Information and Communication Engineering
Pabna University of Science and Technology
Faculty of Engineering and Technology
B.Sc. (Engineering) 2nd Year 2nd Semester Examination-2018
Session: 2016-2017

Course Code: ICE-2206 **Course Title: Signals and Systems Sessional**

Time: 4 hours **Marks: 60**

E1. Plot the following signals: i) Unit Step, ii) Unit Impulse, and iii) Ramp signals.

E2. Let $x(n) = \{1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1\}$. Determine and plot the following Sequence:

↑

$$x1(n) = 2x(n-5) - 3x(n+4).$$

E3. Write MATLAB code to perform the following operations on a Sinusoidal wave:

i) Sampling, ii) Quantization, and iii) Coding.

E4. Determine and plot the following sequences over the indicated interval using MATLAB:

$$x(n) = 2\delta(n+2) - \delta(n-4)\,; \; -5 \le n \le 5.$$

E5. Plot the following signal operations on the following signals:

$x = \{1, 0, 3, 4\}$; $y = \{1, 1, 1, 1\}$; $z = \{3, -1, 0, -4\}$;

↑ ↑ ↑

i) Signal Addition (x + y), and ii) Folding of signal z.

E6. Plot following signal operations:

$x = \{1, 2, 3, 4\}$; $y = \{1, 1, 1, 1\}$; $z = \{-2, 3, 0, 1, 5\}$;

↑ ↑ ↑

i) Signal Multiplication (x*y) and ii) Signal Shifting (z).

E7. Write a MATLAB code to determine the frequency response $H(e^{j\omega})$ of a system characterized by $h(n) = (0.9)^n u(n)$. Plot the magnitude and phase responses.

E8. Let $x_a(t) = e^{-1000|t|}$. Determine and plot its Fourier Transform using MATLAB.

E9. Find the 4-point DFT of a sequence $x(n) = \{1, 2, 1, 1\}$.

E10. Find the 4-point IDFT of a sequence $X(k) = \{4, 0, 0, 0\}$.

Pabna University of Science and Technology

Faculty of Engineering and Technology

Department of

Information and Communication Engineering(ICE)

Lab Report

Course Code: **ICE-2206**

Course Title: **Signals and Systems Sessional**

Submitted By:

Name: Md. Maruf Hossain

ID: 170626
Session: 2016-17
Dept. of Information and Communication Engineering
Pabna University of Science and Technology

Date of issue:

Date of Submission: 13-07-19

Submitted To:

Sohag Sarker
Assistant Professor

Dept. of Information and Communication Engine
Pabna University of Science and Technology

Signature :

INDEX

07.	Write a MATLAB program to determine the frequency response $H(e^{jw})$ of a system characterized by $h(n)=(0.9)^n u(n)$. Plot the magnitude and the phase response.	12
08.	Let $x(t)=e^{-1000|t|}$. Determine and plot it's Fourier Transform using MATLAB.	13
09.	Find the 4-point DFT of a sequence x(n).	14
10.	Find the 4-point IDFT of a sequence x(k).	15

Experiment NO — 01

Experiment name: plot following signal operations using user defined function

i) Addition
ii) Folding.

Theory:

Addition of signals: Consider a pair of continuous time signal $x_1(t)$ & $x_2(t)$. Adding these two signals $x_1(t)$ & $x_2(t)$ result in a signal $y(t)$. The period of the signal is unaltered.

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

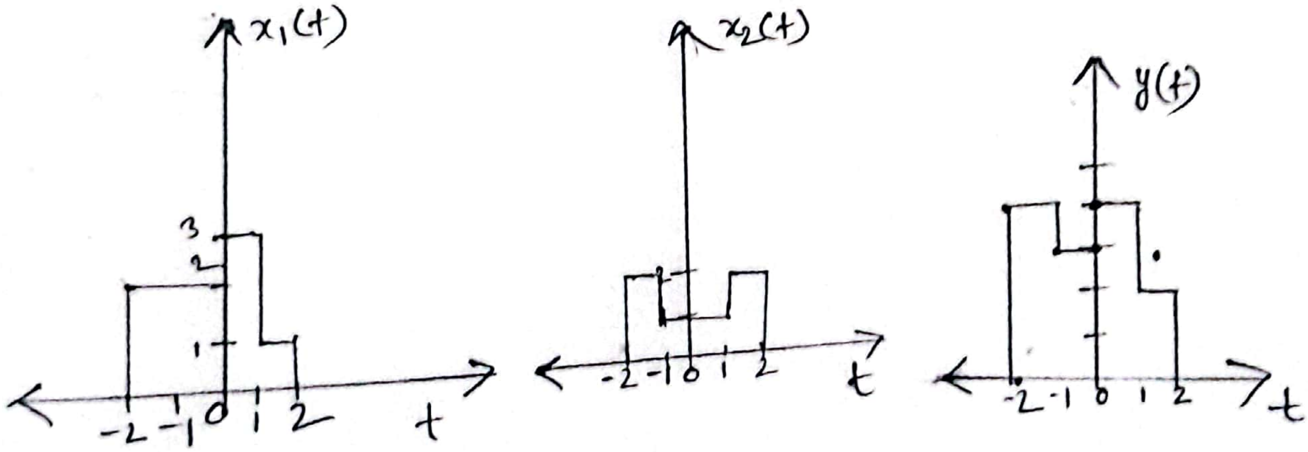


Fig-1: Addition of continuous time signal.

Consider a pair of discrete time signals $x_1(n)$ & $x_2(n)$ in a signal produced these two signals $x_1(n)$ & $x_2(n)$ $y(n)$.

The period of $y(n)$ is unaltered.

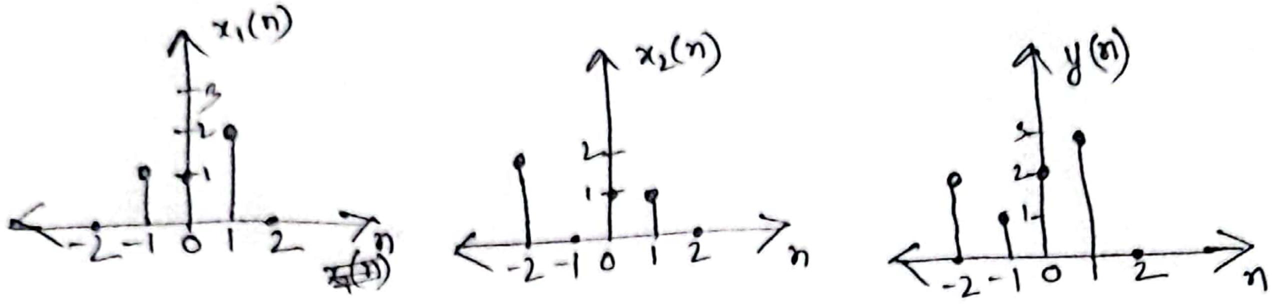


Fig-2: Discrete time signal addition.

Folding of signals: Consider a discrete time signal $x(n)$ folding means converting the position in positive to negative or negative to positive. The period of $x(n)$ is unchanged.

so, $x(n) = x(-n)$,

Example:

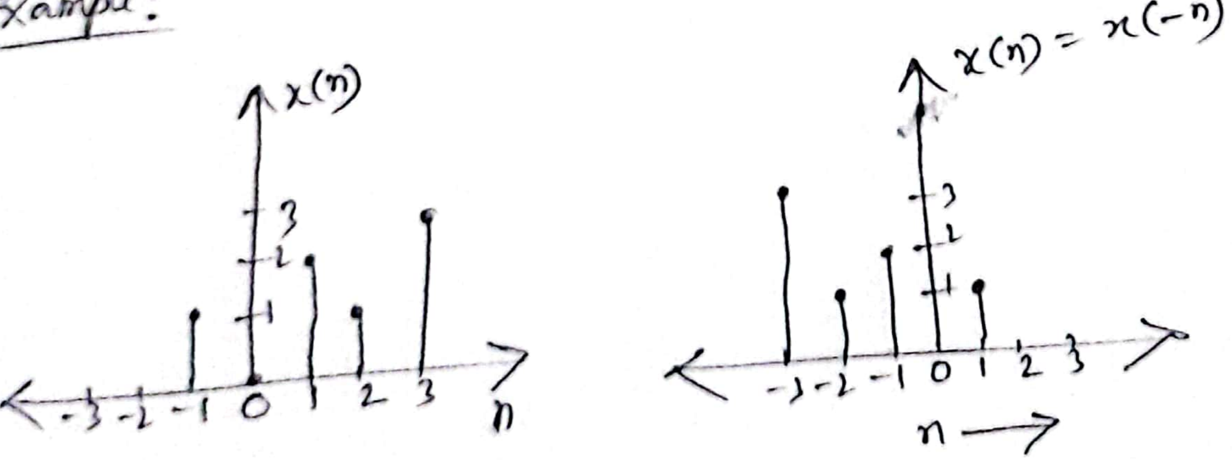


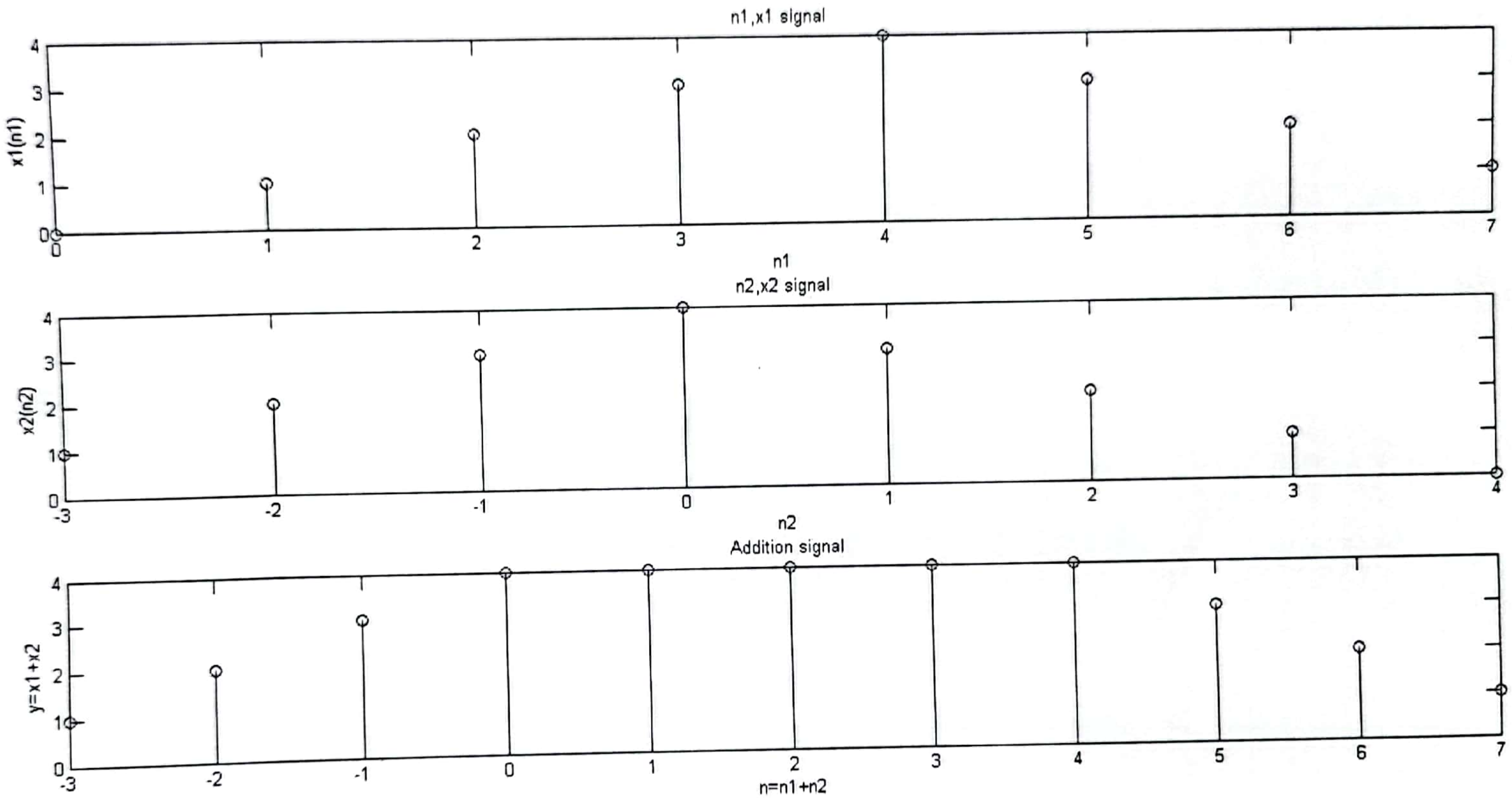
Fig-3: Folding of signals.

MATLAB Source Code:

For Addition of Signal:

```
clc;
closeall;
clearall;
n1 = 0:7;
x1 = [0,1,2,3,4,3,2,1];
subplot(3,1,1);
stem(n1,x1);
title('n1,x1 signal');
xlabel('n1');
ylabel('x1(n1)');
n2 = -3:4;
x2 = [1,2,3,4,3,2,1,0];
subplot(3,1,2);
stem(n2,x2);
title('n2,x2 signal');
xlabel('n2');
ylabel('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;
subplot(3,1,3);
stem(n,y);
title('Addition signal');
xlabel('n=n1+n2');
ylabel('y=x1+x2');
```

output:

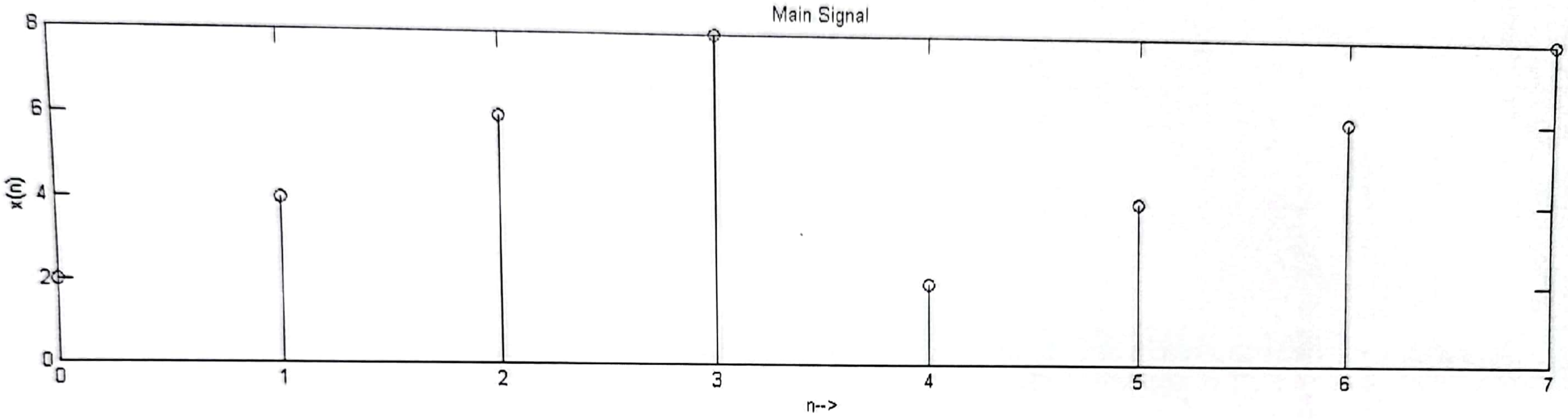


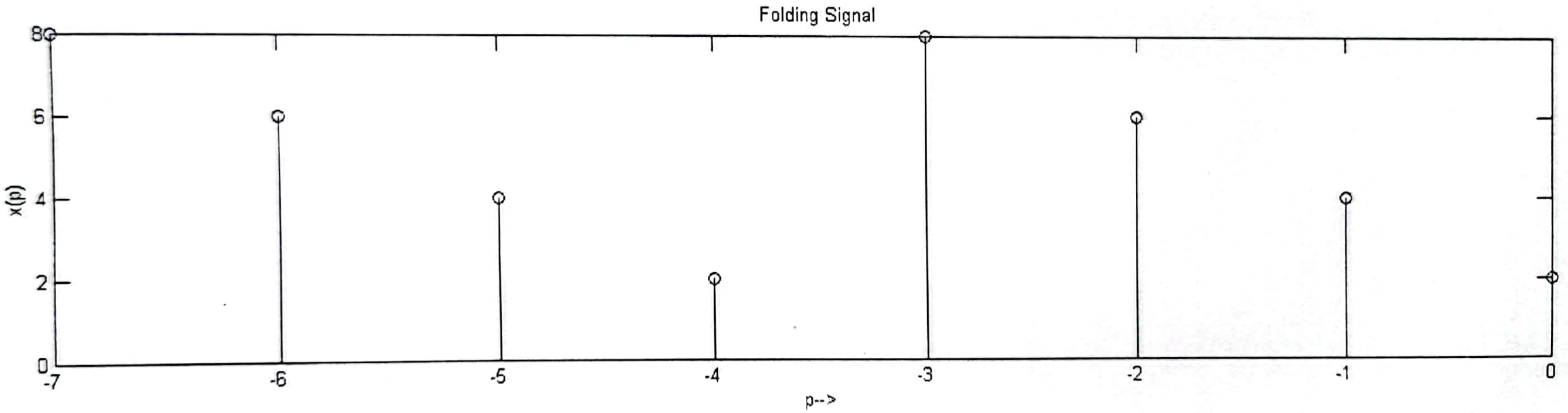
Scanned with CamScanner

Signal Folding Operation:

```
clc;
closeall;
clearall;
n = 0:7;
x = [2,4,6,8,2,4,6,8];
subplot(2,1,1);
stem(n,x);
title('Main Signal');
xlabel('n-->');
ylabel('x(n)');
p = (-1)*n;   %%m=fliplr(n),y=fliplr(x)
subplot(2,1,2);
stem(p,x);
title('Folding Signal');
xlabel('p-->');
ylabel('x(p)');
```

output:


Main Signal
x(n)
n-->
0
1
2
3
4
5
6
7
8


Folding Signal
x(p)
p-->
-7
-6
-5
-4
-3
-2
-1
0
2
4
6
8

Experiment — 02

Experiment name: plot following signals operations using user defined function.

i) signal Multiplication
ii) signal shifting.

Theory:

Signal Multiplication: Multiplication of signal is the basic operation on signals.

Consider a pair of discrete time signal $x_1(n)$ & $x_2(n)$ multiplication of these two discrete time signals, $x_1(n)$ & $x_2(n)$ & resultant signal $y(n)$.

$$\therefore y(n) = x_1(n) * x_2(n).$$

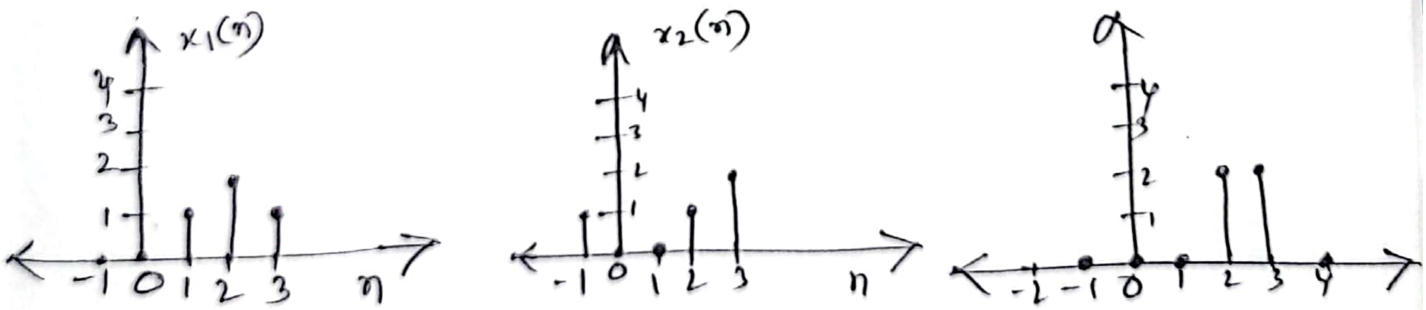


Fig-1: Multiplication of two discrete time signals.

shifting of signal: Let us Consider a discrete time signal $x(n)$. Let $y(n)$ is a signal denote to obtain by shifting the signal $x(n)$ by $(n-n_0)$ that is $y(n) = x(n-n_0)$

$y(n) = x(n-n_0)$ denote right shift
$y(n) = x(n+n_0)$ denote left shift,

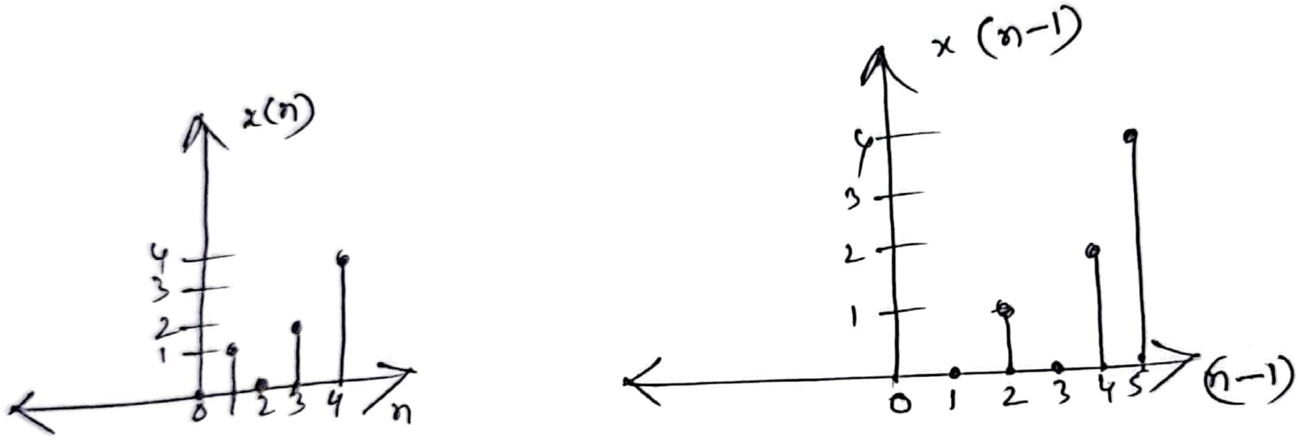


Fig-02: signal shifting operation.

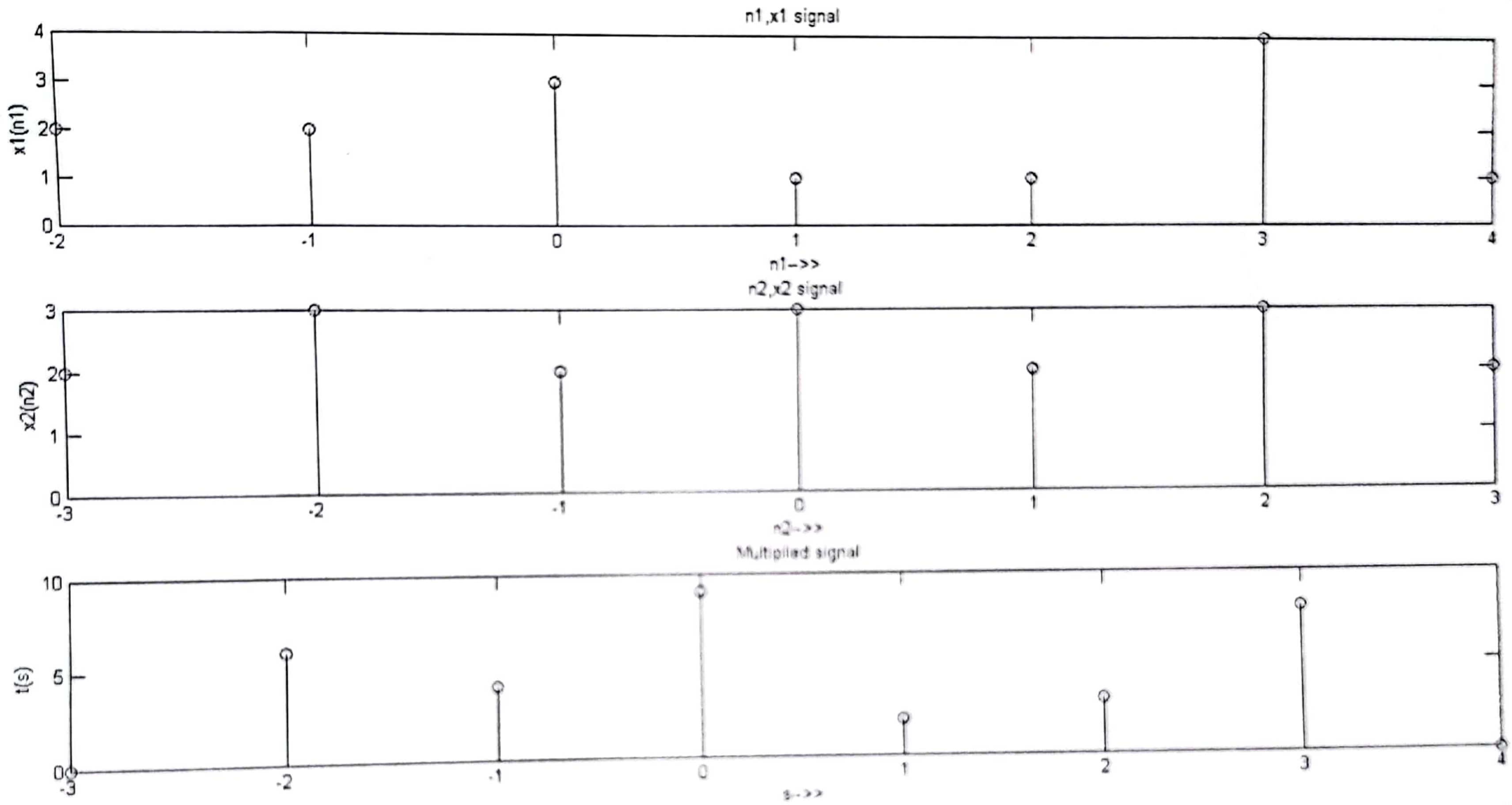
Multiplication Operation:

```
clc;
clearall;
closeall;
n1 = -2:4;
x1 = [2,2,3,1,1,4,1];
subplot(3,1,1);
stem(n1,x1);
title('n1,x1 signal');
xlabel('n1-->>');
ylabel('x1(n1)');
n2 = -3:3;
x2 = [2,3,2,3,2,3,2];
subplot(3,1,2);
stem(n2,x2);
title('n2,x2 signal');
xlabel('n2-->>');
ylabel('x2(n2)');
[s,t] = usermul(n1,x1,n2,x2);
subplot(3,1,3);
stem(s,t);
title('Multiplied signal');
xlabel('s-->>');
ylabel('t(s)');
```

Function usermul.m:

```
function [n,y] = usermul(n1,x1,n2,x2)

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;
end
```

output:



n1,x1 signal
x1(n1)
n1-->>
n2,x2 signal
x2(n2)
n2-->>
Multiplied signal
t(s)
s-->>

MATLAB Source Code:

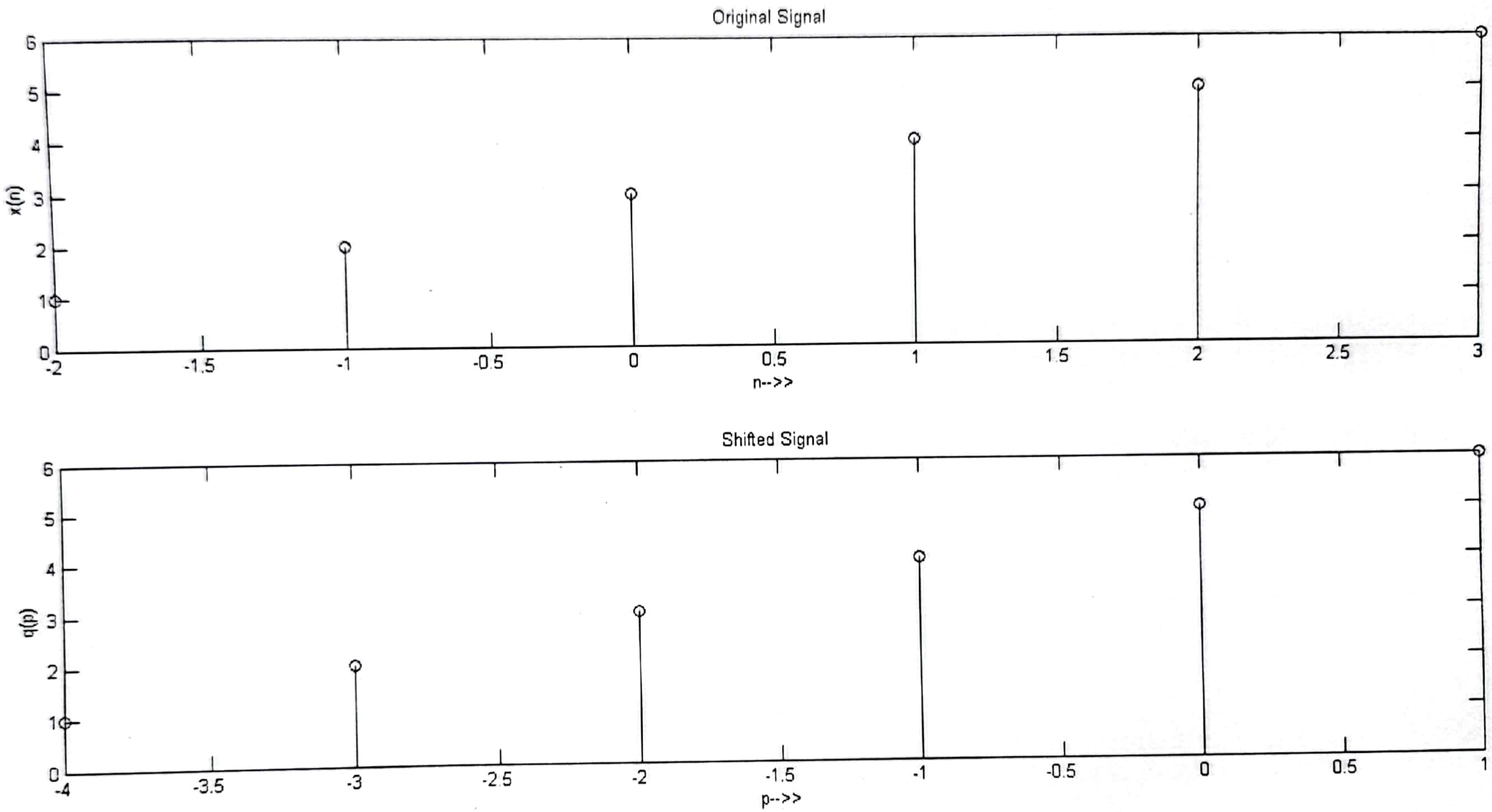
Shifting Operation:

```
clc;
clearall;
closeall;
n = -2:3;
k = -2;
x = [1,2,3,4,5,6];
subplot(2,1,1);
stem(n,x);
title('Original Signal');
xlabel('n-->>');
ylabel('x(n)');
[p,q] = user(k,n,x);
subplot(2,1,2);
stem(p,q);
title('Shifted Signal');
xlabel('p-->>');
ylabel('q(p)');
```

Function user.m:

```
function [r,s] = user(k,n,x)
r = n+k;
s = x;
end
```

output:



Original Signal
x(n)
n-->>
0
1
2
3
4
5
6
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
Shifted Signal
q(p)
p-->>
0
1
2
3
4
5
6
-4
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1

Experiment NO — 03

Experiment Name: Explain & implement following elementary discrete signal using MATLAB.

i) unit sample sequence / unit impulse signal.

ii) unit step signal

iii) unit ramp signal.

Theory:

i) unit sample sequence / unit impulse signal.

The unit impulse signal is denoted as $\delta(n)$ & defined as

$$\delta(n) = \begin{cases} 1, & \text{for } n = 0 \\ 0, & \text{otherwise} \end{cases}$$

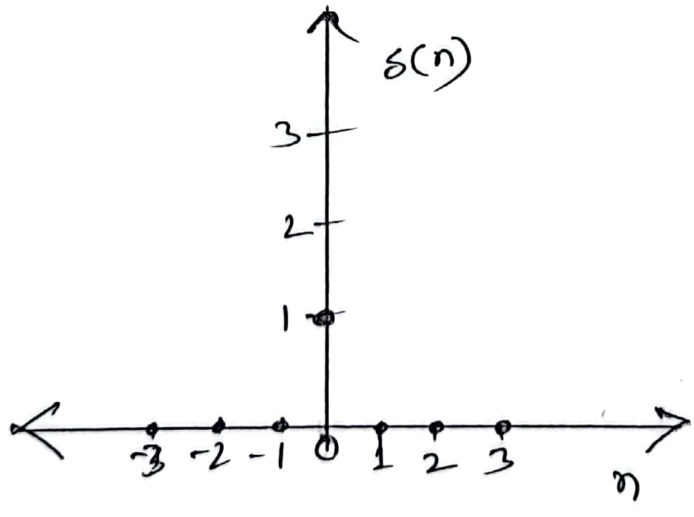


Fig-1: unit impulse signal.

ii) unit step signal

The unit step signal is denoted as $u(n)$ & defined as

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

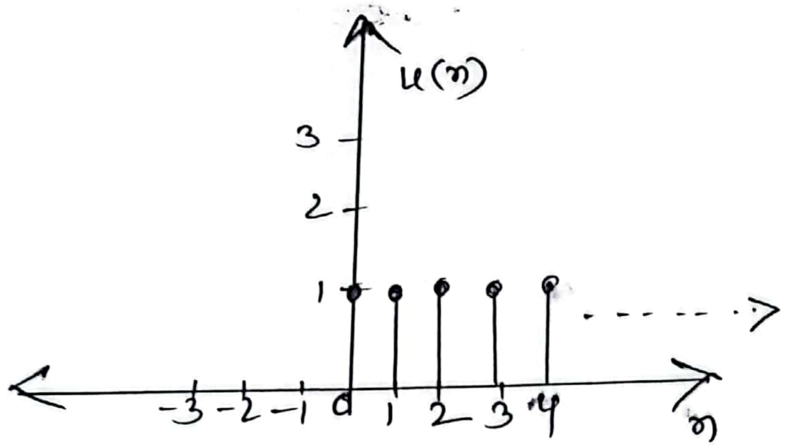


Fig-2: unit step signal.

iii) unit ramp signal

The unit ramp signal is denoted as $ur(n)$ & is defined as

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

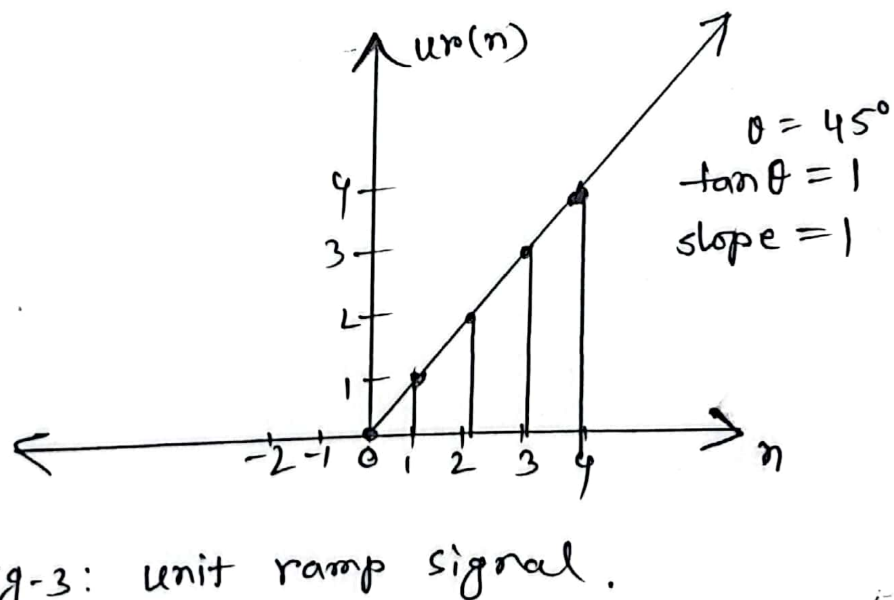
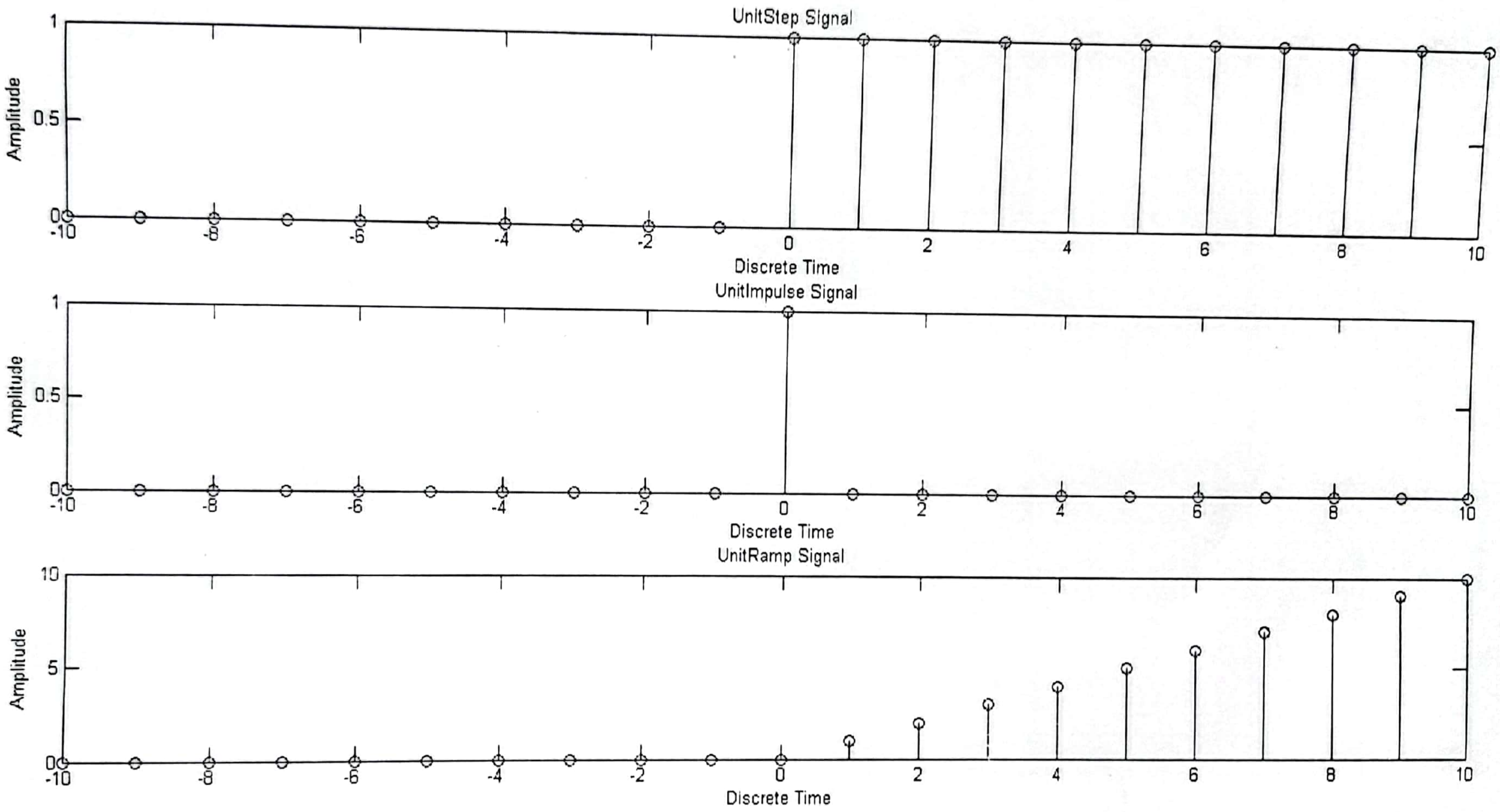


Fig-3: unit ramp signal.

MATLAB source Code:

```
clc;
clearall;
closeall;
n0 = -10:10;
x = [(n0>=0)==1];
subplot(3,1,1);
stem(n0,x);
title('UnitStep Signal');
xlabel('Discrete Time');
ylabel('Amplitude');
y = [(n0==0)==1];
subplot(3,1,2);
stem(n0,y);
title('UnitImpulse Signal');
xlabel('Discrete Time');
ylabel('Amplitude');
k = [(((n0>=1)==1).*n0)];
subplot(3,1,3);
stem(n0,k);
title('UnitRamp Signal');
xlabel('Discrete Time');
ylabel('Amplitude');
```

output:



UnitStep Signal
Amplitude
1
0.5
0
-10
-8
-6
-4
-2
0
2
4
6
8
10
Discrete Time
UnitImpulse Signal
Amplitude
1
0.5
0
-10
-8
-6
-4
-2
0
2
4
6
8
10
Discrete Time
UnitRamp Signal
Amplitude
10
5
0
-10
-8
-6
-4
-2
0
2
4
6
8
10
Discrete Time

Experiment no.— 04

Experiment name : Let $x(n) = \{1, 2, 3, 4, 5, 6, \underset{\uparrow}{7}, 6, 5, 4, 3, 2, 1\}$
Determine & plot the following sequence $x1(n) = 2x(n-5) - 3x(n+4)$
where, $x(0) = 7$.

Theory:

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

Here, we should plotting the $x1(n) = 2x(n-5) - 3x(n+4)$

where,
$x(n-5)$ denotes the right shift of every single point by 5 times.

Then $2x(n-5)$ is denote the multiplication of each sample amplitude by 2.

same for $3x(n+4)$ but here $x(n+4)$ implies only left shift.

Then these two signals subtraction is the resultant signal which is required.

MATLAB Source Code:

```
clc;
clearall;
closeall;
n = [-6:6];
x = [1:7,6:-1:1];
[n1,x1] = shift(n,x,5);
[n2,x2] = shift(n,x,-4);
[n3,x3] =  add(n1,n2,2*x1,(-3)*x2);
stem(n3,x3);
title('The Desired Sequence');
xlabel('n-->>');
ylabel('Amplitude');
```

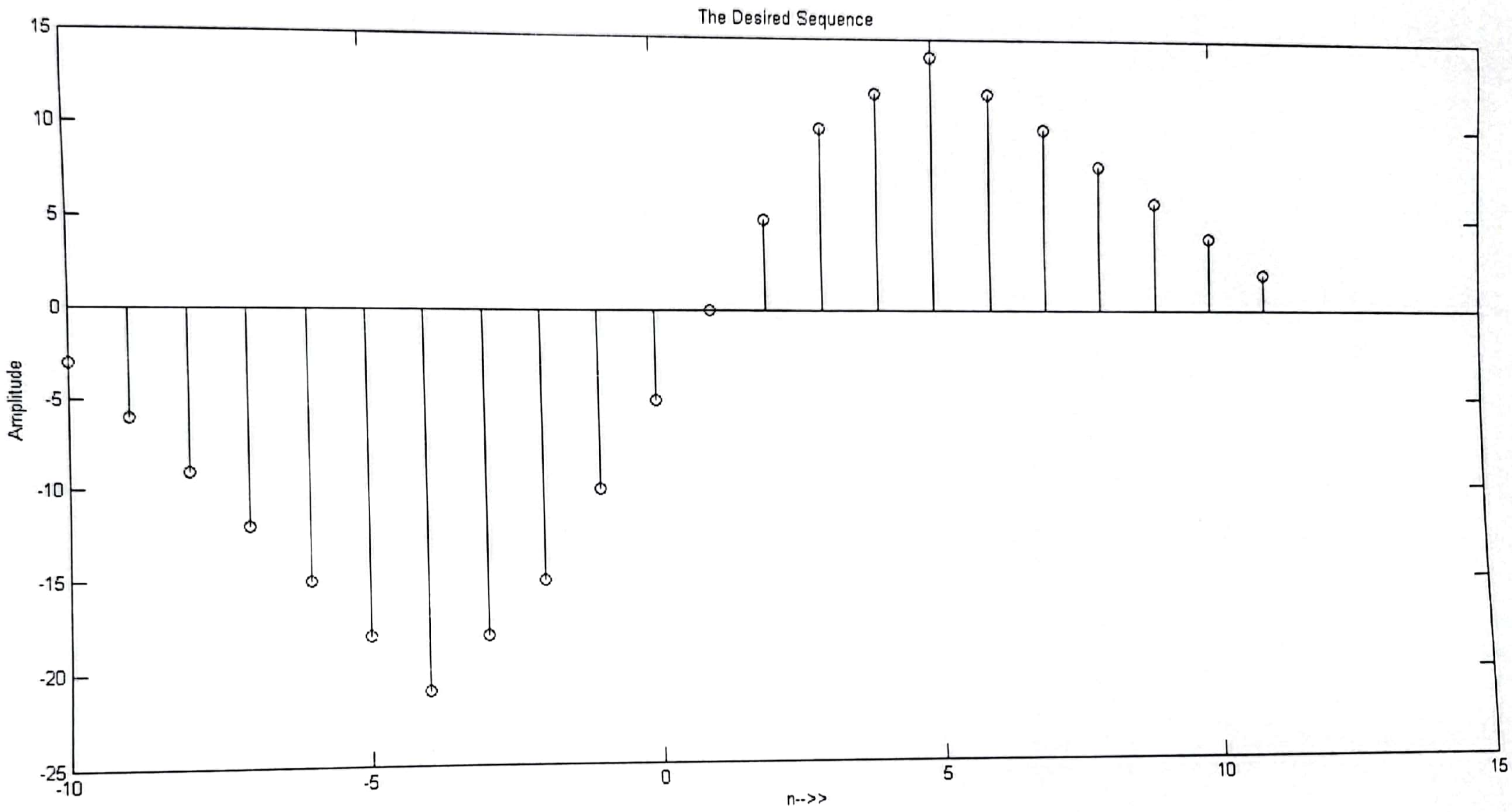
shift function:

```
function[p,q] = shift(n,x,val)
p = n+val;
q = x;
end
```

add function:

```
function [p,q] = add(n1,n2,x1,x2)
p = min(min(n1),min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(p));
y2 = y1;
y1(find((p>=min(n1)) & (p<=max(n1))==1))=x1;
y2(find((p>=min(n2)) & (p<=max(n2))==1))=x2;
q = y1+y2;
end
```

output:



The Desired Sequence
15
10
5
0
-5
-10
-15
-20
-25
Amplitude
-10
-5
0
5
10
15
n-->>

Experiment NO — 05

Experiment Name: Determine & plot the following sequence

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

Theory:

Discrete-time signal: A signal $x(n)$ is said to be discrete time signal if it can be defined for a discrete instant of time (say n)

Unit impulse signal: It is defined in two ways

i) Discrete time

ii) Continuous time.

(i) Discrete time unit impulse signal: It is denoted as $\delta(n)$ and is defined as

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

(ii) Continuous time unit impulse signal: It is denoted as $\delta(t)$ and is defined as

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

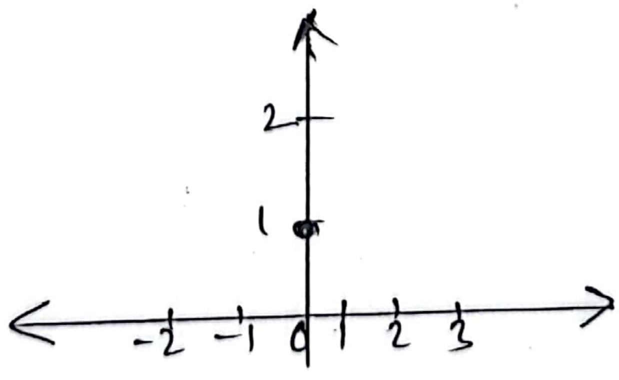


Fig-01: unit impulse signal.

Here, $\delta(n+2)$ is defined as the unit impulse signal amplitude will be unity at the point $n=-2$ because $\delta(n+2)$ denoting the left shift.

This is also true for $\delta(n-4)$, but here is only deform with the shifting.

Then we subtract the two unit impulse signal and hence desired output will be got.

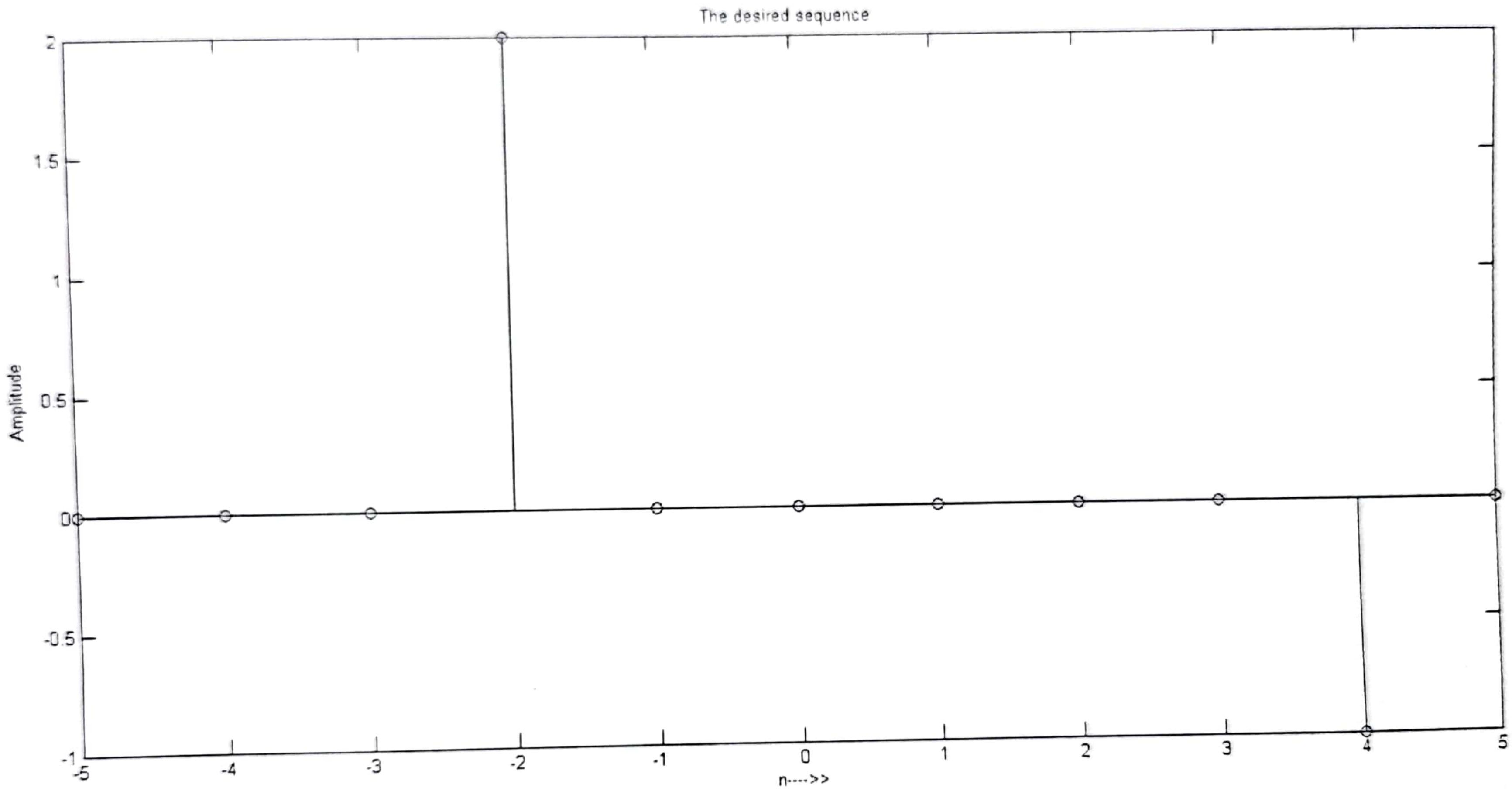
MATLAB Code:

```
clc;
clearall;
closeall;
n1 = -5;
n2 = 5;
n = n1:n2;
[p1,q1]=shifting(n1,n2,-2);
[p2,q2]=shifting(n1,n2,4);
y = ((2.*q1) - q2);
figure(1);
stem(n,y);
title('The desired sequence');
xlabel('n---->>');
ylabel('Amplitude');
```

Method **shifting.m**

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

outputs:



Experiment NO — 06

Experiment name: Write a MATLAB program to perform following operations

i) sampling ii) Quatization iii) Coding.

Theory:

Sampling: This is the conversion of a continuous time signal into a discrete time signal obtained by taking sampleb of the continuous time signal at discrete time instants. Thus if $x_a(t)$ is the input to the sampler the output is $x_a(nT) = x(n)$ where T is called the sampling interval.

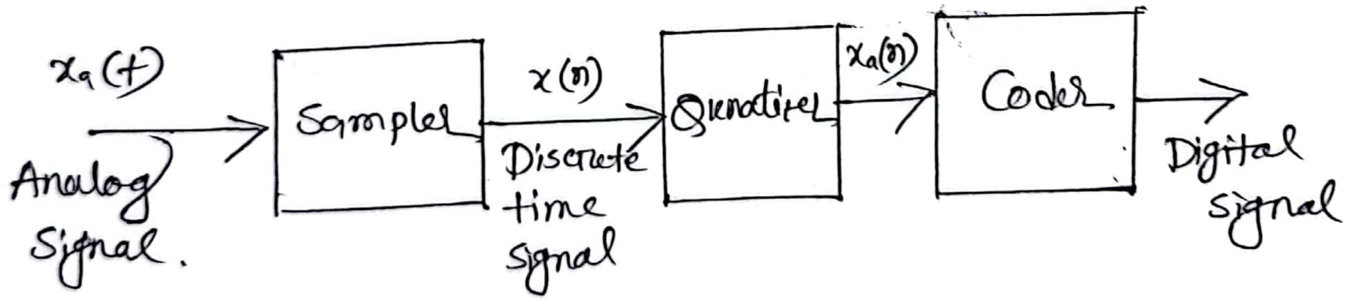


Fig-01: Block diagram of analog to digital Converter.

Quatization: Quatization in mathematics & digital signal processing is the process of maping a large set of input values to a smaller set. Rounding and truncation are typical examples of quantization process.

The difference between an input value & its quantinized value (such as round-off-error) is reffered to as quatinization error. A device or algorithmic function that performs quantization is called quantizer.

Coding: A system of signals used to represent letters or numbers in transmitting message. The instruction in a computer program.

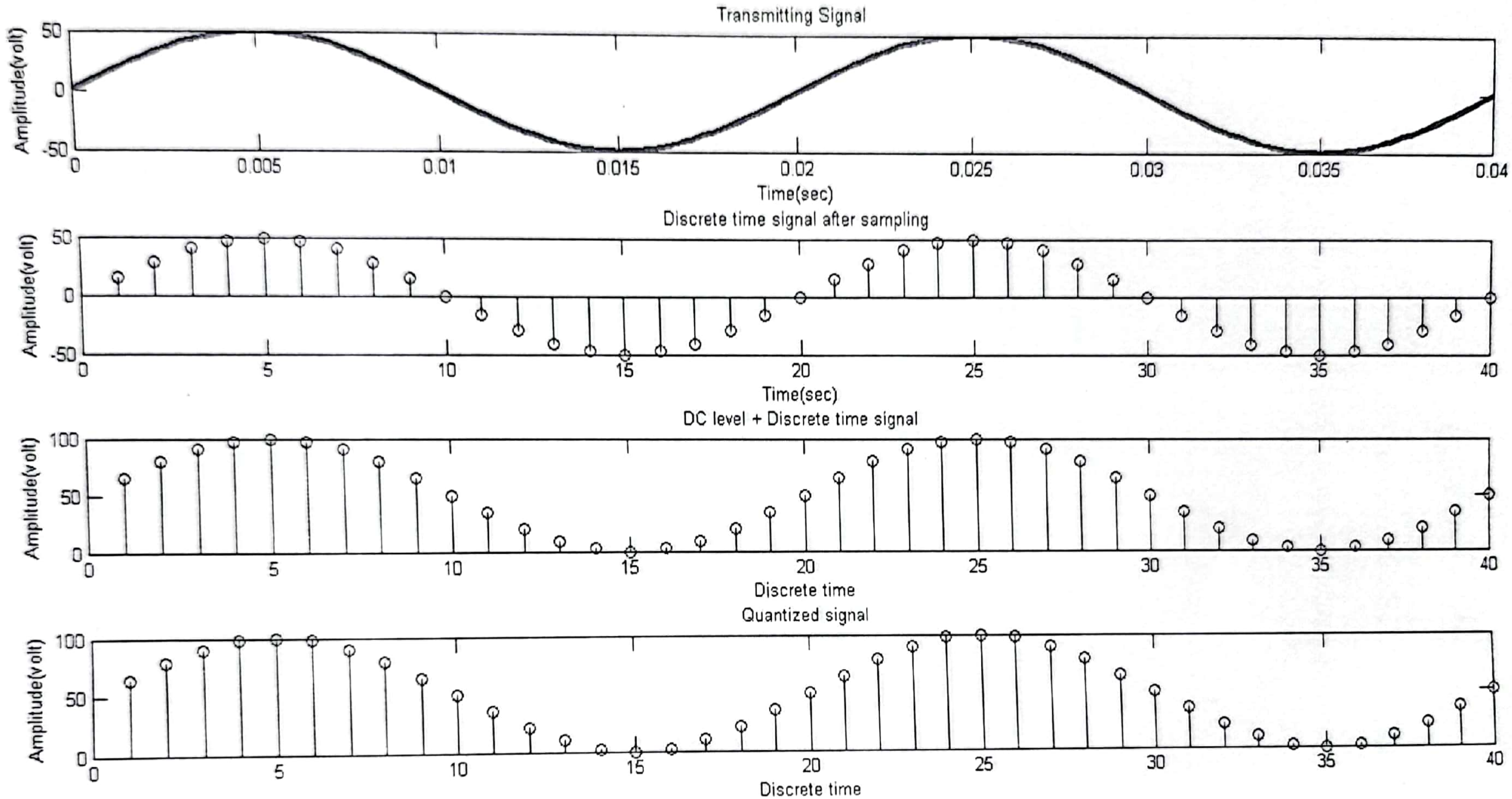
MATLAB Source Code:

```
clc;
closeall;
clearall;
A = input('Enter amplitude of Transmitting Signal: ');

%Transmitting signal generation
f = 50;
T = 1/f;
t = 0:T/100:2*T;
y = A*sin(2*pi*f*t);
subplot(4,1,1);
plot(t,y,'r','Linewidth',3);
title('Transmitting Signal');
xlabel('Time(sec)');
ylabel('Amplitude(volt)');

%Sampling
n = 1:1:40;
y1 = A*sin(2*pi*f*(.001)*n);
subplot(4,1,2);
stem(n,y1);
title('Discrete time signal after sampling');
xlabel('Time(sec)');
ylabel('Amplitude(volt)');

%DC level + Discrete time signal
y2 = A+y1;
subplot(4,1,3);
stem(n,y2);
title('DC level + Discrete time signal');
xlabel('Discrete time');
ylabel('Amplitude(volt)');

%Quantization
y3 = round(y2);
subplot(4,1,4);
stem(n,y3);
title('Quantized signal');
xlabel('Discrete time');
ylabel('Amplitude(volt)');

%Coding
y4 = dec2bin(y3);
disp('Binary Information');
disp(y4);
```

Output:

```
Enter amplitude of Transmitting Signal: 50
Binary Information
1000001
1001111
1011010
1100010
1100100
1100010
1011010
1001111
1000001
0110010
0100011
0010101
0001010
0000010
0000000
0000010
0001010
0010101
0100011
0110010
1000001
1001111
1011010
1100010
1100100
1100010
1011010
1001111
1000001
0110010
0100011
0010101
0001010
0000010
0000000
0000010
0001010
0010101
0100011
0110010
```

output:



Transmitting Signal
Amplitude(volt)
Time(sec)
Discrete time signal after sampling
Amplitude(volt)
Time(sec)
DC level + Discrete time signal
Amplitude(volt)
Discrete time
Quantized signal
Amplitude(volt)
Discrete time

Experiment No — 07

Experiment Name: Write a MATLAB program to determine the frequency response $H(e^{jw})$ of a system characterized by $h(n) = (0.9)^n u(n)$. plot the magnitude & the phase response.

Theory: system response differently to inputs of different frequencies. Some systems may aplies components of certain frequencies, and alternate components of other frequencies. The way that the system output is related to the system input for different frequencies is called frequency response of the system.

The frequency response is the relationship between the system input and output in the Fourier domain

$$x(jw) \longrightarrow \boxed{H(jw)} \longrightarrow y(jw)$$

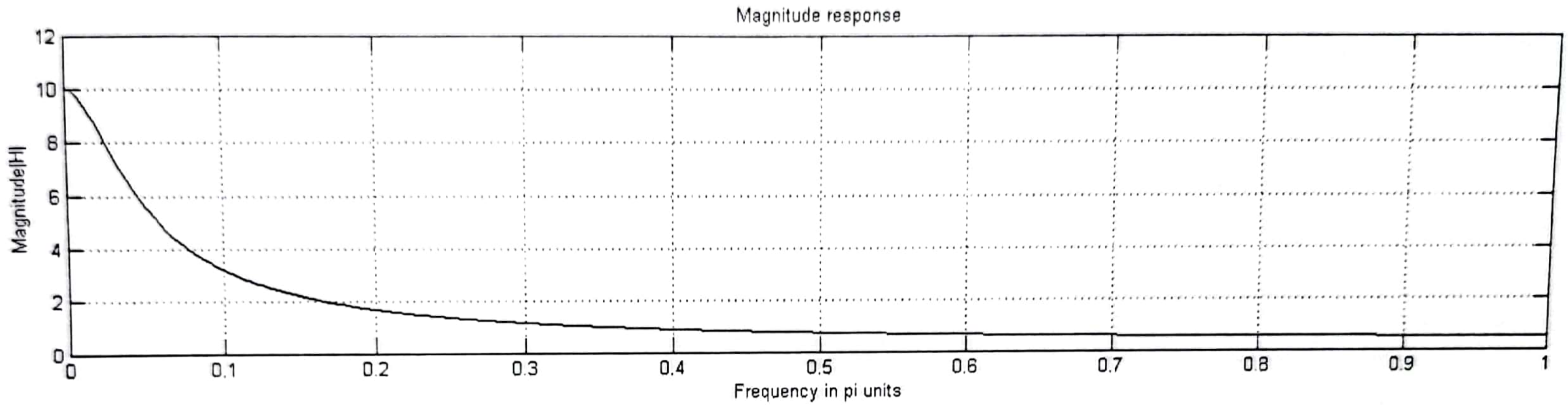
Amplitude response: For each frequency the magnitude represents the system's tendency to amplify alternate ~~the~~ input signal.

Phase response: The phase represents the system tendency to modify the phase of the input sinusoids,

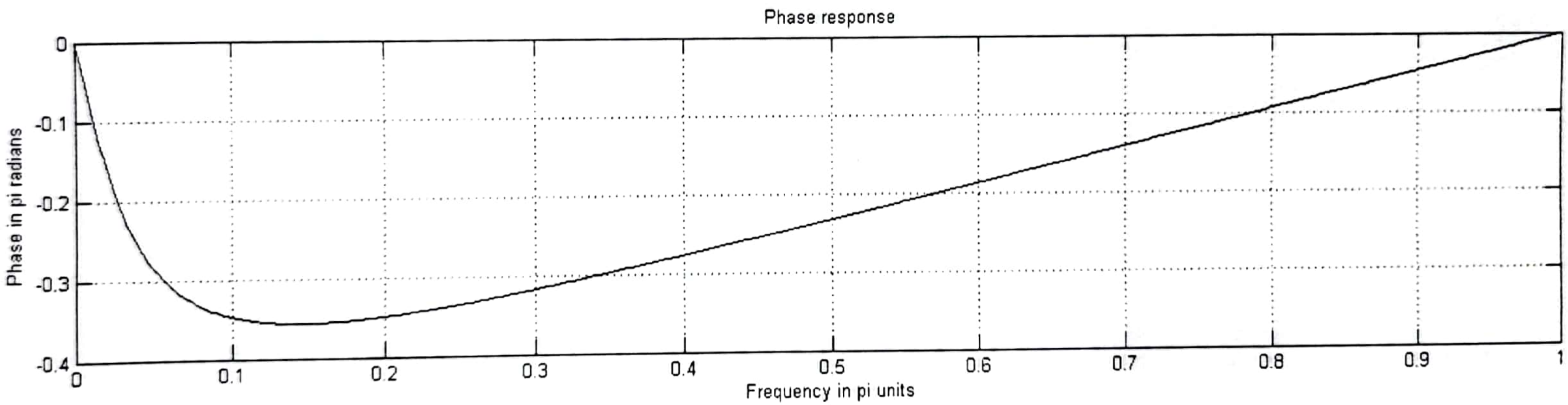
MATLAB Source Code:

```
clc;
clearall;
closeall;
w = [0:1:500]*pi/500;%[0:pi] axis divided into 501
points
H = exp(j*w)./(exp(j*w)-0.9*ones(1,501));
magH = abs(H);
angH = angle(H);
subplot(2,1,1);
plot(w/pi,magH);
gridon;
title('Magnitude response');
xlabel('Frequency in pi units');
ylabel('Magnitude|H|');
subplot(2,1,2);
plot(w/pi,angH/pi);
gridon;
title('Phase response');
xlabel('Frequency in pi units');
ylabel('Phase in pi radians');
```

output:



Magnitude response
Magnitude|H|
12
10
8
6
4
2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Frequency in pi units



Phase response
Phase in pi radians
0
-0.1
-0.2
-0.3
-0.4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Frequency in pi units

Experiment No — 08

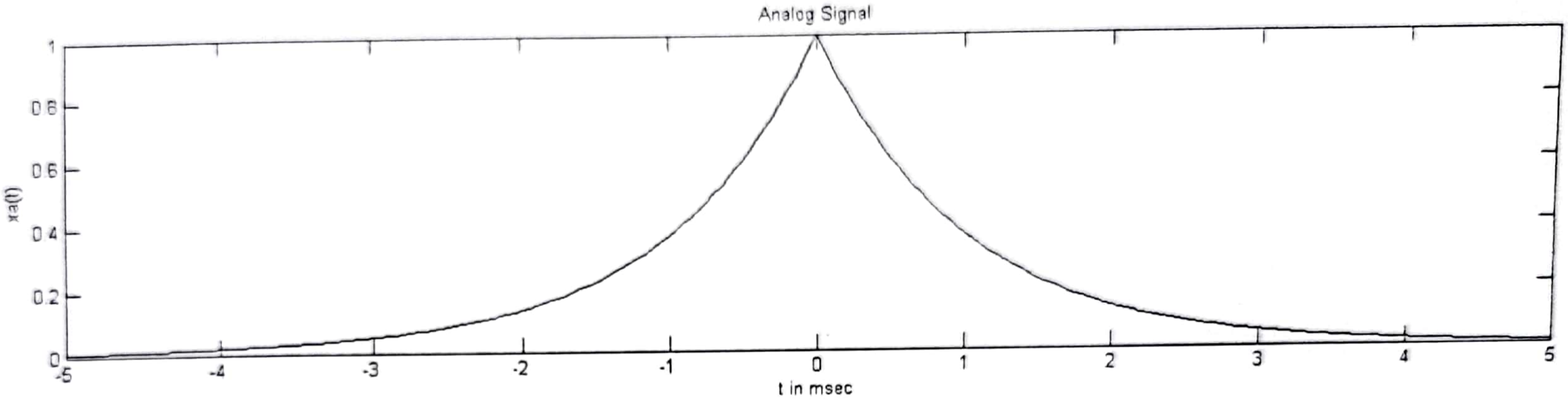
Experiment Name: Let $x(t) = e^{-1000|t|}$. Determine & plot its Fourier transform using MATLAB.

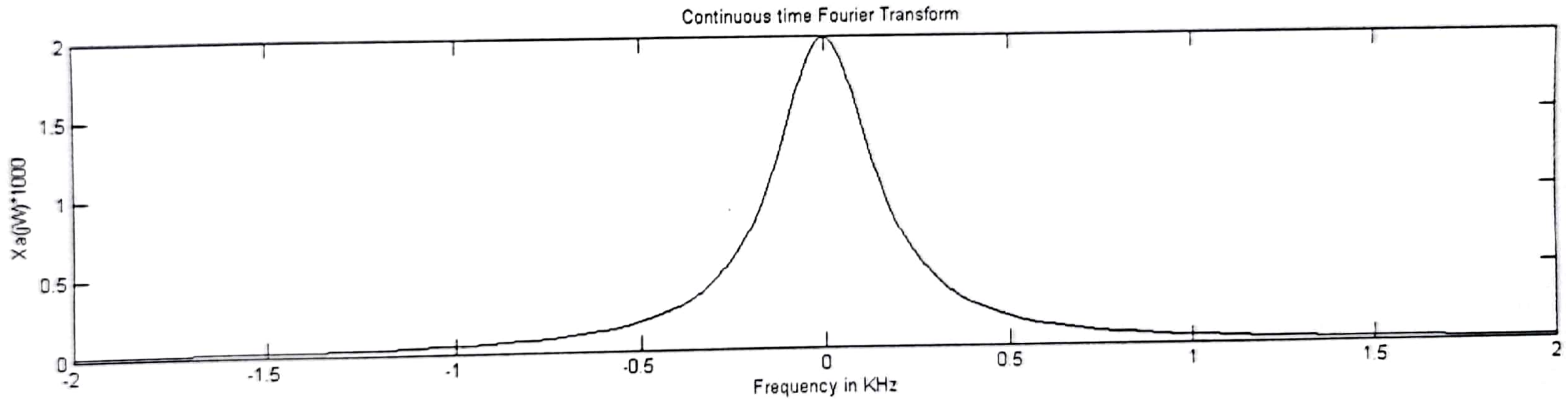
Theory:

MATLAB Source Code:

```
clc;
clearall;
closeall;
%Analog Signal
dt=0.00005;
t = -0.005:dt:.005;
xa = exp(-1000*abs(t));
subplot(2,1,1);
plot(t*1000,xa);
title('Analog Signal');
xlabel('t in msec');
ylabel('xa(t)');
%Continuous time Fourier Transform

Wmax=2*pi*2000;
K = 500;
k = 0:1:K;
W = k*Wmax/K;
Xa = xa*exp(-j*t'*W)*dt;
Xa = real(Xa);
W = [-fliplr(W),W(2:501)];
Xa = [fliplr(Xa),Xa(2:501)];
subplot(2,1,2);
plot(W/(2*pi*1000),Xa*1000);
title('Continuous time Fourier Transform');
xlabel('Frequency in KHz');
ylabel('Xa(jW)*1000');
```

Output:


Analog Signal
xa(t)
t in msec
0
0.2
0.4
0.6
0.8
1
-5
-4
-3
-2
-1
0
1
2
3
4
5


Continuous time Fourier Transform
Xa(jW)*1000
Frequency in KHz
0
0.5
1
1.5
2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Experiment No — 09

Experiment Name: Find the 4-point DFT of a sequence $x(n)$

Theory:

Theoritically $x(t)$ can take any value within $-\infty$ to $+\infty$, but since computer representation must be a finite number, quantization is needed.

DFT means discrete Fourier transform. DFT is similar to DTFT (Discrete time Fourier Transform) with some differences. Use of DFT is widespread since fast & efficient algorithms for implementing DFT on digital computer exist. This algorithms collectively are known as Fast Fourier Transform (FFT) algorithms.

Considering $x(n)$ as an N-point sequence (i.e $n=0, 1, 2, \ldots N-1$), DFT of $x(n)$ is given by

$$x(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$ where $W_N = e^{-j2\pi/N}$

Here, $k = 0, 1, 2, \ldots N-1$ also.

MATLAB Source Code:

```
%DFT sequence
clc;
clearall;
closeall;
N = input('Enter the length of sequence: ');
xn = input('Enter the sequence: ');
n = 0:N-1;
k = 0:N-1;
wN = exp((-j*2*pi)/N);
nk = n'*k;
wNnk = wN.^nk;
xk = xn*wNnk;
disp('xk-->>');
disp(xk);
mag = abs(xk);
subplot(2,1,1);
stem(k,mag);
gridon;
title('Magnitude of Fourier Transform');
xlabel('k--->>');
ylabel('Magnitude(x(k))');
phase = angle(xk);
subplot(2,1,2);
stem(k,phase);
gridon;
title('Phase of Fourier Transform');
xlabel('k--->>');
ylabel('Phase(x(k))');
```

Output:

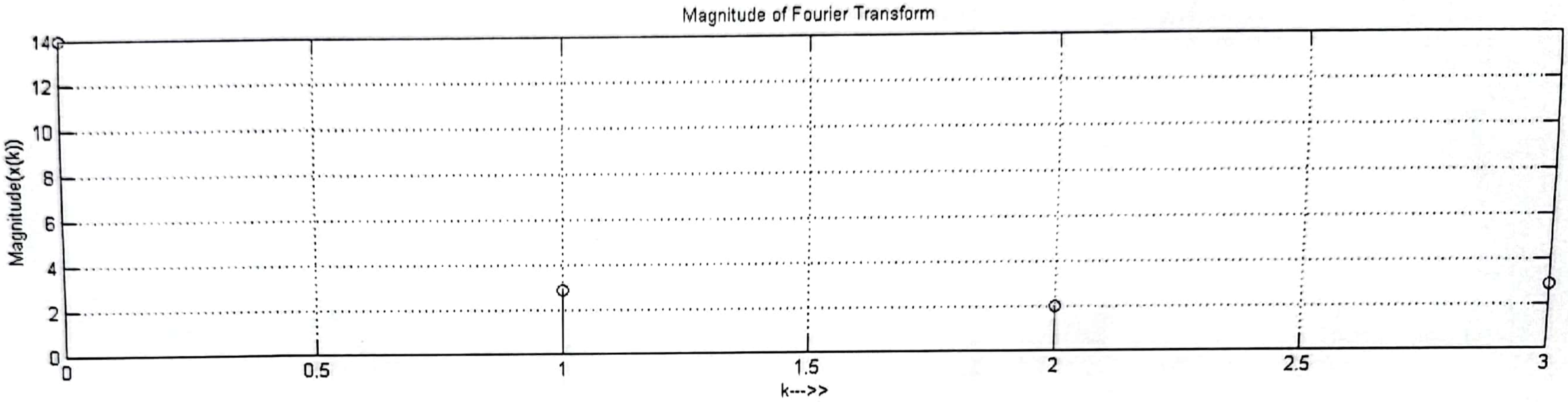
Enter the length of sequence: 4

Enter the sequence: [2,3,4,5]

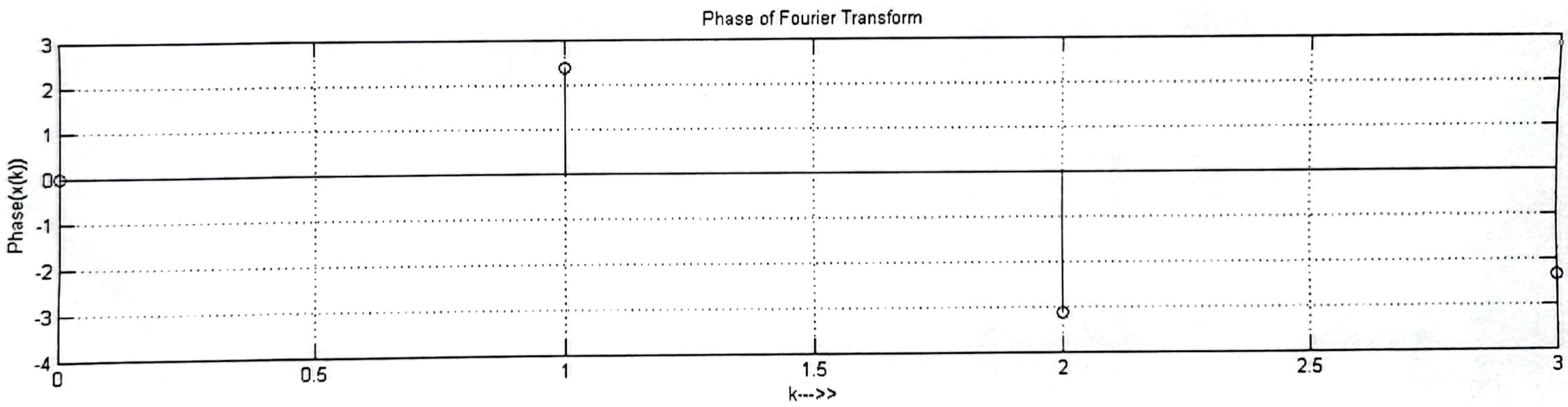
xk-->>

14.0000 -2.0000 + 2.0000i -2.0000 - 0.0000i -2.0000 - 2.0000i

output:



Magnitude of Fourier Transform
Magnitude(x(k))
14
12
10
8
6
4
2
0
0
0.5
1
1.5
2
2.5
3
k--->>



Phase of Fourier Transform
Phase(x(k))
3
2
1
0
-1
-2
-3
-4
0
0.5
1
1.5
2
2.5
3
k--->>

Experiment No — 10

Experiment Name: Find the 4-point IDFT of a sequence $x(K)$.

Theory: The inverse of DFT called IDFT (Inverse Discrete Fourier Transform) is given by

$$x(n) = \frac{1}{N} \sum_{K=0}^{N-1} x(K)\, W_N^{nK} \quad \text{where, } K = 0, 1, 2, \ldots, N-1$$

$$W_N = e^{j2\pi/N}$$

Here, $n = 0, 1, 2, 3, \ldots, N-1$ also.

~~Difference~~ Relation between DFT & IDFT:

1. Both DFT & IDFT are periodic with period N.
2. That is both DFT & IDFT replaces the finite sequence $x(n)$ with it's periodic extensions.
3. If needed zeros should be added to make N equal.

MATLAB Source Code:

```
clc;
clearall;
closeall;
N = input('Enter the length of sequence: ');
xk = input('Enter the sequence: ');
n = 0:N-1;
k = 0:N-1;
wN = exp((j*2*pi)/N);
nk = n'*k;
wNnk = wN.^nk;
xn = (xk*wNnk)/N;
disp('xn-->>');
disp(xn);
mag = abs(xn);
subplot(2,1,1);
stem(n,mag);
gridon;
title('IDFT sequence of xk');
xlabel('n--->>');
ylabel('Magnitude(x(n))');
phase = angle(xn);
subplot(2,1,2);
stem(n,phase);
gridon;
title('Phase of Fourier Transform');
xlabel('n-->>');
ylabel('Phase of x(n)');
```

Ouput:

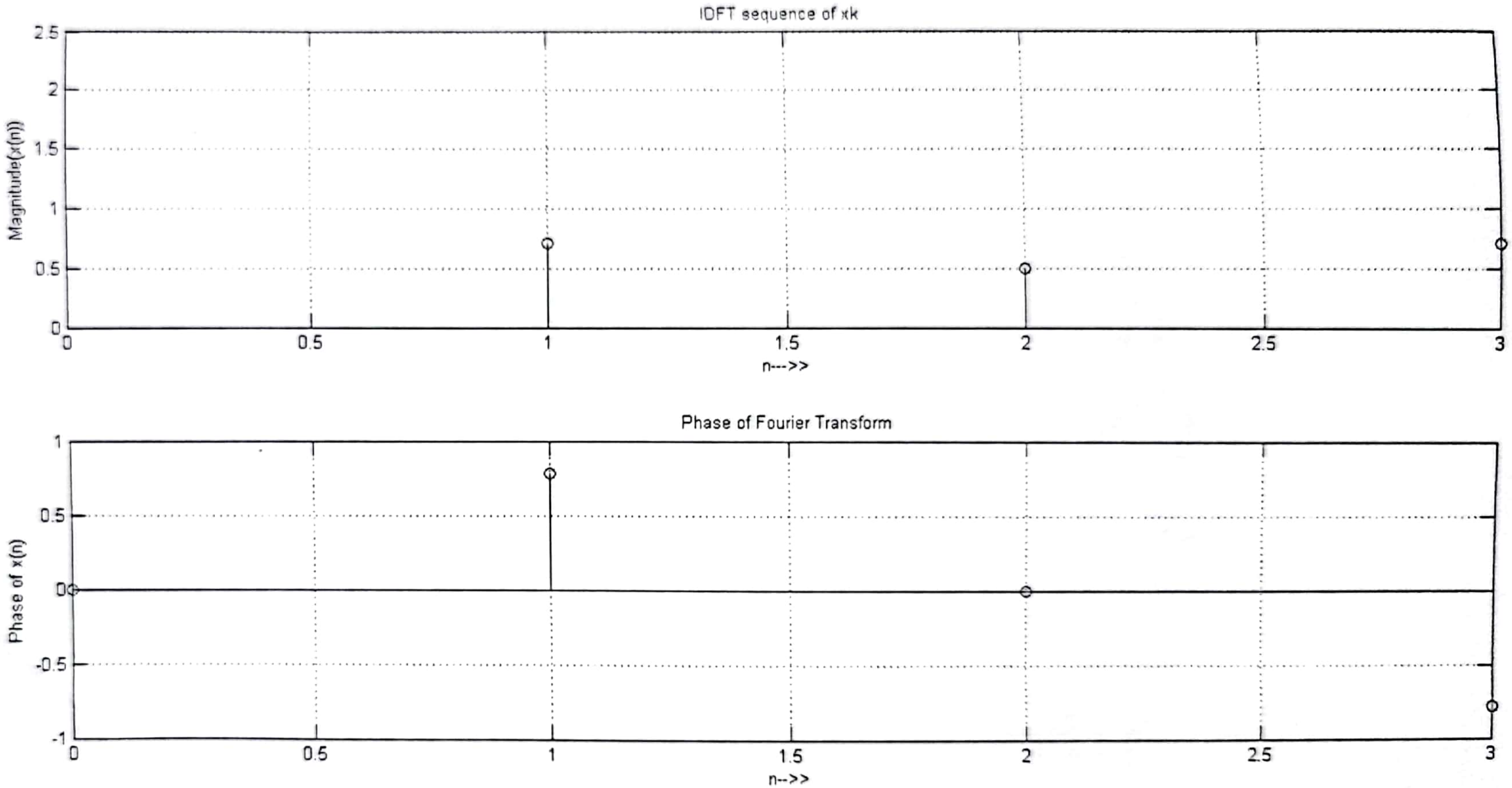
Enter the length of sequence: 4

Enter the sequence: [4 3 2 1]

xn-->>

```
2.5000    0.5000 + 0.5000i   0.5000 + 0.0000i   0.5000 - 0.5000i
```

Output:


IDFT sequence of xk
Magnitude(x(n))
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
n--->>
Phase of Fourier Transform
Phase of x(n)
1
0.5
0
-0.5
-1
0
0.5
1
1.5
2
2.5
3
n-->>