# Signal and Systems
# Simulation Phase 3
# DTMF - Live Tone Detector

**Alireza Moradi**

May 17, 2020

In this phase we had to implement a live DTMF Decoder so the only thing different from previous phases is the sound source.

`sounddevice` and `soundfile` are used to record sound and save it to a `wav` file, `wavfile` is used to load the recorded sound into the program.

[ ]:
```
import sounddevice as sd
import soundfile as sf
from scipy.io import wavfile as wav
from DTMF1copy import DTMF1
```

To detect the tones live, we use laptops microphone, In an infinite loop we record a sound with 1 second length and 44100 $Hz$ sample rate. Then we wait until the recording is finished, then we save the recording in a `wav` file in load it back again into the program, then we feed it to the decoder implemented in phase 1.

Finally if a tone is detected it is printed and the above algorithm runs infinitely.

[ ]:
```
if __name__ == '__main__':
    fs = 44100
    duration = 1  # seconds
    print('Listening...')
    while True:
        result = ''
        myrecording = sd.rec(duration * fs, samplerate=fs, channels=1,␣
↪dtype='float64')
        sd.wait()
        sf.write('file.wav', myrecording, fs)
        rate, signal = wav.read('file.wav')
        result = DTMF1(signal, rate)
        if (result != ''):
            print(result)
            print('Listening...')
```