In [105…
```
import numpy as np
```

# A

In [106…
```
a = np.array([1,2,3])[:,None]
b = np.array([4,5,6])[:,None]
c = np.array([-1,1,3])[:,None]
```

## 1

In [107…
```
2*a-b
```

Out[107]:
```
array([[-2],
       [-1],
       [ 0]])
```

## 2

In [108…
```
a/np.linalg.norm(a)
```

Out[108]:
```
array([[0.26726124],
       [0.53452248],
       [0.80178373]])
```

## 3

In [109…
```
np.linalg.norm(a)
```

Out[109]:
```
3.7416573867739413
```

In [110…
```
i = np.array([1,0,0])[:,None]
j = np.array([0,1,0])[:,None]
k = np.array([0,0,1])[:,None]
```

In [111…
```
np.arccos(np.dot(np.transpose(a),i)/np.linalg.norm(a))
```

Out[111]:
```
array([[1.30024656]])
```

## 4

In [112…
```
np.arccos(np.dot(np.transpose(a),i)/np.linalg.norm(a))
```

Out[112]:
```
array([[1.30024656]])
```

In [113…
```
np.arccos(np.dot(np.transpose(a),j)/np.linalg.norm(a))
```

Out[113]:
```
array([[1.00685369]])
```

In [114…
```
np.arccos(np.dot(np.transpose(a),k)/np.linalg.norm(a))
```

Out[114]:
```
array([[0.64052231]])
```

## 5

In [115…
```
np.arccos(np.dot(np.transpose(a),b)/np.linalg.norm(a)/np.linalg.norm(b))
```

Out[115]:
```
array([[0.22572613]])
```

## 6

In [116…
```
np.dot(np.transpose(a),b)
```

Out[116]:
```
array([[32]])
```

In [117…
```
np.dot(np.transpose(b),a)
```

Out[117]:
```
array([[32]])
```

## 7

In [118…
```
np.linalg.norm(a)*np.linalg.norm(b)*np.cos(np.arccos(np.dot(np.transpose(a),b)/np.linalg.norm(a)/np.linalg.norm(b)))
```

Out[118]:
```
array([[32.]])
```

## 8

In [119…
```
np.dot(np.transpose(b),a/np.linalg.norm(a))
```

Out[119]:
```
array([[8.55235974]])
```

## 9

In [120…
```
np.dot(np.array([0,-3,2]),a)
```
Out[120]:
```
array([0])
```

## 10

In [121…
```
np.cross(np.transpose(a),np.transpose(b))
```
Out[121]:
```
array([[-3,  6, -3]])
```

In [122…
```
np.cross(np.transpose(b),np.transpose(a))
```
Out[122]:
```
array([[ 3, -6,  3]])
```

## 11

In [123…
```
np.cross(np.transpose(a),np.transpose(b))
```
Out[123]:
```
array([[-3,  6, -3]])
```

In [124…
```
np.dot(np.cross(np.transpose(a),np.transpose(b)),a)
```
Out[124]:
```
array([[0]])
```

In [125…
```
np.dot(np.cross(np.transpose(a),np.transpose(b)),b)
```
Out[125]:
```
array([[0]])
```

## 12

In [126…
```
b-3*a+c
```
Out[126]:
```
array([[0],
       [0],
       [0]])
```

## 13

In [127…
```
np.dot(np.transpose(a),b)
```
Out[127]:
```
array([[32]])
```

In [128…
```
np.dot(a,np.transpose(b))
```
Out[128]:
```
array([[ 4,  5,  6],
       [ 8, 10, 12],
       [12, 15, 18]])
```

# B

In [129…
```
A = np.array([[1,2,3],[4,-2, 3],[0,5,-1]])
B = np.array([[1,2,1],[2,1,-4],[3,-2,1]])
C = np.array([[1,2,3],[4,5,6],[-1,1,3]])
d = np.array([1,2,3])[:,None]
```

## 1

In [130…
```
2*A-B
```
Out[130]:
```
array([[ 1,  2,  5],
       [ 6, -5, 10],
       [-3, 12, -3]])
```

## 2

In [131…
```
np.matmul(A,B)
```
Out[131]:
```
array([[ 14,  -2,  -4],
       [  9,   0,  15],
       [  7,   7, -21]])
```

In [132…
```
np.matmul(B,A)
```
Out[132]:
```
array([[  9,   3,   8],
       [  6, -18,  13],
       [ -5,  15,   2]])
```

## 3

In [133…
```
np.transpose(np.matmul(A,B))
```
Out[133]:
```
array([[ 14,   9,   7],
       [ -2,   0,   7],
       [ -4,  15, -21]])
```

In [134…
```
np.matmul(np.transpose(B),np.transpose(A))
```
Out[134]:
```
array([[ 14,   9,   7],
       [ -2,   0,   7],
       [ -4,  15, -21]])
```

## 4

In [135…
```
np.linalg.det(A)
```
Out[135]:
```
54.999999999999964
```

In [136…
```
np.exp(np.linalg.slogdet(A)[1])
```
Out[136]:
```
54.999999999999964
```

## 5

In [137…
```
np.linalg.det(C)
```
Out[137]:
```
7.494005416219837e-16
```

In [138…
```
np.linalg.det(B)
```
Out[138]:
```
-42.00000000000002
```

In [139…
```
np.matmul(B[0,:],B[1,:])
```
Out[139]:
```
0
```

In [140…
```
np.matmul(B[0,:],B[2,:])
```
Out[140]:
```
0
```

In [141…
```
np.matmul(B[1,:],B[2,:])
```
Out[141]:
```
0
```

## 6

In [142…
```
np.linalg.inv(A)
```
Out[142]:
```
array([[-0.23636364,  0.30909091,  0.21818182],
       [ 0.07272727, -0.01818182,  0.16363636],
       [ 0.36363636, -0.09090909, -0.18181818]])
```

In [143…
```
np.linalg.inv(B)
```
Out[143]:
```
array([[ 0.16666667,  0.0952381 ,  0.21428571],
       [ 0.33333333,  0.04761905, -0.14285714],
       [ 0.16666667, -0.19047619,  0.07142857]])
```

## 7

In [144…
```
np.linalg.inv(C)
```
Out[144]:
```
array([[ 1.20095990e+16, -4.00319967e+15, -4.00319967e+15],
       [-2.40191980e+16,  8.00639934e+15,  8.00639934e+15],
       [ 1.20095990e+16, -4.00319967e+15, -4.00319967e+15]])
```

## 8

In [145…
```
np.matmul(A,d)
```
Out[145]:
```
array([[14],
       [ 9],
       [ 7]])
```

## 9

In [146…
```
np.matmul(A,d)/np.linalg.norm(d)
```
Out[146]:
```
array([[3.74165739],
       [2.40535118],
       [1.87082869]])
```

## 10

In [147…
```
np.matmul(A,np.matmul(d,np.transpose(d)))/np.linalg.norm(d)
```
Out[147]:
```
array([[ 3.74165739,  7.48331477, 11.22497216],
       [ 2.40535118,  4.81070235,  7.21605353],
       [ 1.87082869,  3.74165739,  5.61248608]])
```

## 11

In [148…
```
np.matmul(A,d)
```
Out[148]:
```
array([[14],
       [ 9],
       [ 7]])
```

## 12

In [149…
```
np.matmul(np.linalg.inv(B),d)
```

Out[149]:
```
array([[1.00000000e+00],
       [2.77555756e-17],
       [0.00000000e+00]])
```

In [150…
```
np.linalg.solve(B,d)
```

Out[150]:
```
array([[ 1.00000000e+00],
       [ 2.37904934e-17],
       [-1.18952467e-17]])
```

## 13

In [151…
```
np.linalg.solve(C,d)
```

Out[151]:
```
array([[-8.00639934e+15],
       [ 1.60127987e+16],
       [-8.00639934e+15]])
```

# C

In [152…
```
D = np.array([[1,2],[3,2]])
E = np.array([[2,-2],[-2,5]])
F = np.array([[1,2],[2,4]])
```

## 1

In [153…
```
np.linalg.eig(D)
```

Out[153]:
```
(array([-1.,  4.]),
 array([[-0.70710678, -0.5547002 ],
        [ 0.70710678, -0.83205029]]))
```

## 2

In [155…
```
np.dot(np.linalg.eig(D)[1][:,0],np.linalg.eig(D)[1][:,1])
```

Out[155]:
```
-0.19611613513818388
```

## 3

In [157…
```
np.dot(np.linalg.eig(E)[1][:,0],np.linalg.eig(E)[1][:,1])
```

Out[157]:
```
0.0
```

## 4

Since $E$ is symmtric, it's eigenvectors are orthogonal. Proofs are in previous part.

## 5

In [163…
```
np.matmul(F,np.array([0,0]))
```

Out[163]:
```
array([0, 0])
```

## 6

In [161…
```
np.matmul(F,np.array([2,-1]))
```

Out[161]:
```
array([0, 0])
```

In [162…
```
np.matmul(F,np.array([4,-2]))
```

Out[162]:
```
array([0, 0])
```

## 7

In [164…
```
np.matmul(D,np.array([0,0]))
```

Out[164]:
```
array([0, 0])
```

In [ ]: