The objective of this lab is to generate samples of frequency and amplitude modulated sinusoids. We will study the effect of modulation by listening to the results converted to audio via the MATLAB command `sound`. This set of instructions is adapted from your textbook (DSP First - A Multimedia Approach), Section C.4.3.

A. Synthesize Beat Notes: An Example of Amplitude Modulation

1. Write a function named `beat_notes` that implements the sum of two sinusoidal signals of slightly different frequencies, given by the following equation:

\[ X(t) = A \cos(2\pi(f_c - f_\Delta)t) + B \cos(2\pi(f_c + f_\Delta)t) \]

The function should have the following input and output parameters:

Inputs:
- \(A\): amplitude of lower frequency cosine (real)
- \(B\): amplitude of higher frequency cosine (real)
- \(f_c\): center frequency (real and positive)
- \(f_\Delta\): frequency difference (real and positive)
- \(f_{samp}\): sampling rate (integer and positive)
- \(d_{ur}\): total time duration in seconds (real and positive)

Outputs:
- \(t\): time vector ranging from 0 to \(d_{ur}\) with a sampling rate of \(f_{samp}\)
- \(X\): vector containing samples of the sum of two sinusoids

2. An error message should be generated if any of the inputs does not meet the specified conditions.

3. The function should produce the plot of \(X\) versus \(t\) with proper labels and title. Select a small portion of the signal in order to get a clean graph (e.g. select 0.2 seconds if the total duration is 2 seconds).

4. The function should also convert the signal to audio and play it by using the MATLAB function `sound` with two input arguments \(X\) and \(f_{samp}\).
5. Call the function in the command window for the following input sets:
   
i. A=10, B=10, fc=50, fdelta=10, fsamp=8000, dur=4
   ii. A=10, B=10, fc=200, fdelta=10, fsamp=8000, dur=2
   iii. A=10, B=10, fc=2000, fdelta=10, fsamp=8000, dur=1.5

Listen to the signal and verify that the amplitude and frequency of the “envelope” is correct in each case.

B. Synthesize a Chip: An Example of Frequency Modulation

1. Write a function named \texttt{my\_chirp} that synthesizes a “chirp” signal (a sinusoid whose frequency changes linearly over time). The equations of the chirp signal and its instantaneous frequency are as follows
   \[ X(t) = A \cos(2\pi \mu t^2 + 2\pi f_0 t + \phi) \]
   \[ f_i(t) = 2\mu t + f_0 \]

The function should have the following input and output parameters:

Inputs:
- \( A \): amplitude of the sinusoid (real)
- \( f_1 \): starting frequency (real and positive)
- \( f_2 \): ending frequency (real and positive)
- \( \phi \): phase of the sinusoid (real)
- \( \text{fsamp} \): sampling rate (integer and positive)
- \( \text{dur} \): total time duration (real and positive)

Output:
- \( X \): vector containing samples of the chirp signal

2. An error message should be generated if any of the inputs does not meet the specified conditions.

3. To generate the chirp signal, first you need to calculate the slope and intercept of the instantaneous frequency (\( \mu \) and \( f_0 \)) given the values of the starting and ending frequencies (\( f_1 \) and \( f_2 \)).

4. The function should convert the signal to audio using the MATLAB function \texttt{sound}.

5. The function should also compute and plot the spectrogram of the chirp signal using the MATLAB function \texttt{spectrogram} with the following syntax:

\[
\text{win}=256; \text{noverlap}=128; \text{nfft}=256; \\
\text{spectrogram(X, win, noverlap, nfft, fsamp)}
\]
6. Call the function in the command window for the following input sets:
   i. A=0.5, f1=100, f2=2000, phi=0, fsamp=8000, dur=2
   ii. A=0.5, f1=100, f2=2000, phi=0, fsamp=8000, dur=5
   iii. A=0.5, f1=300, f2=15000, phi=0, fsamp=8000, dur=2
   iv. A=0.5, f1=300, f2=15000, phi=0, fsamp=8000, dur=5

   Listen to the signal. What comments can you make regarding the sound of the chirp? Is it linear? Use the sampling theorem (from chapter 4) to explain the behavior of the chirp signal. For each input set, include in your report the spectrogram and a brief description of what you hear and observe.

**Question**

Compute and plot the spectrogram of beat notes by adding the `spectrogram` command to your `beat_notes` function. Use the syntax and input values given in part B-5 for the `spectrogram` command. Call the `beat_notes` function in the command window for the following input set:

   A=10, B=10, fc=1000, fdelta=20, fsamp=8000, dur=2

   How many frequency components do you expect to see in the spectrogram? at what frequencies? should the frequencies vary with time or be constant for the whole duration of the signal? is the plot consistent with your expectation? why? or why not?

   Change the parameters of the spectrogram as follows:
   
   i. win=1024; noverlap=512; nfft=1024;
   ii. win=4096; noverlap=2048; nfft=4096;

   Comment on how the spectrogram changes in each case. How many and what frequency components do you observe now?