Image Processing Primer

Given a set of sample images, write executable functions for

the following operations as directed below.
For all operation, implement with your own functions.

(1) Read both the images as provided. -- 5 M
(2) Convert the color image into a gray level image and use

it in subsequent processing. -- 5M
(3) Add salt and pepper noise where a pixel is turned into dark (0) or white (255) with a probability of (p/2) for each (p to be provided as an input parameter) to 'cap.bmp' image and then perform denoising by median filtering and mean filtering. Display the original, noisy, and filtered images.
-- 20 M
(4) Plot PSNR values against p for each cases (median and mean filters)-- 10 M

(3) Compute edge pixels of image 'lego.tif' using Sobel operator. Use empirical thresholding. Display the x-gradient image, y-gradient image, gradient magnitude image, and thresholded edges. -- 20 M

(4) Partition the image 'cap.bmp' into 8x8 non-overlapping blocks and for each block compute 8x8 DCT coefficients. If any boundary block has less number of rows and columns pad them with zeroes to make them 8x8. Display the resulting block-DCT transformed image. Perform the inverse DCT
operations and get back the original image. Display the inverse DCT images along with original image. -- 20 M

(5) Implement Harris corner detection algorithm and apply on the image 'lego.tif'. Use empirical thresholding to filter the responses. -- 20 M

Bonus: Compare with the results obtained by using standard library/package functions. -- 10 M

You may implement your programs in OpenCV/MATLAB language with necessary user's interfaces and visualization of your results and input. Please provide a documentation for compiling and running the programs in a README file. The whole project should be submitted in a single tar or zip
file.