Experiment No: 01

Experiment Name: To explain and implement Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT).

Theory:

DFT: DFT converts a finite sequence of equally spaced samples of a function into a same length sequence of equally-spaced samples of the discrete-time Fourier transform which is a complex valued function of frequency.

DFT convert the time domain sequence to an equivalent frequency domain.

Considering $x[n]$ as an N-point sequence. Hence, DFT of $x[n]$ is given by

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}nk}$$

IDFT: The Fourier transform converts a time domain signal into a frequency domain. This frequency domain representation is exactly the same signal but in different form. The IDFT brings the signal back to the time domain from the frequency domain.

And the IDFT is given by,

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}nk}$$

Let us consider an example, and we have to determine DFT and IDFT of the given signal.

$x(n) = \{1, 1, 1, 1\}$

$N = L = 4$

The DFT is given by,

$$X[K] = \sum_{n=0}^{N-1} x(n) e^{-J\frac{2\pi}{N}nk}, \quad K = 0, 1, \cdots (N-1)$$

$$K = 0, 1, 2, 3$$

$$= \sum_{0}^{3} x(n) e^{-j\frac{\pi}{2}nk}$$

when $k=0$,

$$X[0] = \sum_{0}^{3} x(n) e^{0}$$
$$= x(0)+x(1)+x(2)+x(3)$$
$$= 1+1+1+1$$
$$= 4$$

when $k=1$,

$$X[1] = \sum_{0}^{3} x(n) e^{-j\frac{\pi}{2}n}$$

$$= x(0)e^{0} + x(1)e^{-j\frac{\pi}{2}} + x(2)e^{-j\pi} + x(3)e^{-j\frac{3\pi}{2}}$$

$$= 1 + 1\left(\cos \pi/2 - j\sin\frac{\pi}{2}\right) + 1(\cos\pi - j\sin\pi) + 1\left(\cos\frac{3\pi}{2} - j\sin\frac{3\pi}{2}\right)$$

$$= 0$$

For $k=2$,

$$X[2] = \sum_{0}^{3} x(n) e^{-jn\pi}$$

$$= x(0)e^{-0} + x(1)e^{-j\pi} + x(2)e^{-j2\pi} + x(3)e^{-j3\pi}$$

$$= 1 + 1[\cos\pi - j\sin\pi] + 1[\cos 2\pi - j\sin 2\pi] + 1[\cos 3\pi - j\sin 3\pi]$$

$$= 0$$

for $k=3$,

$$X[3] = \sum_{0}^{3} x(n)\, e^{-j\frac{3\pi}{2}n}$$

$$= x(0)e^{0} + x(1)e^{-j\frac{3\pi}{2}} + x(2)e^{-j3\pi}$$

$$+ x(3)e^{-j\frac{9\pi}{2}}$$

$$= 0$$

Therefore DFT of $x(n)$ is $X(k)=\{4,0,0,0\}$

To find IDFT,

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, e^{j\frac{2\pi}{N}nk} \qquad n=0,1,\cdots(N-1)$$

$$x(n) = \frac{1}{4}\sum_{0}^{3} X(k)\, e^{j\frac{\pi}{2}nk} \qquad n=0,1,2,3$$

when $n=0$,

$$x(0) = \frac{1}{4}\sum_{0}^{3} X(k)e^{0}$$

$$= \frac{1}{4}\left[X(0)+X(1)+X(2)+X(3)\right]$$

$$= \frac{1}{4}\left[4+0+0+0\right]$$

$$= 1$$

When $n=1$,

$$x(1) = \frac{1}{4} \sum_{k=0}^{3} X(k) e^{j\frac{\pi}{2}k}$$

$$= \frac{1}{4} \left[X(0)e^0 + X(1)e^{j\frac{\pi}{2}} + X(2)e^{j\pi} + X(3)e^{j\frac{3\pi}{2}} \right]$$

$$= \frac{1}{4} [4+0+0+0]$$

$$= 1$$

When $n=2$,

$$x(2) = \frac{1}{4} \sum_{k=0}^{3} X(k) e^{j\pi k}$$

$$= \frac{1}{4} \left[X(0)e^0 + X(1)e^{j\pi} + X(2)e^{j2\pi} + X(3)e^{j3\pi} \right]$$

$$= \frac{1}{4} [4+0+0+0]$$

$$= 1$$

When $n=3$

$$x(3) = \frac{1}{4} \sum_{0}^{3} X(k) e^{j\frac{3\pi}{2}k}$$

$$= \frac{1}{4} \left[X(0)e^0 + X(1)e^{j\frac{3\pi}{2}} + X(2)e^{j3\pi} + X(3)e^{j\frac{9\pi}{2}} \right]$$

$$= \frac{1}{4} [4+0+0+0]$$

$$= 1$$

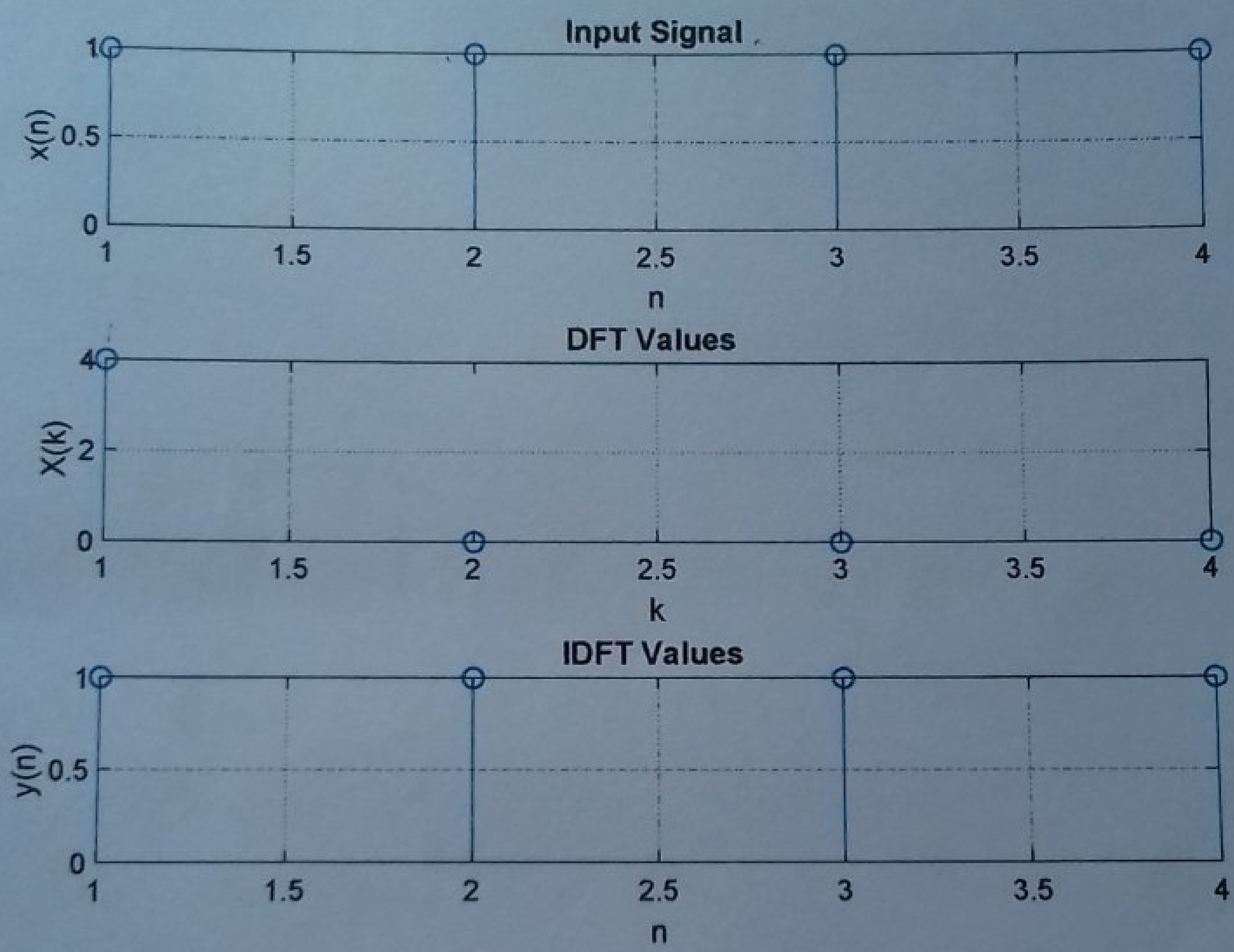
There fore, the IDFT is $x(n)=\{1,1,1,1\}$

Source code:

```
clc;
close all;
clear all;

x = input('Enter the sequence x(n)=');
N = input('Enter n');
disp(N);
subplot(3,1,1);
stem(x);
xlabel('n');
ylabel('x(n)');
title('Input Signal');
grid on;

if N>length(x)
    for i=1:N-length(x)
        x=[x,0];
    end
end
```

```
y = zeros(1,N);
for k=0:N-1
    for n=0:N-1
        y(k+1) = y(k+1) + x(n+1) *
                    exp((-1i * 2*pi *k*n)/N);
    end
end
disp(y);
subplot(3,1,2);
stem(y);
xlabel('k');
ylabel('X(k)');
title('DFT Values');
grid on;

M = length(y);
m = zeros(1,M);
for k=0:M-1
    for n=0:M-1
        m(k+1) = m(k+1)+((1/M)* y(n+1)*
                    exp((1i*2*pi*k*n)/M));
    end
```

```
disp(m);
Subplot(3,1,3);
Stem(m);
xlabel('n');
ylabel('y(n)');
title('IDFT values');
grid on;
```


Input Signal
x(n)
n
DFT Values
X(k)
k
IDFT Values
y(n)
n

Experiment No: 02

Experiment Name: Let,

$x(n) = \{1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1\}$.

Determine and plot the following sequence.

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

Theory: A signal is defined as a function which conveys information. Shifting is an important properties that a signal can perform.

Let us consider $x(n)$ is a discrete time signal.

The shifting version of $x(n)$ is defined by

$y(n) = x(n-n_0)$, here n_0 is the time shift.

if $n_0>0$ then $x(n)$ is shifted to the right.

if $n_0<0$ then $x(n)$ is shifted to the left.

Let us consider the above mentional signal.

$$x(n) = \{1, 2, \underset{\uparrow}{3}, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1\}$$

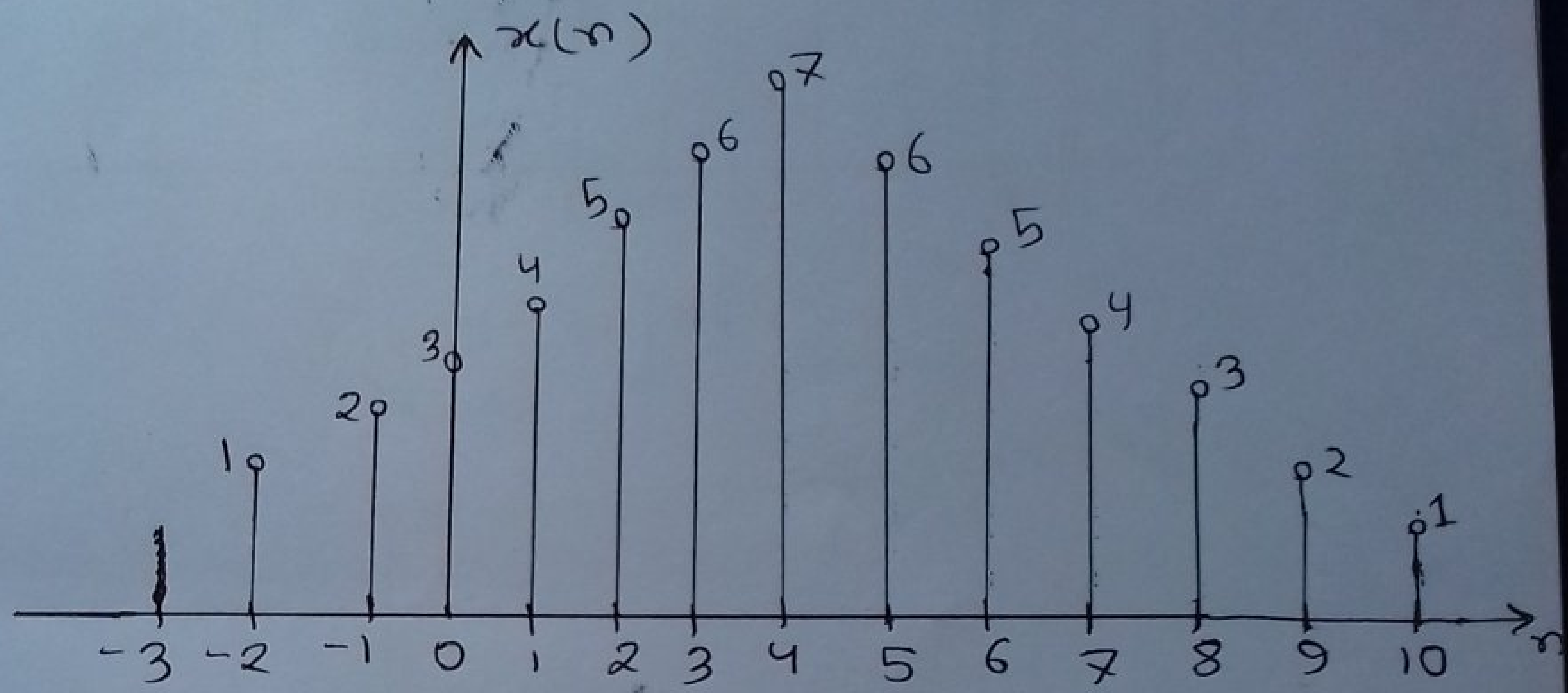
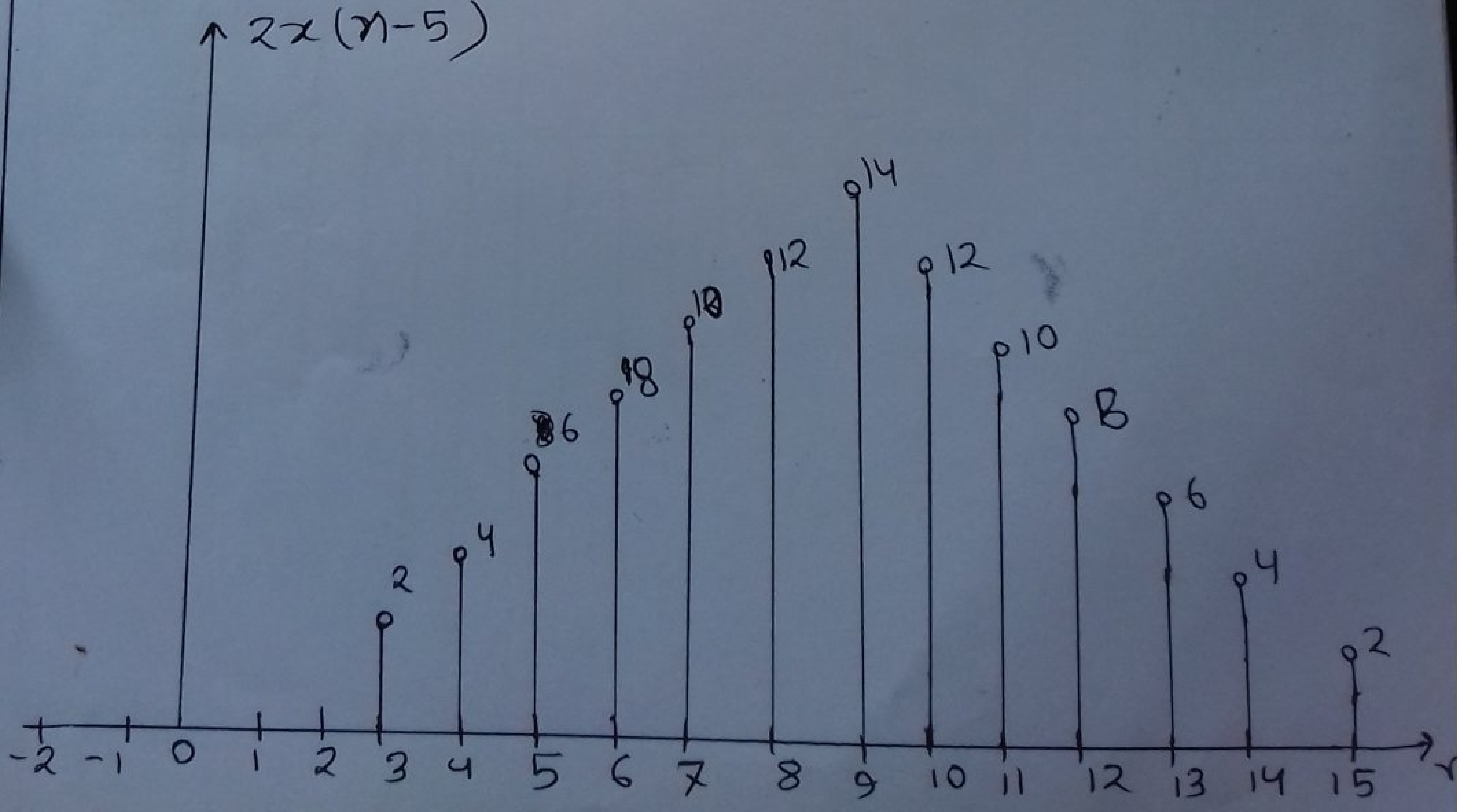


Fig-1 : x(n)



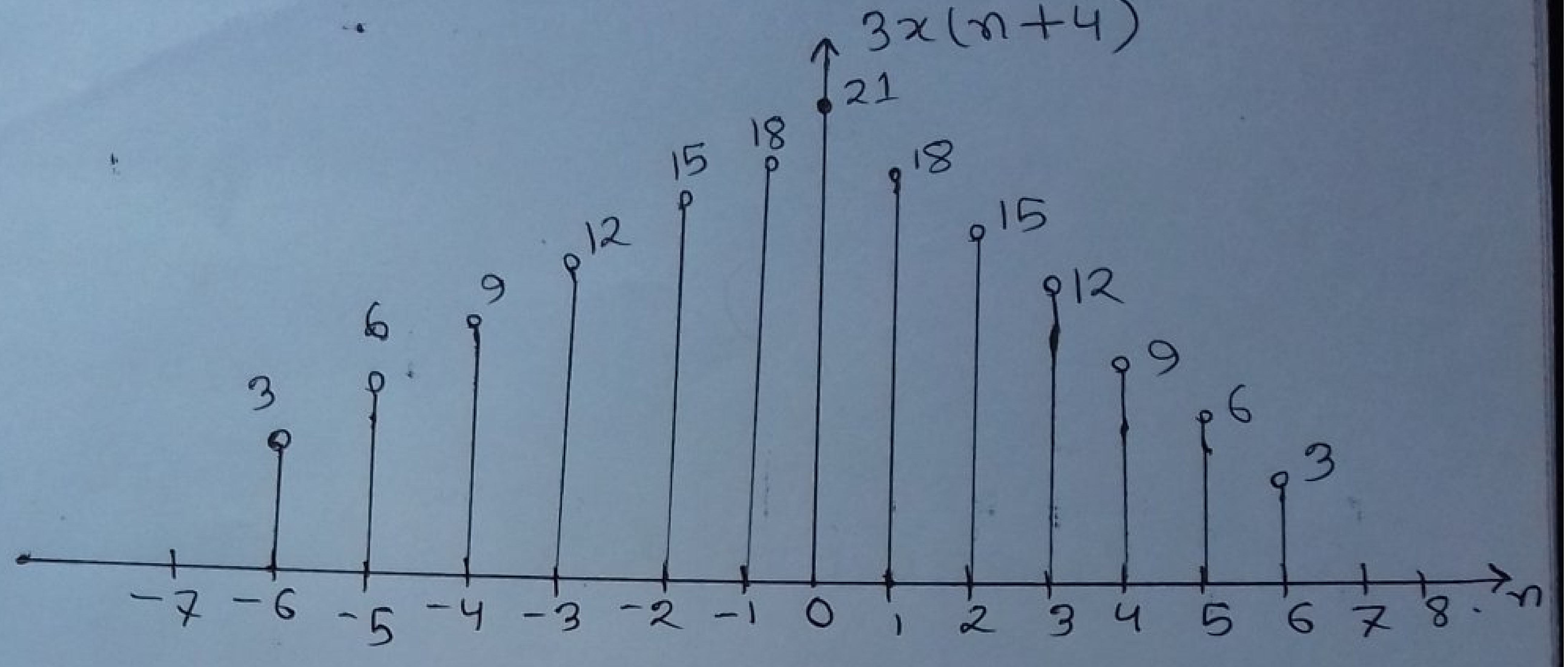


Fig-3: 3x(x+4)

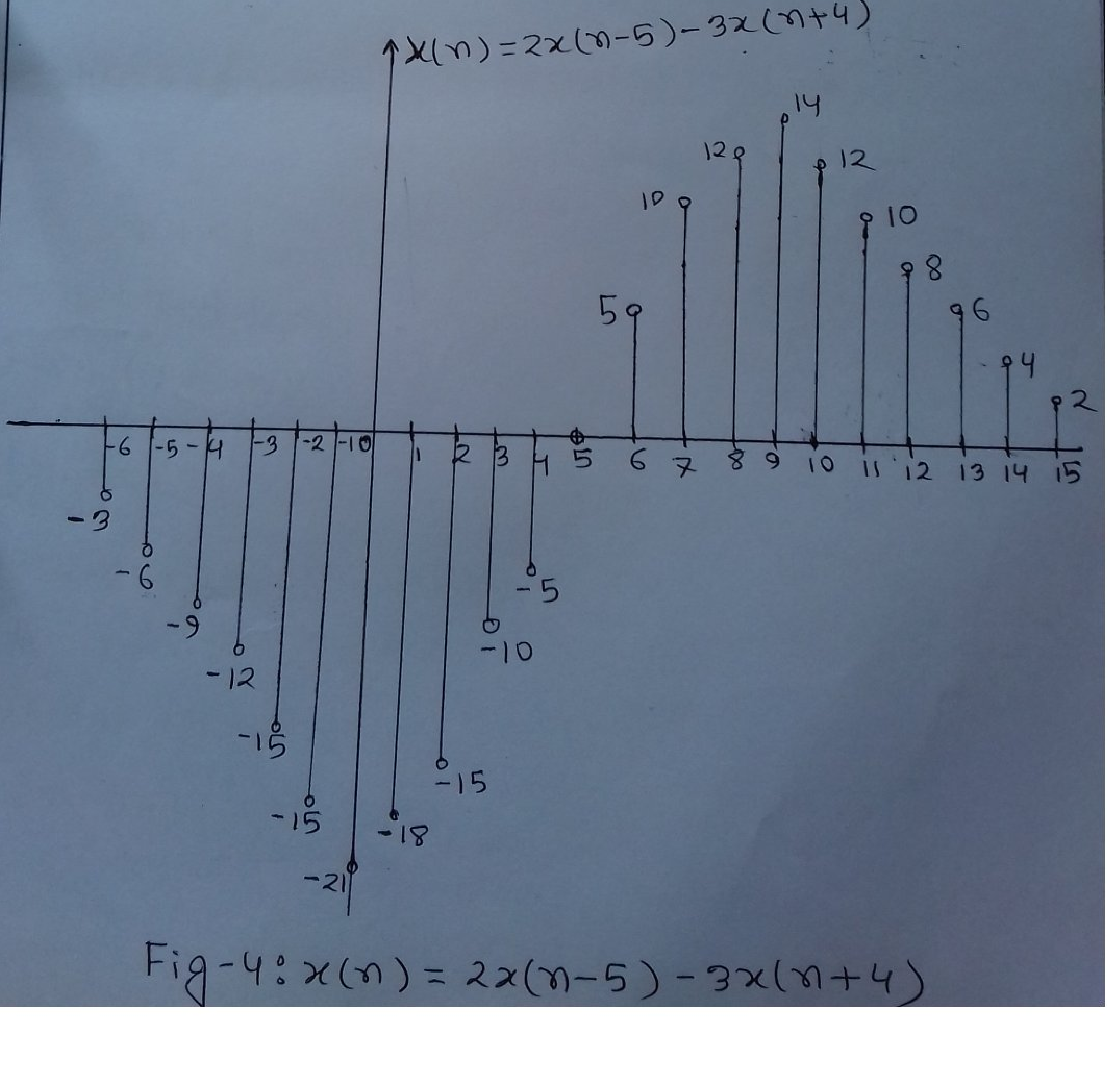


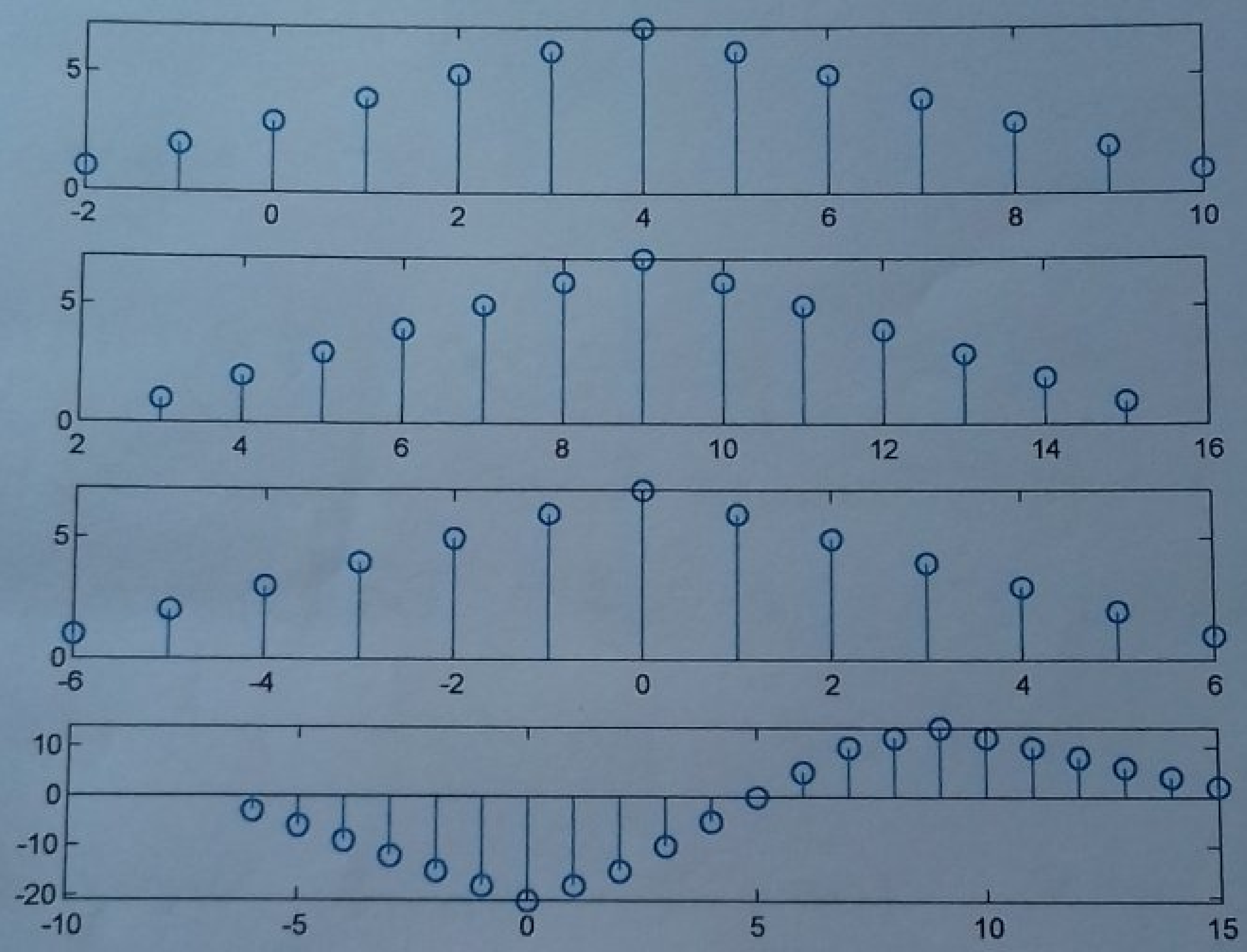
Fig-4: x(n) = 2x(n-5) - 3x(n+4)

Source code:

```
clc;
clear all;
close all;
%figure(1);
x=[1 2 3 4 5 6 7 6 5 4 3 2 1];
n = -2:10;
subplot(4,1,1);
stem(n,x);
%figure(2);
n=3:15;
%a=n+n1;
subplot(4,1,2);
stem(n1,x);
%figure(3);
n2=-6:6;
%b=n-n2;
subplot(4,1,3);
stem(n2,x);
```

```
m= min(min(n1), min(n2)): max(max(n1),
                                   max(n2));
y1 =[];
temp =1;
for i= 1:length(m);
    if (m(i)< min(n1)||m(i)> max(n1));
        y1= [y1,0];
    else
        y1 = [y1.x(temp)];
        temp= temp+1;
    end
end
y2 = [];
temp=1;
for i= 1:length(m);
    if(m(i) <min(n2)|| (m(i)>max(n2));
        y2 = [y2,0];
    else
        y2=[y2 x(temp)];
        temp = temp+1;
    end
```

```
y = (2.* y1)-(3.* y2);
Subplot (4,1,4);
Stem(m, y);
```


5
0
-2 0 2 4 6 8 10
5
0
2 4 6 8 10 12 14 16
5
0
-6 -4 -2 0 2 4 6
10
0
-10
-20
-10 -5 0 5 10 15

Experiment No: 03

Experiment Name: Write a matlab program to perform following operation.

(i) sampling (ii) Quantization (iii) Coding

Theory:

Sampling: Sampling is a procedure in which a continuous time signal is converted to a discrete time signal, by taking samples of the continuous time signal at discrete time instants.

Quantization: Quantization is a process of maping a large set of input values to a smaller set. Rounding is a typical sample of quantization process.

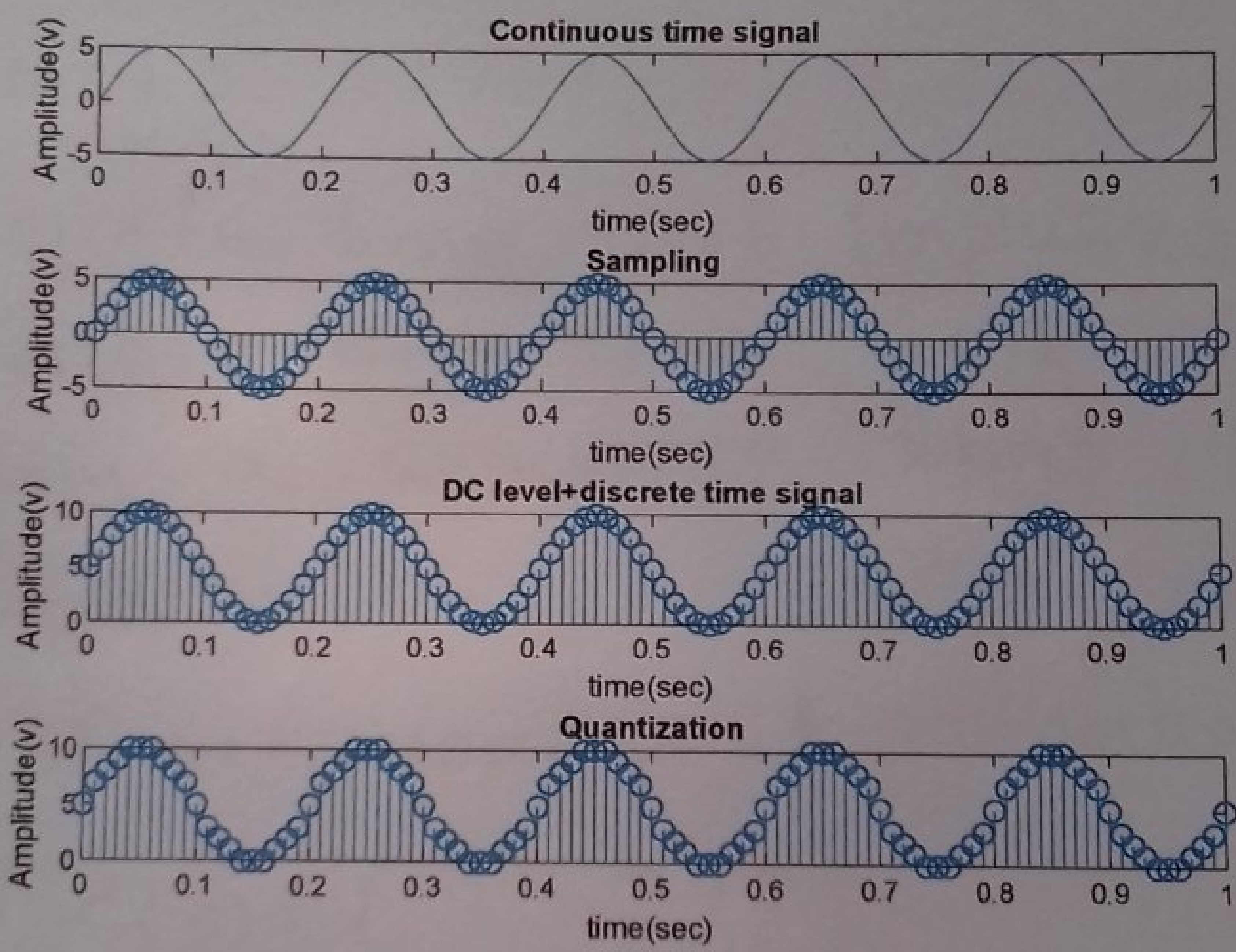
The difference between an input value and its quantized value is reffered to as quantization error.

Coding: A system of signals used to represent letters or numbers in transmitting manages. This system is named as coding.

Source code:

```
clc;
clear all;
close all;
A =5;
f = 5;
t = 0:0.001:1;
x = A*sin(2*pi*f*t);
Subplot(4,1,1);
plot(t,x);
title('Continuous time signal');
xlabel('Time');
ylabel('Amplitude');
```

```
%% After sampling discrete time signal
Subplot(4,1,2);
Stem(t,x);
title('Sampling');
xlabel('Time');
ylabel('Amplitude');

%DC level+discrete time signal
x1 = A+x;
Subplot(t,x1); (4,1,3); stem(t,x1);
title('DC level+ discrete time signal');
xlabel('Time');
ylabel('Amplitude');

%% Quantized
x2=round(x1);
Subplot(4,1,4);
stem(t,x2);
title('Quantization');
xlabel('Time');
ylablel('Amplitude');

%% Coding
x3 = dec2bin(x2);
disp(x3);
```


Continuous time signal
Amplitude(v)
time(sec)
Sampling
Amplitude(v)
time(sec)
DC level+discrete time signal
Amplitude(v)
time(sec)
Quantization
Amplitude(v)
time(sec)

Experiment No: 04

Experiment Name: Determine and plot the following sequences,

$$x(n) = 2\delta(n+2) - \delta(n-4) \quad , \quad -5 \leq n \leq 5$$

Theory:

Discrete time unit impluse: In Discrete time, the unit impluse is simply a sequence that is zero except $n=0$.

In other word, it is defined as,

$$\delta(n) = \begin{cases} 0 \; ; & n \neq 0 \\ 1 \; ; & n = 0 \end{cases}$$

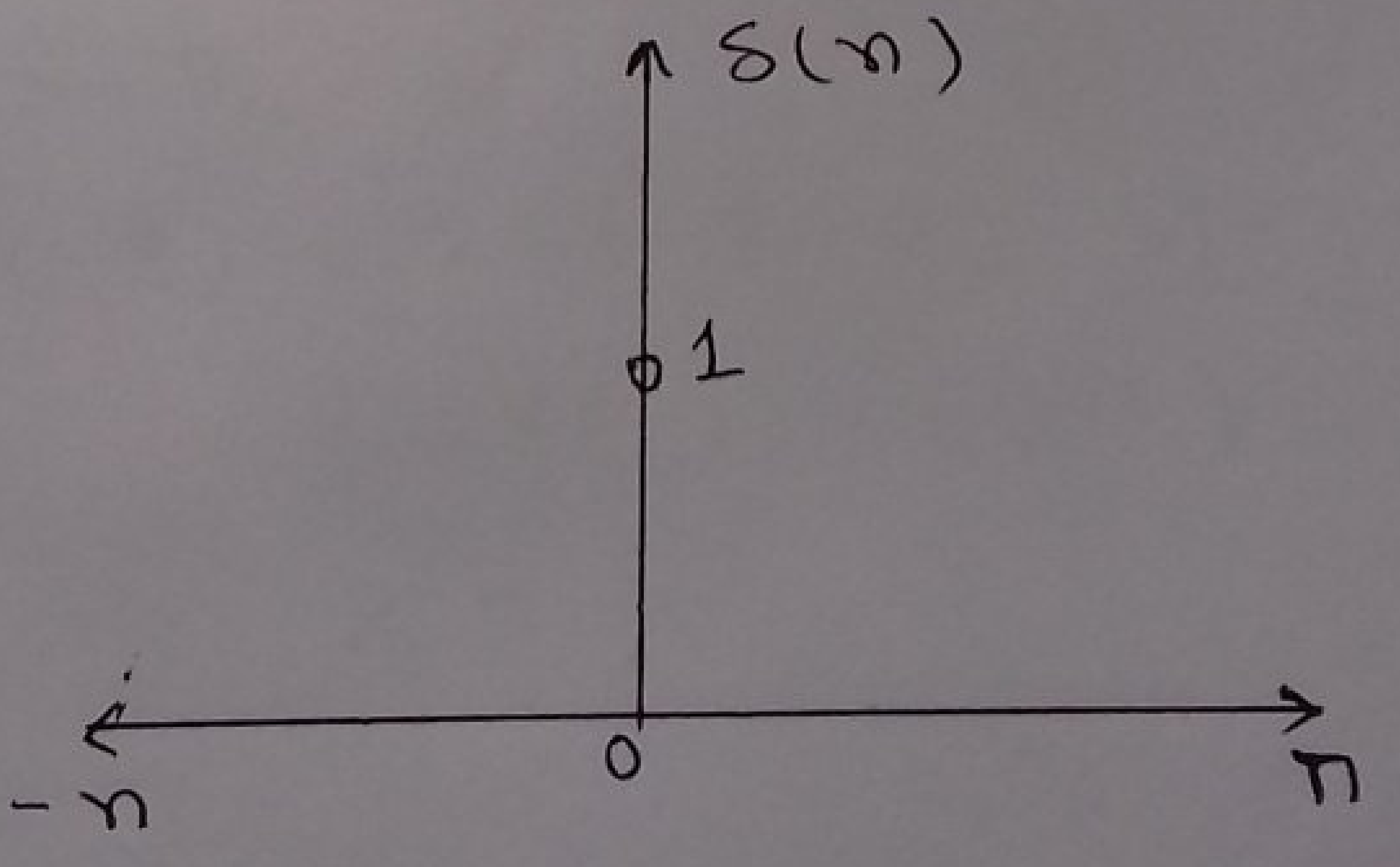


Fig-a: Graphical representation of the unit sample signal

Let us consider the sequence as sample,

$$x(n) = 2\delta(n+2) - \delta(n-4) \quad , -5 \leq n \leq 5$$

when,

$n = -5$,

$$x(-5) = 2\delta(-5+2) - \delta(-5-4)$$
$$= 2\delta(-3) - \delta(-9)$$
$$= 0$$

when, $n = -4$,

$$x(-4) = 2\delta(-4+2) - \delta(-4-4)$$
$$= 2\delta(-2) - \delta(-8)$$
$$= 0$$

when, $n = -3$,

$$x\cancel{n}(-3) = 2\delta(-3+2) - \delta(-3-4)$$
$$= 2\delta(-1) - \delta(-7)$$
$$= 0$$

when $n = -2$,

$$x(-2) = 2\delta(-2+2) - \delta(-2-4)$$
$$= 2\delta(0) - \delta(-6)$$
$$= 2 \times 1 - 0$$
$$= 2$$

when, $n=-1$,

$$x(-1) = 2\delta(-1+2) - \delta(-1-4)$$

$$= 2\delta(1) - \delta(-5)$$

$$= 0$$

when, $n=0$,

$$x(0) = 2\delta(0+2) - \delta(0-4)$$

$$= 2\delta(2) - \delta(-4)$$

$$= 0$$

when, $n=1$,

$$x(1) = 2\delta(1+2) - \delta(1-4)$$

$$= 2\delta(3) - \delta(-3)$$

$$= 0$$

when, $n=2$,

$$x(2) = 2\delta(2+2) - \delta(+2-4)$$

$$= 2\delta(4) - \delta(-2)$$

$$= 0$$

when, $n=3$, $x(3) = 2\delta(3+2) - \delta(3-4)$

$$= 2\delta(5) - \delta(-1)$$

$$= 0$$

when, $n=-1$,

$$x(-1) = 2\delta(-1+2) - \delta(-1-4)$$

$$= 2\delta(1) - \delta(-5)$$

$$= 0$$

when, $n=0$,

$$x(0) = 2\delta(0+2) - \delta(0-4)$$

$$= 2\delta(2) - \delta(-4)$$

$$= 0$$

when, $n=1$,

$$x(1) = 2\delta(1+2) - \delta(1-4)$$

$$= 2\delta(3) - \delta(-3)$$

$$= 0$$

when, $n=2$,

$$x(2) = 2\delta(2+2) - \delta(+2-4)$$

$$= 2\delta(4) - \delta(-2)$$

$$= 0$$

when, $n=3$, $x(3) = 2\delta(3+2) - \delta(3-4)$

$$= 2\delta(5) - \delta(-1)$$

$$= 0$$

when, $n = 4$,

$$x(4) = 2\delta(4+2) - \delta(4-4)$$
$$= 2\delta(6) - \delta(0)$$
$$= 0 - 1$$
$$= -1$$

when, $n = 5$,

$$x(5) = 2\delta(5+2) - \delta(5-4)$$
$$= 2\delta(7) - \delta(1)$$
$$= 0 - 0$$
$$= 0$$

Now the graphical representation of the output of this given sequence will be,

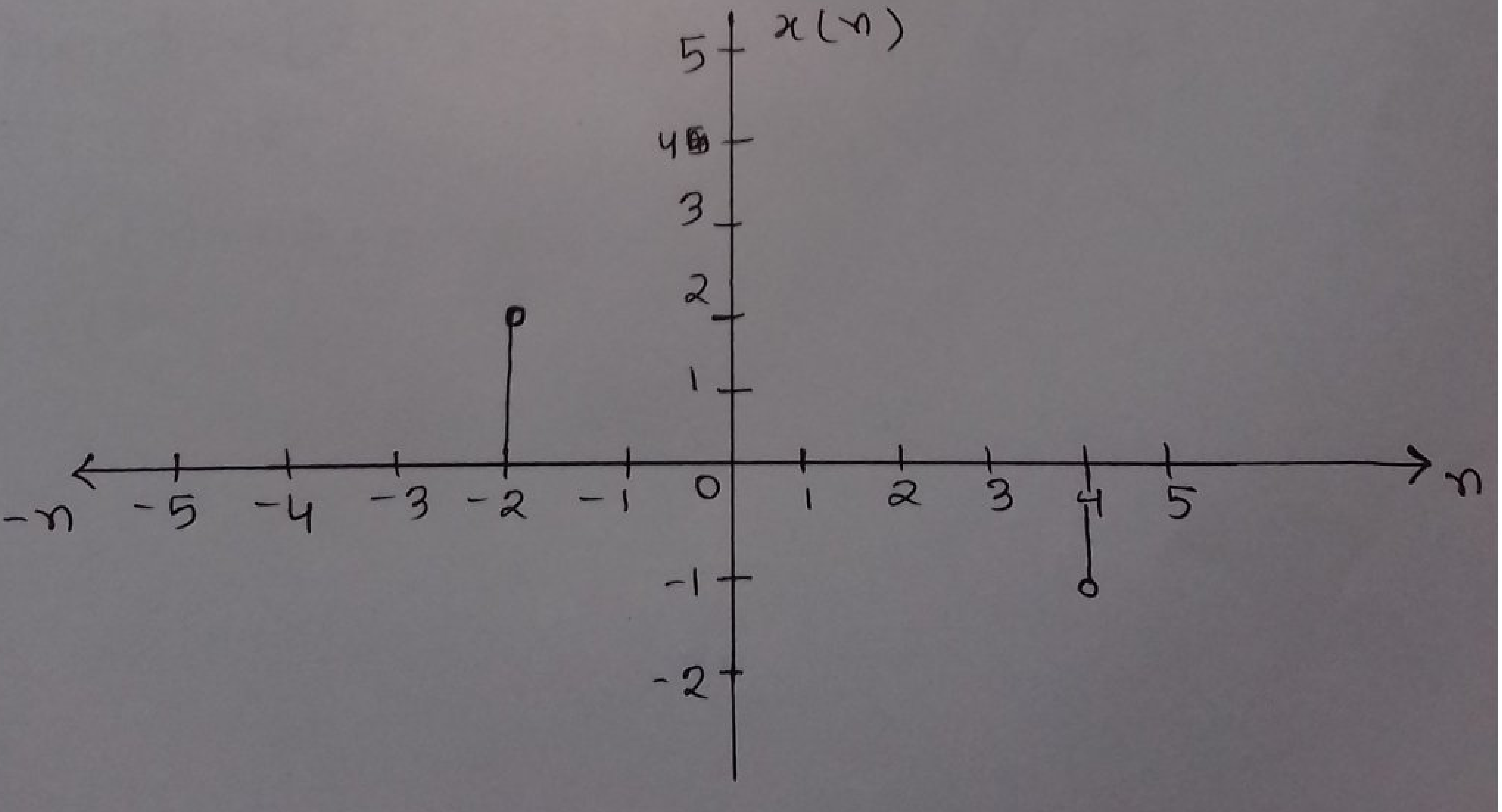
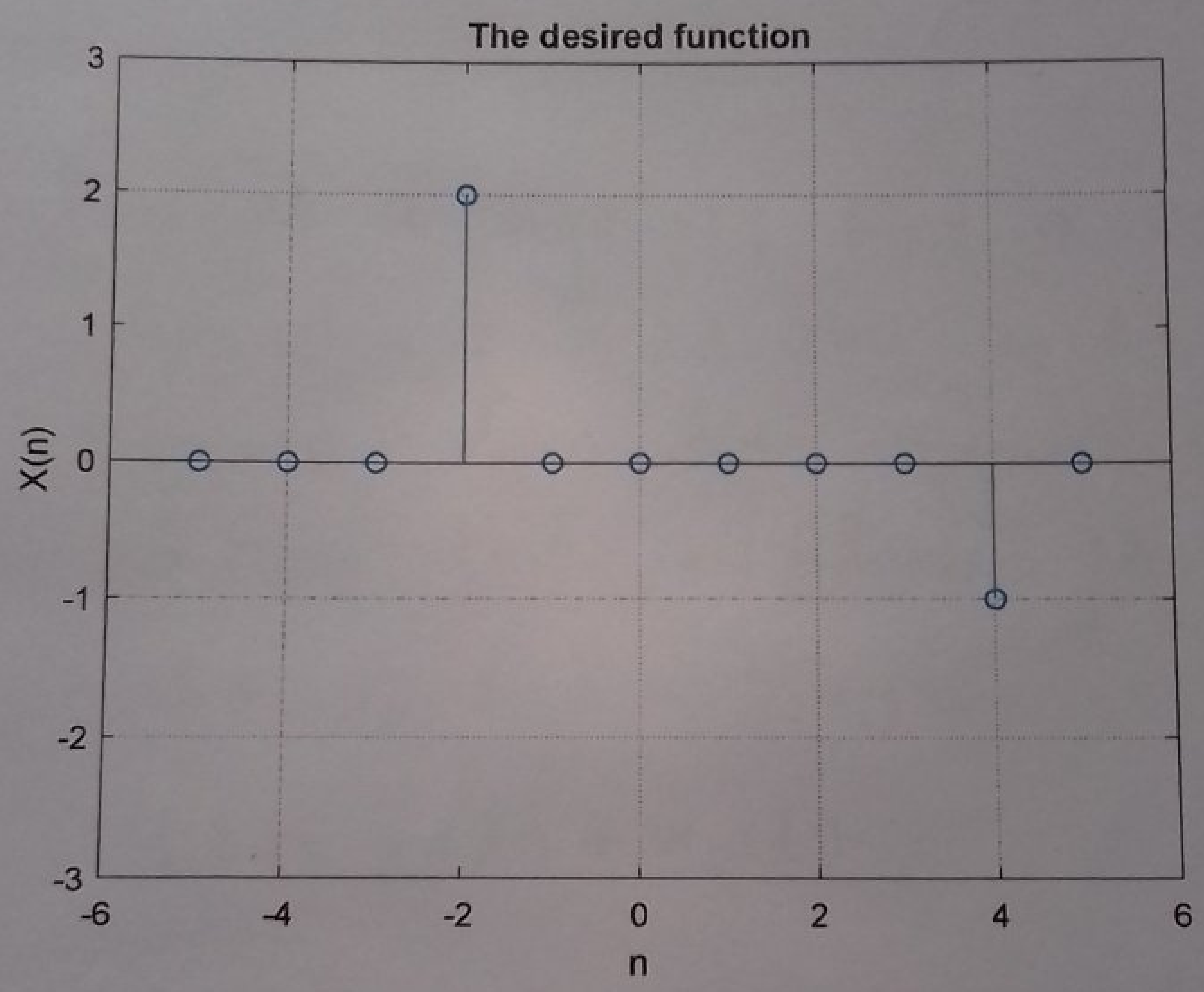


Fig-6: Discrete time impluse sequence

Source code :

```
clc;
clear all;
close all;
n=5:5;
x=2*deltaF(-2,-5,5)-deltaF(4,-5,5);
stem(n,x);
xlabel('n');
ylabel('x(n)');
title('The desired function');
axis([-6 6 -3 3]);
grid on;

function[x,n]=deltaF(n0,n1,n2)
n=n1:n2;
x=(n-n0)==0;
end
```


The desired function
X(n)
n
3
2
1
0
-1
-2
-3
-6
-4
-2
0
2
4
6

Experiment No : 05

Experiment Name : To plot the following signal operation using user defined function i) Addition ii) folding.

Theory :

Addition of a signal : For a continuous time signal, if $x_1(t)$ and $x_2(t)$ are two signals then the signal $x(t)$ obtained by the addition of $x_1(t)$ and $x_2(t)$ is defined by,

$$y(t) = x_1(t) + x_2(t)$$

And if $x_1(n)$ and $x_2(n)$ are two discrete signals then the addition of this two signals is defined by

$$y(n) = x_1(n) + x_2(n)$$

Example of addition of two continuous time signal.

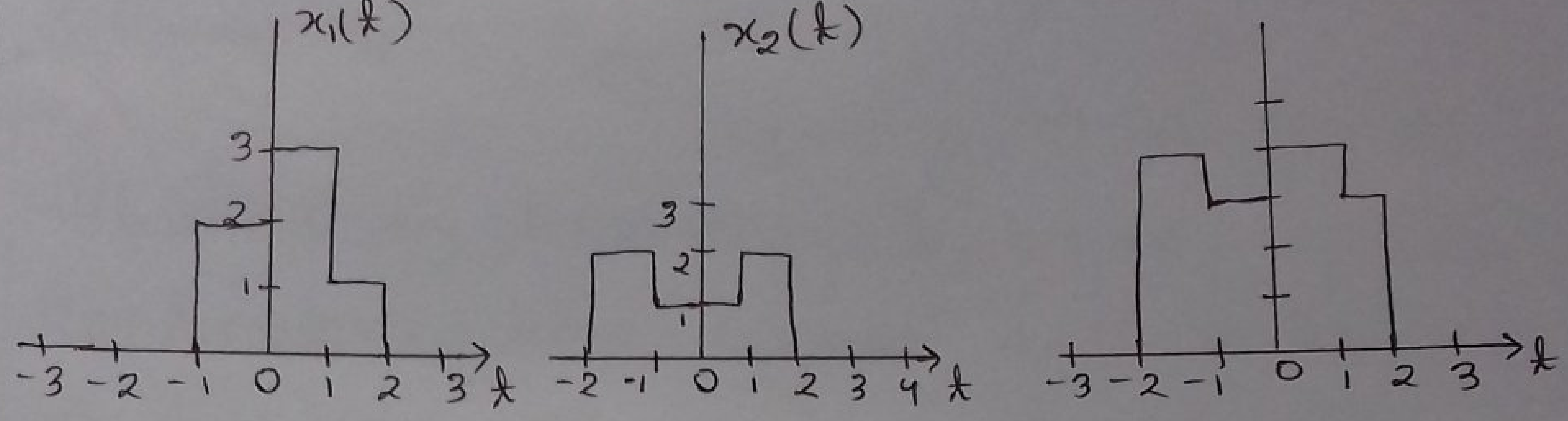


Fig-a: Addition of C-T Time Signal

Example of addition of two discrete time signal.

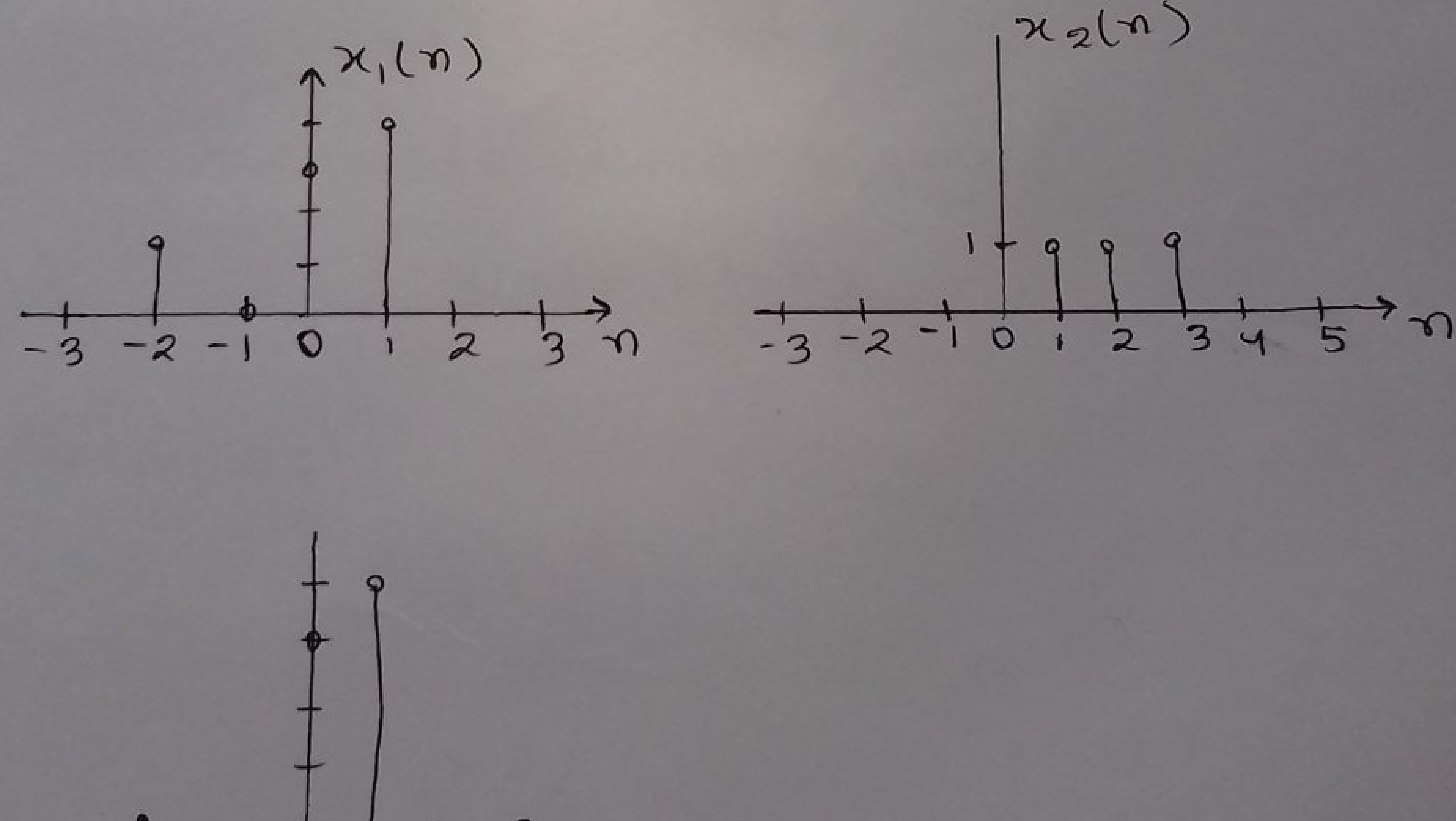


Fig-b: Addition of two discrete time signal.

Folding of a Signal:

Folding of a signal obtained by replacing t by $-t$ is continuous time signal and n by $-n$ is discrete time signal. The period will be unchanged.

Folding of a continuous-time signal will be

$$y(t) = x(-t)$$

For discrete time signal,

$$y(n) = x(-n)$$

Example:

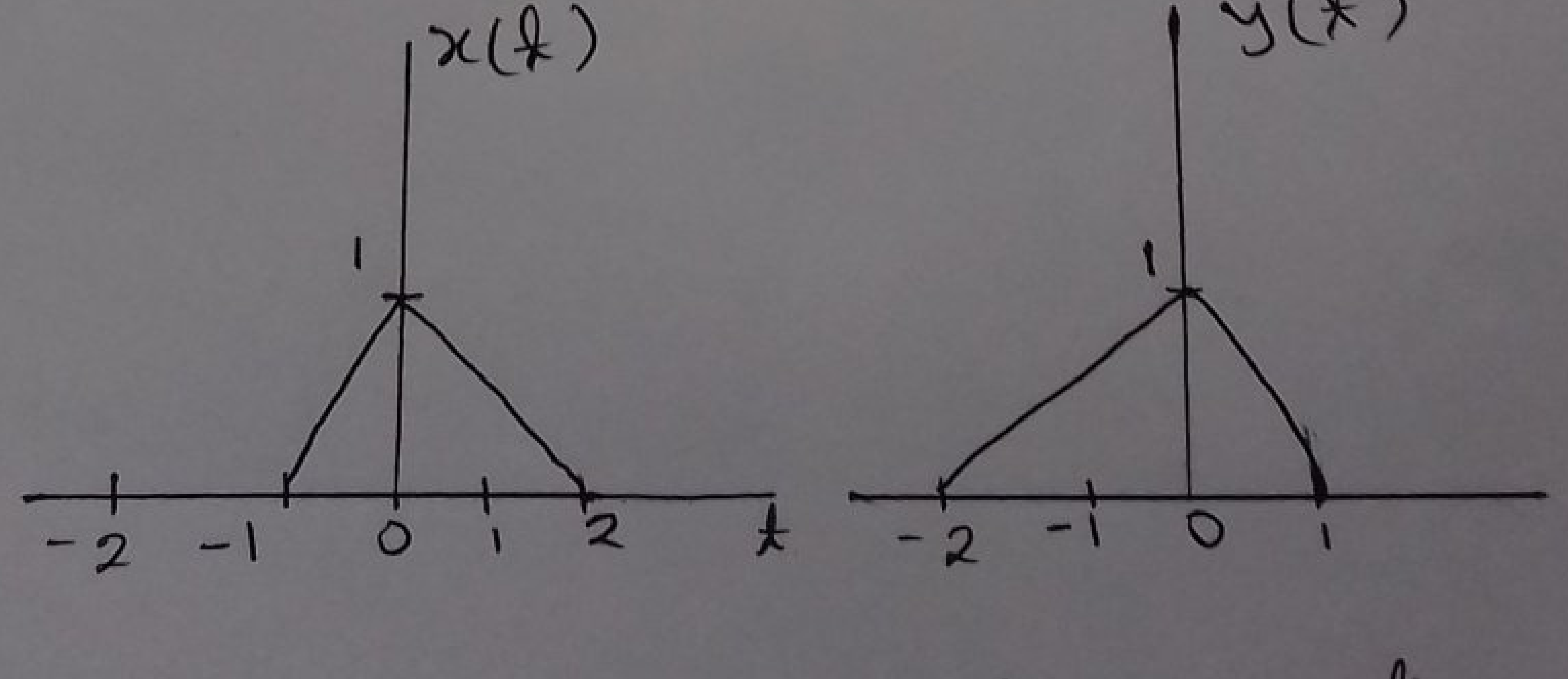


Fig-a: Folding of two continuous signal.

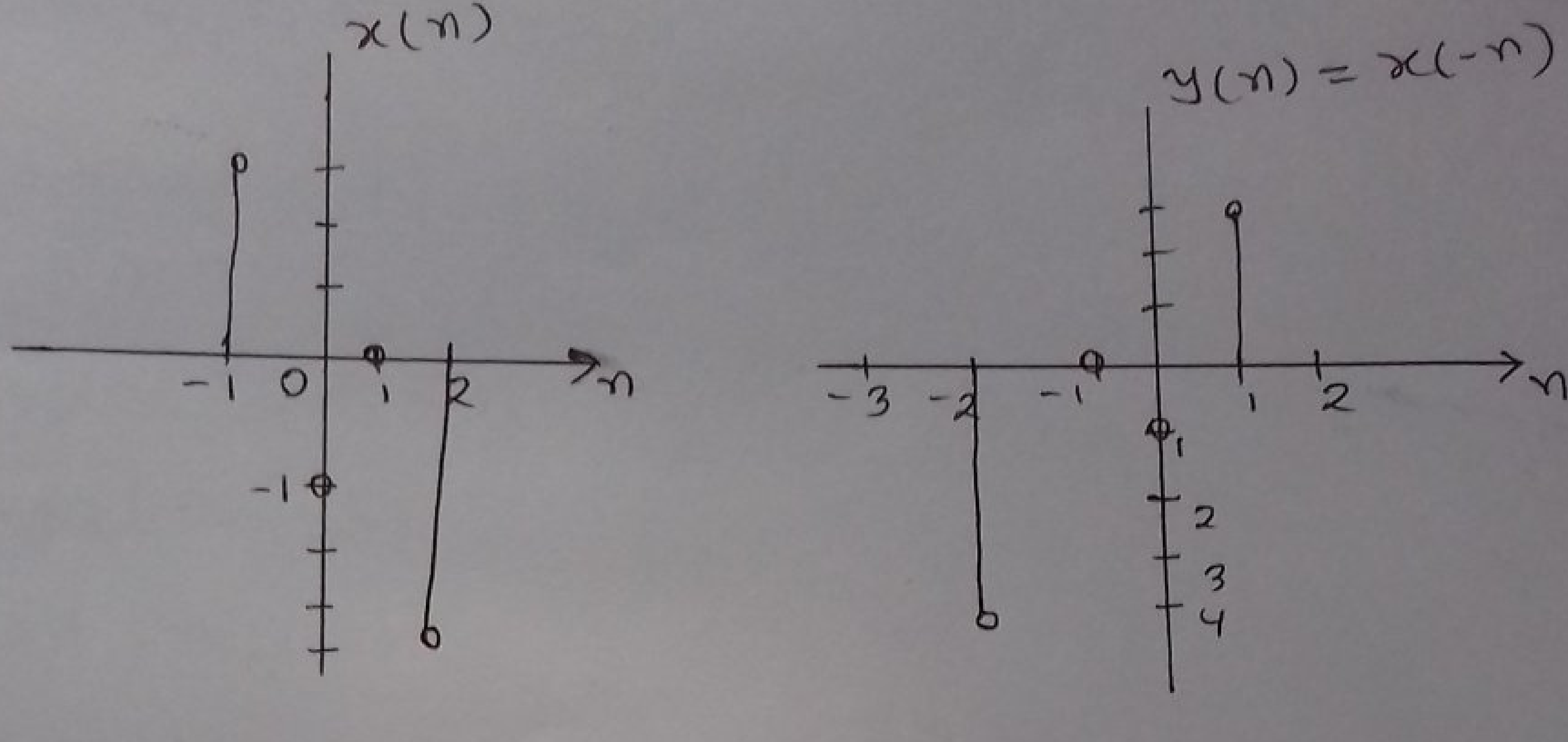


Fig-b: Folding of discrete time signal

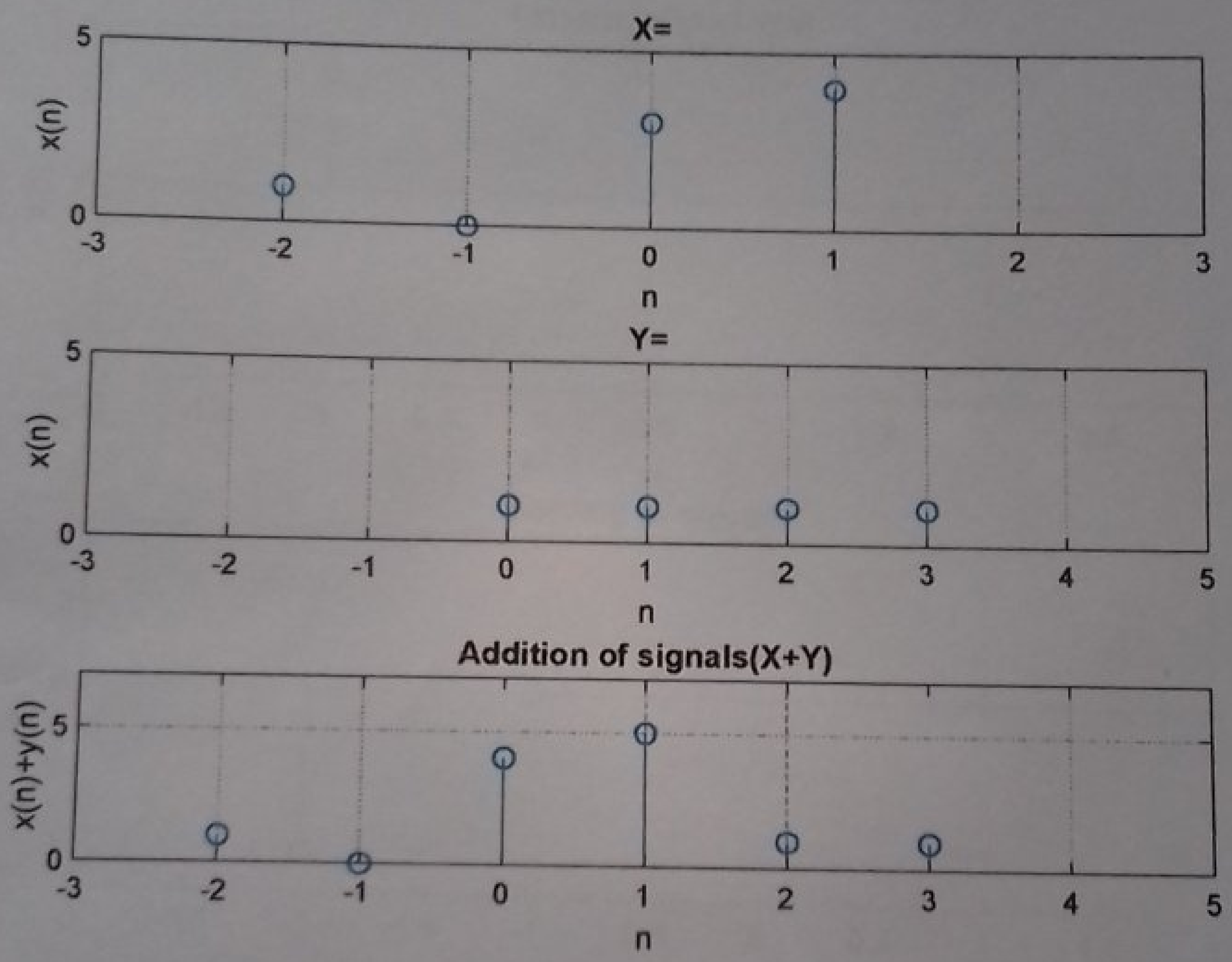
Source Code:

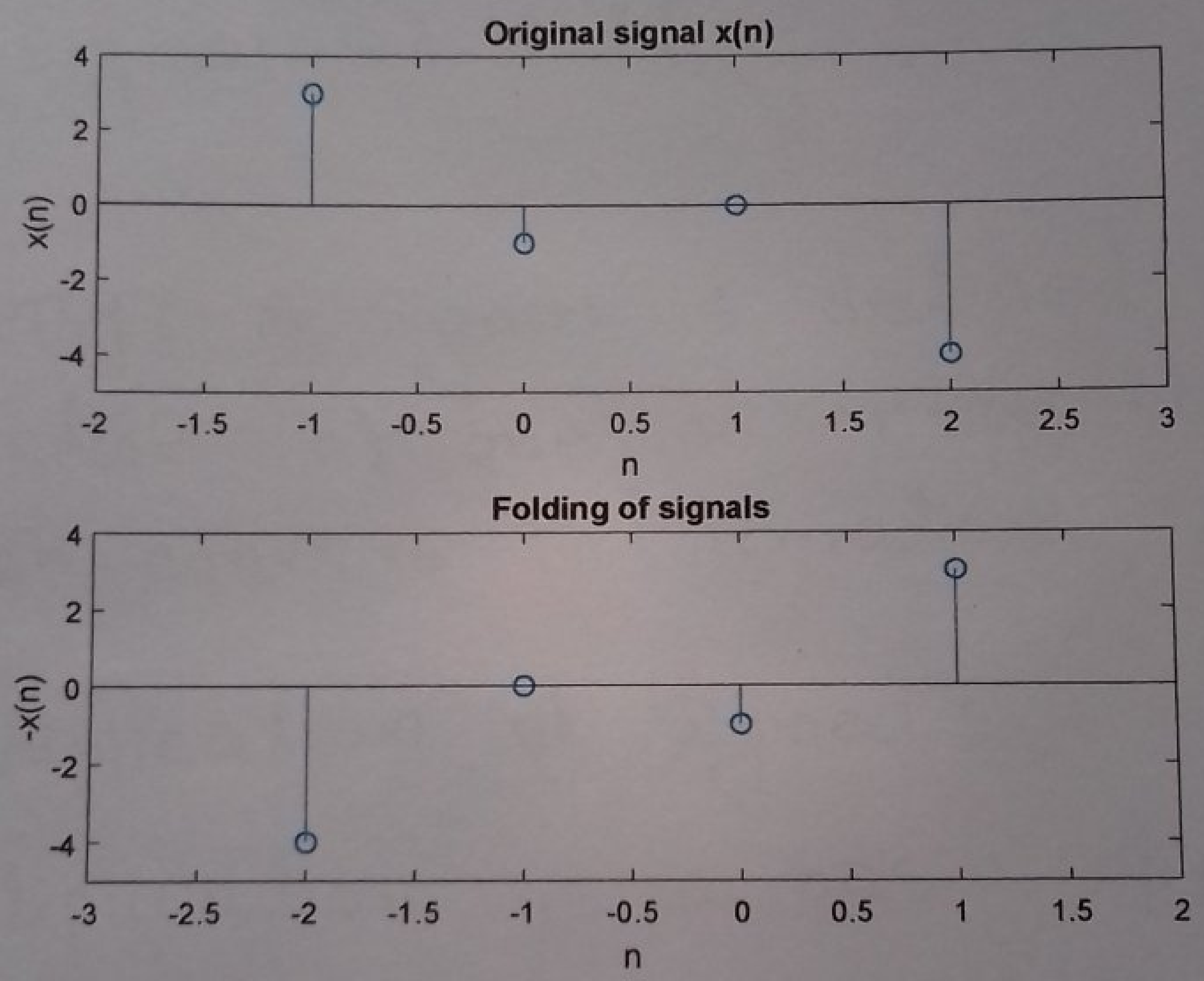
```
clc;
close all;
clear all;
figure(1);
x = [1,0,3,4];
n1 = -2:1;
Subplot(3,1,1);
stem(n1,x);
grid on;
title('x = ');
xlabel('n');
ylabel('x(n)');
```

```
axis([-3,3,0,5]);

y = [1,1,1,1];
n2 = 0:3;
Subplot (3,1,2);
Stem (n2,y);
grid on;
title ('Y=');
xlabel ('n');
ylabel('x(n)');
axis([-3,5 0 5]);
m = min(min(n1), min(z)) : max(max(n1),
                              max(n2));
y1 = [ ];
temp = 1;
for i = 1:length(m)
    if (m(i)<min(n1)|| m(i)>max(n1))
        y1 = [y1,0];
    else
        y1 = [y1 x(temp)];
        temp = temp+1;
    end
end
```

```
y2=[];
temp=1;
for i= 1:length(m)
    if(m(i)<min(n2)||m(i)>max(n2))
        y2 = [y2.0];
    else
        y2 = [y2 y(temp)];
        temp = temp+1;
    end
end

add = y1+y2;
Subplot (3,1,3)
Stem(m, add );
grid on;
title ("Addition of Signals (x+y)');
xlabel ('n');
ylabel ('x(n)+y(n)');
axis([-3,5,0,7]);
```

```
figure(2);
x = [3 -1 0 -4];
n = -1:2;
Subplot (2,1,1);
Stem (n,x);
title ('Original Signal x(n)');
xlabel('n');
ylabel ('x(n)');
axis ([-2 3 -5 4]);
c = fliplr(x);
y = fliplr(-n);
Subplot (2,1,2);
Stem(y,c);
title ('Folding of Signals');
xlabel ('n');
ylabel ('-x(n)');
axis[[-3 2 -5 4]);
```


X=
x(n)
n
Y=
x(n)
n
Addition of signals(X+Y)
x(n)+y(n)
n



Original signal x(n)
x(n)
n
Folding of signals
-x(n)
n

Experiment NO: 06

Experiment Name: Plot the following operation using user defined function.

i) Signal multiplication

ii) Signal ~~sf~~ shifting

Theory: A signal is defined as a function of one or more variables which conveys information.

Multiplication of Signal:

Multiplication is a basic operation on signals.

Let us consider $x_1(n)$ and $x_2(n)$ two discrete signals. Then the resultant signal $y(n)$ obtained by multiplication of $x_1(n)$ and $x_2(n)$ is defined by

$$y(n) = x_1(n) \cdot x_2(n)$$

Let us consider two discrete signals as example and we have to multiply this two signals.

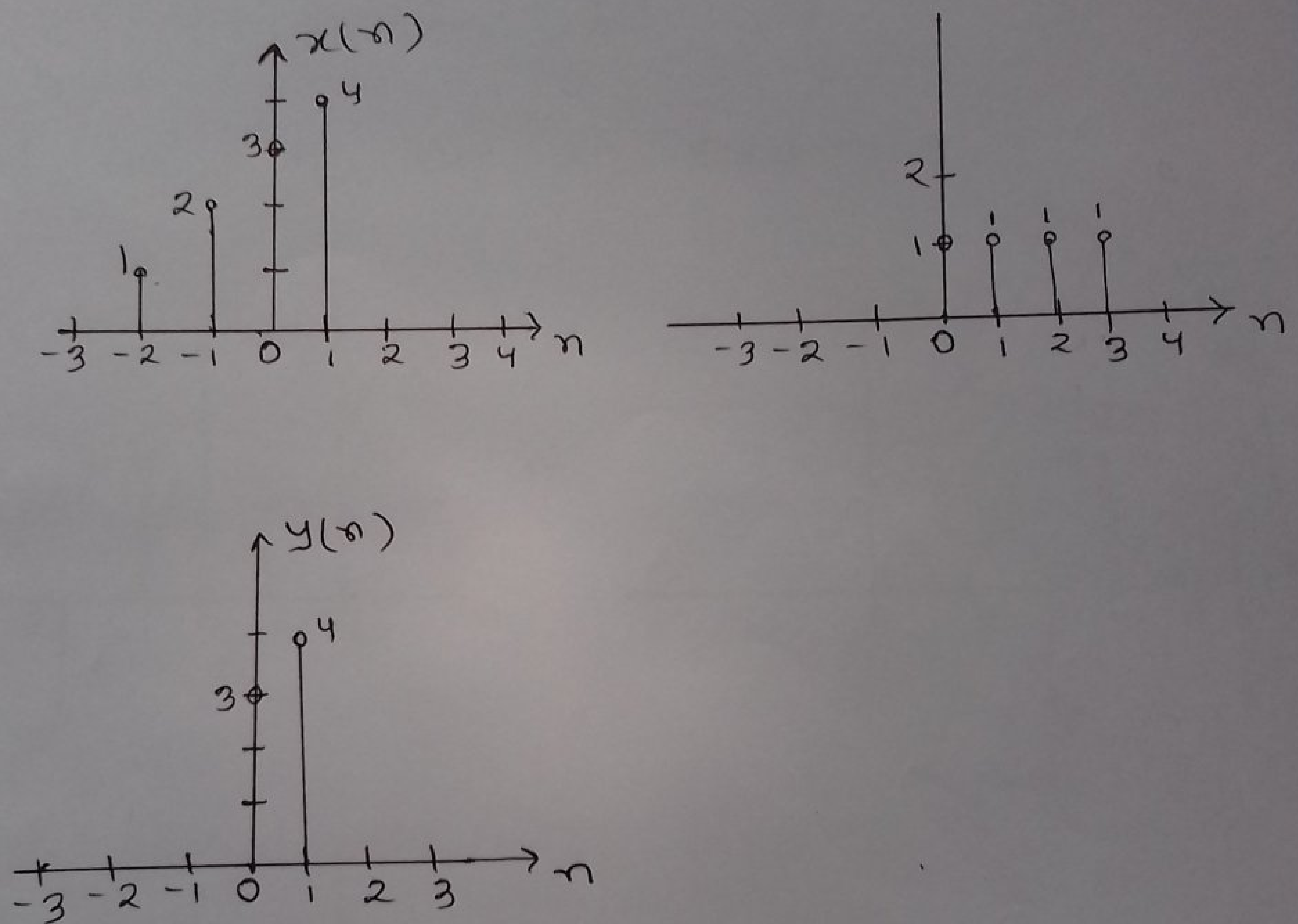


Fig-a: Signal multiplication

Shifting of Signal:

Let us consider $x(n)$ is a discrete time signal. Then the time shifting version of $x(n)$ is defined by,

$y(n) = x(n - n_0)$ here, n_0 is the time shift

- If $n_0 > 0$, then waveform of $x(n)$ is shifted to right.
- If $n_0 < 0$, then the waveform of $x(n)$ is shifted to left.

Let us consider an example,

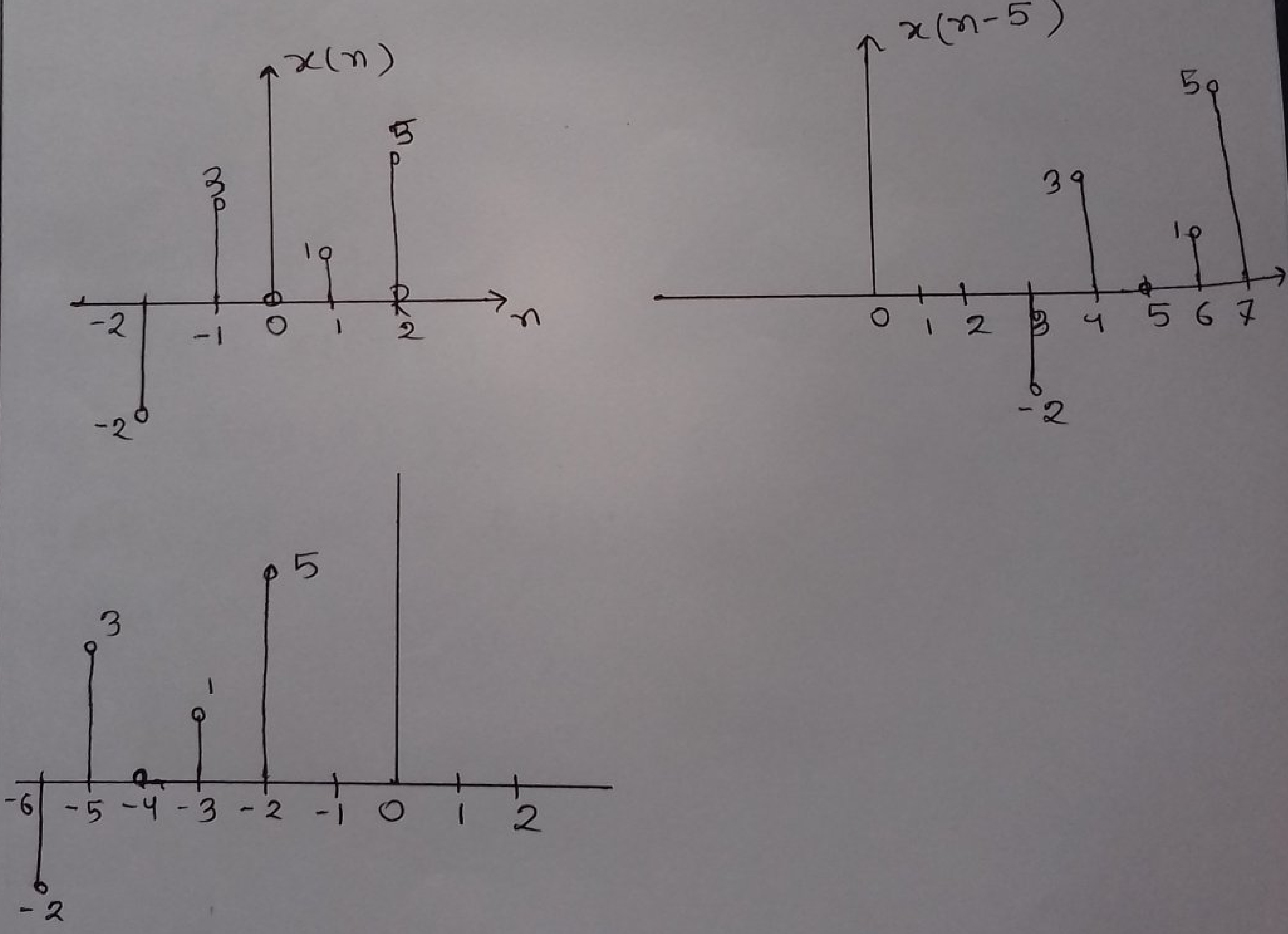


Fig-6: Shifting of discrete signals

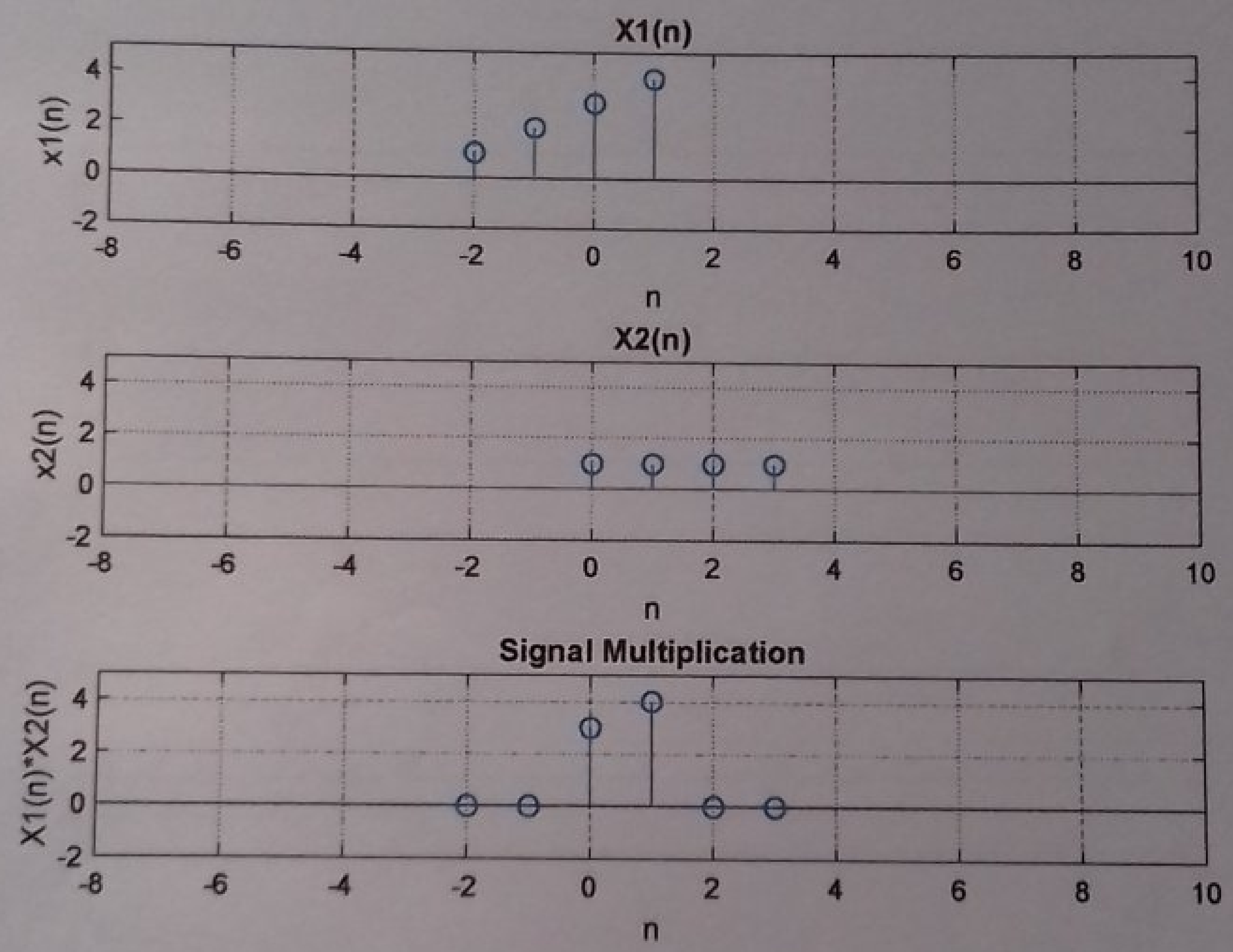
Source code:

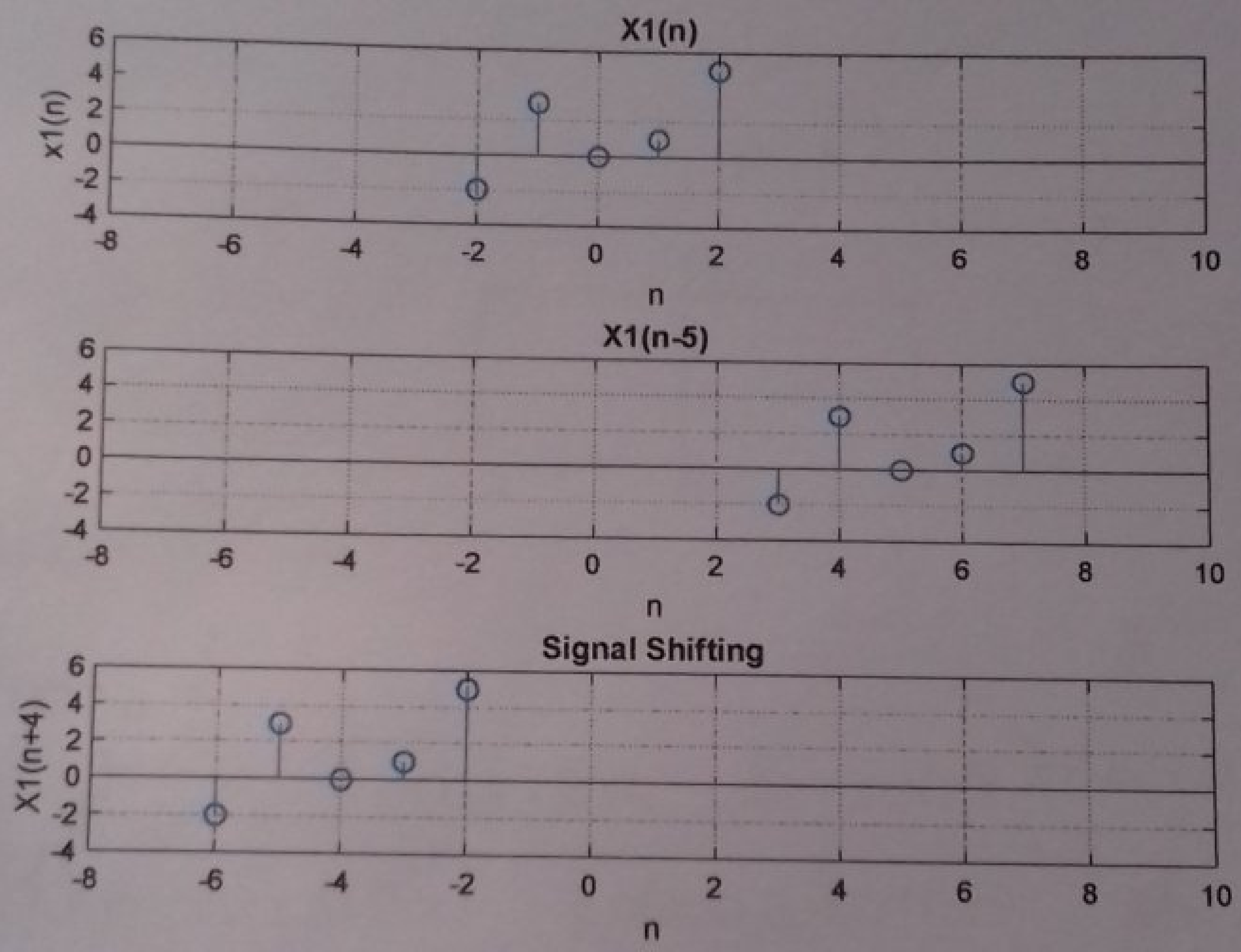
```
clc;
close all;
clear all;

figure(1);
x = [1.2 3 4];
n1 = -2:1;
subplot(3,1,1);
stem(n1,x);
xlabel('n');
ylabel('x1(n)');
title('x1(n)');
axis([-8 10 -2 5]);
grid on;

y = [1 1 1 1];
n2 = 0:3;
subplot(3,1,2);
stem(n2,y);
xlabel('n');
ylabel('x2(n)');
```

```
axis([-8 10 -2 5]);
grid on;

m = min(min(n1), min(n2)) : max((max(n1),
                              max(n2));

y1 = [];
temp = 1;
for i = 1:Length(m)
    if (m(i)< min(n1) || m(i)> max(n1))
        y1 = [y1, 0];
    else
        y1 = [y1 x(temp)];
        temp = temp + 1;
    end
end

y2 = [];
temp = 1;
for i = 1:Length(m)
    if ( m(i)<min(n2) || m(i) >max(n2))
        y2 = [y2 0];
    else
```

```
            y2 = [y2  y(temp)];
            temp = temp+1;
      end
end
mul = y1 * y2;
Subplot (3,1,3);
stem(m, mul);
xlabel('n');
ylabel('x1(n)* x2(n)');
title('signal Multiplication');
axis([-8  10 -2  5]);
grid on

figure(2);
n = -2:2;
x = [-2  3  0  1 5];
Subplot (3,1,1);
stem(n, x);
xlabel ('n');
ylabel ('x1(n)');
title ('x1(n)');
```

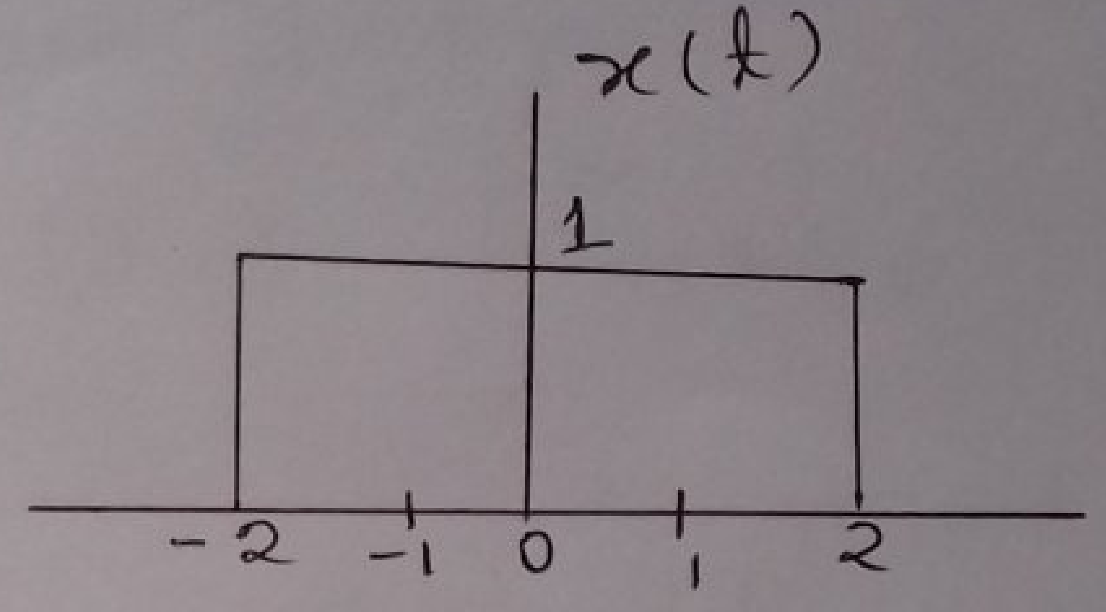
```
axis([-8 10 -4 6]);
grid on;

n1=5;
a=n+n1;
Subplot(3,1,2);
stem(a,x);
xlabel('n');
ylabel('x2(n)');
title('x1(n-5)');
axis([-8 10 -4 6]);
grid on

n2=4;
b=n-n2;
Subplot(3,1,3);
stem(b,x);
xlabel('n');
ylabel('x1(n+4)');
title('Signal shifting');
axis([-8 10 -4 6]);
grid on;
```


X1(n)
x1(n)
n
X2(n)
x2(n)
n
Signal Multiplication
X1(n)*X2(n)
n


X1(n)
x1(n)
n
X1(n-5)
n
Signal Shifting
X1(n+4)
n

Experiment No: 07

Experiment Name: Using MATLAB to plot the Fourier Transform of a Time Function the aperiodic pulse shown below

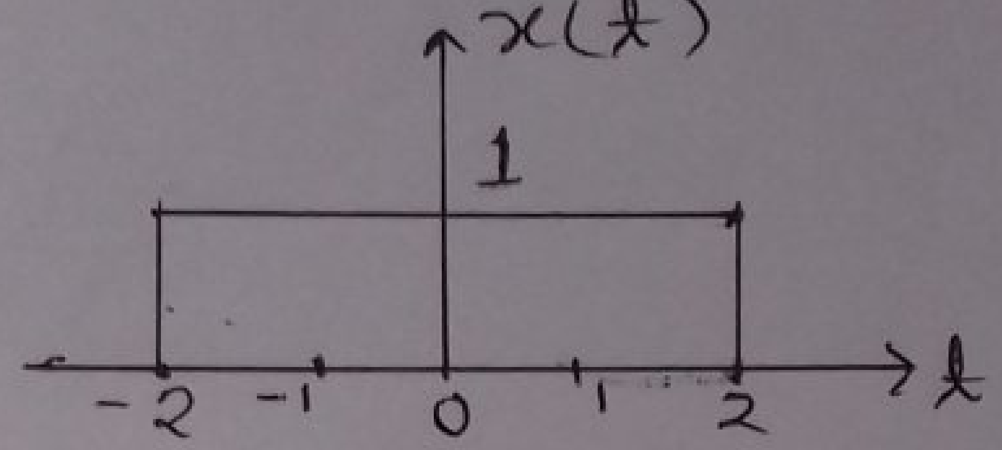


Theory: A Fourier transform is a mathematical transform that decomposes function depending on time into function depending on frequency.

From the definition of continuous time Fourier transform, we know that,

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)\, e^{-j\omega t}\, dt$$

Let us consider this above given example



By definition of FT,

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)\, e^{-j\omega t}\, dt$$

$$= \int_{-2}^{2} 1\, e^{-j\omega t}\, dt$$

$$= \left[\frac{e^{-j\omega t}}{-j\omega} \right]_{-2}^{2}$$

$$= \frac{e^{-2j\omega} - e^{-j\omega(-2)}}{-j\omega}$$

$$= \frac{e^{-2j\omega} + e^{-j2\omega}}{-j\omega}$$

$$= \frac{2}{\omega}\, \frac{e^{j2\omega} - e^{-j2\omega}}{2j}$$

$$= \frac{2}{\omega} \sin 2\omega$$

$$= \frac{2}{\omega} \sin\left(\frac{4\omega}{2}\right)$$

$$= \frac{\sin\left(\frac{4\omega}{2}\right)}{\frac{\omega}{2}}$$

$$= 4 \frac{\sin\left(\frac{4\omega}{2}\right)}{\frac{4\omega}{2}}$$

$$\Rightarrow X[j\omega) = 4 \operatorname{sinc}\left(\frac{4\omega}{2}\right)$$

$$\Rightarrow X(jf) = 4 \operatorname{sinc}\left(4 \frac{2\pi f}{2}\right)$$

$$\Rightarrow X(jf) = 4 \operatorname{sinc}(4\pi f)$$

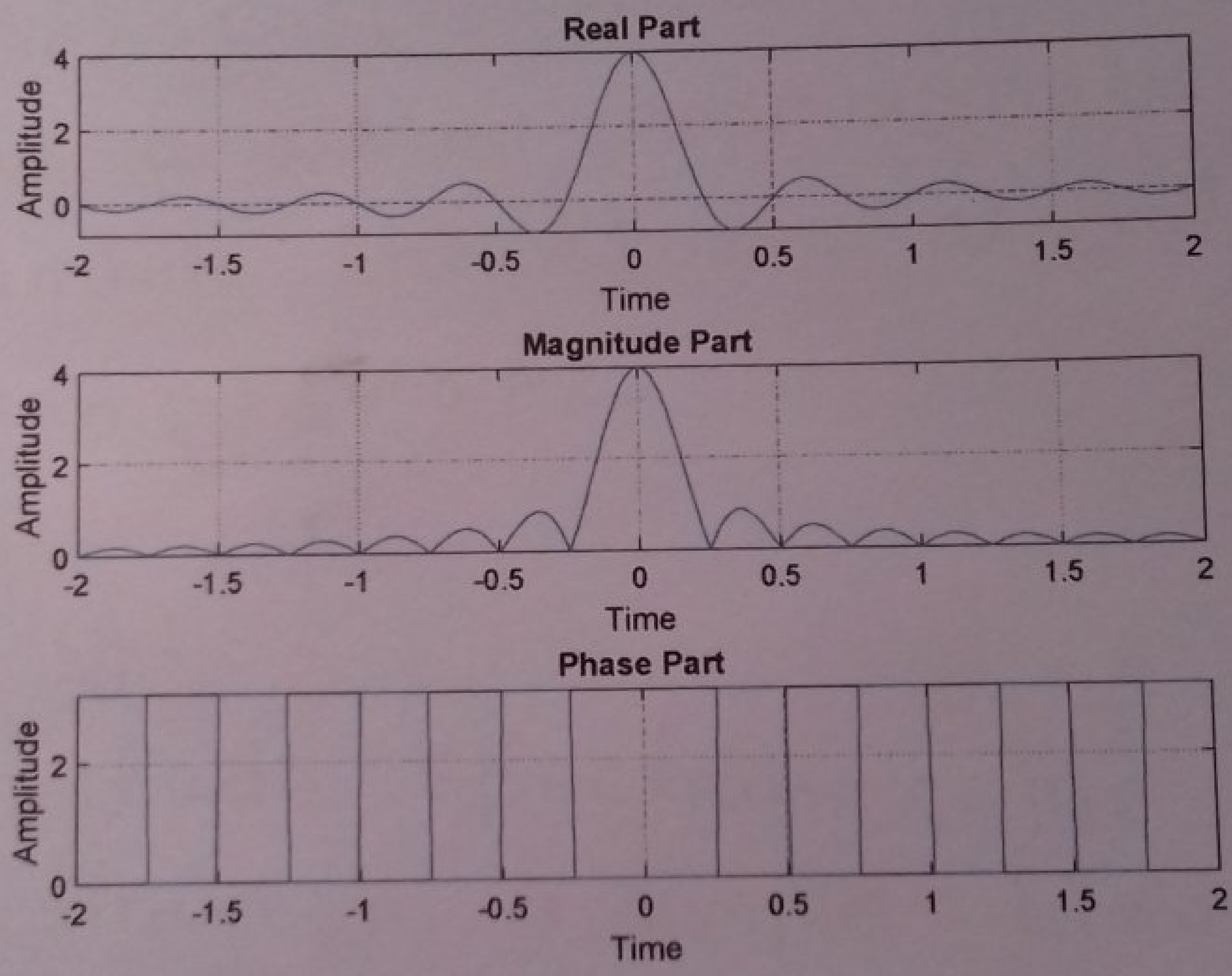
$\therefore$ The aperiodic pulse shown above has a Fourier Transform

$$X(jf) = 4 \operatorname{sinc}(4\pi f)$$

Source code:

```
Clc;
Clear all;
Close all;
t = 2:0.01:2;
x = 4 * Sinc(4*t);
Subplot(3,1,1);
plot(t,x);
xlabel('Time');
ylabel('Amplitude');
title('Real part');
grid on;

subplot(3,1,2);
plot(t, abs(x));
xlabel('Time');
ylabel('Amplitude');
title('Magnitude part');
grid on;

Subplot(3,1,3);
plot(t, angle(x));
```

```
xlabel('Time');
ylabel('Amplitude');
title('Phase part');
grid on;
```


Real Part
Amplitude
Time
Magnitude Part
Amplitude
Time
Phase Part
Amplitude
Time

Experiment No: 08

Experiment Name: To find the amplitude spectrum of the two frequency signal:

$$x(t) = \cos(2\pi 100t) + \cos(2\pi 500t)$$

and also find approximate the Fourier transform of integral for $0 \leq f \leq 800$ Hz.

Theory:

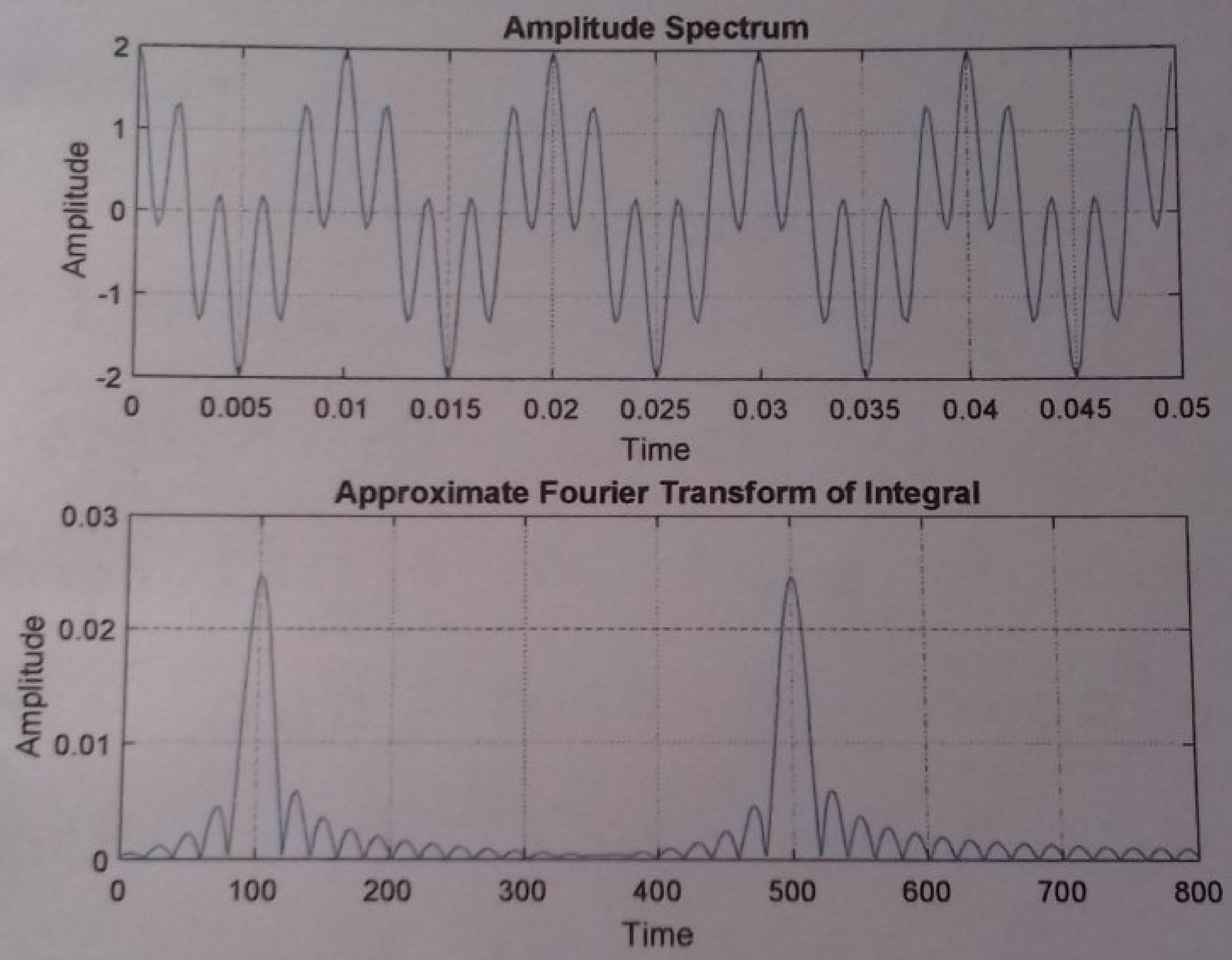
Amplitude Spectrum: The amplitude Spectrum is a simple transformation of the DFT. The amplitude spectrum is the vector that contains the absolute values of the co-efficients of the frequency domain representation of x.

It shows which frequencies contribute more to the magnitude of x.

Source code :

```
clc;
clear all;
close all;
N =250;
ts = 0.0002;
t = [0:N-1]*ts;
x = cos(2*pi*100*t)+cos(2*pi*500*t);
Subplot(2,1,1);
plot(t,x);
xlabel('Time');
ylabel('Amplitude');
title('Amplitude Spectrum');
grid on;


k = 0;
for f=0:1:800
    k=k+1;
    X(k)= trapz(t,x*exp(-1i*pi*2*f*t));

end
```

```
f = 0:800;
Subplot (2,1,2);
Plot (f, abs(x));
xlabel('Time');
ylabel ('Amplitude');
title('Approximate Fourier Transform of
                                    Integral');
grid on;
```



Amplitude Spectrum
Amplitude
Time
0 0.005 0.01 0.015 0.02 0.025 0.03 0.035 0.04 0.045 0.05
2 1 0 -1 -2
Approximate Fourier Transform of Integral
Amplitude
Time
0 100 200 300 400 500 600 700 800
0.03 0.02 0.01 0

Experiment No: 09

Experiment Name: Explain and generate sinusoidal wave with different frequency using MATLAB.

Theory: A sine wave, sinusoidal wave is a mathematical curve defined in terms of the sine trigonometric function of which it is the graph. It is a type of continuous wave and also a smooth periodic function. A sine wave shows how the amplitude of a variable changes with time. It is often use in pure and applied mathematics as well as physics, signal processing engineering and many other field.

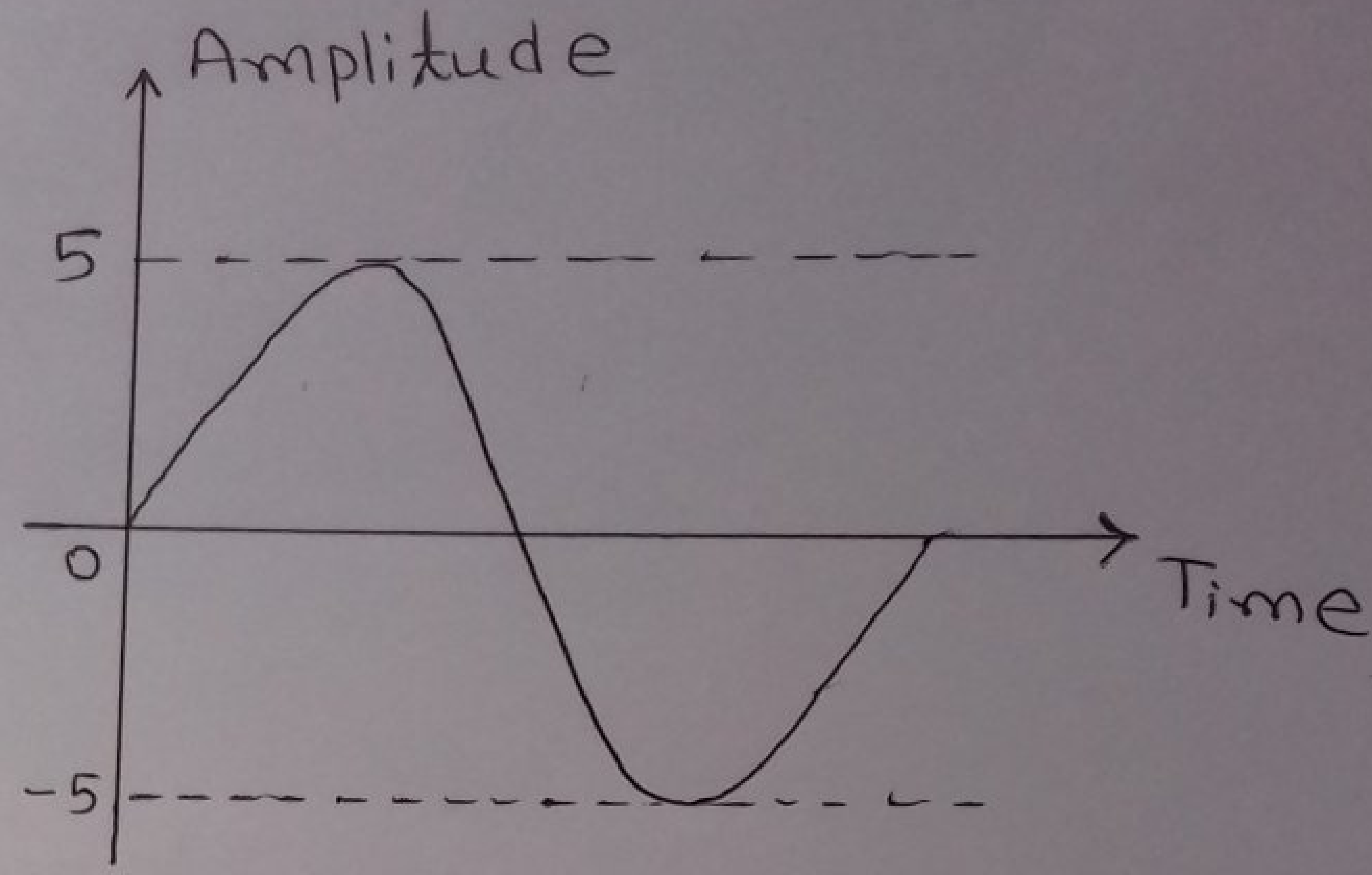


Fig: A Sine wave curve

The equation of the sinusoidal curve is,

$$y = A_m \sin \omega t$$
$$= A_m \sin 2\pi f t$$

where,

A_m = Amplitude

f = frequency

t = period

Phase, $\theta = 0$

Source Code:

```
clc;
close all;
clear all;
f = 100;
t = 1/f;
t1 = 0:t/100:t;
a = 5;
y = a * sin(2*pi*f*t1)
subplot(2,1,1);
plot(y);

Am = 1;
fm = 5;
t = 0:0.001:1;
wm = 2*pi*fm;
msg-sig = Am*sin(wm*t);
subplot(2,1,2);
plot(t, msg-sig);
```

Experiment No: 10

Experiment Name: Explain and implementation of following elementary dicrete signals.

(i) The unit sample sequence.

(ii) The unit step signal.

(iii) The unit ramp signal.

Theory:

(i) Unit sample sequence: Unit sample sequence is also called unit impluse. The unit impluse sequence is a sequence of discrete samples that has unit magnitude at origin and zero-magnitude at all other sample instants.

The discrete time version of unit impluse is defined by

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

Impluse function has zero duration infinite amplitude and unit area under it.

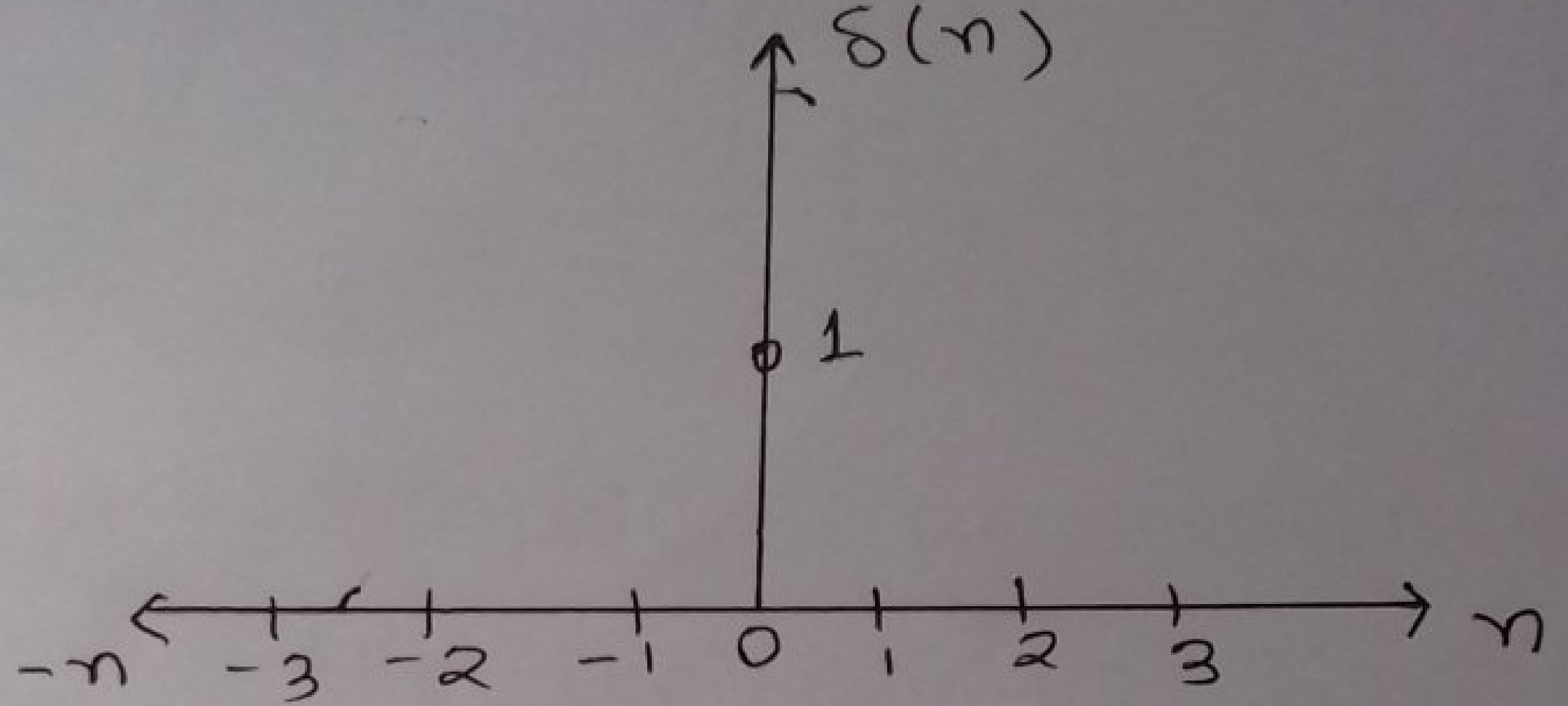


Fig-a: Graphical Representation of the unit sample signal.

Unit step signal:

The discrete time unit step signal is denoted as u(n) and is defined as,

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n \leq 0 \end{cases}$$

The graphical representation of unit step function is,

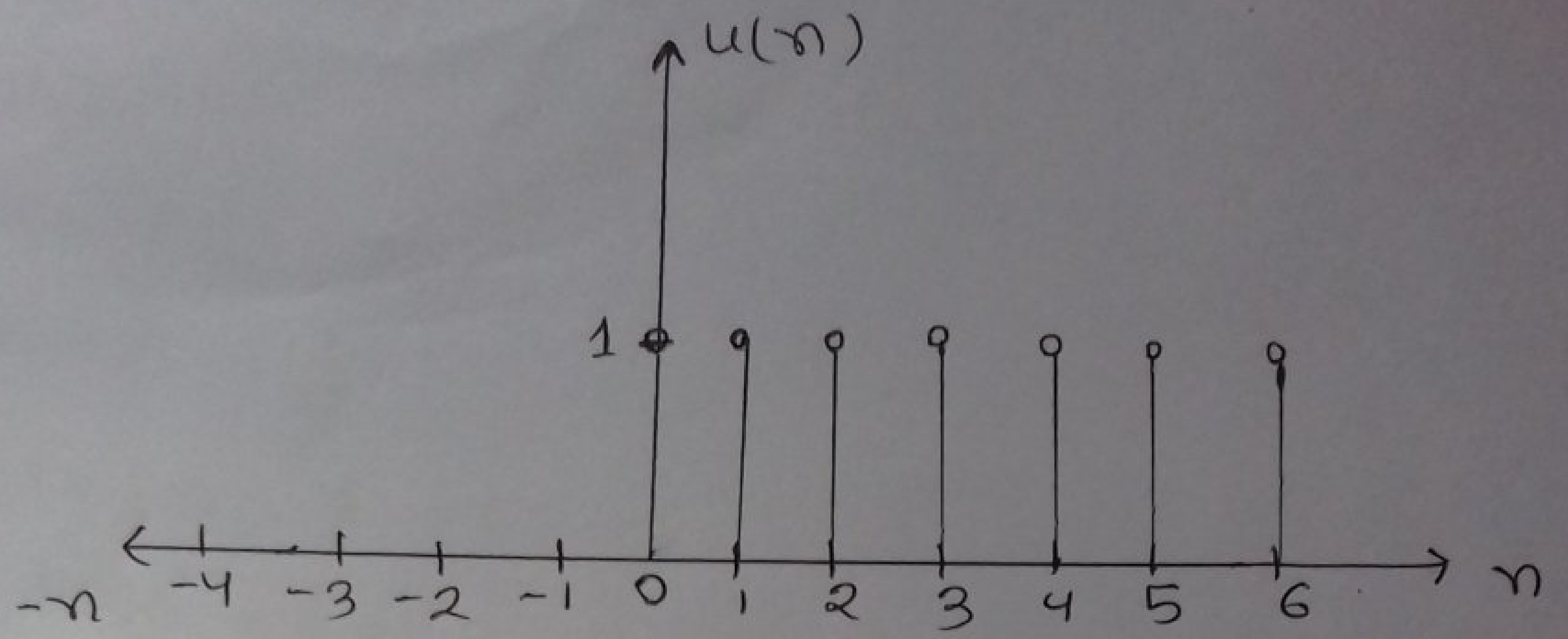
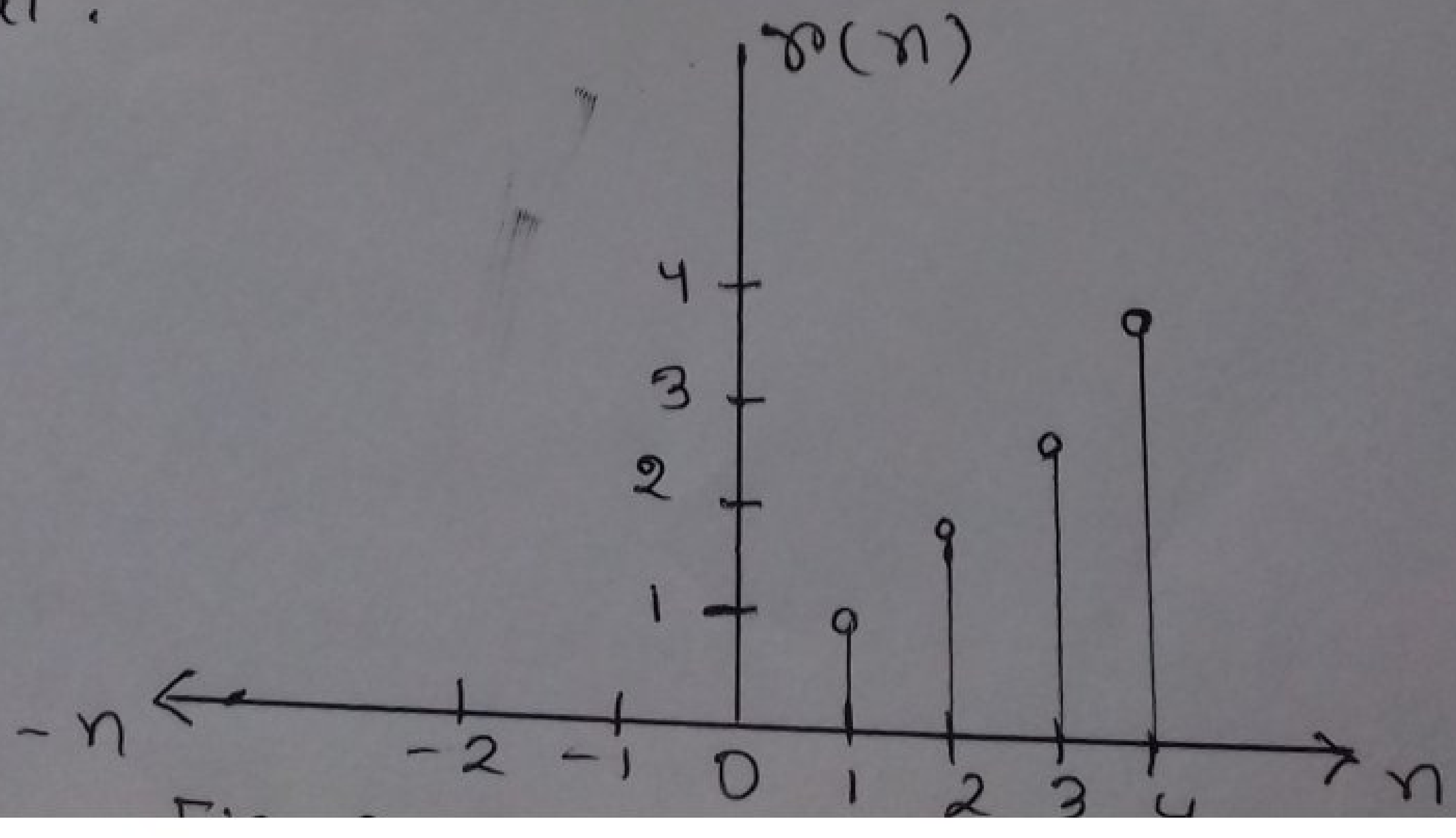


Fig-6: Graphical representation of the unit step signal.

Unit ramp signal: The discrete time unit ramp signal is denoted as $r(n)$ and is defined as,

$$r(n) = \begin{cases} n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

graphical representation of ramp signal.



Source code :

```
clc;
close all;
t = -20:1:20;

unitstep = t>=0;
impluse = t==0;
unitramp = t.* unitstep;

Subplot (3,1,1);
stem( t, unitstep);
xlabel ('Time');
ylabel ('Amplitude');
title('Unitstep Discrete Time');
grid on;
ylim ( [0,2] );

Subplot(3,1,2);
stem(t, impluse);
grid on;
xlabel ('Time');
ylabel ('Amplitude');
title('impluse Discrete Time');
```

```
Subplot(3,1,3);
stem(t, unitramp);
grid on;
xlabel('Time');
ylabel('Amplitude');
title('Unitramp Discrete Time');
```