LAB – 3

Name: Gandevia Keval Dharmeshbhai

Sem: VII

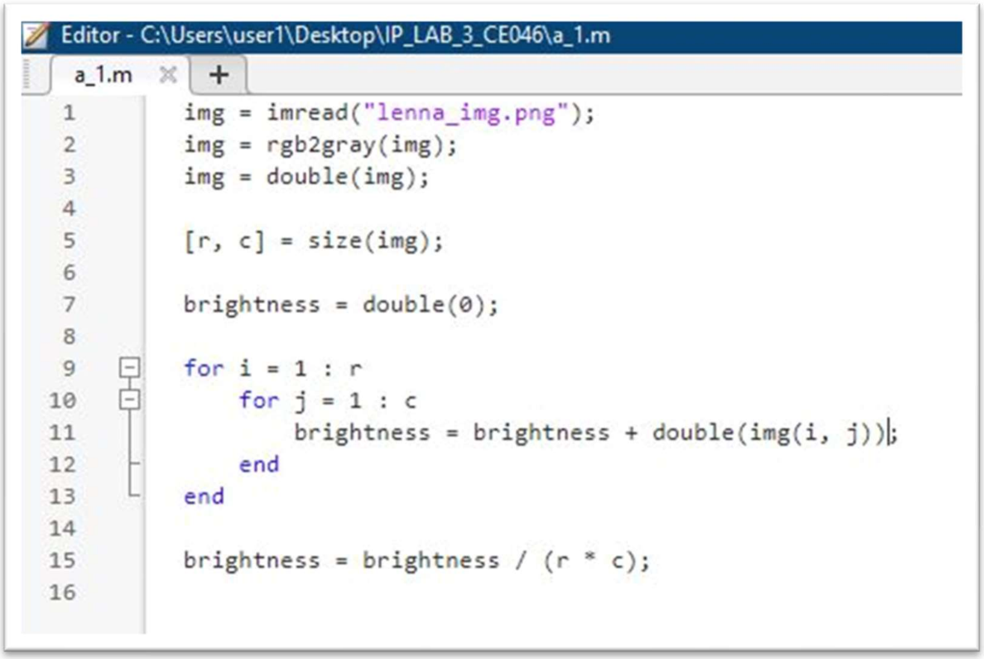
Roll No: CE046

Subject: Image Processing

Q. 1: Calculate the brightness and contrast of the images.

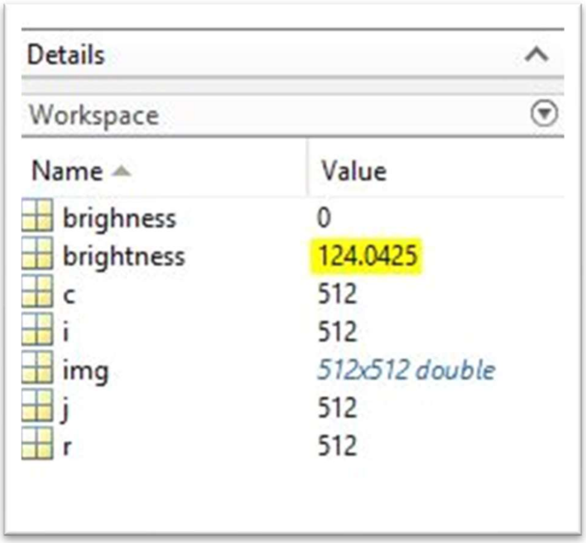
❖ High brightness image

- Code:



```
img = imread("lenna_img.png");
img = rgb2gray(img);
img = double(img);

[r, c] = size(img);

brightness = double(0);

for i = 1 : r
    for j = 1 : c
        brightness = brightness + double(img(i, j));
    end
end

brightness = brightness / (r * c);

```

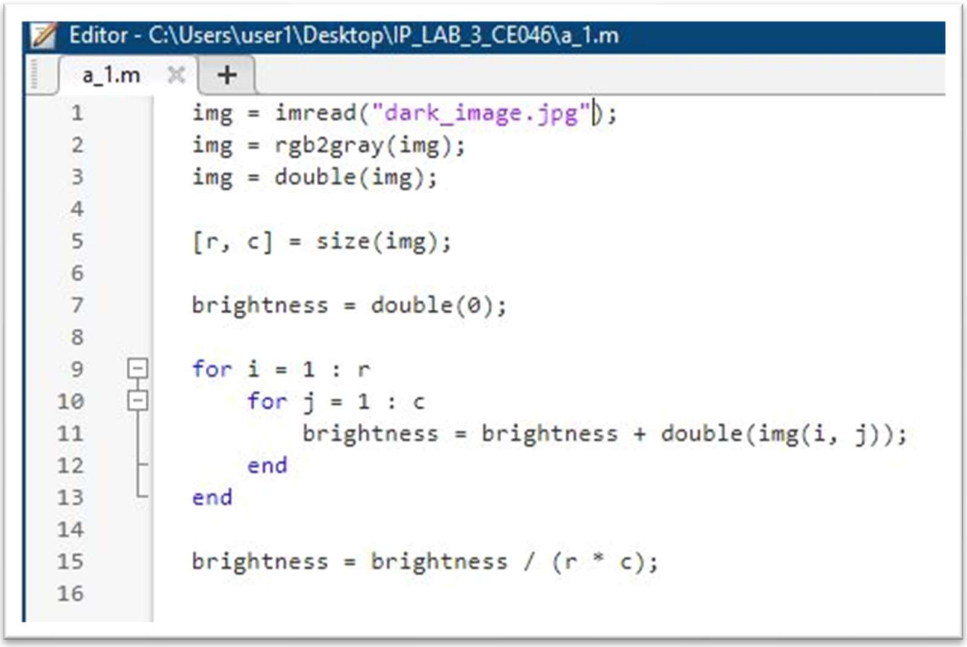
- Output:



Name	Value
brighness	0
brightness	124.0425
c	512
i	512
img	*512x512 double*
j	512
r	512

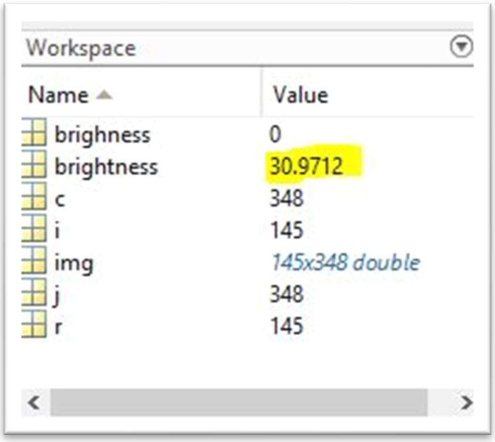
❖ Low brightness image

- Code:



```
img = imread("dark_image.jpg");
img = rgb2gray(img);
img = double(img);

[r, c] = size(img);

brightness = double(0);

for i = 1 : r
    for j = 1 : c
        brightness = brightness + double(img(i, j));
    end
end

brightness = brightness / (r * c);

```

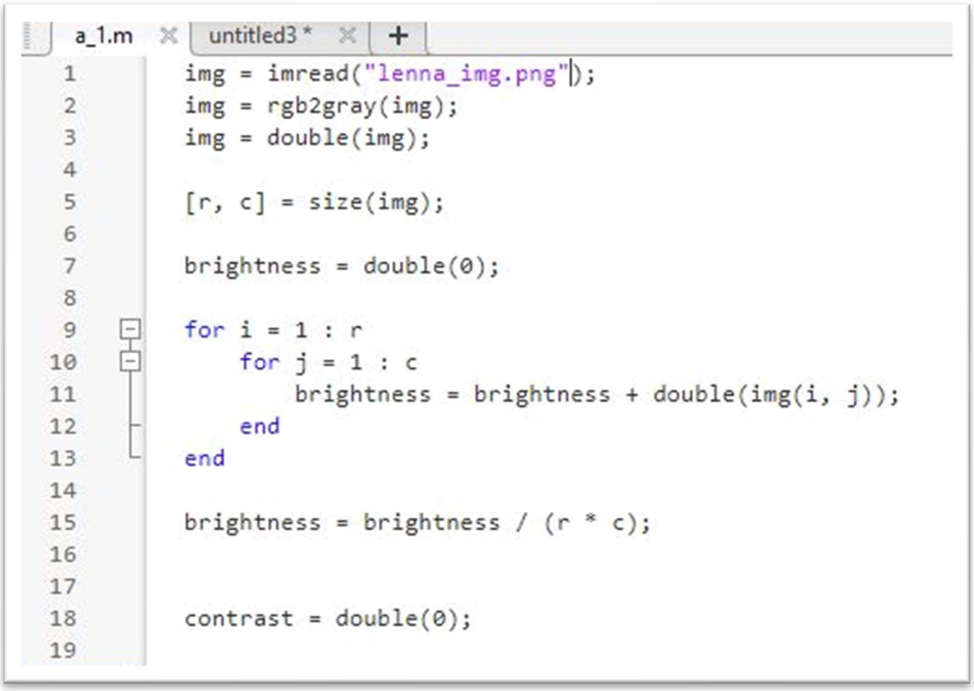
- Output:



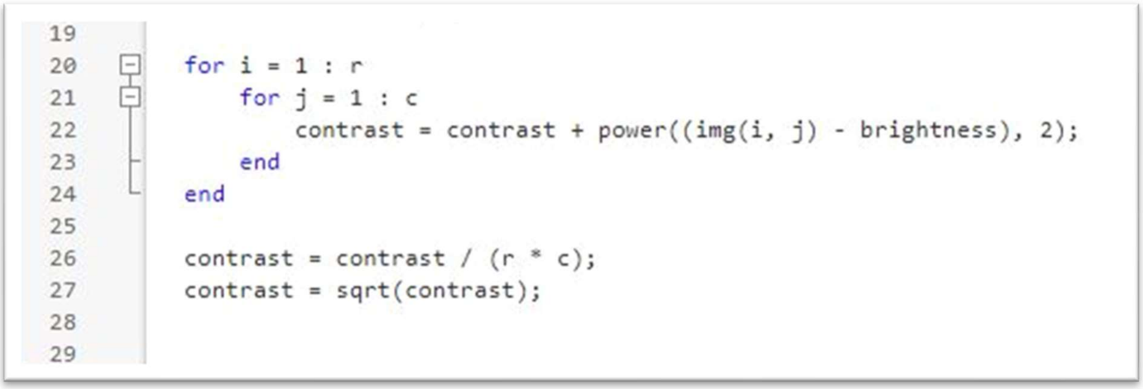
Name	Value
brighness	0
brightness	30.9712
c	348
i	145
img	145x348 double
j	348
r	145

High contrast image

- Code:

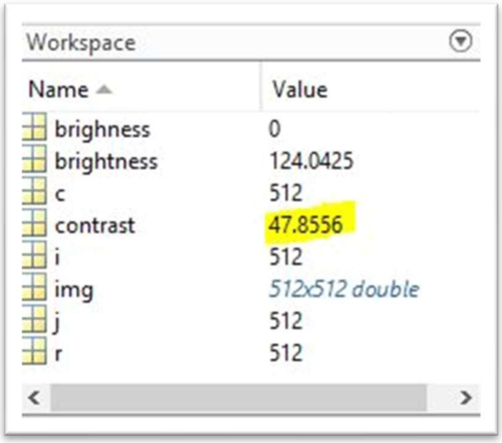


```
img = imread("lenna_img.png");
img = rgb2gray(img);
img = double(img);

[r, c] = size(img);

brightness = double(0);

for i = 1 : r
    for j = 1 : c
        brightness = brightness + double(img(i, j));
    end
end

brightness = brightness / (r * c);


contrast = double(0);

```



```
for i = 1 : r
    for j = 1 : c
        contrast = contrast + power((img(i, j) - brightness), 2);
    end
end

contrast = contrast / (r * c);
contrast = sqrt(contrast);
```

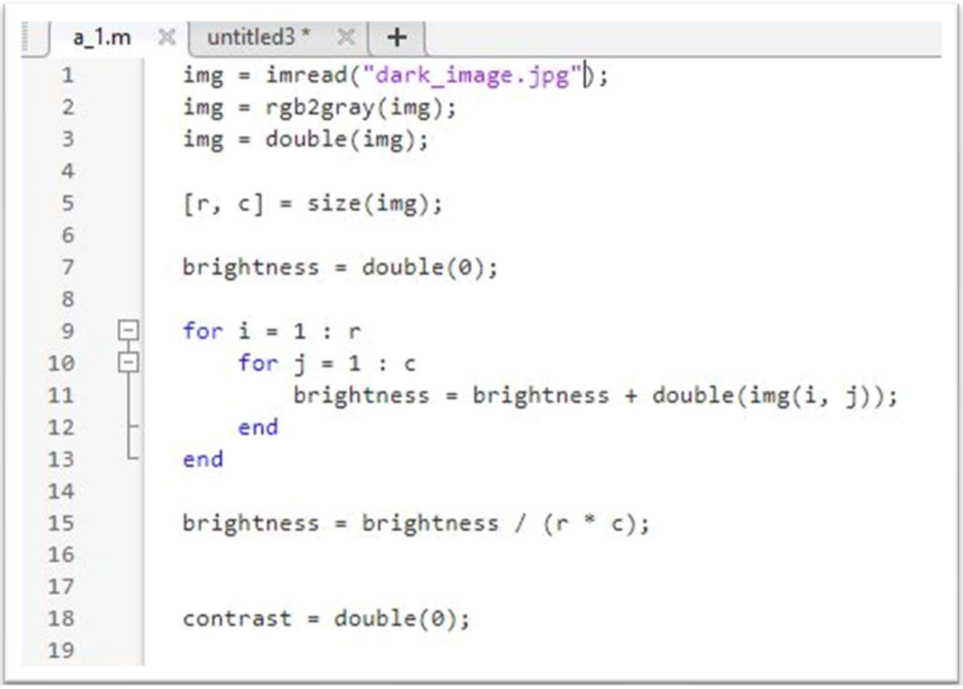
- Output:



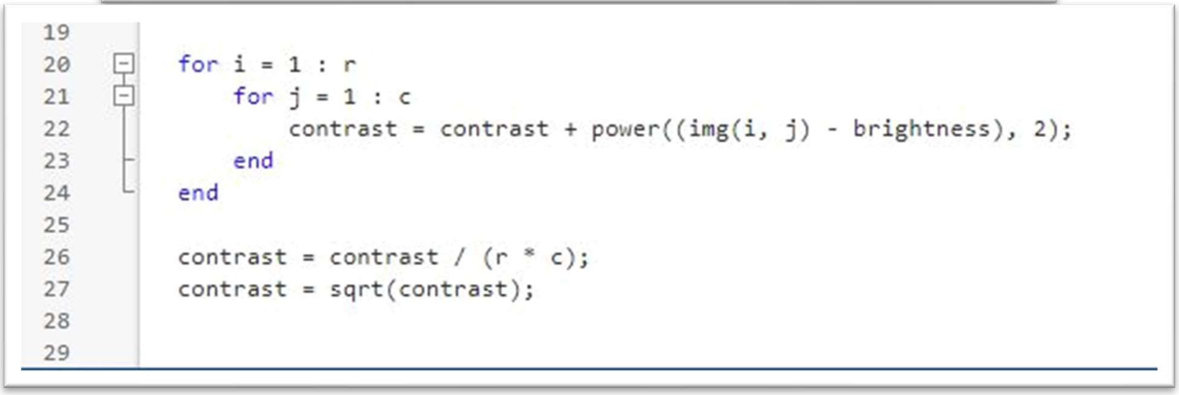
Name	Value
brighness	0
brightness	124.0425
c	512
contrast	47.8556
i	512
img	*512x512 double*
j	512
r	512

❖ <u>Low contrast image</u>

- Code:

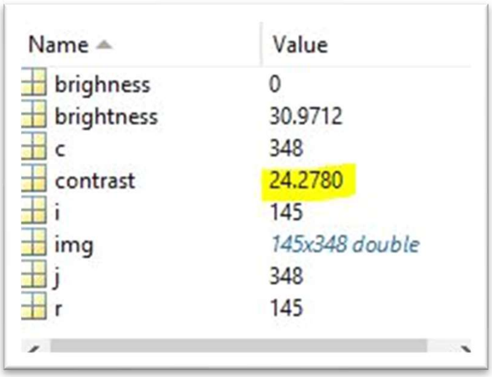


```
img = imread("dark_image.jpg");
img = rgb2gray(img);
img = double(img);

[r, c] = size(img);

brightness = double(0);

for i = 1 : r
    for j = 1 : c
        brightness = brightness + double(img(i, j));
    end
end

brightness = brightness / (r * c);


contrast = double(0);

```



```

for i = 1 : r
    for j = 1 : c
        contrast = contrast + power((img(i, j) - brightness), 2);
    end
end

contrast = contrast / (r * c);
contrast = sqrt(contrast);


```

- Output:

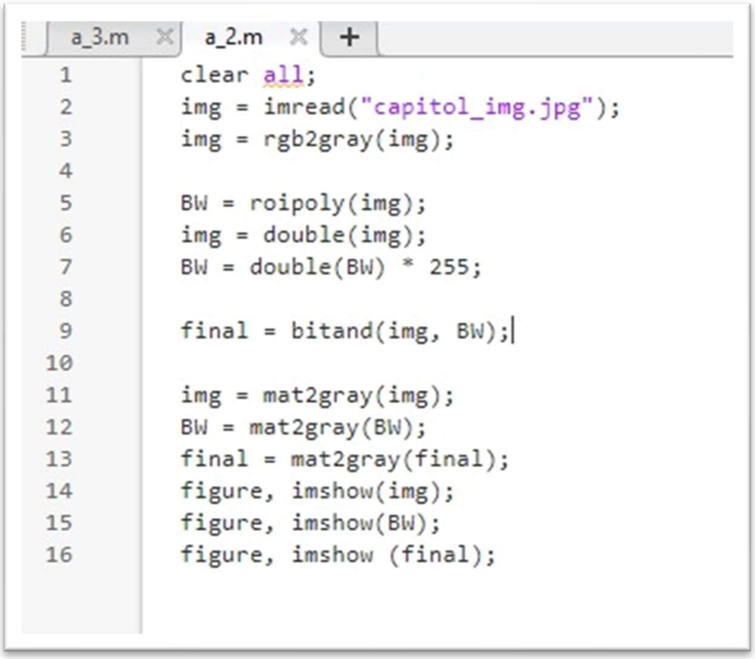


Name ▲	Value
brighness	0
brightness	30.9712
c	348
contrast	24.2780
i	145
img	*145x348 double*
j	348
r	145

Q. 2: Perform AND, NOT, and OR logical operations on the images.

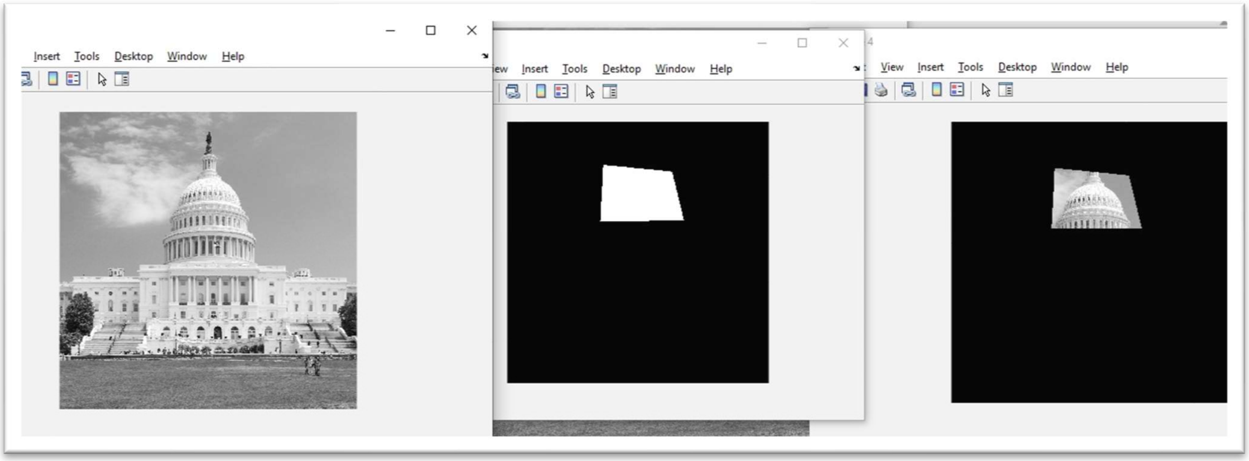
❖ Logical AND

- Code:



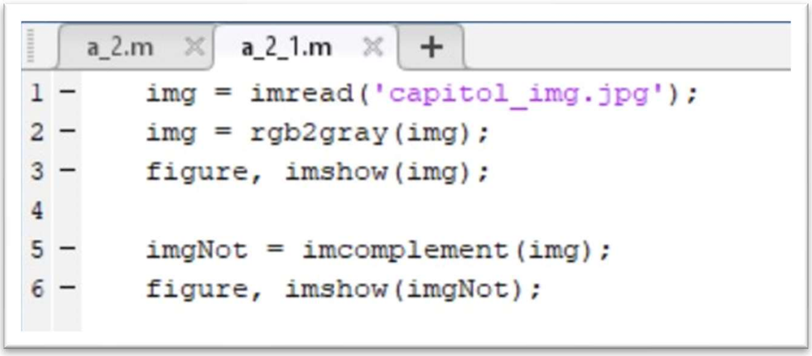
```
clear all;
img = imread("capitol_img.jpg");
img = rgb2gray(img);

BW = roipoly(img);
img = double(img);
BW = double(BW) * 255;

final = bitand(img, BW);

img = mat2gray(img);
BW = mat2gray(BW);
final = mat2gray(final);
figure, imshow(img);
figure, imshow(BW);
figure, imshow (final);
```

- Output:


Insert Tools Desktop Window Help
Insert Tools Desktop Window Help
View Insert Tools Desktop Window Help

❖ Logical NOT

➢ Code:



```
    img = imread('capitol_img.jpg');
    img = rgb2gray(img);
    figure, imshow(img);

    imgNot = imcomplement(img);
    figure, imshow(imgNot);
```

➢ Output:

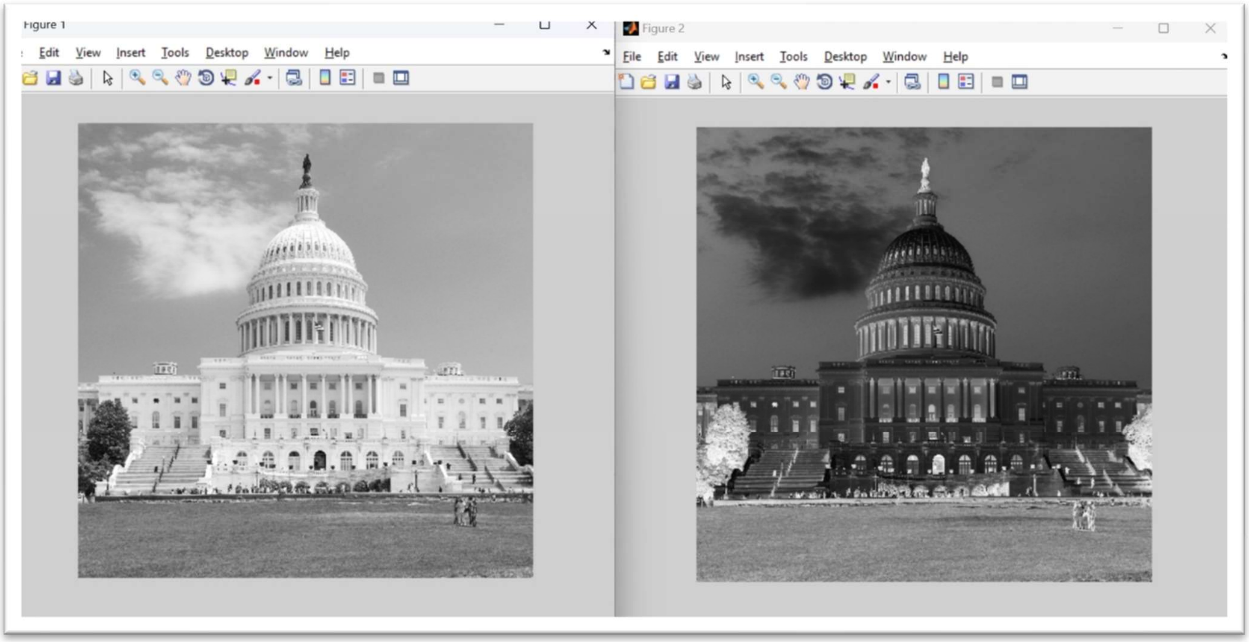
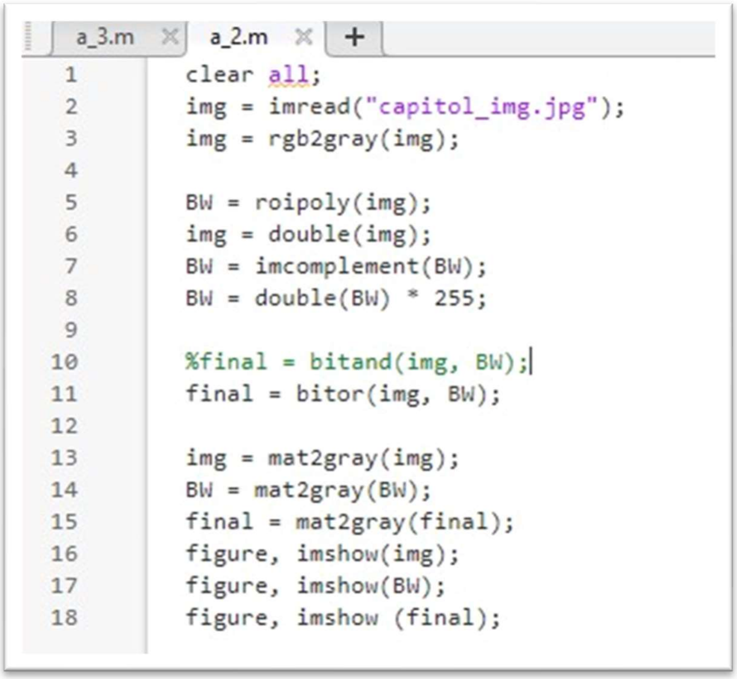


Figure 1
Edit View Insert Tools Desktop Window Help
Figure 2
File Edit View Insert Tools Desktop Window Help

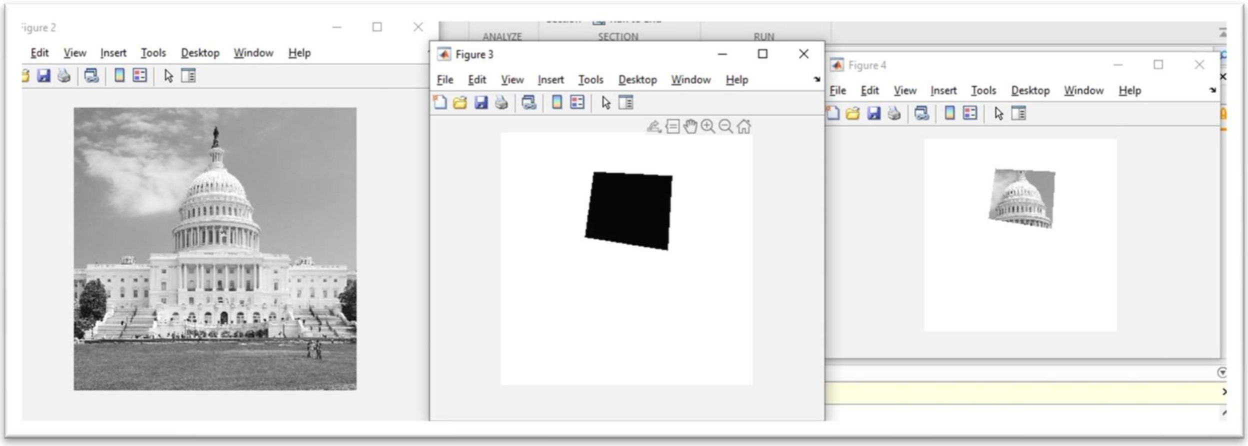
❖ Logical OR

- Code:



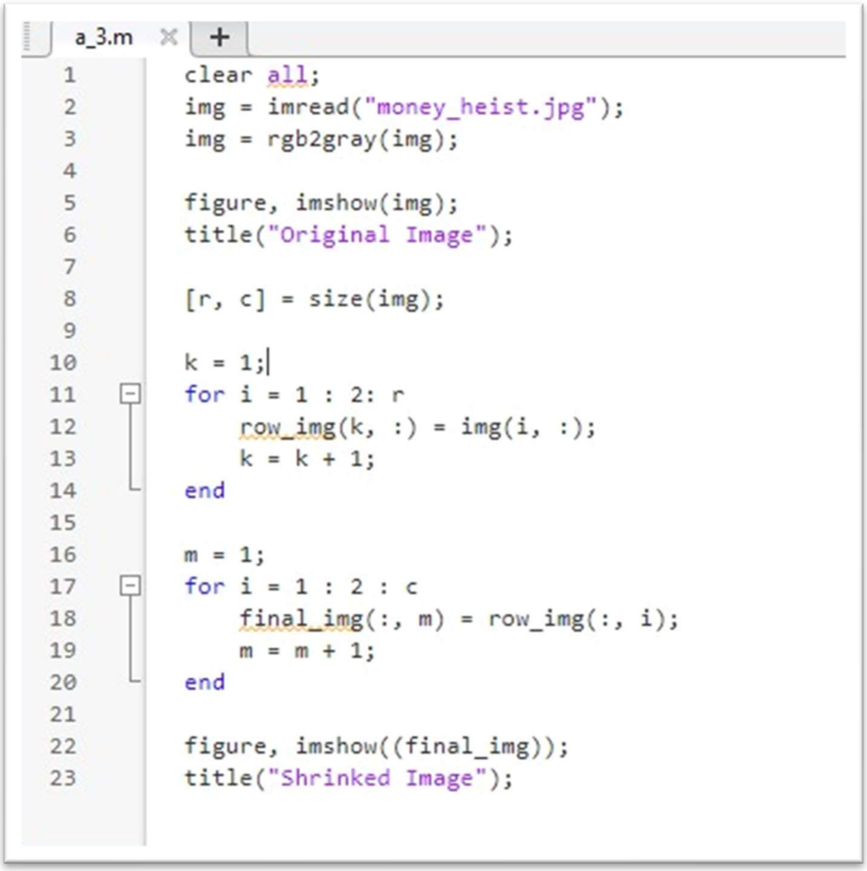
```
clear all;
img = imread("capitol_img.jpg");
img = rgb2gray(img);

BW = roipoly(img);
img = double(img);
BW = imcomplement(BW);
BW = double(BW) * 255;

%final = bitand(img, BW);
final = bitor(img, BW);

img = mat2gray(img);
BW = mat2gray(BW);
final = mat2gray(final);
figure, imshow(img);
figure, imshow(BW);
figure, imshow (final);
```

- Output:


igure 2
Edit View Insert Tools Desktop Window Help
Figure 3
File Edit View Insert Tools Desktop Window Help
Figure 4
File Edit View Insert Tools Desktop Window Help
ANALYZE
SECTION
RUN

Q. 3: Perform image shrinking operation on the image.

❖ Code:



```
clear all;
img = imread("money_heist.jpg");
img = rgb2gray(img);

figure, imshow(img);
title("Original Image");

[r, c] = size(img);

k = 1;
for i = 1 : 2: r
    row_img(k, :) = img(i, :);
    k = k + 1;
end

m = 1;
for i = 1 : 2 : c
    final_img(:, m) = row_img(:, i);
    m = m + 1;
end

figure, imshow((final_img));
title("Shrinked Image");
```

❖ Output:

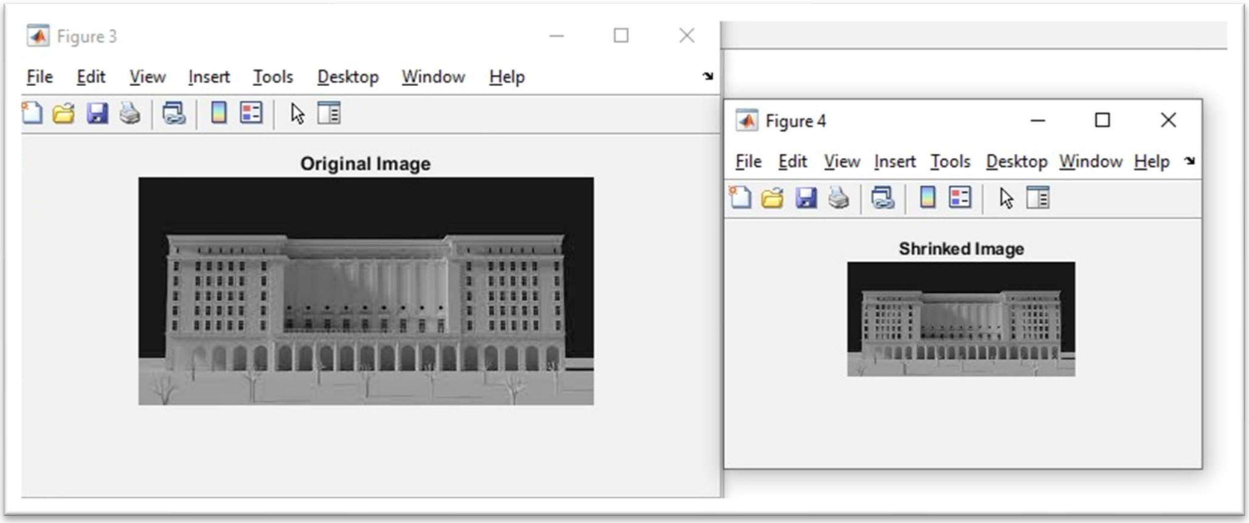
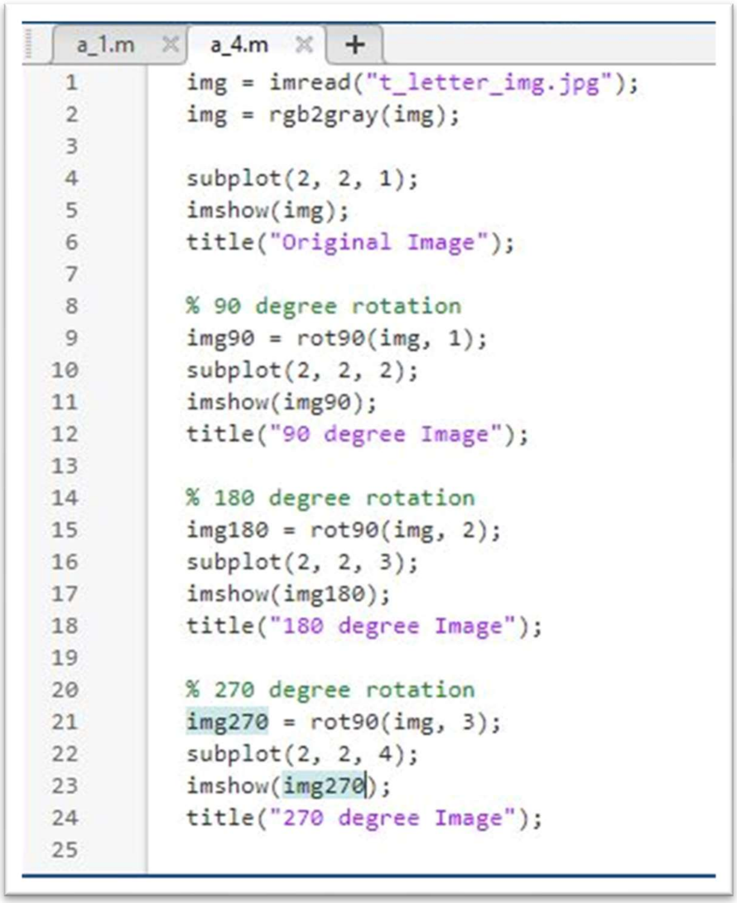


Figure 3
File Edit View Insert Tools Desktop Window Help
Original Image
Figure 4
File Edit View Insert Tools Desktop Window Help
Shrinked Image

Q. 4: Perform image transformation. (Rotation)

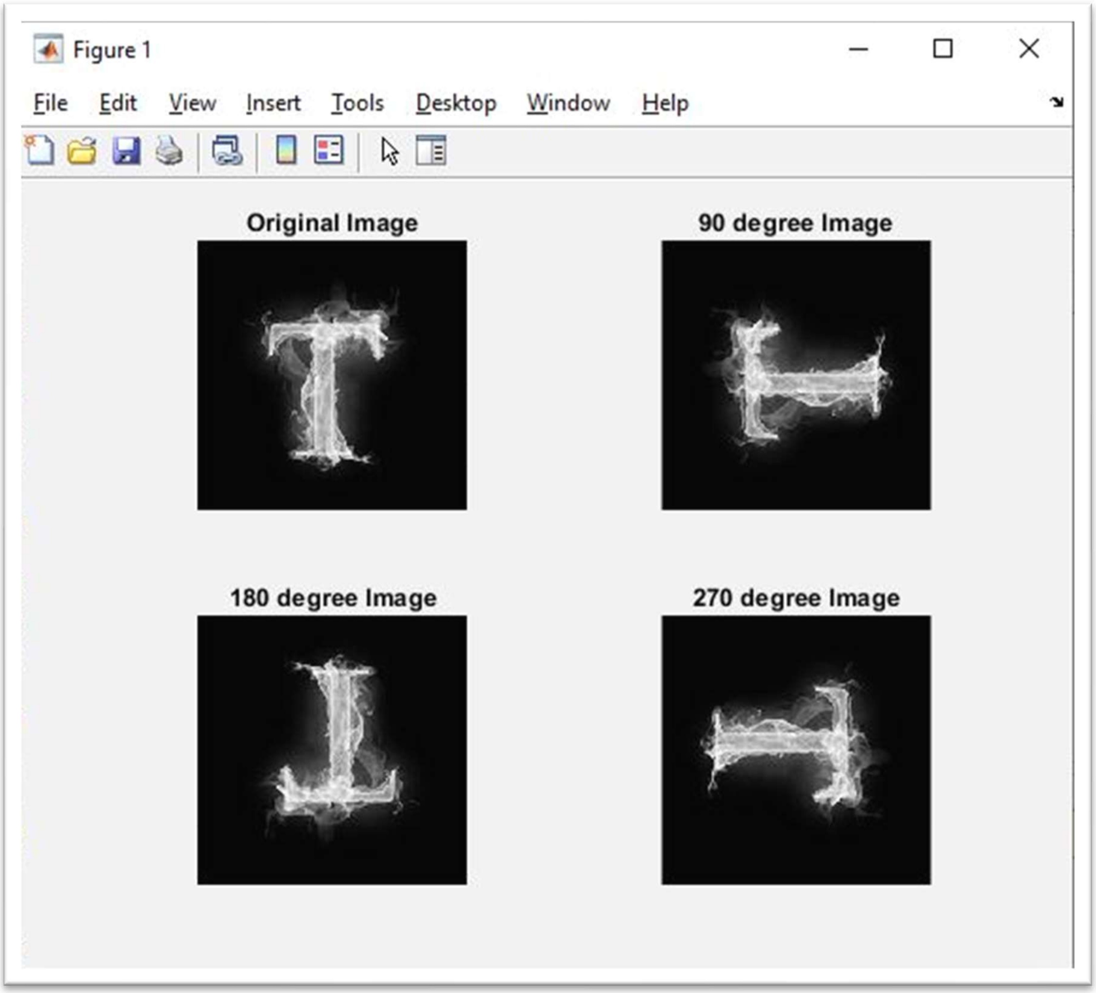
❖ Using rot90() function

➢ Code:



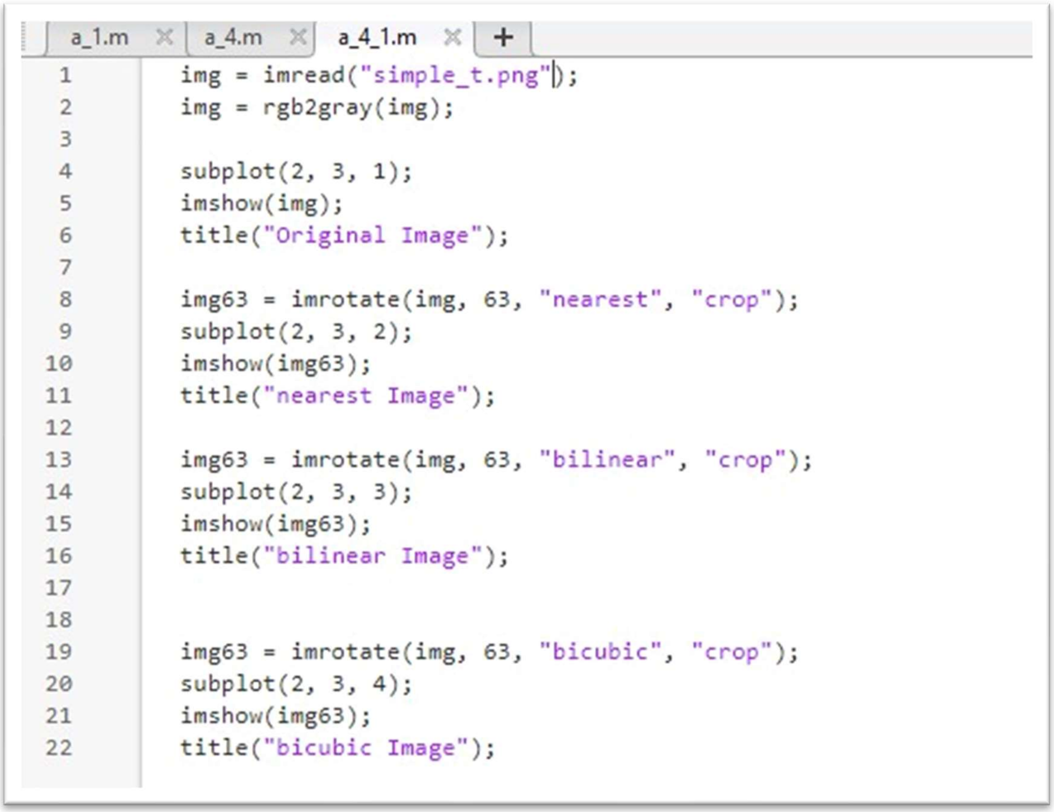
```
img = imread("t_letter_img.jpg");
img = rgb2gray(img);

subplot(2, 2, 1);
imshow(img);
title("Original Image");

% 90 degree rotation
img90 = rot90(img, 1);
subplot(2, 2, 2);
imshow(img90);
title("90 degree Image");

% 180 degree rotation
img180 = rot90(img, 2);
subplot(2, 2, 3);
imshow(img180);
title("180 degree Image");

% 270 degree rotation
img270 = rot90(img, 3);
subplot(2, 2, 4);
imshow(img270);
title("270 degree Image");
```

➢ Output:



❖ Using imrotate() function

- Code:



```
img = imread("simple_t.png");
img = rgb2gray(img);

subplot(2, 3, 1);
imshow(img);
title("Original Image");

img63 = imrotate(img, 63, "nearest", "crop");
subplot(2, 3, 2);
imshow(img63);
title("nearest Image");

img63 = imrotate(img, 63, "bilinear", "crop");
subplot(2, 3, 3);
imshow(img63);
title("bilinear Image");


img63 = imrotate(img, 63, "bicubic", "crop");
subplot(2, 3, 4);
imshow(img63);
title("bicubic Image");
```

- Output:

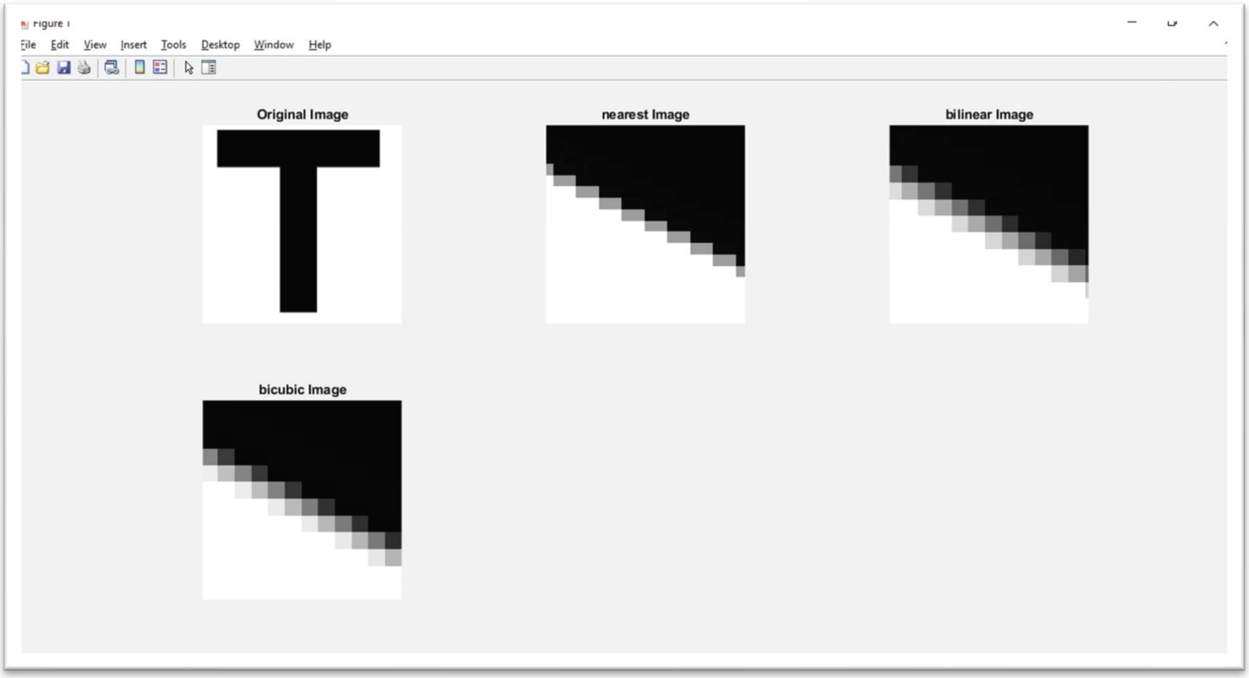


Figure 1
File Edit View Insert Tools Desktop Window Help
Original Image
nearest Image
bilinear Image
bicubic Image