No. of the experiment: 01

Name of the experiment: Explain and implement Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT)

Theory:

Discrete Fourier transform:

In mathematics, the Discrete Fourier transform (DFT) converts a finite set of equally spaced. samples of a function into the list of coefficients of a time combination of complex sinusoids ordered by their frequencies that has there sample values. It can be said to convert the sampled function from its original domain to the following domain.

The DFT of $x(k)$ is denoted $x(i) = \text{DFT} \{x(k)\}$ and defined

$$x(i) = \sum_{K=0}^{N-1} x(K) W_N^{ik}, \quad 0 \leq i \leq N$$

Inverse Discrete Fourier Transform (IDFT):

The continuous time fourier transform has an inverse whose is always identical to the original signal. The inverse of DFT, which is denoted

$x(K) = \text{IDFT}\{x(i)\}$ is computed as follows:

$$x(K) = \frac{1}{N} \sum_{i=0}^{N-1} x(i) \, W_N^{-Ki} \quad , \quad 0 \le K < N$$

Mathematical equation:

DFT,

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

$$0 \le n \le N-1$$

IDFT,

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

$$0 \le K \le N-1$$

CamScanner

Matlab code:

```
clc;
clear all;
close 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*pi*k*n)/N);
    end
end
disp(y);
```

```
subplot(3,1,2);
stem(y);
xlabel('k');
ylabel('x(k)');
title('DFT values');
grid on;

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

input:

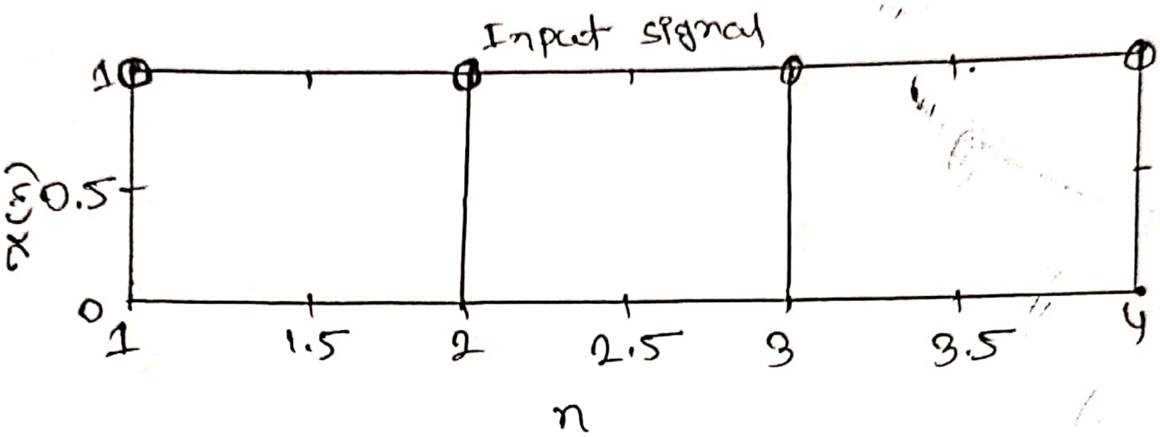
Enter the sequence x[n] = [1 1 1 1]

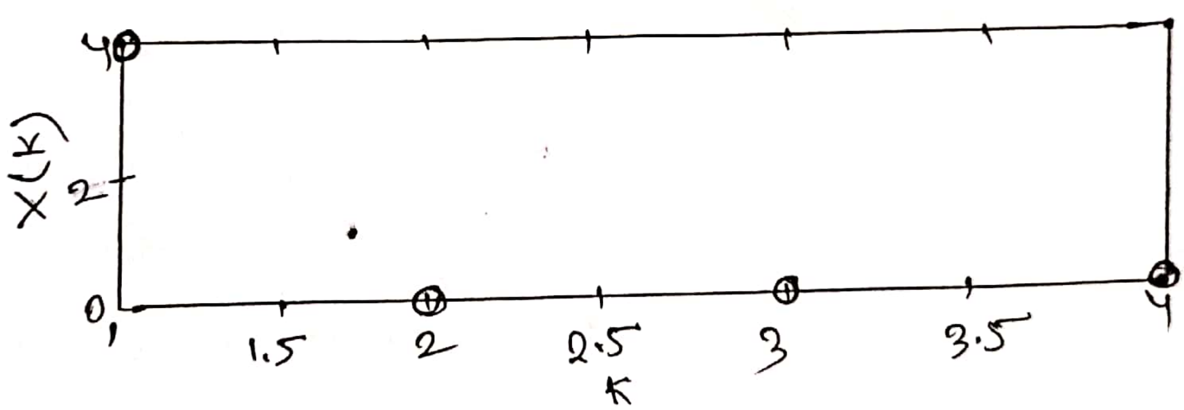
enter n: 4

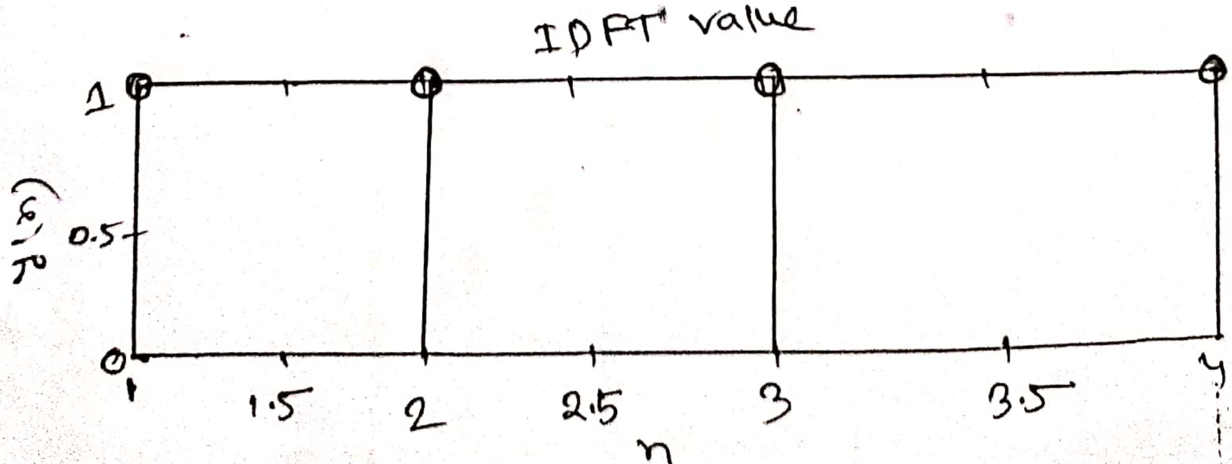
Output:

4.000+0.000i -0.0000-0.0000i 0.0000-0.0000i 0.0000-0.0000i

1.0000-0.0000i 1.0000-0.0000i 1.0000+0.0000i 1.0000+0.0000i







No. of the experiment: 02

Name of the experiment: Let $x(n) = \{1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1\}$. Determine and plot the following sequence. $x_1[n] = 2x(n-5) - 3x(n+4)$

Theory:

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

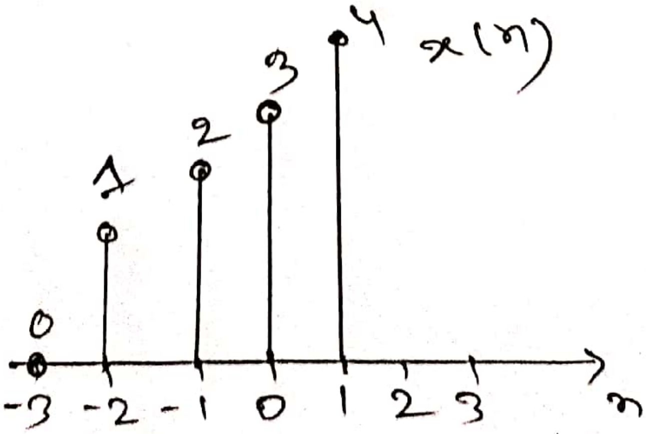
Shifting: Let us consider a discrete time signal $x(n)$. Let $y(n)$ is a signal denote to obtain by shifting the signal $x(n)$ by $(n-n_0)$ that is,

$$y(n) = x(n-n_0)$$

Example,

suppose, $x(n) = \{0, 1, 2, \underset{\uparrow}{3}, 4\}$

determine $y(n) = x(n-1)$



Here, $n = -3+1 = -2$ Since $y(n) = x(n+1)$ that is, right shifting by 1

$x(-2) = 0$

For $n = -2+1 = -1$

$x(-1) = 1$

For $n = -1+1 = 0$

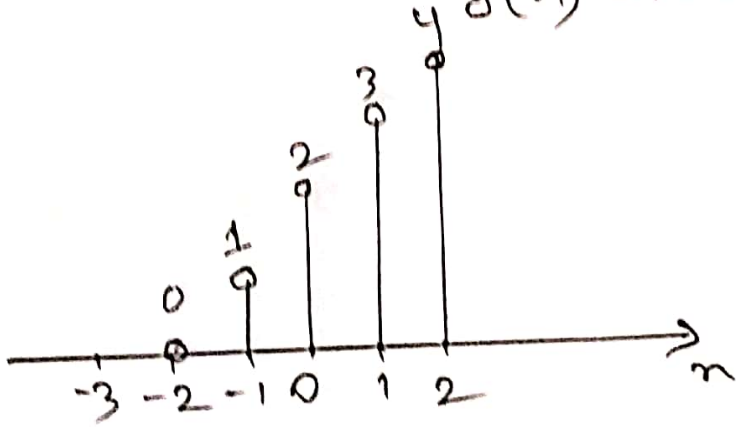
$x(0) = 2$

For $n = 0+1 = 1$

$x(1) = 3$

For $n = 1+1 = 2$

$y(n) = x(n-1)$

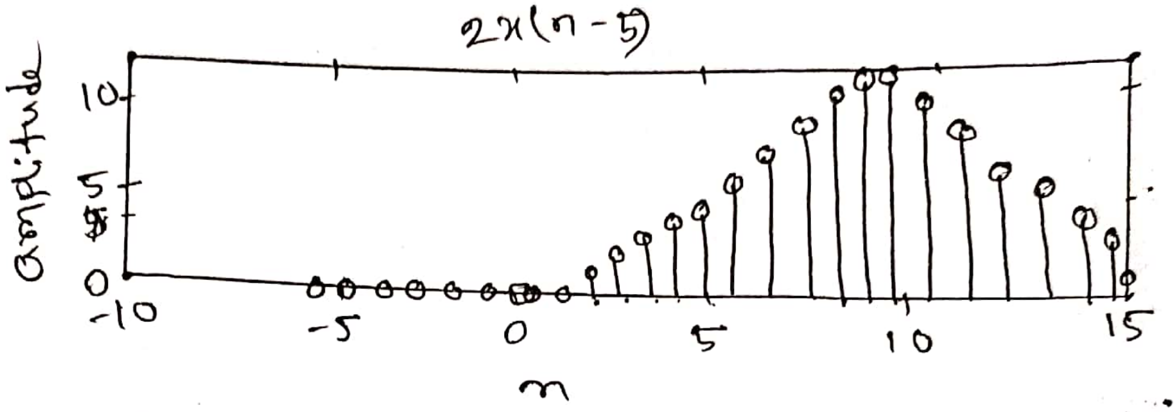


Matlab code:

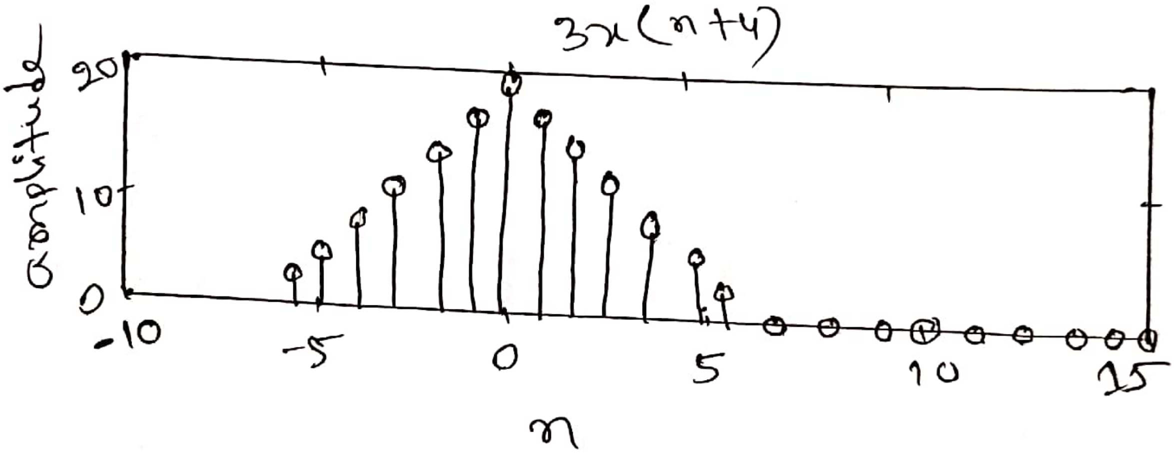
```
clc;
clear all;
close all;
n= -2:10;
x=[1,2,3,4,5,6,7,6,5,4,3,2,1];
n1=3:15;
n2=-6:6;
m=min(min(n1),min(n2)):max(max(n1),max(n2));
y1=[];
temp=1;
for i=1:length(m)
    if (m(i)<min(n1) || m(i)>max(n2))
        y1=[y1 0]
    else
        y1=[y1 x(temp)];
        temp=temp+1;
    end
end
y2=[];
temp=1;
for i=1:length(m)
    if (m(i)<min(n2) || m(i)<max(n2))
        y2=[y2 0];
    else
        y2=[y2 x(temp)]
    end
end
```

```
subplot(3,1,1);
stem(n, 2.*y1);
grid on;
title('2x(n-5) signal');
xlabel('n');
ylabel('Amplitude');

subplot(3,1,2);
stem(n, 3.*y2);
grid on;
title('3x(n+4)');
xlabel('n');
ylabel('Amplitude');
y = (2.*y1)-(3*(y2));
subplot(3,1,3);
stem(n,y);
grid on;
```

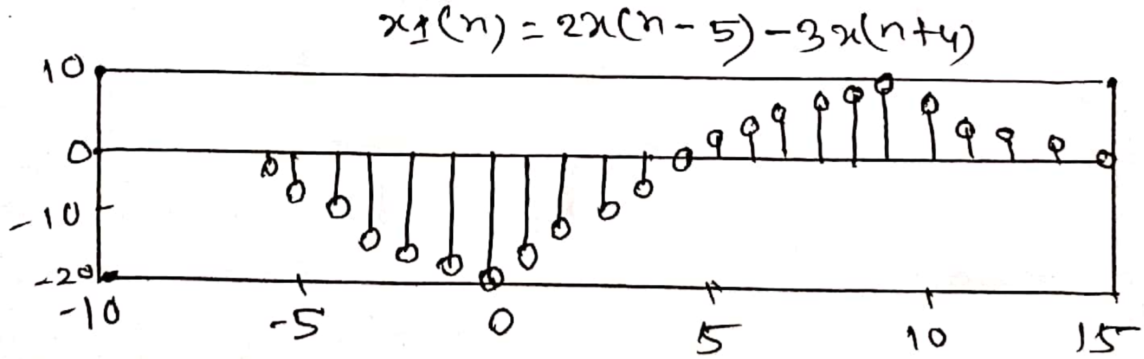
Output:



2x(n-5)
amplitude
10
5
0
-10
-5
0
5
10
15
n



3x(n+4)
amplitude
20
10
0
-10
-5
0
5
10
15
n



x1(n) = 2x(n-5)-3x(n+4)
10
0
-10
-20
-10
-5
0
5
10
15

No. of the experiment: 03

Name of the experiment: Write a program to perform following operations-

(i) Sampling (ii) Quantization (iii) Coding

Theory:

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

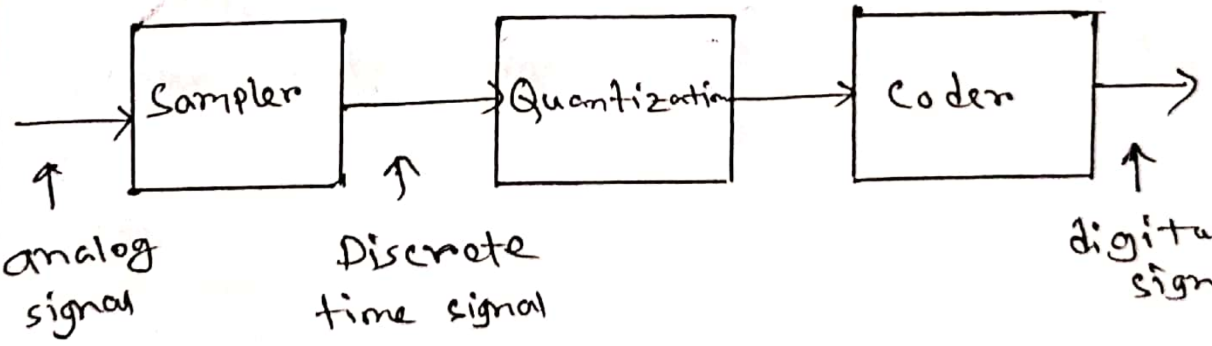


Figure: Basic parts of analog to digital converter

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

the difference between an input value and its quantized value (such as round off error) is referred to as quantization error. A device or algorithmic function that perform quantization is called a quantizer.

Coding: As system of system signal used to represent letter or number in transmitting message. The instruction in a computer program.

Matlab code:

```
clc;
clear all;
close all;
A=input('Enter amplitude of transmitting signal:');
f=50;
T=1/f;
t=0:T/100:2*T;
n=1:40;
y=A*sin(2*pi*f*t);
subplot(4,1,1);
plot(t,y);
title('continuous time message/transmitting signal');
xlabel('Time');
ylabel('Amplitude');
% sampling signal
y1=A*sin(2*pi*f*(0.001)*n);
```

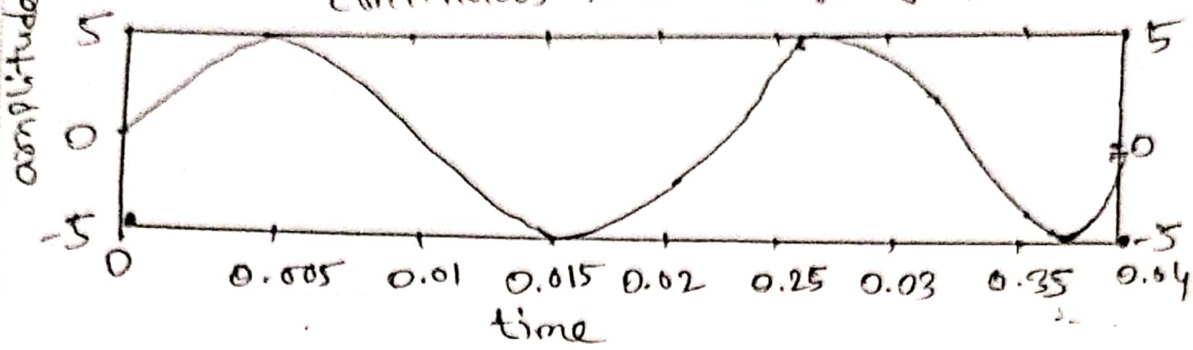
```
subplot(4,1,2);
stem(n,y1);
title('Discrete time signal after sampling");
xlabel('n---');
ylabel('amplitude');

y2=A+y1;
subplot(4,1,3);
stem(n,y2);
title('DC level + Discrete time signal');
xlabel('n---');
ylabel('amplitude');
% quantization
y3=round(y2);
subplot(4,1,4);
stem(n,y3);
title('Quantization(Quantized signal');
xlabel('n---');
ylabel('amplitude');

% coding
y4=dec2bin(y3);
disp('binary information');
disp(y4);
```

input:

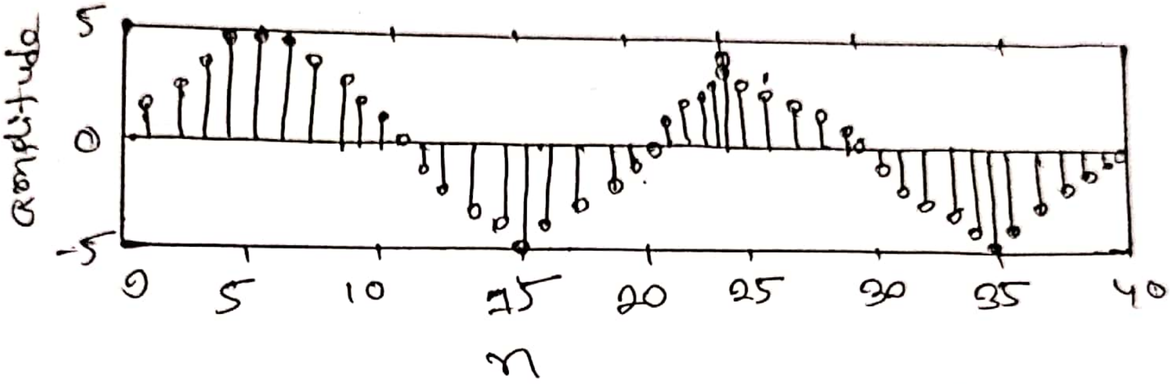
Enter amplitude of transmitting signal: 5

output:

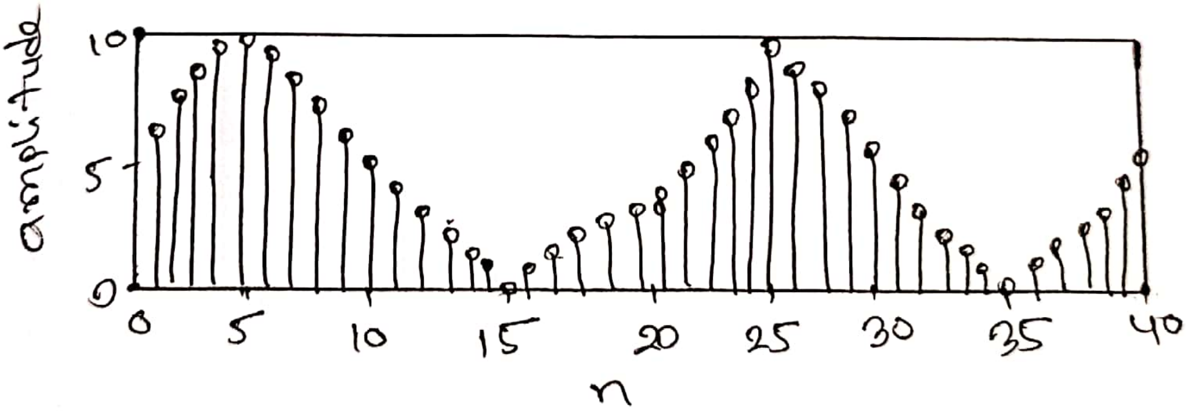
continuous time message signal



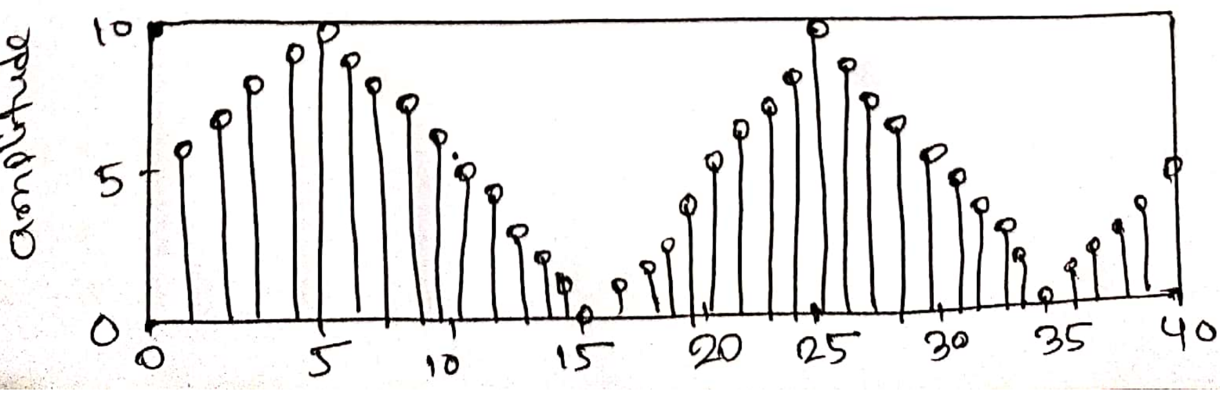
Discrete time after sampling



Dc level + Discrete Time signal



Quantization (Quantized signal)



binary information:

0111
1000
1001
1010
1010
1010
1001
1000
0111
0101
0011
0010
0001
0000
0000
0000
0001
0010
0011
0101
0111
1000
1001
1010
1010
1010

1001
1001
0111
0101
0011
0010
0001
0000
0000
0000
0001
0010
0011
0101

No. of the experiment: 04

Name of the experiment: Determine and plot the following sequence, $x(n) = 2\delta(n+2) - \delta(n-4)$, $-5 \leq n \leq 5$

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 sample: It's defined in two ways:
(i) Discrete time (ii) continuous time

(a) Discrete time unit impulse: It is defined as,

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

(b) continuous time unit impulse: It's defined by,

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

The unit sample sequence is a signal that is everywhere; except at $n=0$ where $t=0$ its value is unity.

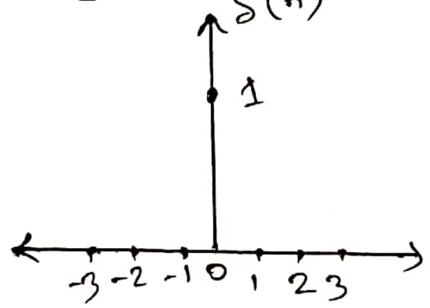


Figure: Graphical representation of the unit sample signal

Ex: Suppose,

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

We know that,

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

For $n = 2$

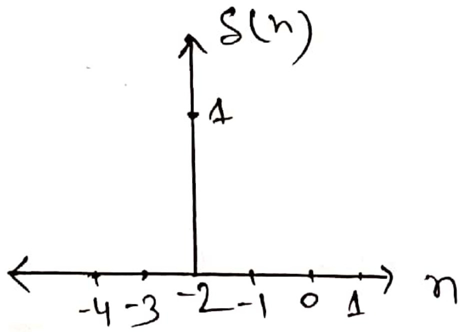
$\delta(-2+2) = \delta(0) = 1$

For $n = -1$, $\delta(-1+2) = \delta(1) = 0$

For $n = 0$, $\delta(0+2) = \delta(2) = 0$

For $n = 2$, $\delta(2+2) = \delta(4) = 0$

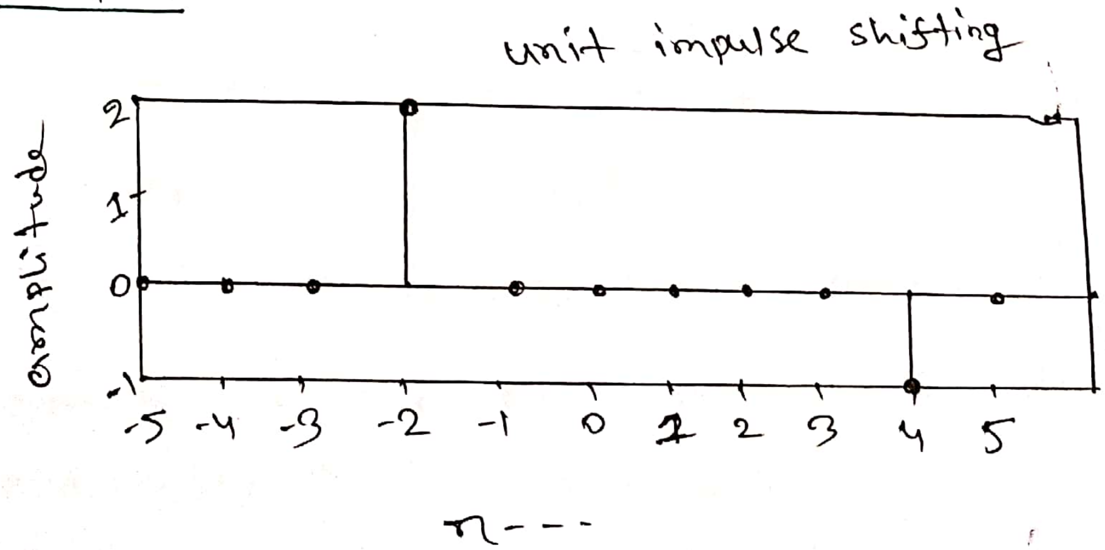
Graphical representation of $x(n)$:



Matlab code:

```
clc;
clear all;
close all;
n = -5:5;
n1 = -2:5;
n2 = -5:4;
y1 =(n==-2);
y2 = (n==4);
y3 = 2*y1 - y2;
subplot(2,1,1);
stem(n,y3);
title('Unit impulse shifting');
xlabel ('n---');
ylabel ('amplitude');
```

output:



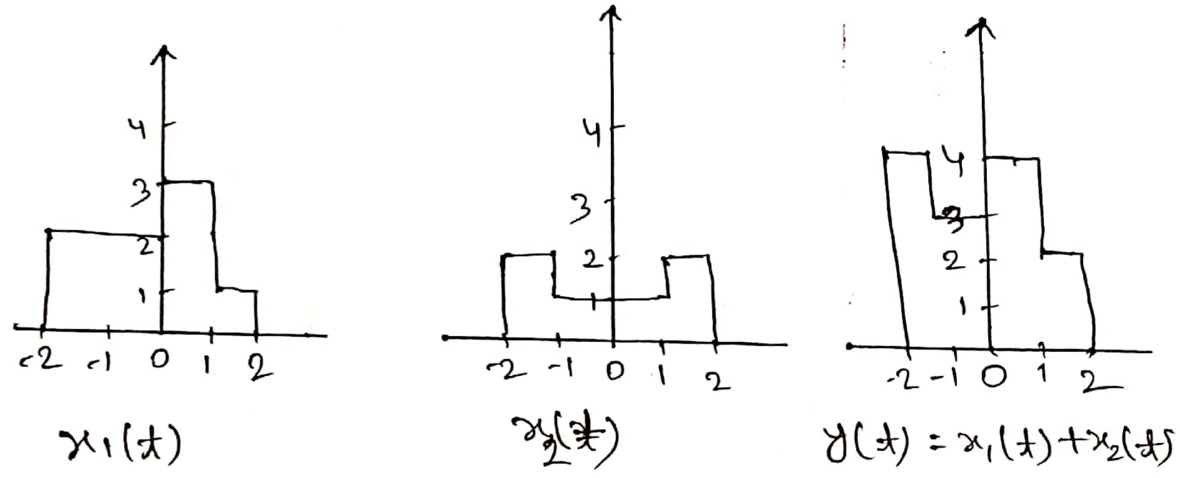
No. of the experiment: 05

Name of the experiment: Plot following signal operations using user defined function (i) Addition (ii) Folding

Theory:

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

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



(a) Addition of signals

consider a pair of discrete time signal $x_1(n)$ and $x_2(n)$ in a signal $y(n)$, the period of signal $y(n)$ is unchanged.

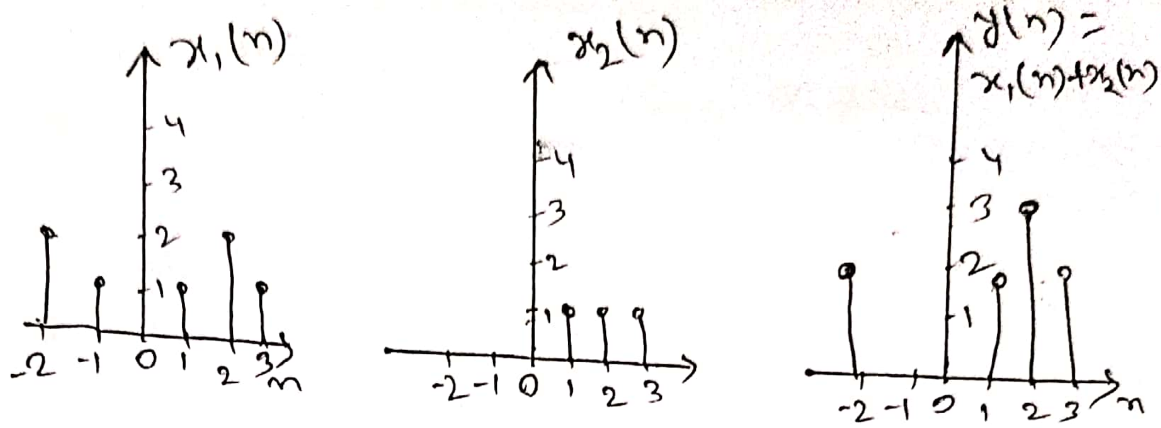


Figure: Addition of signals

<u>Folding of signal:</u> Consider a discrete time $x(n)$ Folding means converting the position in positive to negative and negative to positive. The period of $x(n)$ is unchanged.

So $x(n) = x(-n)$

Example:

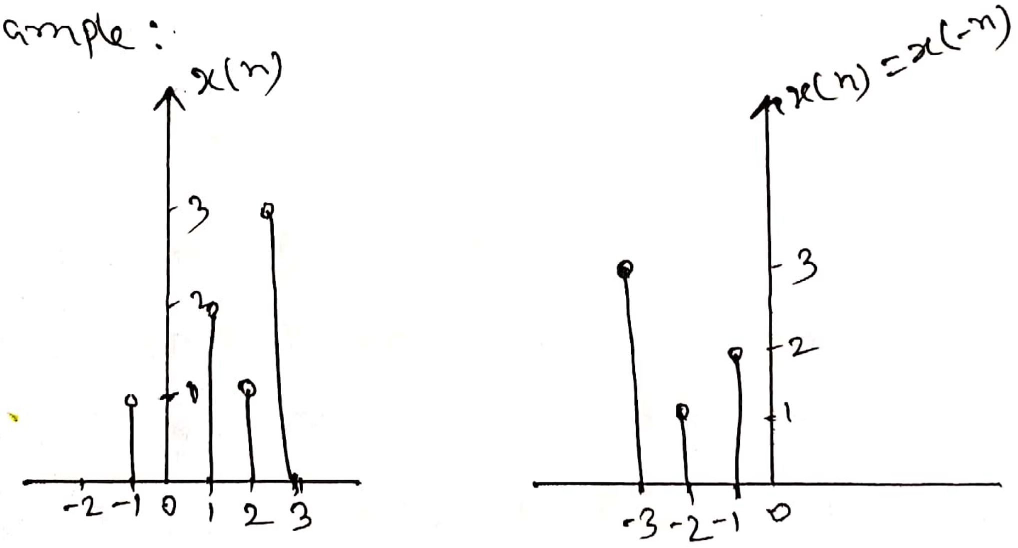


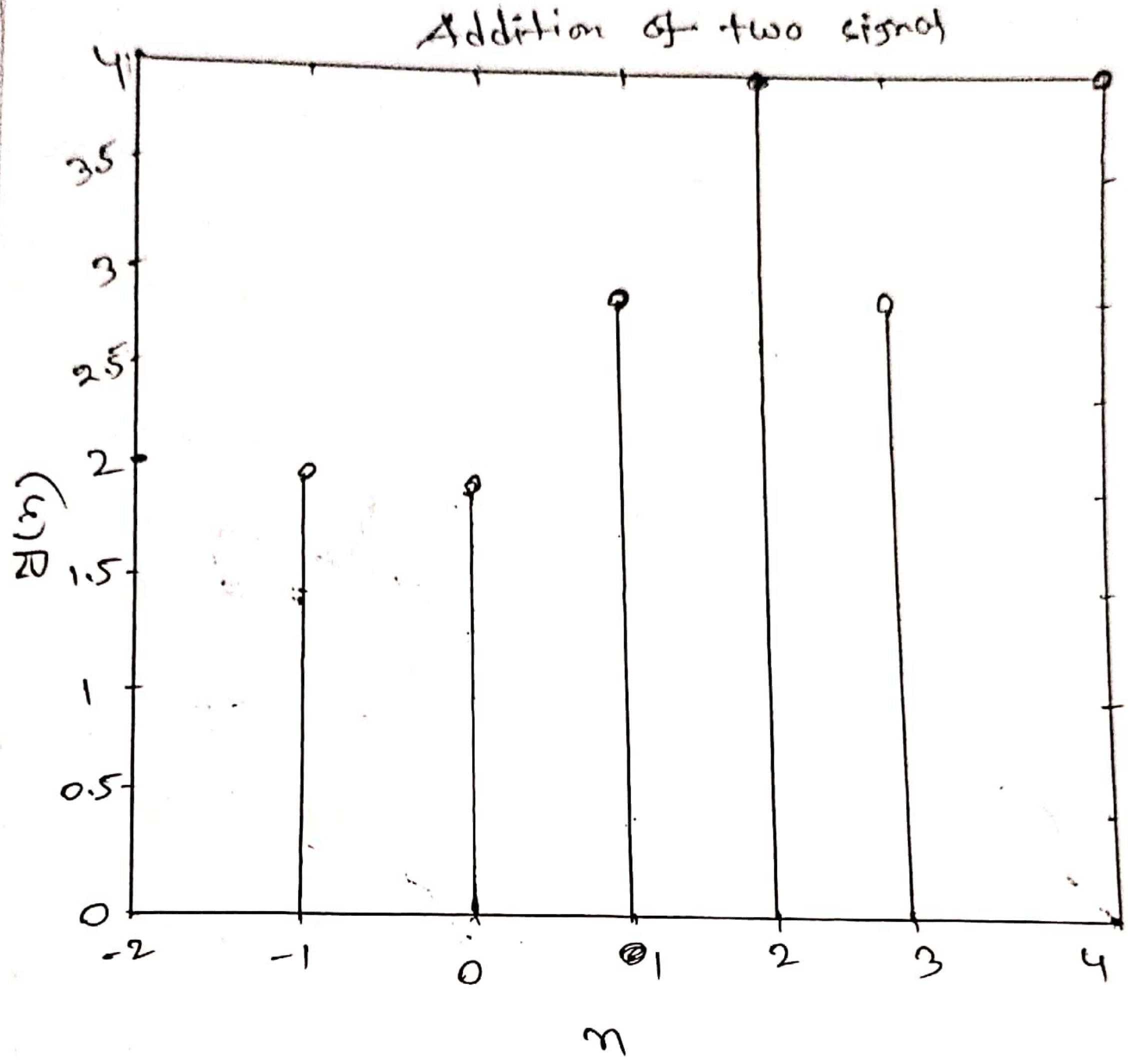
Fig: Folding of signal

Matlab code:

Adding of signals by using Matlab

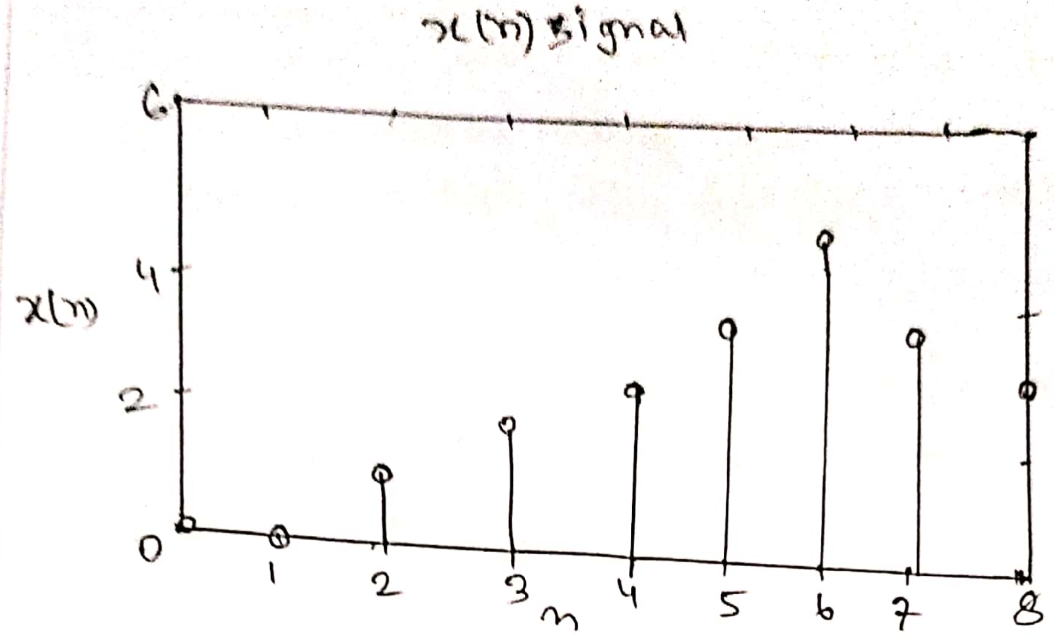
```
clc;
clear all;
close all;
%x1=[4, 2 6 1 4 9];
%x2=[4 3 5 11 1 0];
%n1=-2:3;
%n2=-1:3;
%[y,n]=sigadd(x1,n1,x2,n2);
%stem(x,y);
n1=0:4;
x1=[0 1 2 3 4];
x2=[2 2 2 2 2];
n=min(min(n1),min(n2)):max(max(n1),max(n2));
y1=zeros(1,length(n)); y2=y1;
y1=(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2=(find((n>=min(n2))&(n<=max(n2))==1))x2;
y=y1+y2;
stem(x,y)
```

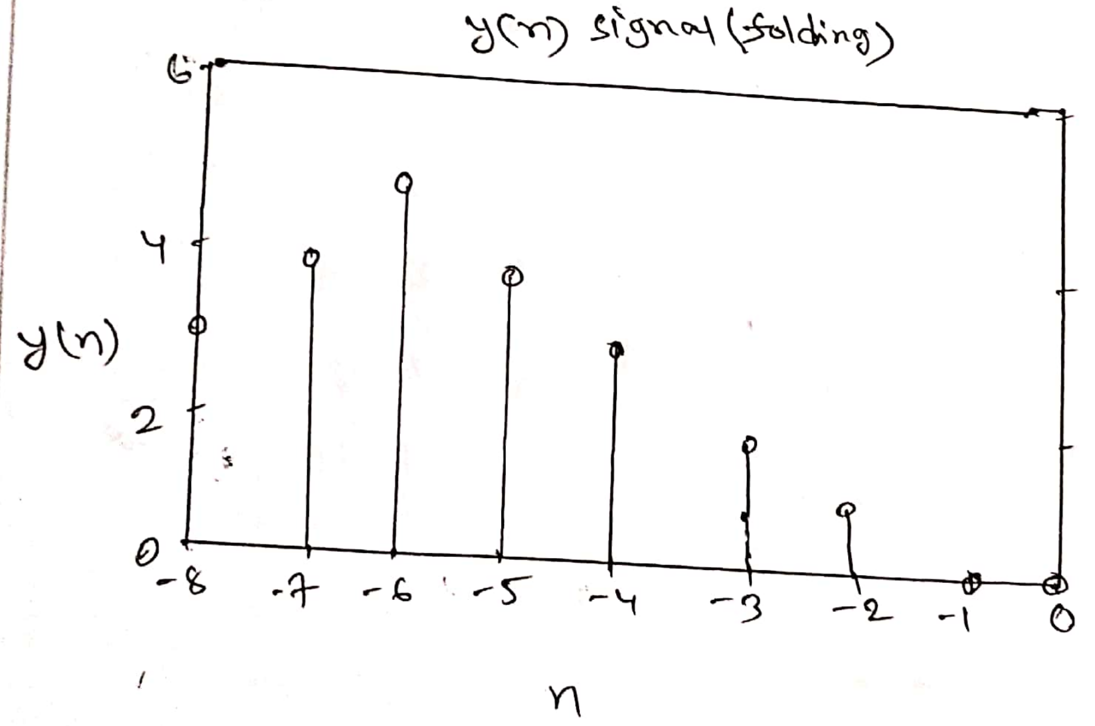
Output:



Folding of signal using Matlab:

```
clc;
clear all;
close all;
n = 0:8;
x = [0 0 1 2 3 4 5 4 3];
subplot(2,1,1);
stem(n,x);
title('x(n)signal');
xlabel('n');
ylabel('x(n)');
m = - fliplr(n);
y = fliplr(x);
subplot(2,1,2);
stem(m,y);
title('y(n) = x(-n)');
xlabel('n');
ylabel('y(n)');
```


x(n) signal
x(n)
n
0
2
4
6
1
2
3
4
5
6
7
8


y(n) signal (folding)
y(n)
n
0
2
4
6
-8
-7
-6
-5
-4
-3
-2
-1
0

No. of the experiment: 06

Name of the experiment: Plot the following signal operations using user defined function (i) signal multiplication (ii) signal shifting

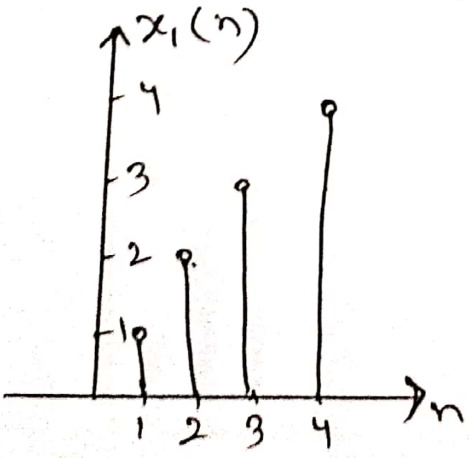
Theory:

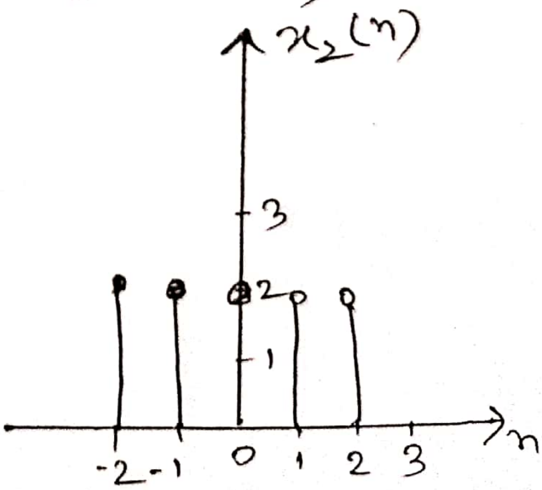
Signal: A signal is defined as a function of one or more variables which conveys information. It is also a physical quantity with time or any other independent variable.

Example- ECG signal, speech signal

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

Consider a pair of discrete time signal $x_1(n)$ and $x_2(n)$. Multiplication of these two discrete time signals $x_1(n)$ and $x_2(n)$ and resulted output signal is, $y(n) = x_1(n) \cdot x_2(n)$





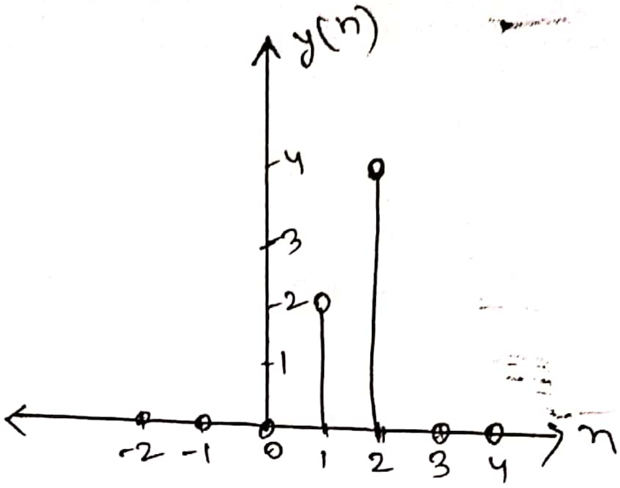
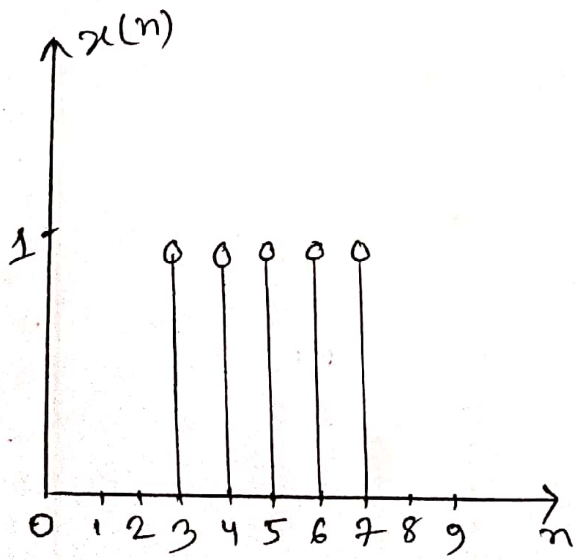


Fig: Multiplication of signal

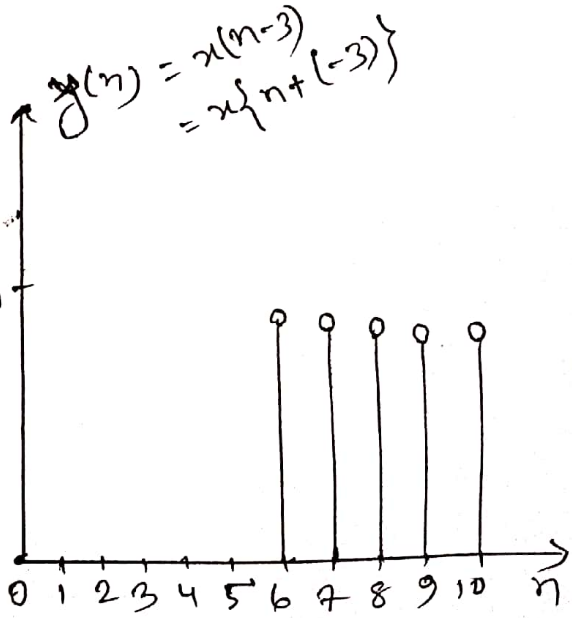
<u>Shifting of signal</u>: Let us consider a discrete time signal $x(n)$. Let $y(n)$ is a signal denote to obtain by shifting the signal $x(n)$ by $(n-n_0)$ that is $y(n) = x(n-n_0)$

Shift $h(-k)$ by n_0 to the right (left) if n_0 is positive (negative) to obtain $h(n_0-k)$

Example-



Fig(a): Original positive signal $x(n)$

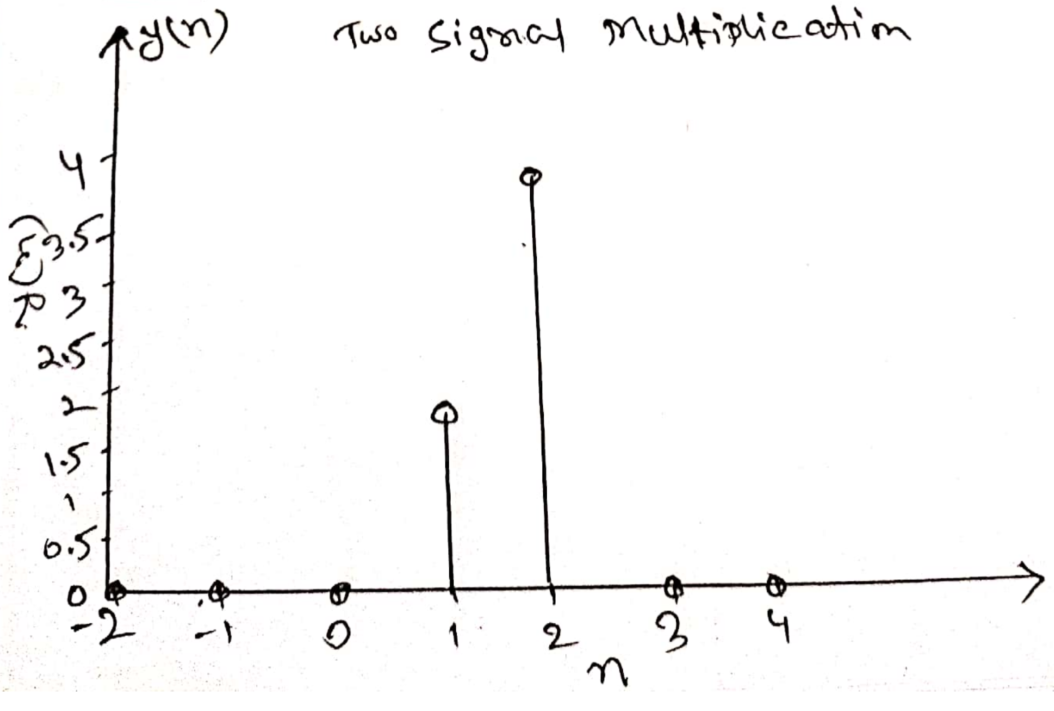


Fig(b): Right shifted signal $x(n-3)$

Signal multiplication using matlab:

```
clc;
clear all;
close all;
% function (y,n) = sigmult(x1, x2, n1, n2);
n1 = 0:4;
n2 = -2:2;
x1 = [0 1 2 3 4];
x2 = [2 2 2 2 2];
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;
stem(n,y);
```

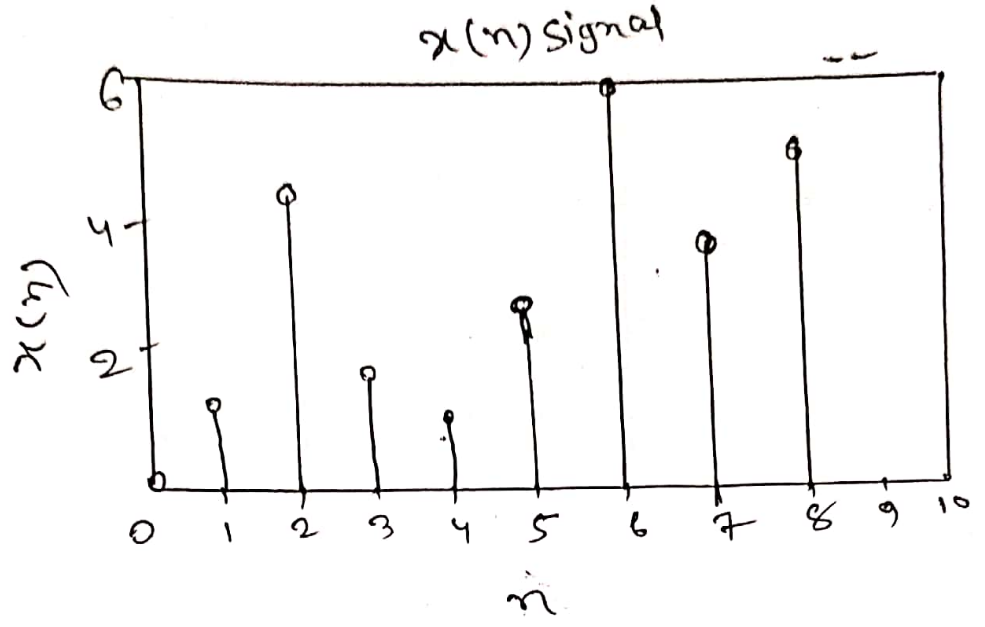
Output:



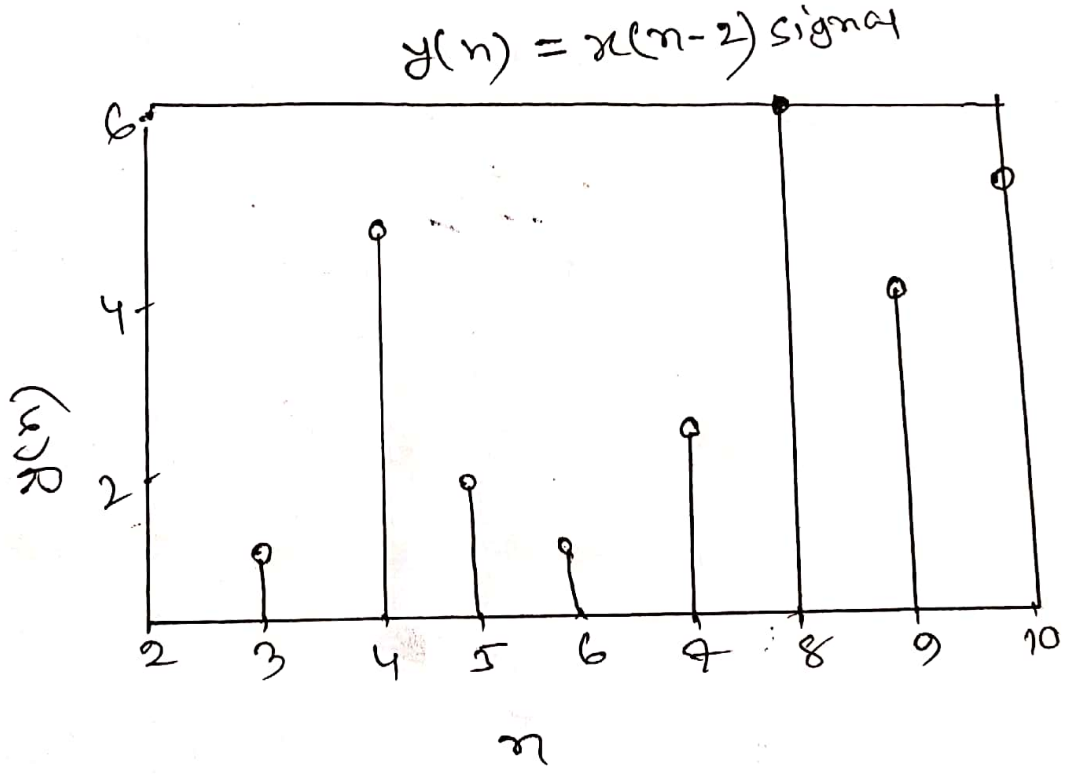
Signal shifting using Matlab code:

```
clc;
clear all;
close all;
n = 0: 8;
x = [0 1 5 2 1 3 6 4 5];
%subplot(n,x); subplot(2,1,1);
stem(n,x);
title('x(n) signal');
xlabel('n');
ylabel('x(n)');
m=n+2;
y=x;
subplot(2,1,2);
stem(m,y);
title('y(n) = x(n-2) signal');
xlabel('n');
ylabel('y(n)');
```

Output:



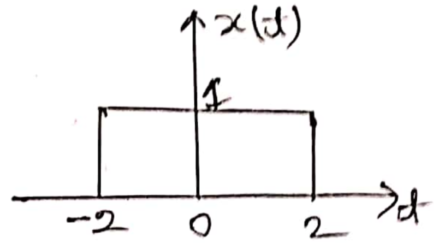
x(n) signal
6
4
2
x(n)
0
1
2
3
4
5
6
7
8
9
10
n



y(n) = x(n-2) signal
6
4
2
y(n)
2
3
4
5
6
7
8
9
10
n

No. of the experiment: 07

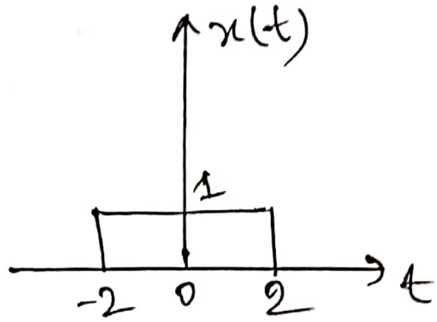
Name of the experiment: using MATLAB to plot the Fourier transform of time function the aperiodic pulse shown below:



Theory:

Fourier Transform: A function derived from a given function and representing it by a series of sinusoid function.

The aperiodic pulse shown below-



has a fourier transform:

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

This can be found using the table of Fourier transform. We can use Matlab to plot this transform. Matlab has built in sinc function. However, the definition of the Matlab sinc function is slightly different than the one used in class and on the Fourier transform table.

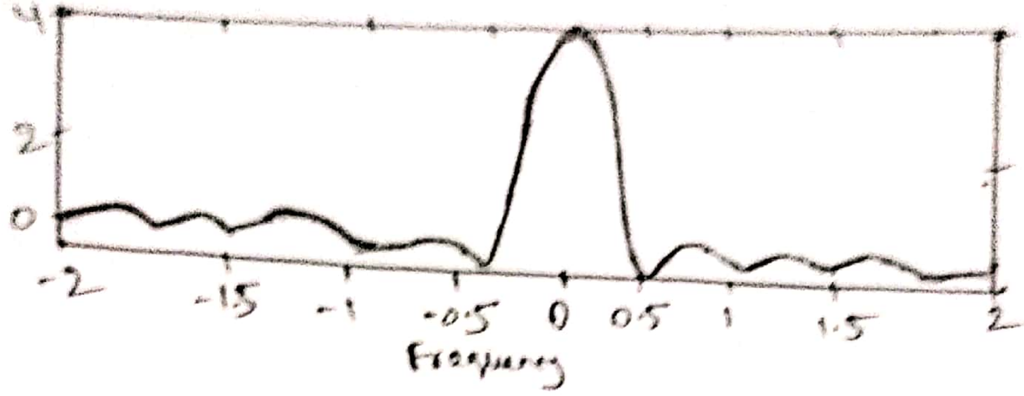
$$\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

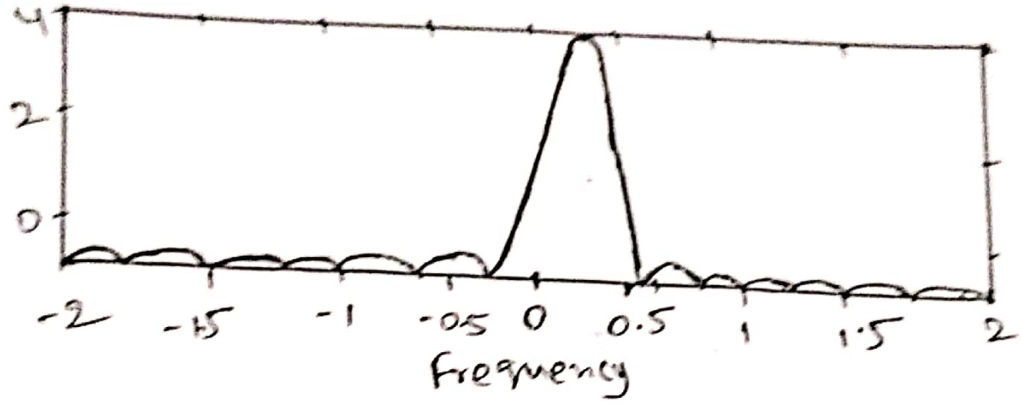
Thus is matlab we write the transform X, using sinc(4f), since the π factor is built into the function.

Matlab code:

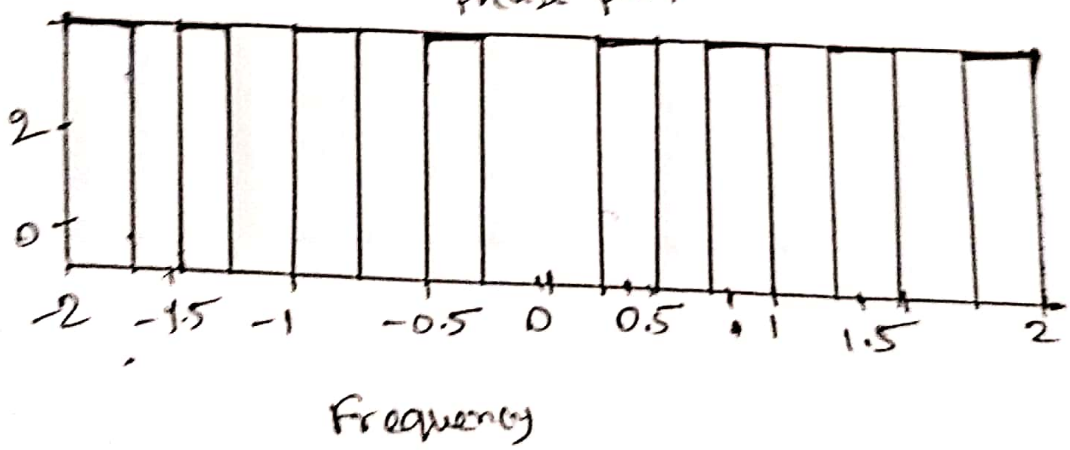
```
clc;
clear all;
close all;
f =-2:.01:2;
x = 4* sin(4*f);
subplot(3,1,1);
plot(f,real(x));
title('Real part');
xlabel('Frequency');
subplot(3,1,2);
plot (f, abs(x));
title('Magnitude part');
xlabel('Frequency');
subplot (3,1,3);
plot(3,1,3);
title('Angle part');
xlabel('Frequency');
```

Real part





Phase part



No of the experiment: 09

Name of the experiment: Explain and generate sinusoidal wave with different frequency using MATLAB.

Theory: A sine wave or sinusoid is a mathematical curve that describes a smooth repititive oscillation. It is named after the function sine of it is the graph. It occurs often in pure and applied mathematical as well as physics, engineering, signal processing and many other field.

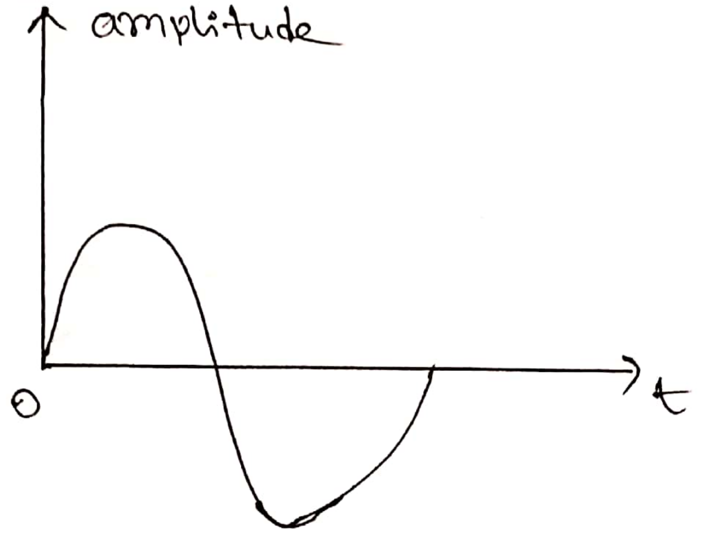


Fig: A sine wave curve

The equation of the sinusoidal wave is,

$$z = x + iy = r(\cos\varphi + i\sin\varphi)$$

$$x = r\cos\varphi$$

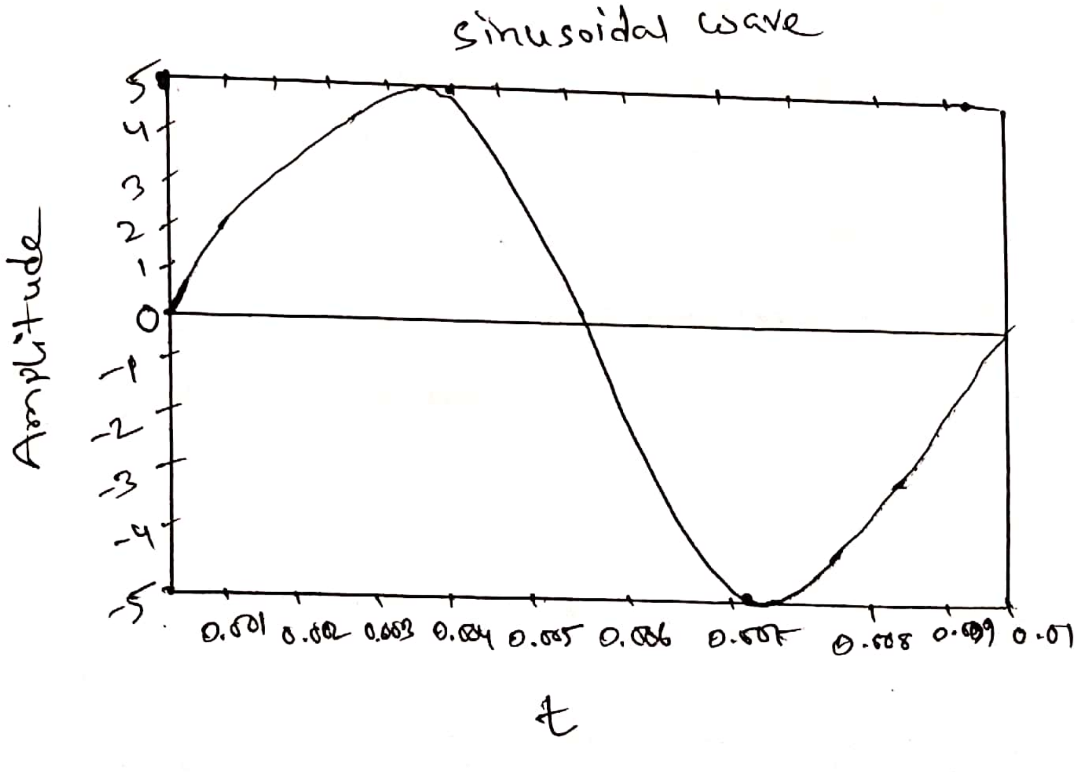
$$y = r\sin\varphi$$

$$\varphi = \omega t$$

Sinusoidal wave with frequency Matlab code:

```
clc;
clear all;
close all;
A = 5;
f = 100;
T = 1/f;
t = 0 : T/100 : T;
y = A*sin(2*pi*f*t);
plot(t,y);
xlabel('t');
ylabel('Amplitude');
title('sinusoidal wave');
```

Output:



No. of the experiment: 10

Name of the experiment: Explain and implementation of following Elementary discrete signal using Matlab.
(1) The unit sample sequence (2) The unit step signal (3) The ramp signal

Theory:

The unit sample sequence: The unit sample sequence is denoted as $\delta(n)$ and is defined as,

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

In words the unit sample sequence is a signal that's zero everywhere, except at, n=0, where it's value is unity. The signal is sometimes reffered to a unit impulse. In constrast to signal $\delta(t)$. Which is also called a unit impuls and is defined to be zero everywhere except at t=0 and has unit area the unit sample sequence is much less mathematically completed the graphical representation is shown in fig-

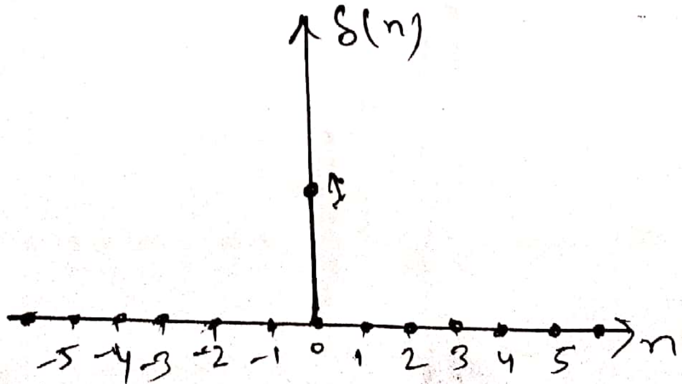


Fig: Graphical representation of the unit sample sig

The unit step signal: The unit step signal is denoted as $u(n)$ and defined as,

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

The graphical representation of the unit step signal is given below–

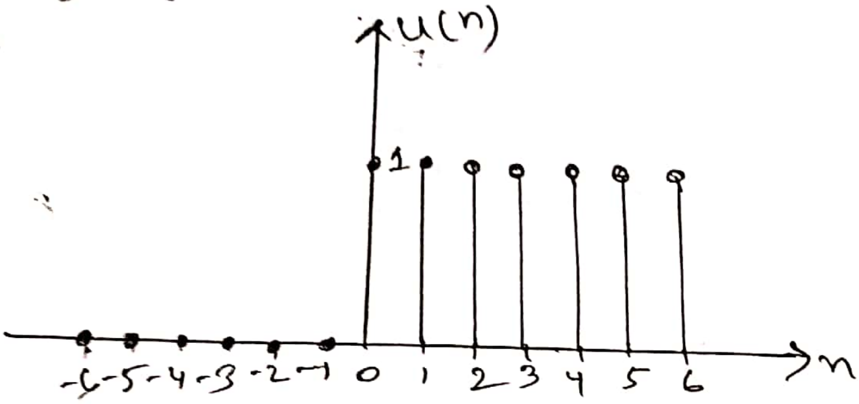


Fig: Graphical representation of the unit step signal

The unit ramp signal: The unit ramp signal is denoted as $ur(n)$ and is defined as,

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

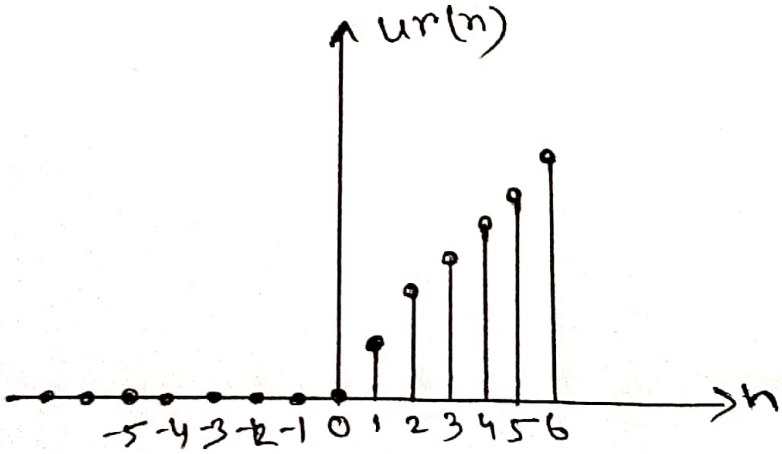


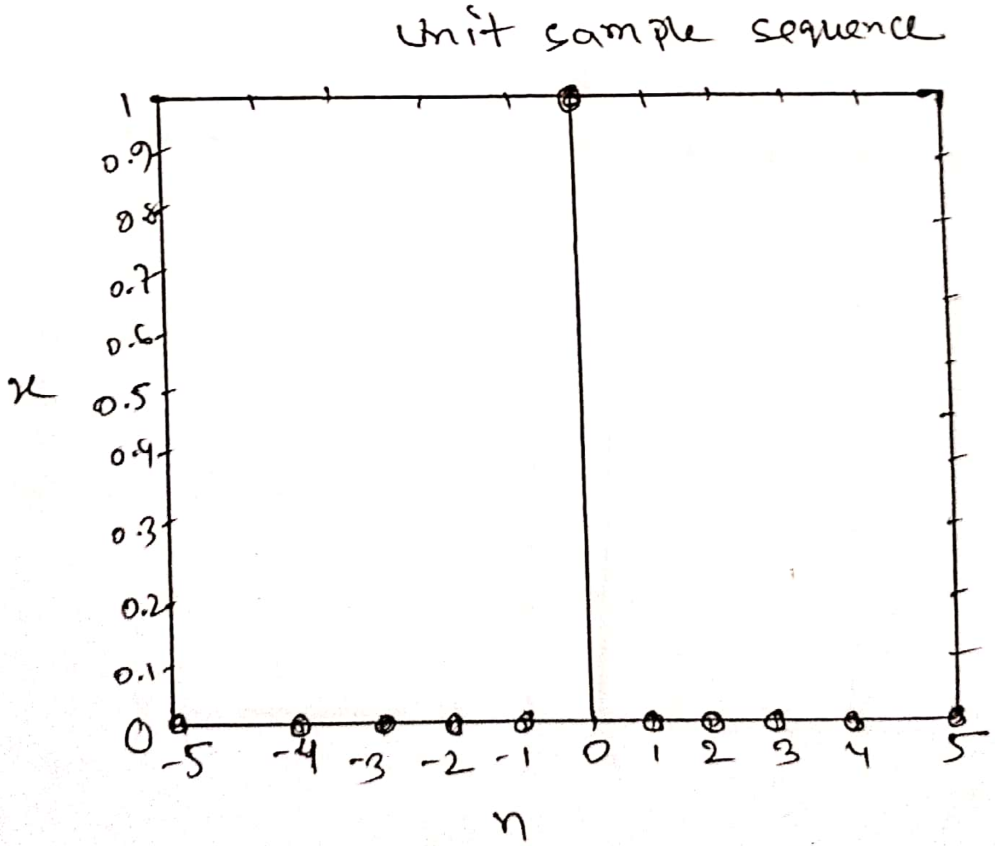
Fig: Graphical representation of unit ramp signal

Matlab code:

The unit sample sequence using matlab:

```
clc;
clear all;
close all;
n=-5:5;
x=[n==0];
stem(n,x);
title('Unit sample sequence');
xlabel('n');
ylabel('x');
```

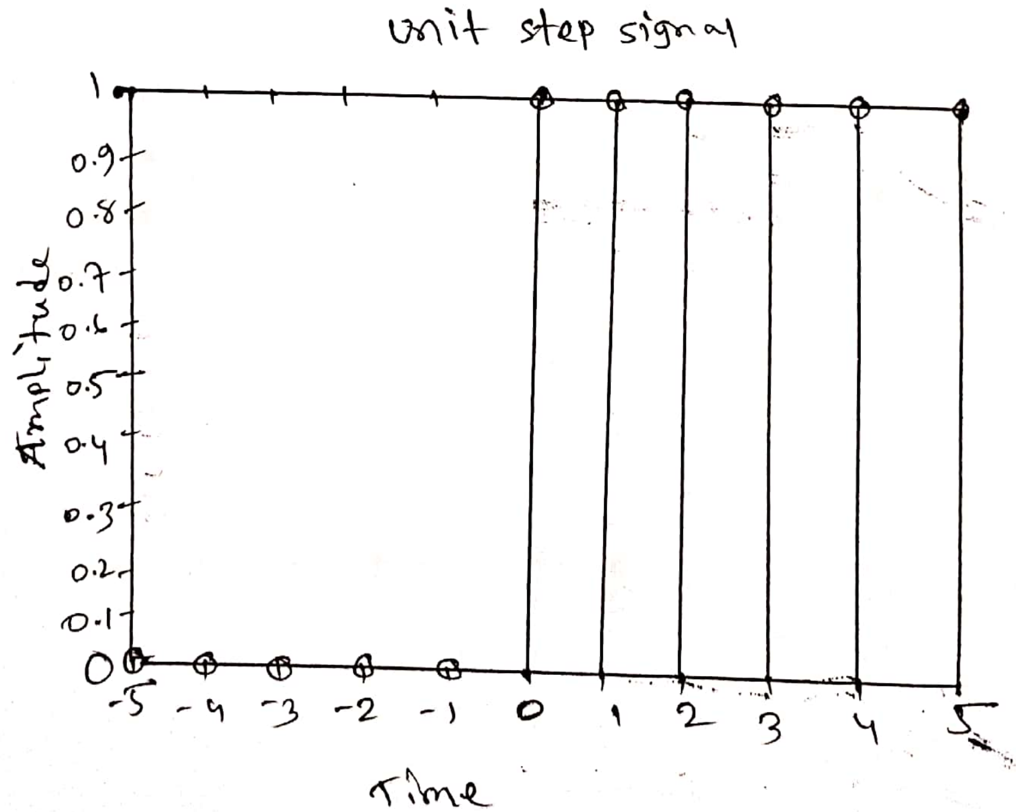
output:



The unit step signal using Matlab code:

```
clc;
clear all;
close all;
N = 5; % number of sample
n = -5:1:5;
x = [(n>=0)==1];
stem(n,x);
xlabel('Time');
ylabel('Amplitude');
title('unit step');
```

Output:



The unit ramp signal using matlab:

```
clc;
clear all;
close all;
n=-5:5;
x=[(((n>=1)==1).*n)]
stem(n,x);
xlabel('time');
ylabel('Amplitude');
title('Unit ramp signal');
```

Output: