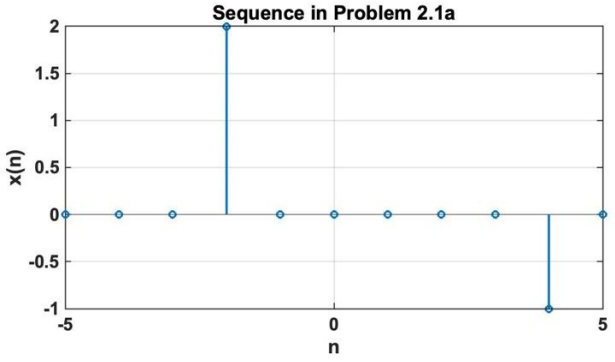
Experiment No: 01 [Example 2.1(a)]
Name of Experiment: Generate and Plot $x(n) = 2\delta(n+2) - \delta(n-4), \quad -5 \le n \le 5$

Program Code:

```
function [x,n] = impseq(n0,n1,n2)
n = [n1:n2]; x = [(n-n0) == 0];
clc;
clear;
close all;
n = [-5:5];
x = 2*impseq(-2,-5,5) - impseq(4,-5,5);
stem(n,x);
title('Sequence in Problem 2.1a');
xlabel('n');
ylabel('x(n)');
```

Output:



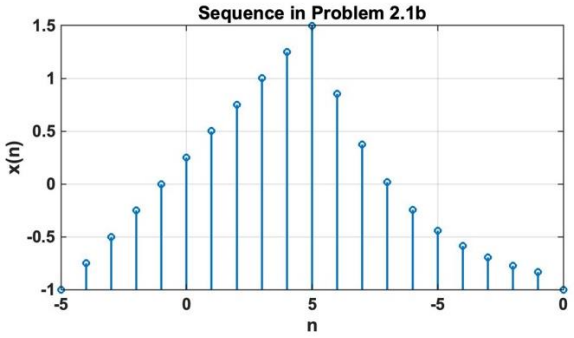
Experiment No: 02 [Example 2.1(b)]
Name of Experiment: Generate and Plot

$$x(n) = n[u(n) - u(n-10)] + 10e^{-0.3(n-10)}[u(n-10) - u(n-20)] \quad 0 \le n \le 20$$

Program Code:

```
function [x,n] = stepseq(n0,n1,n2)
n = [n1:n2]; x = [(n-n0) >= 0];
n = [0:20];
x1 = n.*(stepseq(0,0,20)-stepseq(10,0,20));
x2 = 10*exp(-0.3*(n-
10)).*(stepseq(10,0,20)-stepseq(20,0,20));
x = x1+x2;
subplot(2,2,3);
stem(n,x);
title('Sequence in Problem 2.1b');
xlabel('n');
ylabel('x(n)');
```

Output:



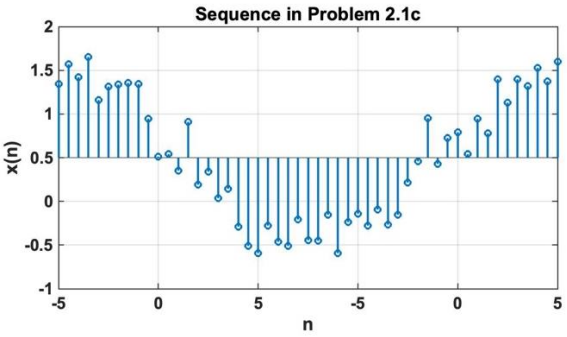
Experiment No: 03 [Example 2.1(c)]
Name of Experiment: Generate and Plot

$$x(n) = \cos(0.04\pi n) + 0.2\omega(n); \qquad 0 \le n \le 50$$

Program Code:

```
n = [0:50];
x = cos(0.04*pi*n)+0.2*randn(size(n));
subplot(2,2,2);
stem(n,x);
title('Sequence in Problem 2.1c');
xlabel('n');
ylabel('x(n)');
```

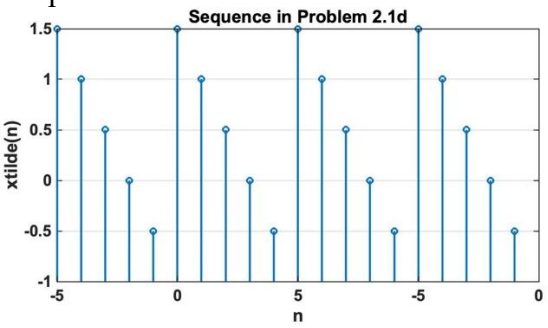
Output:



Experiment No: 04 [Example 2.1(d)]
Name of Experiment: Generate and Plot

$$\tilde{x}(n) = \{\ldots\ldots,5,4,3,2,1,5,4,3,2,1,5,4,3,2,1,\ldots\ldots\ldots\}; \qquad -10 \le n \le 10$$

Program Code:	Output:

n = [-10:9];
x = [5,4,3,2,1];
xtilde = x' * ones(1,4); xtilde = (xtilde(:))';
subplot(2,2,4);
stem(n,xtilde);
title('Sequence in Problem 2.1d');
xlabel('n');
ylabel('xtilde(n)');
	

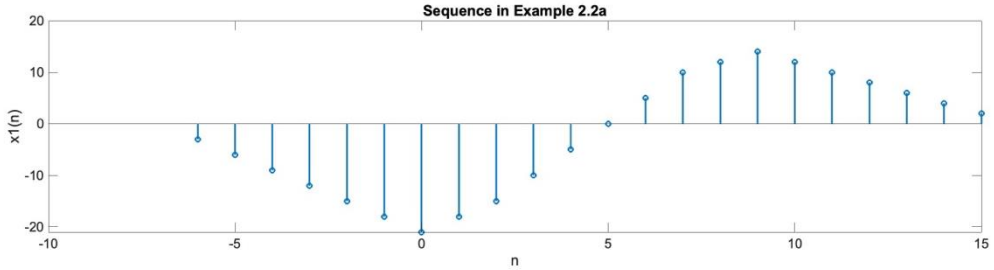
Experiment No: 05 [Example 2.2 (a, b)]
Name of Experiment: Determine and Plot, Let $x(n) = \{1,2,3,4,5,6,7,6,5,4,3,2,1\}$

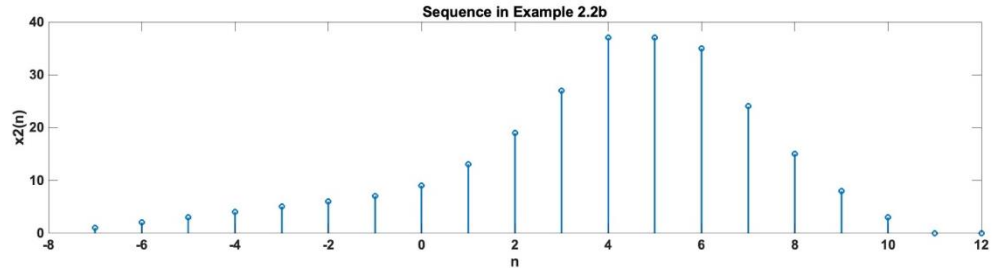
$$x_1(n) = 2x(n-5) - 3x(n+4)$$
$$x_2(n) = x(3-n) + x(n)x(n-2)$$

Program Code 2.2(a):	Program Code 2.2(b):

function [y,n] = sigadd(x1,n1,x2,n2)
n = min(min(n1),min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n)); y2 = y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y = y1+y2;
function [y,n] = sigshift(x,m,k)
n = m+k; y = x;
n = -2:10;
x = [1:7,6:-1:1];
[x11,n11] = sigshift(x,n,5);
[x12,n12] = sigshift(x,n,-4);
[x1,n1] = sigadd(2*x11,n11,-3*x12,n12);
subplot(2,1,1);
stem(n1,x1);
title('Sequence in Example 2.2a');
xlabel('n');
ylabel('x1(n)');
	function [y,n] = sigadd(x1,n1,x2,n2)
	n = min(min(n1),min(n2)):max(max(n1),max(n2));
	y1 = zeros(1,length(n)); y2 = y1;
	y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
	y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
	y = y1+y2;
	function [y,n] = sigshift(x,m,k)
	n = m+k; y = x;
	function [y,n] = sigmult(x1,n1,x2,n2)
	n = min(min(n1),min(n2)):max(max(n1),max(n2));
	y1 = zeros(1,length(n)); y2 = y1;
	y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
	y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
	y = y1 .* y2;
	function [y,n] = sigfold(x,n)
	y = fliplr(x); n = -fliplr(n);
	[x21,n21] = sigfold(x,n);
	[x21,n21] = sigshift(x21,n21,3);
	[x22,n22] = sigshift(x,n,2);
	[x22,n22] = sigmult(x,n,x22,n22);
	[x2,n2] = sigadd(x21,n21,x22,n22);
	subplot(2,1,2);
	stem(n2,x2);
	title('Sequence in Example 2.2b');
	xlabel('n');
	ylabel('x2(n)');

Output:





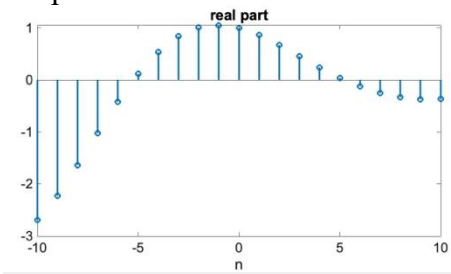
Experiment No: 06 [Example 2.3]
Name of Experiment: Generate the complex-valued signal

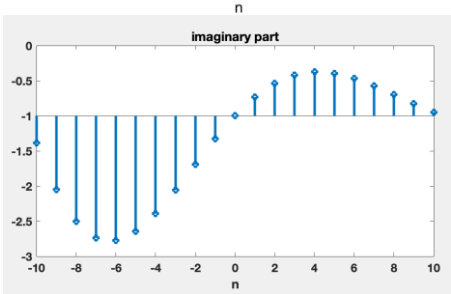
$$x(n) = e^{(-0.1+j0.3)n} \qquad -10 \leq n \leq 10$$

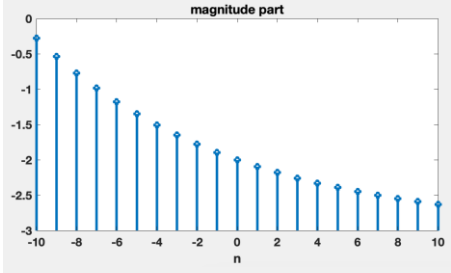
Program Code:

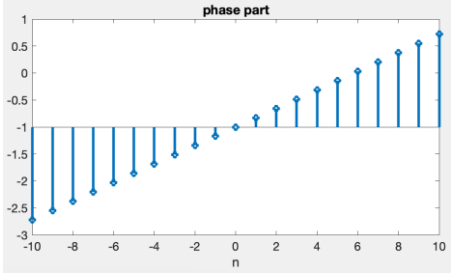
```
clc;
clear;
close all;

n = [-10:1:10];
alpha = -0.1+0.3j;
x = exp(alpha*n);
subplot(2,2,1);
stem(n,real(x));
title('real part');
xlabel('n');
subplot(2,2,2);
stem(n,imag(x));
title('imaginary part');
xlabel('n');
subplot(2,2,3);
stem(n,abs(x));
title('magnitude part');
xlabel('n');
subplot(2,2,4);
stem(n,(180/pi)*angle(x));
title('phase part');
xlabel('n');
```

Output:







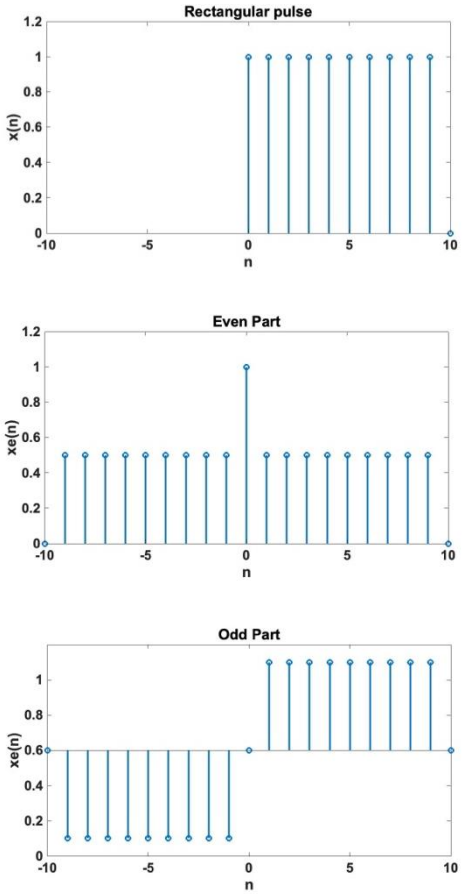


Experiment No: 07 [Example 2.4]
Name of Experiment: Let *x(n) = u(n) - u(n - 10).* Decompose c(n) into even and odd components.

Program Code:

```
function [x,n] = stepseq(n0,n1,n2)
n = [n1:n2]; x = [(n-n0) >= 0];
function [xe, xo, m] = evenodd(x,n)
if any(imag(x) ~= 0)
error('x is not a real sequence')
end
m = -fliplr(n);
m1 = min([m,n]); m2 = max([m,n]); m =
m1:m2;
nm = n(1)-m(1); n1 = 1:length(n);
x1 = zeros(1,length(m)); x1(n1+nm) = x; x =
x1;
xe = 0.5*(x + fliplr(x)); xo = 0.5*(x - fliplr(x));
n = [0:10];
x = stepseq(0,0,10)-stepseq(10,0,10);
[xe,xo,m] = evenodd(x,n);
subplot(2,2,1);
stem(n,x);
title('Rectangular pulse');
xlabel('n');
ylabel('x(n)');
axis([-10,10,0,1.2]);
subplot(2,2,2);
stem(m,xe);
title('Even Part');
xlabel('n');
ylabel('xe(n)');
axis([-10,10,0,1.2]);
subplot(2,2,4);
stem(m,xo);
title('Odd Part');
xlabel('n');
ylabel('xe(n)');
axis([-10,10,-0.6,0.6]);
```

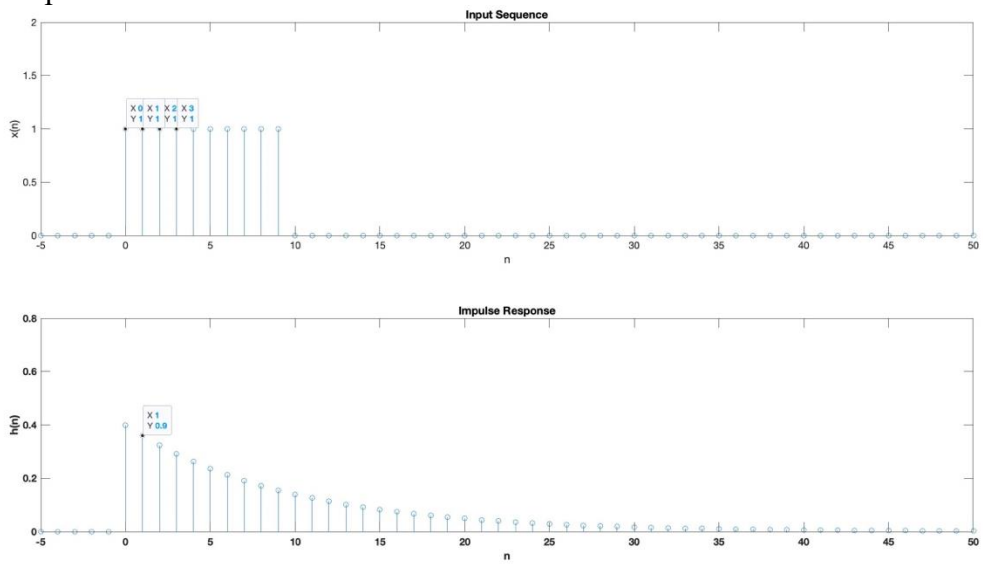
Output:



Experiment No: 08 [Example 2.5]
Name of Experiment: Let the rectangular pulse *x(n) = u(n) - u(n - 10)* of Example 2.4 be an input to an LTI system with impulse response $h(n) = (0.9)^n u(n)$ Determine y(n).

```
Program Code:
function [x,n] = stepseq(n0,n1,n2)
n = [n1:n2]; x = [(n-n0) >= 0];
n = -5:50;
u1 = stepseq(0,-5,50); u2=stepseq(10,-5,50);
x = u1-u2;
h = ((0.9).^n).*u1;
subplot(1,1,1)
subplot(2,1,1); stem(n,x); axis([-5,50,0,2])
title('Input Sequence')
xlabel('n'), ylabel('x(n)')
subplot(2,1,2); stem(n,h); axis([-5,50,0,2])
title('Impulse Response')
xlabel('n'), ylabel('h(n)'); pause
print -deps2 ex0205a.eps
% output response
y = (10*(1-(0.9).^(n+1))).*(u1-u2)+(10*(1-
(0.9)^10)*(0.9).^(n-9)).*u2;
subplot(1,1,1)
subplot(2,1,2); stem(n,y); axis([-5,50,0,8])
title('Output Sequence')
xlabel('n'), ylabel('y(n)')
print -deps2 ex0205b.eps
```

Output:



Experiment No: 09 [Example 2.6]
Name of Experiment: Give the following two sequences

$$x(n) = [3,11,7,0,-1,4,2] \qquad -3 \le n \le 3$$
$$and\ h(n) = [2,3,0,-5,2,1] \qquad -1 \le n \le 4$$

```
Program Code:
x = [3,11,7,0,-1,4,2];
h = [2,3,0,-5,2,1];
y = conv(x,h);
```

```
Output:
y =
     6    31    47     6   -51    -5    41    18   -22    -3     8     2
```

Experiment No: 10 [Example 2.7]
Name of Experiment: Perform the convolution in Example 2.6 using the conv_m function.

```
Program Code:
x = [3,11,7,0,-1,4,2];
nx = [-3:3];
h = [2,3,0,-5,2,1];
ny = [-1:4];
[y,ny] =
conv_m(x,nx,h,ny);
```

```
Output:
y =
     6    31    47     6   -51    -5    41    18   -22    -3     8     2

ny =
    -4    -3    -2    -1     0     1     2     3     4     5     6     7
```

Experiment No: 11 [Example 2.8]
Name of Experiment: Cross correlation sequence. Let

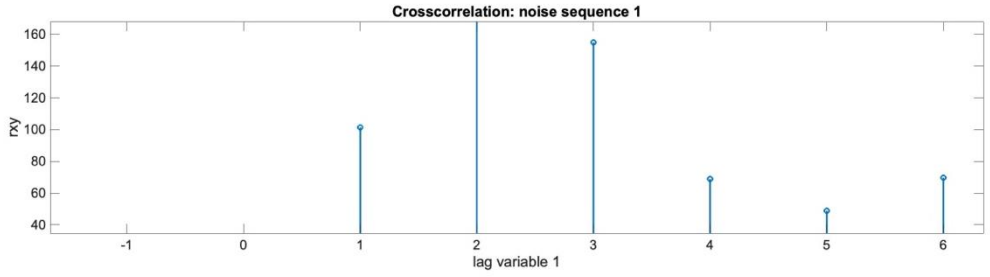
$$x(n) = [3,11,7,0,-1,4,2]$$
$$y(n) = x(n-2) + \omega n$$

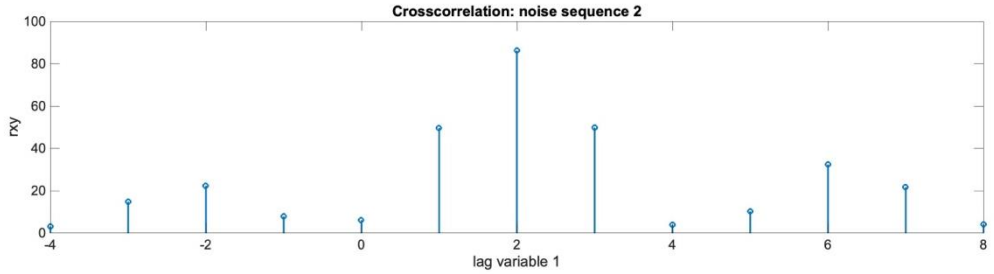
Program Code:

```
function [y,ny] = conv_m(x,nx,h,nh)
nyb = nx(1)+nh(1); nye = nx(length(x)) + nh(length(h));
ny = [nyb:nye]; y = conv(x,h);
function [y,n] = sigfold(x,n)
y = fliplr(x); n = -fliplr(n);
function [y,n] = sigadd(x1,n1,x2,n2)
n = min(min(n1),min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n)); y2 = y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y = y1+y2;
function [y,n] = sigshift(x,m,k)
n = m+k; y = x;
%noise sequence 1
x = [3,11,7,0,-1,4,2];
nx=[-3:3];
[y,ny] = sigshift(x,nx,2);
w=randn(l,length(y));
nw=ny;
[y,ny] = sigadd(y,ny,w,nw);
[x,nx] = sigfold(x,nx);
[rxy,nrxyl = conv_m(y,ny,x,nx);
subplot(1,1,1), subplot(2,1,1);
stem(nrxy,rxy);
axis([-5,10,-50,25O]);
xlabel('lag variable 1')
ylabel('rxy');
title('Crosscorrelation: noise sequence 1');
```

```
%noise sequence 2
x = [3, 11, 7, 0, -1, 4, 2];
nx = [-3:3];
[y,nyl= sigshift(x,nx,2);
w = randn(1,length(y));
nw = ny;
[y,ny] = sigadd(y,ny,w,nw);
[x,nx] = sigfold(x,nx);
[rxy,nrxy] = conv_m(y,ny,x,nx);
subplot(2,1,2);
stem(nrxy,rxy);
axis([-5,10,-50,250);
xlabel('lag variable 1');
ylabel('rxy');
title('Crosscorrelation: noise
sequence 2');
```

Output:





Experiment No: 12 [Example 2.9 (a, b)]
Name of Experiment: Given the following difference equation

$$y(n) - y(n-1) + 0.9y(n-2) = x(n)$$

a. Calculate and plot the impulse response $h(n)\ at\ n = -20, \ldots\ldots\ldots, 100.$
b. Calculate and Plot the unit step sequence response $\delta(n)\ at\ n = -20, \ldots\ldots\ldots, 100.$

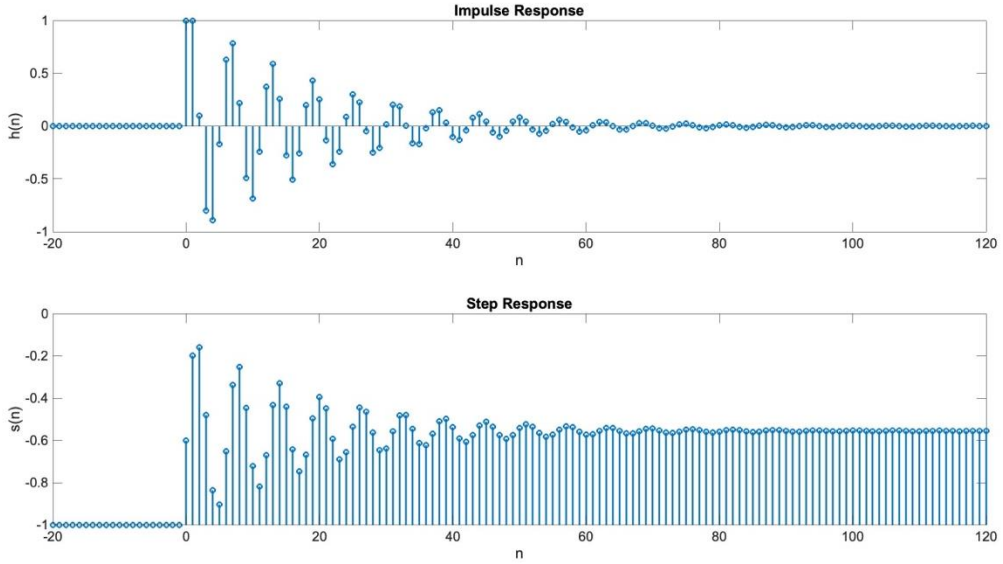
Program Code 2.9(a):

```
function [x,n] = impseq(n0,n1,n2)
n = [n1:n2]; x = [(n-n0) == 0];
b = [1];
a = [1,-1,0.9];
x = impseq(0,-20,120);
n = [-20:120];
h = filter(b,a,x);
subplot(2,1,1);
stem(n,h);
title('Impulse Response');
xlabel('n');
ylabel('h(n)');
```

Program Code 2.9(b):

```
function [x,n] = stepseq(n0,n1,n2)
n = [n1:n2]; x = [(n-n0) >= 0];
x = stepseq(0,-20,120);
s = filter(b,a,x);
subplot(2,1,2);
stem(n,s);
title('Step Response');
xlabel('n');
ylabel('s(n)');
```

Output:



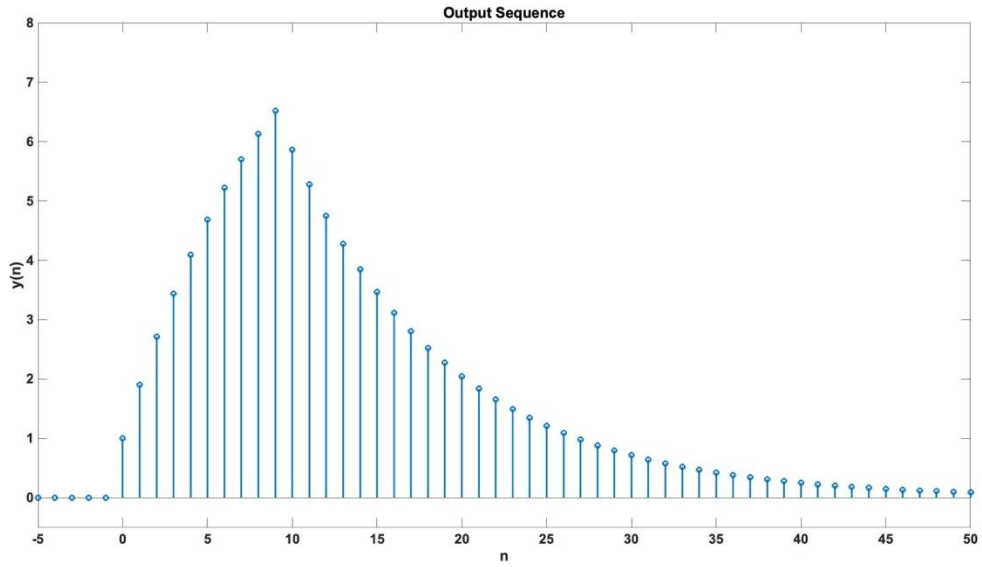
Experiment No: 13 [Example 2.10]
Name of Experiment: Let us consider the convolution given in Example 2.5. The input sequence is of finite duration. $x(n) = u(n) - u(n-10)$. While the impulse response is of infinite duration. $h(n) = (0.9)^n u(n)$

Determine $y(n) = x(n) * h(n)$

Program Code:

```
function [x,n] = stepseq(n0,n1,n2)
n = [n1:n2]; x = [(n-n0) >= 0];
b = [1];
a = [1,-0.9];
n = -5:50;
x = stepseq(0,-5,50) - stepseq(10,-5,50);
y = filter(b,a,x);
subplot(1,1,1);
subplot(2,1,2);
stem(n,y);
title('Output Sequence');
xlabel('n');
ylabel('y(n)');
axis([-5,50,-0.5,8]);
```

Output:



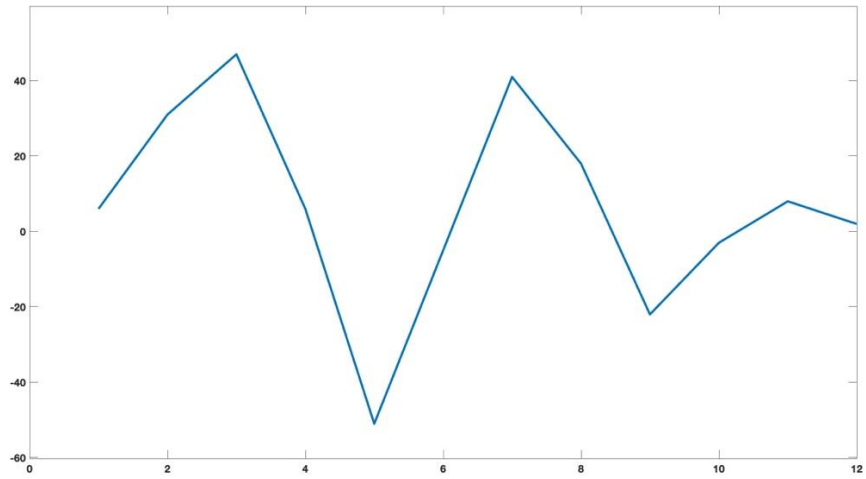
Experiment No: 14
Name of Experiment: Linear Convolution.

Program Code:

```
function[y] = linearconvulation(x,h)
n1 = length(x);
n2 = length(h);
N = (n1+n2) - 1;
x1 = [x zeros(1, N-n1)];
h1 = [h zeros(1, N-n2)];

for n = 1:N
   for m = 1:N
      if n>m
         H(m,n) = 0;
      else
         H(m,n) = h1(m - (n-1));
      end
   end
end
y = H * x1';
end
```

```
clc;
clear;
close all;
x = [3,11,7,0,-1,4,2];
h = [2,3,0,-5,2,1];
[linearoutput] = linearconvulation(x,h);
figure(1);
plot(linearoutput);
```

Output:



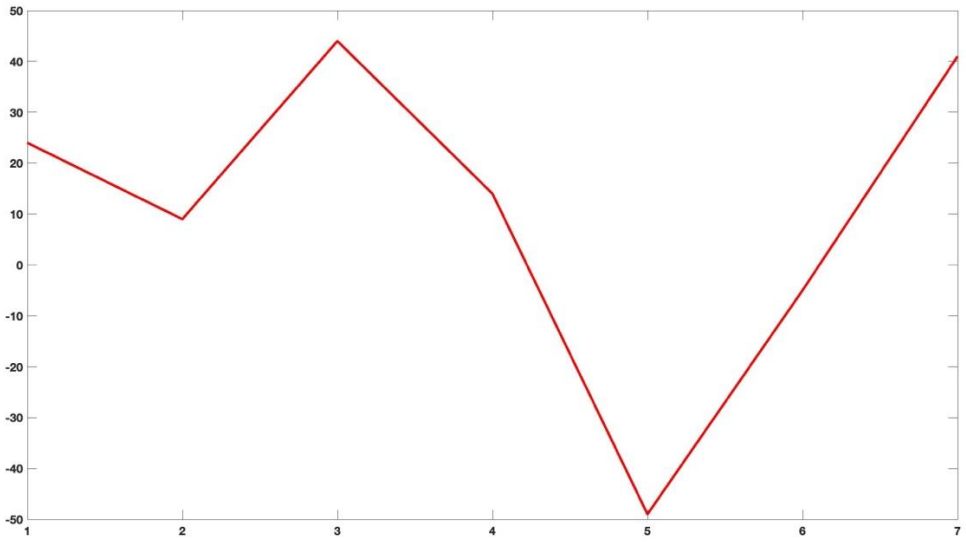
Experiment No: 15
Name of Experiment: Circular Convolution.

Program Code:

```
function[z] = circularconvulation(x1,x2)
n1 = length(x1);
n2 = length(x2);
N = max(n1,n2);
x1 = [x1 zeros(1, N-n1)];
x2 = [x2 zeros(1, N-n2)];
s = n1 - n2;
if(s==0)
   x2 = [x2 zeros(1,s)];
else
   x1 = [x1 zeros(1,-s)];
   x2 = [x2 zeros(1,s)];
end
%circular multiplication
z = [];
for k = 1:N
   y = 0;
   for i = 1:N
      j = (k-i) + 1;
      if(j<=0)
         j = j + N;
      end
      y = y + (x1(i) * x2(j));
   end
   z = [z,y];
end
end
```

```
clc;
clear;
close all;
x = [3,11,7,0,-1,4,2];
h = [2,3,0,-5,2,1];
[circularoutput] = circularconvulation(x,h);
figure(1);
plot(circularoutput);
```

Output:

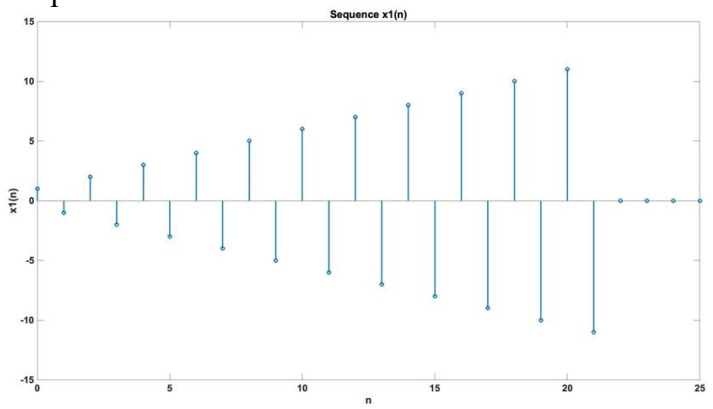


Experiment No: 16 [Problem 2.1(a)]
Name of Experiment: $x_1(n) = \sum_{m=0}^{10}(m+1)[\delta(n-2m) - \delta(n-2m-1)];\ 0 \le n \le 25$

Program Code:

```
function [x,n] = impseq(n0,n1,n2)
n = [n1:n2]; x = [(n-n0) == 0];
clc;
clear;
close all;
n1 = [0:25];
x1 = zeros(1,length(n1));
for m = 0:10
    x1 = x1 +
(m+1)*(impseq(2*m,0,25) -
impseq(2*m+1,0,25));
end
stem(n1,x1);
xlabel('n');
ylabel('x1(n)');
title('Sequence x1(n)');
```

Output:



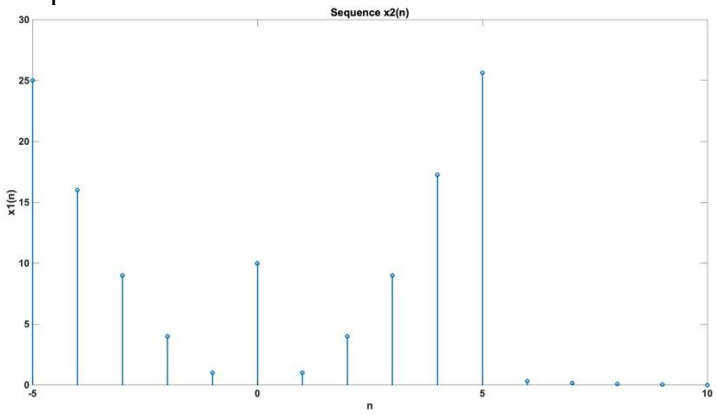
Experiment No: 17 [Problem 2.1(b)]
Name of Experiment:

$$x_2(n) = n^2[u(n+5) - n(n-6)] + 10\delta(n) + 20(0.5)^n[u(n-4) - u(n-10)]$$

Program Code:

```
function [x,n] = impseq(n0,n1,n2)
n = [n1:n2]; x = [(n-n0) == 0];
function [x,n] = stepseq(n0,n1,n2)
n = [n1:n2]; x = [(n-n0) >= 0];
clc;
clear;
close all;
n2 = -5:10;
x2 = (n2.^2).*(stepseq(-5,-5,10)-
stepseq(6,-5,10))+10*impseq(0,-
5,10)+20*((0.5).^n2).*(stepseq(4,
-5,10)-stepseq(10,-5,10));
stem(n2,x2);
xlabel('n');
ylabel('x1(n)');
title('Sequence x2(n)');
```

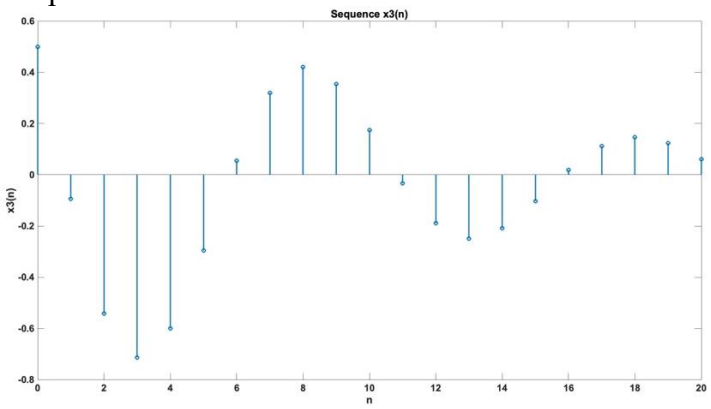
Output:



Experiment No: 18 [Problem 2.1(c)]
Name of Experiment:

$$x_3(n) = (0.9)^n \cos\left(0.2\pi n + \frac{\pi}{3}\right) \qquad 0 \le n \le 20$$

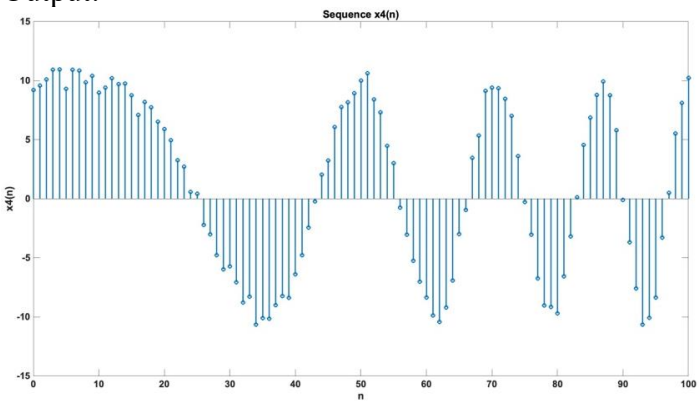
Program Code:	Output:

clc;
clear;
close all;
 	
n3 = [0:20];
x3 =
((0.9).^n3).*cos(0.2*pi*n3+pi/3);
stem(n3,x3);
xlabel('n');
ylabel('x3(n)');
title('Sequence x3(n)');
	

Experiment No: 19 [Problem 2.1(d)]
Name of Experiment:

$$x_4(n) = 10\cos(0.0008\pi n^2) + \omega(n) \qquad 0 \le n \le 100$$

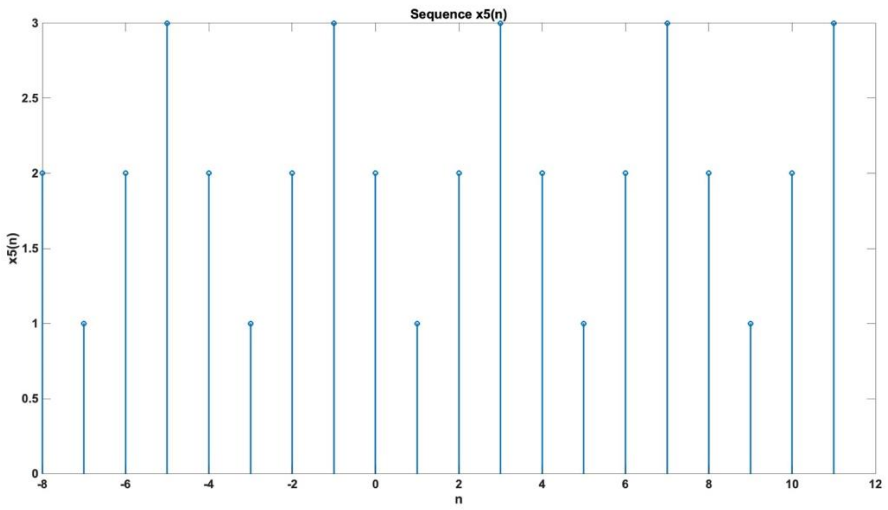
Program Code:	Output:

clc;
clear;
close all;
 	
w = 2*(rand(1,101)-0.5);
n4 = [0:100];
x4 = 10*cos(0.0008*pi*n4.^2)+w;
stem(n4,x4);
xlabel('n');
ylabel('x4(n)');
title('Sequence x4(n)');
	

Experiment No: 20 [Problem 2.1(e)]
Name of Experiment:

$$\tilde{x}_5(n) = \{\dots\dots,1,2,3,2,1,2,3,2,1,\dots\dots.\} \quad Plot\ 5\ Periods$$

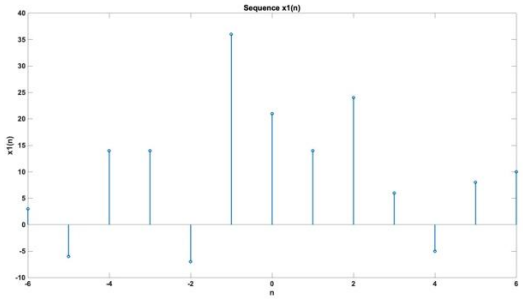
Program Code:	Output:

clc;
clear;
close all;
 	
n5 = [-8:11];
x5 = [2,1,2,3];
x5 = x5'*ones(1,5);
x5 = (x5(:))';
stem(n5,x5);
xlabel('n');
ylabel('x5(n)');
title('Sequence x5(n)');
	

Experiment No: 21 [Problem 2.2(a)]
Name of Experiment: Let $x(n) = \{1, -2, 4, 6, -5, 8, 10\}$

$$x_1(n) = 3x(n+2) + x(n-4) - 2x(n)$$

Program Code:	Output:

function [y,n] = sigshift(x,m,k)
n = m+k; y = x;
function [y,n] = sigadd(x1,n1,x2,n2)
n =
min(min(n1),min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n)); y2 = y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y = y1+y2;
n = [-4:2];
x = [1,-2,4,6,-5,8,10];
 	
[x11,n11] = sigshift(3*x,n,-2);
[x12,n12] = sigshift(x,n,4);
[x13,n13] = sigadd(x11,n11,x12,n12);
[x1,n1] = sigadd(x13,n13,2*x,n);
stem(n1,x1);
xlabel('n');
ylabel('x1(n)');
title('Sequence x1(n)');
	

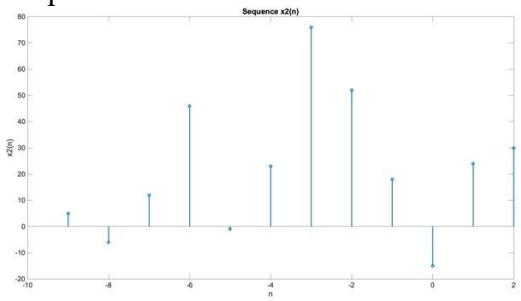
Experiment No: 22 [Problem 2.2(b)]
Name of Experiment: Let $x(n) = \{1, -2, 4, 6, -5, 8, 10\}$

$$x_2(n) = 5x(5+n) + 4x(n+4) + 3x(n)$$

Program Code:

```
function [y,n] = sigshift(x,m,k)
n = m+k; y = x;
function [y,n] = sigadd(x1,n1,x2,n2)
n =
min(min(n1),min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n)); y2 = y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y = y1+y2;
n = [-4:2];
x = [1,-2,4,6,-5,8,10];

[x21,n21] = sigshift(5*x,n,-5);
[x22,n22] = sigshift(4*x,n,-4);
[x23,n23] = sigadd(x21,n21,x22,n22);
[x2,n2] = sigadd(x23,n23,3*x,n);
stem(n2,x2);
xlabel('n');
ylabel('x2(n)');
title('Sequence x2(n)');
```

Output:



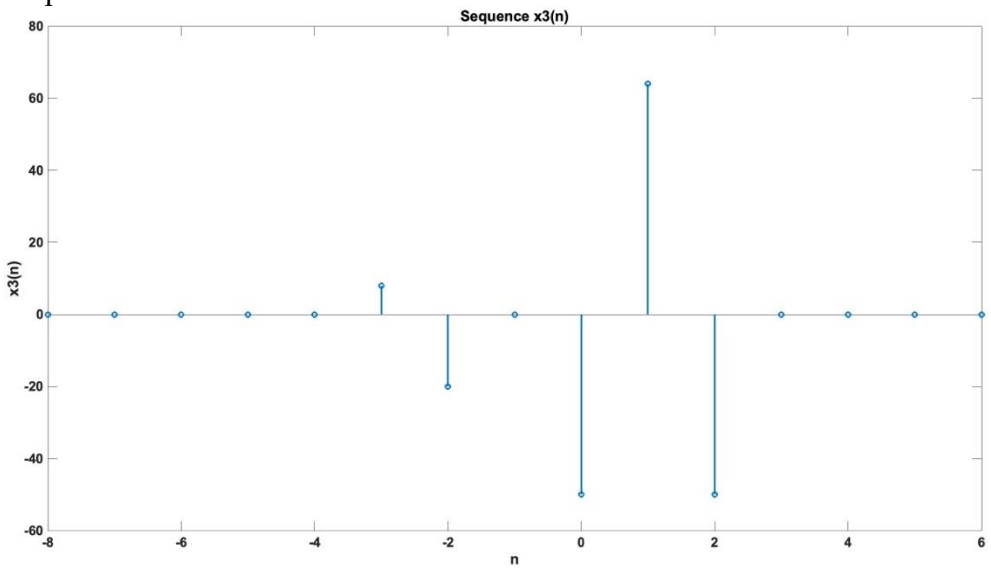
Experiment No: 23 [Problem 2.2(c)]
Name of Experiment: Let $x(n) = \{1, -2, 4, 6, -5, 8, 10\}$

$$x_3(n) = x(n+4)x(n-1) + x(2-n)x(n)$$

Program Code:

```
function [y,n] = sigshift(x,m,k)
n = m+k; y = x;
function [y,n] = sigadd(x1,n1,x2,n2)
n =
min(min(n1),min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n)); y2 = y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y = y1+y2;
function [y,n] = sigmult(x1,n1,x2,n2)
n =
min(min(n1),min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n)); y2 = y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y = y1 .* y2;
function [y,n] = sigfold(x,n)
y = fliplr(x); n = -fliplr(n);

n = [-4:2];
x = [1,-2,4,6,-5,8,10];

[x31,n31] = sigshift(x,n,-4);
[x32,n32] = sigshift(x,n,1);
[x33,n33] = sigmult(x31,n31,x32,n32);
[x34,n34] = sigfold(x,n);
[x34,n34] = sigshift(x34,n34,2);
[x34,n34] = sigmult(x34,n34,x,n);
[x3,n3] = sigadd(x33,n33,x34,n34);
stem(n3,x3);
xlabel('n');
ylabel('x3(n)');
title('Sequence x3(n)');
```

Output:



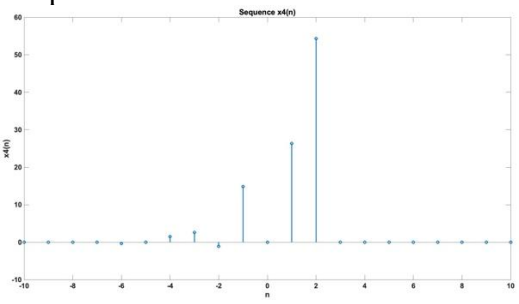
Experiment No: 24 [Problem 2.2(d)]
Name of Experiment: Let $x(n) = \{1, -2, 4, 6, -5, 8, 10\}$

$$x_4(n) = 2e^{0.5n}x(n) + \cos(0.1\pi n)\, x(n+2), \quad -10 \le n \le 10$$

Program Code:

```
function [y,n] = sigshift(x,m,k)
n = m+k; y = x;
function [y,n] = sigadd(x1,n1,x2,n2)
n =
min(min(n1),min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n)); y2 = y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y = y1+y2;
function [y,n] = sigmult(x1,n1,x2,n2)
n =
min(min(n1),min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n)); y2 = y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y = y1 .* y2;
n4 = [-10:10];
n = [-4:2];
x = [1,-2,4,6,-5,8,10];
x41 = 2*exp(0.5*n4);
x412 = cos(0.1*pi*n4);
[x42,n42] = sigmult(x41,n4,x,n);
[x43,n43] = sigshift(x,n,-2);
[x44,n44] = sigmult(x412,n4,x43,n43);
[x4,n4] = sigadd(x42,n42,x44,n44);
stem(n4,x4);
xlabel('n');
ylabel('x4(n)');
title('Sequence x4(n)');
```

Output:



Experiment No: 25 [Problem 2.2(e)]
Name of Experiment: Let $x(n) = \{1, -2,4,6, -5,8,10\}$

$$x_5(n) = \sum_{k=1}^{5} nx(n-k)$$

Program Code:

```
function [y,n] = sigshift(x,m,k)
n = m+k; y = x;
function [y,n] = sigadd(x1,n1,x2,n2)
n =
min(min(n1),min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n)); y2 = y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y = y1+y2;
function [y,n] = sigmult(x1,n1,x2,n2)
n =
min(min(n1),min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n)); y2 = y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y = y1 .* y2;
```

```
n = [-4:2];
x = [1,-2,4,6,-5,8,10];
[x51,n51] = sigshift(x,n,1);
[x52,n52] = sigshift(x,n,2);
[x5,n5] = sigadd(x51,n51,x52,n52);
[x53,n53] = sigshift(x,n,3);
[x5,n5] = sigadd(x5,n5,x53,n53);
[x54,n54] = sigshift(x,n,4);
[x5,n5] = sigadd(x5,n5,x54,n54);
[x55,n55] = sigshift(x,n,5);
[x5,n5] = sigadd(x5,n5,x55,n55);
[x5,n5] = sigmult(x5,n5,n5,n5);
stem(n5,x5);
xlabel('n');
ylabel('x5(n)');
title('Sequence x5(n)');
```

Output: