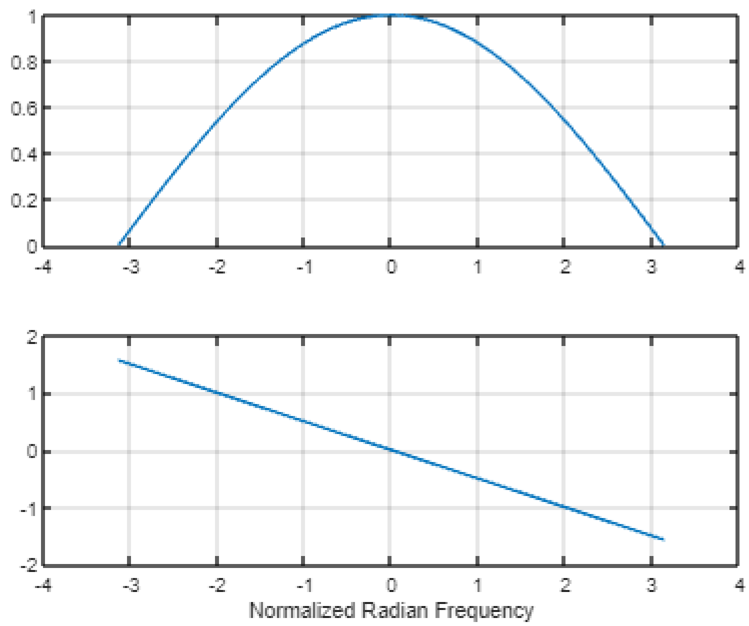
Name: Basil khowaja

Aqeel mehdi

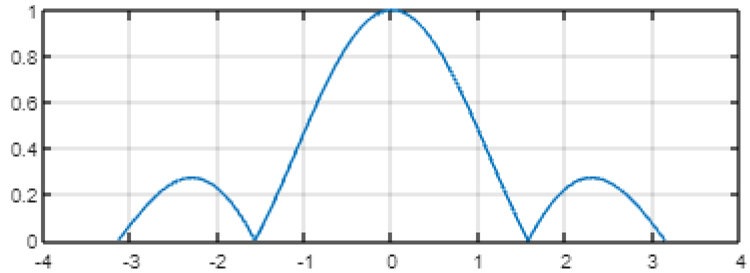
Frequency Response:

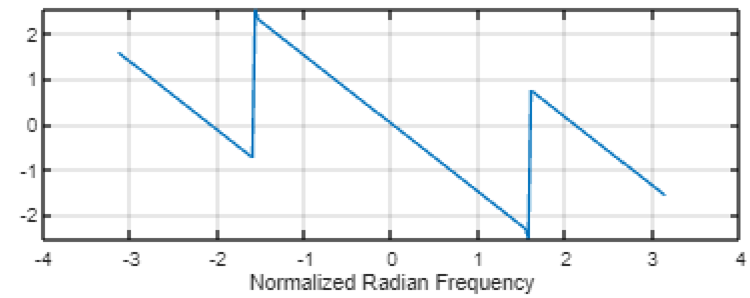
```
bb = [0.5, 0.5]; %-- Filter Coefficients
ww = -pi:(pi/100):pi; %-- Frequency vector
H = freqz(bb, 1, ww); %frequency reponse
figure;
subplot(2,1,1);
plot(ww, abs(H)), grid on
subplot(2,1,2);
plot(ww, angle(H)), grid on
xlabel("Normalized Radian Frequency")
```



Task 1:

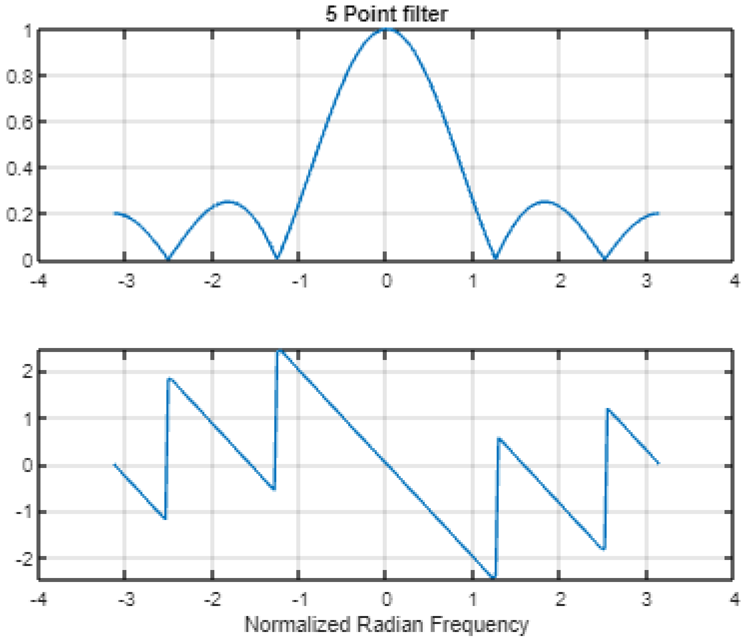
```
bb = [0.25, 0.25, 0.25, 0.25]; %-- Filter Coefficients
% bb = ones(1, 4);
ww = -pi:(pi/100):pi; %-- Frequency vector
H = freqz(bb, 1, ww); %frequency reponse
figure;
subplot(2,1,1);
plot(ww, abs(H)), grid on
subplot(2,1,2);
plot(ww, angle(H)), grid on
xlabel("Normalized Radian Frequency")
```

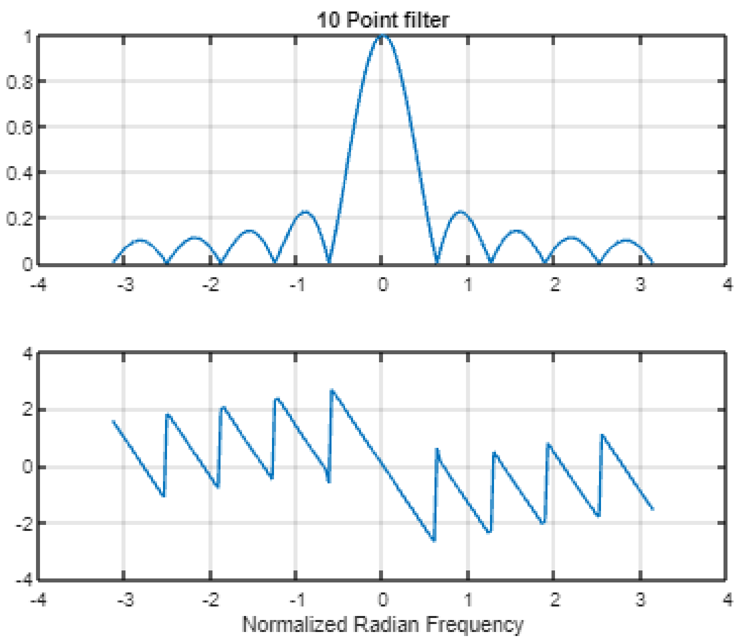


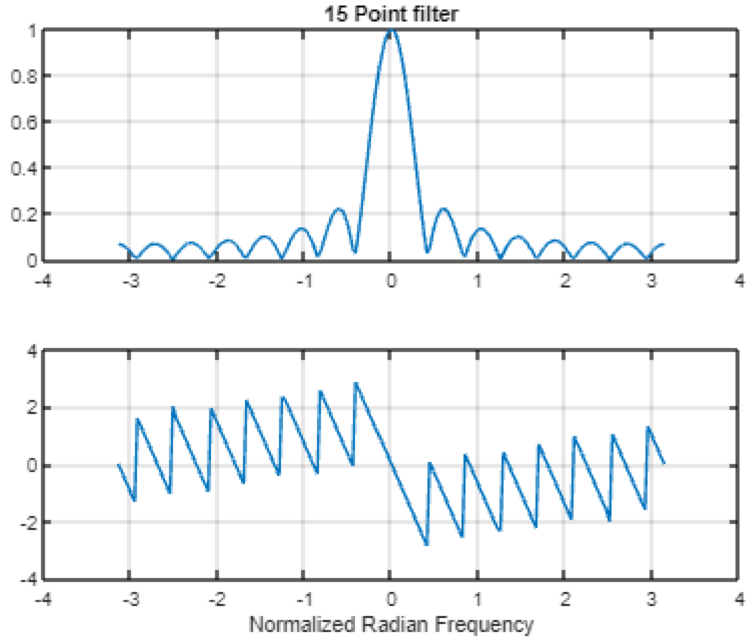



```
% General FIR filter
for j = 1:3
    N = j*5;
    bb = ones(1, N)./N;
%     for i = 1:N
%         bb(i) = 1/N;
%     end
%    bb = [0.5, 0.5]; %-- Filter Coefficients
    ww = -pi:(pi/100):pi; %-- Frequency vector
    H = freqz(bb, 1, ww); %frequency reponse

    figure;
    subplot(2, 1, 1);
    plot(ww, abs(H)), grid on
    title(N + " Point filter")
    subplot(2,1,2);
    plot(ww, angle(H)), grid on
    xlabel("Normalized Radian Frequency")
end
```


5 Point filter
1
0.8
0.6
0.4
0.2
0
-4 -3 -2 -1 0 1 2 3 4
2
1
0
-1
-2
-4 -3 -2 -1 0 1 2 3 4
Normalized Radian Frequency


10 Point filter
1
0.8
0.6
0.4
0.2
0
-4 -3 -2 -1 0 1 2 3 4
4
2
0
-2
-4
-4 -3 -2 -1 0 1 2 3 4
Normalized Radian Frequency



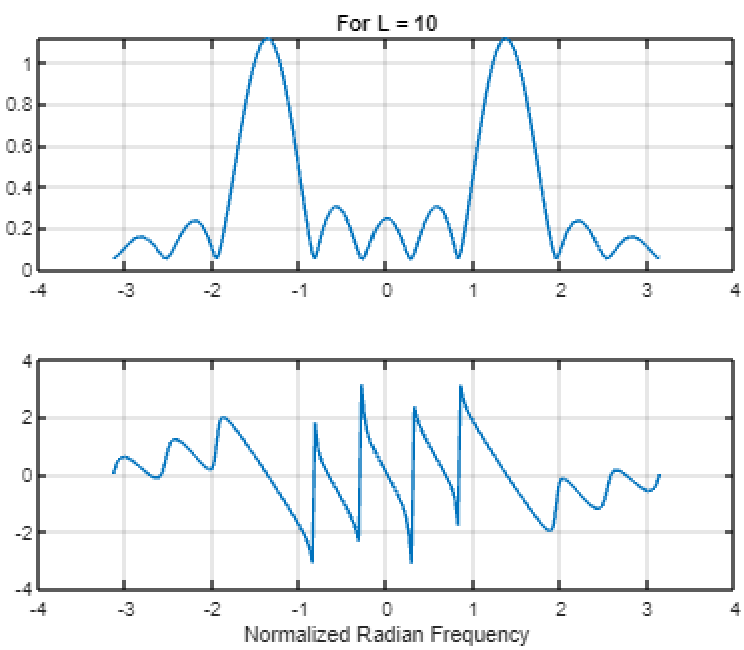
In this code we performs two tasks: In the first task, we compute and plot the frequency response of a simple FIR filter with coefficients [0.25, 0.25, 0.25, 0.25]. In the second task, we generate FIR filters of varying lengths (5, 10, 15) with coefficients set to 1/N, and then find their frequency responses, and plot the magnitude and phase. This filter had a smoothing effect in the previous labs because it attenuates the higher frequencies and usuallythe noise in a signal has very high frequencies so this filter does not allow that noise to pass and we get smoothing effect. In K point averaging filter the more we increase the value of K the less amount of frequencies it will allow andthe frequencies that does not pass would be more.

Task 2:

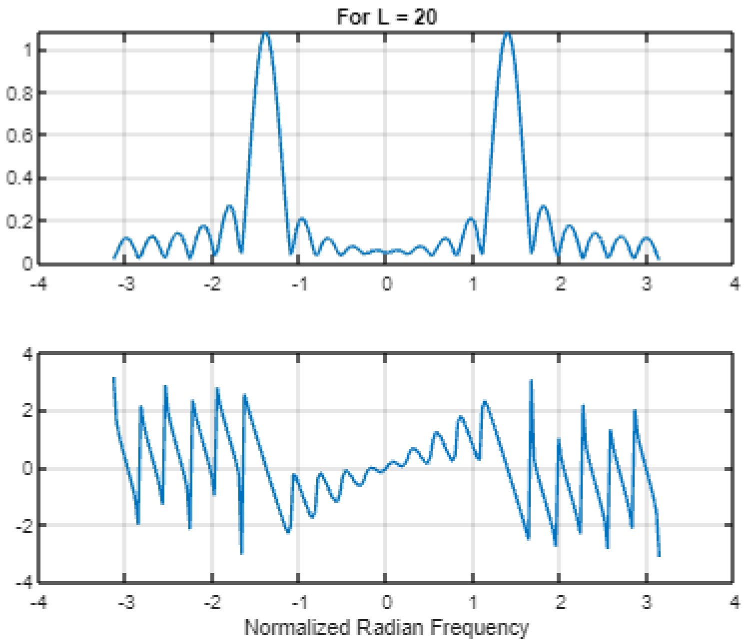
```
ab = [10 20 40];
for i = ab
    L = i;
    wc = 0.44*pi;
    n= 0:L;
    hn = 2/L*(cos(wc*n));
    ww = -pi:(pi/100):pi; %-- Frequency vector
    H = freqz(hn, 1, ww); %frequency reponse
    hmax = max(abs(H));
    freq = zeros(1, length(H));
    for j = 1:length(H)
        if abs(H(j))/hmax > 0.707
            freq(j) = j;
        end
    end
    pass = find(freq)
    for j = 1:length(pass)-1
        if pass(j+1) - pass(j) > 5
            mid = j;
```

```
                break
            end
        end
%       pass1 = freq(pass);
%       width = [abs(ww(pass(1)))-abs(ww(pass(j))) abs(ww(pass(j+1)))-abs(ww((length(pass)-j)))]
        width = [j length(pass)-j]
        figure;
        subplot(2,1,1);
        plot(ww, abs(H)), grid on
        title("For L = " + L);
        subplot(2,1,2);
        plot(ww, angle(H)), grid on
        xlabel("Normalized Radian Frequency")
end
```

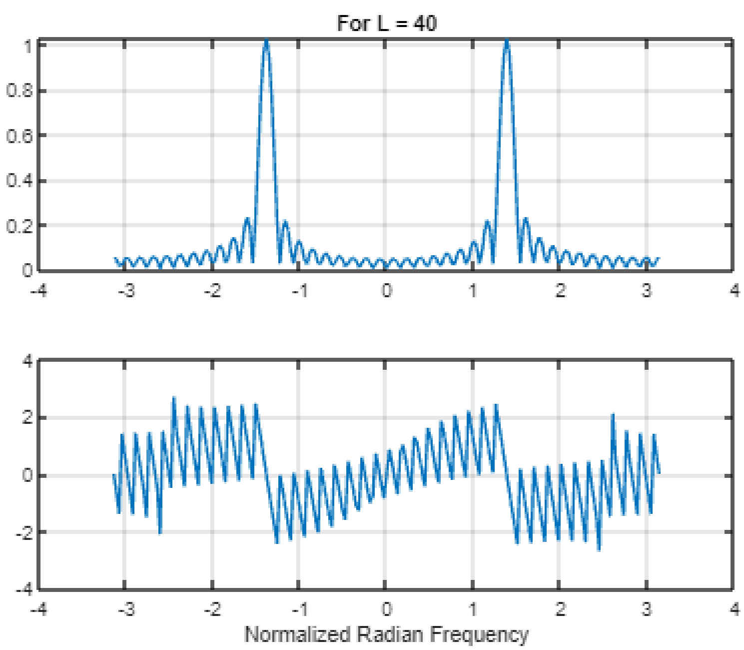
```
pass = 1×32
    50    51    52    53    54    55    56    57    58    59    60    61    62 ···
width = 1×2
    16    16
```




```
pass = 1×16
    53    54    55    56    57    58    59    60   142   143   144   145   146 ···
width = 1×2
     8     8
```




```
pass = 1×10
    55    56    57    58    59   143   144   145   146   147
width = 1×2
     5     5
```



In this code we iterate over a series of lengths (10, 20, 40) and find the frequency response of filters for each length. For each length, we calculate the filter coefficients based on a cosine function, then determine the frequency components corresponding to the passband by finding where the magnitude response exceeds 70.7% of the maximum value. We then find the width of the passband by finding the frequency range between two points where the magnitude response crosses the 70.7% threshold. Then in last we plot the magnitude and phase responses of each filter. When we double the value of L the band of the allowable frequecies decreaeses. The value of L is inverslyproportional to the width of the passband.

```
for j
    =
    2
    :
```

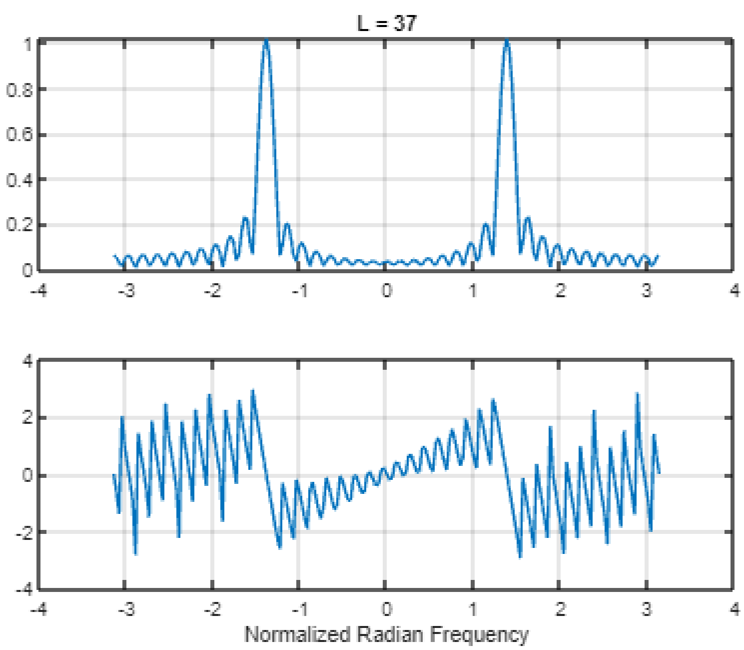
```
h = (2/j)*(cos(wc*n));

    H = freqz(h,1,ww);

    i1 = find(ww ==0.3*pi);
    i2 = find(ww == -0.7*pi);
    i3 = find(ww == -0.3*pi);
    i4 = find(ww == 0.7*pi);
    H_max = max(abs(H));
    if max(abs(H(1:i2))) < H_max/10 && max(abs(H(i3:i1))) < H_max/10 && max(abs(H(i4:length(H)
        L = j
        break;
    end
end
```

```
L = 37
```

```
figure;
subplot(2,1,1);
plot(ww,abs(H)), grid on
title("L = " + L);
subplot(2,1,2);
plot(ww, angle(H)), grid on
xlabel('Normalized Radian Frequency')
```




```
L = 21;
wc = 0.44*pi;
n= 0:L;
hn = 2/L*(cos(wc*n));
ww = -pi:(pi/100):pi; %-- Frequency vector
```

```
H = freqz(hn, 1, ww); %frequency reponse
hmax = max(abs(H));
freq = zeros(1, length(H));
for j = 1:length(H)
    if abs(H(j))/hmax > 0.707
        freq(j) = j;
    end
end
pass = find(freq)
```

```
pass = 1×16
    54    55    56    57    58    59    60    61   141   142   143   144   145 ···
```

```
for j = 1:length(pass)-1
    if pass(j+1) - pass(j) > 5
        mid = j;
        break
    end
end
%     pass1 = freq(pass);
%     width = [abs(ww(pass(1)))-abs(ww(pass(j))) abs(ww(pass(j+1)))-abs(ww((length(pass)-j)))]
width = [pass(j) pass(length(pass)-j)]
```

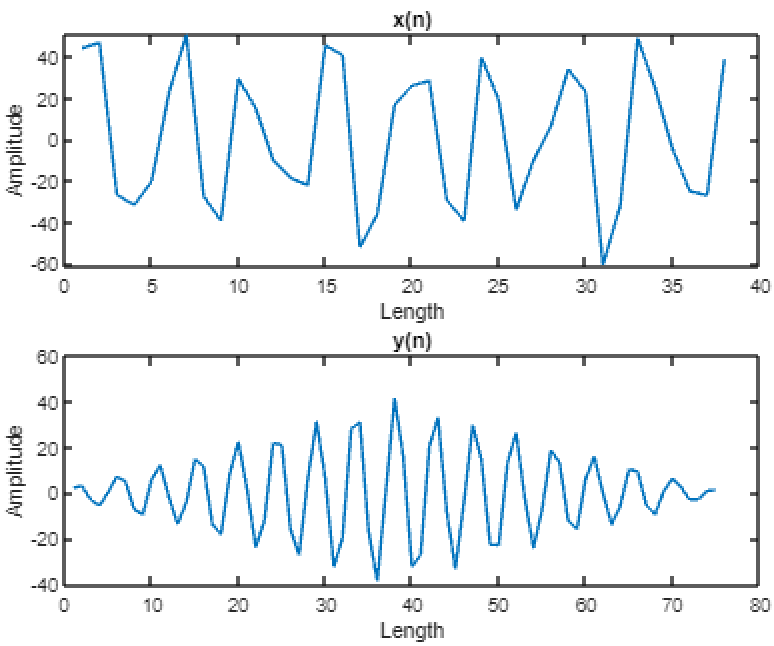
```
width = 1×2
    61    61
```

```
figure;
subplot(2,1,1);
plot(ww, abs(H)), grid on
title("For L = " + L);
subplot(2,1,2);
plot(ww, angle(H)), grid on
xlabel("Normalized Radian Frequency")
```

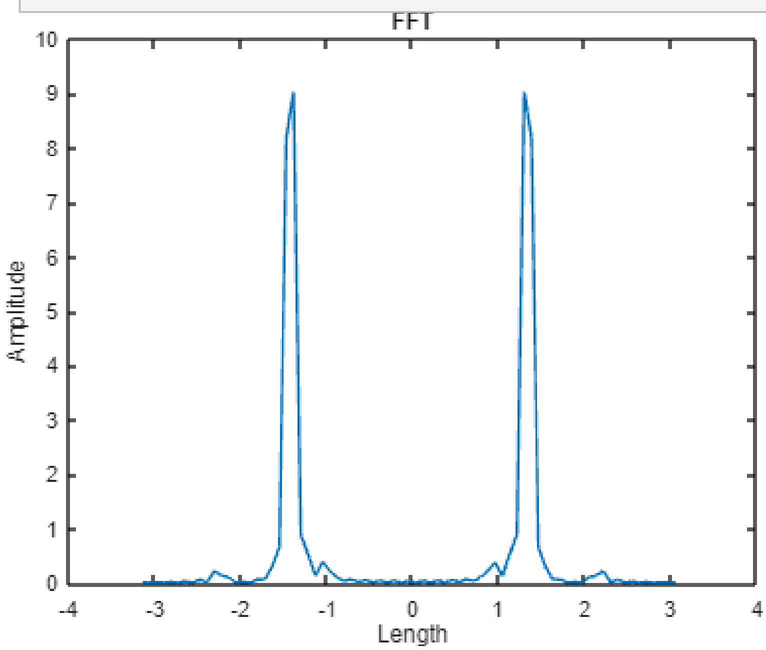
```
L = 37;
n= 0:L;
hn = 2/L*(cos(wc*n));
xn = 10*cos(0.3*pi*n) + 40*cos(0.44*pi*n - pi/3) + 20*cos(0.7*pi*n - pi/4);

yn = conv(xn, hn);
figure;
subplot(2, 1, 1); plot(xn)
title("x(n)")
xlabel("Length")
ylabel("Amplitude")
subplot(2, 1, 2); plot(yn)
title("y(n)")
xlabel("Length")
ylabel("Amplitude")
```

In this code we iterate through different lengths of FIR filters to find the minimum length required for achieving specific side level around given stopband frequencies. Then we calculate the filter coefficients based on a cosine function for each length and find their frequency response using the Fast Fourier Transform. Then we check if the side levels around the stopband frequencies meet certain condition. Once we find the minimum length satisfying the condition, we plot the frequency response of the corresponding filter. Also we design FIR filters with lengths 21 and 37, then convolve a composite signal with each filter, and plot the input and output signals. Then in last we perform FFT of the output signal to visualize its frequency spectrum.



```
fs = 2*pi; % sampling frequency
t = 0:(1/fs):(10-1/fs); % time vector
S= yn; %output acquired in step 1
n = length(S);
X = fft(S) ;
Y = fftshift(X);
fshift = (-n/2:n/2-1)*(fs/n); % zero-centered frequency range
figure;
plot(fshift,abs(Y)/n)
title("FFT")
xlabel("Length")
ylabel("Amplitude")
```



Name: Basil khowaja

Aqeel mehdi

d) $y[n] = \frac{1}{4}x[n] + \frac{1}{4}x[n-1] + \frac{1}{4}x[n-2]$

$+ \frac{1}{4}x[n-3]$

$$H(j\omega) = \frac{1}{4}\left(1 + e^{-j\omega} + e^{-2j\omega} + e^{-3j\omega}\right)$$

$$= \frac{1}{4}e^{-3/2j\omega}\left[e^{j\omega/2} + e^{-j\omega/2} + e^{3j\omega/2} + e^{-3j\omega/2}\right]$$

$$= \frac{1}{4}e^{-3/2j\omega}\left(2\cos(\omega/2) + 2\cos(3\omega/2)\right)$$

$$= \frac{2\cos(0.5\omega) + 2\cos(1.5\omega) \cdot e^{-1.5j\omega}}{4}$$