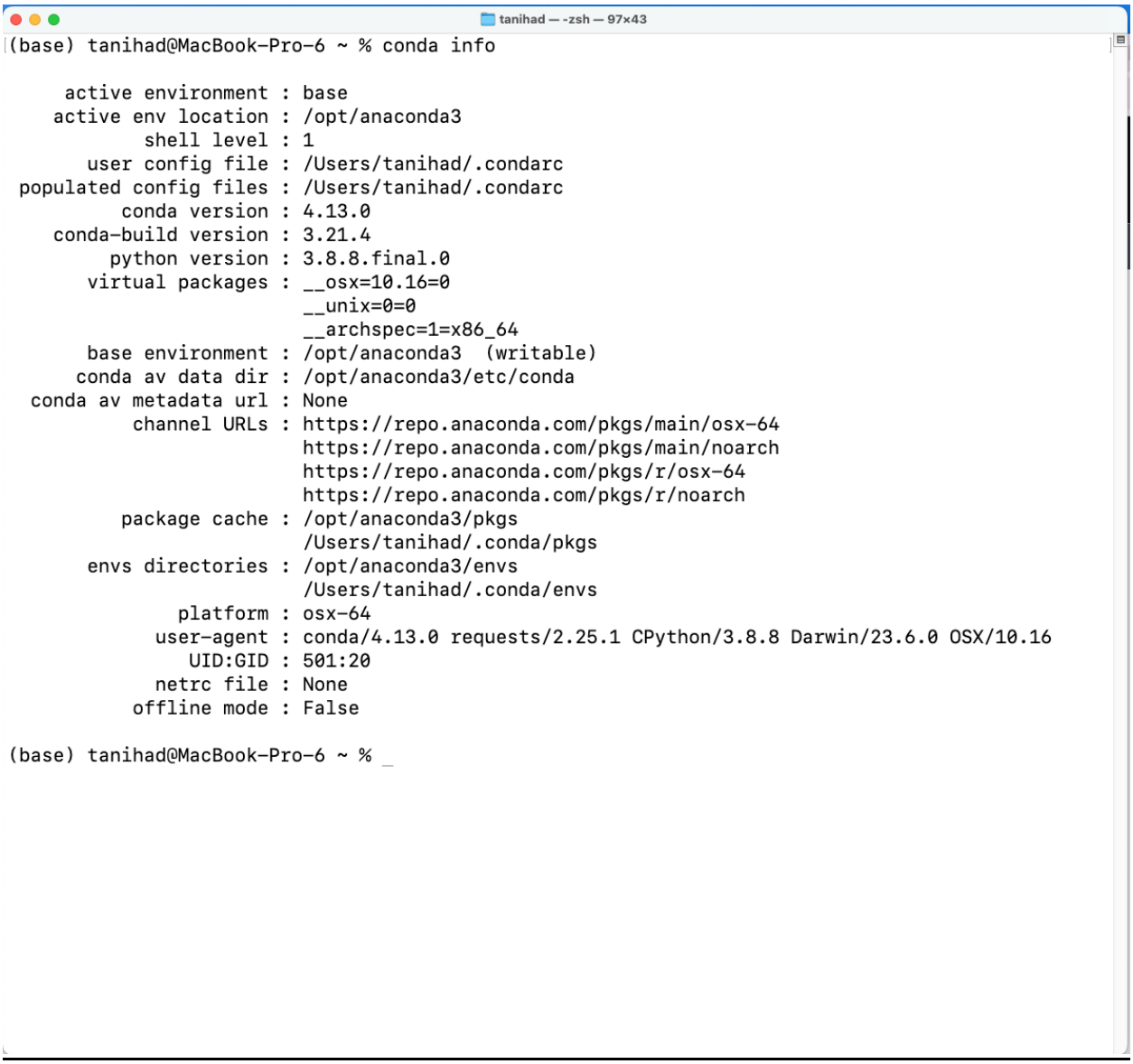
**TASK 1:**

[(base) tanihad®MacBook-Pro-6 ~ % conda info
active environment : base
active env location : /opt/anaconda3
shell level : 1
user config file : /Users/tanihad/.condarc
populated config files : /Users/tanihad/.condarc
conda version : 4.13.0
conda-build version : 3.21.4
python version : 3.8.8. final.0
virtual packages : __osx=10.16=0
-_unix=0=0
__archspec=1=x86_64
base environment : /opt/anaconda3 (writable) conda av data dir:
/opt/anaconda3/etc/conda
conda av metadata url: None
channel URLs : https://repo.anaconda.com/pkgs/main/osx-64
https://repo.anaconda.com/pkgs/main/noarch
https://repo.anaconda.com/pkgs/r/osx-64
https://repo.anaconda.com/pkgs/r/noarch
package cache : /opt/anaconda3/pkgs
/Users/tanihad/.conda/pkgs
envs directories: /opt/anaconda3/envs
/Users/tanihad/.conda/envs
platform: 05x-64
user-agent: conda/4.13.0 requests/2.25.1 CPython/3.8.8 Darwin/23.6.0
OSX/10.16
UID:GID : 501:20
netrc file : None offline mode : False

**Screenshot for reference:**




## TASK 2:

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

In [2]: **import scipy.linalg**

In [3]: a = np.array([1,2])

```
In [265]: np.ndim(a)
```
Out[265]: 1

```
In [5]: np.size(a)
```
Out[266]: 2

```
In [267]: np.shape(a)
```
Out[267]: (2,)

[270]: n = 1

```
In [271]: a.shape[n-1]
```
Out[271]: 2

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

```
In [9]: a = np.array([[7.,8.,9.], [10.,11.,12.]])

In [10]: c = np.array([[13.,14.,15.], [16.,17.,18.]])

In [11]: d = np.array([[19.,20.,21.],[22.,23.,24.]])
```

```
In [12]: e = np.block([[a,b],[c,d]])
```

```
In [13]: print(e)
[[ 7.  8.  9.  1.  2.  3.]
 [10. 11. 12.  4.  5.  6.]
 [13. 14. 15. 19. 20. 21.]
```

```
 [16. 17. 18. 22. 23. 24.]]
```

```
In [273]: a[-1]
```

```
Out[273]: array([10., 11., 12.])
In [15]: array = np.array([[1, 2, 3, 4, 5],[6, 7, 8, 9, 10],[11, 12, 13, 14, 15]]
   ...: )
```

```
In [274]: array[1,4]
```

```
Out[274]: 10
```

```
In [275]: array[1]
```

```
Out[275]: array([ 6,  7,  8,  9, 10])
```

```
In [279]: array = np.array([[1,2,3,4,5],[6,7,8,9,10],[11,12,13,14,15],[16,1
   ...: 7,18,19,20],[26,27,28,29,30],[31,32,33,34,35]])
```

```
In [280]: array[:5]
```

```
Out[280]:
array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10],
       [11, 12, 13, 14, 15],
       [16, 17, 18, 19, 20],
       [26, 27, 28, 29, 30]])
```

```
In [281]: array[-5:]
```

```
Out[281]:
array([[ 6,  7,  8,  9, 10],
       [11, 12, 13, 14, 15],
       [16, 17, 18, 19, 20],
       [26, 27, 28, 29, 30],
```

```
       [31, 32, 33, 34, 35]])
```

```
In [282]: array[0:3,4:9]
```

```
Out[282]:
array([[ 5],
       [10],
       [15]])
```

```
In [284]: array[np.ix_([1,3,4],[0,2])]
```

```
Out[284]:
array([[ 6,  8],
       [16, 18],
       [26, 28]])
```

```
In [291]: array[2:21:2,:]
```

```
Out[291]:
array([[11, 12, 13, 14, 15],
       [26, 27, 28, 29, 30]])
```

```
In [292]: array[::2,:]
```

```
Out[292]:
array([[ 1,  2,  3,  4,  5],
       [11, 12, 13, 14, 15],
       [26, 27, 28, 29, 30]])
```

```
In [293]: array[::-1,:]
```

```
Out[293]:
array([[31, 32, 33, 34, 35],
       [26, 27, 28, 29, 30],
       [16, 17, 18, 19, 20],
       [11, 12, 13, 14, 15],
```

```
       [ 6,  7,  8,  9, 10],
       [ 1,  2,  3,  4,  5]])
```

```
In [296]: array[np.r_[:len(array),0]]
```

```
Out[296]:
array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10],
       [11, 12, 13, 14, 15],
       [16, 17, 18, 19, 20],
       [21, 22, 23, 24, 25],
       [26, 27, 28, 29, 30],
       [31, 32, 33, 34, 35],
       [ 1,  2,  3,  4,  5]])
```

```
In [297]: array.transpose()
```

```
Out[297]:
array([[ 1,  6, 11, 16, 21, 26, 31],
       [ 2,  7, 12, 17, 22, 27, 32],
       [ 3,  8, 13, 18, 23, 28, 33],
       [ 4,  9, 14, 19, 24, 29, 34],
       [ 5, 10, 15, 20, 25, 30, 35]])
```

```
In [298]: array.conj().T
```

```
Out[298]:
array([[ 1,  6, 11, 16, 21, 26, 31],
       [ 2,  7, 12, 17, 22, 27, 32],
       [ 3,  8, 13, 18, 23, 28, 33],
       [ 4,  9, 14, 19, 24, 29, 34],
       [ 5, 10, 15, 20, 25, 30, 35]])

In [299]: a = np.array([[1,2,3],[4,5,6]])
```

```
In [300]: b = np.array([[7.,8.],[9.,10.],[11.,12.]])
```

```
In [301]: a@b
```

```
Out[301]:
array([[ 58.,  64.],
       [139., 154.]])
```

```
In [302]: a = np.array([[1.,2.,3.],[4.,5.,6.]])
```

```
In [303]: b = np.array([[7.,8.,9.],[10.,11.,12.]])
```

```
In [304]: a*b
```

```
Out[304]:
array([[ 7., 16., 27.],
       [40., 55., 72.]])
```

```
In [305]: a/b
```

```
Out[305]:
array([[0.14285714, 0.25      , 0.33333333],
       [0.4       , 0.45454545, 0.5       ]])
```

```
In [306]: a**3
```

```
Out[306]:
array([[  1.,   8.,  27.],
       [ 64., 125., 216.]])
```

```
In [307]: a > 0.5
```

```
Out[307]:
array([[ True,  True,  True],
       [ True,  True,  True]])
```

```
In [5]: np.nonzero(a>0.5)
```

```
Out[5]:
(array([0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2]),
 array([0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3]))
```

```
In [62]: v = np.array([0., 1., 0.9, 0.2])
```

```
In [8]: a[:,np.nonzero(v>0.5)[0]]
```

```
Out[8]:
array([[ 2,  3],
       [ 6,  7],
       [10, 11]])
```

```
In [61]: a = np.array([[1, 2, 3, 4],[5, 6, 7, 8],[9, 10, 11, 12]])

In [62]: v = np.array([0., 1., 0.9, 0.2])
```

```
In [310]: a[:, v.T > 0.5]
```

```
Out[310]:
array([[ 2,  3],
       [ 6,  7],
       [10, 11]])
```

```
In [11]: a = np.array([0.2,3.,0.4,0.5])
```

```
In [9]: a[a<0.5]=0
```

```
In [13]: print (a)
[0.  3.  0.  0.5]
```

```
In [77]: a*(a > 0.5)
Out[77]: array([0. , 0.8, 0. ])

In [80]: a[:] = 3

In [81]: print(a)
[3. 3. 3.]

In [83]: y = a.copy()

In [84]: print(y)
[3. 3. 3.]
In [86]: a = np.array([
   ...:     [1, 2, 3, 4],
   ...:     [5, 6, 7, 8],
   ...:     [9, 10, 11, 12] ])

In [87]: y = a[1, :].copy()

In [88]: print(y)
[5 6 7 8]

In [312]: a.flatten()
Out[312]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])

In [91]: np.arange(1.,11.)
Out[91]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])

In [92]: np.arange(10.)
Out[92]: array([0., 1., 2., 3., 4., 5., 6., 7., 8., 9.])
```

```
In [93]: np.arange(1.,11.)[:,np.newaxis]
```

```
Out[93]:
array([[ 1.],
       [ 2.],
       [ 3.],
       [ 4.],
       [ 5.],
       [ 6.],
       [ 7.],
       [ 8.],
       [ 9.],
       [10.]])
```

```
In [94]: np.zeros((3,4))
```

```
Out[94]:
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

```
In [95]: np.zeros((3,4,5))
```

```
Out[95]:
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 [96]: np.ones((3,4))
```

```
Out[96]:
array([[1., 1., 1., 1.],
       [1., 1., 1., 1.],
       [1., 1., 1., 1.]])
```

```
In [97]: np.eye(3)
```

```
Out[97]:
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])

In [99]: v=np.array([1,2,3,4])
```

```
In [100]: np.diag(v,0)
```

```
Out[100]:
array([[1, 0, 0, 0],
       [0, 2, 0, 0],
       [0, 0, 3, 0],
       [0, 0, 0, 4]])
```

```
In [101]: from numpy.random import default_rng

In [102]: rng = default_rng(42)

In [103]: rng.random((3,4))
```

Out[**103**]:
```
array([[0.77395605, 0.43887844, 0.85859792, 0.69736803],
       [0.09417735, 0.97562235, 0.7611397 , 0.78606431],
       [0.12811363, 0.45038594, 0.37079802, 0.92676499]])
```

In [**104**]: np.linspace(1,3,4)

Out[**104**]: array([1.  , 1.66666667, 2.33333333, 3.  ])

In [**105**]: np.mgrid[0:9.,0:6.]

Out[**105**]:
```
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 [**24**]: np.ogrid[0:9.,0:6.]

```
Out[24]:
[array([[0.],
        [1.],
        [2.],
        [3.],
        [4.],
        [5.],
        [6.],
        [7.],
        [8.]]),
 array([[0., 1., 2., 3., 4., 5.]])]
```

```
In [107]: np.ix_(np.r_[0:9.],np.r_[0:6.])
```

```
Out[107]:
(array([[0.],
        [1.],
        [2.],
        [3.],
        [4.],
        [5.],
        [6.],
        [7.],
        [8.]]),
 array([[0., 1., 2., 3., 4., 5.]]))
```

```
In [108]: np.meshgrid([1,2,4],[2,4,5])
```

```
Out[108]:
[array([[1, 2, 4],
        [1, 2, 4],
        [1, 2, 4]]),
 array([[2, 2, 2],
        [4, 4, 4],
```

```
       [5, 5, 5]])]
```

In [**114**]: a = np.array([[1,2],[3,4]])

In [**115**]: np.tile(a,(2,3))

Out[**115**]:

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

In [**117**]: np.concatenate((a, b), axis=1)

Out[**117**]:

```
array([[ 1.,  2.,  7.,  8.,  9.],
       [ 3.,  4., 10., 11., 12.]])
```

In [**119**]: print(a)

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

In [**120**]: b = np.array([[5,6],[7,8]])

In [**121**]: np.concatenate((a, b))

Out[**121**]:

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

In [**122**]: a.max()

Out[**122**]: 4

```
In [25]: a.max(0)
```
Out[25]: array([7, 5])

```
In [123]: a.max(1)
```
Out[123]: array([2, 4])

```
In [126]: np.maximum(a,b)
```
Out[126]:
array([[5, 6],
       [7, 8]])

```
In [127]: np.sqrt(v@v)
```
Out[127]: 5.477225575051661

```
In [129]: np.logical_and(a,b)
```
Out[129]:
array([[ True,  True],
       [ True,  True]])

In [15]: b = np.array([1,53,7,9])

```
In [17]: np.logical_or(a,b)
```
Out[17]: array([ True,  True,  True,  True])

```
In [130]: a&b
```
Out[130]:
array([[1, 2],
       [3, 0]])

```
In [131]: a|b
```

Out[**131**]:
```
array([[ 5,  6],
       [ 7, 12]])
```

In [**21**]: a = np.array([[1,5],[7,3]])

In [**22**]: np.linalg.inv(a)

Out[**22**]:
```
array([[-0.09375,  0.15625],
       [ 0.21875, -0.03125]])
```

In [**133**]: np.linalg.pinv(a)

Out[**133**]:
```
array([[-2. ,  1. ],
       [ 1.5, -0.5]])
```

In [**134**]: np.linalg.matrix_rank(a)

Out[**134**]: 2

In [**135**]: np.linalg.solve(a,b)

Out[**135**]:
```
array([[-3., -4.],
       [ 4.,  5.]])
```

In [**142**]: np.linalg.solve(a.T ,b.T).T

Out[**142**]:
```
array([[-1.,  2.],
       [-2.,  3.]])
```

In [**143**]: a = np.array([[1,2],[3,4],[5,6]])

```
In [144]: U, S, Vh = np.linalg.svd(a)
```

```
In [145]: print(U)
[[-0.2298477   0.88346102  0.40824829]
 [-0.52474482  0.24078249 -0.81649658]
 [-0.81964194 -0.40189603  0.40824829]]

In [146]: print(S)
[9.52551809 0.51430058]

In [147]: print(Vh)
[[-0.61962948 -0.78489445]
 [-0.78489445  0.61962948]]

In [160]: a = np.array([[2, -1, 0],[-1, 2, -1],[0, -1, 2]])
```

```
In [161]: np.linalg.cholesky(a)
```

```
Out[161]:
array([[ 1.41421356,  0.        ,  0.        ],
       [-0.70710678,  1.22474487,  0.        ],
       [ 0.        , -0.81649658,  1.15470054]])
```

```
In [163]: D,V = np.linalg.eig(a)
```

```
In [164]: print (D)
[3.41421356 2.         0.58578644]

In [165]: print (V)
[[-5.00000000e-01 -7.07106781e-01  5.00000000e-01]
 [ 7.07106781e-01  4.05925293e-16  7.07106781e-01]
 [-5.00000000e-01  7.07106781e-01  5.00000000e-01]]
```

In [**173**]: a = np.array([[1,2],[3,4]])

In [**174**]: b = np.array([[5,6],[7,8]])

In [**175**]: D,V = scipy.linalg.eig(a,b)

In [**176**]: print(D)
[1.00000006+0.j 0.99999994+0.j]

In [**177**]: print(V)
[[ 0.70710679 -0.70710678]
 [-0.70710678  0.70710679]]

In [**189**]: a = np.eye(100)

In [**190**]: D,V = eigs(a,k=3)

In [**191**]: print(D)
[1.+0.j 1.+0.j 1.+0.j]

In [**192**]: print(V)
[[ 2.20849760e-02+0.j  1.71922726e-02+0.j -3.13258644e-02+0.j]
 [-1.07191173e-04+0.j  3.22267962e-04+0.j -1.76184631e-04+0.j]
 [ 7.53570853e-02+0.j -7.38218249e-02+0.j -2.36566621e-01+0.j]
 [-1.00838358e-01+0.j -3.38596090e-02+0.j -2.44256330e-01+0.j]
 [-2.98114211e-02+0.j  2.38808980e-02+0.j  2.67511987e-02+0.j]
 [-5.49108625e-02+0.j  8.34191170e-03+0.j  2.70177015e-02+0.j]
 [ 3.89707946e-02+0.j -1.85140755e-01+0.j -2.03041337e-01+0.j]
 [-2.39993952e-02+0.j  1.13310569e-02+0.j -3.44347011e-02+0.j]
 [-4.09441782e-02+0.j -7.87942517e-02+0.j  4.80215122e-02+0.j]
 [-1.92199265e-02+0.j -9.78823261e-03+0.j  1.52870810e-02+0.j]
 [-6.81559478e-02+0.j  8.00015170e-02+0.j  1.11202102e-01+0.j]
 [ 5.20612022e-02+0.j  2.93299491e-01+0.j -1.73350525e-01+0.j]

[-9.01767483e-02+0.j  1.61206336e-02+0.j -3.83825317e-02+0.j]
[-1.06242134e-01+0.j  1.98346870e-02+0.j  1.13095838e-01+0.j]
[-3.02256912e-02+0.j  4.68233599e-03+0.j -3.37605641e-02+0.j]
[-3.38143491e-02+0.j  3.60344894e-02+0.j  1.18451956e-01+0.j]
[-5.17553612e-02+0.j -1.65564256e-02+0.j  2.99930323e-02+0.j]
[ 2.41784848e-02+0.j  1.60550541e-02+0.j  1.16759889e-02+0.j]
[ 5.89556327e-03+0.j  3.18511737e-03+0.j -4.61584674e-04+0.j]
[ 1.66356643e-02+0.j -1.55487629e-02+0.j  6.75006192e-02+0.j]
[-5.24541174e-02+0.j  7.63523212e-03+0.j  1.15443312e-01+0.j]
[-1.60964934e-01+0.j -5.54933444e-02+0.j  7.51869352e-02+0.j]
[ 7.41149563e-02+0.j  6.93036106e-02+0.j -4.55900406e-02+0.j]
[ 5.68359171e-02+0.j  3.15680856e-02+0.j  5.96446628e-02+0.j]
[ 8.45155680e-02+0.j -3.24275947e-01+0.j  1.79311154e-01+0.j]
[-8.26774593e-02+0.j -2.92231703e-02+0.j  2.32635173e-02+0.j]
[-2.29136534e-02+0.j -1.78033384e-02+0.j  1.95529625e-02+0.j]
[-1.75631020e-02+0.j -6.05599065e-02+0.j -1.14943144e-01+0.j]
[-3.85780606e-02+0.j -4.50283852e-03+0.j  2.76736269e-02+0.j]
[ 8.40729520e-02+0.j  2.01127796e-02+0.j  2.46901557e-01+0.j]
[-3.80019661e-02+0.j -3.58678719e-02+0.j -1.19694496e-01+0.j]
[ 3.93682252e-02+0.j  1.93940345e-02+0.j  3.03381601e-02+0.j]
[ 2.46145690e-01+0.j  2.80925564e-01+0.j  1.84609993e-01+0.j]
[ 2.33575120e-02+0.j -1.11854738e-02+0.j  3.44066873e-02+0.j]
[ 3.95775764e-01+0.j -2.26720258e-01+0.j -2.07145217e-01+0.j]
[ 2.75041944e-02+0.j  1.08813621e-02+0.j  2.92526279e-02+0.j]
[-1.15820228e-03+0.j  3.24385439e-04+0.j  3.94269817e-04+0.j]
[-6.40269571e-02+0.j  1.84633493e-02+0.j  1.04560243e-02+0.j]
[-6.31840367e-02+0.j -1.69671843e-01+0.j  1.61666332e-01+0.j]
[-3.91846831e-02+0.j -1.44356344e-02+0.j -6.17308866e-02+0.j]
[-2.11972773e-02+0.j  1.89762017e-02+0.j  2.86028421e-02+0.j]
[ 1.41198818e-01+0.j -2.08326703e-01+0.j -1.98579792e-01+0.j]
[-6.41266585e-02+0.j -7.74568640e-02+0.j  7.94132545e-02+0.j]
[ 8.50389858e-02+0.j  2.90796906e-02+0.j  1.84061973e-03+0.j]
[-2.45104844e-01+0.j -1.20654384e-01+0.j -9.36956584e-02+0.j]

[ 1.26800252e-02+0.j  3.02231385e-01+0.j -1.75069240e-01+0.j]
[-4.80310046e-03+0.j  1.28861616e-02+0.j -2.13303561e-02+0.j]
[ 2.82421282e-02+0.j -1.81952456e-02+0.j  3.75712584e-02+0.j]
[-9.11652482e-02+0.j -5.16513094e-02+0.j  6.83952361e-02+0.j]
[-6.11359494e-02+0.j  1.78146833e-02+0.j  2.58001479e-02+0.j]
[ 4.92121437e-02+0.j  5.72881854e-02+0.j  5.34850060e-02+0.j]
[ 2.46032978e-02+0.j  9.12809125e-03+0.j  3.11030995e-02+0.j]
[-1.00551399e-02+0.j  2.10397299e-04+0.j -1.61676497e-02+0.j]
[-2.74640392e-01+0.j  1.30768073e-01+0.j  1.00222893e-01+0.j]
[ 7.16144477e-02+0.j  4.82374351e-02+0.j -1.96594507e-02+0.j]
[-2.85050777e-02+0.j -2.57643408e-02+0.j  3.29753403e-02+0.j]
[ 2.29477504e-02+0.j  1.31751176e-01+0.j  2.25419594e-01+0.j]
[-9.68523091e-02+0.j  3.76770010e-02+0.j -7.29691138e-02+0.j]
[ 1.20085208e-01+0.j -5.95077310e-02+0.j -5.18474316e-02+0.j]
[ 1.63039153e-02+0.j -9.59622317e-03+0.j  1.12719199e-02+0.j]
[-1.50994289e-01+0.j  5.77978213e-02+0.j -1.40139029e-01+0.j]
[ 9.68608976e-02+0.j  1.54333287e-01+0.j  2.08969448e-01+0.j]
[-4.01506270e-02+0.j  1.92631177e-03+0.j -1.80109330e-02+0.j]
[-2.05981258e-02+0.j -9.48971338e-03+0.j -1.52089704e-02+0.j]
[-4.61817885e-02+0.j -5.79754887e-02+0.j -5.33527520e-02+0.j]
[-7.88168194e-03+0.j -2.04417973e-02+0.j -2.89260653e-02+0.j]
[ 1.31933366e-01+0.j  5.81573414e-02+0.j -4.40504651e-02+0.j]
[ 1.00378072e-02+0.j -2.03621908e-02+0.j  1.09225968e-02+0.j]
[-1.24605293e-01+0.j -5.98615883e-02+0.j -4.69374228e-02+0.j]
[-2.88317584e-02+0.j  4.59706113e-04+0.j  2.91396707e-02+0.j]
[ 4.47520899e-02+0.j -1.88343316e-03+0.j  3.44324661e-02+0.j]
[ 9.19865355e-02+0.j  6.52502260e-02+0.j -4.48100701e-02+0.j]
[ 1.30223240e-01+0.j  3.14917928e-01+0.j -1.89615237e-01+0.j]
[-2.03456049e-02+0.j -3.27936986e-03+0.j -1.64039926e-02+0.j]
[ 4.39148771e-02+0.j  9.54596748e-03+0.j -5.90826573e-02+0.j]
[ 1.44769753e-03+0.j  1.58004525e-02+0.j  1.33893898e-02+0.j]
[ 2.84631522e-01+0.j -2.09212492e-01+0.j  1.57170126e-01+0.j]
[ 1.29229805e-01+0.j  1.15374314e-01+0.j -7.82361442e-02+0.j]

```
[-1.08164821e-01+0.j -7.14033435e-02+0.j  7.94588883e-02+0.j]
[ 1.99764792e-01+0.j  1.30994048e-01+0.j  2.13460489e-01+0.j]
[-1.48588331e-01+0.j  1.02178686e-01+0.j  1.05724192e-01+0.j]
[-8.47595049e-02+0.j -2.91448367e-02+0.j -5.63291014e-03+0.j]
[-1.44339371e-02+0.j  6.22817771e-03+0.j  5.50333864e-03+0.j]
[ 5.54523171e-02+0.j  3.19100244e-02+0.j  1.20456083e-01+0.j]
[ 2.88177199e-01+0.j  7.93521917e-02+0.j -1.04273877e-02+0.j]
[ 7.68539071e-02+0.j -3.31412398e-02+0.j  7.20963222e-02+0.j]
[ 1.35079158e-02+0.j  1.40382235e-02+0.j -9.40309071e-03+0.j]
[-3.65855176e-02+0.j  1.11772293e-03+0.j -1.78553407e-02+0.j]
[ 5.17918897e-02+0.j -1.88048659e-01+0.j -2.02481785e-01+0.j]
[ 1.45111285e-01+0.j  2.72195157e-01+0.j -1.69289534e-01+0.j]
[ 8.03185846e-02+0.j -3.84248934e-03+0.j  6.64585249e-02+0.j]
[ 4.89418377e-04+0.j -1.72129054e-02+0.j  1.00139773e-02+0.j]
[ 2.12594069e-02+0.j  1.80452567e-03+0.j -1.37508202e-02+0.j]
[-1.31005121e-01+0.j  9.81063157e-02+0.j -7.61238379e-02+0.j]
[-1.30439569e-03+0.j -1.07230261e-02+0.j -1.28533117e-02+0.j]
[-3.95913847e-02+0.j  6.98888435e-04+0.j -6.46431396e-02+0.j]
[-1.11412195e-02+0.j  6.69577049e-02+0.j -1.29796820e-01+0.j]
[ 4.21782704e-02+0.j  1.46093252e-02+0.j -1.62147773e-02+0.j]
[-6.81379942e-02+0.j  4.77257817e-02+0.j  5.41145672e-02+0.j]
[-9.52800247e-02+0.j -4.28875170e-02+0.j -1.94182657e-02+0.j]]
```

```
In [194]: Q,R = scipy.linalg.qr(a)
```

```
In [195]: print(Q)
[[1. 0. 0. ... 0. 0. 0.]
 [0. 1. 0. ... 0. 0. 0.]
 [0. 0. 1. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 1. 0. 0.]
 [0. 0. 0. ... 0. 1. 0.]
```

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

In [196]: print(R)

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

In [197]: P,L,U = scipy.linalg.lu(a)

In [198]: print(P)

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

In [199]: print(L)

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

In [200]: print(U)

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

In [**213**]: a = np.array([[1,2,0],[3,4,0],[0,0,5]])

In [**214**]: b = np.array([1,2,3])

In [**215**]: scipy.sparse.linalg.cg(a,b)

Out[**215**]: (array([-0.93268941,  0.9935026 ,  0.64018077]), 30)

In [**216**]: np.fft.fft(a)

Out[**216**]:

```
array([[ 3. +0.j        ,  0. -1.73205081j,  0. +1.73205081j],
       [ 7. +0.j        ,  1. -3.46410162j,  1. +3.46410162j],
       [ 5. +0.j        , -2.5+4.33012702j, -2.5-4.33012702j]])
```

In [**217**]: np.fft.ifft(a)

Out[**217**]:

```
array([[ 1.        +0.j        ,  0.        +0.57735027j,
         0.        -0.57735027j],
       [ 2.33333333+0.j        ,  0.33333333+1.15470054j,
         0.33333333-1.15470054j],
       [ 1.66666667+0.j        , -0.83333333-1.44337567j,
        -0.83333333+1.44337567j]])
```

In [**218**]: np.sort(a)

Out[**218**]:
```
array([[0, 1, 2],
       [0, 3, 4],
       [0, 0, 5]])
```

In [**219**]: np.sort(a,axis=1)

Out[**219**]:
```
array([[0, 1, 2],
       [0, 3, 4],
       [0, 0, 5]])
array([[0, 1, 2],
       [0, 3, 4],
       [0, 0, 5]])
```

In [**220**]: I = np.argsort(a[:,0])

In [**221**]: print(I)
[2 0 1]
In [**226**]: y = np.array([1,2,3,4])

In [**227**]: z = np.array([[1,1],[1,2],[1,3],[1,4]])

In [**230**]: x = scipy.linalg.lstsq(z,y)

In [**231**]: print(x)
(array([2.66453526e-16, 1.00000000e+00]), 0.0, 2, array([5.77937881, 0.77380911]))

In [**261**]: scipy.signal.resample(x,int(np.ceil(len(x)//2)))

Out[**261**]: array([1.5, 3.5])

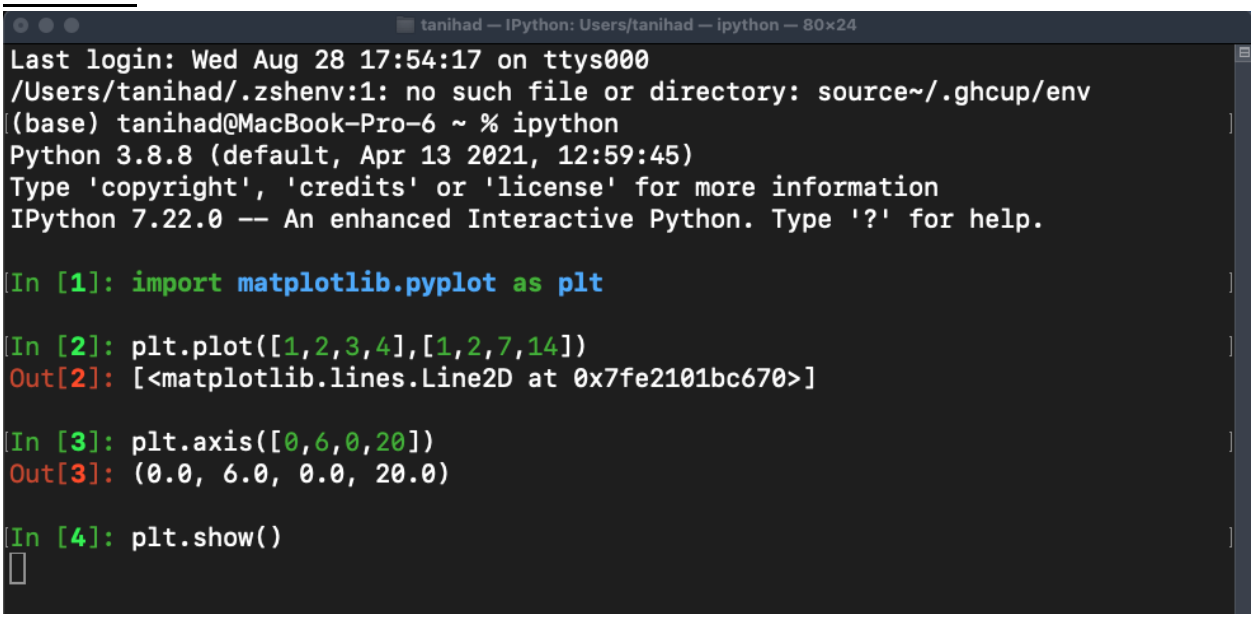
```
In [262]: np.unique(a)
```

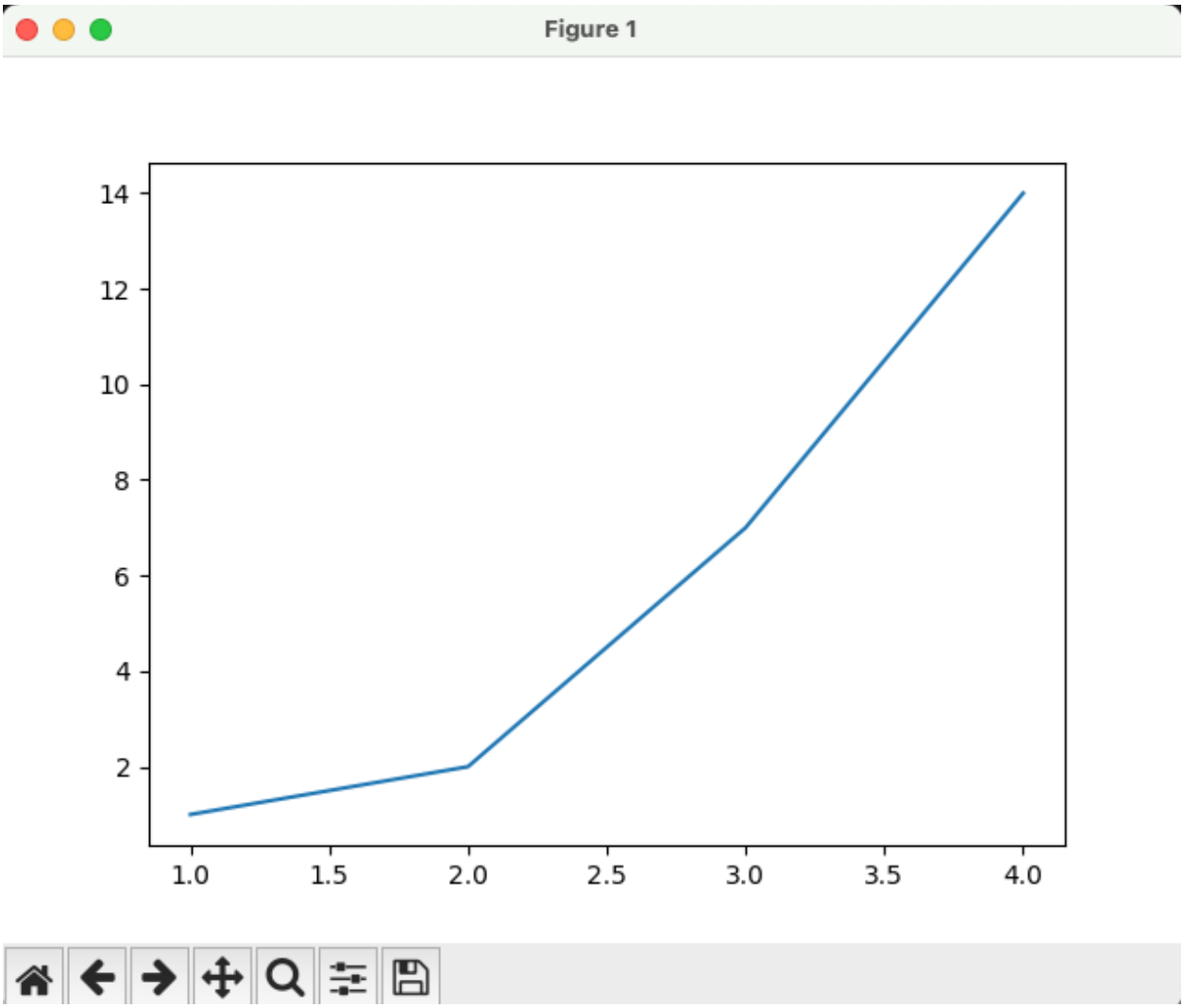
```
Out[262]: array([0, 1, 2, 3, 4, 5])
```

```
In [263]: a.squeeze()
```

```
Out[263]:
array([[1, 2, 0],
       [3, 4, 0],
       [0, 0, 5]])
```

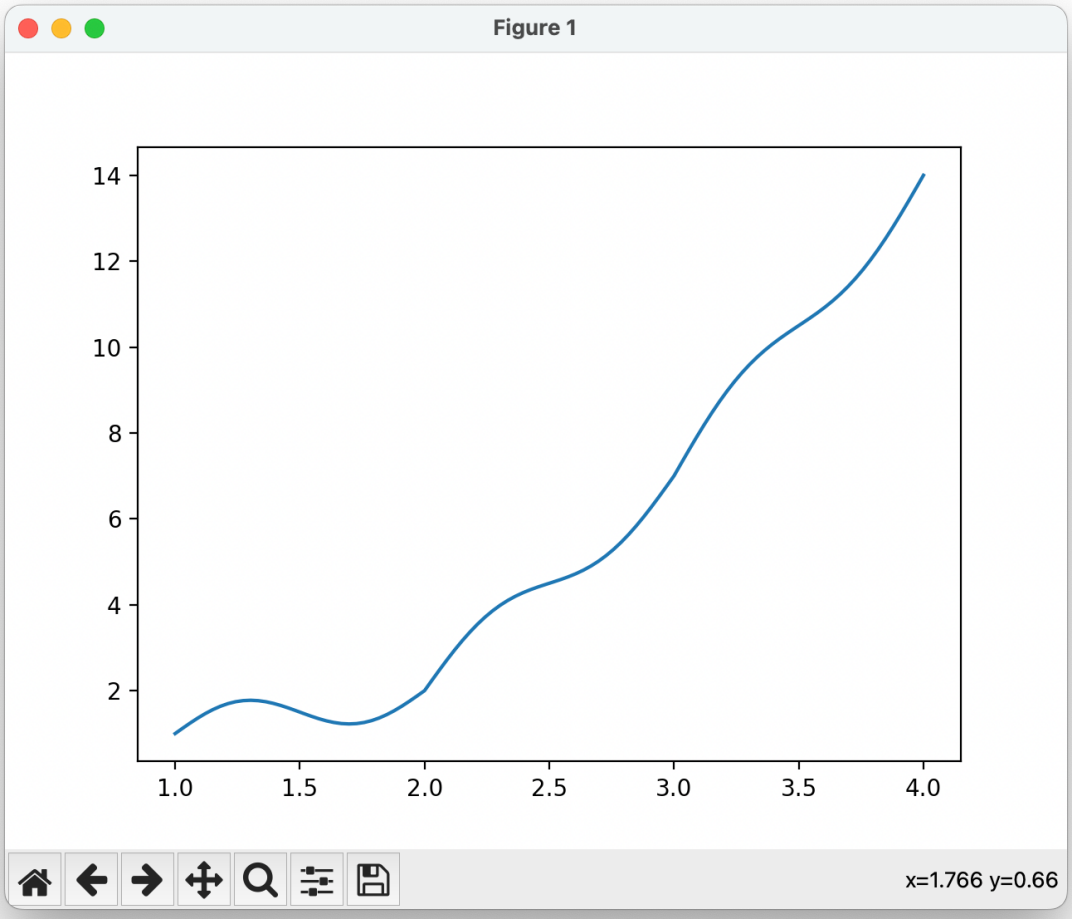
**TASK 3:**






## TASK 4

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

In [2]: import matplotlib.pyplot as plt

In [3]: x = np.linspace(1,4,100)

In [4]: y = np.interp(x,[1,2,3,4],[1,2,7,14]) + 0.5 * np.sin(2
   ...: *np.pi*x)

In [5]: plt.plot(x,y)
Out[5]: [<matplotlib.lines.Line2D at 0x7fa2208f5610>]

In [6]: plt.axis([0,6,0,20])
Out[6]: (0.0, 6.0, 0.0, 20.0)

In [7]: plt.show()
```




**TASK 5:**

Github profile: https://github.com/thanasisrice

**TASK 6**