LAB 1

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

MATLAB BASICS

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

   Use the matrix A to:

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

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

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

   d) Create a six element row vector named *vd* that contains the elements of the second and fourth columns of *A*.

a)

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

va =

    -5     6    34   -21
```

b)

```
>> vb=[a(:,3)']

vb =

    22    34   -18

>>
```

c)

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

vc =

    43    21    22    11    12    17   -18    42

>>
```

d)

```
>> vc=[a(:,2)',a(:,4)']

vc =

    21     6    17    11   -21    42

>>
```

2. Create the following three matrices:

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

a) Calculate $A + B$ and $B + A$ to show that addition of matrices is commutative.
b) Calculate $A + (B + C)$ and $(A + B) + C$ to show that addition of matrices is associative.
c) Calculate $3(A + C)$ and $3A + 3C$ to show that, when matrices are multiplied by a scalar, the multiplication is distributive.
d) Calculate $A * (B + C)$ and $A * B + A * C$ to show that matrix multiplication is distributive.

a)

```
>> A+B

ans =

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

>> B+A

ans =

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

b)

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

ans =

    21    18     3
     1     5    24
     7    -9   -21

>> (A+B)+C

ans =

    21    18     3
     1     5    24
     7    -9   -21

>>
```

c)

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

ans =

    33    33     0
    36     0    48
     9   -18   -42

>> 3*A + 3*C

ans =

    33    33     0
    36     0    48
     9   -18   -42

>>
```

d)

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

ans =

   102    76   -29
    85   -66  -194
   -23    28    49

>> A*B+A*C

ans =

   102    76   -29
    85   -66  -194
   -23    28    49

>>
```

3. Create an array A = [1 2 3 4 5 6] and using built in functions for array find

a. length of A

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

A =

     1     2     3     4     5     6

>> length(A)

ans =

     6

fx >>
```

b. average of the elements of A

```
>> mean(A)

ans =

    3.5000

fx >>
```

c. Maximum element of A

```
>> max(A)

ans =

     6

fx >>
```

d. Minimum element of A

```
>> min(A)

ans =

     1

fx >>
```

e. Sum of all the elements of A

```
>> sum(A)

ans =

    21

>>
```

4. Calculate:

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

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

ans =

   20.3444
```

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.

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

A =

     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

1. Inbuilt functions for image processing in MATLAB

imread() imshow() imwrite() figure() subplot()	size() imresize() imcrop()	imfinfo() rgb2gray() im2bw

imread():-
Read images from various file formats. Read an image as a matrix from the file FILENAME or from the online resource URL.
imshow():-
Display the image IM, where IM can be a 2-dimensional (grayscale image) or a 3-dimensional (RGB image) matrix.
imwrite():-
Write images in various file formats.
figure():-
Create a new figure window for plotting.
subplot():-
Set up a plot grid with ROWS by COLS subwindows and set the current axes for plotting ('gca') to the location given by INDEX.
size():-
Return a row vector with the size (number of elements) of each dimension for the object A.
imresize():-
Scales the image IM by a factor SCALE or into the size M rows by N columns.
imcrop():-
Displays the image IMG in a figure window and waits for the user to select two points defining a bounding box.
imfinfo():-
Returns a structure containing information about the image stored in the file FILENAME.
rgb2gray():-

Transform an image or colormap from red-green-blue (RGB) color space to a grayscale intensity image.

im2bw():-

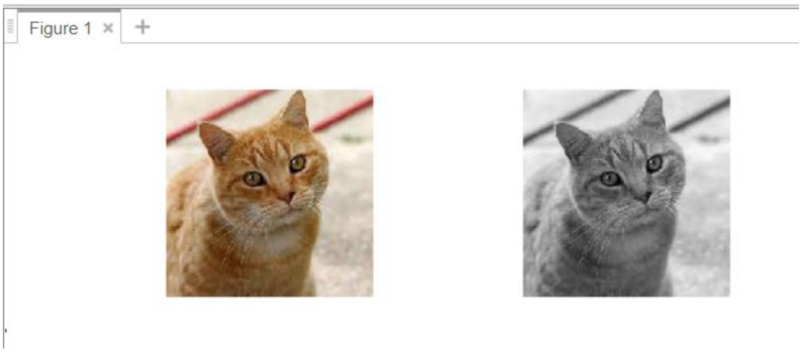
Convert image to binary, black and white, by threshold.

2. Take your own photo (RGB image) and create the following images and save them for future use

a) Gray scale image

lab1.m

```
img=imread('cat.jpg')
subplot(2,2,1)

imshow(img)

temp = rgb2gray(img)

subplot(2,2,2)
imshow(temp)
```

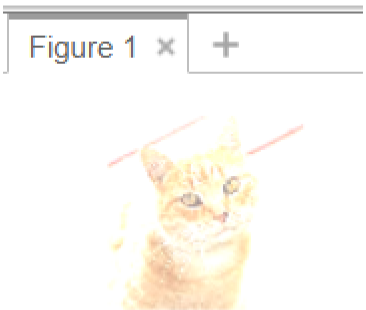


b) Black and White image

```
lab1.m
img=imread('cat.jpg')
subplot(2,2,1)

imshow(img)

temp = im2bw(img)

subplot(2,2,2)
imshow(temp)
```

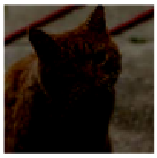
c) Over exposed image

```
lab1.m
img=imread('cat.jpg')
subplot(2,2,1)

imshow(img)

img=img+150;
imshow(img);
```



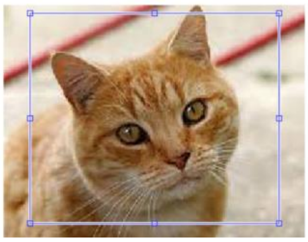
d) Under exposed image

```
img=imread('cat.jpg')
subplot(2,2,1)

imshow(img)

img=img-150;
imshow(img);
```



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

```
img=imread('cat.jpg')

imshow(img)

cp=imcrop(img)
imshow(cp)
```



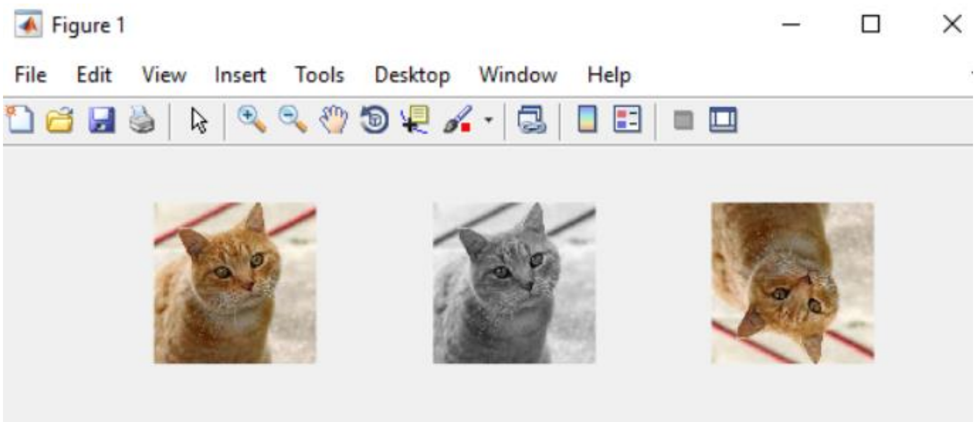
f) Resize the image to 256 x 256

```
img=imread('cat.jpg')

imshow(img)

cp=imresize(img,[256 256])
imshow(cp)
```

3. Take you own photo and process them for following results using loop controlling structures.

a. Flip your image vertically

```
img=imread('cat.jpg');
subplot(3,3,1);

imshow(img);

temp = rgb2gray(img);

subplot(3,3,2);
imshow(temp);

%flip
[r,c,z]=size(img);
for i = 1:r
      new_flip(i,:,:) = img(r-i+1,:,:);
end
subplot(3,3,3);

imshow(new_flip);
```



b. Create the mirror image

```
img=imread('cat.jpg')

imshow(img)

for i=1:1:200
    B(:,200-i+1)=img(:,i)
end
imshow(B)
```

c. Rotate the image by 90 degrees.

lab1.m

```
img=imread('cat.jpg')

imshow(img)
[r,c,z]=size(img)

for i=1:1:r
    for j=1:1:r
        B(j,r-i+1)=img(i,j);
    end
end
imshow(B)
```

d. Rotate the image by 270 degrees.

lab1.m

```
img=imread('cat.jpg')

imshow(img)
[r,c,z]=size(img)
img=rgb2gray(img)
for i=1:1:200
    for j=1:1:200
        B(200-j+1,i)=img(i,j);
    end
end
imshow(B)
```

Figure 1