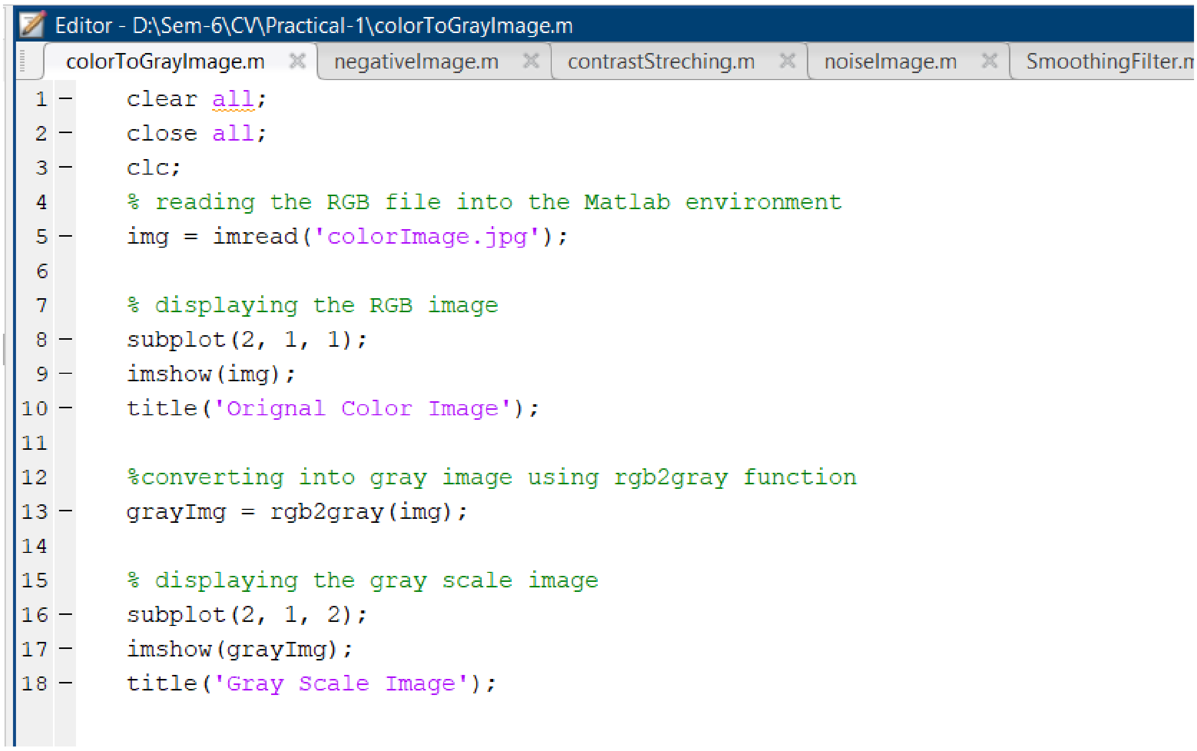
PRACTICAL: 2

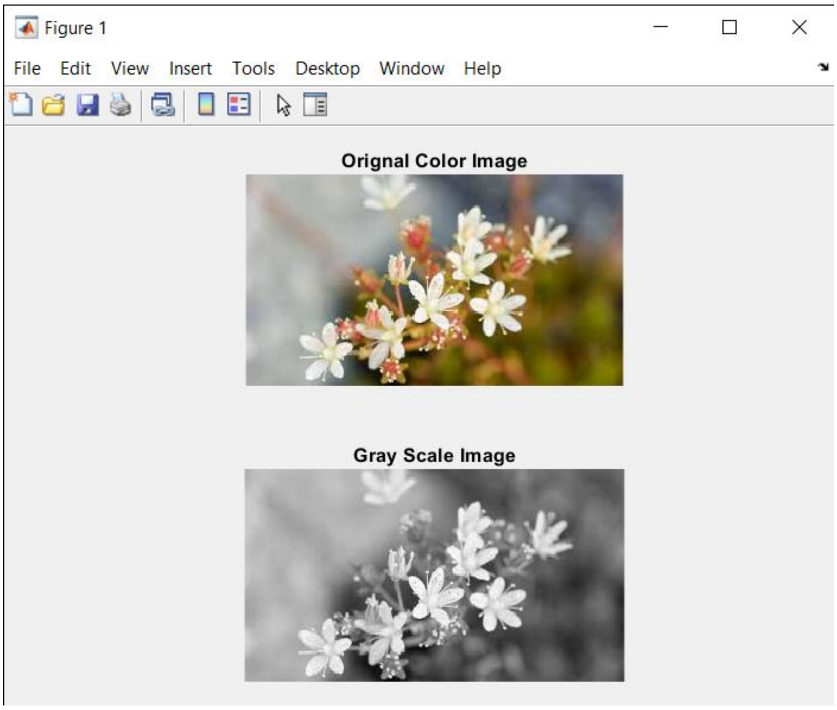
AIM-1: Read your photo. Convert it into gray scale. Inspect the impact of sampling and quantization.

CODE:



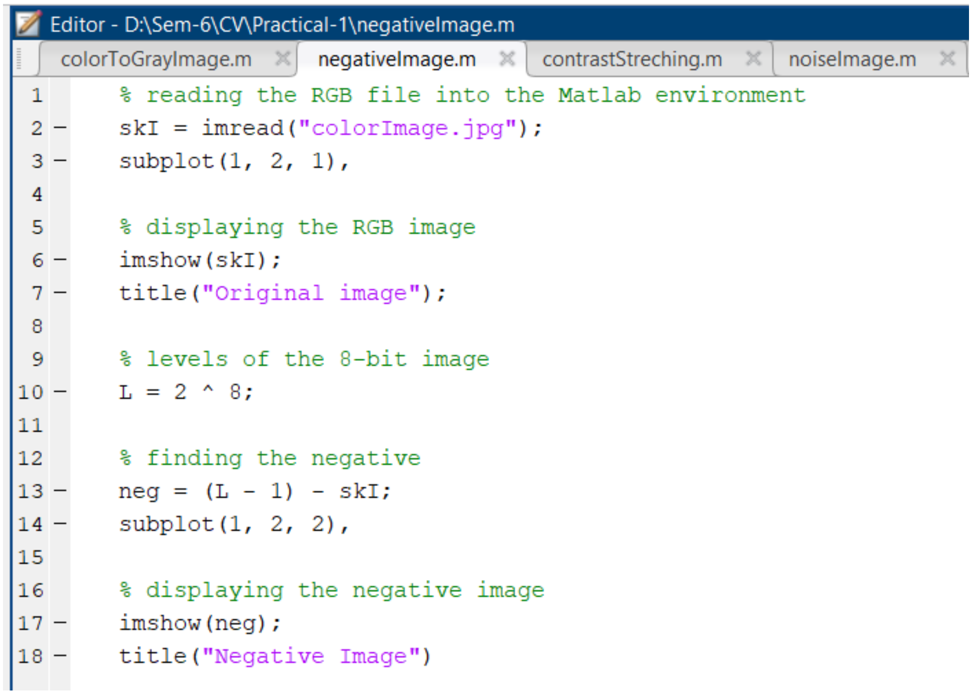
```
clear all;
close all;
clc;
% reading the RGB file into the Matlab environment
img = imread('colorImage.jpg');

% displaying the RGB image
subplot(2, 1, 1);
imshow(img);
title('Orignal Color Image');

%converting into gray image using rgb2gray function
grayImg = rgb2gray(img);

% displaying the gray scale image
subplot(2, 1, 2);
imshow(grayImg);
title('Gray Scale Image');
```

OUTPUT:



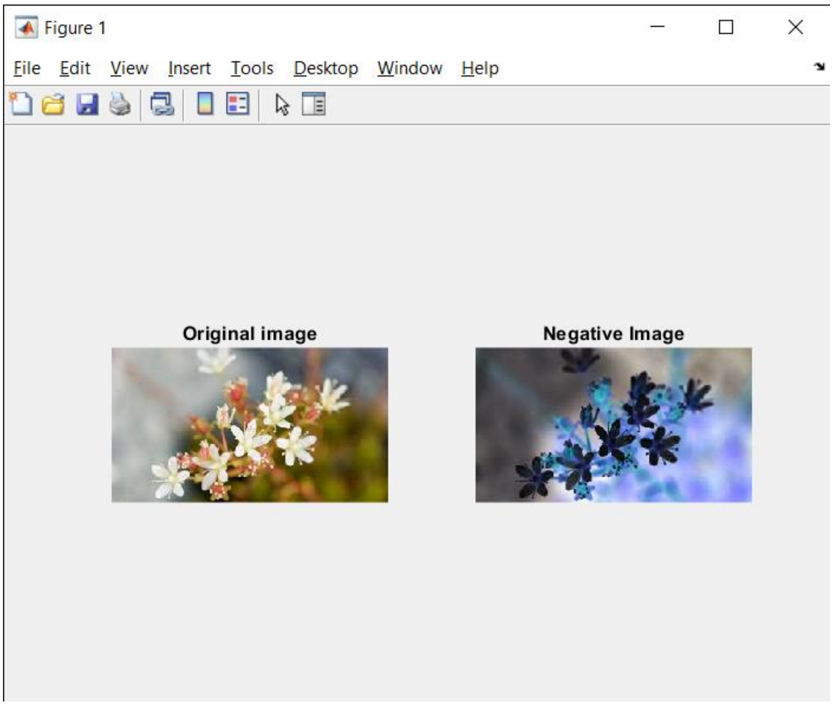
AIM-2 : Inspect the effect of image negative.

CODE:



```
% reading the RGB file into the Matlab environment
skI = imread("colorImage.jpg");
subplot(1, 2, 1),

% displaying the RGB image
imshow(skI);
title("Original image");

% levels of the 8-bit image
L = 2 ^ 8;

% finding the negative
neg = (L - 1) - skI;
subplot(1, 2, 2),

% displaying the negative image
imshow(neg);
title("Negative Image")
```

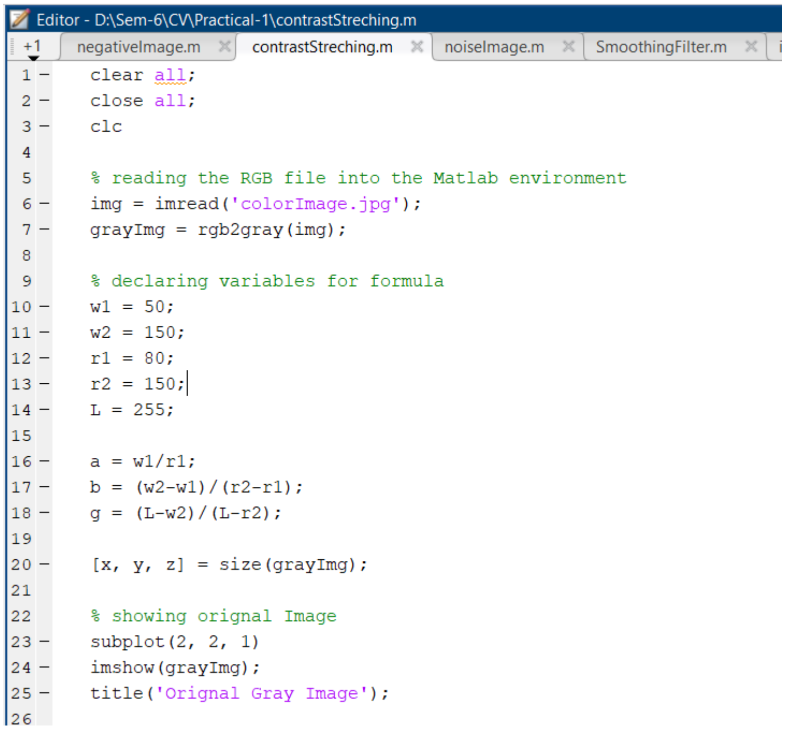
OUTPUT:



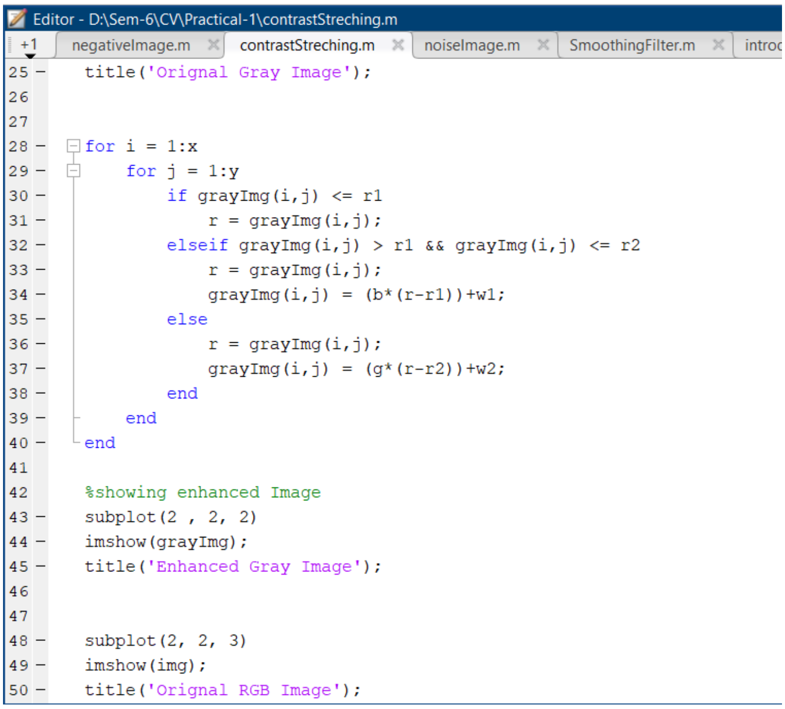
AIM-3: Perform contrast stretching and contrast threshold. Analyze the results for different values of (r1,s1) and (r2,s2).

CODE:

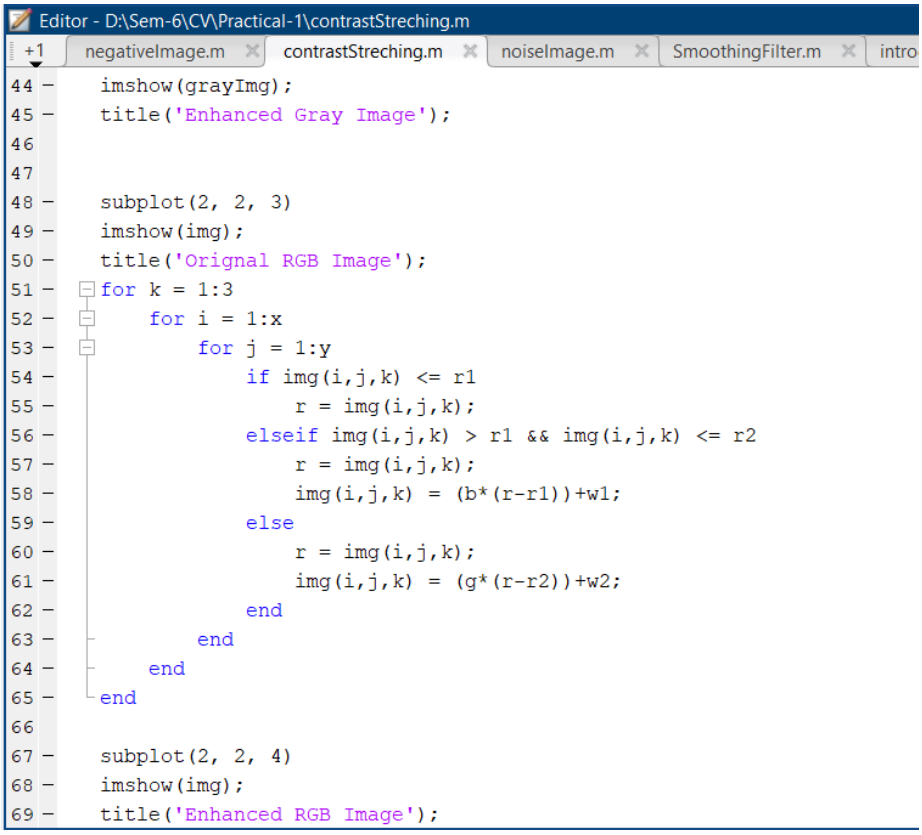
If r1 = 80 and w1 = 50:



```
clear all;
close all;
clc

% reading the RGB file into the Matlab environment
img = imread('colorImage.jpg');
grayImg = rgb2gray(img);

% declaring variables for formula
w1 = 50;
w2 = 150;
r1 = 80;
r2 = 150;
L = 255;

a = w1/r1;
b = (w2-w1)/(r2-r1);
g = (L-w2)/(L-r2);

[x, y, z] = size(grayImg);

% showing orignal Image
subplot(2, 2, 1)
imshow(grayImg);
title('Orignal Gray Image');

```

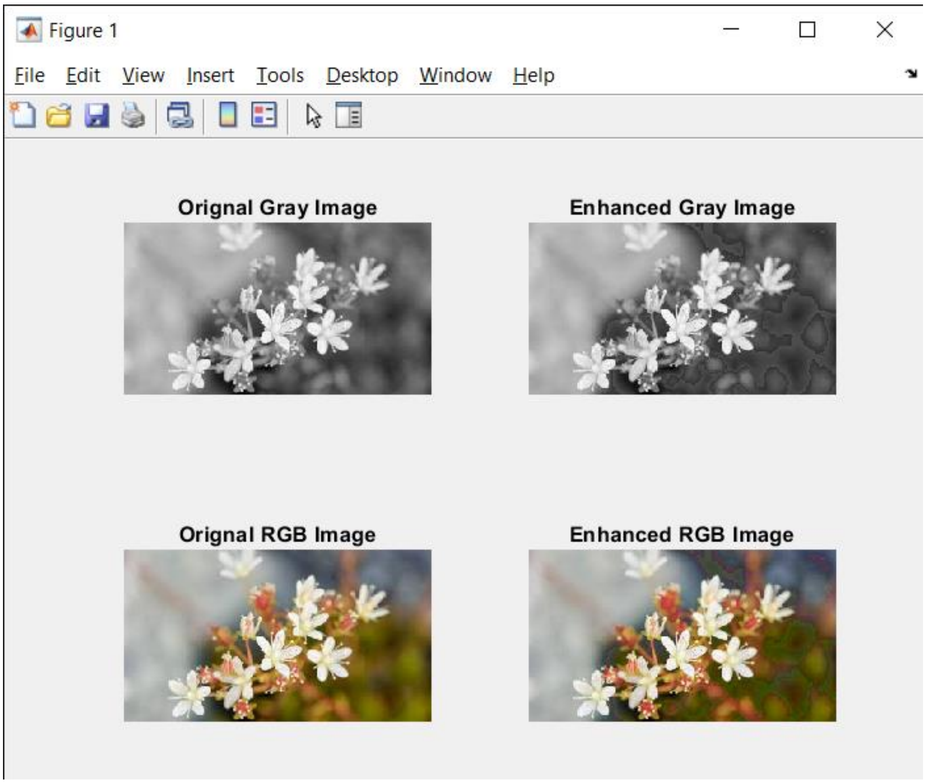


```
title('Orignal Gray Image');


for i = 1:x
    for j = 1:y
        if grayImg(i,j) <= r1
            r = grayImg(i,j);
        elseif grayImg(i,j) > r1 && grayImg(i,j) <= r2
            r = grayImg(i,j);
            grayImg(i,j) = (b*(r-r1))+w1;
        else
            r = grayImg(i,j);
            grayImg(i,j) = (g*(r-r2))+w2;
        end
    end
end

%showing enhanced Image
subplot(2 , 2, 2)
imshow(grayImg);
title('Enhanced Gray Image');


subplot(2, 2, 3)
imshow(img);
title('Orignal RGB Image');
```

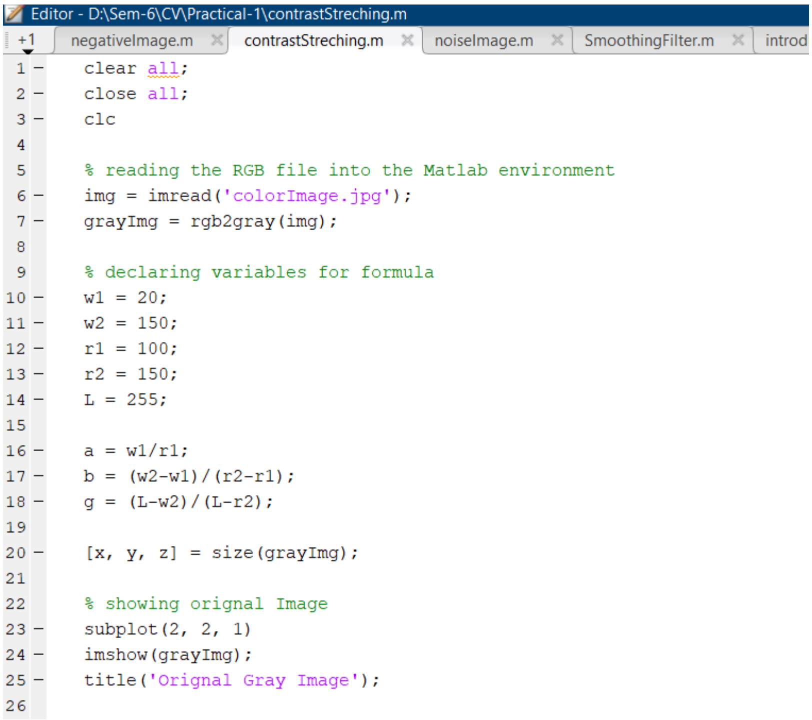


```
imshow(grayImg);
title('Enhanced Gray Image');


subplot(2, 2, 3)
imshow(img);
title('Orignal RGB Image');
for k = 1:3
    for i = 1:x
        for j = 1:y
            if img(i,j,k) <= r1
                r = img(i,j,k);
            elseif img(i,j,k) > r1 && img(i,j,k) <= r2
                r = img(i,j,k);
                img(i,j,k) = (b*(r-r1))+w1;
            else
                r = img(i,j,k);
                img(i,j,k) = (g*(r-r2))+w2;
            end
        end
    end
end

subplot(2, 2, 4)
imshow(img);
title('Enhanced RGB Image');
```

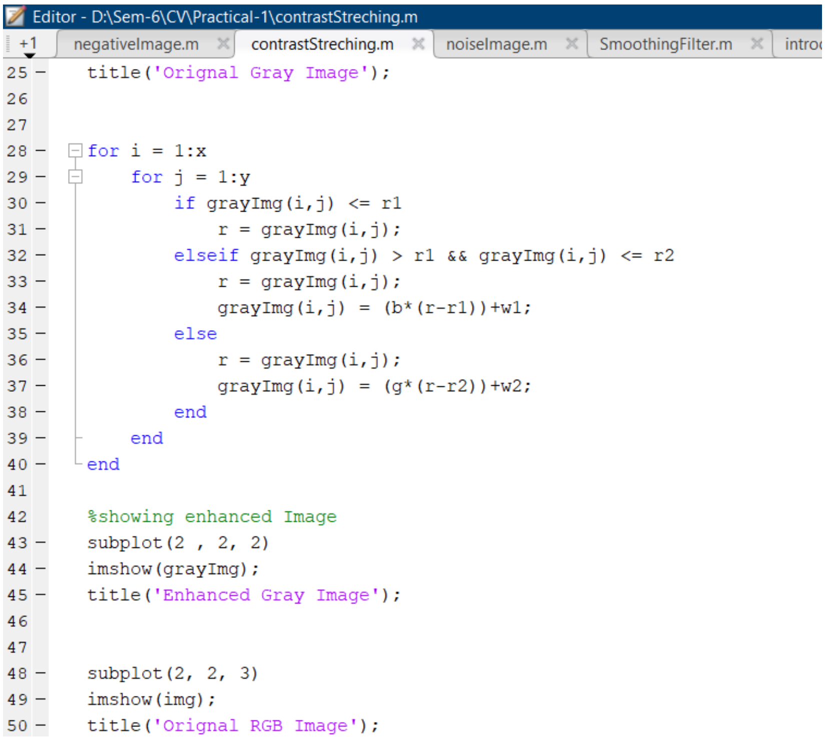
OUTPUT:



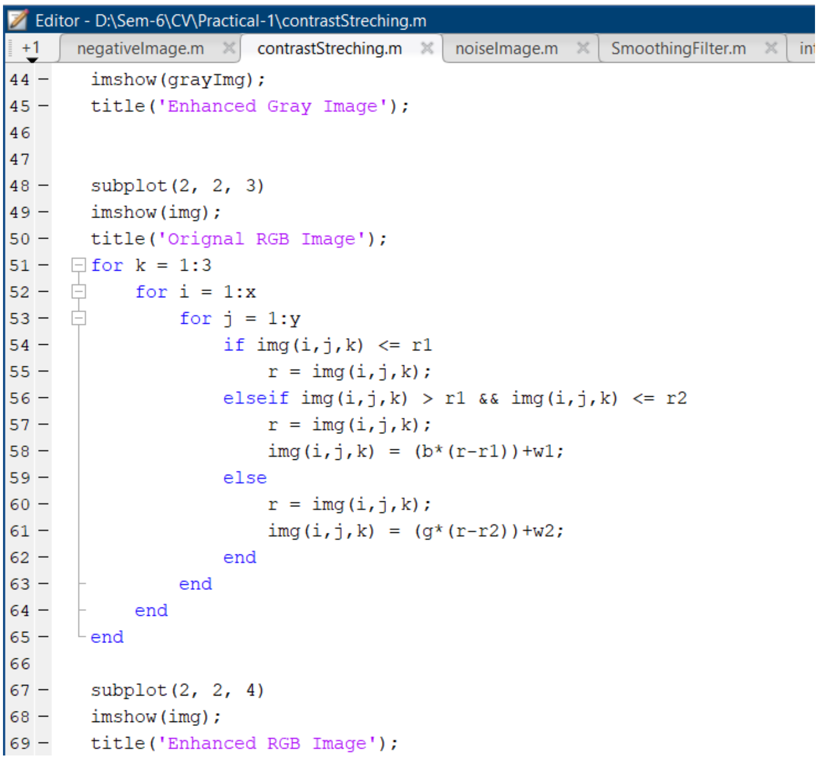
If r1= 100 and w1=20:



```matlab
clear all;
close all;
clc

% reading the RGB file into the Matlab environment
img = imread('colorImage.jpg');
grayImg = rgb2gray(img);

% declaring variables for formula
w1 = 20;
w2 = 150;
r1 = 100;
r2 = 150;
L = 255;

a = w1/r1;
b = (w2-w1)/(r2-r1);
g = (L-w2)/(L-r2);

[x, y, z] = size(grayImg);

% showing orignal Image
subplot(2, 2, 1)
imshow(grayImg);
title('Orignal Gray Image');

```

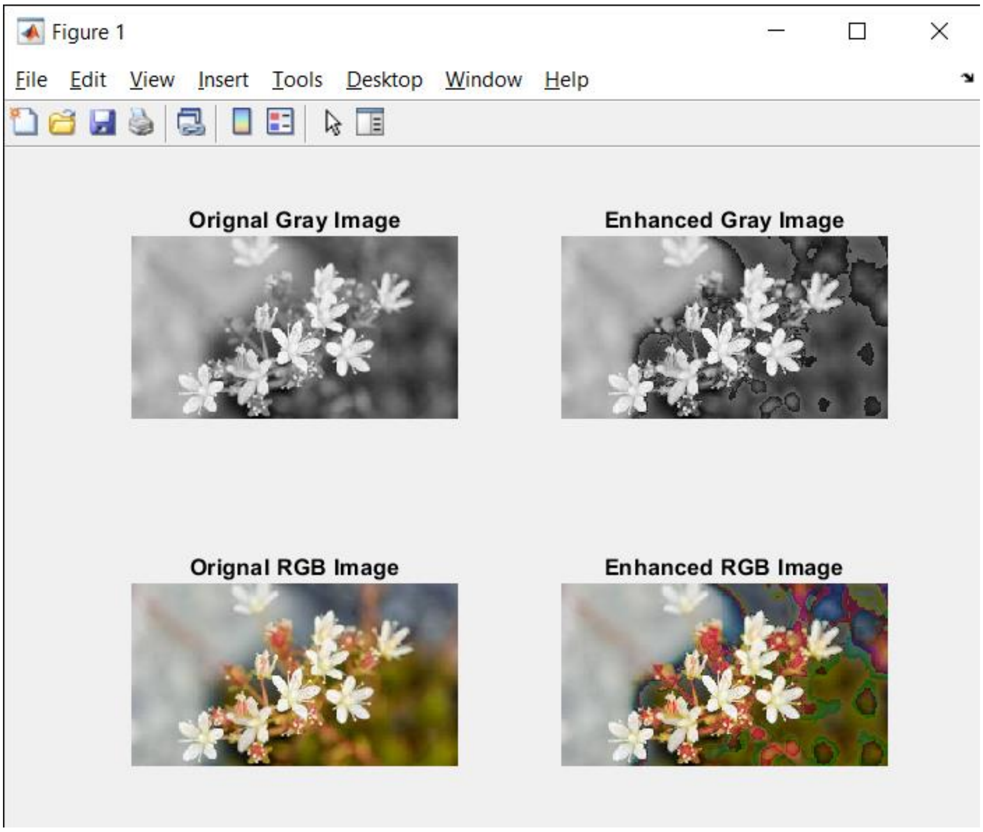


```matlab
title('Orignal Gray Image');


for i = 1:x
    for j = 1:y
        if grayImg(i,j) <= r1
            r = grayImg(i,j);
        elseif grayImg(i,j) > r1 && grayImg(i,j) <= r2
            r = grayImg(i,j);
            grayImg(i,j) = (b*(r-r1))+w1;
        else
            r = grayImg(i,j);
            grayImg(i,j) = (g*(r-r2))+w2;
        end
    end
end

%showing enhanced Image
subplot(2 , 2, 2)
imshow(grayImg);
title('Enhanced Gray Image');


subplot(2, 2, 3)
imshow(img);
title('Orignal RGB Image');
```



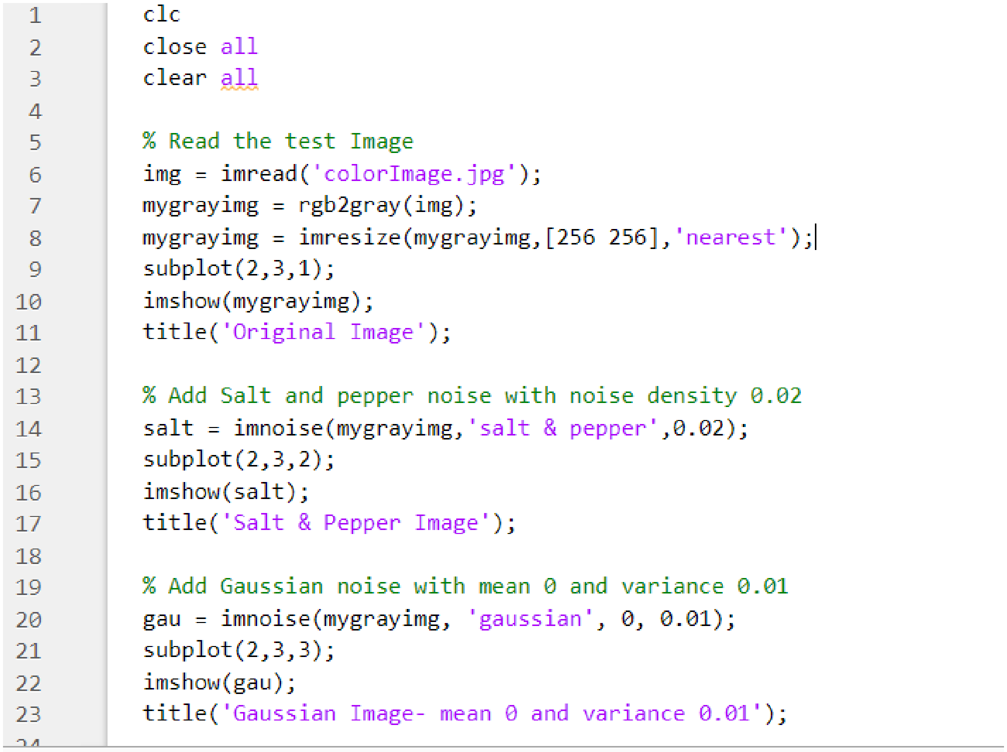
```
imshow(grayImg);
title('Enhanced Gray Image');


subplot(2, 2, 3)
imshow(img);
title('Orignal RGB Image');
for k = 1:3
    for i = 1:x
        for j = 1:y
            if img(i,j,k) <= r1
                r = img(i,j,k);
            elseif img(i,j,k) > r1 && img(i,j,k) <= r2
                r = img(i,j,k);
                img(i,j,k) = (b*(r-r1))+w1;
            else
                r = img(i,j,k);
                img(i,j,k) = (g*(r-r2))+w2;
            end
        end
    end
end

subplot(2, 2, 4)
imshow(img);
title('Enhanced RGB Image');
```

OUTPUT:

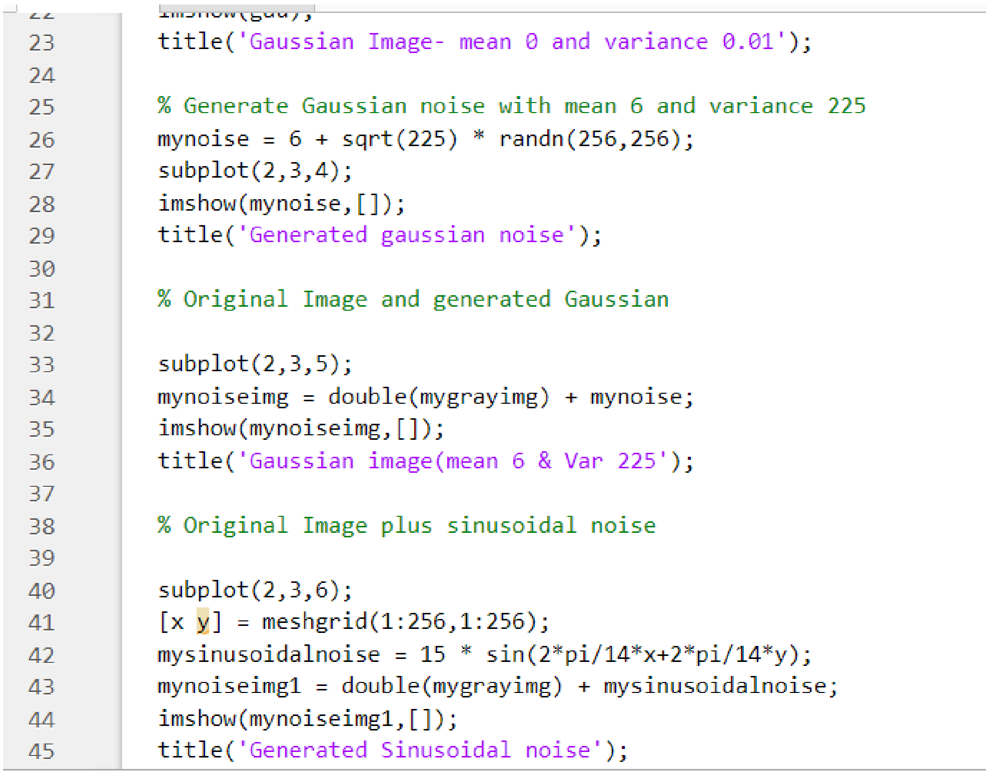


AIM-4: Add noise to your own image.

CODE:

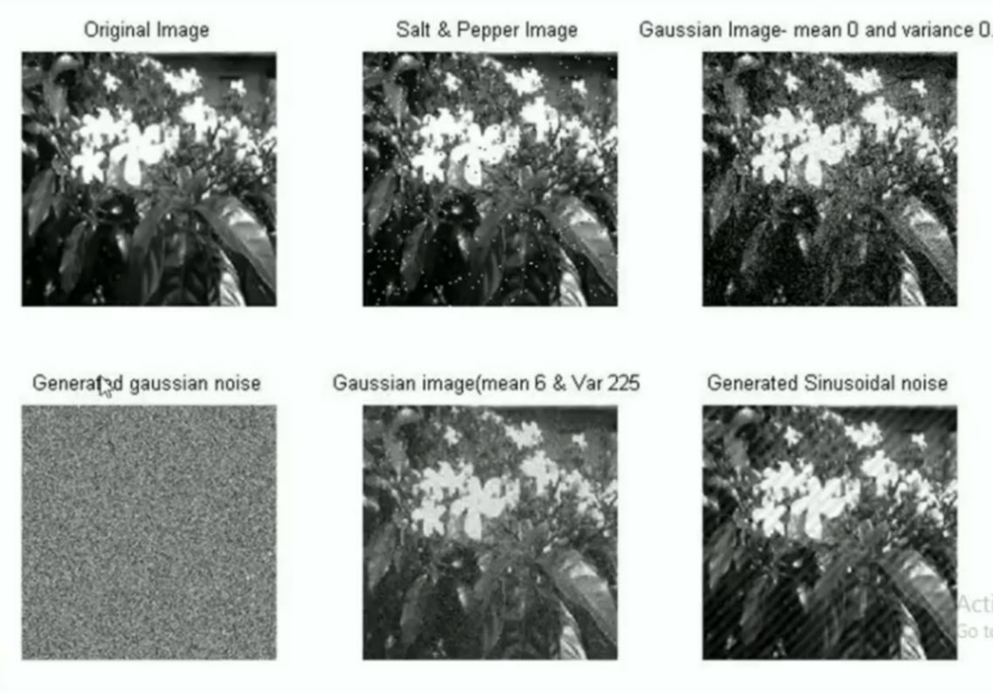


```
clc
close all
clear all

% Read the test Image
img = imread('colorImage.jpg');
mygrayimg = rgb2gray(img);
mygrayimg = imresize(mygrayimg,[256 256],'nearest');
subplot(2,3,1);
imshow(mygrayimg);
title('Original Image');

% Add Salt and pepper noise with noise density 0.02
salt = imnoise(mygrayimg,'salt & pepper',0.02);
subplot(2,3,2);
imshow(salt);
title('Salt & Pepper Image');

% Add Gaussian noise with mean 0 and variance 0.01
gau = imnoise(mygrayimg, 'gaussian', 0, 0.01);
subplot(2,3,3);
imshow(gau);
title('Gaussian Image- mean 0 and variance 0.01');
```



```
title('Gaussian Image- mean 0 and variance 0.01');

% Generate Gaussian noise with mean 6 and variance 225
mynoise = 6 + sqrt(225) * randn(256,256);
subplot(2,3,4);
imshow(mynoise,[]);
title('Generated gaussian noise');

% Original Image and generated Gaussian

subplot(2,3,5);
mynoiseimg = double(mygrayimg) + mynoise;
imshow(mynoiseimg,[]);
title('Gaussian image(mean 6 & Var 225');

% Original Image plus sinusoidal noise

subplot(2,3,6);
[x y] = meshgrid(1:256,1:256);
mysinusoidalnoise = 15 * sin(2*pi/14*x+2*pi/14*y);
mynoiseimg1 = double(mygrayimg) + mysinusoidalnoise;
imshow(mynoiseimg1,[]);
title('Generated Sinusoidal noise');
```

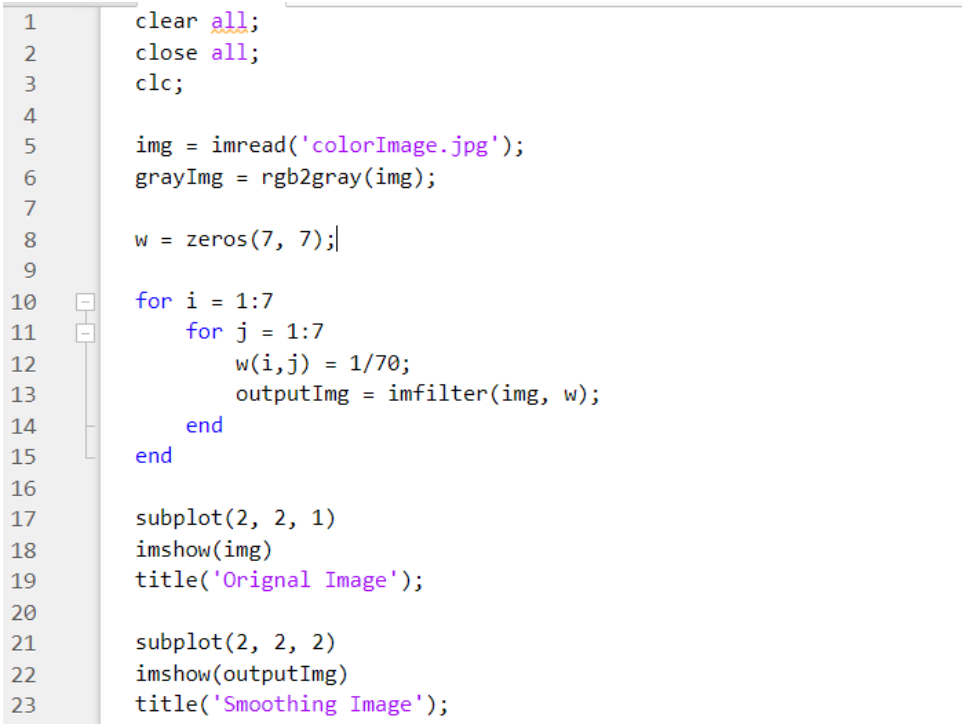
OUTPUT:


Original Image
Salt & Pepper Image
Gaussian Image- mean 0 and variance 0.
Generated gaussian noise
Gaussian image(mean 6 & Var 225
Generated Sinusoidal noise

AIM-5: Implement different spatial domain smoothening filters with different mask size. Analyze the result.

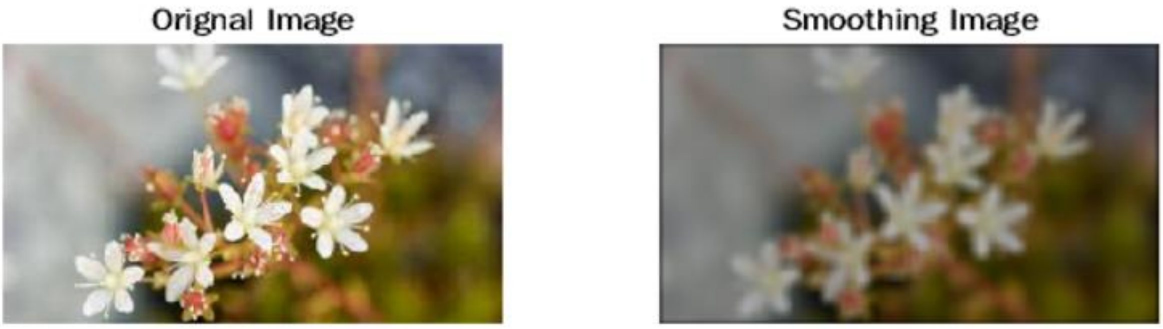
CODE:

If We apply 7, 7 mask size:



```
clear all;
close all;
clc;

img = imread('colorImage.jpg');
grayImg = rgb2gray(img);

w = zeros(7, 7);

for i = 1:7
    for j = 1:7
        w(i,j) = 1/70;
        outputImg = imfilter(img, w);
    end
end

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

subplot(2, 2, 2)
imshow(outputImg)
title('Smoothing Image');
```

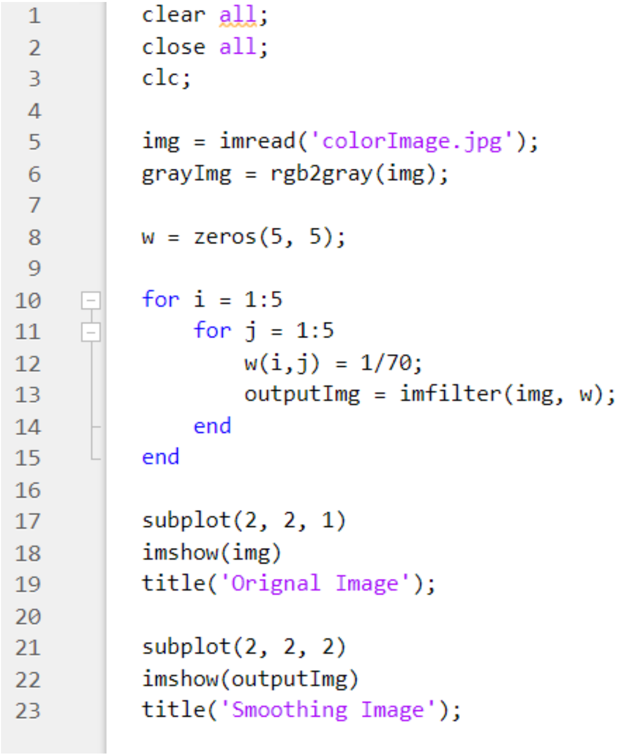
OUTPUT:



Orignal Image

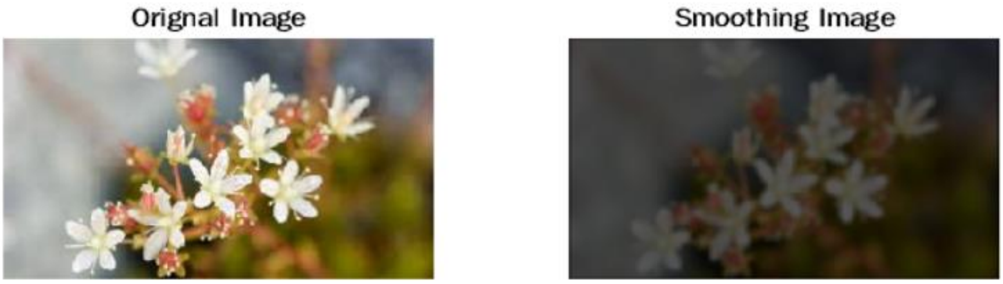
Smoothing Image

If we apply 5, 5 mask size:



```
clear all;
close all;
clc;

img = imread('colorImage.jpg');
grayImg = rgb2gray(img);

w = zeros(5, 5);

for i = 1:5
    for j = 1:5
        w(i,j) = 1/70;
        outputImg = imfilter(img, w);
    end
end

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

subplot(2, 2, 2)
imshow(outputImg)
title('Smoothing Image');
```

OUTPUT:



LEARNING OUTCOME:

In this practical I learned about how we can convert image into gray scale, image into negative, image starching, image thresholding, image noises and image smoothing.