If $h(n) = 6[n-1] - 26[n-4]$ then the filter coefficients are $b_k = \{0, 1, 0, 0, -2\}$.

The difference equation is $y[n] = x[n-1] - 2x[n-4]$. 
PROBLEM 5.2:

(a) 

(b) L=5 ⇒ avg. 5 points
Make table:

<table>
<thead>
<tr>
<th>n</th>
<th>-5</th>
<th>-4</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>x[n]=u[n]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>y[n]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/5</td>
<td>2/5</td>
<td>3/5</td>
<td>4/5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

AVG OVER 5 POINTS

(c) 

(d) General formula:

\[ y[n] = \frac{1}{L} \sum_{k=0}^{L-1} u[n-k] \]

= \frac{1}{L} u[n] + \frac{1}{L} u[n-1] + \frac{1}{L} u[n-2] + ... + \frac{1}{L} u[n-L+1]

For \( n < 0 \), \( y[n] = 0 \)

For \( n \geq L-1 \), \( y[n] = \frac{1}{L} \left( 1 + 1 + ... + 1 \right) = \frac{1}{L} \left( L \right) = 1 \)

Between, for \( 0 \leq n \leq L-1 \), the output is linearly increasing:

\[ y[n] = \frac{(n+1)}{L} \text{ for } 0 \leq n \leq L-1 \]
PROBLEM 5.3:

\[ y[n] = 2x[n] - 3x[n-1] + 2x[n-2] \]

(a) Make a table:

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>\geq 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>x[n]</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>y[n]</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
y[0] = 2x[0] - 3x[-1] + 2x[-2] = 2(1) = 2
\]
\[
y[1] = 2x[1] - 3x[0] + 2x[-1] = 2(2) - 3(1) = 1
\]
\[
y[2] = 2x[2] - 3x[1] + 2x[0] = 2(3) - 3(2) + 2(1) = 2
\]
\[
y[3] = 2(2) - 3(3) + 2(2) = -1
\]
\[
y[4] = 2(1) - 3(2) + 2(3) = 2
\]
\[
y[5] = 2(1) - 3(1) + 2(2) = 3
\]
\[
y[6] = 2(1) - 3(1) + 2(1) = 1
\]
\[
y[7] = 2(1) - 3(1) + 2(1) = 1
\]
\[
y[8] = 2(1) - 3(1) + 2(1) = 1
\]

(b) Impulse Response

\[
h[0] = 2(1) - 3(0) + 2(0) = 2
\]
\[
h[1] = 2(0) - 3(1) + 2(0) = -3
\]
\[
h[2] = 2(0) - 3(0) + 2(0) = 2
\]

Notice \( h[n] \) just "reads out" the filter coefficients:

\( i.e. \ h[n] = b_n \)
PROBLEM 5.3 (more):

Plots via MATLAB

INPUT SIGNAL $x[n]$

OUTPUT SIGNAL $y[n]$

UNIT IMPULSE delta[n]

IMPULSE RESPONSE $h[n]$

TIME INDEX (n)
PROBLEM 5.4:

\[ y[n] = 2x[n] - 3x[n-1] + 2x[n-2]. \]

(a) 

\[ x[n] \rightarrow \text{UNIT DELAY} \rightarrow \times -3 \rightarrow + \rightarrow y[n] \]

(b) Transposed Form

\[ x[n] \rightarrow \text{UNIT DELAY} \rightarrow \times 2 \rightarrow \times -3 \rightarrow + \rightarrow \text{UNIT DELAY} \rightarrow + \rightarrow y[n] \]
PROBLEM 5.5:

(a) \[ y[n] = \sum_{k=0}^{M} b_k x[n-k] \quad \text{for } n < 0 \text{ and } n \geq N \]

Assume \( b_0 \neq 0 \) and \( b_m \neq 0 \)

Since \( x[n] = 0 \) for \( n < 0 \), \( y[0] = b_0 x[0] + b_1 x[1] + \ldots \)

Thus \( y[0] \neq 0 \) if \( x[0] \neq 0 \).

Write out the sum:

\[ y[n] = b_m x[n-M] + b_{m-1} x[n-M+1] + b_{m-2} x[n-M+2] + \ldots \]

To find the largest \( n \) such that \( y[n] \neq 0 \), look at the term \( x[n-M] \). We need \( n-M < N \), otherwise, \( x[n-M] = 0 \)

Thus \( n < N+M \) is the condition \[ P = N+M \]

When \( n = N+M-1 \), then \( x[n-M+1] = x[N+M-1-M+1] = x[N] = 0 \) so the other terms in the sum drop out.

(b) \( x[n] = 0 \) for \( n < N_1 \) and \( n > N_2 \)

This is just a time shifted version of part (a).

The length of \( x[n] \) is \( N_2 - N_1 + 1 \) which takes the place of \( N \).

The output will start at \( n = N_1 \) because the time-invariance property applies. The output will end at \( n = M+N_2 \) because the term \( b_M x[n-M] \) will be the last one used in the sum. \( N_3 = N_1 \) and \( N_4 = N_2 + M \)

Here's a sketch:

---


This page should not be copied or electronically transmitted unless prior written permission has been obtained from the authors. October 8, 2004
PROBLEM 5.6:

Plots for parts (a), (b) and (c) are below. 
(d) This general solution will also apply to part (c).

\[ x[n] = a^n u[n] \]
\[ y[n] = \frac{1}{L} \sum_{k=0}^{L-1} x[n-k] = \frac{1}{L} \sum_{k=0}^{L-1} a^{n-k} u[n-k] \]

There are 3 cases.
1. \( n < 0 \). \( \Rightarrow y[n] = 0 \) because \( u[n-k] \) is always zero

2. \( 0 \leq n \leq L-1 \)
\[ y[n] = \frac{1}{L} \sum_{k=0}^{n} a^{n-k} u[n-k] = \frac{a^n}{L} \sum_{k=0}^{n} a^{-k} \]
\[ \Rightarrow y[n] = \frac{a^n}{L} \left( \frac{1-a^{-n}}{1-a^{-1}} \right) = \frac{1}{L} \left( \frac{a^{n+1} - 1}{a-1} \right) \]

3. \( n \geq L \)
\[ y[n] = \frac{1}{L} \sum_{k=0}^{L-1} a^{n-k} u[n-k] = \frac{a^n}{L} \sum_{k=0}^{L-1} a^{-k} \]
\[ = \frac{a^n}{L} \left( \frac{1-a^{-L}}{1-a^{-1}} \right) = \frac{a^n}{L} \left( \frac{a^{L-1} - 1}{a^L-a^{-1}} \right) \text{ for } n \geq L. \]

The impulse response contains the values of the filter coefficients because

\[ h[n] = \sum_{k=0}^{M} b_k \delta[n-k] \]

Thus,

\[ b_0 = 3, \quad b_1 = 7, \quad b_2 = 13, \quad b_3 = 9, \quad b_4 = 5 \]
Use convolution

\[ n: \ldots -2 -1 0 1 2 3 4 \ldots \]
\[ x[n]: \quad 0 1 0 1 0 1 0 \ldots \]
\[ h[n]: \quad 13 -13 13 \]

\[
\begin{array}{cccccccc}
\ldots & 0 & 13 & 0 & 13 & 0 & \ldots \\
\ldots & -13 & 0 & -13 & 0 & -13 & \ldots \\
\ldots & 0 & 13 & 0 & 13 & 0 & \ldots \\
-13 & 26 & -13 & 26 & -13 & 26 & -13 \\
\end{array}
\]

\[ y[n] = \begin{cases} 
-13 & \text{for } n \text{ even} \\
26 & \text{for } n \text{ odd} 
\end{cases} \]

\[ n=0 \]
PROBLEM 5.9:

Linearity?

(a) YES.

\[ y[n] = (\alpha_1 x_1[n] + \alpha_2 x_2[n]) \cos(0.2\pi n) \]
\[ = \alpha_1 \frac{x_1[n] \cos(0.2\pi n)}{y_1[n]} + \alpha_2 \frac{x_2[n] \cos(0.2\pi n)}{y_2[n]} \]

(b) YES.

\[ y[n] = (\alpha_1 x_1[n] - \alpha_2 x_2[n]) - (\alpha_1 x_1[n-1] + \alpha_2 x_2[n-1]) \]
\[ = \alpha_1 \frac{x_1[n] - x_1[n-1]}{y_1[n]} + \alpha_2 \frac{x_2[n] - x_2[n-1]}{y_2[n]} \]

(c) NO.

Let \( x_1[n] = \delta[n] \) and \( x_2[n] = -2\delta[n] \).

\[ y_1[n] = \delta[n] \quad \text{and} \quad y_2[n] = 2\delta[n] \]

Let \( x[n] = x_1[n] + x_2[n] = \delta[n] - 2\delta[n] = -\delta[n] \).

\[ y[n] = |x[n]| = \delta[n] \quad \text{and} \quad y_1[n] + y_2[n] = \delta[n] + 2\delta[n] = 3\delta[n] \]

(d) NO! if \( B \neq 0 \)

if \( x_1[n] \rightarrow y_1[n] \), test \( 2x_1[n] \rightarrow 2y_1[n] \).

\[ A(2x_1[n]) + B = 2(Ax_1[n] + B) - B \neq 2y_1[n] \]

Time-invariant?

(a) NO!

Let \( x[n] = \delta[n] \), then \( y[n] = \delta[n] \cos(0.2\pi n) = \delta[n] \)

Try \( x[n-1] = \delta[n-1] \), then output is \( \delta[n-1] \cos(0.2\pi n) \neq \delta[n-1] \cos(0.2\pi) \delta[n-1] \).

But \( \cos(0.2\pi) \neq \delta[n-1] \neq y[n-1] = \delta[n-1] \).
PROBLEM 5.9 (more):

**TIME INVARIANT?**

(b) Yes.
If \( x[n] \rightarrow y[n] \), let \( v[n] = x[n-n_0] \)

\[ \text{OUTPUT} = v[n] - v[n-1] = x[n-n_0] - x[n-n_0-1] \]

This is the same as \( y[n-n_0] = x[n-n_0] - x[n-n_0-1] \)

(c) Yes.
output depends only on \( x[\cdot] \) at \( n \), so \( y[n-n_0] = x[n-n_0] \)

(d) Yes
\[ y[n-n_0] = A x[n-n_0] + B \] is always true.

**CAUSAL?**

(a) Yes.
y[\( n \)] at \( n = n_0 \) depends only on \( x[\cdot] \) at \( n = n_0 \), and not on past or future values.

(b) Yes.
y[\( n \)] at \( n = n_0 \) depends only on \( x[\cdot] \) at \( n = n_0 \) \( \forall n = n_0 - 1 \)
so it only uses the "present" and the "past."

(c) Yes
\[ y[n] \] at \( n = n_0 \) depends only on \( x[n] \) at \( n = n_0 \).
\[ y[n_0] = x[n_0] \]

(d) Yes
\[ y[n] \] at \( n = n_0 \) depends only on \( x[n] \) at \( n = n_0 \).
\[ y[n_0] = A x[n_0] + B \]
PROBLEM 5.10:

\[ x[n] = \delta[n] - \delta[n-1] \quad \rightarrow \quad y[n] = \delta[n] - \delta[n-1] + 2\delta[n-3] \]
\[ x[n] = \cos(\pi n/2) \quad \rightarrow \quad y[n] = 2\cos(\pi n/2 - \pi/4) \]

(a) Make a plot of the signal: \( y[n] = \delta[n] - \delta[n-1] + 2\delta[n-3] \).

(b) Use linearity and time-invariance to find the output of the system when the input is

\[ x[n] = 7\delta[n] - 7\delta[n-2] \]

In order to use linearity \& time-invariance, we need to express \( x[n] \) in terms of known signals.

Let \( x_1[n] = 6\delta[n] - 6\delta[n-1] \)

Then \( x[n] = 7(6\delta[n] - 6\delta[n-2]) = 7x_1[n] + 7x_1[n-1] \)

Because \( x_1[n-1] = 6\delta[n-1] - 6\delta[n-2] \).

Now, LTI system \( \rightarrow \)

\[
\begin{align*}
7x_1[n] &\quad \rightarrow \quad 76\delta[n] - 76\delta[n-1] + 146\delta[n-3] \\
7x_1[n-1] &\quad \rightarrow \quad 76\delta[n-1] - 76\delta[n-2] + 146\delta[n-4]
\end{align*}
\]

Add them together:

\[ x[n] \quad \rightarrow \quad 76\delta[n] - 76\delta[n-2] + 146\delta[n-3] + 146\delta[n-4] \]
PROBLEM 5.11:

\[ h[n] = 3 \delta[n] - 2 \delta[n-1] + 4 \delta[n-2] + \delta[n-4] \]

\[ y[n] = 3x[n] - 2x[n-1] + 4x[n-2] + x[n-4] \]

(a) Direct Form:

(b) Transposed Direct Form:
PROBLEM 5.12:

\[ x_1[n] = u[n] \rightarrow y_1[n] = \delta[n] + 2\delta[n-1] - \delta[n-2] \]
\[ x_2[n] = 3u[n] - 2u[n-4] \]

Use linearity and time-invariance:

\[ y_2[n] = 3y_1[n] - 2y_1[n-4] \]
\[ = 3\delta[n] + 6\delta[n-1] - 3\delta[n-2] - 2\delta[n-4] \]
\[ - 4\delta[n-5] + 2\delta[n-6] \]

List of values:

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \leq 0 )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_2[n] )</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>-3</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
PROBLEM 5.13:

(a) \[ y[n] = x[n] - ax[n-1] \]
    \[ = a^n u[n] - a(a^{n-1} u[n-1]) \]
    \[ = a^o \delta[n] + a^n u[n-1] - a^n u[n-1] = \delta[n] \]

(b) Express \( x[n] \) as a sum: \( x[n] = a^n u[n] + (-a^n u[n-10]) \)

Because the FIR filter is an LTI system, we can find the output for \(-a^n u[n-10]\) and add it to the result from part (a)

\[ y_2[n] = -a^n u[n-10] - a(a^{n-1} u[n-11]) \]
\[ = -a^o \delta[n-10] - a^n u[n-11] + a^n u[n-11] \]
\[ = -a^o \delta[n-10] \]

\[ y[n] = \delta[n] - a^o \delta[n-10] \]  
from part (a)
PROBLEM 5.14:

(a) \( h[n] = \delta[n-2] \Rightarrow \) filter is a delay by 2

\[ y[n] = u[n-3] - u[n-6] \]

To find \( x[n] \) we need to "un-delay" \( y[n] \).

\[ \Rightarrow x[n] = u[n-1] - u[n-4] \]

(b) First-difference FIR \( \Rightarrow h[n] = \delta[n] - \delta[n-1] \)

The first-difference filter has a nonzero output at \( n \) when \( x[n] \neq x[n-1] \) are not equal.

If \( y[n] = \delta[n] - \delta[n-4] \), then the input \( x[n] \)
changes value at \( n=0 \) and \( n=4 \). At \( n=0 \), it
jumps up by one; at \( n=4 \), it jumps down.

\[ \Rightarrow x[n] = u[n] - u[n-4] \]

(c) 4-pt averager: \( y[n] = \frac{1}{4} (x[n] + x[n-1] + x[n-2] + x[n-3]) \)

If \( y[n] = -5\delta[n] - 5\delta[n-2] \)

\[ y[0] = -5 = \frac{1}{4} (x[0] + x[-1] + x[-2] + x[-3]) \]

**if we assume \( x[n] = 0 \) for \( n < 0 \), then \( x[0] = -20 \)

\[ y[1] = 0 = \frac{1}{4} (x[1] + x[0] + x[-1] + x[-2]) = \frac{1}{4} x[1] - 5 \]

\[ \Rightarrow x[1] = 20 \]

\[ y[2] = -5 = \frac{1}{4} (x[2] + x[1] + x[0] + x[-1]) \]

\[ = \frac{1}{4} (x[2] + 20 - 20 + 0) = \frac{1}{4} x[2] \]

\[ y[3] = 0 = \frac{1}{4} (x[3] + x[2] + x[1] + x[0]) \]

\[ \Rightarrow x[3] = -20 \]

\[ \Rightarrow x[n] = \begin{cases} 
0 & \text{for } n < 0 \\
-20 & \text{for } n \text{ even} \\
20 & \text{for } n \text{ odd}
\end{cases} \]
PROBLEM 5.15:

(a) \( x[n] = u[n] \) and \( y[n] = u[n-1] \)
We need a "delay by one".
\[ \Rightarrow h[n] = \delta[n-1] \]

(b) \( x[n] = u[n] \) and \( y[n] = \delta[n] \)
Since \( u[n] \) jumps from 0 to 1 at \( n=0 \), we need a filter that detects jumps. This can be done with a first-difference filter.
\[ h[n] = \delta[n] - \delta[n-1] \]

(c) \( x[n] = (\frac{1}{2})^n u[n] \) and \( y[n] = \delta[n-1] \)
Use the convolution sum to write linear equations:
\[ y[n] = \sum_{k=0}^{M} h[k] x[n-k]. \]
\[ y[0] = h[0] x[0] + h[1] x[-1] + \ldots \]
\[ 0 = h[0] \left( \frac{1}{2} \right)^0 + h[1] \left( \frac{1}{2} \right)^{-1} \Rightarrow h[0] = 0 \]
\[ 1 = 0 + h[1] \left( \frac{1}{2} \right)^0 + h[2] \left( \frac{1}{2} \right)^{-1} \Rightarrow h[1] = 1 \]
\[ 0 = 0 + 1 \left( \frac{1}{2} \right)^1 + h[2] \left( \frac{1}{2} \right)^0 \]
\[ 0 = \frac{1}{2} + h[2] \Rightarrow h[2] = -\frac{1}{2} \]
\[ 0 = 0 + 1 \left( \frac{1}{2} \right)^2 - \frac{1}{2} \left( \frac{1}{2} \right)^1 + h[3] \left( \frac{1}{2} \right)^0 \]
\[ 0 = 0 + \frac{1}{4} - \frac{1}{2} + h[3] \Rightarrow h[3] = 0 \]
Similarly for \( n > 3 \)
\[ \therefore h[n] = \delta[n-1] - \frac{1}{2} \delta[n-2] \]
**PROBLEM 5.16:**

Sometimes it is not possible to solve the *deconvolution* process for a given input-output pair. For example, prove that there is no FIR filter that can process the input \( x[n] = \delta[n] + \delta[n-1] \) to give the output \( y[n] = \delta[n] \).

**Solution:** The *deconvolution* filter that turns \( x[n] \) into \( y[n] \) must have an impulse response \( h[n] \) satisfying

\[
y[n] = h[n] * x[n] \quad \text{for all } n
\]

The method of proof will be to assume that \( h[n] \) is the impulse response of an FIR filter and show that we get a contradiction. If \( h[n] \) has finite length, then the most general statement we can make about \( h[n] \) is that it’s zero outside of a finite region, i.e.,

\[
h[n] = 0 \quad \text{for } n < N_1 \text{ or } n > N_2 \quad (1)
\]

Note: it is not necessary to assume that \( h[n] \) is the impulse response of a causal filter, but if it were then \( N_1 \) would be greater than or equal to zero.

We could consider several separate cases depending on whether \( N_1 \) is less than zero, equal to zero, or greater than zero. However, the most general case to consider would be the one where \( N_1 < 0 \), so we now write out a few terms of the convolution equation to see the general form:

\[
\begin{align*}
y[N_1] &= 0 = h[N_1] + h[N_1 - 1] = h[N_1] + 0 \quad \implies \quad h[N_1] = 0 \\
y[N_1 + 1] &= 0 = h[N_1 + 1] + h[N_1] \quad \implies \quad h[N_1 + 1] = 0 \\
y[N_1 + 2] &= 0 = h[N_1 + 2] + h[N_1 + 1] \quad \implies \quad h[N_1 + 2] = 0 \\
&\vdots \quad \vdots \\
y[-1] &= 0 = h[-1] + h[-2] \quad \implies \quad h[-1] = 0 \\
y[0] &= 1 = h[0] + h[-1] = h[0] + 0 \quad \implies \quad h[0] = 1 \\
y[1] &= 0 = h[1] + h[0] = h[1] + 1 \quad \implies \quad h[1] = -1 \\
&\vdots \quad \vdots \\
y[N_2 - 1] &= 0 = h[N_2 - 1] + h[N_2 - 2] \quad \implies \quad h[N_2 - 1] = -1 \\
y[N_2] &= 0 = h[N_2] + h[N_2 - 1] \quad \implies \quad h[N_2] = 1 \\
y[N_2 + 1] &= 0 = h[N_2 + 1] + h[N_2] \quad \implies \quad h[N_2 + 1] = -1 \\
y[N_2 + 2] &= 0 = h[N_2 + 2] + h[N_2 + 1] \quad \implies \quad h[N_2 + 2] = 1 \\
&\vdots \quad \vdots
\end{align*}
\]

where we have assumed that \( N_2 \) is an even integer.

The solution for the values of \( h[n] \) is done by solving the equations one at a time from top to bottom. The final two equations show that \( h[n] \) will be nonzero even when \( n > N_2 \) and thus provide the contradiction of the FIR assumption in equation (1). Hence, we are able to conclude that there is no FIR filter that can process the input \( x[n] = \delta[n] + \delta[n-1] \) to give the output \( y[0] = \delta[0] \).
PROBLEM 5.17:

(a) \( h_1[n] = \delta[n] - \delta[n-1] \)
    \( h_2[n] = \delta[n] + \delta[n-2] \)
    \( h_3[n] = \delta[n-1] + \delta[n-2] \)

(b) The overall \( h[n] \) is the convolution of the \( h_i[n] \).
    \[ h[n] = h_1[n] * h_2[n] * h_3[n] \]
    \[ h_1[n] * h_2[n] = (\delta[n] - \delta[n-1]) * (\delta[n] + \delta[n-2]) \]
    \[ = \delta[n] - \delta[n-1] + \delta[n-2] - \delta[n-3] \]
    Now convolve with \( h_3[n] \)
    \[
    \begin{array}{cccc}
    1 & -1 & 1 & -1 \\
    0 & 1 & 1 & 1 \\
    0 & 0 & 0 & 0 \\
    \end{array}
    \]
    \[
    \begin{array}{cccc}
    0 & 0 & 0 & 0 \\
    1 & -1 & 1 & -1 \\
    1 & -1 & 1 & -1 \\
    \end{array}
    \]
    \[ h[n] = \delta[n-1] - \delta[n-5] \]

(c) \[ y[n] = h[n] * x[n] \]
    \[ = (\delta[n-1] - \delta[n-5]) * x[n] \]
    \[ y[n] = x[n-1] - x[n-5] \]
PROBLEM 5.18:

The MATLAB program has two filters that are added together, and then filtered again

\[ y_1[n] = x[n] + x[n-1] + x[n-2] + x[n-3] \]
\[ y_2[n] = x[n] - x[n-1] - x[n-2] + x[n-3] \]
\[ w[n] = y_1[n] + y_2[n] \]
\[ y[n] = w[n] + w[n-1] + w[n-2] \]

(a)

\[ \{ b_k \} = \{ 1, 1, 1, 1 \} \]

SYSTEM \( S_1 \)

\[ \{ b_k \} = \{ 1, -1, 1, 1 \} \]

SYSTEM \( S_2 \)

\[ \{ b_k \} = \{ 1, 1, 1, 1 \} \]

SYSTEM \( S_3 \)

\( S_1 : h_1[n] = \delta[n] + \delta[n-1] + \delta[n-2] + \delta[n-3] \)
\( S_2 : h_2[n] = \delta[n] - \delta[n-1] - \delta[n-2] + \delta[n-3] \)
\( S_3 : h_3[n] = \delta[n] + \delta[n-1] + \delta[n-2] \)

(b) When \( x[n] = \delta[n] \), \( w[n] = h_1[n] + h_2[n] \)

Then \( y[n] = h_3[n] * w[n] \)

\[ = 2 \delta[n] + 2 \delta[n-3] \]

The overall difference equation is obtained by noting that the filter coefficients are equal to the impulse response values: \( b_k = h[n] \big|_{n=k} \)