Task 1:

```
(base) zeyang@zeyang-desktop:~$ conda info

    active environment : base
   active env location : /home/zeyang/anaconda3
           shell level : 1
      user config file : /home/zeyang/.condarc
populated config files : /home/zeyang/.condarc
         conda version : 4.14.0
   conda-build version : 3.22.0
        python version : 3.9.13.final.0
      virtual packages : __linux=5.4.0=0
                         __glibc=2.27=0
                         __unix=0=0
                         __archspec=1=x86_64
      base environment : /home/zeyang/anaconda3  (writable)
     conda av data dir : /home/zeyang/anaconda3/etc/conda
 conda av metadata url : None
          channel URLs : https://repo.anaconda.com/pkgs/main/linux-64
                         https://repo.anaconda.com/pkgs/main/noarch
                         https://repo.anaconda.com/pkgs/r/linux-64
                         https://repo.anaconda.com/pkgs/r/noarch
         package cache : /home/zeyang/anaconda3/pkgs
                         /home/zeyang/.conda/pkgs
      envs directories : /home/zeyang/anaconda3/envs
                         /home/zeyang/.conda/envs
              platform : linux-64
            user-agent : conda/4.14.0 requests/2.28.1 CPython/3.9.13 Linux/5.4.0-124-generic ubuntu/18.04.6 glibc/2.27
               UID:GID : 1000:1000
            netrc file : /home/zeyang/.netrc
          offline mode : False
```

Task 2:

```
In [10]: import numpy as np
    ...: import scipy.linalg

In [11]: from scipy import linalg

In [12]: from scipy import signal

In [13]: a = np.arange(25).reshape((5, 5))

In [14]: a
Out[14]:
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24]])

In [15]: np.ndim(a)
Out[15]: 2

In [16]: np.size(a)
Out[16]: 25

In [17]: np.shape(a)
Out[17]: (5, 5)
```

```
In [19]: a.shape[1]
Out[19]: 5

In [20]: a.shape[0]
Out[20]: 5

In [21]: np.array([[1. ,2. ,3.], [4. ,5. ,6.]])
Out[21]:
array([[1., 2., 3.],
       [4., 5., 6.]])
```

```
In [23]: b = a = np.arange(25, 50).reshape((5, 5))

In [24]: b = np.arange(25, 50).reshape((5, 5))

In [25]: b
Out[25]:
array([[25, 26, 27, 28, 29],
       [30, 31, 32, 33, 34],
       [35, 36, 37, 38, 39],
       [40, 41, 42, 43, 44],
       [45, 46, 47, 48, 49]])

In [26]: c = np.arange(50, 75).reshape((5, 5))

In [27]: d = np.arange(75, 100).reshape((5, 5))

In [28]: np.block([[a, b], [c, d]])
Out[28]:
array([[25, 26, 27, 28, 29, 25, 26, 27, 28, 29],
       [30, 31, 32, 33, 34, 30, 31, 32, 33, 34],
       [35, 36, 37, 38, 39, 35, 36, 37, 38, 39],
       [40, 41, 42, 43, 44, 40, 41, 42, 43, 44],
       [45, 46, 47, 48, 49, 45, 46, 47, 48, 49],
       [50, 51, 52, 53, 54, 75, 76, 77, 78, 79],
       [55, 56, 57, 58, 59, 80, 81, 82, 83, 84],
       [60, 61, 62, 63, 64, 85, 86, 87, 88, 89],
       [65, 66, 67, 68, 69, 90, 91, 92, 93, 94],
       [70, 71, 72, 73, 74, 95, 96, 97, 98, 99]])

In [29]: a = np.arange(25).reshape((5, 5))

In [30]: np.block([[a, b], [c, d]])
Out[30]:
array([[ 0,  1,  2,  3,  4, 25, 26, 27, 28, 29],
       [ 5,  6,  7,  8,  9, 30, 31, 32, 33, 34],
       [10, 11, 12, 13, 14, 35, 36, 37, 38, 39],
       [15, 16, 17, 18, 19, 40, 41, 42, 43, 44],
       [20, 21, 22, 23, 24, 45, 46, 47, 48, 49],
       [50, 51, 52, 53, 54, 75, 76, 77, 78, 79],
       [55, 56, 57, 58, 59, 80, 81, 82, 83, 84],
       [60, 61, 62, 63, 64, 85, 86, 87, 88, 89],
       [65, 66, 67, 68, 69, 90, 91, 92, 93, 94],
       [70, 71, 72, 73, 74, 95, 96, 97, 98, 99]])

In [31]: a[-1]
Out[31]: array([20, 21, 22, 23, 24])

In [32]: a[1, 4]
Out[32]: 9

In [33]: a[1]
Out[33]: array([5, 6, 7, 8, 9])

In [34]: a[0:5]
Out[34]:
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24]])

In [35]: a[-5:]
Out[35]:
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24]])

In [36]: a = np.block([[a, b], [c, d]])
```

```
In [36]: a = np.block([[a, b], [c, d]])

In [37]: a
Out[37]:
array([[ 0,  1,  2,  3,  4, 25, 26, 27, 28, 29],
       [ 5,  6,  7,  8,  9, 30, 31, 32, 33, 34],
       [10, 11, 12, 13, 14, 35, 36, 37, 38, 39],
       [15, 16, 17, 18, 19, 40, 41, 42, 43, 44],
       [20, 21, 22, 23, 24, 45, 46, 47, 48, 49],
       [50, 51, 52, 53, 54, 75, 76, 77, 78, 79],
       [55, 56, 57, 58, 59, 80, 81, 82, 83, 84],
       [60, 61, 62, 63, 64, 85, 86, 87, 88, 89],
       [65, 66, 67, 68, 69, 90, 91, 92, 93, 94],
       [70, 71, 72, 73, 74, 95, 96, 97, 98, 99]])

In [38]: a[0:3, 4:9]
Out[38]:
array([[ 4, 25, 26, 27, 28],
       [ 9, 30, 31, 32, 33],
       [14, 35, 36, 37, 38]])

In [39]: a[np.ix_([1, 3, 4], [0, 2])]
Out[39]:
array([[ 5,  7],
       [15, 17],
       [20, 22]])

In [40]: a[2:21:2,:]
Out[40]:
array([[10, 11, 12, 13, 14, 35, 36, 37, 38, 39],
       [20, 21, 22, 23, 24, 45, 46, 47, 48, 49],
       [55, 56, 57, 58, 59, 80, 81, 82, 83, 84],
       [65, 66, 67, 68, 69, 90, 91, 92, 93, 94]])

In [41]: a[ ::2,:]
Out[41]:
array([[ 0,  1,  2,  3,  4, 25, 26, 27, 28, 29],
       [10, 11, 12, 13, 14, 35, 36, 37, 38, 39],
       [20, 21, 22, 23, 24, 45, 46, 47, 48, 49],
       [55, 56, 57, 58, 59, 80, 81, 82, 83, 84],
       [65, 66, 67, 68, 69, 90, 91, 92, 93, 94]])
```

```
In [42]: a[::-1,:]
Out[42]:
array([[70, 71, 72, 73, 74, 95, 96, 97, 98, 99],
       [65, 66, 67, 68, 69, 90, 91, 92, 93, 94],
       [60, 61, 62, 63, 64, 85, 86, 87, 88, 89],
       [55, 56, 57, 58, 59, 80, 81, 82, 83, 84],
       [50, 51, 52, 53, 54, 75, 76, 77, 78, 79],
       [20, 21, 22, 23, 24, 45, 46, 47, 48, 49],
       [15, 16, 17, 18, 19, 40, 41, 42, 43, 44],
       [10, 11, 12, 13, 14, 35, 36, 37, 38, 39],
       [ 5,  6,  7,  8,  9, 30, 31, 32, 33, 34],
       [ 0,  1,  2,  3,  4, 25, 26, 27, 28, 29]])

In [43]: a[np.r_[:len(a),0]]
Out[43]:
array([[ 0,  1,  2,  3,  4, 25, 26, 27, 28, 29],
       [ 5,  6,  7,  8,  9, 30, 31, 32, 33, 34],
       [10, 11, 12, 13, 14, 35, 36, 37, 38, 39],
       [15, 16, 17, 18, 19, 40, 41, 42, 43, 44],
       [20, 21, 22, 23, 24, 45, 46, 47, 48, 49],
       [50, 51, 52, 53, 54, 75, 76, 77, 78, 79],
       [55, 56, 57, 58, 59, 80, 81, 82, 83, 84],
       [60, 61, 62, 63, 64, 85, 86, 87, 88, 89],
       [65, 66, 67, 68, 69, 90, 91, 92, 93, 94],
       [70, 71, 72, 73, 74, 95, 96, 97, 98, 99],
       [ 0,  1,  2,  3,  4, 25, 26, 27, 28, 29]])

In [44]: a.transpose()
Out[44]:
array([[ 0,  5, 10, 15, 20, 50, 55, 60, 65, 70],
       [ 1,  6, 11, 16, 21, 51, 56, 61, 66, 71],
       [ 2,  7, 12, 17, 22, 52, 57, 62, 67, 72],
       [ 3,  8, 13, 18, 23, 53, 58, 63, 68, 73],
       [ 4,  9, 14, 19, 24, 54, 59, 64, 69, 74],
       [25, 30, 35, 40, 45, 75, 80, 85, 90, 95],
       [26, 31, 36, 41, 46, 76, 81, 86, 91, 96],
       [27, 32, 37, 42, 47, 77, 82, 87, 92, 97],
       [28, 33, 38, 43, 48, 78, 83, 88, 93, 98],
       [29, 34, 39, 44, 49, 79, 84, 89, 94, 99]])

In [45]: a.conj().transpose()
Out[45]:
array([[ 0,  5, 10, 15, 20, 50, 55, 60, 65, 70],
       [ 1,  6, 11, 16, 21, 51, 56, 61, 66, 71],
       [ 2,  7, 12, 17, 22, 52, 57, 62, 67, 72],
       [ 3,  8, 13, 18, 23, 53, 58, 63, 68, 73],
       [ 4,  9, 14, 19, 24, 54, 59, 64, 69, 74],
       [25, 30, 35, 40, 45, 75, 80, 85, 90, 95],
       [26, 31, 36, 41, 46, 76, 81, 86, 91, 96],
       [27, 32, 37, 42, 47, 77, 82, 87, 92, 97],
       [28, 33, 38, 43, 48, 78, 83, 88, 93, 98],
       [29, 34, 39, 44, 49, 79, 84, 89, 94, 99]])
```

```
In [46]: b = a

In [47]: b
Out[47]:
array([[ 0,  1,  2,  3,  4, 25, 26, 27, 28, 29],
       [ 5,  6,  7,  8,  9, 30, 31, 32, 33, 34],
       [10, 11, 12, 13, 14, 35, 36, 37, 38, 39],
       [15, 16, 17, 18, 19, 40, 41, 42, 43, 44],
       [20, 21, 22, 23, 24, 45, 46, 47, 48, 49],
       [50, 51, 52, 53, 54, 75, 76, 77, 78, 79],
       [55, 56, 57, 58, 59, 80, 81, 82, 83, 84],
       [60, 61, 62, 63, 64, 85, 86, 87, 88, 89],
       [65, 66, 67, 68, 69, 90, 91, 92, 93, 94],
       [70, 71, 72, 73, 74, 95, 96, 97, 98, 99]])

In [48]: a @ b
Out[48]:
array([[ 8300,  8445,  8590,  8735,  8880, 11925, 12070, 12215, 12360,
        12505],
       [10050, 10245, 10440, 10635, 10830, 14925, 15120, 15315, 15510,
        15705],
       [11800, 12045, 12290, 12535, 12780, 17925, 18170, 18415, 18660,
        18905],
       [13550, 13845, 14140, 14435, 14730, 20925, 21220, 21515, 21810,
        22105],
       [15300, 15645, 15990, 16335, 16680, 23925, 24270, 24615, 24960,
        25305],
       [25800, 26445, 27090, 27735, 28380, 41925, 42570, 43215, 43860,
        44505],
       [27550, 28245, 28940, 29635, 30330, 44925, 45620, 46315, 47010,
        47705],
       [29300, 30045, 30790, 31535, 32280, 47925, 48670, 49415, 50160,
        50905],
       [31050, 31845, 32640, 33435, 34230, 50925, 51720, 52515, 53310,
        54105],
       [32800, 33645, 34490, 35335, 36180, 53925, 54770, 55615, 56460,
        57305]])

In [49]: a * b
Out[49]:
array([[   0,    1,    4,    9,   16,  625,  676,  729,  784,  841],
       [  25,   36,   49,   64,   81,  900,  961, 1024, 1089, 1156],
       [ 100,  121,  144,  169,  196, 1225, 1296, 1369, 1444, 1521],
       [ 225,  256,  289,  324,  361, 1600, 1681, 1764, 1849, 1936],
       [ 400,  441,  484,  529,  576, 2025, 2116, 2209, 2304, 2401],
       [2500, 2601, 2704, 2809, 2916, 5625, 5776, 5929, 6084, 6241],
       [3025, 3136, 3249, 3364, 3481, 6400, 6561, 6724, 6889, 7056],
       [3600, 3721, 3844, 3969, 4096, 7225, 7396, 7569, 7744, 7921],
       [4225, 4356, 4489, 4624, 4761, 8100, 8281, 8464, 8649, 8836],
       [4900, 5041, 5184, 5329, 5476, 9025, 9216, 9409, 9604, 9801]])

In [50]: a/b
<ipython-input-50-aae42d317509>:1: RuntimeWarning: invalid value encountered in true_divide
  a/b
Out[50]:
array([[nan,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.]])
```

```
In [51]: a**3
Out[51]:
array([[     0,      1,      8,     27,     64,  15625,  17576,  19683,
         21952,  24389],
       [   125,    216,    343,    512,    729,  27000,  29791,  32768,
         35937,  39304],
       [  1000,   1331,   1728,   2197,   2744,  42875,  46656,  50653,
         54872,  59319],
       [  3375,   4096,   4913,   5832,   6859,  64000,  68921,  74088,
         79507,  85184],
       [  8000,   9261,  10648,  12167,  13824,  91125,  97336, 103823,
        110592, 117649],
       [125000, 132651, 140608, 148877, 157464, 421875, 438976, 456533,
        474552, 493039],
       [166375, 175616, 185193, 195112, 205379, 512000, 531441, 551368,
        571787, 592704],
       [216000, 226981, 238328, 250047, 262144, 614125, 636056, 658503,
        681472, 704969],
       [274625, 287496, 300763, 314432, 328509, 729000, 753571, 778688,
        804357, 830584],
       [343000, 357911, 373248, 389017, 405224, 857375, 884736, 912673,
        941192, 970299]])

In [52]: (a > 0.5)
Out[52]:
array([[False,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True]])

In [53]: np.nonzero(a > 0.5)
Out[53]:
(array([0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4,
        4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6,
        6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8,
        8, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9]),
 array([1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2,
        3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4,
        5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6,
        7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8,
        9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]))
```

```
In [56]: v = np.array([[2],[1],[3],[2],[1],[3],[2],[1],[3],[2]])

In [57]: v
Out[57]:
array([[2],
       [1],
       [3],
       [2],
       [1],
       [3],
       [2],
       [1],
       [3],
       [2]])

In [58]: a[:,np.nonzero(v > 0.5)[0]]
Out[58]:
array([[ 0,  1,  2,  3,  4, 25, 26, 27, 28, 29],
       [ 5,  6,  7,  8,  9, 30, 31, 32, 33, 34],
       [10, 11, 12, 13, 14, 35, 36, 37, 38, 39],
       [15, 16, 17, 18, 19, 40, 41, 42, 43, 44],
       [20, 21, 22, 23, 24, 45, 46, 47, 48, 49],
       [50, 51, 52, 53, 54, 75, 76, 77, 78, 79],
       [55, 56, 57, 58, 59, 80, 81, 82, 83, 84],
       [60, 61, 62, 63, 64, 85, 86, 87, 88, 89],
       [65, 66, 67, 68, 69, 90, 91, 92, 93, 94],
       [70, 71, 72, 73, 74, 95, 96, 97, 98, 99]])
```

```
In [79]: v[v.T[v.T > 8]]
Out[79]: array([[2]])

In [80]: a[a < 0.5]=0

In [81]: a
Out[81]:
array([[ 0,  1,  2,  3,  4, 25, 26, 27, 28, 29],
       [ 5,  6,  7,  8,  9, 30, 31, 32, 33, 34],
       [10, 11, 12, 13, 14, 35, 36, 37, 38, 39],
       [15, 16, 17, 18, 19, 40, 41, 42, 43, 44],
       [20, 21, 22, 23, 24, 45, 46, 47, 48, 49],
       [50, 51, 52, 53, 54, 75, 76, 77, 78, 79],
       [55, 56, 57, 58, 59, 80, 81, 82, 83, 84],
       [60, 61, 62, 63, 64, 85, 86, 87, 88, 89],
       [65, 66, 67, 68, 69, 90, 91, 92, 93, 94],
       [70, 71, 72, 73, 74, 95, 96, 97, 98, 99]])

In [82]: a * (a > 0.5)
Out[82]:
array([[ 0,  1,  2,  3,  4, 25, 26, 27, 28, 29],
       [ 5,  6,  7,  8,  9, 30, 31, 32, 33, 34],
       [10, 11, 12, 13, 14, 35, 36, 37, 38, 39],
       [15, 16, 17, 18, 19, 40, 41, 42, 43, 44],
       [20, 21, 22, 23, 24, 45, 46, 47, 48, 49],
       [50, 51, 52, 53, 54, 75, 76, 77, 78, 79],
       [55, 56, 57, 58, 59, 80, 81, 82, 83, 84],
       [60, 61, 62, 63, 64, 85, 86, 87, 88, 89],
       [65, 66, 67, 68, 69, 90, 91, 92, 93, 94],
       [70, 71, 72, 73, 74, 95, 96, 97, 98, 99]])

In [83]: a[:] = 3

In [84]: a
Out[84]:
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3]])

In [85]: x = a
```

```
In [85]: x = a

In [86]: x
Out[86]:
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3]])

In [87]: y = x.copy()

In [88]: y
Out[88]:
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3]])

In [89]: y = x[1, :].copy()

In [90]: y
Out[90]: array([3, 3, 3, 3, 3, 3, 3, 3, 3, 3])

In [91]: y = x.flatten()

In [92]: y
Out[92]:
array([3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3])

In [93]: np.arange(1., 11.)
Out[93]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

```
In [95]: np.arange(10.)
Out[95]: array([0., 1., 2., 3., 4., 5., 6., 7., 8., 9.])

In [96]: np.arange(1.,11.)[:, np.newaxis]
Out[96]:
array([[ 1.],
       [ 2.],
       [ 3.],
       [ 4.],
       [ 5.],
       [ 6.],
       [ 7.],
       [ 8.],
       [ 9.],
       [10.]])

In [97]: np.zeros((3, 4))
Out[97]:
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])

In [98]: np.zeros((3, 4, 5))
Out[98]:
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 [99]: np.ones((3, 4))
Out[99]:
array([[1., 1., 1., 1.],
       [1., 1., 1., 1.],
       [1., 1., 1., 1.]])

In [100]: np.eye(3)
Out[100]:
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])

In [101]: np.diag(a)
Out[101]: array([3, 3, 3, 3, 3, 3, 3, 3, 3, 3])
```

```
In [114]: np.random.rand(3, 4)
Out[114]:
array([[0.95942051, 0.06768078, 0.86393121, 0.88234934],
       [0.02317476, 0.23053645, 0.17566978, 0.49368622],
       [0.58141339, 0.50972327, 0.10926157, 0.42069358]])

In [115]: np.linspace(1,3,4)
Out[115]: array([1.        , 1.66666667, 2.33333333, 3.        ])

In [116]: np.mgrid[0:9.,0:6.]
Out[116]:
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 [122]: np.meshgrid([1,2,4],[2,4,5])
Out[122]:
[array([[1, 2, 4],
        [1, 2, 4],
        [1, 2, 4]]),
 array([[2, 2, 2],
        [4, 4, 4],
        [5, 5, 5]])]

In [123]: np.tile(a, (3, 2))
Out[123]:
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]])

In [124]: np.concatenate((a,b),1)
Out[124]:
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]])
```

```
In [125]: np.concatenate((a,b))
Out[125]:
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3]])

In [126]: a.max()
Out[126]: 3

In [127]: a.max(0)
Out[127]: array([3, 3, 3, 3, 3, 3, 3, 3, 3, 3])

In [128]: a.max(1)
Out[128]: array([3, 3, 3, 3, 3, 3, 3, 3, 3, 3])

In [129]: np.maximum(a, b)
Out[129]:
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3]])
```

```
In [137]: np.linalg.norm(v)
Out[137]: 11.090536506409418

In [138]: np.logical_and(a,b)
Out[138]:
array([[ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True]])

In [139]: np.logical_or(a,b)
Out[139]:
array([[ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True]])
```

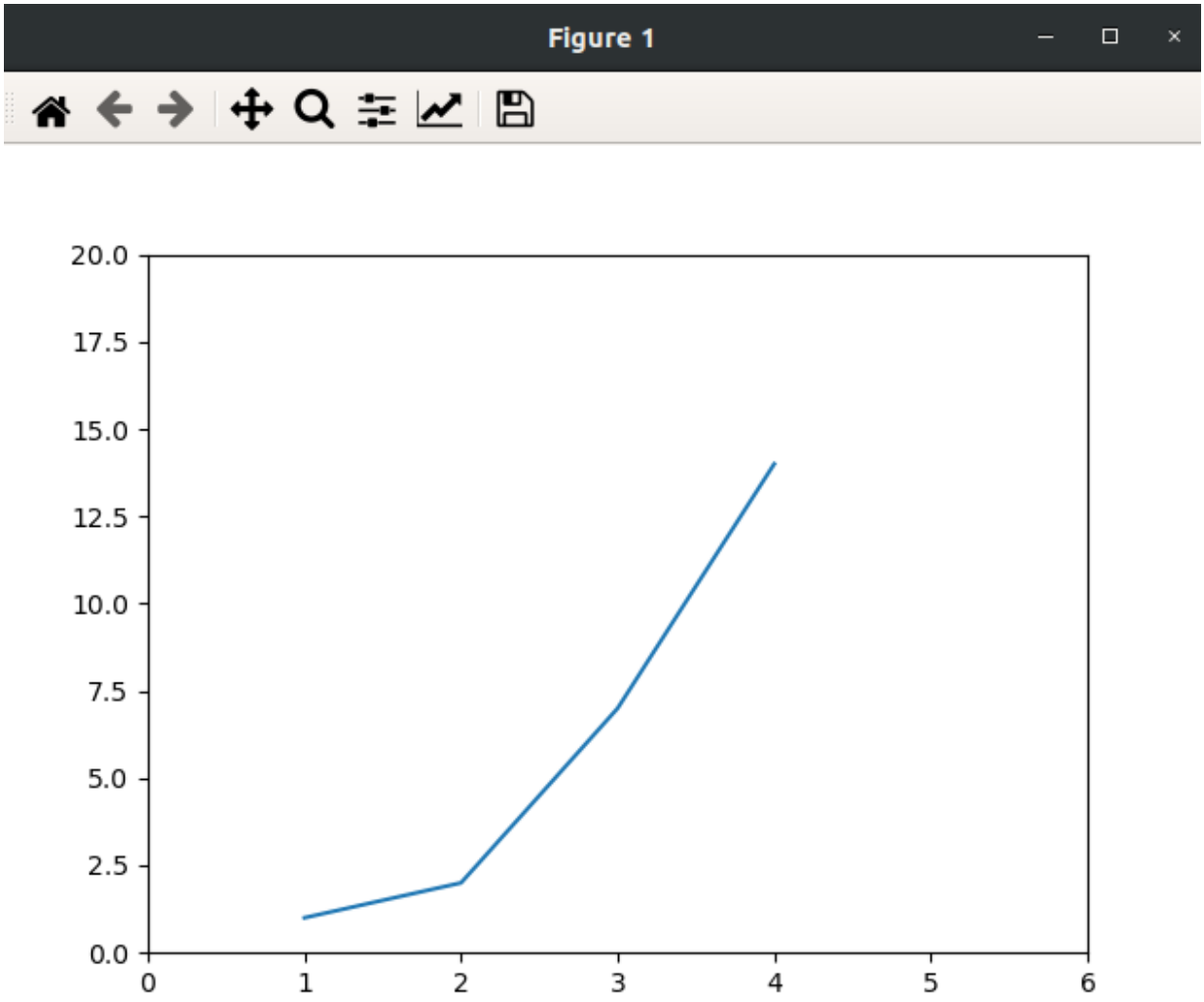
```
In [140]: ^I
     ...: a & b
Out[140]:
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3]])

In [141]: a | b
Out[141]:
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3]])
```

```
In [143]: a
Out[143]:
array([[3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3, 3, 3, 3, 3, 3]])

In [144]: a = np.array([[1., 7.], [4., 2.]])

In [145]: a
Out[145]:
array([[1., 7.],
       [4., 2.]])

In [146]: linalg.inv(a)
Out[146]:
array([[-0.07692308,  0.26923077],
       [ 0.15384615, -0.03846154]])

In [147]: linalg.pinv(a)
Out[147]:
array([[-0.07692308,  0.26923077],
       [ 0.15384615, -0.03846154]])
```

```
In [159]: np.fft.fft(a)
Out[159]:
array([[ 8.+0.j, -6.+0.j],
       [ 6.+0.j,  2.+0.j]])

In [160]: np.fft.ifft(a)
Out[160]:
array([[ 4.+0.j, -3.+0.j],
       [ 3.+0.j,  1.+0.j]])

In [161]: np.sort(a)
Out[161]:
array([[1., 7.],
       [2., 4.]])

In [162]: np.sort(a, axis = 1)
Out[162]:
array([[1., 7.],
       [2., 4.]])

In [163]: I = np.argsort(a[:, 0])

In [164]: b = a[I,:]

In [165]: I
Out[165]: array([0, 1])
```

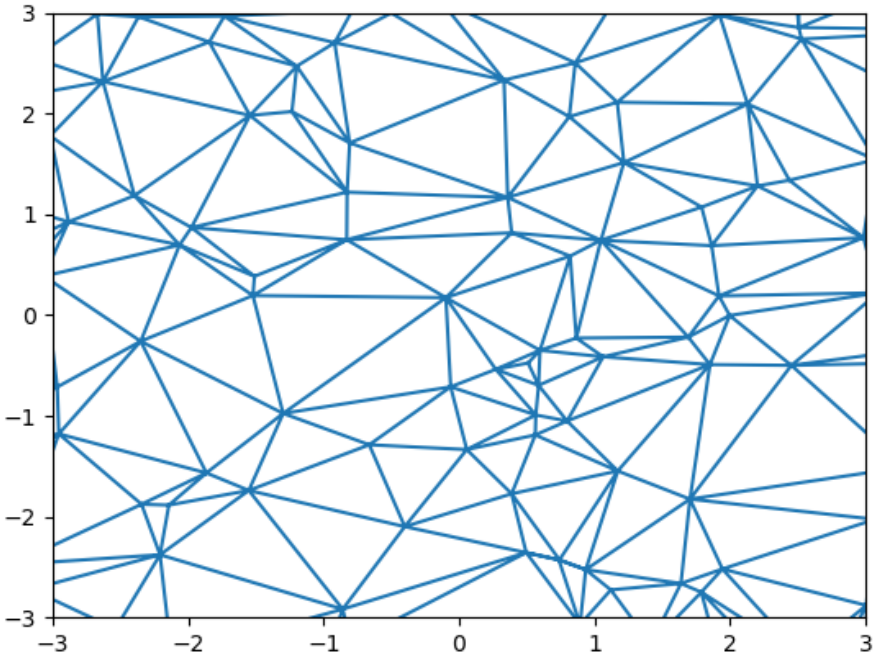
```
In [168]: np.unique(a)
Out[168]: array([1., 2., 4., 7.])

In [169]: a.squeeze()
Out[169]:
array([[1., 7.],
       [4., 2.]])
```

Task 3:



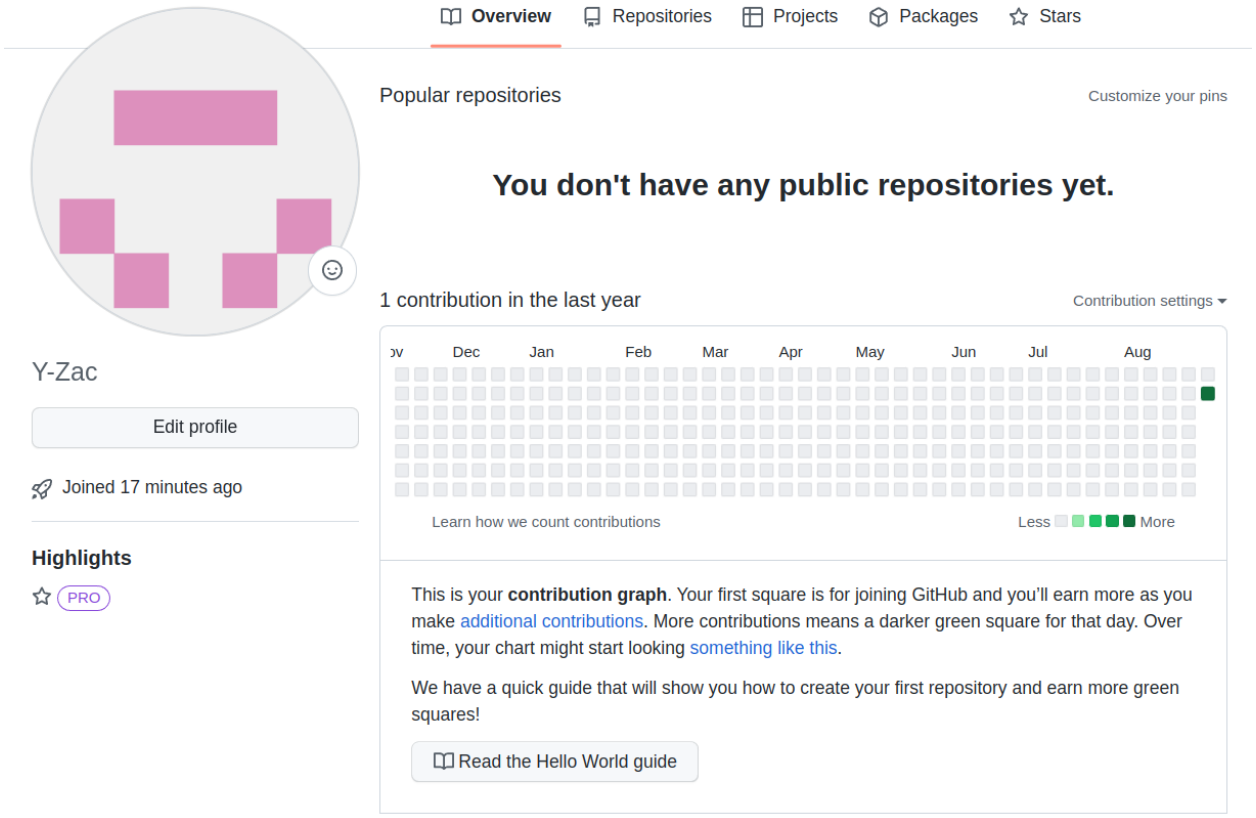
Task 4:

```
In [10]: np.random.seed(1)

In [11]: x = np.random.uniform(-5, 5, 256)

In [12]: y = np.random.uniform(-5, 5, 256)

In [13]: z = (1 - x/2 + x**5 + y**3) * np.exp(-x**2 - y**2)

In [14]: fig, ax = plt.subplots()

In [15]: ax.triplot(x, y)
Out[15]:
[<matplotlib.lines.Line2D at 0x7f562a04b610>,
 <matplotlib.lines.Line2D at 0x7f562a066220>]

In [16]: ax.set(xlim=(-3, 3), ylim=(-3, 3))
Out[16]: [(-3.0, 3.0), (-3.0, 3.0)]

In [17]: plt.show()
```



Task 5:

Github Account: Y-Zac



Task 6:

Public Project Link: https://github.com/Y-Zac/COMP-582-Assignment0