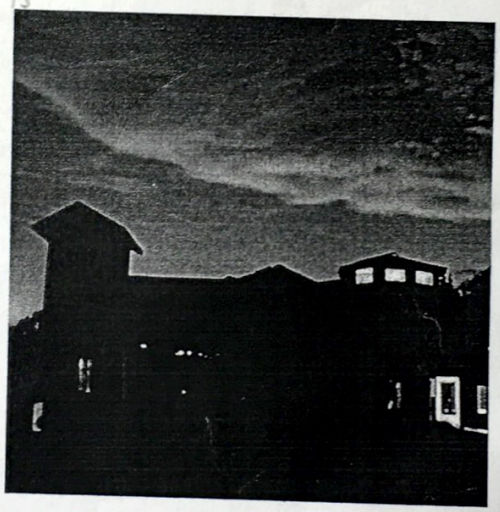
Due: Tuesday 14:00pm, Oct. 31, 2017

NOTE: For all of the homework in this course, do not use the problem-related OpenCV API (neither built-in nor library) to solve your problem.

1. Histogram processing (45%)
   (a). Perform **(Global) Histogram Equalization** on *house512.raw*. Does the result look good? Which portitions it will fail and why? The histograms of input and output images may help your reasoning. (Figure, 10%; Discussion, 5%)
   (b). Use **Histogram Match** to enhance *house512.raw*. Design the histogram specification by yourself. Show the specified histogram, the histograms of input and output images, and the resulted images. (Figure, 10%; Discussion, 5%)
   (c). Use **Local Histogram Equalization** to enhance *house512.raw*. Try at least three different mask sizes to process it. Show the output images and its histograms. (Figure, 10%; Discussion, 5%)



house512.*raw*

2. Bit Plane (10%)
   (a). Slice the bit-planes 7(MSB) to 0(LSB) from both *lenaQ_256.raw* and *baboon_256.raw*. Find and solve the questions embedded in bitplanes of *lenaQ_256.raw*, and embed your answer (you can use OpenCV draw function) into the LSB bitplane of *baboon_256.raw*, and assemble all bitplanes back to *baboon_256_LSB_changed.raw*. Show your each stage output. (Figure, 5%).
   (b). Check the MSE and PSNR between *baboon_256.raw* and *baboon_256_LSB_changed.raw*. (Discussion, 5%)
   Hint for MSE & PSNR: https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio

3. Image Averaging (20%)

(a). Add the Gaussian noise with mean=0 and standard deviation=50 on *lena512.raw* (opencv built-in function is allowed) and show the noisy image. (Figure, 5%)

(b). Repeat (a) to create 3, 5, 50, 100 different gaussian images. Use these different numbers of images to do **image averaging** respectively. Show the output images and discuss the visual difference for each case. (Figure, 10%; discussion, 5%)

4. Smooth & Sharpening (40%)

(a). Use **low-pass filter (Blurring,** example mask as below) to filter *lena512.raw* with a different mask sizes of 3x3 and 7x7 respectively. Explain how you deal with the boundary (only *replication* or *mirror* approach is allowed). Show these three output images, and compare their visual difference with original image. What happened if you apply the filter again (or more times) on the output image? (Figure,10%; Discussion,5%)

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3x3 mask

1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49

7x7 mask

(b). Repeat (a) but using **high-pass filter (Sharpening,** example mask as below) to filter *lena512.raw* with a different mask sizes of 3x3 and 7x7 respectively. (Figure,10%; Discussion,5%)

-1	-1	-1
-1	8	-1
-1	-1	-1

3x3 mask

-10	-5	-2	-1	-2	-5	-10
-5	0	3	4	3	0	-5
-2	3	6	7	6	3	-2
-1	4	7	8	7	4	-1
-2	3	6	7	6	3	-2
-5	0	3	4	3	0	-5
-10	-5	-2	-1	-2	-5	-10

7x7 mask

(c). Try to process *lena512.raw* by combining **LPF (images obtained in (a))** and **arithmetic subtraction operations** to get the sharpened image. You only need to do 3x3 case. Remember to clip the the overflow or underflow pixels. Show your output. By choosing a proper constant *c*, is the result same as the one directly using HPF in (b). (Figure, 5%; Discussion 5%)

Sharpened = Original – c × (LPF of Original), where c is a constnat