Lab 2.1

(a) Write a matlab function called `dampedsine.m`. Your function should be called with a syntax like 
\[
x = \text{dampedsine}(F, B, N, Fs);
\]
The three input parameters should be the frequency of the sine wave, its bandwidth, the number of samples to produce, and the sampling frequency. The output should be a signal constructed as
\[
x[n] = e^{-n\pi B/F_s} \sin \left(\frac{n2\pi F}{F_s}\right), \quad 1 \leq n \leq N
\]
In figure 1, plot 100 samples of a damped sine wave, at 8kHz sampling rate, with a bandwidth of 150Hz and a center frequency of 900Hz, versus sample time (in seconds). Label the axes.

(b) Write a matlab function called `vowel.m`. Your function should be called with a syntax like 
\[
x = \text{vowel}(F0, F1, F2, B1, B2, A1, A2, Fs);
\]
It should call `dampedsine` to produce two damped sine waves, each of length `Fs/F0` samples, at frequencies of `F1` and `F2`, with bandwidths of `B1` and `B2`. These damped sine waves should be scaled by the amplitudes `A1` and `A2`, then added together, then repeated twenty times, in order to output twenty pitch periods of a vowel. Produce an example of the vowel `/a/` (`F1 = 900, F2 = 1200, B1 = 150, B2 = 150, A1 = 1, A2 = 0.75`) at `Fs = 8kHz`. Plot the vowel waveform (versus sample time in seconds; with labeled axes) in figure 2, and play it using `soundsc`.

(c) Use `fft`, which computes the DFT of your twenty-pitch-period waveform. Create two sub-plots in Fig. 3. In the top sub-plot, plot the positive-frequency components of the associated power spectrum, versus frequency in Hertz (from 0 to 4000), with labeled axes. In the bottom sub-plot, plot the positive-frequency components of the power level spectrum (`10 \log_{10} |X(\omega)|^2`) versus frequency in Hertz (frequencies from 0 to 4kHz), with axes labeled “Frequency (Hz)” and “Level (dB).”

(d) Repeat parts (b) and (c) in order to produce the vowel `/i/` (`F1 \approx 300Hz, F2 \approx 2000Hz`). Experiment with bandwidths and amplitudes, to find bandwidths and amplitudes that make the vowel sound as natural as possible. Plot the waveform in Fig. 4; plot the power spectrum and level spectrum in Fig. 5; use `soundsc` to listen to the waveform.