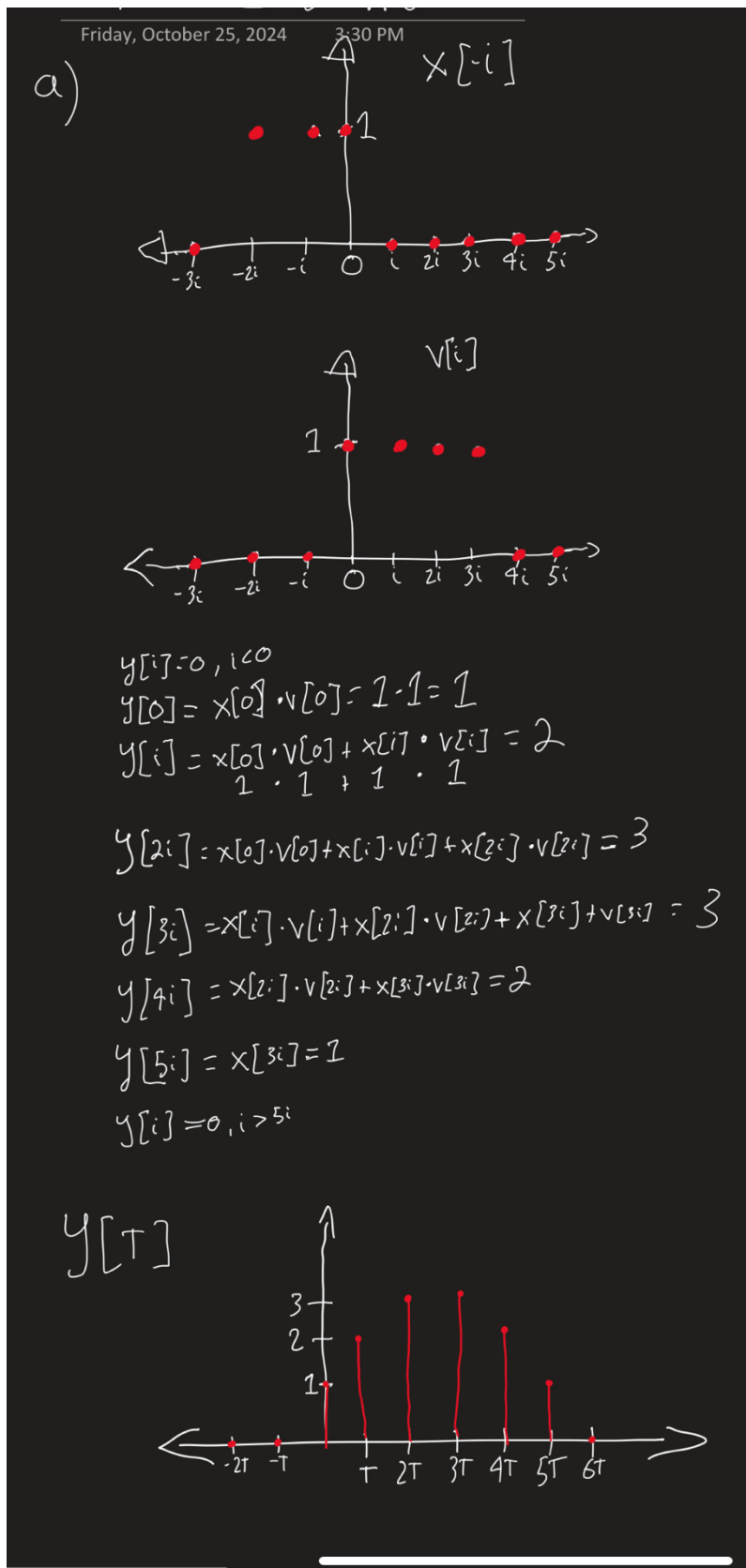
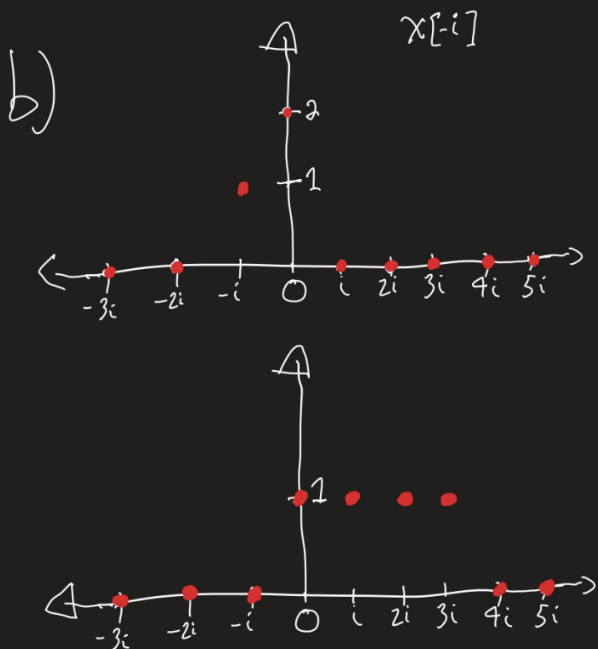


1. a) Manual computations of the convolutions of $x[n]$ and $v[n]$.





$$y[i] = 0, i < 0$$

$$y[0] = x[0] \cdot v[0] = 1 \cdot 2 = 2$$

$$y[i] = \underset{1}{x[0]} \cdot \underset{1}{v[i]} + \underset{2}{x[i]} \cdot \underset{1}{v[i]} = 3$$

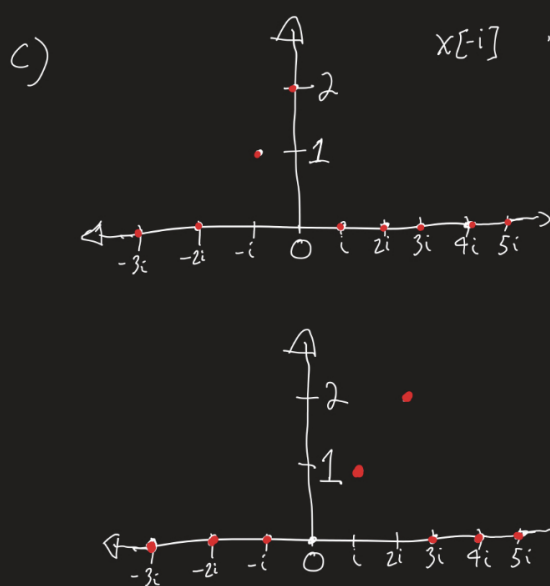
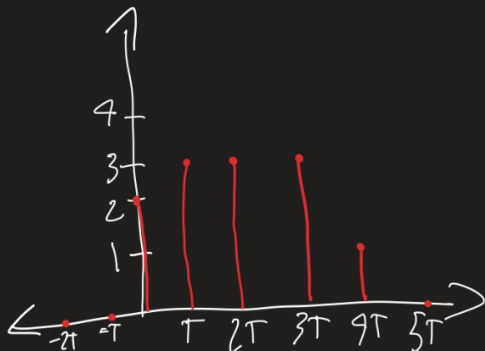
$$y[2i] = x[i] \cdot v[i] + x[2i] \cdot v[2i] = 3$$

$$y[3i] = x[2i] \cdot v[2i] + x[3i] \cdot v[3i] = 3$$

$$y[4i] = x[3i] \cdot v[3i] + \underset{0}{x[4i] \cdot v[4i]} = 1$$

$$y[i] = 0, i > 4i$$

$y[\tau]$



$$y[i] = 0, i < i$$

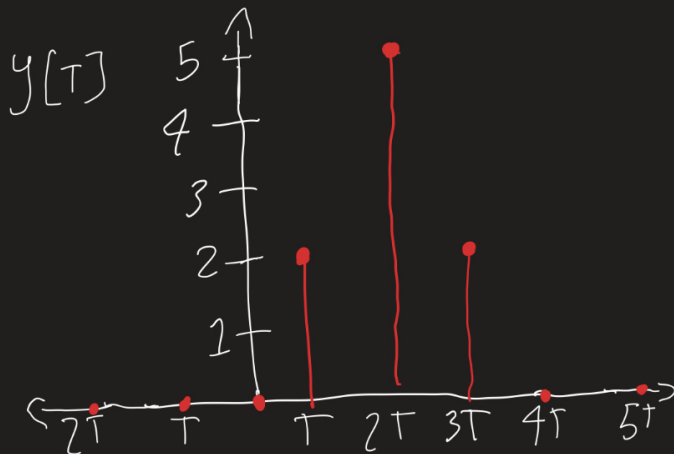
$$y[0] = x[0] \cdot \underset{0}{v[0]} = 0$$

$$y[i] = x[0] \cdot \underset{0}{v[i]} + \underset{1}{x[i]} \cdot v[i] = 2$$

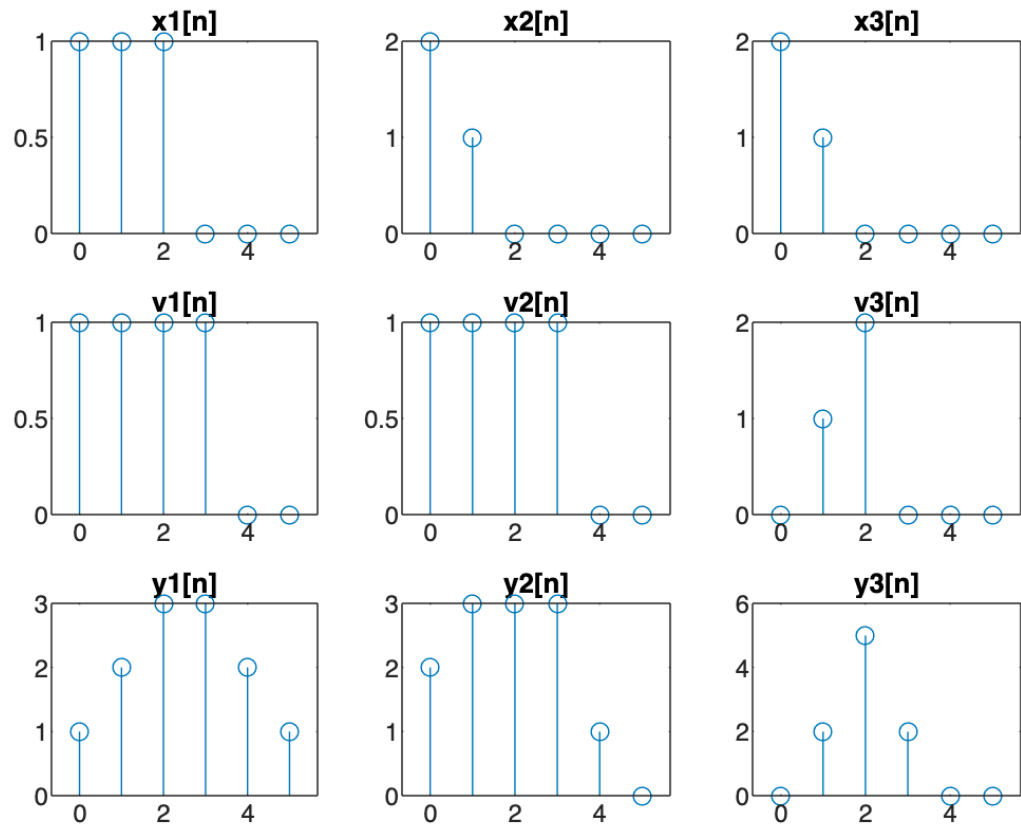
$$y[2i] = x[i] \cdot v[i] + x[2i] \cdot v[3i] = 5$$

$$y[3i] = x[2i] \cdot v[2i] = 2$$

$$y[i] = 0, i > 3i$$



b) Graphs of $x[n]$, $v[n]$, and $y[n]$



2. The highlighted document is our team's audio clip.

Today (12)



Lab2_3TP3_Q3.m



Lab2_3TP3_Q4.m



speechwithecho2.wav



speechwithecho.wav



Lab2_3TP3_Q5.m



speechwithecho1.wav



my_speech_clip.wav

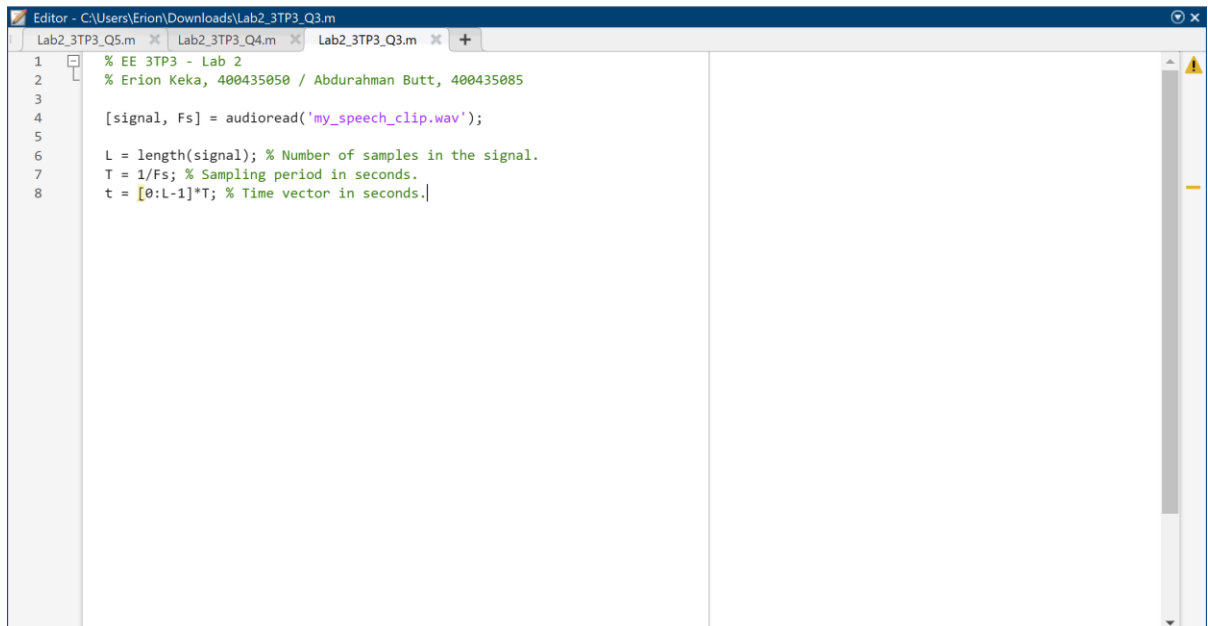


ocenaudio_windows64_3.14.6.exe



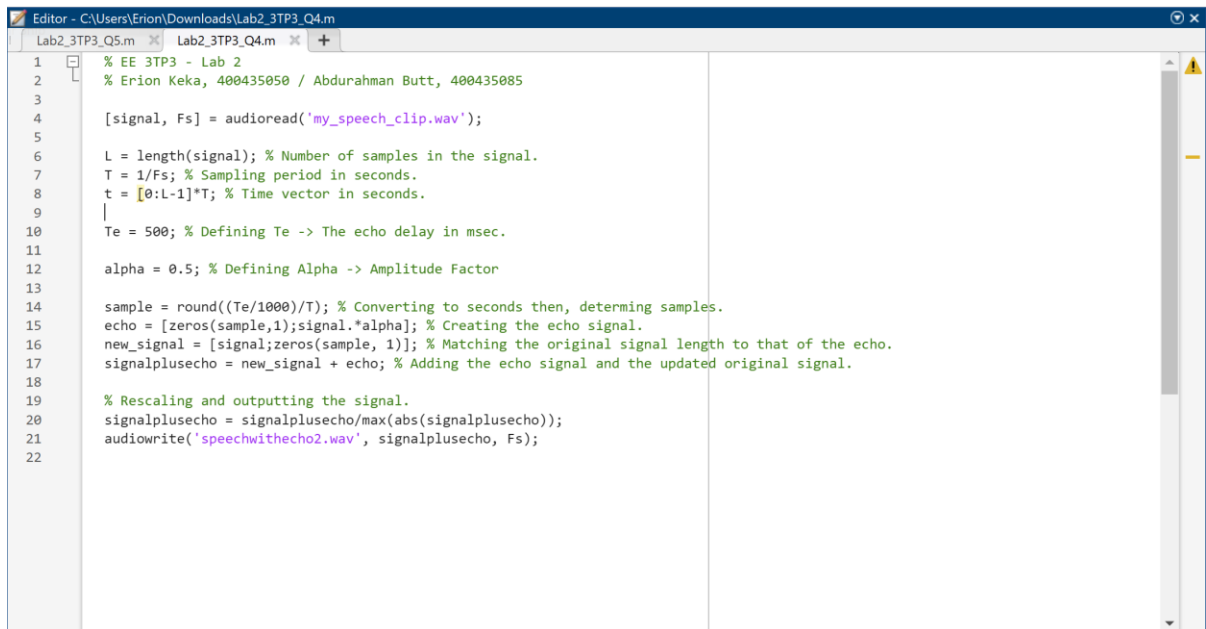
matlab_R2024b_Windows.exe

3. The following is the code from experiment three.



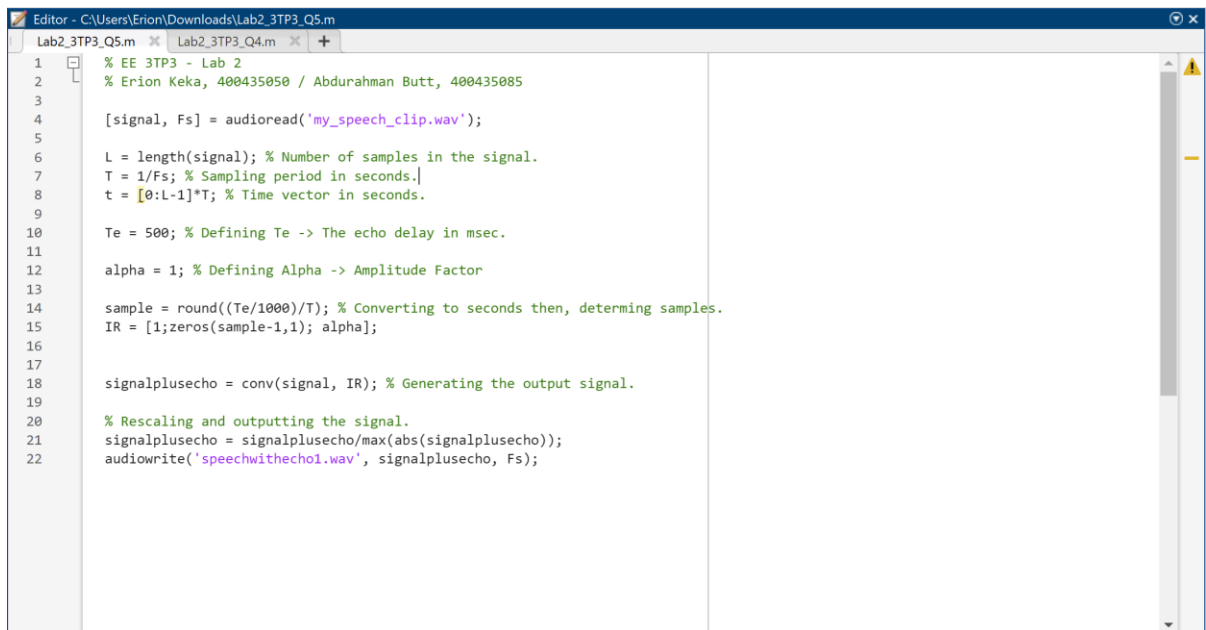
```
Editor - C:\Users\Erion\Downloads\Lab2_3TP3_Q3.m
Lab2_3TP3_Q5.m  Lab2_3TP3_Q4.m  Lab2_3TP3_Q3.m  +
1  % EE 3TP3 - Lab 2
2  % Erion Keka, 400435050 / Abdurahman Butt, 400435085
3
4  [signal, Fs] = audioread('my_speech_clip.wav');
5
6  L = length(signal); % Number of samples in the signal.
7  T = 1/Fs; % Sampling period in seconds.
8  t = [0:L-1]*T; % Time vector in seconds.
```

4. The following is the code from experiment four.

A screenshot of a MATLAB editor window titled 'Editor - C:\Users\Erion\Downloads\Lab2_3TP3_Q4.m'. The window contains a script with 22 lines of code. The code reads a speech file 'my_speech_clip.wav', calculates its length and sampling period, defines an echo delay Te = 500, and an amplitude factor alpha = 0.5. It then creates an echo signal by zero-padding the original signal and scaling it by alpha. The original signal and the scaled echo are added together to create 'signalplusecho', which is then rescaled and saved as 'speechwithecho2.wav'.

```
1 % EE 3TP3 - Lab 2
2 % Erion Keka, 400435050 / Abdurahman Butt, 400435085
3
4 [signal, Fs] = audioread('my_speech_clip.wav');
5
6 L = length(signal); % Number of samples in the signal.
7 T = 1/Fs; % Sampling period in seconds.
8 t = [0:L-1]*T; % Time vector in seconds.
9
10 Te = 500; % Defining Te -> The echo delay in msec.
11
12 alpha = 0.5; % Defining Alpha -> Amplitude Factor
13
14 sample = round((Te/1000)/T); % Converting to seconds then, determining samples.
15 echo = [zeros(sample,1);signal.*alpha]; % Creating the echo signal.
16 new_signal = [signal;zeros(sample, 1)]; % Matching the original signal length to that of the echo.
17 signalplusecho = new_signal + echo; % Adding the echo signal and the updated original signal.
18
19 % Rescaling and outputting the signal.
20 signalplusecho = signalplusecho/max(abs(signalplusecho));
21 audiowrite('speechwithecho2.wav', signalplusecho, Fs);
22
```

5. The following is the code from experiment five. Note: Alpha was set to 1 for Q6.

A screenshot of a MATLAB editor window titled 'Editor - C:\Users\Erion\Downloads\Lab2_3TP3_Q5.m'. The window contains a script with 22 lines of code. The code is similar to the previous one, but alpha is set to 1. Instead of zero-padding and adding, it uses the conv function to generate the output signal 'signalplusecho'. The signal is then rescaled and saved as 'speechwithecho1.wav'.

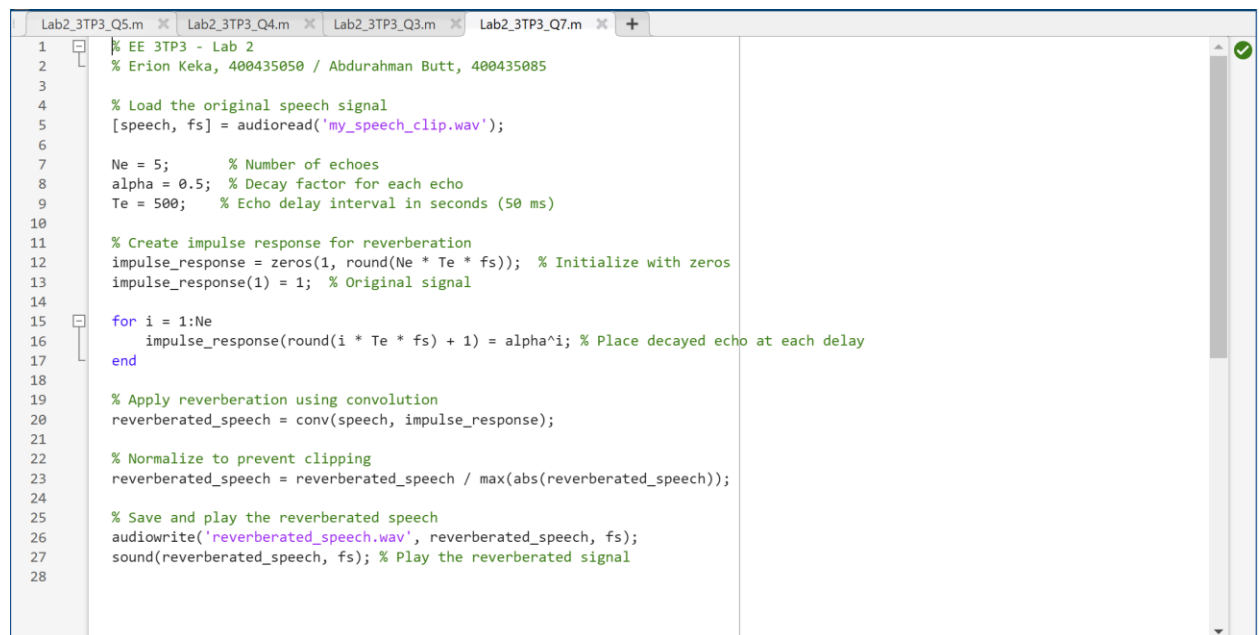
```
1 % EE 3TP3 - Lab 2
2 % Erion Keka, 400435050 / Abdurahman Butt, 400435085
3
4 [signal, Fs] = audioread('my_speech_clip.wav');
5
6 L = length(signal); % Number of samples in the signal.
7 T = 1/Fs; % Sampling period in seconds.
8 t = [0:L-1]*T; % Time vector in seconds.
9
10 Te = 500; % Defining Te -> The echo delay in msec.
11
12 alpha = 1; % Defining Alpha -> Amplitude Factor
13
14 sample = round((Te/1000)/T); % Converting to seconds then, determining samples.
15 IR = [1;zeros(sample-1,1); alpha];
16
17 signalplusecho = conv(signal, IR); % Generating the output signal.
18
19 % Rescaling and outputting the signal.
20 signalplusecho = signalplusecho/max(abs(signalplusecho));
21 audiowrite('speechwithecho1.wav', signalplusecho, Fs);
22
```

An impulse response with a value of 1 was chosen initially with zeros filling the rest until the sample is reached with the echo delay. Upon reaching the sample, there is an alpha value which is the amplitude of the waveform. This convolution method allows for the original sample to come first then, the second sample with the applied amplitude shifted onwards.

6. Upon fixing the value of alpha to 1 and experimenting with a variety of values for Te, we noticed that when Te becomes approximately 18, the quality of audio becomes

acceptable. The audio becomes acceptable at a T_e value of approximately 18 because this is the value in which the echo effect has been eliminated. The audio sounds acceptable as it is two waveforms placed on top of each other rather than one after the other. When the value of α is decreased, the value of T_e does increase for which the audio can be deemed acceptable. This is a result of the value of α being the amplitude of the waveform, thus by decreasing the value of α , we are decreasing the volume of the echo which allows for more headspace when changing the value of T_e .

7. The following is the code from experiment seven.

A screenshot of a MATLAB script editor window. The window has four tabs at the top: 'Lab2_3TP3_Q5.m', 'Lab2_3TP3_Q4.m', 'Lab2_3TP3_Q3.m', and 'Lab2_3TP3_Q7.m'. The active tab is 'Lab2_3TP3_Q7.m'. The script contains MATLAB code for creating a reverberated speech signal. It starts with a comment '% EE 3TP3 - Lab 2' and the author's name '% Erion Keka, 400435050 / Abdurahman Butt, 400435085'. The code loads a speech signal from 'my_speech_clip.wav'. It sets parameters: Ne = 5 (number of echoes), alpha = 0.5 (decay factor), and Te = 500 (echo delay interval in samples, which is 50 ms at 1000 Hz). It creates an impulse response of length round(Ne * Te * fs) + 1, initialized with zeros, and sets the first sample to 1. A for loop from i = 1 to Ne places decayed echoes at each delay. The reverberated speech is calculated using convolution. The signal is then normalized to prevent clipping. Finally, the reverberated speech is saved as 'reverberated_speech.wav' and played back. The script is 28 lines long.

```
1 % EE 3TP3 - Lab 2
2 % Erion Keka, 400435050 / Abdurahman Butt, 400435085
3
4 % Load the original speech signal
5 [speech, fs] = audioread('my_speech_clip.wav');
6
7 Ne = 5; % Number of echoes
8 alpha = 0.5; % Decay factor for each echo
9 Te = 500; % Echo delay interval in seconds (50 ms)
10
11 % Create impulse response for reverberation
12 impulse_response = zeros(1, round(Ne * Te * fs)); % Initialize with zeros
13 impulse_response(1) = 1; % Original signal
14
15 for i = 1:Ne
16     impulse_response(round(i * Te * fs) + 1) = alpha^i; % Place decayed echo at each delay
17 end
18
19 % Apply reverberation using convolution
20 reverberated_speech = conv(speech, impulse_response);
21
22 % Normalize to prevent clipping
23 reverberated_speech = reverberated_speech / max(abs(reverberated_speech));
24
25 % Save and play the reverberated speech
26 audiowrite('reverberated_speech.wav', reverberated_speech, fs);
27 sound(reverberated_speech, fs); % Play the reverberated signal
28
```

Upon repeating the step from experiment six with the reverberated speech, we notice that a T_e value of approximately 8 was the base point to allow for an acceptable audio clip.