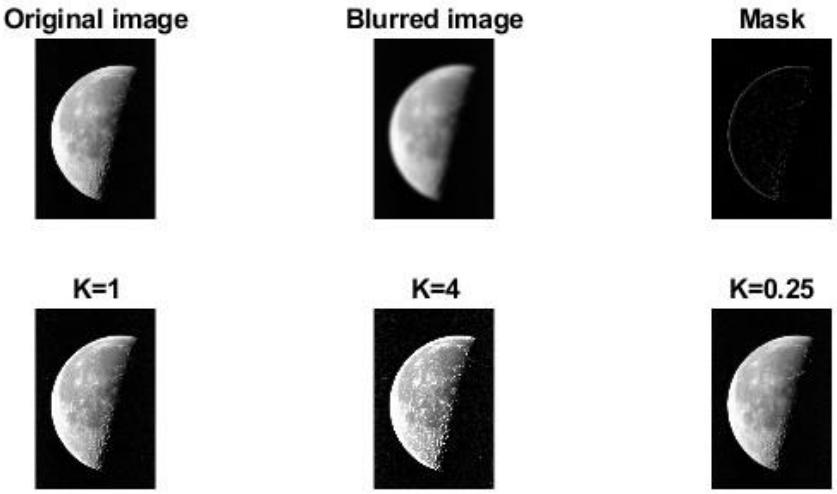
Q3)

```
I=imread("moon.tif");

%%a)
kernel = fspecial('gaussian', [31 31],5);
blurredImage = imfilter(I, kernel);

%%b)
mask = I-blurredImage;

%%c)
I1 = I+mask;
I2 = I+4*mask;
I3 = I+0.25*mask;
figure;
subplot(3,3,1)
imshow(I)
title('Original image')
subplot(3,3,2)
imshow(blurredImage)
title('Blurred image')
subplot(3,3,3)
imshow(mask)
title('Mask')
subplot(3,3,4)
imshow(I1)
title('K=1')
subplot(3,3,5)
imshow(I2)
title('K=4')
subplot(3,3,6)
imshow(I3)
title('K=0.25')
```

The techniques used are image sharpening using unsharp masking and high boost filtering. With the value of K>=1 the resulting image has more detailing as compared to the original image, But K<1 does not have any visible effects on the image.

OUTPUT:



Q4)

```
%%a)
img = imread('lena_color.jpg');
%%b)
blured_img = imgaussfilt3(img,2);
%%c)
median_filtered_img = medfilt3(img);
%%d)
mask1 = img-blured_img;
sharpened_img = img+5*mask1;
figure;
subplot(2,2,1)
imshow(img)
title('Original image')
subplot(2,2,2)
imshow(blured_img)
title('Blured image')
subplot(2,2,3)
imshow(median_filtered_img)
title('Median filtered image')
subplot(2,2,4)
imshow(sharpened_img)
title('Sharpened image K=5')
```

Median filtering the image results in smoothening of sharp details in the image, Whereas gaussian filtering blurs the complete image uniformly. Sharpening the image leads to enhancing the sharp edges present in the image. In this case, due to high value of K, there are some serious unwanted artifacts in the image.

OUTPUT:

Original image

Blured image

Median filtered image

Sharpened image K=5

Q5)

```
sigma = 1;
gauss_kernel = zeros(7,7);
for x=-3:3
    for y=-3:3
        gauss_kernel(x+4,y+4) = (1/(2*pi*(sigma^2)))*exp(-((x^2+y^2)/(2*sigma^2)));
    end
end
image = imread('C:\Program Files\MATLAB\R2022a\toolbox\images\imdata\coins.png');
[l,b] = size(image);
new_image = zeros(l,b);
for i=4:l-3
    for j=4:b-3
        sum=0;
        for m=-3:3
            for n=-3:3
                sum = sum+gauss_kernel(m+4,n+4)*image(i+m,j+n);
            end
        end
        new_image(i,j)=sum;
    end
end
new_image=mat2gray(new_image);
new_image=uint8(255*new_image);
%
%
%
```

Yes, it is possible to use smoothing filters for edge detection.

```
%Yes it is possible to use smoothing filters for edge detection
edges = image-new_image;
figure;
subplot(1,3,1)
imshow(image)
title('Original image')
subplot(1,3,2)
imshow(new_image)
title('Blured image')
subplot(1,3,3)
imshow(edges)
title('Edges')
```

OUTPUT: