1. Consider the image deblurring task described in the slides and text.

   - Download the "boat" image from Waterloo's Image repository.[1] Alternatively, you may use any grayscale image of your choice. You can load the image into MATLAB's memory using the "imread" function (also, you may want to apply the function "im2double" to get an array consisting of doubles).
   - Create a blurring point spread function (PSF) using MATLAB's command "fspecial." For example, you can write

     ```
     PSF = fspecial('motion',20,45);
     ```

     The blurring effect is produced using the "imfilter" function

     ```
     J = imfilter(I,PSF,'conv', 'circ');
     ```

     where I is the original image.
   - Add some white gaussian noise to the image using MATLAB's function "imnoise," as follows:

     ```
     J = imnoise(J, 'gaussian', noise_mean, noise_var);
     ```

     Use a small value of noise variance, such as 10^{-6}.
   - To perform the deblurring, you need to employ the "deconvwnr" function. For example, if J is the array that contains the blurred image (with the noise) and PSF is the point spread function that produced the blurring, then the command

     ```
     K = deconvwnr(J, PSF, C);
     ```

     returns the deblurred image K, provided that the choice of C is reasonable. As a first attempt, select $C = 10^{-4}$. Use various values for C of your choice. Comment on the results.

2. Consider the channel equalization task described in the slides. Write the necessary code to solve the problem using MATLAB according to the following steps:

   (a) Create a signal s_n consisting of 50 equiprobable ± 1 samples. Plot the result using MATLAB's function "stem."

   (b) Create the sequence $u_n = 0.5 s_n + s_{n-1} + \eta_n$, where η_n denotes zero mean Gaussian noise with $\sigma_\eta^2 = 0.01$. Plot the result with "stem."

   (c) Find the optimal $\boldsymbol{w}_* = [w_0, w_1, w_2]^T$, solving the normal equations.

[1] `http://links.uwaterloo.ca/`.

(d) Construct the sequence of the reconstructed signal $\hat{s}_n = \text{sgn}(w_0 u_n + w_1 u_{n-1} + w_2 u_{n-2})$. Plot the result with "stem" using red color for the correctly reconstructed values (i.e., those that satisfy $s_n = \hat{s}_n$) and black color for errors.

(e) Repeat steps (b)-(d) using different noise levels for σ_η^2. Comment on the results.

3. Consider the autoregressive process estimation task described in the Example in the slides. Write the necessary code to solve the problem using MATLAB according to the following steps:

(a) Create 500 samples of the AR sequence $x_n = -a_1 x_{n-1} - a_2 x_{n-2} + \eta_n$ (initializing at zeros), where $a_1 = 0.2$, $a_2 = 0.1$, and η_n denotes zero mean Gaussian noise with $\sigma_n^2 = 0.5$.

(b) Create the sequence $y_n = x_n + v_n$, where v_n denotes zero mean Gaussian noise with $\sigma_v^2 = 1$.

(c) Implement the Kalman filtering algorithm as described in the slides, using y_n as inputs and the matrices F, H, Q, R as described in the AR example in the slides. To initialize the algorithm, you can use $\hat{\boldsymbol{x}}_{1|0} = [0, 0]^T$ and $P_{1|0} = 0.1 \cdot I_2$. Plot the predicted values $\hat{x}_n$ versus the original sequence x_n. Play with the values of the different parameters and comment on the obtained results.