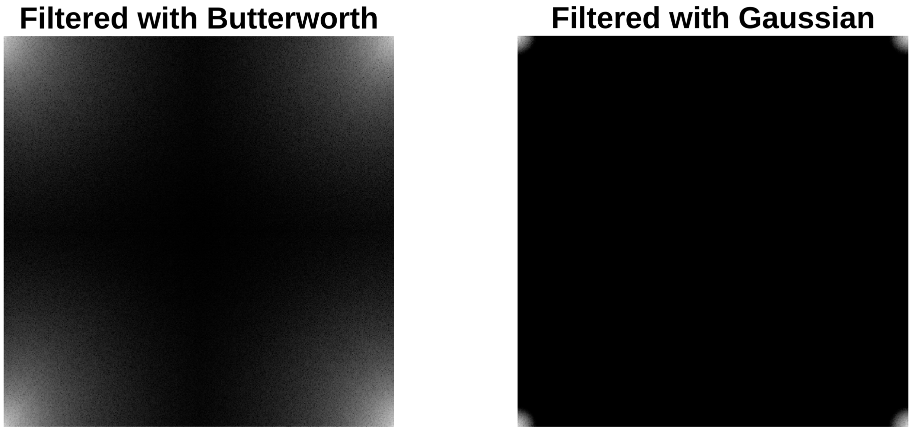
```
% Task 4: Fourier Transform and Filters
clc; clear; close all;

% Read the image
img = imread('image.jpg');
gray_img = rgb2gray(img);

% Apply Fourier Transform
f_transform = fft2(double(gray_img));
f_transform_shifted = fftshift(f_transform);

% Create Butterworth filter
function filter = butterworth_filter(shape, cutoff, order)
    [rows, cols] = size(shape);
    crow = round(rows / 2);
    ccol = round(cols / 2);
    [x, y] = meshgrid(1:cols, 1:rows);
    radius = sqrt((x - ccol).^2 + (y - crow).^2);
    filter = 1 ./ (1 + (radius ./ cutoff).^(2 * order));
end

% Create Gaussian filter
function filter = gaussian_filter(shape, sigma)
    [rows, cols] = size(shape);
    crow = round(rows / 2);
    ccol = round(cols / 2);
    [x, y] = meshgrid(1:cols, 1:rows);
    filter = exp(-((x - ccol).^2 + (y - crow).^2) / (2 * sigma^2));
end

% Apply Butterworth filter
butter_filter = butterworth_filter(gray_img, 50, 2);
filtered_butter = ifftshift(f_transform_shifted .* butter_filter);

% Apply Gaussian filter
gauss_filter = gaussian_filter(gray_img, 10);
filtered_gauss = ifftshift(f_transform_shifted .* gauss_filter);

% Display results
figure;
subplot(1, 2, 1), imshow(log(1 + abs(filtered_butter)), []), title('Filtered with Butterworth');
subplot(1, 2, 2), imshow(log(1 + abs(filtered_gauss)), []), title('Filtered with Gaussian');
```



Filtered with Butterworth

Filtered with Gaussian