Task #1. Read the image from parrot.png. Calculate negative of the image. It means that pixels with 0 value should have value 255 and vice versa if your image has byte type.

For example,

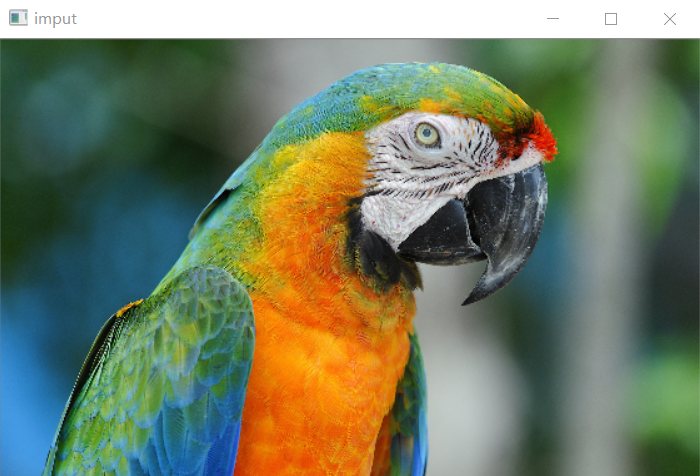
Byte image: 255 -> 0, 0 -> 255, 1 -> 254, 240 -> 15, etc.

Float image: 0 -> 1, 1 -> 0, 0.4 -> 0.6, etc.

Show the image negative.

**import** cv2 **as** cv  
**import** numpy **as** np  
src=cv.imread(**"D:/picture1/parrot.png"**)  
h,w=src.shape[:2]  
print(src.shape)  
cv.imshow(**"imput"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
**for** row **in** range(h):  
 **for** col **in** range(w):  
 b,g,r=src[row,col]  
 b=255-b  
 g=255-g  
 r=255-r  
 src[row,col]=[b,g,r]  
cv.imshow(**"output"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
cv.waitKey(0)  
cv.destroyAllWindows()

(1050, 1680, 3)





Task #2. Read the image from link <https://www.nastol.com.ua/pic/201210/1920x1200/nastol.com.ua-34781.jpg>. In this image, swap the channels so that instead of the RGB order, the channels are in BRG order. Use the function dstack.

Method 1

**import** cv2 **as** cv  
**import** numpy **as** np  
src=cv.imread(**"D:/picture1/tiger.jpg"**)  
h,w=src.shape[:2]  
print(src.shape)  
cv.imshow(**"imput"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
**for** row **in** range(h):  
 **for** col **in** range(w):  
 b,g,r=src[row,col]  
 b=r  
 r=b  
 src[row,col]=[b,g,r]  
cv.imshow(**"output"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
cv.waitKey(0)  
cv.destroyAllWindows()





Method 2

**import** cv2 **as** cv  
**import** numpy **as** np  
src=cv.imread(**"D:/picture1/tiger.jpg"**)  
h,w=src.shape[:2]  
print(src.shape)  
cv.imshow(**"imput"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
b,g,r=cv.split(src)  
src=cv.merge([r,g,b])  
cv.imshow(**"output"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
cv.waitKey(0)  
cv.destroyAllWindows()





Method 3

**import** cv2 **as** cv  
**import** numpy **as** np  
src=cv.imread(**"D:/picture1/tiger.jpg"**)  
h,w=src.shape[:2]  
print(src.shape)  
cv.imshow(**"imput"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
src=src[: , : , : : -1]  
cv.imshow(**"output"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
cv.waitKey(0)  
cv.destroyAllWindows()





Task #3. Read a color image

from <https://www.nastol.com.ua/pic/201210/1920x1200/nastol.com.ua-34781.jpg>. Calculate the grayscale image of it. The resulting image must be single-channel. To calculate the gray channel, use the formula $Y=0.2126⋅R+0.7152⋅G+0.0722⋅B$. Do not forget to first convert the image to real numbers (img\_as\_float function), use transformation and after convert to integers (img\_as\_ubyte function).

**import** cv2 **as** cv  
**import** numpy **as** np  
src=cv.imread(**"D:/picture1/tiger.jpg"**)  
h,w=src.shape[:2]  
print(src.shape)  
cv.imshow(**"imput"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
f\_src=np.float32(src)  
b,g,r=cv.split(f\_src)  
Y=0.2126\*r+0.7152\*g+0.0722\*b  
Y=cv.resize(Y,(w//3,h//3),interpolation=cv.INTER\_LINEAR)  
cv.imshow(**"output"**, Y.astype(np.uint8))  
cv.waitKey(0)  
cv.destroyAllWindows()

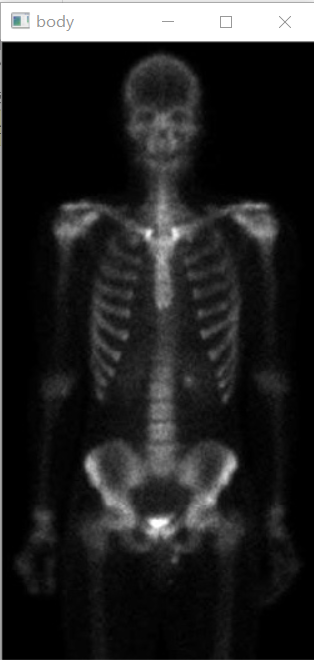


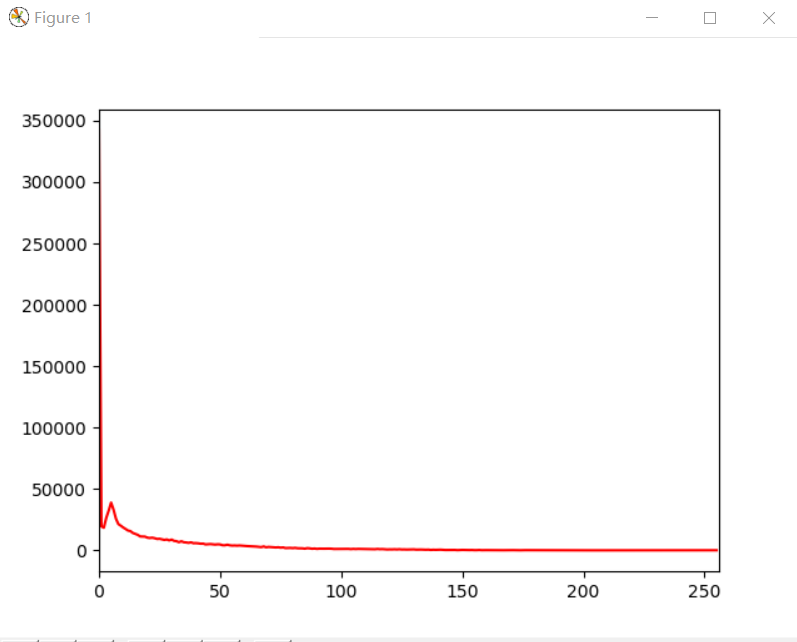


Task #4. Given a grayscale image of the moon surface. Plot histogram of this image. Apply different operaton for increasing and decreasing of contrast and brightness. Plot an image and its histogram after each operation by using subplot.

**import** cv2 **as** cv  
**from** matplotlib **import** pyplot **as** plt  
**import** numpy **as** np  
src=cv.imread(**"D:/picture1/body.tif"**)  
h,w=src.shape[:2]  
print(src.shape)  
cv.imshow(**"body"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
hist=np.zeros([256],dtype=np.int32)  
**for** row **in** range(h):  
 **for** col **in** range(w):  
 pv=src[row,col]  
 hist[pv]+=1  
plt.plot(hist,color=**"r"**)  
plt.xlim([0,256])  
plt.show()  
cv.waitKey(0)  
cv.destroyAllWindows()

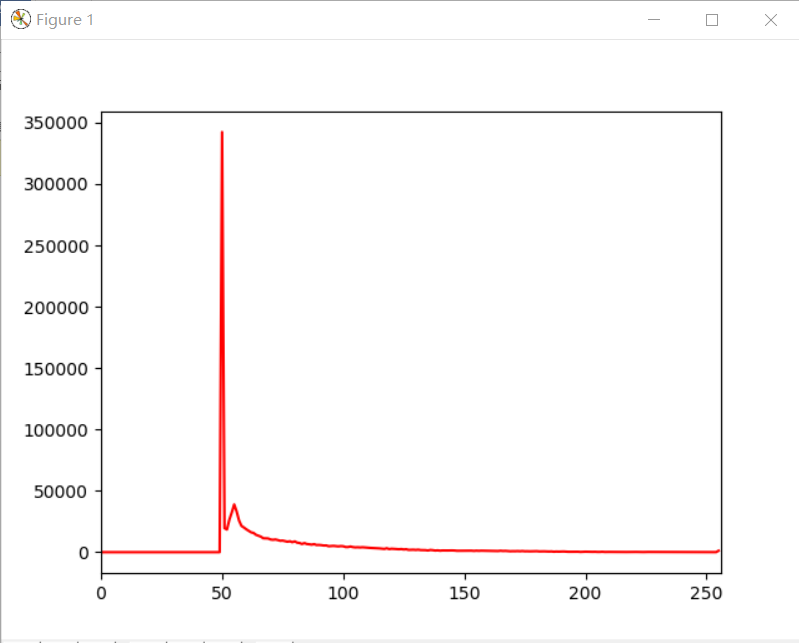
(1482, 750, 3)





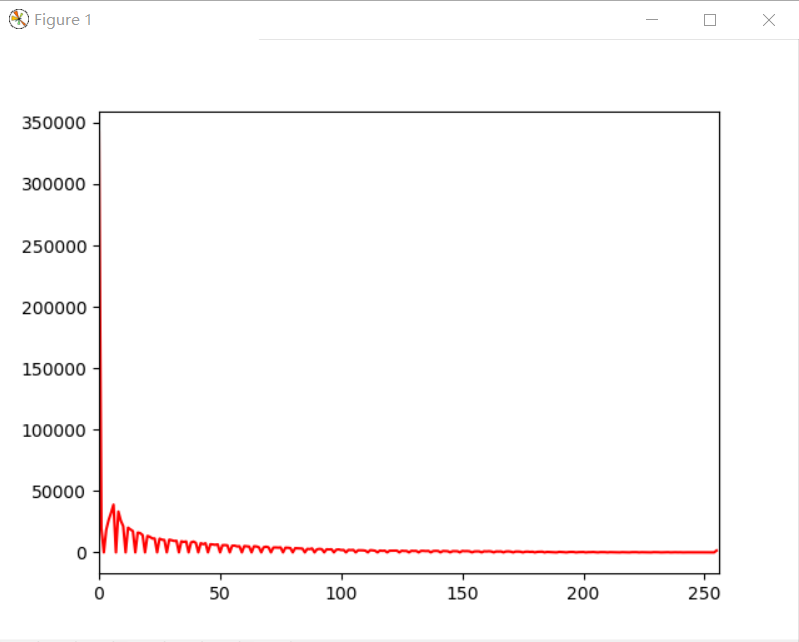
**import** cv2 **as** cv  
**from** matplotlib **import** pyplot **as** plt  
**import** numpy **as** np  
src=cv.imread(**"D:/picture1/body.tif"**)  
h,w=src.shape[:2]  
empty=np.zeros(src.shape,src.dtype)  
src=cv.addWeighted(src,1,empty,0,50)  
cv.imshow(**"body"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
hist=np.zeros([256],dtype=np.int32)  
**for** row **in** range(h):  
 **for** col **in** range(w):  
 pv=src[row,col]  
 hist[pv]+=1  
plt.plot(hist,color=**"r"**)  
plt.xlim([0,256])  
plt.show()  
cv.waitKey(0)  
cv.destroyAllWindows()





**import** cv2 **as** cv  
**from** matplotlib **import** pyplot **as** plt  
**import** numpy **as** np  
src=cv.imread(**"D:/picture1/body.tif"**)  
h,w=src.shape[:2]  
empty=np.zeros(src.shape,src.dtype)  
src=cv.addWeighted(src,1.3,empty,0,0)  
cv.imshow(**"body"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
hist=np.zeros([256],dtype=np.int32)  
**for** row **in** range(h):  
 **for** col **in** range(w):  
 pv=src[row,col]  
 hist[pv]+=1  
plt.plot(hist,color=**"r"**)  
plt.xlim([0,256])  
plt.show()  
cv.waitKey(0)  
cv.destroyAllWindows()





**import** cv2 **as** cv  
**from** matplotlib **import** pyplot **as** plt  
**import** numpy **as** np  
src=cv.imread(**"D:/picture1/body.tif"**)  
h,w=src.shape[:2]  
empty=np.zeros(src.shape,src.dtype)  
src=cv.addWeighted(src,1.3,empty,0,50)  
cv.imshow(**"body"**, cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR))  
hist=np.zeros([256],dtype=np.int32)  
**for** row **in** range(h):  
 **for** col **in** range(w):  
 pv=src[row,col]  
 hist[pv]+=1  
plt.plot(hist,color=**"r"**)  
plt.xlim([0,256])  
plt.show()  
cv.waitKey(0)  
cv.destroyAllWindows()



