INDEX : 190328V Name : KUMARA B.W.J.C.

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

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

02)

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

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

03)

In [3]:
```
Numbers =[i**2 for i in range(1,6)]
for i ,i2 in enumerate(Numbers):
  print(i+1,":",Numbers[i])
```

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

04)

In [4]:
```
Numbers =[i**2 for i in range(1,6)]
for i ,i2 in enumerate(Numbers):
  if not sympy.isprime(i+1):
    print(i+1,":",Numbers[i])
```

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

05) a.

In [5]:
```
import numpy as np

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

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

05) b.

In [6]:
```
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]]
```

06)

In [7]:
```
Arr = np.random.randint(10, size=(5, 7))
sub_Arr = Arr[:2,:4]
print("Array\n",Arr)
print("Sub Array\n",sub_Arr)
```

```
Array
 [[9 4 1 9 9 5 8]
 [8 7 3 3 1 6 1]
 [8 7 9 9 7 8 8]
 [1 8 9 6 9 4 0]
 [6 1 9 6 4 7 1]]
Sub Array
 [[9 4 1 9]
 [8 7 3 3]]
```

07)

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

```
s = a + b
print(s)
```

```
[[ 3  8]
 [ 6 10]]
```

08) a,b

In [9]:

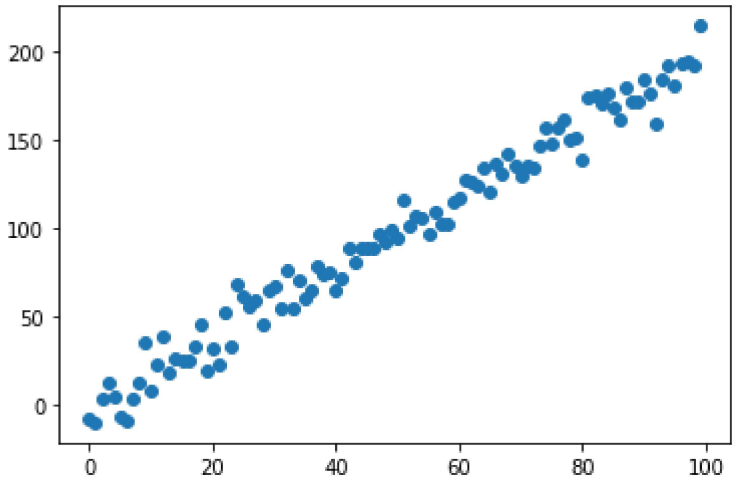
```
import matplotlib.pyplot as plt

from numpy import linalg

m, c = 2 , -4
N = 100
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)
plt.show()

X = np.append(np.ones((N,1)),x,axis =1)
w = linalg.inv(X.T @ X)@X.T @ y
w
```



Out[9]:

```
array([[-3.02313267],
       [ 2.02757364]])
```

09) a. b. c.

In [19]:
```
def HyEst(S):
    a = S
    n =0
    while not (1<=a<100):
        a = int(S//100)
        n+=1
    return ((-190/(a+20))+10)*(10**n)
print(HyEst(1600))
```

```
47.22222222222222
```

In [25]:
```
def Sqrt(N):
    sq_p = HyEst(N)
    sq_n =0

    while True:
        sq_n = 0.5*(sq_p+N/sq_p)
        if (-10e-5)<(sq_n-sq_p)<(10e-5):
            break
        sq_p = sq_n
    return round(sq_n,6)

print("Sqrt of 64:",Sqrt(64))
print("Sqrt of 75:",Sqrt(75))
print("Sqrt of 100:",Sqrt(100))
print("Sqrt of 1600:",Sqrt(1600))
```

```
Sqrt of 64: 8.0
Sqrt of 75: 8.660254
Sqrt of 100: 10.0
Sqrt of 1600: 40.0
```

10)

In [10]:
```
import cv2 as cv
import matplotlib.pyplot as plt

im = cv.imread(r'./Images/gal_gaussian.png')

blur = cv.GaussianBlur(im,(5,5),0)

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

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

11)

In [11]:
```
im2 = cv.imread(r'./Images/gal_sandp.png')

filteredImg = cv.medianBlur(im2, ksize=5)

cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',im2)
cv.waitKey(0)
cv.imshow('Image',filteredImg)
cv.waitKey(0)
cv.destroyAllWindows()
```

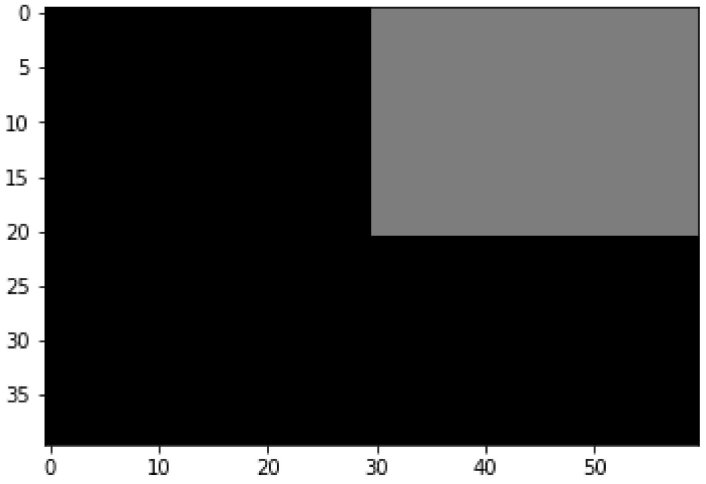
12)

In [12]:
```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

im = np.zeros((40,60),dtype=np.uint8)

im[0:21,30:61] = 125

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

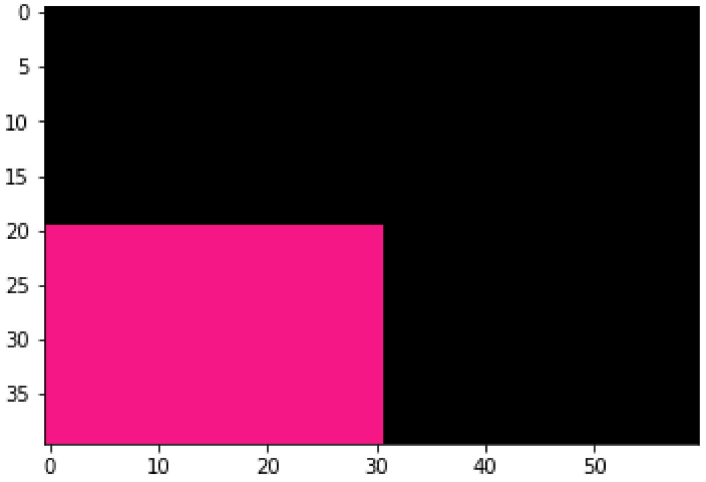


13)

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

im = np.zeros((40,60,3),dtype=np.uint8)
im[20:41,0:31] = [244,24,132]

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



14)

In [15]:
```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

im3 = mpimg.imread(r'./Images/tom_dark.jpg')
imgplot= plt.imshow(im3,cmap = "gray")
plt.show()

im4 = im3+100
imgplot= plt.imshow(im4,cmap = "gray")
plt.show()
```

In [ ]: