CE-MATH Machine Exercise 3
Due: March 25, 2020 08:00 PM

This is an individual work. Intellectual dishonesty will not be tolerated. For every m-file that you will create, write/type the honor code below after your name.

"*I do hereby affirm, on my honor as a student, at the submission of this machine exercise, that the work I am submitting is my own and was not copied or done with others*"

PROBLEM 1: Signal Generation using Matrix Operations

The sawtooth and reverse-sawtooth signals are a combination of sinusoids respectively, defined as

$$y_{\text{sawtooth}}(t) = \frac{A}{2} - \frac{A}{\pi}\sum_{k=1}^{\infty}(-1)^k \frac{\sin(2\pi k f_0 t)}{k} \qquad \text{Eq. (1)}$$

$$y_{\text{reverse\_sawtooth}}(t) = \frac{2A}{\pi}\sum_{k=1}^{\infty}(-1)^k \frac{\sin(2\pi k f_0 t)}{k} \qquad \text{Eq. (2)}$$

For $k = 1$, the sinusoid frequency is f_0 and is called the signal's fundamental frequency. For $k = 2$, the sinusoid frequency is $2f_0$ and is called the second harmonic. For $k = 3$, the sinusoid frequency is $3f_0$ and is called the third harmonic, and so on... In general, the sawtooth and reverse-sawtooth signals are generated by adding the sinusoids whose frequencies are integer multiple of the fundamental frequency.

Using matrix addition, write a function that generates 1-second long of:

- sawtooth signal, for even-number ID number students
- reverse sawtooth signal, for odd-number ID number students

The input parameters are f_0, A and K, where f_0, A are the fundamental frequency and the amplitude of the signal described in Equations 1 and 2, K is the number of harmonics. The output of the function is a plot of the signal. Properly label your axes. No need to save any generated signal, I only need the output plot.

PROBLEM 2: Generate DTMF signal using matrix operation

The human hearing range is from 20 Hz to 20 kHz. The lower the frequency, the lower the tone. The higher the frequency, the higher the tone. A tone is sinusoid in the human frequency range. Therefore, a tone is generated by implementing the sinusoid:

$$y(t) = A\sin(2\pi f_0 t)$$

To generate a tone for 2 seconds with tone frequency equal to 1000 Hz, set f0 = 1000, and end time equal to 2. That is,

```
A = 2;
f0 = 1000;              % Tone frequency is f0 in Hz.
N = 10000;              % 10000 samples per second
t = (0:1/N:1-2/N);      % Start at 0, interval is 1/N and end at 2-1/N (because it starts at 0)
y = A*sin(2*pi*f0*t);   % Use the above equation
soundsc(y,fs);          % listen to the tone, note that it will be played for 2 seconds
```

Notice that if you use lower frequency, say 200 Hz, the tone that you will hear is also lower. Similarly for higher frequency, say 3000 Hz, the tone that you will hear is also higher.

The Dual-Tone Multi-Frequency signalling system uses the human voice-frequency band over telephone lines for use in push-button telephone devices. The layout of a push-button keypad is provided in Figure 1. Each row and column of the keypad corresponds to a unique set of low frequencies and high frequencies, respectively. Every time a button is pressed, a combination of these two tones is activated.

	1209 Hz	**1336 Hz**	**1477 Hz**
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

Figure 1: The DTMF Signalling System

Generate a MATLAB function called called dtmf_sig(x) that accepts an array of integers from 0 to 11, where integer 10 corresponds to the star key (*) and integer 11 corresponds to the pound key (#). The function should output a DTMF signal (series of continuous 0.2-second sound per key) corresponding to the input array. The input vector *x* should be of any length. However I will test only up to 20 integers in *x*. No need to perform error checking, assume that all input is always from 0 to 11.

For example, for input *x* = [4 1 4 3 3 9 7], the function will play the sound (listen to 4143397.wav uploaded file). You don't need to save the signal in a mat or in a wav file. The function just needs to output the DTMF signal using the *soundsc*( ) function.

What to submit? Place all the required m-files in a zip file named <surname_x>.zip where surname is your surname and x is the first letter of your name. Upload the zip file to canvas before the deadline.

Hint: You can use the Internet to discover the equivalent command in MATLAB. For example: "*switch* command in MATLAB" or "length of a vector command in MATLAB" or "*printf* command in MATLAB". You can also use the *help* command for the syntax in MATLAB. For example in the command line you can search, *help switch*, or *help length* then the syntax for these commands will be printed.