Pitch tracking is an important building block of many music analysis systems. This assignment aims to guide you through the basic concepts, implementations, and potential improvements of simple monophonic pitch trackers based on the **Auto Correlation Function (ACF)**.

General guidelines:

- Follow the function templates in the attached folder. Insert code only where you have been asked to.
- Ensure input and output dimensions of your functions are accurate.
- Toolbox methods are not allowed allowed (e.g. xcorr)
- Core MATLAB functions are allowed (e.g. min, max, fft etc.)

A. Block-wise Auto-Correlation based pitch tracker

1. *[10 points]* Implement a MATLAB function: **[xb, timeInSec] = myBlockAudio(x, blockSize, hopSize, fs)** for blocking the input audio signal and storing time-stamps (in seconds) for the blocks. Note that the time stamps for each block will correspond to the starting sample in the block. Please refer to the annotation text files in the zip folder to verify using hopSize = 512 samples and fs = 44100.
2. *[10 points]* Implement a MATLAB function: **[r] = myCompAcf (inputVector, bIsNormalized)** that takes a block of audio and computes the ACF with optional normalization.
3. *[20 points]* Implement a MATLAB function: **[f0] = myGetF0FromAcf(acfVector, fs)** that takes the ACF of a block of audio as input and computes the fundamental frequency f0 of that block in Hz (Textbook Reference: 2.2.6, page 24).
4. *[10 points]* Implement a MATLAB function: **[f0, timeInSec] = myPitchTrackAcf(x, blockSize, hopSize, fs)** that estimates the fundamental frequency f0 of the audio signal based on a block-wise autocorrelation function approach. Note: This function should use all the functions you implemented in 1. to 3. above.

B. Evaluation (windowSize = 1024, hopSize = 512)

1. *[10 points]* In a separate script named **assign1.m**, generate a test signal (sine wave, f = 441 Hz from 0-1 sec and f = 882 Hz from 1-2 sec, fs = 44100), apply your **myPitchTrackerACF()** and plot the f0 curve (expected error <= 5%). Also, plot the absolute error per block and discuss the possible causes for the deviation.
2. [10 points] Implement a MATLAB function: **[pitchInMidi] = myFreq2MidiPitch(pitchInHz)** that converts a column of pitches in Hz to pitches in MIDI (assume A4 = 440.0Hz, Textbook Section: 5.2.5, page 88).
3. *[10 points]* Implement a MATLAB function: **[errCentRms] = myEvaluation(estimation, annotation)** that calls **myFreq2MidiPitch()** to convert frequency f0 to MIDI pitch and computes the RMS of the error. Note: The evaluation should exclude the blocks with annotation == 0
4. *[20 points]* Evaluate your **myPItchTrackerACF()** using the training set (see attachment). Report the overall errCentRms of the training set. What are the potential

solutions to improve their performances? Note: a) The annotation column from left to right is : [onset_seconds, duration_seconds, pitch_frequency, quantized_frequency], you only need to use the 1st and the 3rd column), b) You may use the annotation time-stamps to actually test the time-stamps returned by your **myBlockAudio()** method. c) The average error (errCentRms) for the baseline implementation for the training files should be arround (500-700).

C. (Bonus) Improving your Pitch Tracker

1. *[10 points]* Implement a MATLAB function: **[f0, timeInSec] = myPitchTrackMod(x, windowSize, hopSize, fs)** that modifies your method and improves the performance of your basic ACF pitch tracker thereby providing better f0 estimations. You may create other functions of your own but they should be called from within the myPitchTrackMod() function and a brief description of your functions should be in your report. Please explain your approach in the report. Your function will be tested using a testing set (not provided), and points will be given based on its performance compared to the baseline and the other groups. Best performing group gets 10 points and worst group gets 1 point. [HINT: Think of what modifications will affect the detected pitch, for example: different methods to detect peaks in the ACF function.]