IMAGE PROCESSING LAB-1

Different operation on Image:

- **imread ("filename"):**
  - Read an image as a matrix from the file name or from the online resource url.

- **imwrite (img,"filename"):**
  - Write images in various file formats. The image img can be binary, grayscale, RGB, or multi-dimensional image.

- **rgb2gray (rgb_img):**
  - Transform an image or colormap from red-green-blue (RGB) color space to a grayscale intensity image. The input may be of class uint8, uint16, int8, int16, single or double. The output is of the same class as the input.

- **im2bw (img):**
  - Convert image to binary, black and white, by threshold. The input image img can be either be a grayscale or RGB image.

- **imshow(img):**
  - Display the imge img, where img can be a 2-dimensional (grayscale image) or a 3-dimensional (RGB) matrix.

- **imresize(img,scale):**
  - Resize image with interpolation. Scales image img by a factor scale or into the size M rows by N columns.

Question:1

Create the following matrix A: $A = \begin{bmatrix} 6 & 43 & 2 & 11 & 87 \\ 12 & 6 & 34 & 0 & 5 \\ 34 & 18 & 7 & 41 & 9 \end{bmatrix}$

Use the matrix A to:

a) Create a five-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 fourth column of A.

c) Create a ten-element row vector named `vc` that contains the elements of the first and second rows of A.

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

Question:2

Create the following three matrices:

$$A = \begin{bmatrix} 5 & 2 & 4 \\ 1 & 7 & -3 \\ 6 & -10 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 11 & 5 & -3 \\ 0 & -12 & 4 \\ 2 & 6 & 1 \end{bmatrix} \qquad C = \begin{bmatrix} 7 & 14 & 1 \\ 10 & 3 & -2 \\ 8 & -5 & 9 \end{bmatrix}$$

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 $5(A+C)$ and $5A+5C$ 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.

```
     6    43     2    11    87
    12     6    34     0     5
    34    18     7    41     9

>> A(2,:)
ans =

    12     6    34     0     5

>> A(:,4)
ans =

    11
     0
    41

>> A(1:2,;)
parse error:

  syntax error

>>> A(1:2,;)
          ^

>> A(1:2,:)
ans =

     6    43     2    11    87
    12     6    34     0     5

>> vd=[A(:,2)' A(:,5)']
vd =

    43     6    18    87     5     9
```

A) **Commutative**

B) **Associative**

```
>> (A+B)+c
ans =

    23    21     2
    11    -2    -1
    16    -9    10

>> A+(B+c)
ans =

    23    21     2
    11    -2    -1
    16    -9    10

>> B=[11 5 -3; 0 -12 4; 2 6 1]
B =

    11     5    -3
     0   -12     4
     2     6     1

>> c=[7 14 1; 10 3 -2; 8 -5 9]
c =

     7    14     1
    10     3    -2
     8    -5     9

>> A+B
ans =

    16     7     1
     1    -5     1
     8    -4     1

>> B+A
ans =

    16     7     1
     1    -5     1
     8    -4     1
```

C) Distributive

```
>> 5*(A+c)
ans =

    60    80    25
    55    50   -25
    70   -75    45

>> 5*A+5*c
ans =

    60    80    25
    55    50   -25
    70   -75    45
```

D) Distributive

```
>> A*(B+c)
ans =

   150    81    34
    58   -47   -18
     8   204   -32

>> A*B+A*c
ans =

   150    81    34
    58   -47   -18
     8   204   -32
```

Question:3

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

```
>> res=((power(3,7)*log(76))/(power(7,3)+546))+cbrt(910)
res =  20.344
>>
```

Question:4

Using the ones and Zeros Commands, create 4 * 5 matrix in Which the first 2 rows are 0's and the next 2 rows are 1's.

```
>> p=zeros(2,5)
p =

     0     0     0     0     0
     0     0     0     0     0

>> q=ones(2,5)
q =

     1     1     1     1     1
     1     1     1     1     1

>> ans=[p;q]
ans =

     0     0     0     0     0
     0     0     0     0     0
     1     1     1     1     1
     1     1     1     1     1
```

Question 5:

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

1) Gray scale image

2)Black and white image

3)Over Exposed image

4)Under Exposed image

5) keep your face only-crop rest of the image.

6) Resize the image to 256*256.

Code:

```
pkg load image
img=imread('D:\IP\xr.jpeg');
imshow(img)
img_gray=rgb2gray(img);
imwrite(img_gray,'D:\IP\jaydeep.jpg');
img_bw=im2bw(img);
imwrite(img_bw,'D:\IP\BWjaydeep.jpg');
img_over_exposed=img+70;
imwrite(img_over_exposed,'D:\IP\oejaydeep.jpg');
img_under_expo=img-60;
imwrite(img_under_expo,'D:\IP\uejaydeep.jpg');
img_face=img(40:408,236:604,:);
imwrite(img_face,'D:\IP\cropjaydeep.jpg');
rsize=imresize(img,[256,256]);
imwrite(rsize,'D:\IP\rsizejaydeep.jpg');
```

Original Image:

Gray Scale Image:

Black & White:

Over exposed image:

Under exposed:

Cropped Face only:

Resized:

Question 6:

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

1) flip your image vertically.

2) create the mirror image.

3) rotate the image by 90 degree.

4) rotate the image by 270 degree.

1) Code:

```
img = imread('D:\IP\xr.jpeg');
[x, y, z] = size(img);
for plane = 1 : z;
  len = x;
   for i = 1 : x;
     for j = 1 : y;
       if i < x/2;
         temp = img(i, j, plane);
         img(i, j, plane) = img(len, j, plane);
         img(len, j, plane) = temp;
       end;
     end;
     len = len - 1;
   end;
end;
imwrite(img,'D:\IP\flliped.jpg');
```

Output:

2)Code:

```
img_mirror=uint8(zeros(size(img)));
img = imread('D:\IP\xr.jpeg');
pkg load image
for i=1:1280
 for j=1:853
 img_mirror(i,j,:)=img(i,854-j,:);
 endfor
endfor
imwrite(img_mirror,'D:\IP\mirror.jpg');
```

Output:

3) Code:

```
pkg load image
img=imread('D:\IP\xr.jpeg');
img_90=uint8(zeros(853,1280,3));
for i=1:853
 for j=1:1280
    img_90(i,j,:)=img(1281-j,i,:);
 endfor
endfor
imwrite(img_90,'D:\IP\jaydeep90.jpg');
```

Output:

4) Code:

```
pkg load image
img=imread('D:\IP\xr.jpeg');
image2_90degree=uint8(zeros(853,1280,3));
for i=1:853
 for j=1:1280
 image2_90degree(i,j,:)=img(j,854-i,:);
 endfor
endfor
imwrite(image2_90degree,'D:\IP\jaydeep270.jpg');
```

Output: