1 write a program to read an image to concert gray to binary and show all image in single plot in digital image processing using matlab

img = imread('your\_image.jpg');

gray\_img = rgb2gray(img);

binary\_img = imbinarize(gray\_img);

figure;

subplot(1,3,1), imshow(img), title('Original');

subplot(1,3,2), imshow(gray\_img), title('Grayscale');

subplot(1,3,3), imshow(binary\_img), title('Binary');

2write a program to perform addition ,subtraction multiplication,division by any value to enhance an image and show in subplot in digital image processing using matlab and give short code

img = imread('your\_image.jpg');

gray\_img = rgb2gray(img);

add\_img = gray\_img + 50;

sub\_img = gray\_img - 50;

mul\_img = gray\_img \* 1.5;

div\_img = gray\_img / 1.5;

figure;

subplot(2,3,1), imshow(gray\_img), title('Original');

subplot(2,3,2), imshow(add\_img), title('Addition');

subplot(2,3,3), imshow(sub\_img), title('Subtraction');

subplot(2,3,4), imshow(mul\_img), title('Multiplication');

subplot(2,3,5), imshow(div\_img), title('Division');

3write a program to provide thresholding an image and evaluation of histogram using histogram equazation and illustrate relation between intenities of an image and its histogram in digital image processing using matlab and give short code AND give imp viva questions for these with answers

img = imread('your\_image.jpg');

gray\_img = rgb2gray(img);

% Apply thresholding

threshold = 128;

binary\_img = gray\_img > threshold;

% Histogram equalization

equalized\_img = histeq(gray\_img);

% Display results

figure;

subplot(2,3,1), imshow(gray\_img), title('Original Grayscale');

subplot(2,3,2), imshow(binary\_img), title('Thresholded Image');

subplot(2,3,3), imshow(equalized\_img), title('Histogram Equalized');

subplot(2,3,4), imhist(gray\_img), title('Original Histogram');

subplot(2,3,5), imhist(equalized\_img), title('Equalized Histogram');

4write a program to show geometric transformation like scaling,translation,cropping in digital image processing using matlab and give short code AND give imp viva questions for these with answers

img = imread('your\_image.jpg');

gray\_img = rgb2gray(img);

% Scaling

scaled\_img = imresize(gray\_img, 1.5);

% Translation

translated\_img = imtranslate(gray\_img, [50, 30]);

% Cropping

cropped\_img = imcrop(gray\_img, [50 50 200 200]);

% Display results

figure;

subplot(2,2,1), imshow(gray\_img), title('Original Image');

subplot(2,2,2), imshow(scaled\_img), title('Scaled Image');

subplot(2,2,3), imshow(translated\_img), title('Translated Image');

subplot(2,2,4), imshow(cropped\_img), title('Cropped Image');

write a prog to perform 2d fourier transformation in an image in digital image processing using matlab and give short code AND give imp viva questions for these with answers

img = imread('your\_image.jpg');

gray\_img = rgb2gray(img);

% Compute 2D Fourier Transform

F = fft2(gray\_img);

F\_shifted = fftshift(F);

% Compute magnitude spectrum

magnitude\_spectrum = log(1 + abs(F\_shifted));

% Display results

figure;

subplot(1,2,1), imshow(gray\_img), title('Original Image');

subplot(1,2,2), imshow(mat2gray(magnitude\_spectrum)), title('Magnitude Spectrum');

write a program on spatial resolution (dec sampling levels) make 512\*512 image ti 16\*16 (down sampling) to reduce the size of image for transmission and storage in digital image processing using matlab and give short code AND give imp viva questions for these with answers

img = imread('your\_image.jpg');

gray\_img = rgb2gray(img);

% Downsampling to reduce spatial resolution

small\_img = imresize(gray\_img, [16, 16], 'nearest');

% Upsampling back to original size for visualization

upsampled\_img = imresize(small\_img, [512, 512], 'nearest');

% Display results

figure;

subplot(1,3,1), imshow(gray\_img), title('Original 512x512 Image');

subplot(1,3,2), imshow(small\_img), title('Downsampled 16x16 Image');

subplot(1,3,3), imshow(upsampled\_img), title('Upsampled 512x512 Image');

wap for interpolatiob trchniwue for zooming image (make 64\*64 to 512\*512)by applying types of interpolation (NNI,bilinear interpolatob and bicubic interpolation)-up sampling in digital image processing using matlab and give short code AND give imp viva questions for these with answers

img = imread('your\_image.jpg');

gray\_img = rgb2gray(img);

% Downsampling to 64x64

small\_img = imresize(gray\_img, [64, 64], 'nearest');

% Upsampling using different interpolation techniques

nni\_img = imresize(small\_img, [512, 512], 'nearest');

bilinear\_img = imresize(small\_img, [512, 512], 'bilinear');

bicubic\_img = imresize(small\_img, [512, 512], 'bicubic');

% Display results

figure;

subplot(2,2,1), imshow(gray\_img), title('Original Image');

subplot(2,2,2), imshow(nni\_img), title('Nearest Neighbor Interpolation');

subplot(2,2,3), imshow(bilinear\_img), title('Bilinear Interpolation');

subplot(2,2,4), imshow(bicubic\_img), title('Bicubic Interpolation');

wap on intensity resolution (decreasing quantization levels) Make 8 bit image to 1 bit while keeping spatial resolution constant to observe false contouring in digital image processing using matlab and give short code AND give imp viva questions for these with answers

img = imread('your\_image.jpg');

gray\_img = rgb2gray(img);

% Convert to double for processing

gray\_img = double(gray\_img);

% Reduce intensity levels (quantization)

bit\_4 = uint8(round(gray\_img / 16) \* 16); % 4-bit quantization

bit\_2 = uint8(round(gray\_img / 64) \* 64); % 2-bit quantization

bit\_1 = uint8(gray\_img > 128) \* 255; % 1-bit quantization (Black & White)

% Display results

figure;

subplot(2,2,1), imshow(uint8(gray\_img)), title('Original 8-bit Image');

subplot(2,2,2), imshow(bit\_4), title('4-bit Quantized Image');

subplot(2,2,3), imshow(bit\_2), title('2-bit Quantized Image');

subplot(2,2,4), imshow(bit\_1), title('1-bit Quantized Image (Binary)');

wap for increasing quantization(make 8 bit image (2^8 to 256)to 16 bit image(2^16 to 65536)to observe that there is no visible difference in digital image processing using matlab and give short code AND give imp viva questions for these with answers

img = imread('your\_image.jpg');

gray\_img = rgb2gray(img);

% Convert 8-bit image (0-255) to 16-bit (0-65535)

img\_16bit = uint16(gray\_img) \* 256;

% Display results

figure;

subplot(1,2,1), imshow(gray\_img), title('Original 8-bit Image');

subplot(1,2,2), imshow(img\_16bit, []), title('Converted 16-bit Image');