Name : Tanay Manerikar

Batch : N8

Roll Number : 32343

In [2]:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
image_path = r"../images/"
```

In [44]:

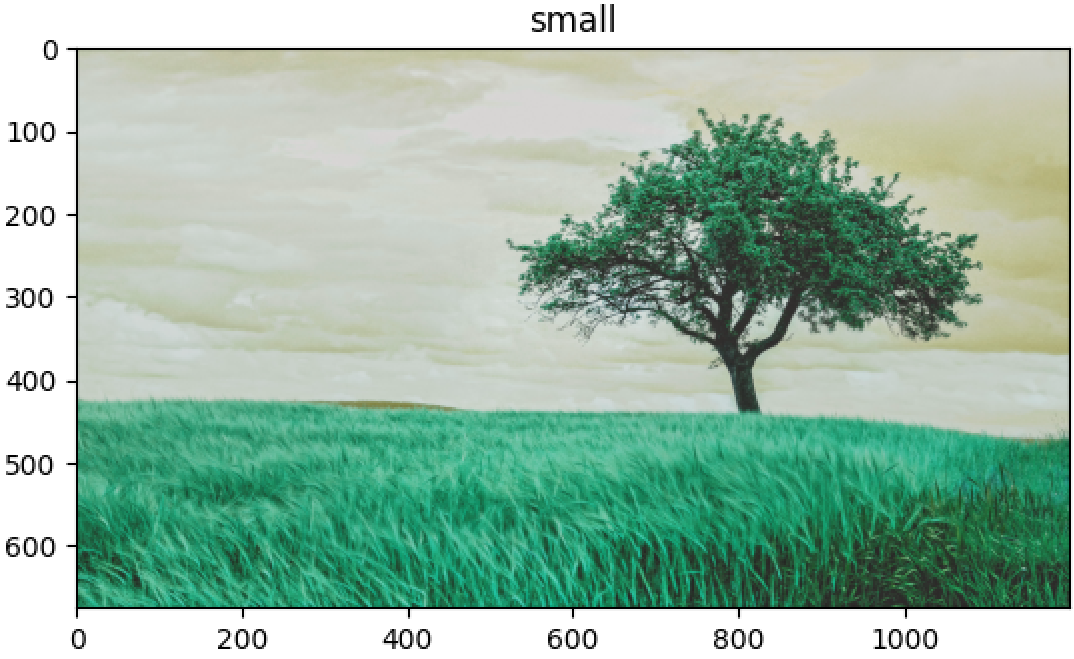
```
#Loads and display image
def img_load(image_address):
    image = cv2.imread(image_address,1)
    return image

def img_disp(image , window_name="Image"):
    cv2.imshow(window_name,image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
    return
def img_disp_plt(image,desc="Image"):
    if(len(image.shape)==3):
        RGB_im = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        plt.title(desc)
        plt.imshow(image)
        plt.show()
    else:
        plt.title(desc)
        plt.imshow(image,cmap='gray')
        plt.show()
    return
```

In [15]:

```
sample_image = img_load(image_path+"1.jpg")
img_disp(sample_image)
```

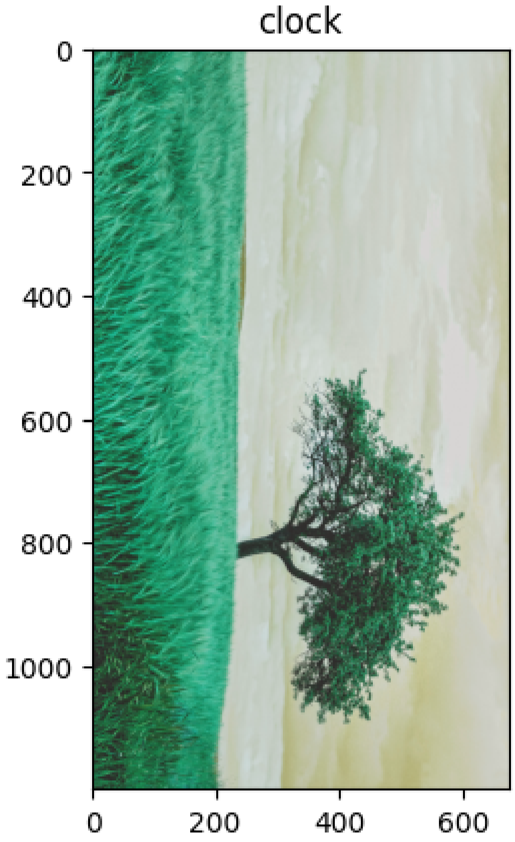
In [50]:

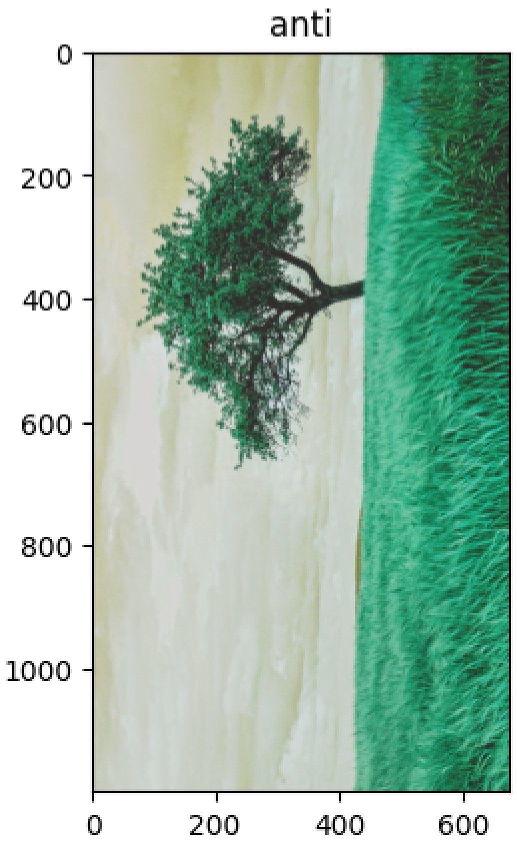
```
# Resize Image
small = cv2.resize(sample_image,(1200,675))
img_disp(small,"small")
img_disp_plt(small,"small")
```

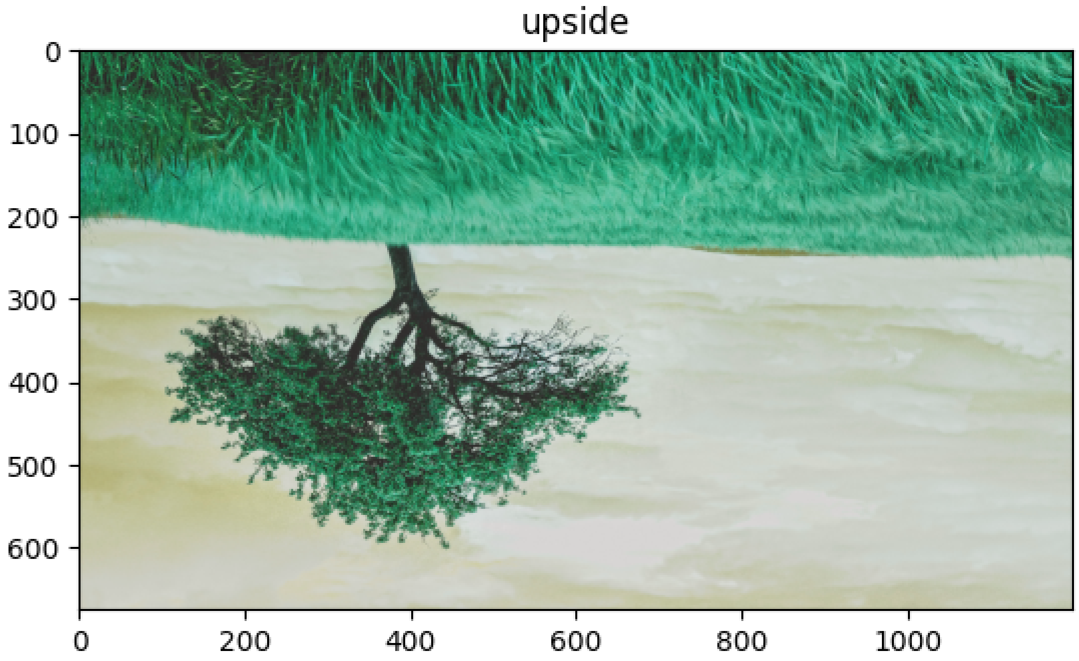


In [35]:

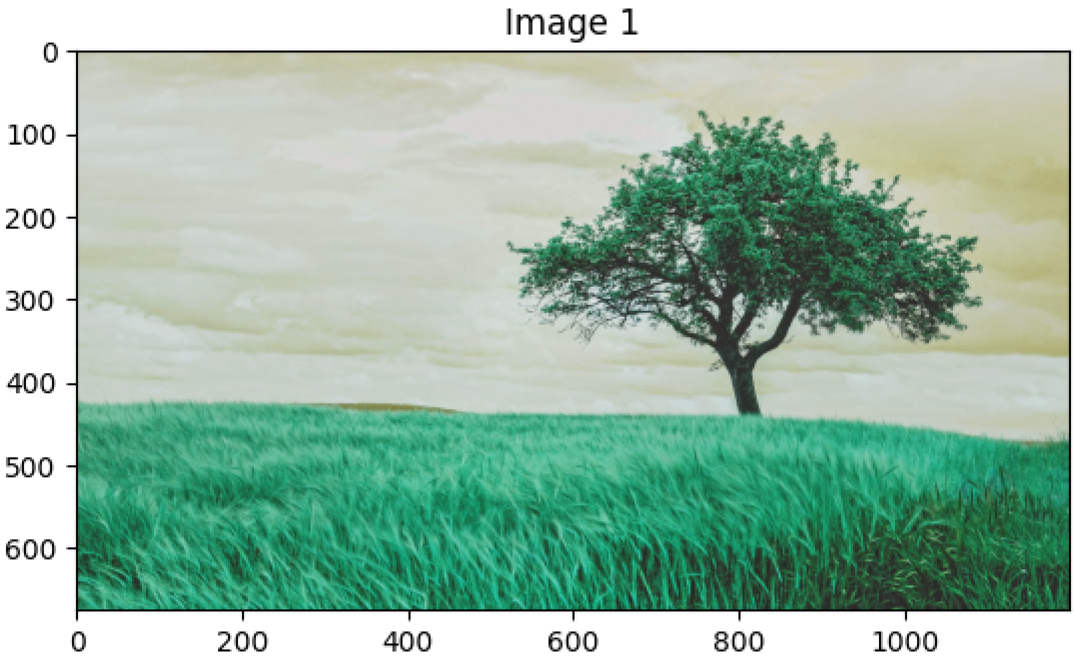
```
# Rotate
clock = cv2.rotate(small,cv2.ROTATE_90_CLOCKWISE)
anti = cv2.rotate(small,cv2.ROTATE_90_COUNTERCLOCKWISE)
upside = cv2.rotate(small,cv2.ROTATE_180)
img_disp(clock,"clock")
img_disp(anti,"anti")
img_disp(upside,"upside")
img_disp_plt(clock,"clock")
img_disp_plt(anti,"anti")
img_disp_plt(upside,"upside")
```

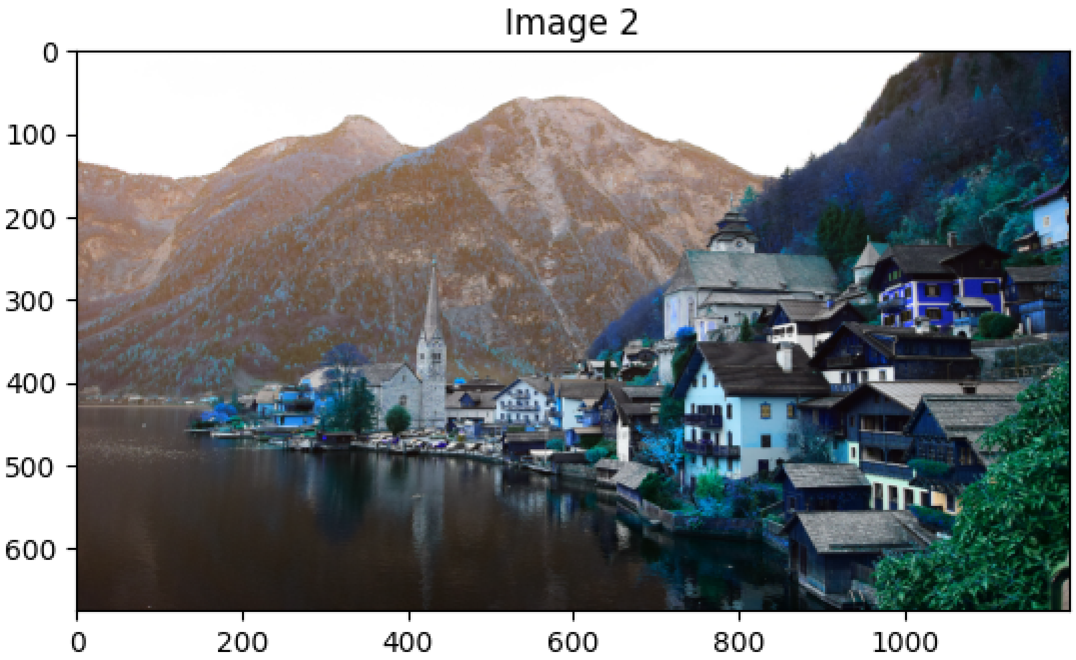


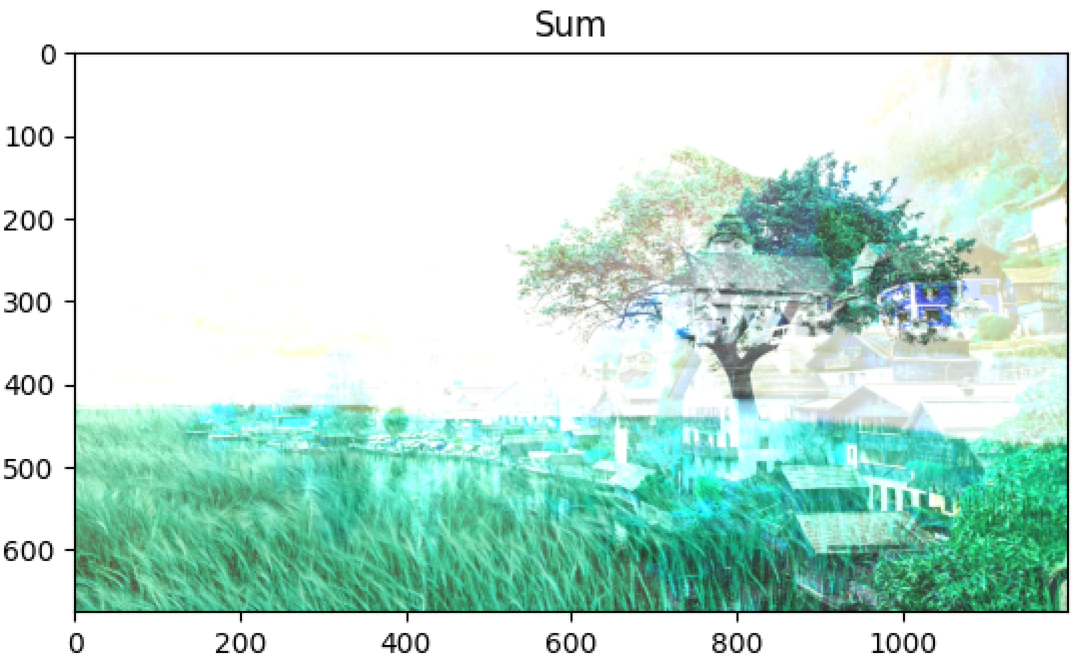

anti
0
200
400
600
800
1000
0
200
400
600


upside
0
100
200
300
400
500
600
0
200
400
600
800
1000

In [53]:
```python
# Add images
img_disp(small,"Image 1")
img_disp_plt(small,"Image 1")


img2 = cv2.imread(image_path+"2.jpg",1)
small2 = cv2.resize(img2,(1200,675))
img_disp(small2,"Image 2")
img_disp_plt(small2,"Image 2")

img_sum = cv2.add(small,small2)

img_disp(img_sum,"Sum")
img_disp_plt(img_sum,"Sum")
```

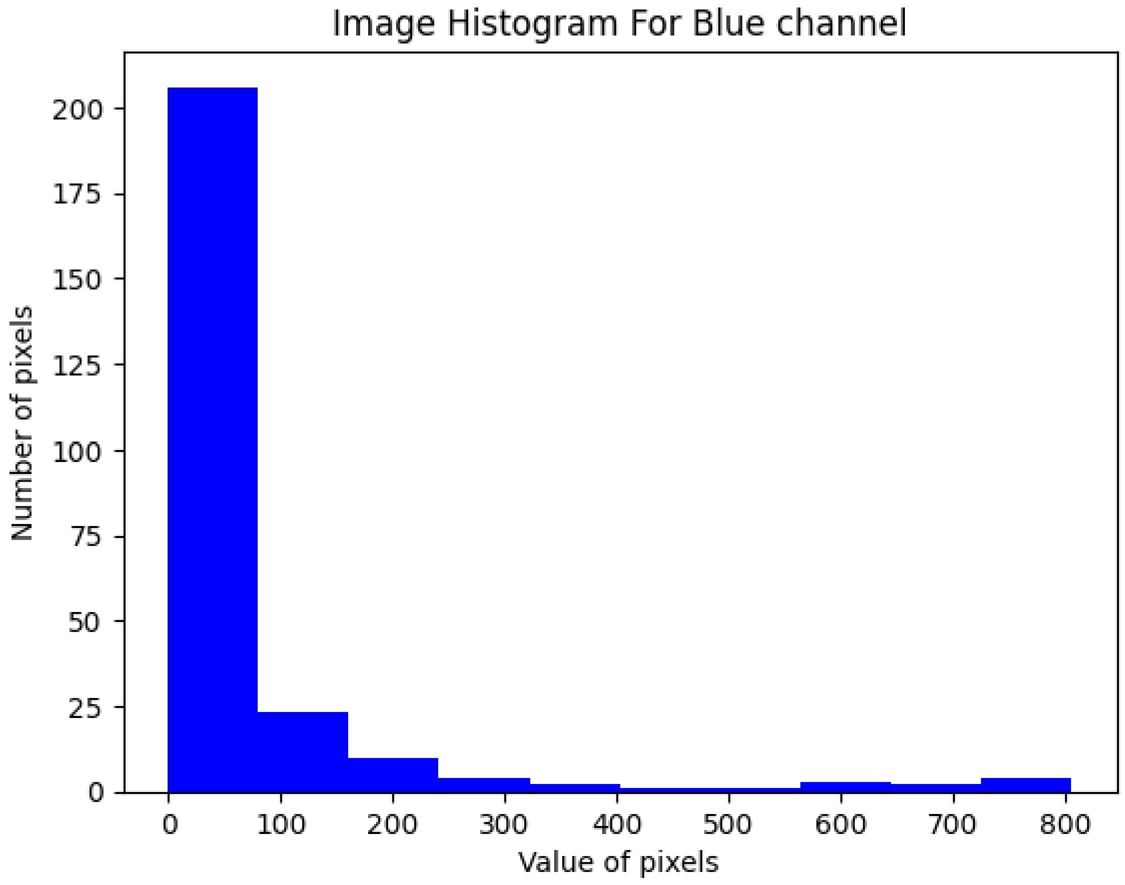





Sum
0
100
200
300
400
500
600
0
200
400
600
800
1000

In [48]:

```
# Plot histogram
b_hist = cv2.calcHist(sample_image,channels=[0],mask=None,histSize=[256],ran
plt.hist(r_hist, color='b')
plt.title('Image Histogram For Blue channel')
plt.xlabel("Value of pixels")
plt.ylabel("Number of pixels")
plt.show()

g_hist = cv2.calcHist(sample_image,channels=[1],mask=None,histSize=[256],ran
plt.hist(r_hist, color='g')
plt.title('Image Histogram For Green channel')
plt.xlabel("Value of pixels")
plt.ylabel("Number of pixels")
plt.show()

r_hist = cv2.calcHist(sample_image,channels=[2],mask=None,histSize=[256],ran
plt.hist(r_hist, color='r')
plt.title('Image Histogram For Red channel')
plt.xlabel("Value of pixels")
plt.ylabel("Number of pixels")
plt.show()
```



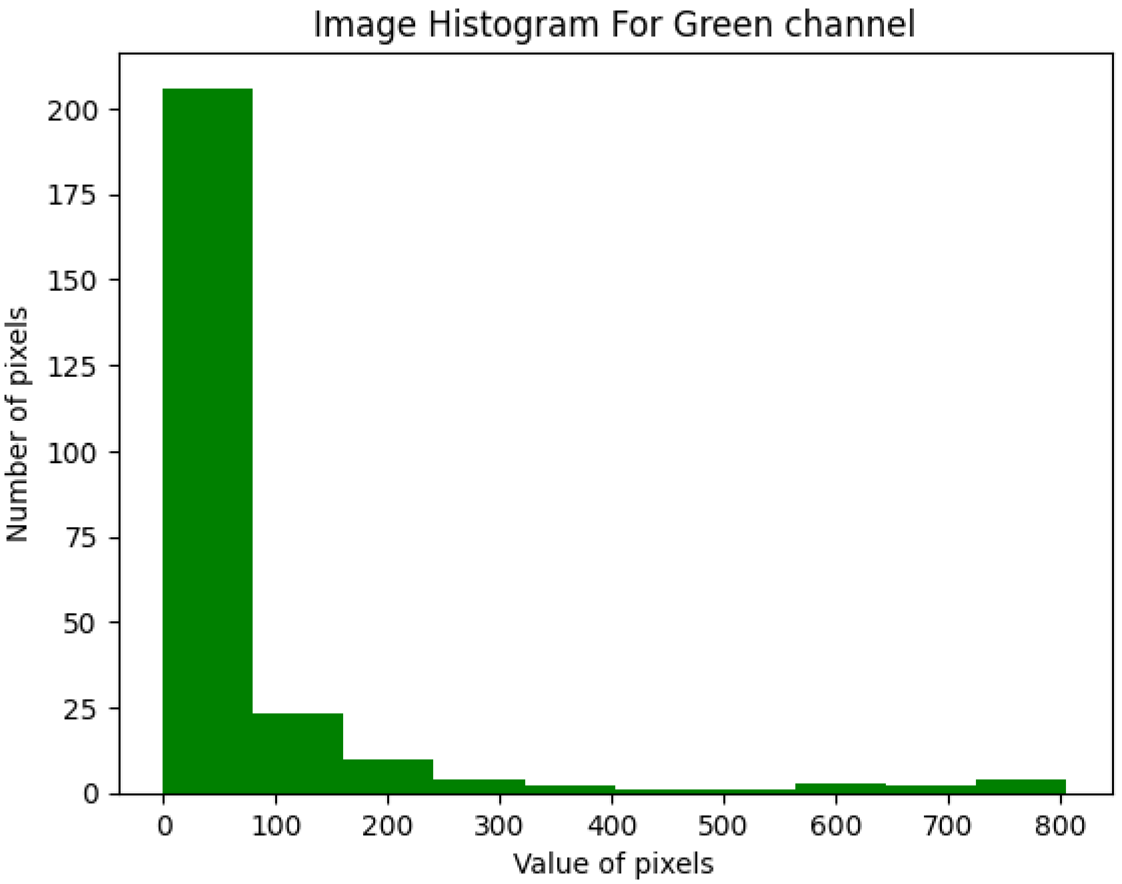


Image Histogram For Green channel

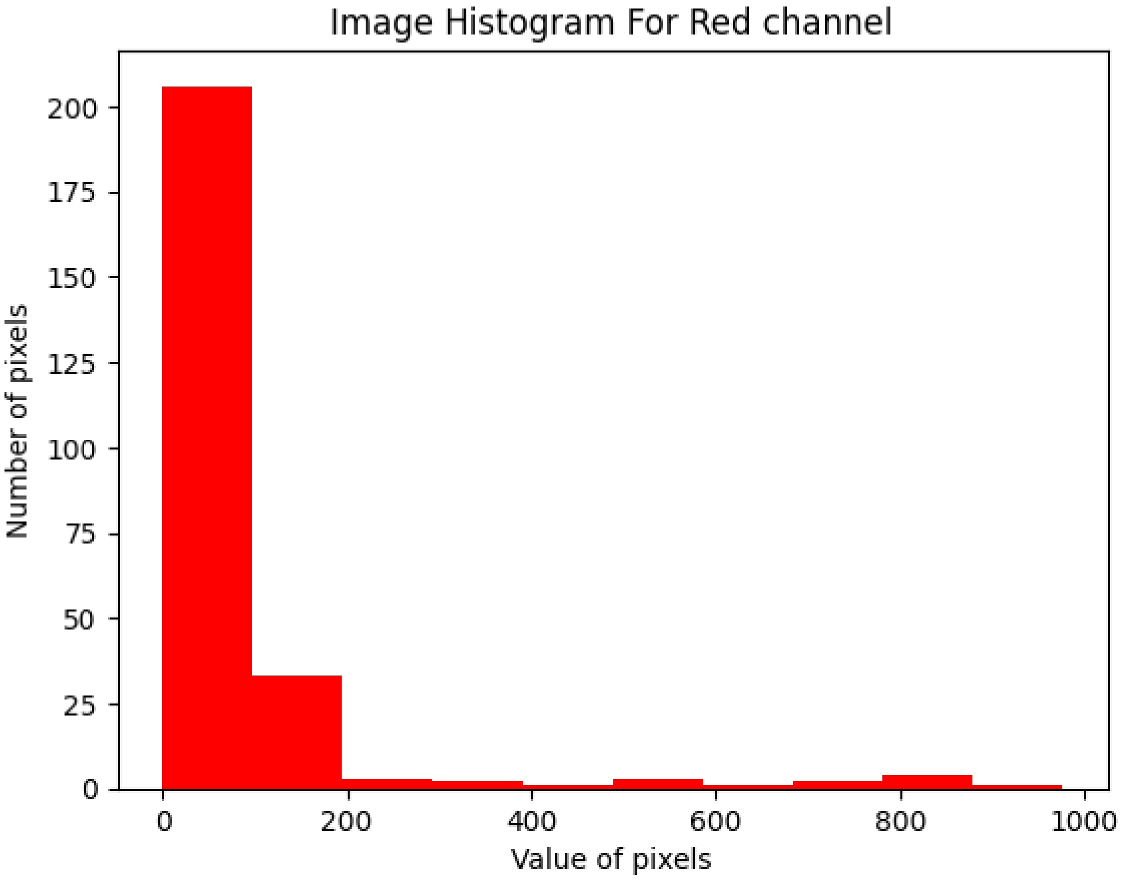
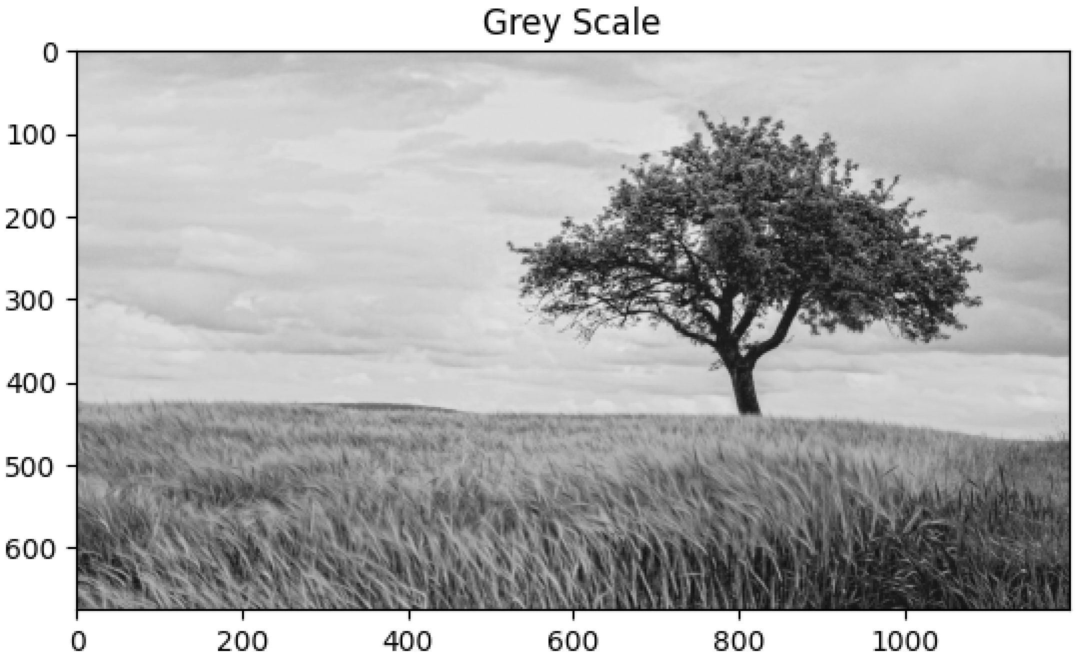


Image Histogram For Red channel

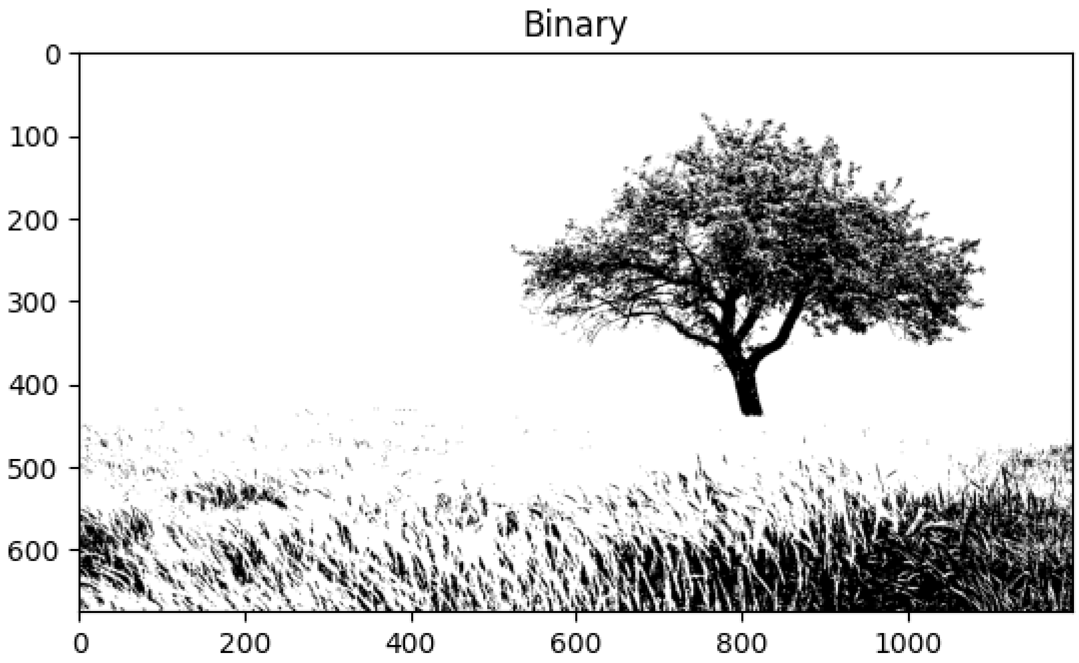
In [45]:

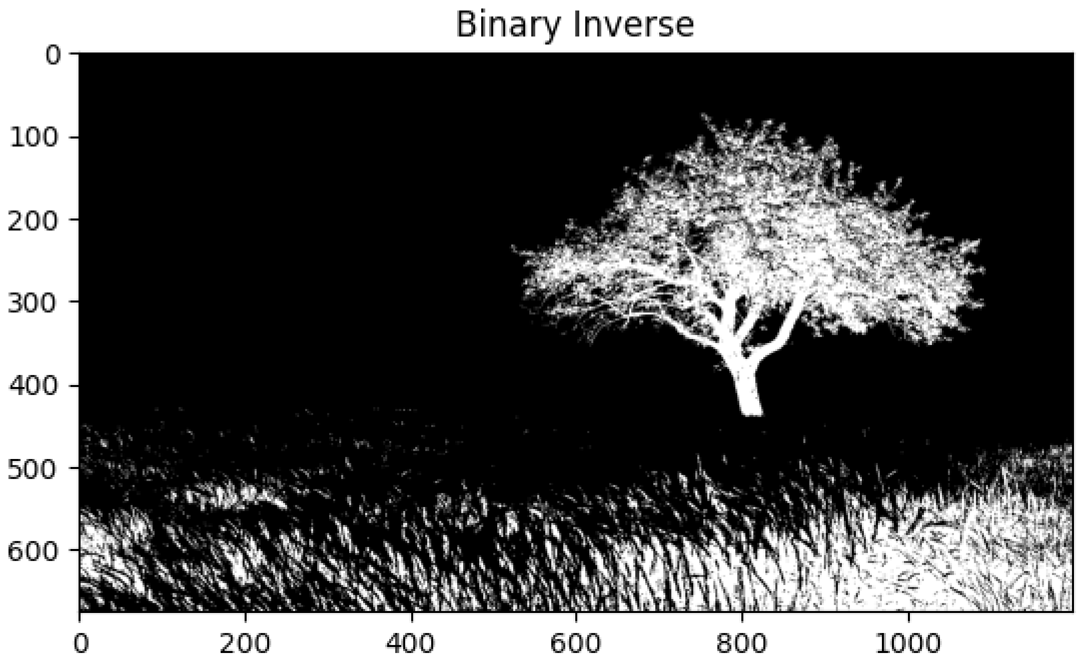
```
# Colour conversion
grey_scale = cv2.cvtColor(small,cv2.COLOR_BGR2GRAY)
img_disp(grey_scale,"Grey Scale")
img_disp_plt(grey_scale,"Grey Scale")
```

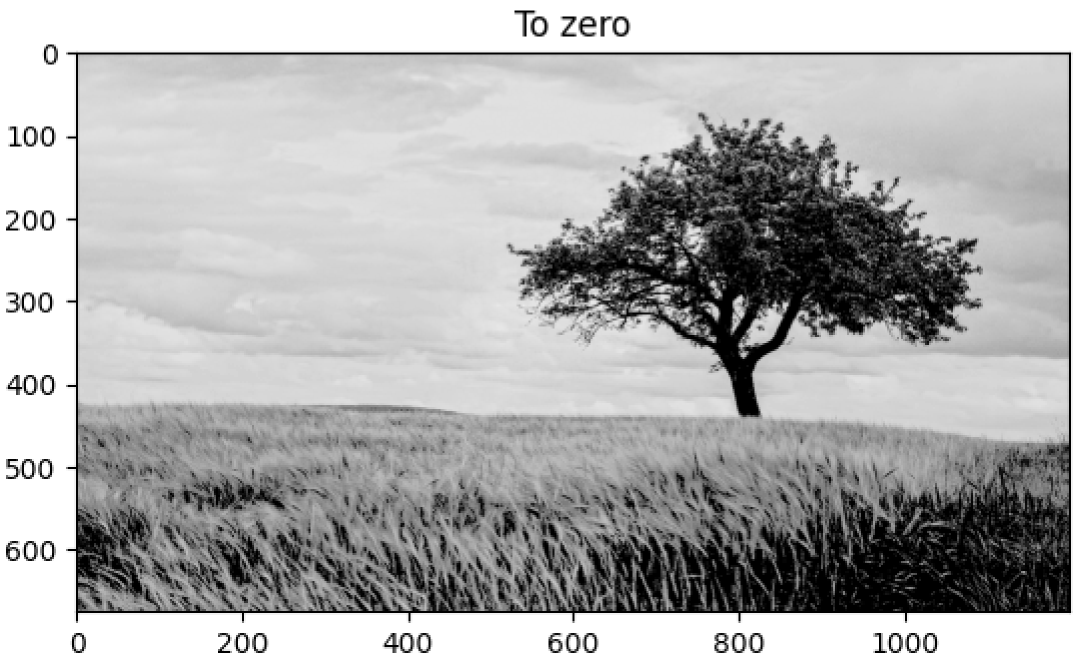


In [46]:

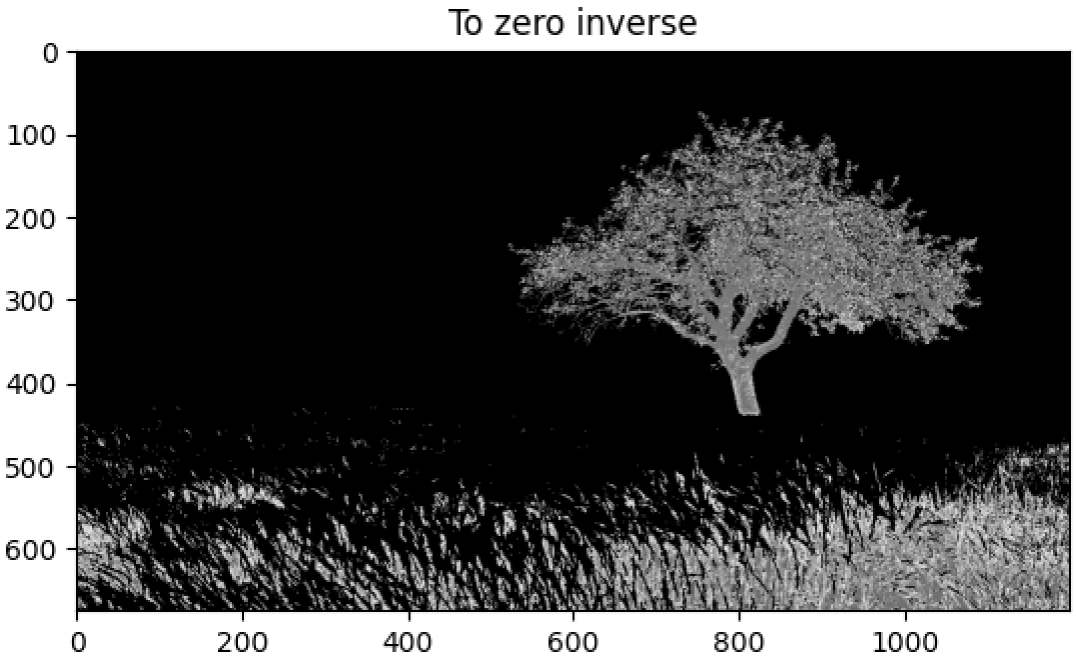
```
# Thresholding
thresh = [[cv2.threshold(grey_scale, 100, 256,cv2.THRESH_BINARY),"Binary"]]
thresh.append([cv2.threshold(grey_scale, 100, 256,cv2.THRESH_BINARY_INV),"Bi
thresh.append([cv2.threshold(grey_scale, 100, 256,cv2.THRESH_TOZERO),"To zer
thresh.append([cv2.threshold(grey_scale, 100, 256,cv2.THRESH_TOZERO_INV),"To
thresh.append([cv2.threshold(grey_scale, 100, 256,cv2.THRESH_TRUNC),"Truncat
for i in thresh:
    img_disp(i[0][1],i[1])
    img_disp_plt(i[0][1],i[1])
```


Binary

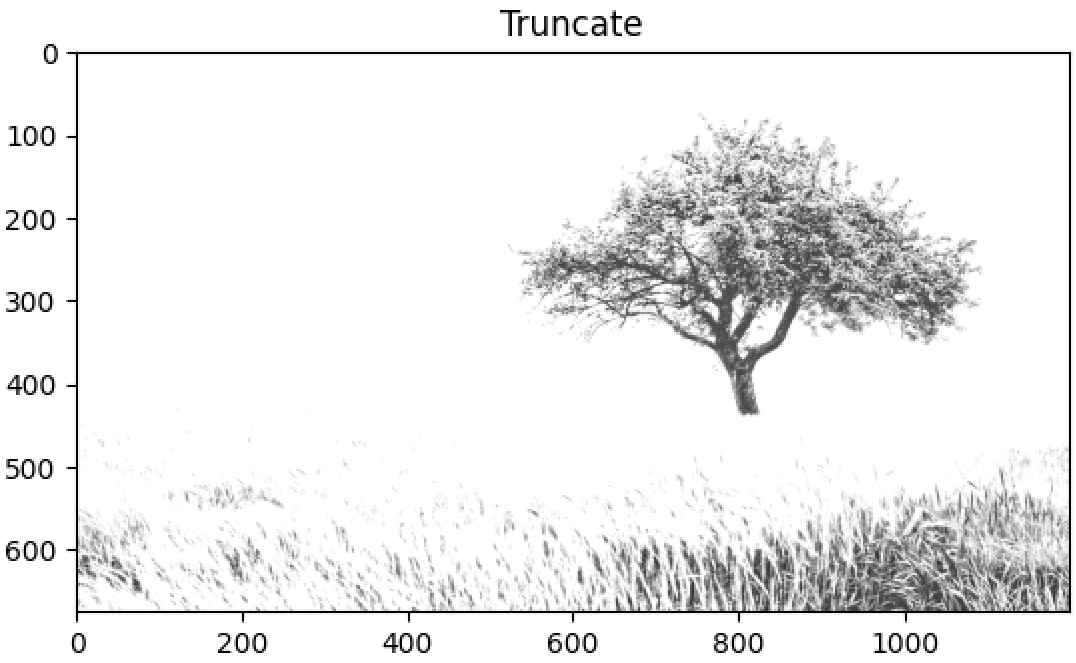

Binary Inverse



To zero

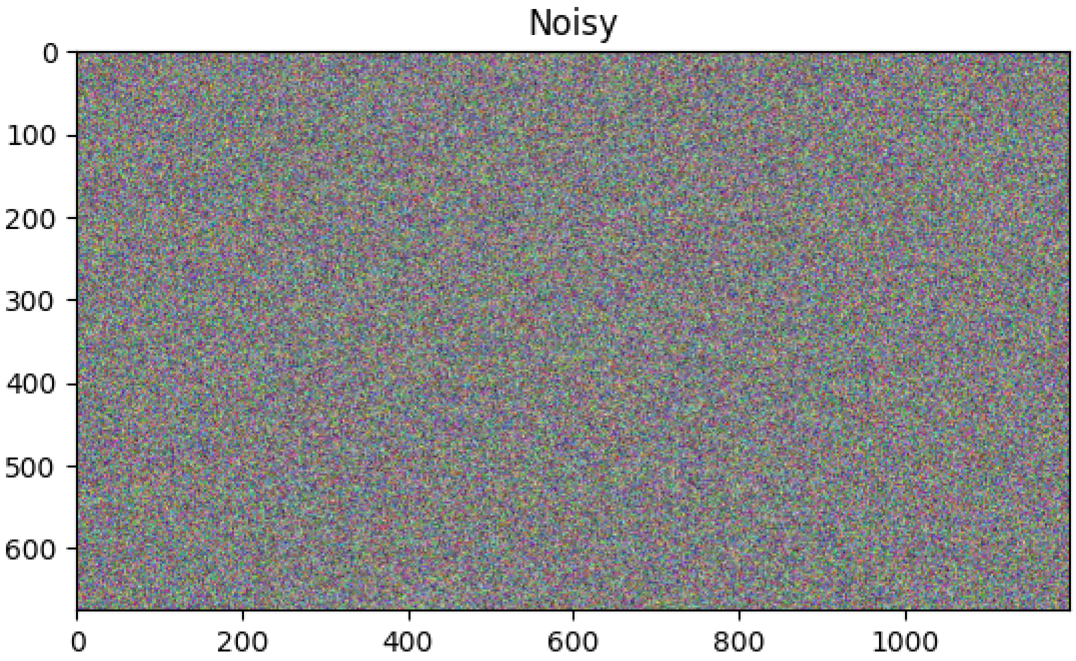


To zero inverse



In [47]:

```
# Noise
mean = (128,128,128)
sigma = (150,150,150)
noisy_img = cv2.randn(small,mean,sigma)
img_disp(noisy_img,"Noisy")
img_disp_plt(noisy_img,"Noisy")
```


Noisy
0
100
200
300
400
500
600
0
200
400
600
800
1000