**Program**

**Ques.** Write a MATLAB Program to mirror the input image.

**Code**

x = imread('./cameraman.tif');

y = imresize(x,[256,256]);

[m,n] = size(y);

s = y;

for i = 1 : m

k = n;

for j = 1 : n

s(i,j) = y(i,k);

k = k - 1;

end

end

subplot(1,2,1);

imshow(y,[]);

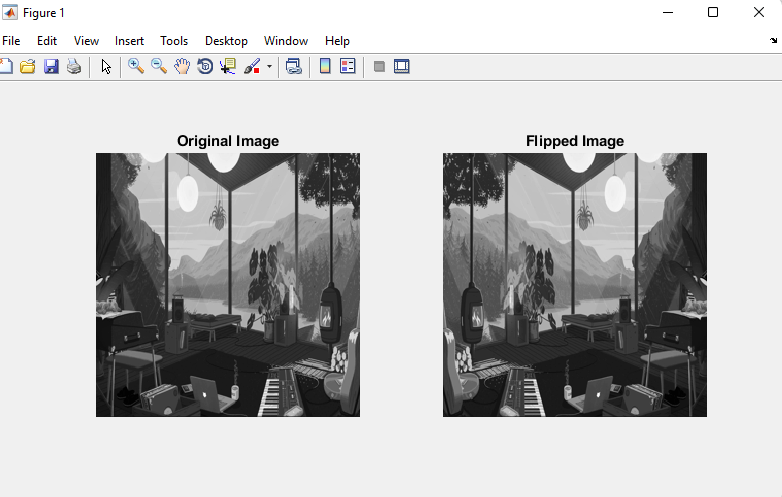
title("Original Image");

subplot(1,2,2);

imshow(s,[]);

title("Flipped Image");

**Output**



**Program**

**Ques.** Perform the below operations over the two different input images.

**(a). Addition**

**Code**

x = imread('./brain\_398.tif');

y = imread('./brain\_492.tif');

x = rgb2gray(x);

y = rgb2gray(y);

x = imresize(x,[256,256]);

y = imresize(y,[256,256]);

[m,n] = size(x);

for i = 1 : m

for j = 1 : n

s(i,j) = y(i,j) + x(i,j);

end

end

subplot(2,2,1)

imshow(x);

title("Image 1");

subplot(2,2,2)

imshow(y);

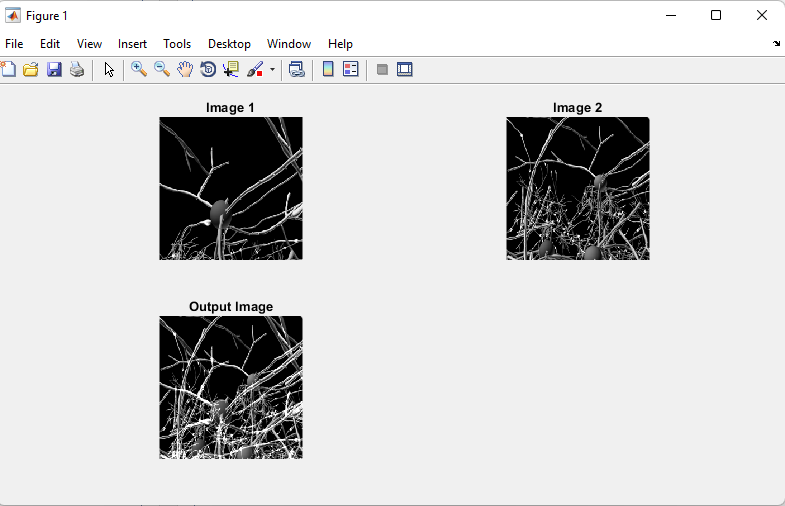
title("Image 2");

subplot(2,2,3)

imshow(s);

title("Output Image");

**Output**

****

**(b). Subtraction**

**Code**

x = imread('./brain\_398.tif');

y = imread('./brain\_492.tif');

x = rgb2gray(x);

y = rgb2gray(y);

x = imresize(x,[256,256]);

y = imresize(y,[256,256]);

[m,n] = size(x);

for i = 1 : m

for j = 1 : n

s(i,j) = y(i,j) - x(i,j);

end

end

subplot(2,2,1)

imshow(x);

title("Image 1");

subplot(2,2,2)

imshow(y);

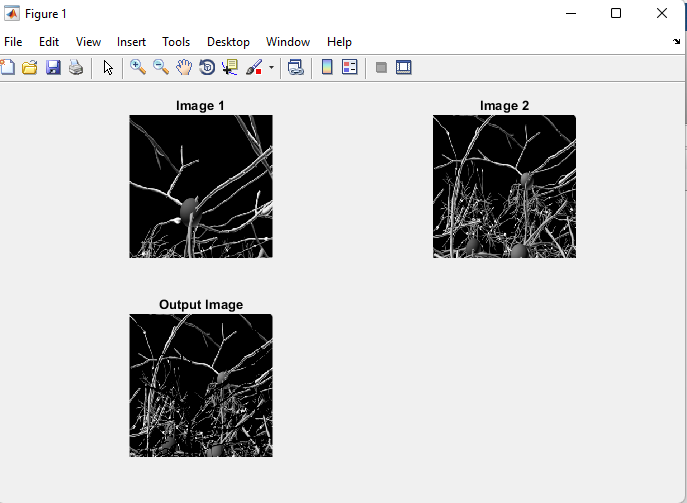
title("Image 2");

subplot(2,2,3)

imshow(s);

title("Output Image");

**Output**



**(c). Average**

**Code**

x = imread('./brain\_398.tif');

y = imread('./brain\_492.tif');

x = rgb2gray(x);

y = rgb2gray(y);

x = imresize(x,[256,256]);

y = imresize(y,[256,256]);

[m,n] = size(x);

for i = 1 : m

for j = 1 : n

s(i,j) = (y(i,j) - x(i,j))/2;

end

end

subplot(2,2,1)

imshow(x);

title("Image 1");

subplot(2,2,2)

imshow(y);

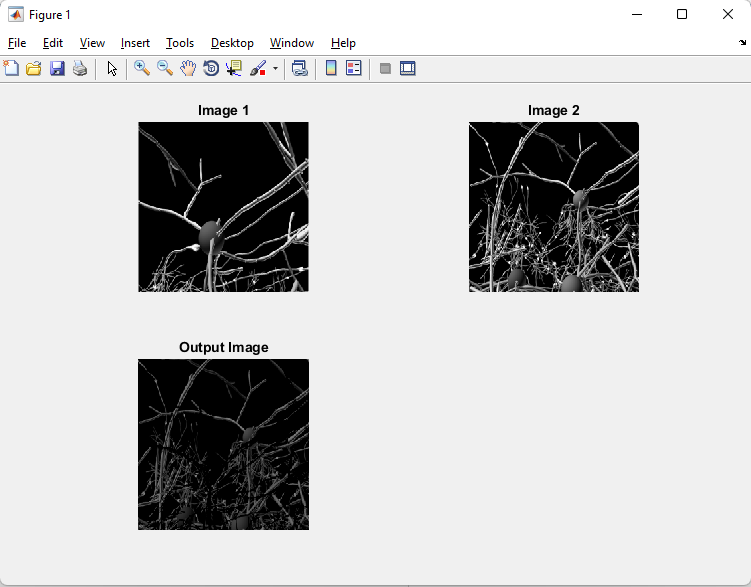
title("Image 2");

subplot(2,2,3)

imshow(s);

title("Output Image");

**Output**



**(d). AND/OR**

**Code**

x = imread('./brain\_398.tif');

y = imread('./brain\_492.tif');

x = rgb2gray(x);

y = rgb2gray(y);

x = imresize(x,[256,256]);

y = imresize(y,[256,256]);

[m,n] = size(x);

for i = 1 : m

for j = 1 : n

s(i,j) = y(i,j) & x(i,j);

end

end

subplot(2,2,1)

imshow(x);

title("Image 1");

subplot(2,2,2)

imshow(y);

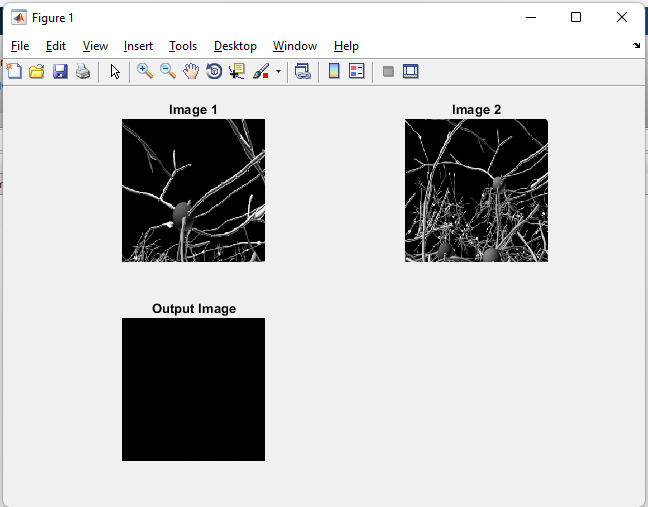
title("Image 2");

subplot(2,2,3)

imshow(s);

title("Output Image");

**Output**



**Program**

**Ques.** Write a MATLAB program to generate salt and pepper noisy image and perform median filtering over the noisy image.

**Code**

x =imread('./test.png');

x = imresize(x,[256,256]);

subplot(1,4,1);

imshow(x);

title("original image");

x1d=rgb2gray(x);

subplot(1,4,2);

imshow(x1d);

title("gray image");

% adding salt and pepper noise in the images

testimage=x1d;

xld = double(testimage);

noisex=imnoise(xld,'salt & pepper',0.02);

subplot(1,4,3);

imshow(noisex,[]);

title("noisy image");

%applying medianfilter directly to original image

output=noisex;

[row,col]=size(noisex);

for x=2:row-1

for y=2:col-1

mask=[noisex(x-1,y-1),noisex(x-1,y),noisex(x-1,y+1),noisex(x,y-1),noisex(x,y),noisex(x,y+1),noisex(x+1,y-1),noisex(x+1,y),noisex(x+1,y+1) ];

masksorted=sort(mask);

mid=masksorted(5);

output(x,y)=mid;

end

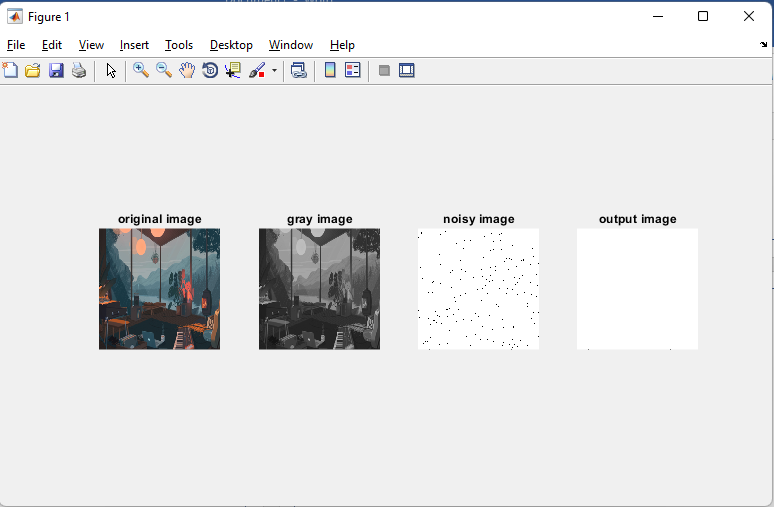
end

subplot(1,4,4);

imshow(output);

title("output image");

**Output**



**Program**

**Ques.** Write a MATLAB program to perform image fusion approach over two different image in spatial domain and also using DWT.

**Code**

**(a).** **Spatial Domain**

x = imread('./brain\_398.tif');

y = imread('./brain\_492.tif');

x = rgb2gray(x);

y = rgb2gray(y);

a1 = double(x);

a2 = double(y);

a1 = imresize(a1,[256,256]);

a2 = imresize(a2,[256,256]);

[m,n] = size(a1);

for i = 1 : m

for j = 1 : n

s(i,j) = a1(i,j) + a2(i,j)/2;

end

end

subplot(1,3,1);

image(a1);

title("Image 1");

subplot(1,3,2);

image(a2);

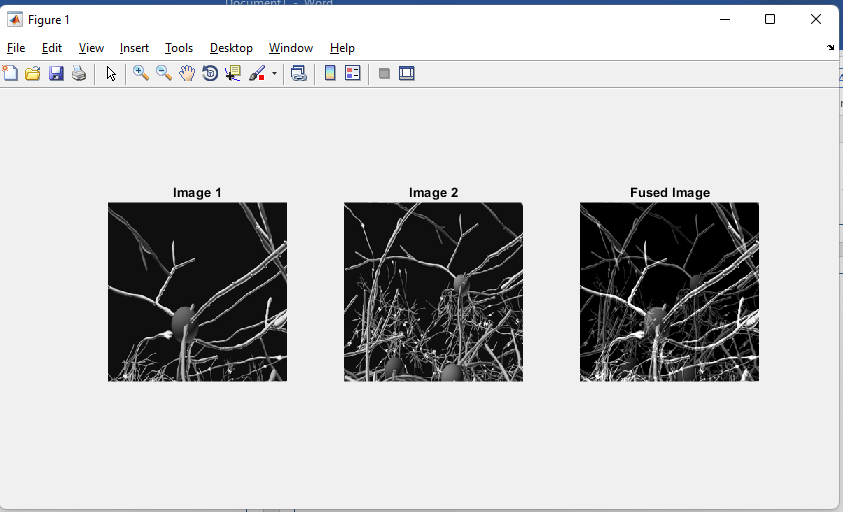
title("Image 2");

subplot(1,3,3);

image(s);

title("Fused Image");

**Output**



**Program**

**Ques.** Write a MATLAB program to perform Power law and Log Transformation on the input image.

**Code**

x = imread('test.png');

x = rgb2gray(x);

x = double(x);

x = imresize(x,[256,256]);

c = 1;

g = 0.5;

[m,n] = size(x);

for i = 1 : m

for j = 1 : n

s1(i,j) = c\*log(1+x(i,j));

end

end

for i = 1 : m

for j = 1 : n

s2(i,j) = c\*(x(i,j)^g);

end

end

subplot(2,2,1)

imshow(x,[]);

title("Original Image");

subplot(2,2,2)

imshow(s1,[]);

title("Log Transformation");

subplot(2,2,3)

imshow(x,[]);

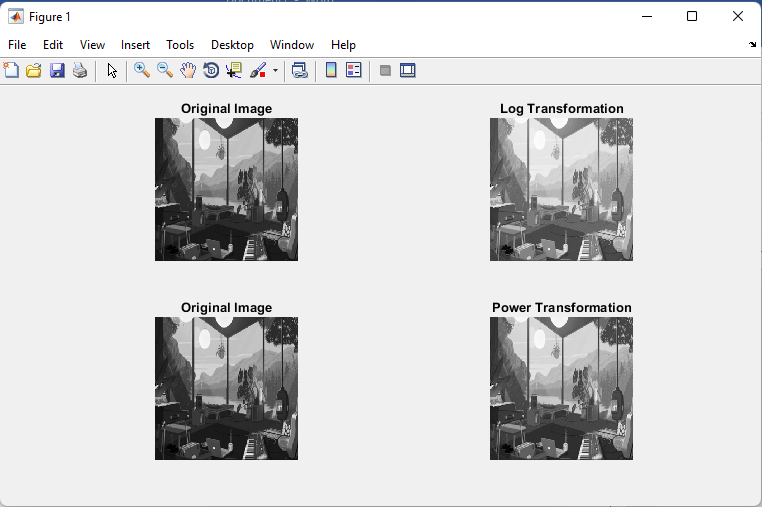
title("Original Image");

subplot(2,2,4)

imshow(s2,[]);

title("Power Transformation");

**Output**



**Program**

**Ques.** Write a MATLAB program to produce the negative image.

**Code**

x = imread('test.png');

x = rgb2gray(x);

x = imresize(x,[256,256]);

[m,n] = size (x)

for i = 1 : m

for j = 1 : n

y(i,j) = 255 - x(i,j);

end

end

subplot(1,2,1)

title("original image");

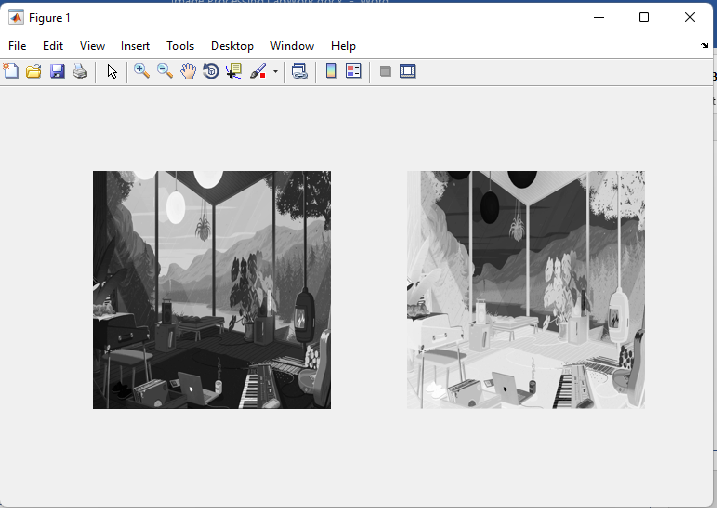
imshow(x,[]);

subplot(1,2,2);

title("new image");

imshow(y,[]);

**Output**



**Program**

**Ques.** Write a MATLAB program to use Prewitt, Sobel and Robert filter on input image.

**Code**

**(a). Prewitt**

x = imread('test.png');

x = rgb2gray(x);

x = imresize(x,[256,256]);

x = double(x)

p\_msk = [-1 0 1; -1 0 1; -1 0 1];

kx = conv2(x,p\_msk,'same');

ky = conv2(x,p\_msk,'same');

ked = sqrt(kx.^2 + ky.^2);

subplot(2,2,1);

imshow(x,[]);

subplot(2,2,2);

imshow(kx,[]);

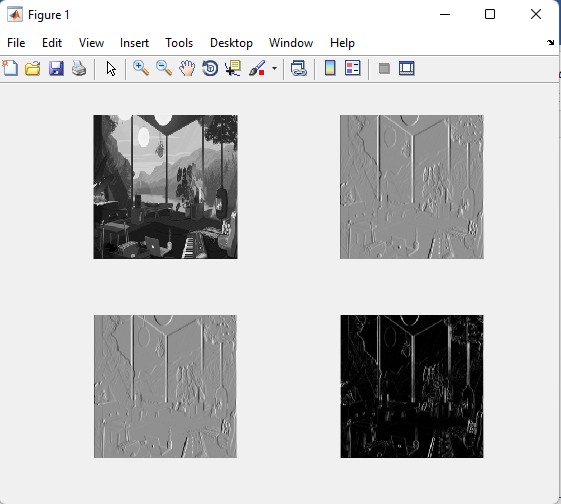
subplot(2,2,3);

imshow(ky,[]);

subplot(2,2,4);

imshow(ked,[]);

**Output**



**(b) Sobel**

**Code**

x = imread('test.png');

x = rgb2gray(x);

x = imresize(x,[256,256]);

x = double(x)

p\_msk = [-1 0 1; -2 0 2; -1 0 1];

kx = conv2(x,p\_msk,'same');

ky = conv2(x,p\_msk,'same');

ked = sqrt(kx.^2 + ky.^2);

subplot(2,2,1);

imshow(x,[]);

subplot(2,2,2);

imshow(kx,[]);

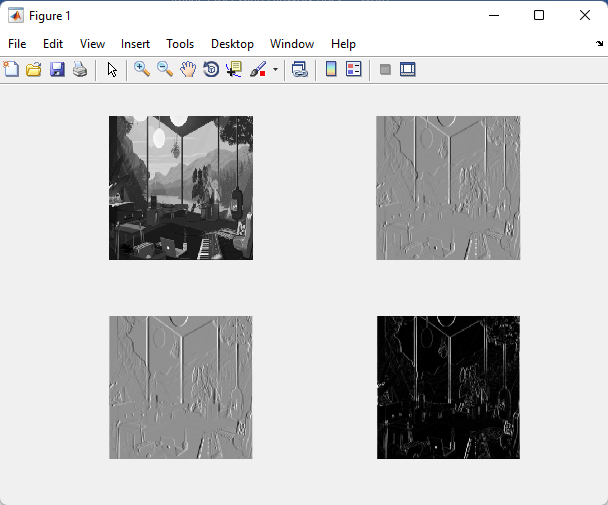
subplot(2,2,3);

imshow(ky,[]);

subplot(2,2,4);

imshow(ked,[]);

**Output**



**(c) Robert**

**Code**

x = imread('test.png');

x = rgb2gray(x);

x = imresize(x,[256,256]);

x = double(x)

p\_msk = [1 0 ; 0 -1 ];

kx = conv2(x,p\_msk,'same');

ky = conv2(x,p\_msk,'same');

ked = sqrt(kx.^2 + ky.^2);

subplot(2,2,1);

imshow(x,[]);

subplot(2,2,2);

imshow(kx,[]);

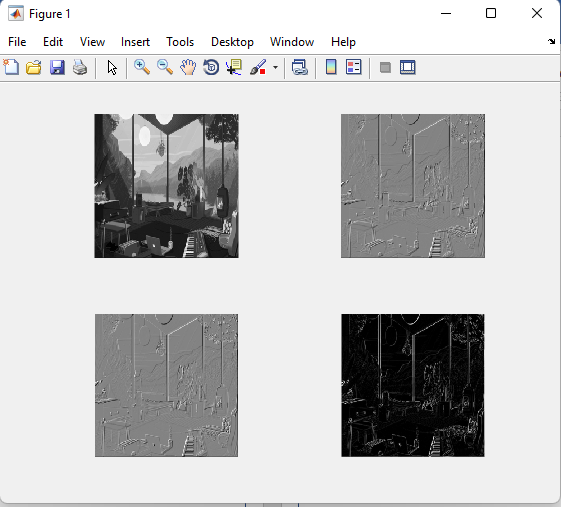
subplot(2,2,3);

imshow(ky,[]);

subplot(2,2,4);

imshow(ked,[]);

**Output**



**Program**

**Ques.**  Write a MATLAB program to perform Bit plane Slicing on input.

**Code**

c = imread('test.png');

cd = double(c);

c1 = mod(cd, 2);

c2 = mod(floor(cd/2), 2);

c3 = mod(floor(cd/4), 2);

c4 = mod(floor(cd/8), 2);

c5 = mod(floor(cd/16), 2);

c6 = mod(floor(cd/32), 2);

c7 = mod(floor(cd/64), 2);

c8 = mod(floor(cd/128), 2);

cc = (2 \* (2 \* (2 \* (2 \* (2 \* (2 \* (2 \* c8 + c7) + c6) + c5) + c4) + c3) + c2) + c1);

subplot(2, 5, 1);

imshow(c);

title('Original Image');

subplot(2, 5, 2);

imshow(c1);

title('Bit Plane 1');

subplot(2, 5, 3);

imshow(c2);

title('Bit Plane 2');

subplot(2, 5, 4);

imshow(c3);

title('Bit Plane 3');

subplot(2, 5, 5);

imshow(c4);

title('Bit Plane 4');

subplot(2, 5, 6);

imshow(c5);

title('Bit Plane 5');

subplot(2, 5, 7);

imshow(c6);

title('Bit Plane 6');

subplot(2, 5, 8);

imshow(c7);

title('Bit Plane 7');

subplot(2, 5, 9);

imshow(c8);

title('Bit Plane 8');

subplot(2, 5, 10);

imshow(uint8(cc));

title('Recombined Image');

**Output**

