```
% Implement both the Floyd-Steinberg and Jarvis-Judice-Ninke dithering
algorithms

% Task 2: Dithering Algorithms
clc; clear; close all;

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

```


% Floyd-Steinberg Dithering
function dithered = floyd_steinberg_dither(image)
    [h, w] = size(image);
    dithered = zeros(h, w);
    for y = 1:h
        for x = 1:w
            old_pixel = image(y, x);
            new_pixel = 255 * (old_pixel > 127);
            dithered(y, x) = new_pixel;
            quant_error = old_pixel - new_pixel;

            % Distribute error to neighboring pixels
            if x < w
                image(y, x + 1) = image(y, x + 1) + quant_error * 7 / 16; %
Right
            end
            if y < h
                if x > 1
                    image(y + 1, x - 1) = image(y + 1, x - 1) + quant_error
* 3 / 16; % Bottom left
                end
                image(y + 1, x) = image(y + 1, x) + quant_error * 5 / 16; %
Bottom
                if x < w
                    image(y + 1, x + 1) = image(y + 1, x + 1) + quant_error
* 1 / 16; % Bottom right
                end
            end
        end
    end
end
```

```


% Jarvis-Judice-Ninke Dithering
function dithered = jarvis_judice_ninke_dither(image)
    [h, w] = size(image);
    dithered = zeros(h, w);
    for y = 1:h
        for x = 1:w
            old_pixel = image(y, x);
            new_pixel = 255 * (old_pixel > 127);
            dithered(y, x) = new_pixel;
            quant_error = old_pixel - new_pixel;

            % Distribute error to neighboring pixels with boundary checks
            if x < w % Right pixel
                image(y, x + 1) = image(y, x + 1) + quant_error * 7 / 48; %
Right
            end
            if y < h % Below row
                if x > 1 % Bottom left pixel
                    image(y + 1, x - 1) = image(y + 1, x - 1) + quant_error
* 5 / 48;
                end
                image(y + 1, x) = image(y + 1, x) + quant_error * 7 / 48; %
Bottom pixel
                if x < w % Bottom right pixel
                    image(y + 1, x + 1) = image(y + 1, x + 1) + quant_error
* 5 / 48;
                end
            end
            if y < h - 1 % Two rows below
                if x > 1 % Two bottom left pixel
                    image(y + 2, x - 1) = image(y + 2, x - 1) + quant_error
* 3 / 48;
                end
                if x < w % Two bottom pixel
                    image(y + 2, x) = image(y + 2, x) + quant_error * 7 /
48;
                end
                if x < w - 1 % Two bottom right pixel
                    image(y + 2, x + 1) = image(y + 2, x + 1) + quant_error
* 5 / 48;
                end
            end
        end
    end
end

% Apply Dithering
floyd_dithered = floyd_steinberg_dither(gray_img);
jarvis_dithered = jarvis_judice_ninke_dither(gray_img);

% Display results
figure;
```

```
subplot(1, 2, 1), imshow(uint8(floyd_dithered)), title('Floyd-Steinberg
Dithering');
subplot(1, 2, 2), imshow(uint8(jarvis_dithered)), title('Jarvis-Judice-Ninke
Dithering');
```