1) Load the an image from your choice, from now on let us denote this image by IMG
2) Determine the pixel resolution of IMG
3) Show the RGB values of the pixel IMG(10, 15)
4) Calculate the brightness of IMG
5) Calculate the contrast of IMG
6) Show the each of the RGB channels separately
7) Convert IMG to gray-level using the average method
8) Convert IMG to gray-level using the weighted formula (using loops, then without loop and finally using python instruction), let us denote the intensity IMG by IMG2
9) Determine the min / max value of IMG and IMG2
10) Calculate the pixel resolution of IMG and IMG2 in terms of MegaPixel
11) Write a small function to calculate the size of IMG
12) Extract SUB_IMG the sub-image from IMG by cropping IMG from line 50 to 70, and from column 20 to 90. Display the two images
13) Quantize IMG2 to 128, 64, 32 levels then display it.
14) Given an RGB image convert it to the different color spaces HSV,CMYK and YCbCr.
15) Binarize IMG2 according to 10, 50, 125, 25 as threshold
16) Calculate the complementary image of IMG2 and its binarized version
17) Add (and subtract) the following values from IMG2: 10, 30,60, 200.
18) To increase / decrease image brightness, multiply IMG2 by 1.7 and 0.6, respectively.
19) Devise the image on 2 and 0.5, respectively
20) Load the another image from your choice (IMG3)
21) Calculate the min / max (IMG2, IMG3)
22) Combine IMG2 and IMG3 linearly using (0.1,0.9), (0.5,0.5),(0.9,0.1) as weights
23) Create two images containing squares (the intersection of squares is not zero)
24) Perform the following logical operations on these two image square: AND, OR, XOR, XNOR
25) Generate two other binary images containing arbitrary shapes
26) Perform the following operations on the image: erosion, dilation, opening and closing, using a 3*3 square binary image as a structural element.
27) Generate the gray-level histogram of IMG2
28) Add 70 to all the image pixels, and compare the histogram before and after this is done
29) Load a low contrast image from your choice (IMG4)
30) Apply the histogram stretching (the two formulas of course), and show the histogram before and after each process.
31) Apply Gamma correction on IMG2, where the values of Gamma are: 1.5, 4.2, 2.1
32) Apply the algorithm of histogram equalization on IMG4, and show the histogram before and after this is done (choose the suitable L_{max} based on the first histogram).
33) Load an image from your choice (IMG5)
34) Add a salt, pepper, salt-pepper, Gaussian noise to IMG5
35) Apply mean and Gaussian filters on the noisy versions of IMG5 (vary the parameters)
36) Apply the median filter with a neighborhood of 3,5 and 7, respectively.
37) Apply Sobel operator on IMG5, and show the horizontal, vertical and the final edge

38) Show the gradient magnitude image of IMG5, and threshold it using 10,50,120,190, show the resulting images.
39) Show the edge image in which you keep only the pixels with the maximum of gradient in the gradient direction.
40) Detect the edges in IMG5 using Laplacian of Gaussian.
41) Detect the edges in IMG5 using canny algorithm.