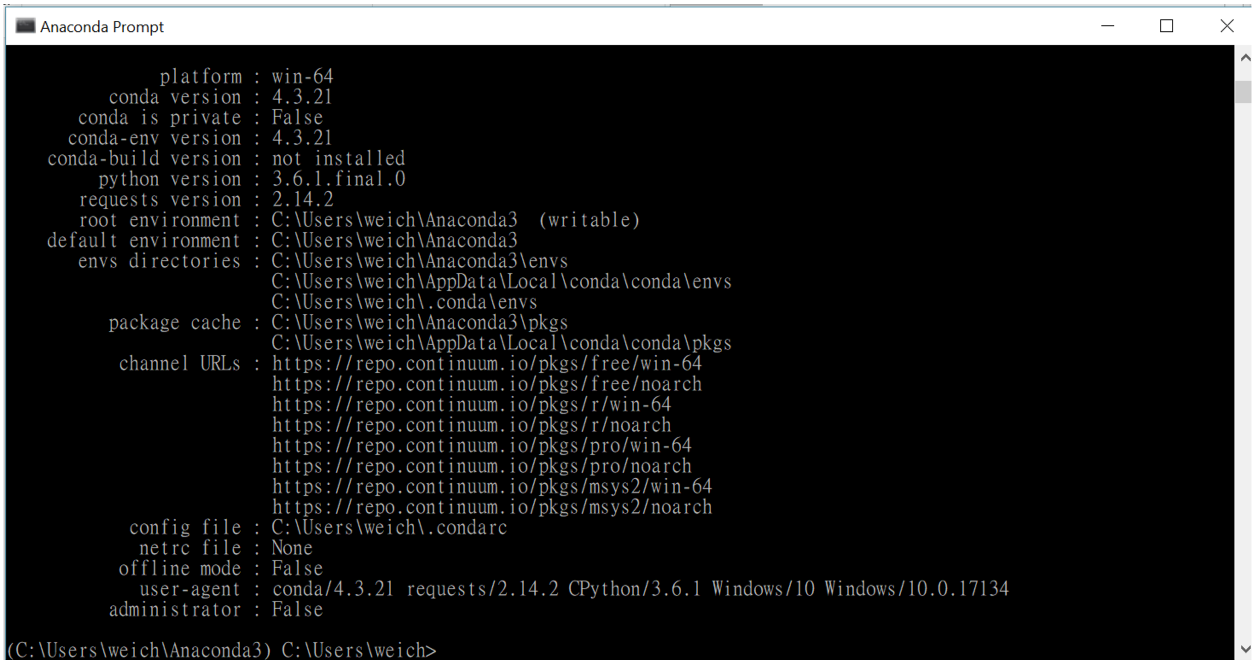
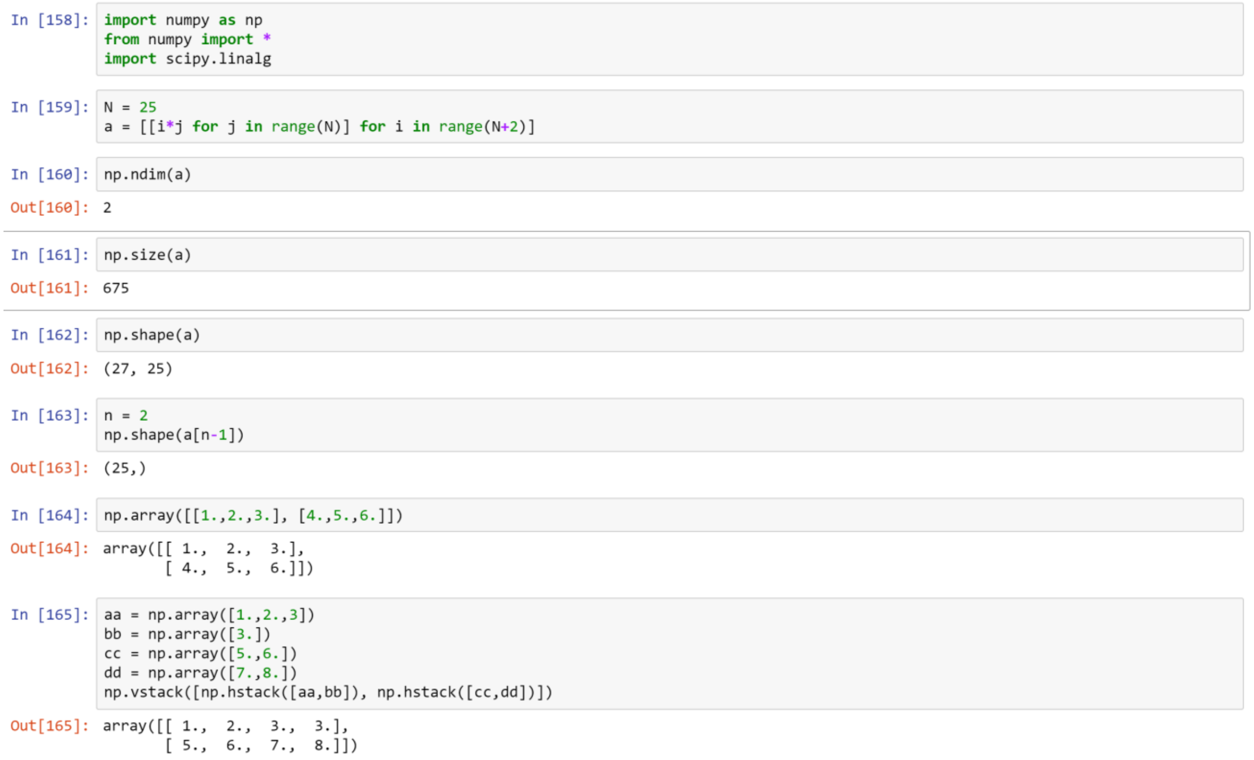
Task 1.



```
             platform : win-64
        conda version : 4.3.21
     conda is private : False
    conda-env version : 4.3.21
  conda-build version : not installed
       python version : 3.6.1.final.0
     requests version : 2.14.2
     root environment : C:\Users\weich\Anaconda3  (writable)
  default environment : C:\Users\weich\Anaconda3
     envs directories : C:\Users\weich\Anaconda3\envs
                        C:\Users\weich\AppData\Local\conda\conda\envs
                        C:\Users\weich\.conda\envs
        package cache : C:\Users\weich\Anaconda3\pkgs
                        C:\Users\weich\AppData\Local\conda\conda\pkgs
         channel URLs : https://repo.continuum.io/pkgs/free/win-64
                        https://repo.continuum.io/pkgs/free/noarch
                        https://repo.continuum.io/pkgs/r/win-64
                        https://repo.continuum.io/pkgs/r/noarch
                        https://repo.continuum.io/pkgs/pro/win-64
                        https://repo.continuum.io/pkgs/pro/noarch
                        https://repo.continuum.io/pkgs/msys2/win-64
                        https://repo.continuum.io/pkgs/msys2/noarch
          config file : C:\Users\weich\.condarc
           netrc file : None
         offline mode : False
           user-agent : conda/4.3.21 requests/2.14.2 CPython/3.6.1 Windows/10 Windows/10.0.17134
        administrator : False

(C:\Users\weich\Anaconda3) C:\Users\weich>
```

Task 2.



```
In [158]: import numpy as np
          from numpy import *
          import scipy.linalg

In [159]: N = 25
          a = [[i*j for j in range(N)] for i in range(N+2)]

In [160]: np.ndim(a)

Out[160]: 2

In [161]: np.size(a)

Out[161]: 675

In [162]: np.shape(a)

Out[162]: (27, 25)

In [163]: n = 2
          np.shape(a[n-1])

Out[163]: (25,)

In [164]: np.array([[1.,2.,3.], [4.,5.,6.]])

Out[164]: array([[ 1.,  2.,  3.],
                 [ 4.,  5.,  6.]])

In [165]: aa = np.array([1.,2.,3])
          bb = np.array([3.])
          cc = np.array([5.,6.])
          dd = np.array([7.,8.])
          np.vstack([np.hstack([aa,bb]), np.hstack([cc,dd])])

Out[165]: array([[ 1.,  2.,  3.,  3.],
                 [ 5.,  6.,  7.,  8.]])
```

```
In [166]: a = range(5)
          a[-1]

Out[166]: 4

In [167]: a = np.array([[[i+j,i*j] for j in range(10)] for i in range(10)])
          a[1,4]

Out[167]: array([5, 4])

In [168]: a[1]

Out[168]: array([[ 1,  0],
                 [ 2,  1],
                 [ 3,  2],
                 [ 4,  3],
                 [ 5,  4],
                 [ 6,  5],
                 [ 7,  6],
                 [ 8,  7],
                 [ 9,  8],
                 [10,  9]])

In [177]: a = np.array([[i+j for j in range(5)] for i in range(5)])
          a[0:5]

Out[177]: array([[0, 1, 2, 3, 4],
                 [1, 2, 3, 4, 5],
                 [2, 3, 4, 5, 6],
                 [3, 4, 5, 6, 7],
                 [4, 5, 6, 7, 8]])

In [178]: a[-5:]

Out[178]: array([[0, 1, 2, 3, 4],
                 [1, 2, 3, 4, 5],
                 [2, 3, 4, 5, 6],
                 [3, 4, 5, 6, 7],
                 [4, 5, 6, 7, 8]])

In [179]: a[0:3][:,4:9]

Out[179]: array([[4],
                 [5],
                 [6]])

In [180]: a[ix_([1,3,4],[0,2])]

Out[180]: array([[1, 3],
                 [3, 5],
                 [4, 6]])

In [181]: a[2:21:2,:]

Out[181]: array([[2, 3, 4, 5, 6],
                 [4, 5, 6, 7, 8]])

In [182]: a[::2,:]

Out[182]: array([[0, 1, 2, 3, 4],
                 [2, 3, 4, 5, 6],
                 [4, 5, 6, 7, 8]])

In [183]: a[::-1,:]

Out[183]: array([[4, 5, 6, 7, 8],
                 [3, 4, 5, 6, 7],
                 [2, 3, 4, 5, 6],
                 [1, 2, 3, 4, 5],
                 [0, 1, 2, 3, 4]])

In [184]: a[r_[:len(a),0]]

Out[184]: array([[0, 1, 2, 3, 4],
                 [1, 2, 3, 4, 5],
                 [2, 3, 4, 5, 6],
                 [3, 4, 5, 6, 7],
                 [4, 5, 6, 7, 8],
                 [0, 1, 2, 3, 4]])
```

In [206]:

```
a = np.array([[i-2*j for j in range(5)] for i in range(5)])
print(a)
```

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

In [207]:

```
a.transpose()
```

Out[207]:

```
array([[ 0,  1,  2,  3,  4],
       [-2, -1,  0,  1,  2],
       [-4, -3, -2, -1,  0],
       [-6, -5, -4, -3, -2],
       [-8, -7, -6, -5, -4]])
```

In [209]:

```
a = np.array([[i-2j for k in range(5)] for i in range(5)])
print(a)
```

```
[[ 0.-2.j  0.-2.j  0.-2.j  0.-2.j  0.-2.j]
 [ 1.-2.j  1.-2.j  1.-2.j  1.-2.j  1.-2.j]
 [ 2.-2.j  2.-2.j  2.-2.j  2.-2.j  2.-2.j]
 [ 3.-2.j  3.-2.j  3.-2.j  3.-2.j  3.-2.j]
 [ 4.-2.j  4.-2.j  4.-2.j  4.-2.j  4.-2.j]]
```

In [210]:

```
a.conj().transpose()
```

Out[210]:

```
array([[ 0.+2.j,  1.+2.j,  2.+2.j,  3.+2.j,  4.+2.j],
       [ 0.+2.j,  1.+2.j,  2.+2.j,  3.+2.j,  4.+2.j],
       [ 0.+2.j,  1.+2.j,  2.+2.j,  3.+2.j,  4.+2.j],
       [ 0.+2.j,  1.+2.j,  2.+2.j,  3.+2.j,  4.+2.j],
       [ 0.+2.j,  1.+2.j,  2.+2.j,  3.+2.j,  4.+2.j]])
```

In [226]:

```
a = np.array([1, -1, 3])
b = np.array([2, 1, 4])
```

In [227]:

```
a@b
```

Out[227]:

```
13
```

In [228]:

```
a*b
```

Out[228]:

```
array([ 2, -1, 12])
```

In [229]:

```
a/b
```

Out[229]:

```
array([ 0.5 , -1.  ,  0.75])
```

In [230]:

```
a**3
```

Out[230]:

```
array([ 1, -1, 27], dtype=int32)
```

In [231]:

```
(a>0.5)
```

Out[231]:

```
array([ True, False,  True], dtype=bool)
```

In [234]:

```
nonzero(a>0.5)
```

Out[234]:

```
(array([0, 2], dtype=int64),)
```

In [255]:

```
v = np.array([0.3, 0.51, 1, -1, 3])
a = np.array([[1, 2, 3, 4, 5],[4, 5, 6, 7, 8]])
a[:,nonzero(v>0.5)[0]]
```

Out[255]:

```
array([[2, 3, 5],
       [5, 6, 8]])
```

In [257]:

```
a[:,v.transpose()>0.5]
```

Out[257]:

```
array([[2, 3, 5],
       [5, 6, 8]])
```

In [260]:

```
a = a/5
a[a<0.5] = 0
print(a)
```

```
[[ 0.   0.   0.6  0.8  1. ]
 [ 0.8  1.   1.2  1.4  1.6]]
```

In [264]:
```
a = np.array([[1, 2, 3, 4, 5],[4, 5, 6, 7, 8]])/5
a*(a>0.5)
```

Out[264]:
```
array([[ 0. ,  0. ,  0.6,  0.8,  1. ],
       [ 0.8,  1. ,  1.2,  1.4,  1.6]])
```

In [266]:
```
a[:] = 3
print(a)
```

```
[[ 3.  3.  3.  3.  3.]
 [ 3.  3.  3.  3.  3.]]
```

In [269]:
```
x = np.array([[3, 4, 2, 1, 6],[2, 5, 8, 9, 1]])
print(x)
```

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

In [271]:
```
y = x.copy()
print(y)
```

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

In [275]:
```
y = x[1,:].copy()
print(y)
```

```
[2 5 8 9 1]
```

In [276]:
```
y = x.flatten()
print(y)
```

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

In [278]:
```
arange(1.,11.)
```

Out[278]:
```
array([  1.,   2.,   3.,   4.,   5.,   6.,   7.,   8.,   9.,  10.])
```

In [279]:
```
arange(10.)
```

Out[279]:
```
array([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9.])
```

In [280]:
```
arange(1.,11.)[:, newaxis]
```

Out[280]:
```
array([[  1.],
       [  2.],
       [  3.],
       [  4.],
       [  5.],
       [  6.],
       [  7.],
       [  8.],
       [  9.],
       [ 10.]])
```

In [282]:
```
zeros((3,4))
```

Out[282]:
```
array([[ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.]])
```

In [283]:
```
zeros((3,4,5))
```

Out[283]:
```
array([[[ 0.,  0.,  0.,  0.,  0.],
        [ 0.,  0.,  0.,  0.,  0.],
        [ 0.,  0.,  0.,  0.,  0.],
        [ 0.,  0.,  0.,  0.,  0.]],

       [[ 0.,  0.,  0.,  0.,  0.],
        [ 0.,  0.,  0.,  0.,  0.],
        [ 0.,  0.,  0.,  0.,  0.],
        [ 0.,  0.,  0.,  0.,  0.]],

       [[ 0.,  0.,  0.,  0.,  0.],
        [ 0.,  0.,  0.,  0.,  0.],
        [ 0.,  0.,  0.,  0.,  0.],
        [ 0.,  0.,  0.,  0.,  0.]]])
```

In [284]:
```
ones((3,4))
```

Out[284]:
```
array([[ 1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.]])
```

In [286]:
```
eye(3)
```
Out[286]:
```
array([[ 1.,  0.,  0.],
       [ 0.,  1.,  0.],
       [ 0.,  0.,  1.]])
```

In [290]:
```
a = np.array([[i+j for j in range(4)] for i in range(4)])
diag(a)
```
Out[290]:
```
array([0, 2, 4, 6])
```

In [291]:
```
diag(a,0)
```
Out[291]:
```
array([0, 2, 4, 6])
```

In [292]:
```
random.rand(3,4)
```
Out[292]:
```
array([[ 0.77242161,  0.49864226,  0.1507609 ,  0.12199909],
       [ 0.94463047,  0.2984594 ,  0.63201078,  0.42345558],
       [ 0.77953453,  0.0951252 ,  0.47119294,  0.14649787]])
```

In [293]:
```
linspace(1,3,4)
```
Out[293]:
```
array([ 1.        ,  1.66666667,  2.33333333,  3.        ])
```

In [294]:
```
mgrid[0:9.,0:6.]
```
Out[294]:
```
array([[[ 0.,  0.,  0.,  0.,  0.,  0.],
        [ 1.,  1.,  1.,  1.,  1.,  1.],
        [ 2.,  2.,  2.,  2.,  2.,  2.],
        [ 3.,  3.,  3.,  3.,  3.,  3.],
        [ 4.,  4.,  4.,  4.,  4.,  4.],
        [ 5.,  5.,  5.,  5.,  5.,  5.],
        [ 6.,  6.,  6.,  6.,  6.,  6.],
        [ 7.,  7.,  7.,  7.,  7.,  7.],
        [ 8.,  8.,  8.,  8.,  8.,  8.]],

       [[ 0.,  1.,  2.,  3.,  4.,  5.],
        [ 0.,  1.,  2.,  3.,  4.,  5.],
        [ 0.,  1.,  2.,  3.,  4.,  5.],
        [ 0.,  1.,  2.,  3.,  4.,  5.],
        [ 0.,  1.,  2.,  3.,  4.,  5.],
        [ 0.,  1.,  2.,  3.,  4.,  5.],
        [ 0.,  1.,  2.,  3.,  4.,  5.],
        [ 0.,  1.,  2.,  3.,  4.,  5.],
        [ 0.,  1.,  2.,  3.,  4.,  5.]]])
```

In [295]:
```
ogrid[0:9.,0:6.]
```
Out[295]:
```
[array([[ 0.],
        [ 1.],
        [ 2.],
        [ 3.],
        [ 4.],
        [ 5.],
        [ 6.],
        [ 7.],
        [ 8.]]), array([[ 0.,  1.,  2.,  3.,  4.,  5.]])]
```

```
In [296]: meshgrid([1, 2, 4], [2, 4, 5])

Out[296]: [array([[1, 2, 4],
                  [1, 2, 4],
                  [1, 2, 4]]), array([[2, 2, 2],
                  [4, 4, 4],
                  [5, 5, 5]])]
```

```
In [297]: ix_([1, 2, 4], [2, 4, 5])

Out[297]: (array([[1],
                  [2],
                  [4]]), array([[2, 4, 5]]))
```

```
In [301]: a = eye(2)
          m = 3
          n = 2
          tile(a, (m, n))

Out[301]: array([[ 1.,  0.,  1.,  0.],
                 [ 0.,  1.,  0.,  1.],
                 [ 1.,  0.,  1.,  0.],
                 [ 0.,  1.,  0.,  1.],
                 [ 1.,  0.,  1.,  0.],
                 [ 0.,  1.,  0.,  1.]])
```

```
In [308]: a = eye(3)
          b = ones((3,3))
          hstack((a,b))

Out[308]: array([[ 1.,  0.,  0.,  1.,  1.,  1.],
                 [ 0.,  1.,  0.,  1.,  1.,  1.],
                 [ 0.,  0.,  1.,  1.,  1.,  1.]])
```

```
In [309]: vstack((a,b))

Out[309]: array([[ 1.,  0.,  0.],
                 [ 0.,  1.,  0.],
                 [ 0.,  0.,  1.],
                 [ 1.,  1.,  1.],
                 [ 1.,  1.,  1.],
                 [ 1.,  1.,  1.]])
```

```
In [323]: a = random.rand(4,4)
          print(a)

          [[ 0.53756815  0.92015113  0.90639546  0.43538416]
           [ 0.09714128  0.04831264  0.22842457  0.88752752]
           [ 0.56190655  0.35613301  0.53642345  0.67068431]
           [ 0.53320405  0.02730631  0.70982818  0.99345332]]
```

```
In [324]: a.max()

Out[324]: 0.9934533225885247
```

```
In [325]: a.max(0)

Out[325]: array([ 0.56190655,  0.92015113,  0.90639546,  0.99345332])
```

```
In [326]: a.max(1)

Out[326]: array([ 0.92015113,  0.88752752,  0.67068431,  0.99345332])
```

```
In [327]: b = eye(4)
          maximum(a,b)

Out[327]: array([[ 1.        ,  0.92015113,  0.90639546,  0.43538416],
                 [ 0.09714128,  1.        ,  0.22842457,  0.88752752],
                 [ 0.56190655,  0.35613301,  1.        ,  0.67068431],
                 [ 0.53320405,  0.02730631,  0.70982818,  1.        ]])
```

```
In [328]: np.linalg.norm(a)

Out[328]: 2.4401676396914533
```

```
In [329]: logical_and(a,b)

Out[329]: array([[ True, False, False, False],
                 [False,  True, False, False],
                 [False, False,  True, False],
                 [False, False, False,  True]], dtype=bool)
```

```
In [330]: logical_or(a,b)

Out[330]: array([[ True,  True,  True,  True],
                 [ True,  True,  True,  True],
                 [ True,  True,  True,  True],
                 [ True,  True,  True,  True]], dtype=bool)
```

```
In [336]: a = 1
          b = 0
          a & b

Out[336]: 0
```

```
In [337]: a | b

Out[337]: 1
```

```
In [344]: a = np.array([[1,2], [3,4]])
          linalg.inv(a)

Out[344]: array([[-2. ,  1. ],
                 [ 1.5, -0.5]])
```

```
In [346]: linalg.pinv(a**2)

Out[346]: array([[-0.8 ,  0.2 ],
                 [ 0.45, -0.05]])
```

```
In [347]: linalg.matrix_rank(a)

Out[347]: 2
```

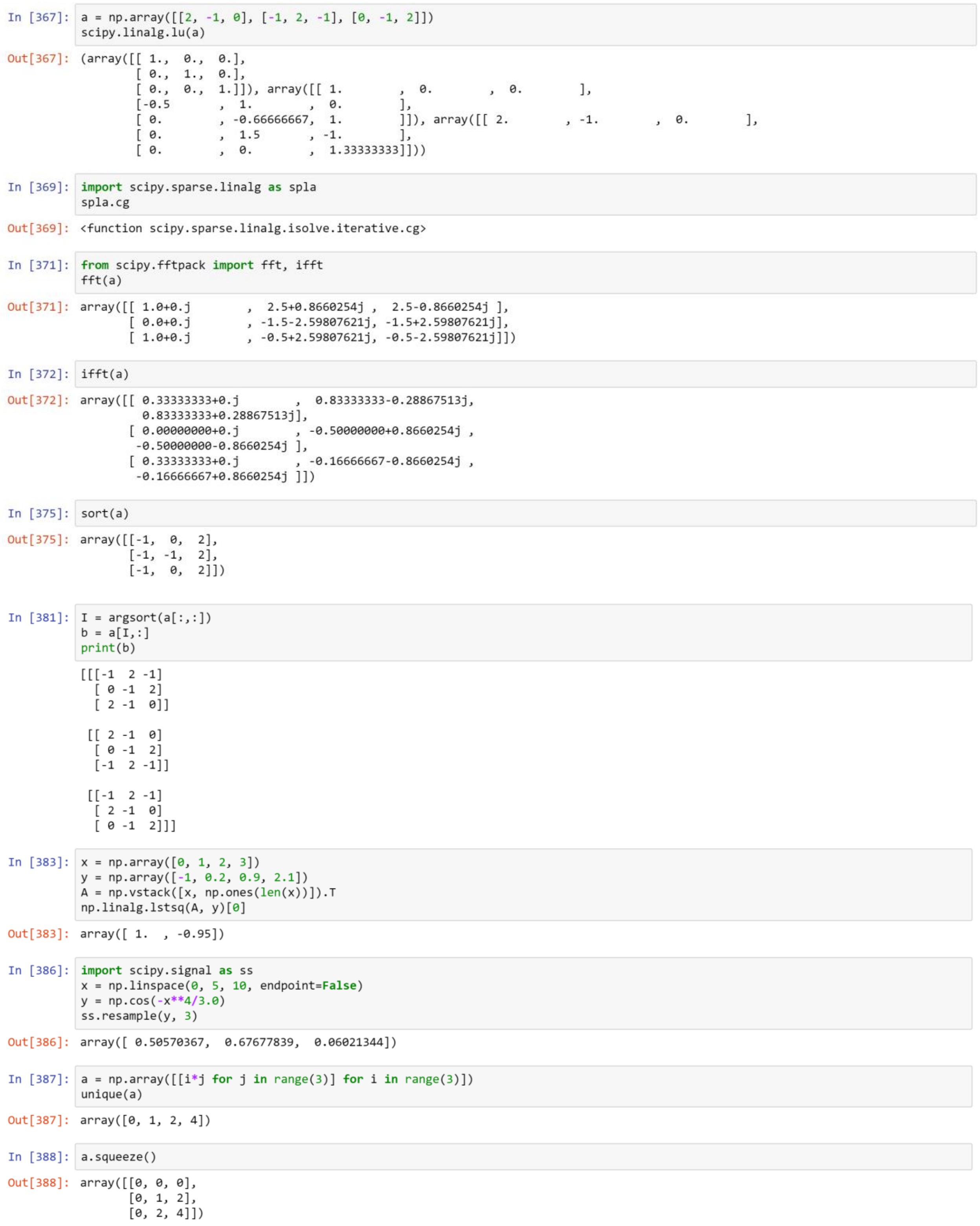
```
In [348]: b = np.array([3,4])
          linalg.solve(a,b)

Out[348]: array([-2. ,  2.5])
```

```
In [353]: U, S, Vh = linalg.svd(a)
          print(U)
          print(S)
          print(Vh)
          print(Vh.transpose())

          [[-0.40455358 -0.9145143 ]
           [-0.9145143   0.40455358]]
          [ 5.4649857   0.36596619]
          [[-0.57604844 -0.81741556]
           [ 0.81741556 -0.57604844]]
          [[-0.57604844  0.81741556]
           [-0.81741556 -0.57604844]]
```

```
In [355]: a = np.array([[2, -1, 0], [-1, 2, -1], [0, -1, 2]])
          linalg.cholesky(a).transpose()

Out[355]: array([[ 1.41421356, -0.70710678,  0.        ],
                 [ 0.        ,  1.22474487, -0.81649658],
                 [ 0.        ,  0.        ,  1.15470054]])
```

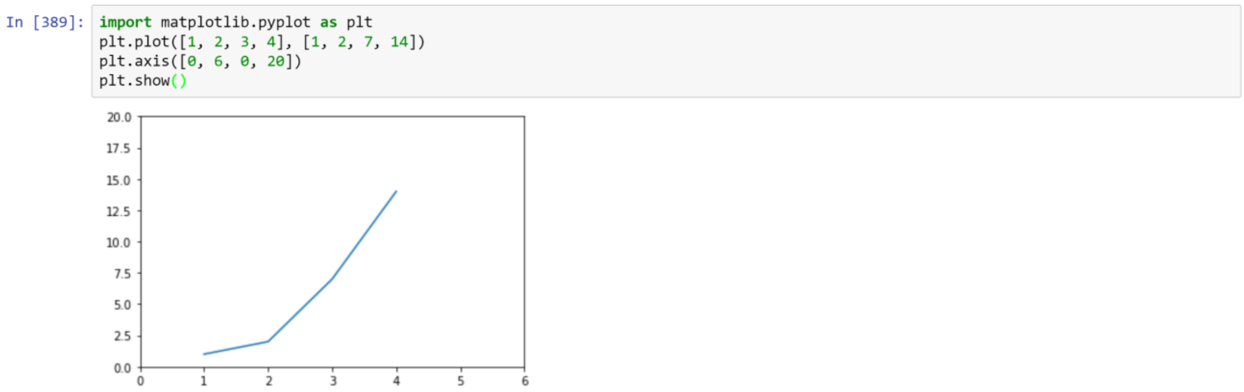
```
In [359]: D, V = linalg.eig(a)
          print(D)
          print(V)

          [ 3.41421356  2.          0.58578644]
          [[ -5.00000000e-01  -7.07106781e-01   5.00000000e-01]
           [  7.07106781e-01   4.05405432e-16   7.07106781e-01]
           [ -5.00000000e-01   7.07106781e-01   5.00000000e-01]]
```

```
In [364]: Q, R = scipy.linalg.qr(a)
          print(Q)
          print(R)

          [[-0.89442719 -0.35856858  0.26726124]
           [ 0.4472136  -0.71713717  0.53452248]
           [-0.          0.5976143   0.80178373]]
          [[-2.23606798  1.78885438 -0.4472136 ]
           [ 0.         -1.67332005  1.91236577]
           [ 0.          0.          1.06904497]]
```



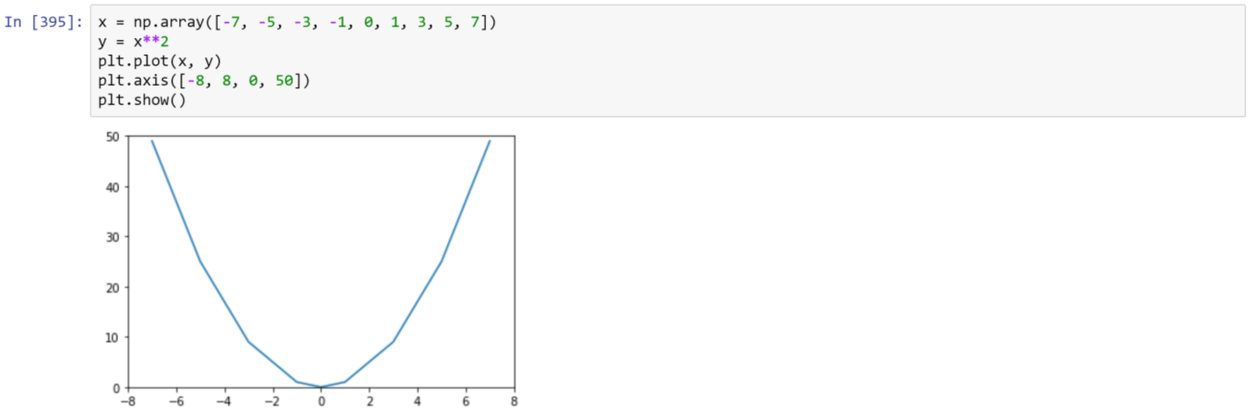
```
In [367]: a = np.array([[2, -1, 0], [-1, 2, -1], [0, -1, 2]])
          scipy.linalg.lu(a)

Out[367]: (array([[ 1.,  0.,  0.],
                  [ 0.,  1.,  0.],
                  [ 0.,  0.,  1.]]), array([[ 1.        ,  0.        ,  0.        ],
                  [-0.5       ,  1.        ,  0.        ],
                  [ 0.        , -0.66666667,  1.        ]]), array([[ 2.        , -1.        ,  0.        ],
                  [ 0.        ,  1.5       , -1.        ],
                  [ 0.        ,  0.        ,  1.33333333]]))

In [369]: import scipy.sparse.linalg as spla
          spla.cg

Out[369]: <function scipy.sparse.linalg.isolve.iterative.cg>

In [371]: from scipy.fftpack import fft, ifft
          fft(a)

Out[371]: array([[ 1.0+0.j        ,  2.5+0.8660254j ,  2.5-0.8660254j ],
                 [ 0.0+0.j        , -1.5-2.59807621j, -1.5+2.59807621j],
                 [ 1.0+0.j        , -0.5+2.59807621j, -0.5-2.59807621j]])

In [372]: ifft(a)

Out[372]: array([[ 0.33333333+0.j        ,  0.83333333-0.28867513j,
                   0.83333333+0.28867513j],
                 [ 0.00000000+0.j        , -0.50000000+0.8660254j ,
                  -0.50000000-0.8660254j ],
                 [ 0.33333333+0.j        , -0.16666667-0.8660254j ,
                  -0.16666667+0.8660254j ]])

In [375]: sort(a)

Out[375]: array([[-1,  0,  2],
                 [-1, -1,  2],
                 [-1,  0,  2]])


In [381]: I = argsort(a[:,:])
          b = a[I,:]
          print(b)

          [[[-1  2 -1]
            [ 0 -1  2]
            [ 2 -1  0]]

           [[ 2 -1  0]
            [ 0 -1  2]
            [-1  2 -1]]

           [[-1  2 -1]
            [ 2 -1  0]
            [ 0 -1  2]]]

In [383]: x = np.array([0, 1, 2, 3])
          y = np.array([-1, 0.2, 0.9, 2.1])
          A = np.vstack([x, np.ones(len(x))]).T
          np.linalg.lstsq(A, y)[0]

Out[383]: array([ 1.  , -0.95])

In [386]: import scipy.signal as ss
          x = np.linspace(0, 5, 10, endpoint=False)
          y = np.cos(-x**4/3.0)
          ss.resample(y, 3)

Out[386]: array([ 0.50570367,  0.67677839,  0.06021344])

In [387]: a = np.array([[i*j for j in range(3)] for i in range(3)])
          unique(a)

Out[387]: array([0, 1, 2, 4])

In [388]: a.squeeze()

Out[388]: array([[0, 0, 0],
                 [0, 1, 2],
                 [0, 2, 4]])
```

Task 3.



```
In [389]: import matplotlib.pyplot as plt
          plt.plot([1, 2, 3, 4], [1, 2, 7, 14])
          plt.axis([0, 6, 0, 20])
          plt.show()
```

Task 4.



```
In [395]: x = np.array([-7, -5, -3, -1, 0, 1, 3, 5, 7])
          y = x**2
          plt.plot(x, y)
          plt.axis([-8, 8, 0, 50])
          plt.show()
```

Task 5.

https://github.com/weichongchen

Task 6.

https://github.com/weichongchen/Assignment0