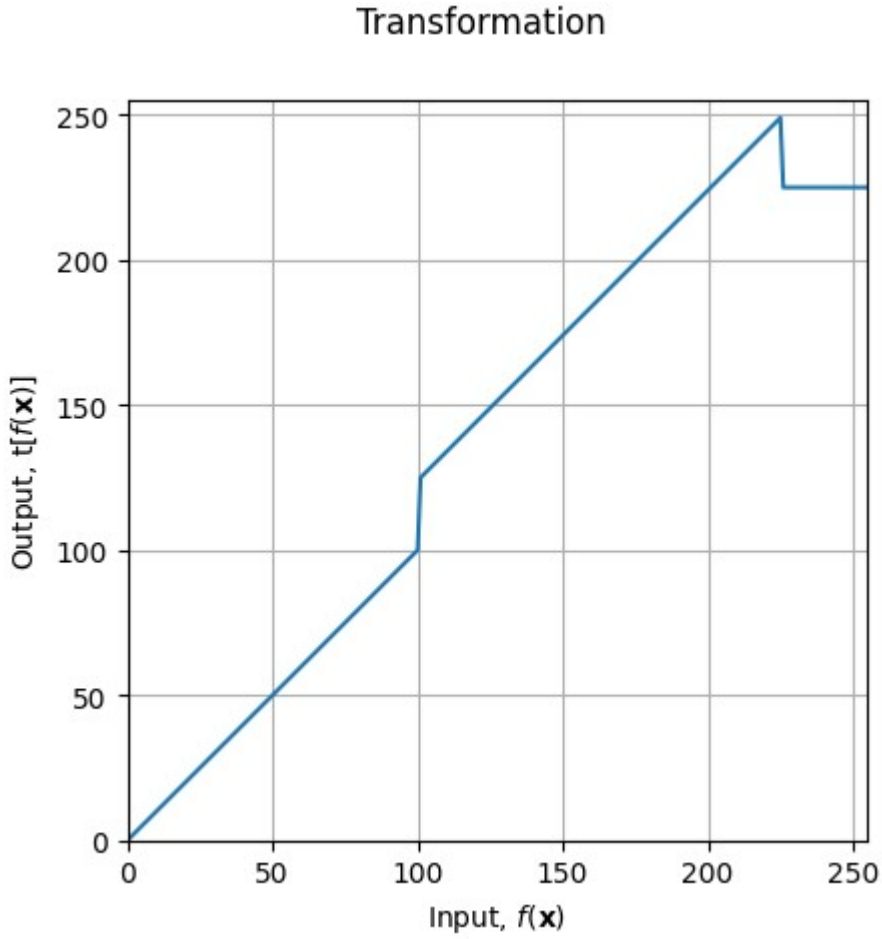
Question 1

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

#Importing the original image in grayscale
im = cv.imread('natasha_grayscale.jpg', cv.IMREAD_GRAYSCALE)
assert im is not None

t1 = np.linspace(0, 100, 101).astype('uint8')
t2 = np.linspace(125, 249, 125).astype('uint8')
t3 = np.linspace(225, 225, 30).astype('uint8')

#Transformation
transform = np.concatenate((t1, t2, t3), axis=0).astype('uint8')

fig, ax = plt.subplots()
fig.suptitle("Transformation")
ax.plot(transform)
ax.set_xlabel(r'Input, $f(\mathbf{x})$')
ax.set_ylabel(r'Output, $\mathrm{t}[f(\mathbf{x})]$')
ax.set_xlim(0, 255)
ax.set_ylim(0, 255)
ax.set_aspect('equal')
ax.grid()

image_transform = cv.LUT(im, transform)

fig, ax = plt.subplots(1, 2, figsize=(10, 20))
ax[0].imshow(im, cmap='gray')
ax[0].set_title("Original")
ax[1].imshow(image_transform, cmap='gray')
ax[1].set_title("Transformed Image")
```

Out[ ]:
```
Text(0.5, 1.0, 'Transformed Image')
```

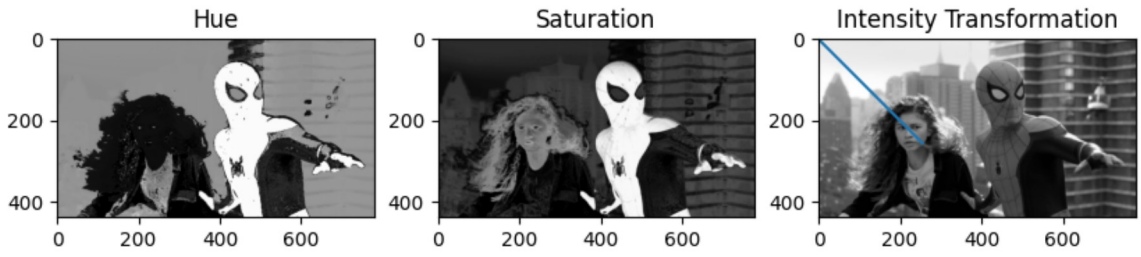

Transformation
Output, t[f(x)]
Input, f(x)
0
50
100
150
200
250


Original
0
100
200
300
0
100
200
300
400
500
600

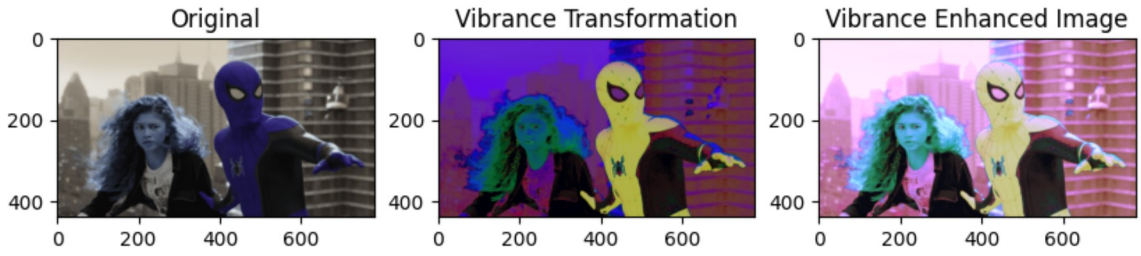

Transformed Image
0
100
200
300
0
100
200
300
400
500
600

Question 2

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

#importing the original image
im = cv.imread('spider.png', cv.IMREAD_COLOR)
assert im is not None

im_HSV = cv.cvtColor(im, cv.COLOR_BGR2HSV)
h_img, s_img, v_img = cv.split(im_HSV)

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')

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('ui
image_transform = cv.LUT(s_img, Y)
plt.title('Intensity Transformation')
plt.plot(Y)

new_HSV = cv.merge([h_img, image_transform, v_img])
output = cv.cvtColor(new_HSV, cv.COLOR_HSV2BGR)
add_img = cv.add(new_HSV, im)
fig, ax = plt.subplots(1, 3, figsize = (10, 20))
ax[0].imshow(im, cmap='gray')
ax[0].set_title('Original')
ax[1].imshow(new_HSV, cmap='gray')
ax[1].set_title('Vibrance Transformation')
ax[2].imshow(add_img, cmap='gray')
ax[2].set_title('Vibrance Enhanced Image')
```

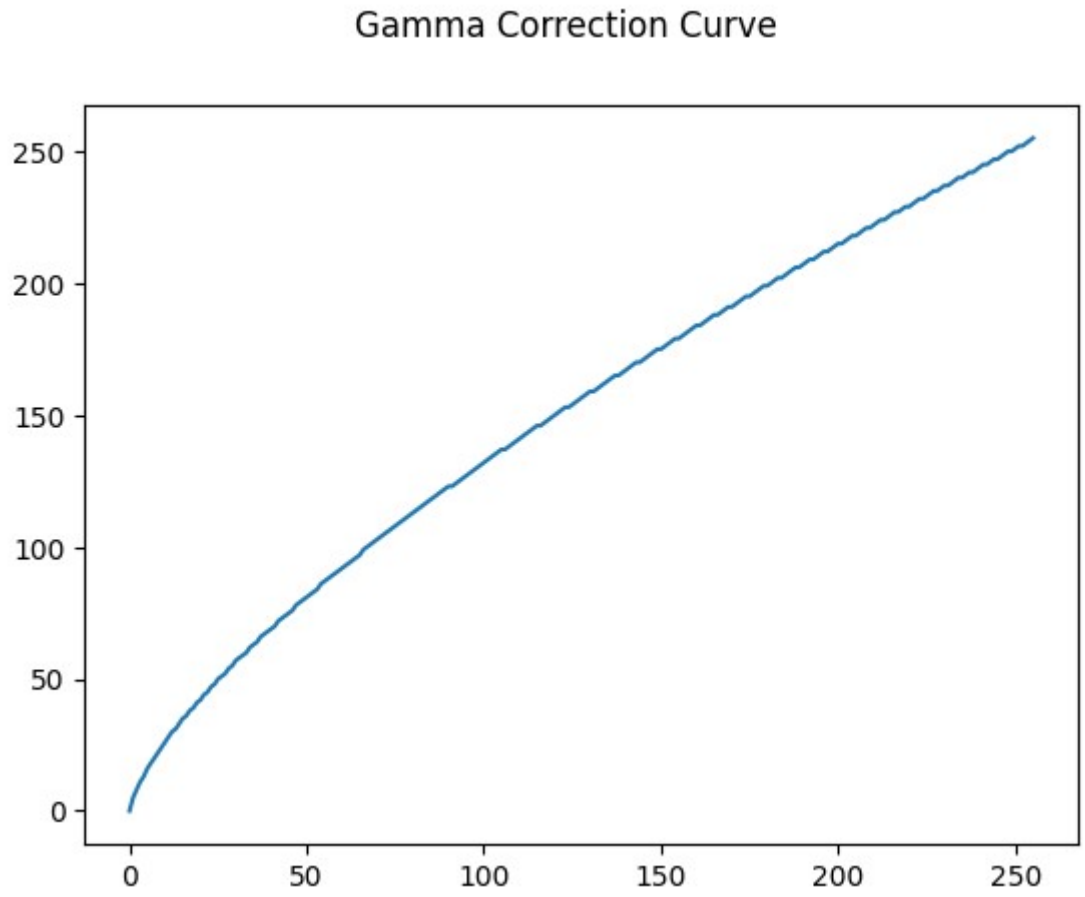


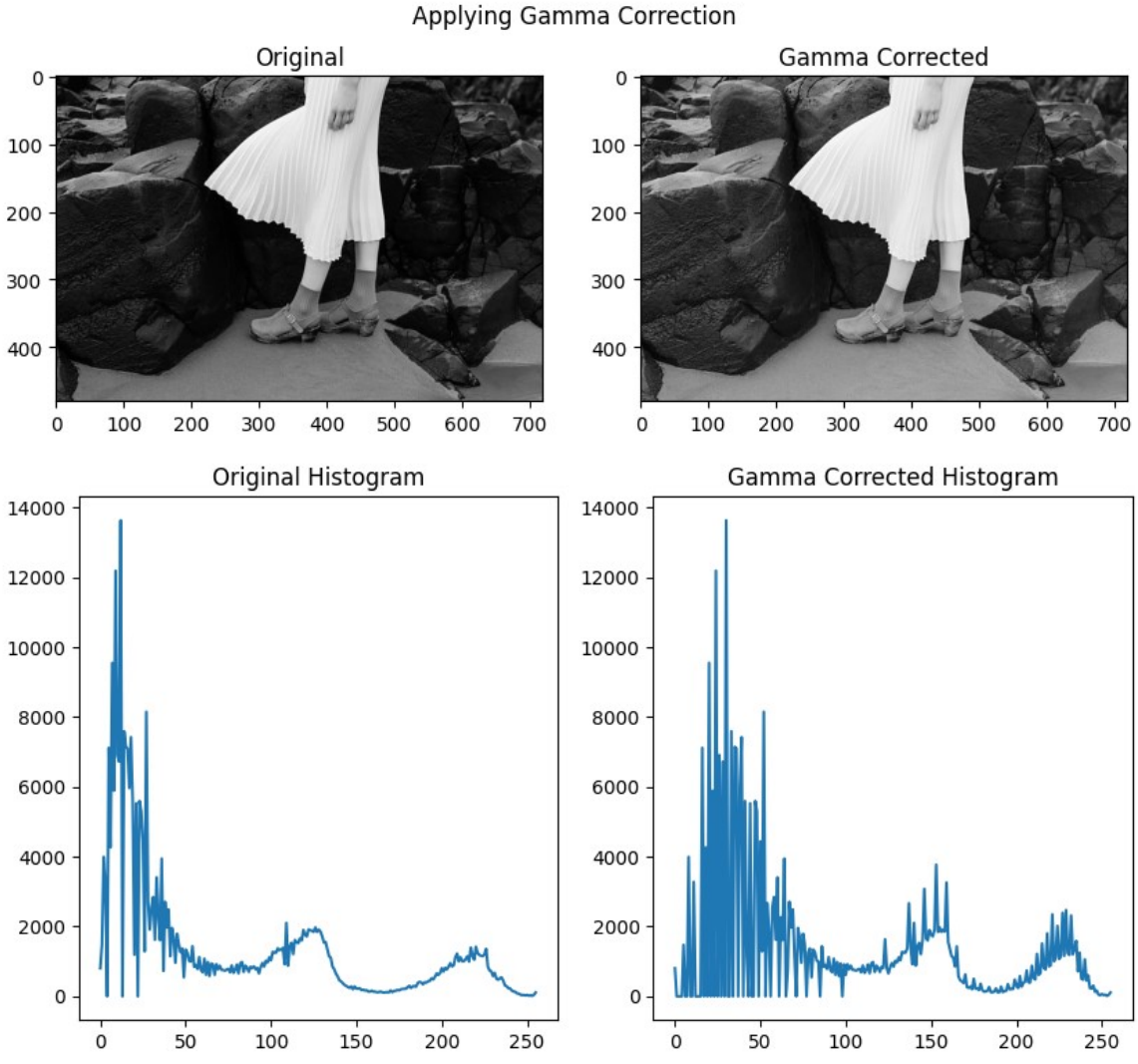
Out[ ]:
```
Text(0.5, 1.0, 'Vibrance Enhanced Image')
```



Question 3

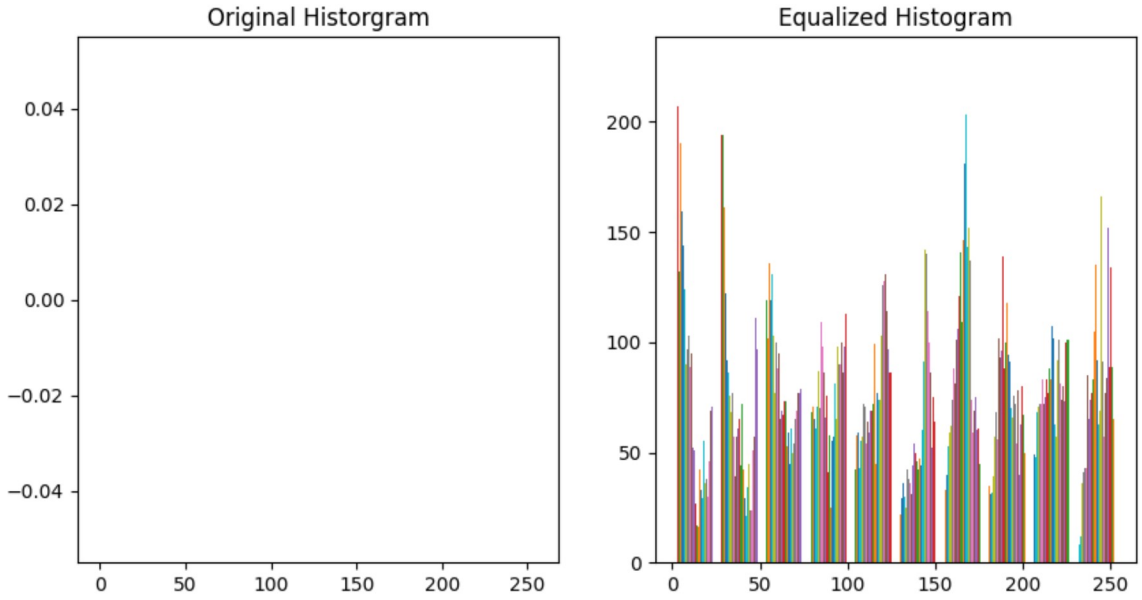
In [ ]:
```
import cv2 as cv
import matplotlib.pyplot as plt
import numpy as np

#Importing the original image
im = cv.imread('highlights_and_shadows.jpg', cv.IMREAD_COLOR)
assert im is not None

im_LAB = cv.cvtColor(im, cv.COLOR_BGR2LAB)
L_im, q_im, r_im = cv.split(im_LAB)

gamma = .7
t = np.array([(i/ 255.)**gamma*255 for i in range (256)], np.uint8)
g = t[L_im]
plt.suptitle("Gamma Correction Curve")
plt.plot(t)
plt.show()

fig, ax = plt.subplots(1, 2, figsize=(10, 3.5))
fig.suptitle("Applying Gamma Correction")
ax[0].imshow(L_im, 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.gca().set_title('Original Histogram')
im_h = cv.calcHist([L_im], [0], None, [256], [0, 256])
plt.plot(im_h)

plt.subplot(1, 2, 2)
plt.gca().set_title('Gamma Corrected Histogram')
g_h = cv.calcHist([g], [0], None, [256], [0, 256])
plt.plot(g_h)
plt.show()
```

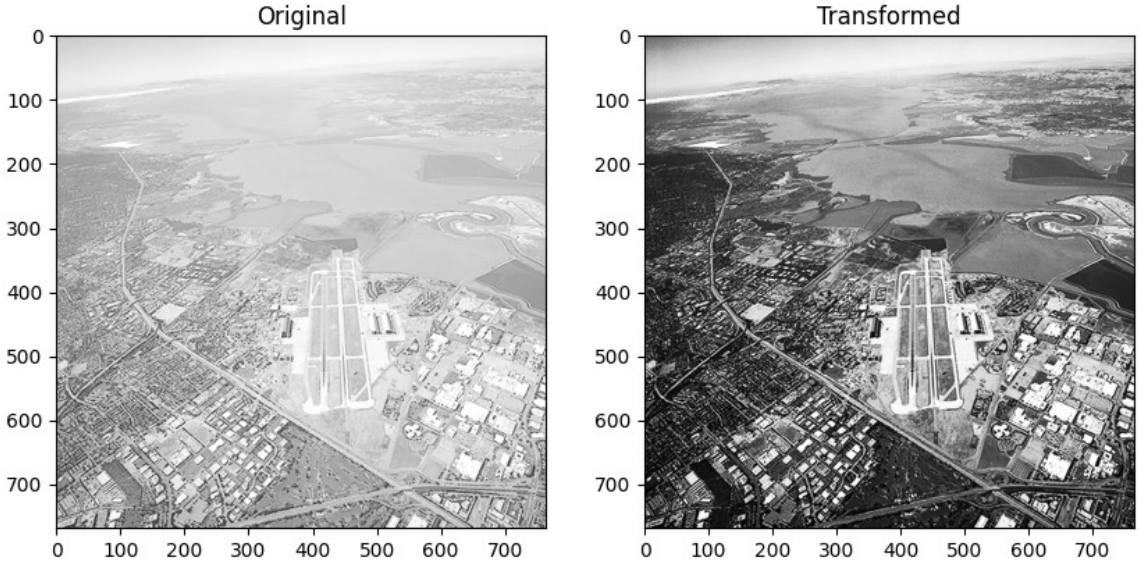

Gamma Correction Curve
250
200
150
100
50
0
0
50
100
150
200
250


Applying Gamma Correction
Original
Gamma Corrected
Original Histogram
Gamma Corrected Histogram

Question 4

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

#Importing the original image
im = cv.imread('washed_out_aerial_image.png', cv.IMREAD_GRAYSCALE)
assert im is not None

plt.figure(figsize = [10, 5])
plt.subplot(1, 2, 1)
plt.gca().set_title('Original Historgram')
h = np.zeros(256)
h = [np.sum(im==1) for i in range (256)]
plt.bar(range(256), h)

plt.subplot(1, 2, 2)
plt.gca().set_title('Equalized Histogram')
eh = cv.equalizeHist(im)
plt.hist(eh)
plt.show()

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

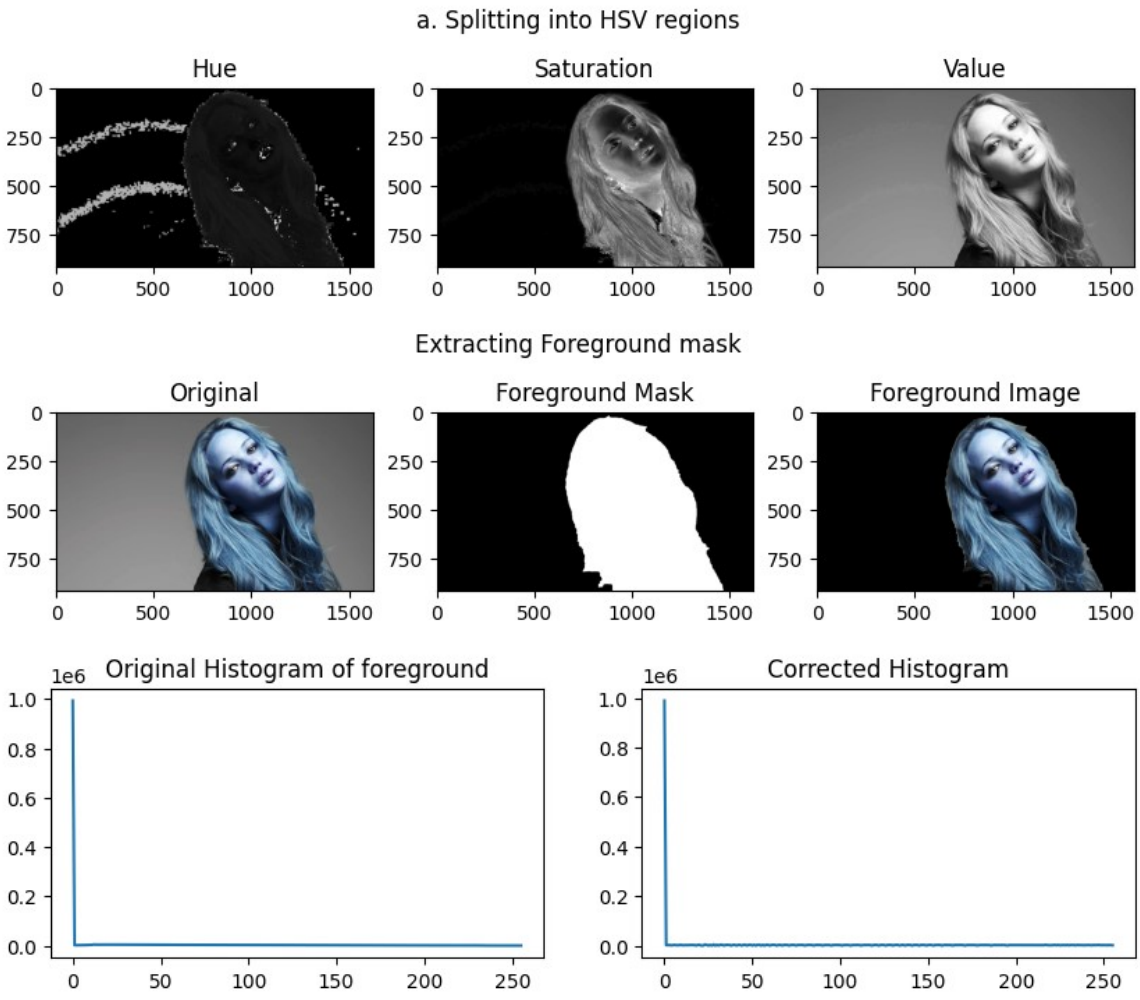


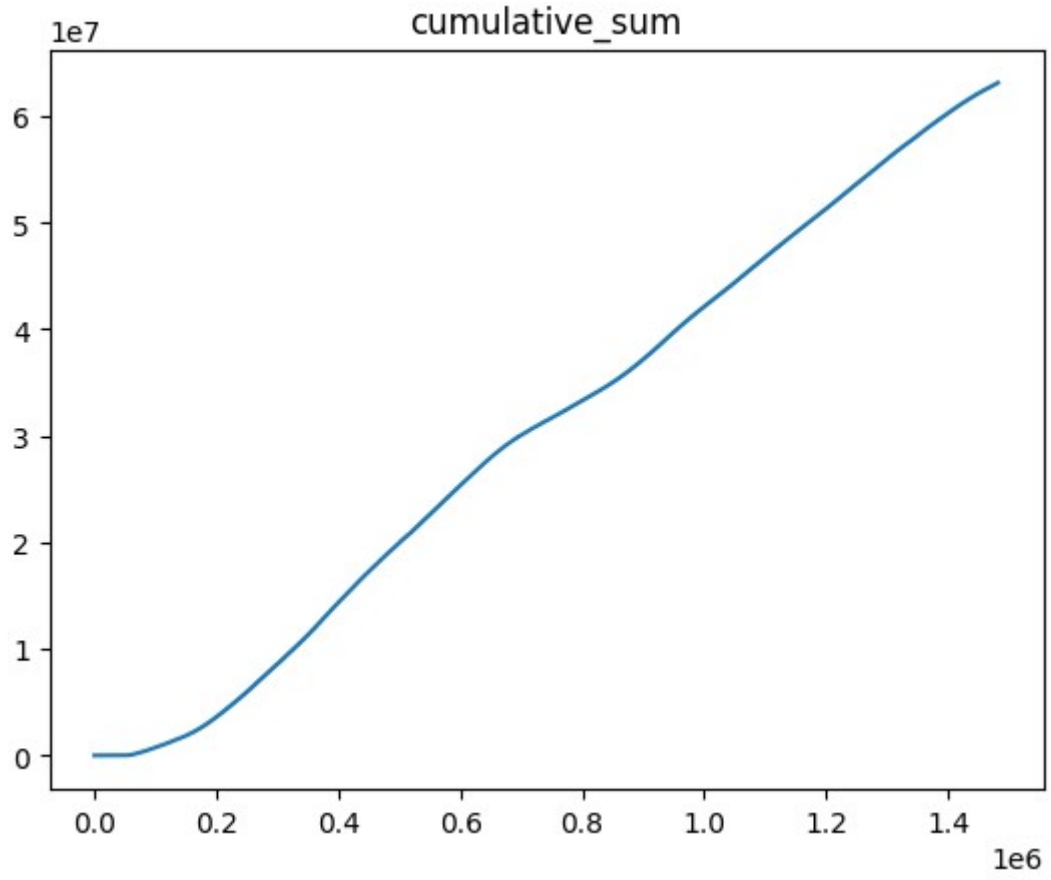


Question 5

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

#Importing the original image
im = cv.imread('jeniffer.jpg', cv.IMREAD_COLOR)
assert im is not None

im1 = cv.cvtColor(im, cv.COLOR_BGR2HSV)
h_img, s_img, v_img = cv.split(im1)

fig, ax= plt.subplots(1, 3, figsize=(10,2.5))
fig.suptitle("a. Splitting into HSV regions", fontsize=12)
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()

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(im, im, mask=mask)

fig, ax = plt.subplots(1, 3, figsize=(10,2.5))
fig.suptitle("Extracting Foreground mask")
ax[0].imshow(im, 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()

#Creating the histogram
plt.figure(figsize = [10, 2.5])
plt.subplot(1, 2, 1)
plt.gca().set_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.gca().set_title('Corrected Histogram')
result1 = cv.cvtColor(result, cv.COLOR_BGR2GRAY)
eh = cv.equalizeHist(result1)
eh1 = cv.calcHist([eh], [0], None, [256], [0,256])
plt.plot(eh1)
plt.show()

cumulative_sum = np.cumsum(eh)
plt.plot(cumulative_sum)
plt.title("cumulative_sum")
plt.show()

mask1 = 255 - morph
bg_img = cv.bitwise_and(im, im, mask=mask1)
bg_img1 = cv.cvtColor(bg_img, cv.COLOR_BGR2GRAY);
```

```
img1 = cv.add(bg_img1,eh)

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(eh, cmap="gray")
ax[1].set_title("Foreground")
ax[2].imshow(img1, cmap="gray")
ax[2].set_title("Added Image")
plt.show()
```





f. Adding background with equalized