LAB 17: Cassandra Cabrera and Mike Menendez **Due:** April 13, 2020

Summary:

For this lab, it was really interesting to change values and try and hear the different chords/sounds produced. We started with the original code and ran the “bach” code, after using that we decided to try to different values that we chose and produced some funky sounds. Then for task 3, we decided that to experiment a bit we would produce a sound at random so we imported random numbers and decided to create a “song” from that and we were able to produce some pretty cool sounds. For task 4, we experimented with random chords that we thought would be close to the youtube video we explored in class. Overall we had a lot of fun with this lab and were able to use a bunch of random values to make a pretty cool sound!

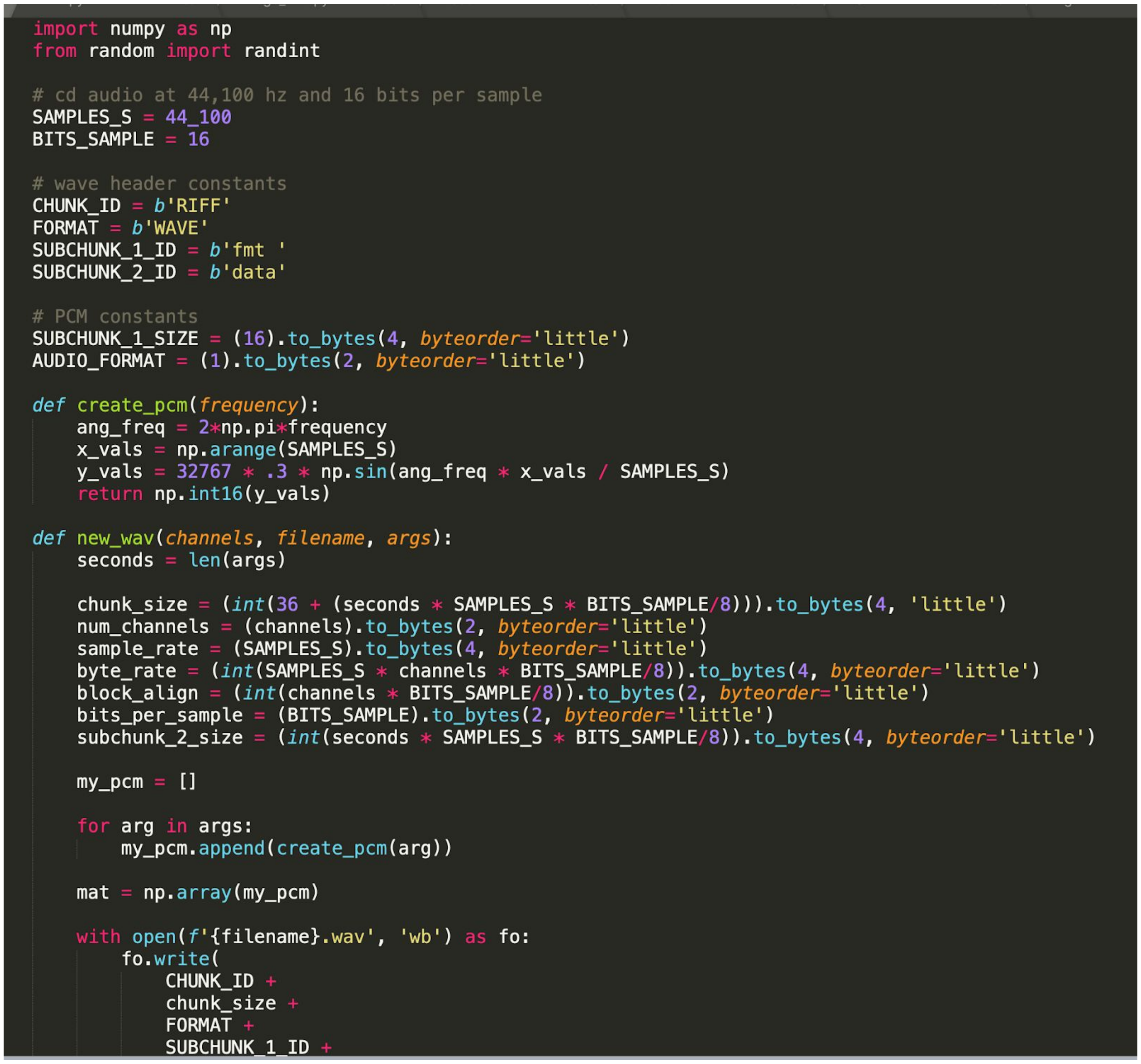
TASK 1:

We installed scipy.

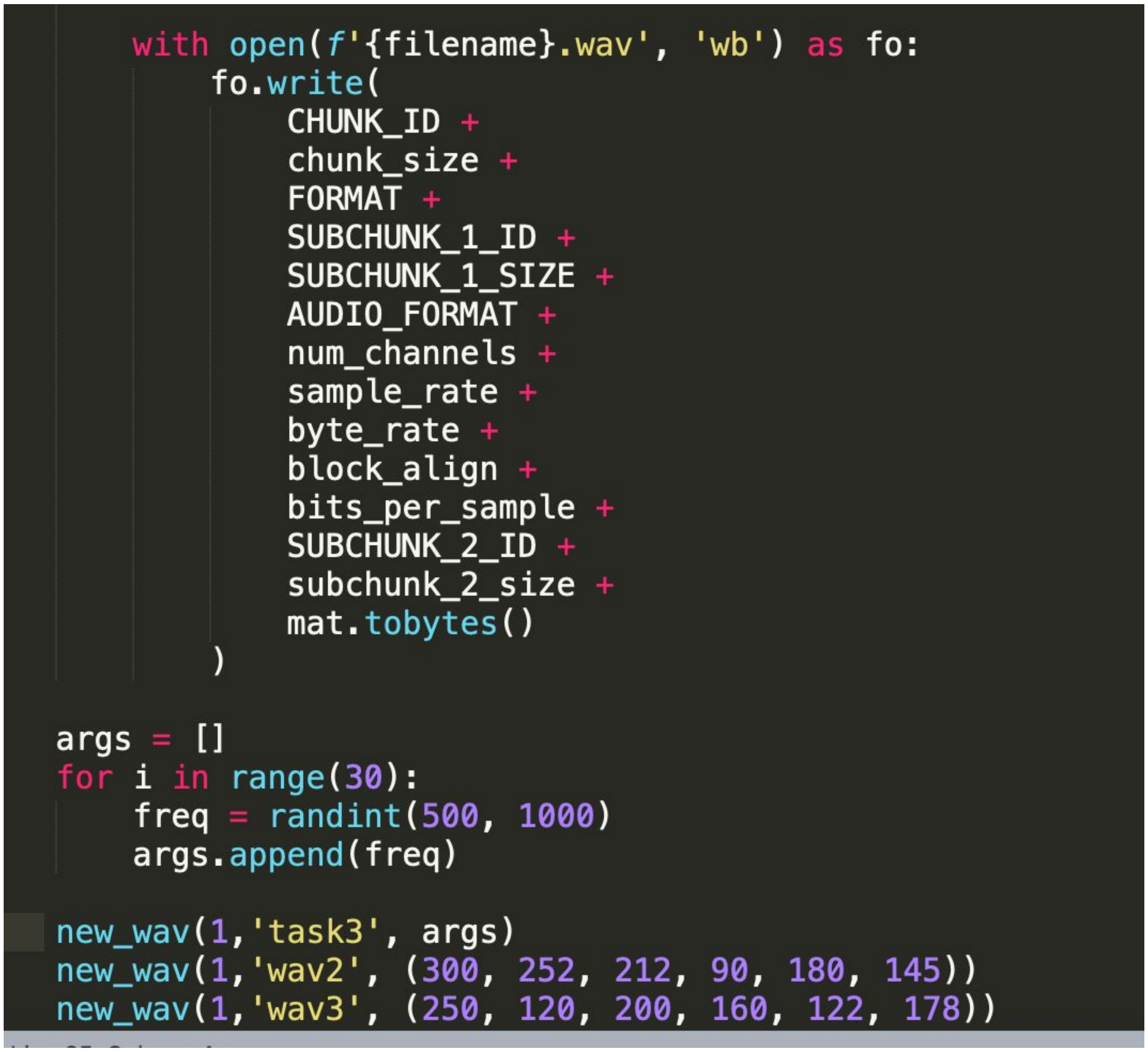
TASK 2:

We installed Audacity.

TASK 3 & 4:



```python
import numpy as np
from random import randint

# cd audio at 44,100 hz and 16 bits per sample
SAMPLES_S = 44_100
BITS_SAMPLE = 16

# wave header constants
CHUNK_ID = b'RIFF'
FORMAT = b'WAVE'
SUBCHUNK_1_ID = b'fmt '
SUBCHUNK_2_ID = b'data'

# PCM constants
SUBCHUNK_1_SIZE = (16).to_bytes(4, byteorder='little')
AUDIO_FORMAT = (1).to_bytes(2, byteorder='little')

def create_pcm(frequency):
    ang_freq = 2*np.pi*frequency
    x_vals = np.arange(SAMPLES_S)
    y_vals = 32767 * .3 * np.sin(ang_freq * x_vals / SAMPLES_S)
    return np.int16(y_vals)

def new_wav(channels, filename, args):
    seconds = len(args)

    chunk_size = (int(36 + (seconds * SAMPLES_S * BITS_SAMPLE/8))).to_bytes(4, 'little')
    num_channels = (channels).to_bytes(2, byteorder='little')
    sample_rate = (SAMPLES_S).to_bytes(4, byteorder='little')
    byte_rate = (int(SAMPLES_S * channels * BITS_SAMPLE/8)).to_bytes(4, byteorder='little')
    block_align = (int(channels * BITS_SAMPLE/8)).to_bytes(2, byteorder='little')
    bits_per_sample = (BITS_SAMPLE).to_bytes(2, byteorder='little')
    subchunk_2_size = (int(seconds * SAMPLES_S * BITS_SAMPLE/8)).to_bytes(4, byteorder='little')

    my_pcm = []

    for arg in args:
        my_pcm.append(create_pcm(arg))

    mat = np.array(my_pcm)

    with open(f'{filename}.wav', 'wb') as fo:
        fo.write(
            CHUNK_ID +
            chunk_size +
            FORMAT +
            SUBCHUNK_1_ID +
```



```python
    with open(f'{filename}.wav', 'wb') as fo:
        fo.write(
            CHUNK_ID +
            chunk_size +
            FORMAT +
            SUBCHUNK_1_ID +
            SUBCHUNK_1_SIZE +
            AUDIO_FORMAT +
            num_channels +
            sample_rate +
            byte_rate +
            block_align +
            bits_per_sample +
            SUBCHUNK_2_ID +
            subchunk_2_size +
            mat.tobytes()
        )

args = []
for i in range(30):
    freq = randint(500, 1000)
    args.append(freq)

new_wav(1,'task3', args)
new_wav(1,'wav2', (300, 252, 212, 90, 180, 145))
new_wav(1,'wav3', (250, 120, 200, 160, 122, 178))
```