Introduction to research experimental methods

TP1: Data acquisition

System : Your computer
Output : Beats played in headphones
Files « PeriodicAlong.wav » and « Aperiodic.wav »
Input : Audio signal recorded with an external microphone (or the built in microphone of your computer)

Script (In octave or python) :

1- Write a script that enables you to record speech while the audio beats are played to the headphones at the same time (Fs=20 000 Hz, 16 bits)

- Load the audio signal stored in a .wav file
  Function read from package scipy.io.wavfile (from scipy.io.wavfile import read)
  fs, audio_signal=read(filename)
- Play the first 10 seconds of this audio signal
  Package sounddevice (import sounddevice as sd)
  Function play from package sounddevice
  sd.play(audio_signal, fs)
  To stop reading before the end of the file : sd.stop()
  To make Python wait a certain amount of time (pause) before executing a next command:
  Function sleep from package time time.sleep(Nb_of_sec)
- Start a mono audio recording (i.e. on one channel), of 10s
  Function rec from package sounddevice
  recorded_signal = sd.rec(int(Nb_of_sec * fs), samplerate=fs, channels=1)
- Play and record simultaneously, for a given duration (e.g. 10s)
  playrec function of the sounddevice package
  recorded_signal = sd.playrec(audio_signal,channels=1,samplerate=fs)
  As before, to stop playback and recording : sd.stop()
  To wait a certain time before executing a next command : time.sleep(Nb_of_sec)
- Save the recorded audio signal
  Function write from package scipy.io.wavfile (from scipy.io.wavfile import write)
  write(Rec_filename, samplerate, recorded_signal)

2- Write a script that enables you to open the recorded audio file and to visualize, one under the other, the beats signal and the audio signal

- Visualize a signal as a function of time
  Package matplotlib.pyplot (from matplotlib import pyplot as plt)
  t = np.arange(0, len(rec_signal))
  fig, axs = plt.subplots()
  axs.plot(t, rec_signal)
  axs.set_title("Signal")
  axs.set_xlabel("Time")
  axs.set_ylabel("Amplitude")
  plt.show()
- View only a portion of this signal
  t = np.arange(0, duration*fs)
  rec_signal[tdeb*fs: (tdeb+duration)*fs]
- Visualize two signals one below the other
  fig,(ax1,ax2) = plt.subplots(2,figsize=(10,12))
  fig.suptitle('The two signals')
  ax1.plot(t,first_signal)
  ax2.plot(t,second_signal)
  ax1.set_title('first signal')
  ax2.set_title('second signal')
  plt.show()

Experiment :

- Plug the headphones and the microphone in output and input of your sound card
- If necessary, configure the inputs/outputs of your soundcards

Method 1: For a longer experiment, prepare a materials list, a wiring diagram and a to-do list and check them before the experiment (as if someone had to replace you at the last moment)

- Run separately the experiment for the two tasks (Speech in PeriodicAlong and Aperiodic) and record the speech signal

Method 2: For a longer experiment, prepare a sheet with the order of conditions
Method 3: On an experiment notebook, note the material, the settings, the subject code (anonymized), the order of the conditions and the name of the files, the possible errors, the observations...
Method 4 : Give clear names to the files to facilitate automatic processing (e.g.: Subject_Task_Speech_Repetition.wav) or keep an order file (Condition column / file number column automatically incremented)