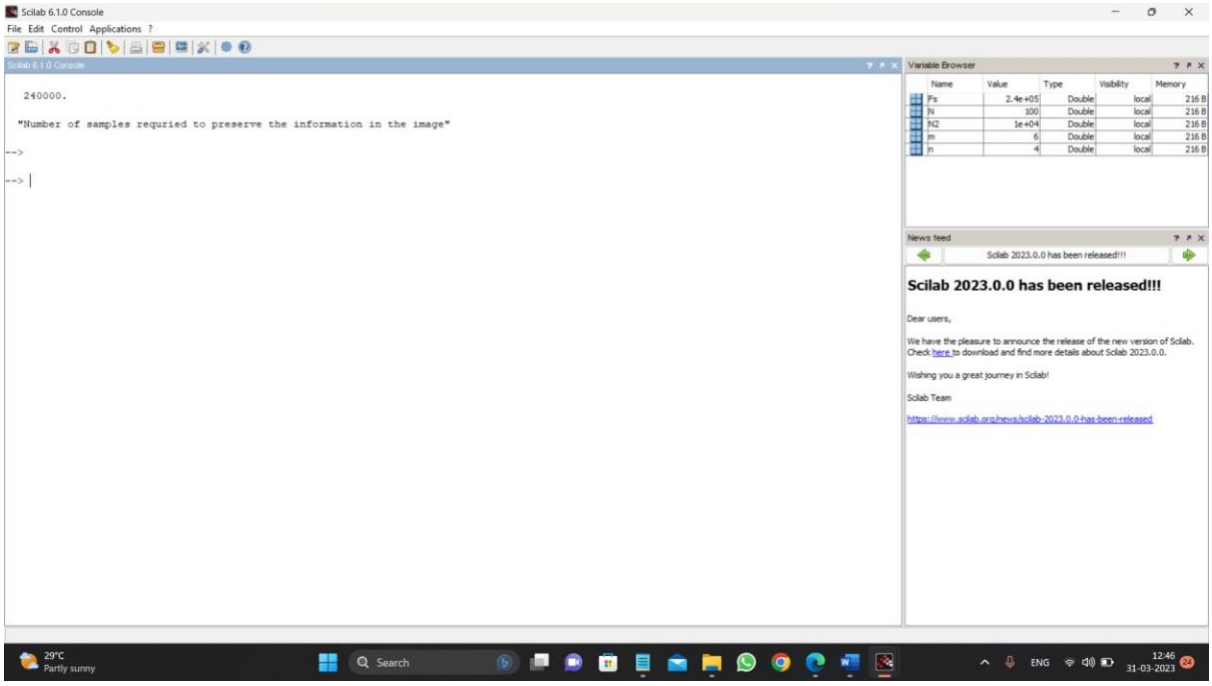
Practical 1

Aim :- Program to calculate number of samples required for an image

Code :-

```
clc ; close
;
// dimesion of the image in inches
m = 6; n = 4;
N = 100;
N2= N*N ; // Number of dots per inch in both direction
Fs= m* n * N2 ;
disp ( Fs , 'Number of samples requried to preserve the information in the image')
```

output :-



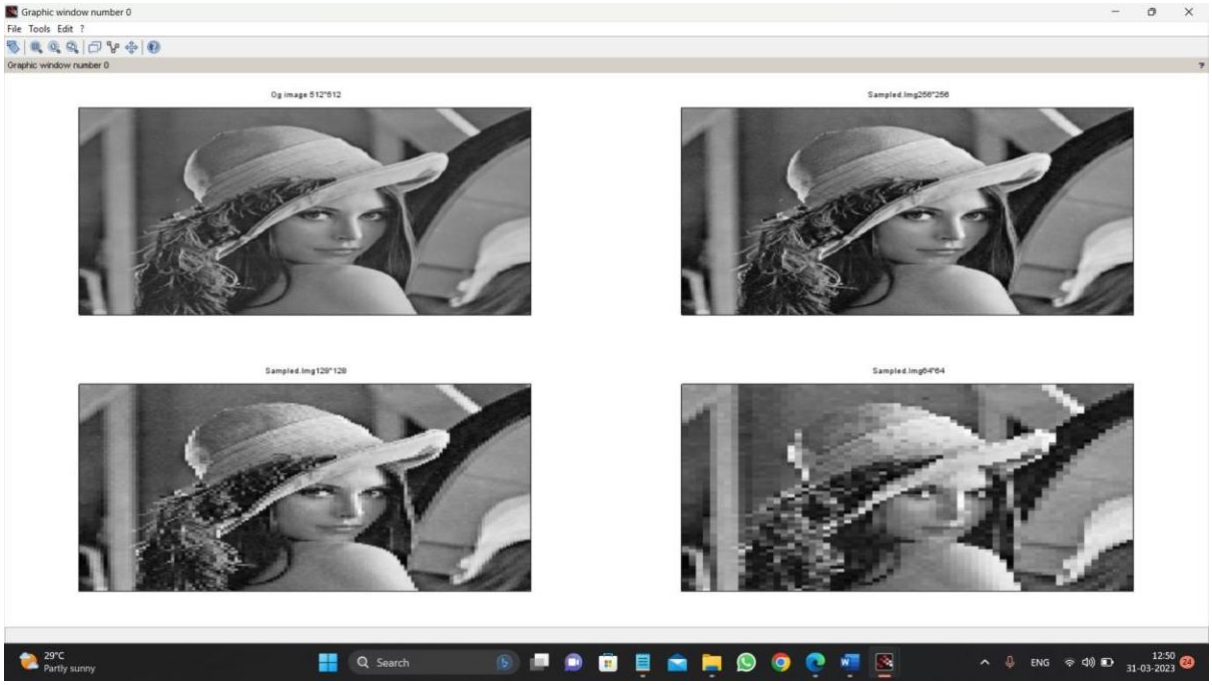
Practical 2

Aim :- Program to study the effects of reducing the spatial resolution of a digital image.

Code :-
```
clc;
clear all;
Img1=imread('lena.png');
Img = rgb2gray(Img1);
//512*512
subplot (2,2,1),imshow(Img),title('Og image 512*512');
//256*256
Samp=zeros(256); m=1;
n=1; for i=1:2:512     for
j=1:2:512
Samp(m,n)=Img(i,j);
n=n+1;                 end
n=1;        m=m+1;   end
SampImg256=mat2gray(Samp);
subplot(2,2,2);
imshow(SampImg256);
title('Sampled.Img256*256')

///////
Samp=zeros(128); m=1;
n=1; for i=1:4:512     for
j=1:4:512
Samp(m,n)=Img(i,j);
n=n+1;                 end
n=1;        m=m+1;   end
SampImg128=mat2gray(Samp);
subplot(2,2,3),imshow(SampImg128),title('Sampled.Img128*128')

////////////////////

Samp=zeros(64); m=1;
n=1; for i=1:8:512     for
j=1:8:512
Samp(m,n)=Img(i,j);
n=n+1;
```

```
    end
n=1;
m=m+1;
end
SampImg64=mat2gray(Samp);
subplot(2,2,4),imshow(SampImg64),title('Sampled.Img64*64')
```

Output :-



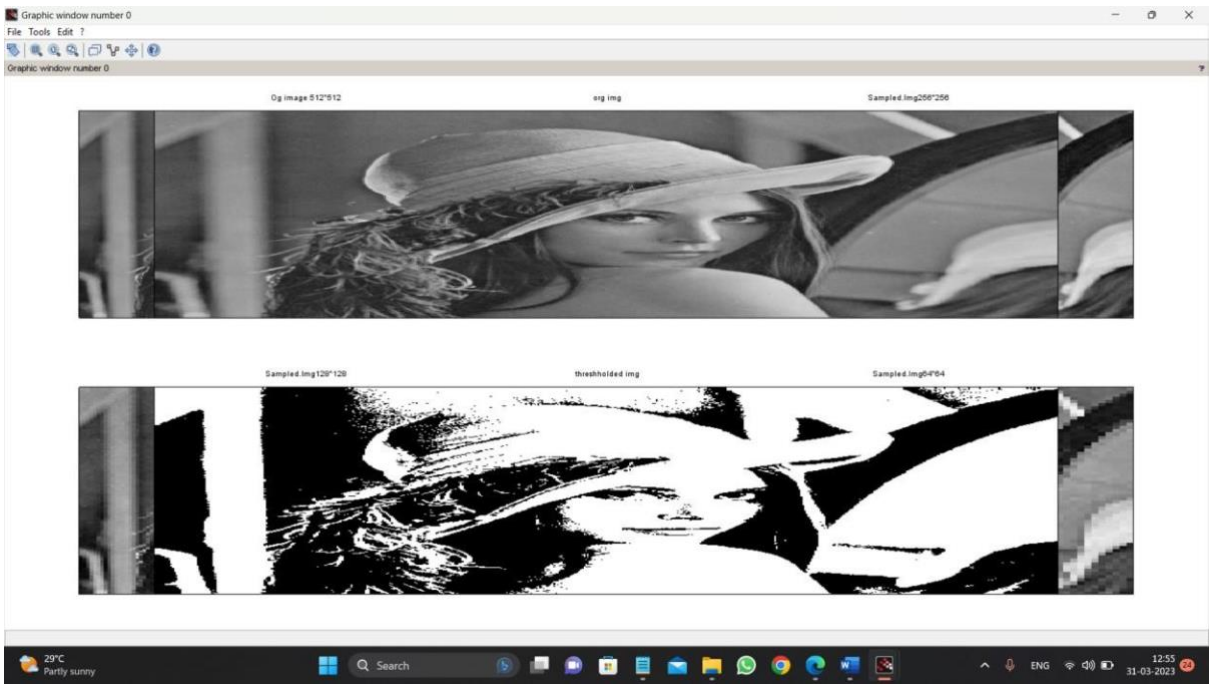
Practical 3

Aim :- Program to perform threshold on an image.

Code :-
```
clc; clear all;
a=imread('lena.jpeg');
a=rgb2gray(a);
subplot(2,1,1);
imshow(a);
title('org img');
T=100; //threshold value
[r,c]=size(a); for
i=1:r     for j=1:c
if (a(i,j)<=T)
```

```
x(i,j)=0;        else
x(i,j)=255;
   end end
end
x=uint8(x);

subplot(2,1,2);
imshow(x);
title('threshholded img');
```

Output :-

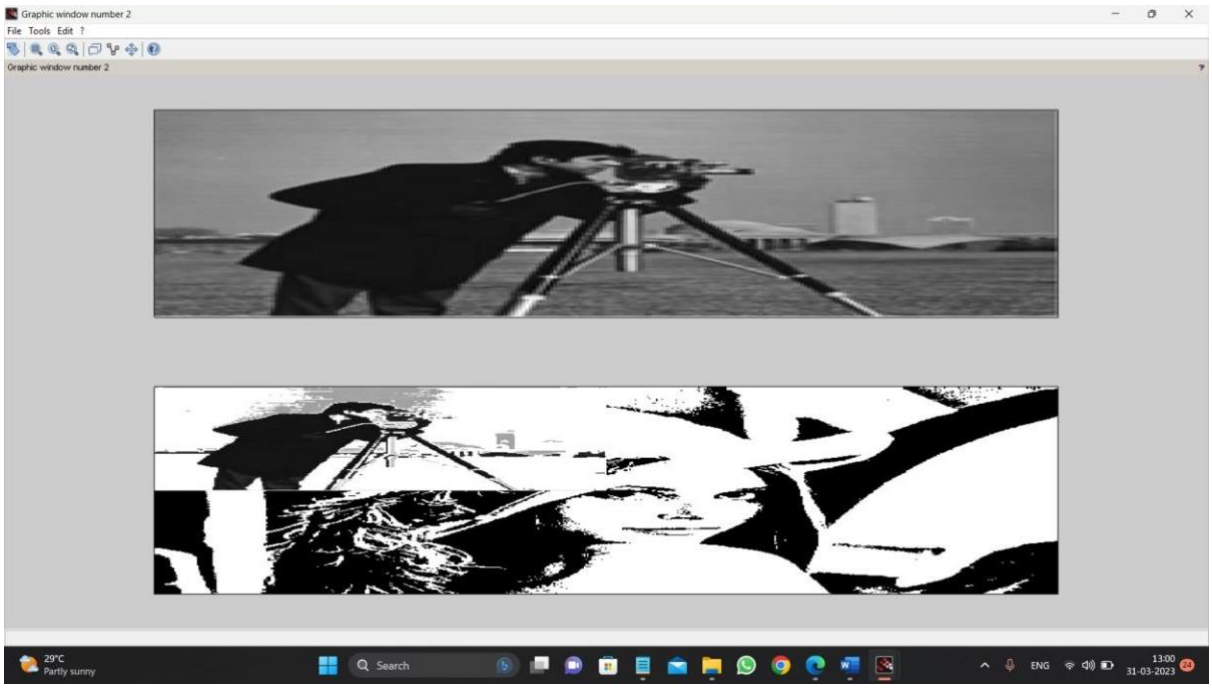


Practical 4

Aim :- Gray-level slicing with and without background.

Code :-
```
clc;
clear all;
a=imread('camera.png');

a1=58; // This value is user defined
b1=158; // This value is user defined
[r,c]=size(a); figure(2);
subplot(2,1,1);
imshow(a); for i=1:r
for j=1:c
      if (a(i,j)>a1 & a(i,j)<b1)
         x(i,j)=255;
else
         x(i,j)=a(i,j);
end     end end
x=uint8(x);
subplot(2,1,2);
imshow(x);
```

Output :-



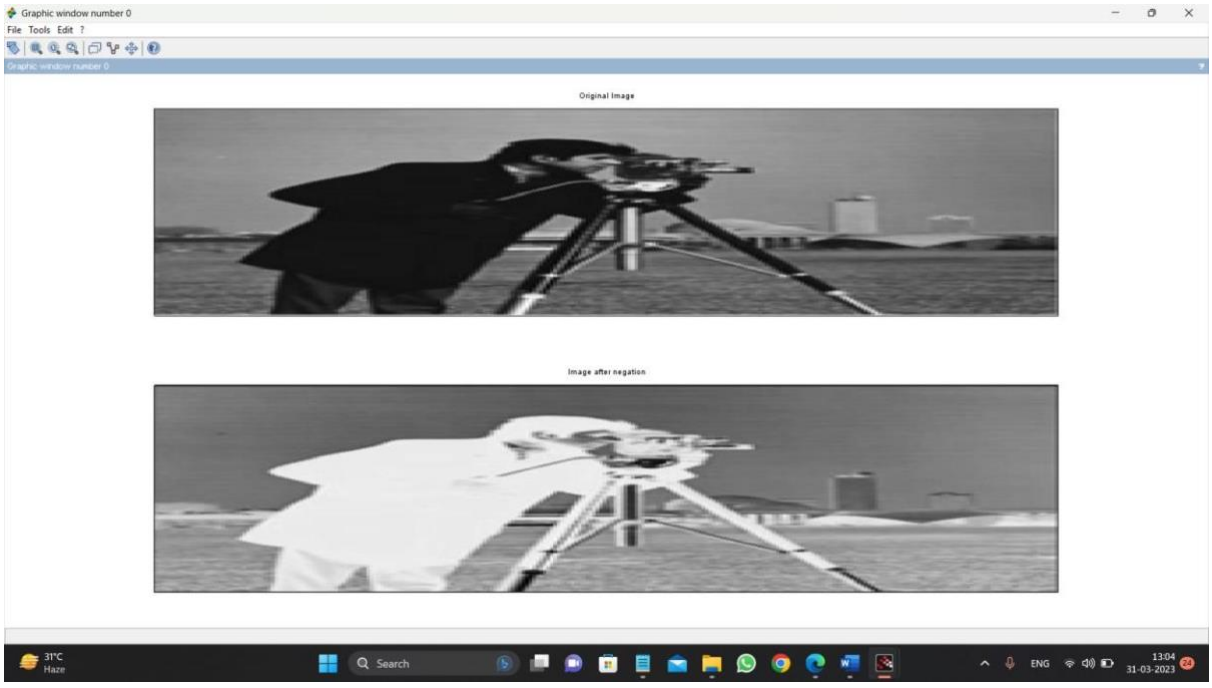
Practical 5

Aim :- Program to perform Image negation.

Code :- *//for gray image*

```
A = imread('camera.png');
A=rgb2gray(A);
subplot(2,1,1); imshow(A);
title('Original Image ');

[row col]=size(A);
for x=1:row     for
y=1:col
      A(x,y)=255-A(x,y);
   end end
subplot(2,1,2);
imshow(A);
title('Image after negation');
```

Output :-



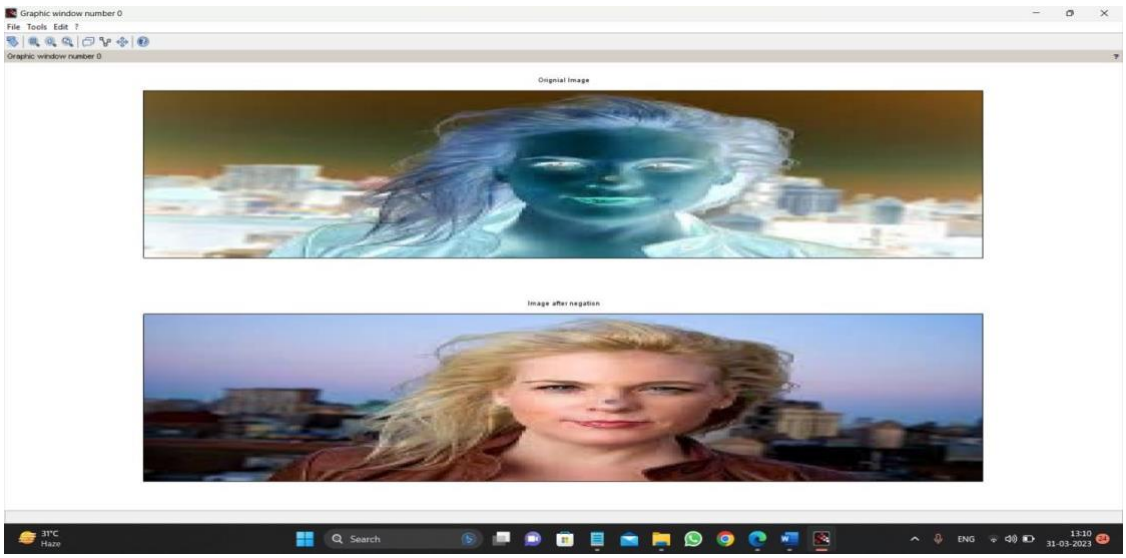
Practical 6

Aim :- Program to perform Image negation(color image).

Code :-

```
A = imread('negimg.jpg');

subplot(2,1,1);
imshow(A);
title('Orignial Image');
R = A(:,:,1);
G = A(:,:,2);
B = A(:,:,3);

[row col]=size(A);
for x=1:row     for
y=1:col
      R(x,y)=255-R(x,y);
      G(x,y)=255-G(x,y);       B(x,y)=255-
B(x,y);
   end end

A(:,:,1)=R;
A(:,:,2)=G;
A(:,:,3)=B;

subplot(2,1,2);
imshow(A);
title('Image after negation');
```

Output :-



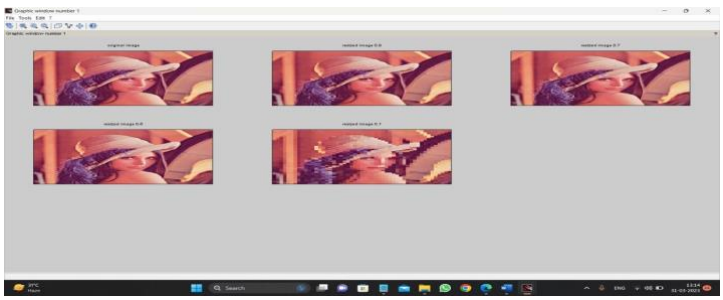
Practical 7

Aim :- Program to study the effects of varying the number of intensity levels in a digital image.

Code :-

```
clc;
clear all;
figure(1)

subplot(3,3,1);
i=imread('lena.jpeg');
imshow(i);
title('original image');
subplot(3,3,2);
j1=imresize(i,0.8);
imshow(j1);
title('resized image 0.8');
subplot(3,3,3);
j2=imresize(i,0.7);
imshow(j2);
title('resized image 0.7');
subplot(3,3,4);
j3=imresize(i,0.6);
imshow(j3);
title('resized image 0.6');
subplot(3,3,5);
j4=imresize(i,0.1);
imshow(j4);
title('resized image 0.1');
```

Output :-

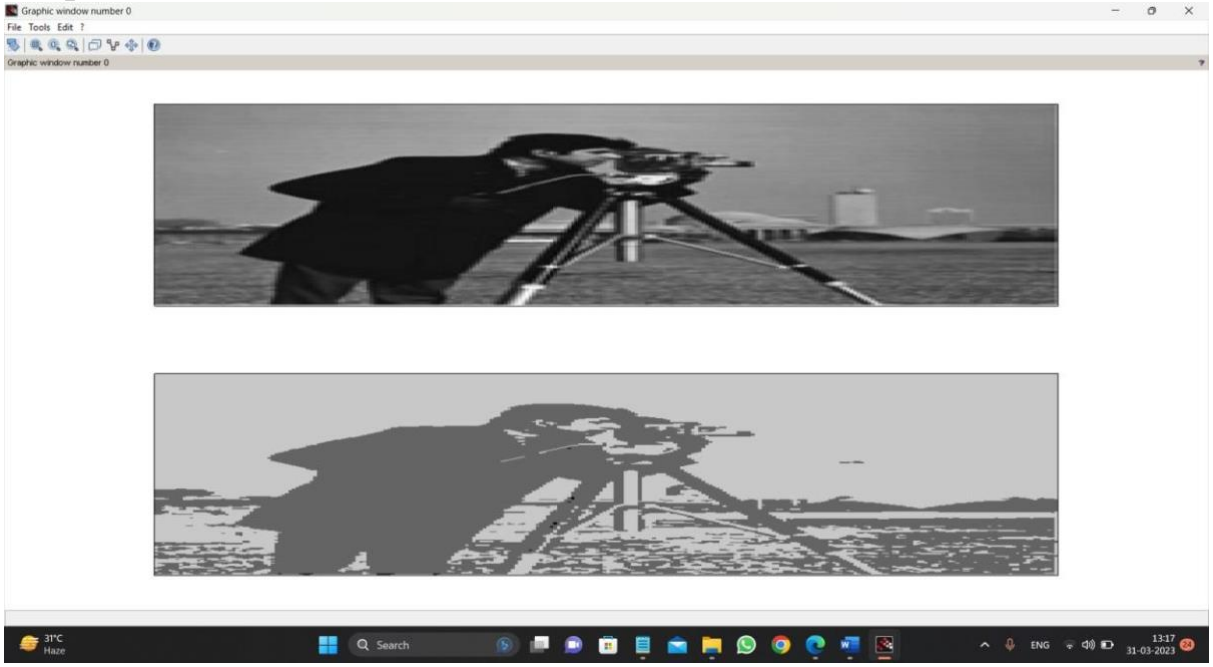


Practical 8

Aim :- Program to perform Log transformation.

Code :-
```
clc; clear all;
a=imread('camera.png');
a=rgb2gray(a);
subplot(2,1,1);
imshow(a); s=a; c=1;
[r1,c1]=size(a); for
i=1:r1     for j=1:c1
b=double(a(i,j));
s(i,j)=c*log10(1+b);
   end
end

new1=uint8(s*100);
//imshow(new1); subplot(2,1,2);
imshow(new1);
```

Output :-

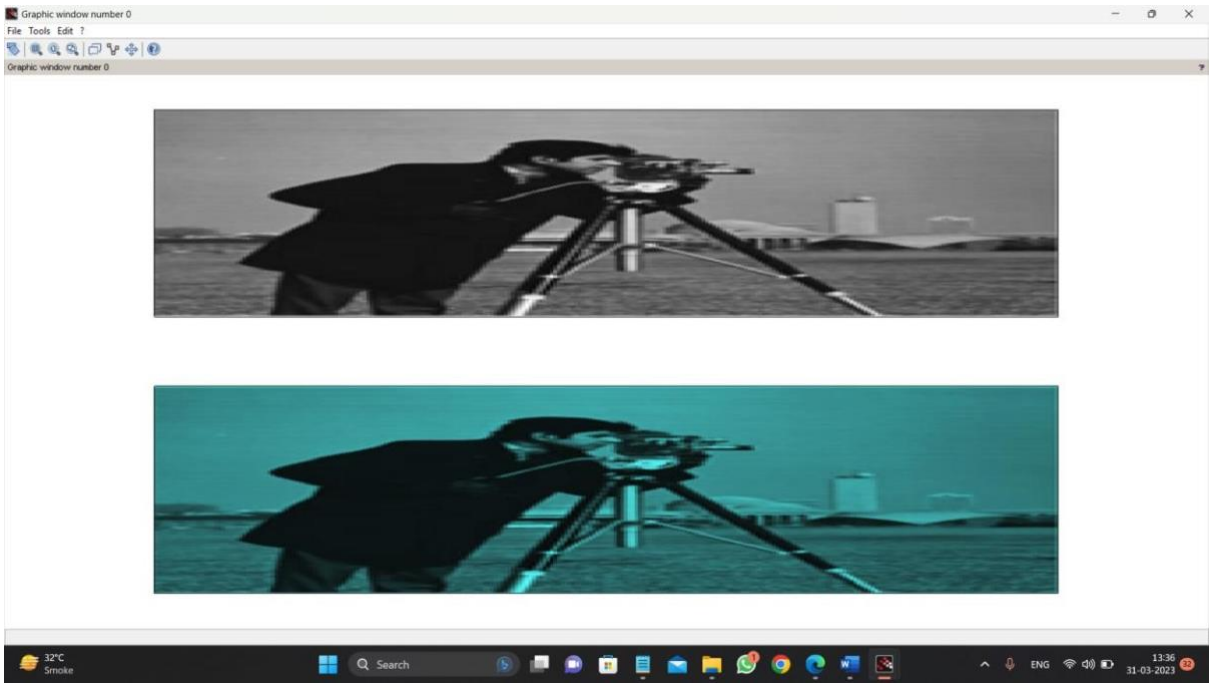


Practical 9

9-Power-law transformations. clc;

```
a=imread('camera.png');
[r,c]=size(a);
subplot(2,1,1);
imshow(a);
x=a; G=0.8;
for i=1:r     for j=1:c
b=double(a(i,j));
x(i,j)=b^G;          end
end new1=uint8(x);
subplot(2,1,2);
imshow(new1);
```

Output :-

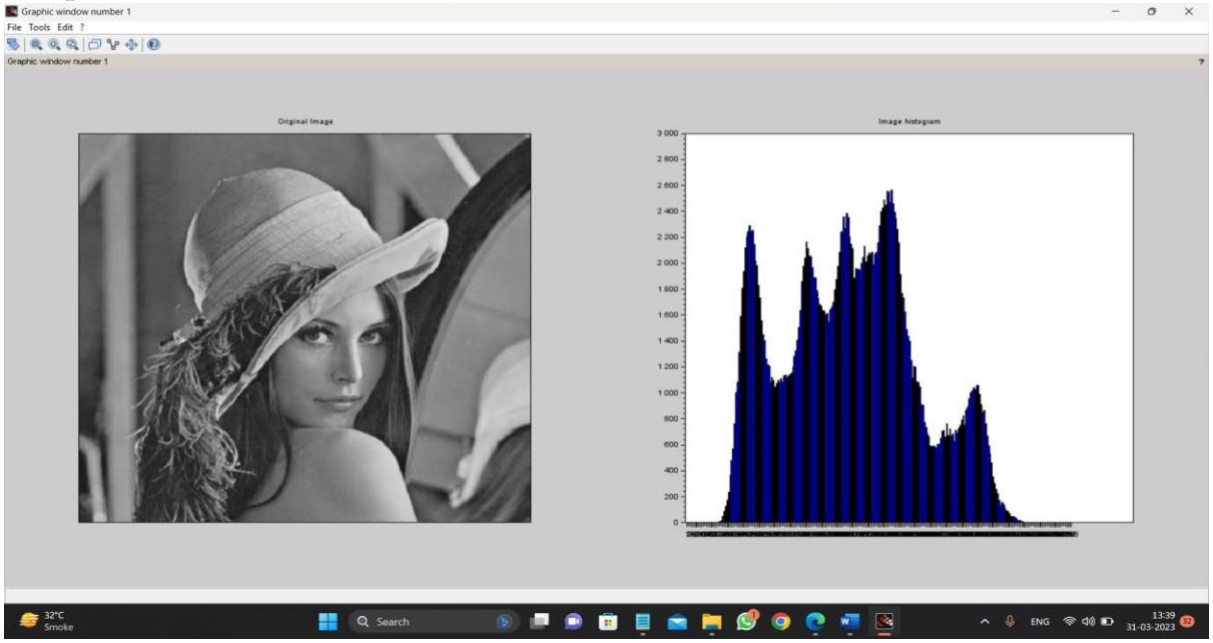


Practical 10

Aim :- Program to plot the histogram of an image.

Code :-
```
clc; clear all;
a=imread('lena.jpeg');
a=rgb2gray(a);
h=zeros(1,258);
[r,c]=size(a); for
i=1:r     for j=1:c
if (a(i,j)==0)
h(0)=h(0)+1;
       end         k=a(i,j);
h(k)=h(k)+1;     end
end figure(1);
subplot(1,2,1);
imshow(uint8(a));
title('Original Image')
subplot(1,2,2); bar(h);
title('Image histogram');
```

Output :-

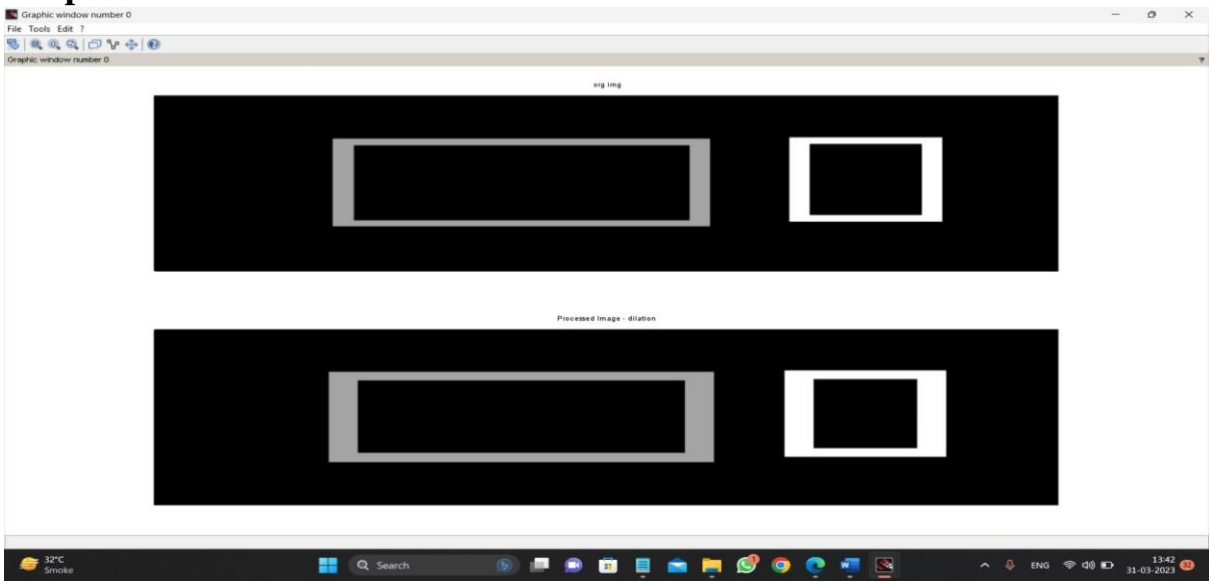


Practical 11

Program to apply dilation
```
clc;
```

```
a=imread('rectb.png');
a=rgb2gray(a); d=a;
A1=a;
[r,c]=size(d);
subplot(2,1,1);
imshow(a);
title('org img'); m=[1
1 1;1 1 1;1 1 1]; //
m=ones(5,5); for
i=2:1:r-1 for j=2:1:c-
1
new=[(m(1)*d(i-1,j-1)) (m(2)*d(i-1,j)) (m(3)*d(i-1,j+1)) (m(4)*d(i,j-1))
(m(5)*d(i,j)) (m(6)*d(i,j+1)) (m(7)*d(i+1,j-1)) (m(8)*d(i+1,j))
(m(9)*d(i+1,j+1))];
A1(i,j)=max(new);
end subplot(2,1,2);
imshow(A1);title('Processed Image - dilation'); end
```

Output :-

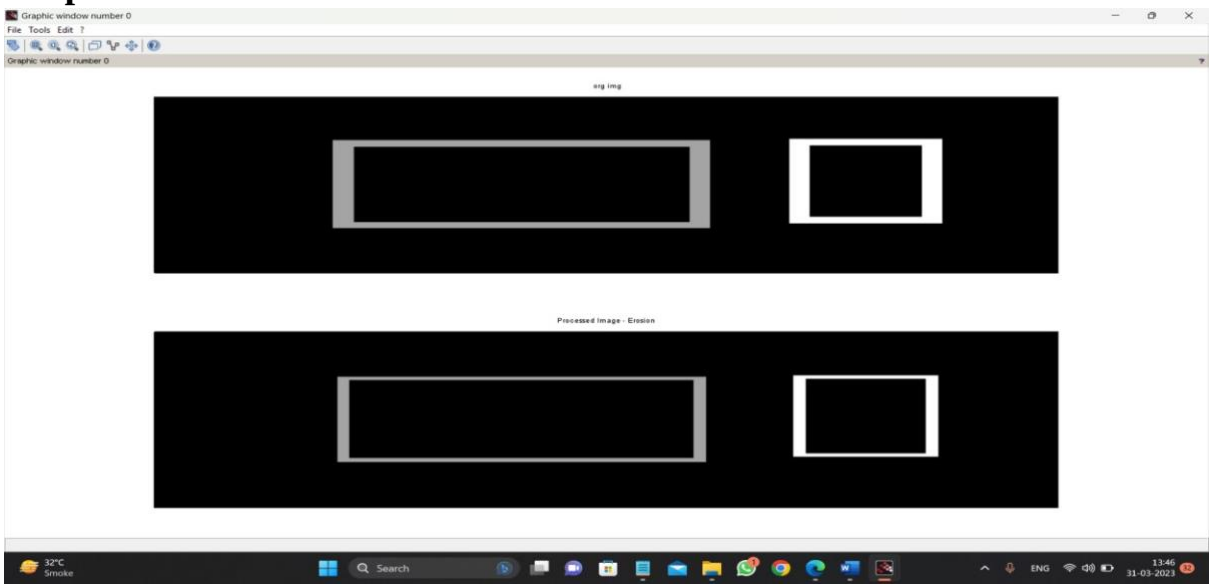


Practical 12

Program to apply erosion.

```
a=imread('rectb.png');
a=rgb2gray(a);
subplot(2,1,1);
imshow(a);
title('org img');
A1=a; d=a;
[r,c]=size(d); m=[1 1
1;1 1 1;1 1 1]; //
m=ones(5,5); for
i=2:1:r-1 for j=2:1:c-
1
   new=[(m(1)*d(i-1,j-1)) (m(2)*d(i-1,j)) (m(3)*d(i-1,j+1)) (m(4)*d(i,j-1))
(m(5)*d(i,j)) (m(6)*d(i,j+1)) (m(7)*d(i+1,j-1)) (m(8)*d(i+1,j))
(m(9)*d(i+1,j+1))];
A1(i,j)=min(new);
end subplot(2,1,2);
title('org img');imshow(A1);title('Processed Image - Erosion'); end
```

Output :-

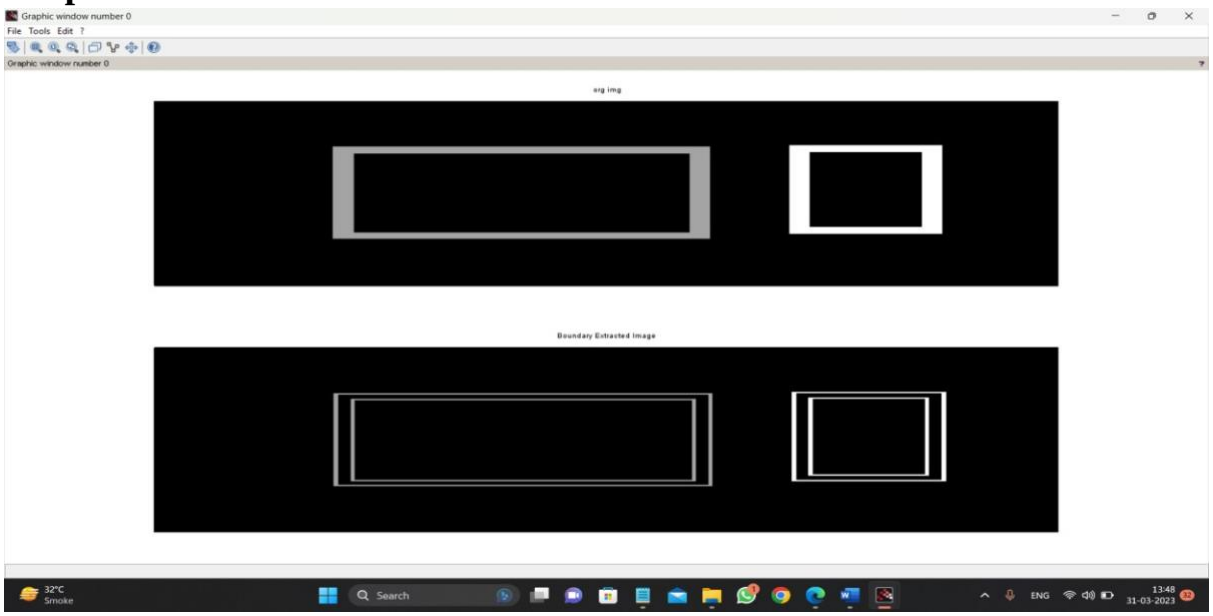


Practical 13

Program for detecting boundary of an image.

```
clc;

a=imread('rectb.png');
a=rgb2gray(a);
subplot(2,1,1);
imshow(a);
title('org img'); d=a;
[r,c]=size(d); m=[1 1
1;1 1 1;1 1 1]; for
i=2:1:r-1 for j=2:1:c-
1
new=[(m(1)*d(i-1,j-1)) (m(2)*d(i-1,j)) (m(3)*d(i-1,j+1)) (m(4)*d(i,j-1))
(m(5)*d(i,j)) (m(6)*d(i,j+1))
(m(7)*d(i+1,j-1)) (m(8)*d(i+1,j)) (m(9)*d(i+1,j+1))];
A2(i,j)=min(new);
aa(i,j)=d(i,j)-A2(i,j);
end end
subplot(2,1,2);
imshow(aa);title('Boundary Extracted Image');
```

Output :-



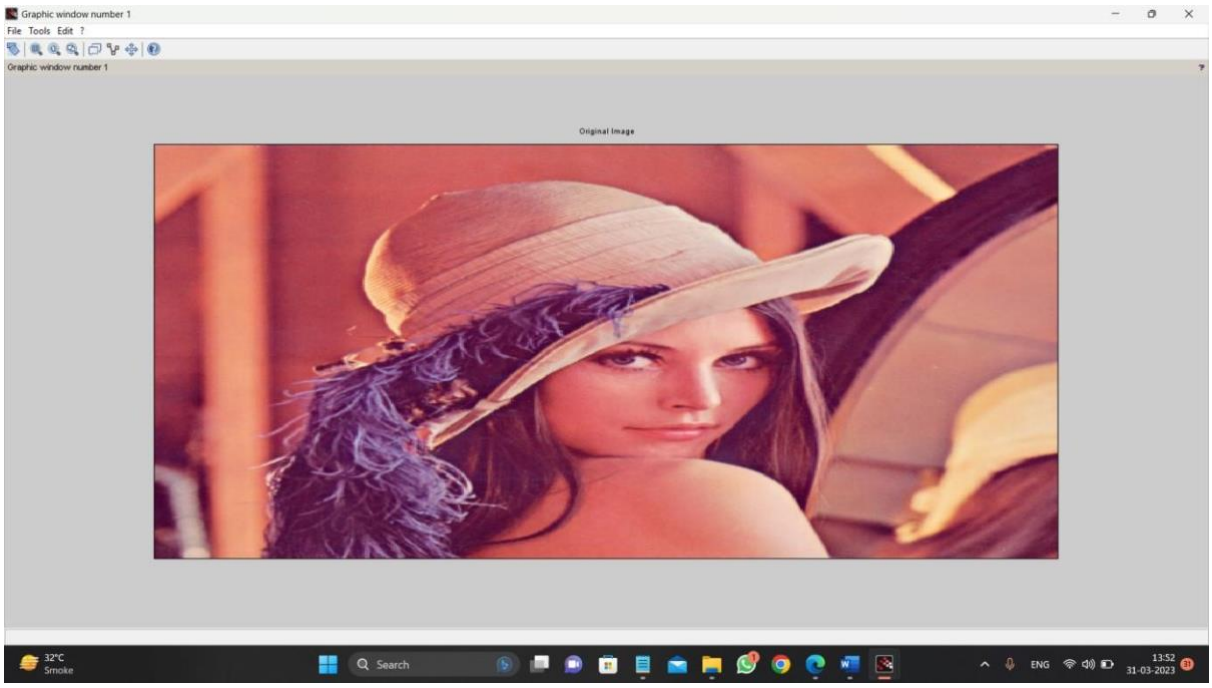
Practical 14

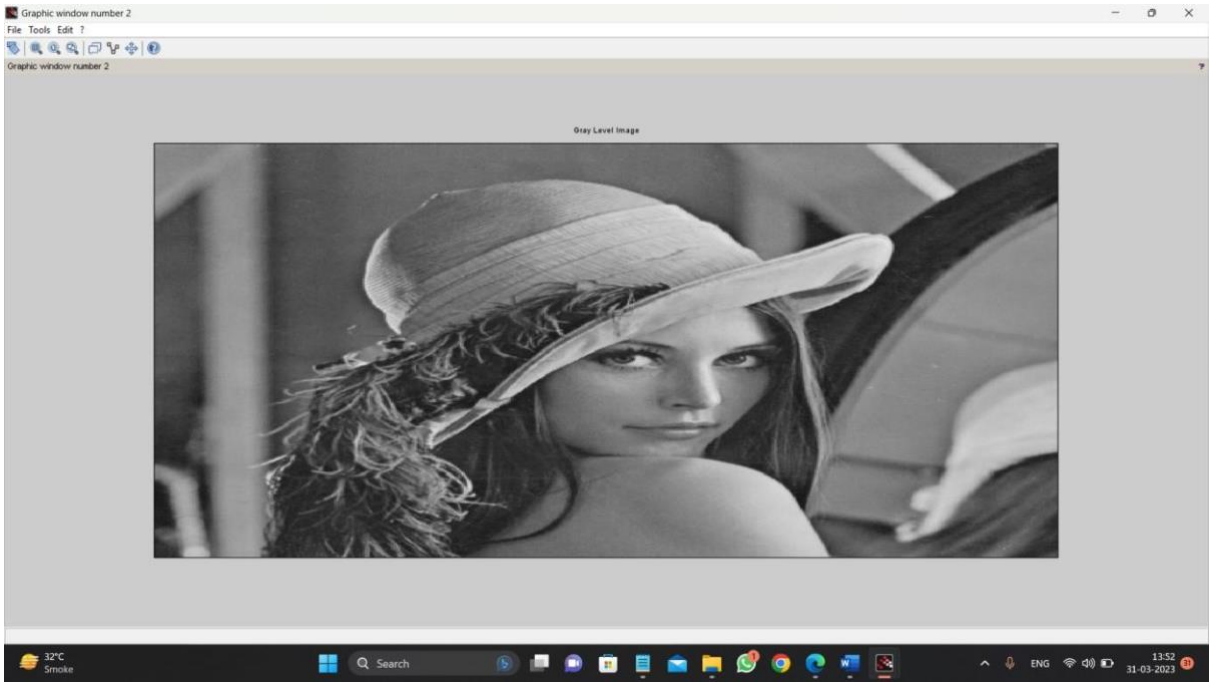
Program to apply false colouring(pseudo) on a gray scale image

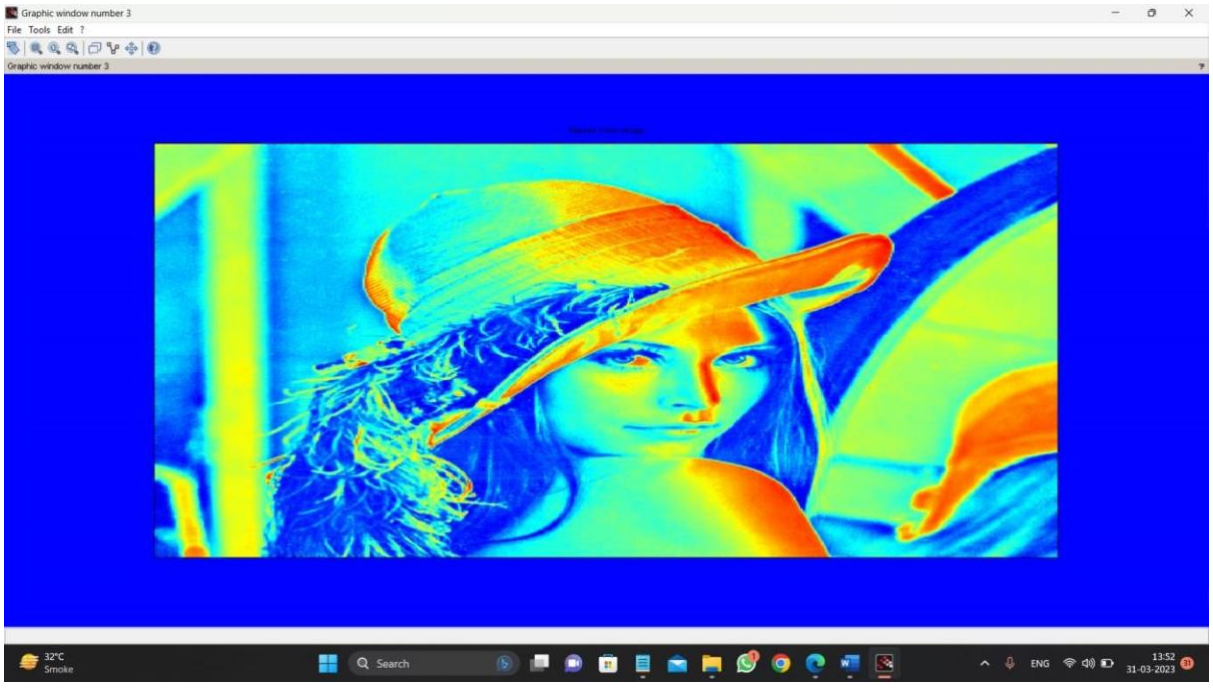
Code

```
clc; close;
a = imread('lena.jpeg');

//Displaying Original RGB image
figure(1); imshow(a);
title("Original Image")

//Displaying Gray level image b
= rgb2gray(a);
figure(2); imshow(b);
title("Gray Level Image")

//Displaying False coloring(Pseudo) image
figure(3) imshow(b,jetcolormap(256));
title("Pseudo Color Image");
```

Output :-




Graphic window number 2
File Tools Edit ?
Graphic window number 2
Gray Level Image
32°C
Smoke
Search
ENG
13:52
31-03-2023

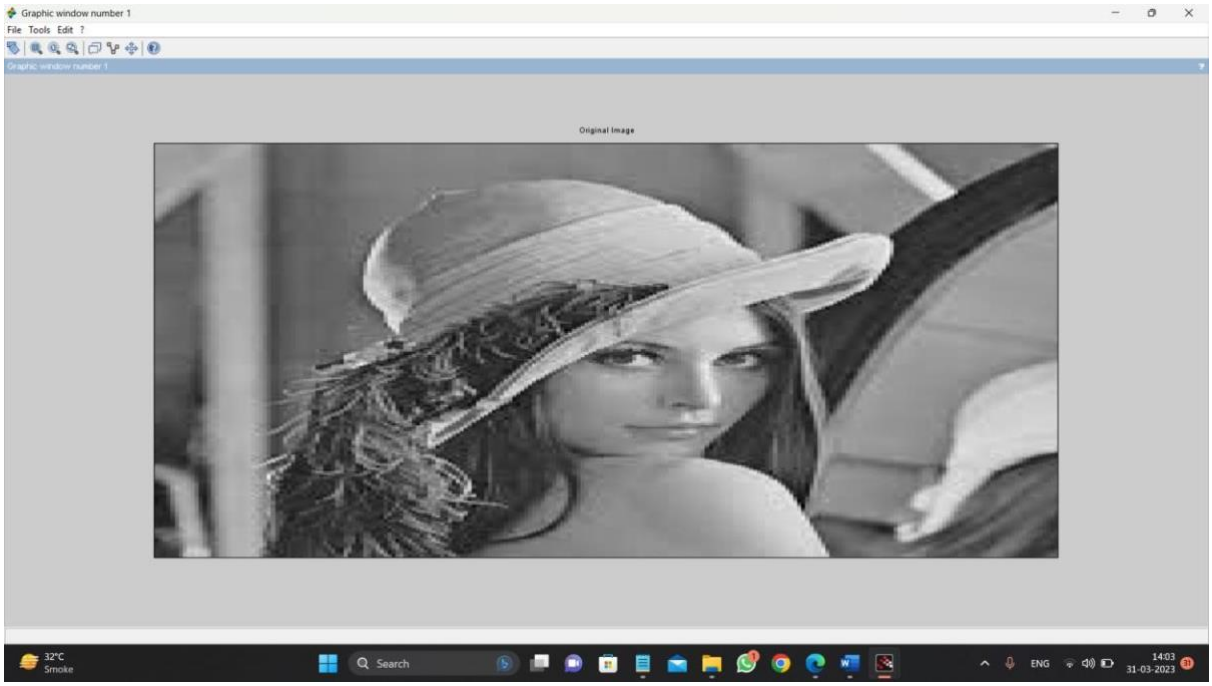

Graphic window number 3
File Tools Edit ?
Graphic window number 3
32°C
Smoke
Search
ENG
13:52
31-03-2023

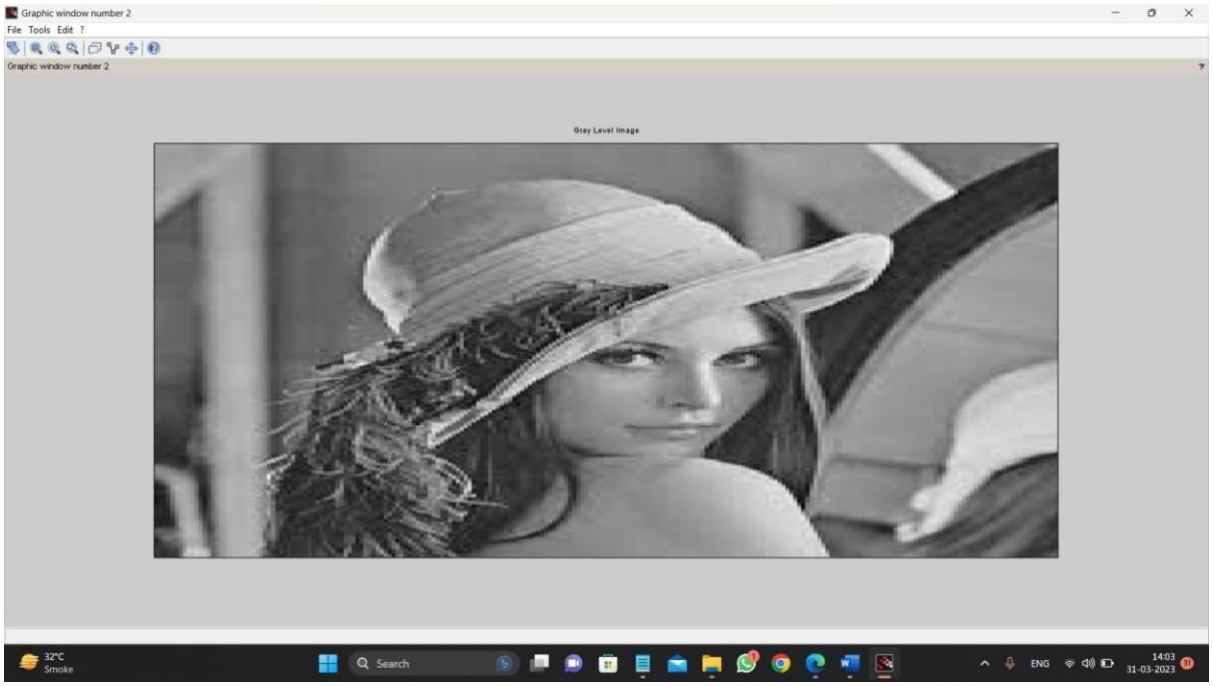
Practical 15

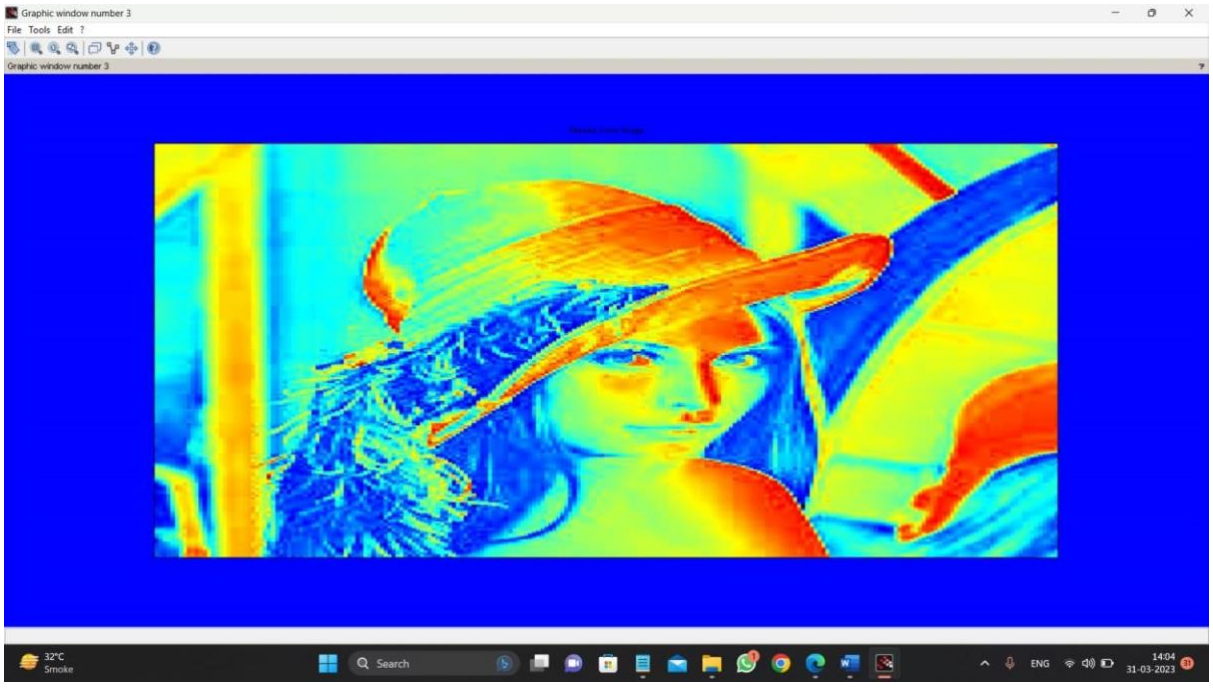
Aim :- Program to apply color to gray image

Code :-
```
clc;
close;

a = imread('lenag.jpeg');

//Displaying Original RGB image
figure(1); imshow(a);
title("Original Image")

//Displaying Gray level image
b = rgb2gray(a); figure(2);
imshow(b);
title("Gray Level Image")

//Displaying False coloring(Pseudo) image
figure(3) imshow(b,jetcolormap(256));
title("Pseudo Color Image");
```

Output :-




Graphic window number 2
File Tools Edit ?
Graphic window number 2
Gray Level Image
32°C
Smoke
Search
ENG
14:03
31-03-2023


Graphic window number 3
File Tools Edit ?
Graphic window number 3
32°C
Smoke
Search
ENG
14:04
31-03-2023