

Question 1

Determine the perimeter of an object by using 4 connected neighborhoods and 8 connected neighborhoods.

Code:

```
>> i1 = imread('img1.png');  
>> bw1 = im2bw(i1);  
>> p1 = bwperim(bw1,4);  
>> p2 = bwperim(bw1,8);  
>> figure,imshowpair(p1,p2,'montage')
```

Output

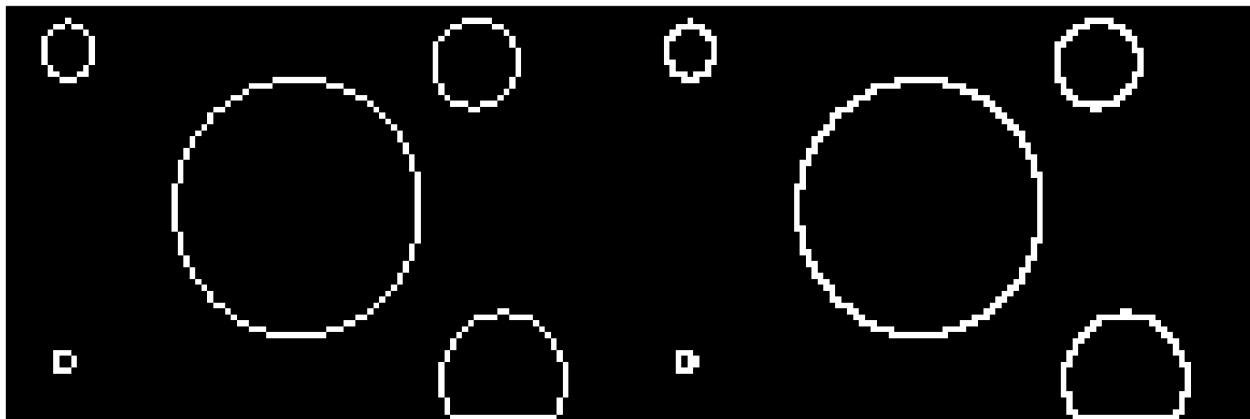


Fig: 4 connected neighborhood

Fig: 8 connected neighborhood

Question 2

Create a binary image using threshold.

Code:

```
>> i2 = imread('img2.png');  
>> bw2 = imbinarize(i2,'adaptive');  
>> figure, imshowpair(i2,bw2,'montage')
```

Output:

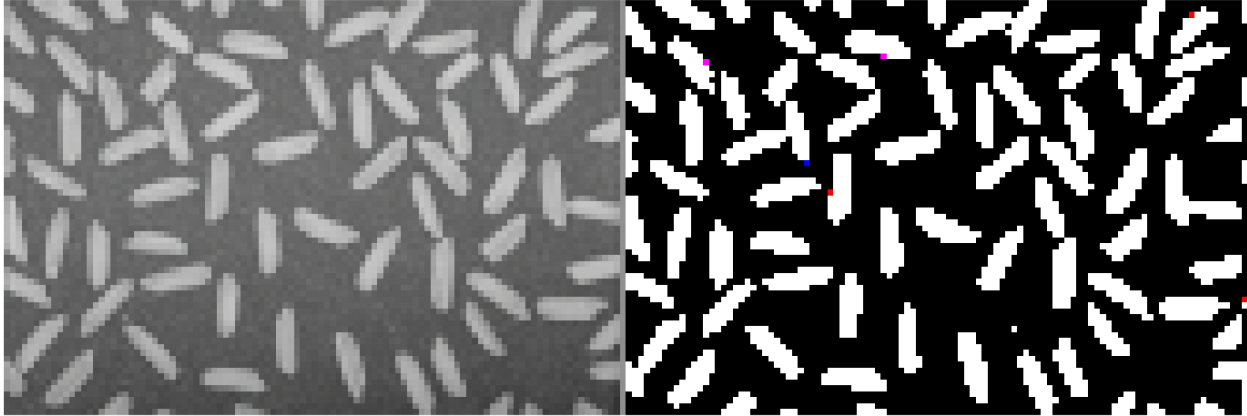


Fig: Grayscale image

Fig: Binary image

Question 3

Determine the number of objects in the binary image generated in Question 2 using the concept of connectivity.

Code:

```
>> se = strel('disk',4);
>> ao = imopen(i2,se);
>> ac = imclose(ao,se);
>> bw3 = im2bw(ac);
>> [L,num]=bwlabel(bw3,4);
```

Output:

```
number_of_objects =

    88
```

Question 4

Find the Euclidean distance between two points of the image.

Code:

```
>> imshow('img2.png')  
>> h=imdistline(gca,[10 100], [10 100]);
```

Output:

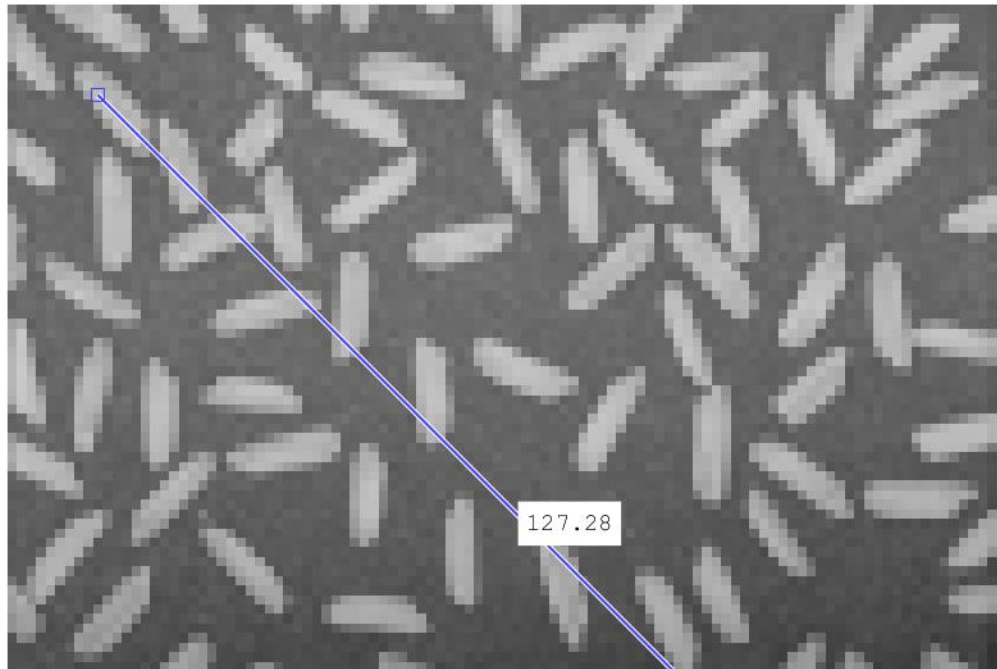


Fig: Distance shown in image

Question 5

Apply the following operations using img1 and img2.

a) Addition

Code:

```
>> resize_image = imresize(i2, [size(i1,1) size(i1,2)]);  
>> add = imadd(i1, resize_image);  
>> subplot(2,2,1);imshow(i1);title('img 1');  
>> subplot(2,2,2);imshow(resize_image);title('img 2');  
>> subplot(2,2,3);imshow(add);title('Additional Image');
```

Output:

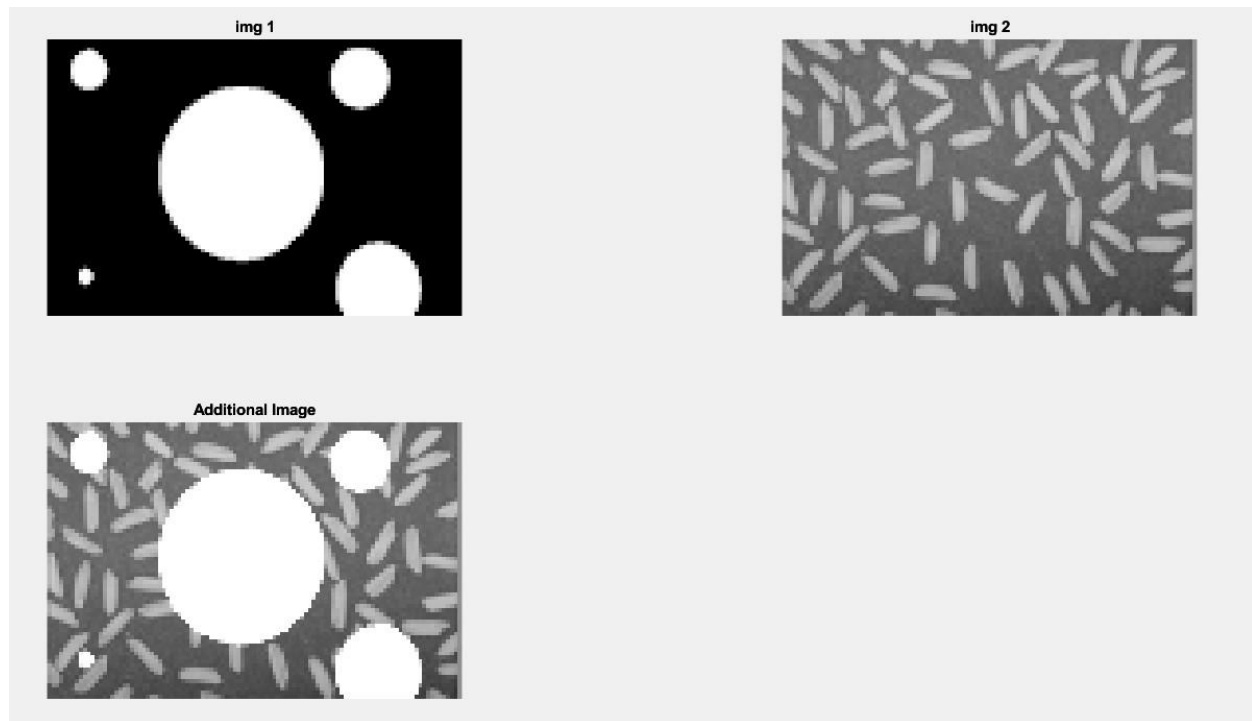


Fig: Both image and added image

b) Subtraction

Code:

```
>> sub = imsubtract(il, resize_image);  
>> figure, imshow(sub)  
>> subplot(2,2,1); imshow(il); title('img 1');  
>> subplot(2,2,2); imshow(resize_image); title('img 2');  
>> subplot(2,2,3); imshow(sub); title('Subtracted Image');
```

Output:

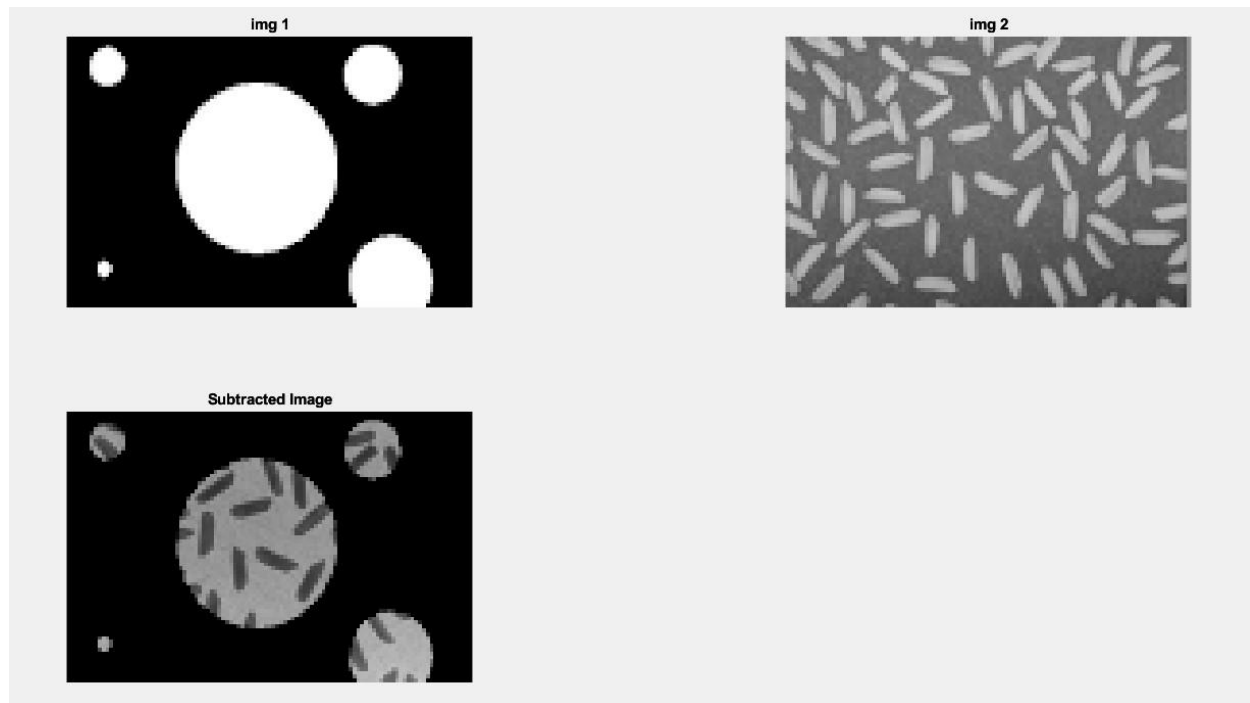


Fig: Both image and subtracted image

c) Multiplication

Code:

```
>> mul = immultiply(il,resize_image);  
>> figure,imshow(mul)  
>> subplot(2,2,1);imshow(il);title('img 1');  
>> subplot(2,2,2);imshow(resize_image);title('img 2');  
>> subplot(2,2,3);imshow(mul);title('Multiplicated Image');
```

Output:

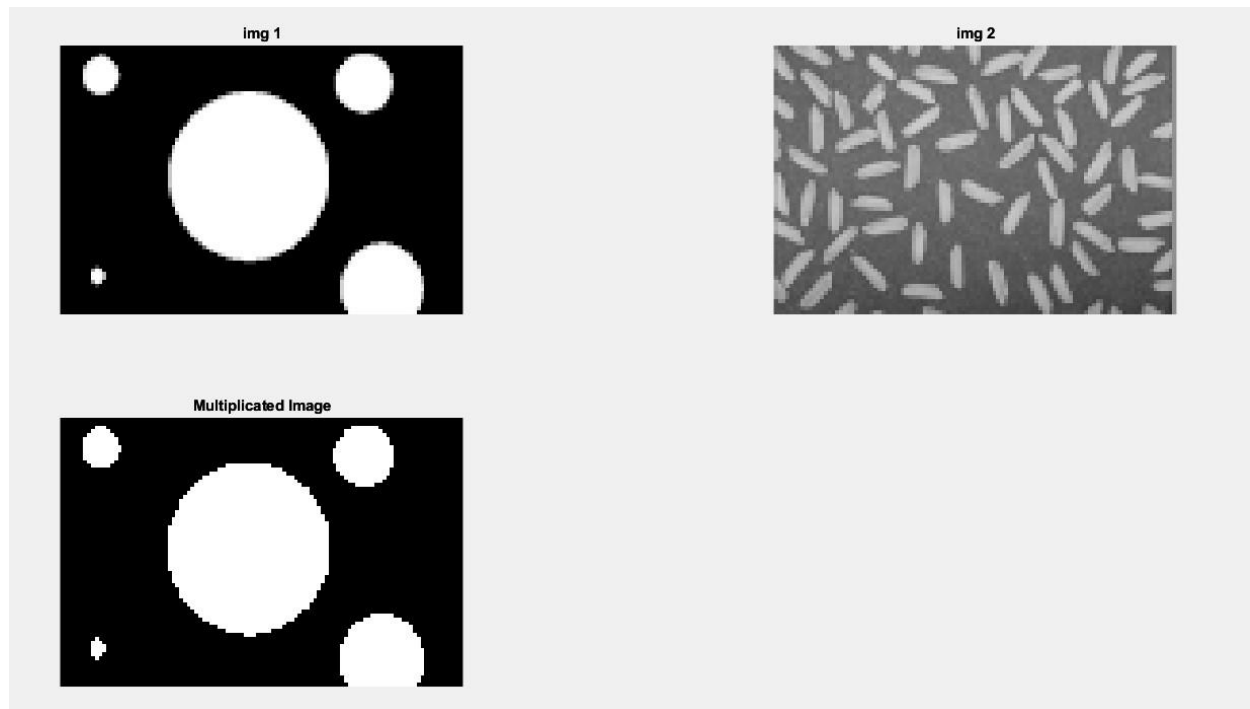


Fig: Both image and multiplied image

d) Division

Code:

```
>> div = imdivide(i2, resize_image);  
>> figure, imshow(div)  
>> subplot(2,2,1);imshow(i1);title('img 1');  
>> subplot(2,2,2);imshow(resize_image);title('img 2');  
>> subplot(2,2,3);imshow(div);title('Divided Image');
```

Output:

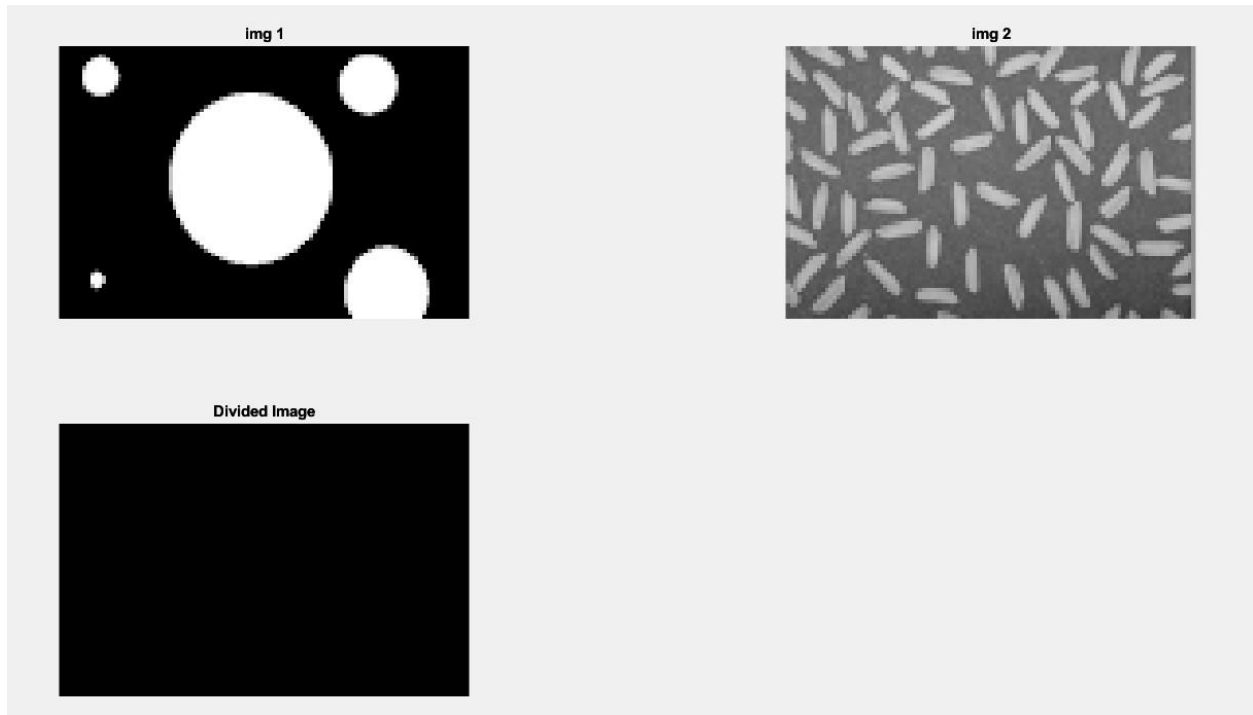


Fig: Both image and divided image

Question 6

Apply the following operations using img1 and img2.

a) AND

Code:

```
>> and = bitand(il,resize_image);  
>> subplot(2,2,1);imshow(il);title('img 1');  
>> subplot(2,2,2);imshow(resize_image);title('img 2');  
>> subplot(2,2,3);imshow(and);title('And Image');
```

Output:



Fig: Both image and AND operation image

b) OR

Code:

```
>> or = bitor(il, resize_image);  
>> subplot(2,2,1);imshow(il);title('img 1');  
>> subplot(2,2,2);imshow(resize_image);title('img 2');  
>> subplot(2,2,3);imshow(or);title('OR Image');  
...
```

Output:

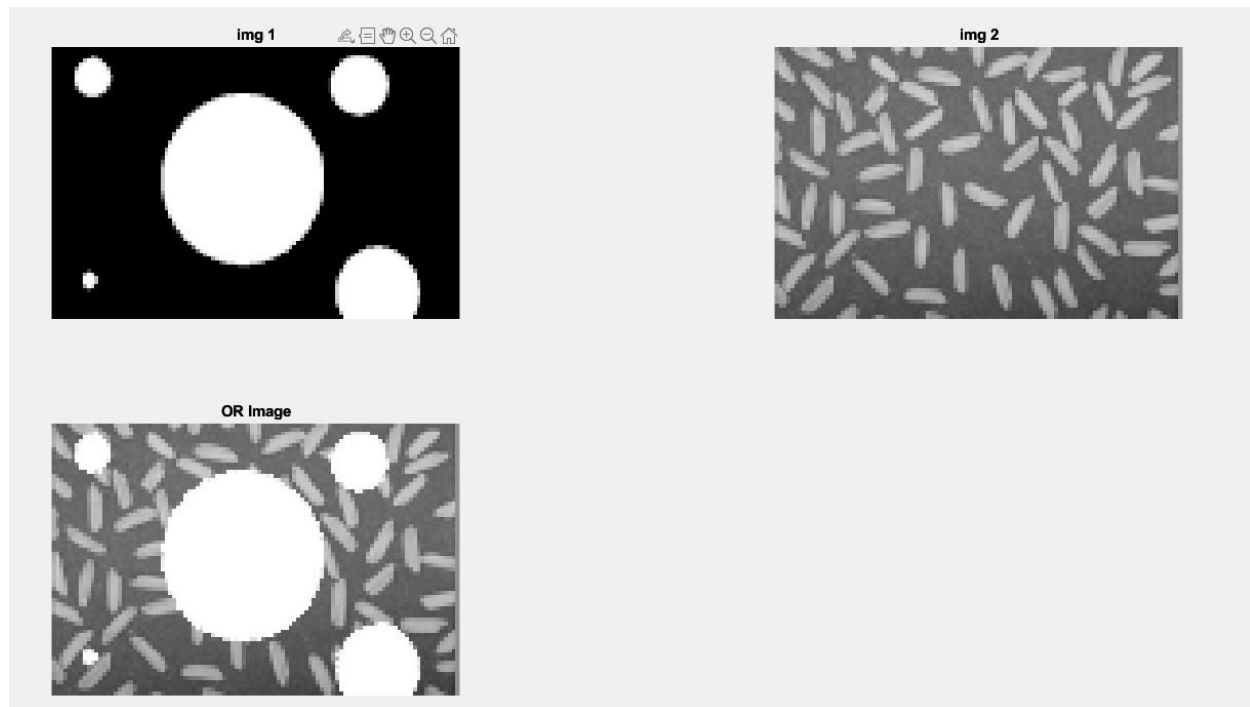


Fig: Both image and OR operation image

c) NOT

Code:

```
>> not1 = bitcmp(i1);
>> not2 = bitcmp(i2);
>> subplot(2,2,1);imshow(i1);title('img 1');
>> subplot(2,2,2);imshow(resize_image);title('img 2');
>> subplot(2,2,3);imshow(not1);title('Not on image1');
>> subplot(2,2,4);imshow(not2);title('Not on image2');
```

Output:

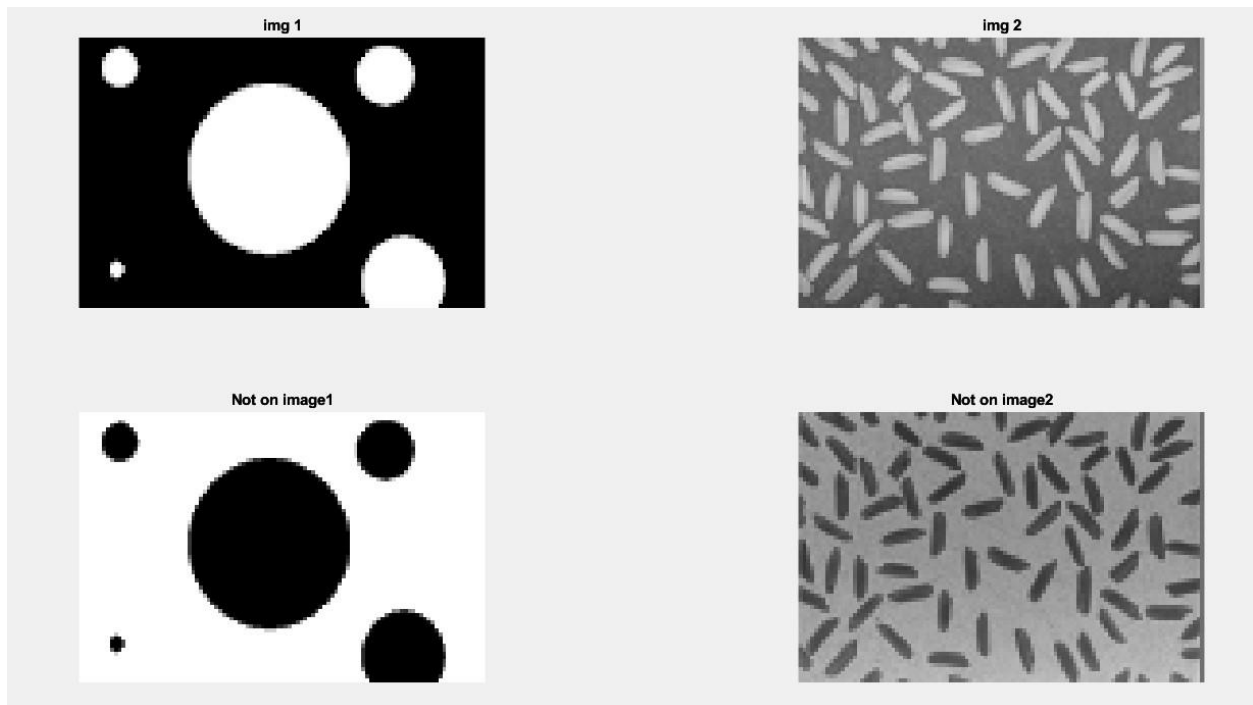


Fig: Both image and both NOT image

Question 7

Find the digital negative of the image.

Code:

```
>> i3 = imread('img3.png');
>> ni = 255 - i3;
>> imshowpair(i3,ni,'montage');
```

Output:

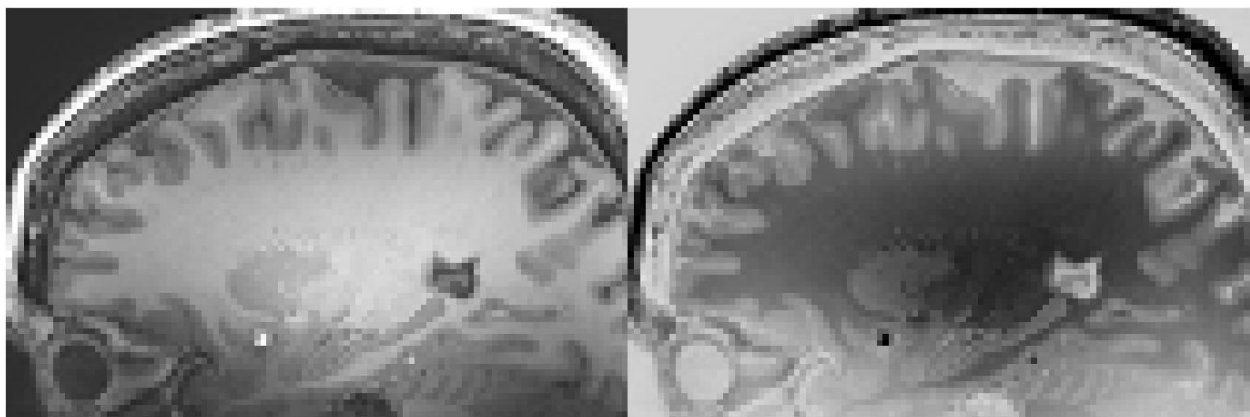


Fig: Normal image

Fig: Negative image

Question 8

Apply bit plane slicing on the image.

Code:

```
i4 = imread('img4.jpg');
i4 = double(i4);
B=bitget(i4,1);
subplot(2,4,1);imshow((B));title('Bit plane 1');
B=bitget(i4,2);
subplot(2,4,2);imshow((B));title('Bit plane 2');
B=bitget(i4,3);
subplot(2,4,3);imshow((B));title('Bit plane 3');
B=bitget(i4,4);
subplot(2,4,4);imshow((B));title('Bit plane 4');
B=bitget(i4,5);
subplot(2,4,5);imshow((B));title('Bit plane 5');
B=bitget(i4,6);
subplot(2,4,6);imshow((B));title('Bit plane 6');
B=bitget(i4,7);
subplot(2,4,7);imshow((B));title('Bit plane 7');
B=bitget(i4,8);
subplot(2,4,8);imshow((B));title('Bit plane 8');
```

Output:

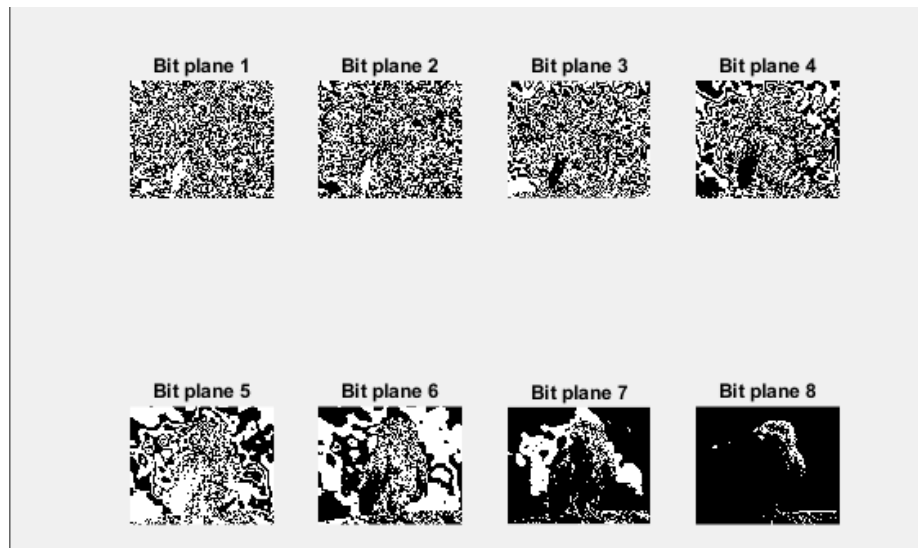


Fig: Plane slicing image

Question 9

Use contrast stretching on the image.

Code:

```
>> i4 = imread('img4.png');
>> cont = imadjust(i4,stretchlim(i4),[]);
>> imshowpair(i4,cont,'montage')
```

Output:



Fig: Normal image

Fig: Contrasted image

Question 9

Change the contrast of the image using Logarithmic Transformation and Power-law Transformation.

Logarithmic Transformation

Code:

```
i21 = imread('img2.png');  
i211 = double(i21);  
x = i211;  
[r,c] = size(i211);  
factor = 0.05;  
for i = 1:r;  
    for j = 1:c;  
        x(i,j) = factor*log(1+i211(i,j));  
    end  
end  
  
imshowpair(i21, x, 'montage')
```

Output:

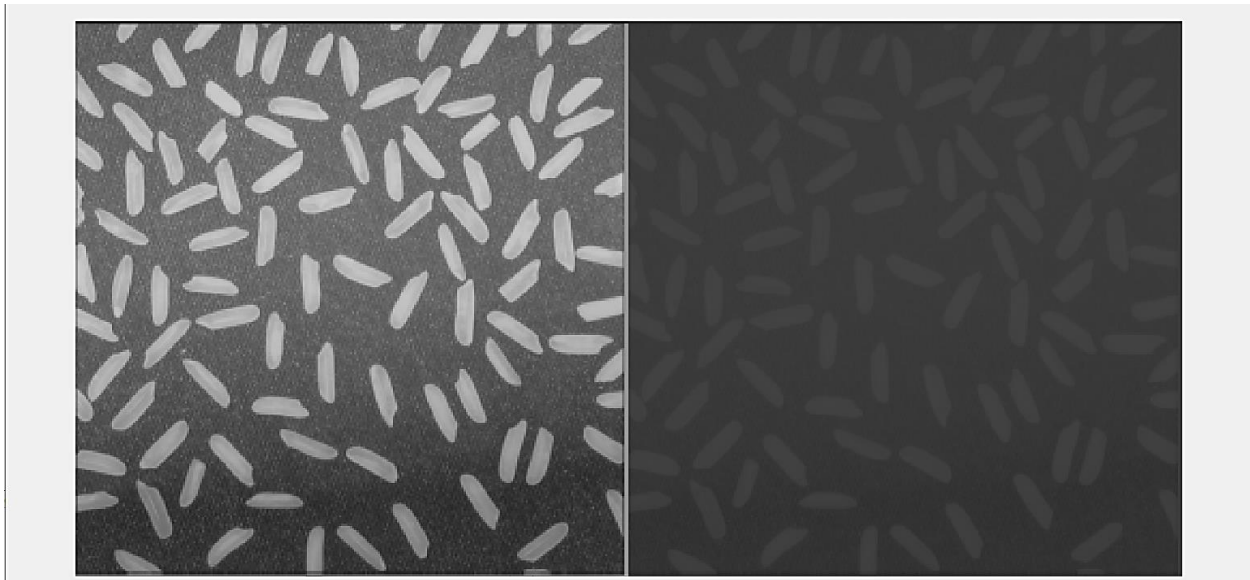


Fig: Normal image

Fig: Logarithmic transformation

Power-law Transformation

Code:

```
i21 = imread('img2.png');  
i211 = double(i21);  
x = i211;  
[r,c] = size(i211);  
factor = .00001;  
for i = 1:r;  
    for j = 1:c;  
        x(i,j) = factor*log(i211(i,j))^2;  
    end  
end  
  
imshowpair(i21, x, 'montage')
```

Output

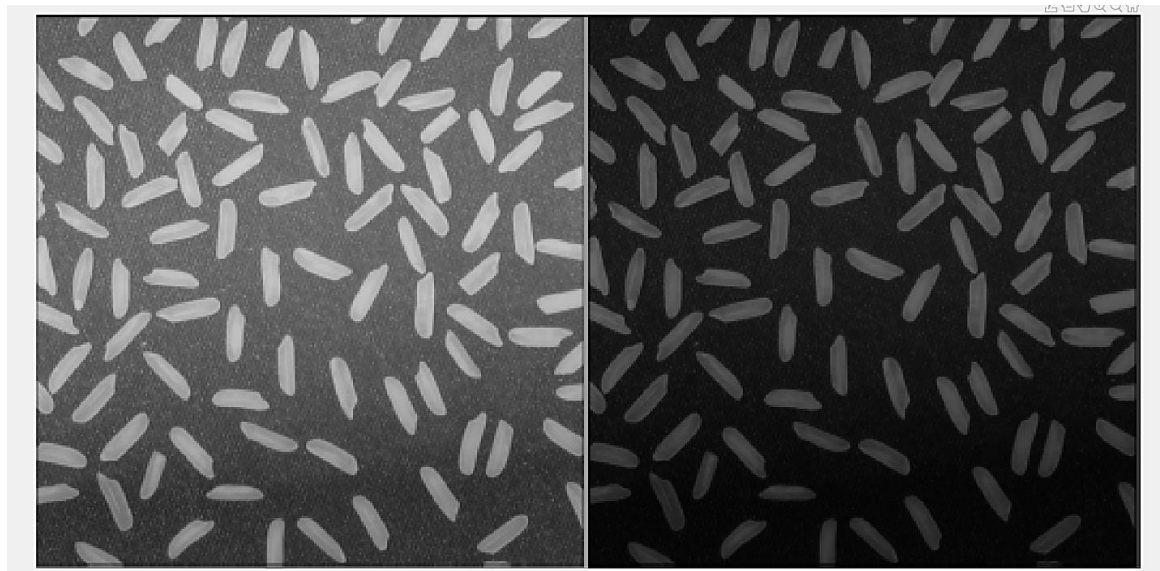


Fig: Normal image

Fig: Power law transformation