1. Design 10 black-and-white half-tone patterns on a 3 x 3 background. The 10 patterns should represent 10 shades of gray between black and white.

2. Write a function (xx_halftone.m) that accepts a gray level uint8 image as an argument and uses your half-tone patterns to generate and return a black-and-white logical image that is a half-tone approximation to the input image.

   \[ B = \text{xx_halftone}(A); \]

   The return image will be three times larger than the input image.

3. Write a script (proj03.m) that reads in the gray level clown image and converts it to a black and white image. The script should display both images in separate figures. It should also write the black-and-white image out to a file.

Turn in printed copies of both the function and the script as well as printed copies of the original gray level and black and white images. Turn in electronic versions of the function and script in a zip or tar archive via the submission system.

Notes:

- Recall that Octave does not show images at their true-size. Use imagej (or other image viewing program) to display and print the images at their true-size.
- I am sure the implementation of your function would be a lot faster if, instead of looping over the pixels in the image, you could figure out how to loop over the 10 shades of gray instead. If your function does this, I will give you bonus points. (I will raise the project grade of your choice to halfway between its current grade and 100%).