PROGRAM NO:1

DATE:-

IMPLEMENT IMAGE ENHANCEMENT TECHNIQUES

AIM :

To implement image enhancement techniques using MATLAB.

ALGORITHM :

STEP 1:- Start the process.

STEP 2:- Select image **rice.png** and paste in the matlab current directory.

STEP 3:- Read and display image using **imread** and **imshow** in inbuilt function.

STEP 4:- To make the background illumination more uniform, create and approximation of the background as a separate image and then subtract this approximate from the original image.

STEP 5:- Using **strel** function to create a disk-shaped structuring element a radius of 15 .

STEP 6:- Use **imadjust** to increase the contrast of the processed image 12.

STEP 7:- To create a binary version of processed image can use **im2bw** function to convert the gray scale image into a binary image.

STEP 8:- The **graythresh** function automatically compute and approximate threshold convert the grayscale image to binary.

STEP 9:- Remove the background noise with the **bwareaopen** function.

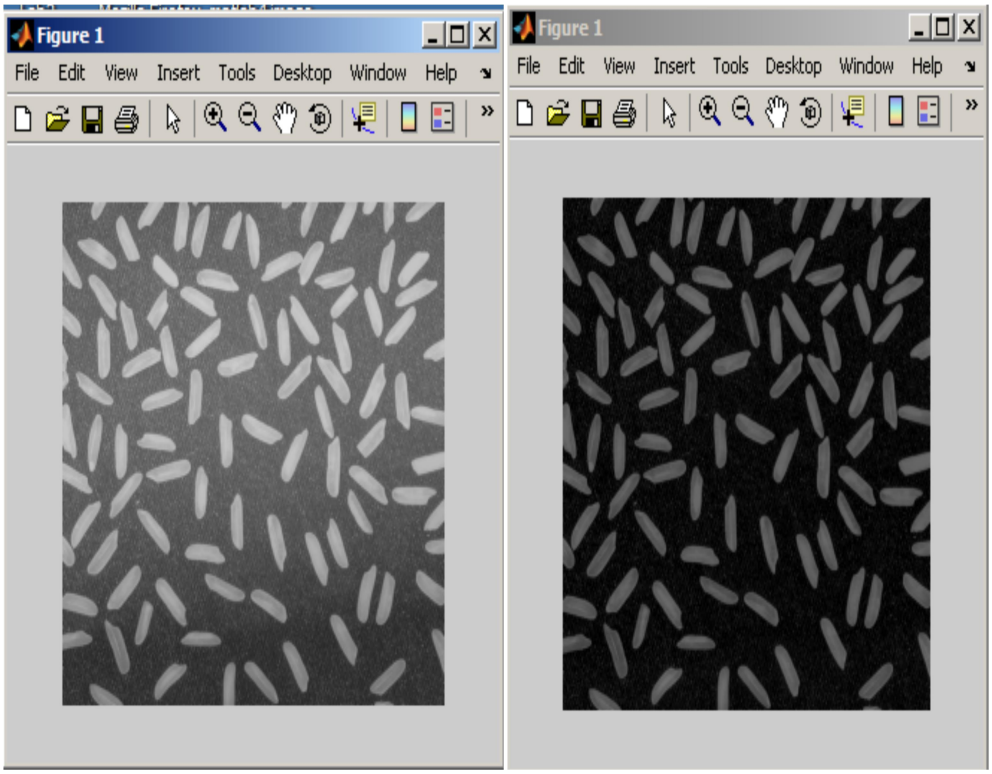
STEP 10:- Corresponding output will be displayed.

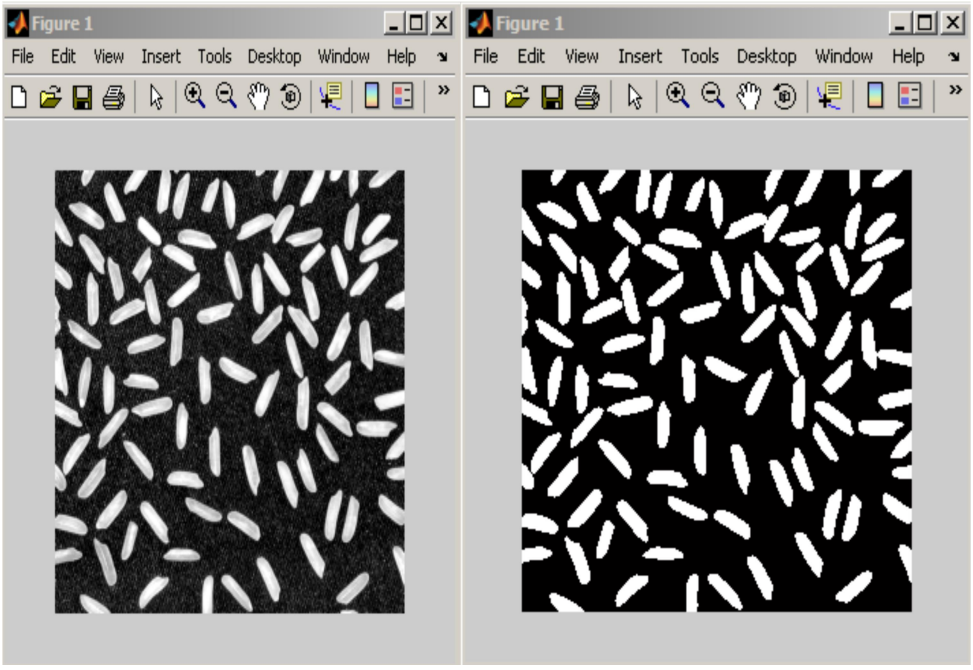
STEP 11:- Stop the process.

1.IMPLEMENT IMAGE ENHANCEMENT TECHIQUES

```
I = imread('rice.png');
imshow(I)
se = strel('disk',15)
background = imopen(I,se);
imshow(background)
 I2 = I - background;
imshow(I2)
 I3 = imadjust(I2);
imshow(I3)
```

OUTPUT





RESULT:

The above program is completed successfully.

PROGRAM NO:02

DATE:-

HISTOGRAM EQUALIZATION.

AIM :

To implement histogram using MATLAB.

ALGORITHM :

STEP 1:- Start the process.

STEP 2:- Read the image tyre.jpg resize the original image using the **imresize** and display the image.

STEP 3:- Use **imagesc** function, which display **im** as image. Each element of **im** corresponds to a rectangular area in the image.

STEP 4:- Use **imhist** function to display a histogram of image data.

STEP 5:- To enhance the contrast of image by transforming the value in an intensity image by using **histeq** inbuilt function. So that the histogram of the output image approximately matches a specified histogram.

STEP 6:- Use **colormap** to set and get the current colormap.

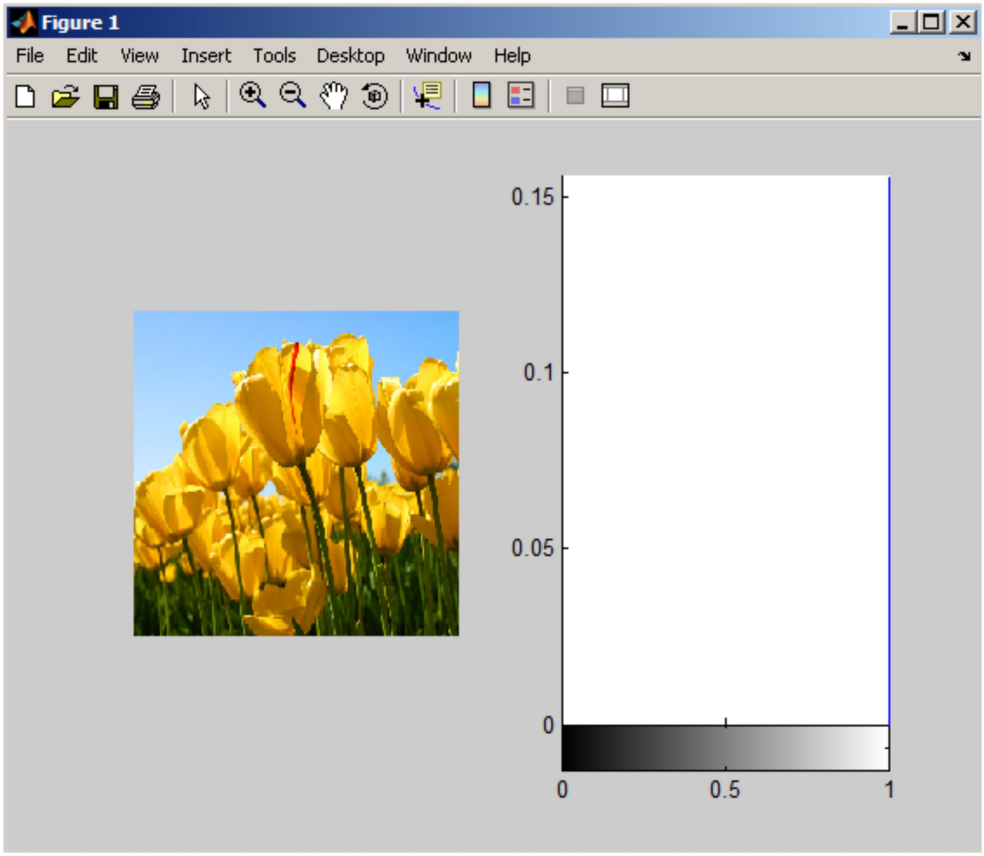
STEP 7:- Corresponding output will be displayed.

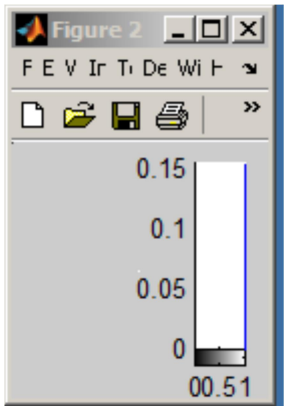
STEP 8:- Stop the process.

2.HISTOGRAM EQUALIZATION

```
I = imread('pout.tif');
figure
subplot(1,2,1)
imshow(I)
subplot(1,2,2)
imhist(I,64)
J = histeq(I);
figure
subplot(1,2,1)
imshow(J)
subplot(1,2,2)
imhist(J,64)
I = imread('pout.tif');
[J,T] = histeq(I);
figure
plot((0:255)/255,T);
```

OUTPUT





RESULT:

The above program is completed successfully.

PROGRAM NO:-03

DATE:-

IMAGE RESTORATION

AIM:

To implement image restoration using MATLAB.

ALGORITHM:

STEP 1:- Start the process.

STEP 2:- Read the image using **im2double**, its makes image as input and returns an image double and display the image.

STEP 3:- Blur the image using **imfilters** it a multidimensional image.

STEP 4:- Add noise to the image by using the inbuilt function **imnoise**.

STEP 5:- Restore the image using the blind deconvolution algorithm .

STEP 6:- Display the image and stop the process.

3.IMAGE RESTORATION

```
l=imread('coins.png');
figure(1);imshow(1);title('source image');
psf=fspecial('disk',15);
blurred=imfilter(1,psf,'circular','conv');
noise_var=0.00001;
noise_mean=0;
blurred=imnoise(blurred,'gaussian',noise_mean,noise_var);
figure(2);imshow(blurred);title('blurred image');
estimated_nsr=noise_var/var(blurred(:));
figure(3);imshow(deconvblind(blurred,psf,100));
title('blind');
h=fspecial('motion',20,45);
motionblur=imfilter(l,h,'replicate');
figure(4);imshow(motionblur);title('motion blur');
```

OUTPUT

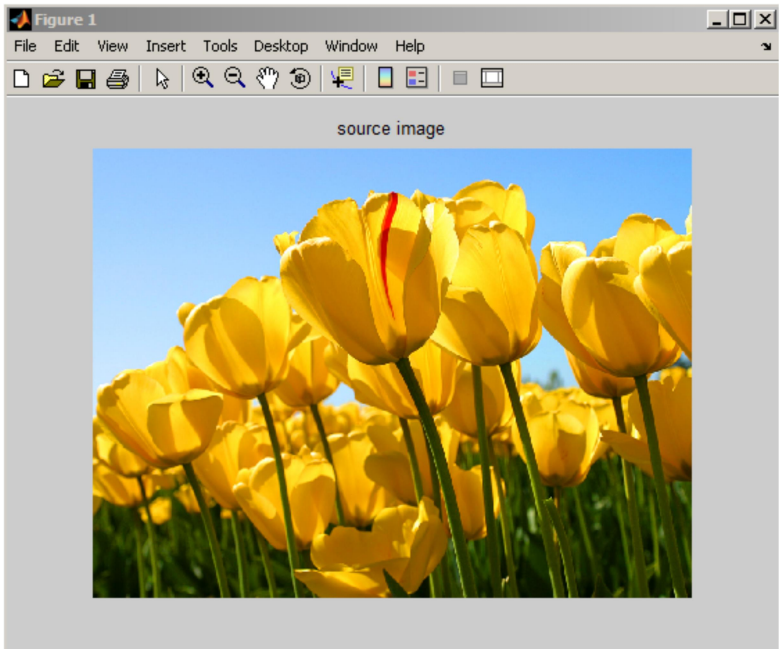


Figure 1
File Edit View Insert Tools Desktop Window Help
source image

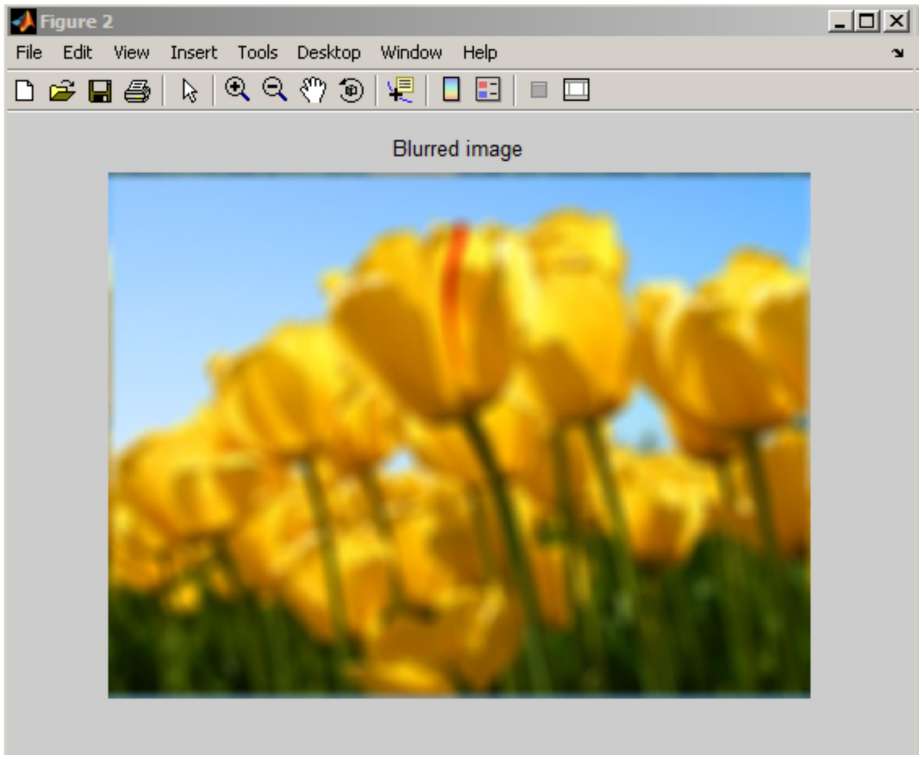
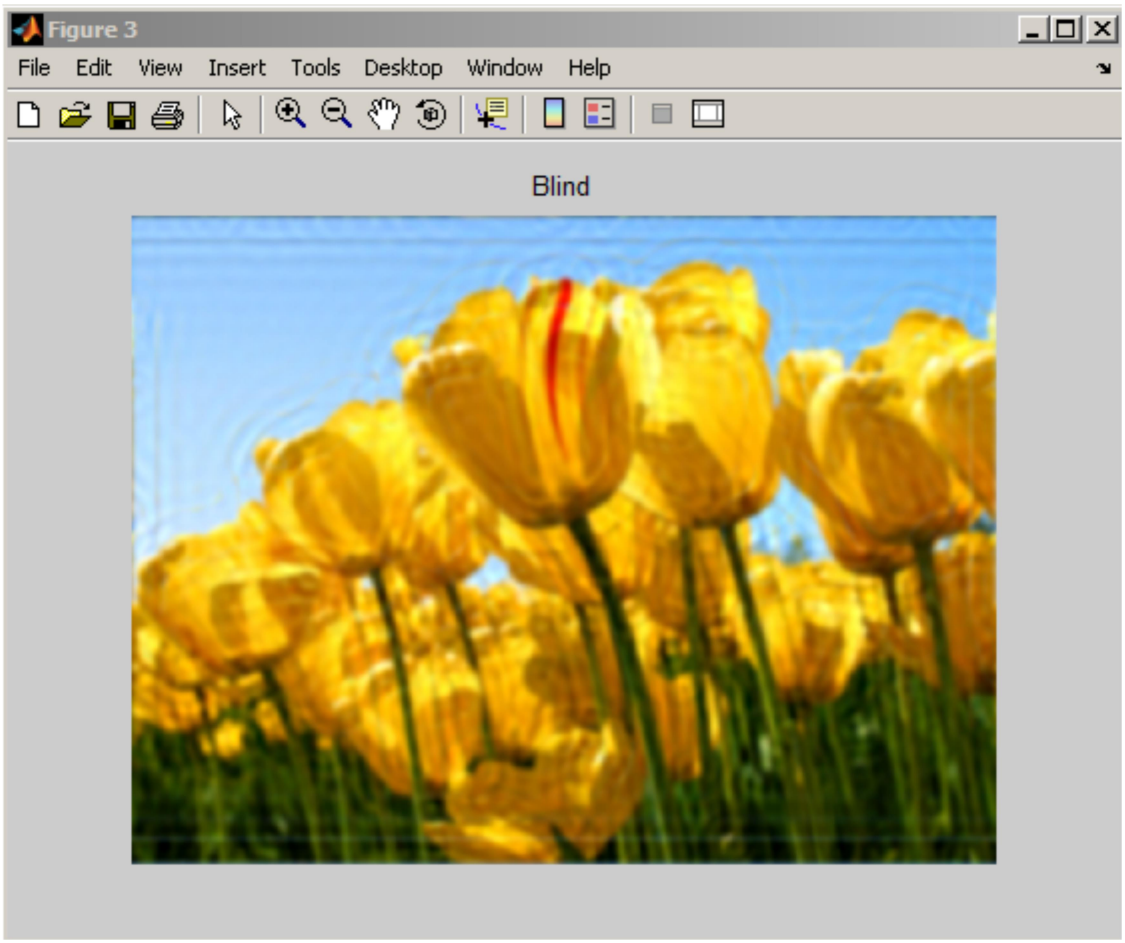


Figure 2
File Edit View Insert Tools Desktop Window Help
Blurred image



RESULT:

The above program is completed successfully.

PROGRAM NO:-04

DATE:-

IMPLEMENT IMAGE FITERING

AIM :

To implement image filtering using MATLAB.

ALGORITHM :

STEP 1:- Start the process.

STEP 2:- Read and display the grayscale image **cameraman.jpg**.

STEP 3:- Add noice to image by using **imnoice** inbuilt function.

STEP 4:- Create a 2-D special filter using **fspecial** function and filter the image using imfilter function.

STEP 5:- Use **wiener2** function to perform two dimensional adaptive noise removal filtering.

STEP 6:- Display all the corresponding outputs.

STEP 7:- Stop the process.

4.IMAGE FILTERING

```
i=imread('cameraman.tif');
imshow(l);
title('original image');
figure;
g=imnoise(l,'gaussian',0.0005,0.0019);
imshow(g);
title('noise added');
figure;
h=fspecial('average',3);
f=imfilter(g,h);
imshow(f);
title('using mean filter');
figure;
w=wiener2(g,[6,6]);
imshow(w);
title('using wiener filter');
```

OUTPUT

RESULT:

The above program is completed successfully.

PROGRAM NO:-O5

DATE:-

EDGE DETECTION USING OPERATIONS

AIM :

To implement edge detection operation like Roberts, Prewitts and Sobel.

ALGORITHM :

STEP 1:- Start the process.

STEP2 :- Read and display the grayscale image.

STEP 3:- Edge take an intensity image and return a binary image BW of the same size as I.

STEP 4:- For the edge detection in an image ,use prewitt,sobel and Robert method.

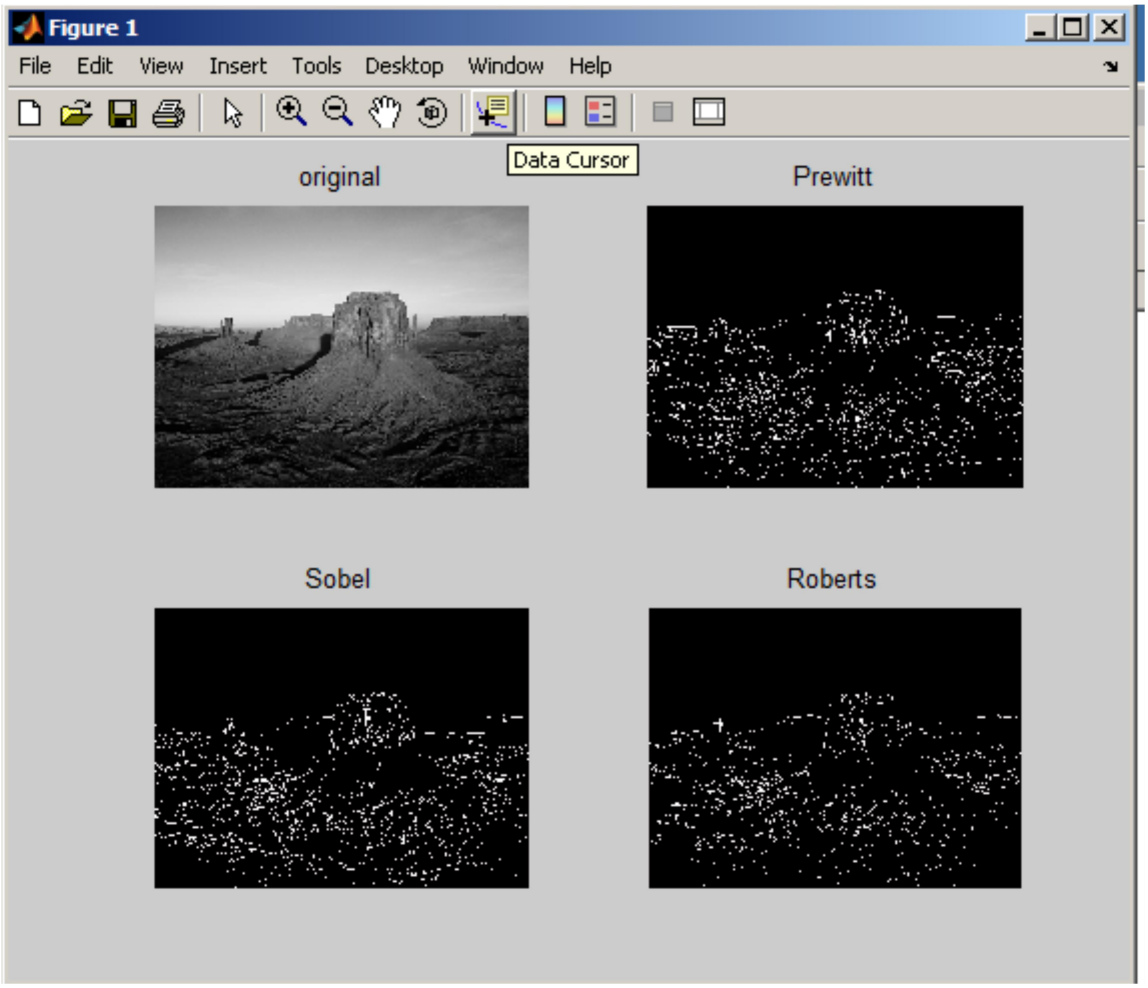
STEP 5:- Display the output.

STEP 6:- Stop the process.

5.EDGE DETECTION USING OPERATION

```
l=imread('coins.png');
bw1=edge(l,'prewitt');
bw2=edge(l,'sobel');
bw3=edge(l,'canny');
subplot(2,2,1);
imshow(l);
title('original');
subplot(2,2,2);
imshow(bw1);
title('prewitt');
subplot(2,2,3);
imshow(bw2);
title('sobel');
subplot(2,2,4);
imshow(bw3);
title('canny');
```

OUTPUT



RESULT:

The above program is completed successfully.

PROGRAM NO:06

DATE:

IMPLEMENT IMAGE COMPRESSION

AIM:-

To implement Image Compression using MATLAB.

ALGORITHM:-

STEP 1:-Start the process.

STEP 2:-Read and display the image.

STEP 3:-To resize an image, use **imresize**inbuilt function.

STEP 4:-Use **dct2(**discrete cosine transform)function to compress the given image.

STEP 5:-Stop the process.

6. IMAGE COMPRESSION

```
im=imread('Desert.jpg');
im1=im(1:206,1:206,:);
im2=imresize(im1,0.5);
im3=imresize(im2,0.5);
subplot('position',[1-1 .5 1/2 1/2]);
imshow(im1);
subplot('position',[1-.5 .5 1/4 1/4]);
imshow(im2);
subplot('position',[1-.25 .5 1/8 1/8]);
imshow(im3);
```

OUTPUT

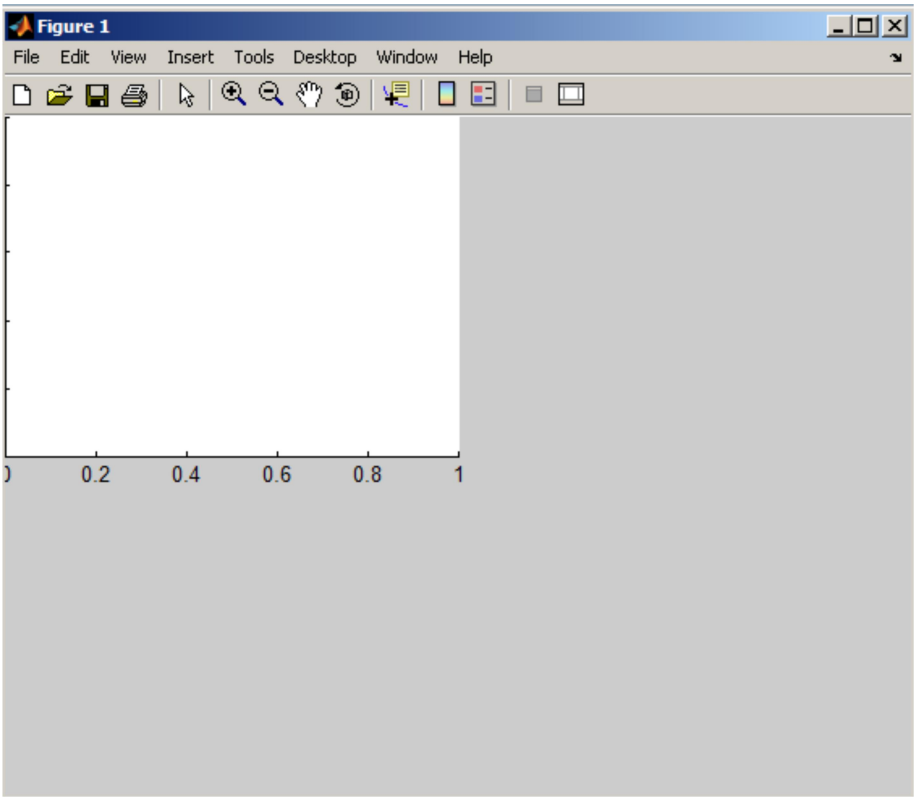


Figure 1
File Edit View Insert Tools Desktop Window Help
0.2
0.4
0.6
0.8
1

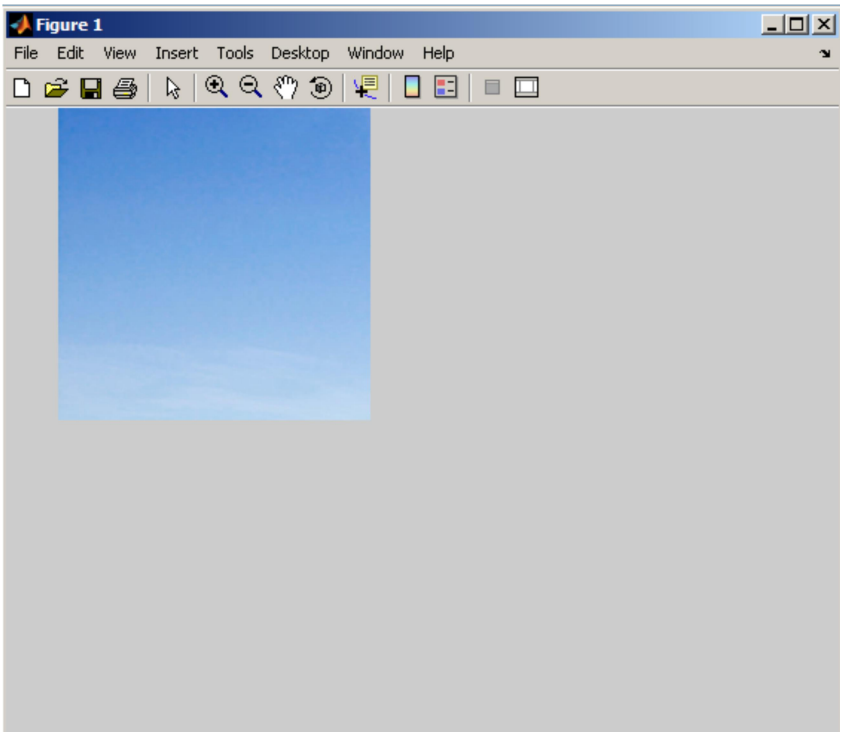
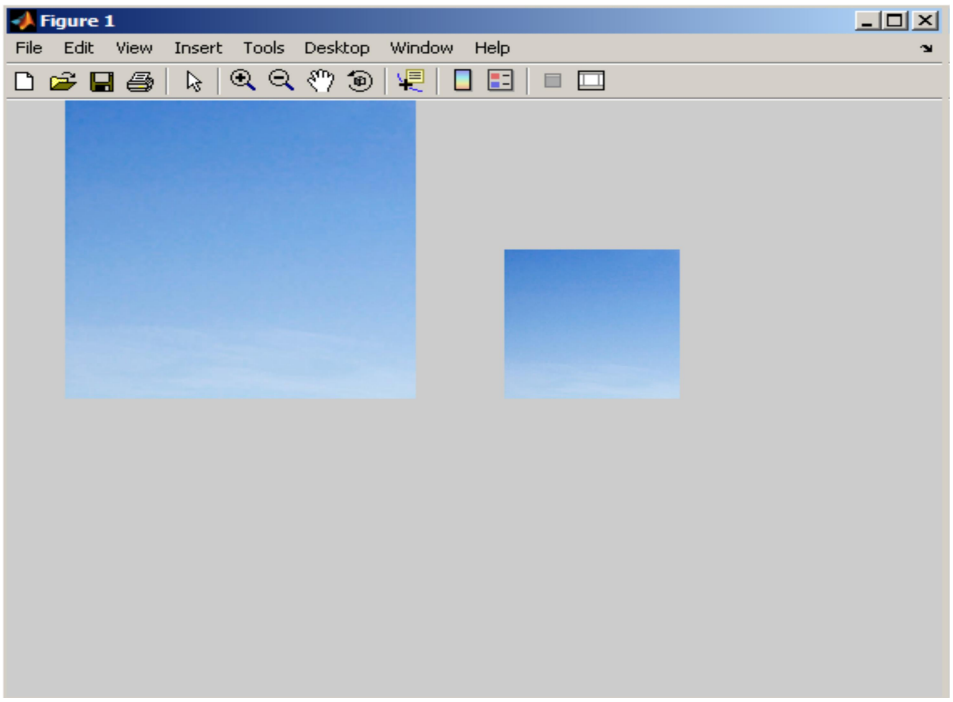
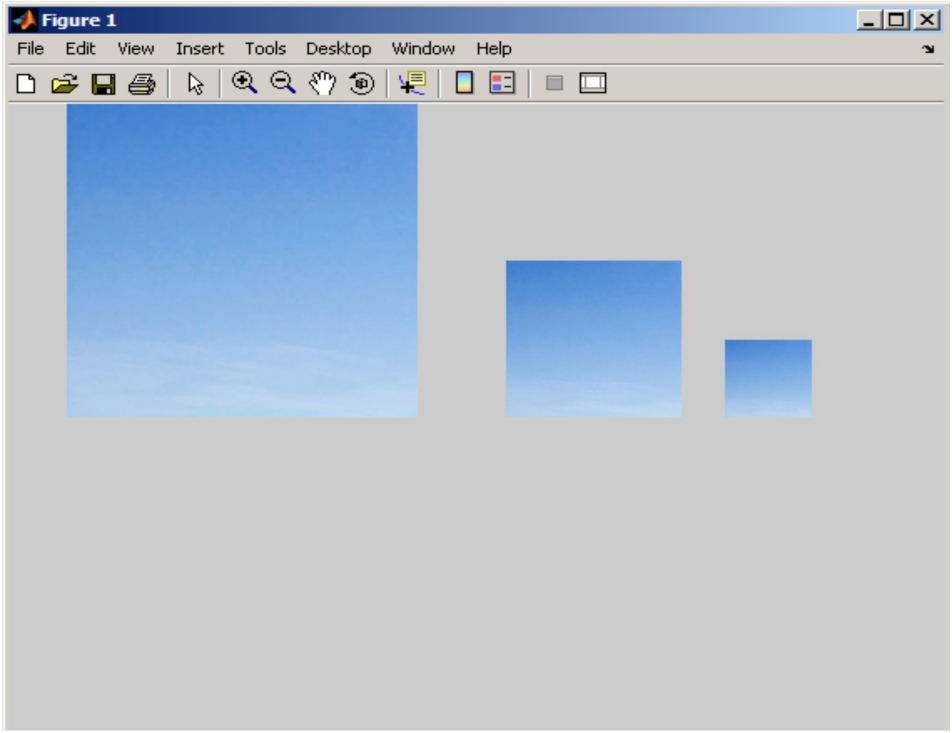


Figure 1
File Edit View Insert Tools Desktop Window Help





RESULT:

The above program is completed successfully.

PROGRAM NO: 07

DATE:

IMAGE SUBTRACTION

AIM:-

To implement Image Subtraction using MATLAB.

ALGORITHM:-

STEP 1:-Start the process.

STEP 2:-Read and display the grayscaleimage**rice**.**jpg.**

STEP 3:-Subtract the background from the image using **imsubtract**inbuilt function.

STEP 4:-Display the image.

STEP 5:-Stop the process.

7.IMAGE SUBTRACTION

```
I=imread('rice.png');
Imshow(I);
Background=imopen(I,strel('disk',15));
Ip=imsubtract(I,background);
Imshow(Ip,[])
```

OUTPUT:

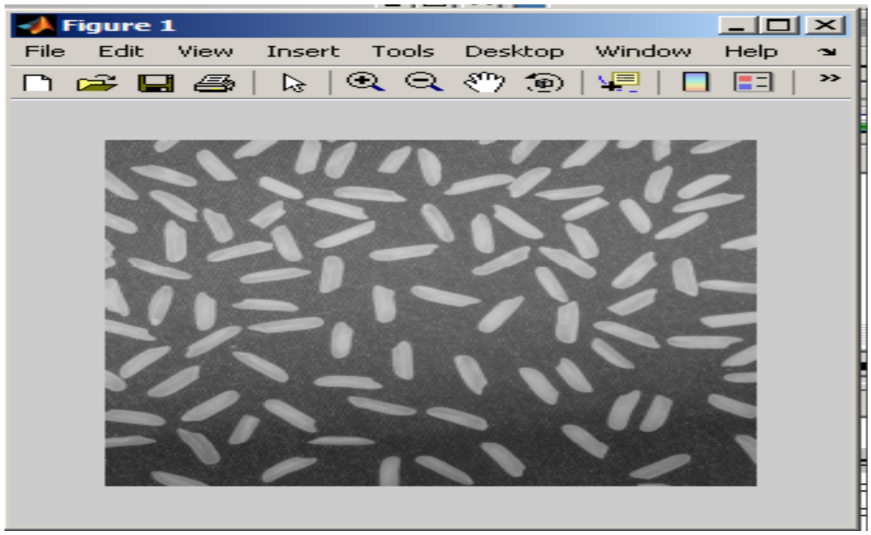


Figure 1
File Edit View Insert Tools Desktop Window Help

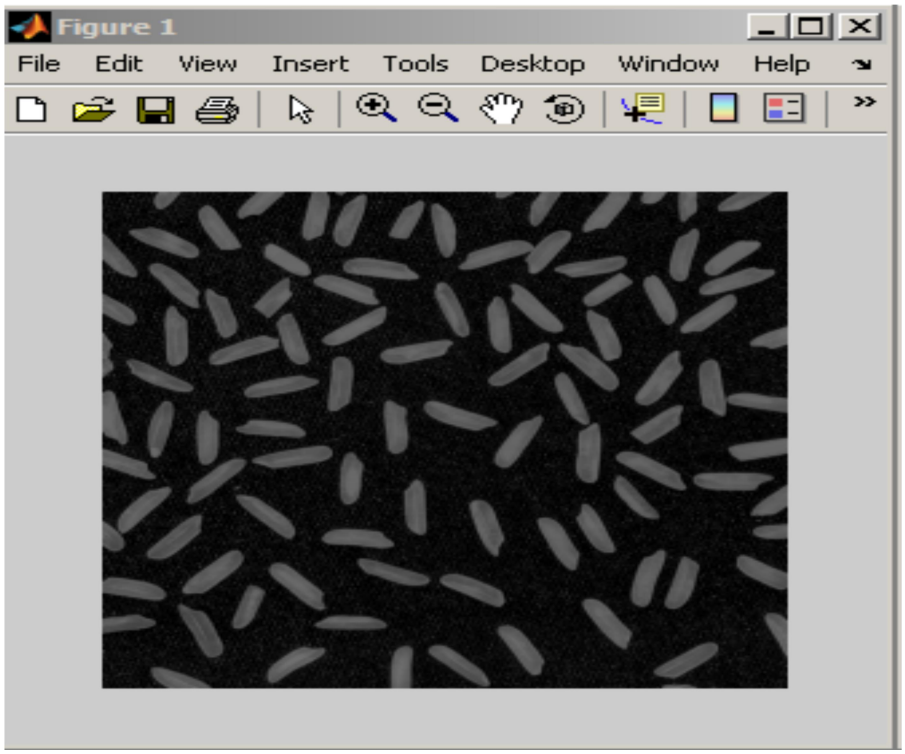
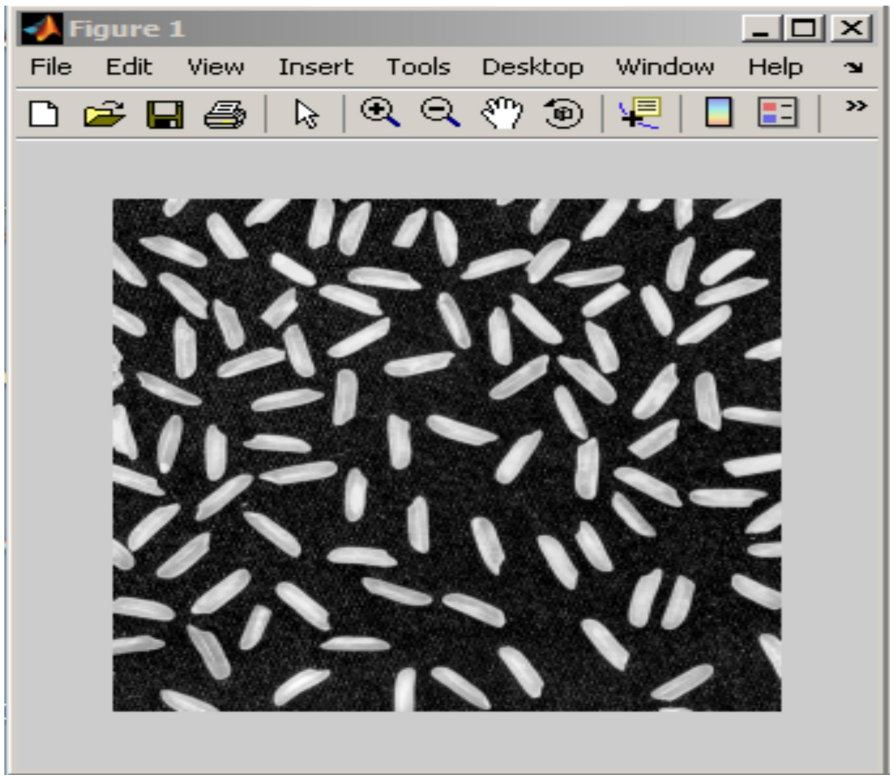


Figure 1
File Edit View Insert Tools Desktop Window Help



RESULT:

The above program is completed successfully.

PROGRAM NO:-08

DATE:-

BOUNDARY EXTRACTION USING MORPHOLOGY

AIM:

To implement a boundary extraction using morphology in **MATLAB.**

ALGORITHAM:

STEP 1:- Start the process.

STEP 2:- Read the image using **imread**function.

STEP 3:- Convert the color image to grayscale image using **grb2gray** function.

STEP 4:- Create a morphological structuring element of the type specified by the shape using**strel** function.

STEP 5:- Convert an image to a binary image using the inbuilt function **im2bw**.

STEP 6:- Erode the image using **imerode** function.

STEP 7:- Display the corresponding output.

STEP 8:- Stop the process.

8. BOUNDARY EXTRACTION

```
A=imread('Tulips.jpg');
C=rgb2gray(A);
C(C<225)=0;
S=strel('disk',40);
D=~im2bw(C);
F=~imerode(D,S);
imshow(A);
title('original image');
imshow(D);
title('Binary image');
imshow(D-F);
title('Boundary image');
```

OUTPUT

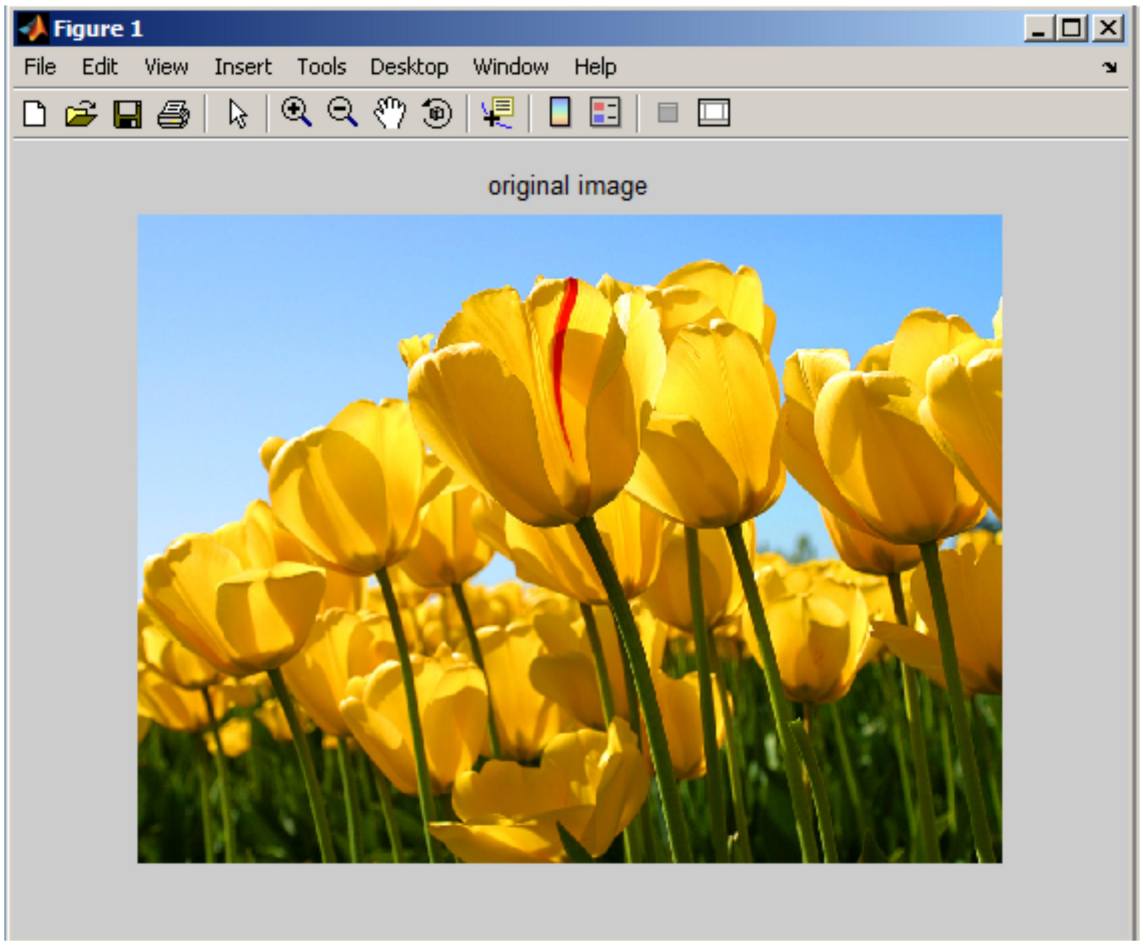
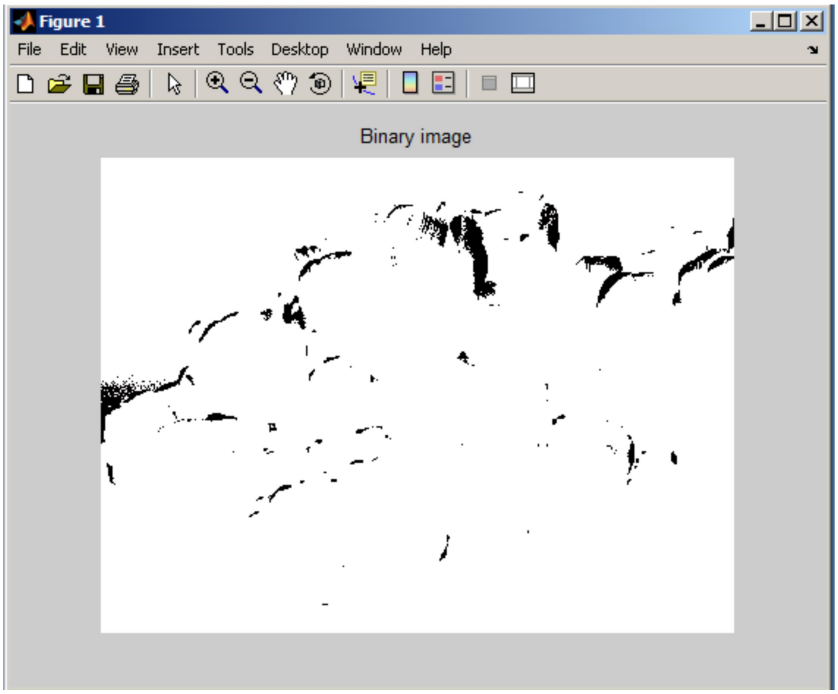
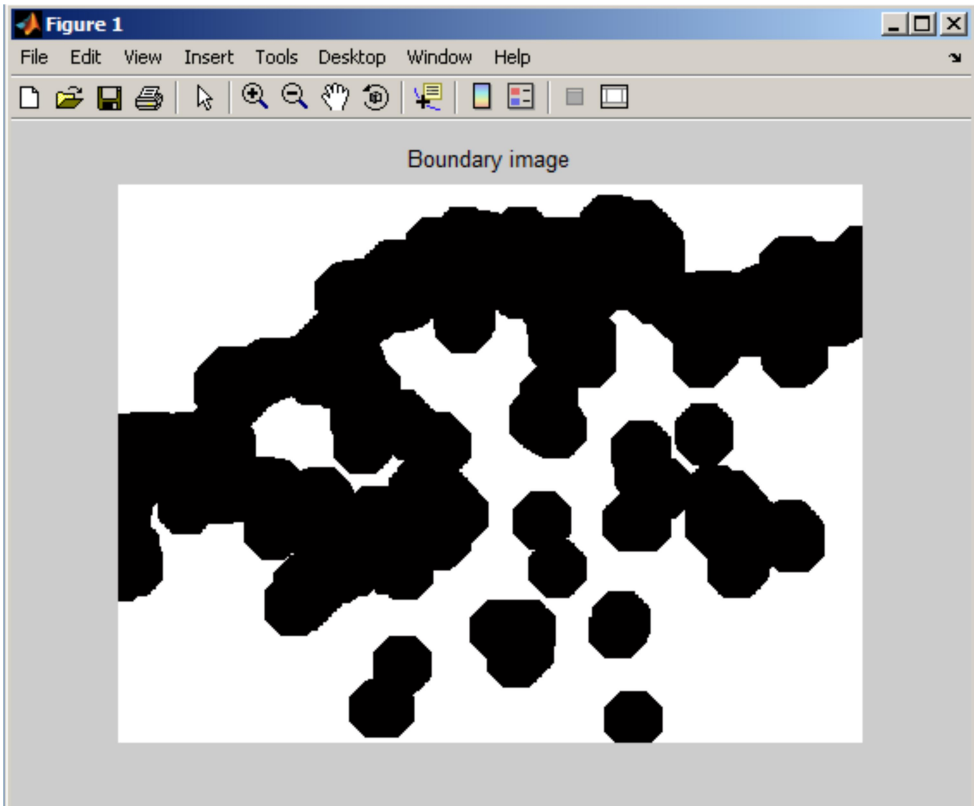


Figure 1
File Edit View Insert Tools Desktop Window Help
original image





RESULT:

The above program is completed successfully.

PROGRAM NO:-09

DATE:-

IMAGE SEGMENTATION

AIM:-

To implement image segmentation using **MATLAB**.

ALGORITHM:-

STEP 1:- Start the process.
STEP 2:-Read and display the image.

STEP 3:- Use the Reshape function, reshape the array of the image and calculate the RGB color channels.

STEP 4:- Calculate the Blueness of the image.

STEP 5:-Apply threshold to segment the foreground.

STEP 6:- Get the label points and mask.

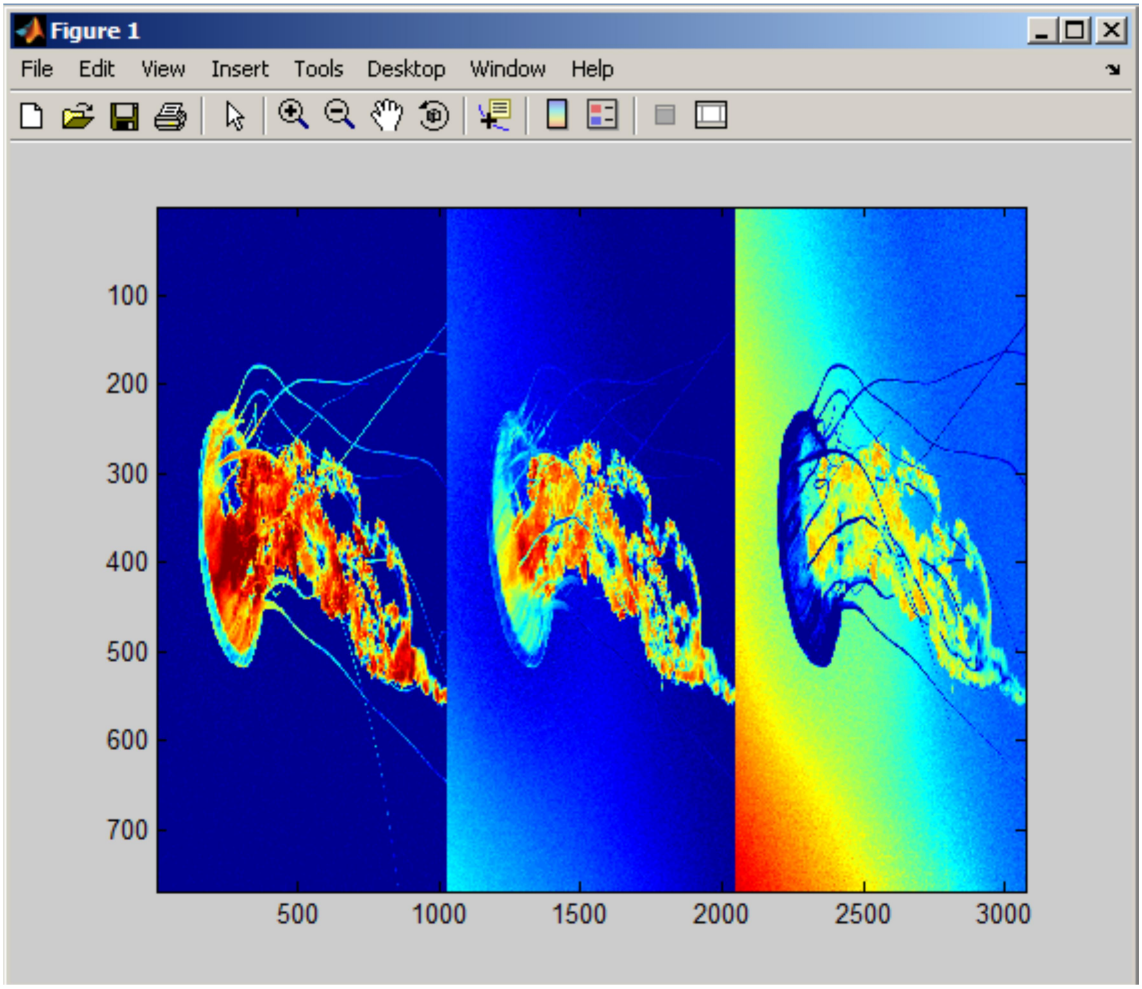
STEP 7:- Display the corresponding outputs.

STEP 8:- Stop the process.

9. IMAGE SEGMENTATION

```
image=imread('Jellyfish.jpg');
[height,width,Planes]=size(image);
bw=reshape(image,height,width*Planes);
imagesc(bw);
```

OUTPUT



RESULT:

The above program is completed successfully.