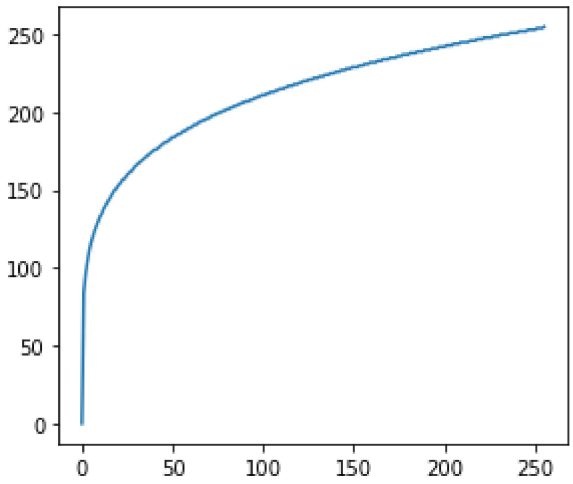
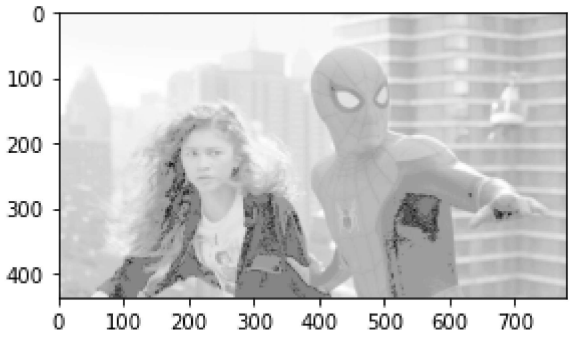
Index : 190144D

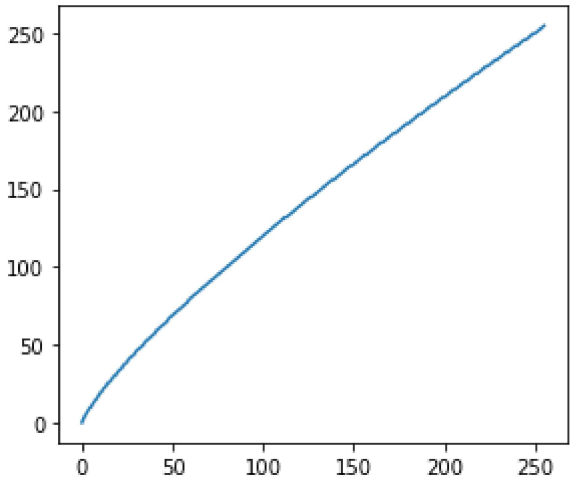
Name: Dilshan J.V.A.P

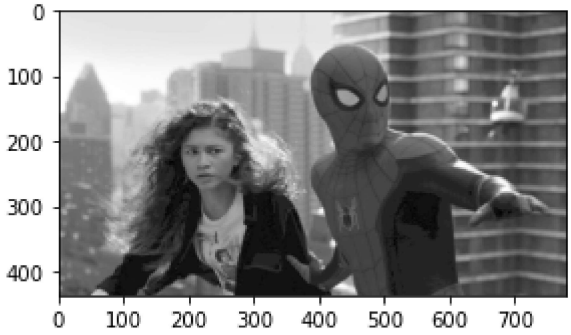
In [ ]:
```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
```

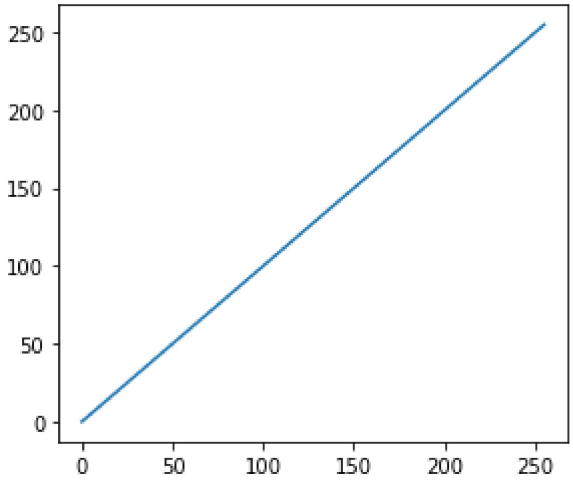
In [ ]:
```
#(1)
im=cv.imread(r'spider.png',cv.IMREAD_GRAYSCALE)
assert im is not None

cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',im)
cv.waitKey(0)
gamma = [0.2,0.8,1,1.2,2]
for i in gamma:
    t = np.array([(p/255)**i*255 for p in range (0,256)]).astype(np.uint8)
    g = cv.LUT(im,t)
    fig,ax = plt.subplots(1,2,figsize=(10,4))
    ax[0].plot(t)
    ax[1].imshow(g,'gray')

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

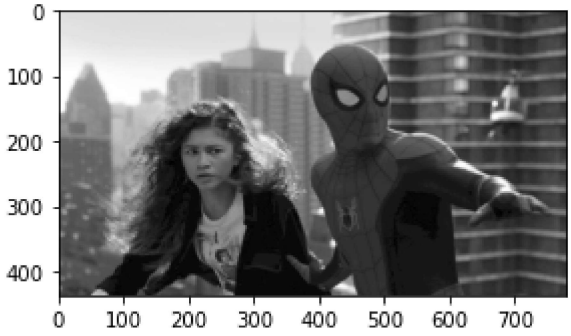


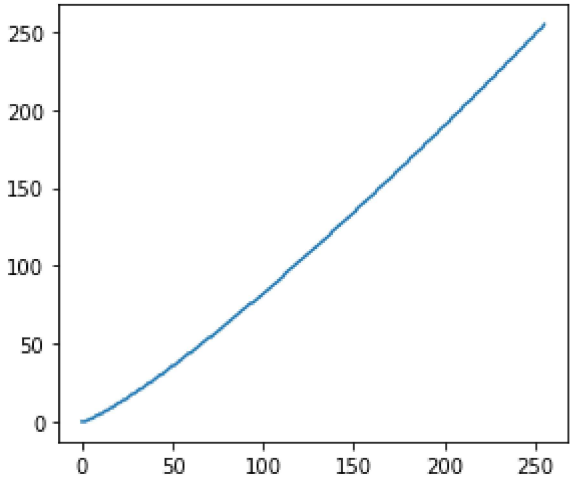


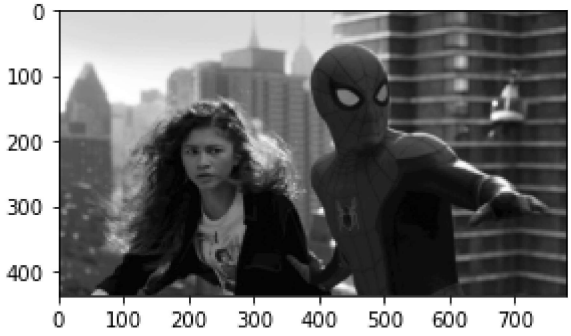

0
50
100
150
200
250

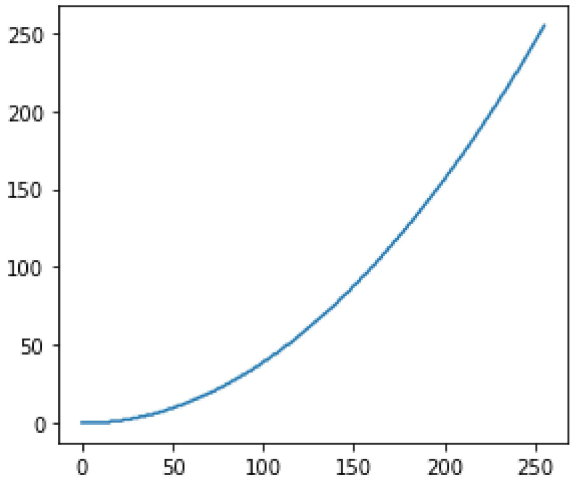

0
100
200
300
400
0
100
200
300
400
500
600
700

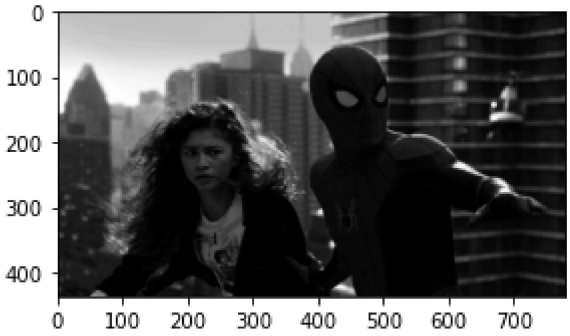

0
50
100
150
200
250


0
100
200
300
400
0
100
200
300
400
500
600
700


0
50
100
150
200
250


0
100
200
300
400
0
100
200
300
400
500
600
700



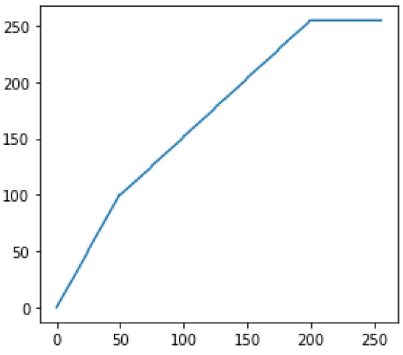


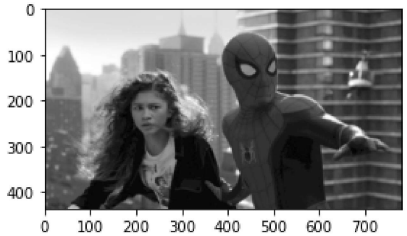
In [ ]:

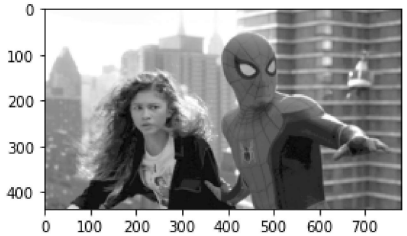
```
#(2)
im=cv.imread(r'spider.png',cv.IMREAD_GRAYSCALE)
t1= np.linspace (0,100,50)
t2= np.linspace (100,255,150)
t3= np.linspace (255,255, 56)
t= np.concatenate((t1,t2,t3), axis =0).astype(np.uint8)

assert len(t) ==256
g= cv.LUT(im,t)
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',im)
cv.waitKey(0)
cv.imshow('Image',g)
cv.waitKey(0)
cv.destroyAllWindows()

fig,ax = plt.subplots(1,3,figsize=(15,4))
ax[0].plot(t)
ax[1].imshow(im,'gray')
ax[2].imshow(g,'gray')
```

Out[ ]:

```
<matplotlib.image.AxesImage at 0x2ac008a0fd0>
```





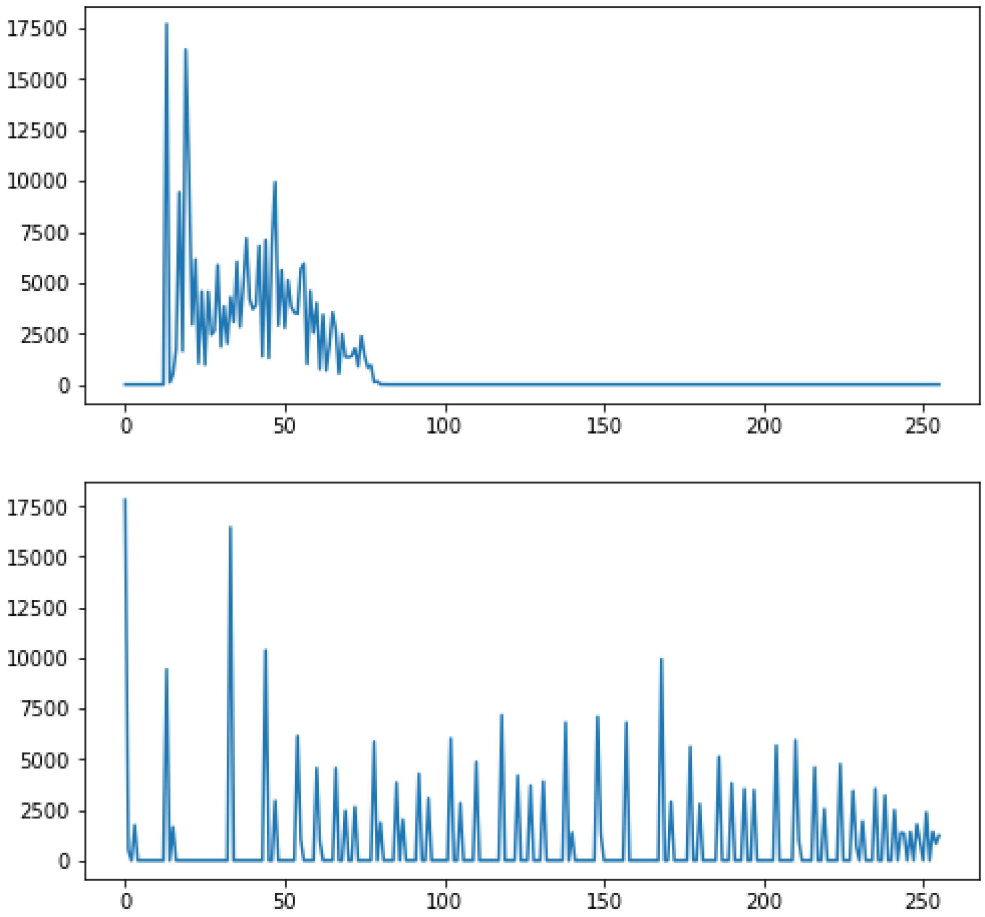


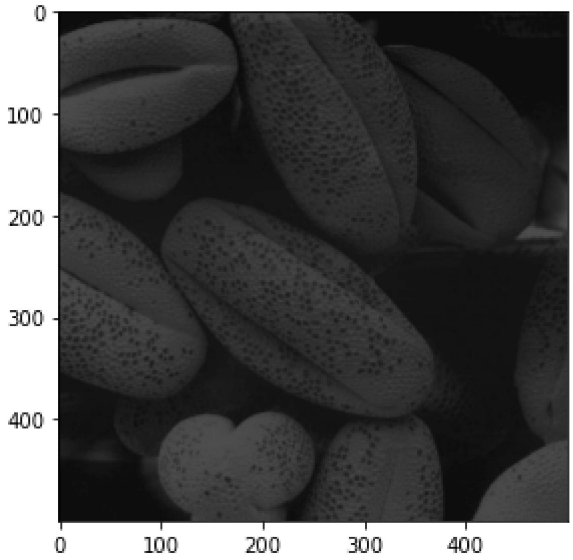
In [ ]:

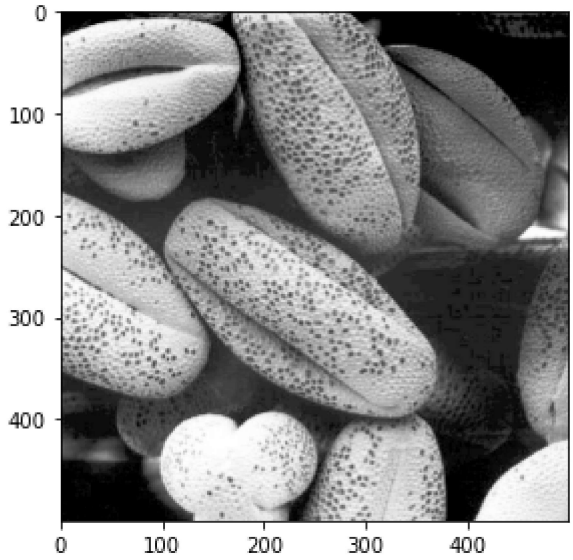
```
#(3)
im=cv.imread('shells.tif',cv.IMREAD_GRAYSCALE)
assert im is not None
hist_im = cv.calcHist([im], [0], None, [256], [0,256])
g = cv.equalizeHist(im)
hist_g = cv.calcHist([g], [0], None, [256], [0,256])
fig, ax = plt.subplots(2,1,figsize= (8,8))
```

```
ax[0].plot(hist_im)
ax[1].plot(hist_g)

cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',im)
cv.waitKey(0)
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',g)
cv.waitKey(0)
cv.destroyAllWindows()
rgb_im = cv.cvtColor(im, cv.COLOR_BGR2RGB)
rgb_g = cv.cvtColor(g, cv.COLOR_BGR2RGB)
fig,ax = plt.subplots(1,2,figsize=(10,8))
ax[0].imshow(rgb_im)
ax[1].imshow(rgb_g)
```

Out[ ]: <matplotlib.image.AxesImage at 0x2ac7fdd6fd0>







In [ ]:

```
#(4)(a)
img =  cv.imread(r'zion_pass.jpg')
assert img is not None;
hsvImg = cv.cvtColor(img, cv.COLOR_BGR2HSV)


print("original")
fig,ax = plt.subplots()
ax.imshow(cv.cvtColor(img, cv.COLOR_BGR2RGB))
plt.show()

#increasing the saturation
satIncImg = np.array(hsvImg)
for x in range(hsvImg.shape[0]):
    for y in range(hsvImg.shape[1]):
        satIncImg[x,y][1] = min(255,satIncImg[x,y][1]+40)


imgRGB = cv.cvtColor(satIncImg, cv.COLOR_HSV2RGB)
fig, ax = plt.subplots()
ax.imshow(imgRGB)
```

```
original
```

Out[ ]:

```
<matplotlib.image.AxesImage at 0x2ac7f7c6b50>
```

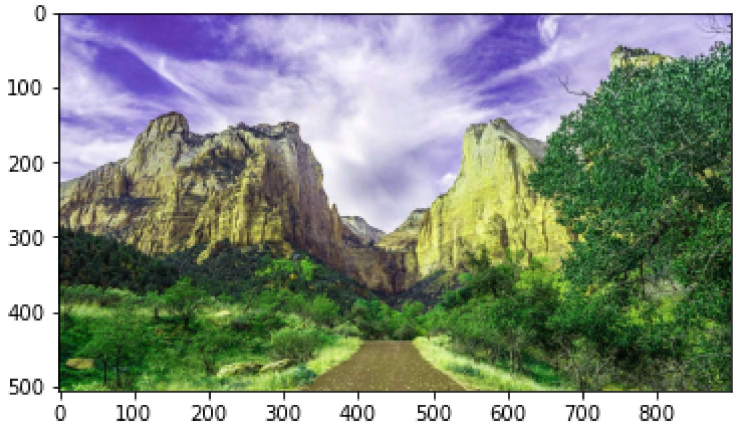
In [ ]:

```
#(4)(b)
img =  cv.imread(r'zion_pass.jpg')
assert img is not None;
hsvImg = cv.cvtColor(img, cv.COLOR_BGR2HSV)



print("original")
fig,ax = plt.subplots()
ax.imshow(cv.cvtColor(img, cv.COLOR_BGR2RGB))
plt.show()

#increasing the hue
hueChangedImg = np.array(hsvImg)
for x in range(hsvImg.shape[0]):
    for y in range(hsvImg.shape[1]):
        hueChangedImg[x,y][0] = min(255,hueChangedImg[x,y][0]+20)



imgRGB = cv.cvtColor(hueChangedImg, cv.COLOR_HSV2RGB)
fig, ax = plt.subplots()
ax.imshow(imgRGB)
```

```
original
```

Out[ ]:

```
<matplotlib.image.AxesImage at 0x2ac7fd31580>
```


0
100
200
300
400
500
0
100
200
300
400
500
600
700
800