Write a program to negative an image.

```
clc;
clear all;
close all;

img = imread('tape.png');

L=2^8;
neg=(L-1)-img;

subplot 211, imshow(img), title('Original Image');
subplot 212, imshow(neg), title('Negative Image');
```

Write a program to Add or Subtract two images.

1st code

```
clc;
clear all;
close all;

a = imread('cameraman.tif');
b = imread('tape.png');

c = imresize(a, [300, 300]);
d = imresize(b, [300, 300]);

for i = 1: 300
    for j = 1:300
        add(i, j) = c(i,j) + d(i,j);
        sub(i, j) = c(i,j) - d(i,j);
    end
end

subplot 221, imshow(a), title('First Image');
subplot 222, imshow(b), title('Secound Image');
subplot 223, imshow(add), title('Addition Image');
subplot 224, imshow(sub), title('Subtracted Image');
```

2nd code

```
clc;
clear all;
close all;

a = imread('cameraman.tif');
b = imread('tape.png');

[r, co] = size(a);

c = imresize(b, [r, co]);

for i = 1: r
    for j = 1:co
        add(i, j) = a(i,j) + c(i,j);
        sub(i, j) = a(i,j) - c(i,j);
    end
end

subplot 221, imshow(a), title('First Image');
subplot 222, imshow(b), title('Secound Image');
subplot 223, imshow(add), title('Addition Image');
subplot 224, imshow(sub), title('Subtracted Image');
```

Write a program for bit plane slicing of 8 bit image.

1st code

```
% clearing the output screen
clc;
clear all;
close all;

% reading image's pixel in c
c = imread('cameraman.tif');

% storing image information in cd
cd = double(c);

% extracting all bit one by one
% from 1st to 8th in variable
% from c1 to c8 respectively
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);

% combining image again to form equivalent to original
grayscale image
cc = (2 * (2 * (2 * (2 * (2 * (2 * (2 * c8 + c7) + c6)
+ c5) + c4) + c3) + c2) + c1);

% plotting original image in first subplot
subplot(2, 5, 1);
imshow(c);
title('Original Image');
```

```
% plotting binary image having extracted bit from 1st
to 8th
% in subplot from 2nd to 9th
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');

% plotting recombined image in 10th subplot
subplot(2, 5, 10);
imshow(uint8(cc));
title('Recombined Image');
```

2nd Code

```
clc;
clear all;
close all;

a = imread('onion.png');
g = rgb2gray(a);

[m n] = size(g);

for i = 1:m
    for j = 1:n
        t = de2bi(g(i,j),8,'right-msb');
        b1(i,j) = t(1,1);
        b2(i,j) = t(1,2);
        b3(i,j) = t(1,3);
        b4(i,j) = t(1,4);
        b5(i,j) = t(1,5);
        b6(i,j) = t(1,6);
        b7(i,j) = t(1,7);
        b8(i,j) = t(1,8);
    end
end

subplot 331, imshow(a), title('Original Image', 'color',
'b');
subplot 332, imshow(logical(b1)), title('Bit Plane 1',
'color', 'b');
subplot 333, imshow(logical(b2)), title('Bit Plane 2',
'color', 'b');
subplot 334, imshow(logical(b3)), title('Bit Plane 3',
'color', 'b');
subplot 335, imshow(logical(b4)), title('Bit Plane 4',
'color', 'b');
subplot 336, imshow(logical(b5)), title('Bit Plane 5',
'color', 'b');
subplot 337, imshow(logical(b6)), title('Bit Plane 6',
'color', 'b');
subplot 338, imshow(logical(b7)), title('Bit Plane 7',
'color', 'b');
subplot 339, imshow(logical(b8)), title('Bit Plane 8',
'color', 'b');
```

3rd code

```
clear all;
close all;
clc;
a=imread('cameraman.tif');

b1=[];
b2=[];
b3=[];
b4=[];
b5=[];
b6=[];
b7=[];
b8=[];

for m=1:256
    for n=1:256
        t=de2bi(a(m,n),8,'left-msb');
        b1(m,n)=t(1,1);
        b2(m,n)=t(1,2);
        b3(m,n)=t(1,3);
        b4(m,n)=t(1,4);
        b5(m,n)=t(1,5);
        b6(m,n)=t(1,6);
        b7(m,n)=t(1,7);
        b8(m,n)=t(1,8);
    end
end

subplot(3,3,1);
imshow(a);
title('image of cameramen','color','r');

subplot(3,3,2);
imshow(b8);
title('image of bit-1','color','r');
subplot(3,3,3);
imshow(b7);
title('image of bit-2','color','r');
```

```
subplot(3,3,4);
imshow(b6);
title('image of bit-3','color','r');
subplot(3,3,5);
imshow(b5);
title('image of bit-4','color','r');
subplot(3,3,6);
imshow(b4);
title('image of bit-5','color','r');
subplot(3,3,7);
imshow(b3);
title('image of bit-6','color','r');
subplot(3,3,8);
imshow(b2);
title('image of bit-7','color','r');
subplot(3,3,9);
imshow(b1);
title('image of bit-8','color','r');
```

4th code

```
clc;
clear all;
close all;

c = imread('cameraman.tif');
% c = rgb2gray(c);     %If the image is rgb
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);

subplot 331, imshow(c), title('Original Image');
subplot 332, imshow(c1), title('Bit Plane 1');
subplot 333, imshow(c2), title('Bit Plane 2');
subplot 334, imshow(c3), title('Bit Plane 3');
subplot 335, imshow(c4), title('Bit Plane 4');
subplot 336, imshow(c5), title('Bit Plane 5');
subplot 337, imshow(c6), title('Bit Plane 6');
subplot 338, imshow(c7), title('Bit Plane 7');
subplot 339, imshow(c8), title('Bit Plane 8');

cc =
(2*(2*(2*(2*(2*(2*(2*c8+c7)+c6)+c5)+c4)+c3)+c2)+c1);
figure, imshow(uint8(cc)), title('Recombined Image');
```

Write a program to Convert into Gray level and then to convert into monocrome.

```
clc;
clear all;
close all;

img = imread('onion.png');

g = 0.299*img(:,:,1) + 0.587*img(:,:,2) +
0.114*img(:,:,3);

t = input('Enter threshold value: ');

[m n] = size(g);

for i = 1:m
    for j = 1:n
        if g(i,j)>t
            bw(i,j) = 255;
        else
            bw(i,j) = 0;
        end
    end
end

subplot 311, imshow(img), title('Original Image');
subplot 312, imshow(g), title('Grayscale Image');
subplot 313, imshow(bw), title('Monochrome Image');
```

Write a program to see the effect of log transformation of an image.

```
clc;
clear all;
close all;

img = imread('peppers.png');

b = im2double(img);

c = 1;
f = c*log(1+b);

figure, imshow(img), title('Original Image');
figure, imshow(f), title('Log transformed Image');
```

Write a program for gamma correcton of an image.

```
clc;
clear all;
close all;

img = imread('cameraman.tif');
% img = rgb2gray(img);    %If the image is rgb

d = im2double(img);
c = 1;
gamma = input('Enter the gamma value: ');

p = c*d.^gamma;

maxval = max(p(:));
minval = min(p(:));

[m n] = size(img);

for i = 1:m
    for j = 1:n
        g(i,j) = 255*p(i,j)/(maxval - minval);
    end
end
g = uint8(g);

figure, imshow(img), title('Original Image');
figure, imshow(g), title('Gamma corrected Image');
```

Write a program to draw histogram of an image.

```
clc;
clear all;
close all;

I = imread('onion.png');

h = zeros(1,256);
[m n] = size(I);

for i = 1:m
    for j = 1:n
        h(I(i,j)+1) = h(I(i,j)+1)+1;
    end
end

subplot 211, imshow(I), title('Image');
subplot 212, stem(h), title('Histogram of the Image');
```

Write a program for image zooming and shrinking.

```
clc;
clear all;
close all;

I = imread('logo.tif');
% I = rgb2gray(I);    %If the image is rgb

[m n] = size(I);

z = input('Enter zooming factor: ');

for i = 1:m*z;
    for j = 1:n*z;
        p = ceil(i/z);
        q = ceil(j/z);
        zoom(i,j) = I(p,q);
    end
end

s = input('Enter shrinking factor: ');

for i = 1:m/s;
    for j = 1:n/s;
        p = floor(i*s);
        q = floor(j*s);
        shrink(i,j) = I(p,q);
    end
end

figure, imshow(I), title('Original Image');
figure, imshow(zoom), title('Zoomed Image');
figure, imshow(shrink), title('Shrinked Image');
```

Write a program to apply Smoothing filter on a Image.

```
clc;
clear all;
close all;

I = imread('cameraman.tif');
% I = rgb2gray(I);    %If the image is rgb
d = im2double(I);

[r,c] = size(I);

f = zeros(r,c);

for i = 1:r-2
    for j = 1:c-2
        sum = 0;
        for k = i:i+2
            for l = j:j+2
                sum = sum+d(k,l);
            end
        end
        f(i+1,j+1) = sum/9;
    end
end

figure, imshow(I), title('Original Image');
figure, imshow(f), title('Smooth Image');
```