MUSIC 208 Homework 1 (Units 1-2)

Name:

Copy all text to a text editor. Submit your answers in plain text format and place that text file in your Hand-in folder inside the course folder. Sending file as email is a less preferred option. Be sure the question numbers and question text is included on your answer page. Do not include Hints or images.

Assumptions

The sample rate is 44100 Hz in all questions.

Show your calculations in commented Octave code.

Sample Math

1. What is the sample period in microseconds to four decimal places
Hint: Get Octave help for format and then try the Octave command.

```
octave:5> format short eng;
```

2. What is the sample number is at the 10 minute mark?

3. What is the time in minutes and seconds of sample number 7,952,700?

Periodic Waveforms

4. The period is 50 samples. What is the frequency?

5 The period is 140 samples. A) What is the frequency? B) What is the closest MIDI note number to that frequency.

6. How many samples are in one period of the note Middle C on the piano. Answer to 2 decimal places.

7. Imagine a clock-like device with only one rotating hand. The hand is rotating

at the frequency of 440 Hz (A-440). The length of the hand is exactly 12 inches from the center of rotation to the tip of the hand.

a) How far will the tip of the hand travelled in one second?

b) What is the speed of the tip of the hand in miles per hour? There are 5280 feet in one mile.

Aliasing

The harmonic series is defined as integer multiplies of a fundamental frequency. For example if the fundamental frequency is 1000 Hz, the the first harmonic is 1000, the second is 2000, the third is 3000, and the nth harmonic is 1000*n.

8. Given a fundamental frequency of A-440 = 440 Hz, what is the largest value of n such that n*440 will remain under the Nyquist frequency?

9. What does the term bandlimited mean?

Octave Function Files

10. Write an Octave function named ms2samp that converts a millisecond time into the sample number.

```
function [ ret ] = ms2samp ( millis )
    # you fill in
    # return value is the sample number
endfunction
```

11. Write an Octave function named samp2ms that converts a sample number into a millisecond time.

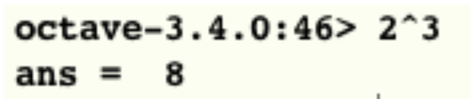
```
function [ ret ] = samp2ms ( sampleNum )
    # you fill in
    # return value is the millisecond time
endfunction
```

12. Write an Octave function named mtof that converts a MIDI note number into a frequency in Hz.

```
function [ ret ] = mtof ( midiNoteNumber )
    # you fill in
    # return value is the frequency
endfunction
```

Hints:

$$frequency = 440 * 2^{\frac{midiNoteNumber - 69}{12}}$$



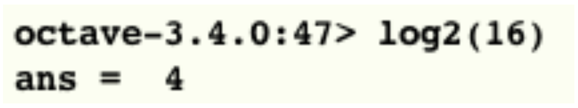
```
octave-3.4.0:46> 2^3
ans =  8
```

13. Write an Octave function named ftom that converts a frequency into a MIDI note number.

```
function [ ret ] = ftom ( frequency )
    # you fill in
    # return value is the MIDI note number with decimal places
endfunction
```

Hints:

$$MIDINoteNumber = 69 + \log_2(\frac{freq}{440})$$



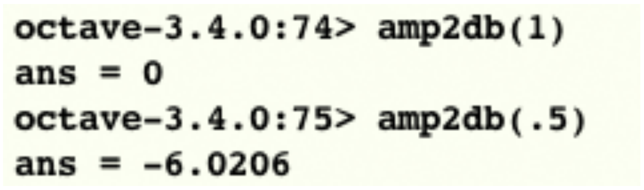
```
octave-3.4.0:47> log2(16)
ans =  4
```

14. Write an Octave function named amp2db that converts amplitude (0-1.0) into decibels.

```
function [ ret ] = amp2db ( amplitude )
    # you fill in
    # amplitude range is 0.0 to 1.0
    # return value is the dB value 0 to -100
endfunction
```

Hints:

$$dB = 20 * \log_{10}(amplitude)$$



```
octave-3.4.0:74> amp2db(1)
ans = 0
octave-3.4.0:75> amp2db(.5)
ans = -6.0206
```

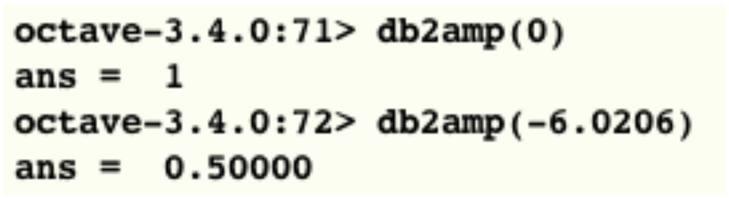
15. Write an Octave function named db2amp that converts decibels into amplitude values.

```
function [ ret ] = db2amp ( decibels )
    # you fill in
    # return value is the amplitude scaled from 0 to 1.0
endfunction
```

Hints:

$$amplitude = 10^{\frac{dB}{20}}$$

Note: amplitudes are entered as negative numbers. 0 dB is maximum amplitude (1.0).



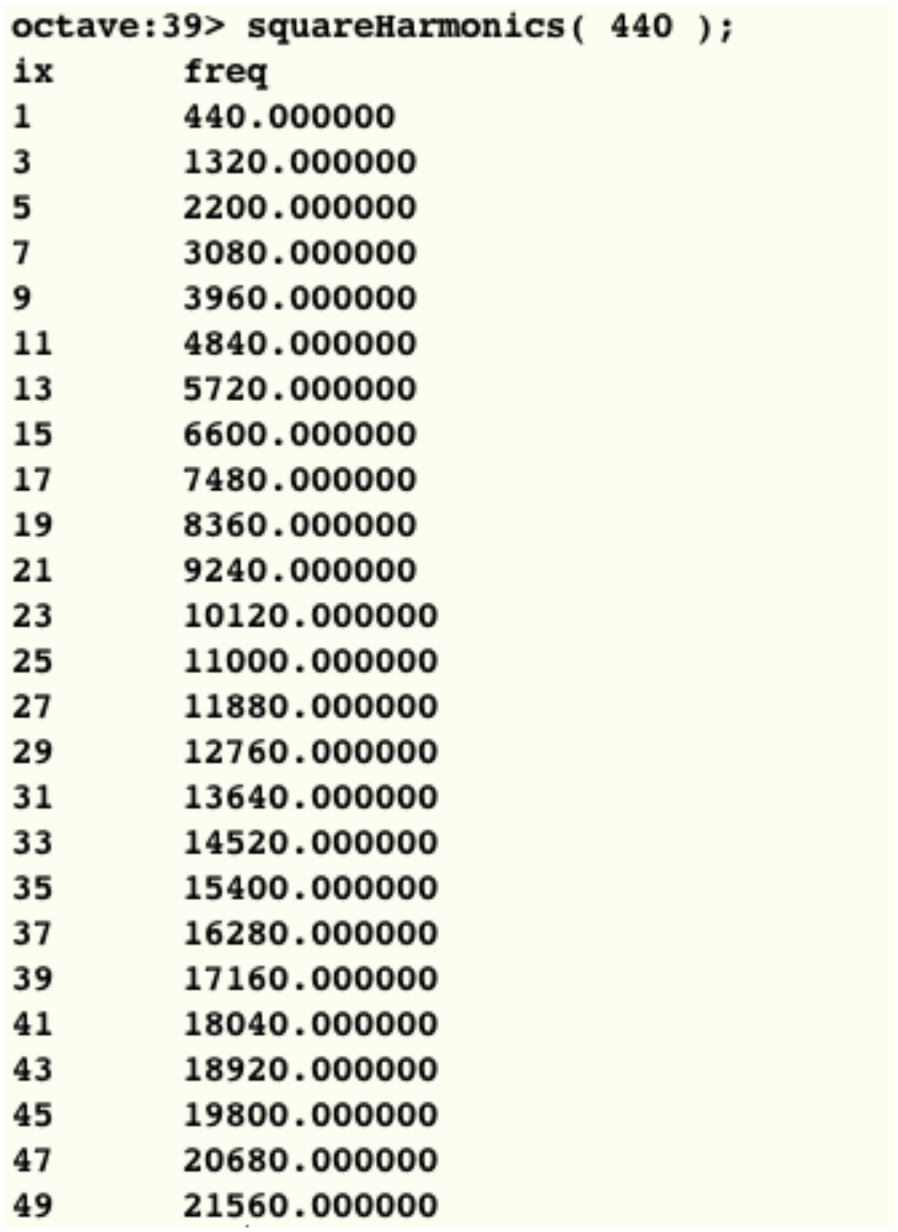
```
octave-3.4.0:71> db2amp(0)
ans =  1
octave-3.4.0:72> db2amp(-6.0206)
ans =  0.50000
```

16. A square wave can be generated by the formula

$$\omega = 2 * \pi * f;$$

$$squareWave = \sin(\omega) + \frac{1}{3}\sin(3\omega) + \frac{1}{5}\sin(5\omega) + \frac{1}{7}\sin(7\omega) + \frac{1}{9}\sin(9\omega)...$$

Write an octave function named squareHarmonics that will print the the values of all the odd numbered harmonics below the Nyquist frequency for a given starting frequency f.



```
octave:39> squareHarmonics( 440 );
ix      freq
1       440.000000
3       1320.000000
5       2200.000000
7       3080.000000
9       3960.000000
11      4840.000000
13      5720.000000
15      6600.000000
17      7480.000000
19      8360.000000
21      9240.000000
23      10120.000000
25      11000.000000
27      11880.000000
29      12760.000000
31      13640.000000
33      14520.000000
35      15400.000000
37      16280.000000
39      17160.000000
41      18040.000000
43      18920.000000
45      19800.000000
47      20680.000000
49      21560.000000
```

Hints: read about and get help on
for loops
break
if (...) statements
printf
rem (as in remainder, needed to test whether a number is odd or even)