Department of

Electronic &Telecommunication Engineering

LAB MANUAL

SIGNAL & SYSTEM LAB

B.Tech– IV Semester

KCT College OF ENGG AND TECH.

VILLAGE FATEHGARH

DISTT.SANGRUR

INDEX

Experiment No. - 1

AIM: - Generation of Continuous and Discrete Unit Step Signal.

Apparatus: - MATLAB SOFTWARE (Version 7.13)

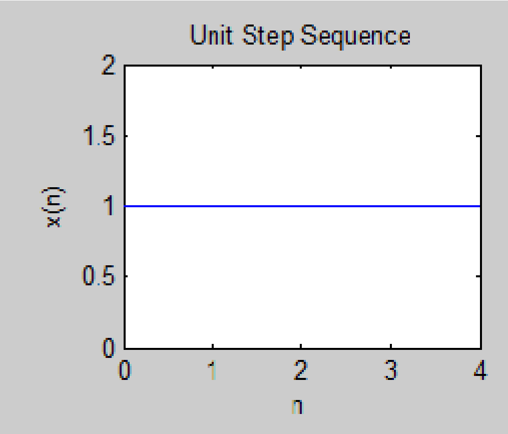
Source Code:-

```
n=input('Enter the Length of the step sequence N=');
t=0:n-1;
y=ones(1,n);
subplot(2,2,1);
plot(t,y);
xlabel('n');
ylabel('x(n)');
title('Unit Step Sequence');
subplot(2,2,2);
stem(t,y);
xlabel('n');
ylabel('x(n)');
title('Unit Step Sequence');
```

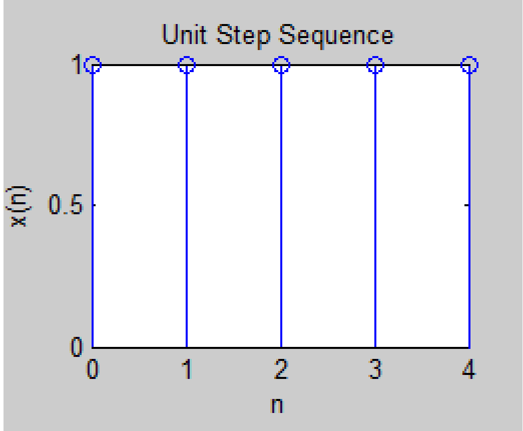
Output:-

Enter the Length of the step sequence N=5

Graphs:-



(Continuous Form)



(Discrete Form)

Experiment No. - 2

AIM: - Generation of Exponential and Ramp Signal in Continuous and Discrete Domain.

Apparatus: - MATLAB SOFTWARE (Version 7.13)

1. Exponential Signal:-

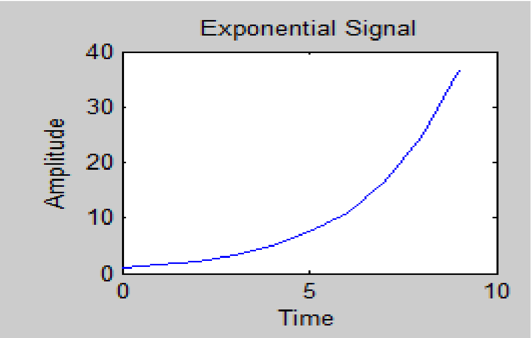
Source Code:-

```
n=input('Enter the Duration of Signal=');
a=input('Enter the Scaling Factor');
t=0:1:n-1;
y=exp(a*t);
subplot(2,2,1);
plot(t,y);
xlabel('Time');
ylabel('Amplitude');
title('Exponential Signal');
subplot(2,2,2);
stem(t,y);
xlabel('Time');
ylabel('Amplitude');
title('Exponential Signal');
```

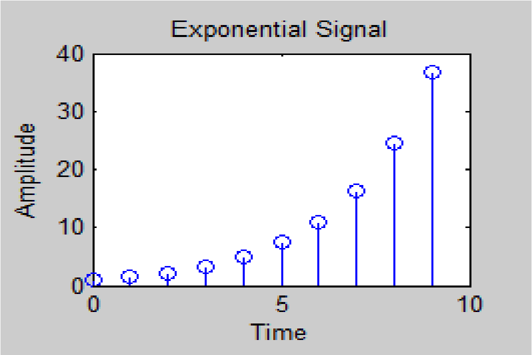
Output:-

```
Enter the Duration of Signal=10
Enter the Scaling Factor=.4
```

Graphs:-



(Continuous Form)



(Discrete Form)

2. Ramp Signal:-

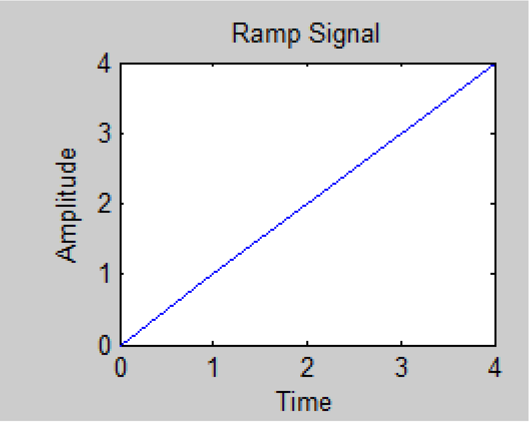
Source Code:-

```
n=input('Enter the Duration of Signal N=');
t=0:n-1;
y=t;
subplot(2,2,1);
plot(t,y);
xlabel('Time');
ylabel('Amplitude');
title('Ramp Signal');
subplot(2,2,2);
stem(t,y);
xlabel('Time');
ylabel('Amplitude');
title('Ramp Signal');
```

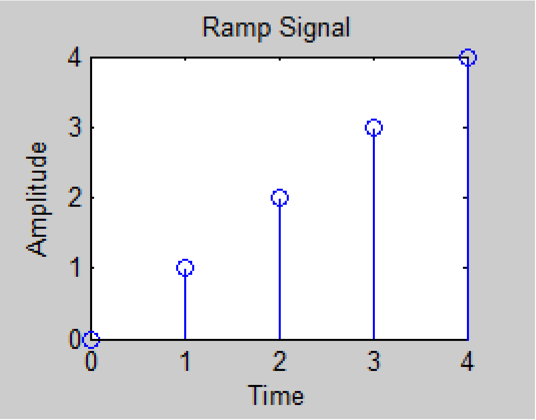
Output:-

Enter the Duration of Signal N=5

Graph:-



(Continuous Form)



(Discrete Form)

Experiment No. - 3

AIM: - Adding and subtracting two Signal (Continuous as well as Discrete Signals).

Apparatus: - MATLAB SOFTWARE (Version 7.13)

1. Addition of Signals:-

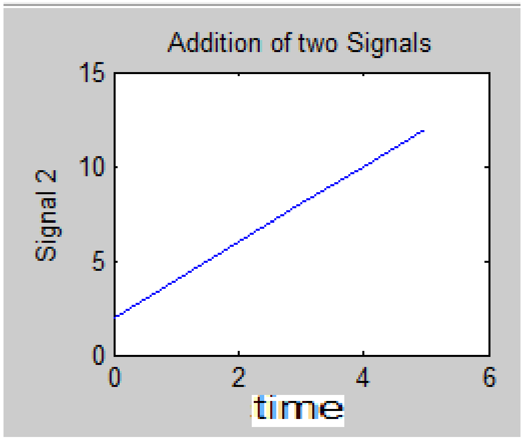
Source Code:-

```
n=input('Enter the Duration of Signal=');
t=0:1:n-1;
a=t+2;
b=t;
c=a+b;
subplot(2,2,1);
plot(t,c);
xlabel('Signal 1');
ylabel('Signal 2');
title('Addition of two Signals');
subplot(2,2,2);
stem(t,c);
xlabel('Signal 1');
ylabel('Signal 2');
title('Addition of Signals');
```

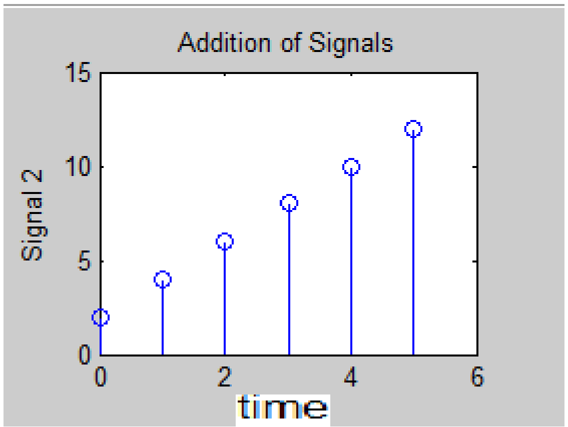
Output:-

Enter the Duration of Signal=6

Graphs:-



(Continuous Form)



(Discrete Form)

2. Subtraction of Signals:-

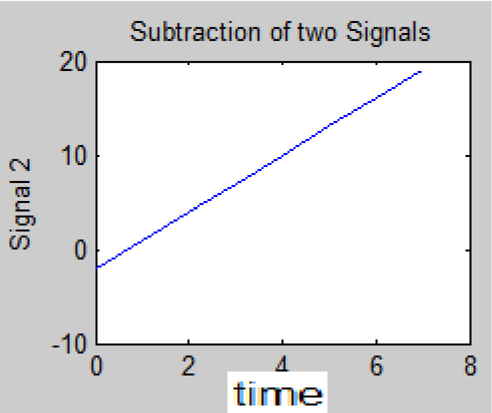
Source Code:-

```
n=input('Enter the Duration of Signal=');
t=0:1:n-1;
a=t*4;
b=t+2;
c=a-b;
subplot(2,2,1);
plot(t,c);
xlabel('Signal 1');
ylabel('Signal 2');
title('Subtraction of two Signals');
subplot(2,2,2);
stem(t,c);
xlabel('Signal 1');
ylabel('Signal 2');
title('Subtraction of Signals');
```

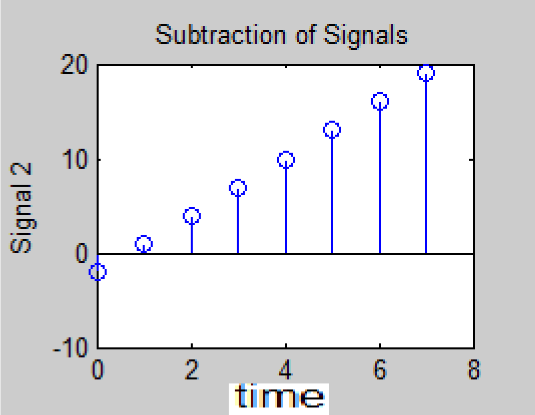
Output:-

Enter the Duration of Signal N=8

Graph:-



(Continuous Form)



(Discrete Form)

Experiment No. - 4

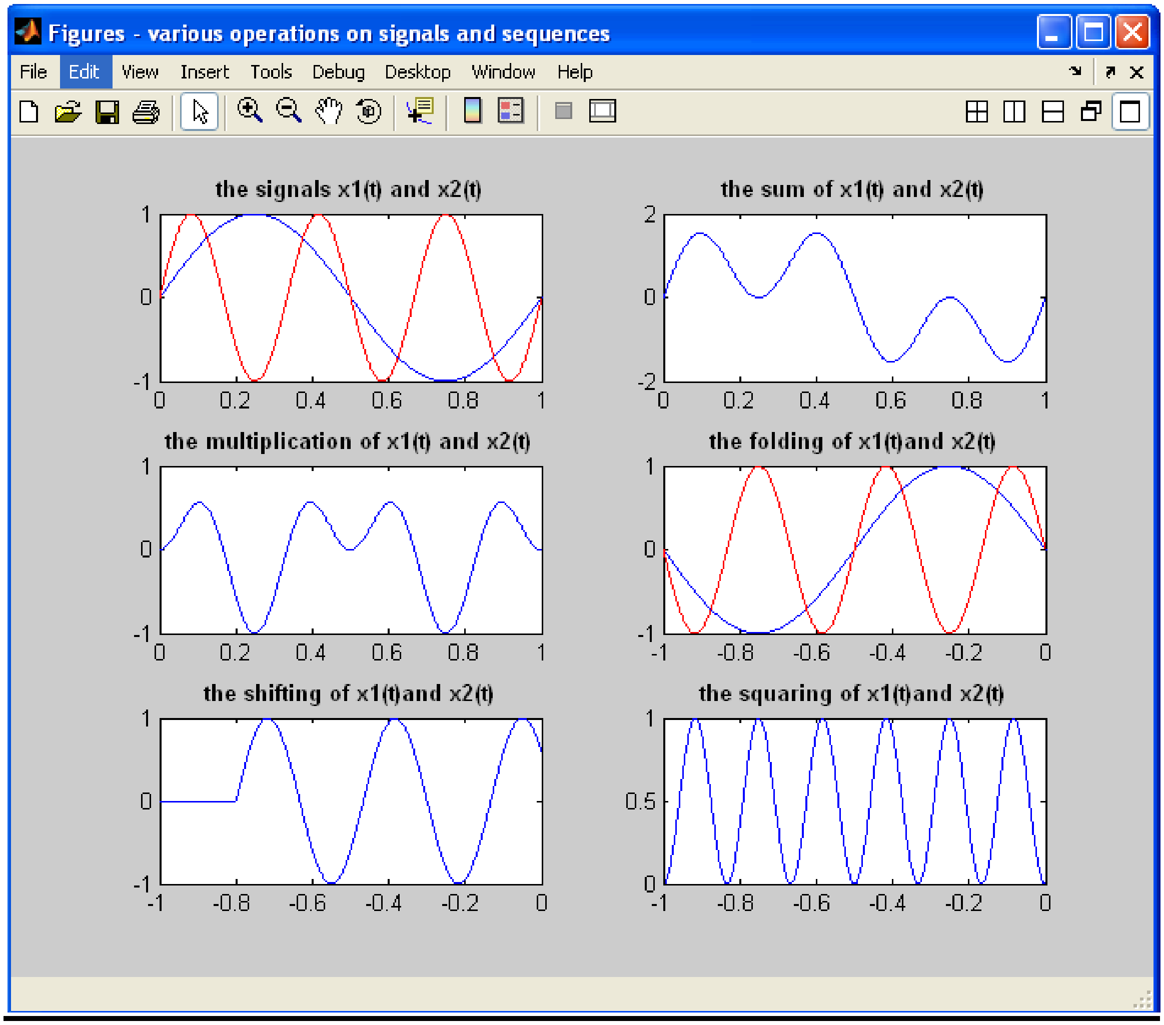
AIM: - To develop program module based on operation on sequences like signal shifting, signal folding, signal addition and signal multiplication.

Apparatus: - MATLAB SOFTWARE (Version 7.13)

Source Code:-

```
clc;
close all;
clear all;
t=0:0.001:1;
L=length(t);
f1=1;
f2=3;
x1=sin(2*pi*f1*t);
x2=sin(2*pi*f2*t);
figure;
subplot(3,2,1);
plot(t,x1,'b',t,x2,'r');
title('the signals x1(t) and x2(t)');
x3=x1+x2;
subplot(3,2,2);
plot(t,x3);
title('the sum of x1(t) and x2(t)');
x4=x1.*x2;
subplot(3,2,3);
plot(t,x4);
title('the multiplication of x1(t) and x2(t)');
t=-1:0.001:0;
x5=sin(2*pi*f1*(-t));
x6=sin(2*pi*f2*(-t));
subplot(3,2,4);
plot(t,x5,'b',t,x6,'r');
title('the folding of x1(t)and x2(t)');
x7=[zeros(1,200),x2(1:(L-200))];
subplot(3,2,5);
plot(t,x7);
title('the shifting of x1(t)and x2(t)');
x8=x2.^2;
subplot(3,2,6);
plot(t,x8);
title('the squaring of x1(t)and x2(t)');
```

Graphs:-


Figures - various operations on signals and sequences
File Edit View Insert Tools Debug Desktop Window Help
the signals x1(t) and x2(t)
1
0
-1
0 0.2 0.4 0.6 0.8 1
the sum of x1(t) and x2(t)
2
0
-2
0 0.2 0.4 0.6 0.8 1
the multiplication of x1(t) and x2(t)
1
0
-1
0 0.2 0.4 0.6 0.8 1
the folding of x1(t)and x2(t)
1
0
-1
-1 -0.8 -0.6 -0.4 -0.2 0
the shifting of x1(t)and x2(t)
1
0
-1
-1 -0.8 -0.6 -0.4 -0.2 0
the squaring of x1(t)and x2(t)
1
0.5
0
-1 -0.8 -0.6 -0.4 -0.2 0

Experiment No. - 5

AIM: - To develop elementary signal function modules (m-files) for unit sample, unit step, exponential and unit ramp sequences.

Apparatus : MATLAB software.

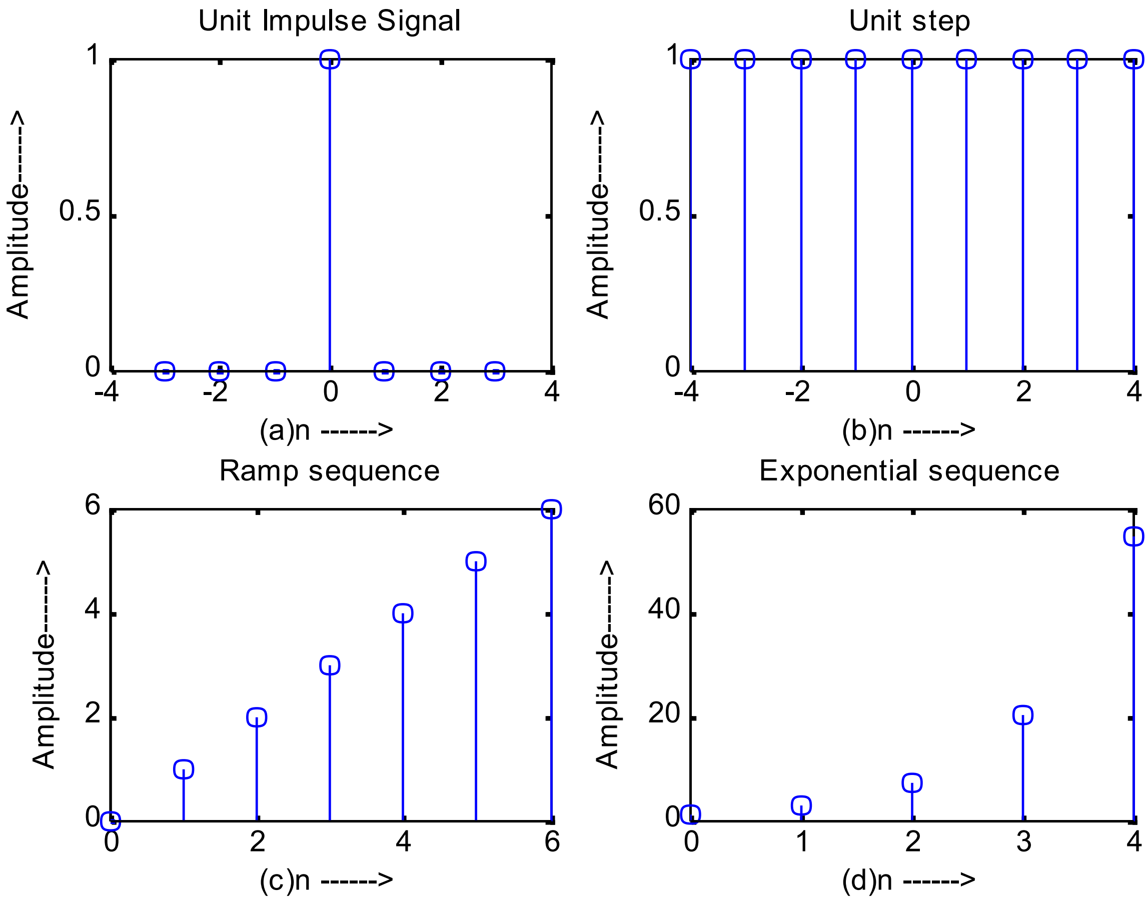
Source Code:-

```
% program for generation of unit sample
clc;clear all;close all;
t = -3:1:3;
y = [zeros(1,3),ones(1,1),zeros(1,3)];
subplot(2,2,1);stem(t,y);
ylabel('Amplitude------>');
xlabel('(a)n ------>');
title('Unit Impulse Signal');

% program for genration of unit step of sequence [u(n)- u(n)-N]
t = -4:1:4;
y1 = ones(1,9);
subplot(2,2,2);stem(t,y1);
ylabel('Amplitude------>');
xlabel('(b)n ------>');
title('Unit step');

% program for generation of ramp signal
n1 = input('Enter the value for end of the seqeuence ');
x = 0:n1;
subplot(2,2,3);stem(x,x);
ylabel('Amplitude------>');
xlabel('(c)n ------>');
title('Ramp sequence');

% program for generation of exponential signal
n2 = input('Enter the length of exponential seqeuence ');   %n2 = <any
value>7 %
t = 0:n2;
a = input('Enter the Amplitude');    %a=1%
y2 = exp(a*t);
subplot(2,2,4);stem(t,y2);
ylabel('Amplitude------>');
xlabel('(d)n ------>');
title('Exponential sequence');
disp('Unit impulse signal');y
disp('Unit step signal');y1
disp('Unit Ramp signal');x
disp('Exponential signal');x
```

Output :

```
Enter the value for end of the seqeuence 6

Enter the length of exponential seqeuence 4

Enter the Amplitude1


Unit impulse signal  y =     0    0    0    1    0    0    0


Unit step signal  y1 = 1    1    1    1    1    1    1    1    1


Unit Ramp signal  x = 0    1    2    3    4    5    6


Exponential signal  x =  0    1    2    3    4    5    6
```

Graph:



Experiment No. - 6

Aim : To develop program for discrete convolution and correlation.

Apparatus : PC having MATLAB software.

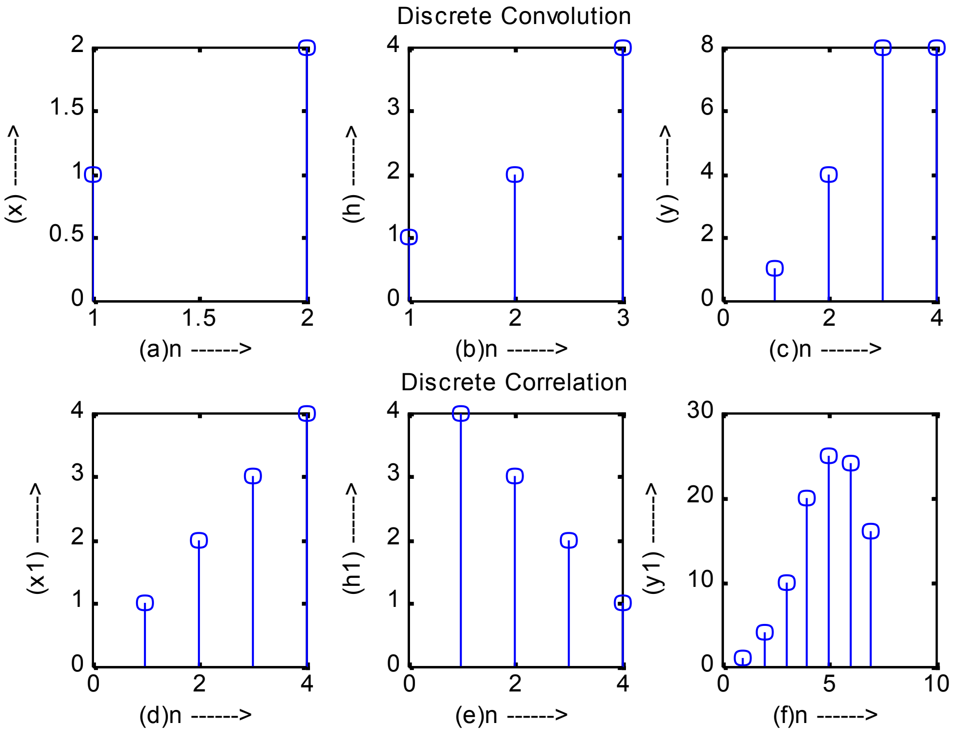
Source code

```
% program for discrete convolution
% of x= [1 2] and h = [1 2 4]
clc;clear all;close all;
x = input('Enter the 1st sequence : ');   %[1 2]
h = input('Enter the 2nd  sequence : ');   %[1 2 4]
y =conv(x,h);
subplot(2,3,1);stem(x);
ylabel('(x) ------>');
xlabel('(a)n ------>');
subplot(2,3,2);stem(h);
ylabel('(h) ------>');
xlabel('(b)n ------>');
title('Discrete Convolution');
subplot(2,3,3);stem(y);
ylabel('(y) ------>');
xlabel('(c)n ------>');
disp(' The resultant Signal is :');y
% program for discrete correlation
% of h =[4 3 2 1]
x1 = input('Enter the 1st sequence : ');   %[1 2 3 4]
h1 = input('Enter the 2nd  sequence : ');   %[4 3 2 1]
y1 =xcorr(x1,h1);
subplot(2,3,4);stem(x1);
ylabel('(x1) ------>');
xlabel('(d)n ------>');
subplot(2,3,5);stem(h1);
ylabel('(h1) ------>');
xlabel('(e)n ------>');
title('Discrete Correlation');
subplot(2,3,6);stem(y1);
ylabel('(y1) ------>');
xlabel('(f)n ------>');
disp(' The resultant Signal is :');y1
```

Output :

```
Convolution :
Enter the 1st sequence : [1 2]
Enter the 2nd  sequence : [1 2 4]
 The resultant Signal is : y =  1    4    8    8
Correlation :
Enter the 1st sequence :   [1 2 3 4]
Enter the 2nd  sequence : [4 3 2 1]
 The resultant Signal is : y1 = 1.0000    4.0000   10.0000   20.0000   25.0000   24.0000   16.0000
```

Graphs:


Discrete Convolution
2
1.5
1
0.5
0
1
1.5
2
(x) ------>
(a)n ------>
4
3
2
1
0
1
2
3
(h) ------>
(b)n ------>
8
6
4
2
0
0
2
4
(y) ------>
(c)n ------>
Discrete Correlation
4
3
2
1
0
0
2
4
(x1) ------>
(d)n ------>
4
3
2
1
0
0
2
4
(h1) ------>
(e)n ------>
30
20
10
0
0
5
10
(y1) ------>
(f)n ------>

Experiment No. - 7

Aim : To develop program for finding the response of the LTI system by difference equation.

Apparatus : MATLAB software.

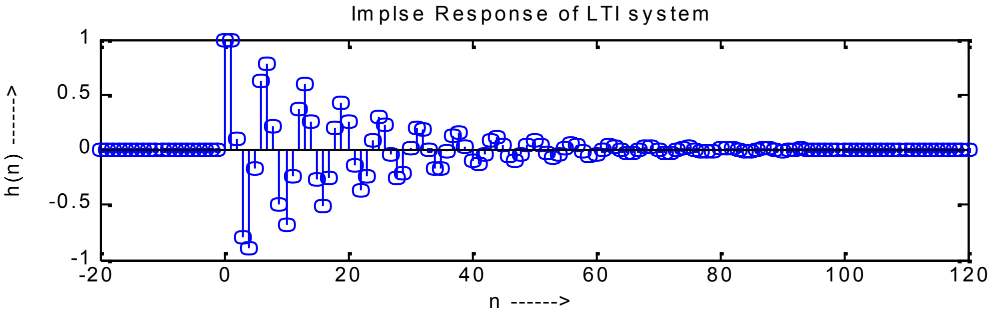
Source Code:-

```
% prog for finding the response of LTI system by difference equation
% let y(n) -y(n-1) +0.9y(n-2)=x(n) plot impulse response h(n) at
% n = 20,...100
b = [1];
a = [1,-1,0.9];      %  coefficient arrays from the =n
x = impseq(0,-20,120); n = [-20:120];
h = filter(b,a,x)
subplot(2,1,1);stem(n,h);
ylabel('h(n) ------>');
xlabel('n ------>');
title('Implse Response of LTI system');


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

Output :

```
h = 0          0        0              0      0      0            0
    0          0        0        0      0      0            0
    0          0        0        0      0      0    1.0000
   1.0000    0.1000   -0.8000   -0.8900   -0.1700    0.6310    0.7840
   0.2161   -0.4895   -0.6840   -0.2434    0.3722    0.5912    0.2563
  -0.2758   -0.5065   -0.2583    0.1976    0.4300    0.2522   -0.1348
  -0.3618   -0.2405    0.0852    0.3016    0.2249   -0.0465   -0.2489
  -0.2071    0.0169    0.2033    0.1881    0.0051   -0.1642   -0.1688
  -0.0210    0.1309    0.1498    0.0320   -0.1028   -0.1316   -0.0391
   0.0794    0.1145    0.0431   -0.0600   -0.0988   -0.0448    0.0441
   0.0844    0.0447   -0.0313   -0.0715   -0.0434    0.0210    0.0600
   0.0411   -0.0129   -0.0499   -0.0383    0.0066    0.0411    0.0351
  -0.0018   -0.0335   -0.0318   -0.0017    0.0269    0.0285    0.0042
  -0.0214   -0.0252   -0.0059    0.0167    0.0221    0.0070   -0.0129
  -0.0192   -0.0076    0.0097    0.0165    0.0078   -0.0070   -0.0141
  -0.0077    0.0049    0.0119    0.0074   -0.0032   -0.0099   -0.0070
   0.0019    0.0083    0.0065   -0.0009   -0.0068   -0.0060    0.0001
   0.0055    0.0054    0.0004   -0.0044   -0.0048   -0.0008    0.0035
   0.0042    0.0011   -0.0027   -0.0037   -0.0013    0.0021    0.0032
  0.0013   -0.0015   -0.0028   -0.0014    0.0011    0.0023    0.0013
 -0.0008
```

Graph:


Implse Response of LTI system
h(n) ------>
n ------>
1
0.5
0
-0.5
-1
-20
0
20
40
60
80
100
120

Experiment No. - 8

Aim : To generate a Gaussian noise and to compute its Mean, Mean Square Value and Probability Distribution function.

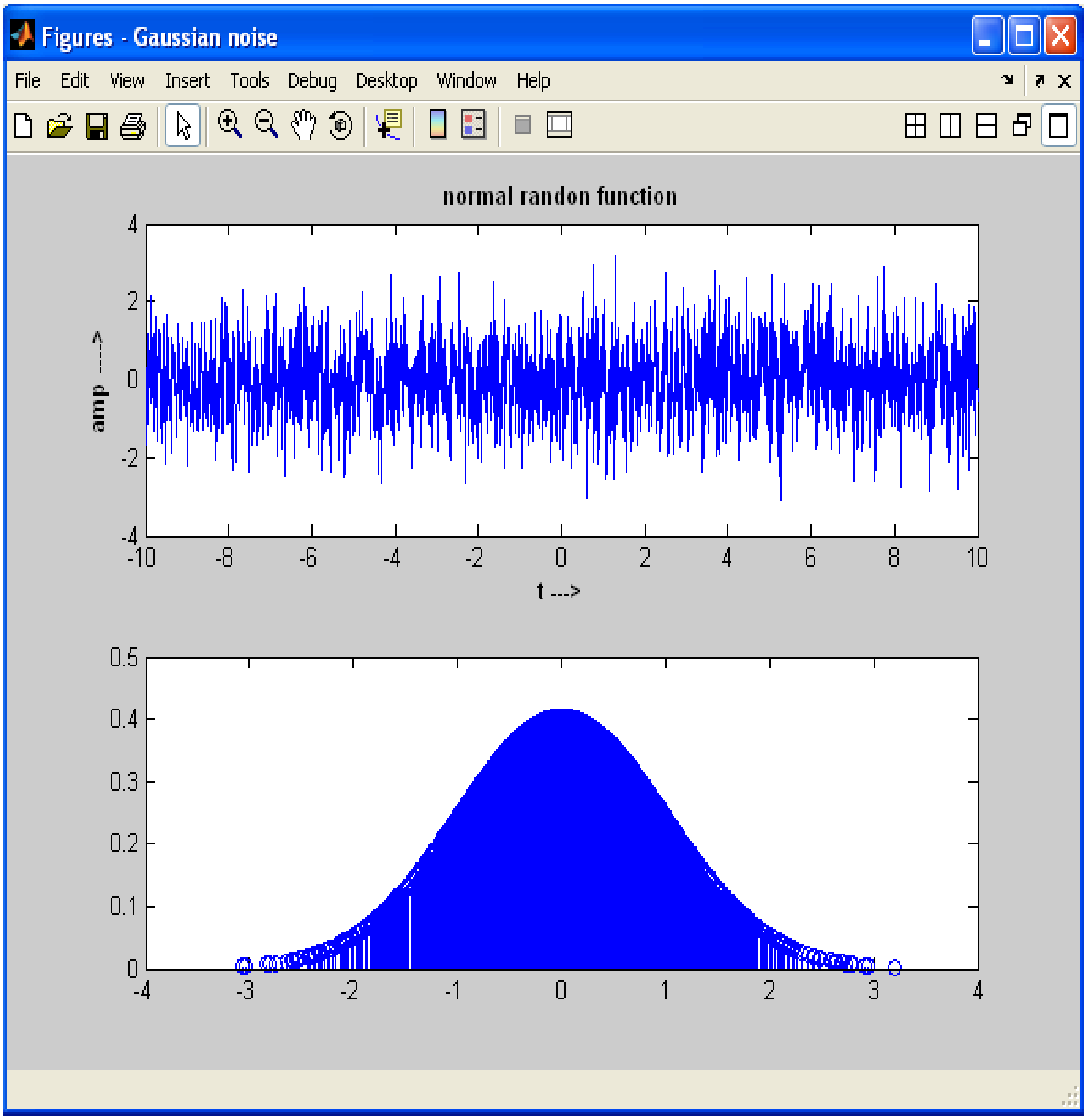
Apparatus : MATLAB software.

Source Code:-

```
clc; clear all; close all;
t=-10:0.01:10;
L=length(t);
n=randn(1,L);
subplot(2,1,1);
plot(t,n);
xlabel('t --->'),ylabel('amp ---->');
title('normal randon function');
nmean=mean(n);
disp('mean=');disp(nmean);
nmeansquare=sum(n.^2)/length(n);
disp('mean square=');disp(nmeansquare);
nstd=std(n);
disp('std=');disp(nstd);
nvar=var(n);
disp('var=');disp(nvar);
p=normpdf(n,nmean,nstd);
subplot(2,1,2);
stem(n,p)
```

OUTPUT:-

```
Mean=
 9.2676e-004

Mean square=
   0.9775

STD=
   0.9889

Var=
   0.9780
```


Figures - Gaussian noise
File Edit View Insert Tools Debug Desktop Window Help
normal randon function
amp ---->
t --->
4
2
0
-2
-4
-10
-8
-6
-4
-2
0
2
4
6
8
10
0.5
0.4
0.3
0.2
0.1
0
-4
-3
-2
-1
0
1
2
3
4

Experiment No. - 9

Aim : To develop program for computing inverse Z-transform.

Apparatus : MATLAB software.

Source Code:-

```
%prog for computing the inverse Z-transform by using residuez function
b=[1,0.4*sqrt(2)];
a=[1,- 0.8*sqrt(2),0.64]; [
R,P,C]=residuez(b,a);
R
P
C
Zplane(b,a);
```

Output :

```
R =

   0.5000 - 1.0000i
   0.5000 + 1.0000i


P =

   0.5657 + 0.5657i
   0.5657 - 0.5657i


C =

     []
```