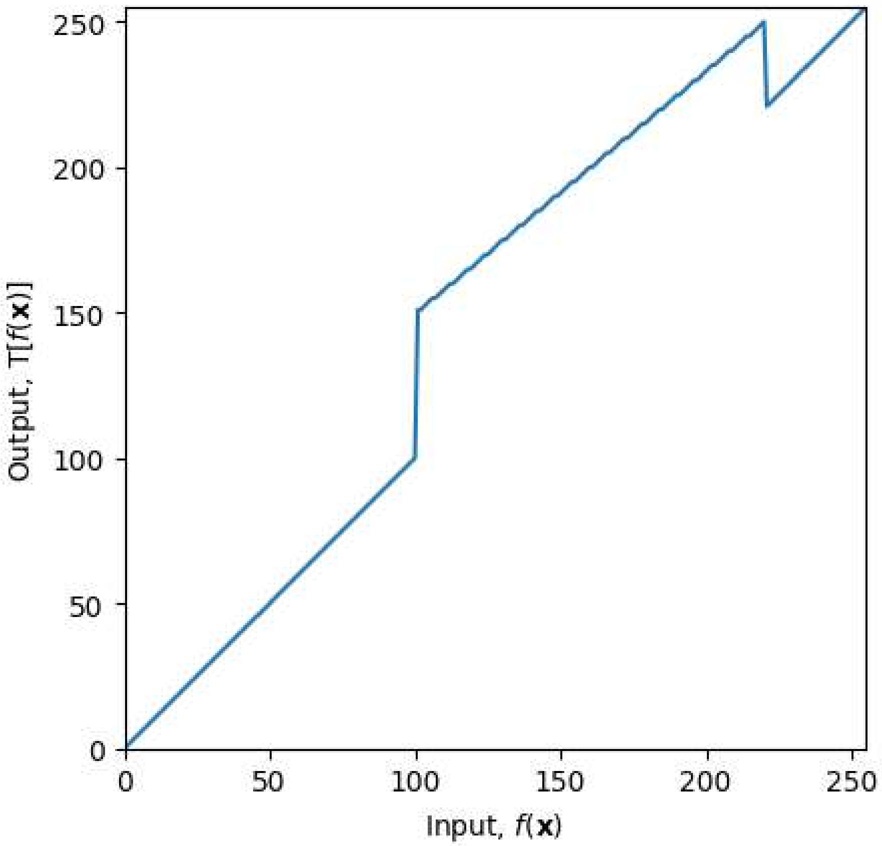
In [ ]:

```
%matplotlib inline
import cv2 as cv
import matplotlib.pyplot as plt
import numpy as np

c = np.array([(100,100), (100,150), (220,250), (220,220)])

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, c[2,1], c[2,0] - c[1,0] ).astype('uint8')
print(len(t3))
t4 = np.linspace(c[2,1]+1, c[3,1], c[3,0] - c[2,0]).astype('uint8')
print(len(t4))
t5 = np.linspace(c[3,1]+1, 255, 255- c[3,0] ).astype('uint8')
print(len(t5))

transform = np.concatenate((t1,t2), axis=0).astype('uint8')
transform = np.concatenate((transform,t3), axis=0).astype('uint8')
transform = np.concatenate((transform,t4), 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.set_aspect('equal')
# plt.savefig('../transform.png')

plt.show()


img_org = cv.imread('../Resources/natasha_grayscale.jpg', cv.IMREAD_GRAYSCALE)
print(img_org.shape)
cv.namedWindow('Image', cv.WINDOW_AUTOSIZE)
cv.imshow('Image', img_org)
cv.waitKey(0)
image_transformed = cv.LUT(img_org,transform)
cv.imshow('Image', image_transformed)
cv.waitKey(0)
cv.destroyAllWindows()
```

```
101
0
120
0
35
256
```

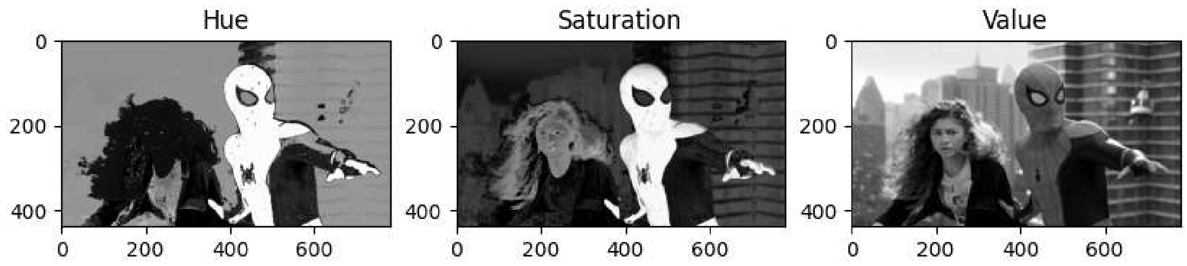


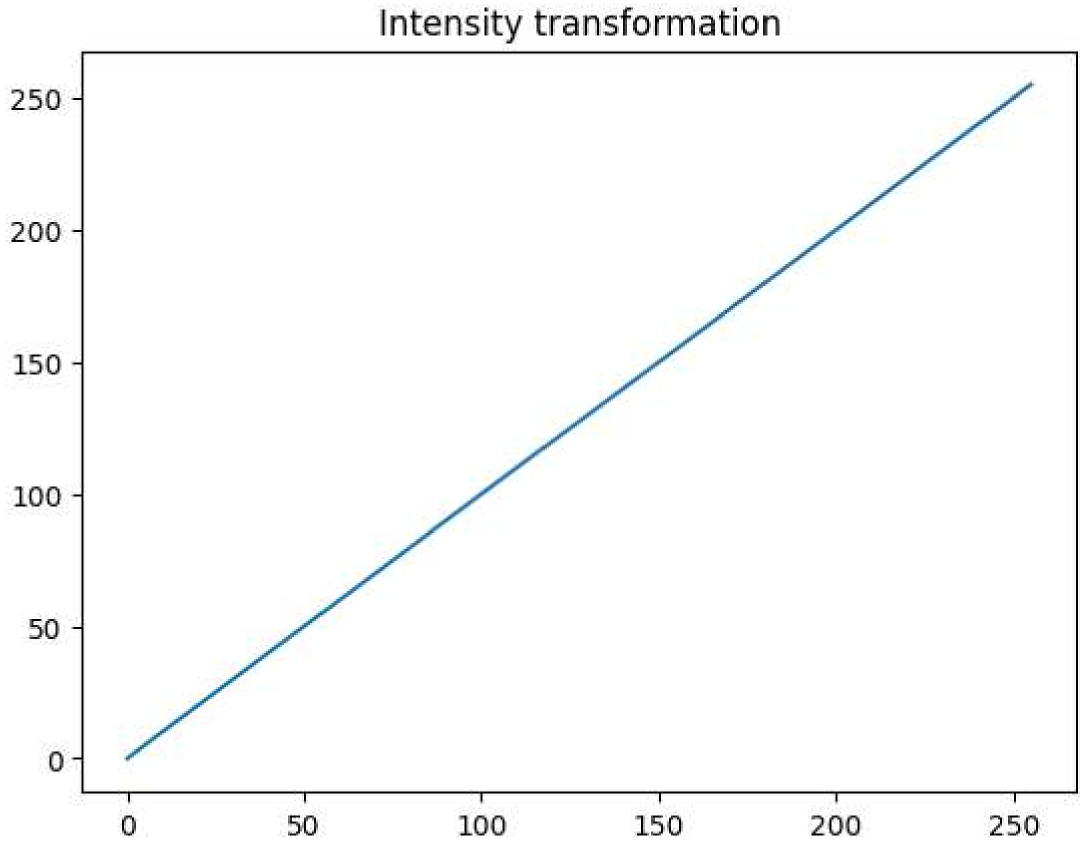
```
(400, 700)
```

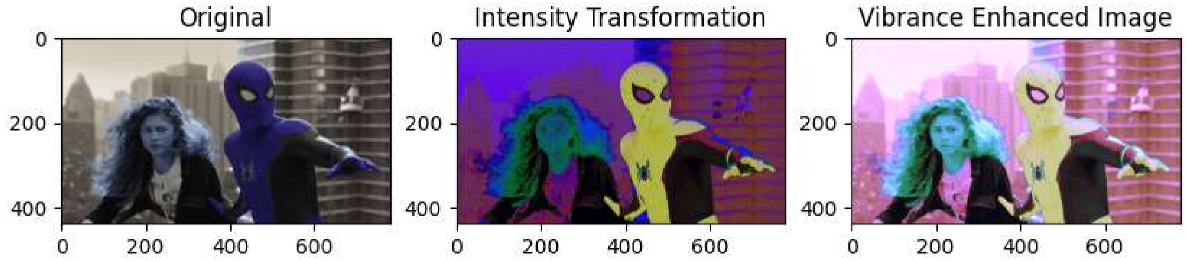
In [ ]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import cv2 as cv
import numpy as np
img = cv.imread('../Resources/spider.png')
if img is None:
    print('Image could not be read')
    assert False
img1 = cv.cvtColor(img, cv.COLOR_BGR2HSV)
hue, saturation, value = cv.split(img1)
fig, ax= plt.subplots(1,3, figsize=(10,20))
ax[0].imshow(hue, cmap="gray")
ax[0].set_title('Hue')
ax[1].imshow(saturation, cmap="gray")
ax[1].set_title('Saturation')
ax[2].imshow(value, cmap="gray")
ax[2].set_title('Value')
plt.show() # result up to here is in answer a

x= np.arange(0, 256).astype('uint8') # define range for x variable
a = 1
sigma = 70
T = np.minimum(((x)+(a*(np.exp(-(x-128)**2/(2*sigma**2))))/128), 255).astype('uint8
image_transform = cv.LUT(saturation, T) # adding transformation to saturation plane
plt.title('Intensity transformation')
plt.plot(T)
plt.show() # result here in in answer b
```

```
new_HSV = cv.merge([hue,image_transform,value])
result =  cv.cvtColor(new_HSV,  cv.COLOR_HSV2BGR)
added_img = cv.add(new_HSV, img)
fig, ax= plt.subplots(1,3, figsize=(10,20))
ax[0].imshow(img, cmap="gray")
ax[0].set_title('Original')
ax[1].imshow(new_HSV, cmap="gray")
ax[1].set_title('Intensity Transformation')
ax[2].imshow(added_img, cmap="gray")
ax[2].set_title('Vibrance Enhanced Image')
plt.show() # result of here is given in answer d
```


Hue
Saturation
Value


Intensity transformation


Original
Intensity Transformation
Vibrance Enhanced Image