Digital Image Processing

Practicals

Q. 1) Write program to read and display digital image using MATLAB or SCILAB :

---

a. Become familiar with SCILAB/MATLAB Basic commands

b. Read and display image in SCILAB/MATLAB

c. Resize given image

d. Convert given color image into gray-scale image

e. Convert given color/gray-scale image into black & white image

f. Draw image profile

g. Separate color image in three R G & B planes

h. Create color image using R, G and B three separate planes

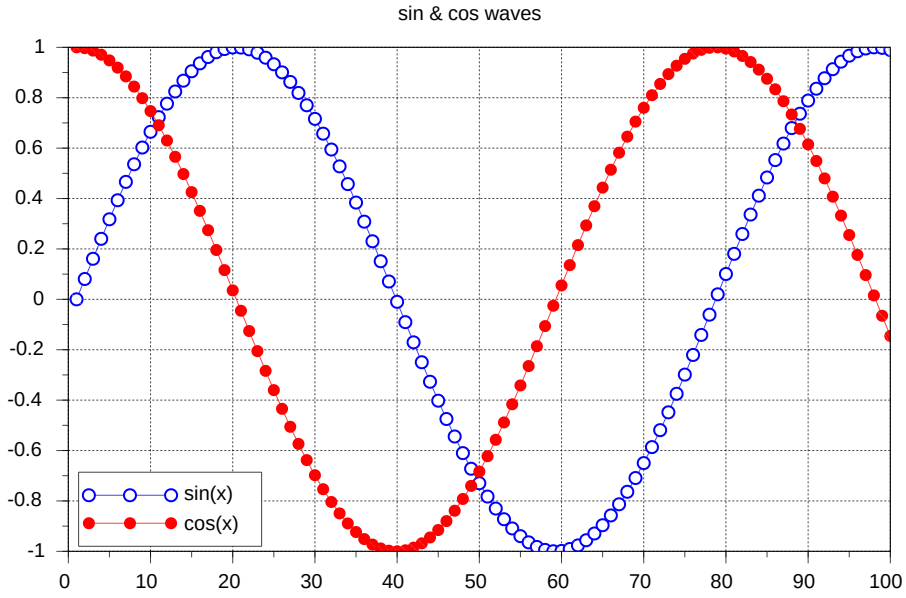
i. Flow control and LOOP in SCILAB

j. Write given 2-D data in image file

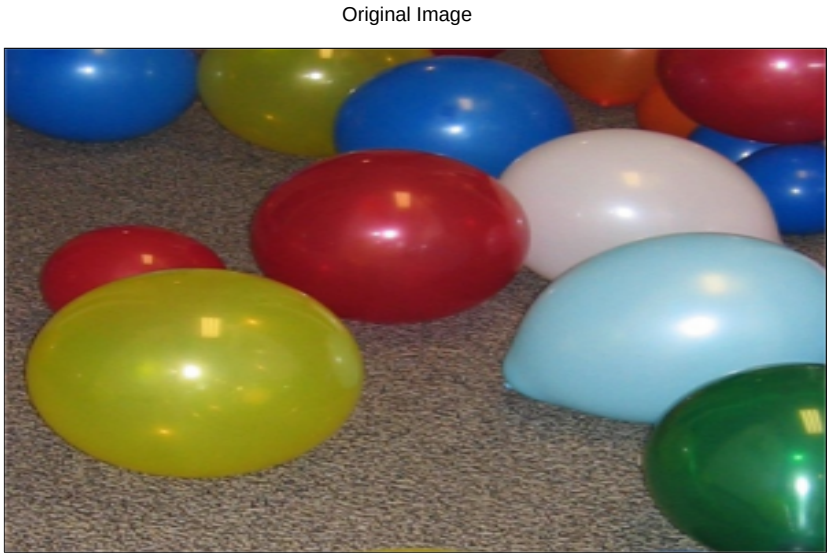
Code)

In [36]:
```
/* a. Scilab Basic Commands */
disp('2 * 3 - 4 + 8 / 3 \ 9 =', 2*3-4+8/3\9);
x = linspace(0,8,100);
plot(sin(x),'o-');
plot(cos(x),'r.-');
xtitle('sin & cos waves');
legend('sin(x)','cos(x)', 3);
xgrid(0,1,7);
```

```
"2 * 3 - 4 + 8 / 3 \ 9 ="

 5.375
```

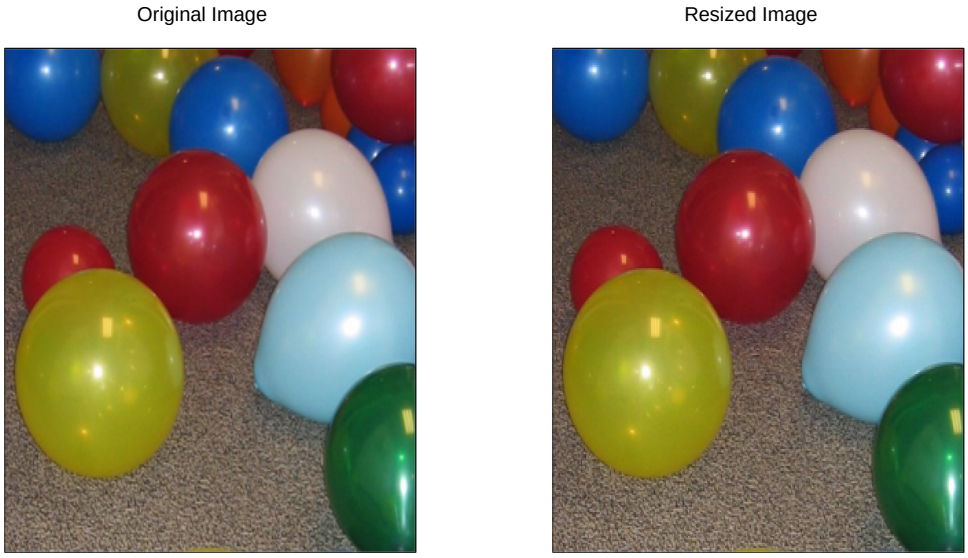


In [45]:
```
/* b. Reading & Displaying Image in Scilab */

img = imread("Test_images/balloons.png");
title('Original Image'),imshow(img);
```

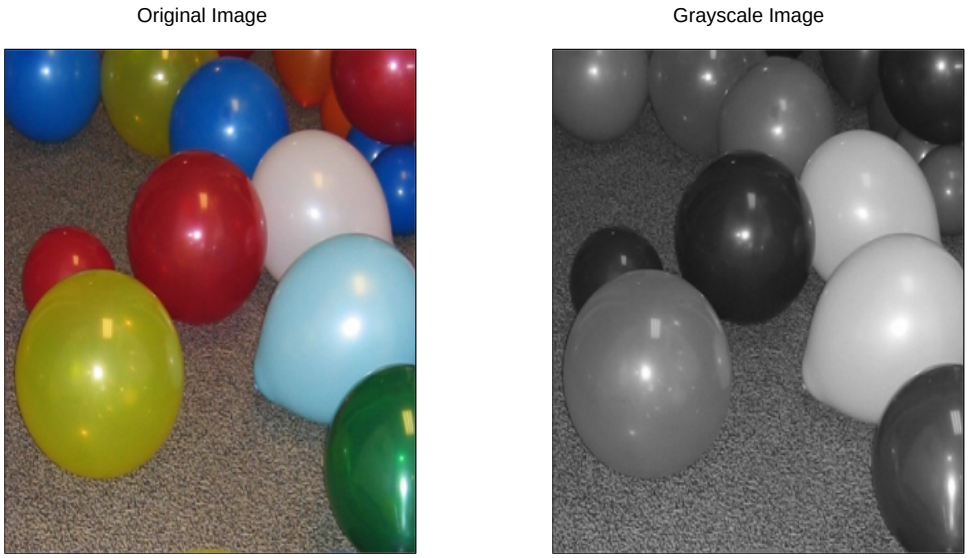


In [46]:
```
/* c. Resize given image */

img1 = imresize(img,2); // resized image

printf("Size of the image : %i %i", size(img));
printf("After Resizing the image by 2 : %i %i", size(img1));

subplot(1,2,1), title('Original Image'), imshow(img);
subplot(1,2,2), title('Resized Image'), imshow(img1);
```

```
Size of the image : 296 300
After Resizing the image by 2 : 592 600
```



In [47]:
```
/* d. RGB to Grayscale Image */

gray_img = rgb2gray(img);

subplot(1,2,1), title('Original Image'), imshow(img);
subplot(1,2,2), title('Grayscale Image'), imshow(gray_img);
```



In [48]:
```
/* e. RGB to B&W Image */
```

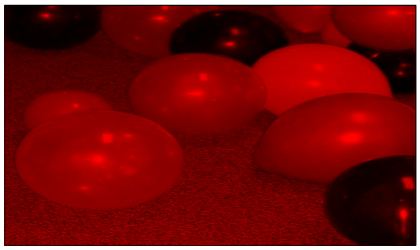
```
bw_img = im2bw(img, 0.5);

subplot(1,2,1), title('Original Image'), imshow(img);
subplot(1,2,2), title('B&W Image'), imshow(bw_img);
```

In [7]:
```
/* f. Drawing Image Profile */
//[xc, yc, pixelval] = improfile(img);
```

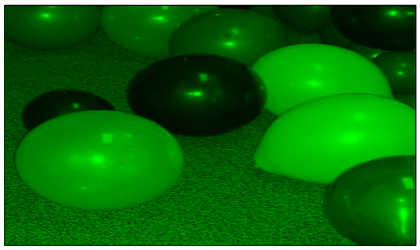
In [49]:
```
/* g. Seperating RGB Planes */

[r,c] = size(img); // no. of rows & columns of image

all_black = zeros(r,c,'uint8'); // A Black Image

// img(:,:,1) = red channel of image
red_img = cat(3, img(:,:,1), all_black, all_black);

// img(:,:,2) = green channel of image
green_img = cat(3,  all_black, img(:,:,2), all_black);

// img(:,:,3) = blue channel of image
blue_img = cat(3, all_black, all_black, img(:,:,3));

// plotting
subplot(2,2,1),title("Original Image"),imshow(img);
subplot(2,2,2),title("Red Plane Image"),imshow(red_img);
subplot(2,2,3),title("Green Plane Image"),imshow(green_img);
subplot(2,2,4),title("Blue Plane Image"),imshow(blue_img);
```

Original Image

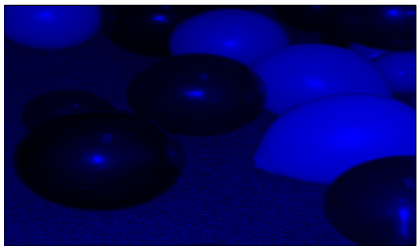
Red Plane Image



Green Plane Image



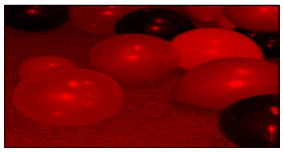
Blue Plane Image



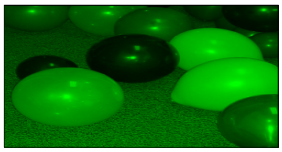
In [50]:
```
/* h.  Creating image from RGB Planes */

// RGB Image = red + green + blue channels
merged = red_img + green_img + blue_img;

subplot(3,3,2),title("Original Image"), imshow(img);
subplot(3,3,4),title("Red Plane Image"), imshow(red_img);
subplot(3,3,5),title("Green Plane Image"), imshow(green_img);
subplot(3,3,6),title("Blue Plane Image"), imshow(blue_img);
subplot(3,3,8),title("Merged RGB Planes Image"), imshow(merged);
```

Original Image

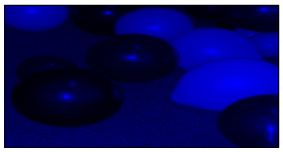
Red Plane Image



Green Plane Image



Blue Plane Image



Merged RGB Planes Image

In [10]:
```
/* i. Flow control & loop in scilab */
for i=1:10
    if modulo(i,2) == 1 then
        disp(i);
    else
```

```
        continue;
    end,
end;
```

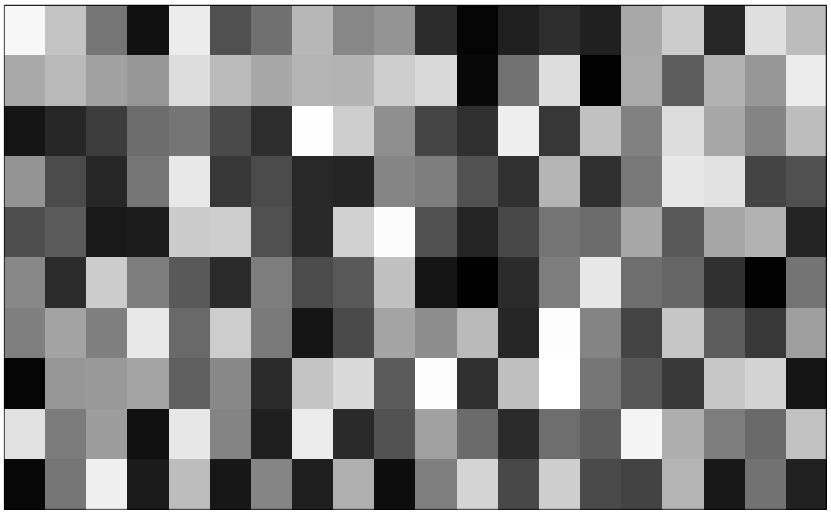
```
   1.

   3.

   5.

   7.

   9.
```

In [51]:

```
/* j. Create image with given 2D data */

mat_2d = modulo(round(rand(10,20) * (17/7) * 256),256);
disp('Given 2D data : ', mat_2d);

title('Created Image from matrix'), imshow(uint8(mat_2d));
```

```
  "Given 2D data : "

         column 1 to 11

   247.   195.   118.    17.   236.    80.   112.   183.   135.   148.    44.
   169.   185.   161.   151.   221.   187.   167.   180.   179.   205.   217.
    21.    39.    61.   109.   117.    74.    46.   254.   205.   143.    69.
   148.    75.    39.   118.   232.    55.    75.    40.    37.   133.   126.
    78.    91.    25.    28.   203.   205.    80.    41.   209.   252.    81.
   136.    44.   204.   126.    89.    43.   126.    75.    89.   192.    21.
   127.   163.   127.   232.   105.   205.   122.    21.    74.   164.   141.
     5.   150.   153.   164.    95.   136.    43.   196.   217.    91.   253.
   225.   124.   156.    17.   231.   132.    31.   235.    42.    82.   160.
     8.   118.   239.    27.   189.    23.   133.    31.   175.    13.   127.

         column 12 to 20

     5.    32.    46.    32.   168.   204.    39.   223.   188.
     8.   114.   221.     2.   170.    93.   178.   151.   235.
    48.   238.    55.   193.   129.   220.   167.   132.   189.
    82.    49.   180.    48.   120.   231.   225.    68.    80.
    37.    72.   117.   108.   167.    89.   166.   178.    36.
     1.    43.   125.   231.   110.   102.    48.     2.   116.
   185.    38.   253.   132.    68.   198.    93.    57.   159.
    48.   191.   255.   118.    87.    57.   199.   211.    20.
   107.    43.   110.    93.   244.   174.   126.   106.   194.
   212.    72.   206.    74.    67.   180.    24.   114.    33.
```

Created Image from matrix



Q. 2) To write and execute image processing programs using point processing method :

---

a. Obtain Negative image

b. Obtain Flip image

c. Thresholding

d. Contrast stretching

Code)

In [53]:
```
/* a. Obtaining Negative Image */

img = imread('Test_images/download.jpeg');

subplot(1,2,1), title('Original Image'), imshow(img);

// for negative image use '~' operator or imcomplement() method
neg_img = 255 - img;
subplot(1,2,2), title('Negative Image'), imshow(neg_img);
```

Original Image

Negative Image

In [54]:
```
/* b. Obtaining flipped images */

subplot(2,2,1), title('Original Image'), imshow(img);
subplot(2,2,3), title('Up to Down Flipped Image'), imshow(flipdim(img, 1));
subplot(2,2,4), title('Left to right Flipped Image'), imshow(flipdim(img, 2));
```

Original Image

Up to Down Flipped Image

Left to right Flipped Image

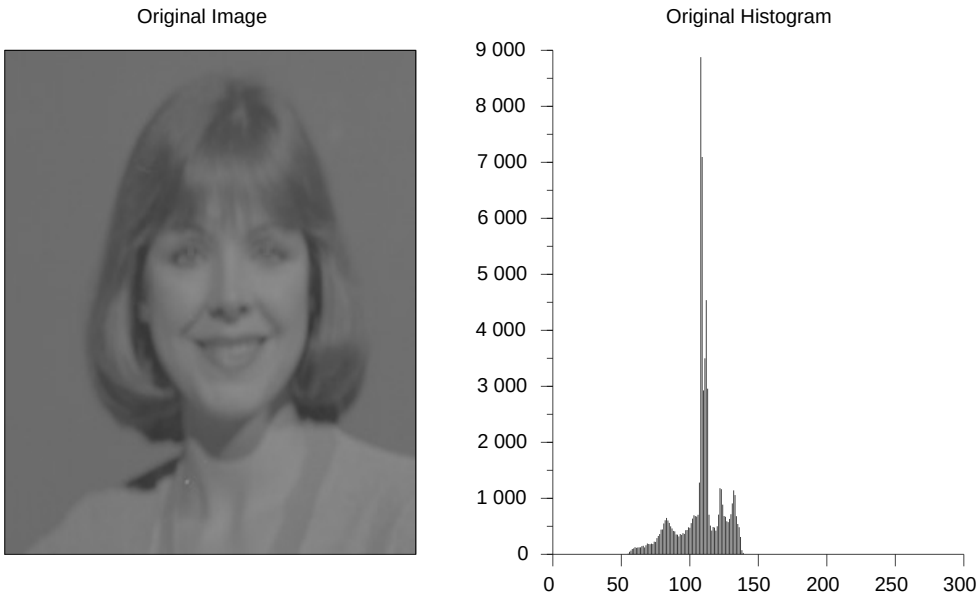
In [64]:
```
/* c. Thresholding */

function bin_img = imthreshold(gray_img, d0)
bin_img = gray_img >= d0;
endfunction;

sam_img = imread('Test_images/wom1.jpg');

subplot(1,2,1), title('Original Image'), imshow(sam_img);
subplot(1,2,2), title('Binary Image at D0 = 110'), imshow(imthreshold(sam_img,
```

Original Image

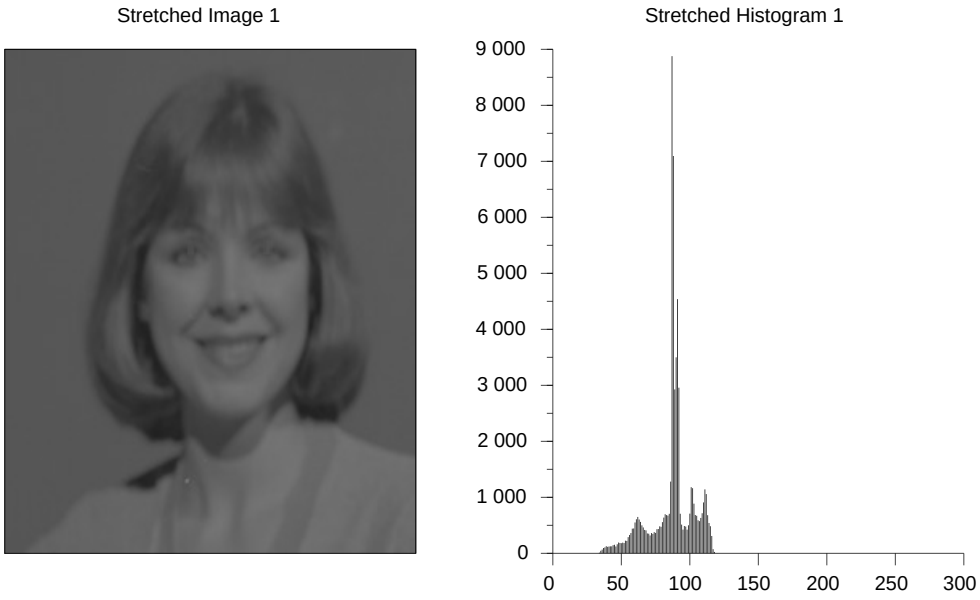
Binary Image at D0 = 110

In [57]:

```
/* d. Contrast Stretching */

function [image, image1] = contrast_stretch(gray_img)
a = min(gray_img(:));
b = max(gray_img(:));
c = perctl(double(gray_img),[5,95]); // 5 & 95 percentile of image

image = (gray_img - c(1)).*((b-a)/(c(2)-c(1))) + a;
image1 = (gray_img - a)*(255/(b-a));
endfunction;

[c_sam, c_sam1] = contrast_stretch(sam_img);

subplot(1,2,1),title("Original Image "),imshow(sam_img);
subplot(1,2,2),title("Original Histogram"),imhist(sam_img,[],1);
```



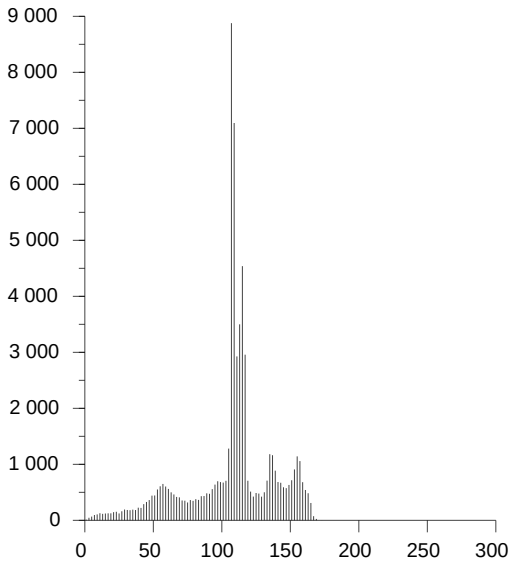
In [58]:
```
subplot(121),title("Stretched Image 1"),imshow(c_sam);
subplot(122),title("Stretched Histogram 1"), plot2d3(imhist(c_sam));
```



In [59]:
```
subplot(121),title("Stretched Image 2"),imshow(c_sam1);
subplot(122),title("Stretched Histogram 2"), plot2d3(imhist(c_sam1));
```

Stretched Image 2

Stretched Histogram 2



Q. 3) To write and execute programs for image arithmetic operations :

---

a. Addition of two images

b. Subtract one image from other image

c. Calculate mean value of image

Code)

In [18]:
```
/* Sample images */

im1 = imread('Test_images/lena_1.jpeg');
im2 = imread('Test_images/download.jpeg');

subplot(1,2,1), title('Image 1'), imshow(im1);
subplot(1,2,2), title('Image 2'), imshow(im2);
```

Image 1

Image 2

In [19]:
```
// a. use '+' between images or use imadd() method
im3 = imadd(im1, im2);

// b. use '-' between images or use imsubtract() method
im4 = im1 - im2;
im5 = im2 - im1;

subplot(1,3,1), title('Image 3 = Image 1 + Image 2'), imshow(im3);
subplot(1,3,2), title('Image 4 = Image 1 - Image 2'), imshow(im4);
subplot(1,3,3), title('Image 5 = Image 2 - Image 1'), imshow(im5);
```

Image 3 = Image 1 + Image 2

Image 4 = Image 1 - Image 2

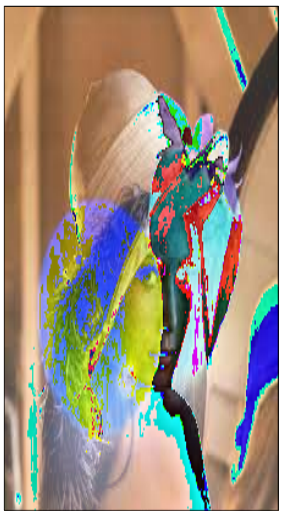
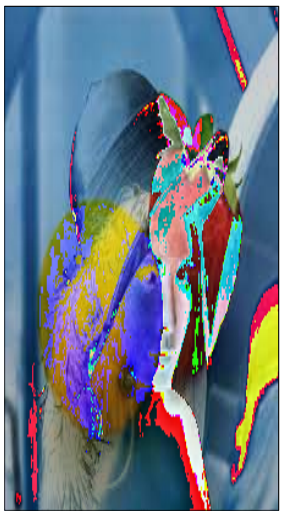


Image 5 = Image 2 - Image 1



In [20]:
```
/*c. Mean of the images */

printf('Mean of the image 1 : %.3f\n',mean(im2double(im1)))
printf('Mean of the image 2 : %.3f',mean(im2double(im2)))
```

Mean of the image 1 : 0.315

Mean of the image 2 : 0.632

Q. 4) To write and execute programs for image logical operations :

---

a. AND operation between two images

b. OR operation between two images

c. Calculate intersection of two images

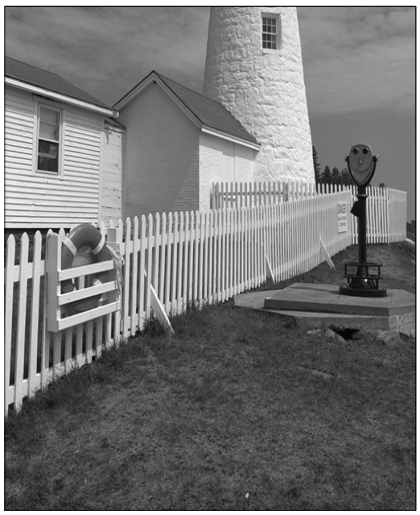
d. NOT operation (Negative image)

Code)

In [21]:
```
/* Sample images */

// for airthmetic, images have to be of same size
img1 = imread('Test_images/girlface.bmp');
img2 = imread('Test_images/lighthouse.bmp');

subplot(1,2,1), title('Image 1'), imshow(img1);
subplot(1,2,2), title('Image 2'), imshow(img2);
```

Image 1

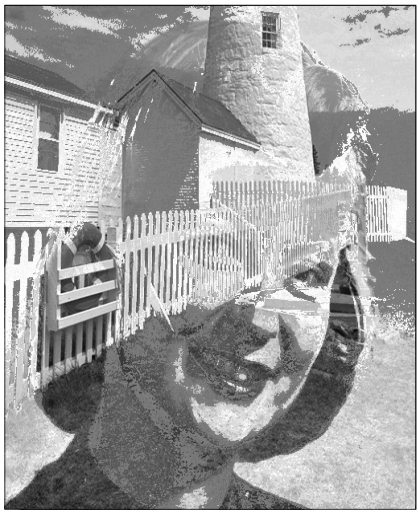
Image 2



In [22]:
```
/* a. And operation on two images */
subplot(1,2,1), title('Image 1 & Image 2'), imshow(img1 & img2);

/* b. Or operation on two images */
subplot(1,2,2), title('Image 1 | Image 2'), imshow(img1 | img2);
```

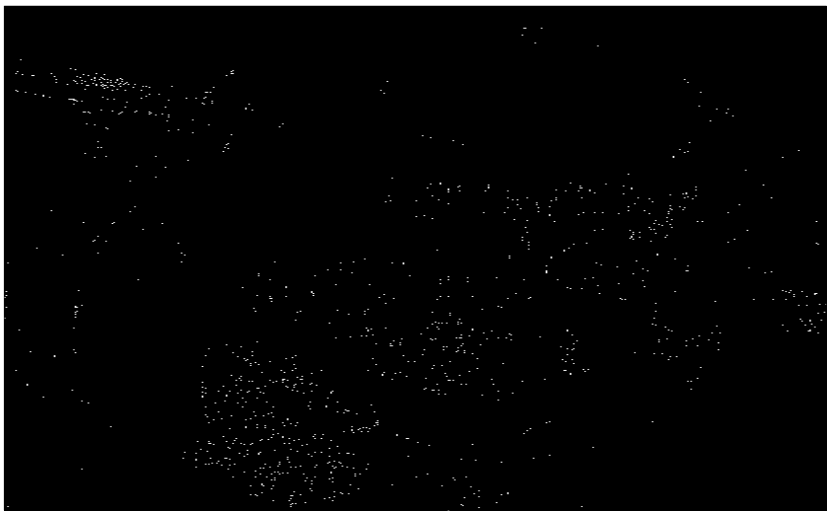
Image 1 & Image 2

Image 1 | Image 2



In [23]:
```
/* c. Intersection of two images */

inter_img = (double(img1) - double(img2)) == 0;
title('Intersection of Image 1 & Image 2'), imshow(inter_img);
```

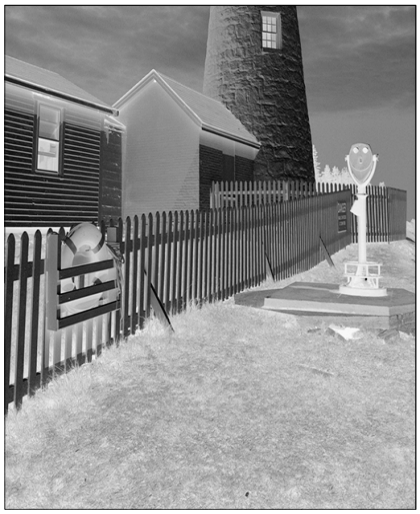
Intersection of Image 1 & Image 2



In [24]:
```
/* d. NOT on the images */

subplot(1,2,1), title('~(Image 1)'), imshow(~img1);
subplot(1,2,2), title('NOT of Image 2'), imshow(~img2);
```

~(Image 1)

NOT of Image 2



Q. 5) To write a program for histogram calculation and equalization using :

---

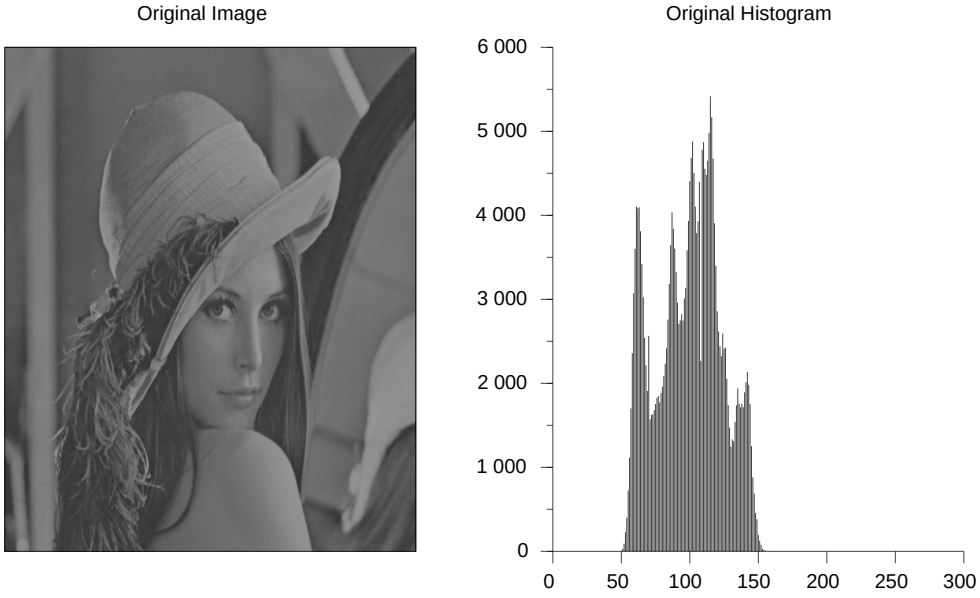
a. Standard MATLAB function

b. Program without using standard MATLAB functions

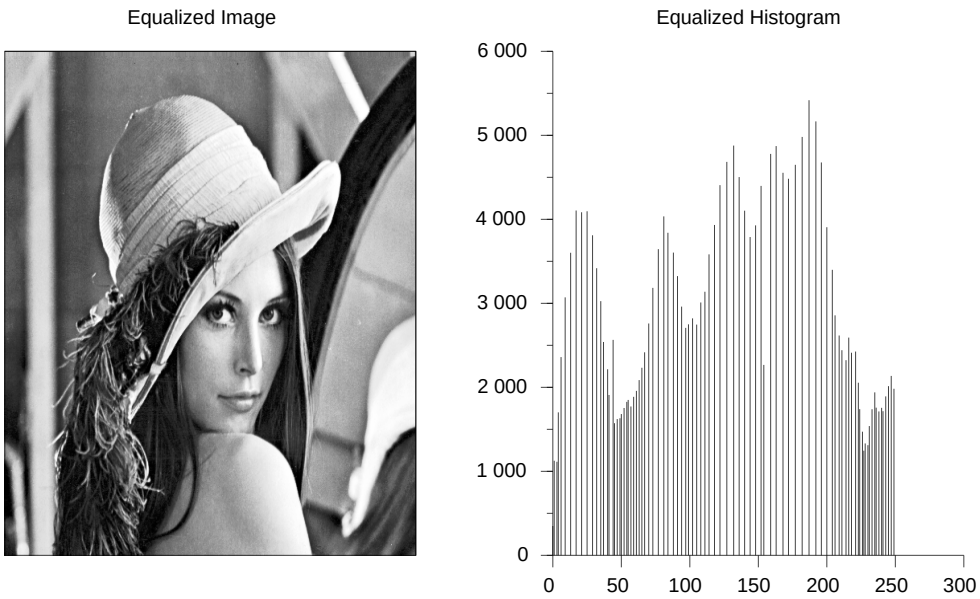
Code)

In [25]:

```
/* a. With standard Matlab function */

lena_img = imread('Test_images/Lena_dark.png');

subplot(1,2,1),title("Original Image "),imshow(lena_img);
subplot(1,2,2),title("Original Histogram"),imhist(lena_img,[],1);
```



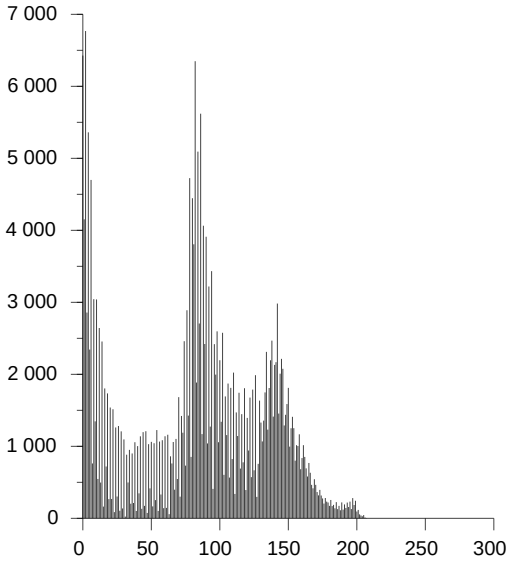
In [26]:
```
h_img = imhistequal(lena_img);

subplot(1,2,1),title("Equalized Image "),imshow(h_img);
subplot(1,2,2),title("Equalized Histogram"),imhist(h_img,[],1);
```



In [89]:
```
/* b. Without standard Matlab function */

g_img = imread('Test_images/girlface.bmp');

subplot(1,2,1),title("Original Image "),imshow(g_img);
subplot(1,2,2),title("Original Histogram"),imhist(g_img,[],1);
```

Original Image

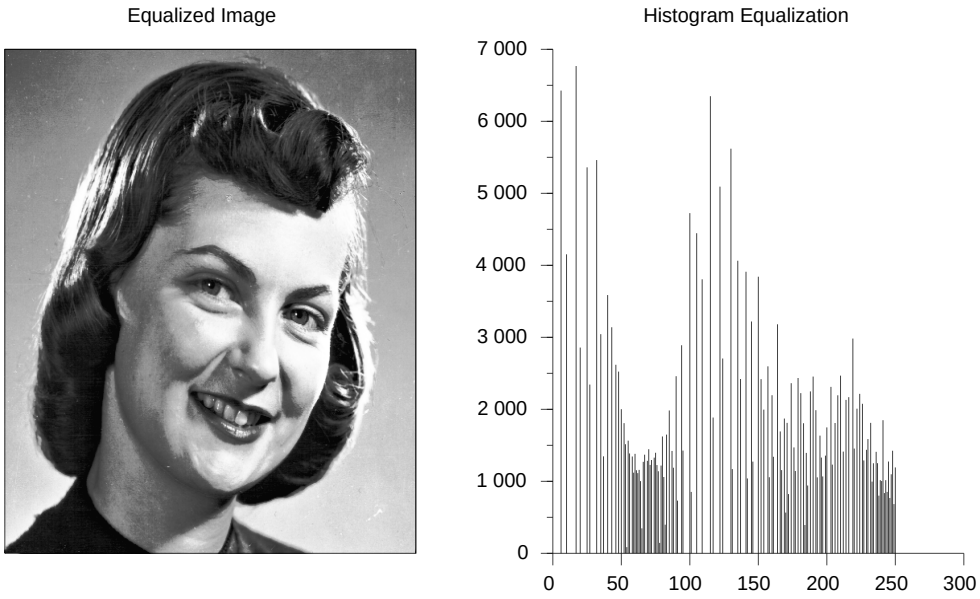
Original Histogram



In [90]:

```
function eq_img = histeq(img)

[mr,nc] = size(img);
no_pixels = mr * nc;
eq_img = zeros(mr,nc,'uint8');
freq = zeros(256,1);
probf = zeros(256,1);
cumsum = zeros(256,1);
probc = zeros(256,1);
bins = 255;
total = 0;
output = zeros(256,1);

// Counting Frequency & probability of every pixel in img

for i = 1:mr
    for j = 1:nc
        val = img(i,j) + 1;
        freq(val) = freq(val) + 1;
        probf(val) = freq(val) / no_pixels;
    end,
end;

// Cumulative Distribution Probability is calculated

for i = 1:256
    total = total + freq(i);
    cumsum(i) = total;
    probc(i) = cumsum(i) / no_pixels;
    output(i) = round(bins * probc(i));
end;

// Creating Equalized Image
for i = 1:mr
    for j = 1:nc
        eq_img(i,j) = output(img(i,j) + 1);
    end,
end;
```

```
endfunction;

he_img = histeq(g_img);
subplot(1,2,1),title("Equalized Image "),imshow(he_img);
subplot(1,2,2),title("Histogram Equalization"),imhist(he_img,[],1);
```



Q. 6) To write and execute program for geometric transformation of image :

---

a. Translation

b. Scaling

c. Rotation

d. Shrinking

e. Zooming

Code)

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Q.7) To understand various image noise models and to write programs for :

---

a. image restoration

b. Remove Salt and Pepper Noise

c. Minimize Gaussian noise

d. Median filter

Code)

In [ ]:

In [ ]:

Q. 8) Write and execute programs to use spatial low pass and high pass filters.

Code)

In [ ]:

In [ ]:

Q. 9) Write and execute programs for image frequency domain filtering :

---

a. Apply FFT on given image

b. Perform low pass and high pass filtering in frequency domain

c. Apply IFFT to reconstruct image

Code)

In [ ]:

In [ ]:

In [ ]:

Q. 10) Write a program in C and MATLAB/SCILAB for edge detection using different edge detection mask.

Code)

In [ ]:

Q. 11) Write and execute program for image morphological operations erosion and dilation.

Code)

In [ ]: