**CSC 745 Advanced Multimedia Programming**

**Assignment: Add reverb or echo to a recorded sound**

**Discussion**

If you’ve ever been in a recording studio, you may have noticed that walls and ceilings were covered with “acoustical” tile and the floors carpeted. The result is an acoustically “dead” or “dry” room, where sound seems to die away as it leaves your mouth. The philosophy in these studios is to record only the primary sound that is emitted from the instrument or vocalist—with no reflections from walls or other surfaces. Later, the recorded sound is processed in order to add “ambiance,” the warmth and vitality that comes from “reverberation” of the sound within a room. The level of reverb added to the sound depends on the room that is being modelled—a baroque trumpet concerto should sound like it was recorded in a small concert hall, for example, while a Gregorian chant should sound like it was recorded in a large, reverberant cathedral. In some cases, the goal is to produce an echo – when the cowboy looks over the desolate canyon, for example, and shouts “Halloooo,” the answer comes ringing back, “Halloooo-lloo-oo.”

**Specifications**

Complete the program *Asn2Reverb.py* to add reverb or echo to a sound; your task is to write code for the reverb() function. In effect, the process is simply mixing the sound with itself, but with some delay. The input file is specified in a variable near the top of the program. Delay is specified, in ms (milliseconds), by a variable defined at the beginning of *main()*. The amplitude of the delayed sound is also specified by a variable near the beginning of *main()* – for example, a delay amplitude of 0.3 will multiply the delayed samples by 0.3. In order to keep from exceeding the dynamic range of the system, you will want to use a weighted average of the current sample with the delayed sample. If the delayed sound is to be 0.1 as loud as the sound itself, for example, the following code will process one sample:

y[n] = 0.9 \* x[n] + 0.1 \* x[n-delay],

where x is the input signal and y is the output signal. Don’t change the input sound – modify a copy of the sound (*np.copy()* returns a copy of an ndarray). Be sure to change *return snd* to return your modified copy.

Do not modify any code outside *reverb(),* with the following exceptions:

1. You may want to experiment with other sound files and with other values for delay and delay\_amp; please restore the defaults before submitting your program.
2. If you use a library other than sounddevice to play sound, take the following steps:
   1. Insert a comment before the imports stating what sound library you’re using
   2. Comment out the statement that imports sounddevice
   3. Insert a statement to import your library immediately before or after the sounddevice import
   4. Comment out the calls to *sd.play()* and *sd.wait()* in *main()*
   5. Insert your call(s) to play sound using your library immediately before or after the calls to sounddevice in *main()*

**Deliverables**

Submit your .py file on Blackboard. I only need your .py file – I don’t need any sounds you experimented with. I will test your program with my sound files, which may include files other than the default provided in the assignment folder.