```
% This function reduces a 24-bit RGB image to the specified bit depth.

% Inputs:
% - image_filename: The path to the source image (must be PNG format)
% - rgb_bit_depth: A 1x3 array specicfying the bit depth of R, G, and B

% Outputs:
% - output_image: The array representing the reduced image.
% This function also writes the output image to the disk in PNG format.
function output_image = uniformQuant(image_filename, rgb_bit_depth)

    % Read in the image
    image = imread(image_filename, 'png');

    [cols, rows] = size(image, 1:2);

    % Calculate the size of the range of colors in each R, G, and B
    % channel that will be reduced to a single color for each bit depth
    partition_size = 256 ./ 2.^uint8(rgb_bit_depth);

    fprintf("Quantizing...\n")
    tic();

    % Iterate over each pixel in the image
    for c = 1:cols
        for r = 1:rows
            % Transform the RGB data into a 1x3 array so it's easier to
            % work with
            rgb = reshape(image(c,r,:),  [1, 3]);

            % Reduce the pixel's color.
            % Divide by the partition size, truncate, then re-multiply by
            % that same partition size
            rgb = floor(rgb ./ partition_size) .* partition_size;

            % Transform the data back to its original configuration
            image(c,r,:) = reshape(rgb, [1, 1, 3]);
        end
    end
    fprintf("Finished after %.2f seconds\n", toc())

    % Write out the image to disk
    output_filename = sprintf("%s_%d_%d_%d.png", image_filename,
 rgb_bit_depth);
    imwrite(image, output_filename)

    % Return
    output_image = image;
end
```

Published with MATLAB® R2022a