In [212]:
```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
```

In [213]:
```
def read_image():
    path = "/media/rifat/STUDY/4-1/LAB/Image_Processing/image/2.jpg"
    gray = cv.imread(path,0)

    return gray
```

In [214]:
```
def OpencvHistogramEqualization(gray):
    equ = cv.equalizeHist(gray)
    return equ
```

In [215]:
```
def ManualHistogramEqualization(img):
    hist,bins = np.histogram(img.flatten(),256,[0,256])
    cdf = hist.cumsum()
    cdf_normalized = cdf * hist.max()/ cdf.max()
    cdf_m = np.ma.masked_equal(cdf,0)
    cdf_m = (cdf_m - cdf_m.min())*255/(cdf_m.max()-cdf_m.min())
    cdf = np.ma.filled(cdf_m,0).astype('uint8')
    img = cdf[img]
    return img
```

In [216]:
```
def img_show(gray,equ1,equ2):
    plt.figure(figsize=(30,20))
    plt.subplot(2,2,1)
    plt.imshow(gray,cmap = 'gray')
    plt.title("Original Image")
    plt.subplot(2,2,3)
    plt.imshow(equ1,cmap = 'gray')
    plt.title("OpencvHistogramEqualization Image")
    plt.subplot(2,2,4)
    plt.imshow(equ2,cmap = 'gray')
    plt.title("ManualHistogramEqualization Image")
    plt.show()
```

In [217]:
```
def plot_show(gray,equ1,equ2):
    plt.figure(figsize=(30,20))
    plt.subplot(2,2,1)
    plt.hist(gray.ravel(),256,[0,256])
    plt.title("Original Image")
    plt.subplot(2,2,3)
    plt.hist(equ1.ravel(),256,[0,256])
    plt.title("OpencvHistogramEqualization Image")
    plt.subplot(2,2,4)
    plt.hist(equ2.ravel(),256,[0,256]);
    plt.title("ManualHistogramEqualization Image")
    plt.show()
```

In [218]:
```
if __name__ == "__main__":
    gray = read_image()
    equ1 = OpencvHistogramEqualization(gray)
    gray = read_image()
    equ2 = ManualHistogramEqualization(gray)
    img_show(gray,equ1,equ2)
    plot_show(gray,equ1,equ2)
```