ECE 3331, Dr. Hebert, Spr 2017, programming assignment HW 10 due 04/11 at 11:59 pm

File ts9_mono is a short wav file (mono, 16 bits per sample, samplerate = 22050/sec) containing a simple recorded guitar. In this project you change the mono wav file to a stereo wav file; then you will will add a flange-effect to the stereo guitar audio file and store the result in an output stereo wav file.

To start your project, write a program to open up a mono wav file and generate an output stereo wav file having the same samplerate and bitspersample. **In stereo wav files, the audio samples for the left and right channels are interleaved; that is, a sample for the left channel is stored, then a sample for the right channel is stored, then a sample for the left channel is stored, etc etc, etc.**
(1) prompt the user for the path and filename for the input mono wav file. Open the provided file as "rb". Read the header and check if it is 16 bitspersample and a mono wav file. If the file doesn't exist, or if it has the wrong wav characteristics, prompt the user to enter a correct filename. Repeat this until a valid filename is provided.
(2) prompt the user for an output filename (filename only, no path) with no file extension (e.g. "mywav"). Create the output file (e.g. "mywav.wav") in the same folder as the input mono wav file was located; use file mode "wb".
(3) generate the output stereo wav header with the same characteristics as the input mono wav file, only stereo - not mono. Do this by copying the header from the input file to the output file and then fseeking to change the values for filesize, number of channels to 2, and data chunksize to twice the size.
(4) Malloc a short int (16-bit) vector to hold the audio data from the input mono wav file, and read the audio data from the input file into the vector. Use this vector to write stereo audio data to the output file. It is easy; you will just write each 2-byte sample into the output file twice. Close the file. Now Play the file that you created and also check its size. The stereo file should play and it should be almost twice the size of the input wav file. Once your stereo file plays correctly, go to step (5).
(5) Now you will add a flange effect to the left channel and a flange effect to the right channel. A flange effect is created by adding a delayed version of the signal to itself, and varying the amount of delay.
If the nth input audio sample and the nth output audio sample are denoted $ai[n]$ and $ao[n]$ a fixed delay version of an audio signal is is added to itself by

$$ao[n] = ai[n] + ai[n-d].$$

Since the sum $ao[n]$ might exceed the range of a short int (for example) we need to divide by 2.0

$$ao[n] = \frac{ai[n] + ai[n-d]}{2}.$$

If the sample rate is 22050 samples per second and we want an echo with a 0.002 second delay, then d= 0.002*22050 = 44.

$$ao[n] = \frac{ai[n] + ai[n-44]}{2}.$$

Here, the echo is the same volume as the direct sound. We can decrease the volume of the "echo" $ai[n-d]$ for example via

$$ao[n] = \frac{ai[n] + 0.7 * ai[n-44]}{1.0 + 0.7}$$

The flange effect is formed by a time-varying delay *d[n]*

$$ao[n] = \frac{ai[n] + 0.7 * ai[\,n - d[n]\,]}{1.0 + 0.7} \text{ where}$$

$$d[n] = \mathrm{int}\left(44\left[1 + A\cos\left(\frac{2\pi f_0 n}{f_s}\right)\right]\right)$$ and f_s is the samples per second of the audio file, f_0 is a small frequency such as 0.8 Hz, and amplitude A equals a value less than or equal to 1.0; for example A=0.9.
In this example, the output equation would be

$$ao[n] = \frac{ai[n] + 0.7 * ai\left[n - \mathrm{int}\left(44\left[1 + 0.9\cos\left(\frac{2\pi f_0 n}{f_s}\right)\right]\right)\right]}{1.0 + 0.7}.$$

To create a stereo flange effect, try using a cosine function for the left channel and a sine wav for the right channel

$$ao_L[n] = \frac{ai[n] + 0.7 * ai\left[n - \mathrm{int}\left(44\left[1 + 0.9\cos\left(\frac{2\pi 0.8 n}{f_s}\right)\right]\right)\right]}{1.0 + 0.7}$$

$$ao_R[n] = \frac{ai[n] + 0.7 * ai\left[n - \operatorname{int}\left(44\left[1 + 0.9\sin\left(\frac{2\pi 0.8n}{f_s}\right)\right]\right)\right]}{1.0 + 0.7}$$

At the start of this algorithm (n=0, 1, 2, etc) $n - \operatorname{int}\left(44\left[1 + 0.9\sin\left(\frac{2\pi 0.8n}{f_s}\right)\right]\right)$ may be less than 0, so that $ai\left[n - \operatorname{int}\left(44\left[1 + 0.9\sin\left(\frac{2\pi 0.8n}{f_s}\right)\right]\right)\right]$ may not exist (i.e. ai[-4] doesn't exist). If $n - \operatorname{int}\left(44\left[1 + 0.9\sin\left(\frac{2\pi 0.8n}{f_s}\right)\right]\right)$<0, use ai[n] in place of $ai\left[n - \operatorname{int}\left(44\left[1 + 0.9\sin\left(\frac{2\pi 0.8n}{f_s}\right)\right]\right)\right]$.

(5) Once your program is working and creating a playable wav file, you can experiment with different values of amplitude A between 0 and 1.0 and different values of f_0 between 0.2 and 1.2.