PROBLEM 6 :

PART A:

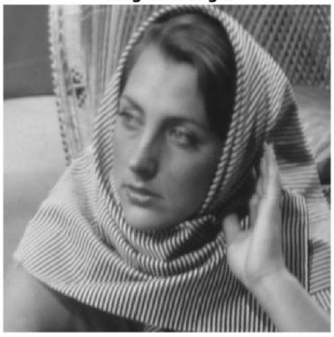
Add zero-mean Gaus sian noise with standard deviation σ = 5

CODE:

```
% Load the images
barbara = imread('barbara256.png');
kodak = imread('kodak24.png');

% Convert the images to double for adding noise
barbara = im2double(barbara);
kodak = im2double(kodak);

% Add zero-mean Gaussian noise with standard deviation σ = 5
sigma = 5 / 255;  % Standard deviation should be normalized for imnoise
barbara_noisy = imnoise(barbara, 'gaussian', 0, sigma^2);
kodak_noisy = imnoise(kodak, 'gaussian', 0, sigma^2);

% Display the noisy images
figure;
subplot(121),imshow(barbara), title('Original Image');
subplot(122),imshow(barbara_noisy), title('After adding Gaussian Noise (σ =
5)');

figure;
subplot(121),imshow(kodak), title('Original Image');
subplot(122),imshow(kodak_noisy), title('After adding Gaussian Noise (σ =
5)');
```

RESULT:

Original Image



After adding Gaussian Noise (σ = 5)

Original Image

After adding Gaussian Noise (σ = 5)

PART B: Implement a bilateral filter and show the outputs of the bilateral filter on both images for the following parameter configurations: ($\sigma s = 2, \sigma r = 2$);($\sigma s = 0.1$, $\sigma r = 0.1$);($\sigma s = 3, \sigma r = 15$)

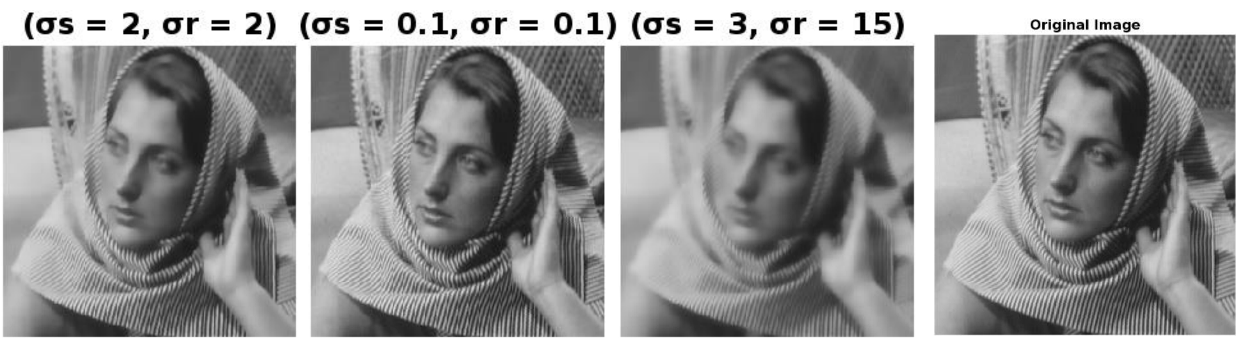
CODE:FUNCTION `mybilateralfilter`:

```
function B = mybilateralfilter(img, sigma_s, sigma_r)

    % Normalize image if not already in range [0, 1]
    if max(img(:)) > 1
        img = im2double(img);
    end

    % Define the window size based on sigma_s
    % Typically, width is set to ~3 * sigma_s to capture most of the filters
influence
    width = ceil(3 * sigma_s);

    % Precompute Gaussian spatial weights
    [X, Y] = meshgrid(-width:width, -width:width);
    G = exp(-(X.^2 + Y.^2) / (2 * sigma_s^2));

    % Get image dimensions
    [rows, cols] = size(img);
    B = zeros(rows, cols);

    % Apply the bilateral filter
    for i = 1:rows
        for j = 1:cols
            % Extract local region
            iMin = max(i - width, 1);
            iMax = min(i + width, rows);
            jMin = max(j - width, 1);
            jMax = min(j + width, cols);
            I = img(iMin:iMax, jMin:jMax);

            % Compute range weights (difference in intensity)
            H = exp(-(I - img(i, j)).^2 / (2 * sigma_r^2));

            % Combine spatial and range weights
            F = H .* G((iMin:iMax) - i + width + 1, (jMin:jMax) - j + width +
1);

            % Calculate the output pixel value
            B(i, j) = sum(F(:) .* I(:)) / sum(F(:));
        end
    end
end
```

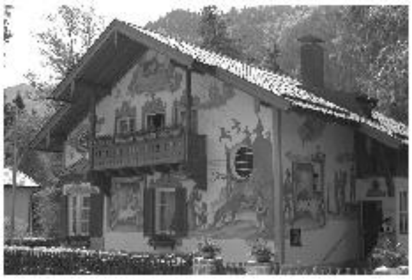
CODE :

```
% Load the images
barbara = imread('barbara256.png');
kodak = imread('kodak24.png');

% Convert the images to double for adding noise
barbara = im2double(barbara);
kodak = im2double(kodak);

% Define the bilateral filter parameters
sigma_s1 = 2; sigma_r1 = 2/255;  % (σs = 2, σr = 2)
sigma_s2 = 0.1; sigma_r2 = 0.1/255;  % (σs = 0.1, σr = 0.1)
sigma_s3 = 3; sigma_r3 = 15/255;  % (σs = 3, σr = 15)

%without noise
% Apply bilateral filter to the noisy images with different parameters
barbara_filtered1 = mybilateralfilter(barbara, sigma_r1, sigma_s1);
barbara_filtered2 = mybilateralfilter(barbara, sigma_r2, sigma_s2);
barbara_filtered3 = mybilateralfilter(barbara, sigma_r3, sigma_s3);

kodak_filtered1 = mybilateralfilter(kodak, sigma_r1, sigma_s1);
kodak_filtered2 = mybilateralfilter(kodak, sigma_r2, sigma_s2);
kodak_filtered3 = mybilateralfilter(kodak, sigma_r3, sigma_s3);

% Display the filtered results for Barbara
figure;
subplot(1, 3, 1), imshow(barbara_filtered1), title('(σs = 2, σr = 2)');
subplot(1, 3, 2), imshow(barbara_filtered2), title('(σs = 0.1, σr = 0.1)');
subplot(1, 3, 3), imshow(barbara_filtered3), title('(σs = 3, σr = 15)');

% Display the filtered results for Kodak
figure;
subplot(1, 3, 1), imshow(kodak_filtered1), title('(σs = 2, σr = 2)');
subplot(1, 3, 2), imshow(kodak_filtered2), title('(σs = 0.1, σr = 0.1)');
subplot(1, 3, 3), imshow(kodak_filtered3), title('(σs = 3, σr = 15)');
```

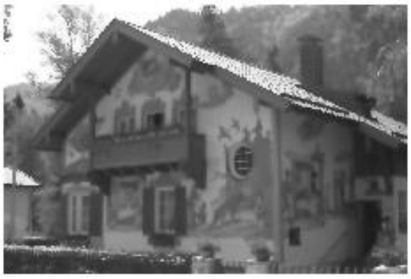
RESULT:



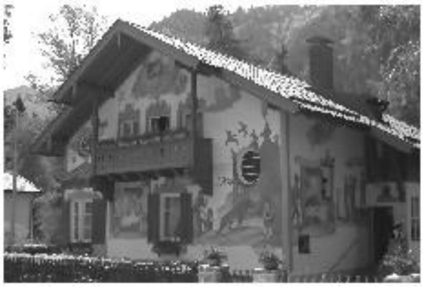
($\sigma s = 0.1, \sigma r = 0.1$)



($\sigma s = 3, \sigma r = 15$)



($\sigma s = 2, \sigma r = 2$)



Original Image

COMMENTS AND OBSERVATION:

Configuration 1:σs=2, σr=2

- Comments: The image processed with this configuration shows the best balance between noise reduction and detail preservation.

Configuration 2: σs=0.1, σr=0.1

- Comments: The filter with these parameters is too conservative, leading to minimal noise reduction. The image retains much of the noise, and the details appear slightly blurred due to the lack of sufficient spatial smoothing. This configuration does not effectively address the noise issue while preserving image quality.

Configuration 3: σs=3, σr=15

- Comments: While this configuration provides strong noise reduction, it causes noticeable smoothing of the image, which results in a loss of fine details.

Summary

For both images, Configuration 1 (σs=2,σr=2) appears to be the most effective, offering a good trade-off between noise reduction and detail preservation. Configurations 2 and 3 show extremes in filtering performance—Configuration 2 is too mild, resulting in insufficient noise reduction, while Configuration 3 provides strong noise reduction at the cost of significant detail loss.

PART C: Repeat when the image is corrupted with zero-mean Gaussian noise of σ = 10 (with the same bilaterial filter parameters)

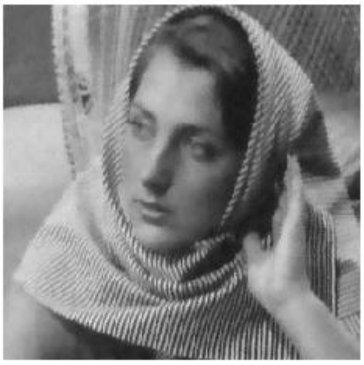
CODE:

```
% Load the images
barbara = imread('barbara256.png');
kodak = imread('kodak24.png');

% Convert the images to double for adding noise
barbara = im2double(barbara);
kodak = im2double(kodak);

% Define the bilateral filter parameters
sigma_s1 = 2; sigma_r1 = 2/255;  % (σs = 2, σr = 2)
sigma_s2 = 0.1; sigma_r2 = 0.1/255;  % (σs = 0.1, σr = 0.1)
sigma_s3 = 3; sigma_r3 = 15/255;  % (σs = 3, σr = 15)

%after adding gaussian noise where sigma=10
% Add zero-mean Gaussian noise with standard deviation σ = 10
sigma = 10 / 255;  % Standard deviation should be normalized for imnoise
barbara_noisy = imnoise(barbara, 'gaussian', 0, sigma^2);
kodak_noisy = imnoise(kodak, 'gaussian', 0, sigma^2);


% Apply bilateral filter to the noisy images with different parameters
barbara_filtered1 = mybilateralfilter(barbara_noisy, sigma_r1, sigma_s1);
barbara_filtered2 = mybilateralfilter(barbara_noisy, sigma_r2, sigma_s2);
barbara_filtered3 = mybilateralfilter(barbara_noisy, sigma_r3, sigma_s3);

kodak_filtered1 = mybilateralfilter(kodak_noisy, sigma_r1, sigma_s1);
kodak_filtered2 = mybilateralfilter(kodak_noisy, sigma_r2, sigma_s2);
kodak_filtered3 = mybilateralfilter(kodak_noisy, sigma_r3, sigma_s3);

% Display the filtered results for Barbara
figure;
subplot(1, 3, 1), imshow(barbara_filtered1), title('(σs = 2, σr = 2)');
subplot(1, 3, 2), imshow(barbara_filtered2), title('(σs = 0.1, σr = 0.1)');
subplot(1, 3, 3), imshow(barbara_filtered3), title('(σs = 3, σr = 15)');

% Display the filtered results for Kodak
figure;
subplot(1, 3, 1), imshow(kodak_filtered1), title('(σs = 2, σr = 2)');
subplot(1, 3, 2), imshow(kodak_filtered2), title('(σs = 0.1, σr = 0.1)');
subplot(1, 3, 3), imshow(kodak_filtered3), title('(σs = 3, σr = 15)');
```

RESULT:

$(\sigma s = 0.1, \sigma r = 0.1)$

$(\sigma s = 2, \sigma r = 2)$



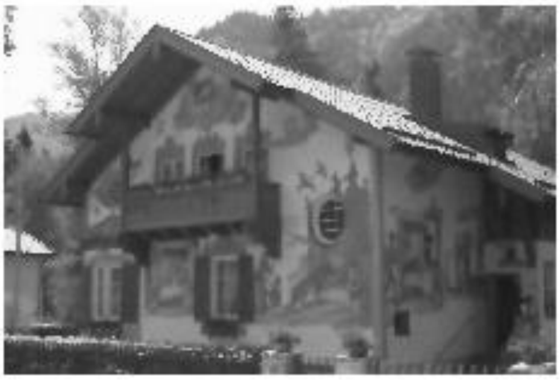
$(\sigma s = 3, \sigma r = 15)$

Original Image

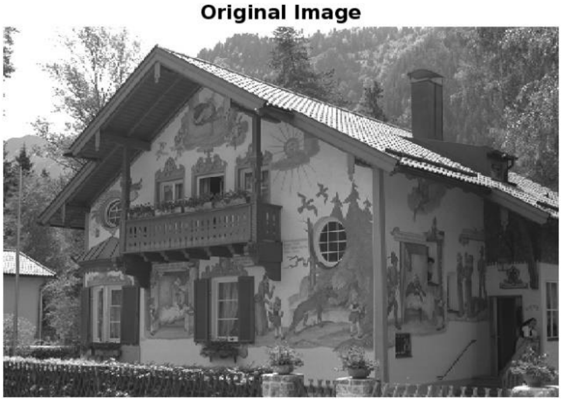
$(\sigma s = 0.1, \sigma r = 0.1)$

$(\sigma s = 2, \sigma r = 2)$

$(\sigma s = 3, \sigma r = 15)$



Original Image



COMMENTS AND OBSERVATION:

Configuration 1: $\sigma s=2,\sigma r=2$

- **Comments**: This configuration still performs well, reducing noise effectively while maintaining reasonable image detail. The balance between noise reduction and detail preservation is maintained, resulting in a clear image with minimized noise.

Configuration 2: $\sigma s=0.1,\sigma r=0.1$

- **Comments**: Despite the increased noise level, this configuration provides a better appearance than in the previous part. While the image still appears noisy, the bilateral filter has managed to reduce some of the noise compared to the noisy original. However, the results are still not as clear as those from Configuration 1, and the image retains a somewhat noisy look.

Configuration 3: $\sigma s=3,\sigma r=15$

- **Comments**: This configuration results in a noticeably blurred image. While it does reduce noise significantly, the high σr value leads to excessive smoothing, which removes too much detail. The image appears overly soft and lacks the sharpness of the other configurations, making it less suitable if detail preservation is a priority..

Summary

For both images and both noise levels, Configuration 1 ($\sigma s=2,\sigma r=2$) consistently provides the best balance between noise reduction and detail preservation. Configuration 2 shows some improvement over the noisy original images but still does not match the performance of Configuration 1. Configuration 3, while effective at noise reduction, introduces excessive blurring and detail loss, especially at higher noise levels.