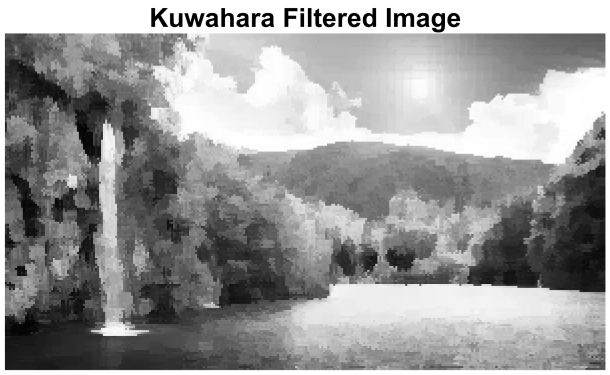
Question 3: Kuwahara Filter in MATLAB

Explanation:

The **Kuwahara filter** is a non-linear smoothing filter that preserves edges. It divides the image into overlapping regions and computes the variance for each region, then replaces the central pixel with the mean value of the region with the smallest variance.

```
% Load the image
image = imread('Nature.jpeg');
gray_image = rgb2gray(image);

% Apply Kuwahara filter
window_size = 5; % Size of the neighborhood
kuwahara_filtered = kuwahara_filter(gray_image, window_size);
figure, imshow(uint8(kuwahara_filtered)), title('Kuwahara Filtered Image');
```




```

```

Function for question-3

```
% Kuwahara filter function
```

```
function output = kuwahara_filter(img, window_size)
    [rows, cols] = size(img);
    output = zeros(rows, cols);
    offset = floor(window_size / 2); % Half-window size for padding

    % Pad the image symmetrically to avoid indexing issues at the borders
    padded_img = padarray(img, [offset, offset], 'symmetric');

    % Loop over each pixel in the original image
    for i = 1:rows
```

```
        for j = 1:cols
            % Extract four subregions around the current pixel
            regions = [
                mean2(padded_img(i:i+offset, j:j+offset)), ...
                mean2(padded_img(i:i+offset, j+offset+1:j+2*offset)), ...
                mean2(padded_img(i+offset+1:i+2*offset, j:j+offset)), ...
                mean2(padded_img(i+offset+1:i+2*offset, j+offset+1:j+2*offset))
            ];

            % Calculate the variance of each subregion
            variances = [
                var(double(padded_img(i:i+offset, j:j+offset)), 0, 'all'), ...
                var(double(padded_img(i:i+offset, j+offset+1:j+2*offset)), 0,
'all'), ...
                var(double(padded_img(i+offset+1:i+2*offset, j:j+offset)), 0,
'all'), ...
                var(double(padded_img(i+offset+1:i+2*offset,
j+offset+1:j+2*offset)), 0, 'all')
            ];

            % Find the subregion with the minimum variance
            [~, min_index] = min(variances);

            % Set the output pixel to the mean of the region with the minimum
variance
            output(i, j) = regions(min_index);
        end
    end
end
```