Problem 1.

a) $\delta_p = \delta_s = 0.15$ $\quad \Delta\omega = \omega_s - \omega_p = 0.25\pi$

$\omega_p = 0.5\pi$ $\quad A = -20\log_{10}\underbrace{\delta_{min}}_{\min(\delta_p,\delta_s)}$

$\omega_s = 0.75\pi$ $\quad = -20\log_{10}(0.15) \approx 16.48\,dB$

$$\beta = \begin{cases} 0.1102(A-8.7) & A > 50 \\ 0.5842(A-21)^{0.4} + 0.07886(A-21) & 21 \le A \le 50 \\ 0 & A < 21 \end{cases}$$

$$N = \frac{16.48-8}{2.285(0.25\pi)} = 4.71 \approx \underline{5}, \quad \underline{\beta = 0.0}$$

b) same as a)

c) $\delta_s = \delta_p = 0.09$, $\quad A \approx 20.92 \rightarrow \underline{\beta = 0.0}$,

$$N = \frac{20.92-8}{2.285(0.25\pi)} = 7.19 \approx \underline{9}$$

d) $\delta_s = \delta_p = 0.09$ $\quad A$: same as c) $\rightarrow \underline{\beta = 0.0}$

$\Delta\omega = 0.15\pi$

$$N = \frac{20.92-8}{2.285(0.15\pi} = 11.99 \rightarrow \approx \underline{N = 13}$$

2. a) Windowing is "near-optimal" but does not guarantee the filter will meet both ripple specifications.

Optimal design minimizes the max error in both δ_p and δ_s

b)

$$N = 2L+1 = \frac{-10\log_{10}(\delta_p\delta_s) - 13}{14.6(\omega_s - \omega_p)/(2\pi)} = \frac{-10\log_{10}(0.005) - 13}{14.6(0.25\cancel{\pi})/(2\cancel{\pi})}$$

$$\approx 5.48 \rightarrow \underline{N = 7}$$

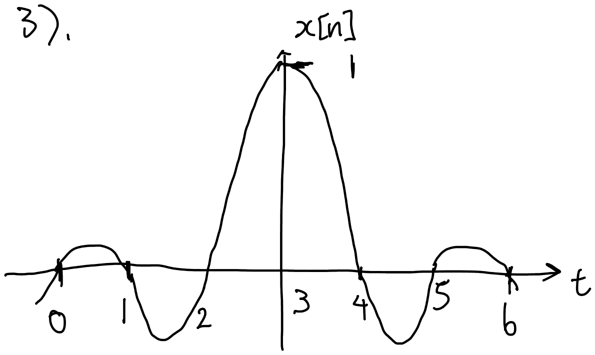
c) $A = -20\log_{10}(0.05) \approx 26.02\,dB$

$$N = \frac{A-8}{2.285\Delta\omega} = \frac{26.02-8}{2.285 \times 0.25\pi} = 10.04 \rightarrow \underline{N = 11}$$

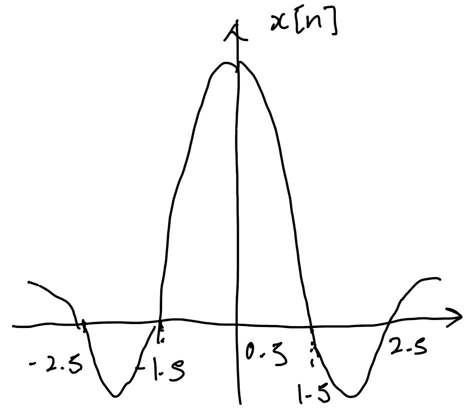
d) $N = 2L + 1 = 7 \rightarrow \underline{L = 3}$

Problem 3).

a)

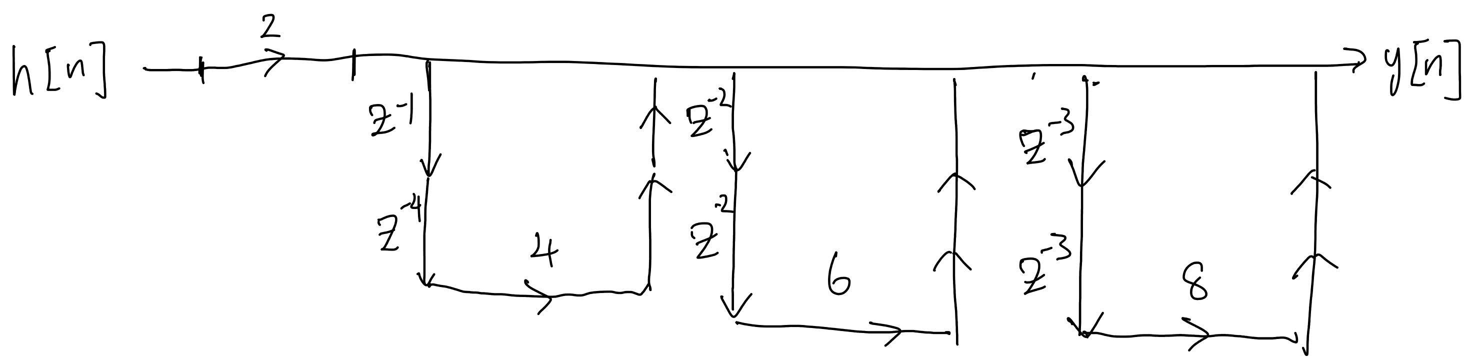


b)



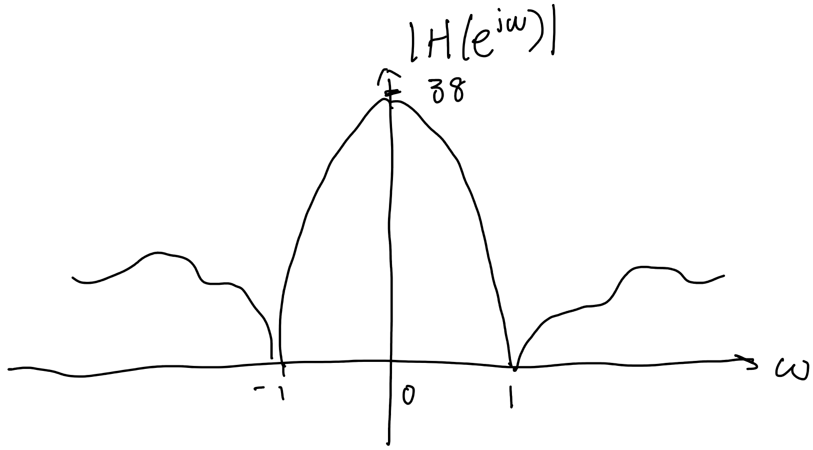
Problem 4. $h[n] = 2\delta[n] + 4\delta[n-1] + 6\delta[n-2] + 8\delta[n-3] + 6\delta[n-4] + 4\delta[n-5] + 8\delta[n-6]$

a) $h[n] = 2\delta[n] + 4(\delta[n-1] + [n-5]) + 6([\delta[n-2] + \delta[n-4]]) + 8(\delta[n-3] + \delta[n-6])$



b) $H(e^{j0}) = \sum h[n] = 2 + 4 + 6 + 8 + 6 + 4 + 8 = 38$

$H(e^{j\pi}) = \sum h[n](-1)^n = 2 - 4 + 6 - 8 + 6 - 4 + 8 = 6$



c) No general linear phase

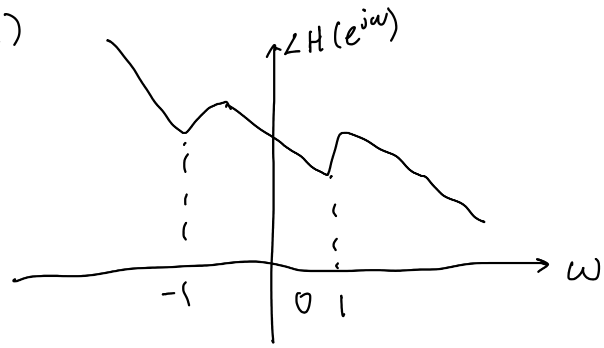
For $N = 7$, must be symmetric or anti-symmetric

$h[n] = h[6-n]$ or $-h[6-n]$

$\downarrow$

$h[0] = 2$, $h[6] = 8$, $h[0] \neq h[8]$

d)



```
% Part A
fp = 0.5;   % Passband edge frequency (Normalized: 0.5 * pi rad/s)
fs = 0.6;   % Stopband edge frequency (Normalized: 0.6 * pi rad/s)
dev_p = 0.1;  % Passband tolerance (delta_p)
dev_s = 0.05; % Stopband tolerance (delta_s)

f_edges = [fp, fs];
mags = [1, 0];    % Desired magnitudes: 1 in passband, 0 in stopband
devs = [dev_p, dev_s];

% firpmord estimates the lowest order (n) that meets the specs.
[n, fo, ao, w] = firpmord(f_edges, mags, devs);

% firpm uses the estimated parameters to find the filter coefficients (b).
b = firpm(n, fo, ao, w);

fprintf('Part A: The lowest estimated order (n) is: %d\n', n);
```

```
Part A: The lowest estimated order (n) is: 17
```

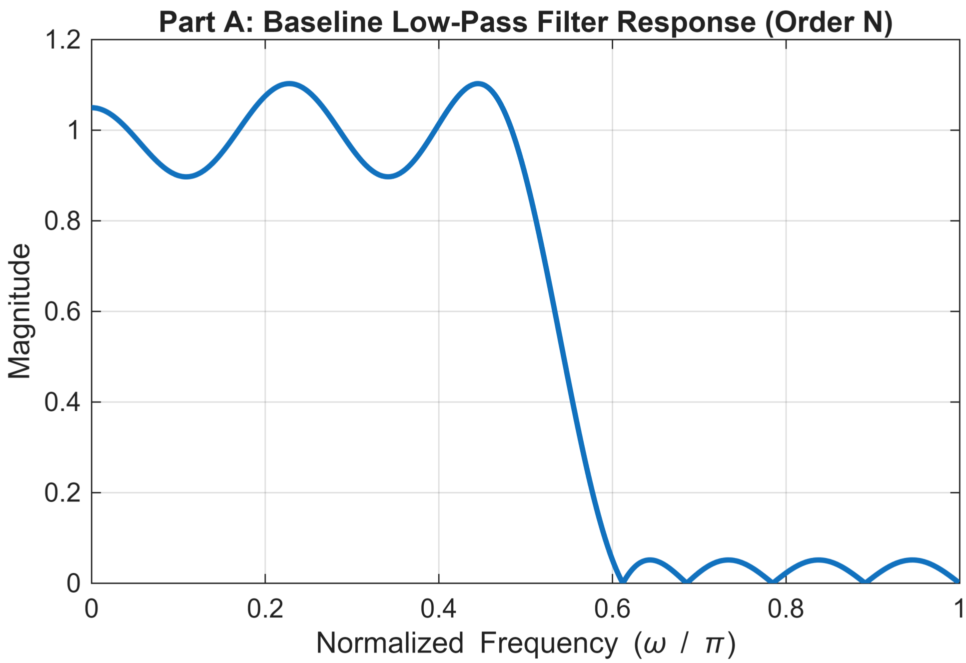
```
fprintf('The length of the impulse response is: %d\n', length(b));
```

```
The length of the impulse response is: 18
```

```

[h, w_rad] = freqz(b, 1, 1024); % 1024 frequency points
magnitude_response = abs(h);

figure(1);
plot(w_rad/pi, magnitude_response, 'LineWidth', 2);
title('Part A: Baseline Low-Pass Filter Response (Order N)');
xlabel('Normalized Frequency (\omega / \pi)');
ylabel('Magnitude');
grid on;
```




```
% Part B
dev_p_B = 0.05;
fs = 0.6; % fs and fp remain the same

f_edges_B = [fp, fs];
devs_B = [dev_p_B, dev_s]; % Use the new dev_p

[n_B, fo_B, ao_B, w_B] = firpmord(f_edges_B, mags, devs_B);

b_B = firpm(n_B, fo_B, ao_B, w_B);
fprintf('\nPart B: Reducing Passband Ripple:\n');
```

```
Part B: Reducing Passband Ripple:
```

```
fprintf('The new estimated order (n) is: %d\n', n_B);
```

```
The new estimated order (n) is: 21
```

```
fprintf('The new length of the impulse response is: %d\n', length(b_B));
```

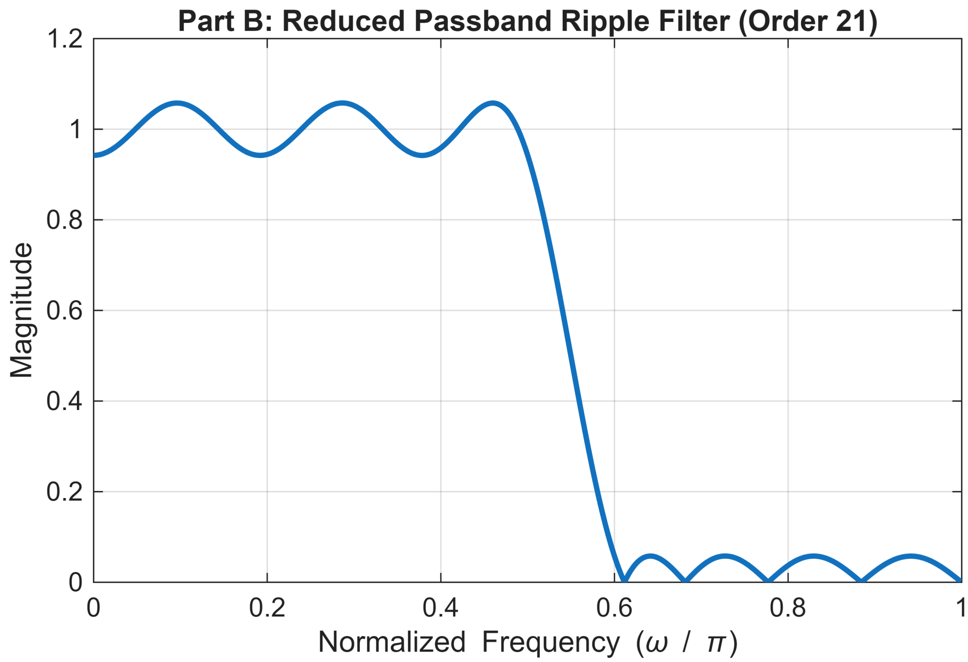
```
The new length of the impulse response is: 22
```

```

[h_B, ~] = freqz(b_B, 1, 1024);

figure(2);
plot(w_rad/pi, abs(h_B), 'LineWidth', 2);
title(['Part B: Reduced Passband Ripple Filter (Order ', num2str(n_B), ')']); % Use
n_B here
xlabel('Normalized Frequency (\omega / \pi)');
ylabel('Magnitude');
```

```
grid on;
```




```
% Part C
fs_C = 0.555;
dev_p_C = 0.05;

f_edges_C = [fp, fs_C]; % Narrower transition band
devs_C = [dev_p_C, dev_s];

[n_C, fo_C, ao_C, w_C] = firpmord(f_edges_C, mags, devs_C);
b_C = firpm(n_C, fo_C, ao_C, w_C);

fprintf('\nPart C: Narrowing Transition Band:\n');
```

```
Part C: Narrowing Transition Band:
```

```
fprintf('The final estimated order (n) is: %d\n', n_C);
```

```
The final estimated order (n) is: 39
```

```
fprintf('The final length of the impulse response is: %d\n', length(b_C));
```

```
The final length of the impulse response is: 40
```

```
[h_C, ~] = freqz(b_C, 1, 1024);

% Assumes h_C and n_C were calculated and mag_C = abs(h_C)
figure(3); % Use a new figure number (e.g., 3)
plot(w_rad/pi, abs(h_C), 'LineWidth', 2);
title(['Part C: Narrowed Transition Band Filter (Order ', num2str(n_C), ')']); %
Use n_C here
```

```
xlabel('Normalized Frequency (\omega / \pi)');
ylabel('Magnitude');
grid on;
```