

LAB – 3

Name: Gandevia Keval Dharmeshbhai

Sem: VII

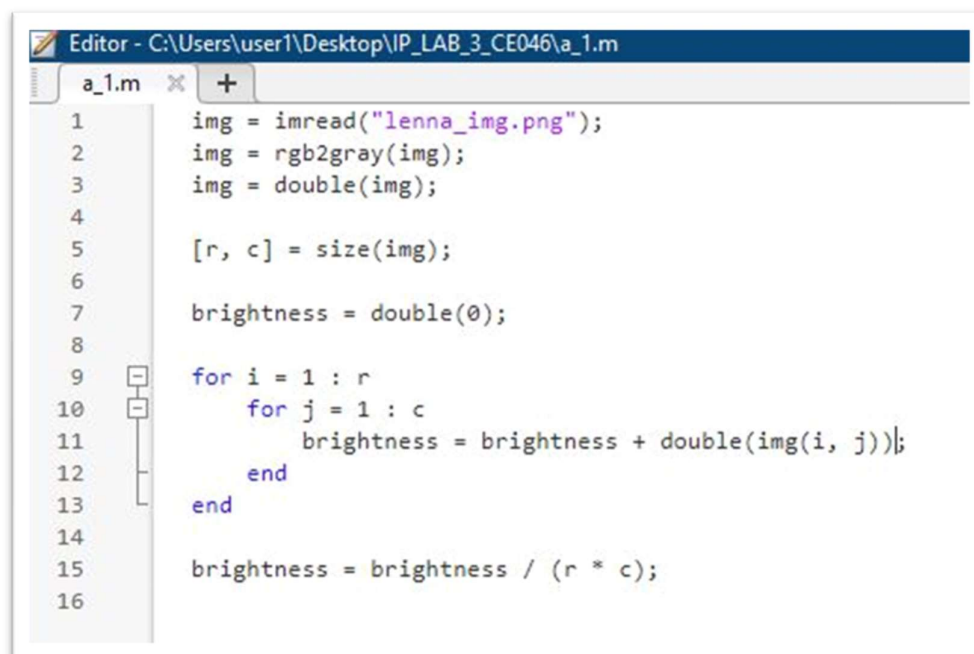
Roll No: CE046

Subject: Image Processing

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

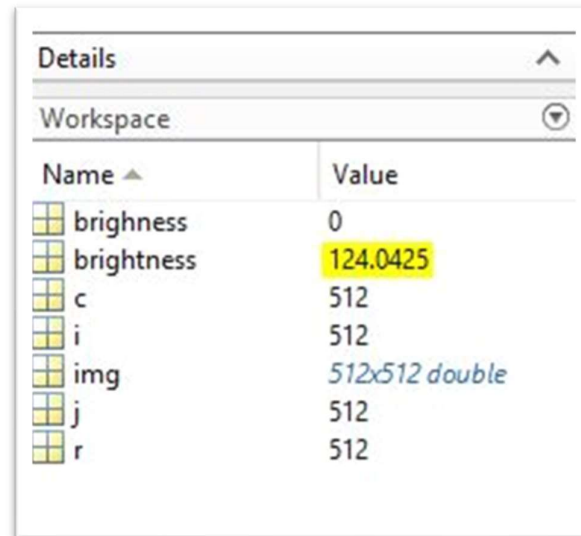
❖ High brightness image

➤ Code:

A screenshot of a MATLAB Editor window. The title bar reads "Editor - C:\Users\user1\Desktop\IP_LAB_3_CE046\a_1.m". The editor has a tab labeled "a_1.m" with a close button and a plus sign. The code is as follows:

```
1  img = imread("lenna_img.png");
2  img = rgb2gray(img);
3  img = double(img);
4
5  [r, c] = size(img);
6
7  brightness = double(0);
8
9  for i = 1 : r
10     for j = 1 : c
11         brightness = brightness + double(img(i, j));
12     end
13 end
14
15 brightness = brightness / (r * c);
16
```

➤ Output:

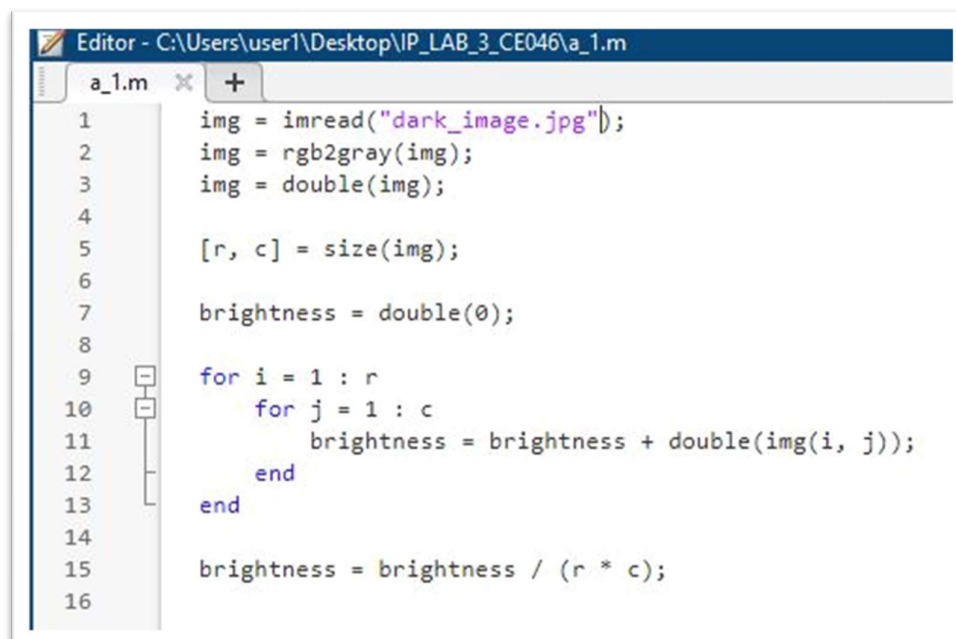


The image shows the MATLAB Workspace window. It has a 'Details' tab and a 'Workspace' section. The workspace contains several variables: 'brightness' (0), 'brightness' (124.0425), 'c' (512), 'i' (512), 'img' (512x512 double), 'j' (512), and 'r' (512). The second 'brightness' value is highlighted in yellow.

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

❖ Low brightness image

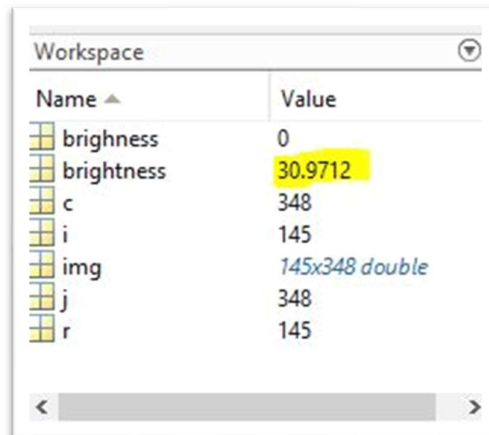
➤ Code:



The image shows the MATLAB Editor window with a script named 'a_1.m'. The script calculates the average brightness of an image. It reads 'dark_image.jpg', converts it to grayscale, and then iterates through each pixel to calculate the average brightness.

```
1  img = imread("dark_image.jpg");
2  img = rgb2gray(img);
3  img = double(img);
4
5  [r, c] = size(img);
6
7  brightness = double(0);
8
9  for i = 1 : r
10     for j = 1 : c
11         brightness = brightness + double(img(i, j));
12     end
13 end
14
15 brightness = brightness / (r * c);
16
```

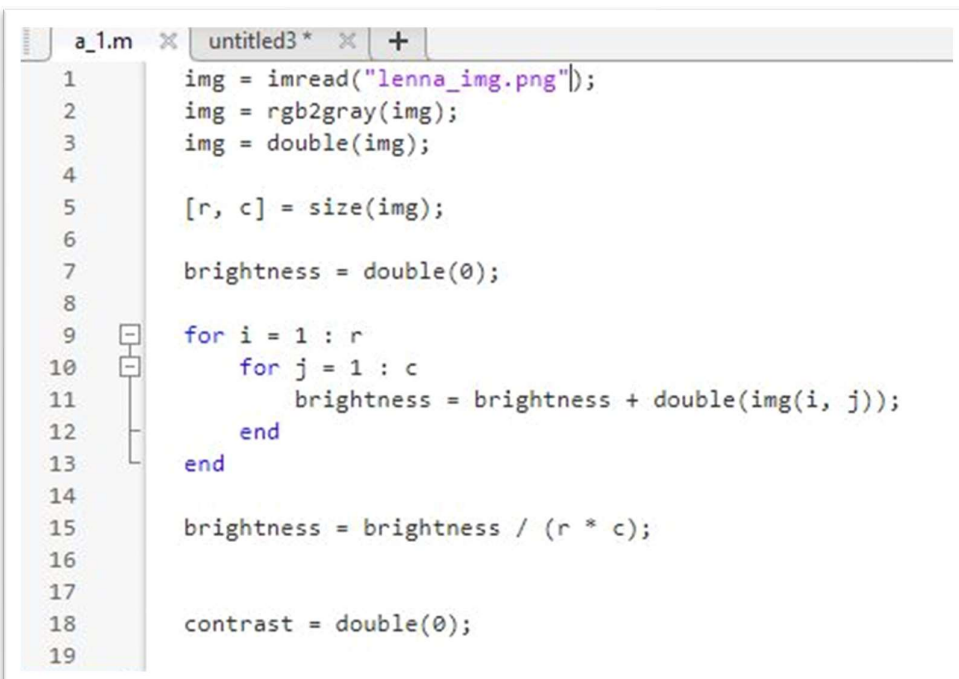
➤ Output:



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

❖ High contrast image

➤ Code:



```
1  img = imread("lenna_img.png");
2  img = rgb2gray(img);
3  img = double(img);
4
5  [r, c] = size(img);
6
7  brightness = double(0);
8
9  for i = 1 : r
10     for j = 1 : c
11         brightness = brightness + double(img(i, j));
12     end
13 end
14
15 brightness = brightness / (r * c);
16
17
18 contrast = double(0);
19
```

```

19
20   for i = 1 : r
21       for j = 1 : c
22           contrast = contrast + power((img(i, j) - brightness), 2);
23       end
24   end
25
26   contrast = contrast / (r * c);
27   contrast = sqrt(contrast);
28
29

```

➤ Output:

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

❖ Low contrast image

➤ Code:

```
a_1.m x untitled3 * x +
1  img = imread('dark_image.jpg');
2  img = rgb2gray(img);
3  img = double(img);
4
5  [r, c] = size(img);
6
7  brightness = double(0);
8
9  for i = 1 : r
10     for j = 1 : c
11         brightness = brightness + double(img(i, j));
12     end
13 end
14
15 brightness = brightness / (r * c);
16
17
18 contrast = double(0);
19
```

```
19
20     for i = 1 : r
21         for j = 1 : c
22             contrast = contrast + power((img(i, j) - brightness), 2);
23         end
24     end
25
26 contrast = contrast / (r * c);
27 contrast = sqrt(contrast);
28
29
```

➤ Output:

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

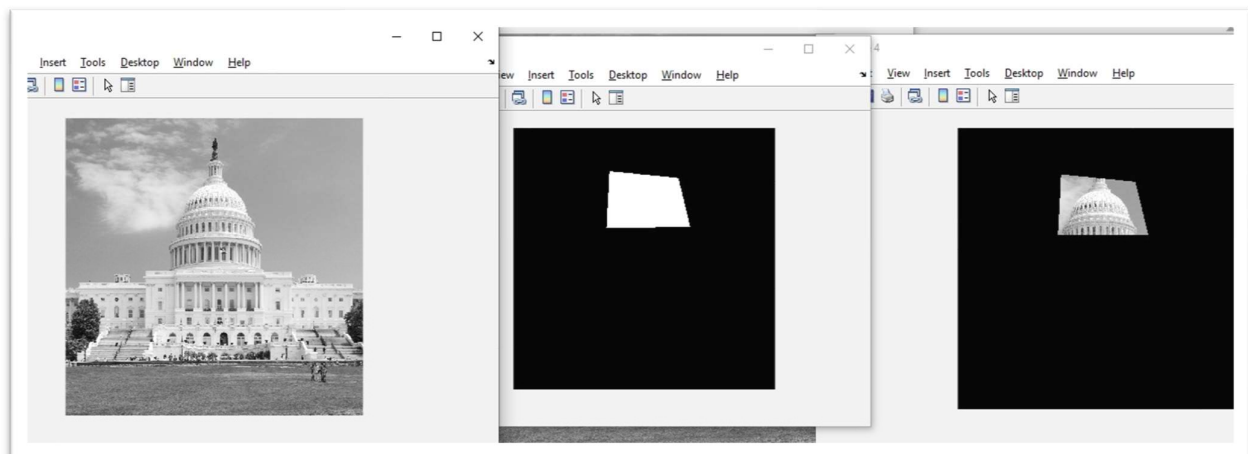
Q. 2: Perform AND, NOT, and OR logical operations on the images.

❖ Logical AND

➤ Code:

```
a_3.m  a_2.m  +
1      clear all;
2      img = imread("capitol_img.jpg");
3      img = rgb2gray(img);
4
5      BW = roipoly(img);
6      img = double(img);
7      BW = double(BW) * 255;
8
9      final = bitand(img, BW);|
10
11     img = mat2gray(img);
12     BW = mat2gray(BW);
13     final = mat2gray(final);
14     figure, imshow(img);
15     figure, imshow(BW);
16     figure, imshow (final);
```

➤ Output:

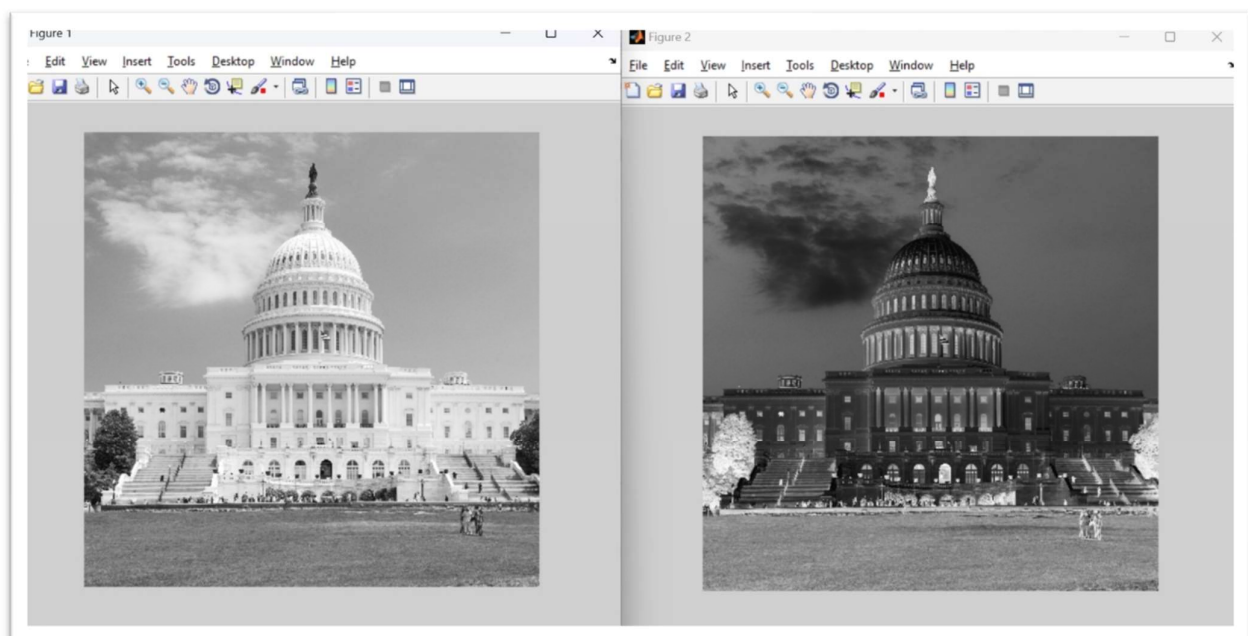


❖ Logical NOT

➤ Code:

```
a_2.m  a_2_1.m  +
1 -     img = imread('capitol_img.jpg');
2 -     img = rgb2gray(img);
3 -     figure, imshow(img);
4
5 -     imgNot = imcomplement(img);
6 -     figure, imshow(imgNot);
```

➤ Output:

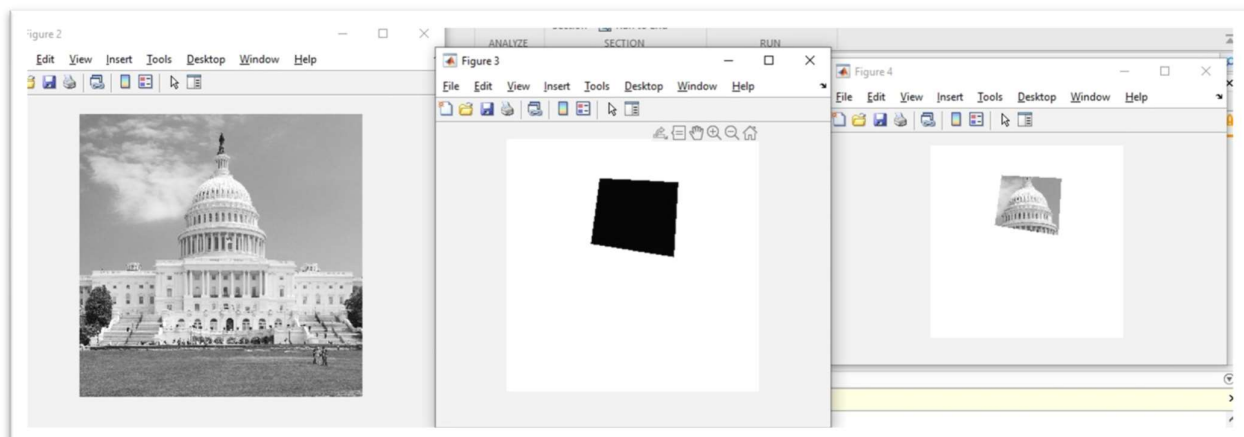


❖ Logical OR

➤ Code:

```
a_3.m  a_2.m  +
1      clear all;
2      img = imread("capitol_img.jpg");
3      img = rgb2gray(img);
4
5      BW = roipoly(img);
6      img = double(img);
7      BW = imcomplement(BW);
8      BW = double(BW) * 255;
9
10     %final = bitand(img, BW);|
11     final = bitor(img, BW);
12
13     img = mat2gray(img);
14     BW = mat2gray(BW);
15     final = mat2gray(final);
16     figure, imshow(img);
17     figure, imshow(BW);
18     figure, imshow (final);
```

➤ Output:

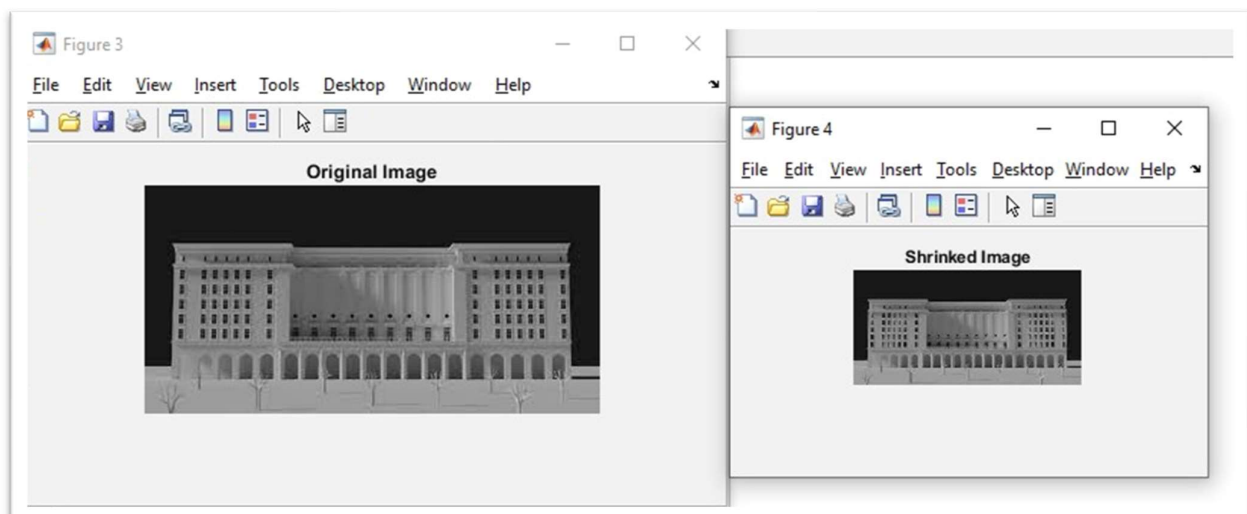


Q. 3: Perform image shrinking operation on the image.

❖ Code:

```
a_3.m x +
1 clear all;
2 img = imread("money_heist.jpg");
3 img = rgb2gray(img);
4
5 figure, imshow(img);
6 title("Original Image");
7
8 [r, c] = size(img);
9
10 k = 1;
11 for i = 1 : 2: r
12     row_img(k, :) = img(i, :);
13     k = k + 1;
14 end
15
16 m = 1;
17 for i = 1 : 2 : c
18     final_img(:, m) = row_img(:, i);
19     m = m + 1;
20 end
21
22 figure, imshow((final_img));
23 title("Shrunked Image");
```

❖ Output:



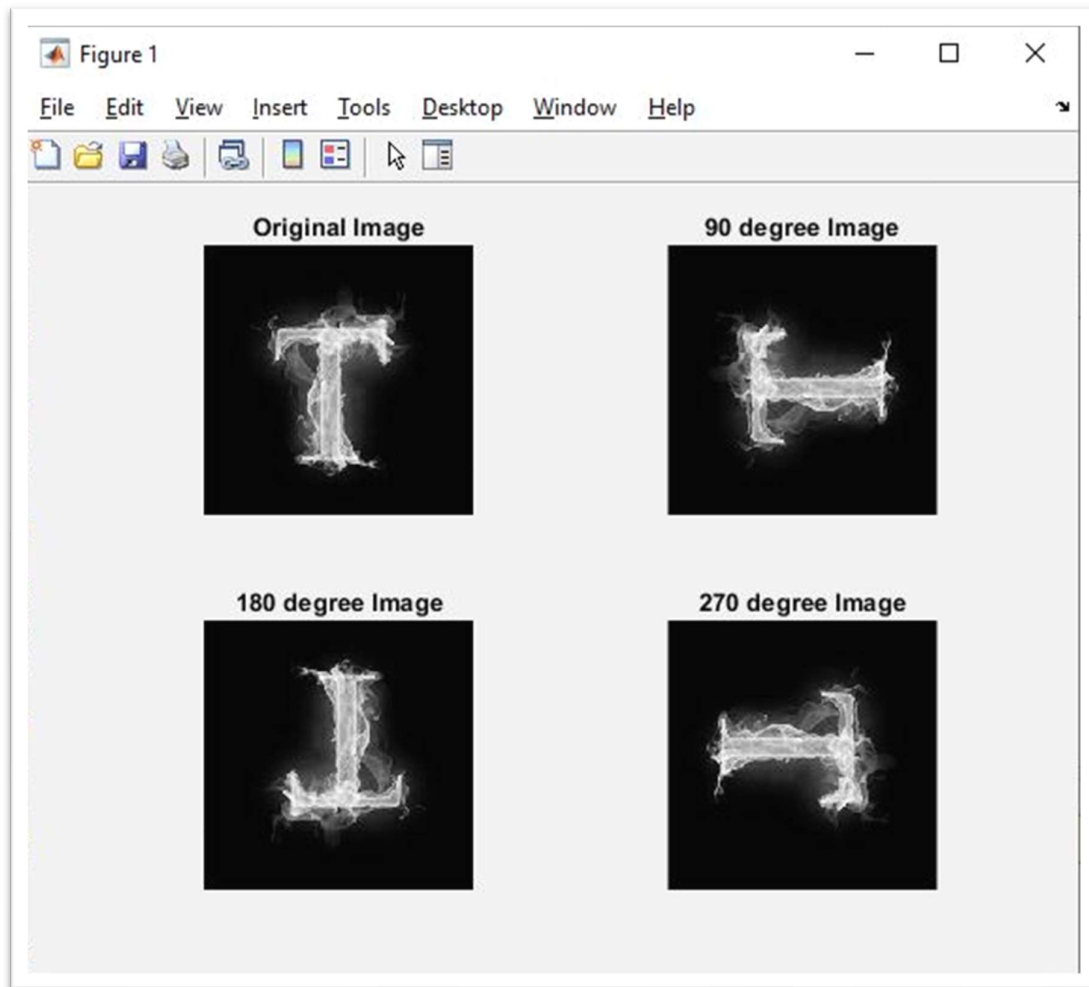
Q. 4: Perform image transformation. (Rotation)

❖ Using rot90() function

➤ Code:

```
1  img = imread("t_letter_img.jpg");
2  img = rgb2gray(img);
3
4  subplot(2, 2, 1);
5  imshow(img);
6  title("Original Image");
7
8  % 90 degree rotation
9  img90 = rot90(img, 1);
10 subplot(2, 2, 2);
11 imshow(img90);
12 title("90 degree Image");
13
14 % 180 degree rotation
15 img180 = rot90(img, 2);
16 subplot(2, 2, 3);
17 imshow(img180);
18 title("180 degree Image");
19
20 % 270 degree rotation
21 img270 = rot90(img, 3);
22 subplot(2, 2, 4);
23 imshow(img270);
24 title("270 degree Image");
25
```

➤ Output:



❖ Using imrotate() function

➤ Code:

```
a_1.m x a_4.m x a_4_1.m x +
1  img = imread("simple_t.png");
2  img = rgb2gray(img);
3
4  subplot(2, 3, 1);
5  imshow(img);
6  title("Original Image");
7
8  img63 = imrotate(img, 63, "nearest", "crop");
9  subplot(2, 3, 2);
10 imshow(img63);
11 title("nearest Image");
12
13 img63 = imrotate(img, 63, "bilinear", "crop");
14 subplot(2, 3, 3);
15 imshow(img63);
16 title("bilinear Image");
17
18
19 img63 = imrotate(img, 63, "bicubic", "crop");
20 subplot(2, 3, 4);
21 imshow(img63);
22 title("bicubic Image");
```

➤ Output:

