In this machine problem, the goal was to create an algorithm to provide more detail to a photograph that has little nuance between greyscale values. The main method to do this was histogram equalization, although I also implemented a linear and quadratic least-squares lighting adjustment function.

For the histogram equalization method, I simply created a histogram using a nested for loop, normalized (using the number of pixels as the divisor), and then found the cumulative distribution function to use as a transformation for each pixel in the image. The hardest part of this was realizing that my type was wrong, as the function originally returned an array of doubles, but this was solved easily enough by casting to a byte-sized integer once this was realized.

Further, I implemented the two least-squares algorithms, though the results seem a bit dark and frankly worse than when I only used regular histogram equalization, which I tried applying after to get even worse results. This process was very straightforward and took maybe half an hour for both functions. The structure of both is to create the necessary matrices (a list of the intensities pixel-by-pixel and a list of the corresponding points with a 1 added to the end for linear, quadratic just required the addition of the quadratic terms) and then use of pinv in MATLAB and some multiplication.