Experiment No. 2

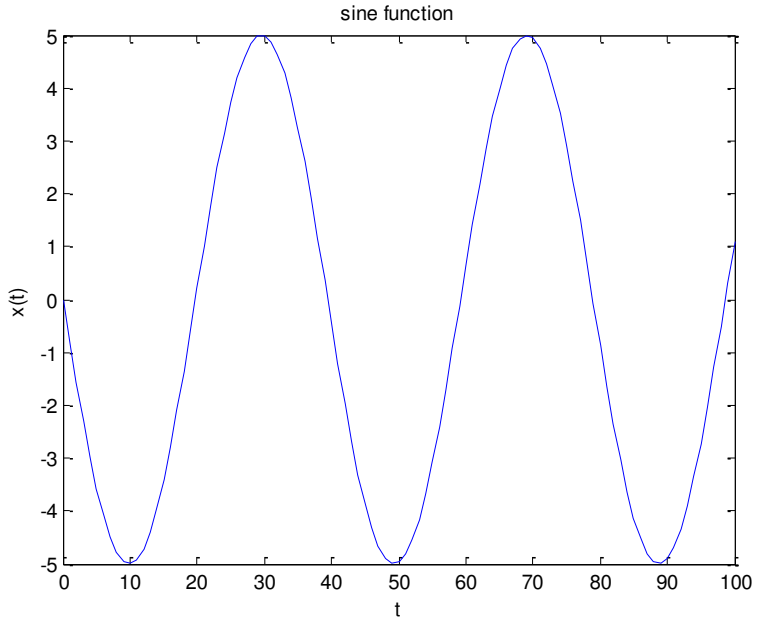
Aim:- (I). Write a Program to plot the following continous signals:

a) x(t)=5*sin(2*pi*f*t),where f=50Hz.

Program:-

```
clc;
t=(0:1:100);
f=50;
x=5*sin(2*3.14*f*t);
plot(t,x);
xlabel('t');
ylabel('x(t)');
title('sine function');
```

Result:-

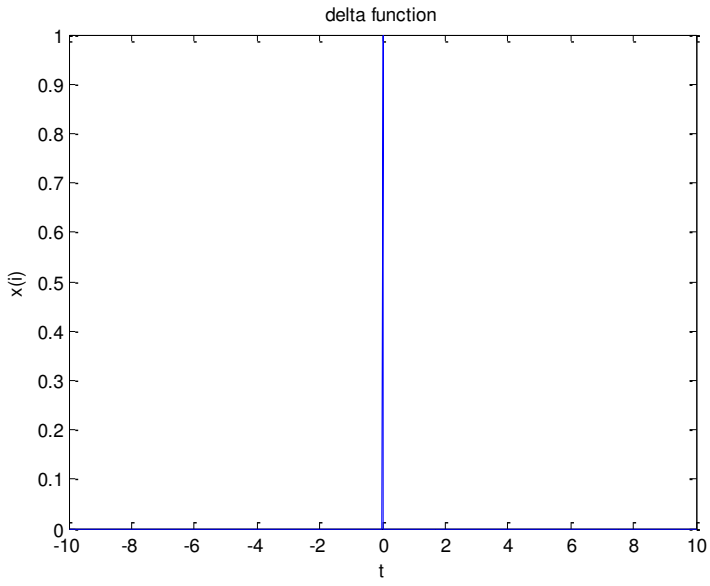


(**b). x(t)=δ(t);**

Program: -

```
clc;
t=(-10:0.01:10);
for i=1:length(t);
if (t(i)==0)
x(i)=1;
else
x(i)=0;
end;
end;
plot(t,x);
xlabel('t');
ylabel('x(i)');
title('delta function');
```

Result:-



(c). x(t)=u(t)

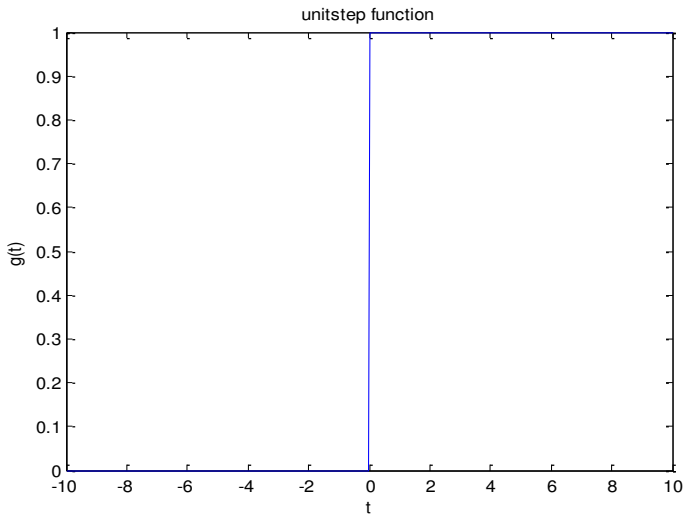
Program:-

```
clc;
t=(-10:0.01:10);
for i=1:length(t);
if (t(i)>=0)
g(i)=1;
else
g(i)=0;
end;
end;
plot(t,g);
xlabel('t');
ylabel('g(t)');
title('unitstep function');
```

OR

```
t1=-2:0.01:0.01;
t2= 0:0.01:2;
y1= 0*t1;
y2= ones(1,length(t2));
t=[t1,t2];
y=[y1,y2];
plot(t,y) ;
xlabel('t');
ylabel('y');
title('unitstep function');
```

Result:-

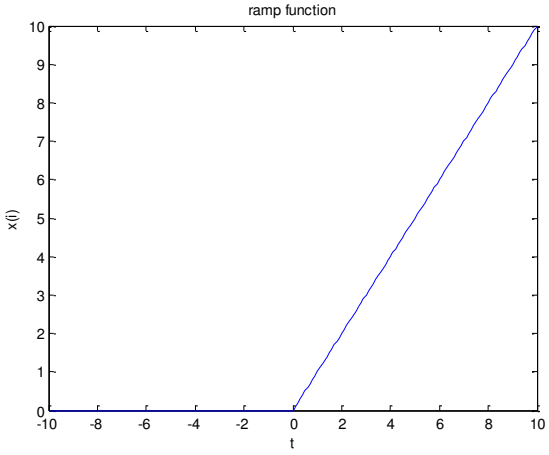


d). x(t)=r(t)

Program:-

```
t=[-10:1:10]
For i=1:length(t)
if(t(i)>=0)
x(i)=t(i)
 else
 x(i)=0
end
end
plot(t,x)
xlabel('t')
ylabel('x(i)')
```

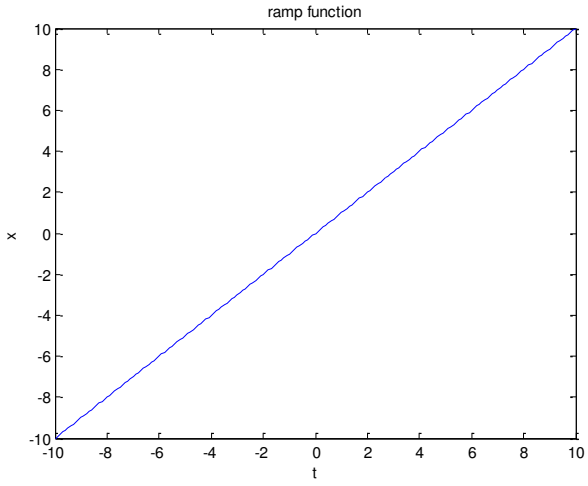
Result:-



OR

```
t=(-10:0.1:10);
x=t;
plot(t,x);
xlabel('t');
ylabel('x');
title('ramp function');
```

Result:-

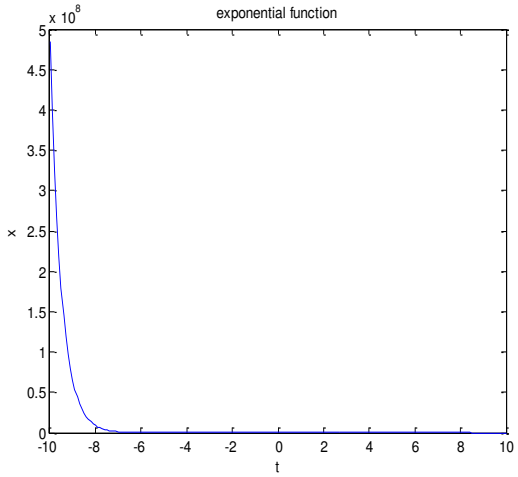


e) x(t)=exp(-a*t)

Program:-

```
clc;
t=(-10:0.1:10);
a=2;
x=exp(-a*t);
plot(t,x);
xlabel('t');
ylabel('x');
title('exponential function')
```

Result:-



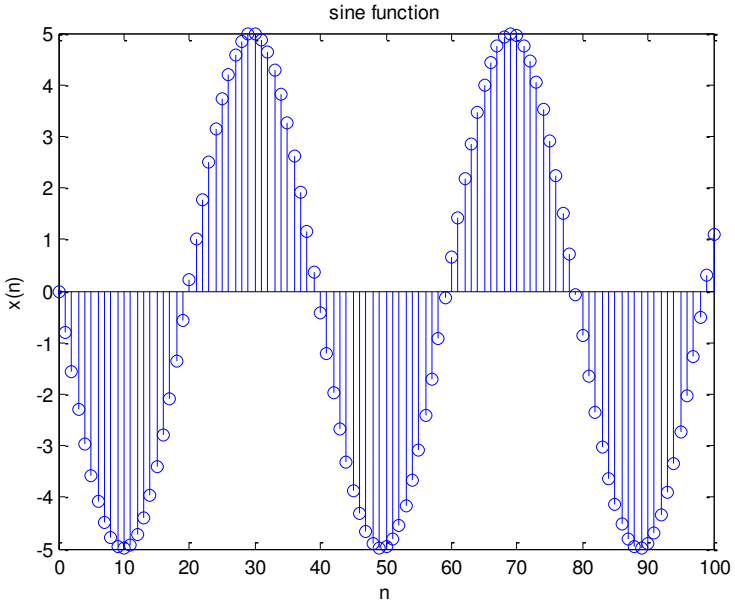
(II) Aim : Write a MATLAB Prg. to plot the following discrete sgnals:

a) x(n)=5*sin(2*pi*f*n),where f=50Hz.

Program:-

```
clc;
n=(0:1:100);
f=50;
x=5*sin(2*3.14*f*n);
stem(n,x);
xlabel('n');
ylabel('x(n)');
title('sine function');
```

Result:-



b) x(n)=δ(n);

Program:-

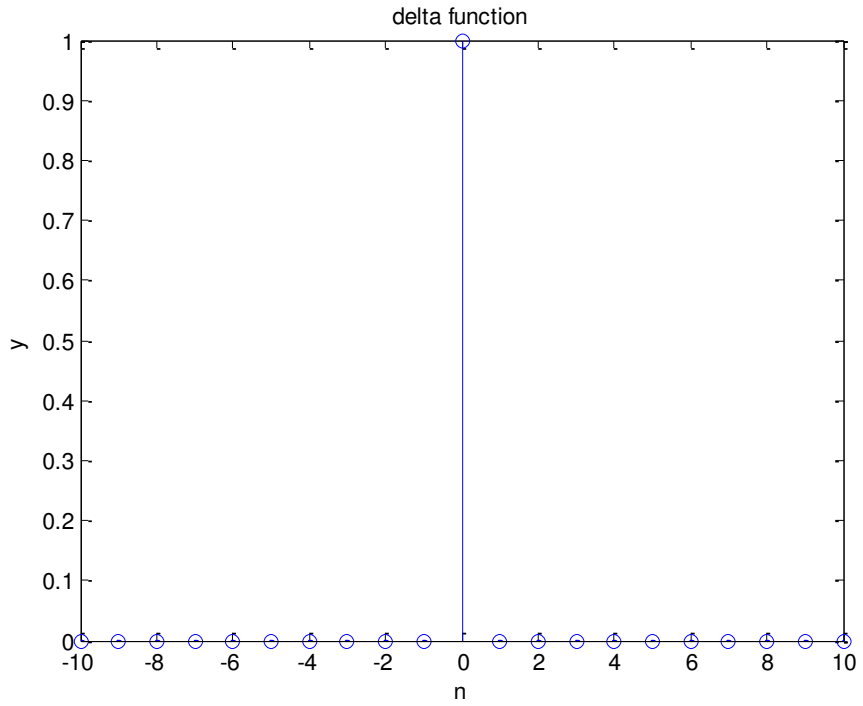
```
n=(-10:10);
for i=1:length(n);
if (n(i)==0)
g(i)=1;
 else
g(i)=0;
 end;
end;
stem(n,g);
xlabel('n');
```

```
ylabel('g(i)');
title('delta function');
```

OR

```
n=-10:10;
y=(n==0);
stem(n,y);
```

Result:-



c). x(n)=u(n)

Program :-

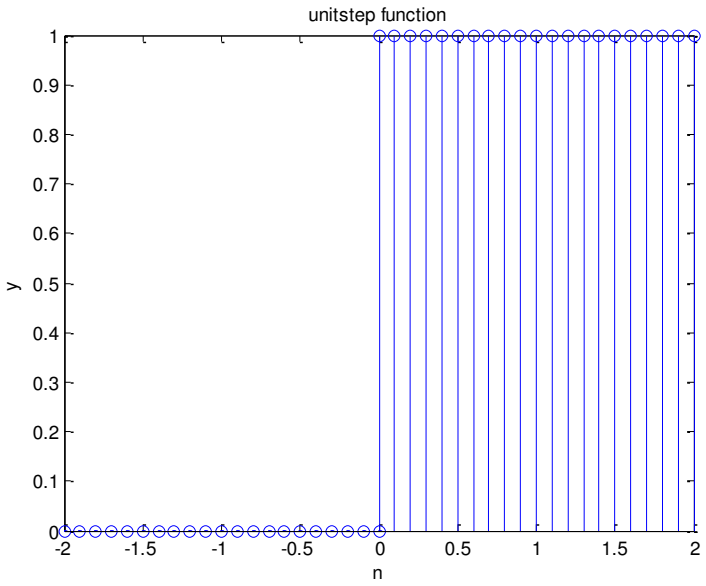
```
clc;
n=(-10:0.01:10);
for i=1:length(n);
if (n(i)>=0)
  g(i)=1;
  else
  g(i)=0;
  end;
  end;
 stem(n,g(i));
```

```
xlabel('n');
ylabel('g(n)');
title('unitstep function');
```

OR

```
n1=-2:0.1:0.01;
n2= 0:0.1:2;
y1= 0*n1;
y2= ones(1,length(n2));
n=[n1,n2];
y=[y1,y2];
stem(n,y) ;
xlabel('n');
ylabel('y');
title('unitstep function');
```

Result:-



d). **x(n)=r(n)**

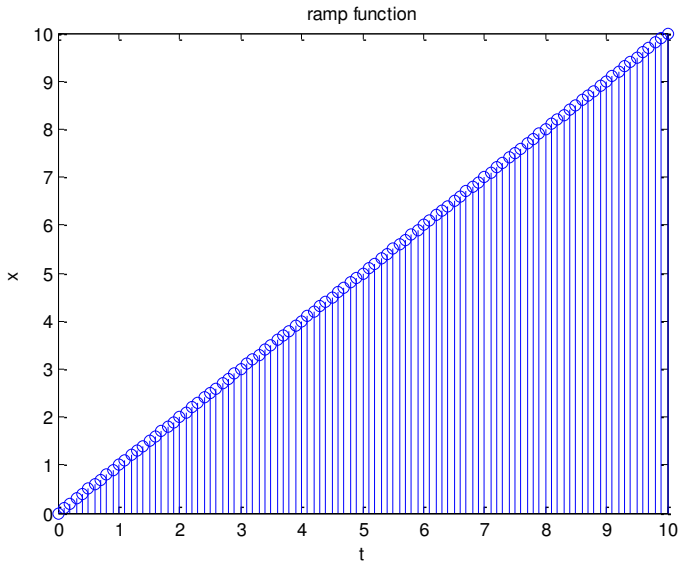
Program:-

```
t=[-10:1:10]
for i=1:length(t)
if(t(i)>=0)
x(i)=t(i)
else
x(i)=0
end
end
stem(t,x)
xlabel('t')
title('ramp function')
```

OR

```
t=(0:0.1:10);
x=t;
stem(t,x);
xlabel('t');
ylabel('x');
title('ramp function');
```

Result:-

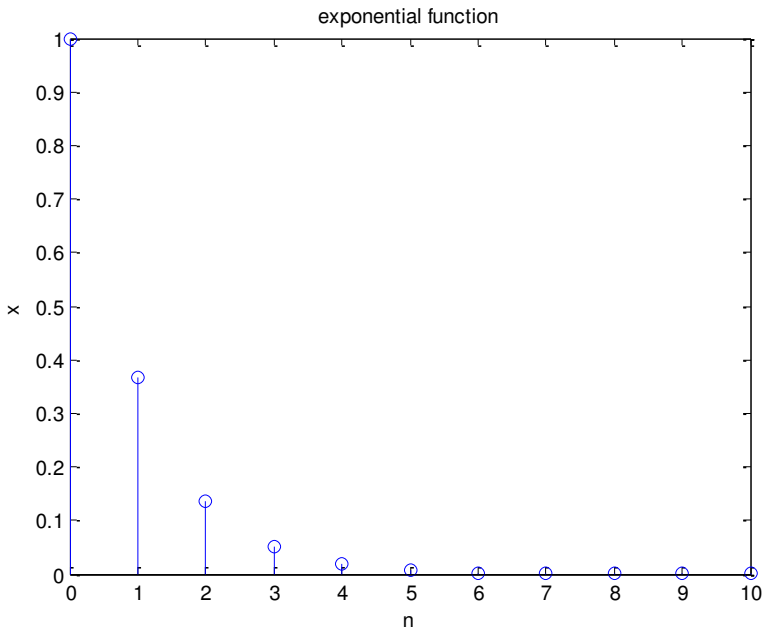


e**). x(n)=exp(-a*n)**

Program:-

```
clc;
n=(0:10);
a=1;
x=exp(-a*n);
stem(n,x);
xlabel('n');
ylabel('x');
title('exponential function');
```

Result:-



Experiment-3

Aim:- (I). Write MALAB codes for generating and plotting the following combinations of two signals over the range, -20<n<20.

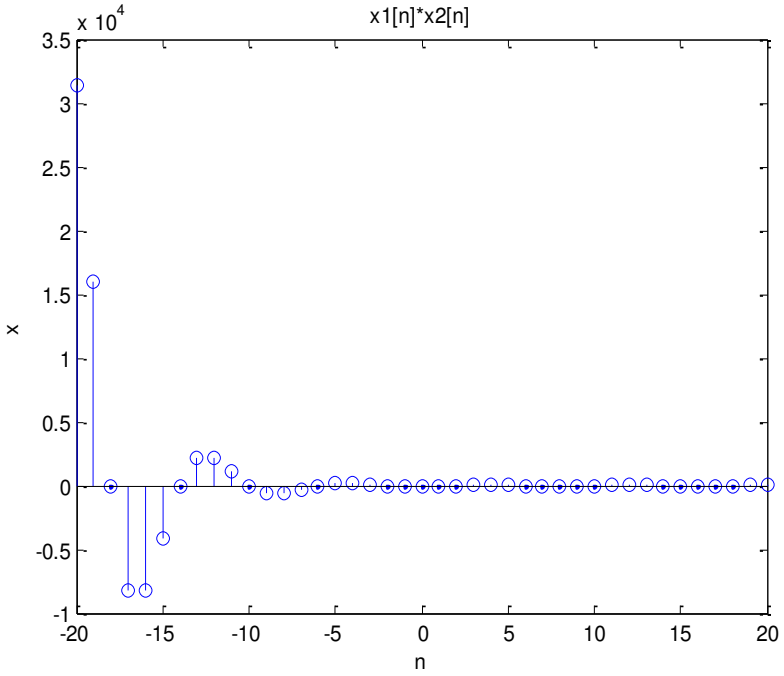
Let x1[n] = 5 cos (2π n/8) and $x_2[n] = -8\, e^{-(n/6)2}$

(a) x[n] = x_1[n] x2[n]

Program:-

```
n=-20:1:20
x1=5*cos((2*pi*n)/8)
x2=-8*exp((-n/6)*2)
 x=x1.*x2
 stem(n,x)
```

Result:-

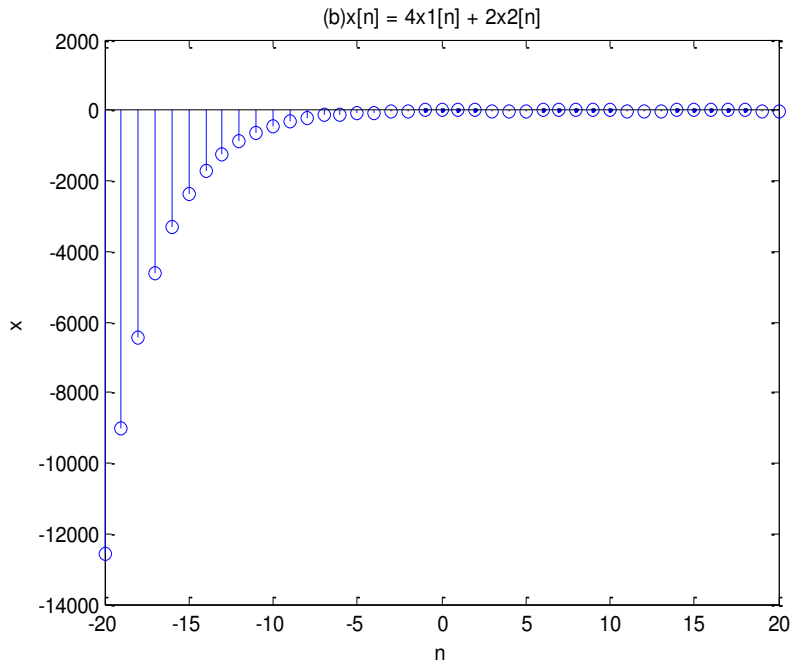


(b) $x[n] = 4x_1[n] + 2x_2[n]$

Program:-

```
n=-20:1:20
x1=5*cos((2*pi*n)/8)
x2=-8*exp((-n/6)*2)
 x=4*x1+2*x2
stem(n,x)
```

Result:-

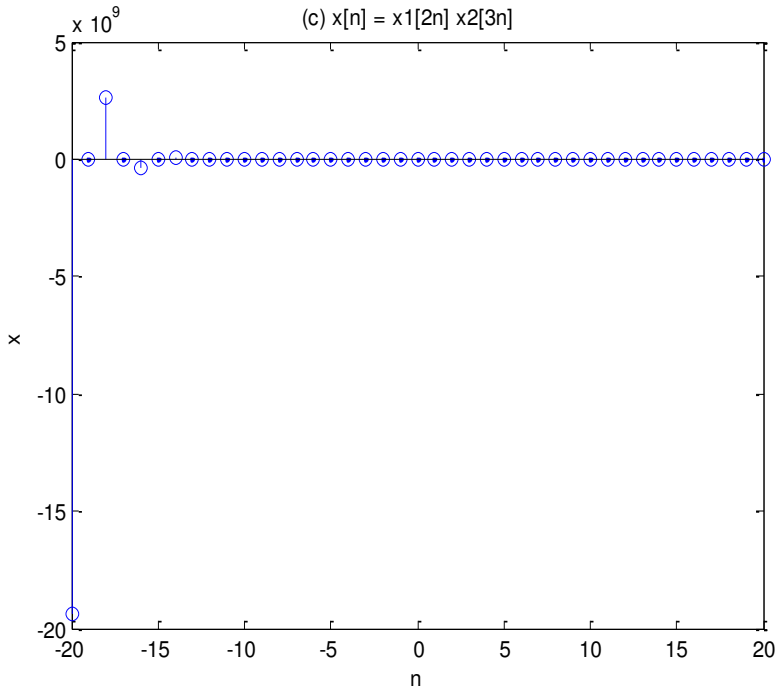


(c) x[n] = $x_1[2n]$ $x_2[3n]$

Program:-

```
n=-20:1:20
x1=5*cos((2*pi*2*n)/8)
x2=-8*exp((-3*n/6)*2)
x=x1.*x2
stem(n,x)
```

Result:-

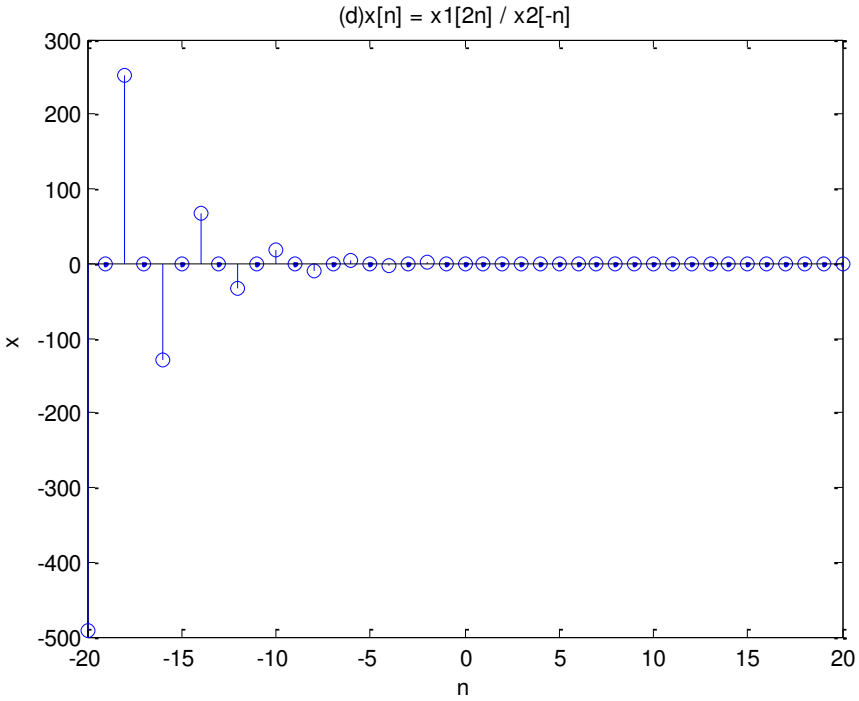


(d) $x[n] = x_1[2n] / x_2[-n]$

Program:-

```
n=-20:1:20
x1=5*cos((2*pi*2*n)/8)
x2=-8*exp((n/6)*2)
x=x1./x2
stem(n,x)
```

Result:-

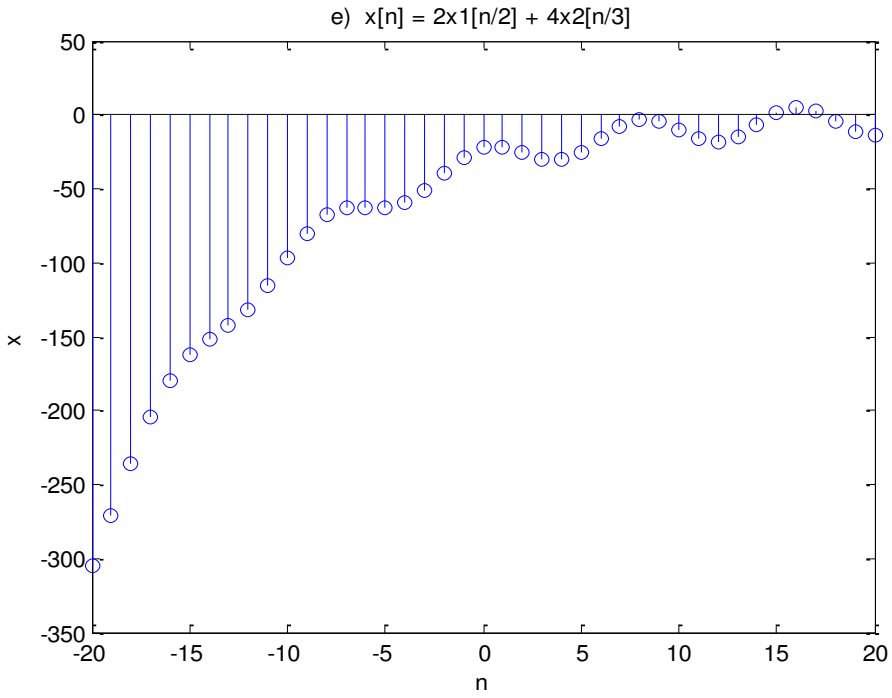


e) $x[n] = 2x_1[n/2] + 4x_2[n/3]$

Program:-

```
n=-20:1:20
x1=5*cos((2*pi*2*n)/16)
x2=-8*exp((-n/18)*2)
x =2.*x1+4.*x2
stem(n,x)
```

Result:-

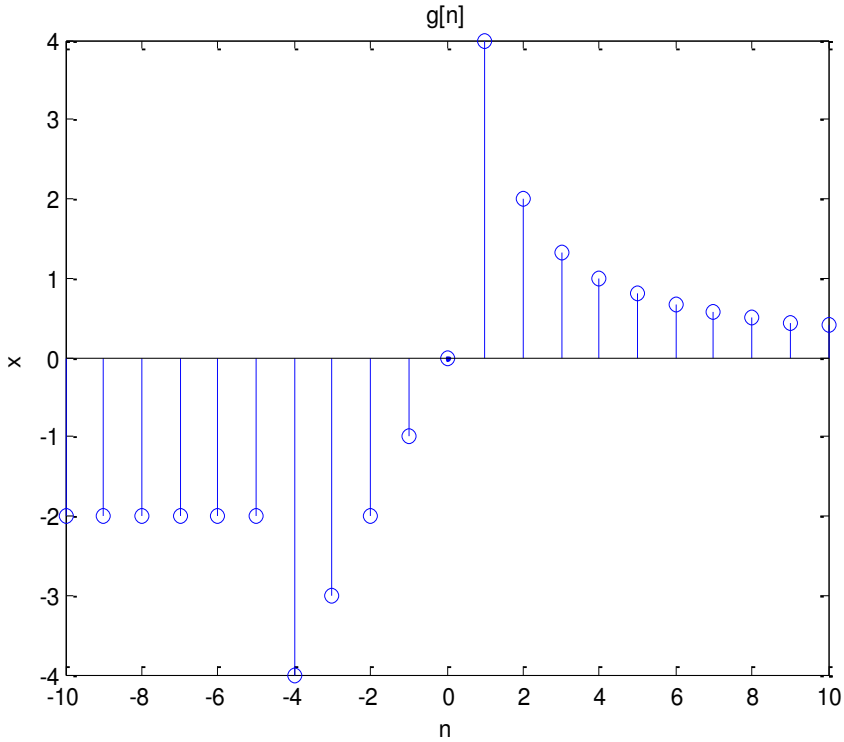


(b) Write MATLAB Codes for generating the following Sequences Sketch g [-n], g [2-n], g[2n] and g[n/2].

Program:- **g[n]**

```
n=-10:1:10
 k=n
for i=1:length(k)
if k(i)<-4
 g(i)=-2
elseif (k(i)>=-4) && (k(i)<1)
 g(i)=k(i)
 elseif k(i)>=1
   g(i)=4./k(i)
else
end
end
stem(n,g)
```

Result:-

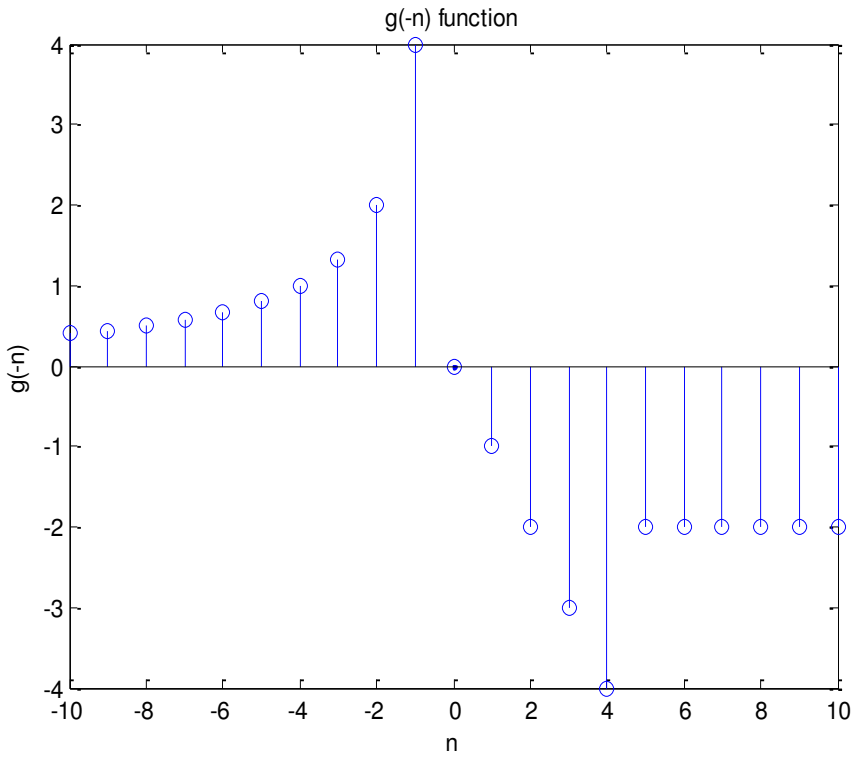


g[-n]

Program:-

```
 n=-10:1:10
 k=-n
for i=1:length(k)
if k(i)<-4
 g(i)=-2
elseif (k(i)>=-4) && (k(i)<1)
g(i)=k(i)
elseif k(i)>=1
 g(i)=4./k(i)
 else
end
end
stem(n,g)
```

Result:-

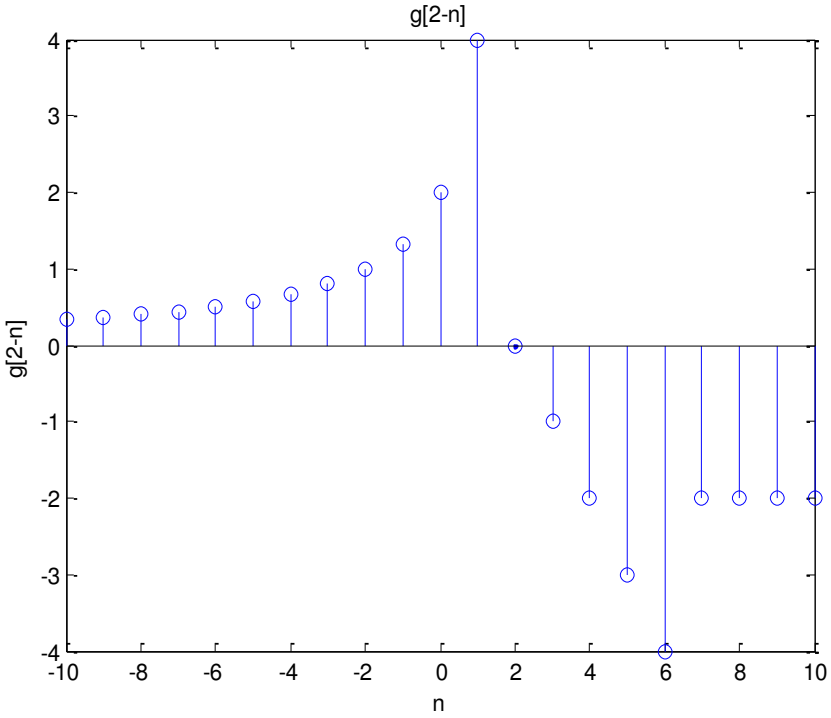


g[2-n]

Program:-

```
n=-10:1:10
k=2-n
for i=1:length(k)
if k(i)<-4
g(i)=-2
elseif (k(i)>=-4) && (k(i)<1)
g(i)=k(i)
elseif k(i)>=1
g(i)=4./k(i)
else
 end
end
stem(n,g)
```

Result:-

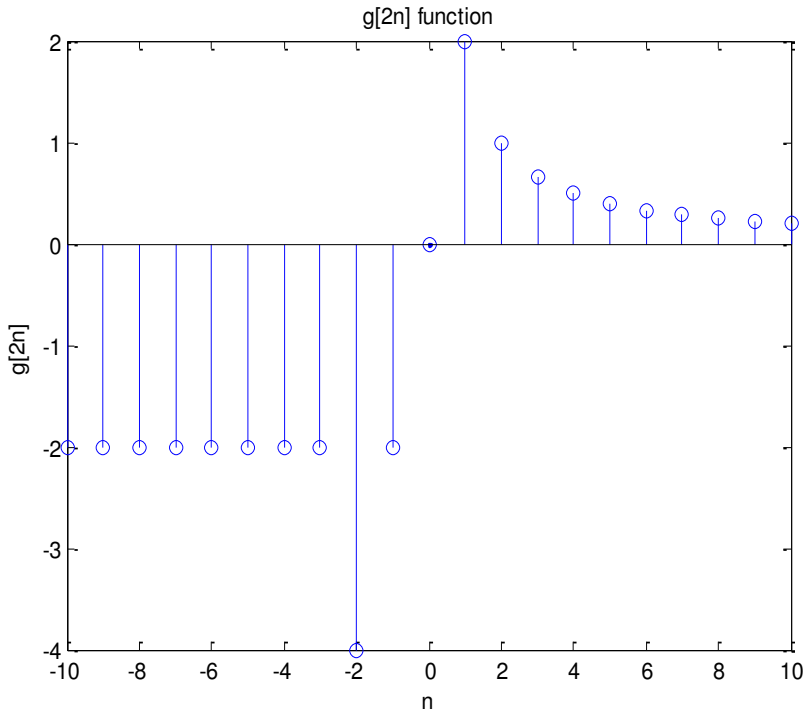


g[2n]

Program:-

```
      n=-10:1:10
 k=2*n
for i=1:length(k)
if k(i)<-4
 g(i)=-2
elseif (k(i)>=-4) && (k(i)<1)
g(i)=k(i)
elseif k(i)>=1
g(i)=4./k(i)
 else
  end
  end
  stem(n,g)
```

Result:-

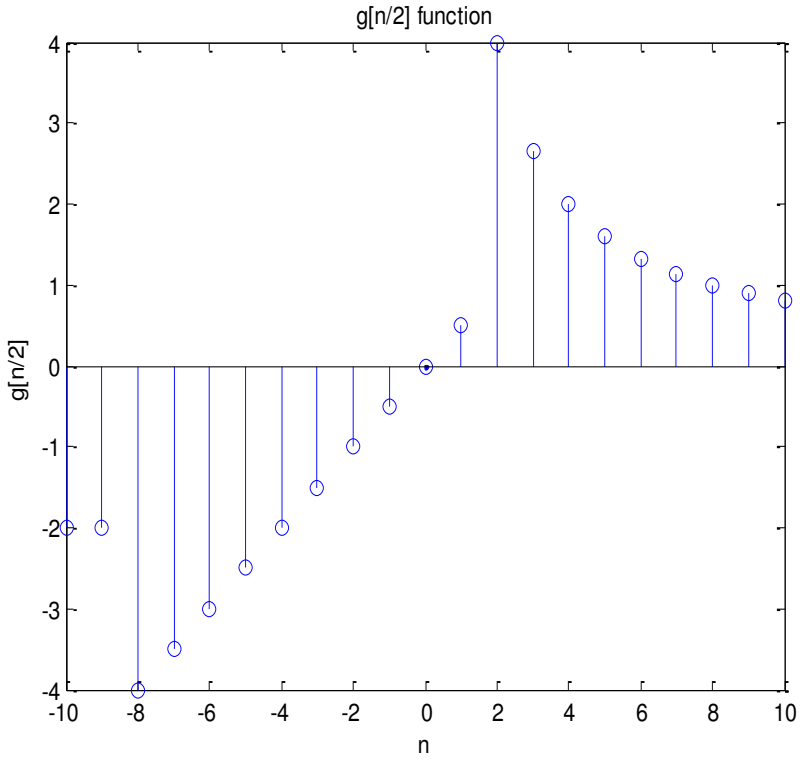


g[n/2]

Program:-

```
    n=-10:1:10
k=n/2
for i=1:length(k)
 if k(i)<-4
 g(i)=-2
 elseif (k(i)>=-4) && (k(i)<1)
 g(i)=k(i)
 elseif k(i)>=1
  g(i)=4./k(i)
 else
 end
 end
 stem(n,g)
```

Result:-



EXPERIMENT NO:-4

Aim:-. Write a program for finding the signal energy or power of the signal

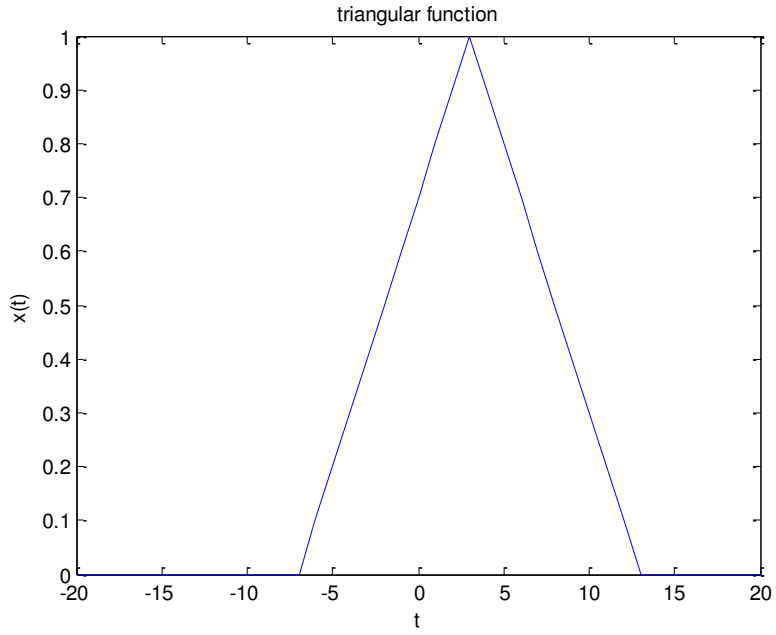
(I). x1(t) = tri ((t-3) /10) where tri is a triangular signal

Program:-

```
t=-20:1:20
x=trimf (t,[-7,3,13])
plot (t,x)
title('triangular function')
xlabel ('t')
ylabel('x(t)')


y=abs(x)
m=y. ^2
power=trapz (m)
```

Result:-



Power=66.6700

(II).A periodic signal of fundamental period of 10 describe over one period by

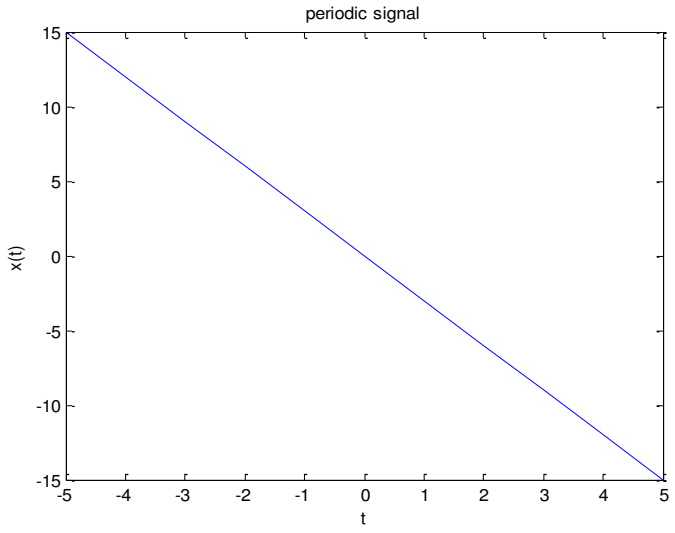
x2(t) = -3*t , -5 < t < 5

Program:-

```
t=-5:1:5
x=-(3*t)
plot(t,x)
xlabel('t')
ylabel('x(t)')
title('periodic signal')

y=abs(x)
m=y.^2
e=-5:1:5
r=trapz(e,m)
Energy=r/10
```

Result:-



Energy=76.500

EXPERIMENT NO:-5

Aim:- Write a program to calculate the convolution sum of two discrete time signals which involves folding, shifting, multiplication and summation. Also plot the output signal. Verify your result using 'conv' command.

(I). x[n] = [-2 0 1 -1 3] and h[n]= [1 2 0 -1]

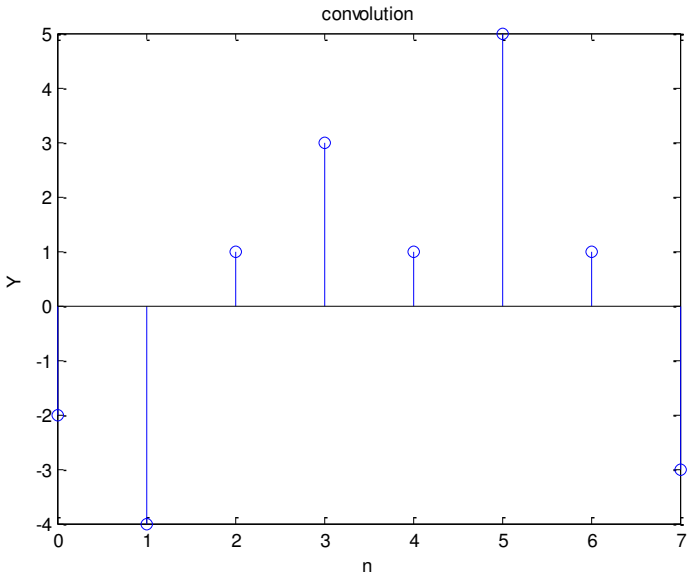
Program:-

```
x=[-2,0,1,-1,3];
h=[1,2,0,-1];
 y=conv(x,h);
stem(y)
```

OR

```
 x=[-2 0 1 -1 3];
 h=[1 2 0 -1];
N=length(x)-1
M = length(h)-1
 for n=0:M+N,
Y(n+1)=0;
for k=0:M,
x_index = n-k;
if (and(x_index >= 0, x_index <= N))
Y(n+1) = h(k+1)*x(x_index+1) + Y(n+1);
end
end
end
n=0:N+M;
stem(n,Y);
```

Result:-



(II) x[n] = u[n] and h[n]= e^{-an} u[n].

Proram:-

```
n=0:1:10;
 h=[ones(1,10)];
x=exp(-n1.*2);
N=length(x)-1
M = length(h)-1
Y(n+1)=0;
for k=0:M,
x_index = n-k;
if (and(x_index >= 0, x_index <= N))
 Y(n+1) = h(k+1)*x(x_index+1) + Y(n+1);
end
end
end
disp(Y);
y=conv(x,h);
disp(y);
n=0:N+M;
stem(n,Y);
```

EXPERIMENT NO-6

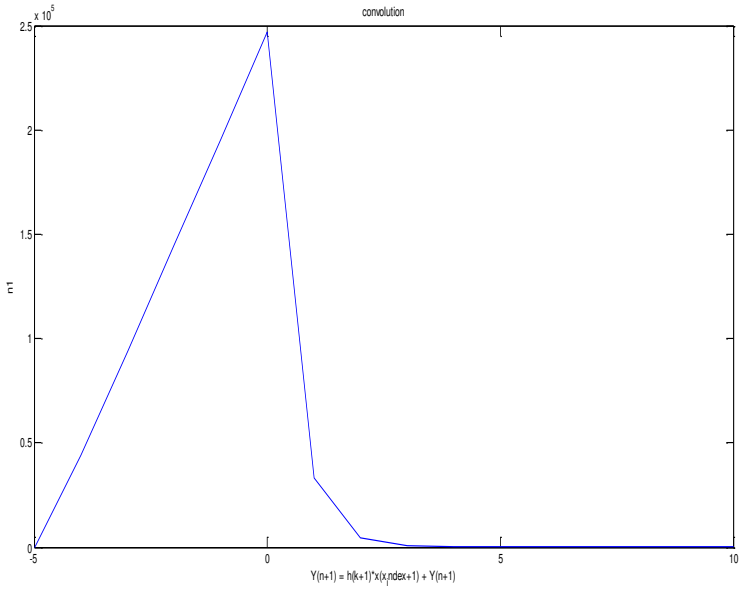
Aim:- Write a program to calculate the convolution integral of two continuous time signals which involves folding, shifting, multiplication and integration. Also plot the output signal.

(I) x[t] = 2t.u (t) and h[t]= e^{-2t}

%convolution using the mathematical formula for convolution that includes folding, shifting , multiplication , summation

Program:-

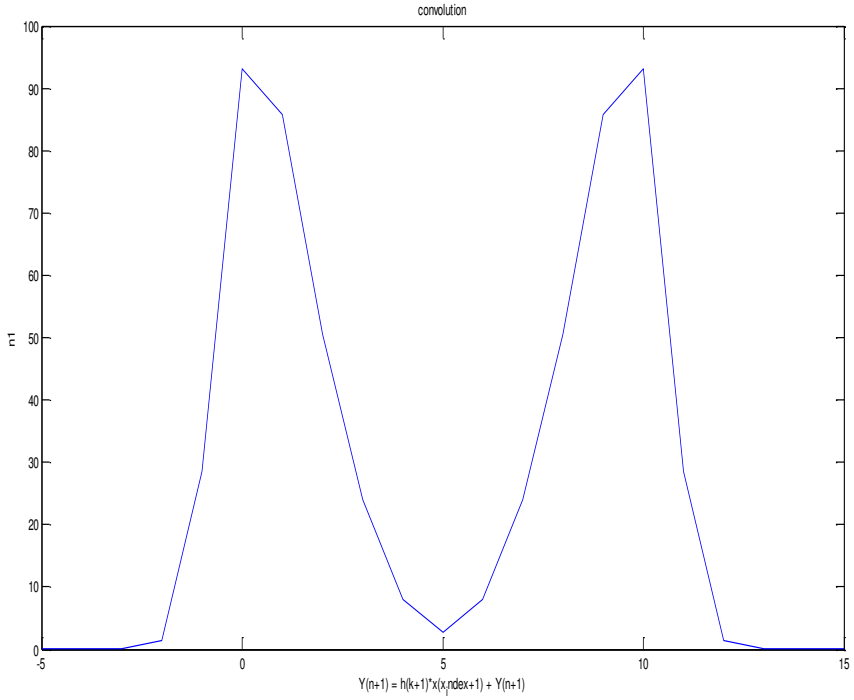
```
t1=-5:1:-1;
t2=0:1:5;
t=[t1 t2];
h=2*t2;
x=exp(-t.*2);
N=length(x)-1;
M = length(h)-1
for n=0:M+N,
Y(n+1)=0;
for k=0:M,
x_index = n-k;
if (and(x_index >= 0, x_index <= N))
Y(n+1) = h(k+1)*x(x_index+1) + Y(n+1)
end
end
end
 n=0:N+M;
stem(n,Y);
n1=(n-5);
plot(n1,Y);
```



(II) x[t] = exp(-t^2) and h[t]= 3t^2 .

Program:-

```
 t1=-5:1:-1;
t2=0:1:5;
t=[t1 t2];
h=3*t.^2;
x=exp(-t.^2);
N=length(x)-1;
 M = length(h)-1
for n=0:M+N,
Y(n+1)=0;
for k=0:M,
x_index = n-k;
if (and(x_index >= 0, x_index <= N))
Y(n+1) = h(k+1)*x(x_index+1) + Y(n+1);
end
end
end
 n=0:N+M;
stem(n,Y);
n1=(n-5);
plot(n1,Y);
```


convolution
100
90
80
70
60
50
40
30
20
10
0
n1
-5
0
5
10
15
Y(n+1) = h(k+1)*x(xndex+1) + Y(n+1)

EXPERIMENT NO:-7

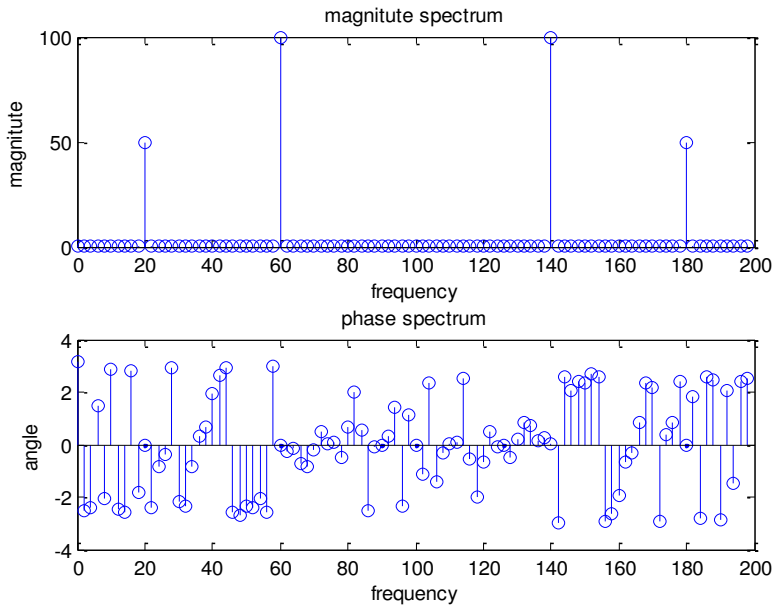
Aim:-Write a program to evaluate Fourier Transform of x(t) = 2*sin(2*pi*50*t) + 3*sin(2*pi*120*t); numerically and using the DFT. Plot the signal and its magnitude & phase over the frequency range -20 < f < 20 .

Program:-

```
clc;
N=0:99;
fs=200;
Ts=1/fs;
x = 2*sin (2*pi*50*t) + 3*sin (2*pi*120*t);
X=fft(x);
m=0:length(X)-1;
subplot(2,1,1)
stem(m*fs/length(X),abs(X));
title('magnitude spectrum')
xlabel('frequency')
ylabel('magnitude')

subplot(2,1,2)
stem(m*fs/length(X),angle(X));
title ('phase spectrum')
xlabel('frequency')
ylabel('angle')
```

Result:-



EXPERIMENT NO:-8

Aim:-. Write a program to evaluate the Laplace Transform of continuous time signal x(t) and h(t) . Finding the ROC of the transform.

a). x[t] = 2t.u (t)

Program:-

```
clc
syms t:
x=2*t*heaviside(t);
k=laplace(x);
```

Result:-

```
2/s^2
```

b). h[t]= e^{-2t}

Program:-

```
clc
syms t:
h=exp(-2*t);
k=laplace(h);
```

Result:-

```
1/(s+2)
```

EXPERIMENT NO:-9

Aim:- Write a program to evaluate the Z-Transform of a discrete-time signal x(n) and h(n) . Also Find ROC of the transform. Check stability of the system using .

a). x[n] = u[n]

Program:

```
syms n z a;
x=heaviside(n)
ztrans(x,z)
```

Result:-

1/(z - 1) + 1/2

b). h[n]= e^{-an} u[n].

Program:-

```
syms n z a;
x=exp(-a*n)*heaviside(n)
ztrans(x,z)
```

Result:-

1/ (z*exp(a) - 1) + 1/2