



RAMANUJAN COLLEGE

(UNIVERSITY OF DELHI)

**SUBJECT-DIGITAL IMAGE PROCESSING
(PRACTICAL FILE)**

**NAME=MOHD SHOHEL
COLLEGE ROLL NO. = 20201417
EXAMINATION ROLL NO. = 20020570022
SUBMITTED TO = MRS. BHAVYA AHUJA
SEMESTER = 5th**

Question 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

Solution :

Code -

```
% (a)
a=imread("C:\Users\MOHD SHOHEL\Downloads\lion.jpg");
imshow(a);
title("original image");

% (b)
a=imread("C:\Users\MOHD SHOHEL\Downloads\lion.jpg");
imshow(a);
title("original image");

% (c)
img = imread('C:\Users\MOHD SHOHEL\Downloads\fish.jpg');

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

subplot(2,2,1), title('Original Image'), imshow(img);
xlabel(strcat(string(size(img)),'* '));
```

```
subplot(1,2,2), title('Resized Image'), imshow(img1);
xlabel(strcat(string(size(img1)),'* '));

% (d)
RGB=imread("C:\Users\MOHD SHOHEL\Downloads\flower.jpg");
figure
imshow(RGB);
title("Original image");
I=rgb2gray(RGB);
figure
imshow(I);
title("gray-scale image");

% (e)
im1 = imread('C:\Users\MOHD SHOHEL\Downloads\fish.jpg');
bw_img = im2bw(img1, 0.5);

subplot(1,2,1), title('Original Image'), imshow(img1);
subplot(1,2,2), title('B&W Image'), imshow(bw_img);

% (f)
i = imread("C:\Users\MOHD SHOHEL\Downloads\fish.jpg");
improfile(i);

% (g),(h)
i = imread("C:\Users\MOHD SHOHEL\Downloads\fish.jpg");
redChannel = i(:,:,1); // Red channel
greenChannel = i(:,:,2); // Green channel
blueChannel = i(:,:,3); // Blue channel
// Create an all black channel.
allBlack = zeros(size(i, 1), size(i, 2), 'uint8');
// Create color versions of the individual color channels.
just_red = cat(3, redChannel, allBlack, allBlack);
just_green = cat(3, allBlack, greenChannel, allBlack);
just_blue = cat(3, allBlack, allBlack, blueChannel);
// Recombine the individual color channels to create the original RGB image again.
recombinedRGBImage = cat(3, redChannel, greenChannel, blueChannel);
// Display them all.
figure
```

```

subplot(3, 3, 2);
imshow(i);
fontSize = 2;
title('Original RGB Image', 'FontSize', fontSize)
subplot(3, 3, 4);
imshow(just_red);
title('Red Channel in Red', 'FontSize', fontSize)
subplot(3, 3, 5);
imshow(just_green)
title('Green Channel in Green', 'FontSize', fontSize)
subplot(3, 3, 6);
imshow(just_blue);
title('Blue Channel in Blue', 'FontSize', fontSize)
subplot(3, 3, 8);
imshow(recombinedRGBImage);
title('Recombined to Form Original RGB Image Again', 'FontSize', fontSize)

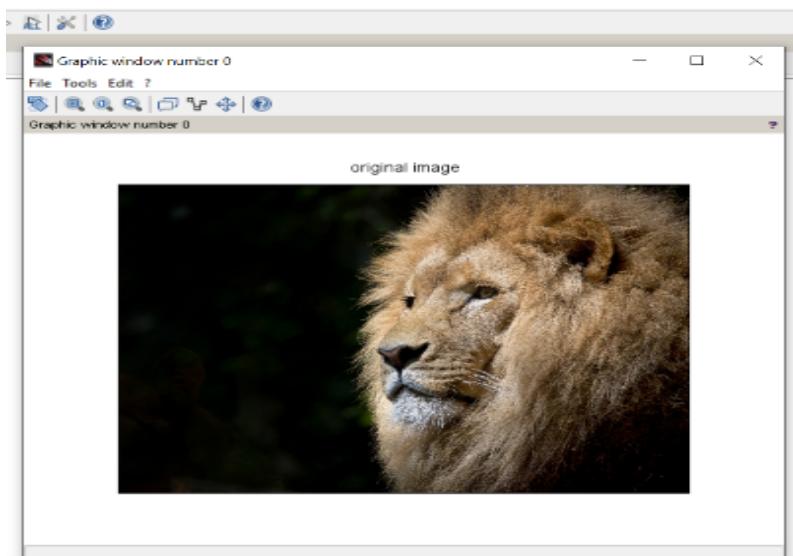
% (i)
for i=1:10
    if modulo(i,2) == 1 then
        disp(i);
    else
        continue;
    end,
end;
% (j)
mat = zeros(20,20,'uint8');
mat(5:10, 5:10) = 1;      // box
mat(2, 3:15) = 1;         //vertical line
mat(2:18, 3) = 1;         //horizontal line
mat(1:4:20,1:4:20) = 1;   // linear points

disp('Given 2D data : ', mat);
title('Created Image from matrix'), imshow(mat2gray(mat));

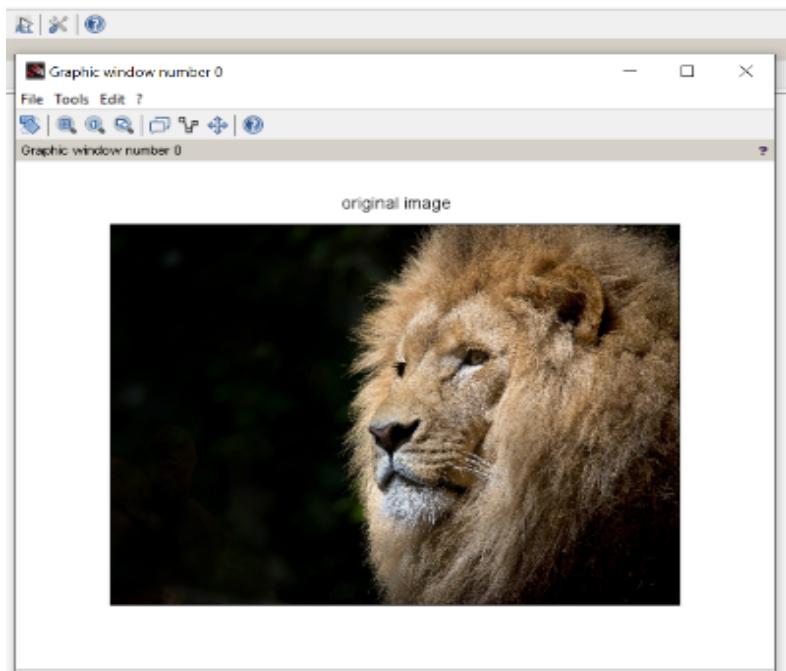
```

Output :

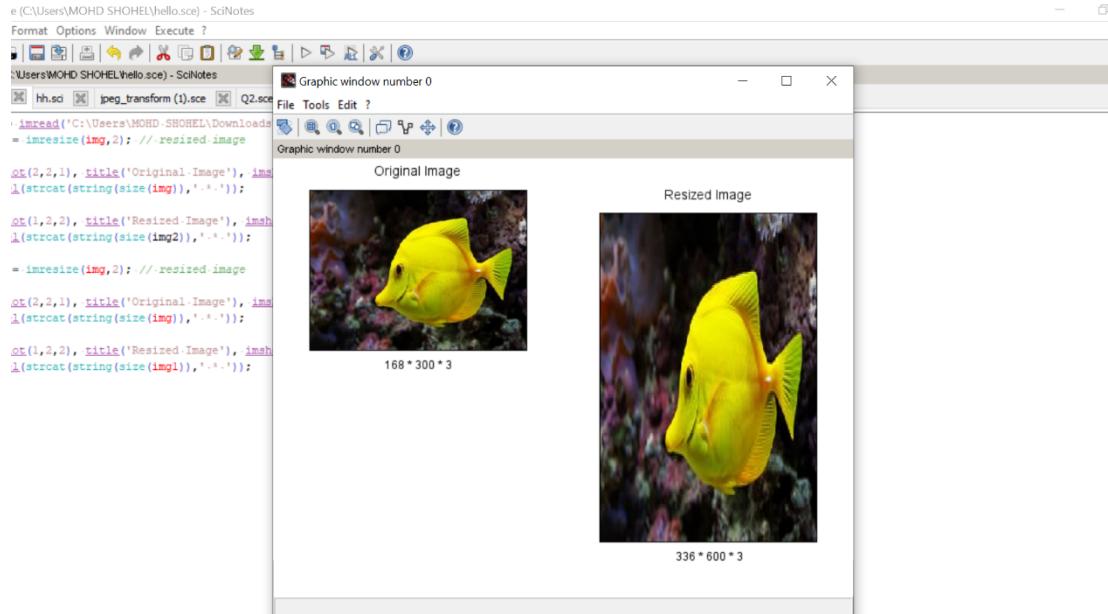
(a)



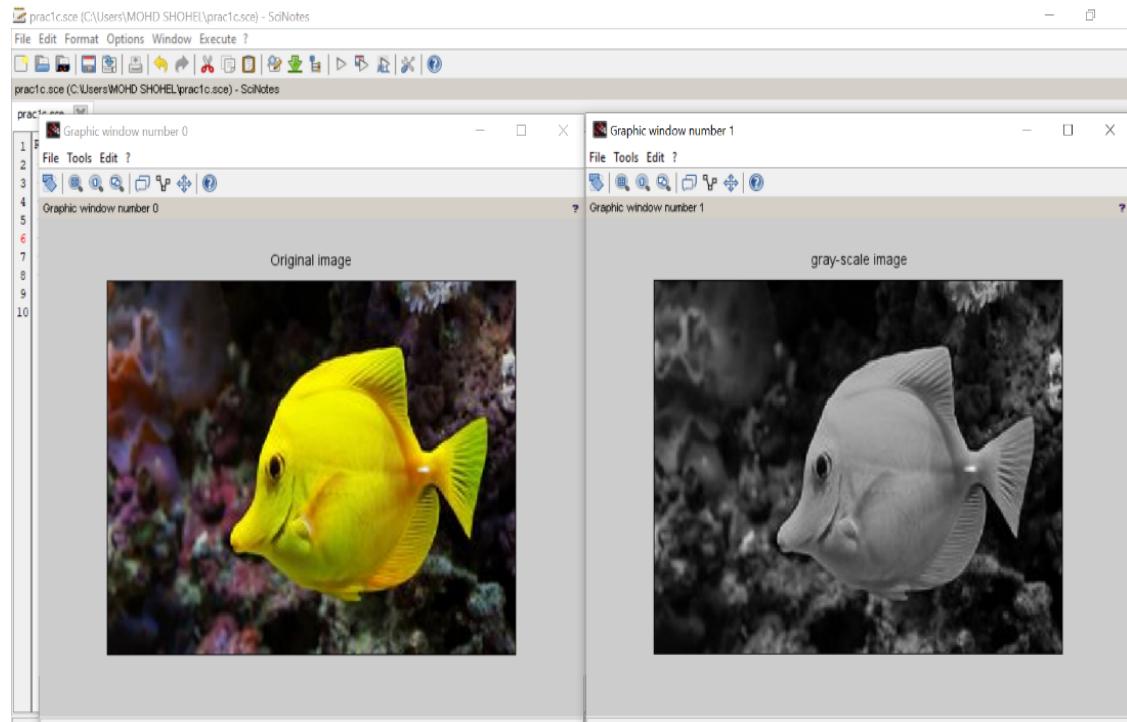
(b)



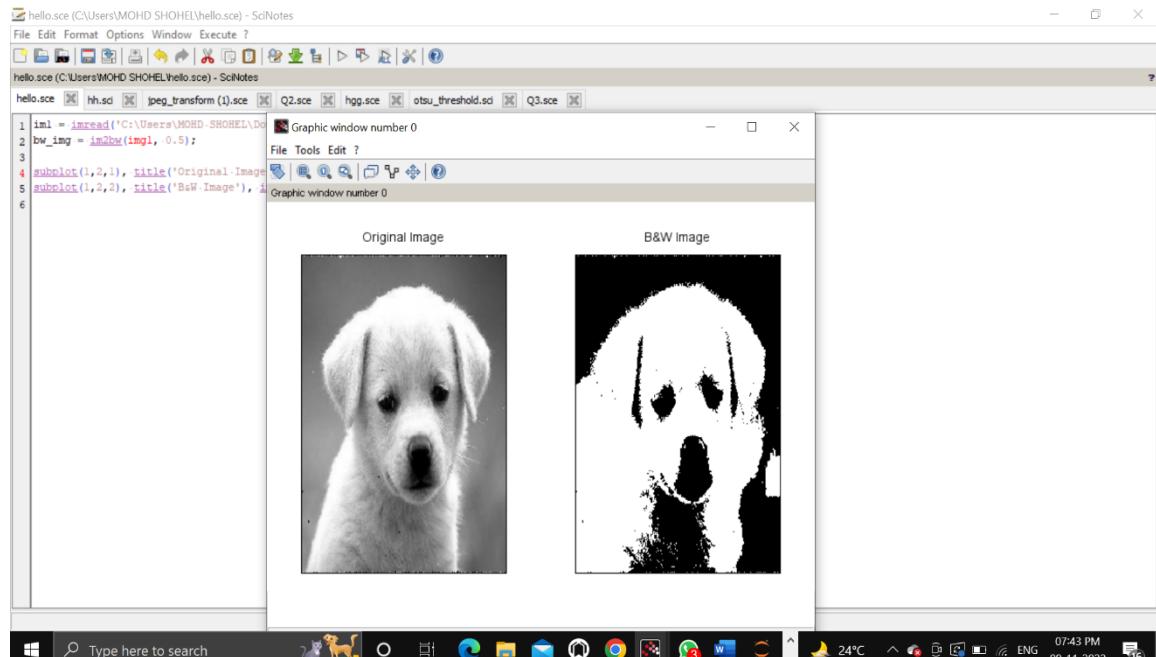
(c)



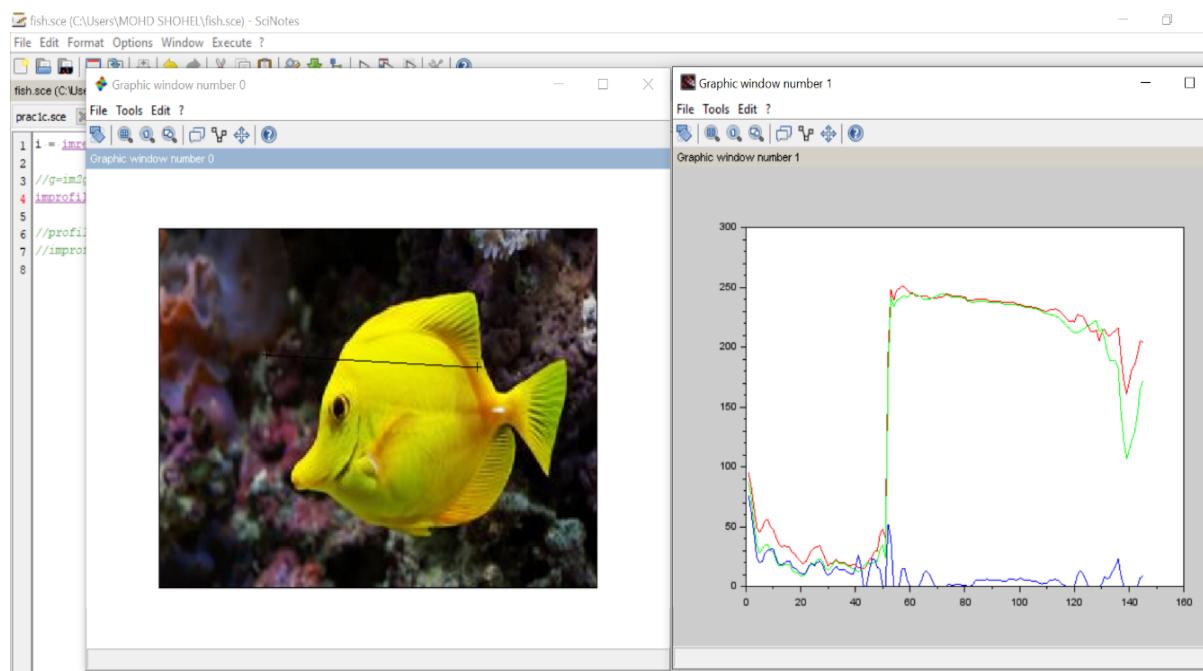
(d)



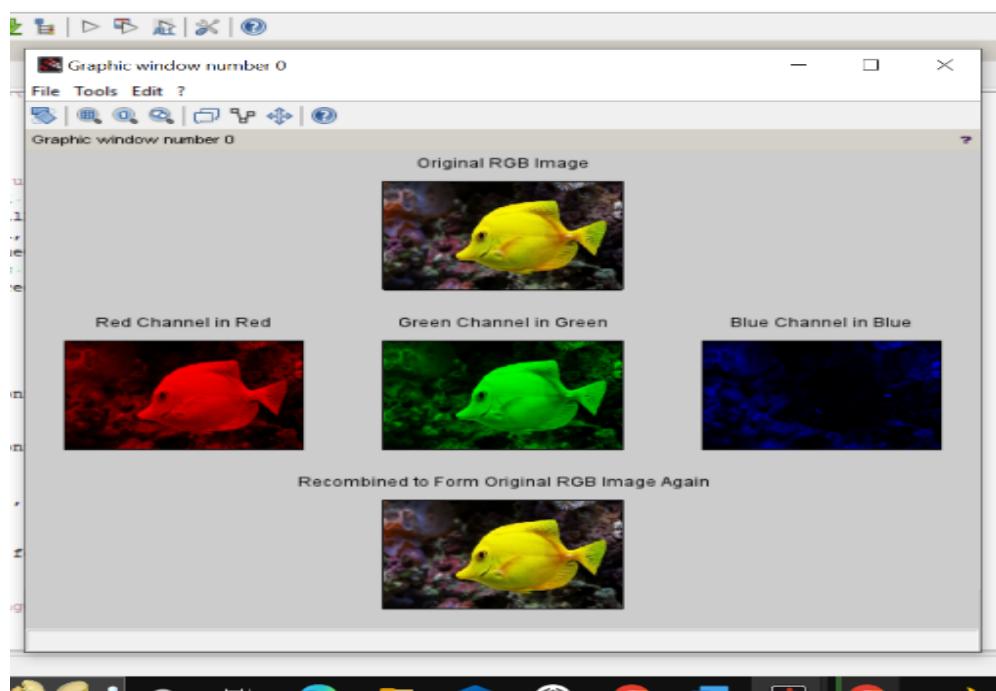
(e)



(f)



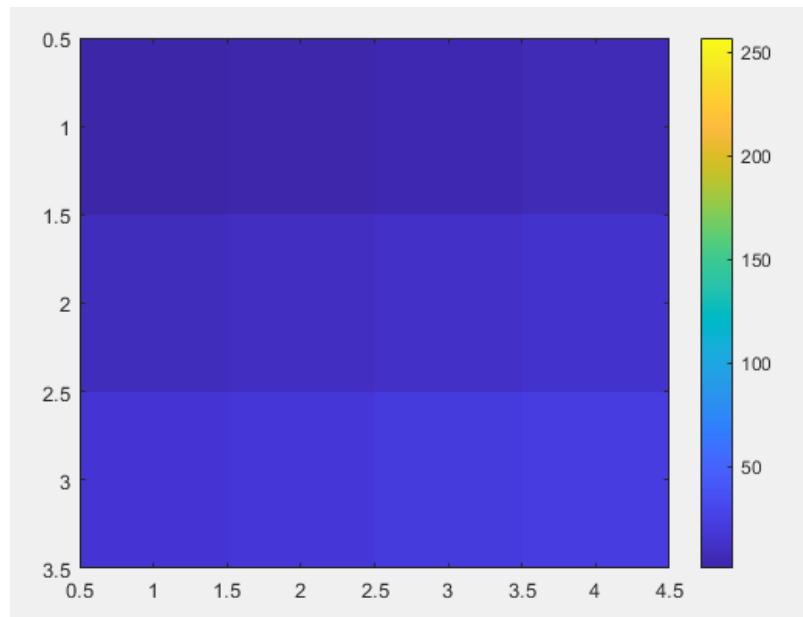
(g),(h)



(i)

```
>> ques1i
Enter a Number 4
The Number is Even
Bipolar Signal
    1      1      1      1      1      1      1      1      1      1
```

(j)



Question 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

Solution :

Code -

```
% (a)
a=imread("C:\Users\MOHD SHOHEL\Downloads\fish.jpg");
figure
imshow(a);
title("Original Image")
d=255-a;
figure
imshow(d);
title("Negative Image")

% (b)
I=imread("C:\Users\MOHD SHOHEL\Downloads\fish.jpg");
S2 = imrotate(I,180);

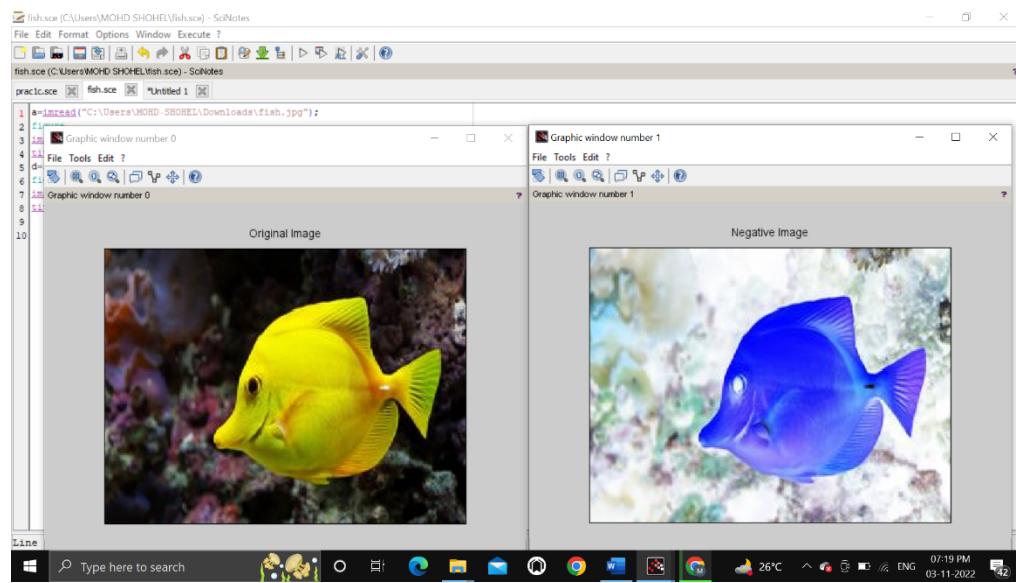
imshow(S2);
% (c)
sam_img = imread('C:\Users\MOHD SHOHEL\Downloads\fish.jpg');

subplot(1,2,1), title('Original Image'), imshow(sam_img);
subplot(1,2,2), title('Binary Image at D0 = 110'), imshow(im2bw(sam_img, 110/256));
% (d)
X=imread("C:\Users\MOHD SHOHEL\Downloads\duck.jpg");
//X = imread('image.jpg'); //%%reading a grayscale image
figure(1);
imshow(X);
title('Original Image')
a = min(X(:)); // %minimum pixel of image X
b = max(X(:)); //%%maximum pixel of image X
X= (X-a).*(255/(b-a)); //%%just using the formula above
figure(2);
```

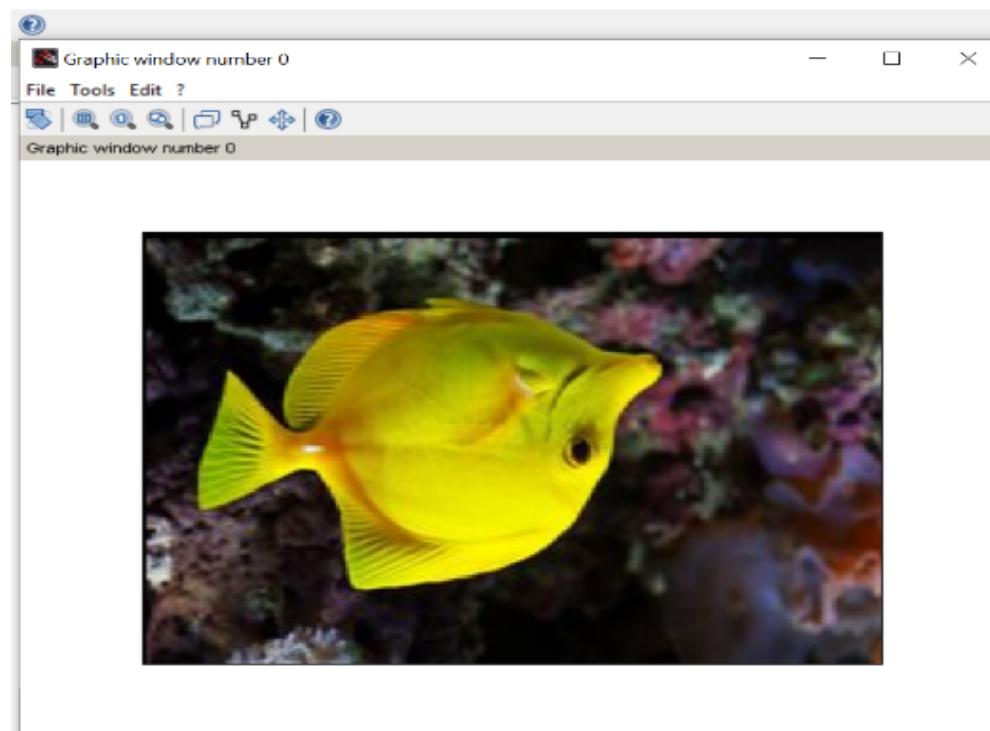
```
imshow(X);
title('Contrast Streached Image')
```

Output :

(a)



(b)



(c)

The figure shows the SciNotes software interface with the following details:

- Title Bar:** hello.sce (C:\Users\MOHD SHOHEL\hello.sce) - SciNotes
- Toolbar:** Standard file operations (New, Open, Save, Print, etc.) and a search icon.
- File Menu:** File, Edit, Format, Options, Window, Execute ?
- Graphic Window Number 0:** A dropdown menu listing open files: hello.sce, hh.sd, peg_transform (1).sce, Q2.sd.
- Code Editor:** Displays MATLAB-style pseudocode:

```
1 sam_img = imread('C:\Users\MOHD SHOHEL\Downl...  
2  
3 subplot(1,2,1), title('Original Image'), im...  
4 subplot(1,2,2), title('Binary Image at D0 = ...  
5
```
- Graphics Area:** Contains two side-by-side images.
 - Original Image:** A vibrant yellow tang fish against a dark, textured background.
 - Binary Image at D0 = 110:** A high-contrast black and white version of the fish, where the yellow body is mostly white against a black background.

(d)

The screenshot shows two windows from the SciNotes application. The left window, titled 'pp.sce (C:\Users\MOHD SHOHEL\pp.sce) - SciNotes', contains MATLAB code for reading an image, creating a figure, displaying it, and then stretching the contrast. The right window, titled 'Graphic window number 2', displays the original image of a white duck standing on grass. The second window, titled 'Contrast Streached Image', displays the same image with increased contrast.

```
1 X=imread("C:\Users\MOHD SHOHEL\Downloads\white duck.jpg");
2 //X=.imread Graphic window number 1
3 figure(1);
4 imshow(X);
5 title('Original Image');
6 a = min(X(:));
7 b = max(X(:));
8 X= (X-a).*(b-a);
9 figure(2);
10 imshow(X);
11 title('Contrast Streached Image');
```

Question 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

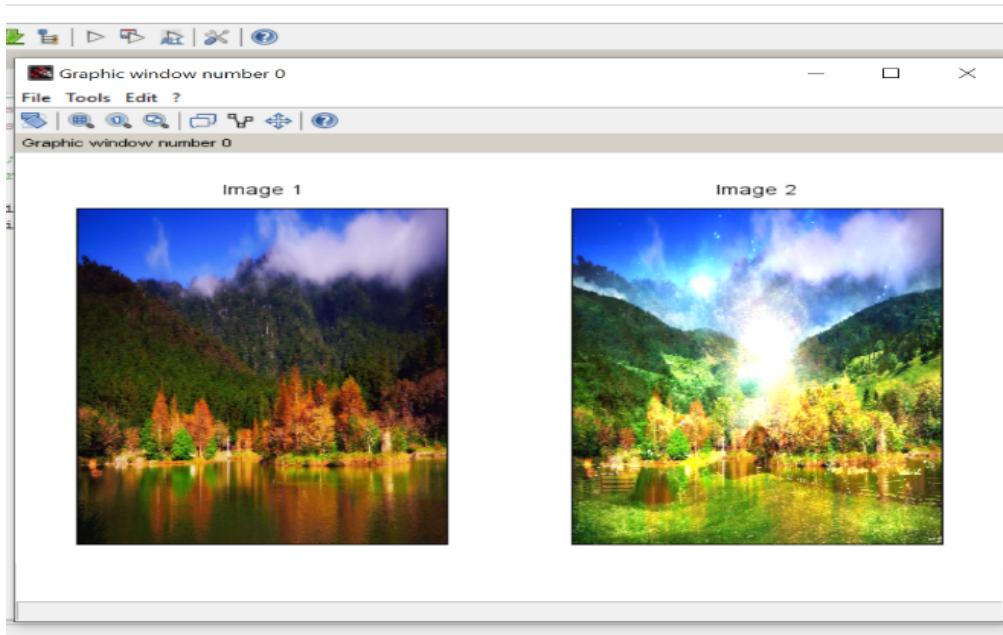
Solution :

Code -

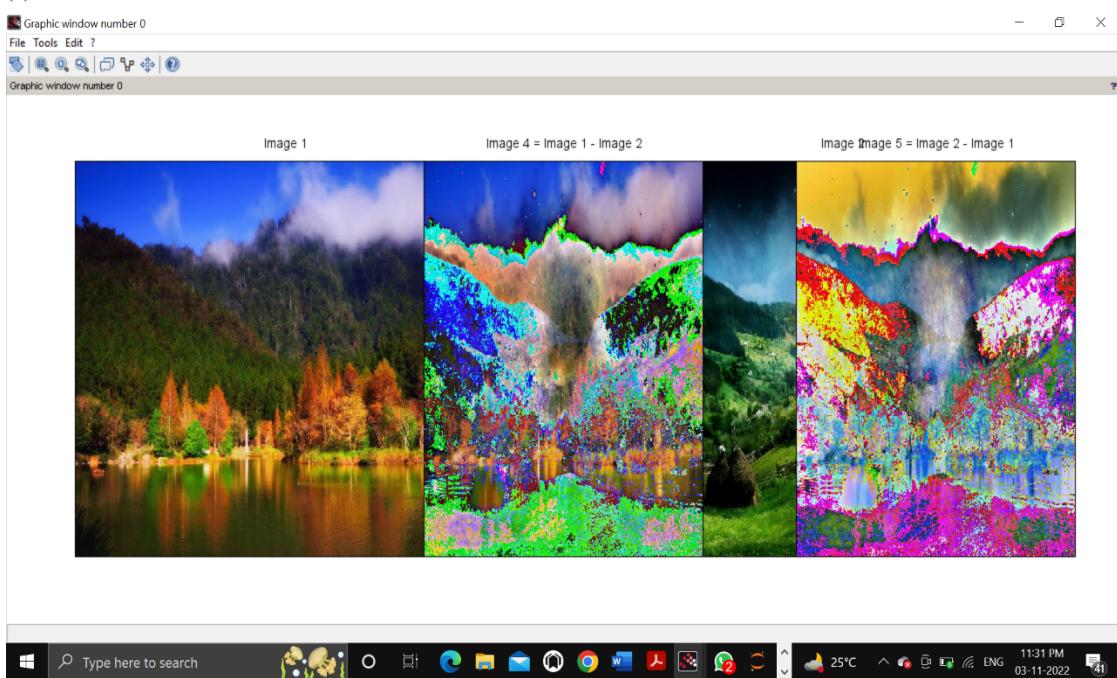
```
% (a)  
% Read two grayscale uint8 images into the workspace.  
im1=imread("C:\Users\MOHD SHOHEL\Downloads\nature1.jpg");  
im2=imread("C:\Users\MOHD SHOHEL\Downloads\nature2.jpg");  
  
//im1 = imread('Test_images/lena_1.jpeg');  
//im2 = imresize(imread('Test_images/coloredChips.png'), [225,225]);  
  
subplot(1,2,1), title('Image 1'), imshow(im1);  
subplot(1,2,2), title('Image 2'), imshow(im2);  
  
im3 = imadd(im1, im2);  
imshow(im3);  
% (b)  
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);  
  
% (c)  
printf('Mean of the image 1 : %.3f\n',mean(im2double(im1)));  
printf('Mean of the image 2 : %.3f,mean(im2double(im2)));
```

Output :

(a)

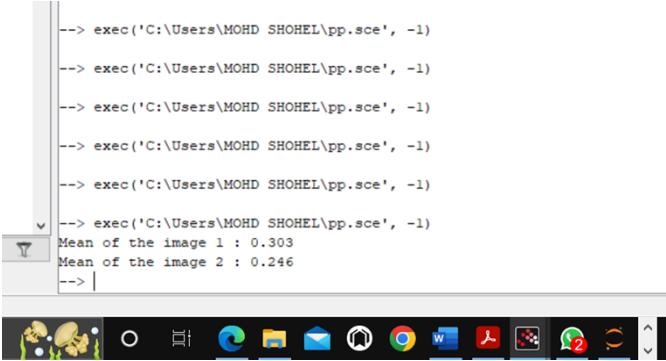


(b)



(c)





```
--> exec('C:\Users\MOHD SHOHEL\pp.sce', -1)
Mean of the image 1 : 0.303
Mean of the image 2 : 0.246
--> |
```

Question 4 : To write and execute programs for image logical operations

- AND operation between two images
- OR operation between two images
- Calculate intersection of two images
- NOT operation (Negative image)

Solution :

Code -

```
% (a)
i= imread('C:\Users\MOHD SHOHEL\Downloads\f1.jfif');
j= imread('C:\Users\MOHD SHOHEL\Downloads\f2.jfif');
andimage=imabsdiff(i,j);
imshow(andimage);

% (b)
i= imread('C:\Users\MOHD SHOHEL\Downloads\f1.jfif');
j= imread('C:\Users\MOHD SHOHEL\Downloads\f2.jfif');
orimage=bitor(i,j);
imshow(orimage);

% (c)
a= imread('C:\Users\MOHD SHOHEL\Downloads\f1.jfif');

b=imread('C:\Users\MOHD SHOHEL\Downloads\f2.jfif');

intersectedImage=bitand(a,b);
subplot(2,2,3), imshow(a), title('rice');
subplot(2,2,4), imshow(b), title('smile');
subplot(2,2,2), imshow(intersectedImage), title('Intersection');

% (d)
```

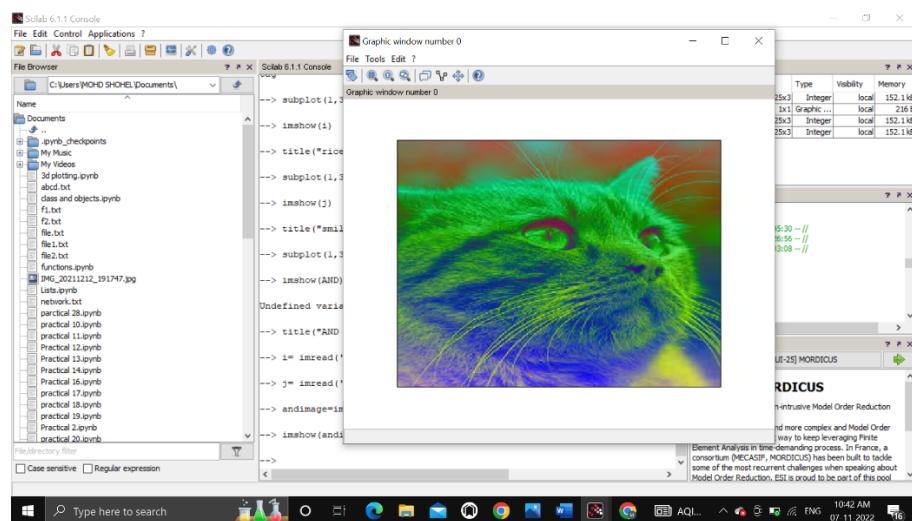
```

W= imread('C:\Users\MOHD SHOHEL\Downloads\f1.jfif');
NotW= bitcmp(W);
figure
subplot(1,2,1)
imshow(W)
title("cat")
subplot(1,2,2)
imshow(NotW)
title("NOT Operation")

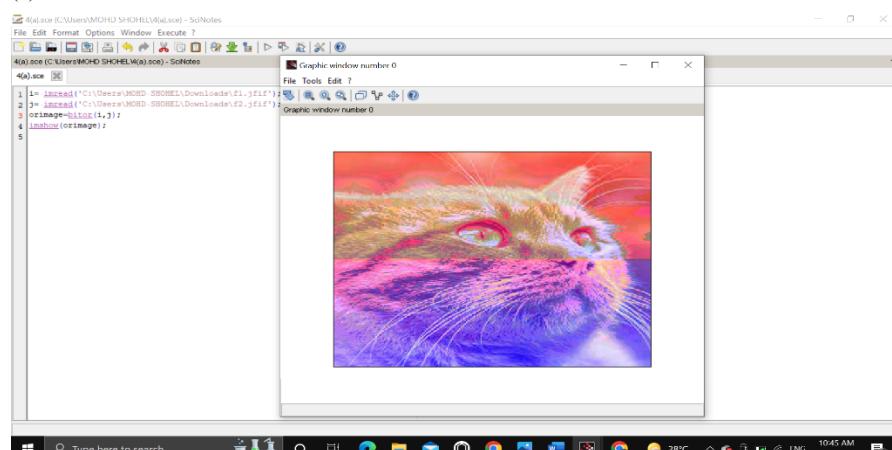
```

Output :

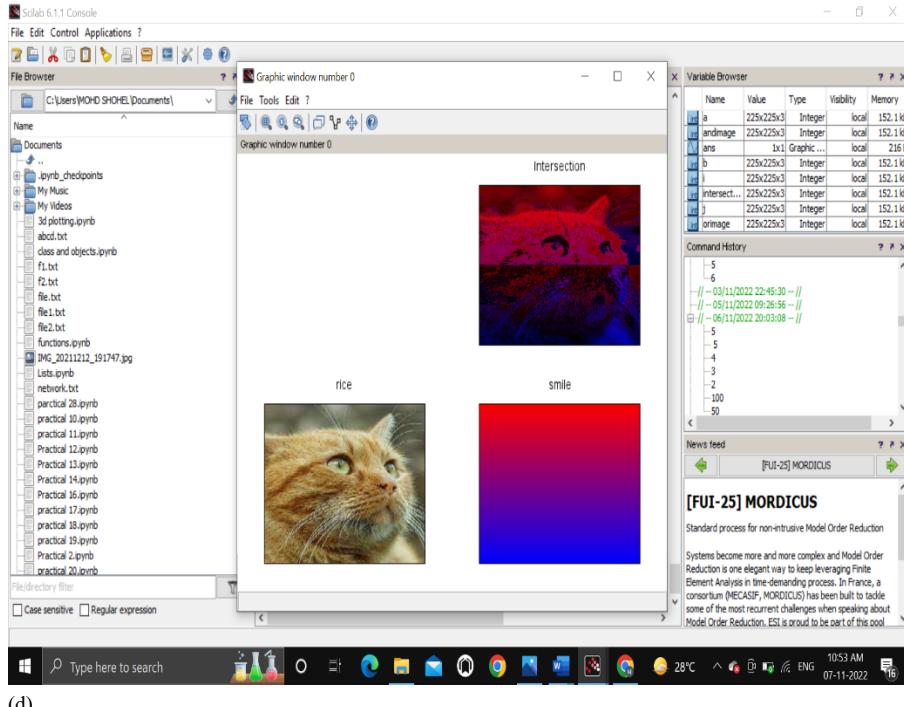
(a)



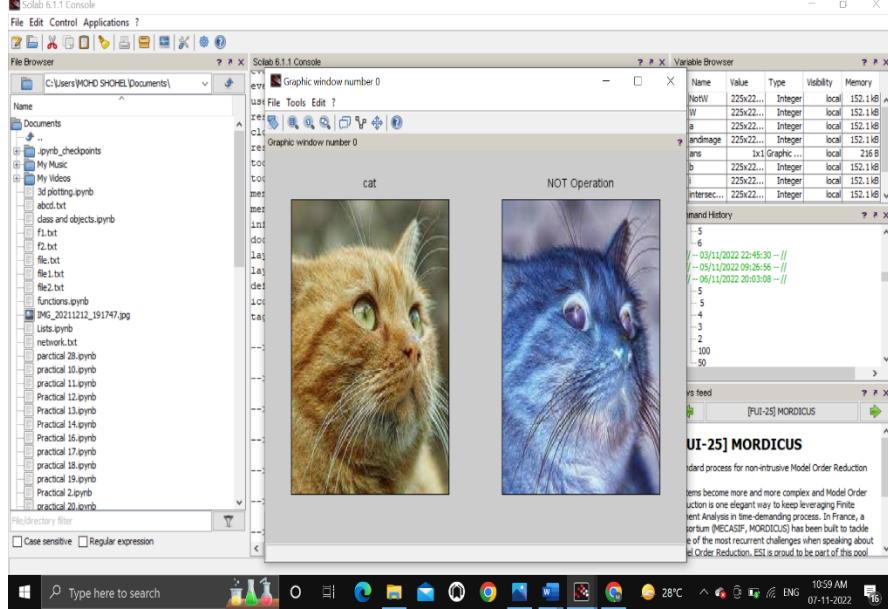
(b)



(c)



(d)



Question 5 : To write a program for histogram calculation and equalization using

- Standard MATLAB function
- Program without using standard MATLAB functions

Solution :

Code -

% (a)

```
I=rgb2gray(imread("C:\Users\MOHD SHOHEL\Downloads\fish.jpg"));
```

```
subplot(2,2,1);
imshow(I);
subplot(2,2,2);
imhist(I,[],1);
J=imhisteq(I);
```

```
subplot(2,2,3);
imshow(J);
subplot(2,2,4);
imhist(J,[], 1);
```

% (b)

```
GIm=imread('penguin.jpg');
numofpixels=size(GIm,1)*size(GIm,2);
figure
imshow(GIm);
title('Original Image');
HIm=uint8(zeros(size(GIm,1),size(GIm,2)));
freq=zeros(256,1);
probff=zeros(256,1);
probcc=zeros(256,1);
cum=zeros(256,1);
output=zeros(256,1);

%freq counts the occurrence of each pixel value.
%The probability of each occurrence is calculated by probf.
for i=1:size(GIm,1)
    for j=1:size(GIm,2)
        value=GIm(i,j);
        freq(value+1)=freq(value+1)+1;
        probff(value+1)=freq(value+1)/numofpixels;
    end
end
sum=0;
```

```

no_bins=255;

%The cumulative distribution probability is calculated.

for i=1:size(prob)
    sum=sum+freq(i);
    cum(i)=sum;
    probc(i)=cum(i)/numofpixels;
    output(i)=round(probc(i)*no_bins);
end

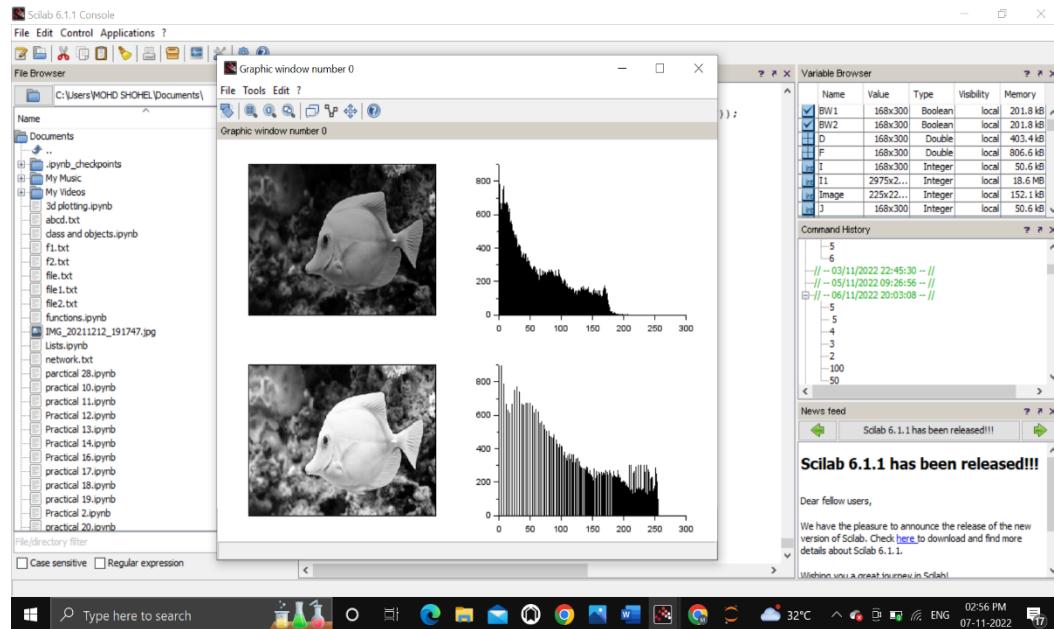
for i=1:size(GIm,1)
    for j=1:size(GIm,2)
        HIm(i,j)=output(GIm(i,j)+1);
    end
end

figure,imshow(HIm);
title('Histogram equalization');

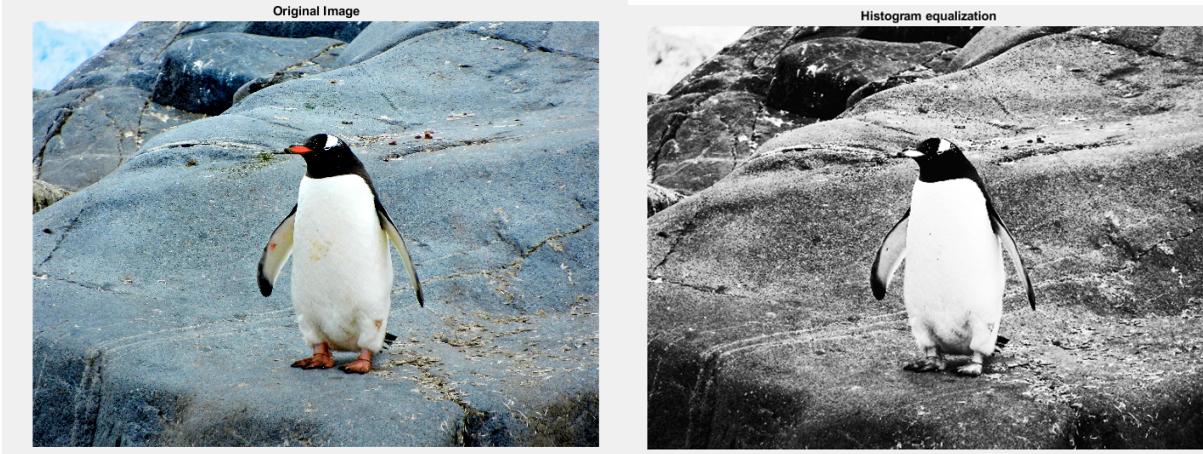
```

Output :

(a)



(b)



Question 6 : To write and execute program for geometric transformation of image

- a. Translation
- b. Scaling
- c. Rotation
- d. Shrinking
- e. Zooming

Solution :

Code -

```
% (a)
```

```
S1=imread("C:\Users\MOHD SHOHEL\Downloads\fish.jpg");
```

```
mat = [ 1 0 0;...
        0 1 0;...
        20 0 1];
S2 = imtransform(S1,mat,'affine');
```

```
mat = [ 1 0 0;...
        0 1 0;...
        0 -20 1];
S3 = imtransform(S1,mat,'affine');
```

```

mat = [ 1 0 0;...
        0 1 0;...
        -20 30 1];
S4 = imtransform(S1,mat,'affine');

subplot(2,2,1), title('Original Image'), imshow(S1);
subplot(2,2,2), title('Translation for x = 20'), imshow(S2);
subplot(2,2,3), title('Translation for y = -20'), imshow(S3);
subplot(2,2,4), title('Translation for (-20,30)'), imshow(S4);

% (b)

```

```
s_img=imread("C:\Users\MOHD SHOHEL\Downloads\fish.jpg");
```

```

width = size(s_img, 'c');
height = size(s_img, 'r');

w = 2;
h = 1;
```

```

mat = [ w 0;
        0 h;
        0 0];
```

```
sc1 = imtransform(s_img, mat, 'affine', width*w, height*h);
```

```
w = 1;
h = 2;
```

```

mat = [ w 0;
        0 h;
        0 0];
```

```
sc2 = imtransform(s_img, mat, 'affine', width*w, height*h);
```

```
w = 2;
h = 2;

mat = [ w 0;
         0 h;
         0 0];

sc3 = imtransform(s_img, mat, 'affine', width*w, height*h);

function s = str(img)
    s = 'Size : ' + strcat(string(size(img)), ' * ');
endfunction;

subplot(3,3,1), title('Original Image'), xlabel(str(s_img)), imshow(s_img);
subplot(3,2,2), title('Image scaling width by 2'), xlabel(str(sc1)), imshow(sc1);
subplot(2,3,4), title('Image scaling height by 2'), xlabel(str(sc2)), imshow(sc2);
subplot(2,2,4), title('Image scaling by 2'), xlabel(str(sc3)), imshow(sc3);

% (c)
subplot(2,2,1), title('Original Image'), imshow(s_img);
subplot(2,2,2), title('Image rotation by 45'), imshow(imrotate(s_img, 45));
subplot(2,2,3), title('Image rotaion by -45'), imshow(imrotate(s_img, -45));
subplot(2,2,4), title('Image rotaion by 180'), imshow(imrotate(s_img, 180));

% (d)
[r c] = size(s_img);
f = 0.5;
im_50 = zeros(r, c, 'uint8');
shrinked = rgb2gray(imresize(s_img, f));
im_50(48:143, 40:120) = shrinked;

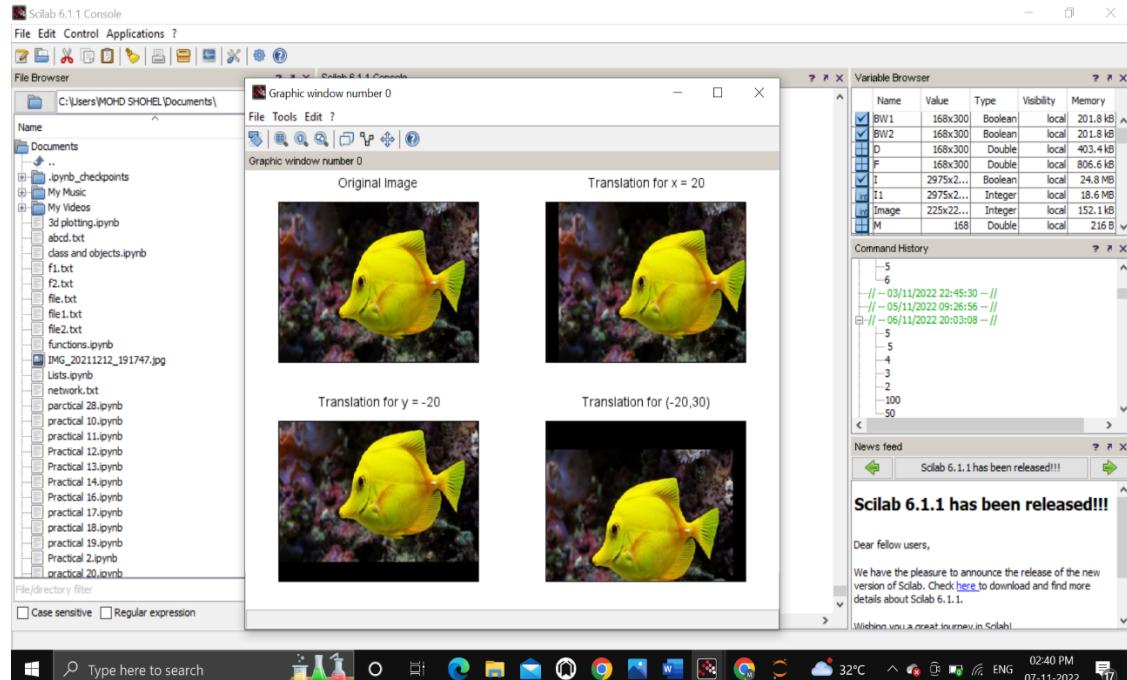
subplot(121), title('Original Image'), imshow(rgb2gray(s_img));
subplot(122), title('Image Shrunk by 50%'), imshow(im_50);

% (e)
```

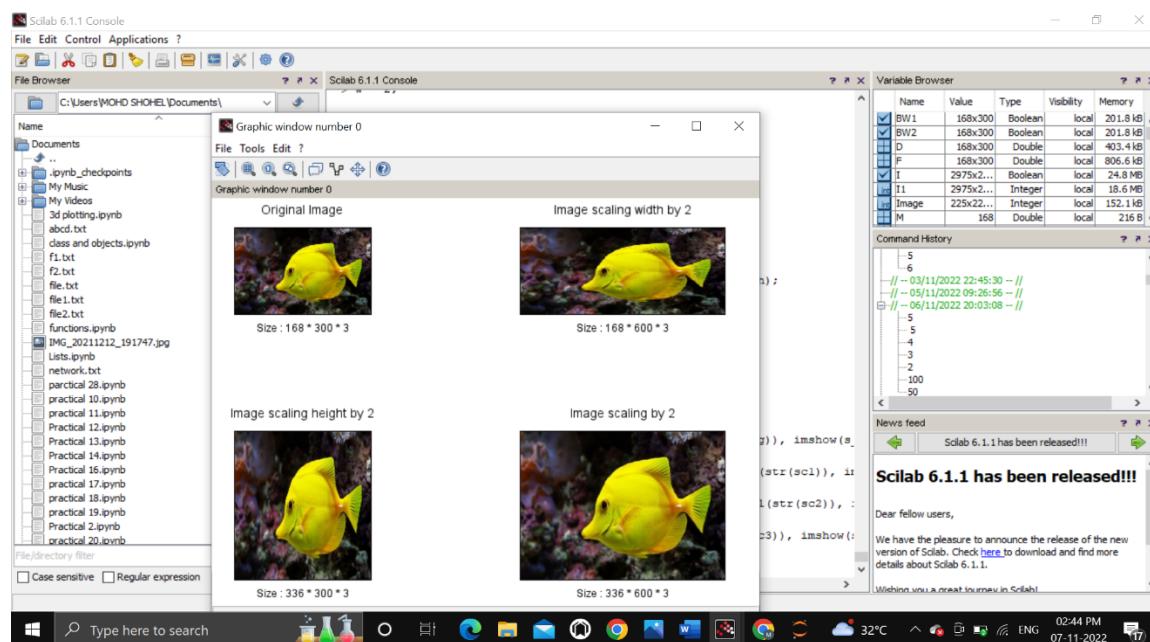
```
f = 2;
im2 = imresize(s_img, f);
subplot(121), title('Original Image'), imshow(s_img);
subplot(122), title('Image zoomed by 200%'), imshow(im2(96:287, 81:241, :));
```

Output :

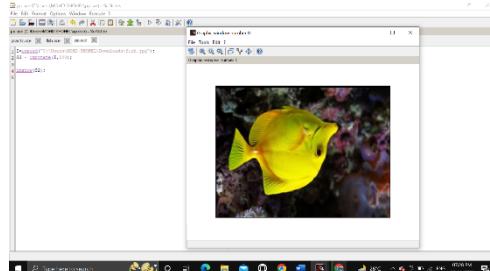
(a)



(b)

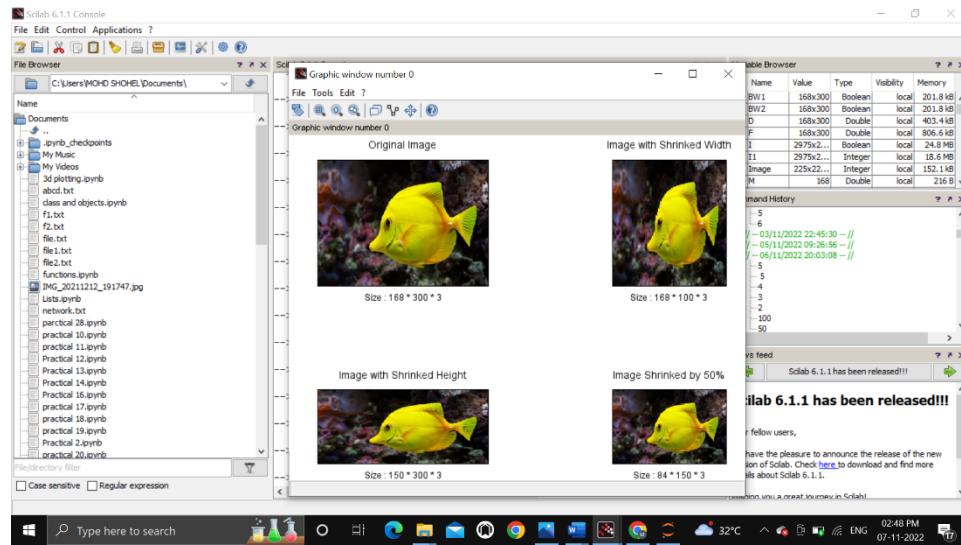


(c)



```
>> ques6c
Enter the rotation degree: 75
```

(d)



(e)





Question 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

Solution :

Code -

```
% (a)
im1 = imread('C:\Users\MOHD SHOHEL\Downloads\fish.jpg');
f = fspecial('gaussian', [8, 8], 2);

subplot(121), title('Noisy Image'), imshow(im1);
subplot(122), title('Filtered Image'), imshow(imfilter(im1, f));
% (b)
im2 = rgb2gray(imread('C:\Users\MOHD SHOHEL\Downloads\fish.jpg'));
im3 = imnoise(im2, 'salt & pepper', 0.3);

subplot(131), title('Original Image'), imshow(im2);
subplot(132), title('Salt & Pepper Noised Image'), imshow(im3);
subplot(133), title('Filtered Image'), imshow(immedian(im2,3));

% (c)
```

```

im1 = imread('C:\Users\MOHD SHOHEL\Downloads\fish.jpg');
im2 = imnoise(im1, 'gaussian');
f = fspecial('average', 3);

subplot(131), title('Original Image'), imshow(im1);
subplot(132), title('Gaussian Noised Image'), imshow(im2);
subplot(133), title('Filtered Image'), imshow(imfilter(im1, f));

```

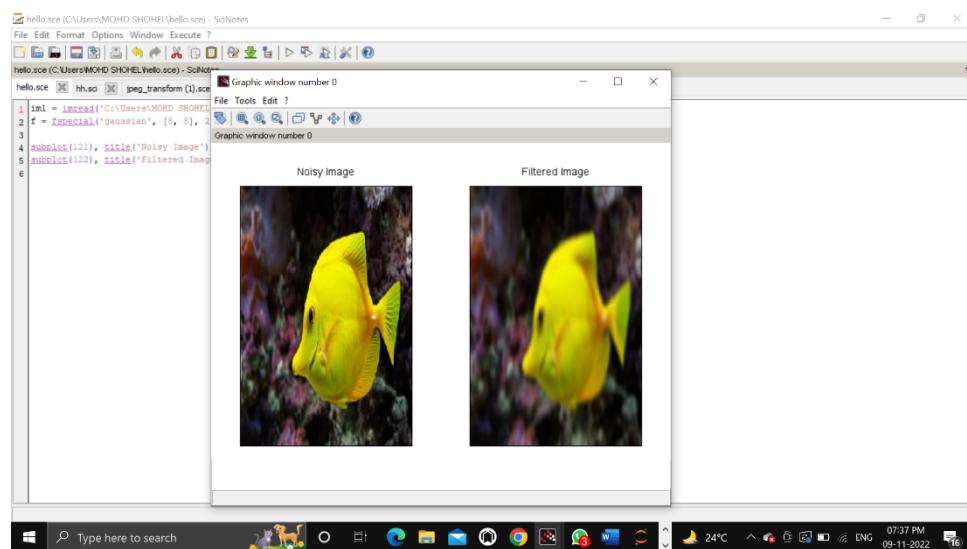
```

% (d)
RGB=imread("C:\Users\MOHD SHOHEL\Downloads\nature1.jpg");
I = rgb2gray(RGB);
imshow(I);
title('Original GrayScale Image');
J = imnoise(I,'gaussian',0,0.025);
figure();
imshow(J);
title('Portion of the Image with Added Gaussian Noise');
K = wiener2(J,[5 5]);
figure();
imshow(K);
title('Portion of the Image with Noise Removed by Wiener Filter');

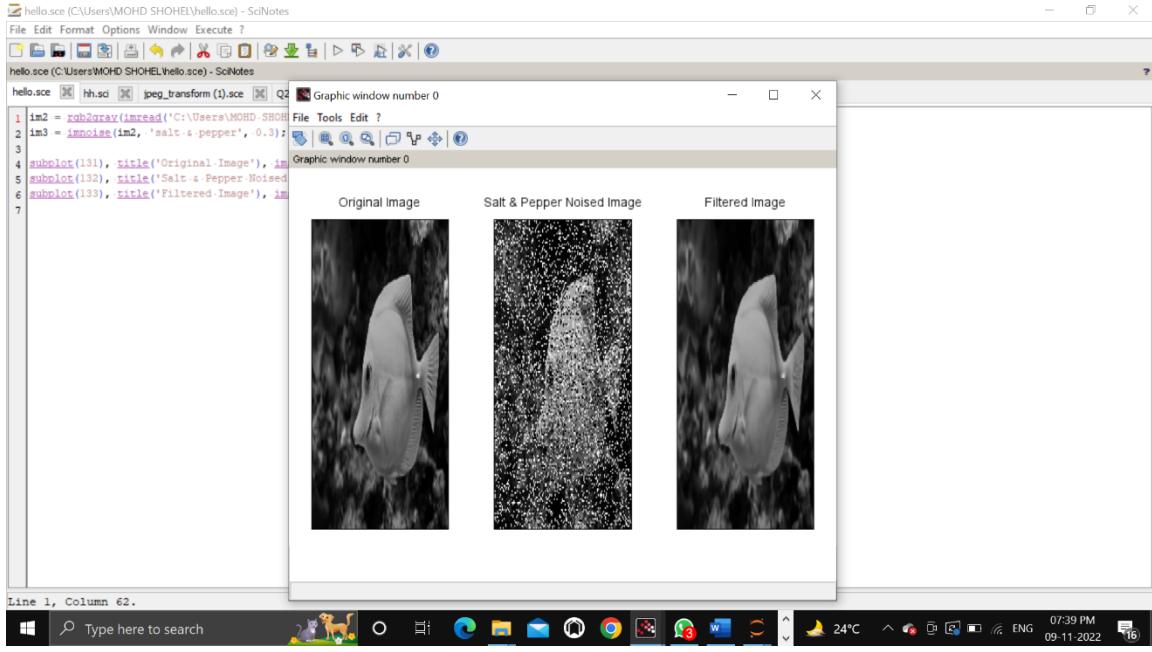
```

Output:

(a)



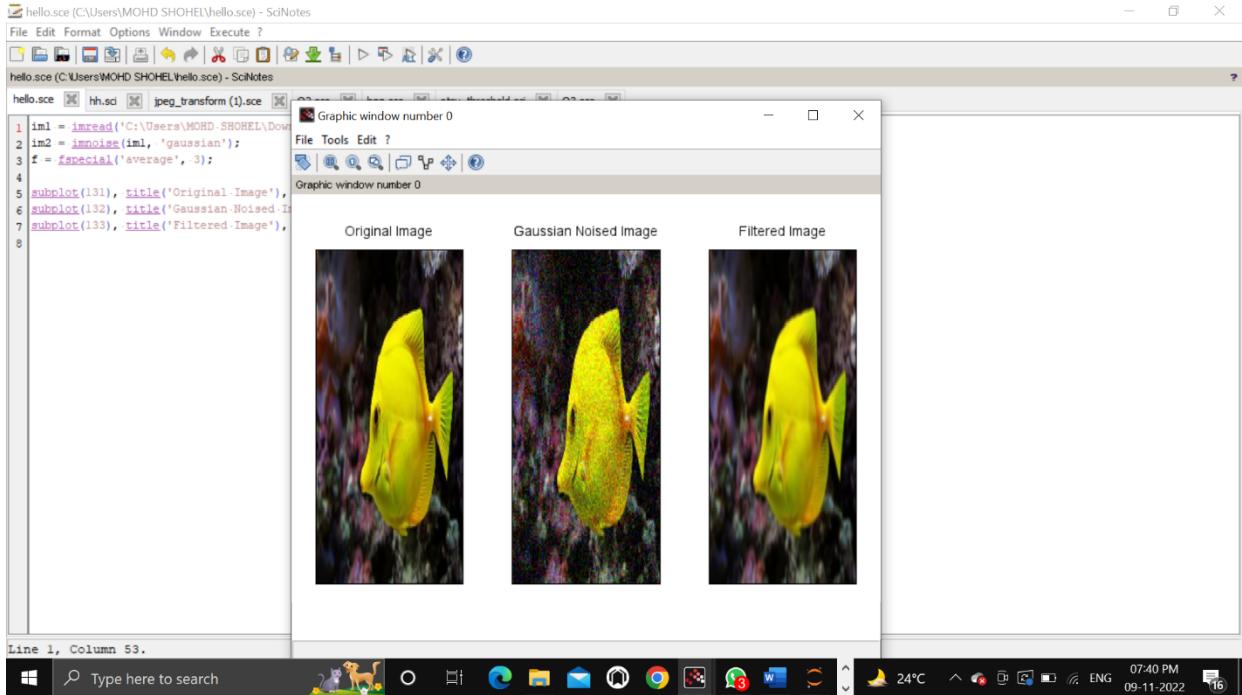
(b)



Line 1, Column 62.

Type here to search 24°C ENG 07:39 PM 09-11-2022

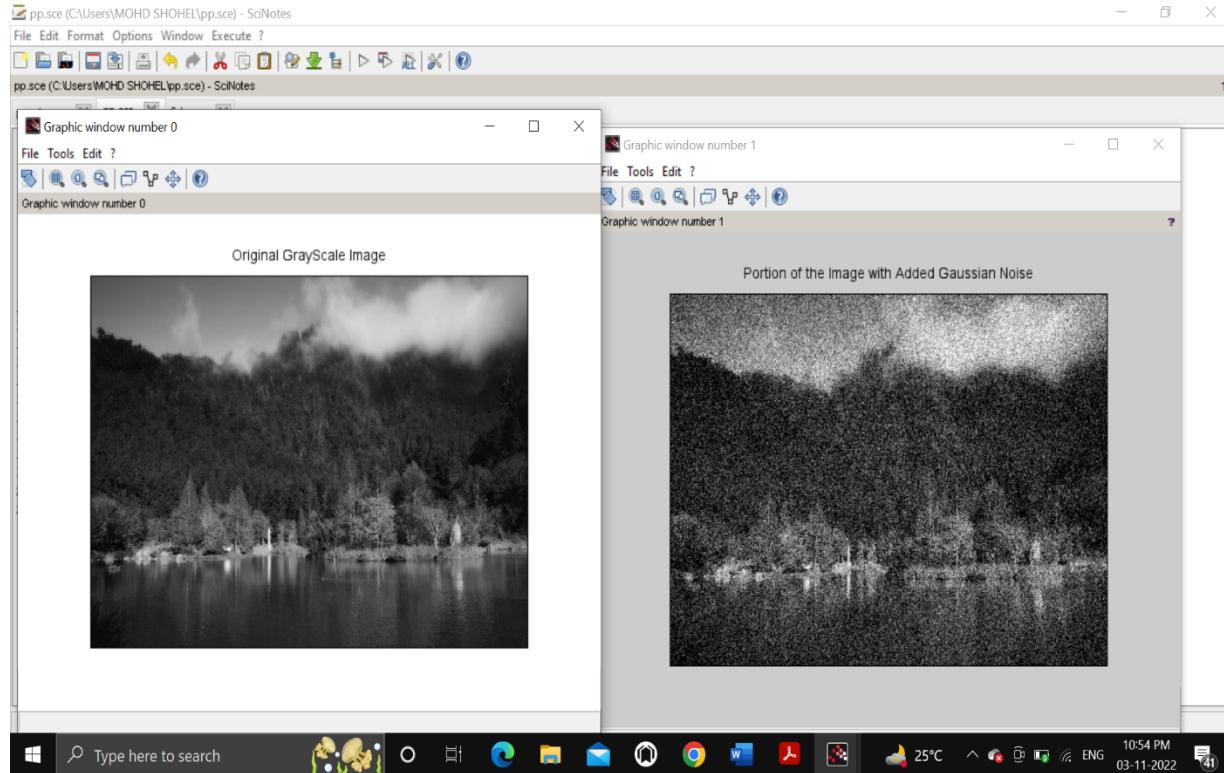
(c)



Line 1, Column 53.

Type here to search 24°C ENG 07:40 PM 09-11-2022

(d)



Question 8: Write and execute programs to use spatial low pass and high pass filters.

Solution :

Code -

```
//High pass
i1= imread('C:\Users\MOHD SHOHEL\Downloads\fish.jpg');
g_filter = fspecial('gaussian');
i2 = imfilter(i1, g_filter);
g_filter2 = fspecial('gaussian',[8,8], 10);
i3 = imfilter(i1, g_filter2);
g_filter3 = fspecial('gaussian',[25,25], 31);
i4 = imfilter(i1, g_filter3);
subplot(2,2,1), title('Original Image'), imshow(i1);
subplot(2,2,2), title('Default Gaussian kernel'), imshow(i2);
subplot(2,2,3), title('Gaussian kernel with 8 * 8 with sigma = 10'), imshow(i3);
subplot(2,2,4), title('Gaussian kernel with 25 * 25 with sigma = 31'), imshow(i4);
```

//low pass

```
i1=imread("C:\Users\MOHD SHOHEL\Downloads\fish.jpg");
```

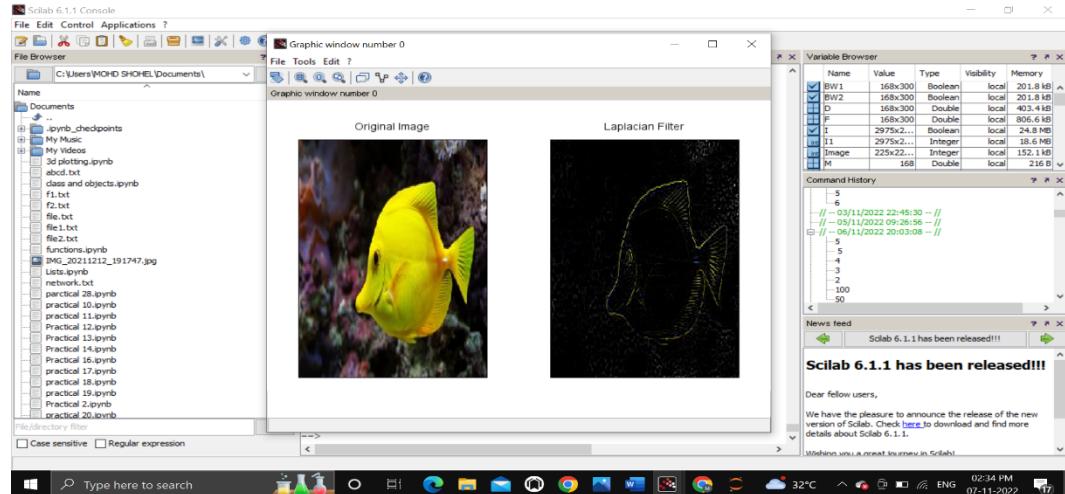
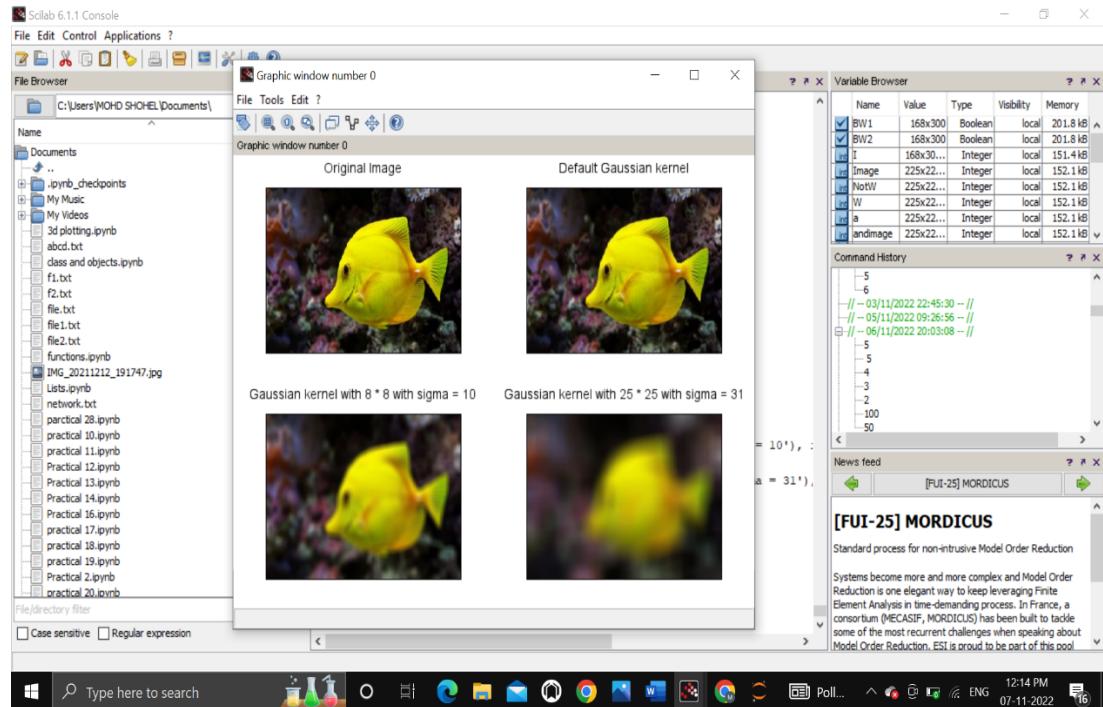
```

l_filter = fspecial('laplacian');
i2 = imfilter(i1, l_filter);

subplot(1,2,1), title('Original Image'), imshow(i1);
subplot(1,2,2), title('Laplacian Filter'), imshow(i2);

```

Output:



Question 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

Solution :

Code -

```
% (a)
img = rgb2gray(imread('C:\Users\MOHD SHOHEL\Downloads\fish.jpg'));
ft_img = fft(double(img));

subplot(1,2,1), title('Original Image'), imshow(img);
subplot(1,2,2), title('Direct Fourier Transformed Image'),imshow(ft_img);

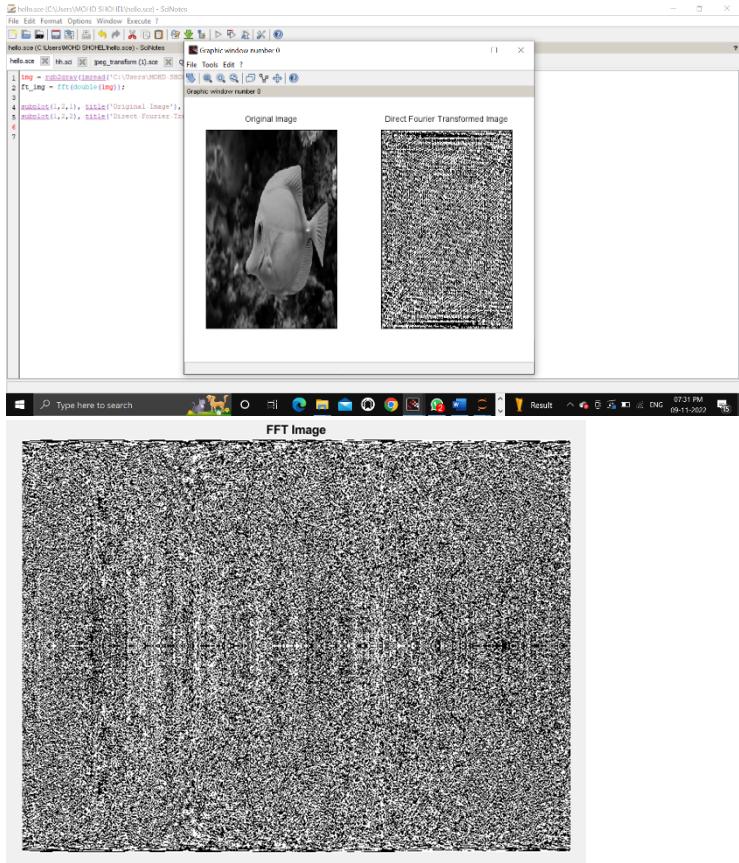
% (b)
// Butterworth Filters
G11 = mkfftfilter(img, 'butterworth1', 0.3);
H11 = 1 - G11;

subplot(121), title('DFT Butterworth Low Pass Image'), imshow(G11);
subplot(122), title('DFT Butterworth High Pass Image'),imshow(H11);
(c)
S2 = ft_img .* fftshift(G11);
bwh_l = uint8(ifft(S2));
S2 = ft_img .* fftshift(H11);
bwh_h = uint8(ifft(S2));

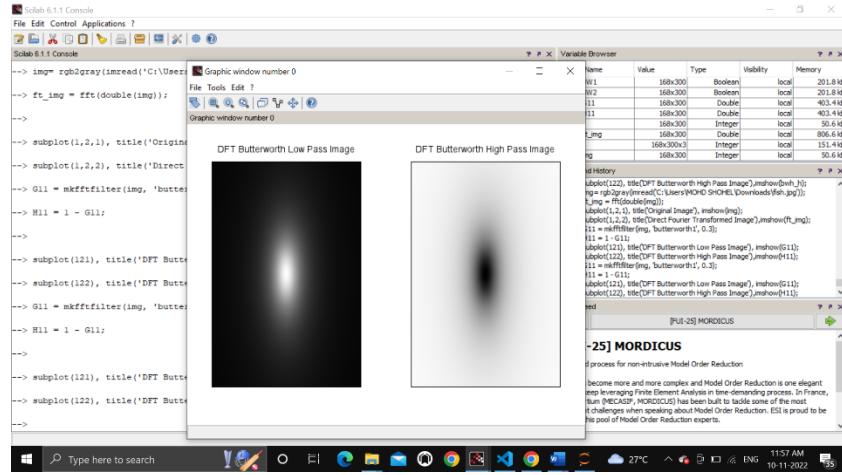
subplot(121), title('DFT Butterworth Low Pass Image'), imshow(bwh_l);
subplot(122), title('DFT Butterworth High Pass Image'),imshow(bwh_h);
```

Output:

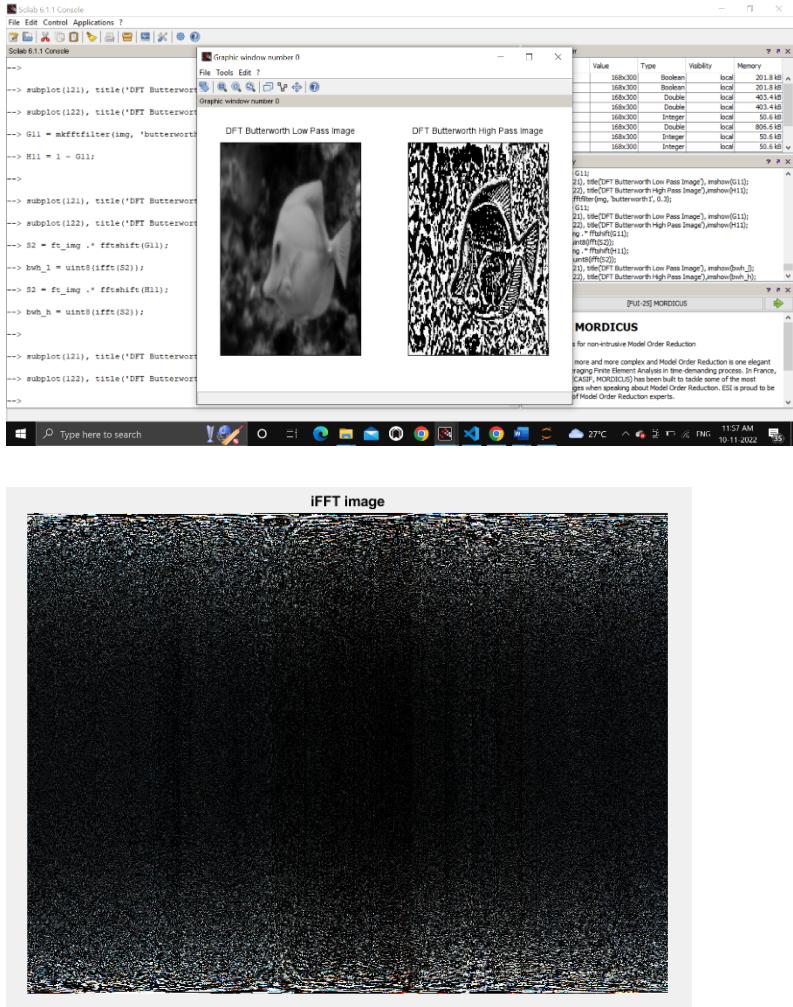
(a)



(b)



(c)



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

Solution :

Code -

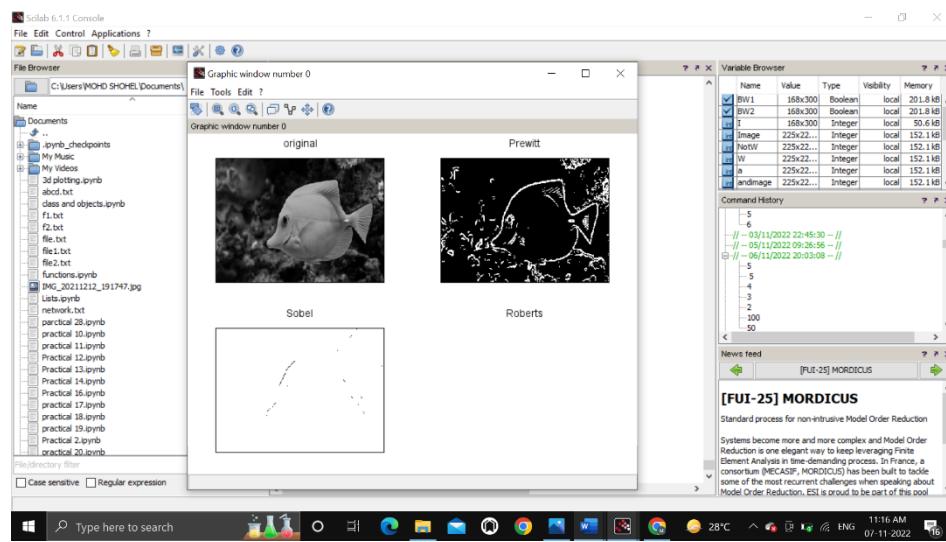
```

i= imread('C:\Users\MOHD SHOHEL\Downloads\fish.jpg');
I=rgb2gray(i);
BW1=edge(I,"prewitt");
BW2=edge(I,"sobel");
BW3=edge(I,"roberts");
subplot(2,2,1);
imshow(I);
title("original");
subplot(2,2,2);

```

```
imshow(BW1);
title("Prewitt");
subplot(2,2,3);
imshow(BW2);
title("Sobel");
subplot(2,2,4);
imshow(BW3);
title("Roberts");
```

Output:



Question 11: Write and execute a program for image morphological operations erosion and dilation.

Solution :

Code -

```
I1=imread('C:\Users\MOHD SHOHEL\Downloads\grey.jpg');
I=im2bw(I1,0.5);
se=imcreatese('cross',3,3);
subplot(2, 3, 1);
imshow(I);
title("Original image");
```

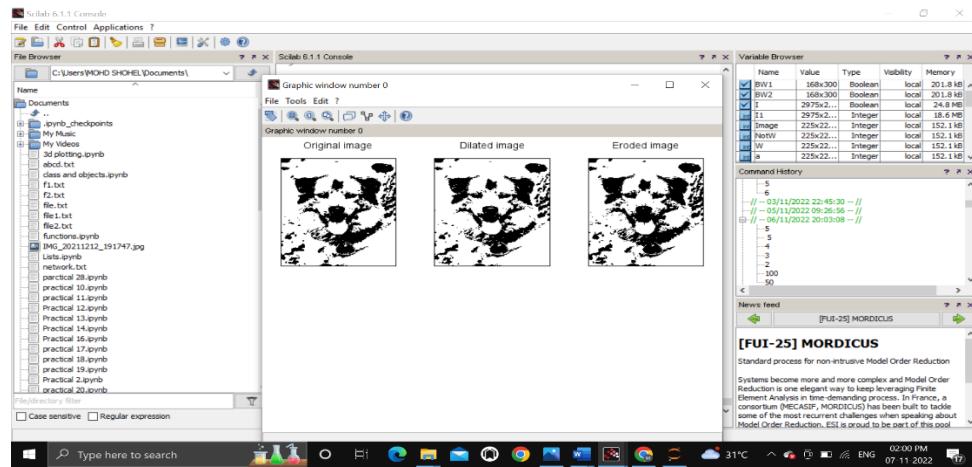
```
dilate = imdilate(I, se);
subplot(2, 3, 2);
imshow(dilate);
title("Dilated image");
```

```

erode = imerode(I, se);
subplot(2, 3, 3);
imshow(erode);
title("Eroded image");

```

Output:



Thank You