Assignment 2
Vikram Shahapur
14597876
Ece 620

Part 1

Answer 1)
Code: bit_plane.m

```
function [image] = bit_plane(image, bit_plane_number)

image = double(image);
% Extract the specified bit plane
bit_plane_image = bitget(image, bit_plane_number);

% Display the bit plane
figure;
imshow(bit_plane_image);

end

clc
close all

peppers = imread('peppers.tif');
baboon = imread('baboon.tif');
%barbara = imread('Barbara.bmp');
bit_plane_number = 4;

peppers = bit_plane(peppers, bit_plane_number);
baboon = bit_plane(baboon, bit_plane_number);
%barbara = bit_plane(barbara, bit_plane_number);
```

The highest bitplane for Peppers is bitplane 3 or 4 that no longer resembles image contect instead it appears noise. And for baboon it is bitplane 4 or 5.

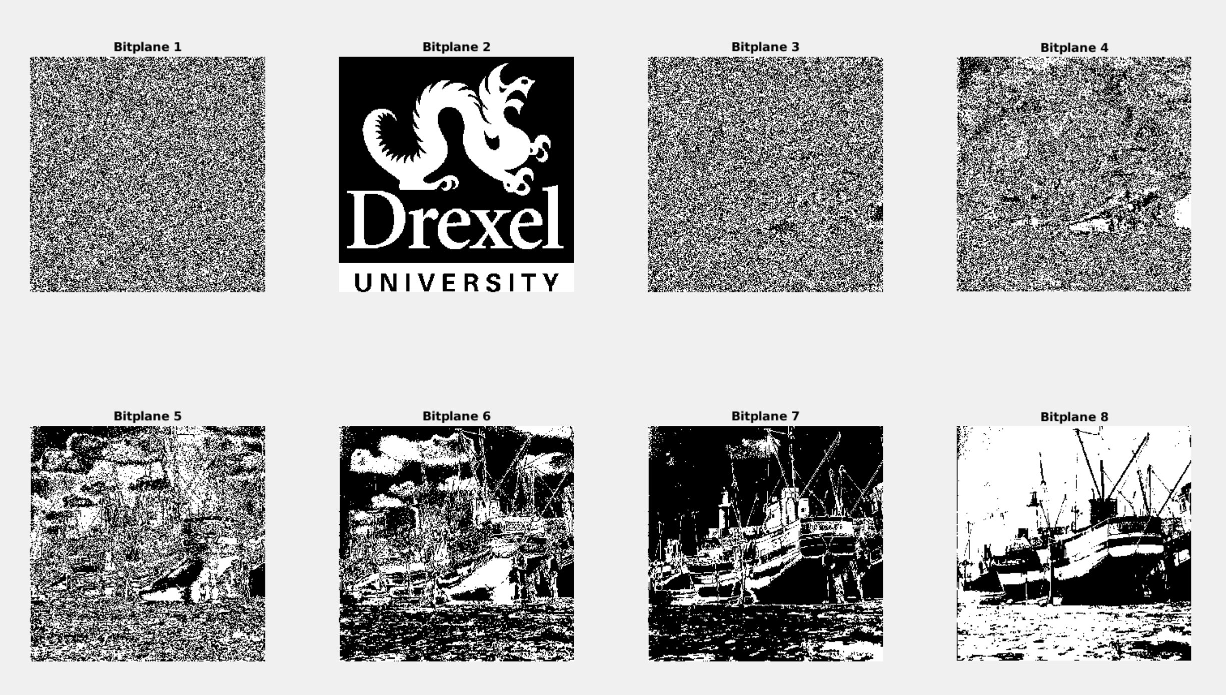
The peppers image has more details and structures, so higher bitplane retain more content. Baboon has less complex structures and details, so higher bitplanes already appear noise.

Answer 2)
Code: bitplane_imgs.m

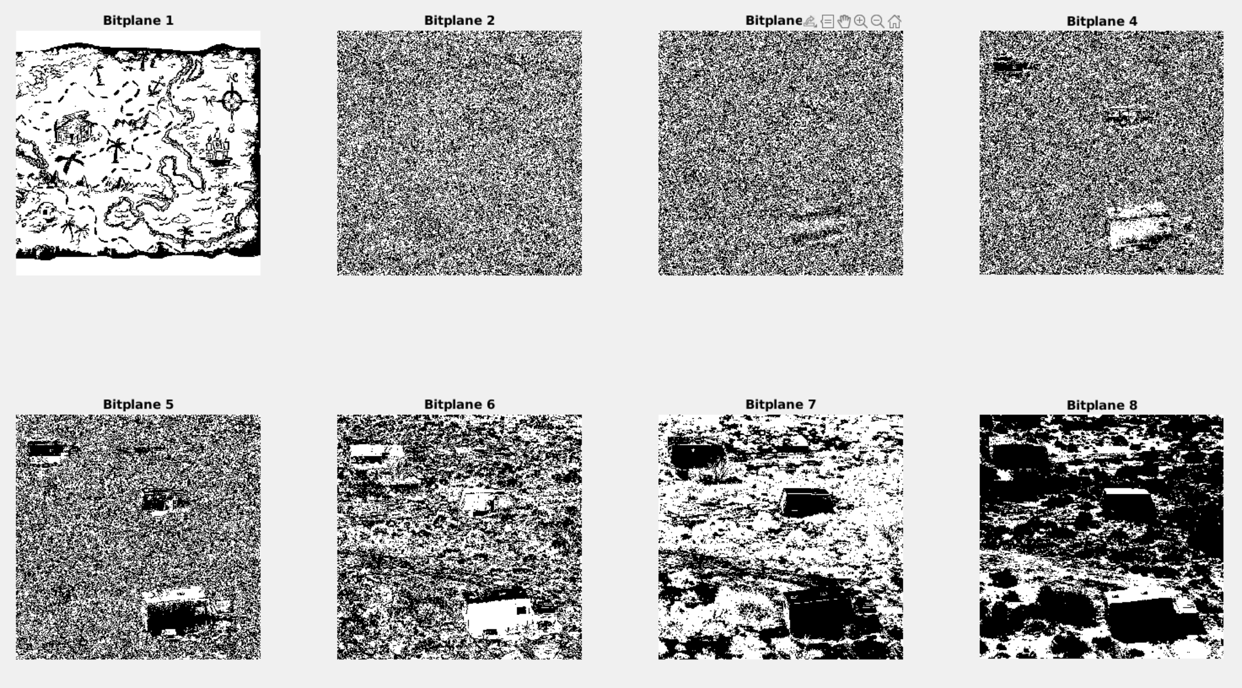
```
 function [image] = bitplane_imgs(image)


%image_two = imread('LSBwmk2.tiff');
%image_three = imread('LSBwmk3.tiff');

image = double(image);
%image_t = double(image_two);
%image_th = double(image_three);

bitplane_imgo1 = bitget(image, 1);
bitplane_imgo2 = bitget(image, 2);
bitplane_imgo3 = bitget(image, 3);
bitplane_imgo4 = bitget(image, 4);
bitplane_imgo5 = bitget(image, 5);
bitplane_imgo6 = bitget(image, 6);
bitplane_imgo7 = bitget(image, 7);
bitplane_imgo8 = bitget(image, 8);

figure;
subplot(2,4,1);
imshow(bitplane_imgo1);
title('Bitplane 1');
subplot(2,4,2);
imshow(bitplane_imgo2);
title('Bitplane 2');
subplot(2,4,3);
imshow(bitplane_imgo3);
title('Bitplane 3');
subplot(2,4,4);
imshow(bitplane_imgo4);
title('Bitplane 4');
subplot(2,4,5);
imshow(bitplane_imgo5);
title('Bitplane 5');
subplot(2,4,6);
imshow(bitplane_imgo6);
title('Bitplane 6');
subplot(2,4,7);
imshow(bitplane_imgo7);
title('Bitplane 7');
subplot(2,4,8);
imshow(bitplane_imgo8);
title('Bitplane 8');

end
```

```
clc
close all

img1 = imread('LSBwmk1.tiff');
img2 = imread('LSBwmk2.tiff');
img3 = imread('LSBwmk3.tiff');

img1 = bitplane_imgs(img1);
img2 = bitplane_imgs(img2);
img3 = bitplane_imgs(img3);
```

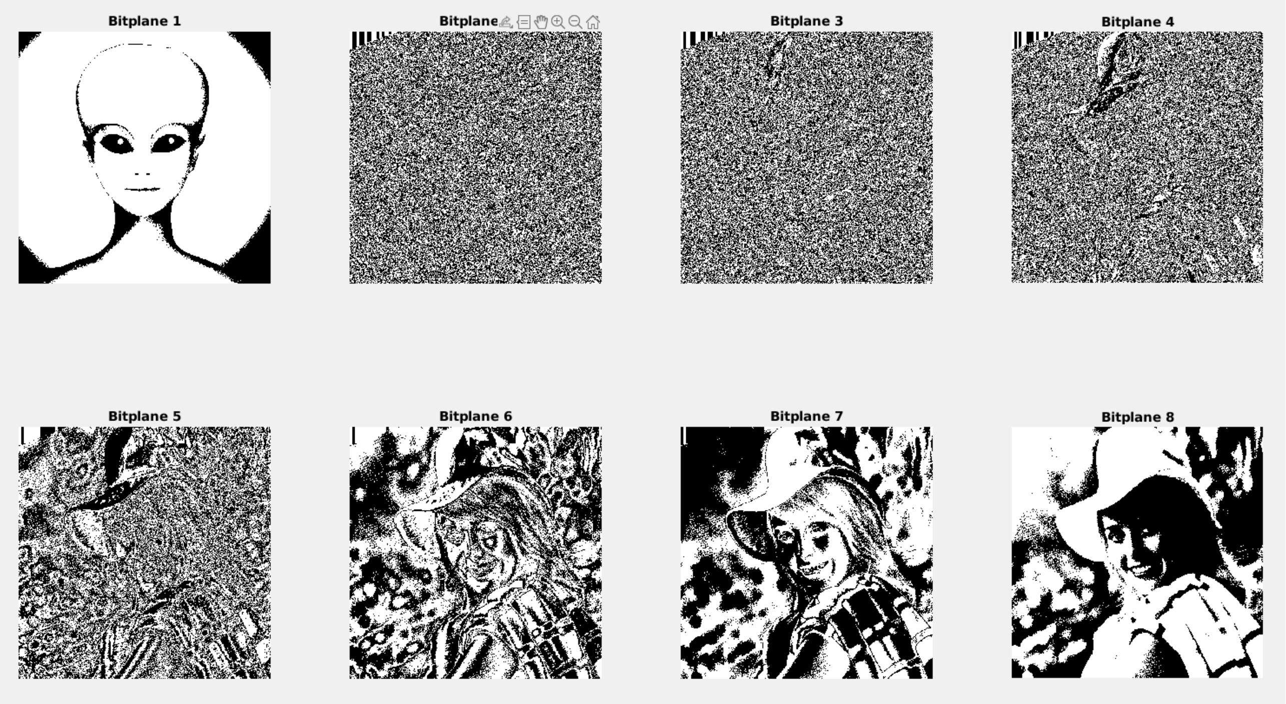
LSBwmk1



LSBwmk2



LSBwmk3



Answer 3)
Code: hide_image.m

```
function [stego] = hide_image(image, hidden_image, N)

oi1 = double(image);
hi = double(hidden_image);

%N = 6;

hishift = bitshift(hi,-(8-N));
size(hishift)
showhi = uint8(hishift);
showhi = bitshift(showhi, 8-N);


%now zero out imbed bits in cover image
coverzero = oi1;
for i=1:N
   coverzero=bitset(coverzero,i,0);
end
%now add message image and cover image
coverzero1=imresize(coverzero,[512,512]);
coverzero1=double(coverzero1);
size(coverzero1)
stego = uint8(coverzero1-hishift);

figure(),
subplot(1,2,1);imshow(showhi);title('embed Image to Hide ');
subplot(1,2,2);imshow(stego);title('Stego image');

imwrite(stego,'modified_image.bmp');
end
```

```
clc
close all

peppers = imread('peppers.tif');
hidden_image = imread('Barbara.bmp');


hide1 = hide_image(peppers, hidden_image, 4);
```

4 bit planes of Barabara can be embedded before noticing the distortion.

```
clc
close all

baboon = imread("baboon.tif");
hidden_image = imread('Barbara.bmp');

hide2 = hide_image(baboon, hidden_image,5);
```

5 bit planes of Barbara can be embedded before noticing the distortion.

5 bit planes of Barbara can be embed in Pepper before you notice the hidden image.
6 bit planes of Barbara can be embed in Baboon before you notice the hidden image.
The number of bit planes you can embed is different because the details of the image are different and it depends on the contrast and complexity of the hidden image.

Part 2

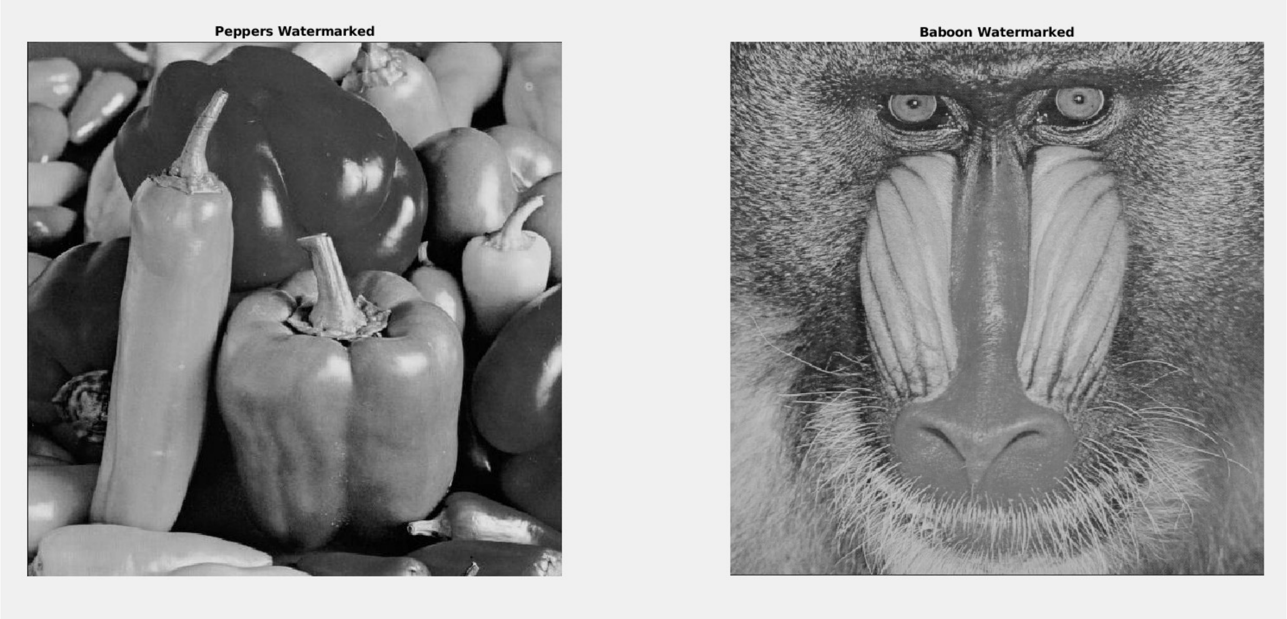
Answer 1)
Code: yeung_mintzer_watermark.m

```
function [watermarked_image] = yeung_mintzer_watermark(image, watermark, key)

% Convert image to double precision
    image = im2double(image);
    watermark = im2double(watermark);

% Generate lookup table using key
    rng(key);
    lut = double(rand(256,2) > 0.5);
    size(lut)
% Flatten image and watermark
    flat_image = reshape(image, 1, []);
    flat_watermark = reshape(watermark, 1, []);

% Embed watermark into image
    for i = 1:length(flat_image)
        pixel_value = round(max(min(flat_image(i)*255, 255), 0));
        watermark_bit = logical(flat_watermark(i));
        embedded_bit = lut(pixel_value+1, watermark_bit+1);
        flat_image(i) = (pixel_value + embedded_bit) / 255;
    end

% Reshape image back to original dimensions
    watermarked_image = im2uint8(reshape(flat_image, size(image)));

    figure;
    imshow(watermarked_image);

    imwrite(watermark,'modified_image.tif');
end
```

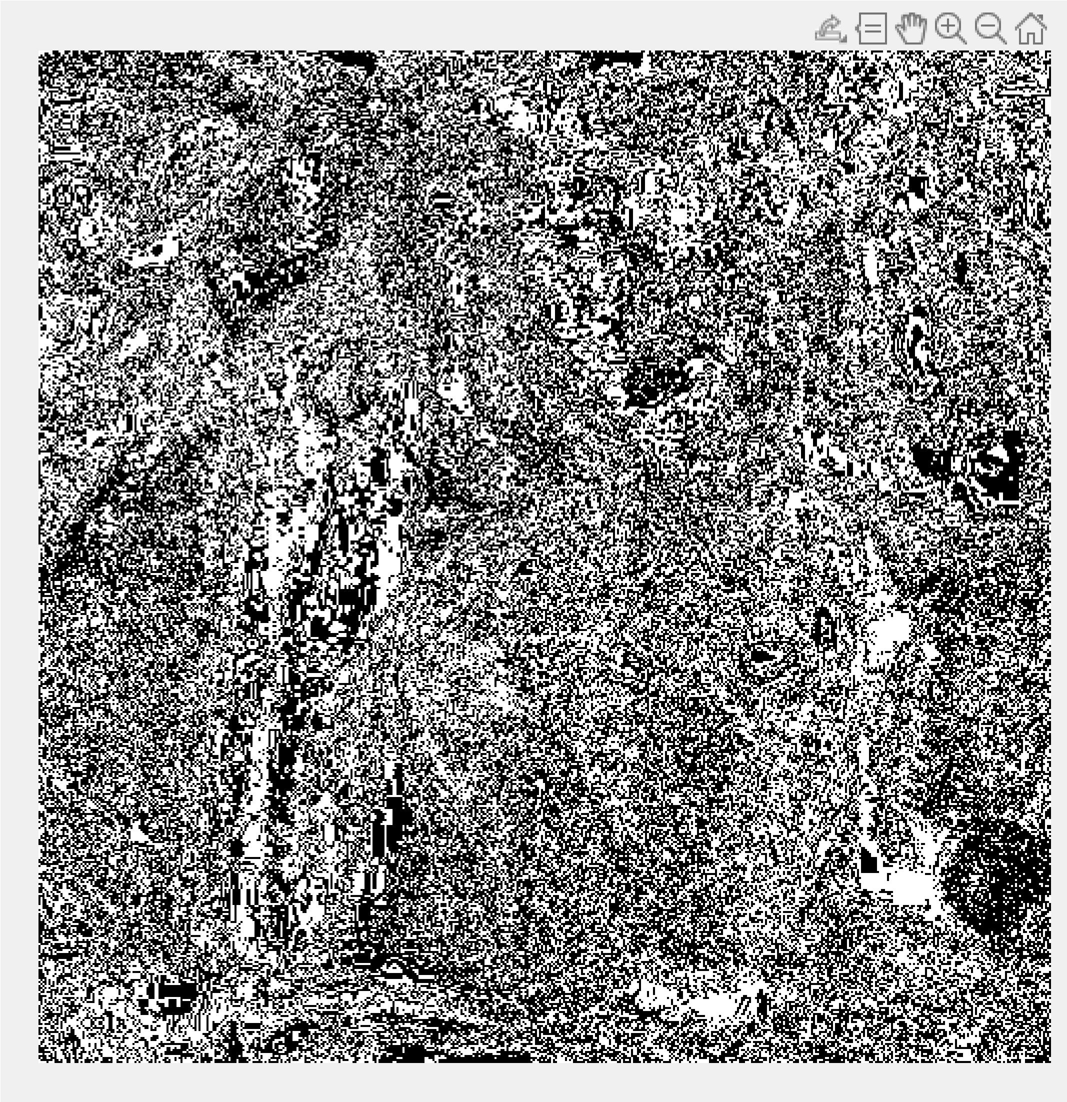
Answer 2)
Code:

```
% Load images
peppers = imread('peppers.tif');
baboon = imread('baboon.tif');
watermark = imread('Barbara.bmp');

% Extract the most significant bit plane of the watermark
msb_watermark = bitget(watermark, 8);

% Embed watermark into peppers image
key = 123;
peppers_watermarked = yeung_mintzer_watermark(peppers, msb_watermark, key);

% Embed watermark into baboon image
baboon_watermarked = yeung_mintzer_watermark(baboon, msb_watermark, key);

% Display watermarked images
figure;
subplot(1, 2, 1);
imshow(peppers_watermarked);
title('Peppers Watermarked');

subplot(1, 2, 2);
imshow(baboon_watermarked);
title('Baboon Watermarked');
%imwrite(peppers_watermarked,'pwmk.bmp');

lsb_pepperswmkd = bit_plane(peppers_watermarked,1);
lsb_baboonwmkd = bit_plane(baboon_watermarked,1);

psnr_value = psnr(peppers,peppers_watermarked);
psnr_value1 = psnr(baboon,baboon_watermarked);

fprintf('PSNR value is %f dB\n', psnr_value);
fprintf('PSNR value is %f dB\n', psnr_value1);
```

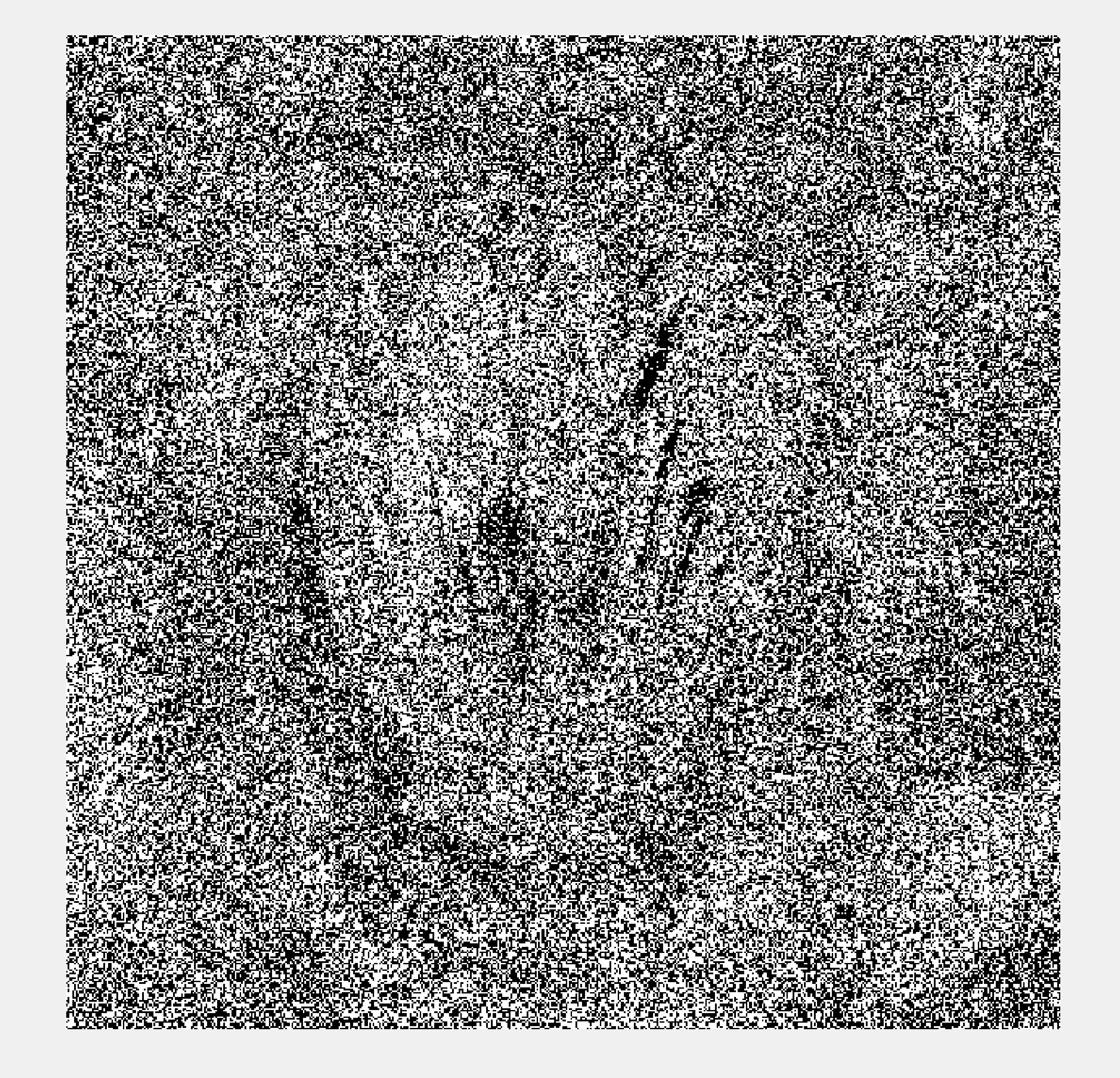
No the hidden watermark is not visually detectable.



Least significant bit plane of watermarked Peppers image.



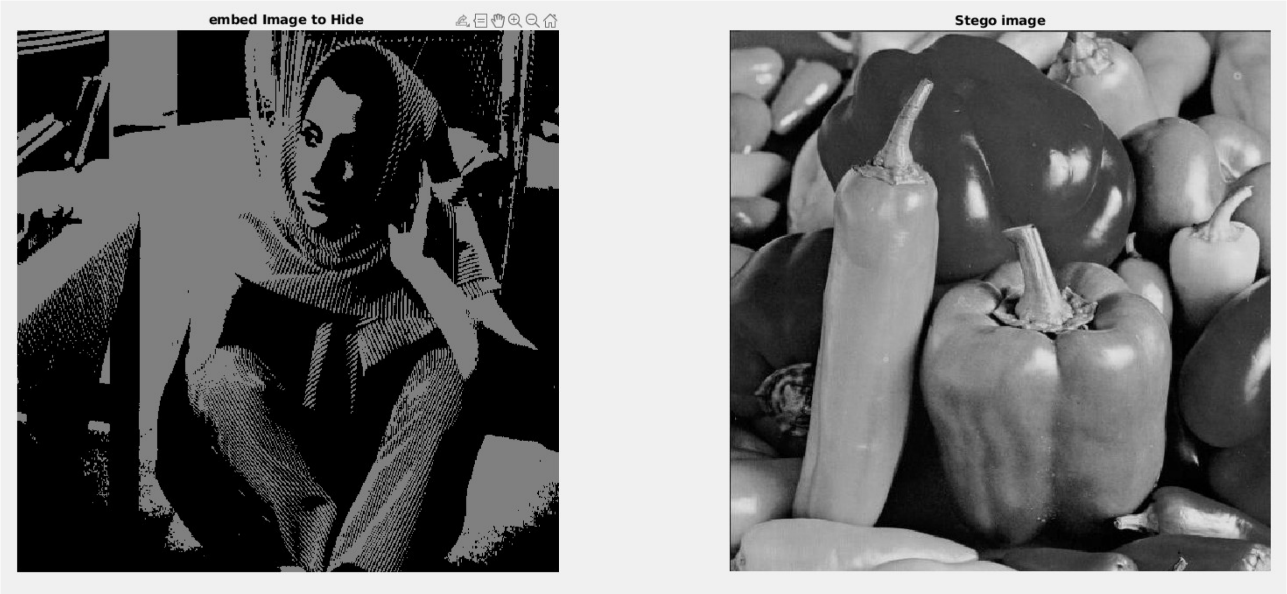
Least significant bit plane of watermarked Baboon image.

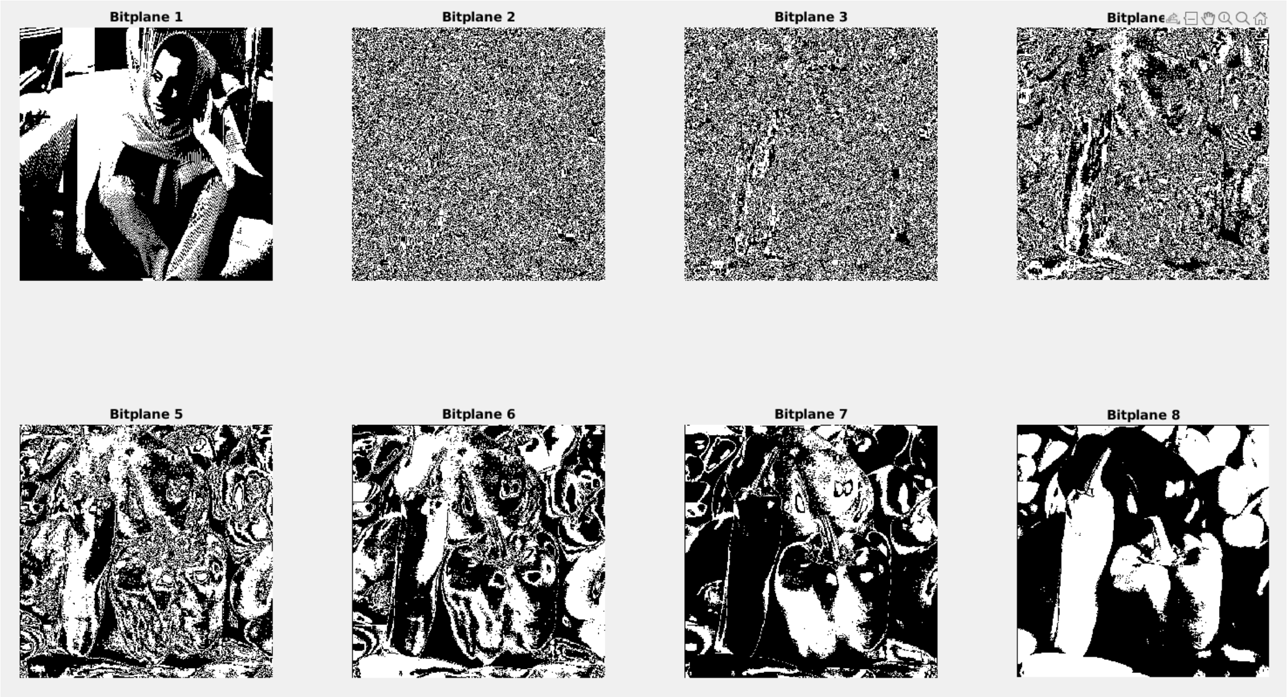


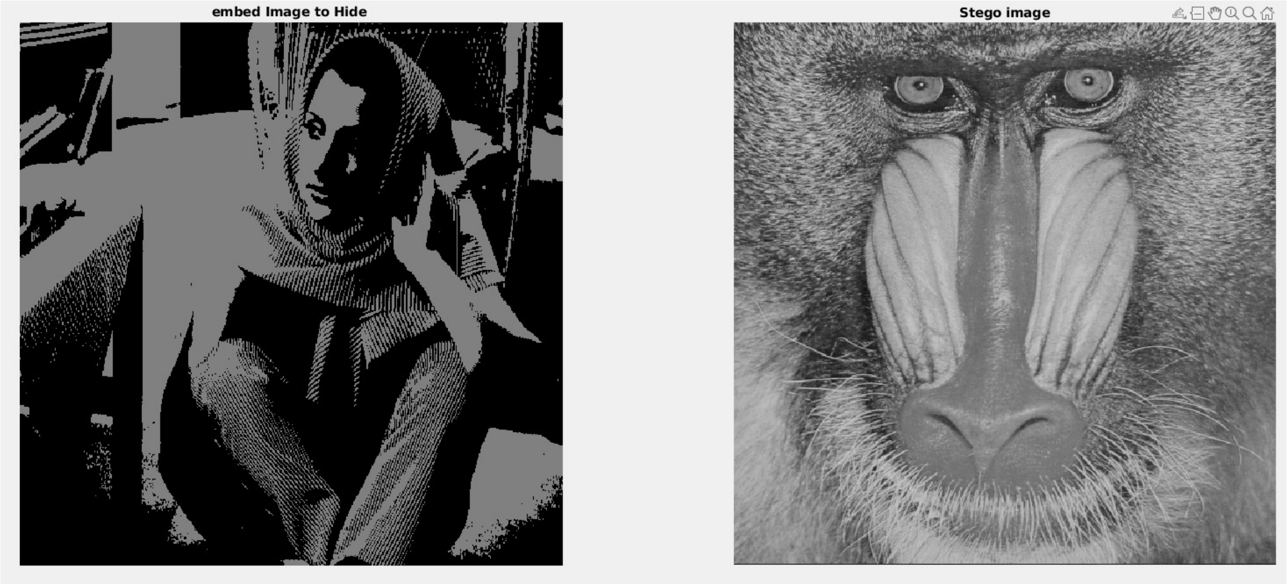
The PSNR between the original version of Peppers image and itsYeung Mintzer watermarked version is 51.03690 dB.
The PSNR between the original version of Baboon image and itsYeung Mintzer watermarked version is 51.056026 dB.

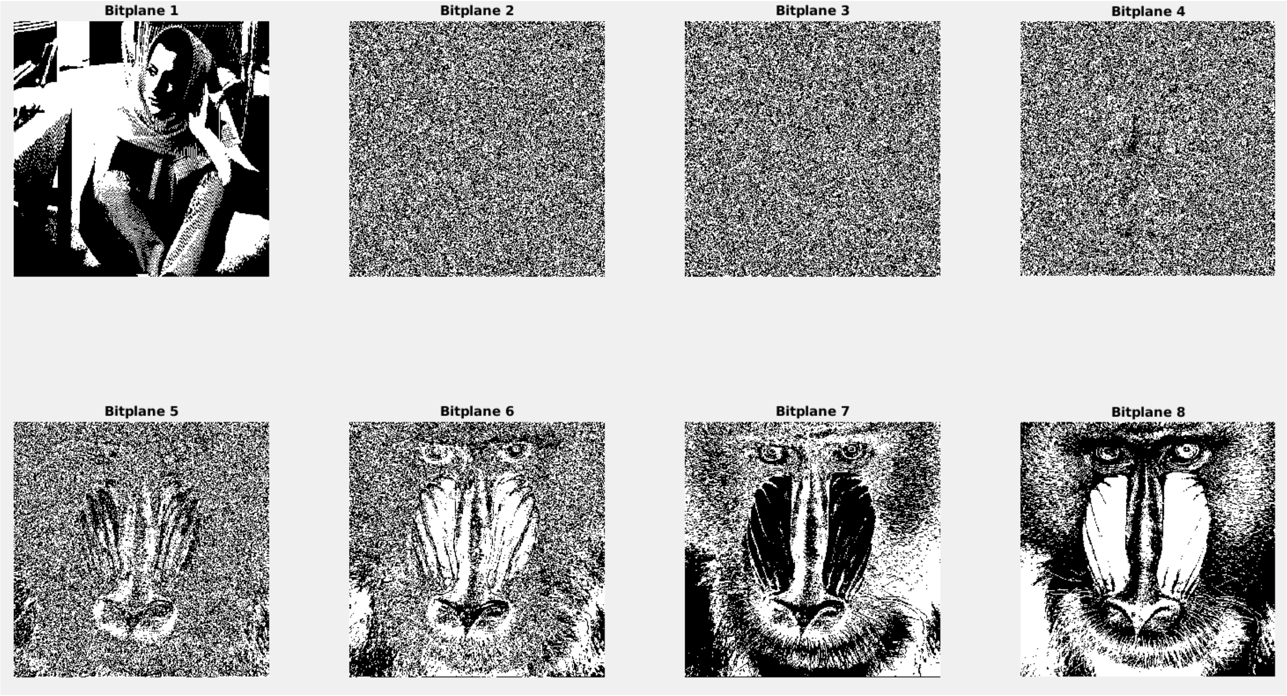
```
clc
close all

peppers = imread('peppers.tif');
baboon = imread('baboon.tif');

hidden_img = imread('Barbara.bmp');
```

```
% Embed the hidden bit plane in the LSB of both images
stego1 = hide_image(peppers, hidden_img,1);
stego2 = hide_image(baboon, hidden_img,1);

% Calculate the PSNR between original and stego images
psnrp = psnr(peppers, stego1);
psnrb = psnr(baboon, stego2);

% Display the PSNR values
disp(['PSNR for peppers.tif: ', num2str(psnrp)])
disp(['PSNR for baboon.tif: ', num2str(psnrb)])

img = bitplane_imgs(stego1);
img1 = bitplane_imgs(stego2);
```









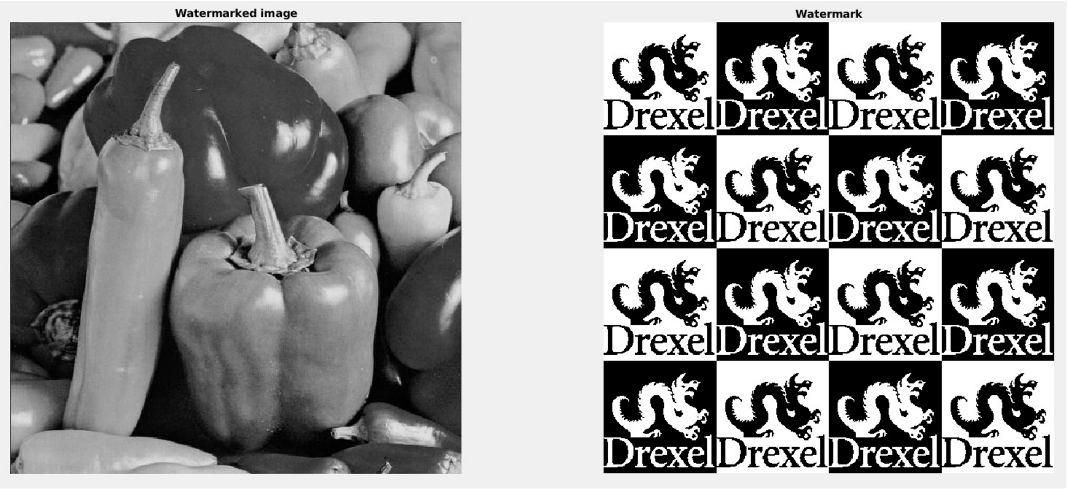
The PSNR between the original version of Peppers image and its watermark using LSB embedding function is 47.0891 dB.
The PSNR between the original version of Baboon image and its watermark using LSB embedding function is 47.0601 dB.
These PSNR values are lower than the PSNR values obtained using Yeung Mintzer algorithm. The reason is LSB algorithm will change the LSB's of original image to embed the watermark, these will lead to distortions in image. Yeung Mintzer uses a quantization step to embed the watermark, which distorts the image less.

Answer 3)
Code:

```
function [watermark] = yeung_mintzer_extract(image, key)

% Convert image to double precision
image = im2double(image);

% Generate lookup table using key
rng(key);
lut = double(rand(256, 2) > 0.5);

% Flatten image
flat_image = reshape(image, 1, []);

% Extract watermark from image
watermark = zeros(size(flat_image));
for i = 1:length(flat_image)
    pixel_value = round(max(min(flat_image(i)*255, 255), 0));
    watermark_bit = lut(pixel_value+1, 1);
    watermark(i) = watermark_bit;
end


% Reshape watermark back to original dimensions
watermark = im2double(reshape(watermark, size(image)));

figure;
subplot(1,2,1);
imshow(image);title('Watermarked image');
subplot(1,2,2);
imshow(watermark);title('Watermark');

end
```

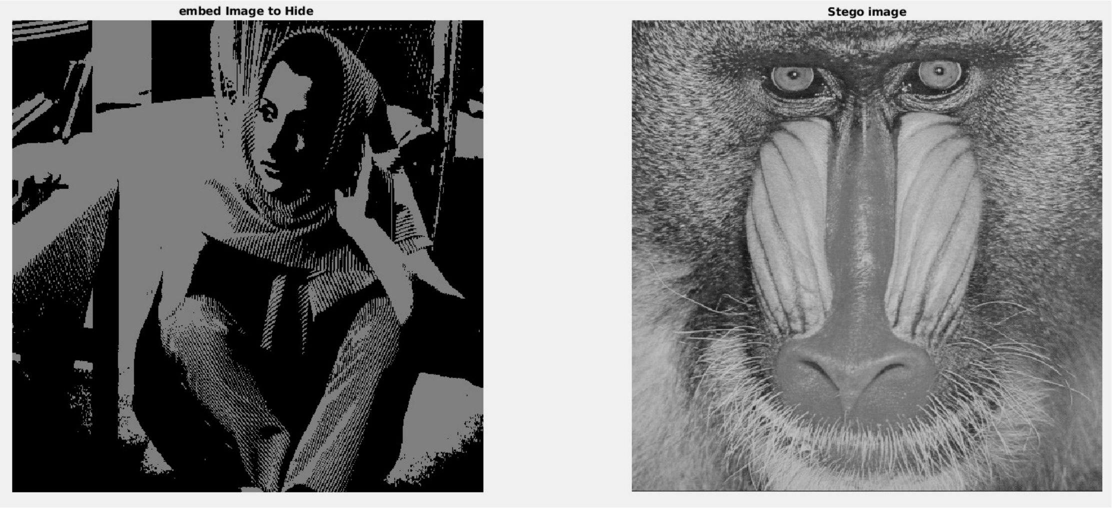
YMwmkedKey435

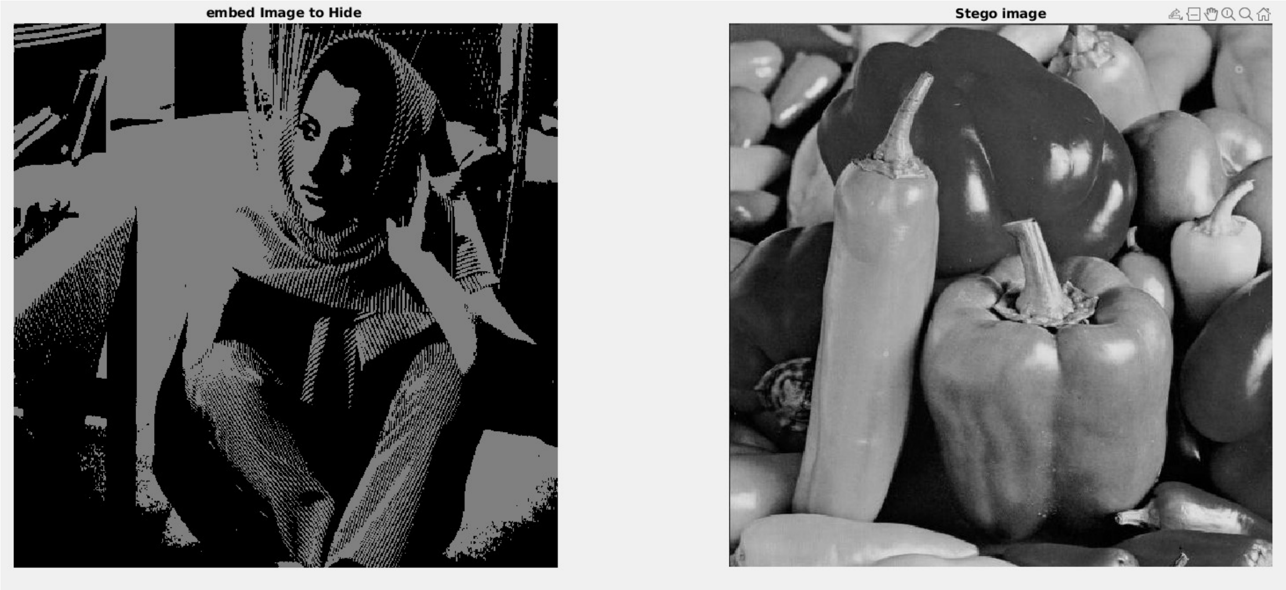
```
clc
close all

image = imread('YMwmkedKey435.tiff');
key = 435;
watermark = yeung_mintzer_extract(image, key);
```

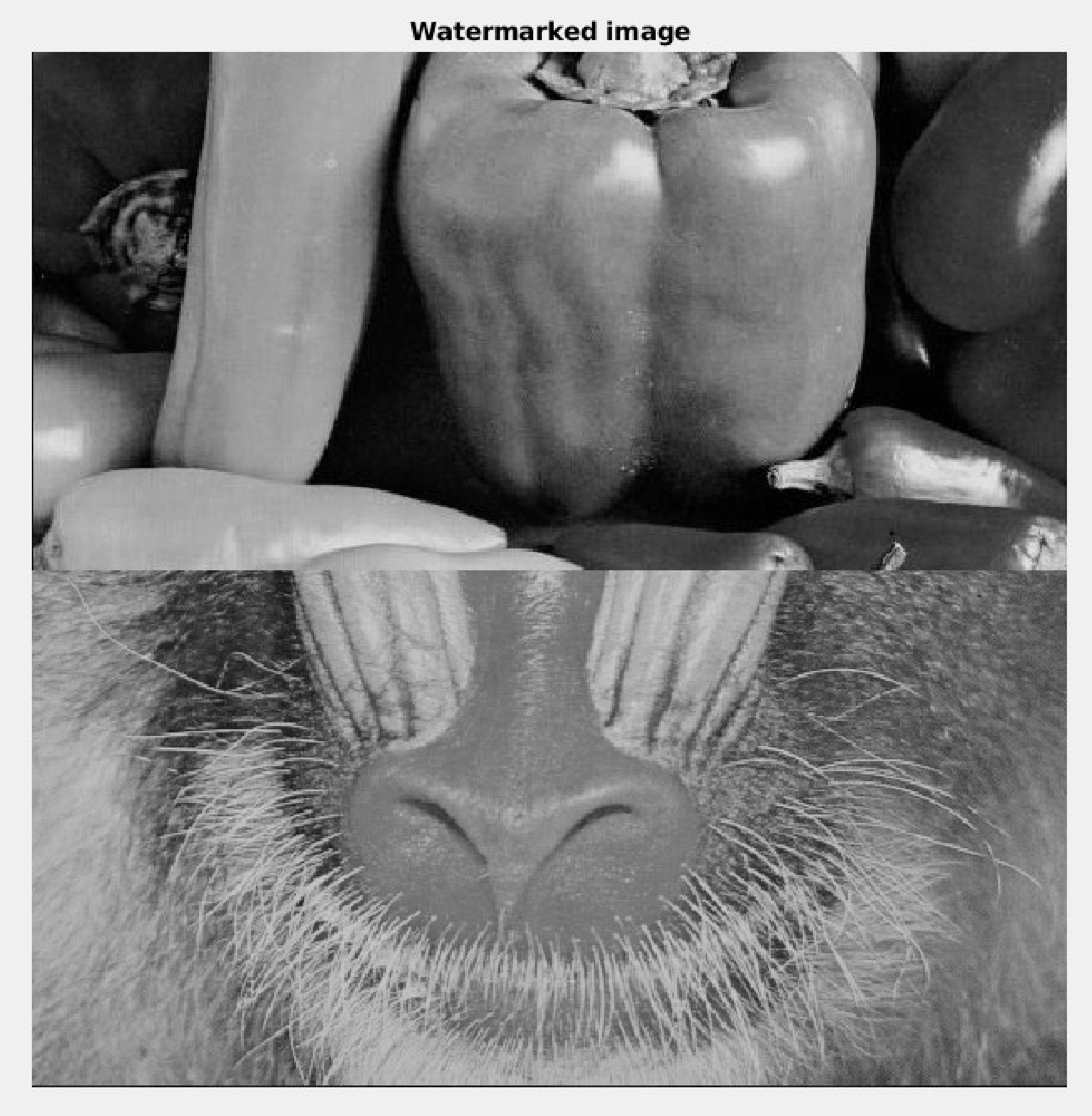


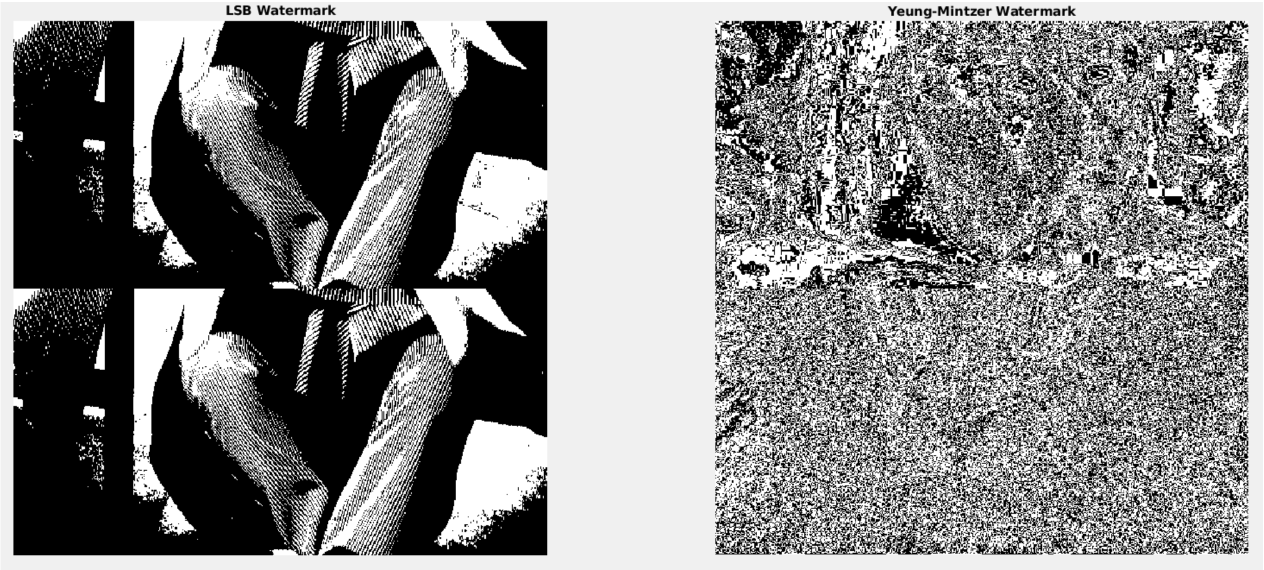
Answer 4)
Code:

```
clc
close all

baboon = imread('baboon.tif');

hidden_img = imread('Barbara.bmp');
lsb_baboons = hide_image(baboon, hidden_img,1);


% Embed watermark into peppers image
key = 123;
ymw_baboons_watermarked = yeung_mintzer_watermark(baboon, hidden_img, key);

% Load the images
peppers = imread('peppers.tif');

% Get the dimensions of the images
[height, width] = size(lsb_baboons);

% Replace the top half of the images
half_height = round(height / 2);
lsb_baboons(1:half_height,:,:) = peppers(half_height+1:end,:,:);
ymw_baboons_watermarked(1:half_height,:,:) = peppers(half_height+1:end,:,:);

%lsb_baboons
% Extract the watermarks
watermark_lsb = extract_LSB_watermark(lsb_baboons);
watermark_yeung = yeung_mintzer_extract(ymw_baboons_watermarked,key);


% Display the watermarks
figure;
subplot(1,2,1);
imshow(watermark_lsb,[]);
title('LSB Watermark');
subplot(1,2,2);
imshow(watermark_yeung);
title('Yeung-Mintzer Watermark');
```

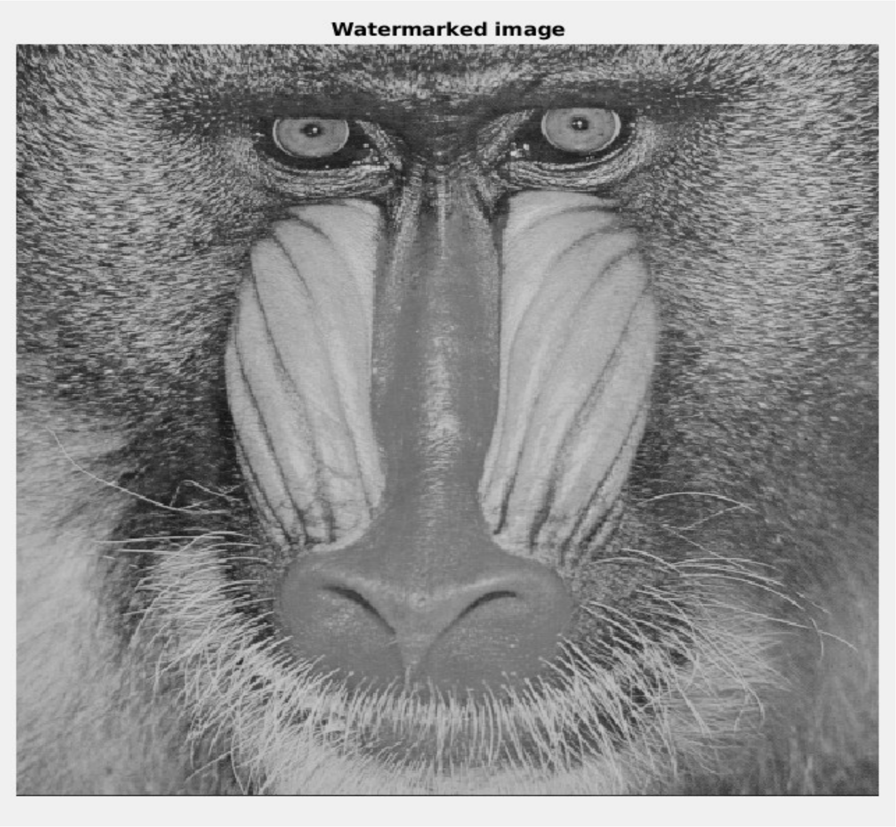


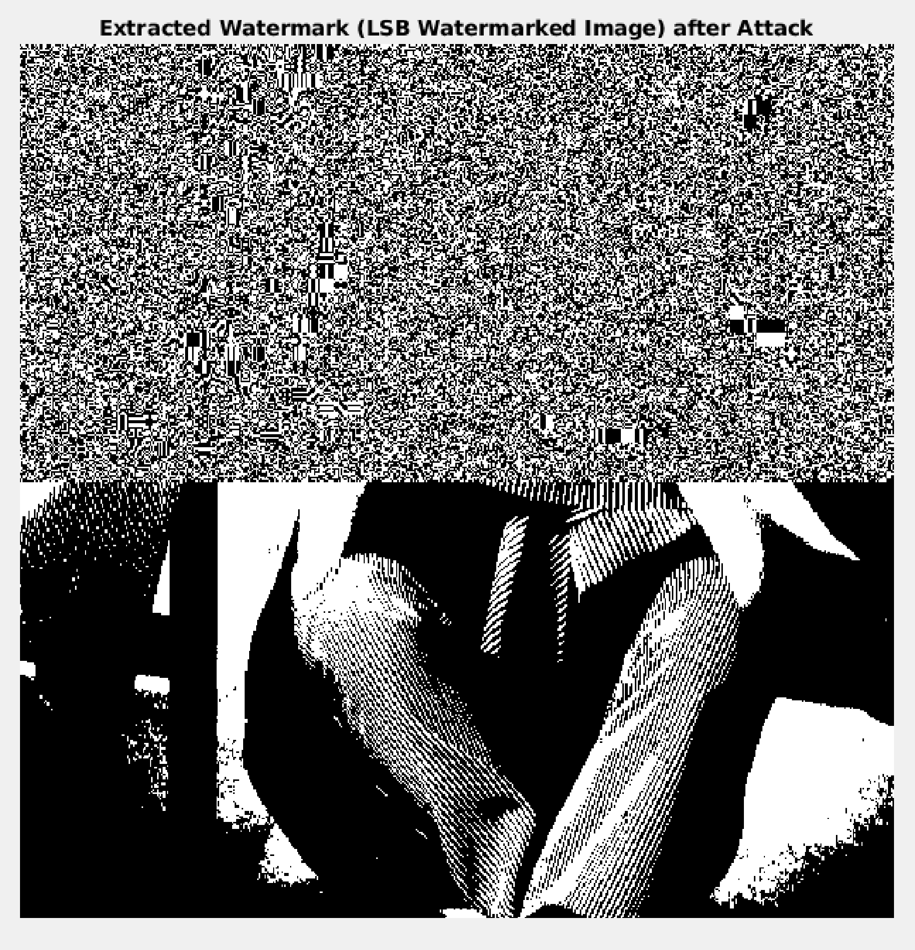

embed Image to Hide
Stego image

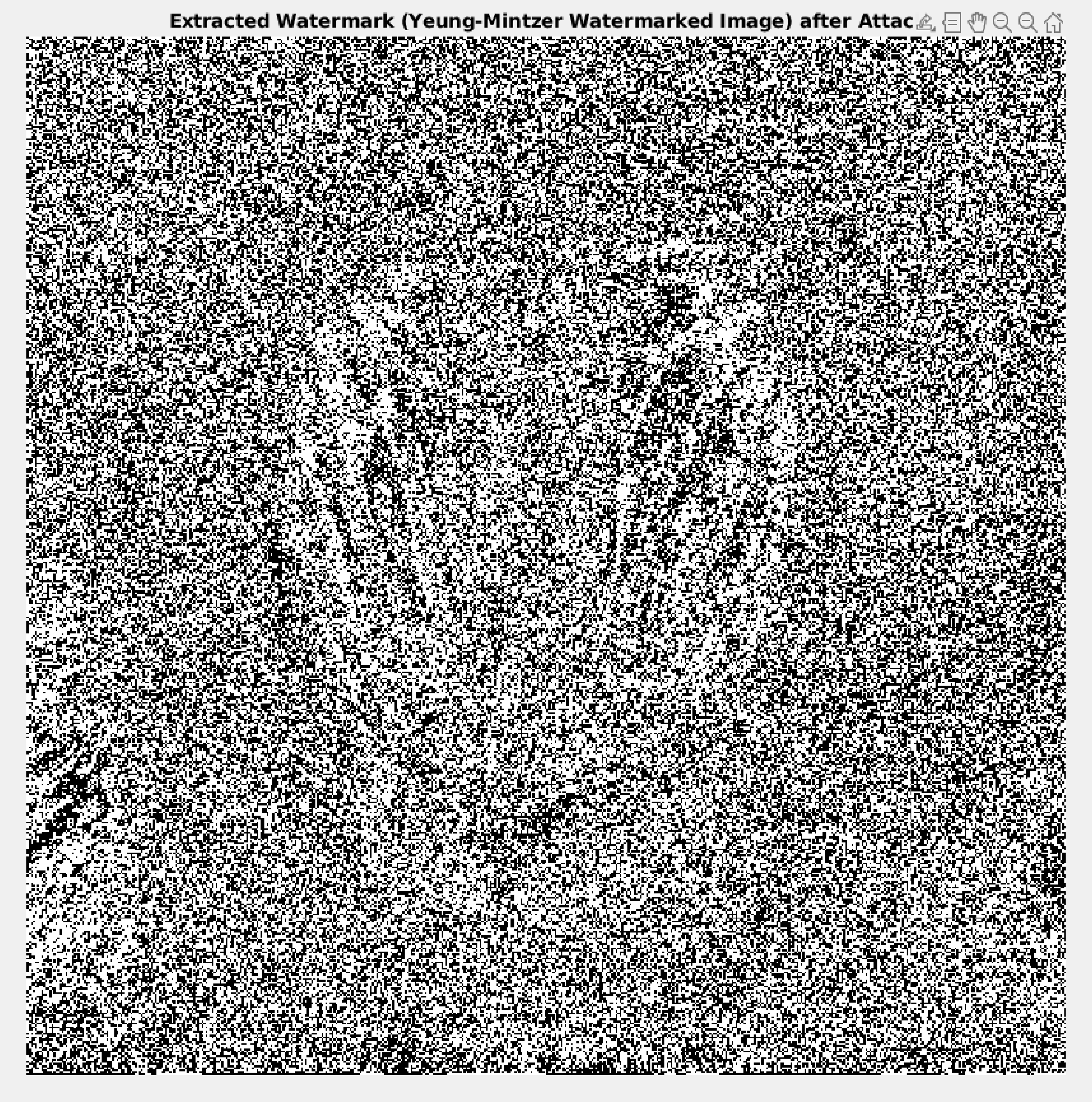

Watermarked image



```
clc
close all

%peppers = imread('peppers.tif');
baboon = imread('baboon.tif');

hidden_img = imread('Barbara.bmp');
lsb_baboons = hide_image(baboon, hidden_img,1);


% Embed watermark into peppers image
key = 123;
ymw_baboons_watermarked = yeung_mintzer_watermark(baboon, hidden_img, key);


% load the watermarked images
%%baboon_lsb = imread('baboon_lsb_watermarked.bmp');
%baboon_yeung = imread('baboon_yeung_watermarked.bmp');

peppers = imread('peppers.tif');
peppers_bottom = peppers(ceil(size(peppers,1)/2):end,:,:);
peppers_bottom = peppers_bottom(1:size(lsb_baboons,1)/2,:,:);

% replace the top half of baboon_lsb with the bottom half of peppers
baboon_lsb_attack = lsb_baboons;
baboon_lsb_attack(1:ceil(size(lsb_baboons,1)/2),:,:) = peppers_bottom;

% extract the watermark from the attacked images
watermark_lsb_attack = extract_LSB_watermark(baboon_lsb_attack);
watermark_yeung_attack = yeung_mintzer_extract(ymw_baboons_watermarked, key);

% display the extracted watermarks
figure(1);
imshow(watermark_lsb_attack,[]);
title('Extracted Watermark (LSB Watermarked Image) after Attack');

figure(2);
imshow(watermark_yeung_attack);
title('Extracted Watermark (Yeung-Mintzer Watermarked Image) after Attack');
```


Watermarked image


Extracted Watermark (LSB Watermarked Image) after Attack


Extracted Watermark (Yeung-Mintzer Watermarked Image) after Attac

It can be done for LSB water marked image by selecting pixels that have the same LSB value as the original pixels in top half of Baboon image.
It is not possible to extract for Yeung Mintzer watermaked image without knowing the key or look up table. It involves permutation process that rearranges the image pixels based on key or look up table. It is not possible replace without destroying the watermark.

Thank you !!!