**Task1:**

active environment : None

user config file : C:\Users\KRQ\.condarc

populated config files : C:\Users\KRQ\.condarc

conda version : 4.13.0

conda-build version : 3.21.8

python version : 3.9.12.final.0

virtual packages : \_\_cuda=11.6=0

\_\_win=0=0

\_\_archspec=1=x86\_64

base environment : D:\Anaconda (writable)

conda av data dir : D:\Anaconda\etc\conda

conda av metadata url : None

channel URLs : https://repo.anaconda.com/pkgs/main/win-64

https://repo.anaconda.com/pkgs/main/noarch

https://repo.anaconda.com/pkgs/r/win-64

https://repo.anaconda.com/pkgs/r/noarch

https://repo.anaconda.com/pkgs/msys2/win-64

https://repo.anaconda.com/pkgs/msys2/noarch

package cache : D:\Anaconda\pkgs

C:\Users\KRQ\.conda\pkgs

C:\Users\KRQ\AppData\Local\conda\conda\pkgs

envs directories : D:\Anaconda\envs

C:\Users\KRQ\.conda\envs

C:\Users\KRQ\AppData\Local\conda\conda\envs

platform : win-64

user-agent : conda/4.13.0 requests/2.27.1 CPython/3.9.12 Windows/10 Windows/10.0.22000

administrator : True

netrc file : None

offline mode : False

**Task2:**

In [1]: import numpy as np

In [2]: import scipy.linalg

In [3]: a=[1,2,3,4,5]

In [4]: np.ndim(a)

Out[4]: 1

In [5]: np.size(a)

Out[5]: 5

In [6]: np.shape(a)

Out[6]: (5,)

In [8]: b=np.array([[1,2,3,4],[5,6,7,8]])

In [9]: b.shape

Out[9]: (2, 4)

In [10]: c=np.block([[1,2],[3,4]])

In [11]: c

Out[11]:

array([[1, 2],

[3, 4]])

In [12]: a[-1]

Out[12]: 5

In [14]: b[1,3]

Out[14]: 8

In [15]: b[1]

Out[15]: array([5, 6, 7, 8])

In [17]: b[:3]

Out[17]:

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

[5, 6, 7, 8]])

In [18]: b[:2]

Out[18]:

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

[5, 6, 7, 8]])

In [19]: b[0:1,2:3]

Out[19]: array([[3]])

In [21]: b[np.ix\_([0,1],[2,3])]

Out[21]:

array([[3, 4],

[7, 8]])

In[22]: c=np.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]])

In [23]: c[:,2:7:2]

Out[23]:

array([[ 3, 5, 7],

[13, 15, 17],

[23, 25, 27]])

In [24]: c[:,::2]

Out[24]:

array([[ 1, 3, 5, 7, 9],

[11, 13, 15, 17, 19],

[21, 23, 25, 27, 29]])

In [25]: c[:,::-1]

Out[25]:

array([[10, 9, 8, 7, 6, 5, 4, 3, 2, 1],

[20, 19, 18, 17, 16, 15, 14, 13, 12, 11],

[30, 29, 28, 27, 26, 25, 24, 23, 22, 21]])

In [26]: c[np.r\_[:len(c),0]]

Out[26]:

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],

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

In [28]: c.transpose()

Out[28]:

array([[ 1, 11, 21],

[ 2, 12, 22],

[ 3, 13, 23],

[ 4, 14, 24],

[ 5, 15, 25],

[ 6, 16, 26],

[ 7, 17, 27],

[ 8, 18, 28],

[ 9, 19, 29],

[10, 20, 30]])

In [29]: c.conj().transpose()

Out[29]:

array([[ 1, 11, 21],

[ 2, 12, 22],

[ 3, 13, 23],

[ 4, 14, 24],

[ 5, 15, 25],

[ 6, 16, 26],

[ 7, 17, 27],

[ 8, 18, 28],

[ 9, 19, 29],

[10, 20, 30]])

In [30]: x=np.array([[1,2],[3,4]])

In [33]: y=np.array([[1,-1],[-1,1]])

In [34]: x\*y

Out[34]:

array([[ 1, -2],

[-3, 4]])

In [36]: x@y

Out[36]:

array([[-1, 1],

[-1, 1]])

In [37]: x\*\*2

Out[37]:

array([[ 1, 4],

[ 9, 16]], dtype=int32)

In [38]: (x>2)

Out[38]:

array([[False, False],

[ True, True]])

In [39]: np.nonzero(x>2)

Out[39]: (array([1, 1], dtype=int64), array([0, 1], dtype=int64))

In [40]: x[:,np.nonzero(x>2)[0]]

Out[40]:

array([[2, 2],

[4, 4]])

In [41]: z=x[1,:].copy()

In [42]: z

Out[42]: array([3, 4])

In [43]: x.flatten()

Out[43]: array([1, 2, 3, 4])

In [46]: a=np.arange(1.,11.)

In [47]: a

Out[47]: array([ 1., 2., 3., 4., 5., 6., 7., 8., 9., 10.])

In [49]: b=np.zeros((3,4))

In [50]: b

Out[50]:

array([[0., 0., 0., 0.],

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

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

In [51]: c=np.ones((3,4))

In [52]: c

Out[52]:

array([[1., 1., 1., 1.],

[1., 1., 1., 1.],

[1., 1., 1., 1.]])

In [54]: d=np.eye(3)

In [55]: d

Out[55]:

array([[1., 0., 0.],

[0., 1., 0.],

[0., 0., 1.]])

In [56]: np.diag(d)

Out[56]: array([1., 1., 1.])

In [57]: np.linspace(1,3,4)

Out[57]: array([1. , 1.66666667, 2.33333333, 3. ])

In [58]: np.mgrid[0:9.,0:6.]

Out[58]:

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 [60]: np.meshgrid([1,2,4],[2,4,5])

Out[60]:

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

[1, 2, 4],

[1, 2, 4]]),

array([[2, 2, 2],

[4, 4, 4],

[5, 5, 5]])]

In [61]: c.max()

Out[61]: 1.0

In [63]: np.maximum(b,c)

Out[63]:

array([[1., 1., 1., 1.],

[1., 1., 1., 1.],

[1., 1., 1., 1.]])

In [64]: v=np.array([1,4,5,3])

In [65]: np.linalg.norm(v)

Out[65]: 7.14142842854285

In [68]: m=np.array([[1,2,3,4],[5,6,7,8],[9,8,7,6],[5,4,3,2]])

In [72]: import scipy.linalg as linalg

In [73]: linalg.inv(m)

Out[73]:

array([[-1.23848990e+16, 1.01330992e+16, -6.75539944e+15,

4.50359963e+15],

[ 1.41863388e+17, -1.01330992e+17, 4.05323966e+16,

-2.00000000e+00],

[-2.46572080e+17, 1.72262686e+17, -6.07985950e+16,

-1.35107989e+16],

[ 1.17093590e+17, -8.10647933e+16, 2.70215978e+16,

9.00719925e+15]])

In [74]: linalg.pinv(m)

Out[74]:

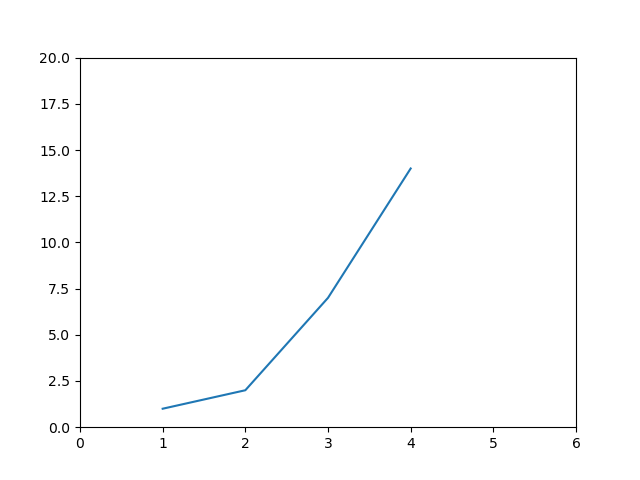
array([[-0.07241379, -0.06896552, 0.08103448, 0.07758621],

[-0.01982759, -0.01293103, 0.03706897, 0.03017241],

[ 0.03275862, 0.04310345, -0.00689655, -0.01724138],

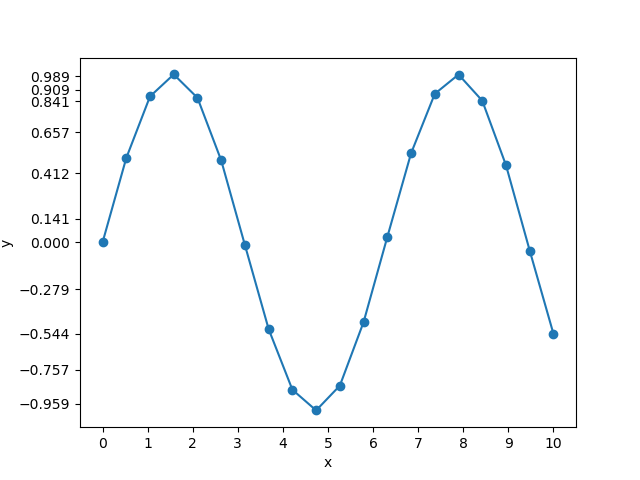
[ 0.08534483, 0.09913793, -0.05086207, -0.06465517]])

**Task3:**



**Task4:**

import matplotlib.pyplot as plt  
import numpy as np  
x = np.linspace(0, 10, 20)  
y = np.sin(x)  
plt.xlabel("x")  
plt.ylabel("y")  
xticks = np.linspace(0, 10, 11)  
yticks = np.sin(xticks)  
plt.xticks(xticks)  
plt.yticks(yticks)  
plt.plot(x, y, linestyle="-", marker="o", label="line")  
plt.show()

****

**Task5:**

# GITHUB:[RuiqiKuang](https://github.com/RuiqiKuang)