IVP Lab Assignment – 1

1. Write a program that calculates the average intensity value of the pixels in the image Moon.bmp and then thresholds this image based on this intensity. Use this new function from the command line on the image moon.tiff and it should have the effect as shown in fig 1. Compare the execution times of program using loops vs vectorization on various image sizes (16X16,32X32,64X64,128X128,256X256) and plot the graph

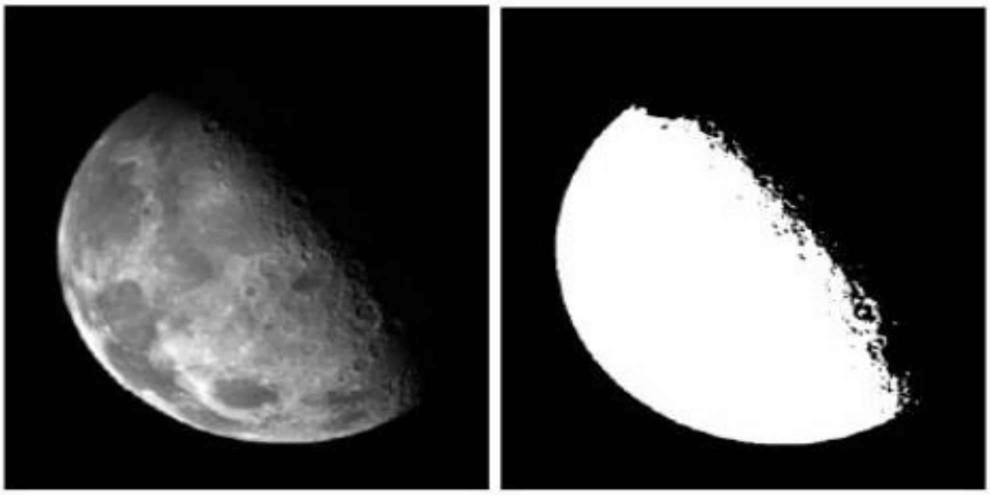
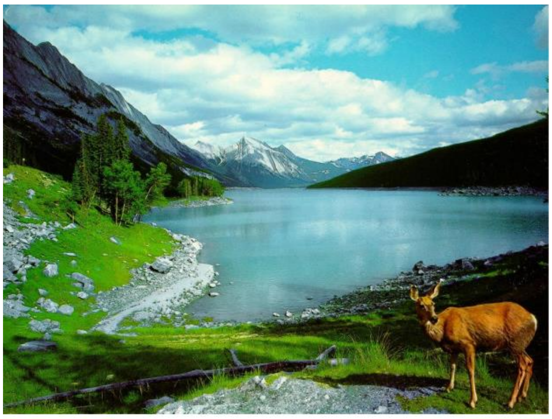
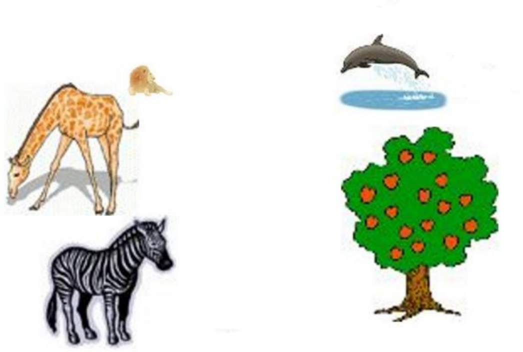
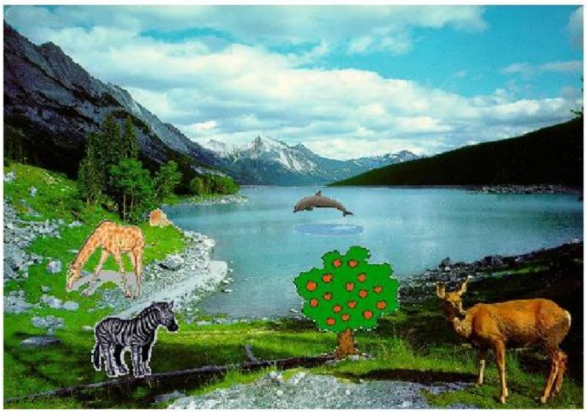


Fig1: The Moon

2. Using basic image operations combine the scene and object image to get the compound image as shown below.







3. For the attached image 'spine.tiff', enhance it using (a) The log transformation and (b) A power-law transformation. In (a) the only free parameter is c, but in (b) there are two parameters, c and r for which values have to be selected. By experimentation, obtain the best visual enhancement possible with the methods in (a) and (b). Once (according to your judgment) you have the best visual result for each transformation, explain the reasons for the major differences between them.

4. Write a function, generateHistogram, which generates the histogram of an image. The function should take an image data array (with pixel values in the range 0 – 255) as its only parameter and return an array containing the histogram of the image. The histogram can be displayed using the built-in plotting function. Use this new function to generate and display histograms for the following images histogram1.jpg, histogram2.jpg and histogram3.jpg.

5. Implement a histogram equalization function. Use it to enhance the above images. Compare the output of your implementation with any built-in library function.

6. Binarization or Thresholding typically is a crucial step towards machine-based understanding content of document images. For the sample document images (leaf-1.jpg and leaf-2.jpg) are attached separately. Write a function BinarizeImage which outputs a binarized version of input image im. Note that the input image may be RGB.

7. The function domIntensity(im, k) takes an image im and an integer k and returns a list of k most frequently occurring graylevel intensity in that image.
1. Implement the function. The code should work for grayscale images.
2. Implement a display function which takes the image im as input, the list returned by domIntensity(im, k) and displays the image, the most dominant graylevel intensity and the palette of k dominant intensities (see Figure 1).

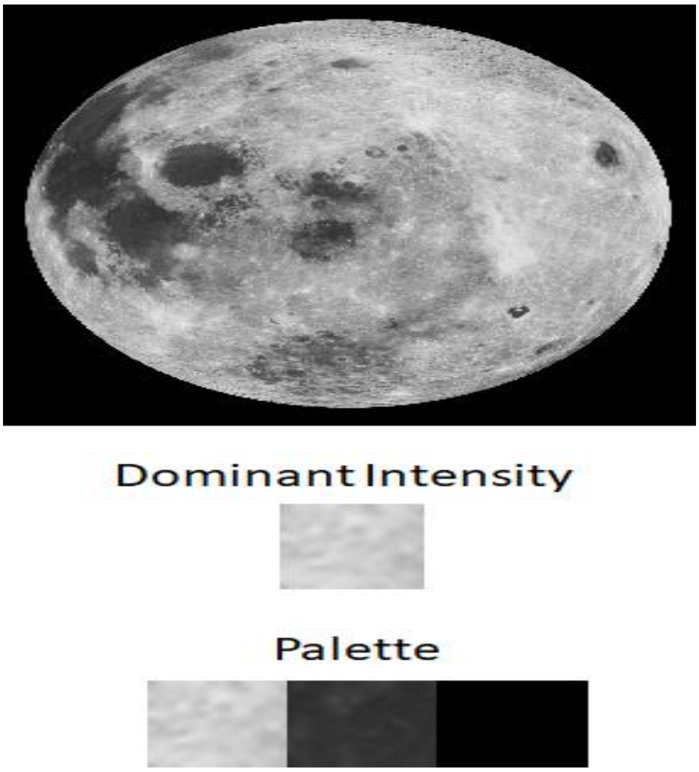


Figure 1: An example of input image and output for question 1.