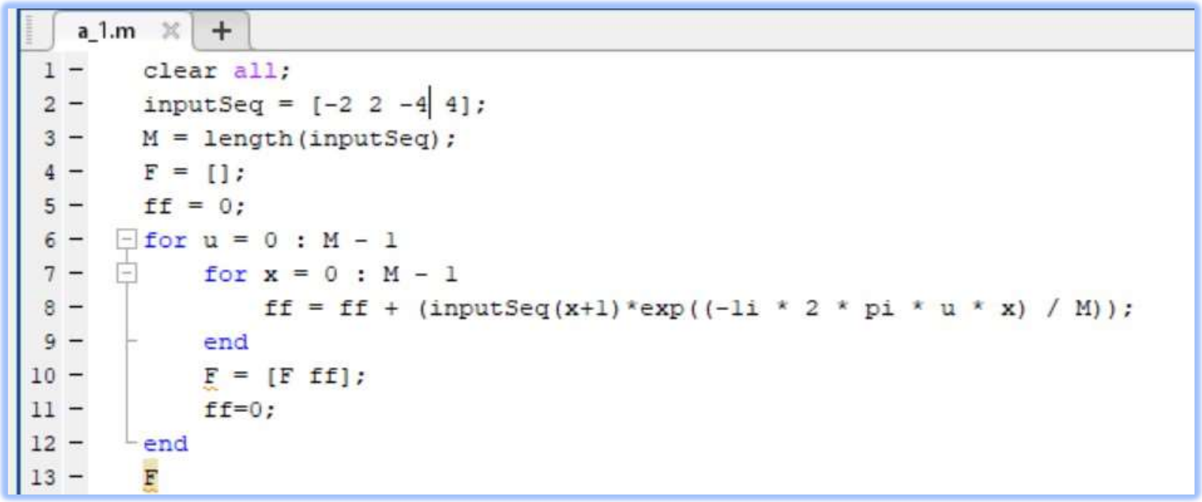
LAB - 9

Name	Keval D Gandevia
Roll Number	CE046
ID	19CEUEG017
Subject	Image Processing

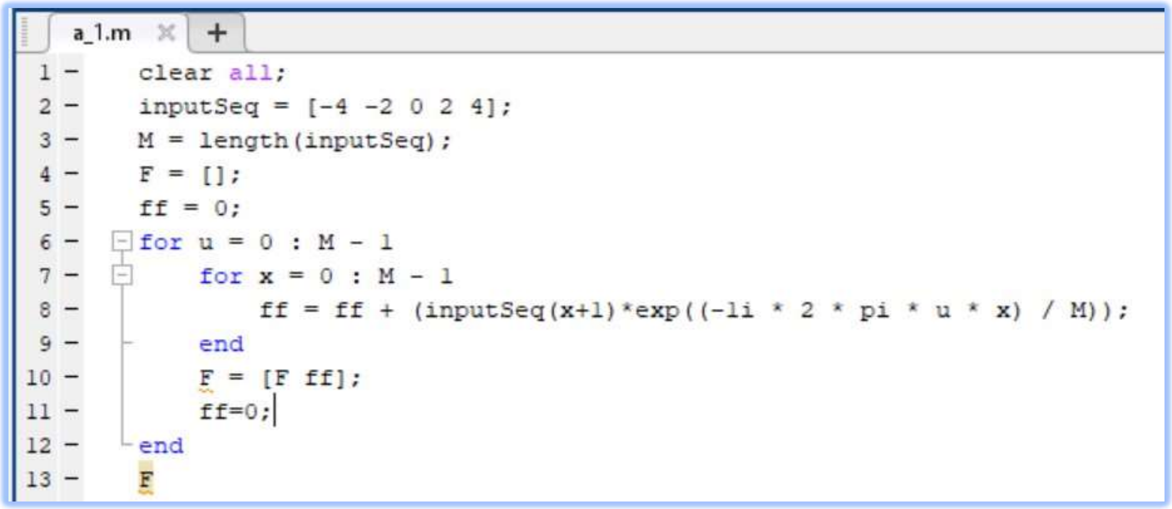
Aim: Fourier Transform and frequency domain analysis in image processing.

Q. 1: Preliminary: Implement DFT in MATLAB for 1 D array and use your script to perform the following DFT calculations: a. Input Sequence = [-2 2 -4 4] b. Input Sequence = [-4 -2 0 2 4].

- **Code:**

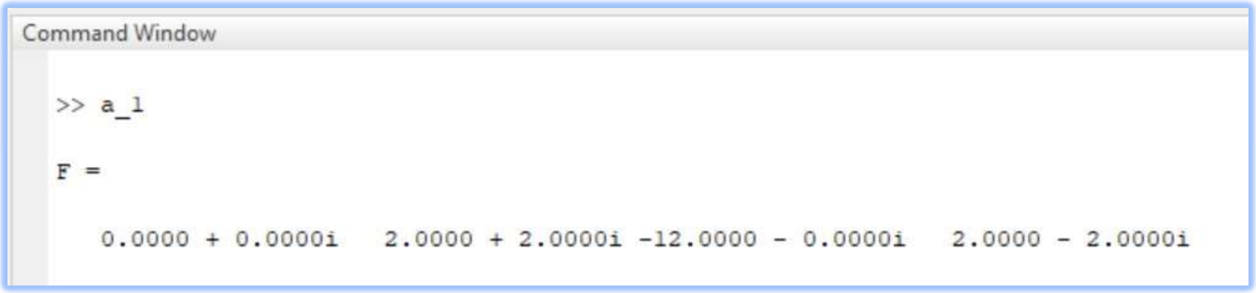


```
clear all;
inputSeq = [-2 2 -4 4];
M = length(inputSeq);
F = [];
ff = 0;
for u = 0 : M - 1
    for x = 0 : M - 1
        ff = ff + (inputSeq(x+1)*exp((-1i * 2 * pi * u * x) / M));
    end
    F = [F ff];
    ff=0;
end
F
```

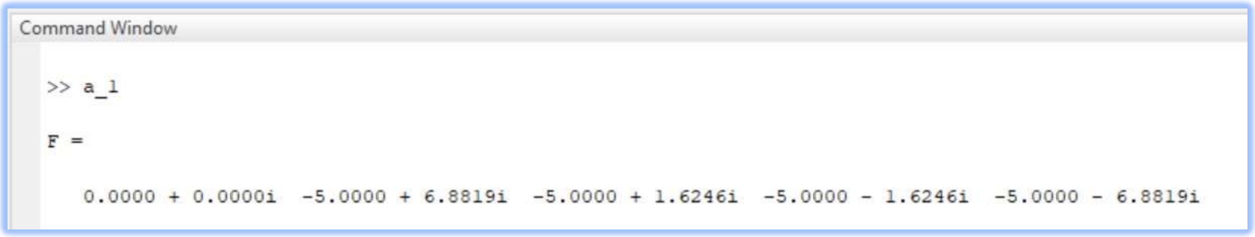


```
clear all;
inputSeq = [-4 -2 0 2 4];
M = length(inputSeq);
F = [];
ff = 0;
for u = 0 : M - 1
    for x = 0 : M - 1
        ff = ff + (inputSeq(x+1)*exp((-1i * 2 * pi * u * x) / M));
    end
    F = [F ff];
    ff=0;
end
F
```

❖ **Output:**



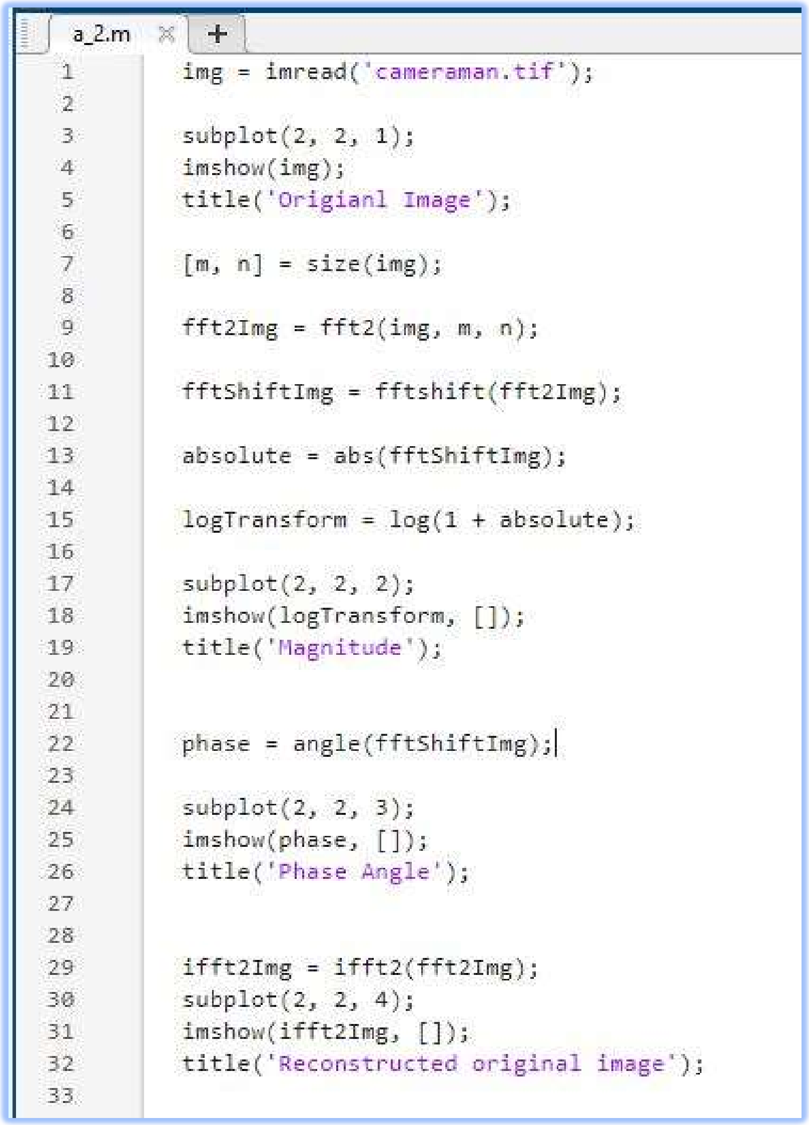
```
>> a_1

F =

   0.0000 + 0.0000i   2.0000 + 2.0000i -12.0000 - 0.0000i   2.0000 - 2.0000i
```



```
>> a_1

F =

   0.0000 + 0.0000i  -5.0000 + 6.8819i  -5.0000 + 1.6246i  -5.0000 - 1.6246i  -5.0000 - 6.8819i
```

Q. 2: The MATLAB routines for computing the 2-D DFT and the inverse 2-D DFT are the routines fft2 and ifft2. Using the image file cameraman.tif. Read in the image and perform DFT computations such that you get the following results.

- **Code:**



```
img = imread('cameraman.tif');

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

[m, n] = size(img);

fft2Img = fft2(img, m, n);

fftShiftImg = fftshift(fft2Img);

absolute = abs(fftShiftImg);

logTransform = log(1 + absolute);

subplot(2, 2, 2);
imshow(logTransform, []);
title('Magnitude');


phase = angle(fftShiftImg);

subplot(2, 2, 3);
imshow(phase, []);
title('Phase Angle');


ifft2Img = ifft2(fft2Img);
subplot(2, 2, 4);
imshow(ifft2Img, []);
title('Reconstructed original image');

```

❖ Output:

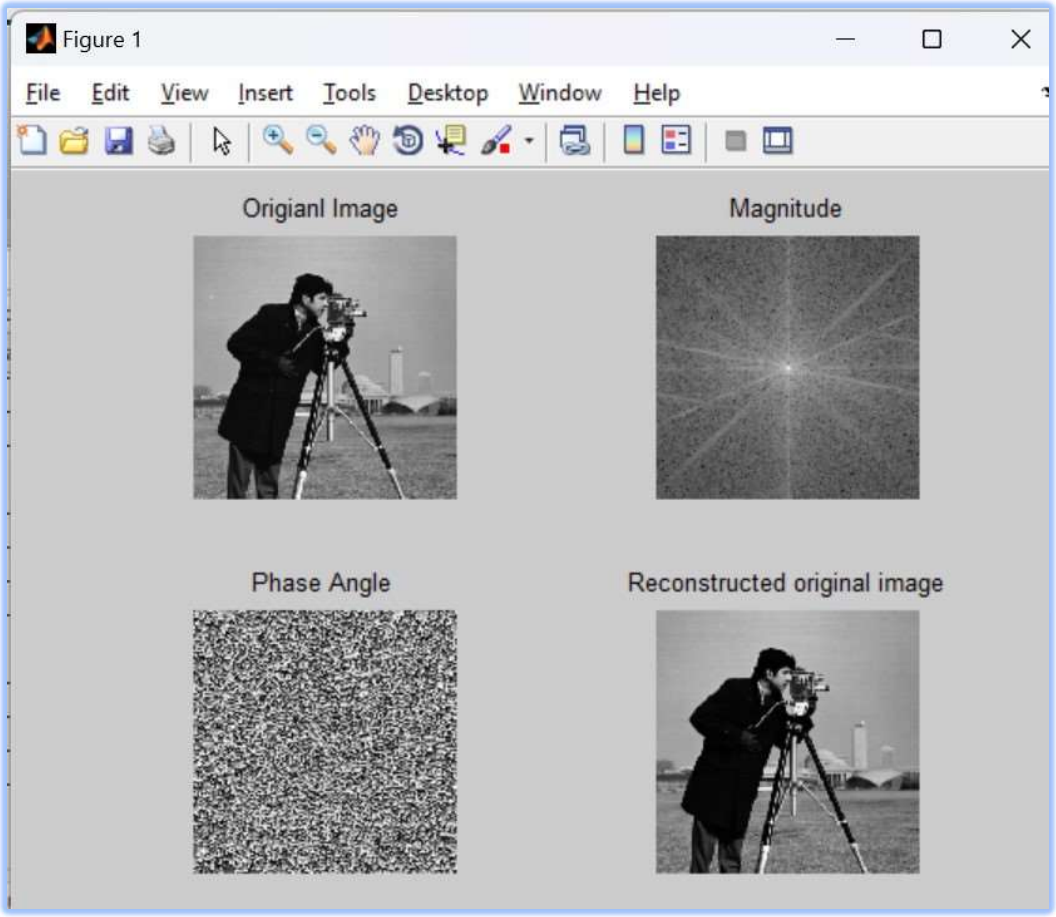
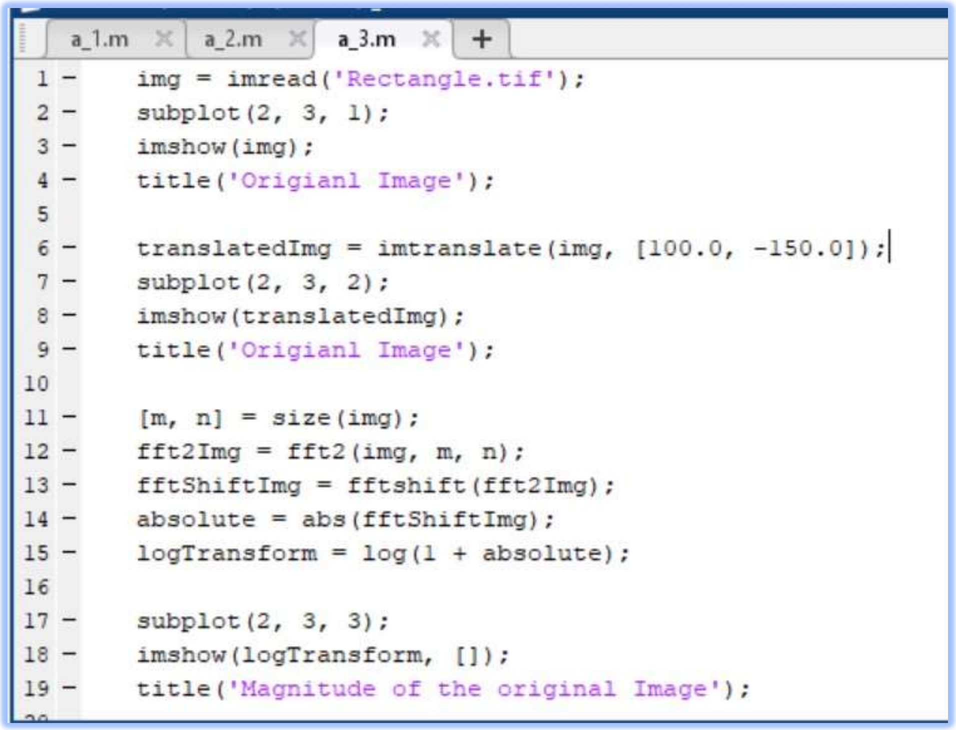


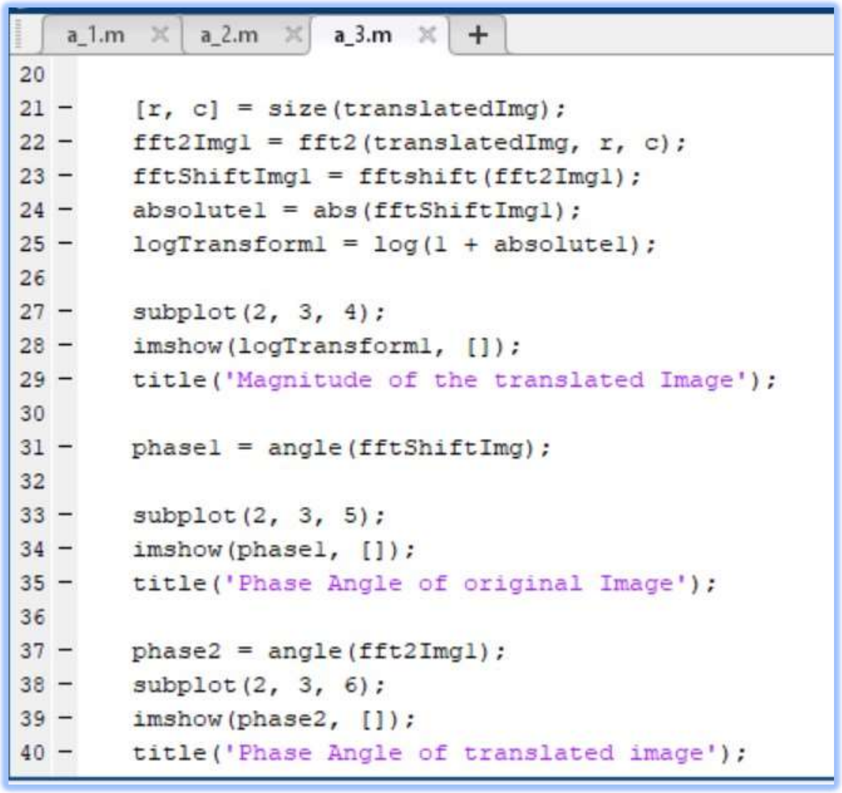
Figure 1
File Edit View Insert Tools Desktop Window Help
Origianl Image
Magnitude
Phase Angle
Reconstructed original image

Q. 3: Use the image Rectangle.tif, validate a). If we translate the image using imtranslate ( ) by (x0, y0) = (100.0, - 150.0), then magnitude spectrum of the original and translated image remains the same. However phase angle changes. You should get approximately the following result b). If we rotate the image using imrotate ( ) by 45 degrees in clockwise direction, then magnitude spectrum as well as phase of the rotated image changes. You should get approximately the following result.

❖ Code with imtranslate:

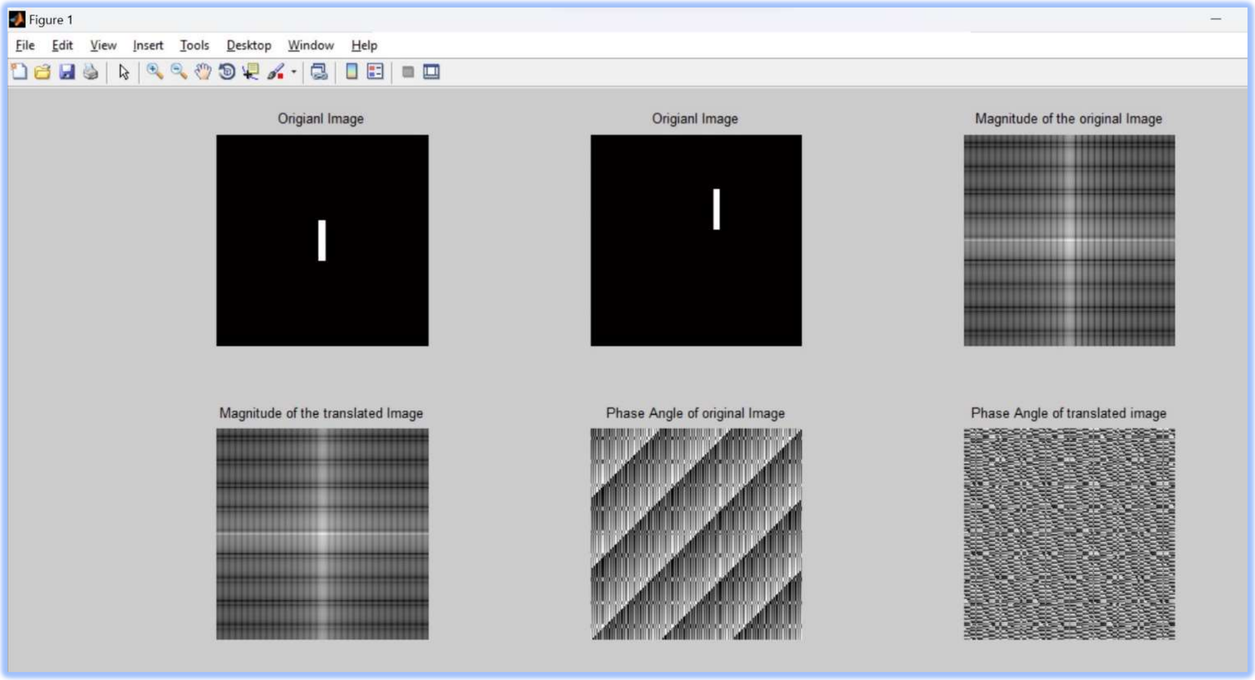


```
img = imread('Rectangle.tif');
subplot(2, 3, 1);
imshow(img);
title('Origianl Image');

translatedImg = imtranslate(img, [100.0, -150.0]);
subplot(2, 3, 2);
imshow(translatedImg);
title('Origianl Image');

[m, n] = size(img);
fft2Img = fft2(img, m, n);
fftShiftImg = fftshift(fft2Img);
absolute = abs(fftShiftImg);
logTransform = log(1 + absolute);

subplot(2, 3, 3);
imshow(logTransform, []);
title('Magnitude of the original Image');
```



```

[r, c] = size(translatedImg);
fft2Img1 = fft2(translatedImg, r, c);
fftShiftImg1 = fftshift(fft2Img1);
absolute1 = abs(fftShiftImg1);
logTransform1 = log(1 + absolute1);

subplot(2, 3, 4);
imshow(logTransform1, []);
title('Magnitude of the translated Image');

phase1 = angle(fftShiftImg);

subplot(2, 3, 5);
imshow(phase1, []);
title('Phase Angle of original Image');

phase2 = angle(fft2Img1);
subplot(2, 3, 6);
imshow(phase2, []);
title('Phase Angle of translated image');
```

❖ Output:



Origianl Image

Origianl Image

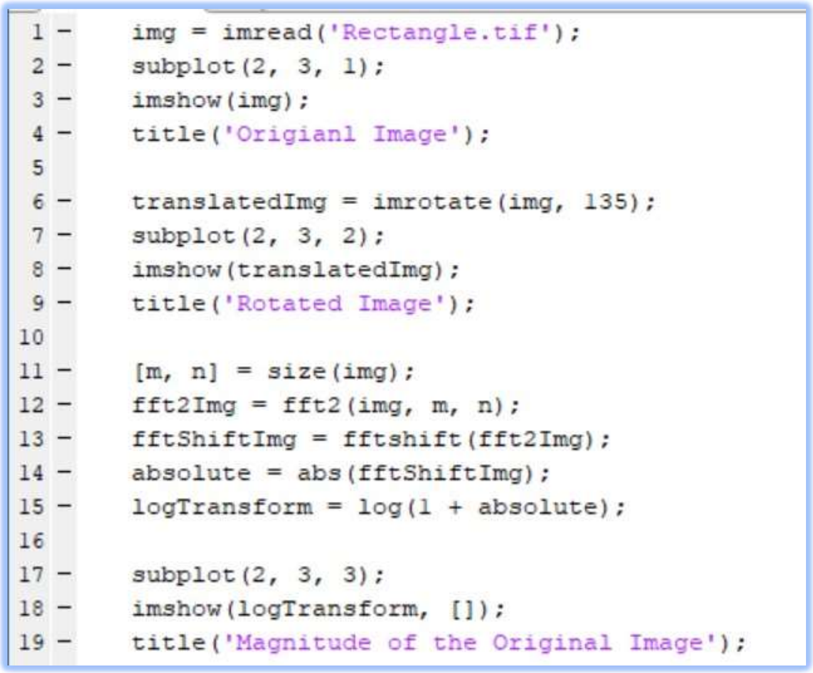
Magnitude of the original Image

Magnitude of the translated Image

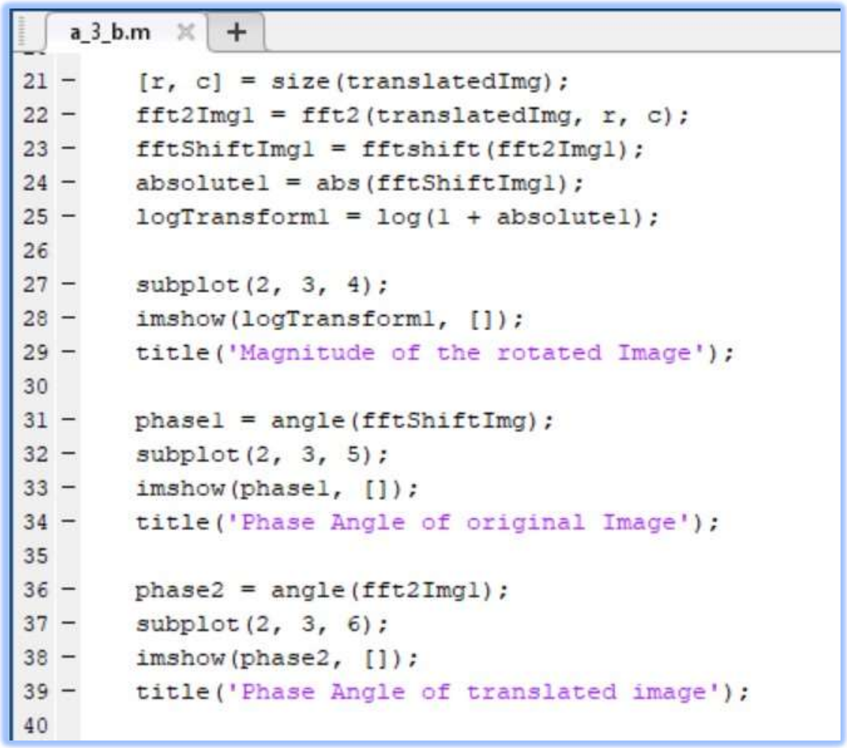
Phase Angle of original Image

Phase Angle of translated image

❖ Code with imrotate:



```
img = imread('Rectangle.tif');
subplot(2, 3, 1);
imshow(img);
title('Origianl Image');

translatedImg = imrotate(img, 135);
subplot(2, 3, 2);
imshow(translatedImg);
title('Rotated Image');

[m, n] = size(img);
fft2Img = fft2(img, m, n);
fftShiftImg = fftshift(fft2Img);
absolute = abs(fftShiftImg);
logTransform = log(1 + absolute);

subplot(2, 3, 3);
imshow(logTransform, []);
title('Magnitude of the Original Image');
```



```
    [r, c] = size(translatedImg);
    fft2Img1 = fft2(translatedImg, r, c);
    fftShiftImg1 = fftshift(fft2Img1);
    absolute1 = abs(fftShiftImg1);
    logTransform1 = log(1 + absolute1);

    subplot(2, 3, 4);
    imshow(logTransform1, []);
    title('Magnitude of the rotated Image');

    phase1 = angle(fftShiftImg);
    subplot(2, 3, 5);
    imshow(phase1, []);
    title('Phase Angle of original Image');

    phase2 = angle(fft2Img1);
    subplot(2, 3, 6);
    imshow(phase2, []);
    title('Phase Angle of translated image');

```

❖ Output:

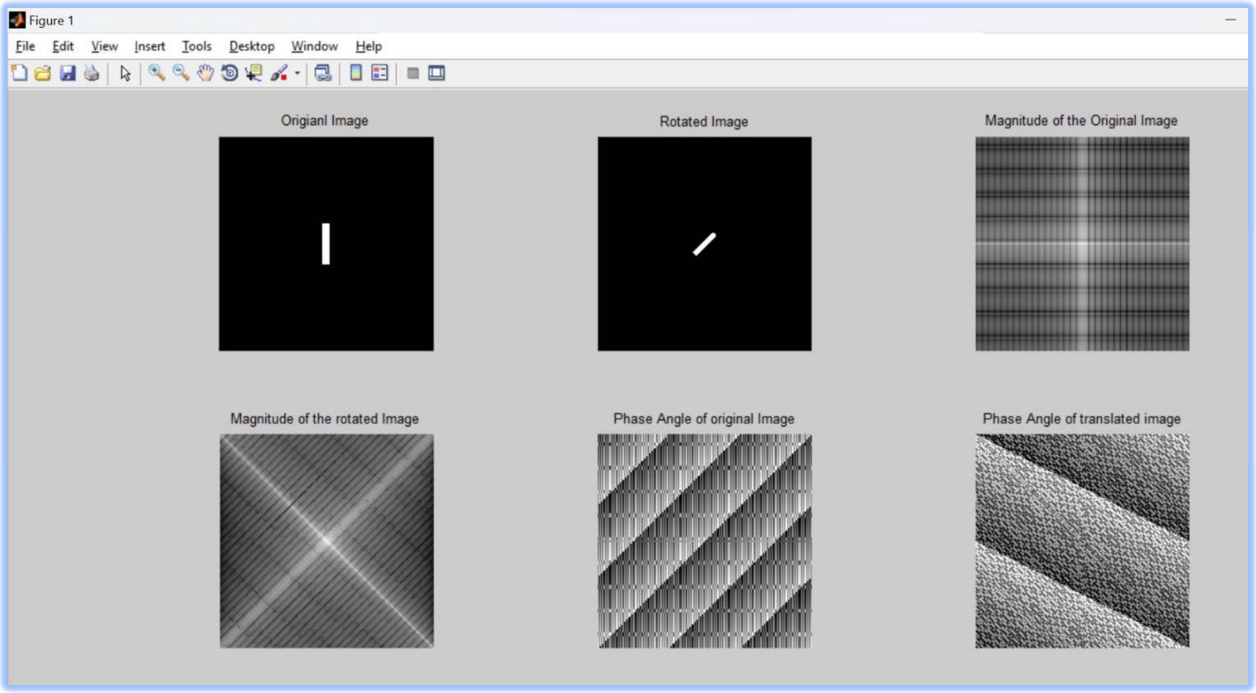
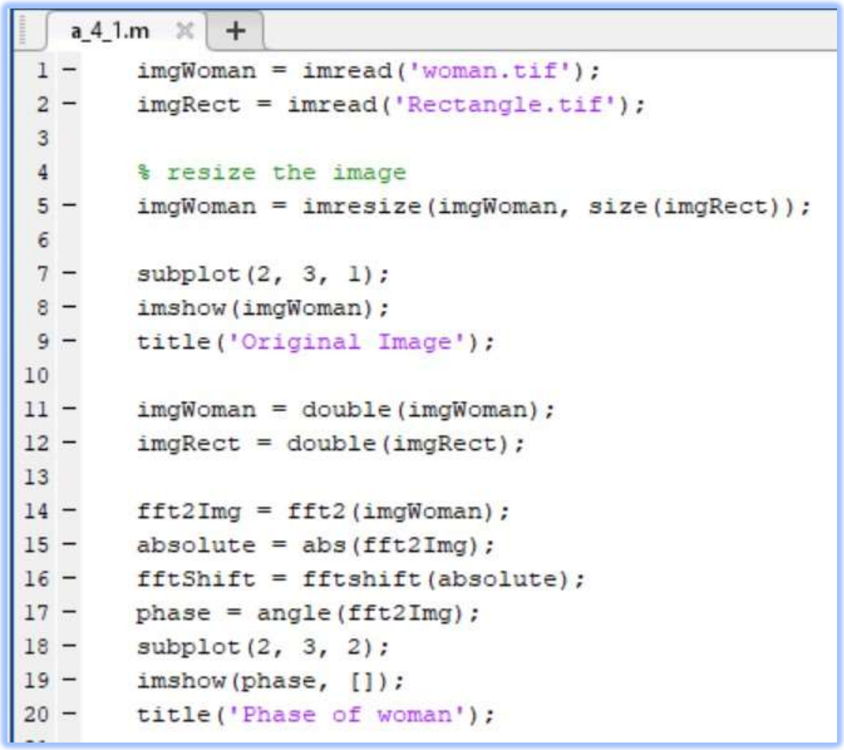


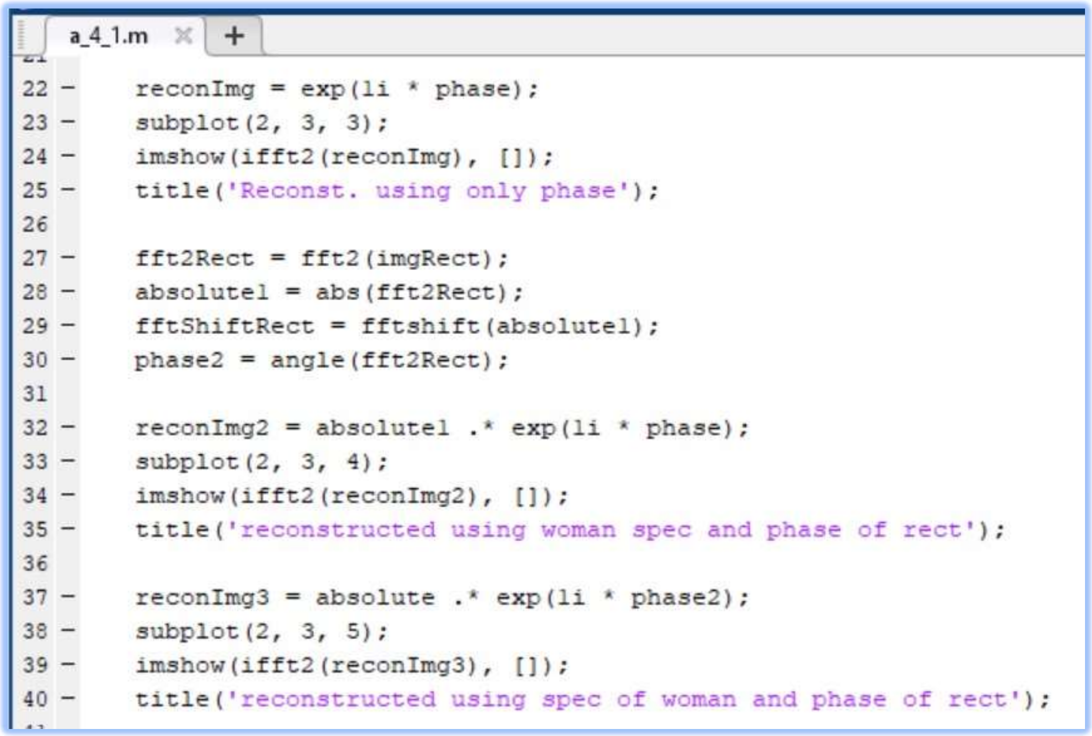
Figure 1
File Edit View Insert Tools Desktop Window Help
Origianl Image
Rotated Image
Magnitude of the Original Image
Magnitude of the rotated Image
Phase Angle of original Image
Phase Angle of translated image

Q. 4: 4. Importance of DFT phase Using the image woman.tif and Rectangle.tif, perform 2D DFT and IDFT computations to get the following results. A). Phase angle of woman. B). Woman reconstructed using only the phase angle. C). Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle. D). Reconstruction using the phase of the rectangle and the spectrum of the woman. E). Reconstruction using woman spectrum and phase. It should match the original image.

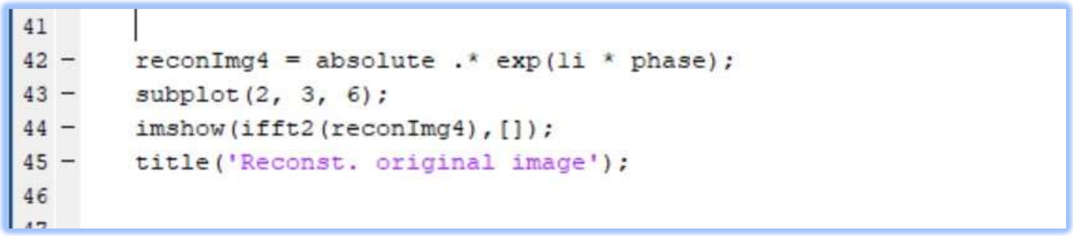
- **Code:**



```
imgWoman = imread('woman.tif');
imgRect = imread('Rectangle.tif');

% resize the image
imgWoman = imresize(imgWoman, size(imgRect));

subplot(2, 3, 1);
imshow(imgWoman);
title('Original Image');

imgWoman = double(imgWoman);
imgRect = double(imgRect);

fft2Img = fft2(imgWoman);
absolute = abs(fft2Img);
fftShift = fftshift(absolute);
phase = angle(fft2Img);
subplot(2, 3, 2);
imshow(phase, []);
title('Phase of woman');
```



```
reconImg = exp(1i * phase);
subplot(2, 3, 3);
imshow(ifft2(reconImg), []);
title('Reconst. using only phase');

fft2Rect = fft2(imgRect);
absolute1 = abs(fft2Rect);
fftShiftRect = fftshift(absolute1);
phase2 = angle(fft2Rect);

reconImg2 = absolute1 .* exp(1i * phase);
subplot(2, 3, 4);
imshow(ifft2(reconImg2), []);
title('reconstructed using woman spec and phase of rect');

reconImg3 = absolute .* exp(1i * phase2);
subplot(2, 3, 5);
imshow(ifft2(reconImg3), []);
title('reconstructed using spec of woman and phase of rect');
```



```

reconImg4 = absolute .* exp(1i * phase);
subplot(2, 3, 6);
imshow(ifft2(reconImg4),[]);
title('Reconst. original image');

```

❖ Output:

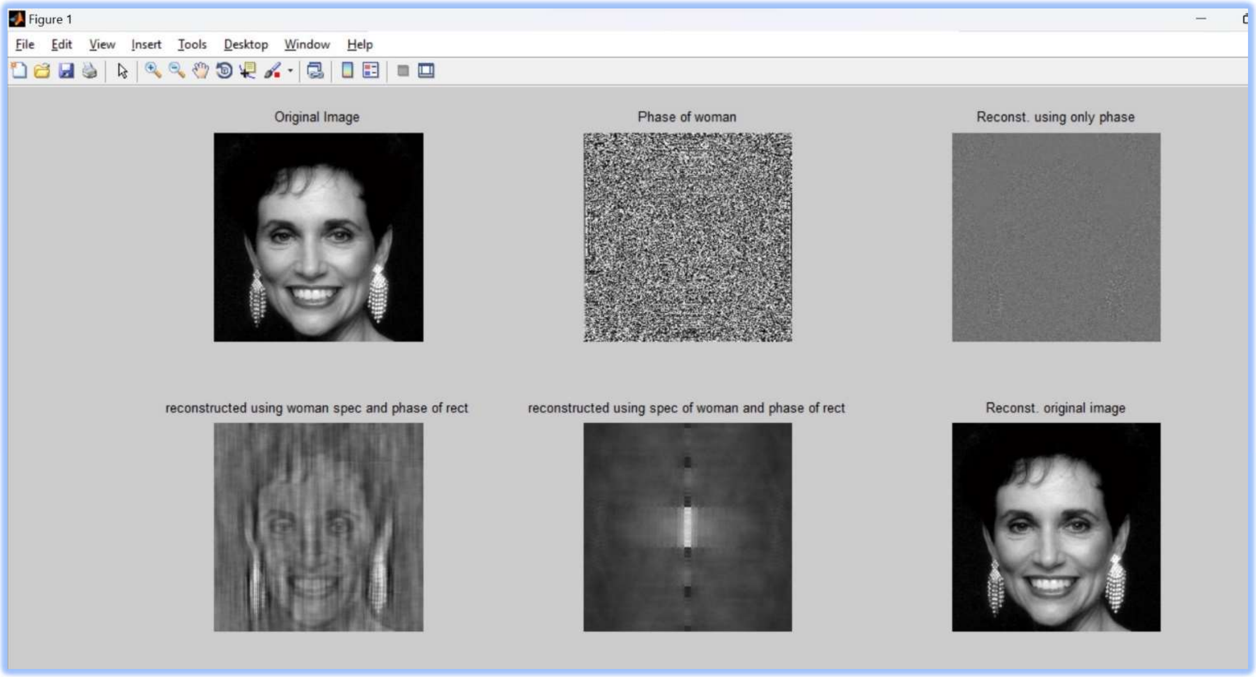


Figure 1
File Edit View Insert Tools Desktop Window Help
Original Image
Phase of woman
Reconst. using only phase
reconstructed using woman spec and phase of rect
reconstructed using spec of woman and phase of rect
Reconst. original image