Audio Filter Graphical User Interface (GUI)	Programs in this unit use: Pyaudio Tkinter FigureCanvasTkAgg Animation									
				Input				**Plots**		
		Numpy	**Slider**	**From file**	**From mic**	**Play output**	**Filter**	**signal**	**frequency response**	**Spectrum**
prog_01.py	Read a signal from a wave file, play the signal using Pyaudio. Plot the waveform using the animation function in a Tkinter window.			Y		Y		Y		
prog_02_gain.py	Add a slider to adjust the gain.		Y	Y		Y		Y		
prog_03_gain_np.py	Uses Numpy	Y	Y	Y		Y		Y		
prog_04_pause.py	Add buttons for pause and play.		Y	Y		Y		Y		
prog_05_pause_np.py	Uses Numpy	Y	Y	Y		Y		Y		
prog_06_filter.py	Implement a filter as a recursive difference equation. Adjust the cut-off frequency with a slider using Tkinter. Plot the output waveform using the animation function, plot the frequency response of the filter.	Y	Y	Y		Y	Y	Y	Y	
prog_07_filter_spectrum.py	Also plot the Fourier transform of the output signal.	Y	Y	Y		Y	Y	Y	Y	Y
prog_08_mic_v1	Implement a filter as a recursive difference equation where the the input is from the microphone. Use Pyaudio to acquire the microphone signal and play the output signal. Adjust the cut-off frequency with a slider using Tkinter. Plot the frequency response of the filter.	Y	Y		Y	Y	Y		Y	
prog_09_mic_v2	Also plot the waveform of the output signal using the animation function.	Y	Y		Y	Y	Y	Y	Y	
prog_10_spectrum.py	Also plot the Fourier transform of the output signal.	Y	Y		Y	Y	Y	Y	Y	Y
Exercises										
Instead of a filter, implement the AM effect or vibrato effect, with sliders to adjust their parameters.										
Instead of a high-pass filter, implement a shelving filter, with sliders to adjust the cut-off frequency and shelf gain.										
Modify the play/pause programs to have a single button that toggles between play and pause.										