**Signals Lab project’s report**

**Name: Ahmed Tarek ID:52-22174**

**Name: Omar Ahmed ID:52-23411**

**T-3**

Text

Description automatically generatedFirst on Spyder IDE, we imported the following libraries, numpy, matplotlib. pyplot and sounddevice.

After that, we initialized the linspace which represents the time to start from 0 up to 3 with 12\*1024 sample.

Then we saved the frequencies of the 3rd and 4th octaves in two separate arrays.

We assigned the number of pairs to be 8, and we saved ti and Ti in two separate arrays.

Text

Description automatically generatedThen we iterated over the number of pairs and every time we add the new signal to the previous one and we plot the signal every iteration.

Then, we played the sound using the play method, but in order this method to work, we had to write “*pip install sounddevice --user*” in the console and then restart the spyder IDE.

Now, in the noise cancellation part, we first imported from “scipy.fftpack” “fft”.



Then, we tranformed our signal from the time domain to the frequancy domain (x\_f).

Text

Description automatically generated

After that, we created two random frequancy for generating the noise signal.



Then, we added the old signal to the noise signal to get (Xn), and we changed the resulted one to the frequency domain (Xn\_f).

Text

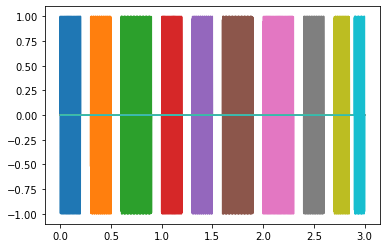
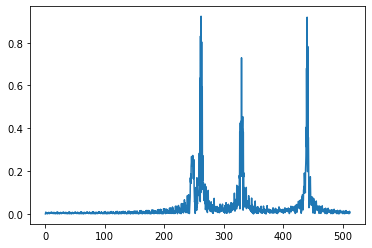
Description automatically generated

Now in order to cancel this noise, we looped through the resulted signal in the frequency domain to get the two frequencies at which ths signal amplitude becomes greater than the maximum points in the old signal. With these both frequencies, we generated a new signal and subtracted it from (Xn) and we got (xfiltered) and lastly we changed it to the frequency domain.

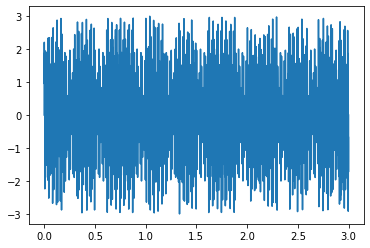
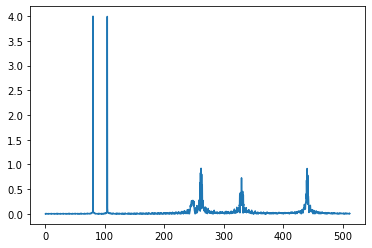
Text

Description automatically generated

**Figure** 1: Original Song

**Figure** 2: Song with noise

**Figure** 3: Filtered song

