Digital signal processing Lab
Experiment-1
Sampling

a)Sampling of a sinusoidal waveform:-

Problem statement:-

- To see the effect of N(number of samples of analog signal taken) of a sampled signal which is sampled at a frequency Fs.
- Analog signal: x(t) = 10*cos(2*pi*1000*t) + 6*cos(2*pi*2000*t) + 2*cos(2*pi*4000*t)
- Fs=12Khz;
- Obtain DFT of x(t) with N=12,64,128,256.
- This signal has Fm=4KHz,so the Nyquist rate of this signal is 8KHz.

Note:-In the matlab we can't get DFT of the signal as it has infinite values,so here we can only get FFT of the signal(which is the repeating part of the DFT over infinite range).

Results:-

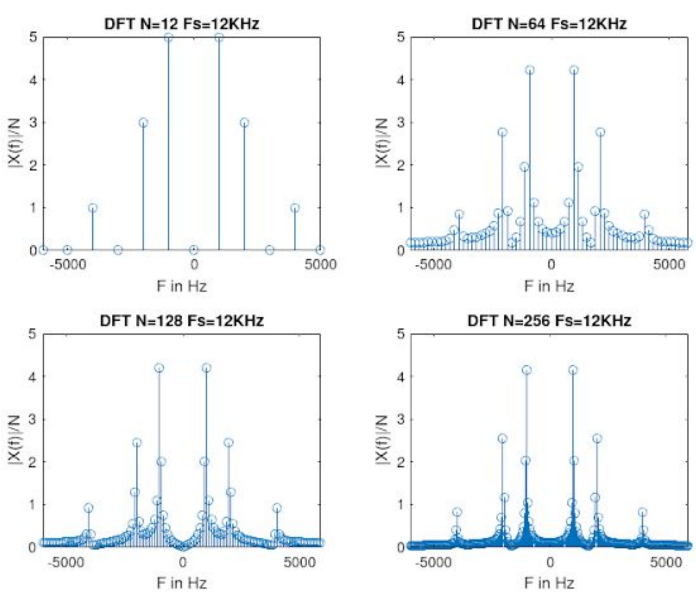

Fig.a

- Above 4 subplots depicts the part of the DFT(from F=-5KHz to F=5KHz) centered at frequency=0 of the analog signal x(t) with different number of samples (N) i.e.,different signal length.

Observation & Discussion:-

- As we can see from plots with varying N,the sampling of the fft of the signal also gets varied which can also be evident from the formulae for fft of the signal(it has only N values where N is the number of samples of the analog signal we have taken).
- Even though our analog signal has 3 frequencies but the FFT of the signal has infinite frequencies,which is because we have taken only a truncated part of our signal,i.e., windowed our signal with rectangular function, as rectangular function has infinite frequencies.so, also here we got the infinite frequencies in our FFT.
- Higher the N the better the FFT gets which is because it turns into a more original signal with increasing our N(i.e.,increasing the size of the window of the rectangular function i.e., increasing the signal length).
- We can also see that |X(f)|/N is half of the amplitude of the signal at a particular frequency,because the signal consists of positive and negative parts of frequencies, sothe amplitude gets distributed half at positive and negative frequencies of the signal.

b) Sampling at below nyquist rate and effect of aliasing:-

Problem statement:-

- To see the aliases of original frequencies when the signal is sampled below the nyquist rate.
- Repeat the above part with diff values of Fs=8,5,4 KHz.

Results:-

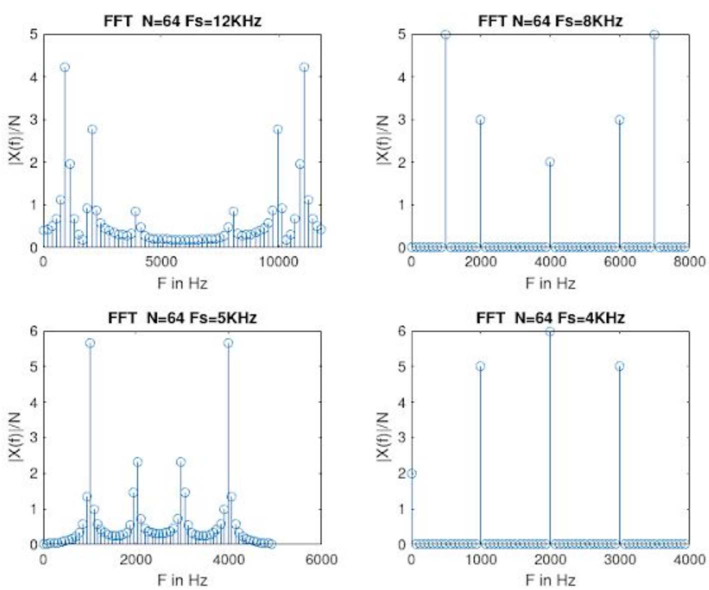

Fig.b_1

- Above 4 subplots depict the FFT of the signal with N=64,with different values of sampling frequency which are below the nyquist rate.

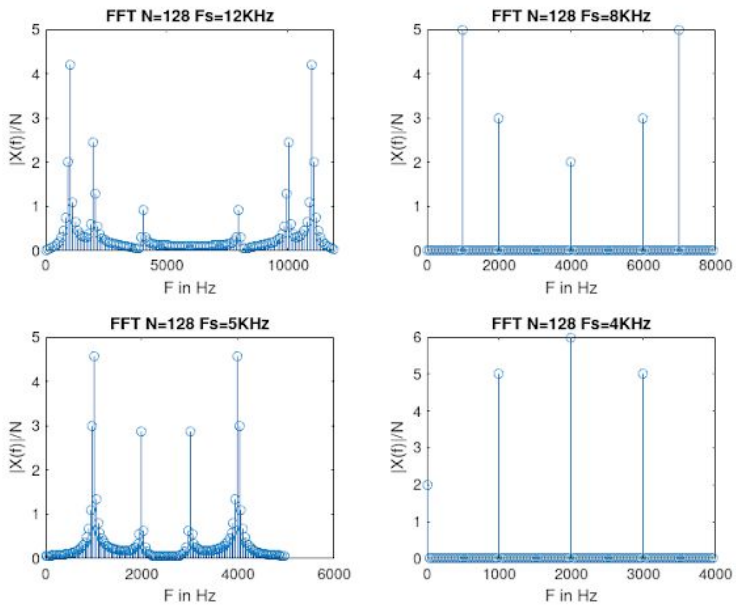

Fig.b_2

- Above 4 subplots depict the FFT of the signal with N=128,with different values of sampling frequency which are below the nyquist rate.

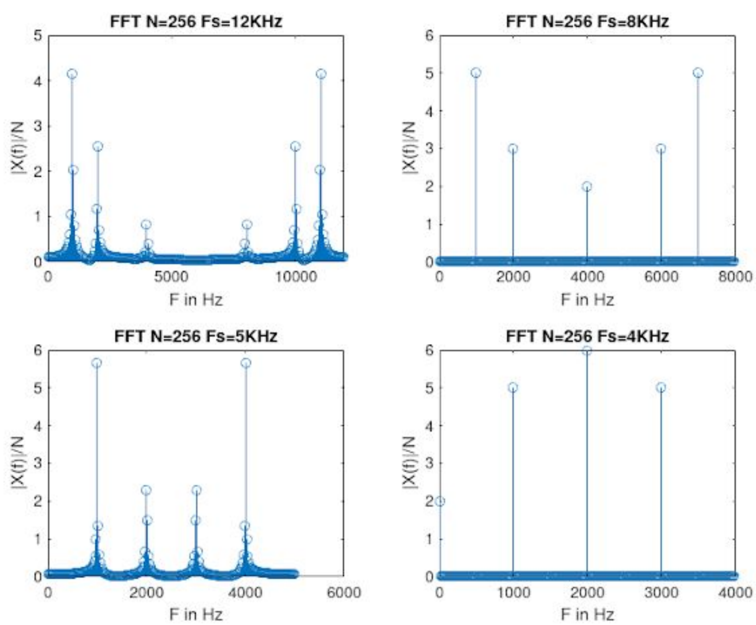

Fig.b_3

- Above 4 subplots depict the FFT of the signal with N=256,with different values of sampling frequency which are below the nyquist rate.

Observation & Discussion:-

- With sampling frequencies below nyquist rate we get different aliasing frequencies arising in the FFT of the signal:
  - For Fs=12KHz,there are no aliasing frequencies as it is higher than the nyquist rate.
  - For Fs=8KHz,there are no aliasing frequencies as it is equal to the nyquist rate.
  - For Fs=5KHz,From the plots we can observe that F=3KHz(this is due to the 2KHZ part in the signal) is an aliasing frequency,i.e., there is a peak at frequency=3KHz.
  - For Fs=4KHz,F1=0 KHz(this is due to the 4KHz part in the signal) and F2=3KHz(this is due to the 1KHz part in the signal) are the aliasing frequencies of the original frequencies.
- We can also observe there is change in the peak values(|X(f)|/N) at different frequencies.
  - For Fs=8KHz in FFT the value |X(f)|/N at diff frequencies:-
    - At 1KHz it is 5,which is the same as when Fs=12KHz(above nyquist rate).
    - At 2KHz it is 3,which is the same as when Fs=12KHz(above nyquist rate).
    - At 4KHz it is 2,which is double the value when Fs=12KHz,due to the frequency part 4KHz.
  - For Fs=5KHz in FFt the value |X(f)|/N at diff frequencies:-
    - At 1KHz it is 6=5+1,an effect due to the 1KHz and 4KHz frequency components in the signal.
    - At 2KHz it is 3,same as when Fs=12KHz.
    - At 3KHz it is 3,an effect due to the 2KHz frequency component in the signal.
    - At 4KHz it is 6=5+1,an effect due to the 4KHz and 1KHz frequency components in the signal.
  - For Fs=4KHz in FFt the value |X(f)|/N at diff frequencies:-
    - At 0KHz it is 2=1+1,an effect due to the 4KHz frequency component in the signal.
    - At 1KHz it is 5,same as when Fs=12KHz.
    - At 2KHz it is 6=3+3,an effect due to the 2KHz frequency component in the signal.
    - At 3KHz it is 5,an effect due to the 1KHz frequency component in the signal.
    - At 4KHz it is 2=1+1,which is double the value when Fs=12KHz,due to the frequency part 4KHz.

c) Spectrum of the square wave:-

Problem statement:-

- To obtain the DFT of the sampled square wave with N=256,Fs=20KHz,where the time period of the square wave is 1ms.

Results:-

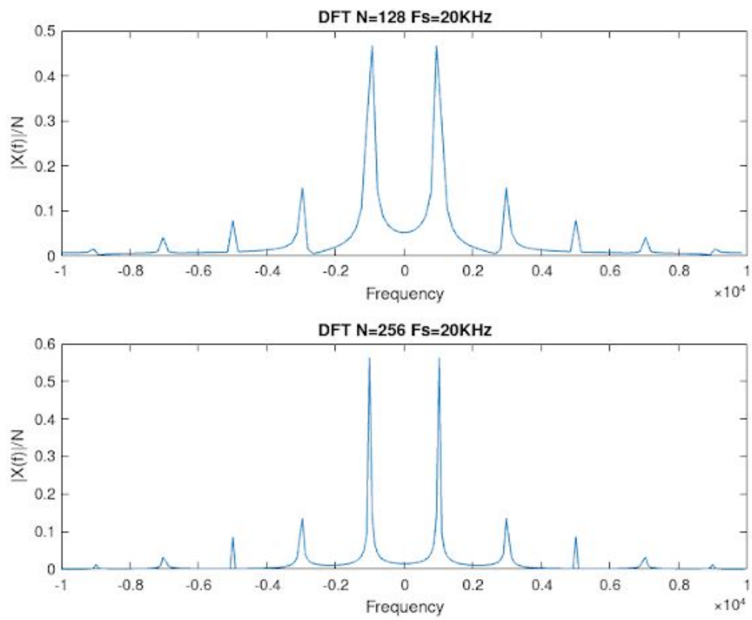

Fig.c

- Above 2 subplots depict the part of the DFT(from F=-10KHz to F=10KHz) centered at frequency=0 of the square wave with time period 1ms and Fs=20KHz,for different signal length.

Observation & Discussion:-

- As we know that the time period of the square wave is 1ms,i.e., frequency of the square wave is 1KHz.
- From the plot we can see that DFT of the square wave has peaks at all the odd integer multiples of the frequency of the square wave.i.e., peaks at 3KHz,5KHz,7KHz,9KHz, which is evident from the fourier series of the square wave.

$$x(t) = \sum_{n=1}^{n=\infty} b_n \sin n w_0 t$$

Where,w_o is the frequency of the square wave.

<u>d) Interpolation or Upsampling:-</u>

Problem statement:-

- To upsample/interpolate the given low pass signal which is sampled at a sampling frequency Fs1 and obtain the sampled signal with different sampling frequency Fs2.
- One of the low pass signal of bandwidth 6KHz:
  - x(t) = 5*cos(2*pi*F*5*t/6) + 4*cos(2*pi*2*F*t/3) + 3*cos(2*pi*F*t/2) + 2*cos(2*pi*F*t/3) + cos(2*pi*F*t/6)
  - The below plots are obtained by taking the signal length to be N=256 and Fs1=12KHz.;

- This signal is sampled with Fs1=12KHz and we obtain the sampled signal at different sampling frequency Fs2=24KHz.

Results:-

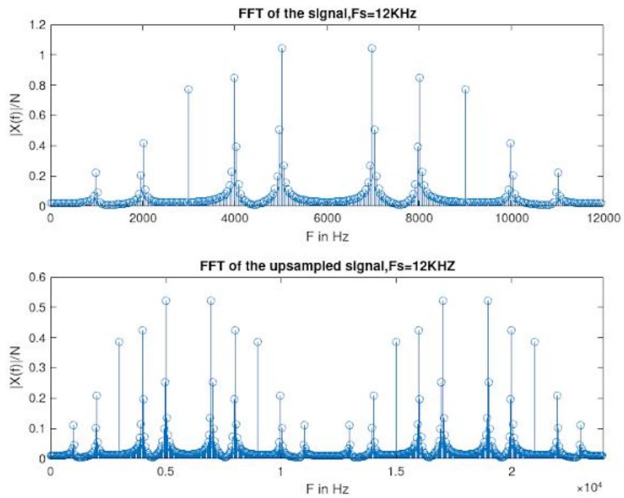

Fig.d_1

- The subplot 1 is the FFT of the taken signal x(t) with Fs=12KHz.
- The subplot 2 is the FFT of the signal x(t)(upsampled signal) after inserting zeroes between each sample.

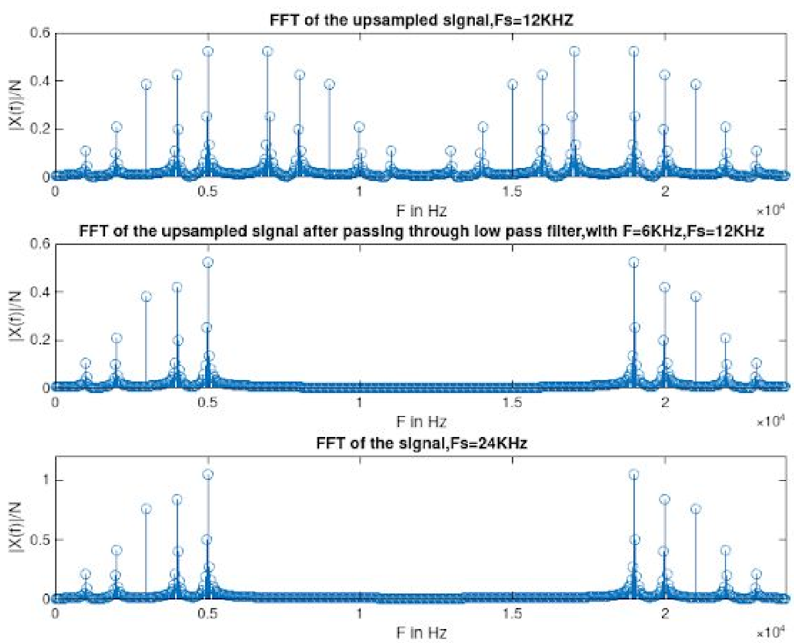

Fig.d_2

- The subplot2 is the FFT of the upsampled signal after passing through the low pass filter which has a cut off frequency of 6KHz.
- The subplot3 is the FFT of the sampled original signal with FS=24KHz.

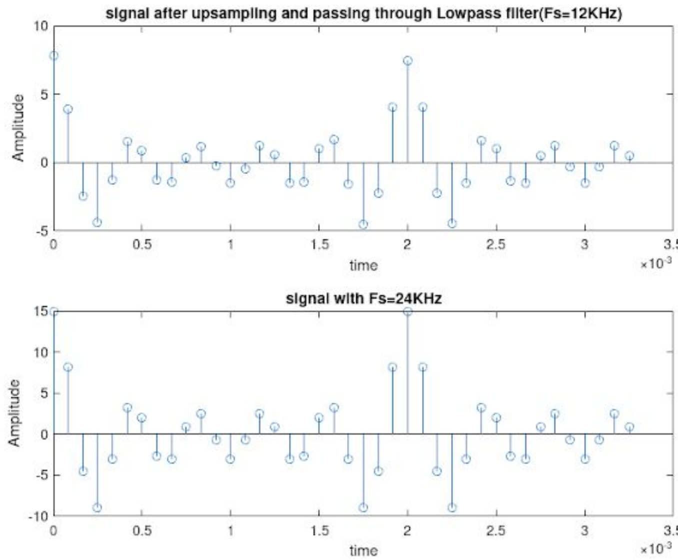

Fig.d_3

- The subplot1 is the sampled signal after interpolation.and taken only 40 points from t=0;
- The subplot2 is the sampled signal with double the sampling frequency.and taken only 40 points from t=0;

Observation & Discussion:-

- Upsampling in the time domain makes a replica of FFT in the frequency domain of the original frequency which can be seen in the fig.d_1.i.e.,the new frequency components get included.
- If we pass a signal through the low pass filter with a particular cutoff frequency Fc, then the signal losses those frequency components from the original signal which are greater than than the Fc,which can be observed from the Fig.d_2: 1 and 2 subplots ,plot 1 is the FFT before passing through the low pass filter(it has frequencies up to 12KHz) and plot 2 is the FFT of the filtered output (there are no high frequency components i.e., there are only frequency components which are less than the 6KHz).

(Codes for MATLAB simulations are there in the APPENDIX below)

Name:Harshavardhan Alimi
Roll No.:18EC10021

APPENDIX:

Part a:

```
Fs=12000; %%sampling frequency
Ts=1/Fs;
%%
N=12; %%signal length
t=0:Ts:(N-1)*Ts; %%sampling
%signal:X(t)
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y=fft(x,N); %%fft of the signal X(t)
y=fftshift(y); %%fftshift of the fft so that 0 frequency comes to the centre
f=-(1*Fs)/2:Fs/N:(N-1)*Fs/(2*N);
%%
N=64; %%%same as above with different N value
t=0:Ts:(N-1)*Ts;
x2=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y2=fft(x2,N);
y2=fftshift(y2);
f2=-(1*Fs)/2:Fs/N:(N-1)*Fs/(2*N);
%%
N=128; %%%same as above with different N value
t=0:Ts:(N-1)*Ts;
x3=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y3=fft(x3,N);
y3=fftshift(y3);
f3=-(1*Fs)/2:Fs/N:(N-1)*Fs/(2*N);
%%
N=256; %%%same as above with different N value
t=0:Ts:(N-1)*Ts;
x4=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y4=fft(x4,N);
y4=fftshift(y4);
f4=-(1*Fs)/2:Fs/N:(N-1)*Fs/(2*N);
%%
%%plotting the results:
subplot(221)
stem(f,abs(y)/12);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('DFT N=12 Fs=12KHz');

subplot(222)
stem(f2,abs(y2)/64);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('DFT N=64 Fs=12KHz');
subplot(223)

stem(f3,abs(y3)/128);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('DFT N=128 Fs=12KHz');
```

```
subplot(224)
stem(f4,abs(y4)/256);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('DFT N=256 Fs=12KHz');
```

Part b:

```
N=64; %%figure(1) with N=64->signal length
%%%%Fs=12KHz
Fs=12000; %%sampling frequency
t=0:1/Fs:(N-1)/Fs; %%sampling
%signal X(t):
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y1=fft(x,N); %%fft of the signal
f1=0:Fs/N:(N-1)*Fs/(N);
%%%%same as above but with different Fs=8KHz
Fs=8000;
t=0:1/Fs:(N-1)/Fs;
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y2=fft(x,N);
f2=0:Fs/N:(N-1)*Fs/(N);
%%%%same as above but with different Fs=5KHz
Fs=5000;
t=0:1/Fs:(N-1)/Fs;
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y3=fft(x,N);
f3=0:Fs/N:(N-1)*Fs/(N);
%%%%%same as above but with different Fs=4KHz
Fs=4000;
t=0:1/Fs:(N-1)/Fs;
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y4=fft(x,N);
f4=0:Fs/N:(N-1)*Fs/(N);
%%%%plotting the results for N=64 with different Fs values
figure(1)
subplot(221)
stem(f1,abs(y1)/N);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT N=64 Fs=12KHz');

subplot(222)
stem(f2,abs(y2)/N);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT N=64 Fs=8KHz');

subplot(223)
stem(f3,abs(y3)/N);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT N=64 Fs=5KHz');

subplot(224)
```

```
stem(f4,abs(y4)/N);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT N=64 Fs=4KHz');
%%
N=128; %%figure(2) with N=128->signal length
%%%Fs=12Kz
Fs=12000; %%sampling frequency
t=0:1/Fs:(N-1)/Fs; %%sampling
%signal X(t):
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y1=fft(x,N); %%FFT of the given signal
f1=0:Fs/N:(N-1)*Fs/(N);
%%%%same as above but with different Fs=8KHz
Fs=8000;
t=0:1/Fs:(N-1)/Fs;
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y2=fft(x,N);
f2=0:Fs/N:(N-1)*Fs/(N);
%%%%same as above but with different Fs=5KHz
Fs=5000;
t=0:1/Fs:(N-1)/Fs;
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y3=fft(x,N);
f3=0:Fs/N:(N-1)*Fs/(N);
%%%%%same as above but with different Fs=4KHz
Fs=4000;
t=0:1/Fs:(N-1)/Fs;
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y4=fft(x,N);
f4=0:Fs/N:(N-1)*Fs/(N);
%%%%plotting the results for N=128 with different Fs values
figure(2)

subplot(221)
stem(f1,abs(y1)/N);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT N=128 Fs=12KHz');

subplot(222)
stem(f2,abs(y2)/N);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT N=128 Fs=8KHz');

subplot(223)
stem(f3,abs(y3)/N);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT N=128 Fs=5KHz');

subplot(224)
stem(f4,abs(y4)/N);
```

```
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT N=128 Fs=4KHz');
%%
N=256; %%figure(3) with N=256->signal length
%%%Fs=12KHz
Fs=12000; %%sampling frequency
t=0:1/Fs:(N-1)/Fs; %%sampling
%%%signal X(t):
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y1=fft(x,N); %%FFT of the given signal
f1=0:Fs/N:(N-1)*Fs/(N);
%%%%same as above but with different Fs=8KHz
Fs=8000;
t=0:1/Fs:(N-1)/Fs;
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y2=fft(x,N);
f2=0:Fs/N:(N-1)*Fs/(N);
%%%%same as above but with different Fs=5KHz
Fs=5000;
t=0:1/Fs:(N-1)/Fs;
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y3=fft(x,N);
f3=0:Fs/N:(N-1)*Fs/(N);
%%%%%same as above but with different Fs=4KHz
Fs=4000;
t=0:1/Fs:(N-1)/Fs;
x=10*cos(2*pi*1000*t)+6*cos(2*pi*2000*t)+2*cos(2*pi*4000*t);
y4=fft(x,N);
f4=0:Fs/N:(N-1)*Fs/(N);
%%%%plotting the results for N=256 with different Fs values
figure(3)

subplot(221)
stem(f1,abs(y1)/N);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT N=256 Fs=12KHz');

subplot(222)
stem(f2,abs(y2)/N);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT N=256 Fs=8KHz');

subplot(223)
stem(f3,abs(y3)/N);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT N=256 Fs=5KHz');

subplot(224)
stem(f4,abs(y4)/N);
xlabel('F in Hz');
```

```
ylabel('|X(f)|/N');
title('FFT N=256 Fs=4KHz');
```

Part c:

```
F = 1000; %%Frequency of the square wave i.e. time period =1ms
N1 = 128; %%signal length
Fs = 20000; %%sampling frequency
Ts = 1/Fs;
t = 0:Ts:(N1-1)*Ts; %%sampling the time
x1 = 0;
%% evaluates the value of square wave from the fourier series expansion and takes upto 10 harmonics.
for i = 1:10
j = 2*(i-1)+1;
x1 = x1 + (4*sin(2*pi*F*j*t))/(pi*j);
end

X1 = fft(x1); %%fft of the square wave
X1 = fftshift(X1); %%fftshift of the fft so that 0 frequency comes to the centre
f1 = (-N1/2:N1/2-1)*(Fs/N1);
%%
%%same as above but with different signal length N=256
F = 1000;
T = 1/F;
N2 = 256;
Fs = 20000;
Ts = 1/Fs;

t = 0:Ts:(N2-1)*Ts;
x2 = 0;
for i = 1:10
j = 2*(i-1)+1;
x2 = x2 + (4*sin(2*pi*F*j*t))/(pi*j);
end

X2 = fft(x2);
X2 = fftshift(X2);
f2 = (-N2/2:N2/2-1)*(Fs/N2);
%%
%%results of the above processes
subplot(211)
plot(f1,abs(X1)/N1);
xlabel("Frequency");
ylabel("|X(f)|/N");
title('DFT N=128 Fs=20KHz');

subplot(212)
plot(f2,abs(X2)/N2);
xlabel("Frequency");
ylabel("|X(f)|/N");
title('DFT N=256 Fs=20KHz');
```

Part d:

```
Fs1=12000; %%initial sampling Frequency
F=6000; %%signal bandwidth
Fs2=24000; %%to be obtained sampling frequency
Ts1=1/Fs1;
Ts2=1/Fs2;
%%
N=256; %%signal length
t1=0:Ts1:(N-1)*Ts1; %%sampling
%Low pass signal which has frequency components 1KHz,2KHz,3KHz,4KHz,5KHz.it
%is bandlimited to 6KHZ
x1=5*cos(2*pi*F*5*t1/6)+4*cos(2*pi*2*F*t1/3)+3*cos(2*pi*F*t1/2)+2*cos(2*pi*F*t1/3)+cos(2*pi*F*t1/6);
y4=fft(x1); %%fft of the taken signal
f4=0:Fs1/(N):(N-1)*Fs1/(N); %frequency sampling
%%%
x1_0=upsample(x1,2); %%inserting zeroes between the samples of the original signal.
t1_0=0:Ts1:(2*N-1)*Ts1; %%sampling the upsampled signal.
y1=fft(x1_0); %%fft of the upsampled signal
f1=0:Fs2/(2*N):(2*N-1)*Fs2/(2*N);
%passing signal through low pass filter which has cutoff frequency F and sampling frequency of the signal Fs
y=lowpass(x1_0,F,Fs2);
y2=fft(y); %%fft the signal which was passed through the low pass filter
%%
N=256*2; %%signal length
t2=0:Ts2:(N-1)*Ts2; %%sampling at the signal with sampling frequency Fs2
x2=5*cos(2*pi*F*5*t2/6)+4*cos(2*pi*2*F*t2/3)+3*cos(2*pi*F*t2/2)+2*cos(2*pi*F*t2/3)+cos(2*pi*F*t2/6);
y3=fft(x2); %%fft of the signal
%%
%seeing the effect on FFT by inserting the zeroes between the samples which is sampled at Fs=12KHz
figure(1);

subplot(211)
stem(f4,abs(y4)/(N));
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT of the signal,Fs=12KHz');

subplot(212)
stem(f1,abs(y1)/(2*N));
axis([0 24000 0 0.6]);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT of the upsampled signal,Fs=12KHZ');
%%
%seeing the effect of low pass filter
figure(2);

subplot(311)
stem(f1,abs(y1)/(2*N));
axis([0 24000 0 0.6]);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT of the upsampled signal,Fs=12KHZ');
```

```
subplot(312)
stem(f1,abs(y2)/(2*N));
axis([0 24000 0 0.6]);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT of the upsampled signal after passing through low pass filter,with F=6KHz,Fs=12KHz');

subplot(313)
stem(f1,abs(y3)/(2*N));
axis([0 24000 0 1.2]);
xlabel('F in Hz');
ylabel('|X(f)|/N');
title('FFT of the signal,Fs=24KHz');
%%
%comparing the signals which is obtained by interpolating the sampled signal and the other
%which is obtained by sampling at double the frequency
figure(3)

subplot(211);
stem(t1_0(1:40),y(1:40));
xlabel("time");
ylabel("Amplitude");
title("signal after upsampling and passing through Low Pass filter(Fs=12KHz)");

subplot(212);
stem(2*t2(1:40),x2(1:40));
xlabel("time");
ylabel("Amplitude");
title("signal with Fs=24KHz");
```