

Name : Rukmal M.A.D Index Number : 190531L

```
In [2]: import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
import sympy
from numpy import linalg
```

```
In [10]: for i in range (1,6):
print(i,":",i**2)
```

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

```
In [14]: for i in range (1,6):
if not sympy.isprime(i):
print(i,":",i**2)
else:
print(i)
```

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

```
In [17]: squares = [i**2 for i in range(1,6)]
for i,val in enumerate(squares):
print (i+1,":",val)
```

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

```
In [18]: squares = [i**2 for i in range(1,6)]
for i,val in enumerate(squares):
if sympy.isprime(i+1):
print(i+1)
else:
print (i+1,":",val)
```

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

```
In [19]: A =np.array([[1,2],[3,4],[5,6]])
B = np.array([[7,8,9,1],[1,2,3,4]])
C =np.matmul(A,B)
print (C)
```

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

```
In [20]: A = np.array([[1,2],[3,4],[5,6]])
        B = np.array([[3,2],[5,4],[3,1]])
        print(np.multiply(A,B))
```

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

```
In [39]: R = np.random.random_integers(0,10,(5,7))
        extract = R[1:4,0:2]
        print(R)
        print(extract)
        print(extract.shape)
```

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

C:\Users\diluk\AppData\Local\Temp\ipykernel\_10904\113242100.py:1: DeprecationWarning:  
This function is deprecated. Please call randint(0, 10 + 1) instead  
R = np.random.random\_integers(0,10,(5,7))

```
In [42]: P = np.array([[1,2,3],[4,5,6],[7,8,9]])
        Q = np.array([2,3,4])
```

```
R = P + Q
S = P*Q
T = P-Q
print(R)
print(S)
print(T)
```

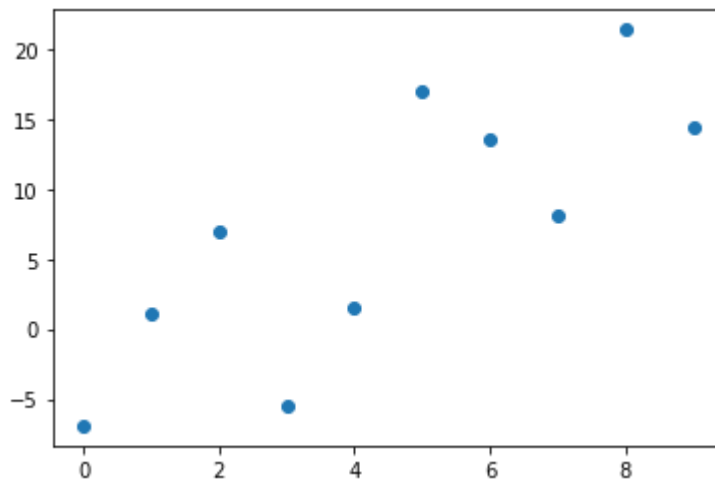
```
[[ 3  5  7]
 [ 6  8 10]
 [ 9 11 13]]
[[ 2  6 12]
 [ 8 15 24]
 [14 24 36]]
[[-1 -1 -1]
 [ 2  2  2]
 [ 5  5  5]]
```

```
In [52]: 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 ) )
        plt.scatter(x,y)

        X = np.append(np.ones((N,1)),x,axis=1)
```

```
w = linalg.inv(X.T@X)@(X.T)@y
w
```

```
Out[52]: array([[ -4.04818217],
               [ 2.49097026]])
```



```
In [13]: img = cv.imread(r'D:\Image processing module\Lecture 1\Images\gal_gaussian.png')
assert img is not None

blur = cv.GaussianBlur(img,(5,5),0)
cv.namedWindow('sample image',cv.WINDOW_AUTOSIZE)
cv.imshow('sample image',blur)
cv.waitKey(0)
cv.destroyAllWindows()
```

```
In [14]: img = cv.imread(r'D:\Image processing module\Lecture 1\Images\gal_sandp.png')
assert img is not None

median = cv.medianBlur(img,5)
cv.namedWindow('sample image',cv.WINDOW_AUTOSIZE)
cv.imshow('sample image',median)
cv.waitKey(0)
cv.destroyAllWindows()
```

```
In [24]: im = np.zeros((40,60), dtype=np.uint8)
im[0:21,30:61] = 125

cv.namedWindow('created',cv.WINDOW_AUTOSIZE)
cv.imshow('created',im)
cv.waitKey(0)
cv.destroyAllWindows()
```

```
In [3]: im = np.zeros((40,60,3), dtype=np.uint8)
im[20:41,0:31] = [132,24,218]

cv.namedWindow('created',cv.WINDOW_AUTOSIZE)
cv.imshow('created',im)
cv.waitKey(0)
cv.destroyAllWindows()
```

```
In [12]: img = cv.imread(r'D:\Image processing module\Lecture 1\Images\tom_dark.jpg')
assert img is not None
brightned = img*np.array((2))
```

```
cv.namedWindow('sample image',cv.WINDOW_AUTOSIZE)  
cv.imshow('sample image',brightned)  
cv.waitKey(0)  
cv.destroyAllWindows()
```

In [ ]: