Problem 1C: Ariana Grande and her 5-octave range ……

Pertinent reading for Problem #1c:

Ariana Grande + Mac Miller: The Way

This video is pretty old-school, and it came out back in the days when she was going out with Mac Miller (RIP !!) The sound clip you'll be analyzing is approximately from 1:26 – end of the song.

https://www.youtube.com/watch?v=_sV0S8qWSy0

How human vocal cords work, and what are overtones / harmonics…..

https://www.youtube.com/watch?v=qBpAU-j_FWg
https://www.youtube.com/watch?v=dJSiQrAerLk

Highest + lowest vocals (world records)

https://www.youtube.com/watch?v=Ia57VfDaESw
https://www.youtube.com/watch?v=AaPtiFO-NLc

Part A: Loading your audio file + separating out the left / right channels

1. Using the *audioread* function, load *Ariana_Grande_TheWay.wav* into matlab. Make sure your sampling frequency is 44.1 kHz before you start anything else !

$$f_{sample} = 44.1\ kHz$$

2. Separate the left vs. right channels into 2 separate column vectors. For the rest of this problem, you can just focus on channel 1's data.

Part B: Create the spectrogram of the original song

1. Using a window length of:

$$L = \frac{1}{5} f_{sample} = 8820\ points = 0.2\ second\ worth\ of\ data\ per\ window$$

a) First, add the appropriate # of *ghost-zero nodes* at the end of the audio file for our choice of L

b) Create a *non-overlapping* spectrogram using our window length L. Store your spectrogram data in a giant matrix (you don't have to echo it), and it should be ready to-be plotted with the *pcolor* command.

2. Using matlab's *pcolor* command, plot your spectrogram. For this diagram, you should apply the same caxis limits on your colorbar such that you can actually see the Fourier coefficients:

```
caxis( [ 0  0.005 ])
```

Part C: Filtering everyone out so that you can only hear her whistles

If you've listened to the sound clip (and if you've plotted the corresponding spectrogram), you will surely have noticed some fairly conspicuous features within this song, where there are a series of high whistles at the very end of the song

Your job: Filter the song such that you

i) Keep the DC component of this song (ie. Don't delete d_0)
ii) Remove (or more accurately, attenuate) almost everything you hear, but….
iii) You can still hear the big whistles at the end of the song

Your tasks for Part C:

1) Delete all necessary Fourier coefficients d_k in your spectrogram matrix. Make a *pcolor* plot of your post-filtered spectrogram (this plot will also serve as a self-check debugging step !!)

Hint: Think……. will you be doing:

a) Low-pass filtering, or
b) High-pass filter ???

Hint: Even though this will NOT be a low-pass filter operation like Pokeman.wav, when you're deleting the Fourier coefficients, you have to make sure you are still deleting things symmetrically with respect to the p-hops d_k coefficient !

2) Then, reverse-engineer your post-filtered sound file by using your filtered d_k coefficients at each time slice. Store your post-filtered, stitched audio data as 1 giant horizontal vector.

$$y1 = \begin{bmatrix} LPF \\ channel\ 1 \\ data \\ \\ (left\ speaker) \end{bmatrix}$$

3) Package your audio file into a 2-column matrix "y"

$$y = \begin{bmatrix} LPF \\ channel\ 1 \\ data \\ \\ (left\ speaker) \end{bmatrix} \begin{bmatrix} LPF \\ channel\ 1 \\ data \\ \\ (right\ speaker) \end{bmatrix}$$

Note: You have to convert your "y1" into a column vector before inserting it into "y" !!

4) Using the *audiowrite* command, write your filtered song "y" as:

Ariana_whistles_only.wav

Part D: Filtering out Mac Miller's voice

You may also have noticed there is a solo section featuring Mac Miller. What I want you do is to:

i) Keep the DC component of this song (ie. Don't delete d_0)
ii) Remove (or more accurately, attenuate) Mac Miller's voice, but….
iii) You can still sort of hear Ariana's voice throughout the song

Your tasks for Part D:

1) Repeat the tasks you did in Part C, but this time, we want to attenuate Mac Miller's voice altogether

Hint: When Mac Miller starts his rap routine, you'll see a significant change in the overall spectrum !

2) Using the *audiowrite* command, write your filtered song "y" as:

Ariana_voice_only.wav