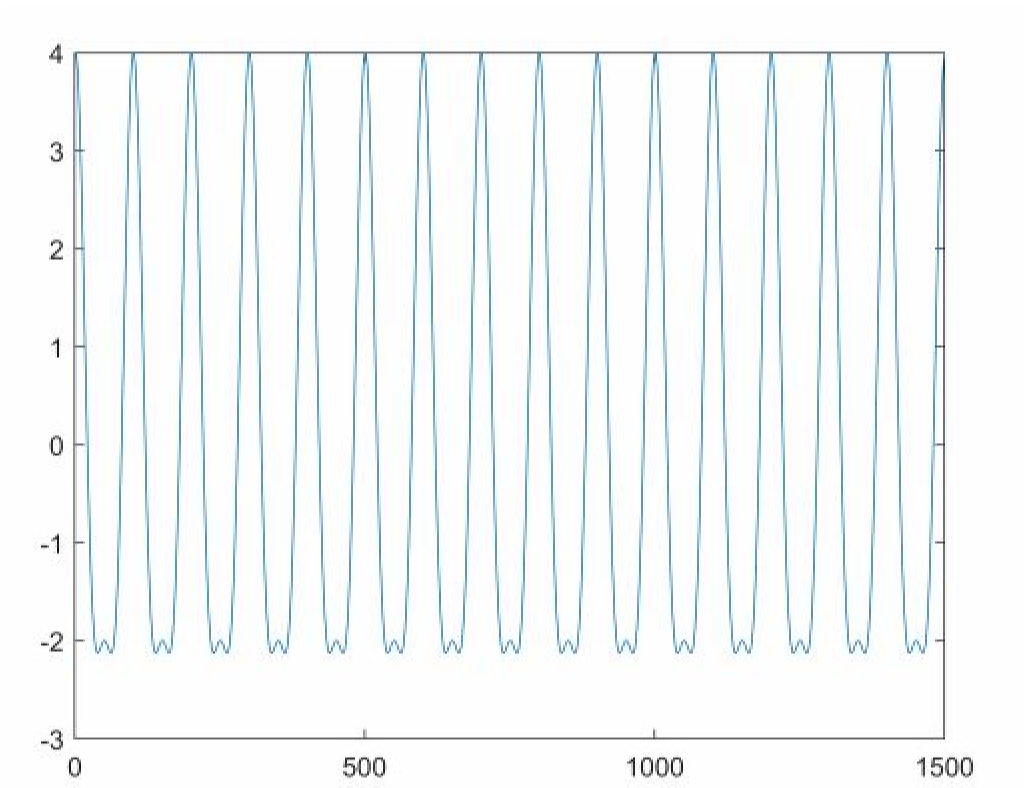
LAB REPORT - 1

Name: Ajay Choudhury
Roll no.: 18018
Experiment: Applications of Fourier Transform
Date: 20-01-2021

Solutions

Q. 1. a.

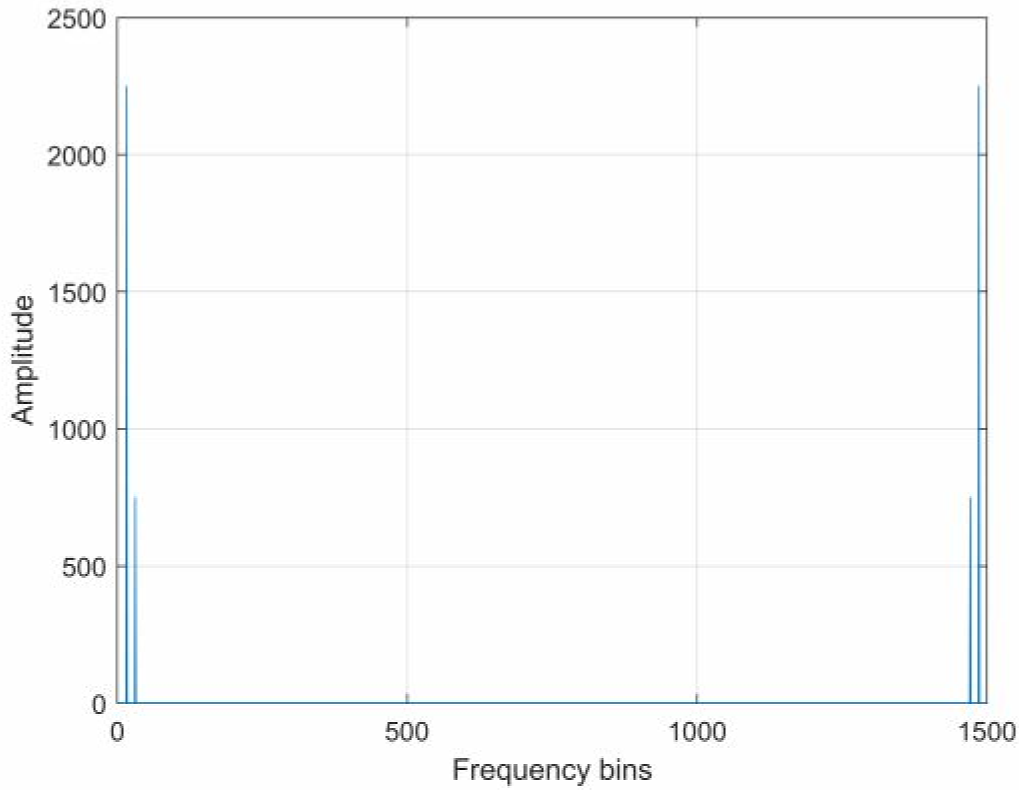
```
% Here the values of phases are taken as zero
fs = 1000;
t = 0 : 1/fs : 1.5 - 1/fs;
f1 = 10;
f2 = 20;
x = 3*cos(2*pi*f1*t) + 1*cos(2*pi*f2*t);
xlim('auto')
ylim('auto')
plot(x)
```



Q. 1. b.

```
% Fourier transform code:
y = fft(x);
length(x);
```

```
X_mag = abs(y);
plot(X_mag)
xlabel('Frequency bins')
ylabel('Amplitude')
xlim('auto')
grid on;
ylim('auto')
```



Interpretation: There are 2 frequencies present (let us consider the left half on the above graph for further inspection) and let us call the frequency on the right as F-low and the frequency on the left as F-high. Now, it can be seen from the above graph that the amplitude of F-low is 750Hz and the amplitue of F-high is 2250 Hz, which means that mathematically F-high = 3*F-low, which is true as the function s(t) which was used had 2 frequencies and the amplitude of one was 3 times the other (s = 3*cos(2*pi*f1*t) + 1*cos(2*pi*f2*t)). Therefore the phase values are reliable as well.

Q. 1. c.

```
X_phase = angle(y);
% The bin Number of F-high is 16, it can be seen after zooming into the
% above graph.
X_phase(16)
```

ans = 1.5940e-15

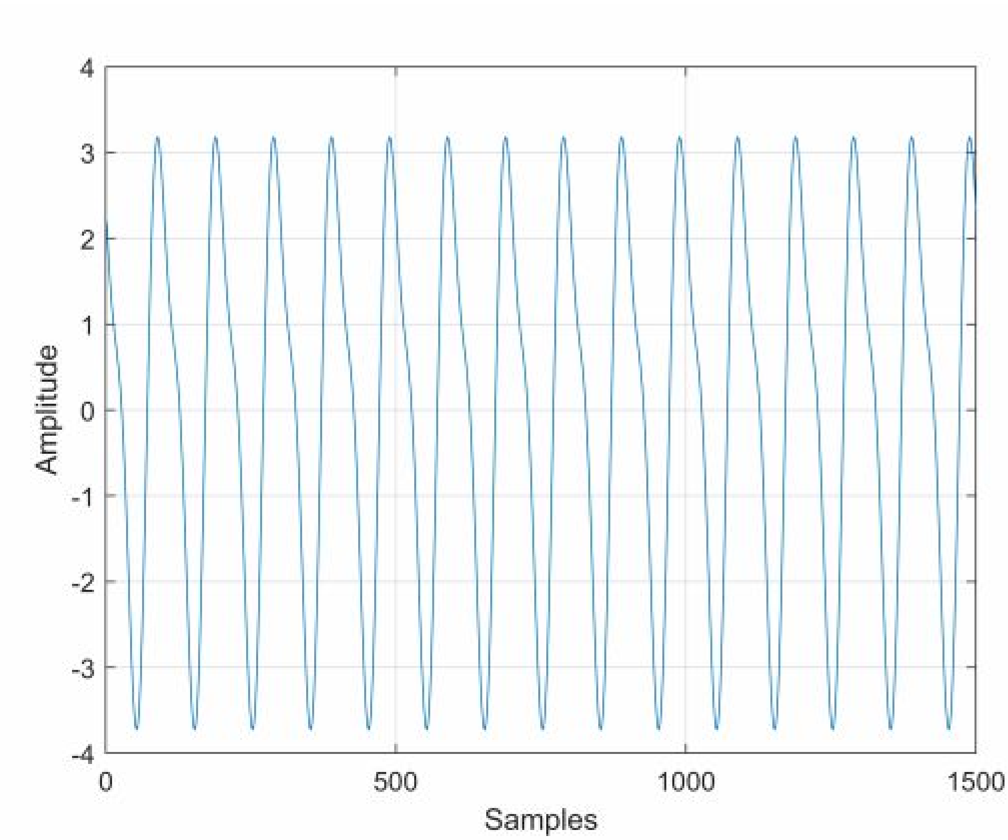
```
% The bin Number of F-low is 31, it can be seen after zooming into the
% above graph.
X_phase(31)
```

ans = 3.0133e-15

The values of X_phase(16) and X_phase(31) are too low and close to zero.

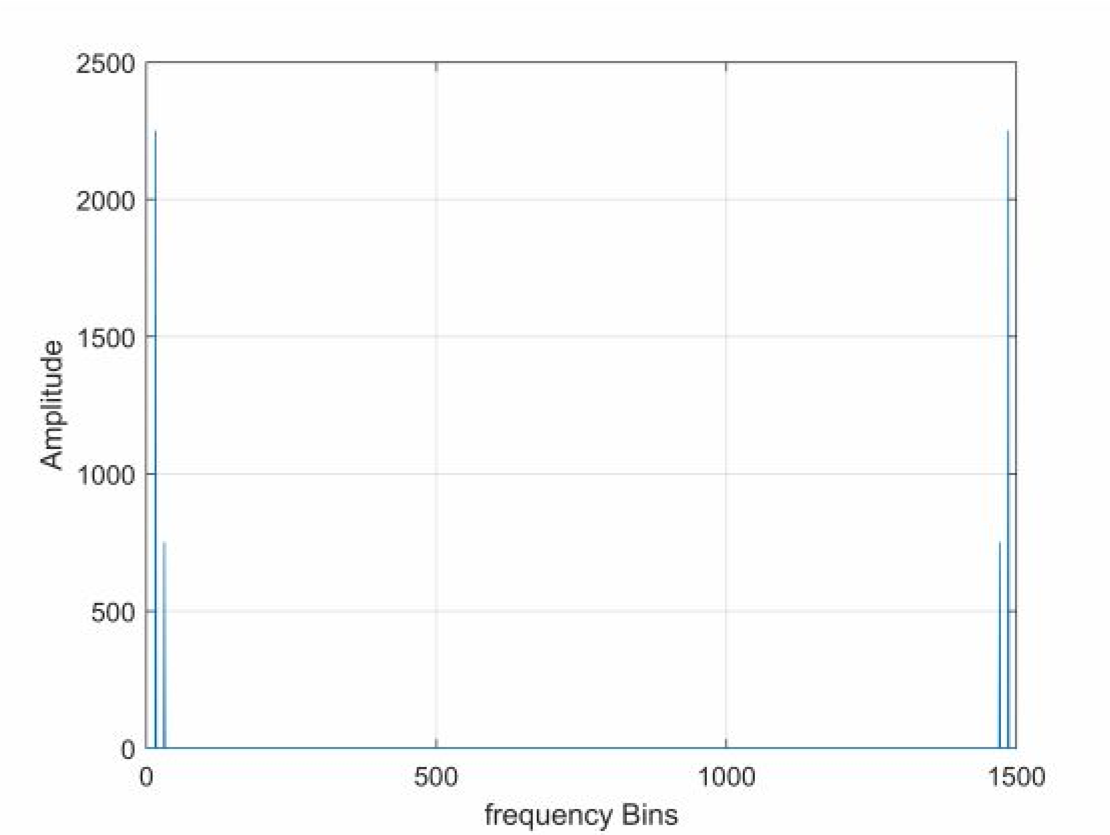
Q. 1. d.

```
% let us assume values of phases this time.
fs = 1000;
t = 0 : 1/fs : 1.5 - 1/fs;
f1 = 10;
f2 = 20;
x = 3*cos(2*pi*f1*t + 0.2) + 1*cos(2*pi*f2*t + 2.4);
plot(x)
xlabel('Samples')
ylabel('Amplitude')
xlim('auto')
grid on;
ylim('auto')
```




```
% Fourier transform code:
y = fft(x);
```

```
length(x);
X_mag = abs(y);
plot(X_mag)
xlabel('frequency Bins')
ylabel('Amplitude')
xlim('auto')
grid on;
ylim('auto')
```




```
X_phase = angle(y);
% The bin Number of F-high is 16, it can be seen after zooming into the
% above graph.
X_phase(16)
```

ans = 0.2000

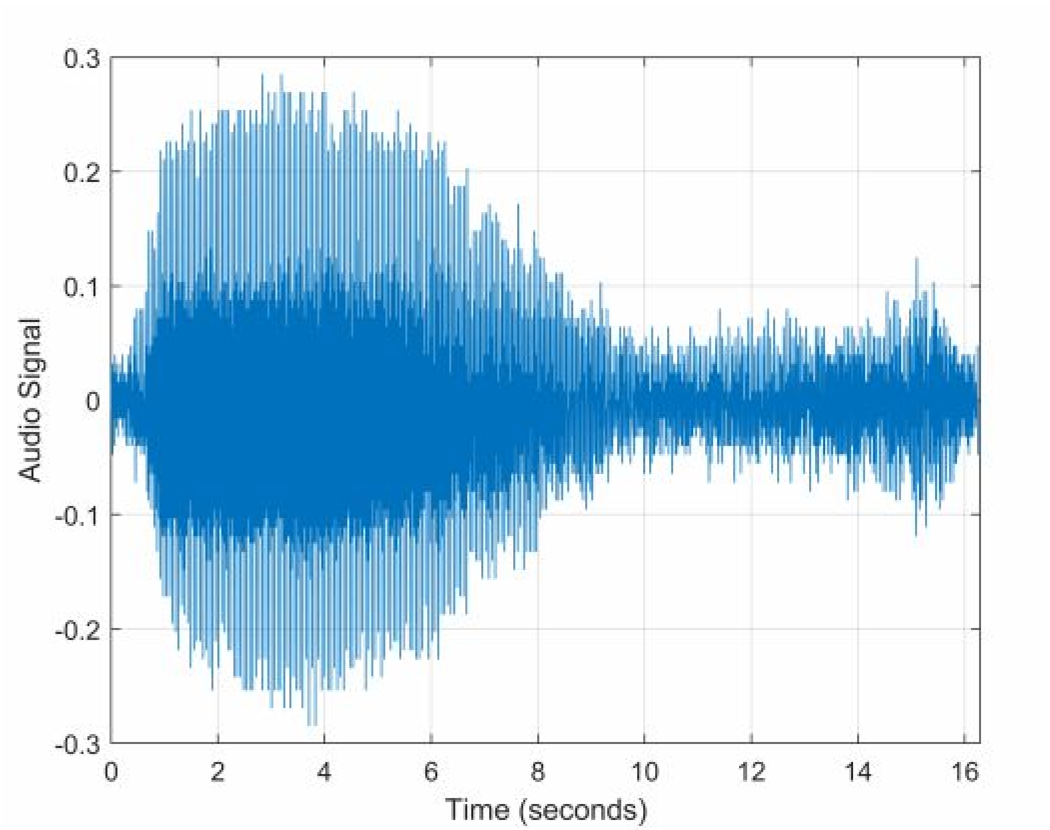
```
% The bin Number of F-low is 31, it can be seen after zooming into the
% above graph.
X_phase(31)
```

ans = 2.4000

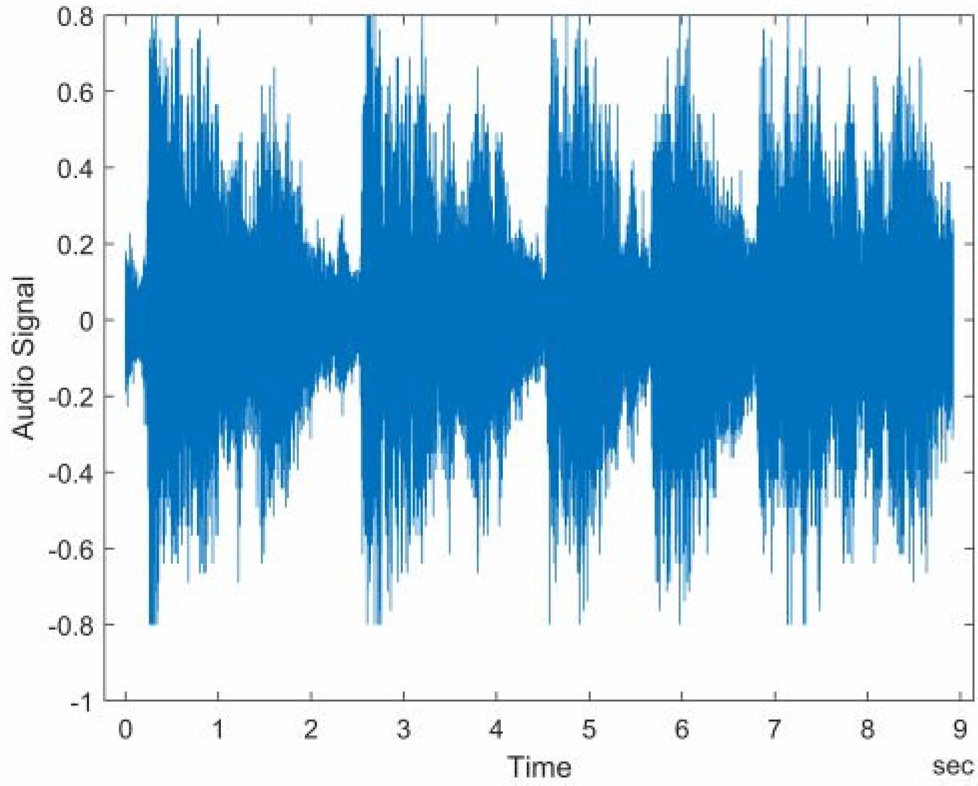
Now we can see the different value of phase.

Q. 2. a.

```
% we will now import 2 audio signals
% importing audio sample file 1
whaleFile = 'bluewhale.au';
[~,fs] = audioread(whaleFile);
t = 10*(0:1/fs:(length(whaleMoan)-1)/fs);
plot(t,whaleMoan)
xlabel('Time (seconds)')
ylabel('Audio Signal')
xlim([0 t(end)])
xlim([0.0 16.3])
grid on;
ylim('auto')
```




```
% importing audio sample file 2
handel = 'handel.wav';
[y,fs] = audioread(handel);
t = 0:seconds(1/fs):seconds(info.Duration);
t = t(1:end-1);
plot(t,y)
xlabel('Time')
ylabel('Audio Signal')
```

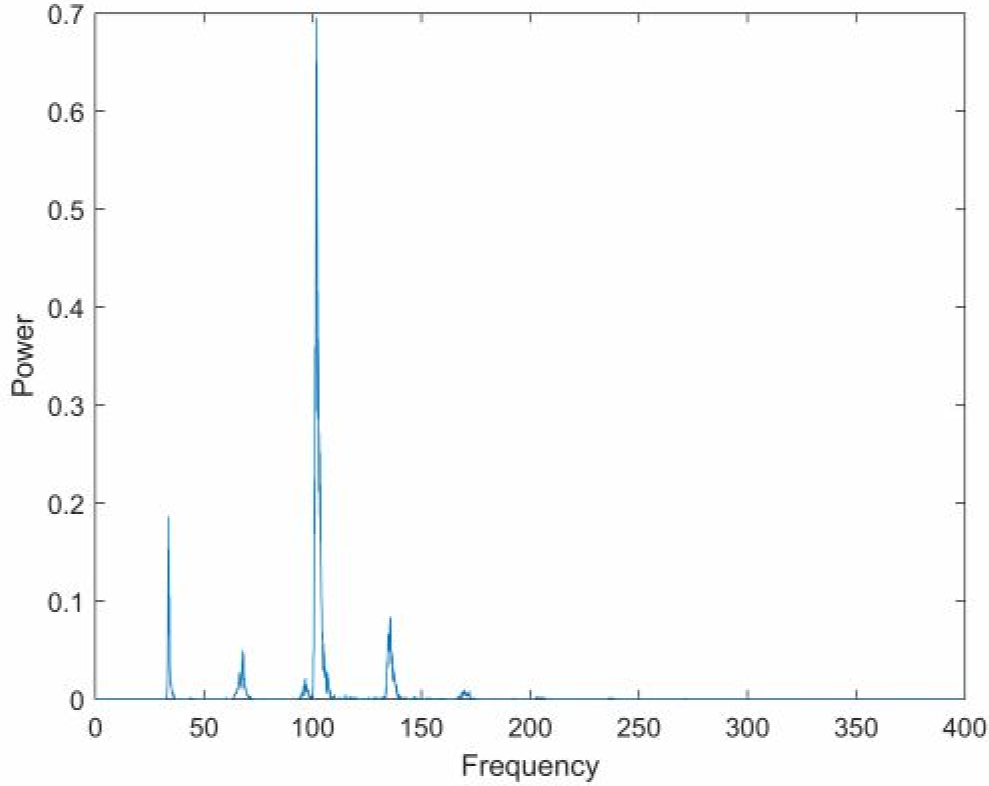



```
% importing audio sample file 3
[z,fs] = audioread('D:\ALL SEMESTERS\Semester VI\ECS 330 - Lab II\recording.wav');
```

Q. 2. b.

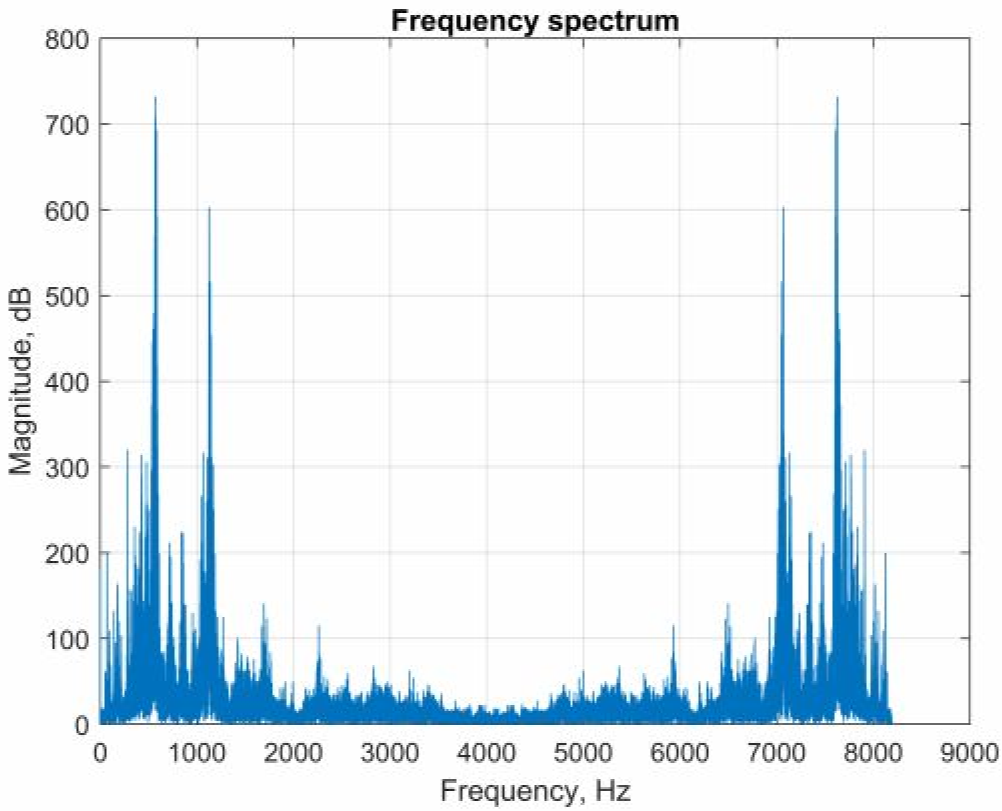
We will try to determine 5 prominent frequencies of both audio signals.

```
% For Audio sample file - 1
m = length(whaleMoan);
n = pow2(nextpow2(m));
y = fft(whaleMoan,n);
% frequency vector
f = (0:n-1)*(fs/n)/10;
% power spectrum
power = abs(y).^2/n;
plot(f(1:floor(n/2)),power(1:floor(n/2)))
xlabel('Frequency')
ylabel('Power')
```



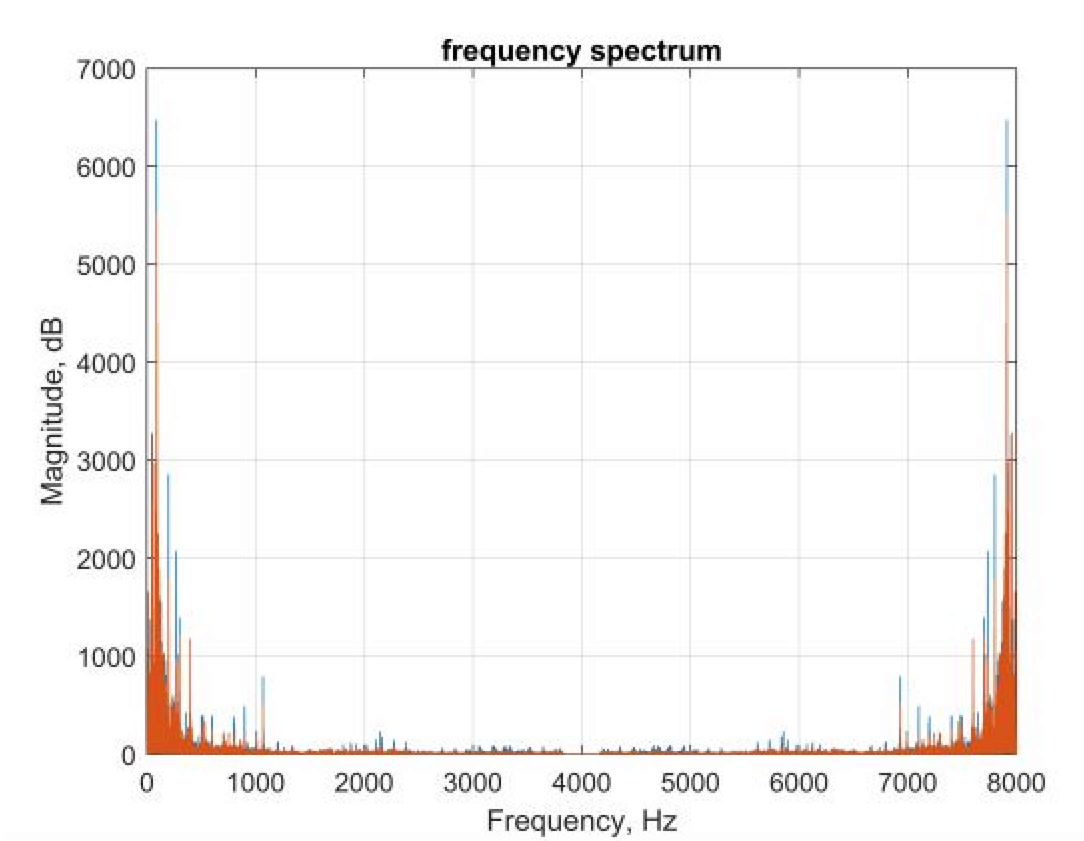
The 5 prominent frequencies of the above audio file are: 34.6Hz, 69.4Hz, 104Hz, 104.3Hz and 139Hz.

```
% For Audio sample file - 2
[y,fs] = audioread(handel);
N = length(y); % Length of vector y, number of samples
Y = fft(y,N); % Fourier transform of y
F = ((0:1/N:1-1/N)*fs); % Frequency vector
magnitudeY = abs(Y); % Magnitude of the FFT
figure (1);
plot(F, magnitudeY);
grid on;
xlabel('Frequency, Hz');
ylabel('Magnitude, dB');
xlim('auto')
ylim('auto');
title('Frequency spectrum');
```



The 5 prominent frequencies of the above audio file are: 567Hz, 1124.5Hz, 7067.5Hz, 7625Hz and 7908Hz.

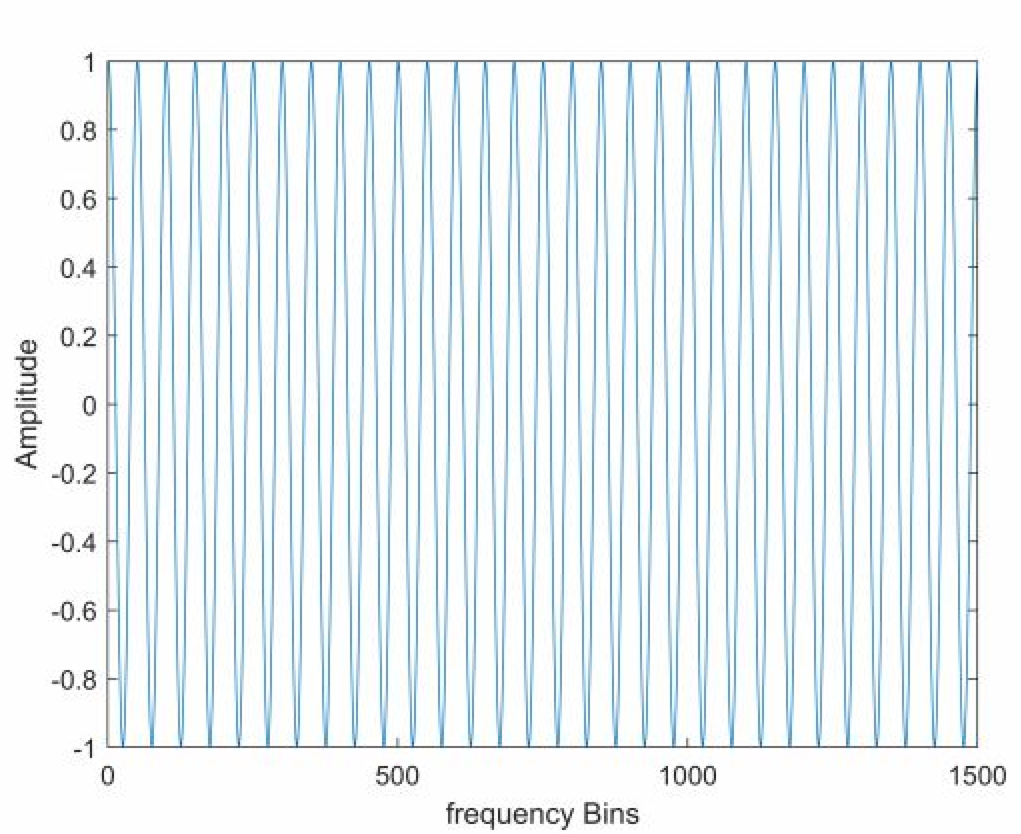
```
% For Audio sample file - 3
[y,fs] = audioread('D:\ALL SEMESTERS\Semester VI\ECS 330 - Lab II\recording.wav');
N = length(y); % Length of vector y, number of samples
Y = fft(y,N); % Fourier transform of y
F = ((0:1/N:1-1/N)*fs); % Frequency vector
magnitudeY = abs(Y); % Magnitude of the FFT
figure (1);
plot(F, magnitudeY);
grid on;
xlabel('Frequency, Hz');
ylabel('Magnitude, dB');
title('frequency spectrum');
xlim('auto')
ylim('auto')
```



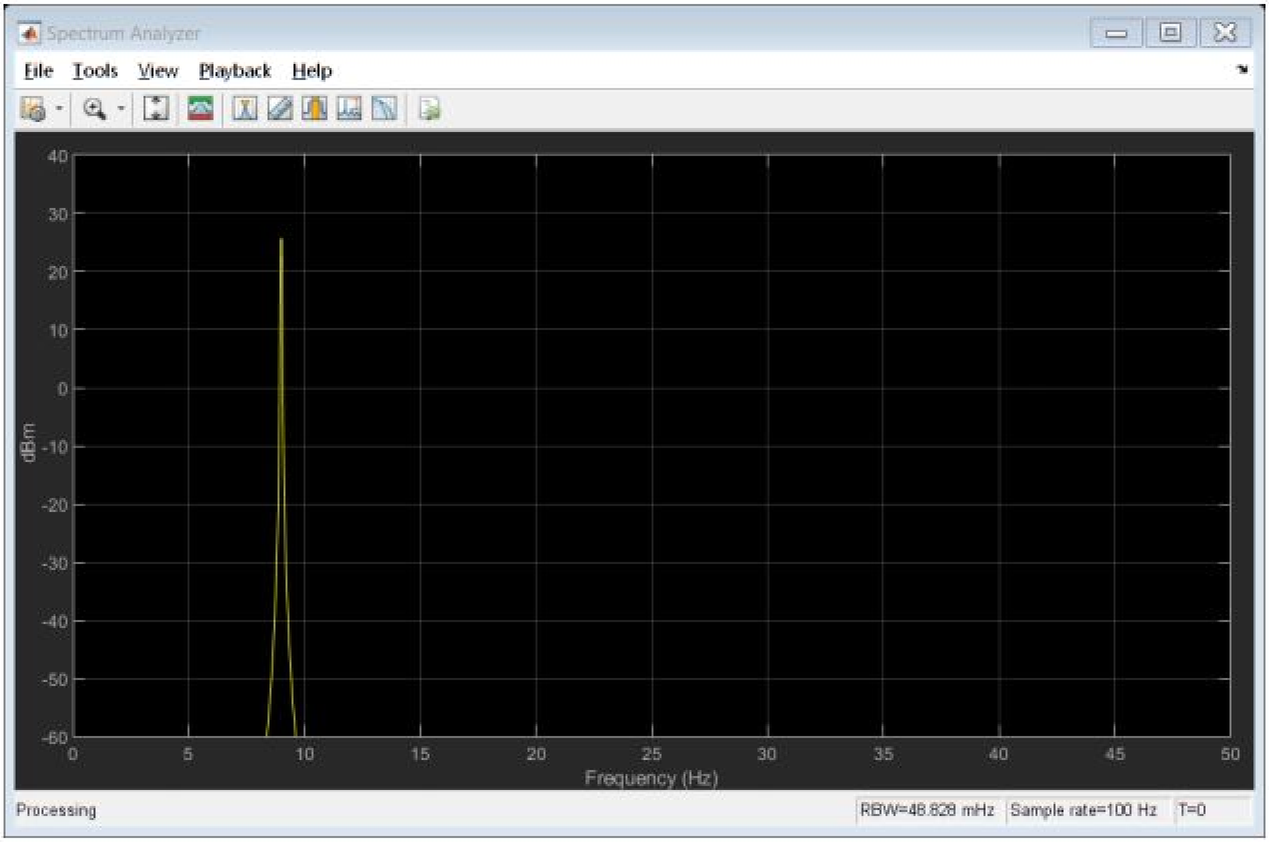
The 5 prominent frequencies of the above audio file are: 88.72Hz, 199.6Hz, 6931.4Hz, 7800Hz and 7911,2Hz.

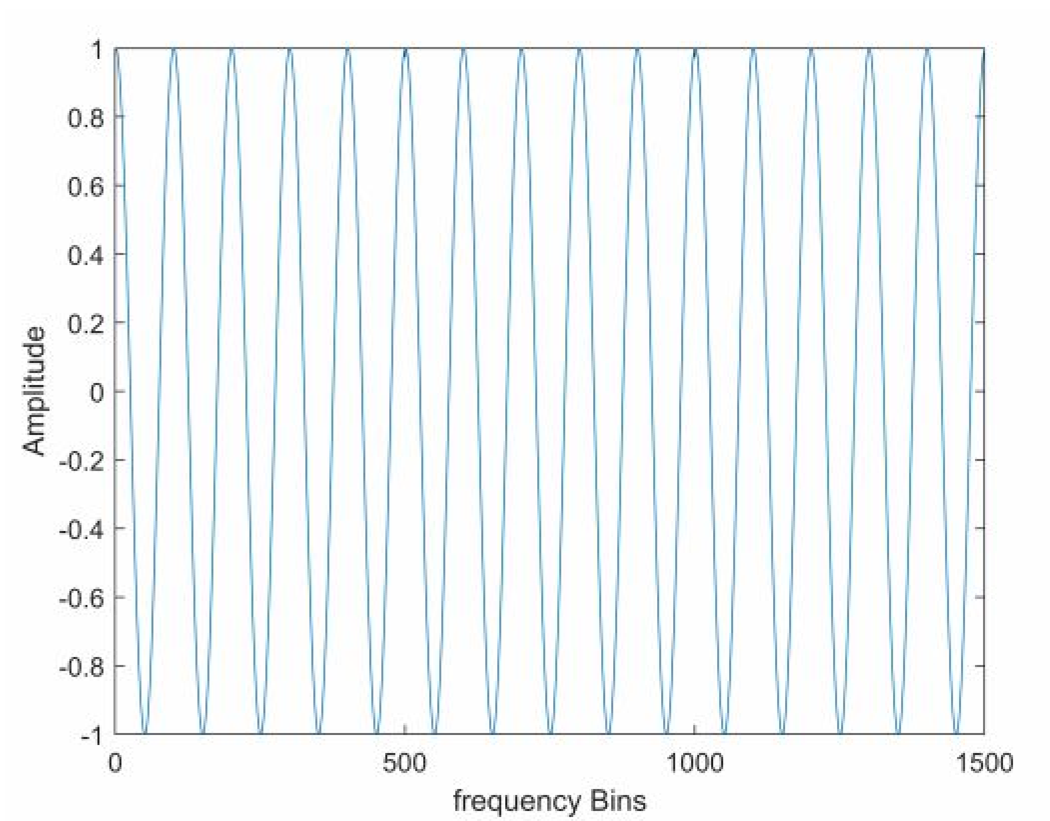
Q. 3. a.

```
% signal plot:
fs = 1000;
t = 0 : 1/fs : 1.5 - 1/fs;
fm = 10; \
fc = 20;
mt = cos(2*pi*fm*t); % message signal
ct = cos(2*pi*fc*t); % carrier wave
plot(ct)
xlabel('frequency Bins')
ylabel('Amplitude')
```




```
plot(mt)
xlabel('frequency Bins')
ylabel('Amplitude')
% Amplitude Modulation
fs = 100;
t = (0:1/fs:100)';
fc = 10;
x = sin(2*pi*t);
ydouble = ammod(x,fc,fs);
ysingle = ssbmod(x,fc,fs);
sa = dsp.SpectrumAnalyzer('SampleRate',fs,
... 'PlotAsTwoSidedSpectrum',false,
... 'YLimits',[-60 40]);
step(sa,ysingle)
```



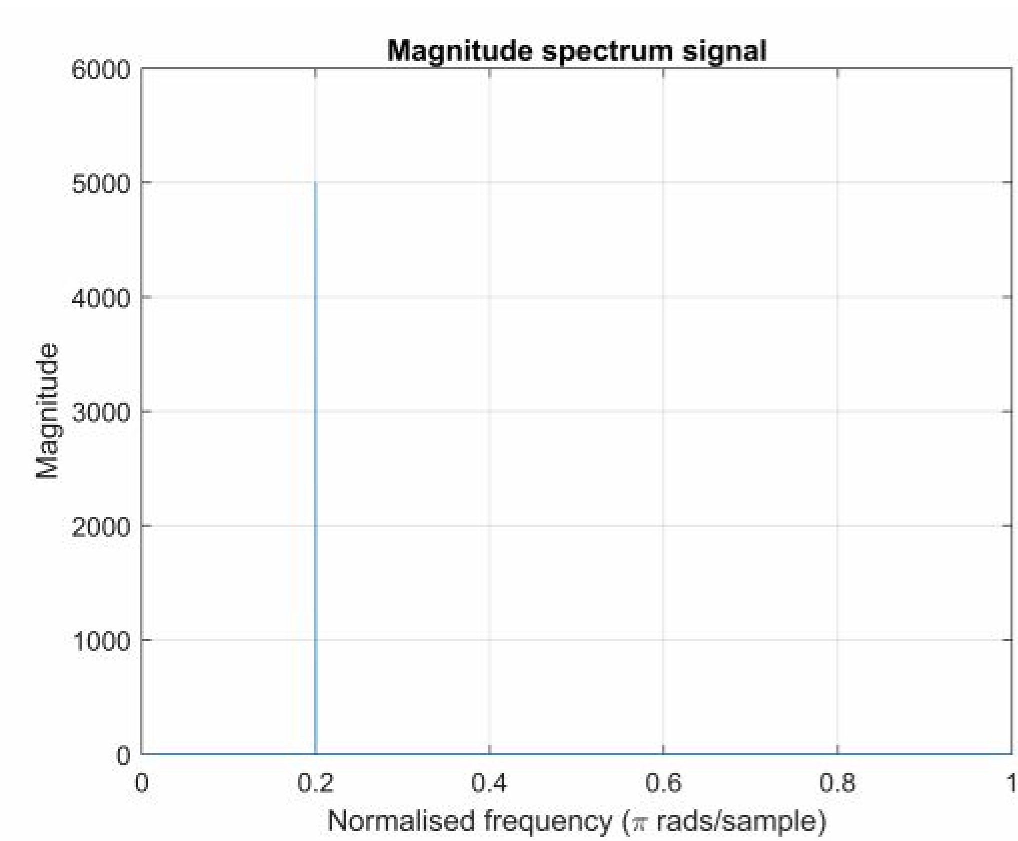



```
step(sa,ydouble)
```

I am Unable to plot both Step double and single in a single report.

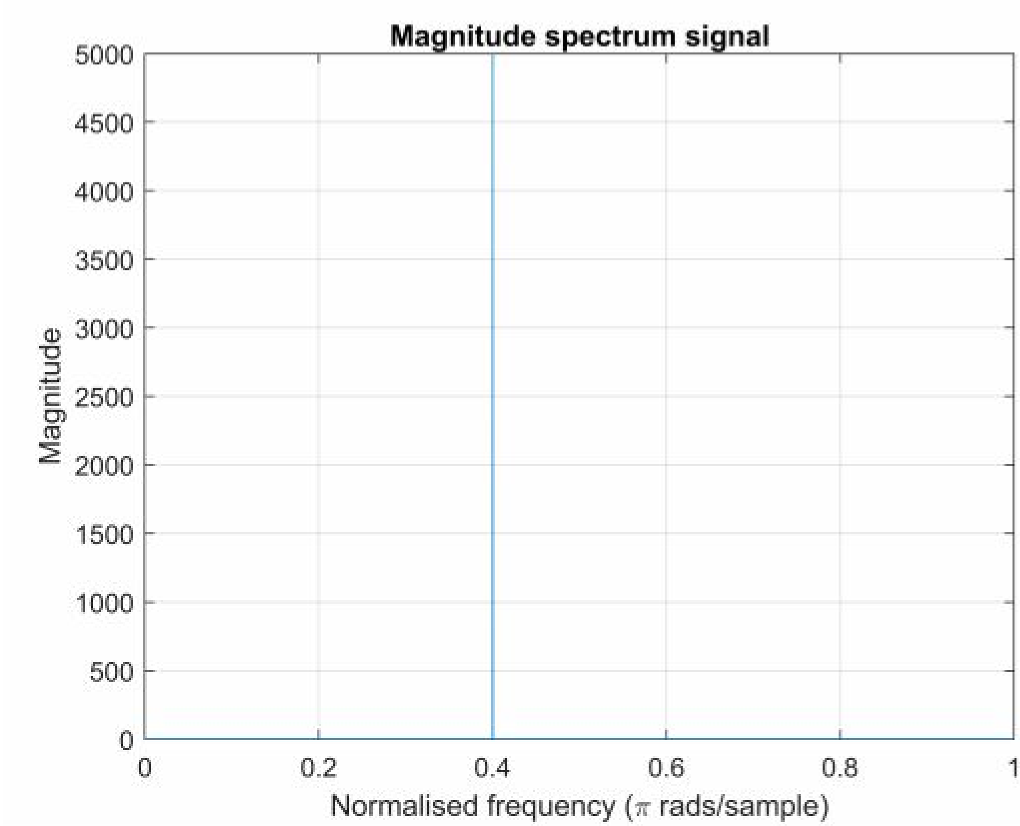
Q. 3. b.

```
% Magnitude spectrum
fm = 10;
fc = 20;
x1 = cos(2*pi*fm*t);
x2 = cos(2*pi*fc*t);
x1(end) = [];
x2(end) = [];
%plot first half of DFT (normalised frequency)
Y_mags1 = abs(fft(x1));
num_bins1 = length(Y_mags1);
Y_mags2 = abs(fft(x2));
num_bins2 = length(Y_mags2);
plot((0:1/(num_bins1/2 -1):1), Y_mags1(1:num_bins1/2));
grid on;
title('Magnitude spectrum signal');
xlabel('Normalised frequency (\pi rads/sample)');
ylabel('Magnitude');
```



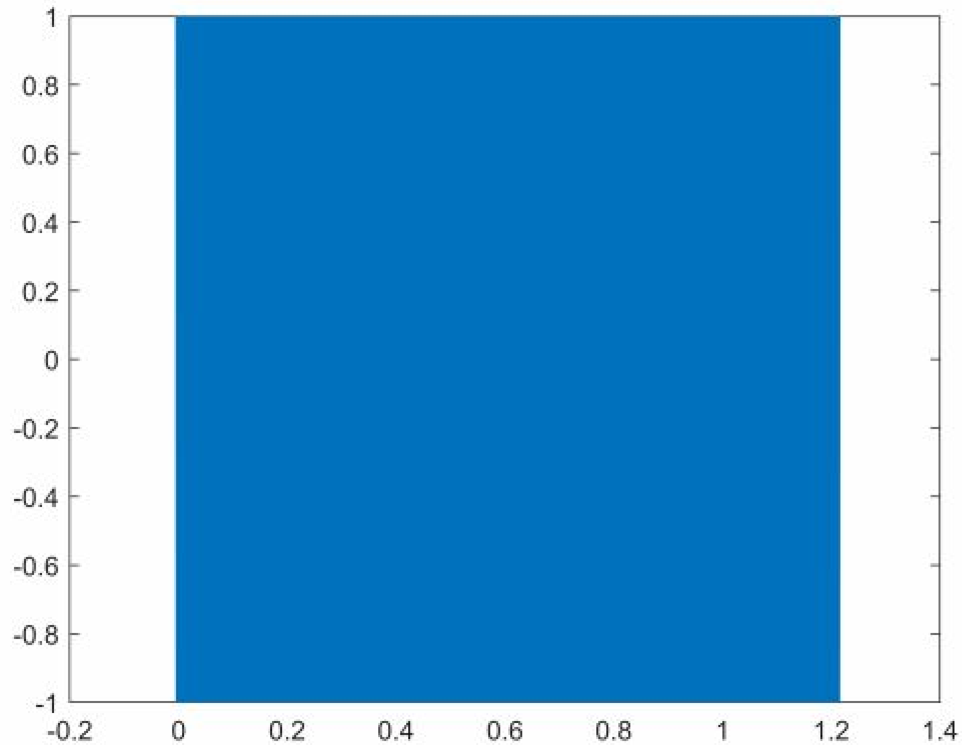

```
plot((0:1/(num_bins2/2 -1):1), Y_mags2(1:num_bins2/2)),
grid on;
title('Magnitude spectrum signal');
```

```
xlabel('Normalised frequency (\pi rads/sample)');
ylabel('Magnitude');
```

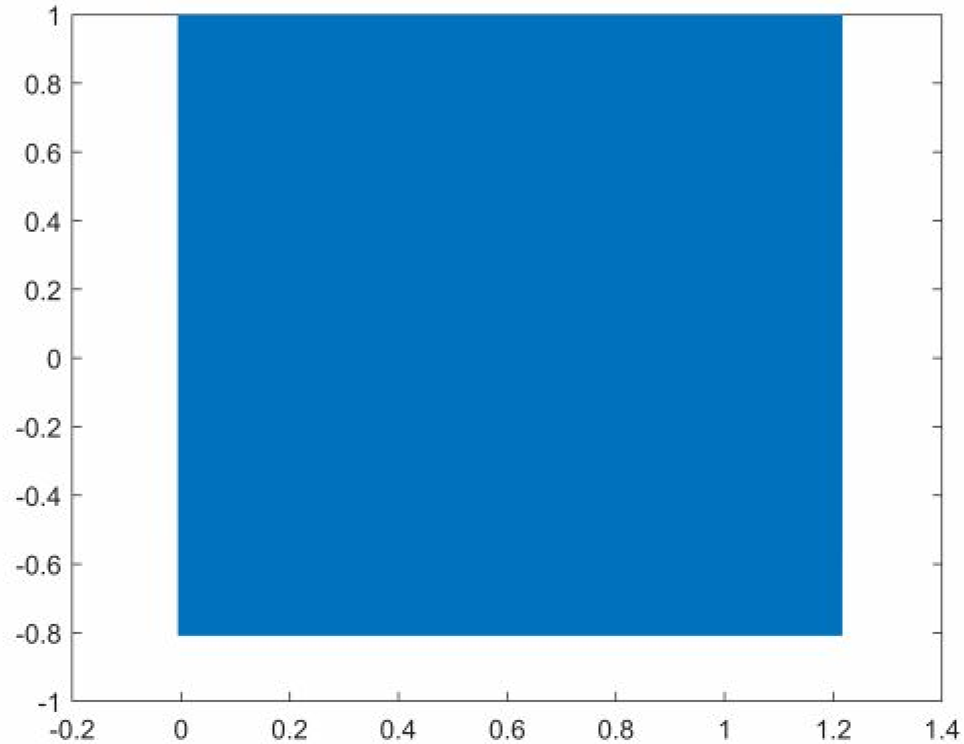


Q. 3. c.

```
% Time domain
fm = 10;
fc = 20;
x1 = cos(2*pi*fm*t);
x2 = cos(2*pi*fc*t);
tSampling=1/Fs;
n1=numel(x1);
t=-0.005:tSampling:-0.005+(n1-1)*tSampling;
plot(t,x1);
```




```
tSampling=1/Fs;
n2=numel(x2);
t=-0.005:tSampling:-0.005+(n2-1)*tSampling;
plot(t,x2);
```


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4