

Name: Chirantha Kurukulasuriya

Index Number: 190337X

Q1

```
In [ ]: for i in range(1, 6):  
        print(i, ': ', i**2)
```

```
1 : 1  
2 : 4  
3 : 9  
4 : 16  
5 : 25
```

Q2

```
In [ ]: from sympy import isprime  
  
for i in range(1, 6):  
    if not isprime(i):  
        print(i, ': ', i**2)
```

```
1 : 1  
4 : 16
```

Q3

```
In [ ]: A = [(i, i**2) for i in range(1,6)]  
for i in range(len(A)):  
    print(A[i][0], ': ', A[i][1])
```

```
1 : 1  
2 : 4  
3 : 9  
4 : 16  
5 : 25
```

Q4

```
In [ ]: A = [(i, i**2) for i in range(1,6) if not isprime(i)]  
for i in range(len(A)):  
    print(A[i][0], ': ', A[i][1])
```

```
1 : 1  
4 : 16
```

Q5

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

a) Matrix Multiplication

```
In [ ]: A = np.array([[1, 2], [3, 4], [5, 6]])
        B = np.array([[7, 8, 9, 1], [1, 2, 3, 4]])
        C = A @ B
        print(C)
```

```
[[ 9 12 15  9]
 [25 32 39 19]
 [41 52 63 29]]
```

b) Element-wise multiplication

```
In [ ]: A = np.array([[1, 2], [3, 4], [5, 6]])
        B = np.array([[3, 2], [5, 4], [3, 1]])
        C = A * B
        print(C)
```

```
[[ 3  4]
 [15 16]
 [15  6]]
```

Q6

```
In [ ]: A = np.random.randint(11, size = (5, 7)) # 5 x 7 array
        subA = A[1:4, 0:2] # array consisting rows 2 to 4 (inclusive) and first two columns
        print('A =\n', A)
        print('subarray of A =\n', subA)
        print('Shape of the sub-array: ', subA.shape)
        print('Size of the sub-array: ', subA.size)
```

```
A =
[[ 7  0  5  7  7  0  4]
 [ 9  7  3  8  4  0  3]
 [ 8  4  8  4  6 10  1]
 [ 7  4  6  6  6  4  6]
 [ 0  8  3  2  8  9  3]]
subarray of A =
[[9 7]
 [8 4]
 [7 4]]
Shape of the sub-array: (3, 2)
Size of the sub-array: 6
```

Q7

(i) An array and a scalar value are combined in an operation

```
In [ ]: a = np.array([1, 2, 3])
        b = 2.0
        print(a * b)
```

```
[2.  4.  6.]
```

```
In [ ]: a = np.array([[1, 2, 3], [4, 5, 6]])
        b = 2.0
        print(a * b)
```

```
[[ 2.  4.  6.]
 [ 8. 10. 12.]]
```

(ii) A 2D array and a 1D array combined in an operation

```
In [ ]: A = np.array([[1, 2], [3, 4], [5, 6]])
        B = np.array([3, 4])
        print(A + B)
```

```
[[ 4  6]
 [ 6  8]
 [ 8 10]]
```

(iii) A column vector and a row vector combined in an operation

```
In [ ]: # Outer product of 4x1 column vector A, with row vector B of shape (3,)
        # yields a 4x3 array(matrix).
        A_trans = np.array([1, 2, 3, 4], ndmin=2)
        A = A_trans.transpose()
        B = np.array([1, 2, 3])
        print(A * B)
```

```
[[ 1  2  3]
 [ 2  4  6]
 [ 3  6  9]
 [ 4  8 12]]
```

Q8

```
In [ ]: m, c = 2, -4
        N = 10
        x = np.linspace(0, N-1, N).reshape (N, 1)
        sigma = 10
        y = m*x + c + np.random.normal(0, sigma, (N, 1))
```

```
In [ ]: X = np.append(np.ones((N,1)), x, axis = 1)
```

```
In [ ]: from numpy import linalg
        print(linalg.inv(X.T @ X) @ X.T @ y)
```

```
[[ -9.28389721]
 [  3.41709837]]
```

Q10

```
In [ ]: import cv2 as cv

        im = cv.imread(r'Images/gal_gaussian.png')
        assert im is not None

        im_gausblur = cv.GaussianBlur(im, ksize=(9,5), sigmaX = 0)

        cv.imshow('Image', im)
        cv.waitKey(0)
```

```
cv.imshow('GausBlur_Image', im_gausblur)
cv.waitKey(0)

cv.destroyAllWindows()
```

Q11

```
In [ ]: im = cv.imread(r'Images/gal_sandp.png')
        assert im is not None

        im_median = cv.medianBlur(im, 5)

        cv.imshow('Image', im)
        cv.waitKey(0)

        cv.imshow('Medianfilter_Image', im_median)
        cv.waitKey(0)

        cv.destroyAllWindows()
```

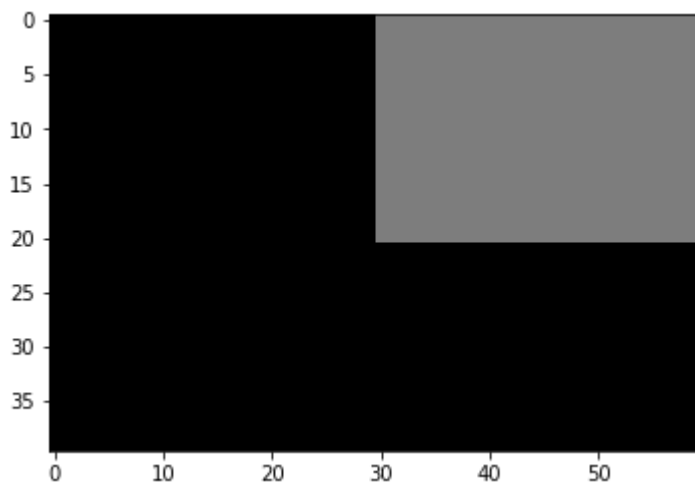
Q12

```
In [ ]: import matplotlib.pyplot as plt
        %matplotlib inline

        im = np.zeros((40,60), dtype=np.uint8)
        im[0:im.shape[0]//2+1, im.shape[1]//2: ] = 125

        fig, ax = plt.subplots()
        ax.imshow(im, cmap='gray', vmin = 0, vmax= 255)
        plt.show()

        # cv.imshow('colourPatch', im)
        # cv.waitKey(0)
        # cv.destroyAllWindows()
```



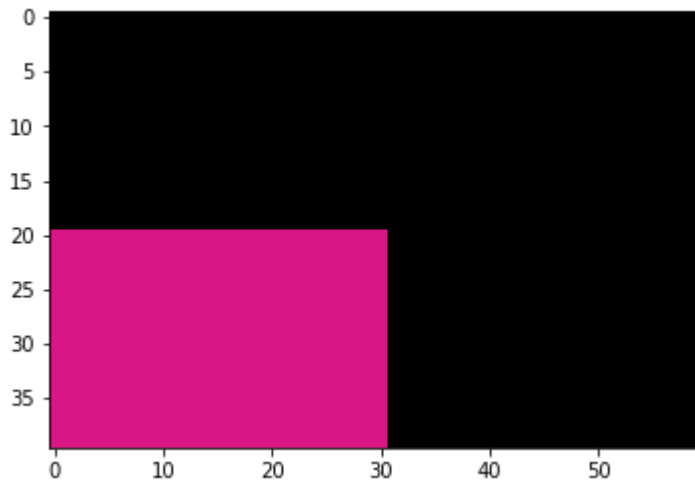
Q13

```
In [ ]: import matplotlib.pyplot as plt
        %matplotlib inline

        im = np.zeros((40,60,3), dtype=np.uint8)
        im[im.shape[0]//2:, 0:im.shape[1]//2 + 1] = int(255*0.85), int(255*0.09), int(255*0.52)

        fig, ax = plt.subplots()
        ax.imshow(im, vmin = 0, vmax= 255)
        plt.show()

        # cv.imshow('BarbiePink', im)
        # cv.waitKey(0)
        # cv.destroyAllWindows()
```



Q14

```
In [ ]: def inc_brightness(im, value=25):
        hsv = cv.cvtColor(im, cv.COLOR_BGR2HSV)
        h, s, v = cv.split(hsv)

        v[v > 255 - value] = 255
        v[v <= 255 - value] += value

        hsv_adjusted = cv.merge((h, s, v))
        im = cv.cvtColor(hsv_adjusted, cv.COLOR_HSV2BGR)
        return im
```

```
In [ ]: im = cv.imread(r'Images/tom_dark.jpg')

        im = inc_brightness(im, 40)
        cv.imshow('Mr. Tom Cruise', im)

        cv.waitKey(0)
        cv.destroyAllWindows()
```