CPS109 Lab 10

Most of the questions in this lab come from Chapter 8 of the course text, Introduction to Computing and Programming in Python, by Guzdial and Ericson.

This lab is graded differently, since it is in the last week of classes. We want you to do what you can during the lab and submit that during your lab. As long as you submit what looks like a reasonable effort, then your lab will be marked as 100%. There will also be a quiz, with the same marking. Please put your answers (numbered) in a **lab10.py** file and submit it on D2L.

The **learning objectives** for Chapter 8 include:

- to splice sounds
- to reverse and mirror sounds
- to iterate an index variable for an array across an appropriate range

To do:

For each question, include in your solution code in lab10.py.

1) Write a function **increaseDecrease(sound, position)**, where position is a number from 0 to 1, inclusive, which indicates where in the sound to switch from increasing the sound by a factor of 2, to decreasing the sound by a factor of 0.2. For example, if position is 0, you just multiply all amplitudes by 0.2. If position is 1, you multiply all sample values by 2. If position is 0.3, then for the first 0.3 of the sound you multiply by 2 and for the last 0.7 of the sound you multiply by 0.2.
2) Write a function **increaseDecrease2(sound)**, which normalizes the sound in the first second, and then slowly decreases the sound in steps of 1/5 for each following second. Recall that the number of samples in a second is obtained with the function **getSamplingRate**(). So, you find the maximum absolute value of samples in the first second, and linearly scale the first second so that the maximum becomes 32,767. In the next second, multiply all values by 0.8. In the third second, multiply all values by 0.6, and so on.

3) Write a function **thisistest**(), which uses the test.wav sound to produce and return a new sound which says 'This is test', rather than what the original sound says ('This is a test'). Explore the test sound first to see where to clip and splice.
4) Write a function **reverse(sound)**, which does not use the function **getSampleValueAt**(), (so that your function is slightly different from the one in the text). The function reverses the order of all the sample values in the sound.
5) Write a function **clip(sound, start, end)**, which returns a sound obtained from the input sound starting at position start and ending at position end-1. Do not use function **getSampleValueAt**(), just so your version if different from the book version.