```
A = np.matrix('9 2; 9 4')
p = 2
q = 3
L = (p + q) * A
R = p * A + q * A
print(L)
```

```
[[45 10]
 [45 20]]
```

In [48]:
```
print(R)
```

```
[[45 10]
 [45 20]]
```

In [49]:
```
A = np.matrix('9 2; 9 4')
p = 2
q = 3
L = (p * q) * A
R = p * (q * A)
print(L)
```

```
[[54 12]
 [54 24]]
```

In [50]:
```
print(R)
```

```
[[54 12]
 [54 24]]
```

In [51]:
```
A = np.matrix('9 2; 9 4')
B = np.matrix('5 9; 7 9')
k = 3
L = k * (A + B)
R = k * A + k * B
print(L)
```

```
[[42 33]
 [48 39]]
```

In [52]:
```
print(R)
```

```
[[42 33]
 [48 39]]
```

In [53]:
```
A = np.matrix('1 6 9; 8 2 9')
B = np.matrix('9 1 5; 9 9 12')
C = A + B
print(C)
```

```
[[10  7 14]
 [17 11 21]]
```

In [54]:
```
A = np.matrix('1 2; 3 4')
B = np.matrix('5 6; 7 8')
L = A + B
R = B + A
print(L)
```

```
[[ 6  8]
 [10 12]]
```

In [53]:
```
print(R)
```

```
[[ 6  8]
 [10 12]]
```

In [55]:
```
A = np.matrix('1 9; 3 9')
B = np.matrix('5 6; 7 8')
C = np.matrix('19 9; 9 9')
L = A + (B + C)
R = (A + B) + C
print(L)
```

```
[[25 24]
 [19 26]]
```

In [56]:
```
print(R)
```

```
[[25 24]
 [19 26]]
```

In [57]:
```
A = np.matrix('1 9; 3 9')
Z = np.matrix('0 0; 0 0')
L = A + (-1)*A
print(L)
```

```
[[0 0]
 [0 0]]
```

In [58]:
```
print(Z)
```

```
[[0 0]
 [0 0]]
```

In [59]:
```
A = np.matrix('9 9 3; 9 5 6')
B = np.matrix('7 8; 9 1; 2 3')
C = A.dot(B)
print(C)
```

```
[[150  90]
 [120  95]]
```

In [60]:
```
A = np.matrix('9 9; 9 4')
B = np.matrix('9 6; 7 8')
C = np.matrix('2 4; 9 8')
L = A.dot(B.dot(C))
R = (A.dot(B)).dot(C)
print(L)
```

```
[[1422 1584]
 [ 992 1124]]
```

In [60]:
```
print(R)
```

```
[[192 252]
 [436 572]]
```

In [61]:
```
A = np.matrix('9 2; 3 4')
B = np.matrix('5 6; 9 8')
C = np.matrix('9 4; 7 8')
L = A.dot(B + C)
R = A.dot(B) + A.dot(C)
print(L)
```

```
[[158 122]
 [106  94]]
```

In [62]:
```
print(R)
```

```
[[158 122]
 [106  94]]
```

In [63]:
```
A = np.matrix('9 2; 3 4')
B = np.matrix('5 9; 9 8')
L = A.dot(B)
R = B.dot(A)
print(L)
```

```
[[63 97]
 [51 59]]
```

In [64]:
```
print(R)
```

```
[[ 72  46]
 [105  50]]
```

In [65]:
```
A = np.matrix('9 2; 3 4')
E = np.matrix('1 9; 9 1')
L = E.dot(A)
R = A.dot(E)
print(L)
```

```
[[36 38]
 [84 22]]
```

In [66]:
```
print(R)
```

```
[[27 83]
 [39 31]]
```

In [67]:
```
print(A)
```

```
[[9 2]
 [3 4]]
```

In [68]:
```
A = np.matrix('1 9; 3 9')
Z = np.matrix('0 0; 0 0')
L = Z.dot(A)
R = A.dot(Z)
print(L)
```

```
[[0 0]
 [0 0]]
```

In [69]:
```
print(R)
```

```
[[0 0]
 [0 0]]
```

In [70]:
```
print(Z)
```

```
[[0 0]
 [0 0]]
```

In [71]:
```
A = np.matrix('-4 -1 9; 10 4 -1; 9 3 9')
print(A)
```

```
[[-4 -1  9]
 [10  4 -1]
 [ 9  3  9]]
```

In [72]:
```
np.linalg.det(A)
```

Out[72]: -111.00000000000007

In [73]:
```
A = np.matrix('-4 -1 2; 10 9 -1; 9 3 1')
print(A)
```

```
[[-4 -1  2]
 [10  9 -1]
 [ 9  3  1]]
```

In [74]:
```
print(A.T)
```

```
[[-4 10  9]
 [-1  9  3]
 [ 2 -1  1]]
```

In [75]:
```
det_A = round(np.linalg.det(A), 7)
det_A_t = round(np.linalg.det(A.T), 7)
print(det_A)
```

```
-131.0
```

In [76]:
```
print(det_A_t)
```

```
-131.0
```

In [77]:
```
A = np.matrix('-8 -1 2; 0 0 0; 5 4 2')
print(A)
```

```
[[-8 -1  2]
 [ 0  0  0]
 [ 5  4  2]]
```

In [78]:
```
 np.linalg.det(A)
```

Out[78]: 0.0

In [79]:
```
A = np.matrix('-8 -1 4; 10 8 -1; 8 3 1')
print(A)
```

```
[[-8 -1  4]
 [10  8 -1]
 [ 8  3  1]]
```

In [80]:
```
B = np.matrix('10 4 -5; -4 -1 5; 8 5 1')
print(B)
```

```
[[10  4 -5]
 [-4 -1  5]
 [ 8  5  1]]
```

In [81]:
```
 round(np.linalg.det(A), 3)
```

Out[81]: -206.0

In [82]:
```
round(np.linalg.det(B), 3)
```

Out[82]: -24.0

In [83]:
```
A = np.matrix('-7 -1 2; -7 -1 2; 7 3 1')
print(A)
```

```
[[-7 -1  2]
 [-7 -1  2]
 [ 7  3  1]]
```

In [84]:
```
np.linalg.det(A)
```

Out[84]: 0.0

In [85]:
```
A = np.matrix('-8 -1 8; 10 8 -1; 8 8 1')
print(A)
```

```
[[-8 -1  8]
 [10  8 -1]
 [ 8  8  1]]
```

In [86]:
```
k = 2
B = A.copy()
B[2, :] = k * B[2, :]
print(B)
```

```
[[-8 -1  8]
 [10  8 -1]
 [16 16  2]]
```

In [87]:
```
det_A = round(np.linalg.det(A), 7)
det_B = round(np.linalg.det(B), 7)
det_A * k
```

Out[87]: 36.0

In [88]:
```
det_B
```

Out[88]: 36.0

In [89]:
```
A = np.matrix('-6 -1 6; -4 -6 2; 8 3 1')
B = np.matrix('-6 -1 6; 6 3 2; 6 3 1')
C = A.copy()
C[1, :] += B[1, :]
print(C)
```

```
[[-6 -1  6]
 [ 2 -3  4]
 [ 8  3  1]]
```

In [90]:
```
print(A)
```

```
[[-6 -1  6]
 [-4 -6  2]
 [ 8  3  1]]
```

In [91]:
```
print(B)
```

```
[[-6 -1  6]
 [ 6  3  2]
 [ 6  3  1]]
```

In [92]:
```
round(np.linalg.det(C), 7)
```

Out[92]: 240.0

In [93]:
```
round(np.linalg.det(A), 7) + round(np.linalg.det(B), 7)
```

Out[93]: 280.0

In [94]:
```
A = np.matrix('-7 -7 2; 10 7 -1; 8 3 7')
k = 2
B = A.copy()
B[1, :] = B[1, :] + k * B[0, :]
print(A)
```

```
[[-7 -7  2]
 [10  7 -1]
 [ 8  3  7]]
```

In [95]:
```
print(B)
```

```
[[-7 -7  2]
 [-4 -7  3]
 [ 8  3  7]]
```

In [96]:
```
round(np.linalg.det(A), 3)
```

Out[96]: 130.0

In [97]:
```
round(np.linalg.det(B), 3)
```

Out[97]: 130.0

In [98]:
```
A = np.matrix('-6 -1 6; 10 4 -6; 8 3 6')
print(A)
k = 2
A[1, :] = A[0, :] + k * A[2, :]
print(A)
round(np.linalg.det(A), 3)
```

```
[[-6 -1  6]
 [10  4 -6]
 [ 8  3  6]]
[[-6 -1  6]
 [10  5 18]
 [ 8  3  6]]
```

Out[98]: 0.0

In [100]:
```
A = np.matrix('1 -5; 2 5')
A_inv = np.linalg.inv(A)
print(A_inv)
```

```
[[ 0.33333333  0.33333333]
 [-0.13333333  0.06666667]]
```

In [101]:
```
A = np.matrix('1. -5.; 2. 5.')
A_inv = np.linalg.inv(A)
A_inv_inv = np.linalg.inv(A_inv)
print(A)
print(A_inv_inv)
```

```
[[ 1. -5.]
 [ 2.  5.]]
[[ 1. -5.]
 [ 2.  5.]]
```

In [102]:
```
A = np.matrix('1. -5.; 2. 5.')
L = np.linalg.inv(A.T)
R = (np.linalg.inv(A)).T
print(L)
print(R)
```

```
[[ 0.33333333 -0.13333333]
 [ 0.33333333  0.06666667]]
[[ 0.33333333 -0.13333333]
 [ 0.33333333  0.06666667]]
```

In [103]:
```
A = np.matrix('1. -5.; 2. 5.')
B = np.matrix('7. 5.; 5. 8.')
L = np.linalg.inv(A.dot(B))
R = np.linalg.inv(B).dot(np.linalg.inv(A))
print(L)
print(R)
```

```
[[ 0.10752688  0.07526882]
 [-0.08387097 -0.03870968]]
[[ 0.10752688  0.07526882]
 [-0.08387097 -0.03870968]]
```

In [104]:
```
m_eye = np.eye(9)
print(m_eye)
```

```
[[1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 1.]]
```

In [105]:
```
rank = np.linalg.matrix_rank(m_eye)
print(rank)
```

```
9
```

In [106]:
```
m_eye[7][7] = 0
print(m_eye)
rank = np.linalg.matrix_rank(m_eye)
print(rank)
```

```
[[1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 1.]]
8
```

Самостоятельно задание

Матричный метод:

In [15]:
```
import numpy as np
A = [[15, 25, 35],[9, 8, 7],[8, 12, 10]]
B = [12, 13, 14]
OM = np.linalg.inv(A)
X2 = np.matmul(OM, B)
print('X2 = ',X2)
```

```
X2 =  [ 1.0984127   1.33650794 -1.08253968]
```

Метод Крамера:

In [30]:
```
import numpy as np
A1 = [[15, 25, 35],[9, 8,7],[18, 12, 10]]
B1 = [12, 13, 14]
def Kram(A,B):
    m = len(A)
    op = np.linalg.det(A)
    r = list()
    for i in range(m):
        VM=np.copy(A)
        VM[:,i]=B
        r.append(np.linalg.det(VM)/op)
    return r

X3 = Kram(A1,B1)
print(X3)
print(np.matmul(A, X3))
```

```
[-1.6476190476190486, 6.828571428571435, -3.8285714285714296]
[12.         13.         30.47619048]
```