LAB – 1

Name: Gandevia Keval Dharmeshbhai

Sem: VII

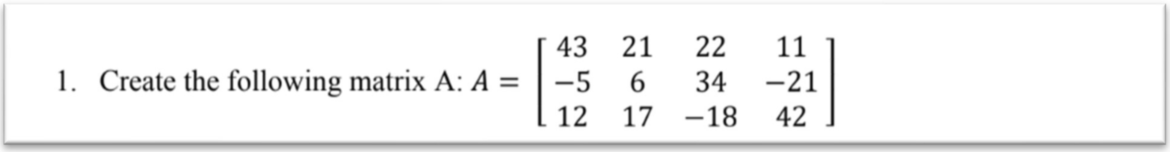
Roll No: CE018

Subject: Image Processing

Aim: Getting familiar with MATLAB and performing basic operations on image.

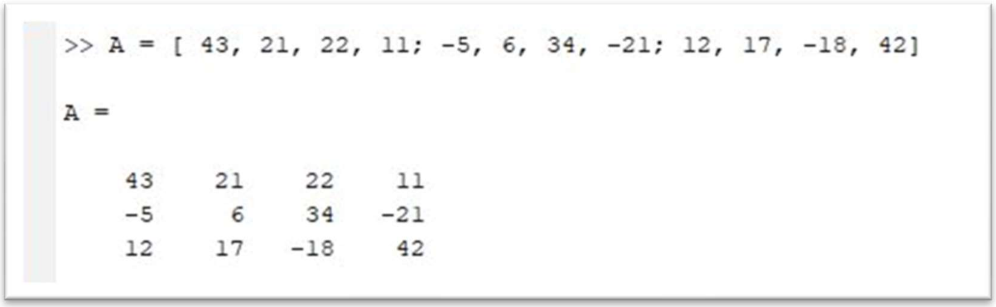
MATLAB BASICS

Q. 1:



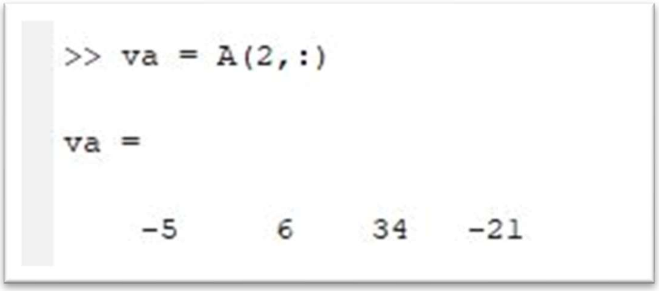
1. Create the following matrix A: $A = \begin{bmatrix} 43 & 21 & 22 & 11 \\ -5 & 6 & 34 & -21 \\ 12 & 17 & -18 & 42 \end{bmatrix}$

- **Creation of given matrix:**



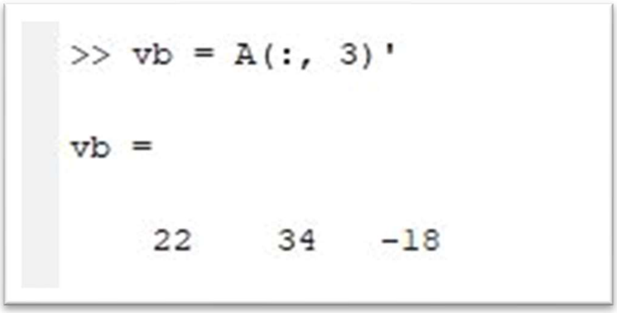
```
>> A = [ 43, 21, 22, 11; -5, 6, 34, -21; 12, 17, -18, 42]

A =

    43    21    22    11
    -5     6    34   -21
    12    17   -18    42
```

a) **Create a four-element row vector named *va* that contains the elements of the second row of A.**



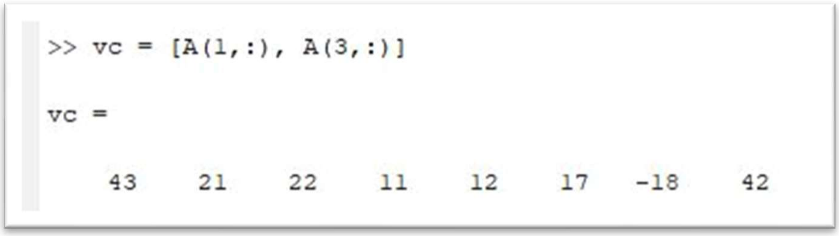
```
>> va = A(2,:)

va =

    -5     6    34   -21
```

b) **Create a three-element row vector named *vb* that contains the elements of the third column of A.**



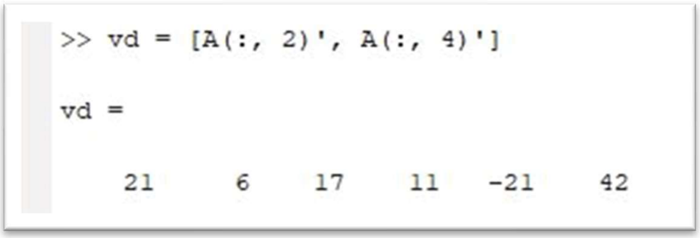
```
>> vb = A(:, 3)'

vb =

    22    34   -18
```

c) **Create an eight-element row vector named *vc* that contains the elements of the first and third rows of A.**



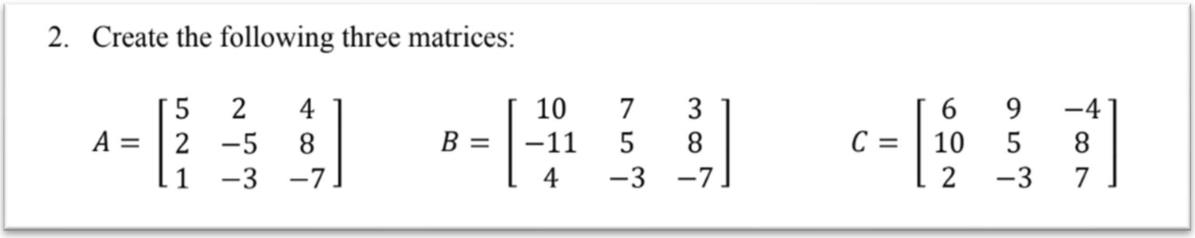
```
>> vc = [A(1,:), A(3,:)]

vc =

    43    21    22    11    12    17   -18    42
```

d) Create an eight-element row vector named *vd* that contains the elements of the first and third rows of A.



```
>> vd = [A(:, 2)', A(:, 4)']

vd =

    21     6    17    11   -21    42
```

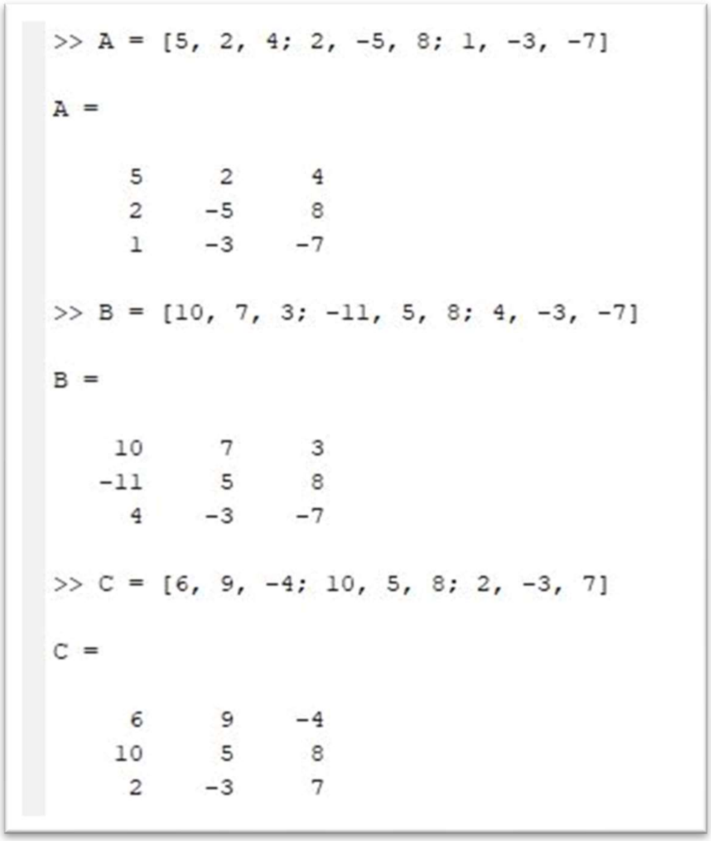
Q. 2:



2. Create the following three matrices:

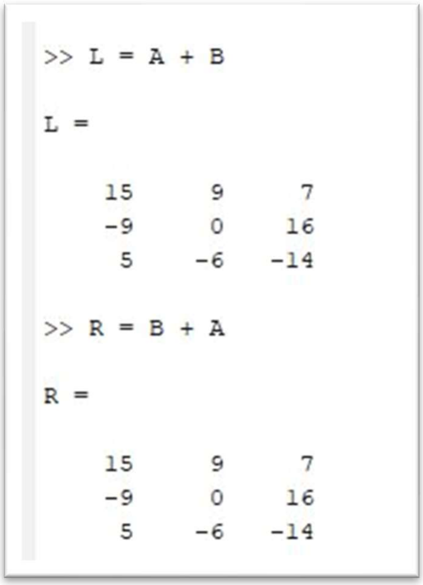
$$A = \begin{bmatrix} 5 & 2 & 4 \\ 2 & -5 & 8 \\ 1 & -3 & -7 \end{bmatrix} \qquad B = \begin{bmatrix} 10 & 7 & 3 \\ -11 & 5 & 8 \\ 4 & -3 & -7 \end{bmatrix} \qquad C = \begin{bmatrix} 6 & 9 & -4 \\ 10 & 5 & 8 \\ 2 & -3 & 7 \end{bmatrix}$$

❖ Creation of matrices:



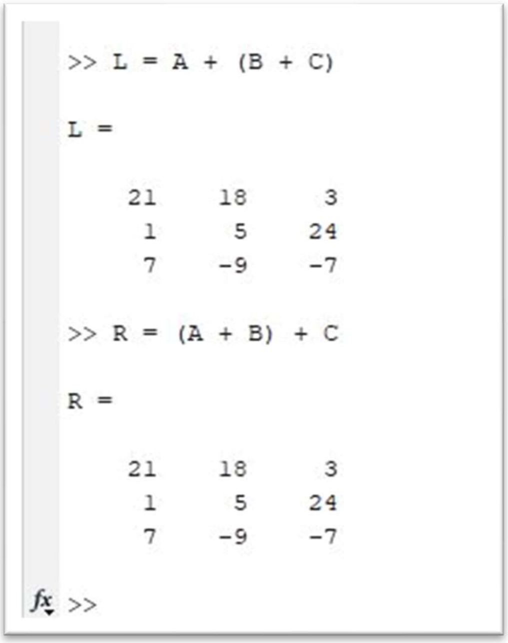
```
>> A = [5, 2, 4; 2, -5, 8; 1, -3, -7]

A =

     5     2     4
     2    -5     8
     1    -3    -7

>> B = [10, 7, 3; -11, 5, 8; 4, -3, -7]

B =

    10     7     3
   -11     5     8
     4    -3    -7

>> C = [6, 9, -4; 10, 5, 8; 2, -3, 7]

C =

     6     9    -4
    10     5     8
     2    -3     7
```

a) Calculate A + B and B + A to show that addition of matrices is commutative.



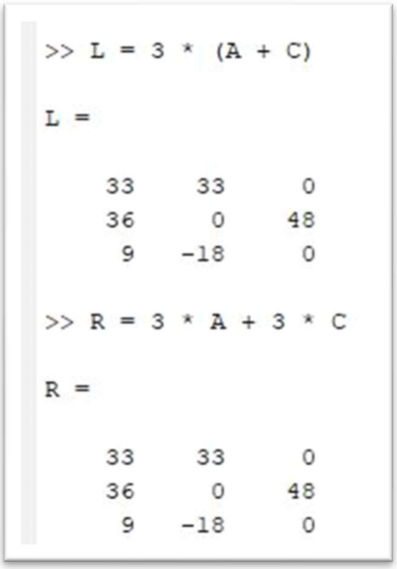
```
>> L = A + B

L =

    15     9     7
    -9     0    16
     5    -6   -14

>> R = B + A

R =

    15     9     7
    -9     0    16
     5    -6   -14
```

b) **Calculate A + (B + C) and (A + B) + C to show that addition of matrices is associative.**



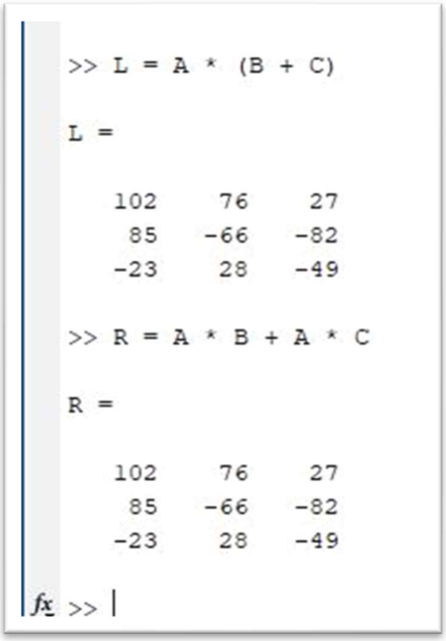
```
>> L = A + (B + C)

L =

    21    18     3
     1     5    24
     7    -9    -7

>> R = (A + B) + C

R =

    21    18     3
     1     5    24
     7    -9    -7

fx >>
```

c) **Calculate 3(A + C) and 3A + 3C to show that, when matrices are multiplied by a scalar, the multiplication is distributive.**



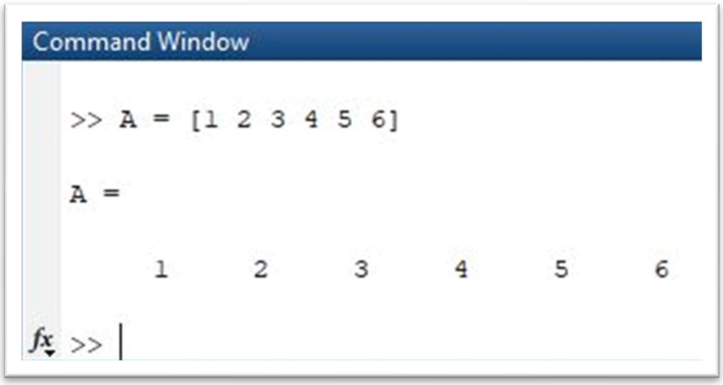
```
>> L = 3 * (A + C)

L =

    33    33     0
    36     0    48
     9   -18     0

>> R = 3 * A + 3 * C

R =

    33    33     0
    36     0    48
     9   -18     0
```

d) Calculate A * (B + C) and A * B + A * C to show that matrix multiplication is distributive.



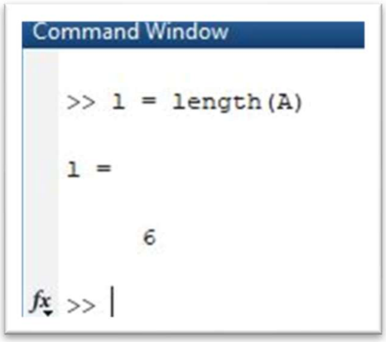
```
>> L = A * (B + C)

L =

   102    76    27
    85   -66   -82
   -23    28   -49

>> R = A * B + A * C

R =

   102    76    27
    85   -66   -82
   -23    28   -49

>> 
```

Q. 3: Create an array A = [1 2 3 4 5 6] and using built in functions for array solve following questions.



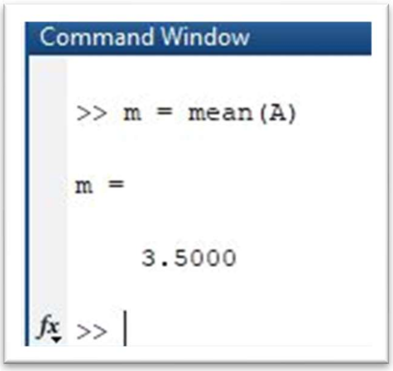
```
>> A = [1 2 3 4 5 6]

A =

     1     2     3     4     5     6

>> 
```

a) Length of A.



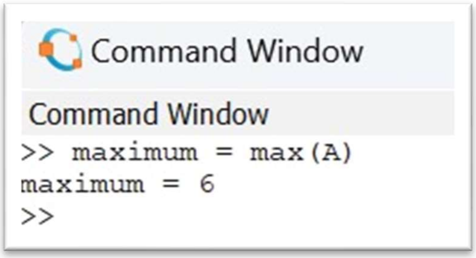
```
>> l = length(A)

l =

     6

>> 
```

b) Average of elements of A.



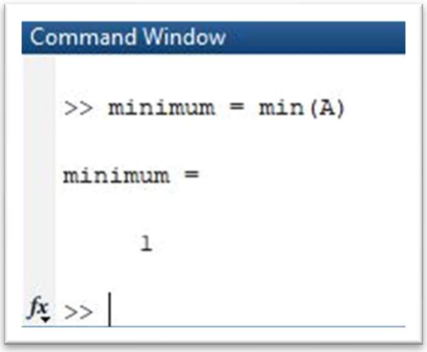
```
>> m = mean(A)

m =

    3.5000

>> 
```

c) Maximum of A.



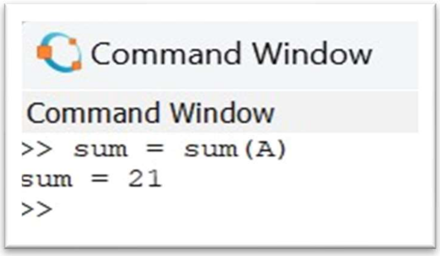
```
>> maximum = max(A)
maximum = 6
>>
```

d) Minimum of A.



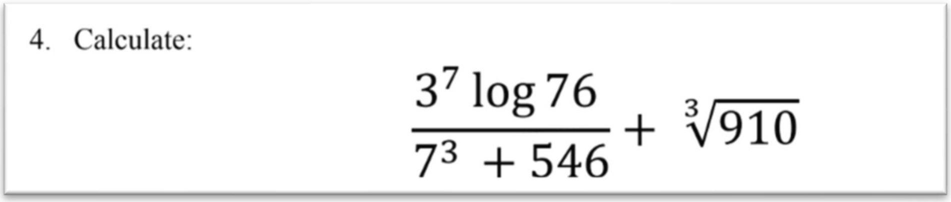
```
>> minimum = min(A)

minimum =

     1

fx >>
```

e) Sum of all the elements of A.



```
>> sum = sum(A)
sum = 21
>>
```

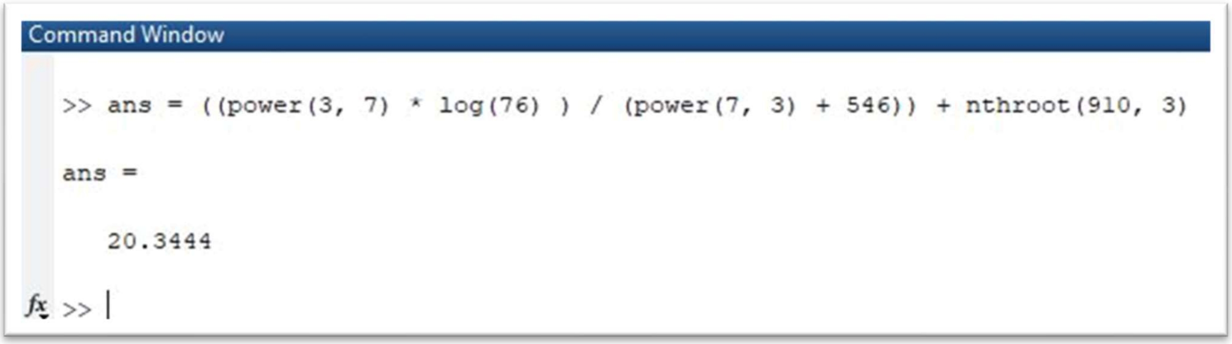
Q. 4:



4. Calculate:

$$\frac{3^7 \log 76}{7^3 \;+ 546} + \sqrt[3]{910}$$

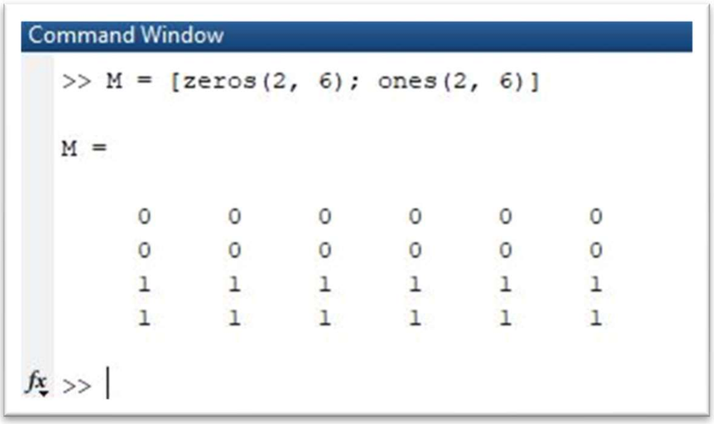
❖



```
>> ans = ((power(3, 7) * log(76) ) / (power(7, 3) + 546)) + nthroot(910, 3)

ans =

   20.3444

>> 
```

Q. 5: Using the ones and zeros commands, create a 4 X 6 matrix in which the first two rows are 0’s and the next two rows are 1’s.

❖



```
>> M = [zeros(2, 6); ones(2, 6)]

M =

     0     0     0     0     0     0
     0     0     0     0     0     0
     1     1     1     1     1     1
     1     1     1     1     1     1

>> 
```

IMAGE PROCESSING TOOLBOX IN MATLAB

Q. 1: Inbuilt functions for image processing in MATLAB.

a) imread ():

- ✓ Syntax: imread (filename)
- ✓ Description: It reads the image from the specified filename and convert that image into matrix form.

b) imshow ():

- ✓ Syntax: imshow (image)
- ✓ Description: It is used to display the specified image in figure.

c) imwrite ():

- ✓ Syntax: imwrite (image, filename)
- ✓ Description: It is used to write image into specified filename, inferring the file from the extension.

d) figure ():

- ✓ Syntax: figure
- ✓ Description: It is used to create a new figure using default property values.

e) subplot ():

- ✓ Syntax: subplot (m, n, p)
- ✓ Description: It divides the current figure into an m-by-n grid and creates axes in the position specified by p.

f) size ():

- ✓ Syntax: size (M)
- ✓ Description: This function helps us to know the dimension of the matrix.

g) imresize ():

- ✓ Syntax: imresize (image, scale)
- ✓ Description: It returns the new image that is scale times the size of old image. The input image can be grayscale, RGB, binary, or categorical image.

h) imcrop ():

- ✓ Syntax: imcrop (image)
- ✓ Description: This function helps to display the grayscale, truecolor, or binary image in a figure window and creates an interactive crop image tool associated with the image.

i) imfinfo ():

- ✓ Syntax: imfinfo (filename)
- ✓ Description: This function returns a structure whose fields contain information about an image in graphics file, filename.

j) rgb2gray ():

- ✓ Syntax: rgb2gray(rgb)
- ✓ Description: It is used to convert the RGB image into gray scale image by eliminating the hue and saturation information and retaining the luminance.

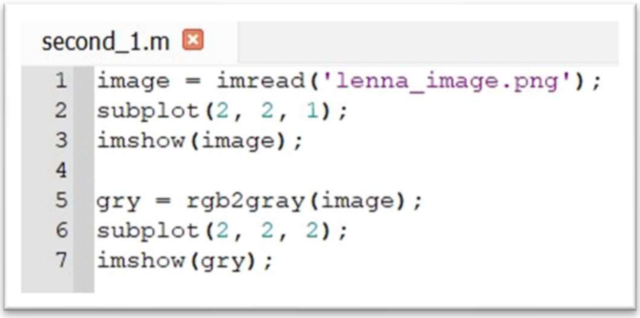
k) im2bw ():

- ✓ Syntax: im2bw (image, level)
- ✓ Description: It converts the grayscale image to binary image, by replacing all pixels in the input image with luminance greater than the level with the value 1 and replacing all other pixels with the value 0.

Q. 2: Take your own photo (RGB Image) and create the following images:

a) Gray scale image.

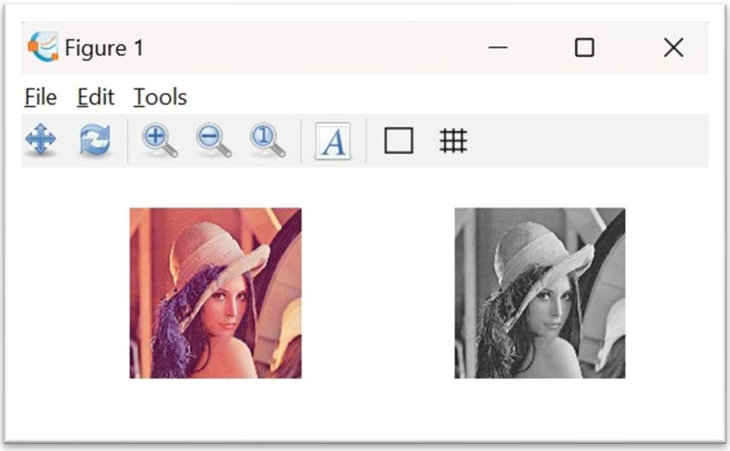
❖ **Code:**



second_1.m

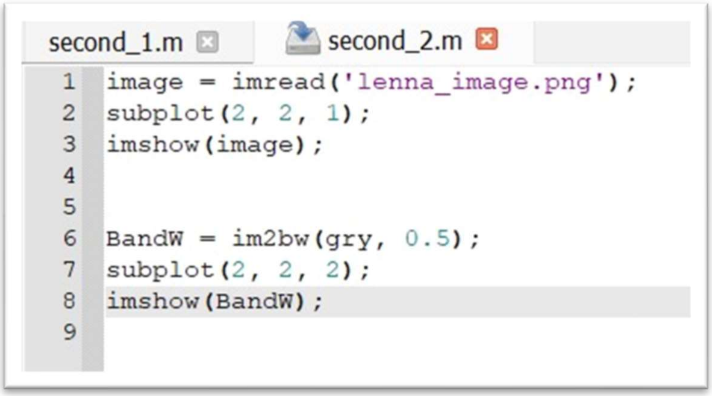
```
image = imread('lenna_image.png');
subplot(2, 2, 1);
imshow(image);

gry = rgb2gray(image);
subplot(2, 2, 2);
imshow(gry);
```

❖ **Output:**



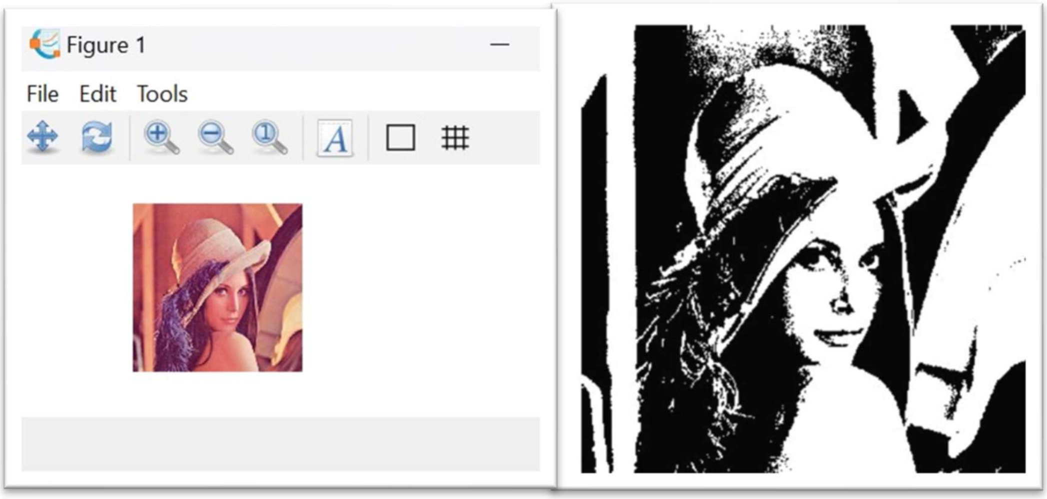
b) Black and white image.

- **Code:**



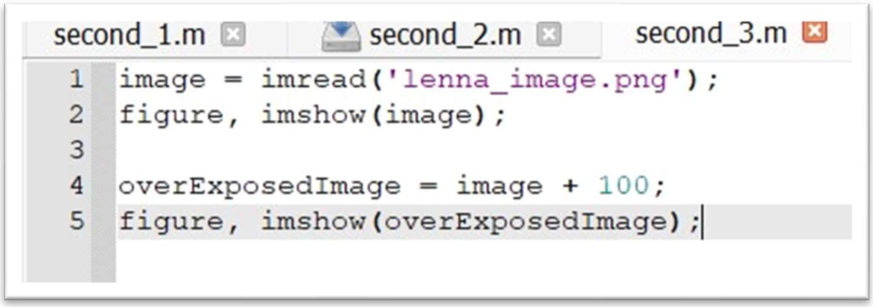
```
image = imread('lenna_image.png');
subplot(2, 2, 1);
imshow(image);


BandW = im2bw(gry, 0.5);
subplot(2, 2, 2);
imshow(BandW);
```

- **Output:**



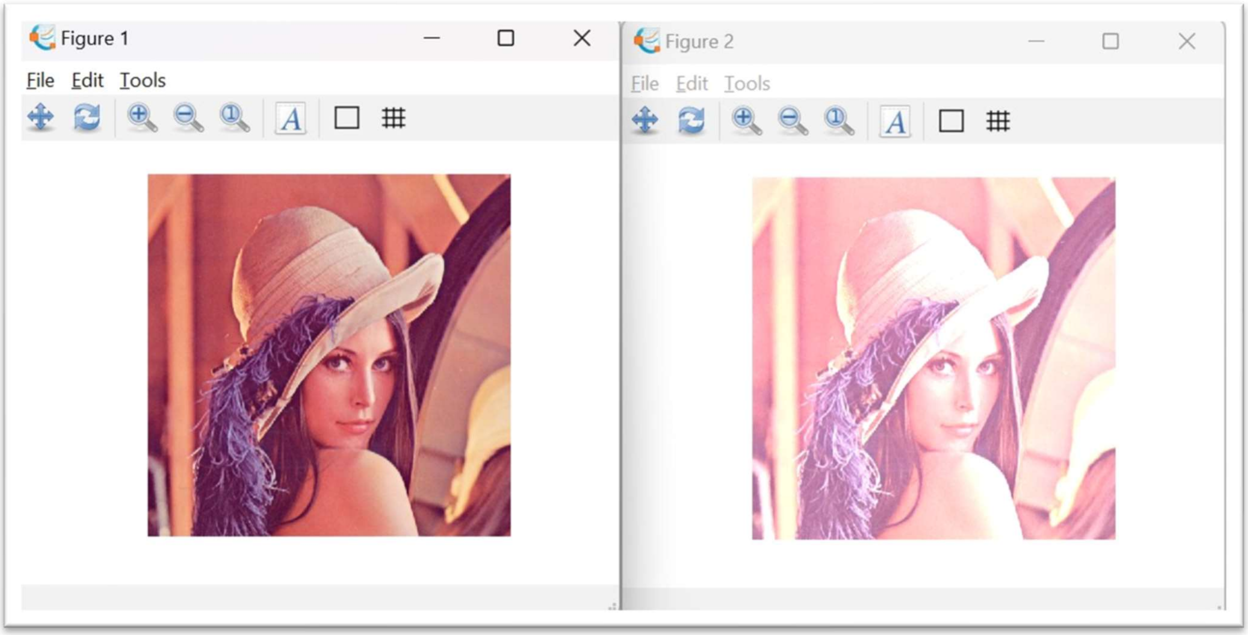
c) Over exposed image.

- **Code:**



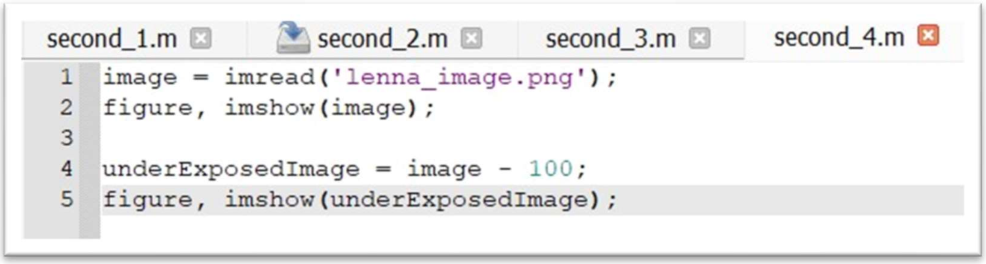
```
image = imread('lenna_image.png');
figure, imshow(image);

overExposedImage = image + 100;
figure, imshow(overExposedImage);
```

- **Output:**



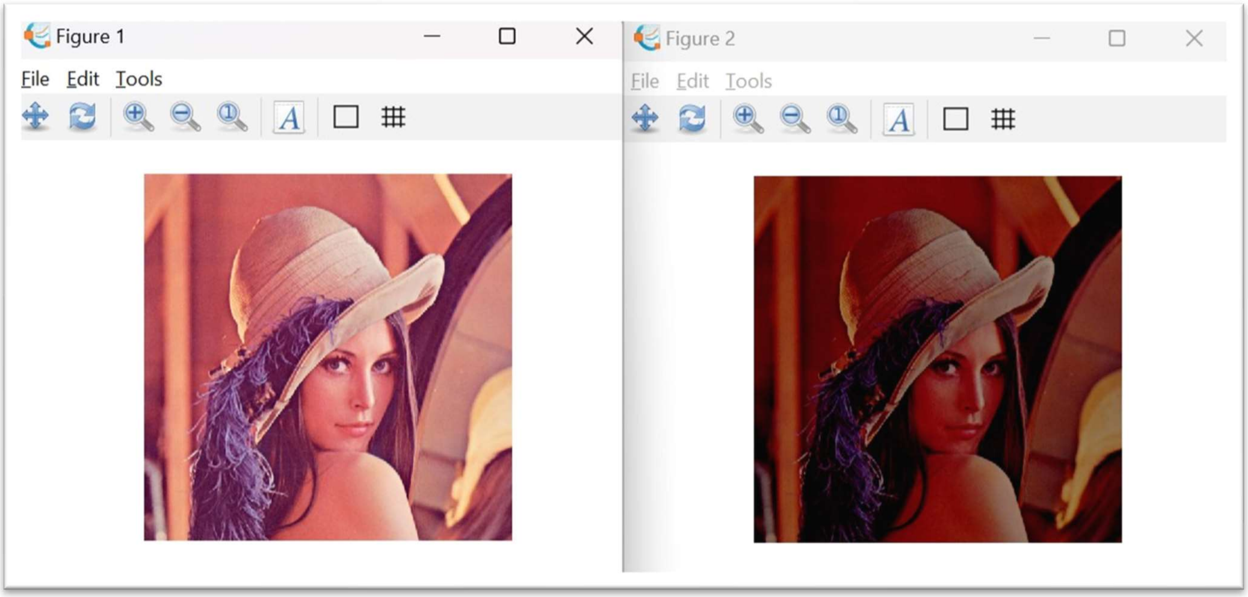
d) Under exposed image.

- **Code:**



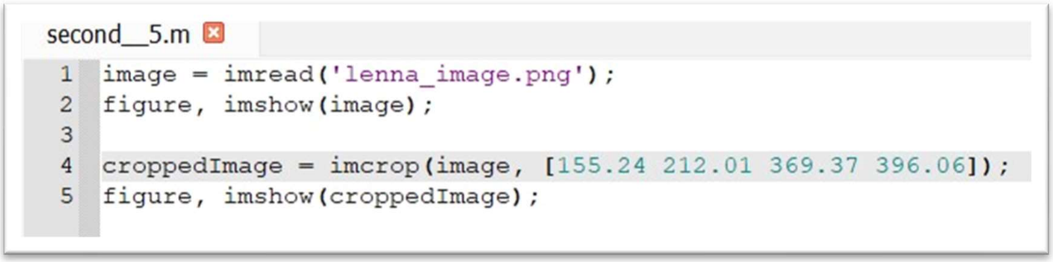
```
image = imread('lenna_image.png');
figure, imshow(image);

underExposedImage = image - 100;
figure, imshow(underExposedImage);
```

- **Output:**



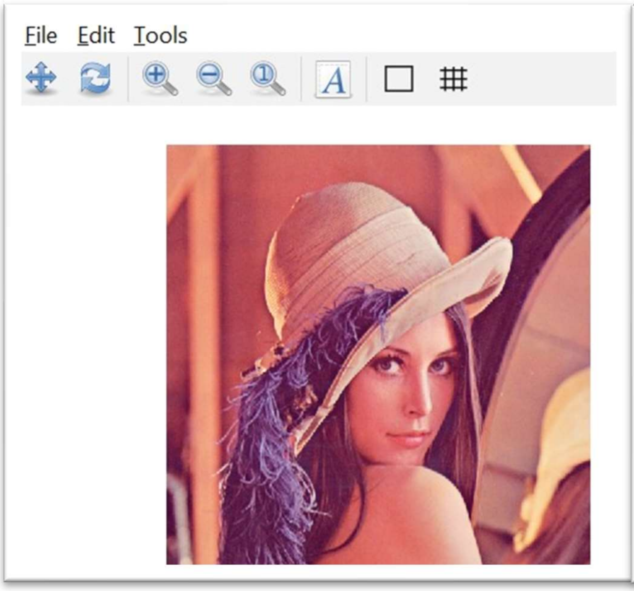
e) Keep your face only and crop the rest of the part.

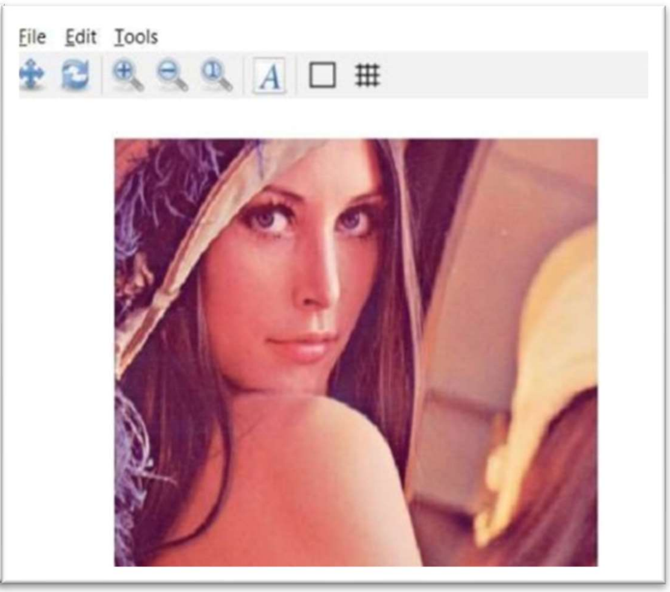
- **Code:**



```
image = imread('lenna_image.png');
figure, imshow(image);

croppedImage = imcrop(image, [155.24 212.01 369.37 396.06]);
figure, imshow(croppedImage);
```

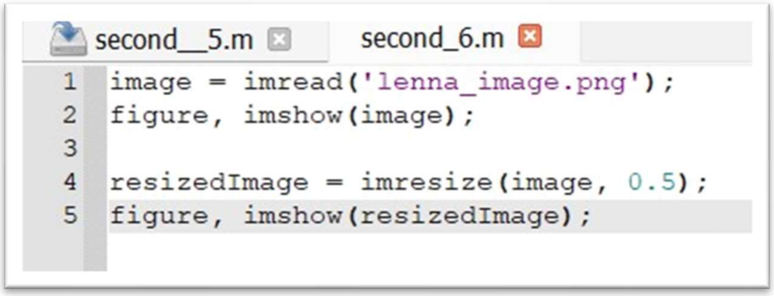
- **Output:**





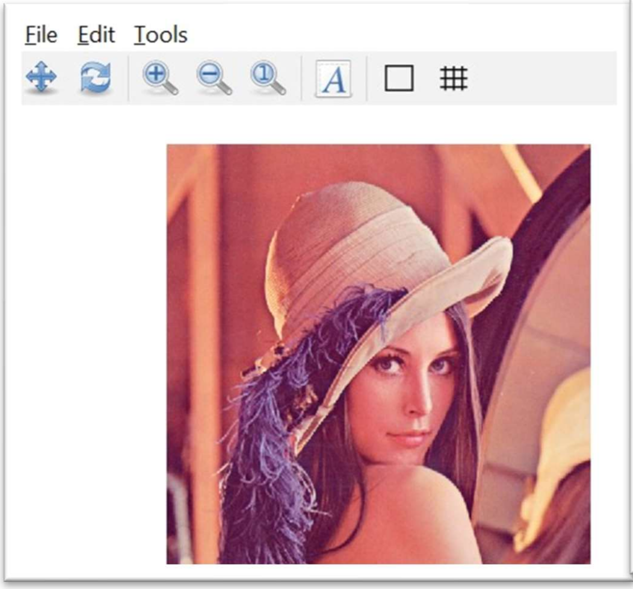
f) Resize the image to 256 X 256.

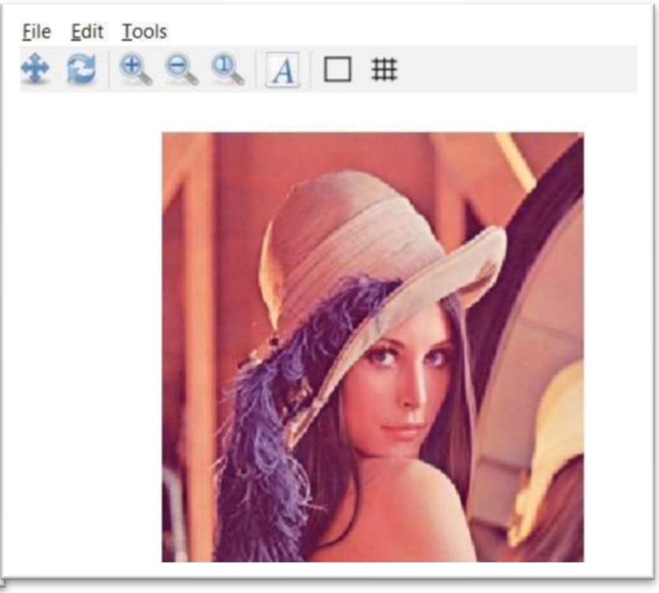
- **Code:**



```
image = imread('lenna_image.png');
figure, imshow(image);

resizedImage = imresize(image, 0.5);
figure, imshow(resizedImage);
```

- **Output:**

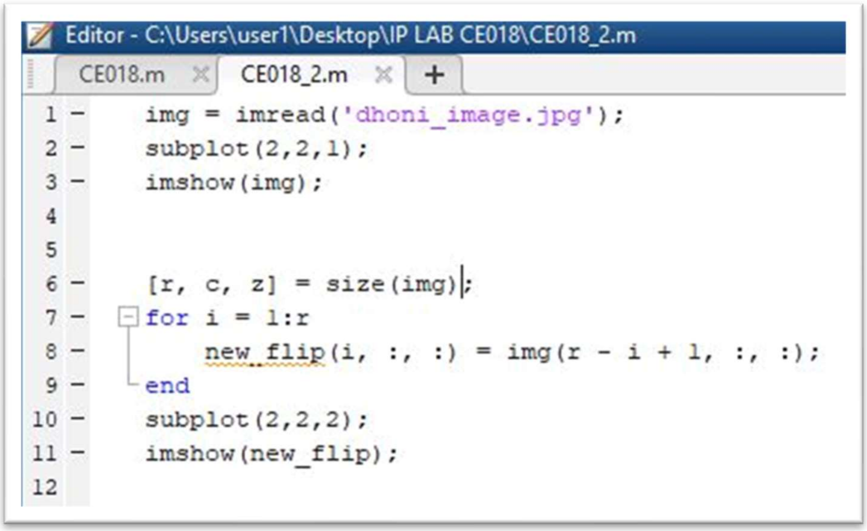




Q. 3: Take your own photo and process them for following results using loop controlling structures.

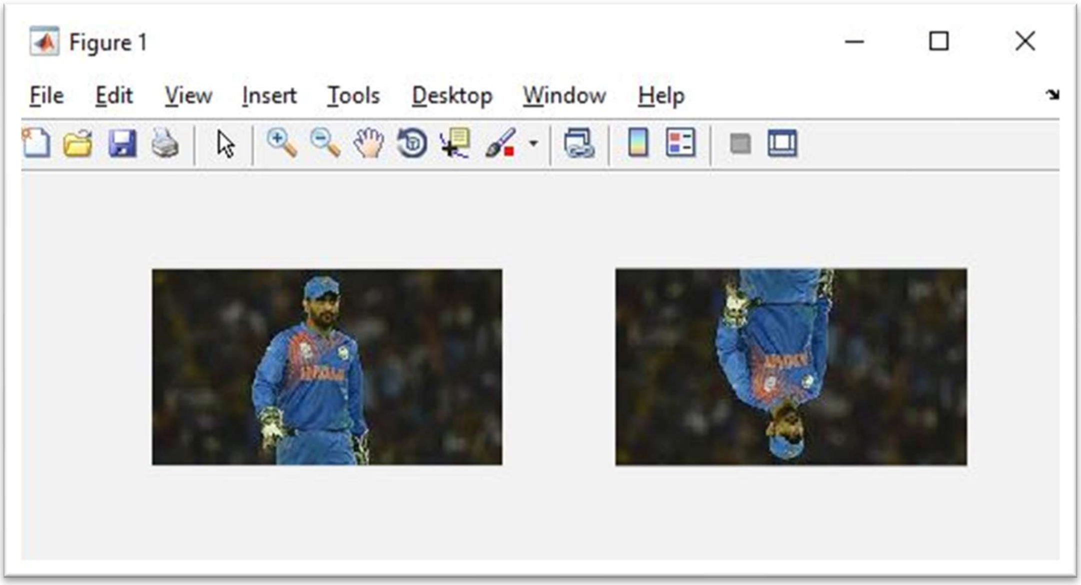
a) Flip your image vertically.

- **Code:**



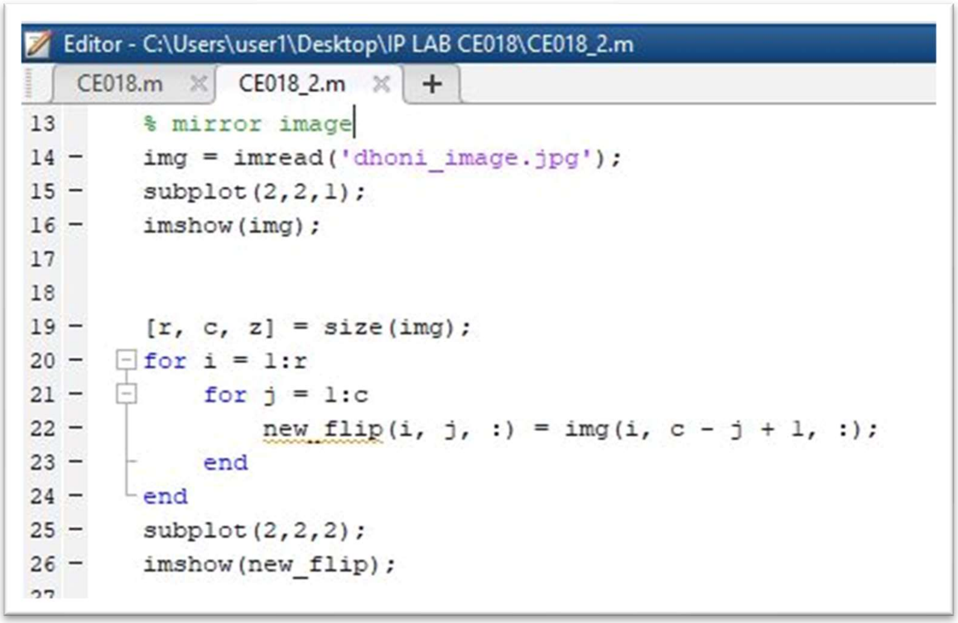
```
img = imread('dhoni_image.jpg');
subplot(2,2,1);
imshow(img);


[r, c, z] = size(img);
for i = 1:r
    new_flip(i, :, :) = img(r - i + 1, :, :);
end
subplot(2,2,2);
imshow(new_flip);

```

- **Output:**



b) Create the mirror image.

- **Code:**

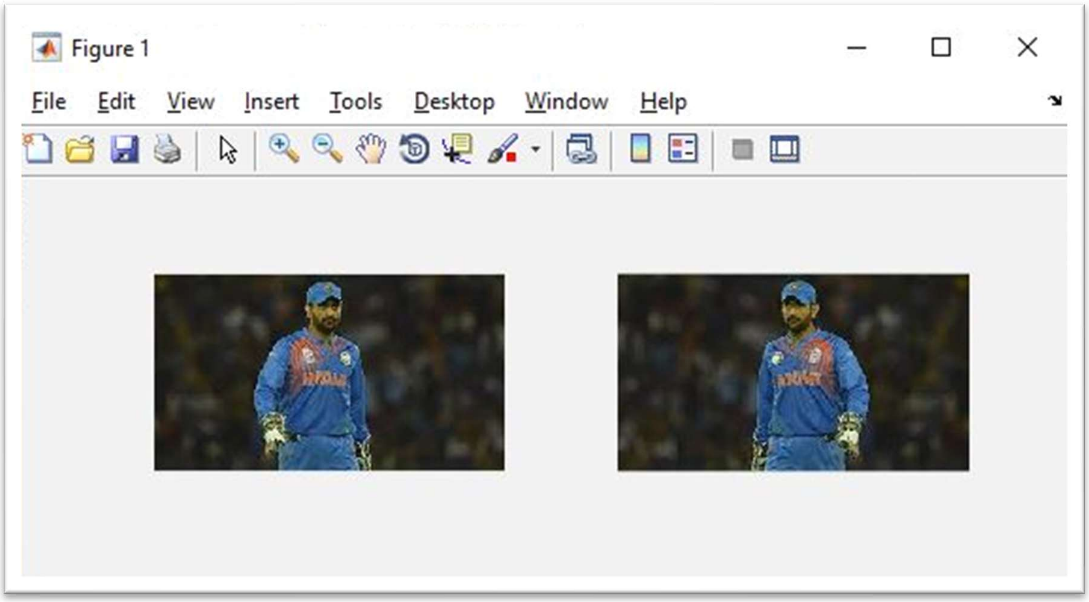


Editor - C:\Users\user1\Desktop\IP LAB CE018\CE018_2.m

CE018.m CE018_2.m

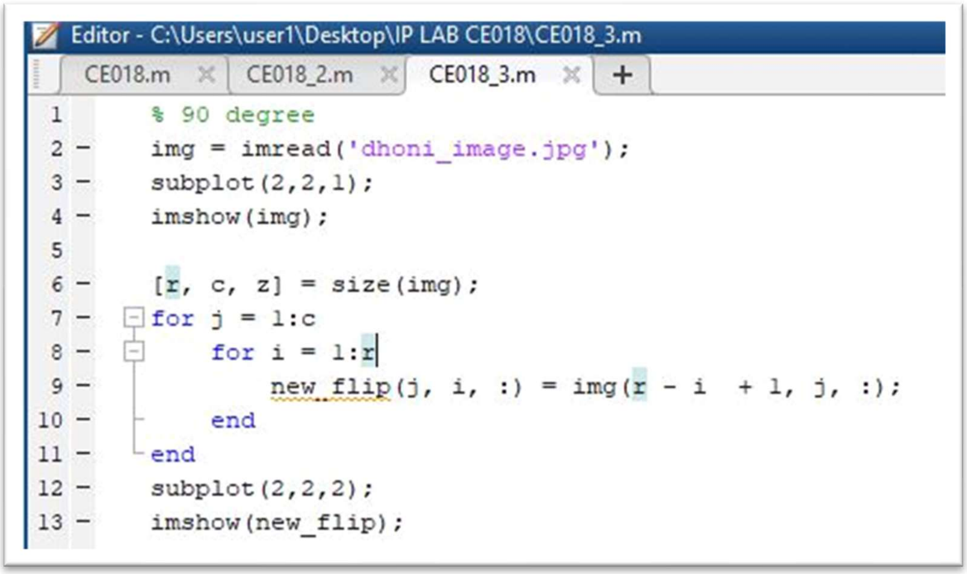
```
% mirror image
img = imread('dhoni_image.jpg');
subplot(2,2,1);
imshow(img);


[r, c, z] = size(img);
for i = 1:r
    for j = 1:c
        new_flip(i, j, :) = img(i, c - j + 1, :);
    end
end
subplot(2,2,2);
imshow(new_flip);
```

- **Output:**



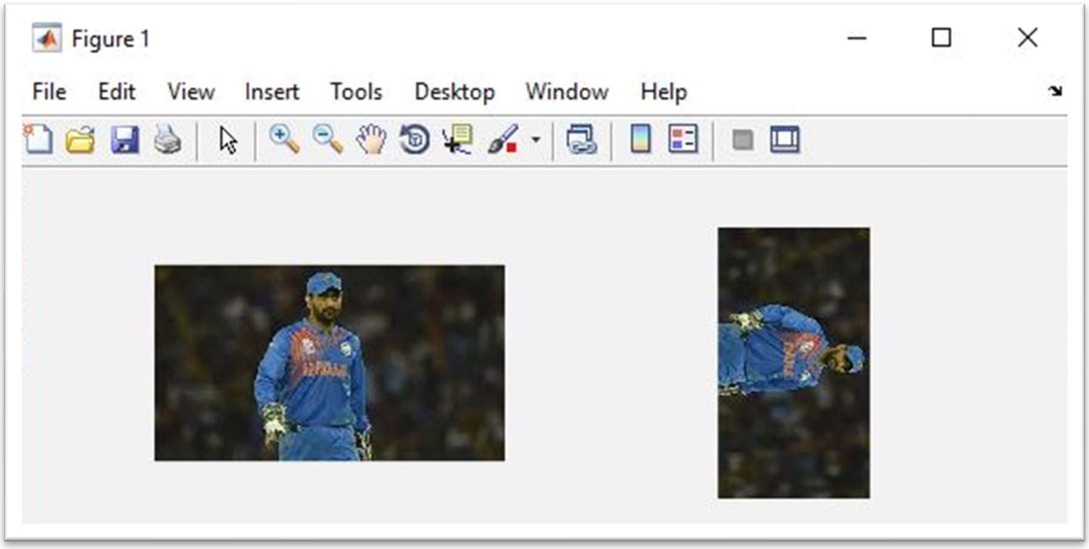
c) Rotate the image by 90 degrees.

- **Code:**



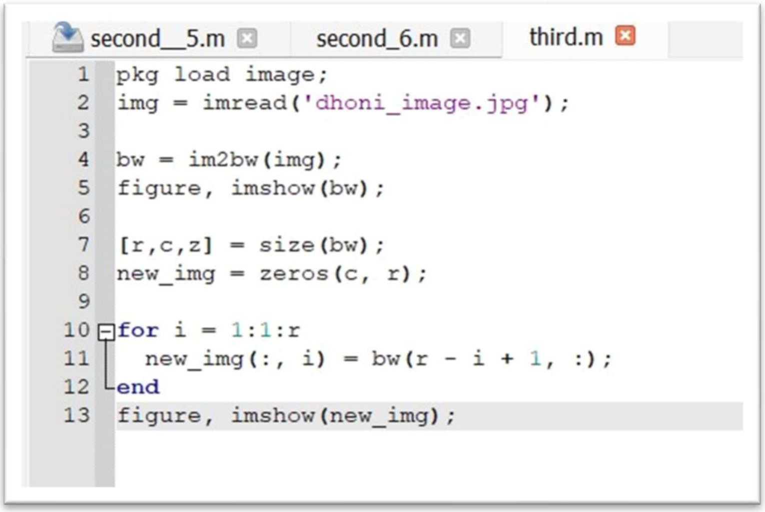
```
% 90 degree
img = imread('dhoni_image.jpg');
subplot(2,2,1);
imshow(img);

[r, c, z] = size(img);
for j = 1:c
    for i = 1:r
        new_flip(j, i, :) = img(r - i  + 1, j, :);
    end
end
subplot(2,2,2);
imshow(new_flip);
```

- **Output:**



d) Rotate the image by 270 degrees.

- **Code:**



```
pkg load image;
img = imread('dhoni_image.jpg');

bw = im2bw(img);
figure, imshow(bw);

[r,c,z] = size(bw);
new_img = zeros(c, r);

for i = 1:1:r
  new_img(:, i) = bw(r - i + 1, :);
end
figure, imshow(new_img);
```

- **Output:**

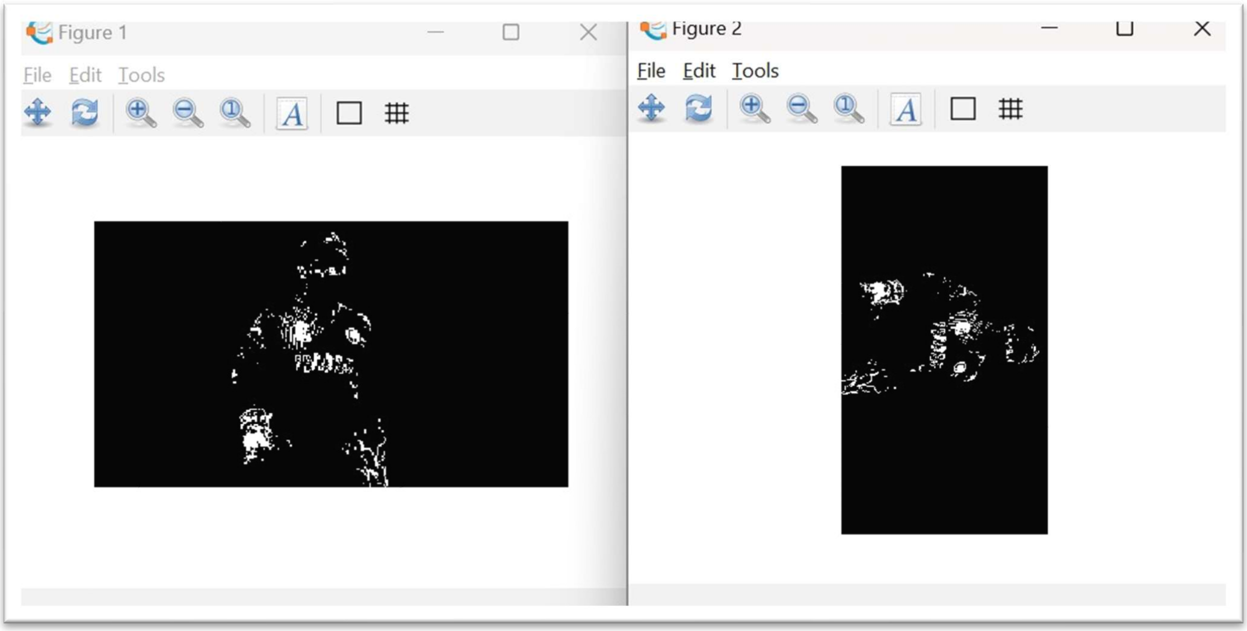


Figure 1
File Edit Tools
Figure 2
File Edit Tools