Lab 1: Basic Signal Representation and Convolution in MATLAB

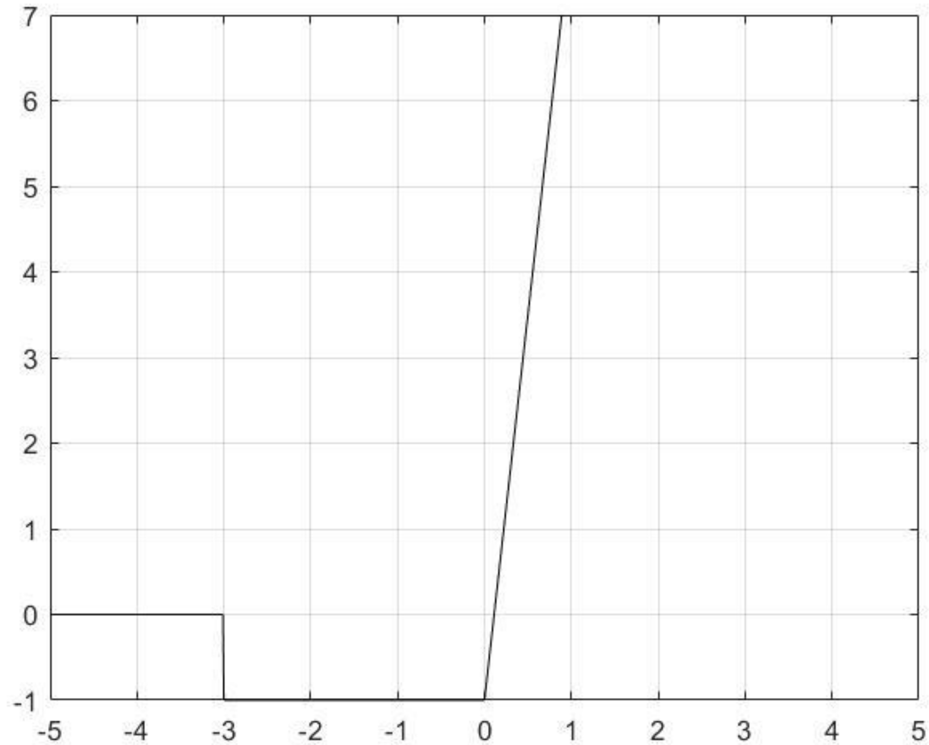
PART 1: Basic Signal Representation in MATLAB

1. **Write a Matlab program and necessary functions to generate the following signal:**

%code

```
clc;

Ts=0.01;
t= -5:Ts:5;

y1 = ramp(t,3,3);
y2 = ramp(t,-6,1);
y3 = ramp(t,3,0);

y4 = ustep(t,-3);

y = y1-2*y2+3*y3-y4;

plot(t,y,'k');

axis([-5 5 -1 7]);
grid


function y = ramp(t,m,ad)
        y = (m .* (t + ad)) .* ustep(t, ad);
end

function y = ustep(t, ad)
    y = (t >= (0 + ad));
end
```

%output



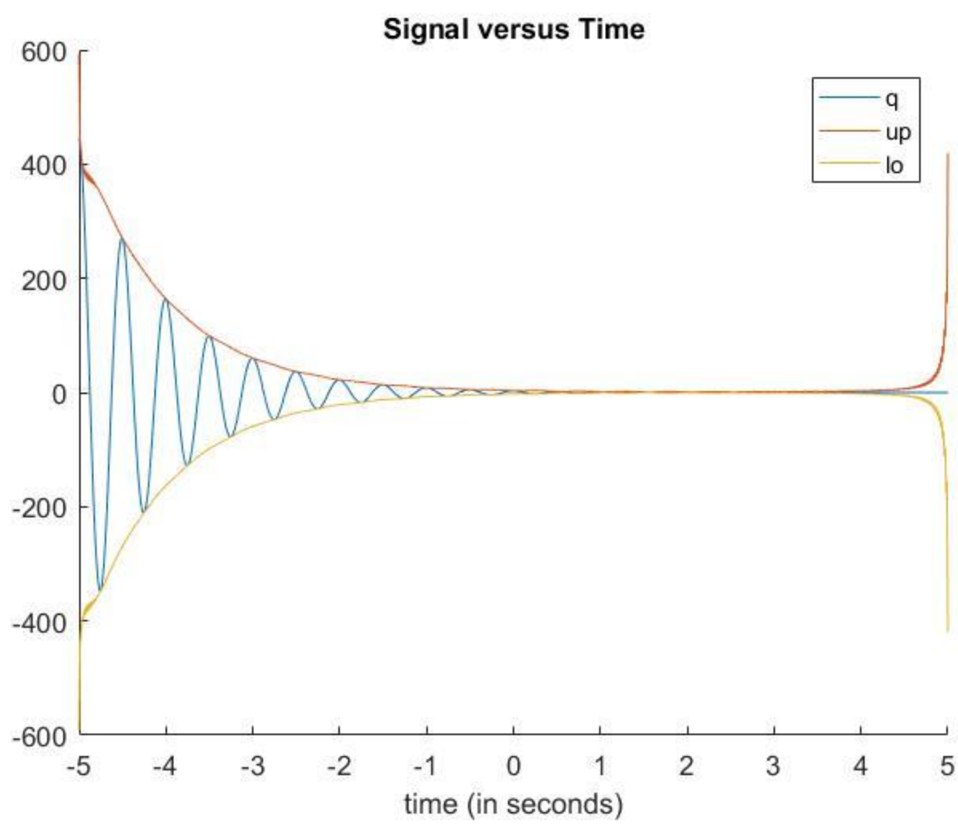
2. **For the damped sinusoidal signal $x(t) = 3e-t\cos(4\pi t)$ write a MATLAB program to generate x(t) and its envelope, then plot.**

%code

```
clc;

Ts=0.01;
t= -5:Ts:5;

q = 3 .*exp(-t) .* cos(4*pi*t);
% Plot the signal versus time:

[up,lo] = envelope(q);
figure;
hold on
plot(t,q,t,up,t,lo)
legend('q','up','lo')


xlabel('time (in seconds)');
```

```
title('Signal versus Time');

hold off
```

%output



PART 2: Time-Domain Convolution

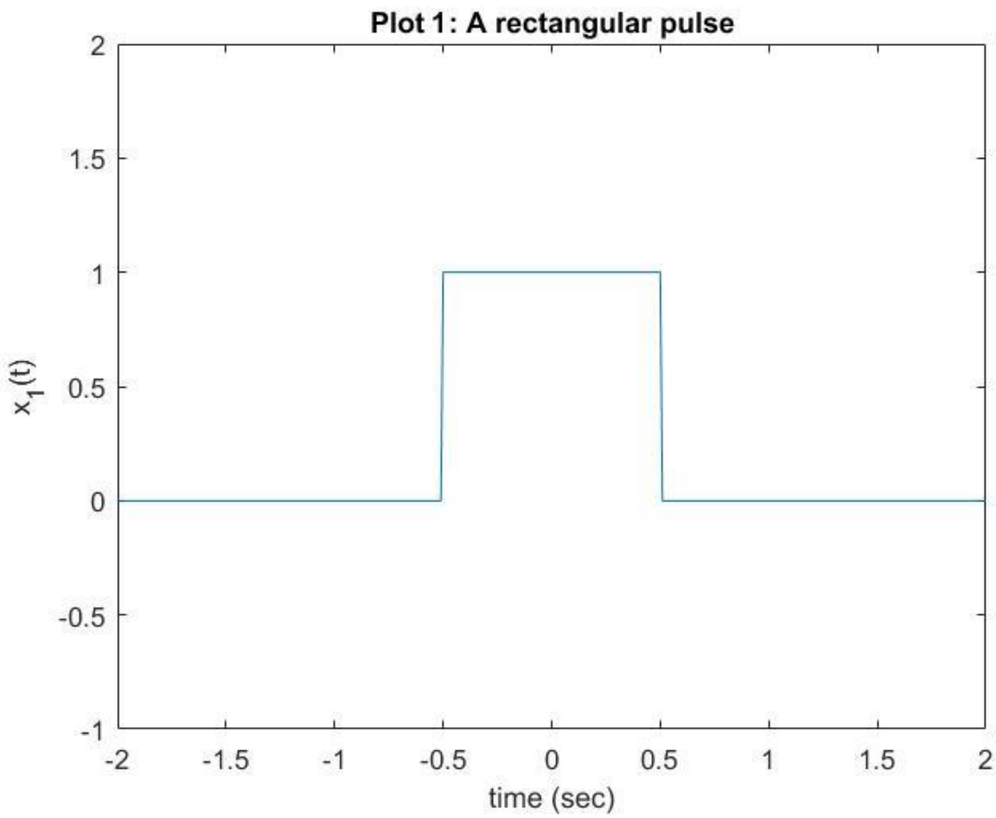
1. Creating a rectangular pulse in MATLAB

%code

```
clc;

f_s = 100;
T_s = 1/f_s;
```

```
t = -5:T_s:5;

x1 = rect(t);
plot(t,x1);

axis( [-2 2 -1 2]);

xlabel( 'time (sec)' );
ylabel( 'x_1(t)' );
title ('Plot 1: A rectangular pulse');


function x = rect(t)
    b = 0.5;
    a = -0.5;

    N = numel(t);
    x = zeros(N,1);
    start_time = find(t== a);
    end_time = find(t== b);

    pop_time = start_time:end_time;

    x(pop_time) = 1;
end
```

%output



2. Elementary signal operations

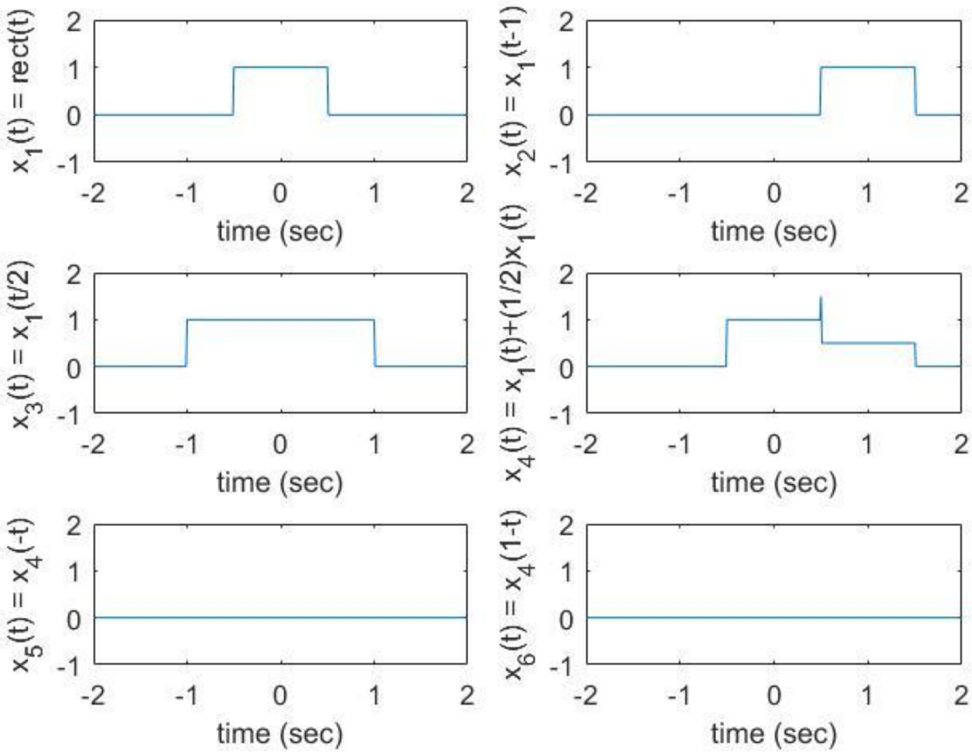
%code

```
clc;

f_s = 100;
T_s = 1/f_s;

t = -5:T_s:5;

b1 = 0.5;
a1 = -0.5;

x1 = rect(t);
plot(t,x1);

axis( [-2 2 -1 2]);
```

```
xlabel( 'time (sec)' );
ylabel( 'x_1(t)' );
title ('Plot 1: A rectangular pulse');

% %Elementary signal operations
x2 = rect(t-1);
plot(t,x2);
axis( [-2 2 -1 2])

x3 = rect(t/2);
plot(t,x3);
axis( [-2 2 -1 2]);

x4 = rect(t)+(1/2) * rect(t-1);

x5 = rect(-t)+(1/2) * rect(-t-1);

x6 = rect(1-t)+(1/2) * rect(-t);

subplot(3,2,1)
plot(t,x1)

axis( [-2 2 -1 2]);
xlabel( 'time (sec)' )
ylabel('x_1(t) = rect(t)')

subplot(3,2,2)
plot(t,x2)

axis( [-2 2 -1 2]);
xlabel( 'time (sec)' )
ylabel('x_2(t) = x_1(t-1)')

subplot(3,2,3)
plot(t,x3)

axis( [-2 2 -1 2]);
xlabel( 'time (sec)' )
ylabel('x_3(t) = x_1(t/2)')

subplot(3,2,4)
plot(t,x4)

axis( [-2 2 -1 2]);
xlabel( 'time (sec)' )
ylabel('x_4(t) = x_1(t)+(1/2)x_1(t)')

subplot(3,2,5)
plot(t,x5)

axis( [-2 2 -1 2]);
```

```
xlabel( 'time (sec)' )
ylabel('x_5(t) = x_4(-t)')

subplot(3,2,6)
plot(t,x6)

axis( [-2 2 -1 2]);
xlabel( 'time (sec)' )
ylabel('x_6(t) = x_4(1-t)')


function x = rect(t)
    b = 0.5;
    a = -0.5;

    N = numel(t);
    x = zeros(N,1);
    start_time = find(t== a);
    end_time = find(t== b);

    pop_time = start_time:end_time;

    x(pop_time) = 1;
end
```

%code



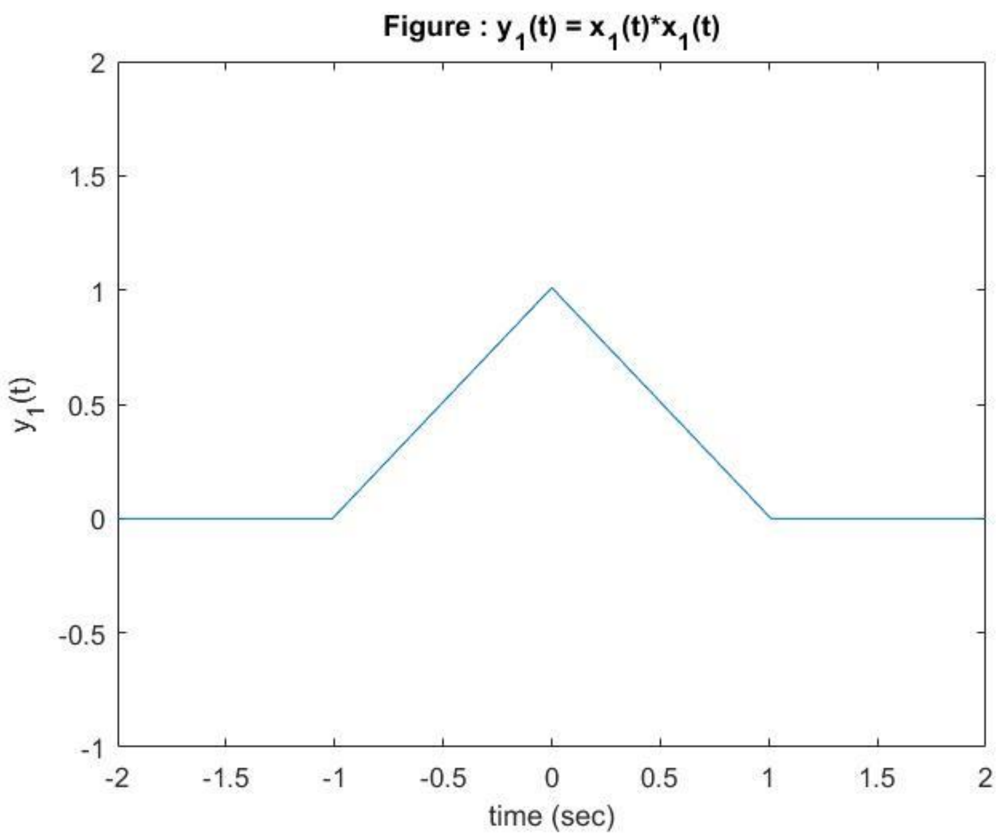
3. Convolution

%code

```
clc;

f_s = 100;
T_s = 1/f_s;

y = conv(x1,x1);

close all;

length(y)
length(t)

t_y = -10:T_s:10;

% plot ( t, y);

% plot( t_y, y)

y1 = T_s*conv(x1,x1);
```

```
plot(t_y, y1);

axis( [-2 2 -1 2] );
xlabel( 'time (sec)');
ylabel('y_1(t)')
title('Figure : y_1(t) = x_1(t)*x_1(t)');
```

%output



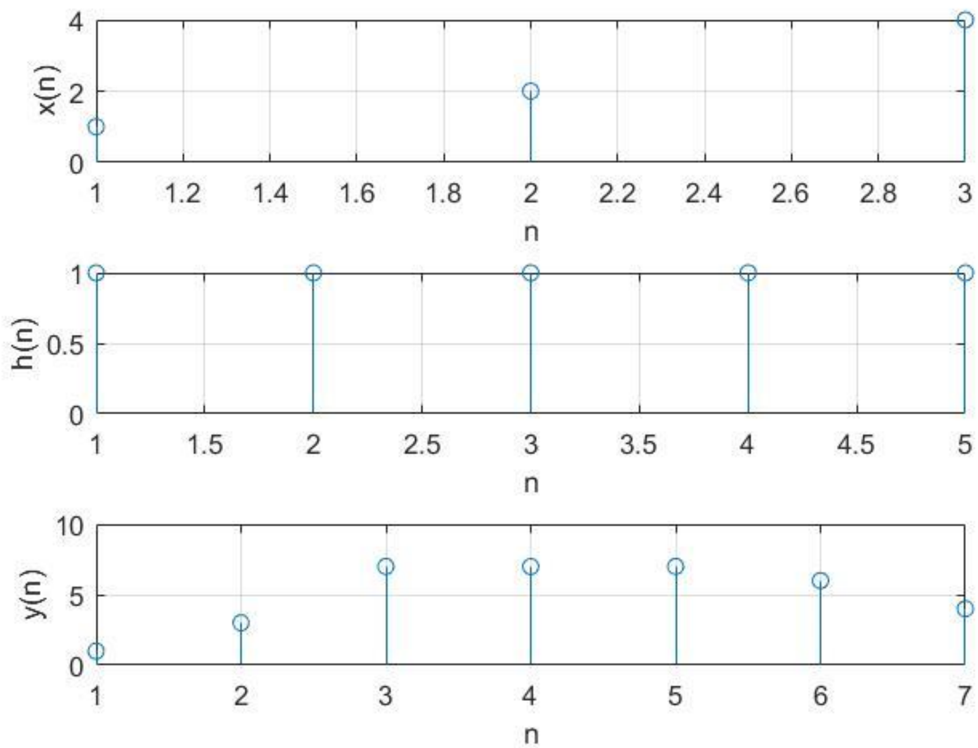
Exercise

1. **Perform convolution on discrete time signals x(n) and h(n), i.e., y(n) = x(n)*h(n) using MATLAB. For each set of signals, plot x(n), h(n) and y(n) as subplots in the same figure.**
   - **x(n) = { 1,2,4 }, h(n) = {1,1,1,1,1}**

%code

```
x = [ 1,2,4 ];
h = [1,1,1,1,1];

y = conv(x,h);

subplot(3,1,1);
stem(x);
grid
xlabel( 'n' ) ;
ylabel( 'x(n)' ) ;

subplot(3,1,2);
stem(h);
grid
xlabel( 'n' ) ;
ylabel( 'h(n)' ) ;

subplot(3,1,3);
stem(y);
grid
xlabel( 'n' ) ;
ylabel( 'y(n)' ) ;
```

%output



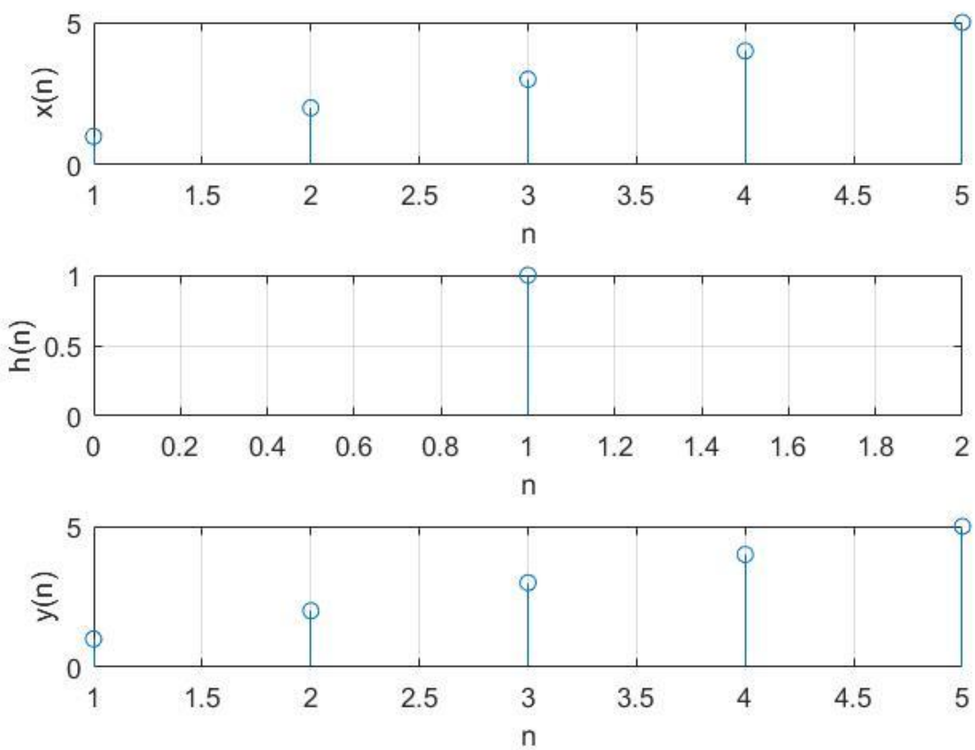
- **x(n) = { 1,2,3,4,5 }, h(n) = {1}**

%code

```
x = [ 1,2,3,4,5 ];
h = [1];

y = conv(x,h);

subplot(3,1,1);
stem(x);
grid
xlabel( 'n' ) ;
ylabel( 'x(n)' ) ;

subplot(3,1,2);
stem(h);
grid
xlabel( 'n' ) ;
ylabel( 'h(n)' ) ;

subplot(3,1,3);
stem(y);
grid
```

```
xlabel( 'n' ) ;
ylabel( 'y(n)' ) ;
```

%output



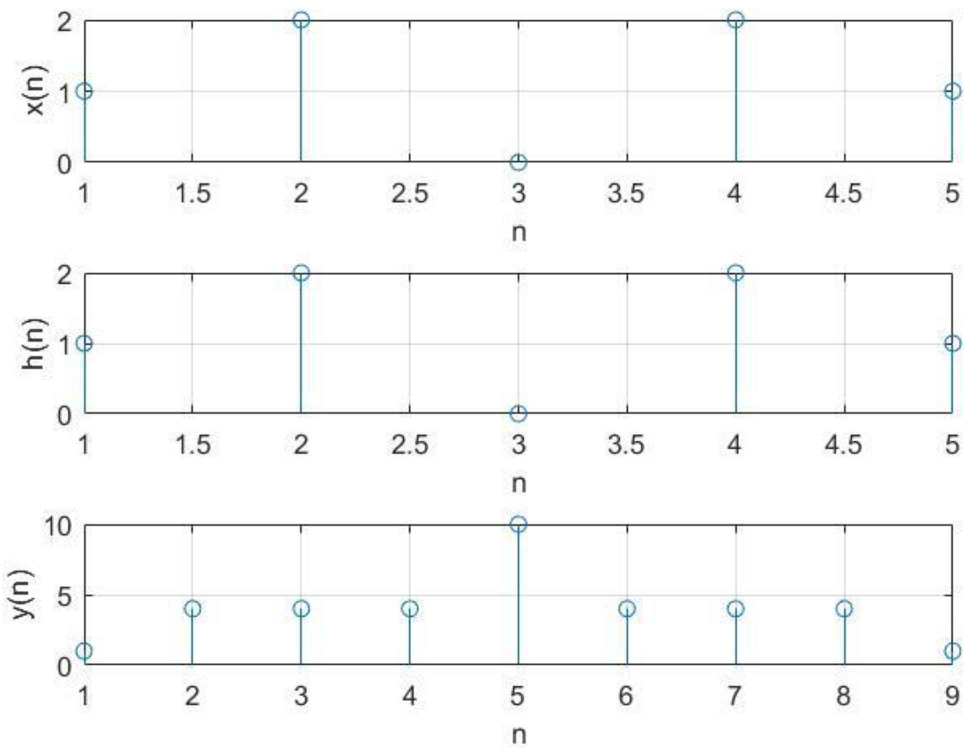
- **x(n) = h(n) ={ 1,2,0,2,1}**

%code

```
x = [ 1,2,0,2,1 ];
h = [ 1,2,0,2,1 ];

y = conv(x,h);

subplot(3,1,1);
stem(x);
grid
xlabel( 'n' ) ;
ylabel( 'x(n)' ) ;

subplot(3,1,2);
stem(h);
```

```
grid
xlabel( 'n' ) ;
ylabel( 'h(n)' ) ;

subplot(3,1,3);
stem(y);
grid
xlabel( 'n' ) ;
ylabel( 'y(n)' ) ;
```

%output



2. Assume a system with the following impulse response:

h(n) = (0.5)n   for 0<= n < 4

= 0   elsewhere

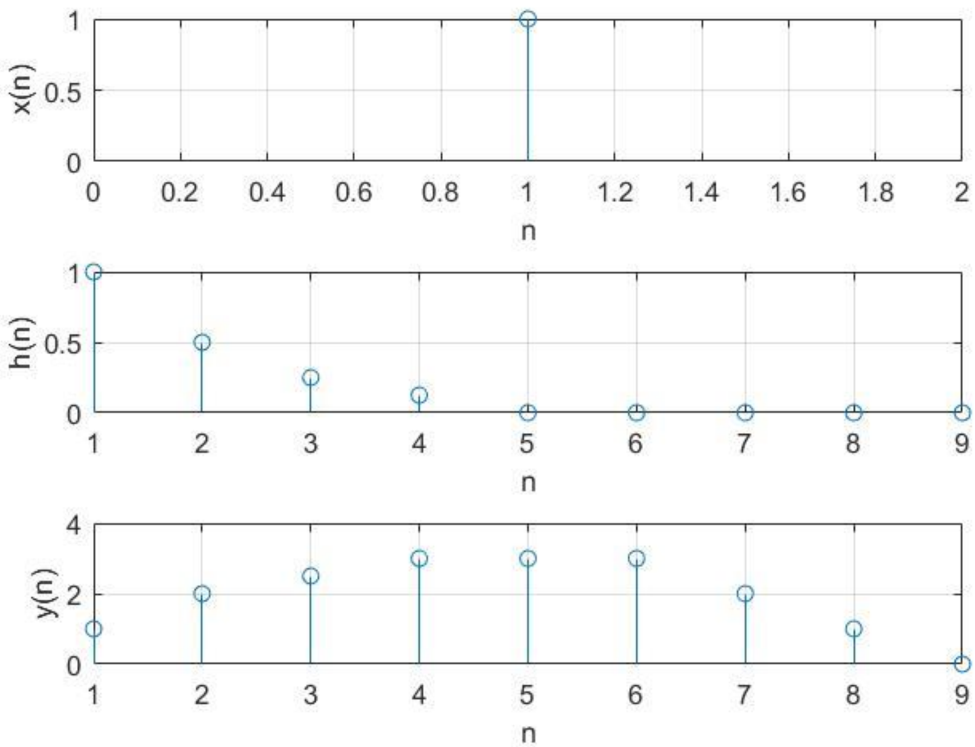
Determine the input x (n) that will generate the output sequence y(n) = {1, 2, 2.5, 3, 3, 3, 2, 1,0...}. Plot h(n), y (n) and x (n) in one figure.

%code

```
y = [1,2,2.5,3,3,3,2,1,0];

h = [1,0.5,0.25,0.125,0,0,0,0,0];

[x,r] = deconv(y,h)

subplot(3,1,1);
stem(x);
grid
xlabel( 'n' ) ;
ylabel( 'x(n)' ) ;

subplot(3,1,2);
stem(h);
grid
xlabel( 'n' ) ;
ylabel( 'h(n)' ) ;

subplot(3,1,3);
stem(y);
grid
xlabel( 'n' ) ;
ylabel( 'y(n)' ) ;

y1 = conv(x,h) + r
```

```
%output
```

X(n) = 2



1
x(n) 0.5
0
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2
n
1
h(n) 0.5
0
1 2 3 4 5 6 7 8 9
n
4
y(n) 2
0
1 2 3 4 5 6 7 8 9
n