

1. 1) Getting familiar with image manipulation in Matlab (or your favorite language) (10/10)

Write a program/function that will:

- (a) Read and display an image

```
Img = imread('im1.jpeg');
```

```
imshow('im1.jpeg');
```



```
if size(Img, 3) == 3
```

```
    grayImage = rgb2gray(Img);
```

```
else
```

```
    grayImage = Img;
```

```
end
```

```
imshow(grayImage);
```

```
title('Grayscale Image');
```



b) Calculate the size (total number of pixels) of the image

```
Imgsize = numel(grayImage);
```

```
fprintf(' Image Size (Total Number of Pixels): %d\n', Imgsize);
```

```
-----  
Image Size (Total Number of Pixels): 1500000  
|
```

Or

```
Imgsize = size(grayImage);
```

```
Imgsize
```

```
Imgsize =
```

```
1500      1000
```

```
1500 * 1000 = 1500000
```

c) `maxPixelValue = max(grayImage(:));`

```
fprintf(' Maximum Pixel Value: %d\n', maxPixelValue)
```

```
Maximum Pixel Value: 254
```

```
d) meanPixelValue = mean(grayImage(:));  
fprintf(' Mean Pixel Value: %.2f\n', meanPixelValue);
```

```
Mean Pixel Value: 126.89
```

e)

```
Img = im2double(imread('im1.jpeg'));  
Img = Img(:,:,1);
```

```
% Performing thresholding
```

```
Img_Threshold = Img;  
Img_Threshold(Img > 126.89) = 1;  
Img_Threshold(Img < 126.89) = 0;
```

```
% Displaying the threshold image
```

```
figure();  
subplot(1, 2, 1);  
imshow(Img, []);  
title('Original Image');  
subplot(1, 2, 2);  
imshow(Img_Threshold); % Corrected this line  
title('Thresholded Image');
```



2)

a) % Read the input image

```
Img = imread('im1.jpeg');
```

% Create a subplot layout

```
figure;
```

% Display the original grayscale image

```
subplot(231);
```

```
Img1 = rgb2gray(Img);
```

```
imshow(Img1)
```

```
title('Original grayscale image');
```

```
% Downsample the image to 256x256 pixels
```

```
Img2 = Img1(1:2:end, 1:2:end);
```

```
subplot(232);
```

```
imshow(Img2)
```

```
title('Sample image (256x256)');
```

```
% Downsample the image to 128x128 pixels
```

```
Img3 = Img1(1:4:end, 1:4:end);
```

```
subplot(233);
```

```
imshow(Img3)
```

```
title('Sample image (128x128)');
```

```
% Downsample the image to 64x64 pixels
```

```
Img4 = Img1(1:8:end, 1:8:end);
```

```
subplot(234);
```

```
imshow(Img4)
```

```
title('Sample image (64x64)');
```

```
% Downsample the image to 32x32 pixels
```

```
Img5 = Img1(1:16:end, 1:16:end);
```

```
subplot(235);
```

```
imshow(Img5)
```

```
title('Sample image (32x32)');
```

```
% Downsample the image to 16x16 pixels
```

```
Img6 = Img1(1:32:end, 1:32:end);
```

```
subplot(236);
```

```
imshow(Img6)
```

```
title('Sample image (16x16)');
```



OR

(I tried in other way possible not sure if its correct prof)

```
Img = imread('im1.jpeg');
```

```
% Create a subplot layout
```

```
figure;
```

```
% Define the downsampling factors
```

```
factors = [1, 2, 4, 8, 16, 32];
```

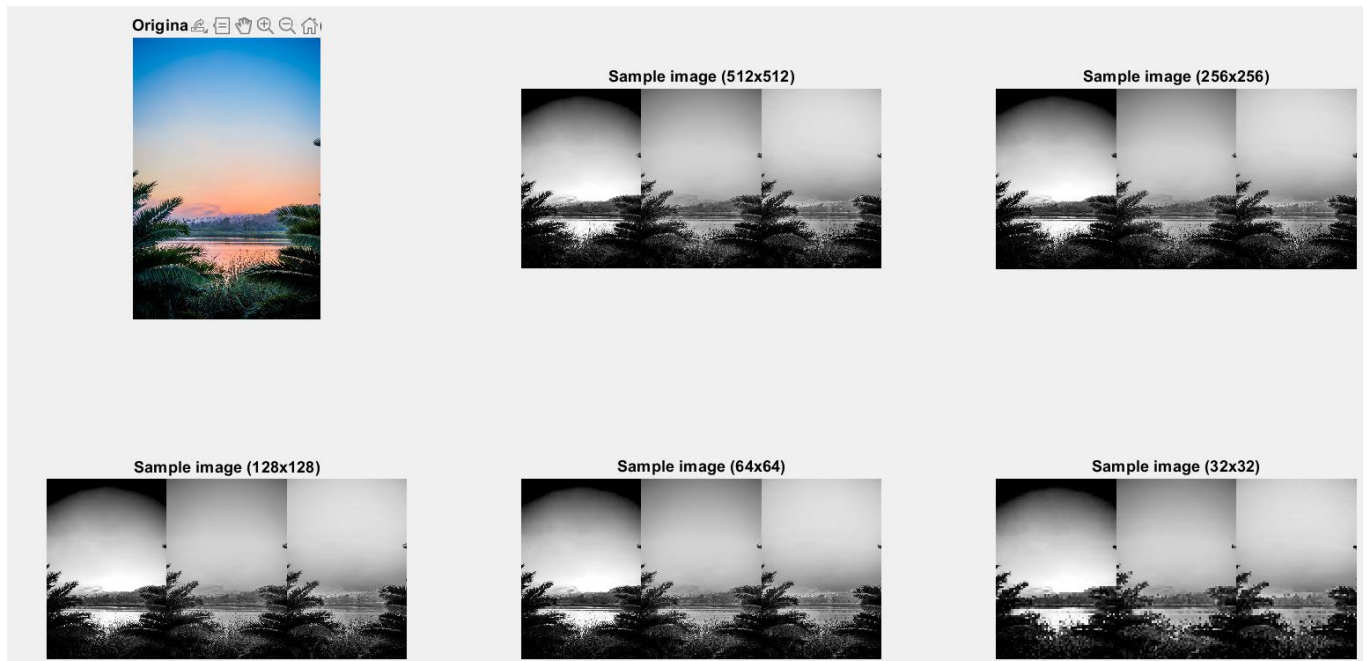
```
% Display the original image
```

```
subplot(2, 3, 1);
```

```
imshow(Img);
```

```
title('Original image (512x512)');
```

```
% Loop through downsampling factors and display the corresponding images
for i = 1:length(factors)
    downsampledImg = Img(1:factors(i):end, 1:factors(i):end);
    subplot(2, 3, i + 1);
    imshow(downsampledImg);
    title(['Sample image (' num2str(512/factors(i)) 'x' num2str(512/factors(i)) ')']);
end
```



3)

```
pic = grayImage;
pic1 = reduceGrayLevel(pic, 128);
pic2 = reduceGrayLevel(pic, 64);
pic3 = reduceGrayLevel(pic, 32);
pic4 = reduceGrayLevel(pic, 16);
pic5 = reduceGrayLevel(pic, 8);
pic6 = reduceGrayLevel(pic, 4);
pic7 = reduceGrayLevel(pic, 2);

subplot(2, 4, 1), imshow(pic, [0, 255]), title('Original');
```

```
subplot(2, 4, 2), imshow(pic1, [0, 255]), title('Level 128');  
subplot(2, 4, 3), imshow(pic2, [0, 255]), title('Level 64');  
subplot(2, 4, 4), imshow(pic3, [0, 255]), title('Level 32');  
subplot(2, 4, 5), imshow(pic4, [0, 255]), title('Level 16');  
subplot(2, 4, 6), imshow(pic5, [0, 255]), title('Level 8');  
subplot(2, 4, 7), imshow(pic6, [0, 255]), title('Level 4');  
subplot(2, 4, 8), imshow(pic7, [0, 255]), title('Level 2');
```

```
function quantizedPic = reduceGrayLevel(pic, level)  
  
    num = 256 / level;  
  
    quantizedPic = uint8(zeros(size(pic)));  
  
    for r = 1:size(pic, 1)  
        for c = 1:size(pic, 2)  
            quantizedPic(r, c) = floor(double(pic(r, c)) / num) * (255 / (level - 1));  
        end  
    end  
end
```




Design your program such that the desired number of gray levels does not have to be a power of 2.

=

```

pic = grayImage;
pic1 = reduceGrayLevel(pic, 129);
pic2 = reduceGrayLevel(pic, 65);
pic3 = reduceGrayLevel(pic, 34);
pic4 = reduceGrayLevel(pic, 17);
pic5 = reduceGrayLevel(pic, 9);
pic6 = reduceGrayLevel(pic, 5);
pic7 = reduceGrayLevel(pic, 3);

subplot(2, 4, 1), imshow(pic, [0, 255]), title('Original');
subplot(2, 4, 2), imshow(pic1, [0, 255]), title('Level 129');
subplot(2, 4, 3), imshow(pic2, [0, 255]), title('Level 65');
subplot(2, 4, 4), imshow(pic3, [0, 255]), title('Level 33');
subplot(2, 4, 5), imshow(pic4, [0, 255]), title('Level 17');
subplot(2, 4, 6), imshow(pic5, [0, 255]), title('Level 9');
subplot(2, 4, 7), imshow(pic6, [0, 255]), title('Level 5');

```

```
subplot(2, 4, 8), imshow(pic7, [0, 255]), title('Level 3');
```

```
function quantizedPic = reduceGrayLevel(pic, level)
```

```
    num = 256 / level;
```

```
    quantizedPic = uint8(zeros(size(pic)));
```

```
    for r = 1:size(pic, 1)
```

```
        for c = 1:size(pic, 2)
```

```
            quantizedPic(r, c) = floor(double(pic(r, c)) / num) * (255 / (level - 1));
```

```
        end
```

```
    end
```

```
end
```

