

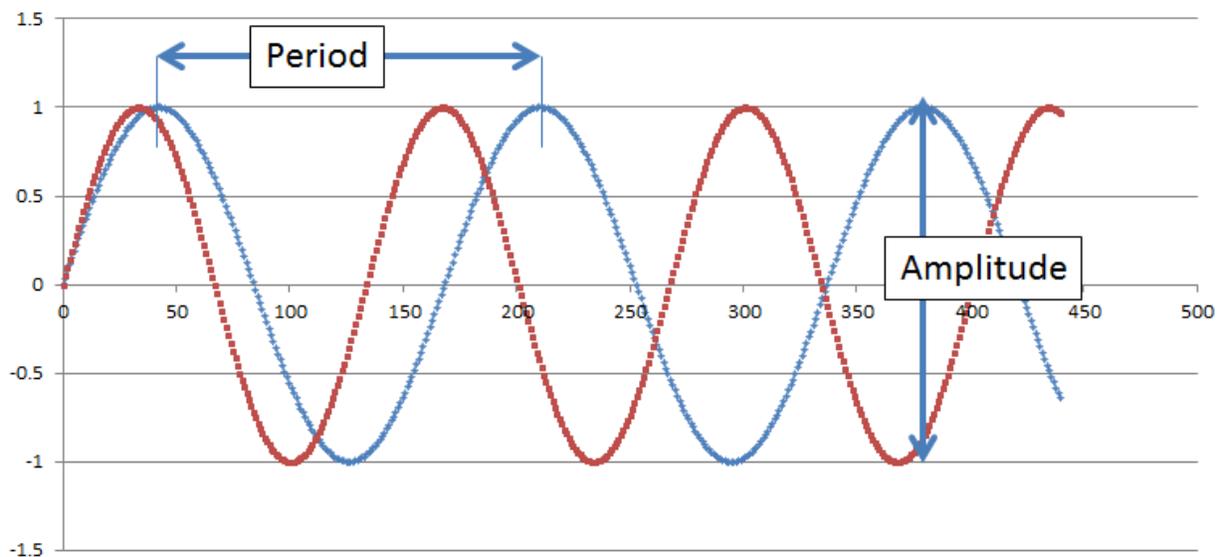
HW3: CS 110X C 2014

Note: This homework (and all remaining homework assignments) is a **partner homework** and must be completed by each partner pair. When you complete this assignment, you must not share your answers with any other student. Only one person from a partner pair needs to submit the assignment, but make sure that you submit before the deadline!

Domain Information

Sound is a traveling longitudinal wave which is an oscillation of pressure. An individual wave is defined by its period (the distance in time between two high points) and amplitude (the total distance vertically from the highest point to the lowest point). The amplitude represents the energy of the wave or its "loudness". For this assignment assume that all wave forms are normalized between $[-1, 1]$.

The human ear interprets a sound wave by converting it into a musical pitch (or note). Each musical note corresponds to a specific frequency which is measured in hertz, or the number of complete cycles per second of a periodic phenomenon (in this case, the sound wave). Numerous studies have demonstrated that the range of hearing for an infant child is 20 Hertz to 20,000 Hertz. The middle C Tone on a piano keyboard is tuned to the frequency of 261.626 Hertz, which is well within this range. See [Wikipedia entry on piano frequencies](#) for additional frequency values. If you were to sample this sound wave 44,100 times per second, then you would compute 44,100 individual values, of which the first 450 are shown below in the blue time series. The horizontal axis (t-axis) represents time, while the vertical y-axis represents the energy contained in the wave at time t.



To interpret the above blue sound wave you need to know the sampling rate and the time when the blue sound wave completes a full period. The blue wave period is about 169 time units. Since there are

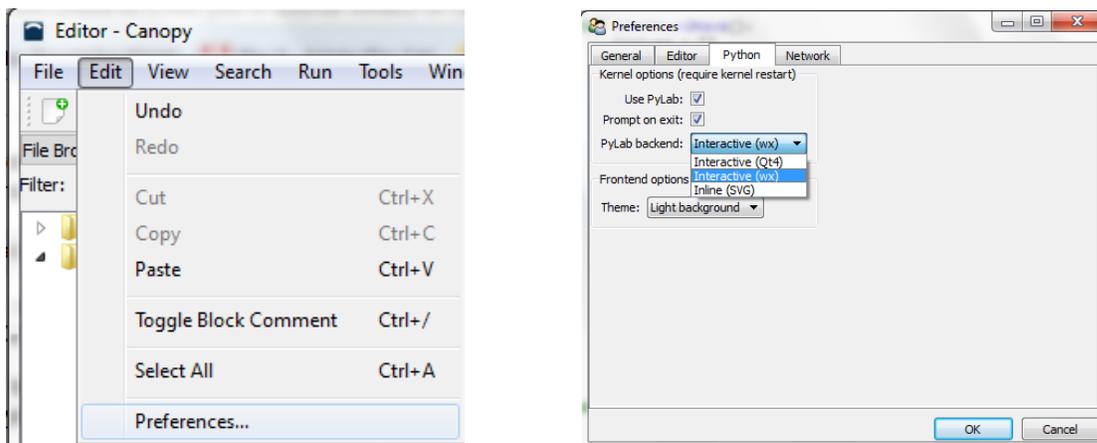
Final Version: 1/29/2014

44,100 total samples, the computed frequency of the blue wave is $44100/169$ or 260.95, which is very close to the middle C Tone stated earlier.

In the same graph above a second red sound wave is plotted, which represents the tone when playing the E key just above middle C. Based on a period of 135 time units, its frequency is computed to be 326.66 which is very close to its actual value of 329.628.

Canopy Issues

If you are running Canopy then you have to make a small configuration change for this homework to work properly. From within the Canopy Editor, select menu item **Edit | Preferences....**



Then in the Preferences window, select the **Python** tab and be sure that **PyLab backend** is set to "Interactive (wx)". I have tested this on Windows and on a MacBook.

Homework Instructions

This Homework has **seven** questions.

For each question be sure you understand exactly the format of the output that is requested. You will lose points if you do not exactly follow the format of the output for the individual questions. Should you have any questions, be sure to post on the HW3 discussion forum.

Q1	Demonstrate if statement use
<div style="border: 1px solid black; padding: 5px;"> <p>Skills</p> <p>CS-1 CS-3 PF-3</p> </div>	<p>Write a function <code>isLeapYear(year)</code> that returns a Boolean value if the given year is a leap year. Find the information for this computation on the associated Wikipedia page.</p> <p>You can assume that year is an integer greater than 1970.</p> <p>Your function must return a Boolean value.</p>
Sample Output in IDLE	<pre>>>> isLeapYear(2014) False</pre>
Sample Output in Canopy	<pre>In[1]: isLeapYear(2016) Out[1]: True</pre>

Q2	Demonstrate knowledge of for loop using indexing
<div style="border: 1px solid black; padding: 5px;"> <p>Skills</p> <p>PF-3 DT-9 CS-5 CS-9</p> </div>	<p>Write a function <code>listAdd(list1, list2)</code> that takes two lists of equal length containing numeric values and returns a new list that represents their numeric addition. Note that <code>list1</code> and <code>list2</code> must not be changed by <code>listAdd</code>.</p> <p><code>listAdd([1,2], [9,5])</code> must return <code>[10,7]</code> because $1+9=10$ and $2+5=7$.</p> <p>Note that you must not invoke any method from numpy to solve this problem. You will find the repetition operator for lists useful in this function (as mentioned in lecture on Jan-30)</p>
<div style="border: 1px solid black; padding: 5px;"> <p>Lecture Dependency</p> <p>Jan-30</p> </div>	
Sample Output in IDLE	<pre>>>> listAdd([1, 2, 4], [8, 3, 7]) [9, 5, 11]</pre>
Sample Output in Canopy	<pre>In[1]: listAdd([1,2], [6,7]) Out[1]: [7, 9]</pre>

Q3	Demonstrate knowledge of if statement, else , elif , definite for loop							
<table border="1"> <tr> <td>Skills</td> </tr> <tr> <td>CS-1</td> </tr> <tr> <td>CS-2</td> </tr> <tr> <td>CS-3</td> </tr> <tr> <td>CS-9</td> </tr> <tr> <td>DT-6</td> </tr> <tr> <td>SM-2</td> </tr> </table>	Skills	CS-1	CS-2	CS-3	CS-9	DT-6	SM-2	<p>The game of Fizz Buzz is a group word game used to explain division. Players take turns starting with 1 to count incrementally, replacing any number divisible by three with the word "Fizz" and any number divisible by five with the word "Buzz". If the corresponding number is divisible by both 3 and 5, then replace with the combined word "Fizz Buzz".</p> <p>Define a function <code>fizzBuzz(n)</code> that returns a list of string literals containing the first <code>n</code> responses in this game.</p>
Skills								
CS-1								
CS-2								
CS-3								
CS-9								
DT-6								
SM-2								
<table border="1"> <tr> <td>Lecture Dependency</td> </tr> <tr> <td>Jan-30</td> </tr> </table>	Lecture Dependency	Jan-30	<p>Note that all values in the returned list are string literals.</p> <p>You will find the modulo operator (%) useful in this function, as mentioned in lecture on Jan-30.</p>					
Lecture Dependency								
Jan-30								
Sample Output in IDLE	<pre>>>> fizzBuzz(16) ['1', '2', 'Fizz', '4', 'Buzz', 'Fizz', '7', '8', 'Fizz', 'Buzz', '11', 'Fizz', '13', '14', 'Fizz Buzz', '16']</pre>							
Sample Output in Canopy	<pre>In[1]: fizzBuzz(3) Out[1]: ['1', '2', 'Fizz']</pre>							

Q4	Demonstrate if and for loop			
<table border="1"> <tr> <td>Skills</td> </tr> <tr> <td>PF-3</td> </tr> <tr> <td>DT-6</td> </tr> </table>	Skills	PF-3	DT-6	<p>In mathematics, an arithmetic sequence is a sequence of numbers such that the difference between the consecutive terms is constant. For instance, the sequence 5, 7, 9, 11, 13, 15 ... is an arithmetic progression with common difference of 2 that starts at 5.</p>
Skills				
PF-3				
DT-6				
<table border="1"> <tr> <td>Lecture Dependency</td> </tr> <tr> <td>Jan-24</td> </tr> </table>	Lecture Dependency	Jan-24	<p>A sequence is uniquely determined by a_0 (starting value), d (the common difference) and n (the number of terms). You are to write a Python function that determines whether a list (containing 3 or more values) is an arithmetic sequence.</p> <p>Define a function <code>isArithmeticSequence(values)</code> that returns a Boolean value <code>True</code> if the list of values is an arithmetic sequence or <code>False</code> if it is not.</p>	
Lecture Dependency				
Jan-24				
Sample Output in IDLE	<pre>>>> isArithmeticSequence ([5, 7.5, 10.0, 12.5, 15.0, 17.5, 20.0]) True</pre>			
Sample Output in Canopy	<pre>In[1]: isArithmeticSequence([6, 7, 8, 10]) Out[1]: False</pre>			

Q5	Demonstrate knowledge of if statement, else , definite for loop										
<table border="1" data-bbox="191 268 370 667"> <tr> <td data-bbox="191 268 370 300">Skills</td> </tr> <tr> <td data-bbox="191 308 370 340">CS-1</td> </tr> <tr> <td data-bbox="191 348 370 380">CS-2</td> </tr> <tr> <td data-bbox="191 388 370 420">CS-3</td> </tr> <tr> <td data-bbox="191 428 370 459">CS-9</td> </tr> <tr> <td data-bbox="191 468 370 499">DT-6</td> </tr> <tr> <td data-bbox="191 508 370 539">SM-2</td> </tr> <tr> <td data-bbox="191 548 370 579">Lecture</td> </tr> <tr> <td data-bbox="191 588 370 619">Dependency</td> </tr> <tr> <td data-bbox="191 627 370 659">Jan-30</td> </tr> </table>	Skills	CS-1	CS-2	CS-3	CS-9	DT-6	SM-2	Lecture	Dependency	Jan-30	<p>There are seven days of the week. Let's consider Sunday to be weekday 0, Monday to be weekday 1, (and so on... until) Saturday to be weekday 6.</p> <p>If you know the day of the week for January 1st in any given year, then you should be able to compute the day of the week for any date in that year. In the template solution the following function is provided for you which returns an integer representing the day of the week for January 1st in the given year (either 0, 1, 2, 3, 4, 5 or 6).</p> <pre>def januaryFirst(year): year = year - 1 return (year + year/4 - year/100 + year/400 + 1) % 7</pre> <p>Write a date(month, day, year) function that returns a string literal "DOW Month DD YYYY" where DOW is the valid day of the week, Month is the English name for the month, and DD and YYYY represent the day and year.</p> <p>You can assume that the parameters reflect only valid days, for example, that when asked for a February 29th the year will always be a leap year; also, no one will ever ask for the 33rd of a month, or the 31st of September.</p> <p>You will find it useful to create several lists representing key concepts such as: the number of days in each of the twelve months, the names of the 12 months, the names of the seven week days.</p> <p>You will also use the isLeapYear(year) function defined earlier. You will find the modulo operator (%) useful in this function, as mentioned in lecture on Jan-30.</p> <p>Helpful Hint: To understand how to solve this problem, try to work the answer out manually for a date of your choosing. You might want to write down the steps before beginning the coding.</p>
Skills											
CS-1											
CS-2											
CS-3											
CS-9											
DT-6											
SM-2											
Lecture											
Dependency											
Jan-30											
Sample Output in IDLE	<pre>>>> date (1, 29, 2014) 'Wednesday January 29 2014'</pre>										
Sample Output in Canopy	<pre>In[1]: date(7, 3, 2016) Out[1]: 'Sunday July 3 2016'</pre>										

Q6	Demonstrate computational ability
<div data-bbox="191 268 370 300" style="border: 1px solid black; padding: 2px;">Skills</div> <div data-bbox="191 308 370 407" style="border: 1px solid black; padding: 2px;">PM-7 DT-6</div>	<p>Returning to the sound wave mentioned at the beginning of this homework, you are now going to generate some sound wave data. Assuming a sampling rate of 44,100 samples per second (the industry standard) every second of audio has 44,100 individual samples, where each sample is a real number in the range [-1, 1].</p>
<div data-bbox="191 457 370 520" style="border: 1px solid black; padding: 2px;">Lecture Dependency</div> <div data-bbox="191 529 370 558" style="border: 1px solid black; padding: 2px;">Jan-27</div>	<p>For this assignment you are to generate a sound wave containing just a single tone. For example, the Middle C tone of the piano scale has frequency = 261.626 Hertz.</p>
	<p>You will assume that the amplitude of the sound wave is 2 (that is, the sound wave ranges from values of -1 to +1) and you can generate this data using the built-in <code>math.sin(x)</code> function, as well as the <code>math.pi</code> constant.</p> <p>Define a <code>generateSoundWave(frequency, n)</code> function that returns a list of n values, each representing a sample of the sound wave over time, assuming a sampling rate of 44,100 samples per second. To do this, recognize that at time unit t_i the value of the idealized sound wave of that frequency is equal to</p> $f(t_i) = \sin(2 * \pi * t_i * frequency)$ <p>The function must return a list containing n real numbers representing the values of the sound wave sampled at discrete times t_i. Note that $t_i = i/44100.0$ for values of i in the range $(0, n)$.</p> <p>BE CAREFUL NOT TO OUTPUT LARGE LISTS TO THE CONSOLE BECAUSE THAT WILL FREEZE YOUR IDLE/CANOPY INSTANCES. IF THIS HAPPENS YOU HAVE TO TERMINATE THE APPLICATION AND START AGAIN.</p> <p>Hint: The description of this question is much longer than the code itself!</p>
Sample Output in IDLE	<pre>>>> generateSoundWave(261.626, 10) [0.0, 0.03726675716643086, 0.07448173986101789, 0.11159324554186605, 0.14854971542704604, 0.18529980612488656, 0.22179246096502583, 0.25797698093112376, 0.2938030950966868, 0.32922103046615064]</pre>
Sample Output in Canopy	<pre>In[1]: generateSoundWave (329.628, 5) Out[1]: [0.0, 0.0469467696588042, 0.09379001160049445, 0.1404264264088237, 0.18675317076582615]</pre>

Q7	Combine generated sound data
<div data-bbox="191 264 370 373" style="border: 1px solid black; padding: 2px;"> <p><i>Skills</i></p> <p>PF-3 DT-6</p> </div> <div data-bbox="191 411 370 520" style="border: 1px solid black; padding: 2px;"> <p><i>Lecture Dependency</i></p> <p>Jan-27</p> </div>	<p>The idealized sound wave only occurs in isolation. When multiple tones are played “at the same time” they overlap with each other. The energy of each wave is simply combined together which makes it easy to imagine how chords would sound. For example, consider the chord Middle C, E above Middle C and G above Middle C:</p> <div data-bbox="397 407 1383 724" style="display: flex; justify-content: space-around;"> <div data-bbox="397 407 828 724"> </div> <div data-bbox="896 407 1383 724"> </div> </div> <p>This is composed of waves of frequency 261.626, 329.628 and 391.995. The sound wave looks like the above. Note that it retains periodicity but is more complicated than a simple sign wave.</p> <p>Define a <code>combineSoundWave(frequencies, n)</code> function that takes a list of frequencies and returns a single list of wave form data, normalized to the range [-1, 1]. That is, the wave data is added and then averaged by dividing each individual sample total by the total number of frequencies.</p> <p>This function should use the <code>generateSoundWave(frequency, n)</code> function defined earlier.</p> <p>This function should invoke <code>listAdd(list1, list2)</code> which you defined earlier.</p> <p>The <code>plotWave</code> function has been provided for you in the template. The invocation below produces the combined wave form shown above.</p>
<p>Sample Output in IDLE</p>	<pre>>>> combineSoundWave ([261.626, 329.628, 391.995], 4) [0.0, 0.046678106849955159, 0.093246422596120745, 0.13959543715231698] >>> plotWave(combineSoundWave([261.626,329.628,391.995],1500))</pre>

How To Get Started On This Assignment

A template HW3.py file is provided to you with some sample functions already provided.

Much of the work for this assignment will be spent trying to understand the domain of sound waves and writing the appropriate Python code. In many ways, that is as it should be! The job of a programmer is more than learning a particular syntax. You need to know how to produce code relevant for a specific problem. Sometimes the code you write is only 5 lines of code (but it will be just the right five lines of code).

Submit your HW3.py file using the web-based turnin system. As we have mentioned in class, only one of the team members needs to submit the assignment. But just make sure that something gets submitted!

Change Log

1. Nothing yet...