ECE 3331, Dr. Hebert, Fall 2023 Programming Assignment 09 due Saturday 10/28 at 11:59 pm

In this project, you will create a mono WAVE audio file whose audio data is a sine wave whose frequency (Hz) and duration (secs) is specified by the user.
The name of the audio file will be tone1.wav and it will be located in the directory (folder) from which your executable is run.
The sample rate will be 44100 samples per second, 16 bits per sample, mono.
You will use malloc( ) or calloc( ) to allocate a 1-D array to hold the mono data.
Your main program will declare the appropriate variable for the fields of the header.
Your main program will assign the appropriate values to those variables.
Your main program will call a subroutine to fill-in the audio sample values.
Your main program will write out each field of the header to the file using one fwrite( ) per field.
Then your main program will write out the audio data to the file using a single fwrite( ).
If your program is correct, when you double-click on the WAVE file created by your program , your computer's default audio app will open the file and play it through your computer's speakers.

Be careful to use a data type for each field that has the correct number of bytes.
And be sure to include the blank space in the FORMAT field "WAV " and the SUBCHUNK1ID field "fmt ".

Prompt the user for a length (float) of audio in seconds, and for the frequency (float) in Hz.
Fill-in all of the variables for the fields of the wav header.
For example, the char chunkid[4] should hold 'R' 'I' 'F' 'F'.
There are many correct ways to fill-in the characters in the header. For example, to write RIFF into the chunkid you could
(a) load in the 4 characters one character at a time
(b) assign the 4 characters in the same statement where you declare the 1-D char array.
(c) You could use sprintf(chunkid,"%s","RIFF"); but this tries to write 5 bytes into chunkid, the 5th character being '\0' the string termination character. So instead, you should use strncpy(chunkid,"RIFF",4).
Be careful when you write "fmt " **(important: this must have the blank space after fmt)** to subchunk1id, you could also use strncpy(subchunk1id,"fmt ",4); .

The wav file will contain 16-bit audio data (short int, with both positive and negative integer values),
and it will be at 44100 samples per second.
bitspersample=16;
blockalign = etc. etc...

Allocate (malloc or alloc) a **1-D array** of **short int** variables of dimension sufficient to hold all of the audio data.
Note that audio audio data is signed, not unsigned.
Make sure that you correctly calculate how many audio samples are required.
For every 1.0 second of audio, you will need N=samplerate audio samples (e.g. at samplerate=44,100 you will need 44,100 samples for every 1.0 seconds of audio.

Your main program will then call a subroutine that fills in the audio data with a sinusoid whose frequency was selected by the user .
This subroutine will receive (as an argument) the pointer to short int audio-data array, the float frequency from the main( ) function, and the number of samples the subroutine should generate and load into the audio-data array..
Use a sine wave amplitude of 32,700.0 (as this is close to the max for a short int 2-byte value : 2^15 - 1).
The sample rate for your mono wav file is samplerate=44,100 samples per second.
A continuous sinusoidal tone with the amplitude listed above has the equation
tone(t) = 32700.0*sin(2*pi*freq*t).
We want to form samples of tone(t) at t = 1.0/samplerate, t=2.0/samplerate, t=3.0/samplerate,…. yielding the sampled data data[n]
data[n]=32700.0* sin(2*pi*freq*(float)n/samplerate) ;
Careful not to do integer division : (n/samplerate)=0 whenever int n < int samplerate.
If you use a for( ) loop to write the audio samples, it would look like
for(i=0 ; i<nsamples ; i++){psi[i]= (short int) (32700.0* sin(2*pi*freq*(float)i/samplerate)) ;}
where pi is appropriately defined and valued.

Don't forget to fclose( ) the wav file to free the allocated memory.