

Task 1) First read the image that provided in the question and convert it into gray scale image using “`cvtColor(resized_img, gray_image, COLOR_BGR2GRAY);`” code.  
Here is my output of gray scale image (Garden\_grayscale.jpg)



Task 2) We create a header file with name “ImageFilters.h” and then include it in task2 file.  
we have 2 function for calculate min and max filter which as a arguman we pass an image and a number for kernel size to the functions , for each of them we use 4 nested loop with time complexity  $O(n^4)$  which talked about it in course.  
at the beginning of the program its important to check if the kernel size is odd or not using this code “ `kernel_size % 2 == 0` “

a) for Lena\_corrupted min filter with kernel 3 gave us best result



b) for Astronaut\_salt\_pepper.png maybe max filter with kernel size 3 gave us best result



c) for max Garden\_grayscale filter with kernel size 5 gave us best result



Task 3 and 4) here is output of apply a 3 kernel with median filter:

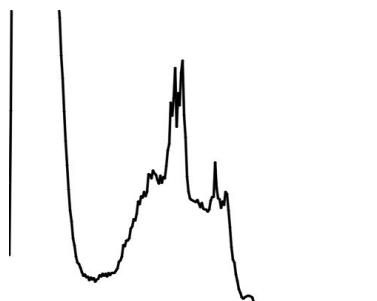


and output of apply a 3\*3 kernel with Gaussian filter:



Task 5 and 6 ) in this task we reads an image file provided as a command-line argument, computes and visualizes the histogram of the grayscale image, and performs histogram equalization to enhance image contrast. It then displays both the original and equalized images along with their respective histograms using OpenCV functions.

Here is the output of original histogram of Garden\_grayscale:



and this is Equalized Histogram:

