EX.NO:01	# IMAGE ENHANCEMENT
DATE:	

AIM:

To implement an image enhancement program using matlab.

ALGORITHM:

Step 1: Open MATLAB and select new script.

Step 2: Write the source code.

Step 3: Copy the path of the image and paste it in the required line.

Step 4: Assign the original image as 'I' and resize the image as 'j'.

Step 5: Assign the values to resize the image as required.
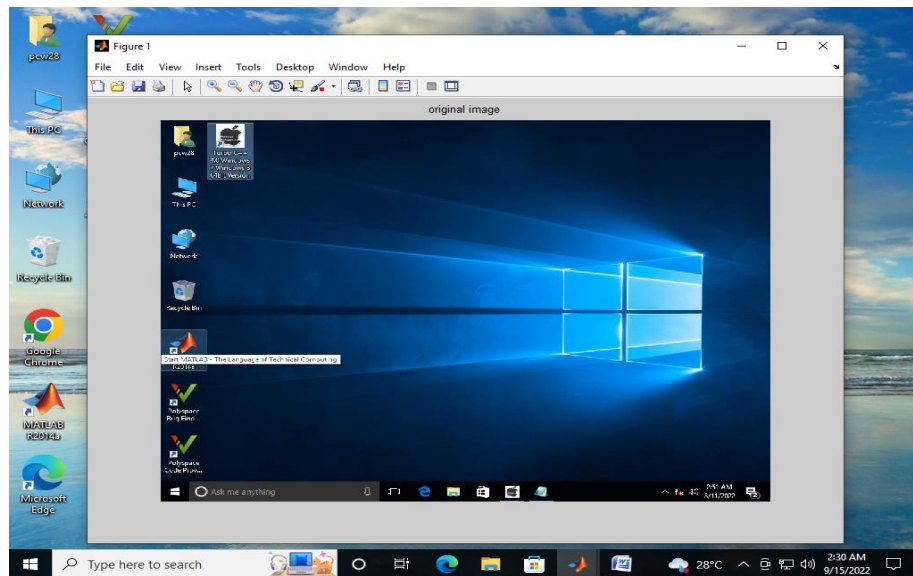
Step 6: Save and run the program.
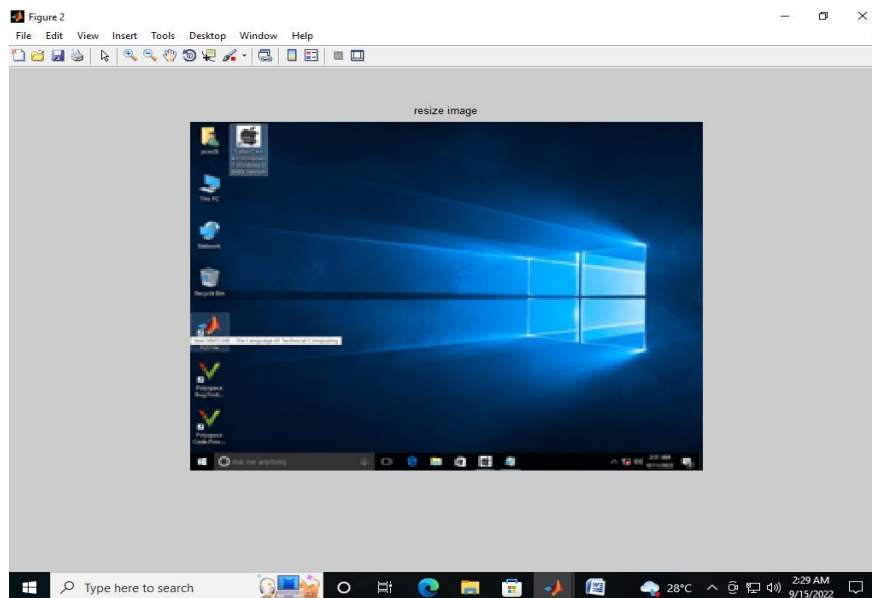
Step 7: Stop the process.

CODING:

```
I=imread('C:\Documents and Settings\Administrator.PCW11\Desktop\Winter.jpg');
j=imresize(I,0.3);
figure
imshow(I);
title('original image')
figure
imshow(j);
title('resized image')
```

OUTPUT:

ORIGINAL IMAGE

RESIZED IMAGE

RESULT:

Thus the program is executed and verified successfully.

EX.NO:02	HISTOGRAM EQUALIZATION
DATE:	

AIM:

To implement histogram equalization program using matlab.

ALGORITHM:

Step 1: Open MATLAB and select new script.

Step 2: Write the source code.

Step 3: Copy the path of the image and paste it in the required line.

Step 4: Assign the original image as A.

Step 5: imhist is used to represent the histogram of RGB colors in the graph.

Step 6: Assign the value to RGB as required.

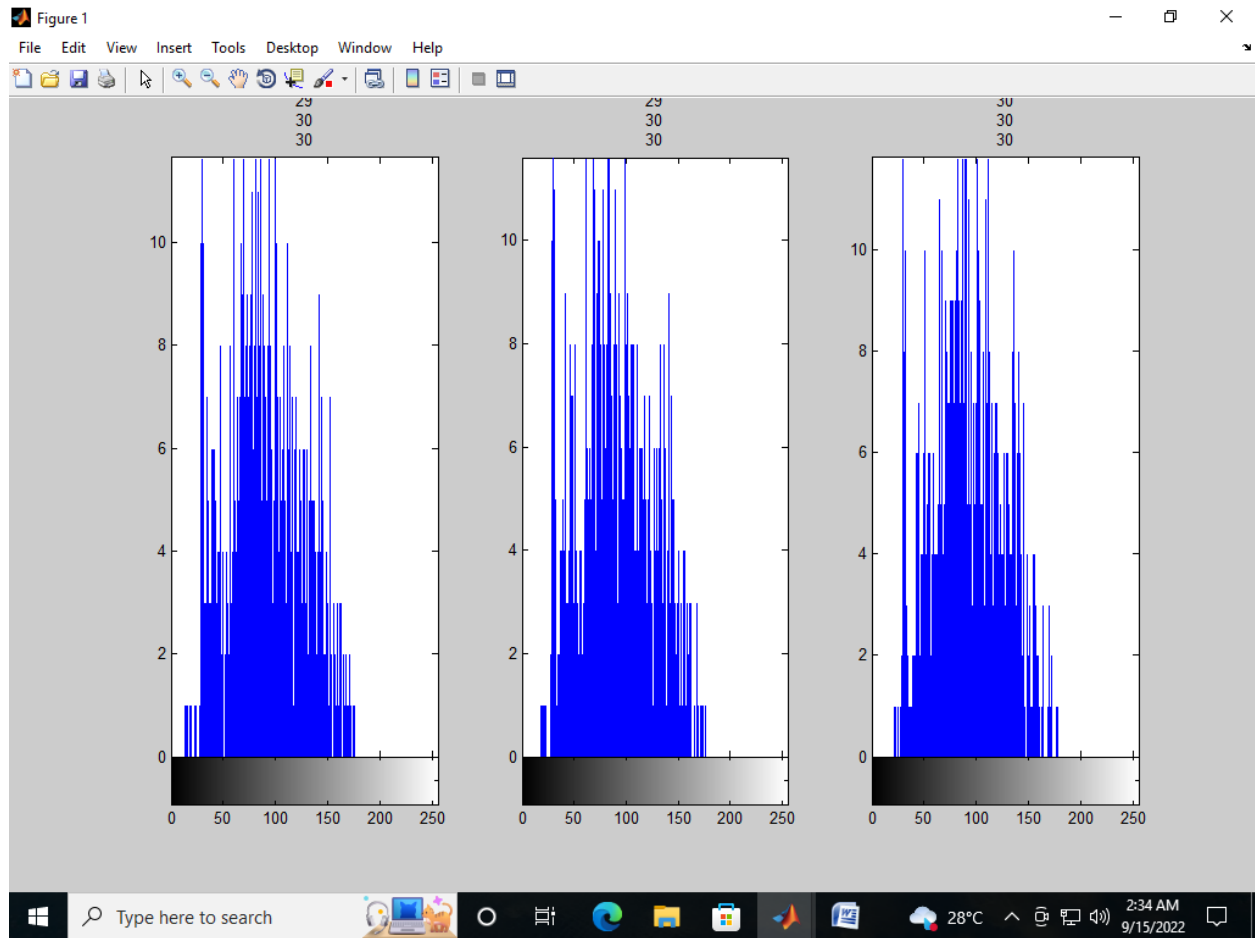Step 7: Subplot is used to plot the values of RGB in the graph.

Step 8: Save and run the program.

Step 9: Stop the process.

CODING:

```
A=imread('C:\Documents and
Settings\Administrator.PCW11\Desktop\Winter.jpg');
A_gray=rgb2gray(A);
imhist(A_gray);
imhist(A_gray,128);
imhist(A_gray,32);
R=A(:,1,1);
G=A(:,1,2);
B=A(:,1,3);
subplot(1,3,1),imhist(R),title(R);
subplot(1,3,2),imhist(G),title(G);
subplot(1,3,3),imhist(B),title(B);
```

OUTPUT:

RESULT:

Thus the program is executed and verified successfully.

EX.NO:03	IMAGE RESTORATION
DATE:	

AIM:

To implement an image restoration program using matlab.

ALGORITHM:

Step 1: Open MATLAB and select new script.

Step 2: Write the source code.

Step 3: Copy the path of the image and paste it in the required line.

Step 4: Assign the length (LEN) as required.

Step 5: Use noise_var=0.001 to avoid noise, if the noise is more increase the value.

Step 6: Save and run the program.
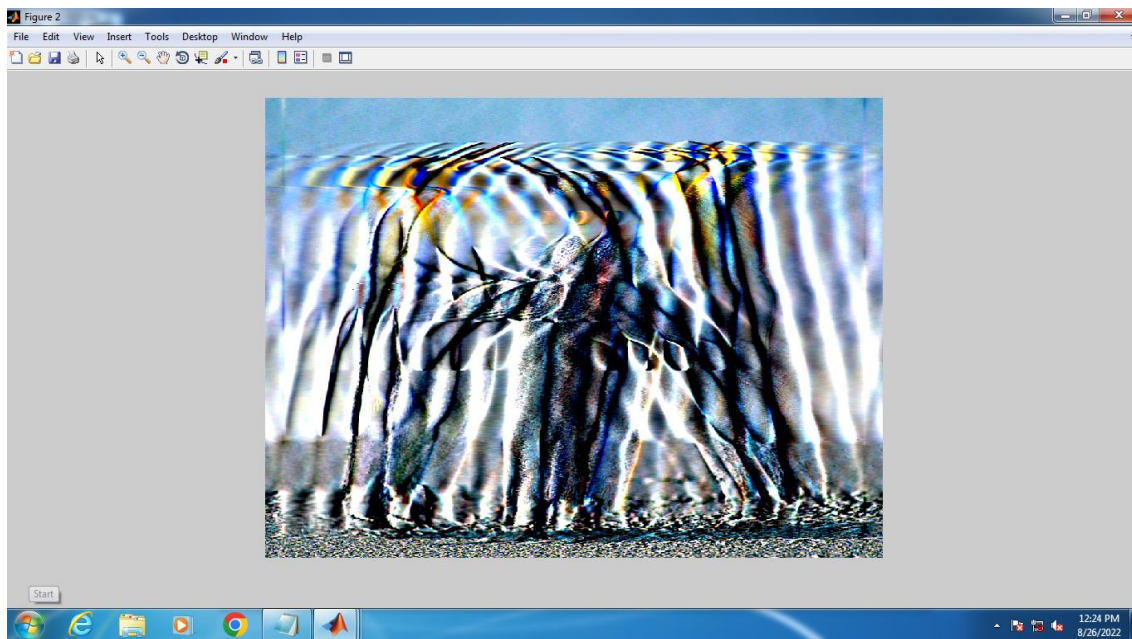
Step 7: Stop the process.

CODING:

```
I=im2double(imread('C:\Documents and Settings\All Users\Documents\My
Pictures\Sample Pictures\images.jpg'));
LEN=60;
THETA=0;
noise_var=0.001;
est_nsr=noise_var/var(I(:));
PSF=fspecial('motion',LEN,THETA);
wnr=deconvwnr(I,PSF,est_nsr);
imshow(wnr);
```

OUTPUT:

ORIGINAL IMAGE

RESTORED IMAGE

RESULT:

Thus the program is executed and verified successfully.

EX.NO:04	IMAGE FILTERING
DATE:	

AIM:

To implement an image filtering program using matlab.

ALGORITHM:

Step 1: Open MATLAB and select new script.

Step 2: Write the source code.

Step 3: Copy the path of the image and paste it in the required line.

Step 4: Assign the name of the output as h.

Step 5: The (5,5) average filter and black and gray filters can be used for filtering the image.

Step 6: To display the image in black and gray the imfilter can be used.

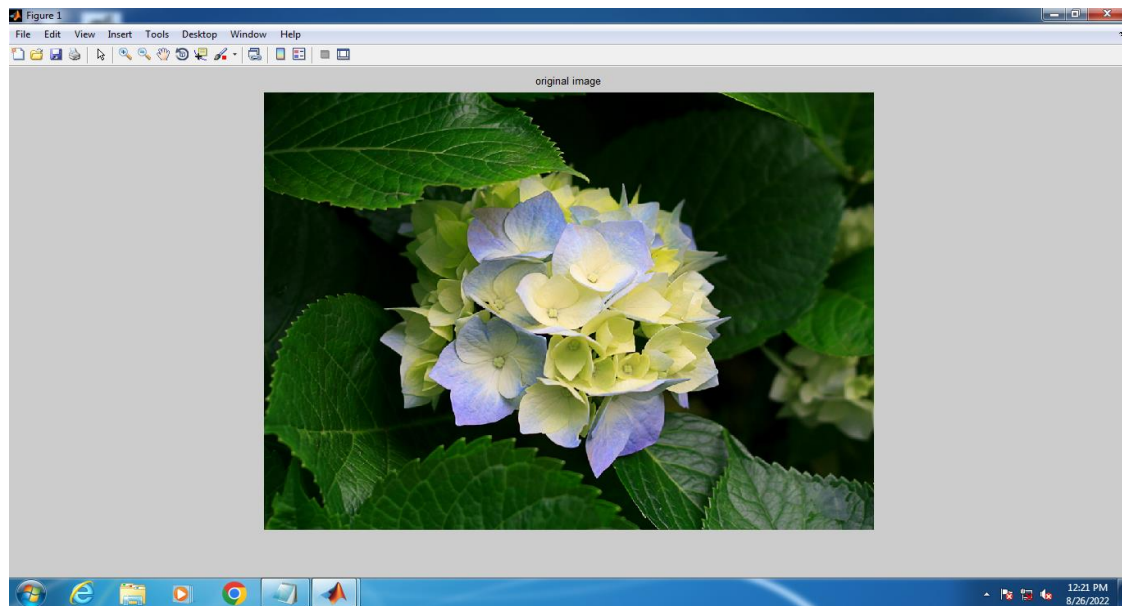Step 7: Save and run the program.
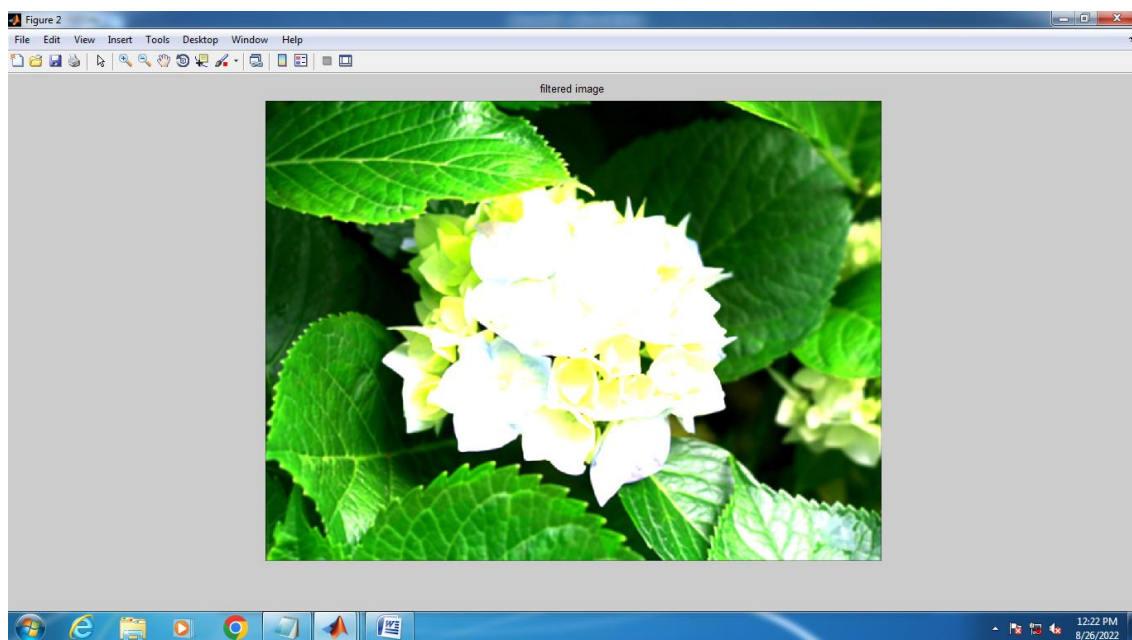
Step 8: Stop the process.

CODING:

```
I=imread('C:\Documents and Settings\Administrator.PCW11\Desktop\Winter.jpg');
h=ones(5,5)/25;
I2=imfilter(I,h);
imshow(I);
title('original image');
figure
imshow(I2);
title('filtered image')
```

OUTPUT:

ORIGINAL IMAGE

FILTERED IMAGE

RESULT:

Thus the program is executed and verified successfully.

EX.NO:05	**EDGE DETECTION USING OPERATORS**
DATE:	

AIM:

To implement edge detection using operators- program using matlab.

ALGORITHM:

Step 1: Start the program.

Step 2: Open MATLAB and select new script.

Step 3: Write the source code.

Step 4: Bw is represented as Binary Image.

Step 5: Bw1, Bw2 and Bw3 is represented for prewitt, sobel, Roberts.

Step 6: Prewitt, sobel and Roberts is used for edge detection.

Step 7: Copy the path of the image in the required line.

Step 8: Save and run the program.

Step 9: The image will be detected and displayed in three different formats.

Step 10: Stop the process.

CODING:

```
i=imread('C:\Documents and Settings\Administrator.PCW11\Desktop\Winter.jpg');
I=rgb2gray(i);
Bw1=edge(I,'prewitt');
Bw2=edge(I,'sobel');
Bw3=edge(I,'roberts');
subplot(2,2,1);
imshow(I);
title('original image');
subplot(2,2,2);
imshow(Bw1);
title('prewitt');
subplot(2,2,3);
imshow(Bw2);
title('sobel');
subplot(2,2,4);
imshow(Bw3);
title('roberts');
```

OUTPUT:
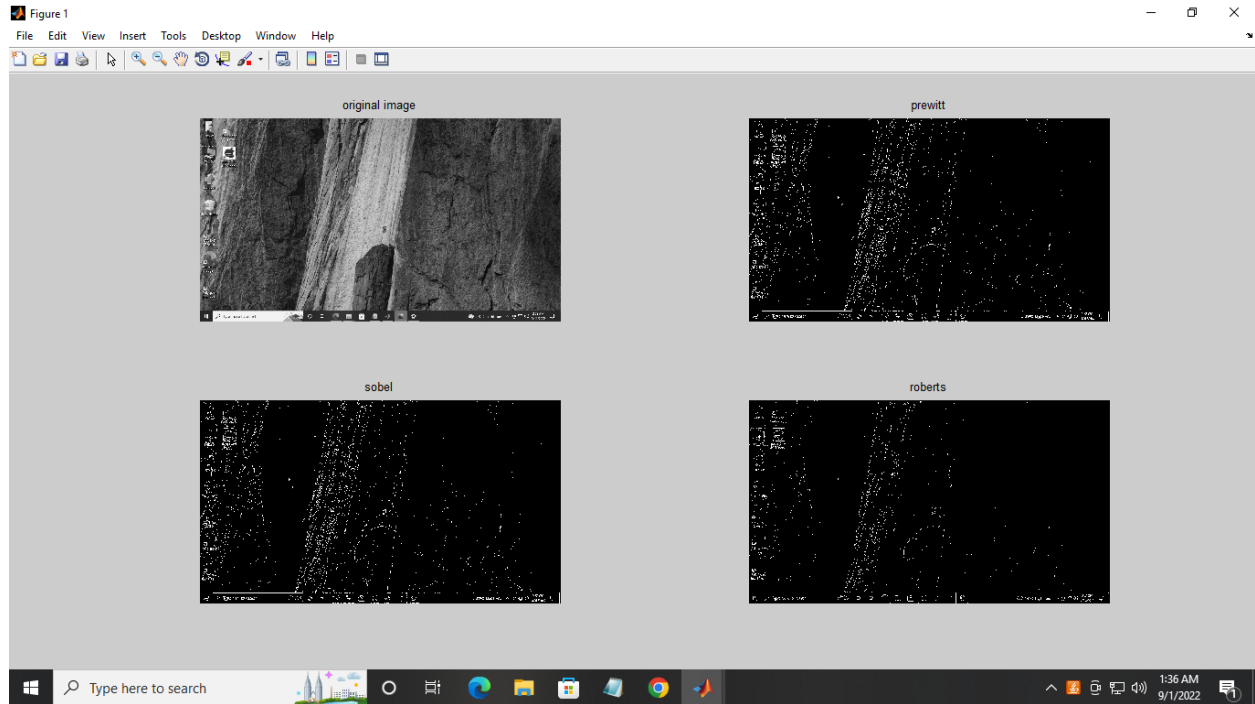
RESULT:

Thus the program is executed and verified successfully.

EX.NO:06	**IMAGE COMPRESSION**
DATE:	

AIM:

To implement an image compression program using matlab.

ALGORITHM:

Step 1: Start the program.

Step 2: Open MATLAB and select new script.

Step 3: Write the source code.

Step 4: wavelet filter is used in image compression.

Step 5: The original image is compressed with the compression ratio.

Step 6: The reconstructed image will be displayed on the window.

Step 7: Copy the path of the image which is selected.

Step 8: Paste the path of the image in the required line.

Step 9: Save and run the program.

Step 10: The output of the image will be compressed and displayed in different window.

Step 11: Stop the process.

CODING:

```
clear all;
close all;
input_image1=imread('C:\pictures\CARU9KLZ.jpg');
input_image=imnoise(input_image1,'speckle',.01);
figure;
imshow(input_image);
n=input('enter the decomposition level=');
[Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('haar');
[c,s] = wavedec2(input_image,n,Lo_D,Hi_D);
disp('the decomposition vector output is');
disp(c);
[thr,nkeep] = wdcbm2(c,s,1.5,3*prod(s(1,:)));
[Compressed_image,TREE,Comp_ratio,PERFL2] =
wpdencmp(thr,'s',n,'haar','threshold',5,1);
disp('Compression ratio in %');
disp(Comp_ratio);
re_ima1 = waverec2(c,s,'haar');
re_ima = uint8(re_ima1);
subplot(1,3,1);
imshow(input_image);
title('I/P image');
subplot(1,3,2);
imshow(Compressed_image);
title('Compressed_image');
subplot(1,3,3);
imshow(re_ima);
title('reconstructed image');
```

OUTPUT:
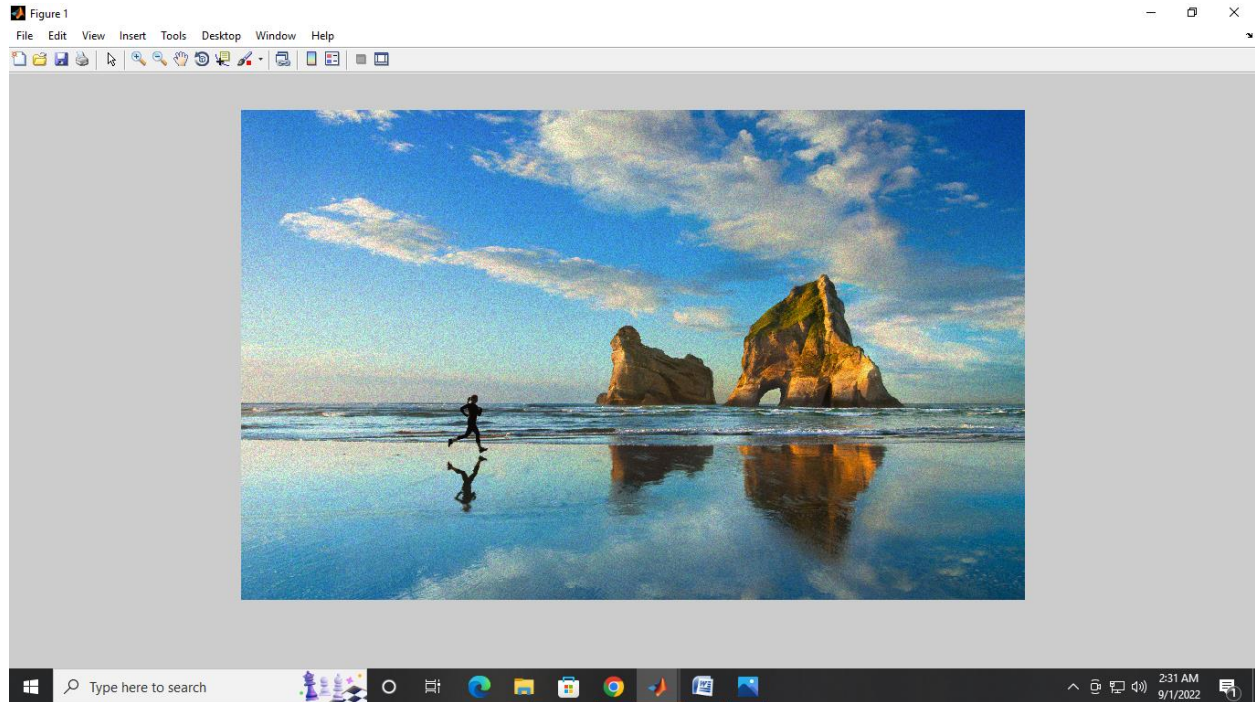
RESULT:

Thus the program is executed and verified successfully.

EX.NO: 07	IMAGE SUBTRACTION
DATE:	

AIM:

To implement an image subtraction program using matlab.

ALGORITHM:

Step 1: Start the program.

Step 2: Open MATLAB and select new script.

Step 3: Write the source code.

Step 4: Copy the path of the image and paste it in the required line.

Step 5: Assign the name of the output as IP.

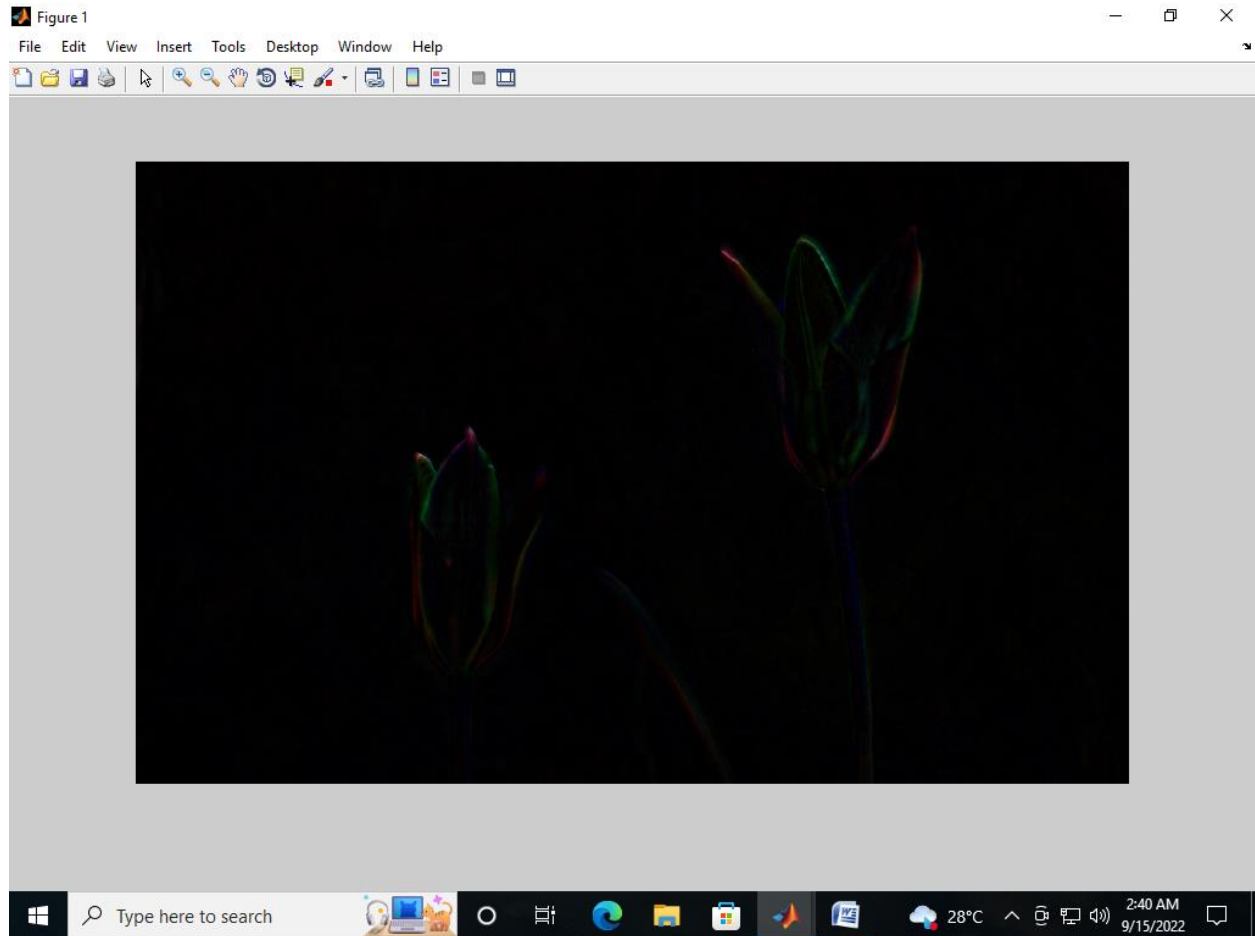Step 6: The imsubtract is used to subtract the background from the image.

Step 7: Save and run the program.

Step 8: Stop the process.

CODING:

```
I=imread('D:\miriam\CAP0SJH1.jpg');
background=imopen(I,strel('disk',15));
IP=imsubtract(I,background);
imshow(IP,[])
```

OUTPUT:

RESULT:

Thus the program is executed and verified successfully.

EX.NO:08	BOUNDARY EXTRACTION
DATE:	

AIM:

To implement an image to extract boundary using matlab.

ALGORITHM:

Step 1: Start the program.

Step 2: Open MATLAB and select new script.

Step 3: Write the source code.

Step 4: Assign the original image as A, F as erode and S as strel.

Step 5:The image Ais converted into binary image.

Step 6: Erode binary image A by structuring element F.

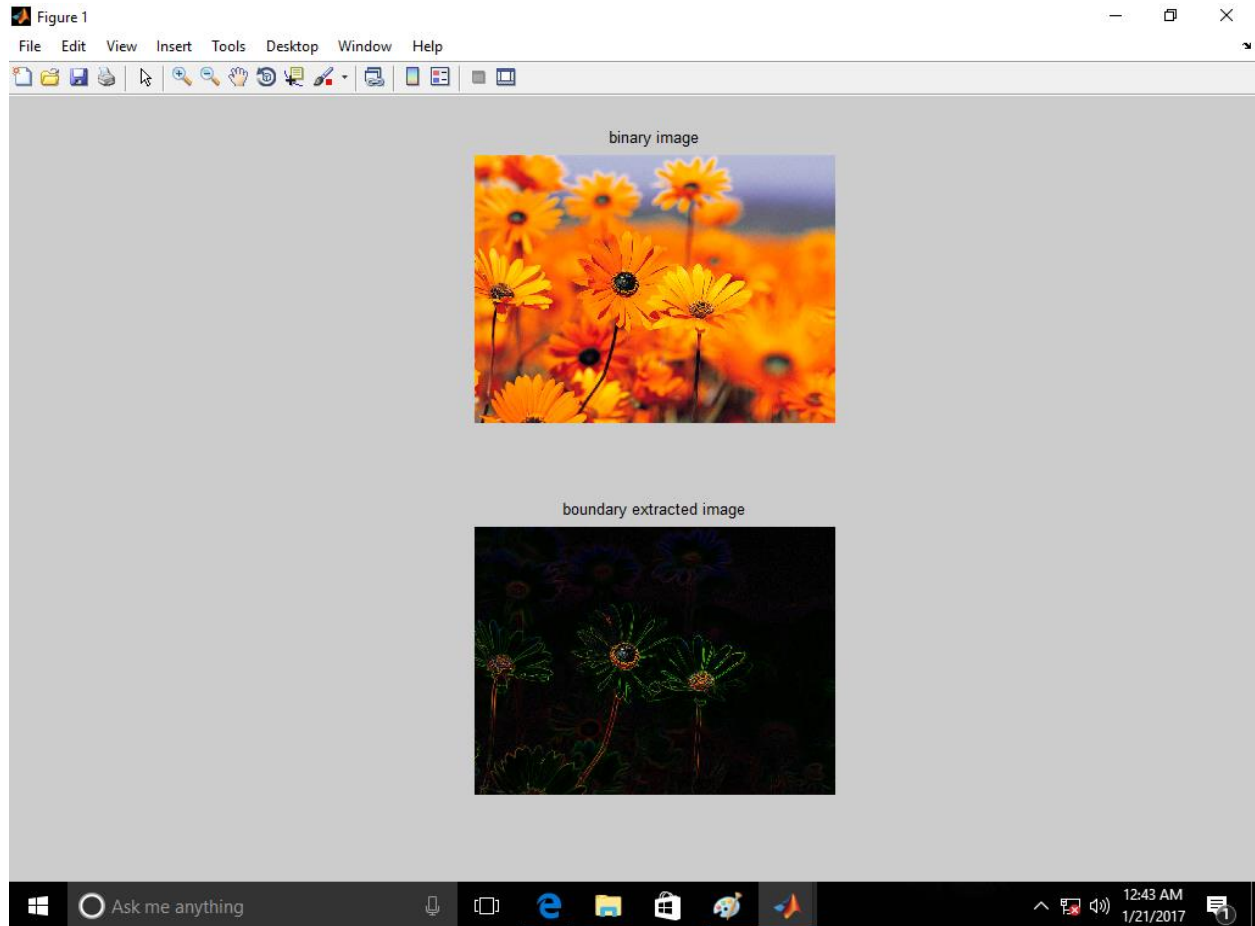Step 7: Subtract the binary image A from the Erode image F.

Step 8: Save and run the program.

Step 9: Stop the process.

CODING:

```
A=imread('D:\miriam\durdle_door_coastline_5k-t1.jpg')
S=strel('disk',2.0);
F=imerode(A,S);
subplot(2,1,1);
imshow(A);title('binary image');
subplot(2,1,2);
imshow(A-F);title('boundary extracted image');
```

OUTPUT:

RESULT:

Thus the program is executed and verified successfully.

EX.NO:09	IMAGE SEGMENTATION
DATE:	

AIM:

To implement an image segmentation program using matlab.

ALGORITHM:

Step 1: Start the program.

Step 2: Open MATLAB and select new script.

Step 3: Write the source code.

Step 4: Copy the path of the image and paste it in the required line.

Step 5: Assign the original image with sizes like height, width and planes.

Step 6: Reshape is used to represent the size of RGB color in the graph.

Step 7: Assign the value to RGB as required.

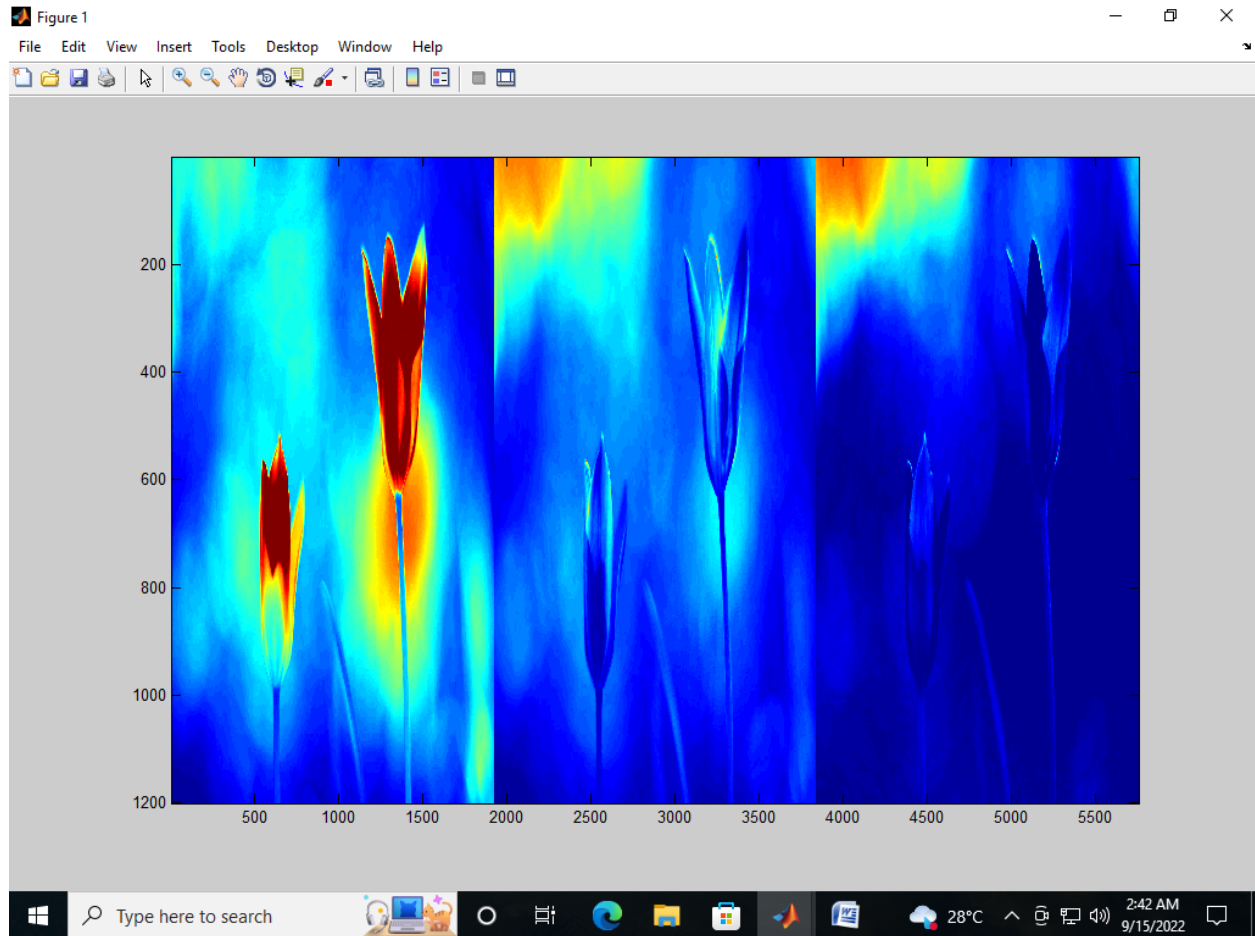Step 8: Colorbaron is used to plot the value in of RGB in the graph.

Step 9: Save and run the program.

Step 10: Stop the process.

CODING:

```
image=imread('C:\Documents and Settings\All Users\Documents\My
Pictures\Sample Pictures\images.jpg');
[height,width,planes]=size(image);
rgb=reshape(image,height,width*planes);
imagesc(rgb);
colorbaron
r=image(:,:,1);
g=image(:,:,2);
b=image(:,:,3);
```

OUTPUT:

RESULT:

Thus the program is executed and verified successfully.