Kuwahara Filter:

The filter works by first computing the variances of the four neighborhoods around *p*, and then the output is the mean of the neighborhood with the lowest variance. It should do a smoothing effect to the filtered image and removes noise.

```
function output = kuwahara_filter(input_image)
    % Convert the image to double precision for computations
    cd = double(input_image);

    % Step 1: Apply the mean and variance filters
    cdm = imfilter(cd, ones(3) / 9, 'symmetric');  % Mean filter
    cd2f = imfilter(cd.^2, ones(3) / 9, 'symmetric');  % Squared values
filter
    cdv = cd2f - cdm.^2;  % Variance calculation

    % Initialize the output image
    [rows, cols] = size(cd);
    output = zeros(size(cd));

    % Step 2: Loop over every pixel except the borders
    for i = 2:rows-1
        for j = 2:cols-1
            % Define the 2x2 neighborhood (corners)
            vars = [cdv(i-1, j-1), cdv(i-1, j+1), cdv(i+1, j-1), cdv(i+1,
j+1)];
            means = [cdm(i-1, j-1), cdm(i-1, j+1), cdm(i+1, j-1), cdm(i+1,
j+1)];

            % Step 3: Find the index of the minimum variance
            [~, idx] = min(vars);

            % Step 4: Assign the mean corresponding to the lowest variance
to the output
            output(i, j) = means(idx);
        end
    end

    % Convert the output back to the original data type if necessary
    output = uint8(output);
end
```

```
noisy_image = imread('Digital Image Processing/Assignment 2/charlie.png');
kuwahara_filtered_image = kuwahara_filter(noisy_image);
imshow(noisy_image);
title('Gray Input Image');
```


```
imshow(kuwahara_filtered_image);
title('Kuwahara Filtered Image');
```