

09_Numpy_3

May 10, 2022

Colab Link: <https://colab.research.google.com/drive/1qPeU7TMFJbmUdO8oiGcIB5jJLIQ2dJ3q?usp=sharing>

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

```
[ ]: # Broadcasting
```

```
[ ]: x = np.array([[0, 10, 20, 30]])  
x
```

```
[ ]: array([[ 0, 10, 20, 30]])
```

```
[ ]: x.shape
```

```
[ ]: (1, 4)
```

```
[ ]: y = np.array([[1], [2], [3], [4]])  
y
```

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

```
[ ]: y.shape
```

```
[ ]: (4, 1)
```

```
[ ]: x + y
```

```
[ ]: array([[ 1, 11, 21, 31],  
          [ 2, 12, 22, 32],  
          [ 3, 13, 23, 33],  
          [ 4, 14, 24, 34]])
```

```
[ ]: x = np.arange(16).reshape(4, 4)
```

```
[ ]: x
```

```
[ ]: array([[ 0,  1,  2,  3],
           [ 4,  5,  6,  7],
           [ 8,  9, 10, 11],
           [12, 13, 14, 15]])
```

```
[ ]: y = np.arange(16).reshape(4, 4)
```

```
[ ]: x + y
```

```
[ ]: array([[ 0,  2,  4,  6],
           [ 8, 10, 12, 14],
           [16, 18, 20, 22],
           [24, 26, 28, 30]])
```

```
[ ]: x = np.array([[0, 10, 20]])
     x.shape
```

```
[ ]: (1, 3)
```

```
[ ]: x
```

```
[ ]: array([[ 0, 10, 20]])
```

```
[ ]: y = np.array([[1], [2], [3], [4]])
     y.shape
```

```
[ ]: (4, 1)
```

```
[ ]: x + y
```

```
[ ]: array([[ 1, 11, 21],
           [ 2, 12, 22],
           [ 3, 13, 23],
           [ 4, 14, 24]])
```

```
[ ]: x = np.arange(3).reshape(1, 3)
     y = np.arange(4).reshape(4, 1)
     x + y
```

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

```
[ ]: x
```

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

```
[ ]: y
```

```
[ ]: array([[ 0,  1,  2,  3],
           [ 4,  5,  6,  7],
           [ 8,  9, 10, 11],
           [12, 13, 14, 15]])
```

```
[ ]: x.shape, y.shape
```

```
[ ]: ((2, 4), (4, 4))
```

```
[ ]: 4 / 2
```

```
[ ]: 2.5
```

```
[ ]: x + y
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-169-cd60f97aa77f> in <module>()
----> 1 x + y

ValueError: operands could not be broadcast together with shapes (2,4) (4,4)
```

```
[ ]: # Broadcasting Works When
```

```
# - both arrays same shape
# - one of the dimension match (in some cases), and the other is 1
```

```
[ ]: # !gdown 17tYTDPU5hpby9t0kGd7w_-zBsby7sEd
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]: # !gdown 1o-8yqdTM7cfz_mAaNCi2nH0urFu7pcqI
```

```
[ ]: ## Splitting
```

```
[ ]: x = np.arange(9) * 5
     x
```

```
[ ]: array([ 0,  5, 10, 15, 20, 25, 30, 35, 40])
```

```
[ ]: np.split(x, 3)
```

```
[ ]: [array([ 0,  5, 10]), array([15, 20, 25]), array([30, 35, 40])]
```

```
[ ]: ? np.split
```

```
[ ]: np.split(x, [2, 3, 5])
```

```
[ ]: [array([0, 5]), array([10]), array([15, 20]), array([25, 30, 35, 40])]
```

```
[ ]: x = np.arange(16).reshape(4, 4)
x
```

```
[ ]: array([[ 0,  1,  2,  3],
          [ 4,  5,  6,  7],
          [ 8,  9, 10, 11],
          [12, 13, 14, 15]])
```

```
[ ]: np.split(x, 4, axis=1)
```

```
[ ]: [array([[ 0],
          [ 4],
          [ 8],
          [12]]), array([[ 1],
          [ 5],
          [ 9],
          [13]]), array([[ 2],
          [ 6],
          [10],
          [14]]), array([[ 3],
          [ 7],
          [11],
          [15]])]
```

```
[ ]: np.vsplit(x, 4)
```

```
[ ]: [array([[0, 1, 2, 3]]),
      array([[4, 5, 6, 7]]),
      array([[ 8,  9, 10, 11]]),
      array([[12, 13, 14, 15]])]
```

```
[ ]: ## Merging
```

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

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

```
[ ]: _ = np.hstack((a, a, a))
```

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

```
[ ]: np.split(_, 3)
```

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

```
[ ]: b = np.array([5, 6, 7, 8])
```

```
[ ]: np.hstack((a, b, b))
```

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

```
[ ]: np.vstack((a, b))
```

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

```
[ ]: np.stack((a,b), axis=1)
```

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

```
[ ]: x = np.arange(8).reshape(2, 4)  
     np.hstack((x, x))
```

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

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

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

```
[ ]: ## Copies of arrays
```

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

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

```
[ ]: y = x
y
```

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

```
[ ]: x[2] = -75
x
```

```
[ ]: array([ 1,  2, -75,  4])
```

```
[ ]: y
```

```
[ ]: array([ 1,  2, -75,  4])
```

```
[ ]: RAM = [.....x1 (32 bytes)..y1 | (32 bytes) [0, 1, -75, 2]..]
```

```
x = np.array([....])
```

```
X-Metdata = x_id = epfjdsbjv3hg, 32 bytes, 1, 2, -75, 4
```

```
x = X
```

```
x = x_id
```

```
X-Metdata = y_id = epfjoi3hg, 32 bytes, 1, 2, 3, 4
```

```
[ ]: y = x
```

```
y = x.copy()
```

```
[ ]: x
```

```
[ ]: array([ 1,  2, -75,  4])
```

```
[ ]: y
```

```
[ ]: array([ 1,  2, -75,  4])
```

```
[ ]: x[2] = 1000
x
```

```
[ ]: array([ 1, 2, 1000, 4])
```

```
[ ]: y
```

```
[ ]: array([ 1, 2, -75, 4])
```

```
[ ]: z = x
```

```
[ ]: x
```

```
[ ]: array([ 1, 2, 1000, 4])
```

```
[ ]: z
```

```
[ ]: array([ 1, 2, 1000, 4])
```

```
[ ]: x[2] = -90
```

```
[ ]:
```

```
[ ]: array([ 1, 2, -90, 4])
```

```
[ ]: x = np.array([96, 56, 64, 75])
```

```
[ ]: z
```

```
[ ]: array([ 1, 2, -90, 4])
```

```
[ ]: # quiz  
  
a = np.array([0, 1, 2, 3, 4, 5])  
b = a[a%1 == 0]  
b[0] = 10  
a[:2]
```

```
[ ]: array([10, 1])
```

```
[ ]: a%1 == 0
```

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

```
[ ]: b
```

```
[ ]: array([10, 2, 3])
```

```
[ ]: ## 3 Days
```

```
[ ]: x = np.arange(8).reshape(2, 2, 2)
x
```

```
[ ]: array([[[0, 1],
           [2, 3]],

          [[4, 5],
           [6, 7]]])
```

```
[ ]: x[0, :, :]
```

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

```
[ ]: x[0, :1, :]
```

```
[ ]: array([[0, 1]])
```

```
[ ]: x + x
```

```
[ ]: array([[[ 0,  2],
           [ 4,  6]],

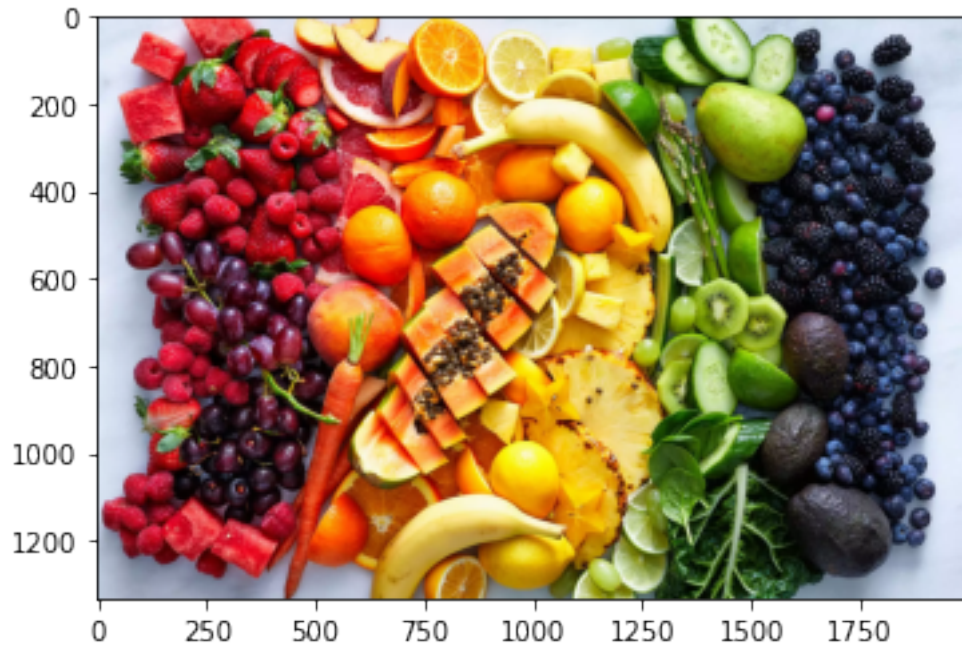
          [[ 8, 10],
           [12, 14]]])
```

```
[ ]: ## Images with Numpy.
```

```
[ ]: import matplotlib.pyplot as plt
```

```
[ ]: x = plt.imread('fruits.png')
plt.imshow(x)
```

```
[ ]: <matplotlib.image.AxesImage at 0x7f450d8737d0>
```

```
[ ]: x.shape
```

```
[ ]: (1333, 2000, 3)
```

```
[ ]: # R, G, B
```

```
# R = (1333, 2000)
```

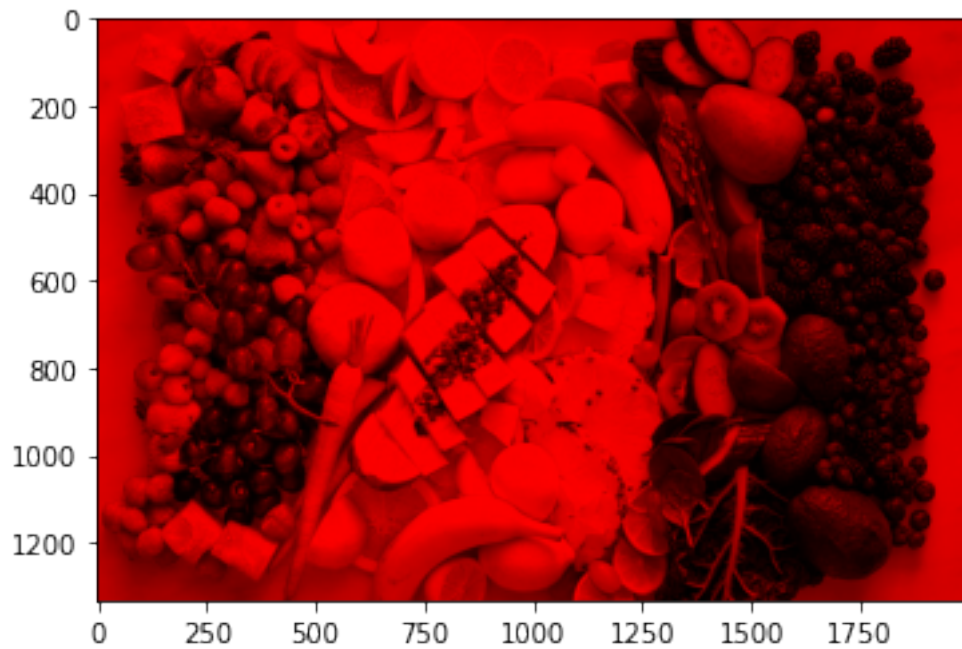
```
# R = [[124, 200], [80, 90]]
```

```
# B = [[104, 100], [80, 90]]
```

```
# G = [[75, 100], [80, 90]]
```

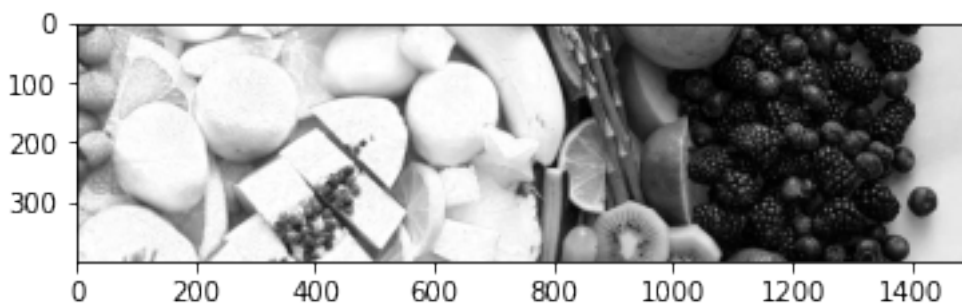
```
[ ]: r = x.copy()
r[:, :, (1, 2)] = 0
# r[:, :, 2] = 0
plt.imshow(r)
```

```
[ ]: <matplotlib.image.AxesImage at 0x7f450dc61f10>
```



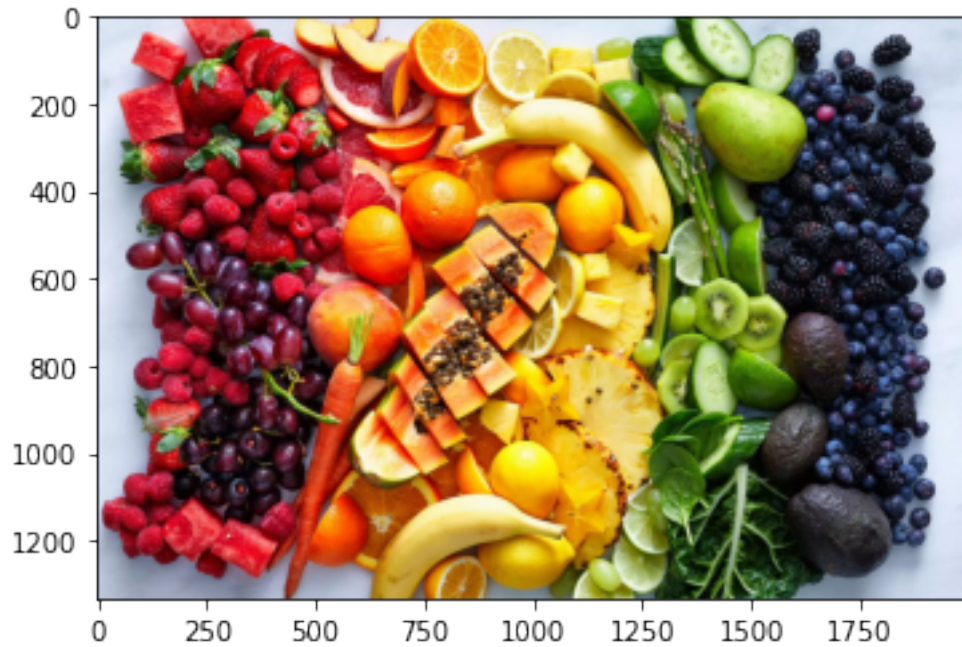
```
[ ]: c = x.copy()
c = c[300:700, 500:9000, 0]
plt.imshow(c, cmap='gray')
```

```
[ ]: <matplotlib.image.AxesImage at 0x7f450d105710>
```



```
[ ]: plt.imshow(x[:, :, :])
```

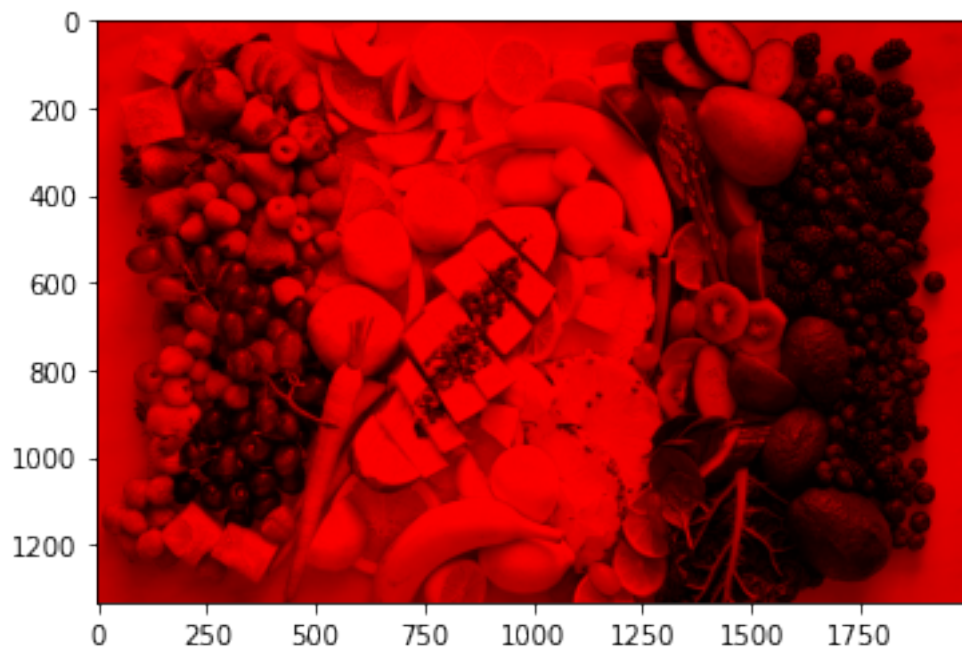
```
[ ]: <matplotlib.image.AxesImage at 0x7f450d211990>
```



```
[ ]: R_img, B_img, G_img = x.copy(), x.copy(), x.copy()
     R_img[:, :, (1, 2)] = 0
     G_img[:, :, (0, 2)] = 0
     B_img[:, :, (0, 1)] = 0
```

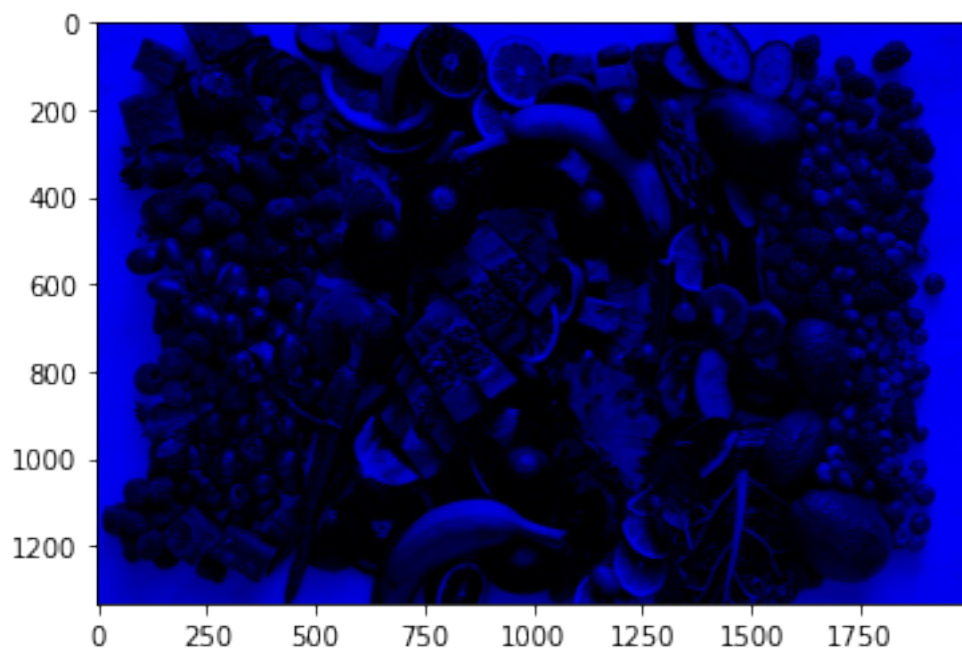
```
[ ]: plt.imshow(R_img)
```

```
[ ]: <matplotlib.image.AxesImage at 0x7f450d7ed750>
```



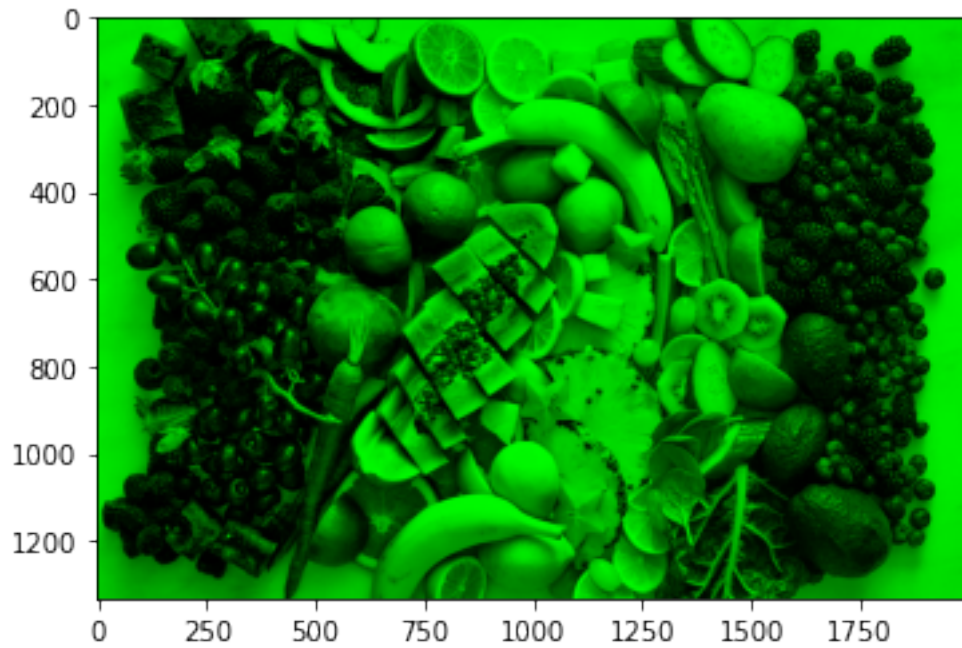
```
[ ]: plt.imshow(B_img)
```

```
[ ]: <matplotlib.image.AxesImage at 0x7f450d748f50>
```



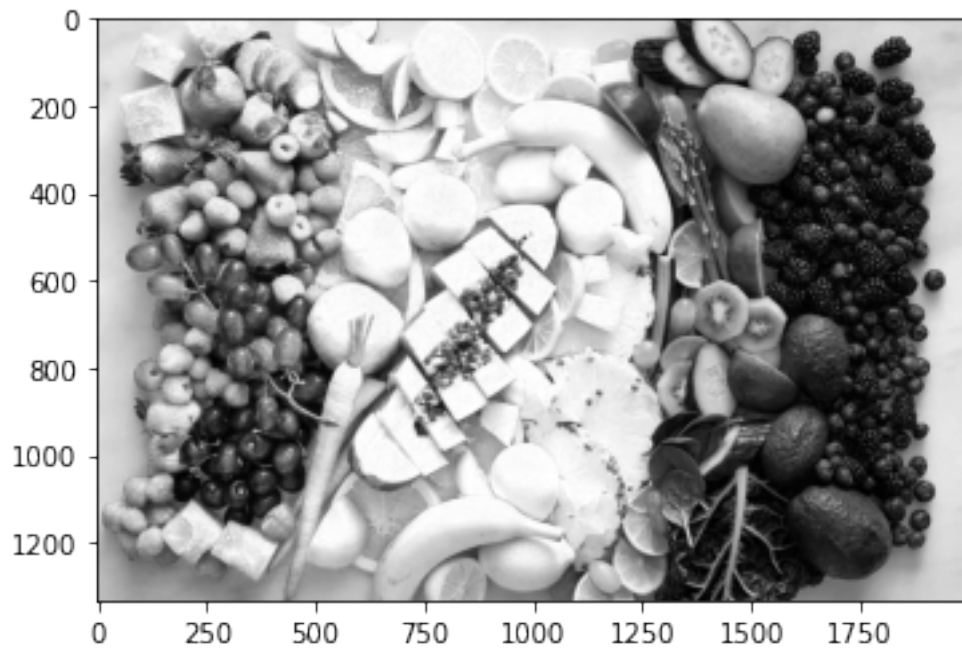
```
[ ]: plt.imshow(G_img)
```

```
[ ]: <matplotlib.image.AxesImage at 0x7f450d6bfb10>
```



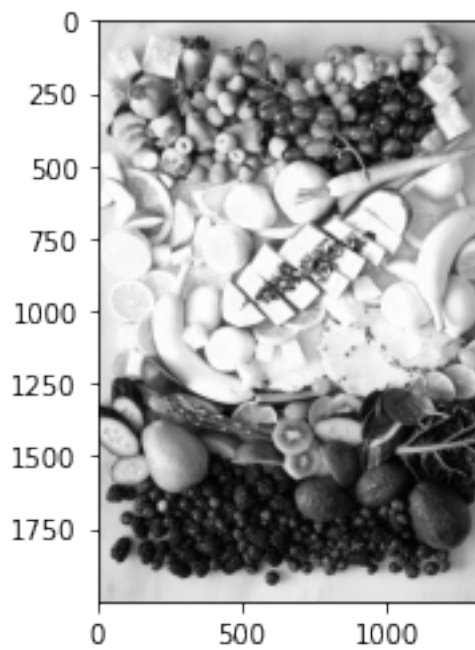
```
[ ]: plt.imshow(x[:, :, 0], cmap='gray')
```

```
[ ]: <matplotlib.image.AxesImage at 0x7f450d5bd250>
```

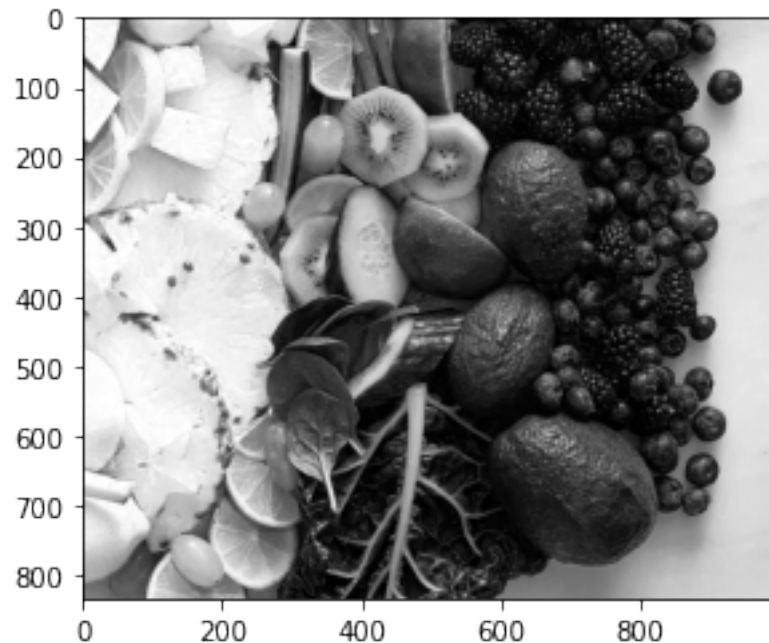
```
[ ]: plt.imshow(np.transpose(x[:, :, 0]), cmap='gray')
```

```
[ ]: <matplotlib.image.AxesImage at 0x7f450d5a1690>
```



```
[ ]: plt.imshow(x[500:, 1000:, 0], cmap='gray')
```

```
[ ]: <matplotlib.image.AxesImage at 0x7f450d47b490>
```



```
[ ]: X = np.arange(12).reshape((3, 4))
row = np.array([0, 1, 2])
mask = np.array([1, 0, 1, 0], dtype=bool)
print(X[row[:, np.newaxis], mask])
```

```
[[ 0  2]
 [ 4  6]
 [ 8 10]]
```

```
[ ]: X
```

```
[ ]: array([[ 0,  1,  2,  3],
          [ 4,  5,  6,  7],
          [ 8,  9, 10, 11]])
```

```
[ ]: row[:, np.newaxis]
```

```
[ ]: array([[0],
          [1],
          [2]])
```

```
[ ]: np.array([0], [1]).shape, np.array([0, 2]).shape
```

```
[ ]: ((2, 1), (2,))
```

```
[ ]: np.tile(np.array([0, 2]), (2, 1))
```

```
[ ]: array([[0, 2],  
          [0, 2]])
```

```
[ ]:
```

```
[ ]: np.array([[0], [1]]) + np.tile(np.array([0, 2]), (2, 1))
```

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

```
[ ]: (np.array([[0], [1]]) + np.array([0, 2]))
```

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

```
[ ]: X[[[0], [1]], [0, 2]]
```

```
[ ]: array([[0, 2],  
          [4, 6]])
```

```
[ ]: X[row[:, np.newaxis], mask]
```

```
[ ]: array([[ 0,  1,  2,  3],  
          [ 4,  5,  6,  7],  
          [ 8,  9, 10, 11]])
```

```
[ ]: row = np.arange(20)  
     row * 10
```

```
[ ]: array([ 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120,  
          130, 140, 150, 160, 170, 180, 190])
```

```
[ ]: def func(x):  
     x // 2  
  
     np.vecotrize(func)
```

```
[ ]: ans = []  
     for i in row:  
         ans.append(i*10)  
     np.array(ans)
```



```
[ ]: array([ 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120,
          130, 140, 150, 160, 170, 180, 190])
```

```
[ ]: row[:2, np.newaxis].shape
```

```
[ ]: (2, 1)
```

```
[ ]: mask
```

```
[ ]: array([ True, False,  True, False])
```

```
[ ]: X = np.arange(12).reshape((3, 4))
```

```
[ ]: X
```

```
[ ]: array([[ 0,  1,  2,  3],
          [ 4,  5,  6,  7],
          [ 8,  9, 10, 11])
```

```
[ ]: ? np.transpose
```

```
[ ]: arr = np.arange(48).reshape(2,4,3,2)
```

```
[ ]: arr[0, 1, 0, 0]
```

```
[ ]: 6
```

```
[ ]: arr
```

```
[ ]: 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, 25],
```

```

        [26, 27],
        [28, 29]],

        [[30, 31],
         [32, 33],
         [34, 35]],

        [[36, 37],
         [38, 39],
         [40, 41]],

        [[42, 43],
         [44, 45],
         [46, 47]]]])

```

```
[ ]: np.sum(arr, axis=1)
```

```
[ ]: array([[ 36,  40],
           [ 44,  48],
           [ 52,  56]],

           [[132, 136],
            [140, 144],
            [148, 152]])
```

```
[ ]: a = np.array([0,1,2]) #id(a[0]) >> someId id(a[0]) >> someAnotherId
```

```
[ ]: id(a)
```

```
[ ]: 139934548228208
```

```
[ ]: id(a)
```

```
[ ]: 139934548228208
```

```
[ ]: x = 1
     y =x * 2
```

```
[ ]: y
```

```
[ ]: 2
```

```
[ ]:
```