JNTUH COLLEGE OFENGINEERING

KUKATPALLY,HYDERABAD-500 085

CERTIFICATE

Certified that the is the bonafide record of the practical work done during the academic year**: 2014 – 2015** by

Name: **GORLA PRAVEEN**

Roll number : **1301143449** class: B.TECH **2** ND YEAR **1**st SEM

In the laboratory of: **BASIC SIMULATION LAB**

Of the department of:ELECTRONICS & COMMUNICATION ENGINEERING

Signature of the staff member:……..

Signature of Head Of Department:……..

Date of Examination:……..

Signature of Examiner/s:……

EXPERIMENT NO:1

NAME OF THE EXPERIMENT:BASIC OPERATIONS ON MATRICES

1.AIM:GENERATION OF MATRIX:

Theory: Introduction to Matrices in Matlab

A basic introduction to defining and manipulating matrices is given here. It is assumed that you know the basics on how to define and manipulate vectors (*Introduction to Vectors in Matlab*) using matlab.

- *Defining Matrices*
- *Matrix Functions*
- *Matrix Operations*

Defining Matrices

Defining a matrix is similar to defining a vector (*Introduction to Vectors in Matlab*). To define a matrix, you can treat it like a column of row vectors (note that the spaces are required!):

```
>> A = [ 1 2 3 ; 3 4 5; 6 7 8]

A=  1     2     3

    3     4     5

    6     7     8
```

2.AIM:operation on matrices:

a=[1 3 4;2 5 1;6 9 5]

a =

1 3 4

2 5 1

6 9 5

b=[2 8 5;4 8 2;8 0 3];

a+b

ans =

3 11 9

6 13 3

14 9 8

a-b

ans =

-1 -5 -1

-2 -3 -1

-2 9 2

a*b

ans =

46 32 23

32 56 23

88 120 63

a.*b

ans =

2 24 20

8 40 2

48 0 15

a/b

ans =

0.9167 -0.5417 0.1667

-0.0167 0.6417 -0.0667

0.5833 0.5417 0.3333

a./b

ans =

0.5000 0.3750 0.8000

0.5000 0.6250 0.5000

0.7500 Inf 1.6667

a\b

ans =

0.4545 -6.7273 -1.6136

0.6364 4.1818 0.8409

-0.0909 0.5455 1.0227

a.\b

ans =

2.0000 2.6667 1.2500

2.0000 1.6000 2.0000

1.3333 0 0.6000

a^2

ans =

31 54 27

18 40 18

54 108 58

a.^2

ans =

1 9 16

4 25 1

36 81 25

a'

ans =

1 2 6

3 5 9

4 1 5

a.'

ans =

1 2 6

3 5 9

4 1 5

3.AIM:**EXTRACTION OF MATRICES**

b(1:1,1:3)

ans = 2 8 5

b(1:3,1:1)

ans 2

4

8

b(1:2,1:2)

ans =

2 8

4 8

b(2:3,2:3)

ans =

8 2

0 3

inv(b)

ans =

-0.1000 0.1000 0.1000

-0.0167 0.1417 -0.0667

0.2667 -0.2667 0.0667

size(b)

ans =

3 3

det(b)

ans =

-240

4.AIM:SOLVING LINEAR EQUATIOS

Theory:

Solving Linear Equations:-

To find a particular solution of a linear equation in a Galois field, use the \ or / operator on Galois arrays. The table below indicates the equation that each operator addresses, assuming that A and B are previously defined Galois arrays. OperatorLinear EquationSyntaxEquivalent Syntax Using \Backslash (\)A * x = Bx = A \ BNot applicable Slash (/)x * A = Bx = B / Ax = (A'\B')'

Operator	Linear Equation	Syntax	Equivalent Syntax Using \
Backslash (\)	`A * x = B`	`x = A \ B`	Not applicable
Slash (/)	`x * A = B`	`x = B / A`	`x = (A'\B')'`

- The results of the syntax in the table depend on characteristics of the Galois array A:
- If A is square and nonsingular, the output x is the unique solution to the linear equation.
- If A is square and singular, the syntax in the table produces an error.
- If A is not square, MATLAB attempts to find a particular solution. If A'*A or A*A' is a singular array, or if A is a tall matrix that represents an overdetermined system, the attempt might fail.

Note : An error message does not necessarily indicate that the linear equation has no solution. You might be able to find a solution by rephrasing the problem. For example, gf([1 2; 0 0],3) \ gf([1; 0],3) produces an error but the mathematically equivalent gf([1 2],3) \ gf([1],3) does not. The first syntax fails because gf([1 2; 0 0],3) is a singular square matrix.

MATLAB CODE:

```
a =

    3    4   -2    2

    4    9   -3    5

   -2   -3    7    5

    1    4    6    7
```

b=[2;8;10;2];

x=inv(a)*b

```
x =

   -2.4000

   -2.0857

   -3.7429

    5.0286

a =

    2    8    5

    4    8    2

    8    0    3
```

e=eig(a)

```
e =

  -4.5172

  13.6149

   3.9023
```

AIM: TO DETERMINE SUBMATRIX

MATLAB CODE

```
Clc
a=[ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8 ;5 6 7 8 9]
b=a(2:4,2:3)
```

OUTPUT:

```
a =

   1   2   3   4   5

   2   3   4   5   6

   3   4   5   6   7

   4   5   6   7   8

   5   6   7   8   9

b =   3   4

      4   5

      5   6
```

AIM: TO DETERMINE INVERSE OF A MATRIX

MATLAB CODE:

```
Clc

a=[ 1 2 3; 4 1 1;1 3 2];
```

b=inv(a)

OUTPUT:

a = 1 2 3

4 1 1

1 3 2

b = -0.0556 0.2778 -0.0556

-0.3889 -0.0556 0.6111

0.6111 -0.0556 -0.3889

AIM: TO FIND DETERMINANT OF A MATRIX

MATLAB CODE

clc
a=[1 2 3; 4 1 1;1 3 2]
b=det(a)

OUTPUT:

a =

1 2 3

4 1 1

1 3 2

b = 18

AIM: TO DETERMINE EIGEN VALUES OF A MATRIX

MATLAB CODE

```
clc
a=[ 1 2 3; 4 1 1;1 3 2]
b=eig(a)
```

OUTPUT:

a =

1 2 3

4 1 1

1 3 2

b =

6.0000

-1.0000 + 1.4142i

-1.0000 - 1.4142i

AIM: CONCATINATION OF A MATRIX

THEORY:

Matrix concatenation is the process of joining one or more matrices to make a new matrix. The brackets [] operator discussed earlier in this section serves not only as a matrix constructor, but also as the MATLAB concatenation operator. The expression C = [A B] horizontally concatenates matrices A and B. The expression C = [A; B] vertically concatenates them.

This example constructs a new matrix C by concatenating matrices A and B in a vertical direction:

```
A = ones(2, 5) * 6;       % 2-by-5 matrix of 6's

B = rand(3, 5);           % 3-by-5 matrix of random values

C = [A; B]                % Vertically concatenate A and B
```

Output:

```
C =

   6.0000   6.0000   6.0000   6.0000   6.0000

   6.0000   6.0000   6.0000   6.0000   6.0000

   0.9501   0.4860   0.4565   0.4447   0.9218

   0.2311   0.8913   0.0185   0.6154   0.7382

   0.6068   0.7621   0.8214   0.7919   0.1763
```

MATLAB CODE:

```
clc
a=[ 1 2 3; 4 1 1;1 3 2;7 5 3]
b=[a a+9 a+4]
```

OUTPUT

```
a =

   1   2   3

   4   1   1
```

```
   1   3   2

   7   5   3

b =

   1   2   3   10   11   12   5   6   7

   4   1   1   13   10   10   8   5   5

   1   3   2   10   12   11   5   7   6

   7   5   3   16   14   12   11   9   7
```

AIM: TO DELETE A ROW FROM A MATRIX

MATLAB CODE

```
clc

a=[ 1 2 3; 4 1 1;1 3 2;7 5 3]

a(2,:)=[]
```

OUTPUT

```
a =
   1   2   3
   4   1   1
   1   3   2
   7   5   3

a =
   1   2   3
   1   3   2
   7   5   3
```

AIM: TO DELETE A COLUMN FROM A MATRIX

MATLAB CODE

```
clc

a=[ 1 2 3; 4 1 1;1 3 2;7 5 3]
a(:,2)=[]
```

OUTPUT

```
a =
```

```
    1    2    3

    4    1    1

    1    3    2

    7    5    3

a =

    1    3

         4         1
         1         2

         7    3
```

EXPERIMENT NO:2

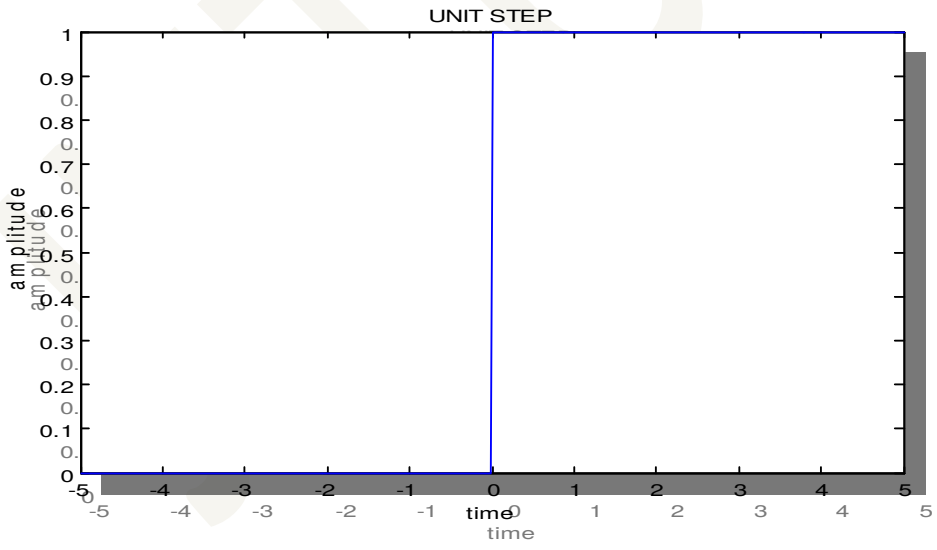
NAME OF THE EXPERIMENT :Generation of various signals

AIM:TO GENERATE UNIT STEP

MATLABCODE:

```
t=-5:.01:5;

a=[zeros(1,500) ones(1,501)];
plot(t,a);
xlabel('time')
ylabel('amplitude')
title('UNIT STEP')
```

FIGURE:

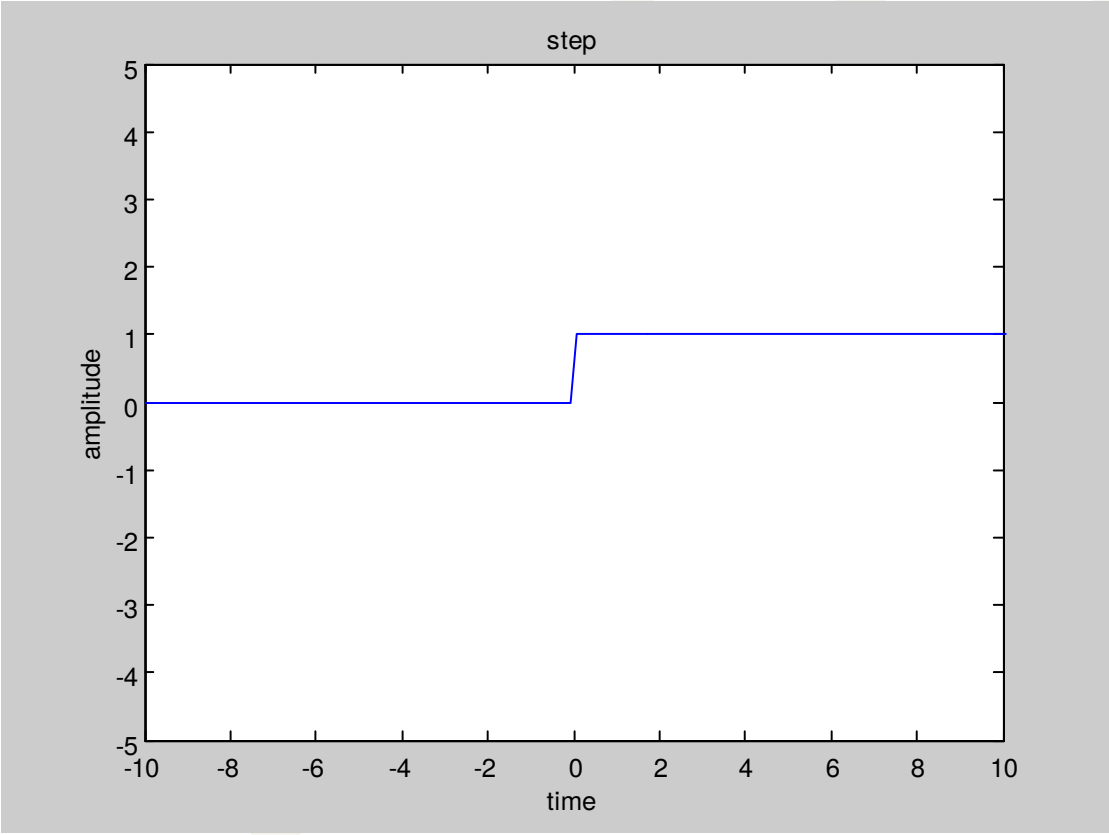


AIM:

UNIT STEP USING HEAVISIDE

```
%unitstep
clear all;
syms t;
x=heaviside(t);
ezplot(x,[-20 20]);
axis([-10 10 -5 5]);
xlabel('time');
ylabel('amplitude');
title('step');
```

OUTPUT:

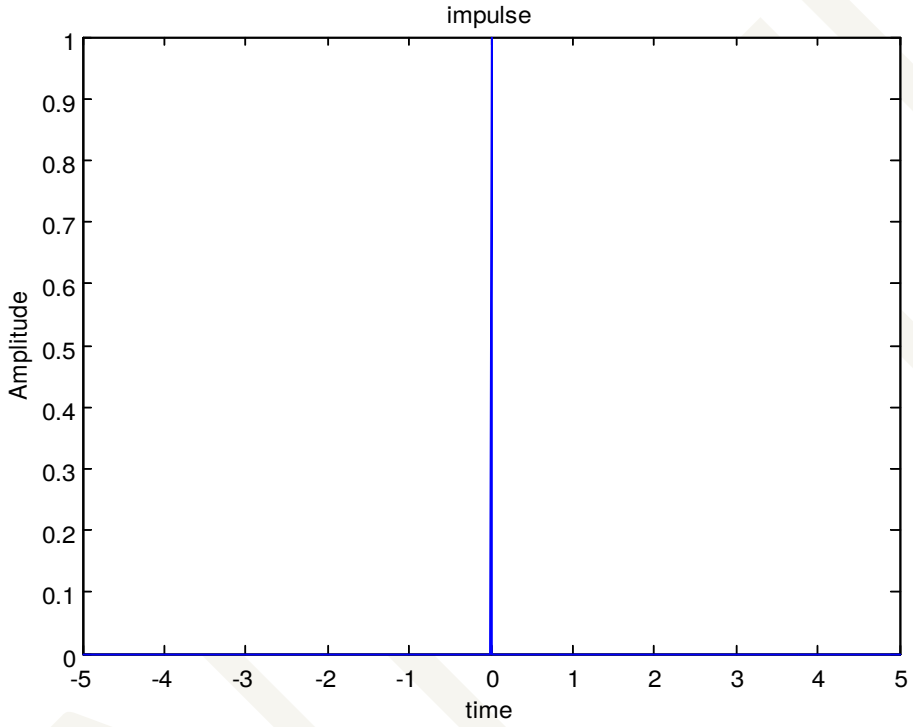


AIM:TO GENERATE UNIT IMPULSE

MATLABCODE:

```
t=-5:.01:5;
y=[zeros(1,500) ones(1) zeros(1,500)];
xlabel('time');
ylabel('Amplitude');
title('impulse');
plot(t,y);
```

FIGURE:



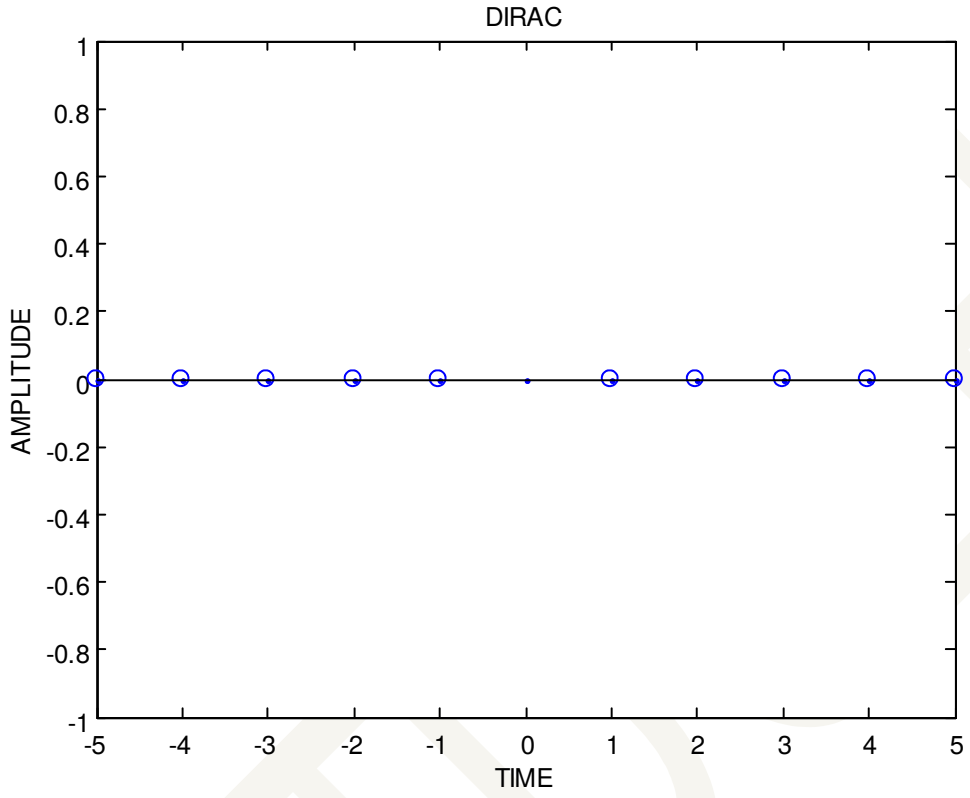
AIM:

UNIT IMPULSE USING DIRAC FUNCTION

MATLAB CODE:

```
clear all
t=-5:1:5
x=dirac (t);
xlabel('time')
ylabel('amplitude')
title('dirac')
stem(t,x)
```

OUTPUT :



AIM:

% IMPULSE USING HEAVISIDE

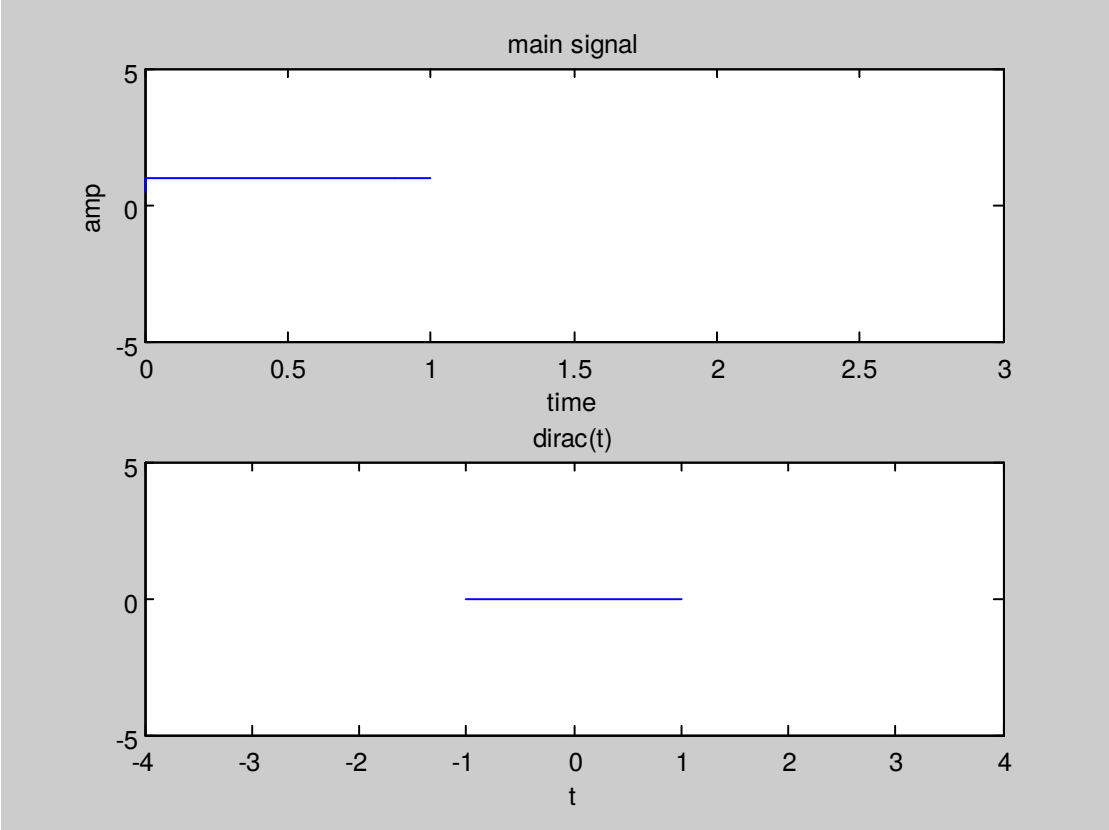
MATLAB CODE:

```
clear all;
syms t;
x=input('x1=');
t=input('time period =');
t1=input('tmin=');
t2=input('tmax=');
e=diff(x);
subplot(211);
ezplot(x,[-1 1]);
axis([0 3 -5 5]);
xlabel('time');
ylabel('amp');
title('main signal');
subplot(2,1,2);
ezplot(e,[-1 1]);
axis([-4 4 -5 5]);
```

INPUT:

x1=HEAVISIDE(t)
t1=0
t2=10

output:

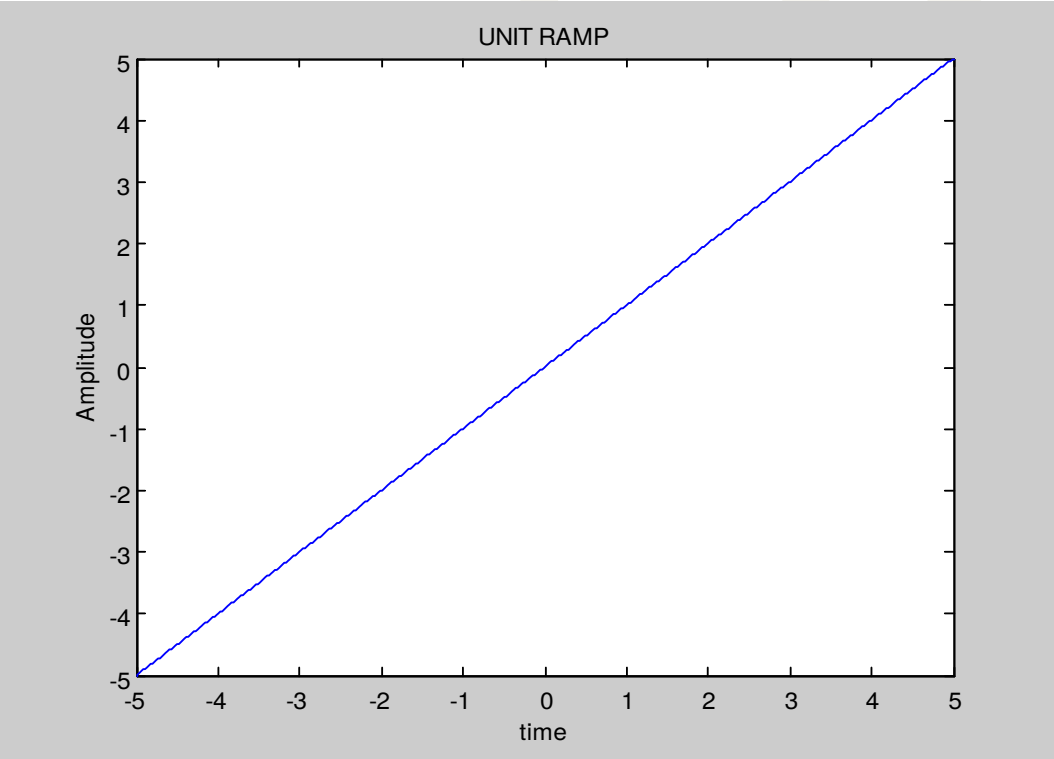

main signal
amp
time
dirac(t)
t

AIM:TO GENERATE UNIT RAMP

MATLAB CODE:

```
t=0:.01:5;
y=t;

plot(t,y);

xlabel('time');
ylabel('Amplitude');
title('impulse');
```

FIGURE:



AIM:

%ramp function using heavide

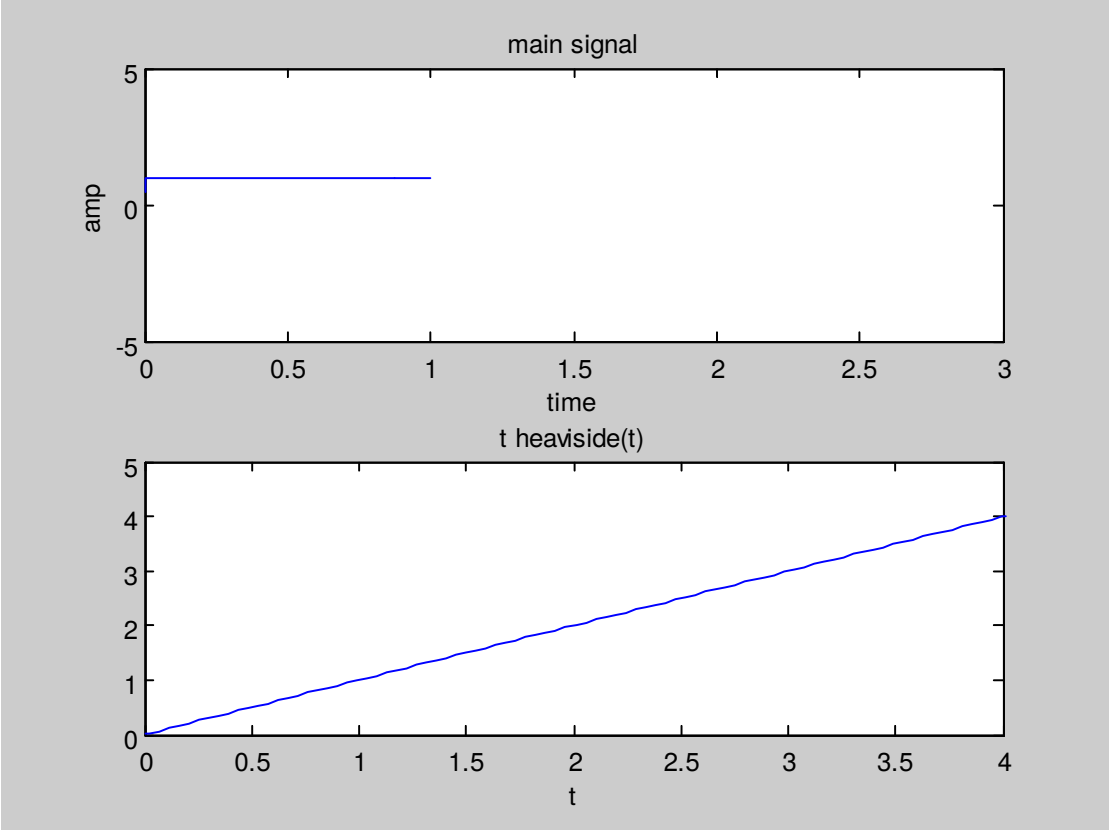
MATLAB CODE:

```
clear all;
syms t;
x=input('x1=');
t=input('time period =');
t1=input('tmin=');
t2=input('tmax=');
e=int(x);
subplot(211);
ezplot(x,[-1 1]);
axis([0 3 -5 5]);
xlabel('time');
ylabel('amp');
title('main signal');
subplot(2,1,2);
ezplot(e,[-10 10]);
axis([0 4 0 5]);
```

INPUT:

X1=heaviside(t)
t1=0
t2=10

OUTPUT:

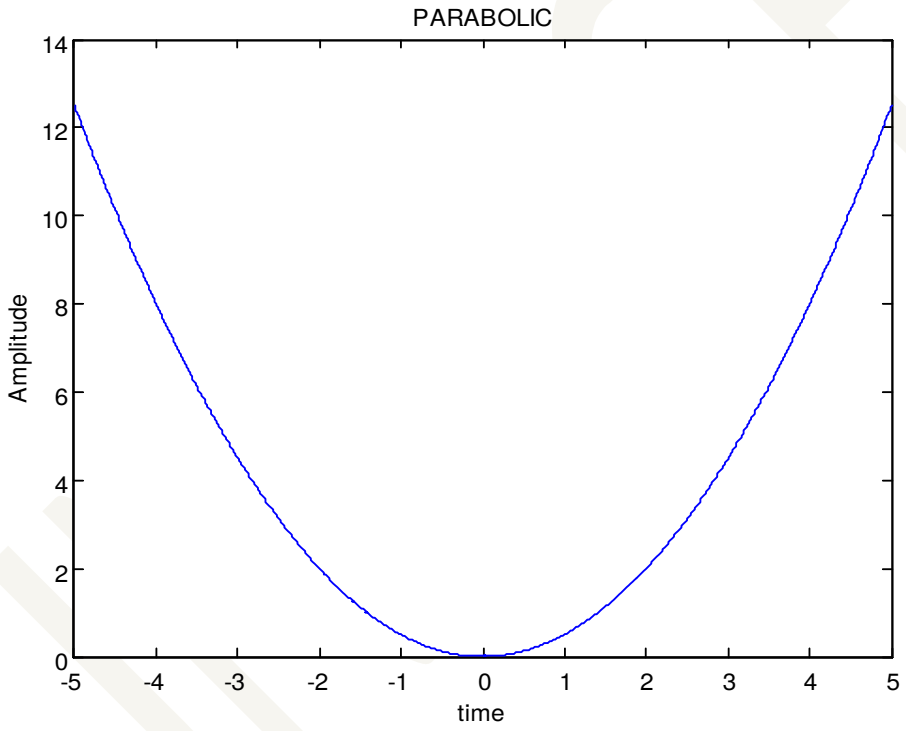

main signal
amp
time
t heaviside(t)
t

AIM:TO GENERATE PARABOLIC FUNCTION

MATLAB CODE:

```
t=-5:.01:5;
y=t.^2/2;

plot(t,y);

xlabel('time');
ylabel('Amplitude');
title('parabolic');
```

FIGURE:



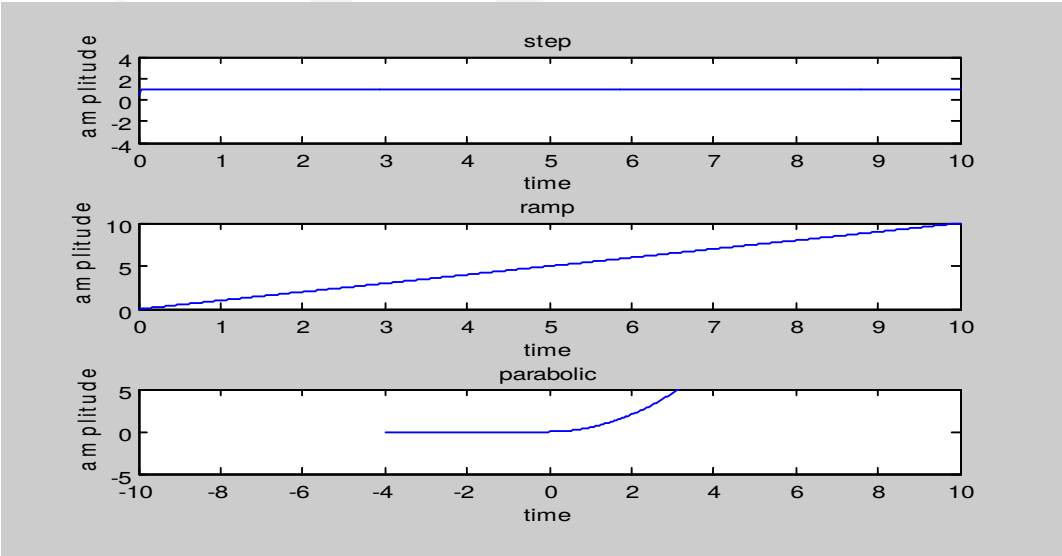
AIM:

PARABVOLIC FUNCTION USING HEAVISIDE

MATLAB CODE:

```
clear all;
syms t;
x=heaviside(t);
subplot(311);
ezplot(x,[0 10]);
axis([0 10 -4 4]);
xlabel('time');
ylabel('amplitude');
title('step');
y=int(x);
subplot(312);
ezplot(y,[0 10]);
axis([0 10 0 10]);
xlabel('time');
ylabel('amplitude');
title('ramp');
z=int(y);
subplot(313);
ezplot(z,[-4 4]);
axis([-10 10 -5 5]);
xlabel('time');
ylabel('amplitude');
title('parabolic');
```

OUTPUT:

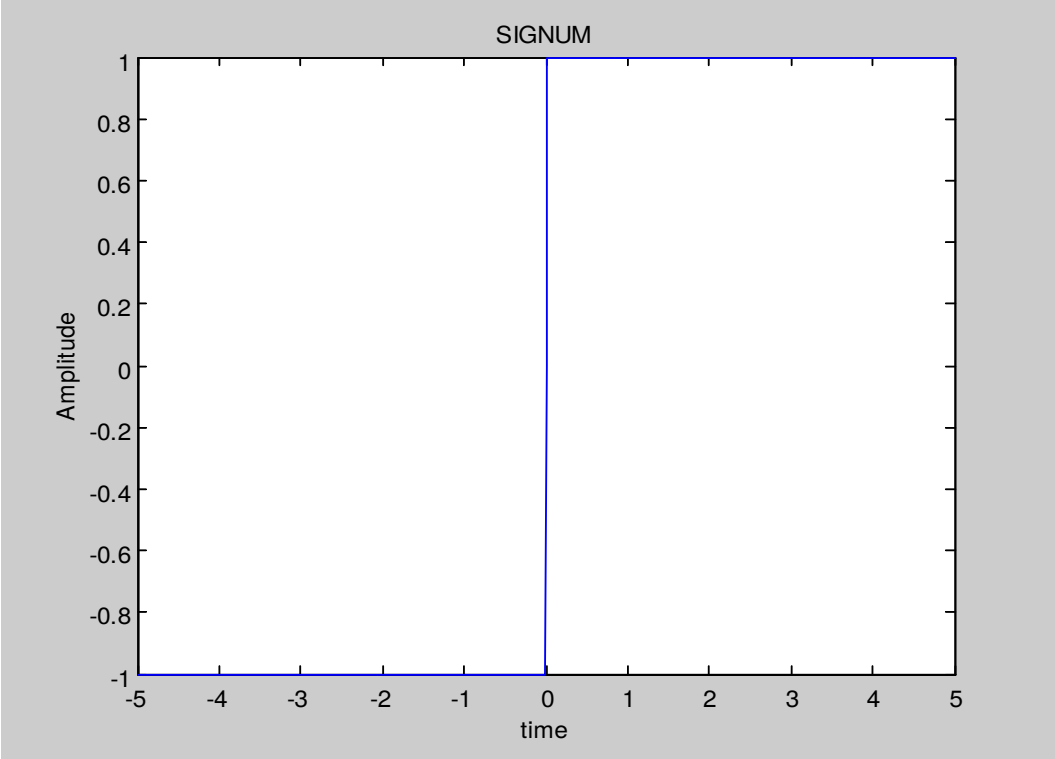


AIM:TO GENERATE SIGNUM FUNCTION

MATLAB CODE:

```
t=-5:.01:5;
y=sign(t);

plot(t,y);

xlabel('time');
ylabel('Amplitude');
title('SIGNUM');
```

FIGURE:



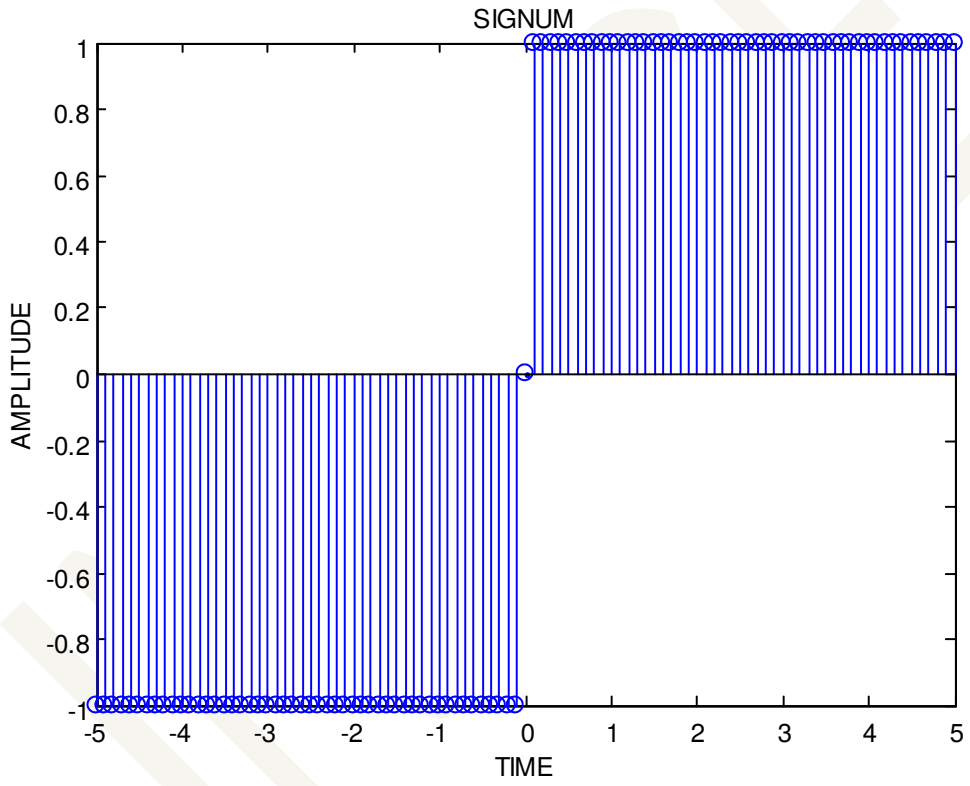
AIM:

SIGNUM FUNCTION USING HEAVISIDE

MATLAB CODE:

```
t=-5:0.1:5
a=heaviside(t)-heaviside(-t)
xlabel('time')
ylabel('amplitude')
title('signum')
STEM(t,a)
```

OUTPUT :

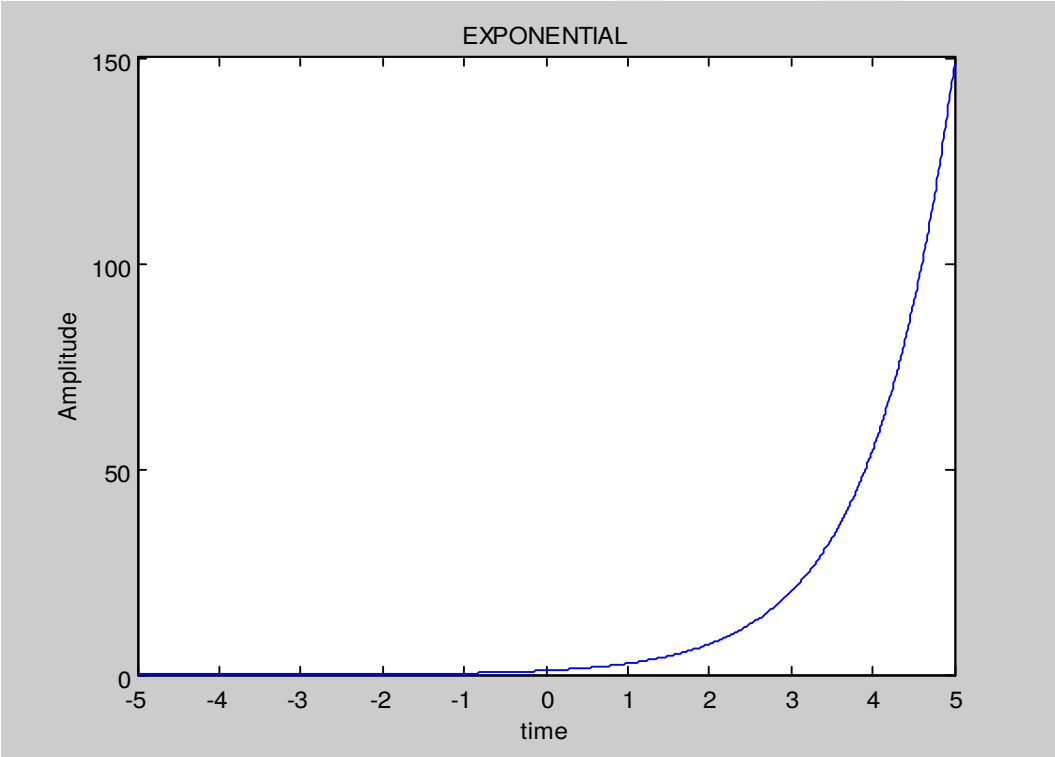


AIM:TO GENERATE EXPONENTIAL FUNCTION

MATLAB CODE:

```
t=-5:.01:5;
y=exp(t);

plot(t,y);

xlabel('time');
ylabel('Amplitude');
title('EXPONENTIAL');
```

FIGURE:



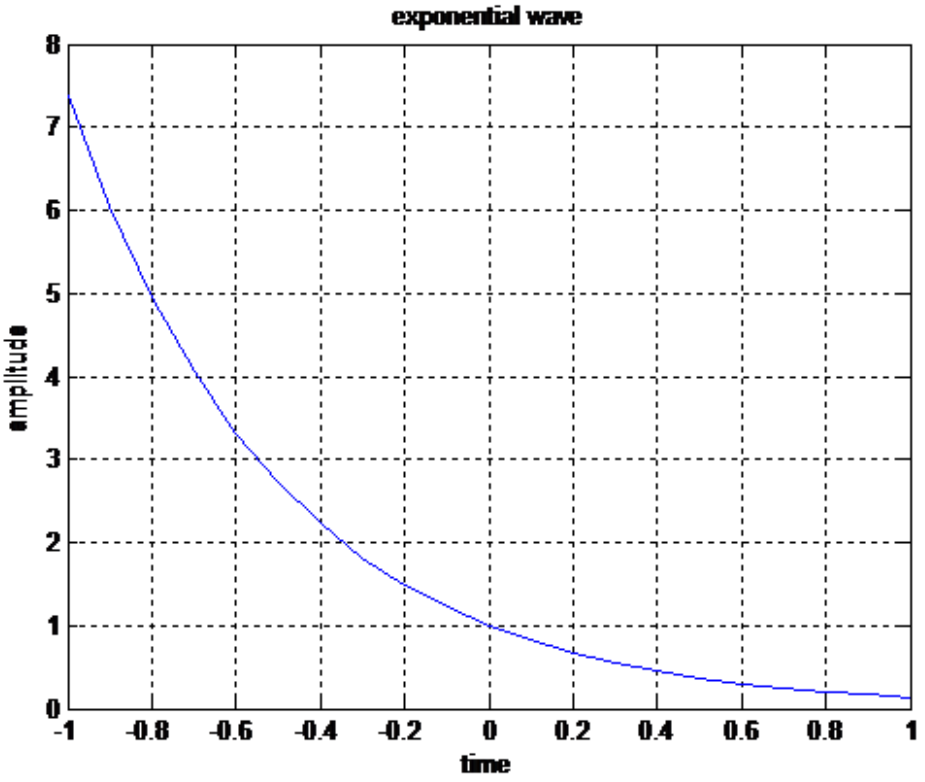
AIM :

GENERATION OF EXPONENTIAL FUNCTION

MATLAB CODE:

```
clc
clearall
closeall
t=-1:0.1:1
a=input(' enter value of a ')
x=exp(-a*t)
plot(t,x)
xlabel(' time ')
ylabel(' amplitude ')
title(' exponential wave ')
grid on;
```

OUTPUT:

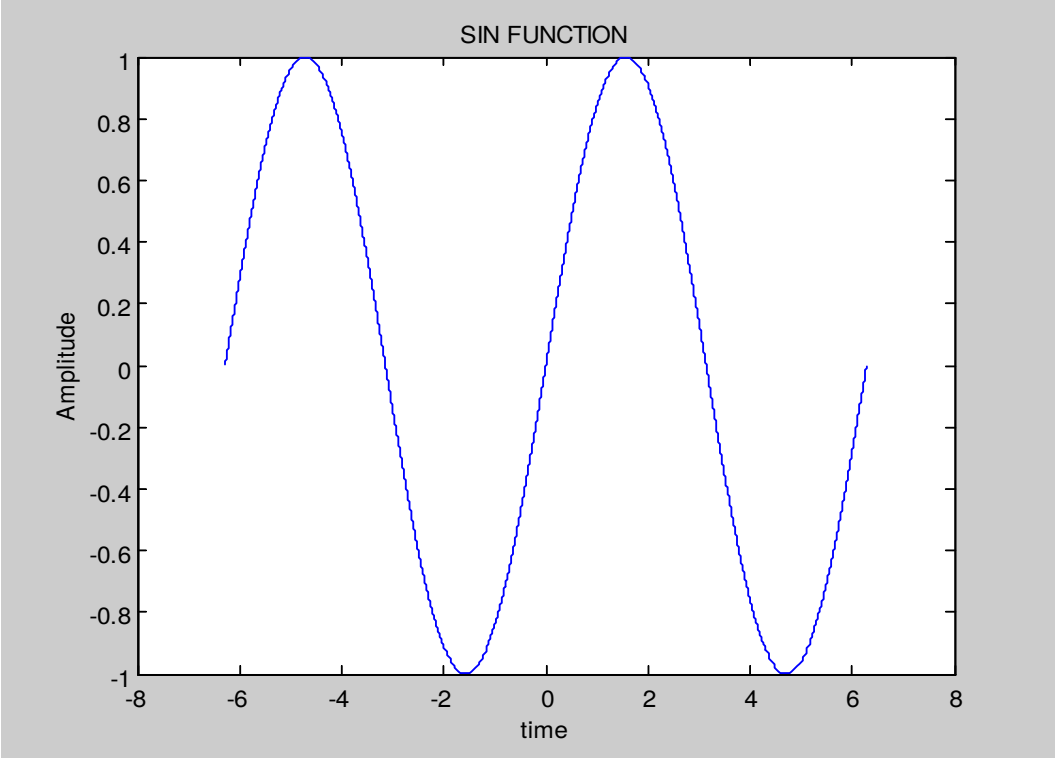


AIM:TO GENERATE SIN FUNCTION

MATLAB CODE:

```
t=-2*pi:.001:2*pi;
y=sin(t);

plot(t,y);

xlabel('time');
ylabel('Amplitude');
title('SIN FUNCTION')
```

FIGURE:

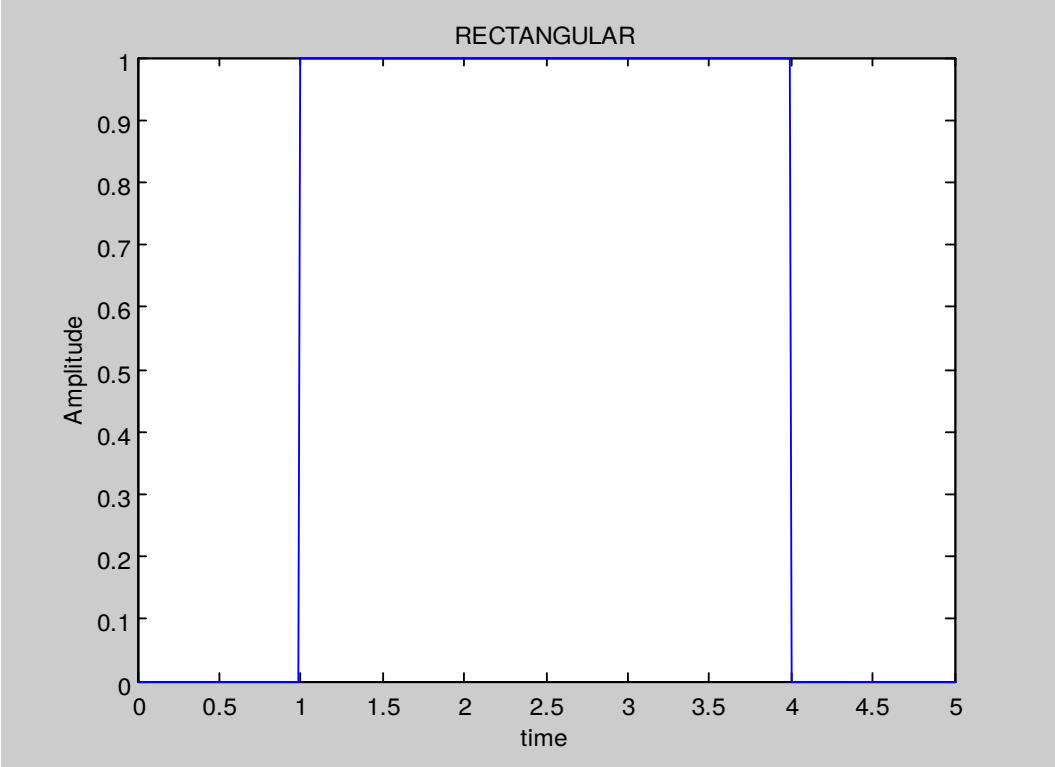


AIM:TO GENERATE RECTANGULAR FUNCTION

MATLAB CODE:

```
t=0:.01:5;
y=[zeros(1,100) ones(1,300) zeros(1,101)];

plot(t,y);

xlabel('time');
ylabel('Amplitude');
title('RECTANGULAR');
```

FIGURE:

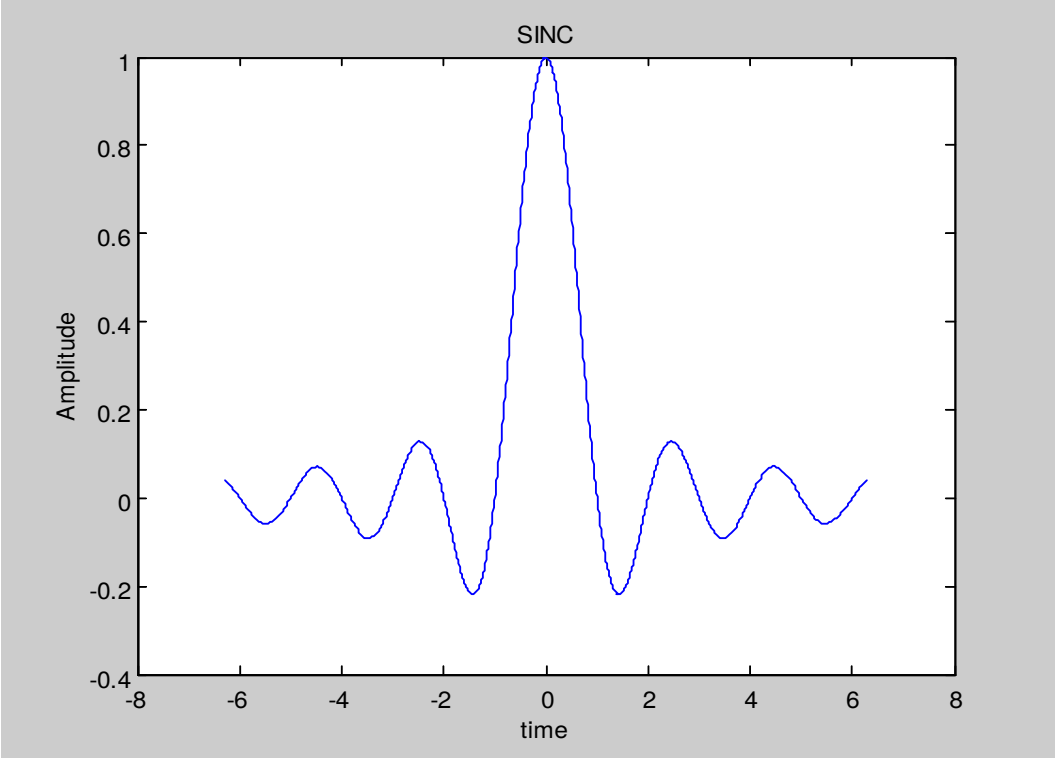


AIM:TO GENERATE SINC FUNCTION

MATLAB CODE:

```
t=-2*pi:.01:2*pi;
y=sinc(t);

plot(t,y);

xlabel('time');
ylabel('Amplitude');
title('SINC');
```

FIGURE:



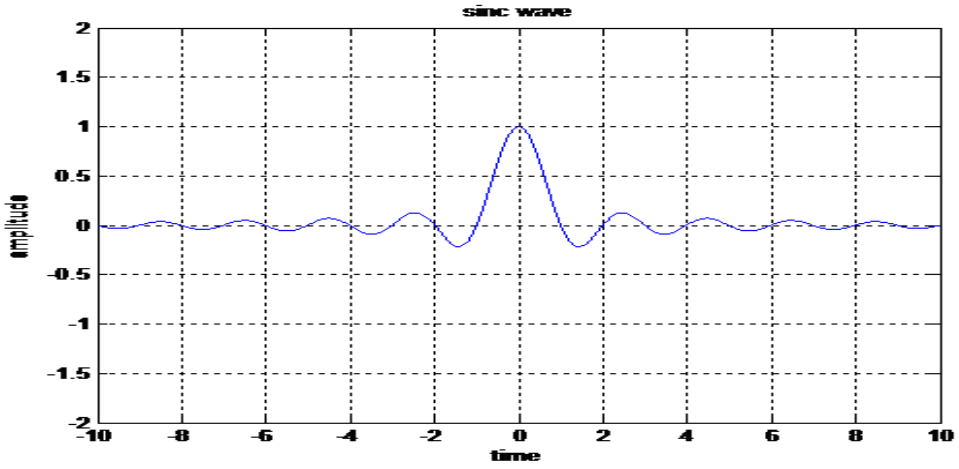
AIM : ***GENERATION OF SINC FUNCTION USING SINE FUNCTION***

MATLAB CODE:

```
%sinc wave
clc
clearall
closeall
t=-10:0.001:10
y=sin(pi*t)./(pi*t)
plot(t,y)
axis([-10 10 -2 2])
xlabel(' time ')
```

```
ylabel(' amplitude ')
title(' sinc wave ')
gridon;
```

OUTPUT:



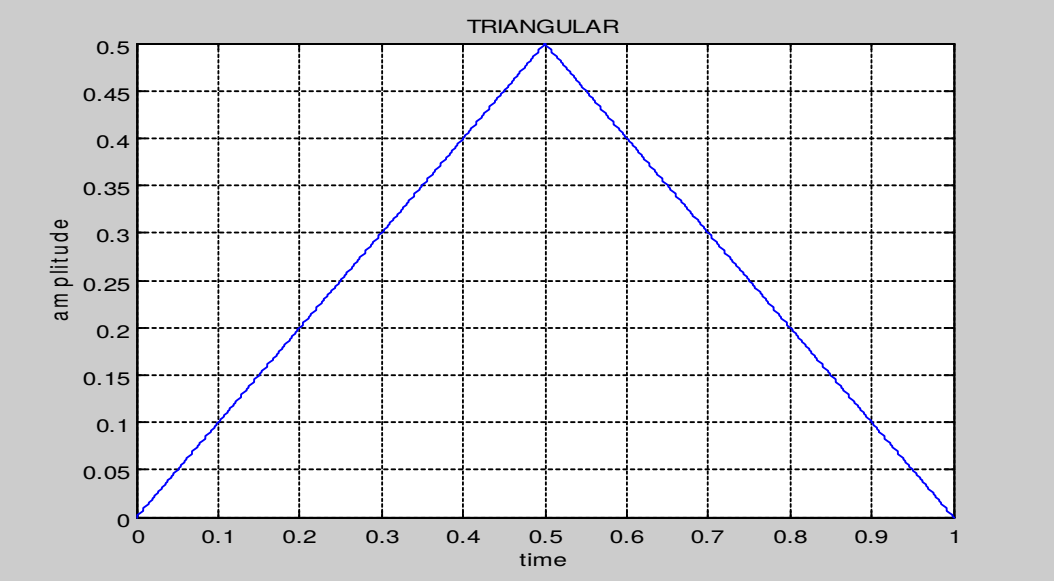
AIM:TO GENERATE TRIANGULAR FUNCTION

MATLAB CODE:

```
Clc

clear all

close all
t=0:0.001:1;
l=length(t);
for i=1:l;
if t(i)<.5
u(i)=t(i);
elseif t(i)>=.5
u(i)=1-t(i);
end
end
plot(t,u);
xlabel('time')
```

```
ylabel('amplitude')
title('TRIANGULAR')
```

FIGURE:

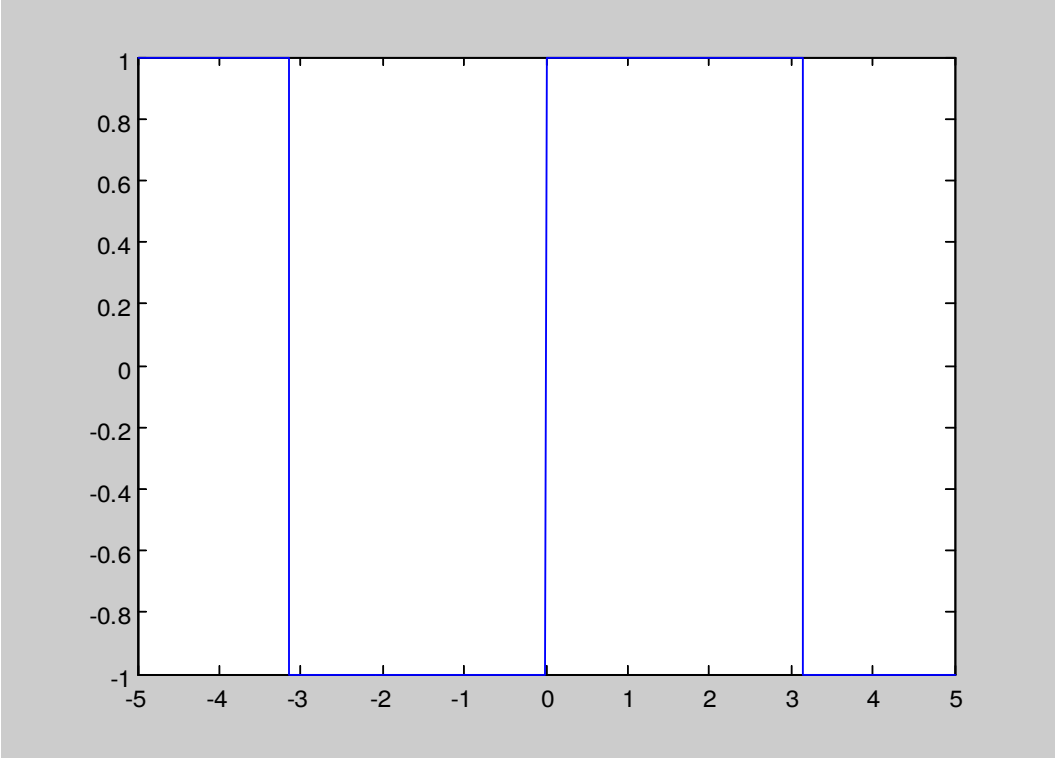


AIM:TO GENERATE SQUARE FUNCTION

MATLAB CODE:

```
t=-5:.001:5;
y=square(t);

xlabel('time');
ylabel('Amplitude');
title('SQUARE FNCTION');
plot(t,y);
```

FIGURE:



EXPERIMENT NO :3

NAME OF THE EXPERIMENT: SUM OF TWO SIGNALS

AIM:TO GENERATE SUM OF TWO SIGNALS

MATLAB CODE:

```
clear all;
      n=input('c=');
      t=-n:.01:n;
      a=input('enyer the first signal');
      b=input('enter the 2nd signal');
      out=a+b;
```

```
disp('out');
xlabel('time');
ylabel('amplitude');
title('SIGNAL 1');
subplot(3,3,1);
stem(t,a);
xlabel('time');
ylabel('amplitude');
title('SGNAL 2');

subplot(3,3,2);
stem(t,b);
xlabel('title');
ylabel('amplitude');
title('sum');
subplot(3,3,3);
stem(t,out);
```

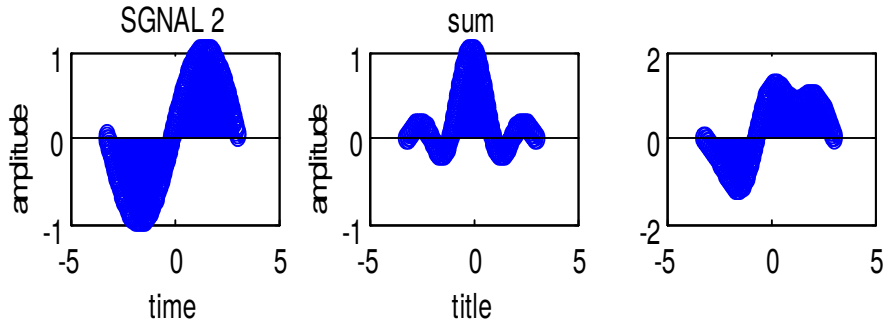
INPUT:

c=pi

enyer the first signalsin(t)

enter the 2nd signalsinc(t)

out



EXPERIMENT NO:

AIM:TO DETERMINE ADDITION OF 2 SIGNALS

MATLAB CODE:

In order to perform this task

We define a function to pad zeros

Function for padding zeros:

```
function[s1out s2out]=padding2(s1,m,s2,n)


 k=abs(min(m)-min(n))

if min(m)<min(n)
for i=1:k
      s2=[0 s2];
end
else
 for i=1:k
       s1=[0 s1];
 end
end

%pad to the max limits

 K1=abs(max(m)-max(n))

if max(m)<max(n)
for i=1:k1
      s1=[s1 0];
end
else
 for i=1:k1
       s2=[s2 0];
 end
end
```

AIM:TO PAD ZEROS FOR 2 SIGNALS.

MATLAB CODE:

```
clear all
```

```
clc
s1=[1 3 4 5 6 7];
m= [-2 -1 0 1 2 3];
s2=[5 8 9 ];
n=[0 1 2];
[s1 s2]=padding2(s1,m,s2,n);
 s1
 s2
```

calling funtion:

```
function[s1 s2 m1]=padding2(s1,m,s2,n)


 k=abs(min(m)-min(n))

if min(m)<min(n)
for i=1:k
      s2=[0 s2];
end
else
 for i=1:k
       s1=[0 s1];
 end
end

%pad to the max limits

 k1=abs(max(m)-max(n))

if max(m)<max(n)
for i=1:k1
      s1=[s1 0];
end
else
 for i=1:k1
       s2=[s2 0];
 end
end
m1=min(min(m),min(n)):max(max(m),max(n))
```

output:

s1=

1 3 4 5 6 7

s2=

0 0 5 8 9 0

EXPERIMENT NO:3

AIM:TO GENERATION ADDITION OF 2 SIGNALS

Matlab code:

```
clear all
clc
s1=input('');
m= [-2 -1 0 1 2 3];
s2=[5 8 9 ];
n=[0 1 2];
[s1 s2]=padding2(s1,m,s2,n);
 s1
 s2
s1+s2
```

output:

```
s1 =

    1    3    4    5    6    7


s2 =

    0    0    5    8    9    0


ans =

    1    3    9   13   15    7
```

METHOD : 2

AIM:IMPLEMENTATION OF TO GENERATION ADDITION OF 2 SIGNALS

MAIN CODE:

```
clc
clear all
s1=input('enter signal');
s2=input('enter signal');
[s]=sumof2signals(s1,s2) ;
s  ;
```

CALLING FUNCTION FOR "SUMOF2SIGNALS"

```
function[s]=sumof2signals(s1,s2)
 l1=length(s1);
m=[ ]
for i=0:1:l1-1
   m=[m i] ;
end;
l2=length(s2) ;
n=[ ]
for i=0:1:l2-1
   n=[n i] ;
end;
[s1 s2 m1]=padding2(s1,m,s2,n)
 s=s1+s2
end
```

INPUT:

```
enter signal:[1 2 3 4 5]
enter signal:[1 2 3 ]
```

OUTPUT:

```
s1 =
```

```
    1    2    3    4    5

s2 =

    1    2    3    0    0

m1 =

    0    1    2    3    4

s =

    1    4    9    0    0
```

AIM:MULTIPLICATION OF 2 SIGNALS

MATLAB CODE:

```
clear all
clc
s1=[1 3 4 5 6 7];
m= [-2 -1 0 1 2 3];

s2=[5 8 9 ];
n=[0 1 2];
[s1 s2]=padding2(s1,m,s2,n);

s1

s2

s1.*s2
```

OUTPUT:

```
s1 =


    1    3    4    5    6    7



s2 =


    0    0    5    8    9    0


ans =


    0    0   20   40   54    0
```

METHOD: 2
MAIN code:

```
clc
clear all
s1=input('enter signal');
s2=input('enter signal');
[s]=mulof2signals(s1,s2) ;
s  ;
```

INPUT:

```
enter signal:[1 2 3 4 5]
enter signal:[1 2 3 ]
```

OUTPUT:

```
s1 =

    1    2    3    4    5


s2 =

    1    2    3    0    0


m1 =

    0    1    2    3    4


s =

    1    4    9    0    0
```

CALLING FUNCTION FOR "MULOF2SIGNALS"

```
function[s]=mulof2signals(s1,s2)
 l1=length(s1);
m=[ ]
for i=0:1:l1-1
   m=[m i] ;
end;

l2=length(s2) ;
n=[ ]
for i=0:1:l2-1
   n=[n i] ;
end;
[s1 s2 m1]=padding2(s1,m,s2,n)
 s=s1+s2
end
```

EXPERIMENT:4

NAME OF THE EXPERIMENT: ODD AND EVEN PARTS OF A

SIGNAL

AIM:TO FIND THE ODD AND EVEN PART OF A SIGNAL

MATLAB CODE:

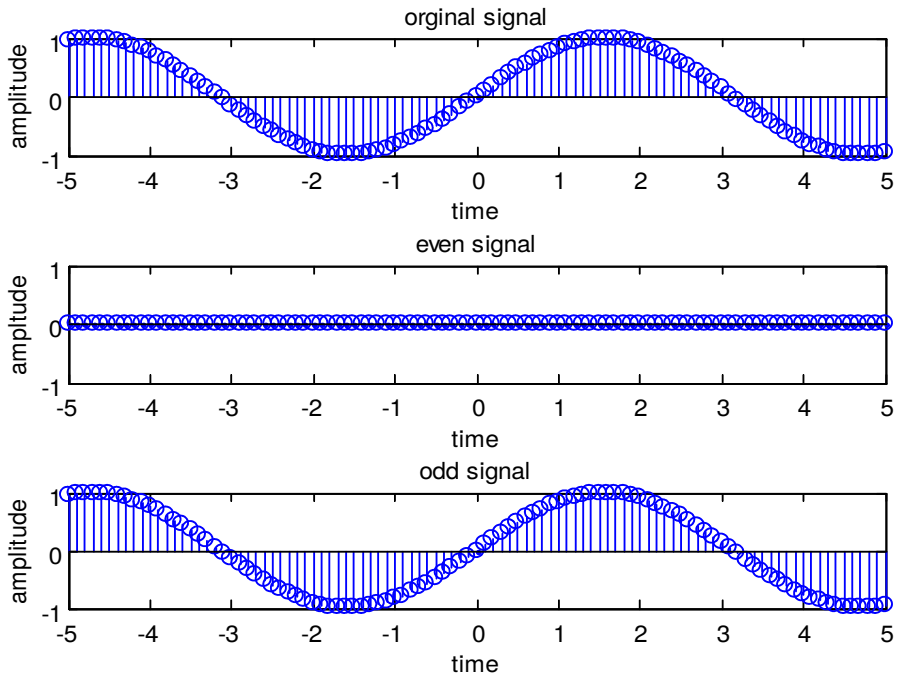
```
n=input('c=');
      t=-n:.1:n;
      a=input('enter the signal')
      xlabel('time');
      ylabel('amplitude');
      title('orginal signal');
      subplot(3,1,1);
      stem(t,a);
      b=fliplr(a);
      d=(a+b)*.5;
      xlabel('time');
      ylabel('ampltude');
      title('even signal');

      subplot(3,1,2);
      stem(t,d);
      e=(a-b)*.5;
      xlabel('time');
      ylabel('amplitude');
      title('odd signal');

      subplot(3,1,3);
      stem(t,e);
```

INPUT:

c=5

enter signalsin(t)

FIGURE:



EXPERIMENT NO:5

NAME OF THE EXPERIMENT: ORTHOGNALITY OF TWO SIGNALS

AIM:TO TEST THE ORTHOGONALITY OF TWO SIGNALS

MATLAB CODE:

```
claer all;
syms t
a=input('ist signal');
b=input('2nd=');
t1=input('tmin=');
t2=input('tmax=');
x1=int(a*b,t1,t2);
if(x1==0)
    disp('orthogonal');
else
    disp('not orthogonal');
end;
```

INPUT:

1st signalsin(2*pi*t)

2nd=cos(3*pi*t/2)

tmin=0

tmax=3

non orthogonal

EXPERIMENT NO:6

NAME OF THE EXPERIMENT: SHIFTING OF GENERAL SIGNALS

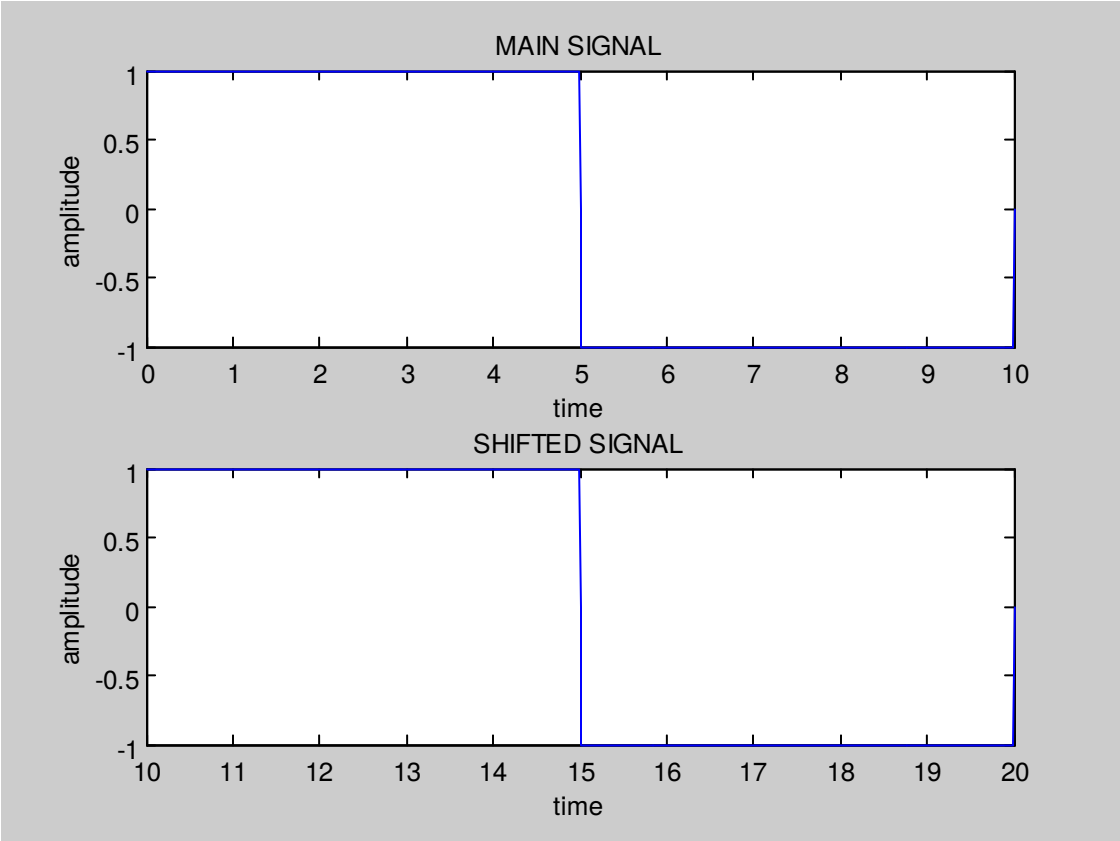
AIM:TO SHIFT A GIVEN SIGNAL

SIGNAL:1

MATLAB CODE:

```
clear all;
n1=input('min=');
n2=input('max=');
for i=1:2
n=input('for orginal n=0,if shifted n=');
if n==0;
t1=n1+n:.001:n2+n;
x=input('signal=');
a=x;
b=t1;
else
 t2=n1+n:.001:n2+n;
 t1=t2;
y=x;
end;
end;
subplot(2,1,1);
plot(b,a);
xlabel('time');
ylabel('amplitude');
title('MAIN SIGNAL');
subplot(2,1,2);
plot(t2,y);
xlabel('time');
ylabel('amplitude');
title('SHIFTED SIGNAL');
```

FIGURE:

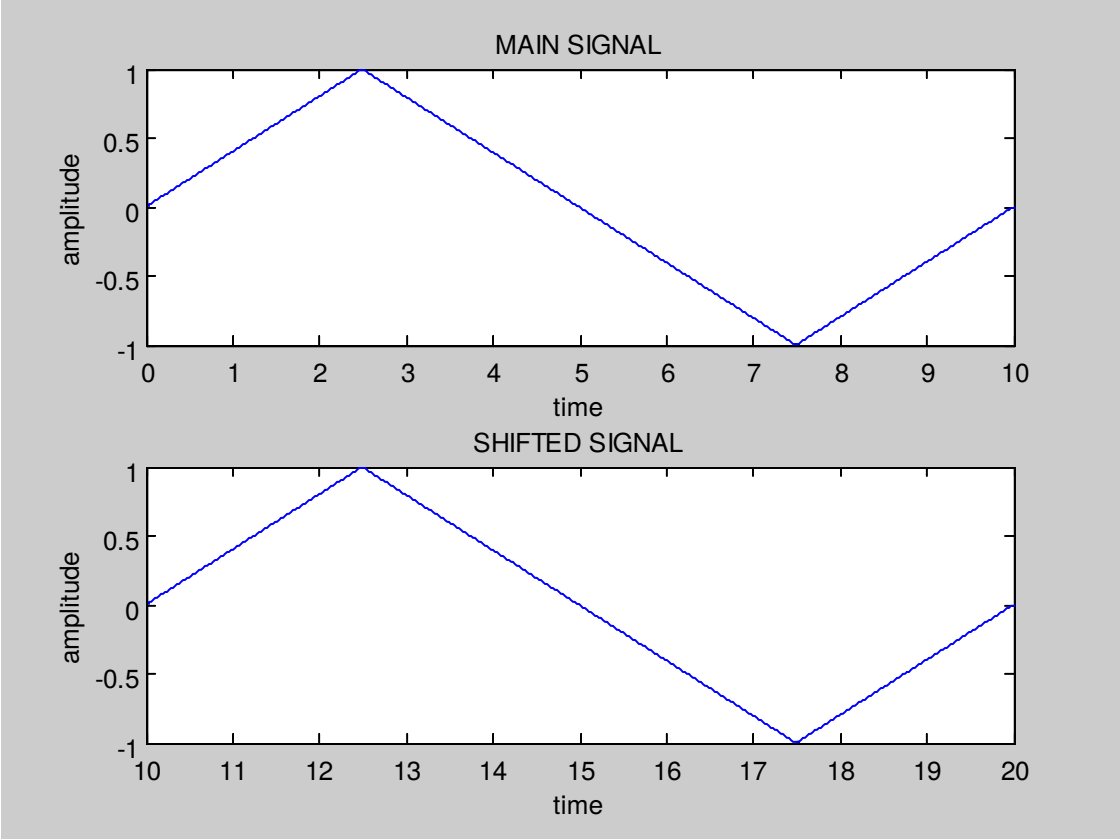


SIGNAL:2

MATLAB CODE:

```
clear all;
n1=input('min=');
n2=input('max=');
for i=1:2
n=input('for orginal n=0,if shifted n=');
if n==0;
t1=n1+n:.001:n2+n;
x=input('signal=');
a=x;
b=t1;
else
 t2=n1+n:.001:n2+n;
 t1=t2;
y=x;
end;
end;
subplot(2,1,1);
plot(b,a);
xlabel('time');
ylabel('amplitude');
title('MAIN SIGNAL');
subplot(2,1,2);
plot(t2,y);
xlabel('time');
ylabel('amplitude');
title('SHIFTED SIGNAL');
```

```
min=0
max=10
for orginal n=0,if shifted n=0
signal=(t1/2.5).*(t1<2.5)+((5-
t1)/2.5).*(t1>=2.5&t1<5)+((5-
t1)/2.5).*(t1>=5&t1<7.5)+((t1-
10)/2.5).*(t1>=7.5&t1<=10)
for orginal n=0,if shifted n=10
```



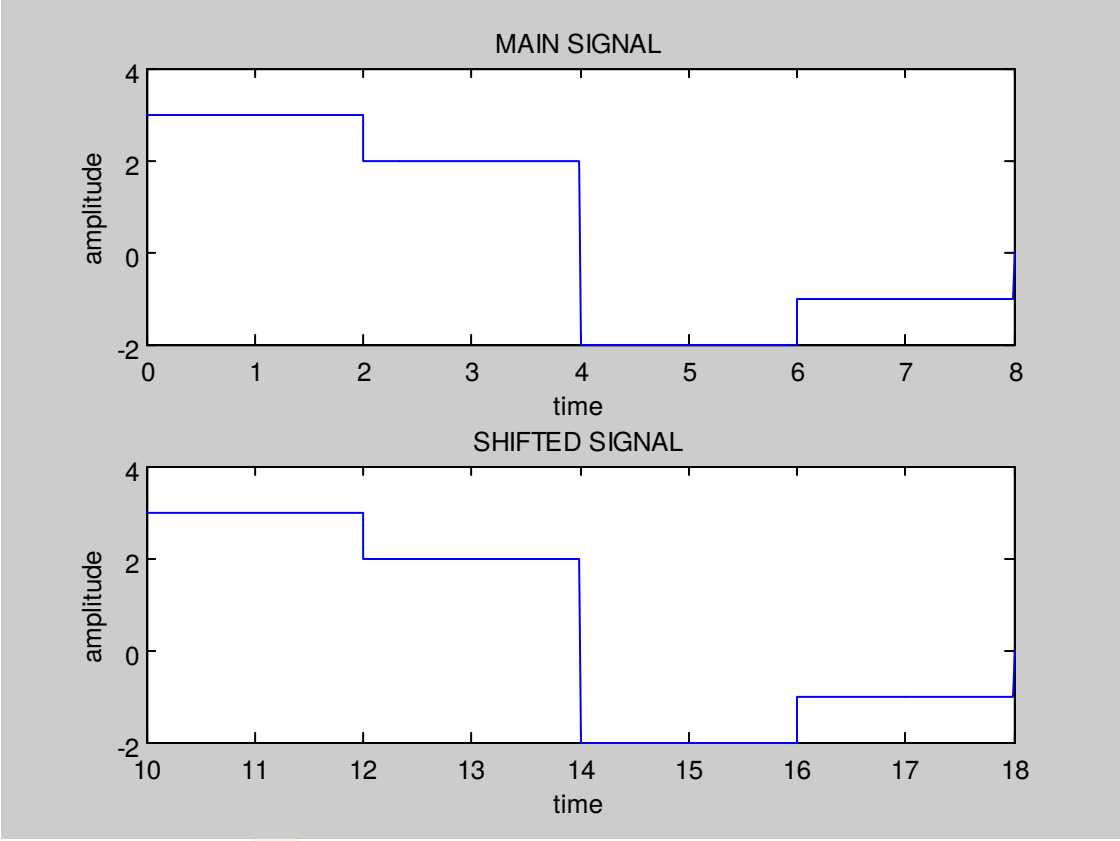
SIGNAL:3

MATLAB CODE:

```
clear all;
n1=input('min=');
n2=input('max=');
for i=1:2
n=input('for orginal n=0,if shifted n=');
if n==0;
t1=n1+n:.001:n2+n;
x=input('signal=');
a=x;
b=t1;
else
 t2=n1+n:.001:n2+n;
 t1=t2;
```

```
y=x;
end;
end;
subplot(2,1,1);
plot(b,a);
xlabel('time');
ylabel('amplitude');
title('MAIN SIGNAL');
subplot(2,1,2);
plot(t2,y);
xlabel('time');
ylabel('amplitude');
title('SHIFTED SIGNAL');
```

FIGURE:



SIGNAL:4

MATLAB CODE:

```
clear all;
n1=input('min=');
n2=input('max=');
for i=1:2
n=input('for orginal n=0,if shifted n=');
if n==0;
t1=n1+n:.001:n2+n;
x=input('signal=');
a=x;
b=t1;
else
 t2=n1+n:.001:n2+n;
 t1=t2;
y=x;
end;
end;
subplot(2,1,1);
plot(b,a);
xlabel('time');
ylabel('amplitude');
title('MAIN SIGNAL');
subplot(2,1,2);
plot(t2,y);
xlabel('time');
ylabel('amplitude');
title('SHIFTED SIGNAL');
```

min=0

max=8

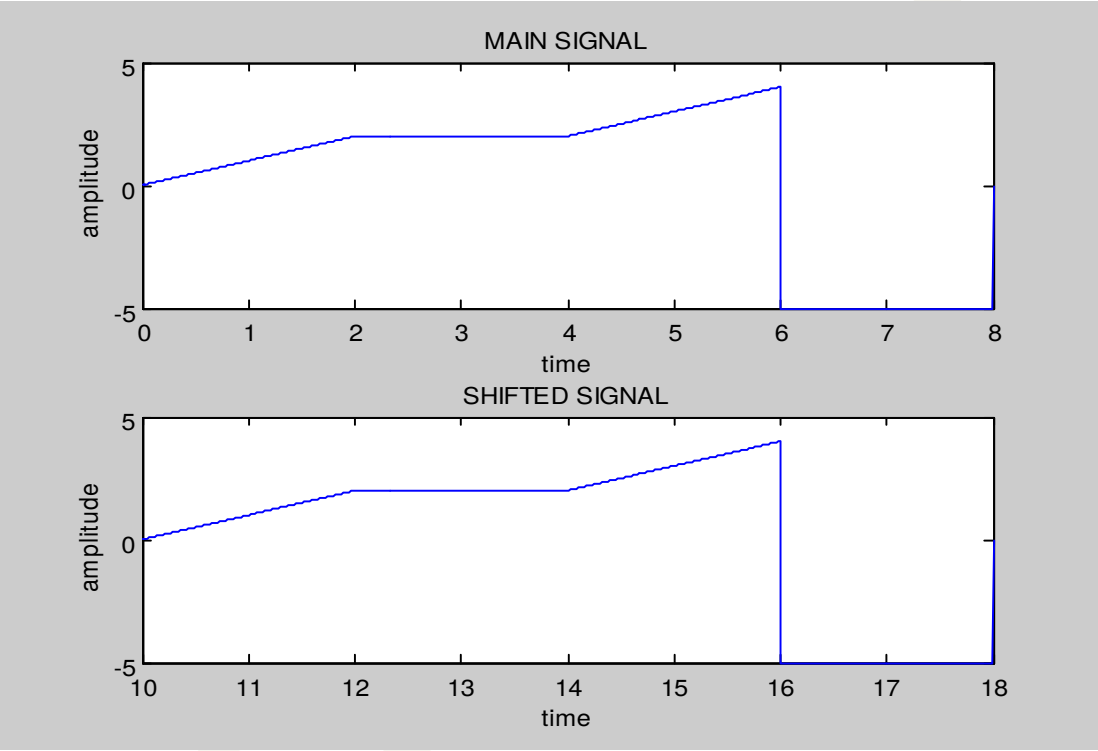
for orginal n=0,if shifted n=0

signal=t1.*(t1<2+n)+2.*(t1>=2+n&t1<4+n)+(t1-2).*(t1>=4+n&t1<6+n)-

5.*(t1>=6+n&t1<8+n)

for orginal n=0,if shifted n=10

FIGURE:



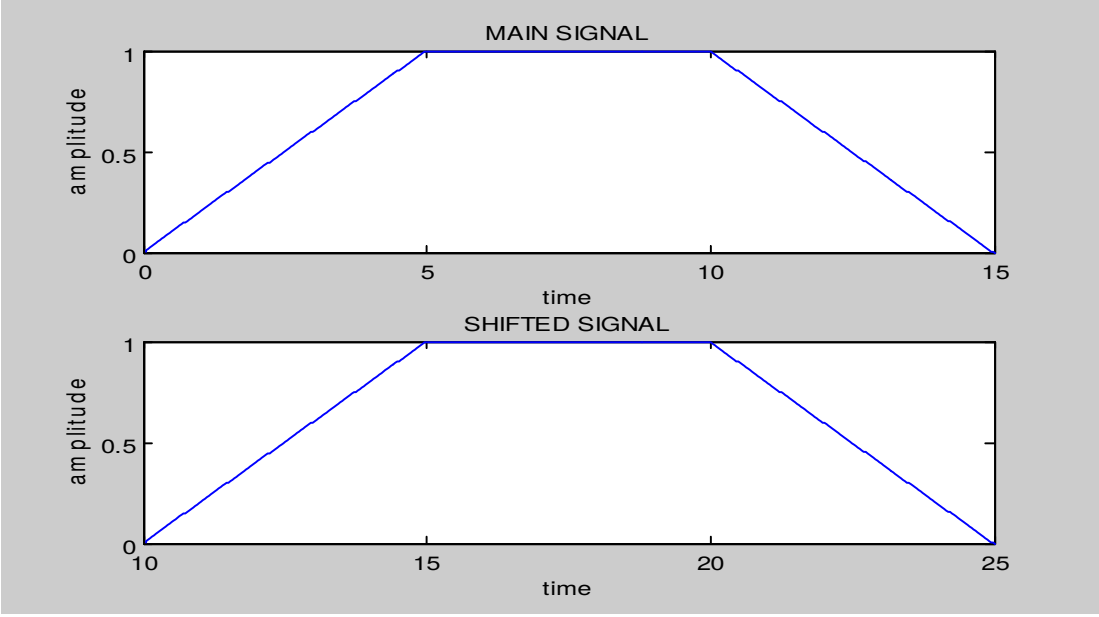
SIGNAL:4

MATLAB CODE:

```
clear all;
n1=input('min=');
```

```
n2=input('max=');
for i=1:2
n=input('for orginal n=0,if shifted n=');
if n==0;
t1=n1+n:.001:n2+n;
x=input('signal=');
a=x;
b=t1;
else
 t2=n1+n:.001:n2+n;
 t1=t2;
y=x;
end;
end;
subplot(2,1,1);
plot(b,a);
xlabel('time');
ylabel('amplitude');
title('MAIN SIGNAL');
subplot(2,1,2);
plot(t2,y);
xlabel('time');
ylabel('amplitude');
title('SHIFTED SIGNAL');
```

FIGURE:



NAME OF THE EXPERIMENT:SCALING OF A GIVEN SIGNAL.

AIM:TO SCALE A GIVEN SIGNAL

SIGNAL:1

MATLAB CODE:

```
clear all;
n1=input('min=');
n2=input('max=');
for i=1:2
n=input('for orginal n=1,if not scaled n=');
if n==1;
t1=n1*n:.001:n2*n;
x=input('signal=');
a=x;
b=t1;
else
 t2=n1*n:.01:n2*n;
 t1=t2;
y=x;
end;
end;
subplot(2,1,1);
plot(b,a);
xlabel('time');
ylabel('amplitude');
title('MAIN SIGNAL');
subplot(2,1,2);
plot(t2,y);
xlabel('time');
ylabel('amplitude');
title('SCALED');
```

min=0

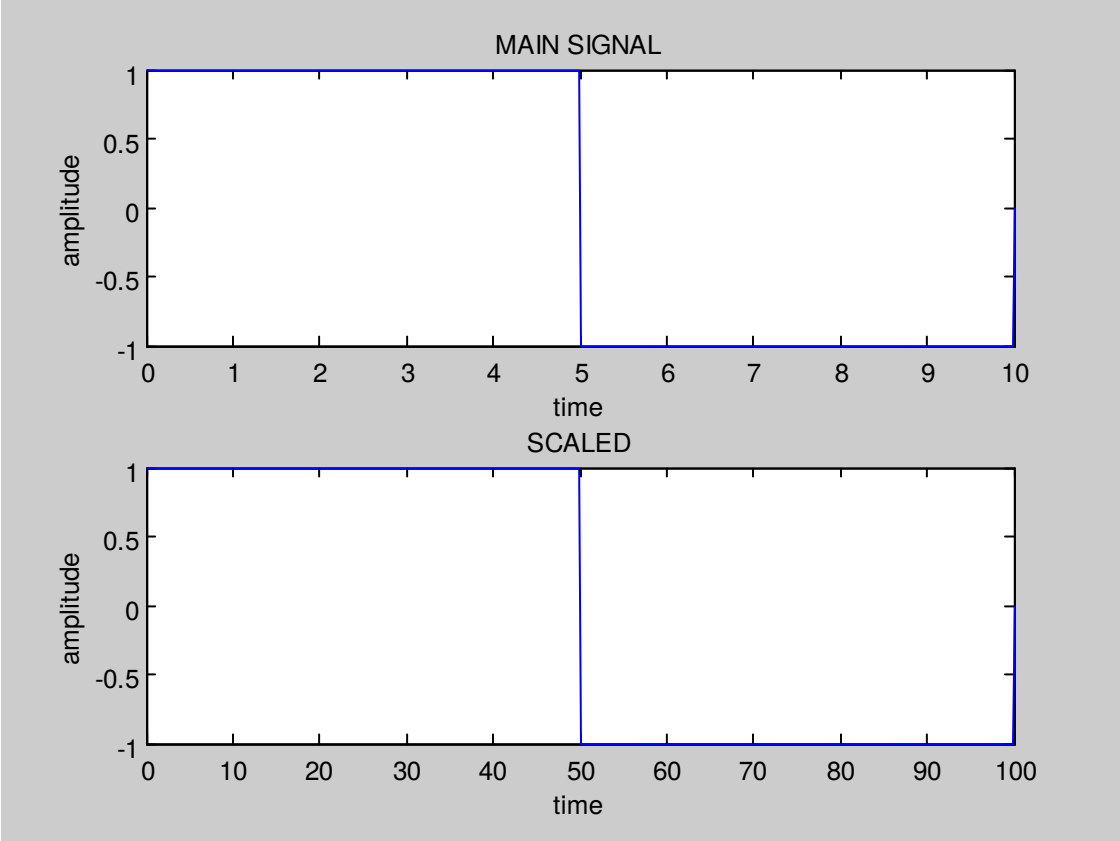
max=10

for orginal n=1,if not scaled n=1

signal=1.*(t1<5*n)-1.*(t1>=5*n&t1<10*n)

for orginal n=1,if not scaled n=10

FIGURE:



SIGNAL:2

MATLAB CODE:

```
clear all;
n1=input('min=');
n2=input('max=');
for i=1:2
n=input('for orginal n=1,if not scaled n=');
if n==1;
t1=n1*n:.001:n2*n;
x=input('signal=');
a=x;
b=t1;
else
 t2=n1*n:.01:n2*n;
 t1=t2;
y=x;
end;
end;
subplot(2,1,1);
plot(b,a);
xlabel('time');
ylabel('amplitude');
title('MAIN SIGNAL');
subplot(2,1,2);
plot(t2,y);
xlabel('time');
ylabel('amplitude');
title('SCALED');
```

min=0

max=10

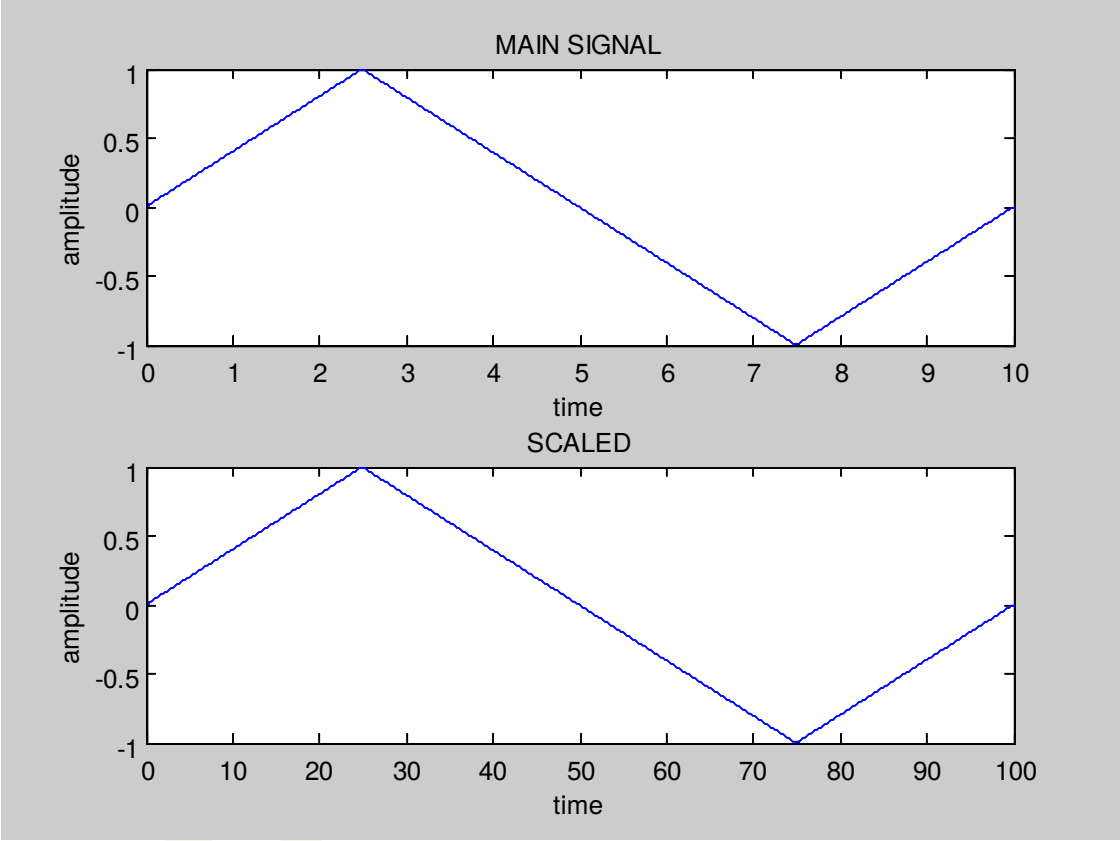
for orginal n=1,if not scaled n=1

signal=(t1/2.5).*(t1<2.5)+((5-t1)/2.5).*(t1>=2.5&t1<5)+((5-

t1)/2.5).*(t1>=5&t1<7.5)+((t1-10)/2.5).*(t1>=7.5&t1<=10)

for orginal n=1,if not scaled n=10

FIGURE:



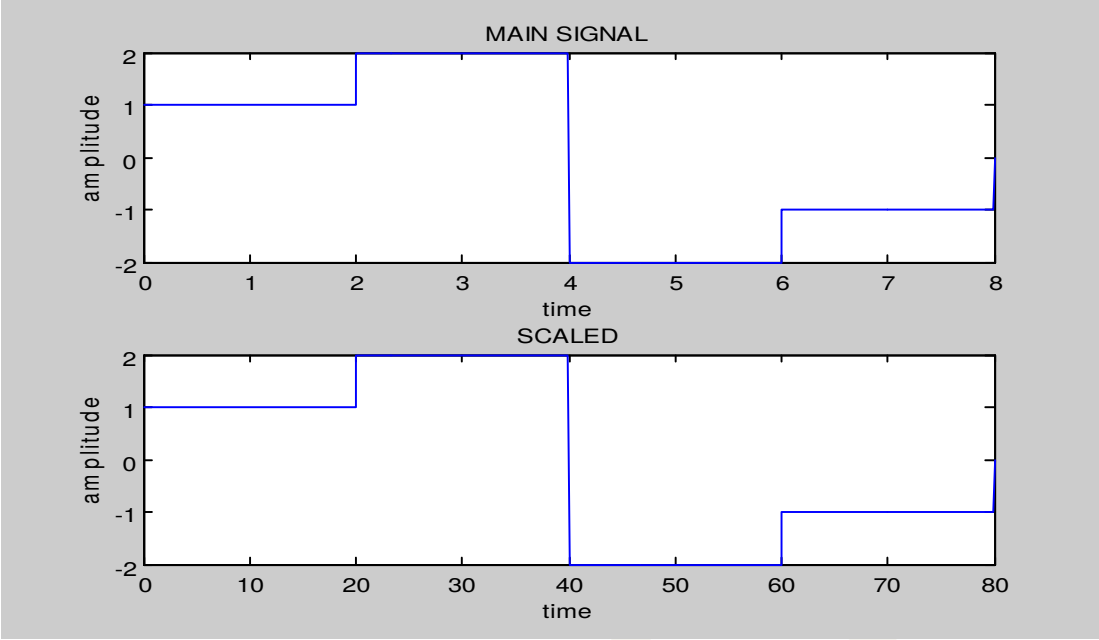
SIGNAL:3

MATLAB CODE:

```
clear all;

n1=input('min=');
n2=input('max=');
```

```
for i=1:2
n=input('for orginal n=1,if not scaled n=');
if n==1;
t1=n1*n:.0001:n2*n;
x=input('signal=');
a=x;
b=t1;
else
 t2=n1*n:.001:n2*n;
 t1=t2;
y=x;
end;
end;
subplot(2,1,1);
plot(b,a);
xlabel('time');
ylabel('amplitude');
title('MAIN SIGNAL');
subplot(2,1,2);
plot(t2,y);
xlabel('time');
ylabel('amplitude');
title('SCALED');


min=0
max=8
for orginal n=1,if not scaled n=1

signal=1.*(t1<2*n)+2.*(t1>=2*n&t1<4*n)-
2.*(t1>=4*n&t1<6*n)-1.*(t1>=6&t1<8)
for orginal n=1,if not scaled n=10
```

FIGURE:



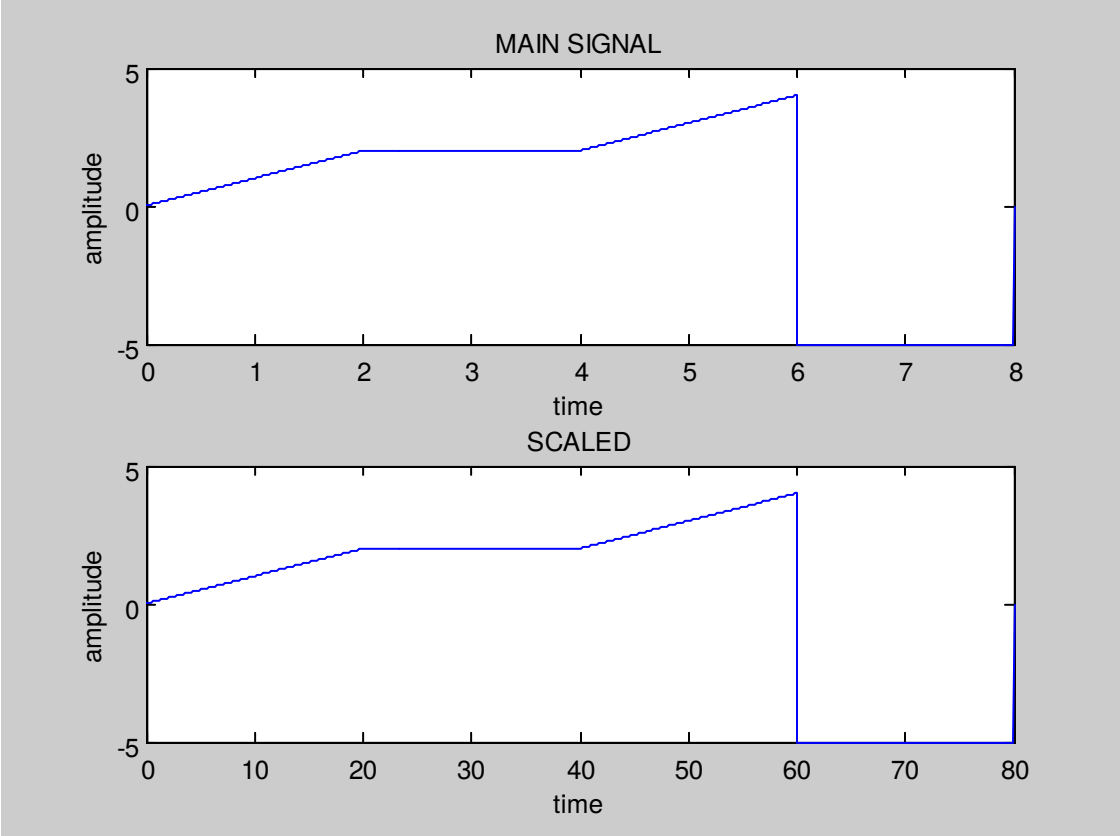
SIGNAL:4

MATLAB CODE:

```
clear all;
n1=input('min=');
n2=input('max=');
for i=1:2
n=input('for orginal n=1,if not scaled n=');
if n==1;
t1=n1*n:.001:n2*n;
x=input('signal=');
a=x;
b=t1;
else
 t2=n1*n:.01:n2*n;
 t1=t2;
y=x;
end;
end;
subplot(2,1,1);
plot(b,a);
xlabel('time');
ylabel('amplitude');
title('MAIN SIGNAL');
subplot(2,1,2);
```

```
plot(t2,y);
xlabel('time');
ylabel('amplitude');
title('SCALED');


min=0
max=8
for orginal n=1,if not scaled n=1
signal=t1.*(t1<2*n)+2.*(t1>=2*n&t1<4*n)+(t1-
2).*(t1>=4*n&t1<6*n)-5.*(t1>=6*n&t1<8*n)
for orginal n=1,if not scaled n=10
```

FIGURE:

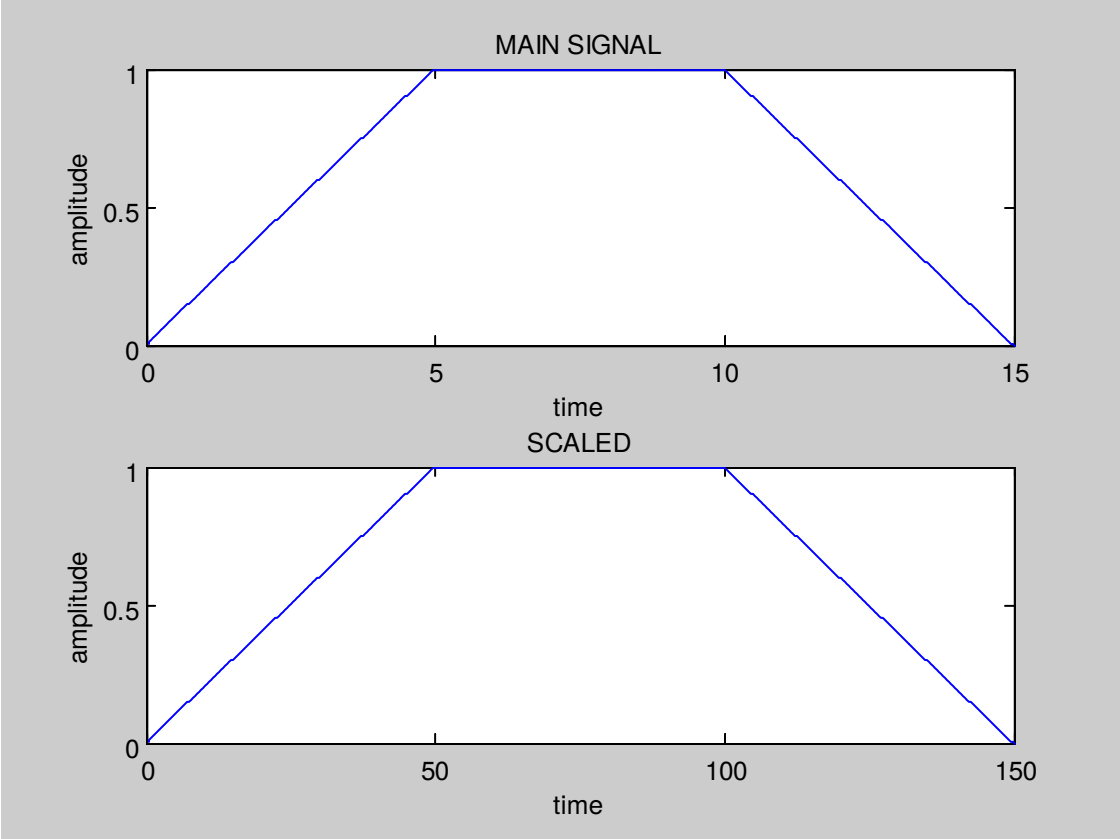


SIGNAL:5

MATLAB CODE:

```
clear all;
n1=input('min=');
n2=input('max=');
for i=1:2
n=input('for orginal n=1,if not scaled n=');
if n==1;
t1=n1*n:.0001:n2*n;
x=input('signal=');
a=x;
b=t1;
else
 t2=n1*n:.001:n2*n;
 t1=t2;
y=x;
end;
end;
subplot(2,1,1);
plot(b,a);
xlabel('time');
ylabel('amplitude');
title('MAIN SIGNAL');
subplot(2,1,2);
plot(t2,y);
xlabel('time');
ylabel('amplitude');
title('SCALED');


min=0
max=15
for orginal n=1,if not scaled n=1
signal=t1/5.*(t1<5*n)+1.*(t1>=5*n&t1<10*n)+((15-
t1)/5).*(t1>=10*n&t1<=15*n)
for orginal n=1,if not scaled n=10
```

FIGURE:



AIM: TO DETERMINE SCALING OF SIGNAL by using samples

MATLAB CODE:

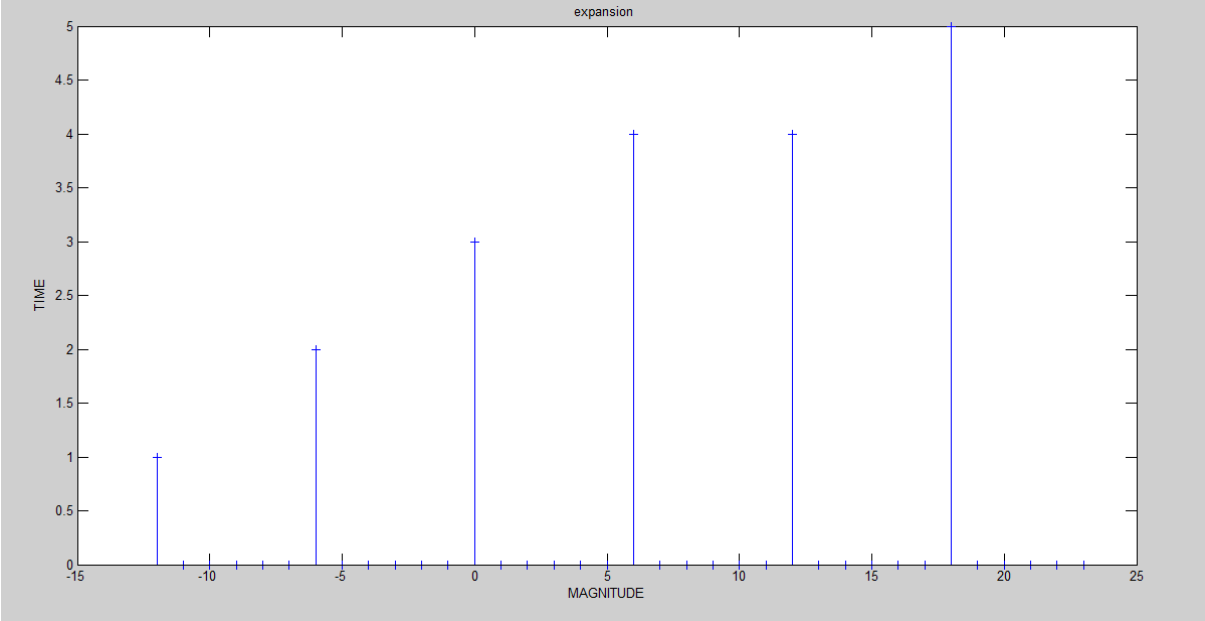
```
clc
clear all
% scaling of a function by using logic-
% expansion of a signal x(n/2)
x=[1 2 3 4 4 5 ]
n=-2:3
a=input('enter scalar: ' );
% y(1)=x(1)
y=cell(1,a)
for i=1:length(x)
 y{i}=[x(i) ]
 for j=1:a
y{i}=[y{i} 0 ]
 end
end
s=cell2mat(y)
```

```
n1=min(n)*(a+1)
n2=max(n)*(a+1)
ns=n1:n2+a
stem(ns,s)
```

input:

enter scalar: 5

OUTPUT:



AIM:FOLDING OF A SIGNAL

MATLAB CODE:

```
clear all
clc
s1=input('enter signal:');
x=length(s1)
m=[ ]
for i=0:1:x-1
   m=[m i];
end
m
```

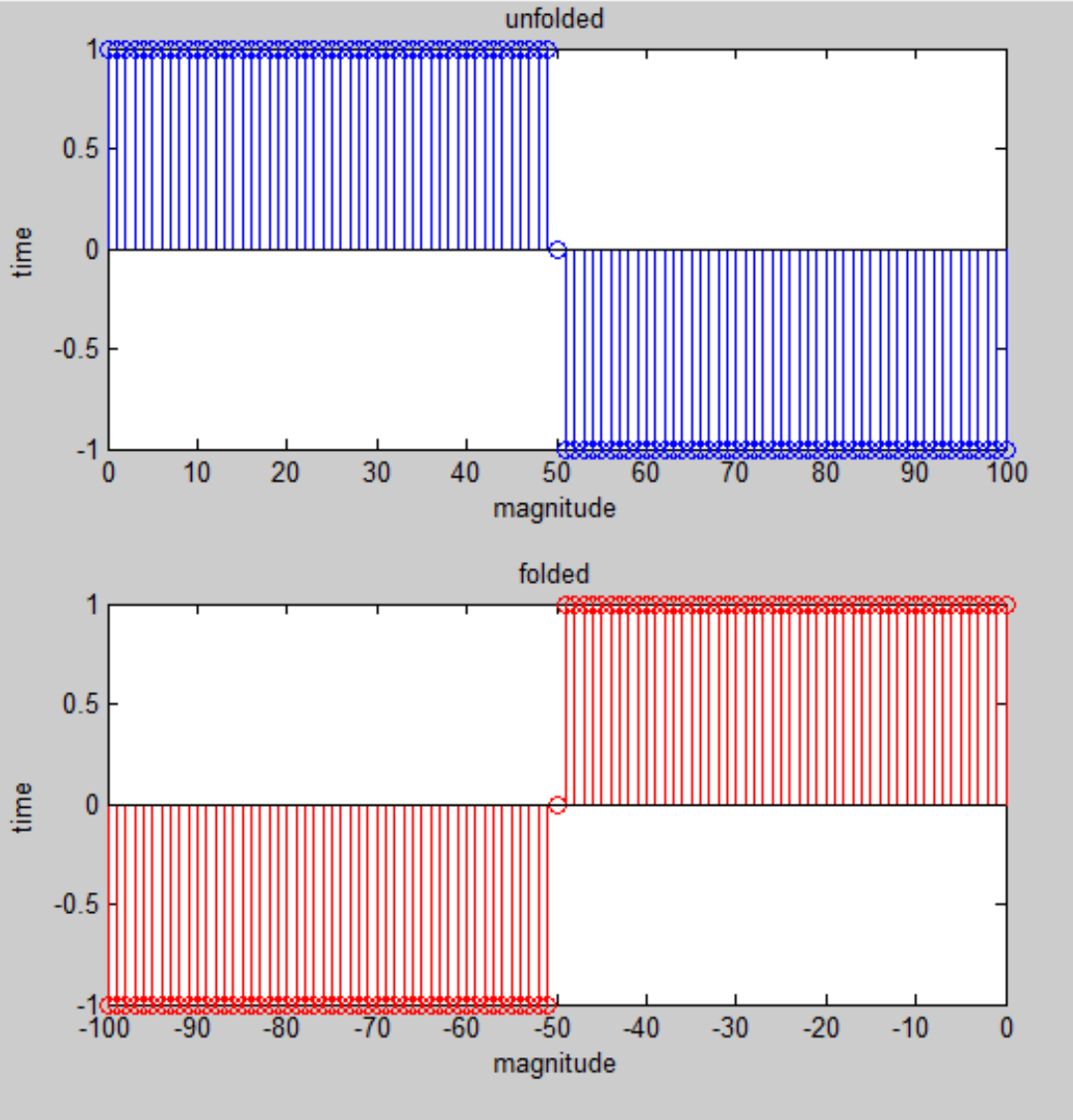
```
subplot(2,1,1)
stem(m,s1,'blue')
xlabel('magnitude')
ylabel('time')
title('unfolded')
k=min(m)
k1=max(m)
m1=[ ]
for i=k:k1
   m1=[-i m1];
end
m1
s2=fliplr(s1)
subplot(2,1,2)
stem(m1,s2,'red')
xlabel('magnitude')
ylabel('time')
title('folded')
```

INPUT:

enter signal:signal1

```
x =101
m =[ ]
k = 0

k1 =100

m1 =[ ]
```

OUTPUT:


unfolded
time
magnitude
folded
time
magnitude

AIM: CONVOLUTION OF 2 SIGNALS

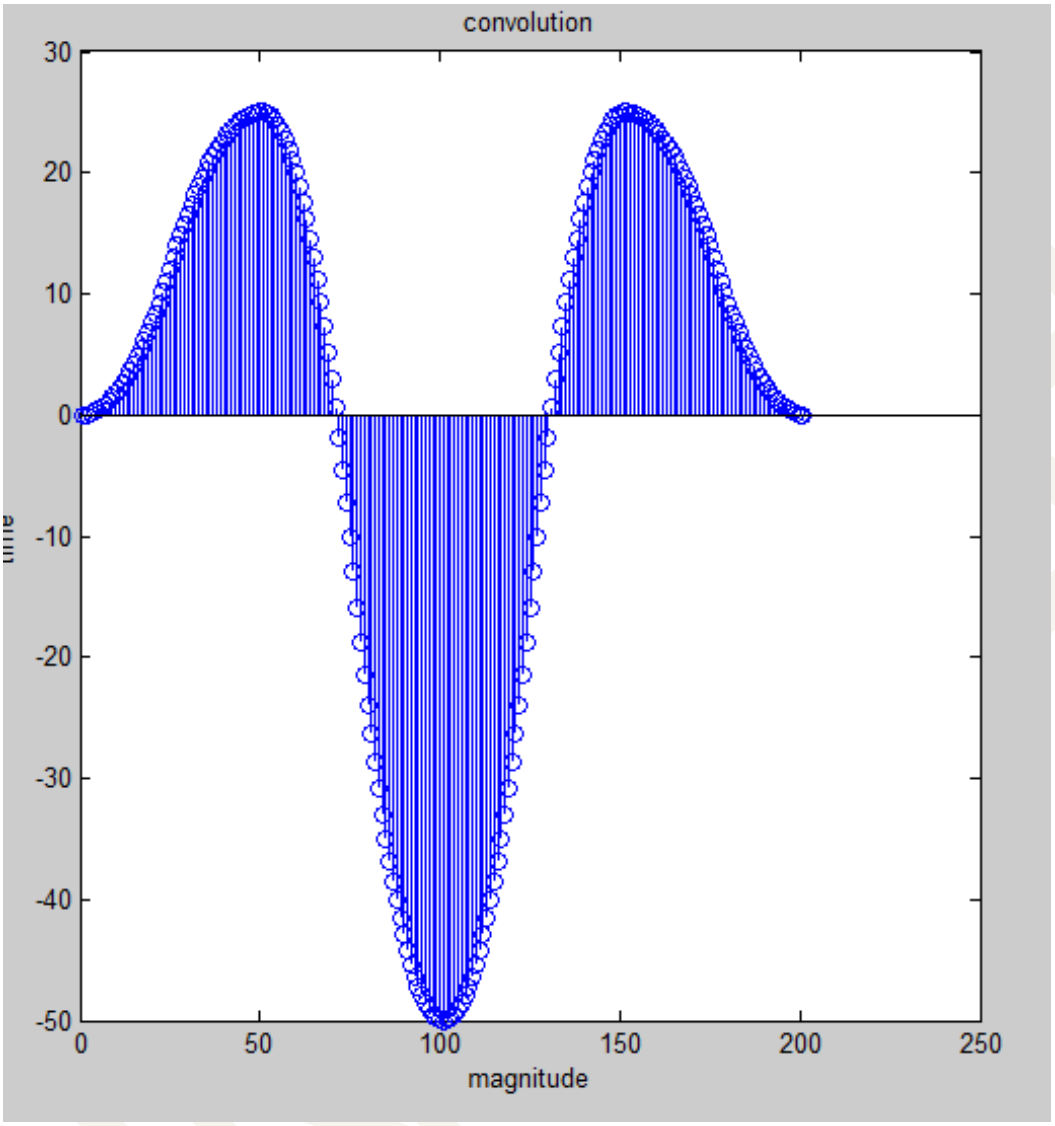
MATLAB CODE:

```
clear all
clc
x=input('enter signal:');
h=input('enter signal:');
m=length(x);
n=length(h);
l1=[ ];
for i=0:1:m
    l1=[l1 i];
end

l2=[ ];
for i=0:1:n
    l2=[l1 i];
end
[x h]=padding2(x,l1,h,l2)
m=length(x);
n=length(h);
a=[x,zeros(1,n)];
b=[h,zeros(1,m)];
for i=1:n+m-1
    y(i)=0;
for j=1:m
    if(i-j+1>0)
      y(i)=y(i)+a(j)*b(i-j+1);
    else
    end
end
end
stem(y);
```

INPUT:

```
enter signal:SIGNAL1
enter signal:SIGNAL2
```

OUTPUT:


convolution
30
20
10
0
-10
-20
-30
-40
-50
0
50
100
150
200
250
magnitude

AIM:Correlation OF 2 SIGNALS

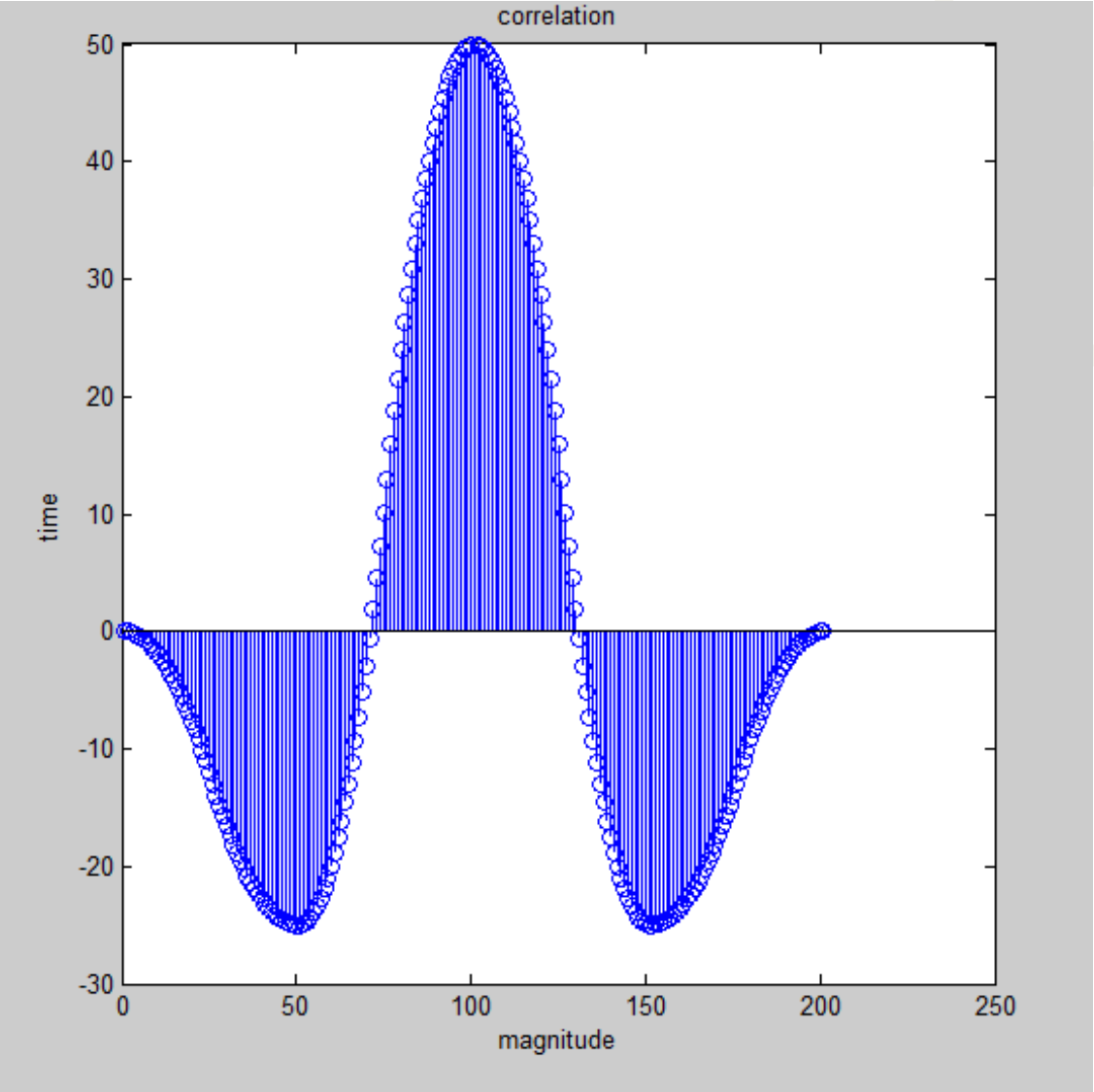
MATLAB CODE:

```
clear all
clc
x=input('enter signal:');
s=input('enter signal:');
h=fliplr(s);
m=length(x);
n=length(h);
l1=[ ];
for i=0:1:m
    l1=[l1 i];
end

l2=[ ];
for i=0:1:n
    l2=[l1 i];
end
[x h]=padding2(x,l1,h,l2)
m=length(x);
n=length(h);
a=[x,zeros(1,n)];
b=[h,zeros(1,m)];
for i=1:n+m-1
    y(i)=0;
for j=1:m
    if(i-j+1>0)
     y(i)=y(i)+a(j)*b(i-j+1);
    else
    end
end
end
stem(y);
xlabel('magnitude')
ylabel('time')
title('correlation.')
```

input:

```
enter signal:SIGNAL1
enter signal:SIGNAL2
```

output:


correlation
50
40
30
20
10
0
-10
-20
-30
time
0
50
100
150
200
250
magnitude

NAME OF THE EXPERIMENT:GENERATE THE ENERGY OFA SIGNAL

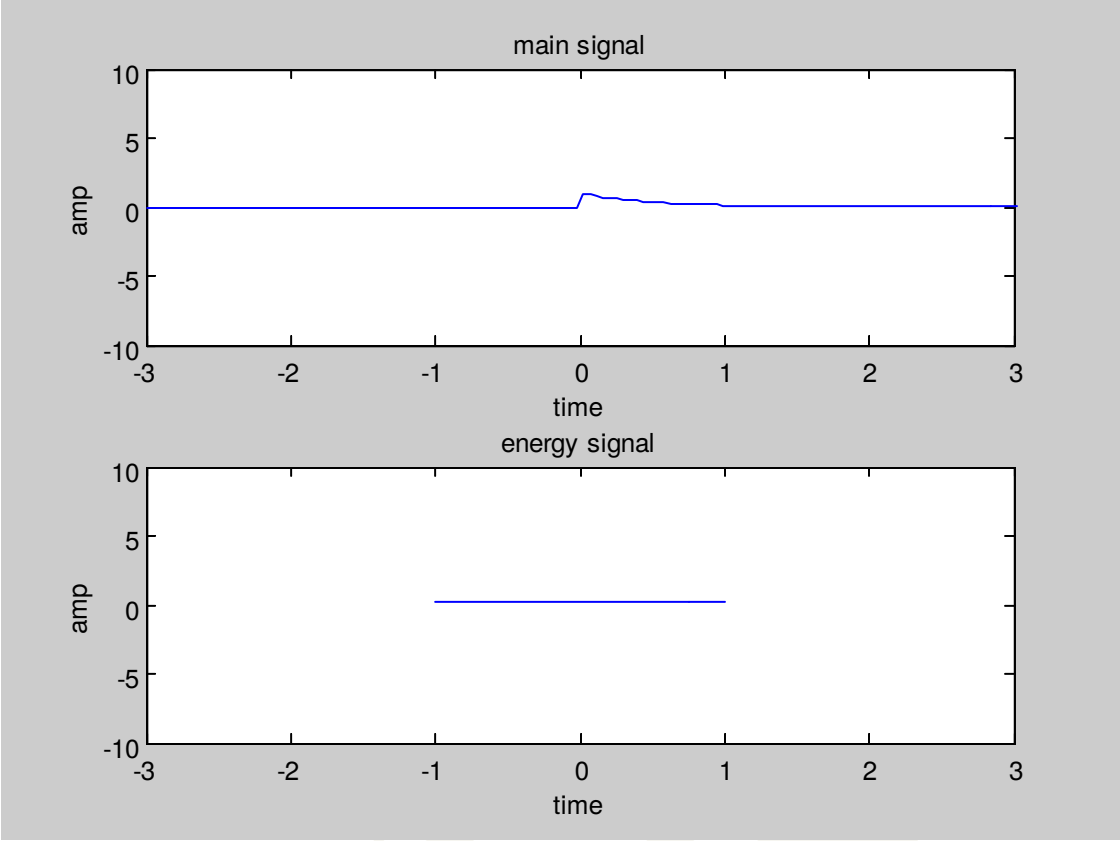
AIM:TO GENERATE THE ENERGY OF THE SIGNAL

MATLAB CODE:

```
%energy of a signal
clc;
clear all;
close all;
syms t;
x=exp(-2*t).*heaviside(t)

t1=input('tmin=');
t2=input('tmax=');
e=int(x.^2,t1,t2)
subplot(2,1,1);
ezplot(x,[-10 10]);
axis([-3 3 -10 10]);
xlabel('time');
ylabel('amp');
title('main signal');
subplot(2,1,2);
ezplot(e,[-1 1]);
axis([-3 3 -10 10]);
xlabel('time');
ylabel('amp');
xlabel('time');
ylabel('amp');
title('energy signal');
```

output:

```
x =

heaviside(t)/exp(2*t)


tmin=0

tmax=2


e =

1/4 - 1/(4*exp(8))
```


main signal
amp
time
energy signal
amp
time

NAME OF THE EXPERIMENT:FIND THE POWER OF THE SIGNAL

AIM:TO FIND THE POWER OF THE SIGNAL

MATLAB CODE:

```
%power of signal
clc;
clear all;
close all;
syms t;
x1=input('x1=');
x2=input('x2=');
x=x1+x2
t=input('time period =');
t1=input('tmin=');
t2=input('tmax=');
e1=(int((x1.^2),t1,t2))./t;
e2=(int((x2.^2),t1,t2))./t;
e=e1+e2
subplot(2,1,1);
ezplot(x,[-10 10]);
axis([-3 3 -10 10]);
xlabel('time');
ylabel('amp');
title('main signal');
subplot(2,1,2);
ezplot(e,[-1 1]);
axis([-3 3 -10 10]);
xlabel('time');
ylabel('amp');
title('power signal');
```

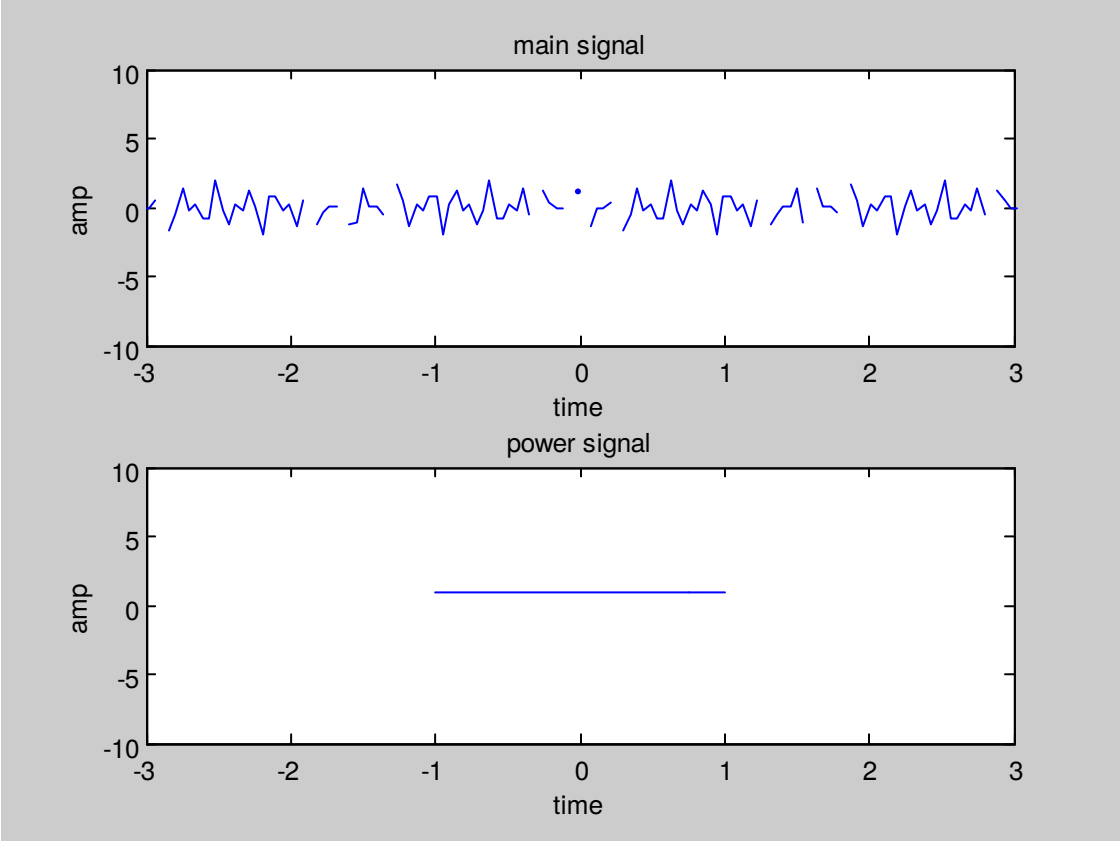
output:

```
x =


cos(30*t) + sin(pi/2 + 50*t)


time period =pi/5

tmin=0

tmax=pi/5
```

e =

1

figure



NAME OF THE EXPERIMENT: FIND THE FOURIER OF HEAVISIDE

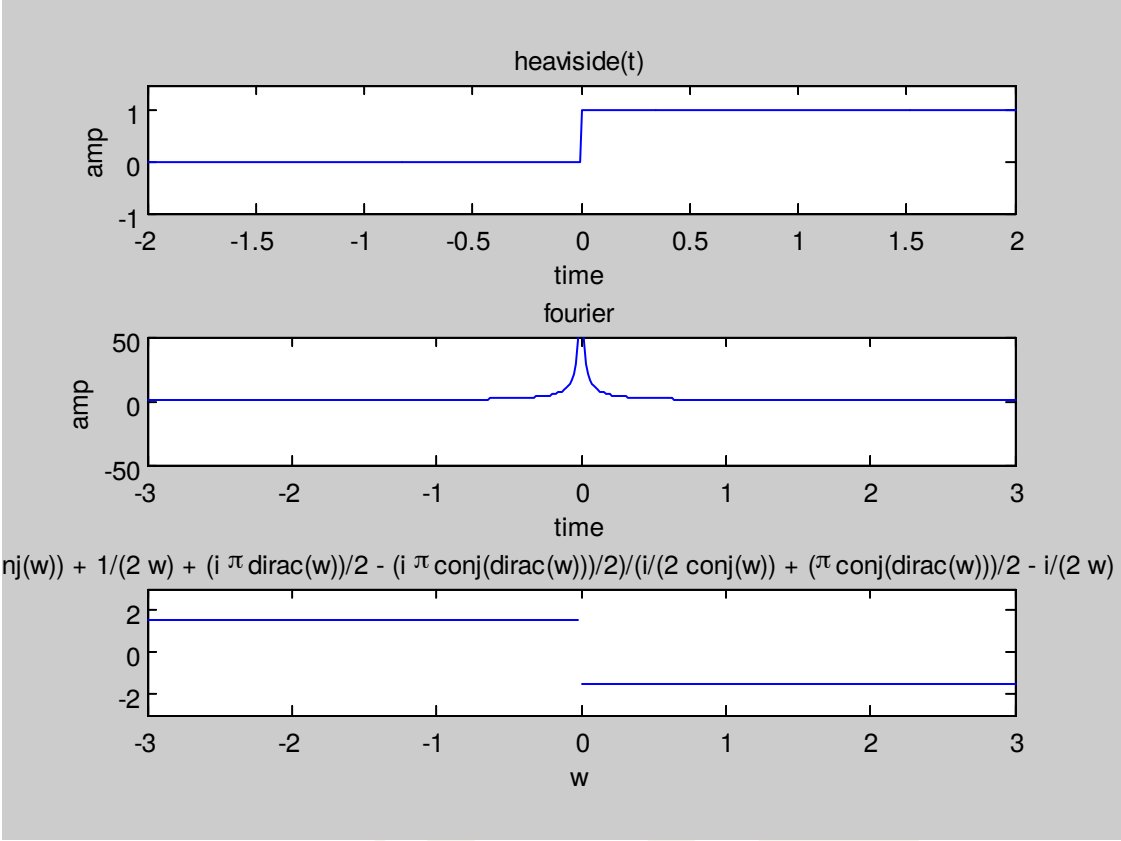
AIM:TO FIND THE FOURIER OF HEAVISIDE

MATLAB CODE:

```
%fourier heaviside
clc;
clear all;
close all;
syms t;
x=input('x=');
subplot(3,1,1);
ezplot(x,[-2 2]);
xlabel('time');
ylabel('amp');
axis([-2 2 -1 1.5])
y=fourier(x)
subplot(3,1,2);
ezplot(abs(y),[-3 3]);
xlabel('time');
ylabel('amp');
title('fourier');
axis([-3 3 -50 50]);
subplot(3,1,3);
ezplot(atan(imag(y)/real(y)),[-3 3]);
axis([-3 3 -3 3]);
```

output:

```
x=heaviside(t)


y =


pi*dirac(w) - i/w
```


heaviside(t)
amp
time
fourier
amp
time
nj(w)) + 1/(2 w) + (i π dirac(w))/2 - (i π conj(dirac(w)))/2)/(i/(2 conj(w)) + (π conj(dirac(w)))/2 - i/(2 w)
w

NAME OF THE EXPERIMENT:FIND THE FOURIER OF DIRAC FUNCTION

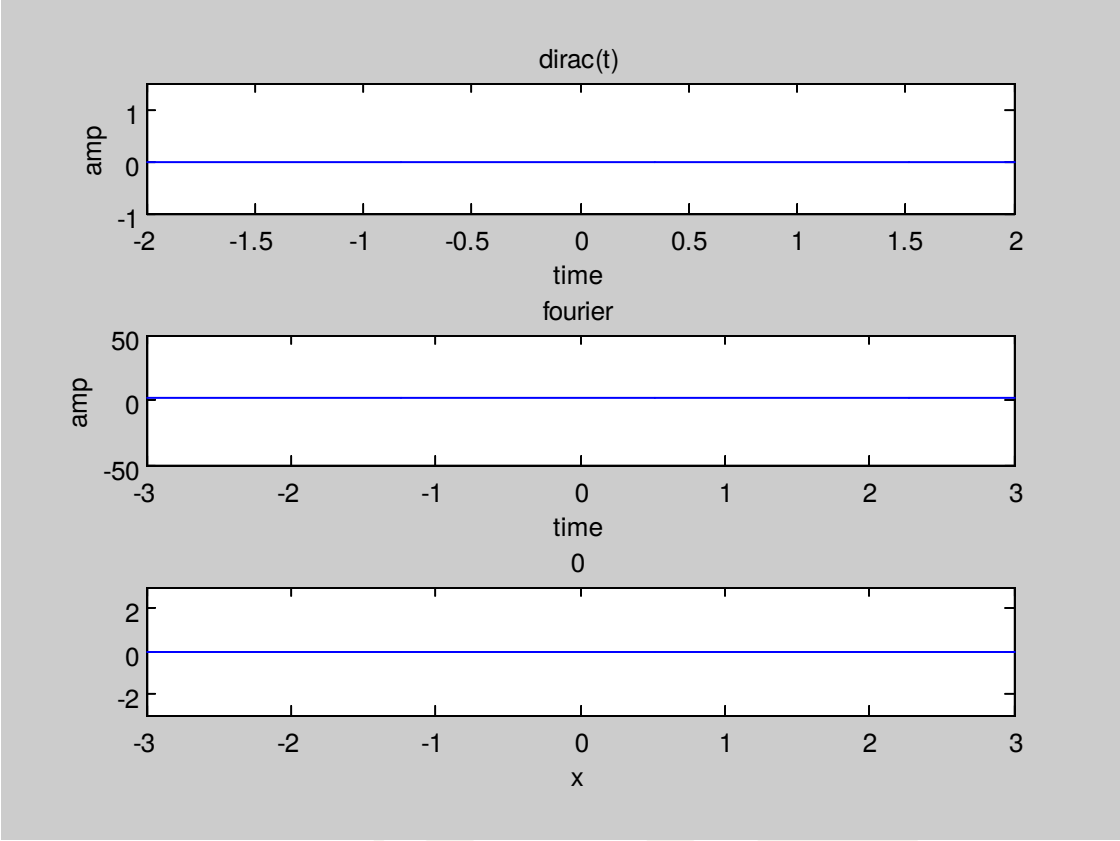
AIM:TO FIND THE FOURIER OF DIRAC FUNCTION

MATLAB CODE:

```
%fourier dirac(t)
clc;
clear all;
close all;
syms t;
x=input('x=');
subplot(3,1,1);
ezplot(x,[-2 2]);
xlabel('time');
ylabel('amp');
axis([-2 2 -1 1.5])
y=fourier(x)
subplot(3,1,2);
ezplot(abs(y),[-3 3]);
xlabel('time');
ylabel('amp');
title('fourier');
axis([-3 3 -50 50]);
subplot(3,1,3);
ezplot(atan(imag(y)/real(y)),[-3 3]);
axis([-3 3 -3 3]);
```

output:

```
x=dirac(t)


y =


1
```


dirac(t)
amp
1
0
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
time
fourier
amp
50
0
-50
-3
-2
-1
0
1
2
3
time
0
2
0
-2
-3
-2
-1
0
1
2
3
x

NAME OF THE EXPERIMENT:FIND THE FOURIER TRANSFORM OF PRODUCT OF EXPONENTIAL AND UNITSTEP

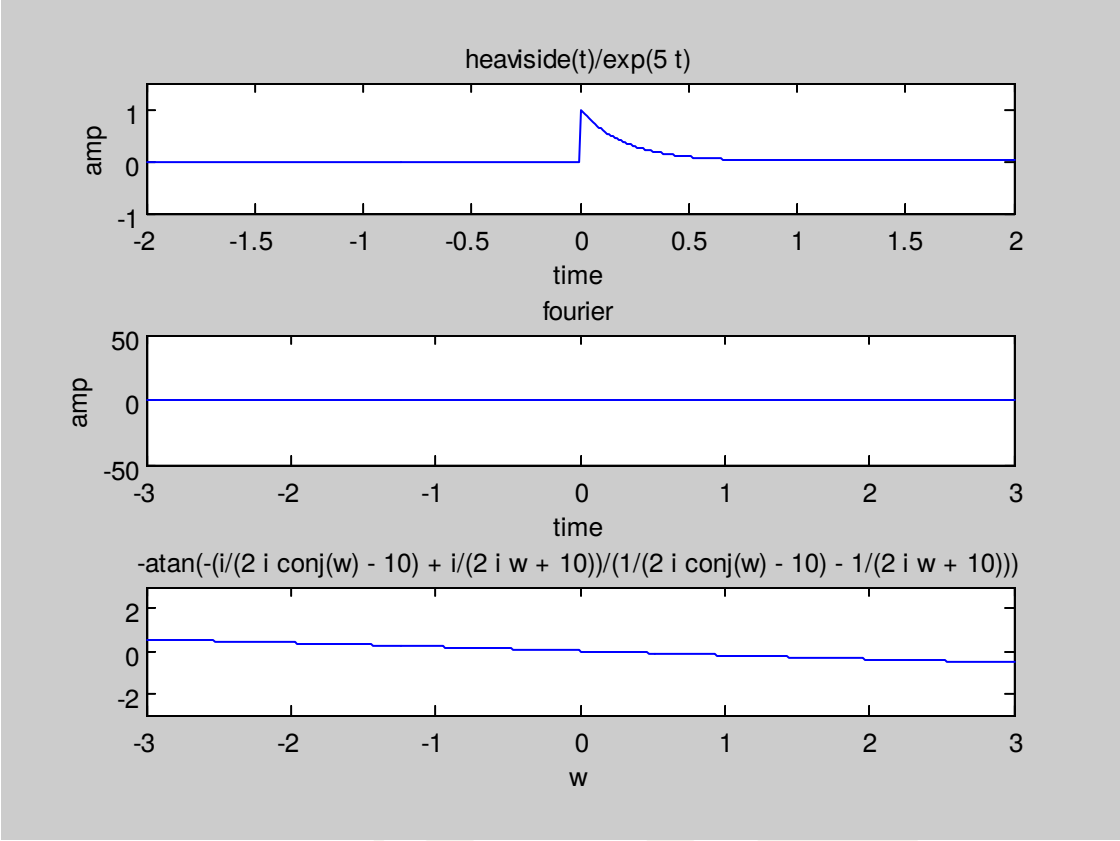
AIM:TO FIND THE FOURIER TRANSFORM OF PRODUCT OF EXPONENTIAL AND UNIT STEP

MATLAB CODE:

```
%fourier exp(-2t)*u(t)
clc;
clear all;
close all;
syms t;
x=input('x=');
subplot(3,1,1);
ezplot(x,[-2 2]);
xlabel('time');
ylabel('amp');
axis([-2 2 -1 1.5])
y=fourier(x)
subplot(3,1,2);
ezplot(abs(y),[-3 3]);
xlabel('time');
ylabel('amp');
title('fourier');
axis([-3 3 -50 50]);
subplot(3,1,3);
ezplot(atan(imag(y)/real(y)),[-3 3]);
axis([-3 3 -3 3]);
```

output:

```
x=exp(-5*t)*heaviside(t)


y =


1/(5 + w*i)
```


heaviside(t)/exp(5 t)
amp
time
fourier
amp
time
-atan(-(i/(2 i conj(w) - 10) + i/(2 i w + 10))/(1/(2 i conj(w) - 10) - 1/(2 i w + 10)))
w

NAME OF THE EXPERIMENT: FIND THE LAPLACE OF DIRACE FUNCTION

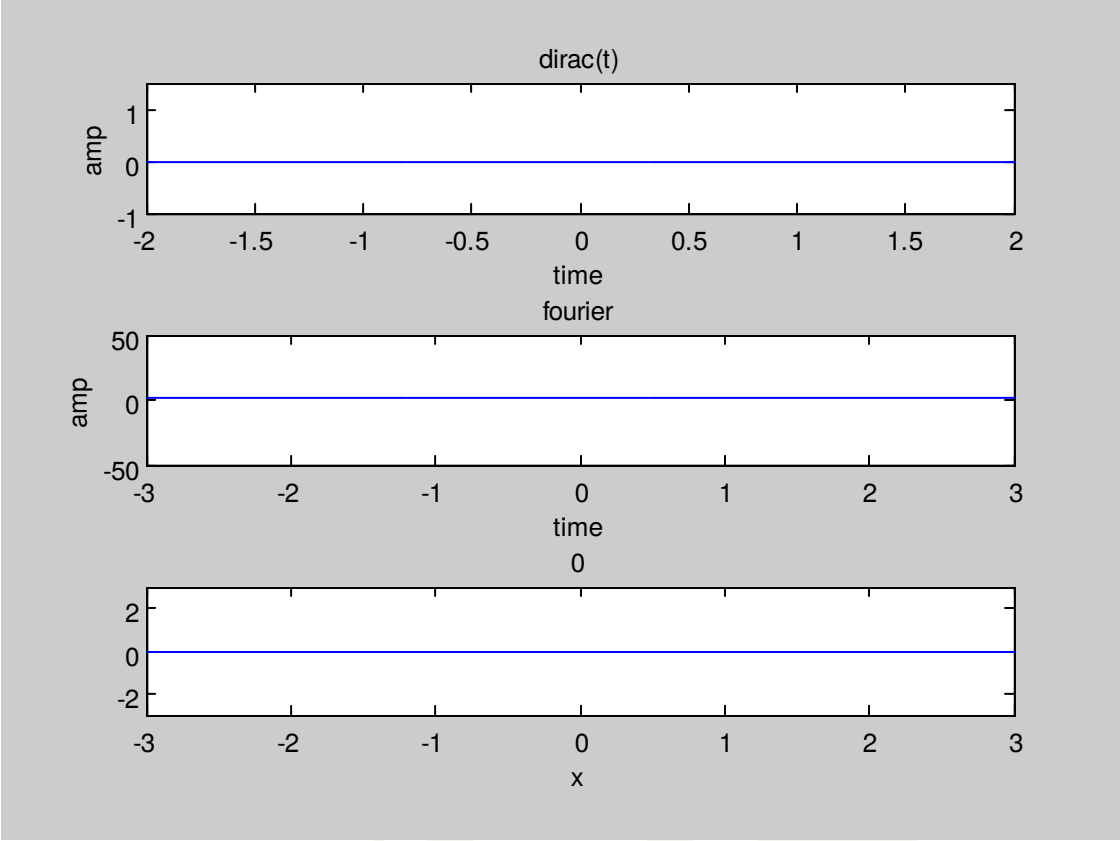
AIM:TO FIND THE LAPLACE OF DIRACE FUNCTION

MATLAB CODE:

```
%laplace of dirac(t)
clc;
clear all;
close all;
syms t;
x=input('x=');
subplot(3,1,1);
ezplot(x,[-2 2]);
xlabel('time');
ylabel('amp');
axis([-2 2 -1 1.5])
y=laplace(x)
subplot(3,1,2);
ezplot(abs(y),[-3 3]);
xlabel('time');
ylabel('amp');
title('laplace');
axis([-3 3 -50 50]);
subplot(3,1,3);
ezplot(atan(imag(y)/real(y)),[-3 3]);
axis([-3 3 -3 3]);
```

output:

```
x=dirac(t)


y =


1
```


dirac(t)
amp
1
0
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
time
fourier
amp
50
0
-50
-3
-2
-1
0
1
2
3
time
0
2
0
-2
-3
-2
-1
0
1
2
3
x

NAME OF THE EXPERIMENT: FIND THE LAPLACE TRANSFORM OF DIRACE FUNCTION

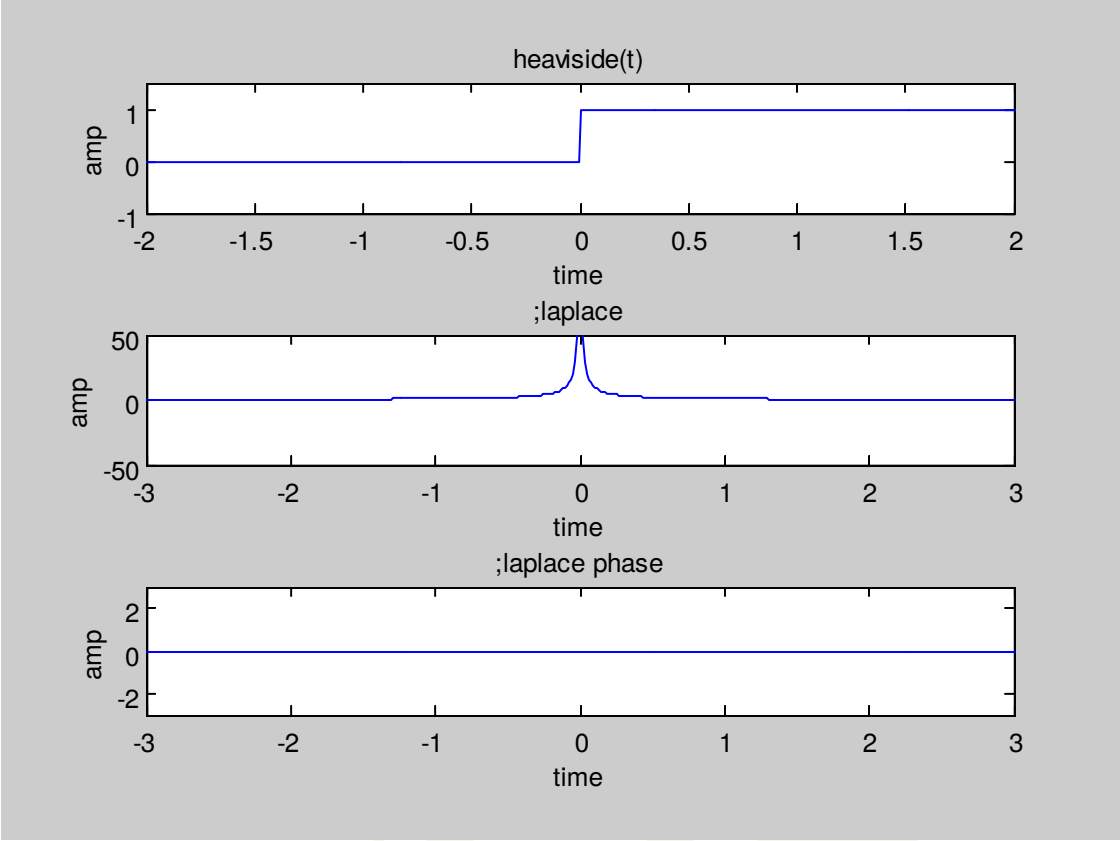
AIM:TO FIND THE LAPLACE TRANSFORMS OF DIRACE FUNCTION

MATLAB CODE:

```
%laplace of dirac(t)
clc;
clear all;
close all;
syms t;
x=input('x=');
subplot(3,1,1);
ezplot(x,[-2 2]);
xlabel('time');
ylabel('amp');
axis([-2 2 -1 1.5])
y=laplace(x)
subplot(3,1,2);
ezplot(abs(y),[-3 3]);
xlabel('time');
ylabel('amp');
title(';laplace');
axis([-3 3 -50 50]);
subplot(3,1,3);
ezplot(atan(imag(y)/real(y)),[-3 3]);
xlabel('time');
ylabel('amp');
title(';laplace phase')
axis([-3 3 -3 3]);
x=heaviside(t)
```

output:

```
y =


1/s
```


heaviside(t)
amp
time
;laplace
amp
time
;laplace phase
amp
time

NAME OF THE EXPERIMENT:FIND THE LAPLACE TRANSFORM OF PRODUCT OF EXPONENTIAL AND UNIT STEP FUNCTION

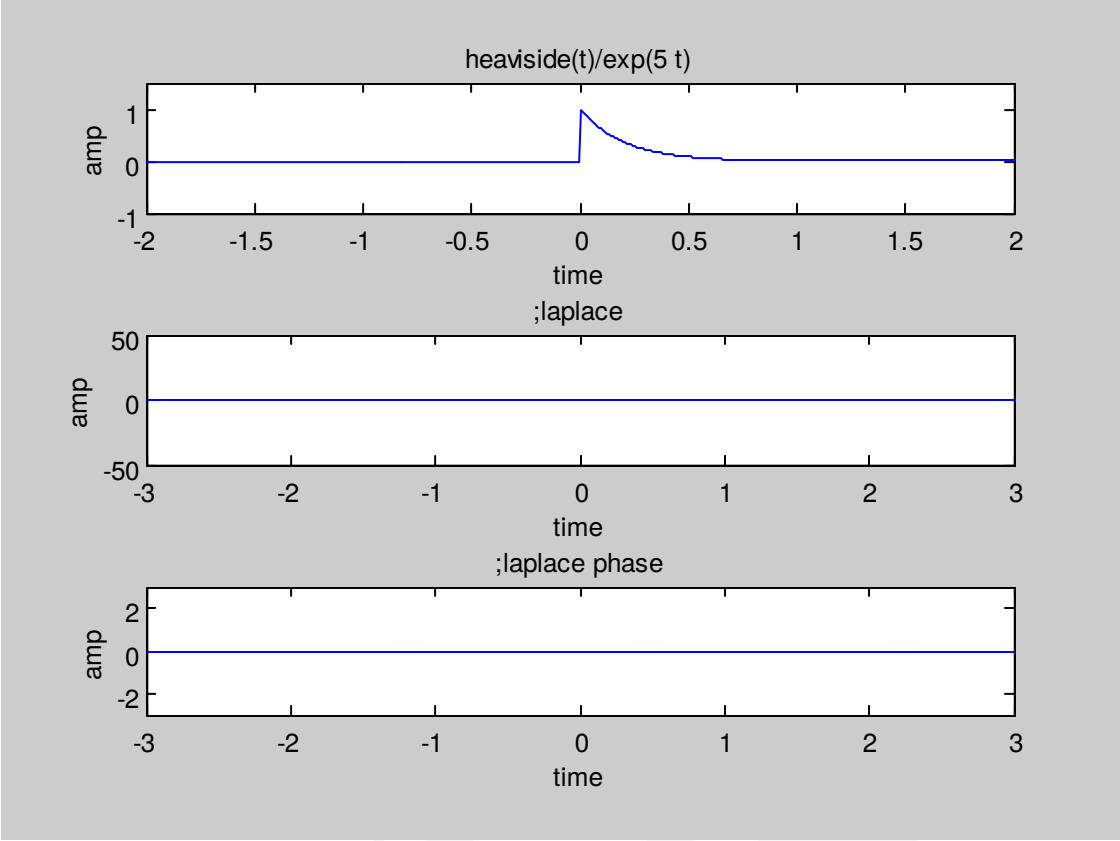
AIM:TO FIND THE LAPLACE TRANSFORM OF PRODUCT OF EXPONENTIAL AND UNIT STEP FUNCTION

MATLAB CODE:

```
%laplace of exp(-5t).heaviside(t)
clc;
clear all;
close all;
syms t;
x=input('x=');
subplot(3,1,1);
ezplot(x,[-2 2]);
xlabel('time');
ylabel('amp');
axis([-2 2 -1 1.5])
y=laplace(x)
subplot(3,1,2);
ezplot(abs(y),[-3 3]);
xlabel('time');
ylabel('amp');
title(';laplace');
axis([-3 3 -50 50]);
subplot(3,1,3);
ezplot(atan(imag(y)/real(y)),[-3 3]);
xlabel('time');
ylabel('amp');
title(';laplace phase')
axis([-3 3 -3 3]);
```

output:

```
x=exp(-5*t).*heaviside(t)


y =


1/(s + 5)
```


heaviside(t)/exp(5 t)
amp
time
;laplace
amp
time
;laplace phase
amp
time

NAME OF THE EXPERIMENT:FIND THE GIBBS PHENOMENON

AIM:TO FIND THE GIBBS PHENOMENON

MATLAB CODE:

AIM:GIBBS PHENOMENON

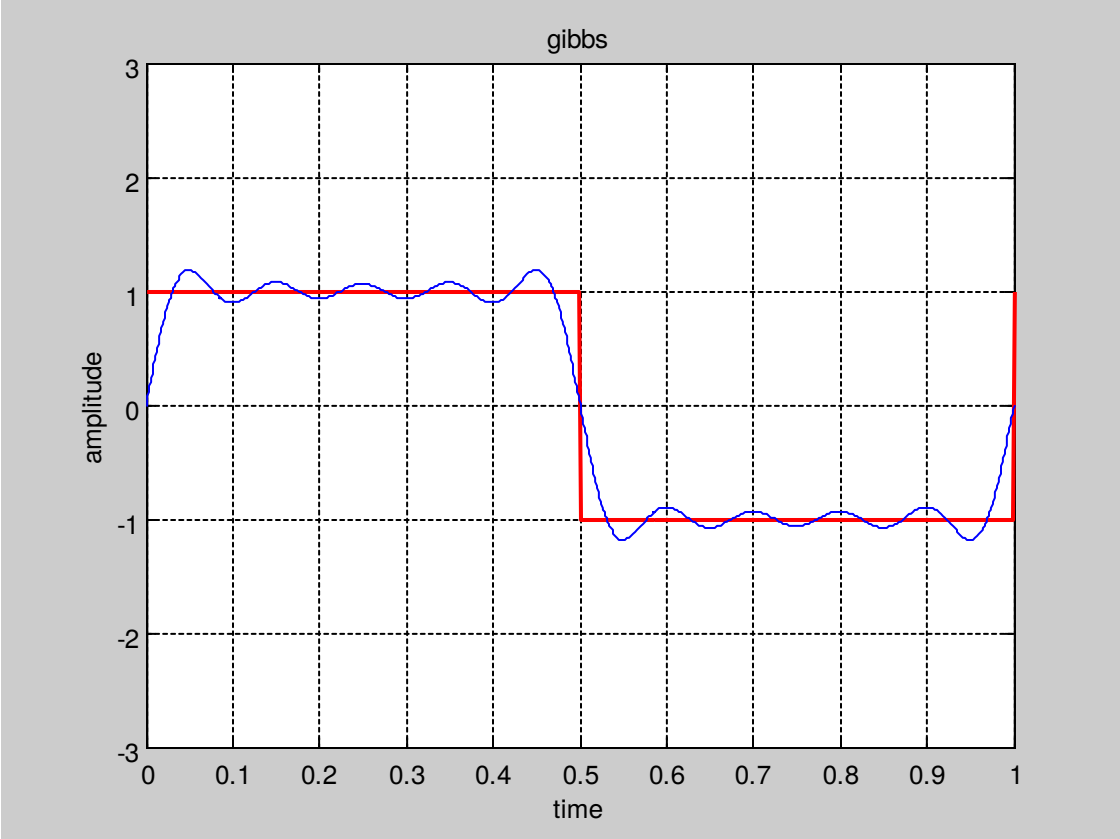
```
clear all;
n=input('type the no of hormonics');
t=0:0.001:1;
y=square(2*pi*t);
plot(t,y,'r','linewidth',2);
axis([0 1 -3 3]);
hold;
sq=zeros(size(t));
for n=1:2:n
    sq=sq+(4/(pi*n)).*sin(2*pi*n*t);
end
plot(t,sq);
grid
xlabel('time');
ylabel('amplitude');
title('gibbs');
```

INPUT:

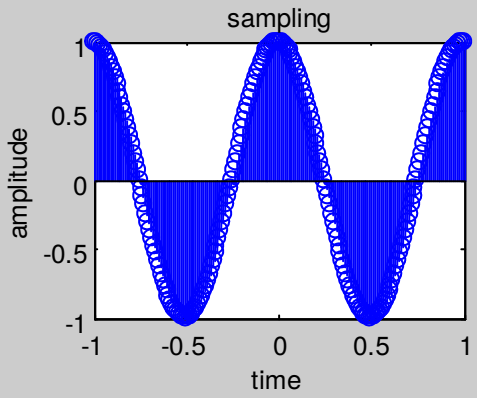
Type the no of harmonics:

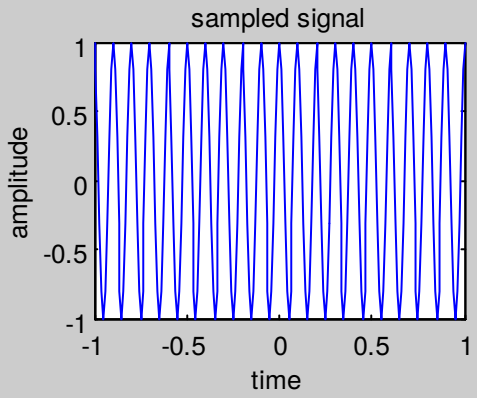
10

FIGURE:




```
%SAMPLING
clc;
clear all;
fs=input('fs=')
ts=1/fs;
n=-1:0.01:1;
x=cos(2*pi.*1.*n)
subplot(2,2,1);
stem(n,x);
xlabel('time');
ylabel('amplitude')
title('sampling')
y=cos(2*pi.*1.*n.*ts)
subplot(2,2,2);
plot(n,y);
xlabel('time');
ylabel('amplitude')
title('sampled signal')
```


sampling
amplitude
time
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1


sampled signal
amplitude
time
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1