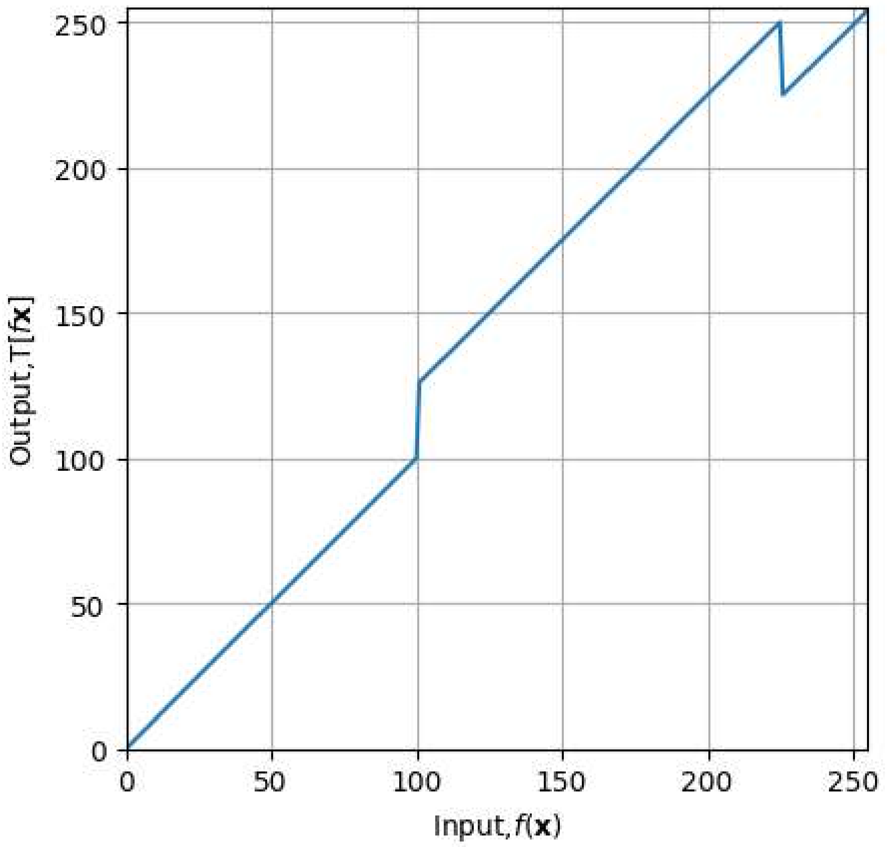
D/ENG/21/0060/EE - MASM PERERA

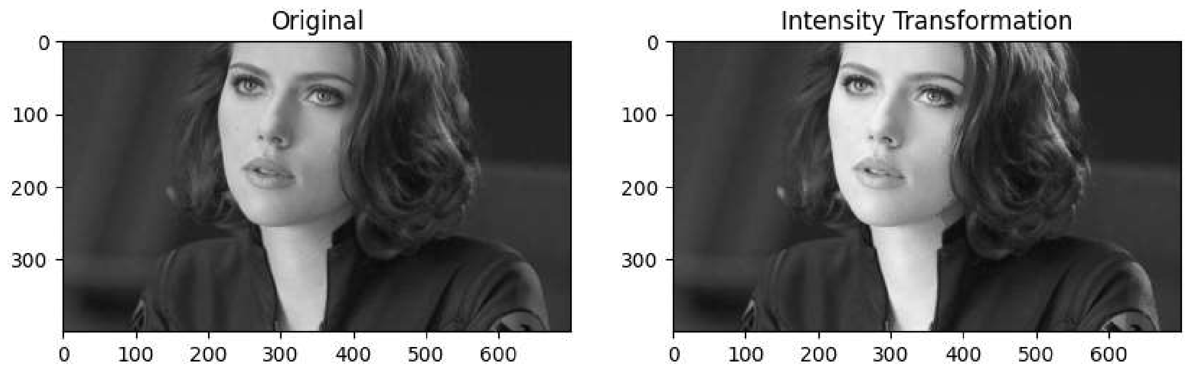
GITHUB LINK - https://github.com/mitharaperera/imageProcessingAssignment

Question 1

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

c=np.array([(100,100),(100,125),(250,255)])
t1=np.linspace(0,c[0,1],c[0,0]+1-0).astype('uint8')
print(len(t1))
#t2=np.linspace(c[0,1]+1,c[1,1],c[1,0]-c[0,0]).astype('uint8')
#print(len(t2))
t3=np.linspace(c[1,1]+1,250,225-c[1,0]).astype('uint8')
print(len(t3))
#t4=np.linspace(c[2,0]+1,225,225-c[1,0]).astype('uint8')
#print(len(t4))
t5=np.linspace(225,255,255-225).astype('uint8')
print(len(t5))
transform=np.concatenate((t1,t3),axis=0).astype('uint8')
transform=np.concatenate((transform,t5),axis=0).astype('uint8')

print(len(transform))
fig,ax=plt.subplots()
ax.plot(transform)
ax.set_xlabel(r'Input,$f(\mathbf{x})$')
ax.set_ylabel('Output,$\mathrm{T}[f{\mathbf{x}}]$')
ax.set_xlim(0,255)
ax.set_ylim(0,255)
ax.grid(True)
ax.set_aspect('equal')
plt.savefig('transform.png')
plt.show()
img_orig=cv.imread('Images/natasha_grayscale.jpg', cv.IMREAD_GRAYSCALE)
cv.namedWindow("Image",cv.WINDOW_AUTOSIZE)
cv.imshow("Image",img_orig)
cv.waitKey(0)
image_transformed=cv.LUT(img_orig,transform)
cv.imshow("Image",image_transformed)
cv.waitKey(0)
cv.destroyAllWindows()

fig, ax= plt.subplots(1,2, figsize=(10,20))
ax[0].imshow(img_orig, cmap="gray")
ax[0].set_title('Original')
ax[1].imshow(image_transformed, cmap="gray")
ax[1].set_title('Intensity Transformation')
plt.show()
```

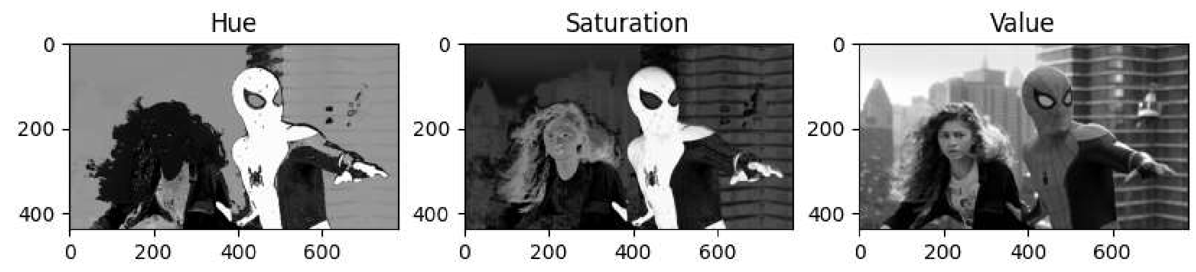
```
101
125
30
256
```

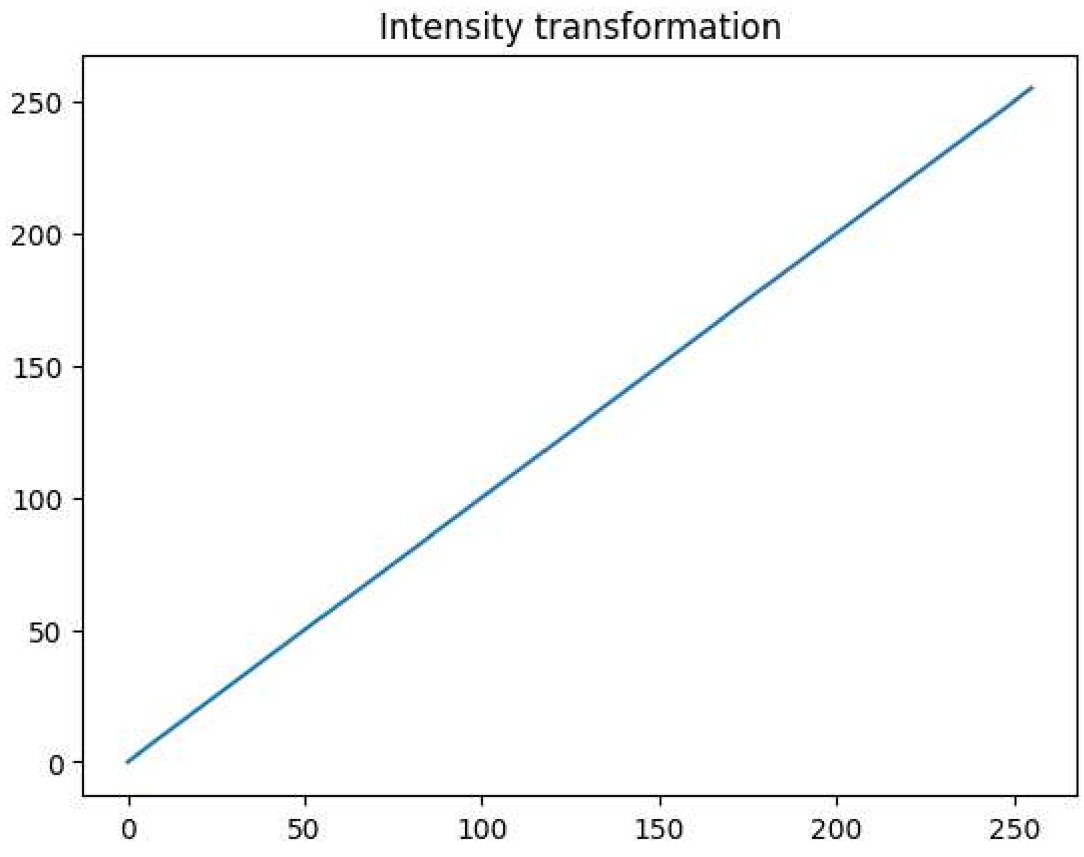


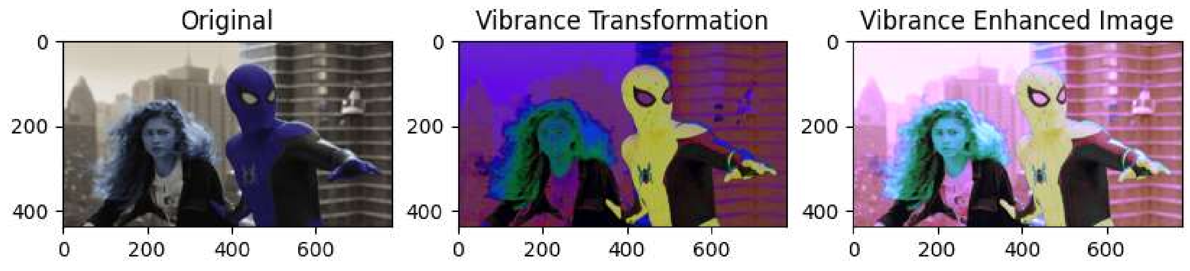


In [ ]:
```python
import cv2 as cv
import matplotlib.pyplot as plt
import numpy as np

#not changed
img = cv.imread('Images\spider.png', cv.IMREAD_COLOR)
assert img is not None

m = cv.cvtColor(img, cv.COLOR_BGR2HSV)
h_img,s_img,v_img = cv.split(m)

fig, ax= plt.subplots(1,3, figsize=(10,20))
ax[0].imshow(h_img, cmap="gray")
ax[0].set_title('Hue')
```

```
ax[1].imshow(s_img, cmap="gray")
ax[1].set_title('Saturation')
ax[2].imshow(v_img, cmap="gray")
ax[2].set_title('Value')
plt.show()

x= np.arange(0, 256).astype('uint8')
a = .1
sigma = 70
Y = np.minimum(((x)+(a*(np.exp(-(x-128)**2/(2*sigma**2))))/128), 255).astype('uint8
image_transform = cv.LUT(s_img, Y)
plt.title('Intensity transformation')
plt.plot(Y)
plt.show()

newHSVtrans = cv.merge([h_img,image_transform,v_img])
result =  cv.cvtColor(newHSVtrans,  cv.COLOR_HSV2BGR)
added_img = cv.add(newHSVtrans, img)
#Y2 = (a/128)*Y1
#Y = np.add(Y, Y1)
fig, ax= plt.subplots(1,3, figsize=(10,20))
ax[0].imshow(img, cmap="gray")
ax[0].set_title('Original')
ax[1].imshow(newHSVtrans, cmap="gray")
ax[1].set_title('Vibrance Transformation')
ax[2].imshow(added_img, cmap="gray")
ax[2].set_title('Vibrance Enhanced Image')
plt.show()
```



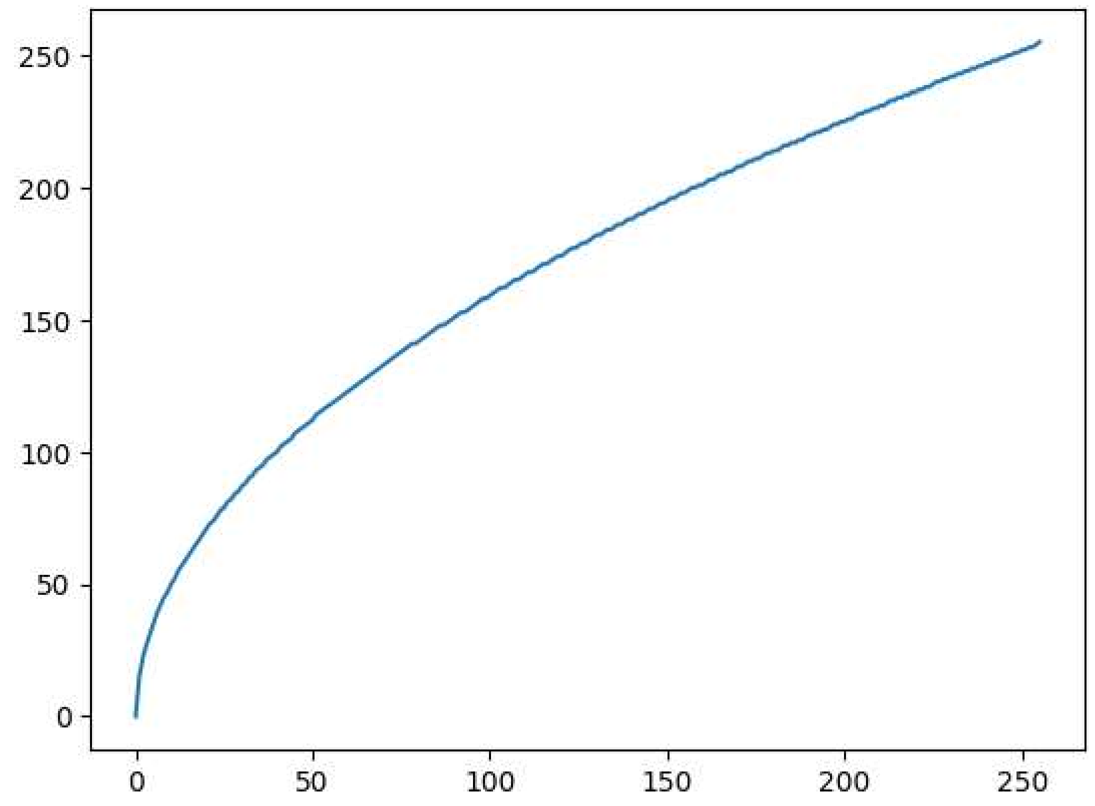


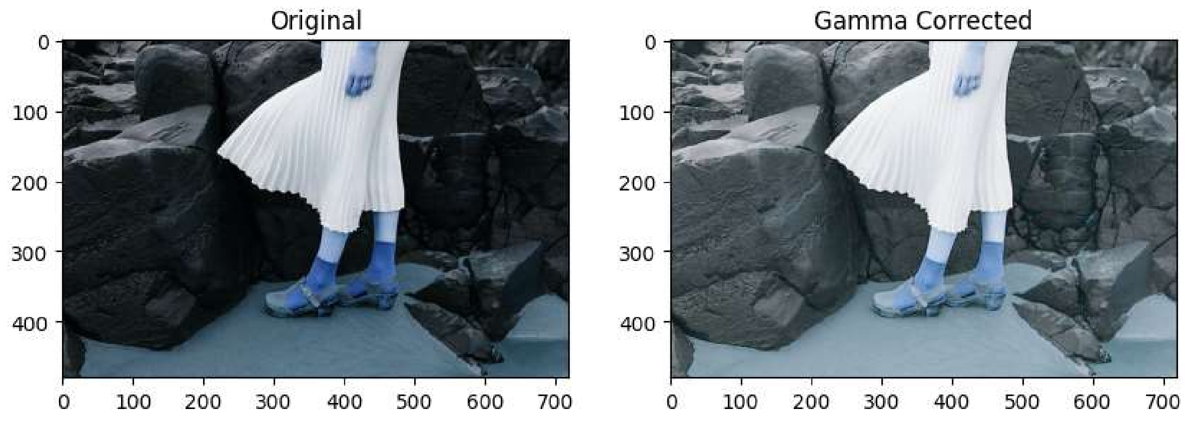


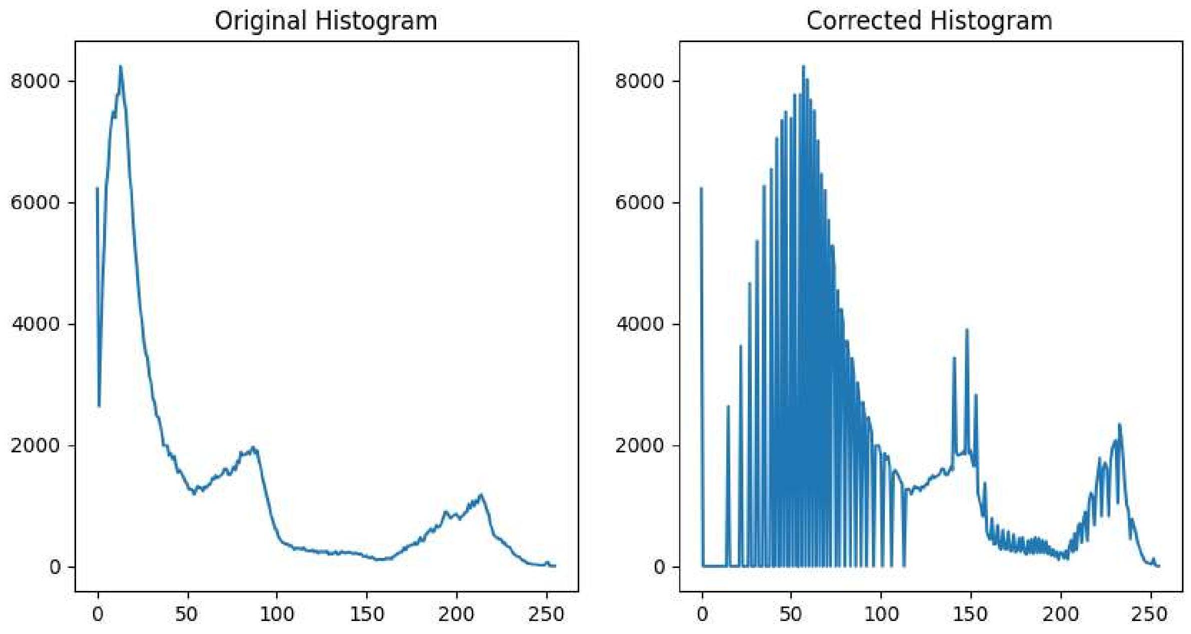
Question 3

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

        img = cv.imread ('Images\highlights_and_shadows.jpg', cv.IMREAD_COLOR)
        assert img is not None

        img_LAB = cv.cvtColor(img, cv.COLOR_BGR2LAB)

        gamma = .5
        t = np.array([(i/255.)**gamma*255 for i in range (256)], np.uint8)
        g = t[img]

        plt.plot(t)
        plt.show()

        fig, ax = plt.subplots(1,2, figsize=(10,20))
```

```
ax[0].imshow(img, cmap="gray")
ax[0].set_title("Original")
ax[1].imshow(g,cmap="gray")
ax[1].set_title("Gamma Corrected")
plt.show()

plt.figure(figsize = [10, 5])
plt.subplot(1, 2, 1)
plt.title('Original Histogram')
im_h = cv.calcHist([img],[0],None,[256],[0,256])
plt.plot(im_h)
plt.subplot(1, 2, 2)
plt.title('Corrected Histogram')
g_h = cv.calcHist([g],[0],None,[256],[0,256])
plt.plot(g_h)
plt.show()
```

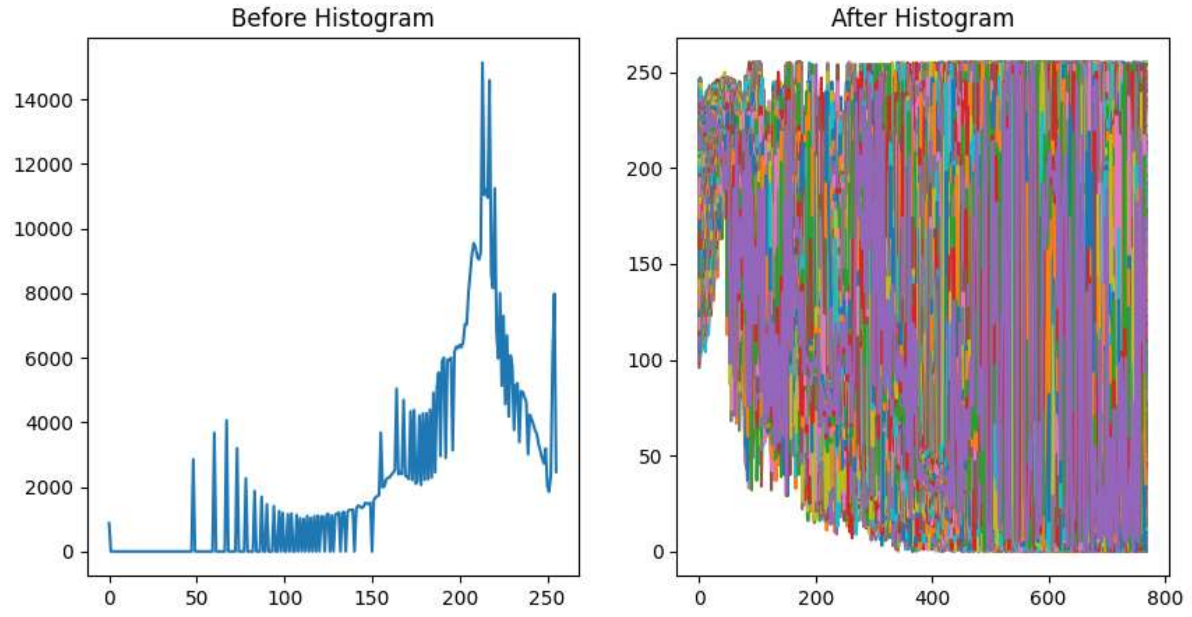


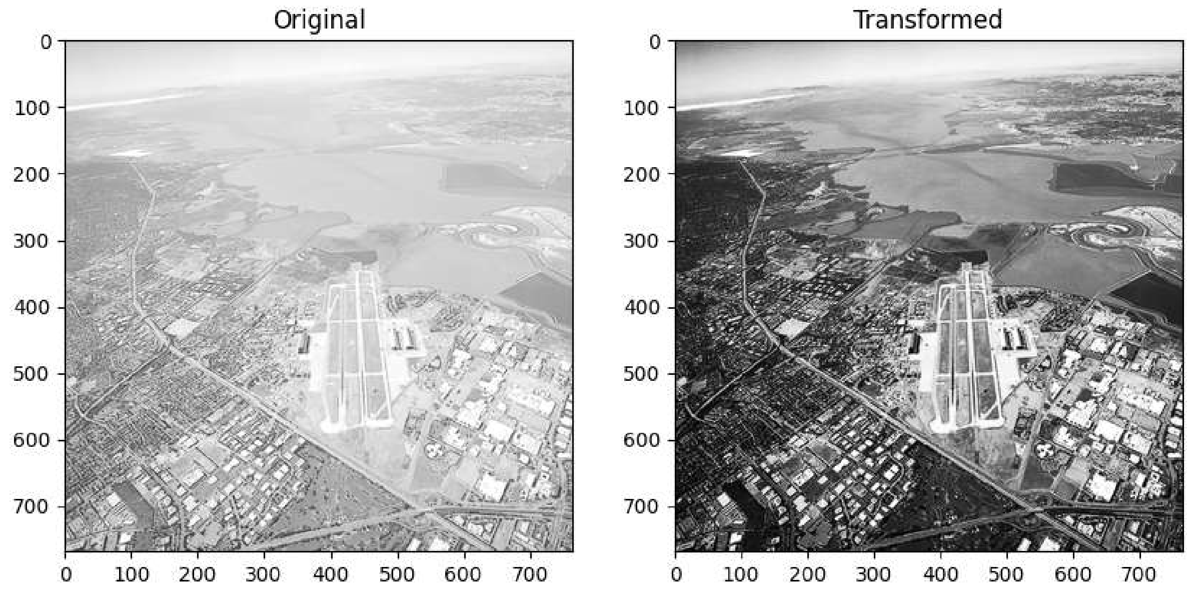




Question 4

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

img = cv.imread('Images\washed_out_aerial_image.png', cv.IMREAD_GRAYSCALE)
assert img is not None
equalized_img = cv.equalizeHist(img)

plt.figure(figsize = [10, 5])
plt.subplot(1, 2, 1)
plt.title('Before Histogram')
img_b = cv.calcHist([img],[0],None,[256],[0,256])
plt.plot(img_b)

plt.subplot(1, 2, 2)
plt.title('After Histogram')
img_a = cv.calcHist([equalized_img],[0],None,[256],[0,256])
plt.plot(equalized_img)
plt.show()

fig, ax= plt.subplots(1,2, figsize=(10,20))
ax[0].imshow(img, cmap="gray")
ax[0].set_title('Original')
ax[1].imshow(equalized_img, cmap="gray")
ax[1].set_title('Transformed')
plt.show()

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

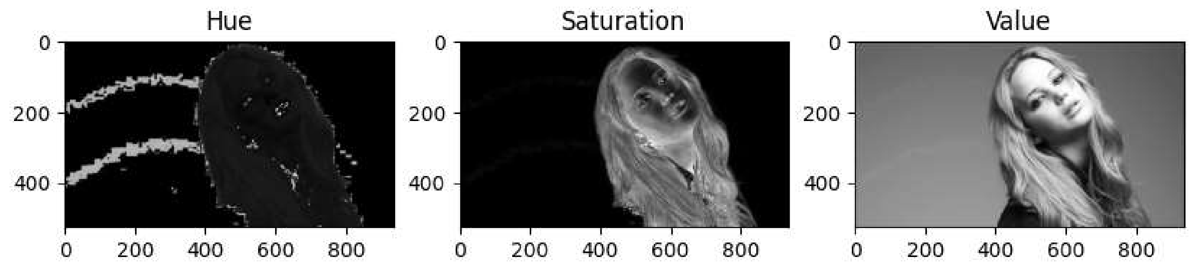




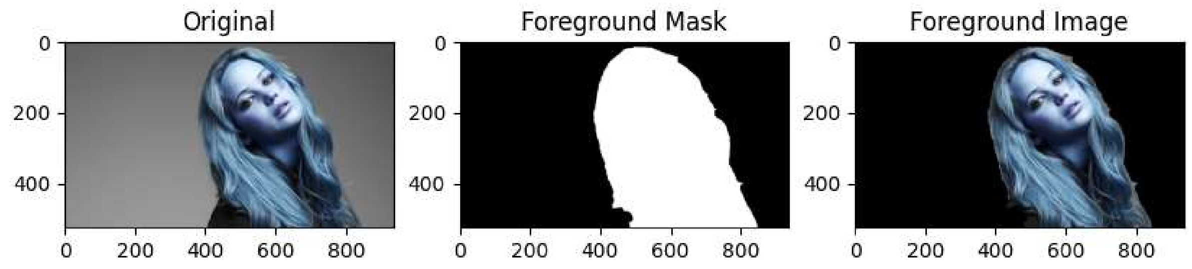
Question 5

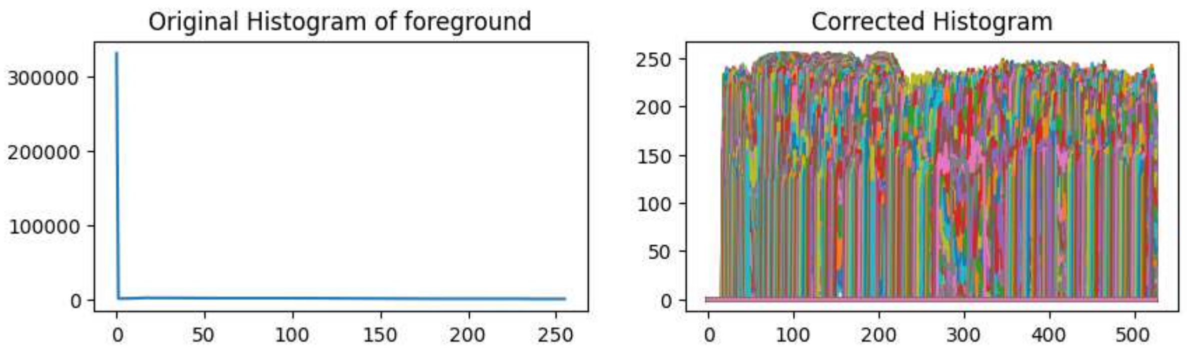
```
In [ ]: import cv2 as cv
        import matplotlib.pyplot as plt
        import numpy as np

        #not changed
        img = cv.imread('Images\jeniffer.jpg', cv.IMREAD_COLOR)
        assert img is not None

        m = cv.cvtColor(img, cv.COLOR_BGR2HSV)
        h_img,s_img,v_img = cv.split(m)

        fig, ax= plt.subplots(1,3, figsize=(10,20))
        ax[0].imshow(h_img, cmap="gray")
        ax[0].set_title('Hue')
        ax[1].imshow(s_img, cmap="gray")
        ax[1].set_title('Saturation')
```

```
ax[2].imshow(v_img, cmap="gray")
ax[2].set_title('Value')
plt.show()

lower = np.array([200, 200, 200])
upper = np.array([255, 255, 255])
thresh = cv.inRange(s_img, 15, 230)
kernel = cv.getStructuringElement(cv.MORPH_ELLIPSE, (20,20))
morph = cv.morphologyEx(thresh, cv.MORPH_CLOSE, kernel)
mask = morph
result = cv.bitwise_and(img, img, mask=mask)

fig, ax = plt.subplots(1,3, figsize=(10,2.5))
fig.suptitle("b. Extracting Foreground mask")
ax[0].imshow(img, cmap="gray")
ax[0].set_title("Original")
ax[1].imshow(mask, cmap="gray")
ax[1].set_title("Foreground Mask")
ax[2].imshow(result, cmap="gray")
ax[2].set_title("Foreground Image")
plt.show()

#histogram
cumulative_sum = np.cumsum(result) #cumulative sum

plt.figure(figsize = [10, 2.5])
plt.subplot(1, 2, 1)
plt.title('Original Histogram of foreground')
fg_h = cv.calcHist([result],[0],None,[256],[0,256])
plt.plot(fg_h)

plt.subplot(1, 2, 2)
plt.title('Corrected Histogram')
result1 = cv.cvtColor(result, cv.COLOR_BGR2GRAY)
eh = cv.equalizeHist(result1)
plt.plot(eh)
plt.show()

#background image
mask1 = 255 - morph
bg_img = cv.bitwise_and(img, img, mask=mask1)
bg_img1 = cv.cvtColor(bg_img, cv.COLOR_BGR2GRAY);
#added image
img1 = cv.addWeighted(bg_img1,0.5, result1,0.5,0.0)

fig, ax = plt.subplots(1,3, figsize=(10,2.5))
fig.suptitle("f. Adding background with equalized")
ax[0].imshow(bg_img, cmap="gray")
ax[0].set_title("Background")
ax[1].imshow(result, cmap="gray")
ax[1].set_title("Foreground")
ax[2].imshow(img1, cmap="gray")
ax[2].set_title("Added Image")
plt.show()
```



b. Extracting Foreground mask





f. Adding background with equalized