SIGNAL PROCESSING CODING ASSIGNMENT

SCILAB

LIFE LONG LEARNING ASSESSMENT(LLT)

By SRIDHARAN S
RA2411004010179

Exercise 1: Generate the basic deterministic CT and DT signals such as unit impulse, unit step, sinusoidal and exponential signals.

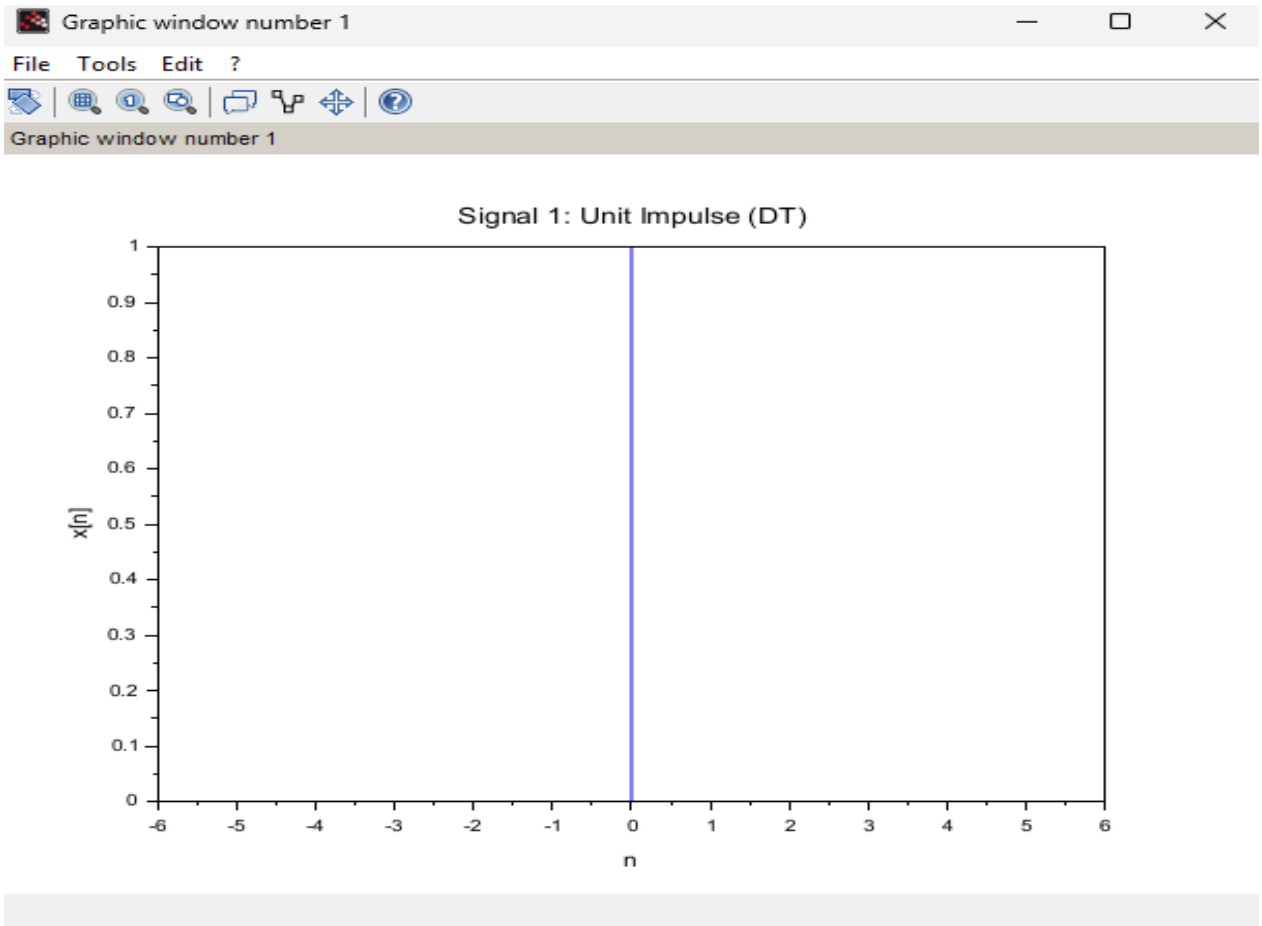
Code:

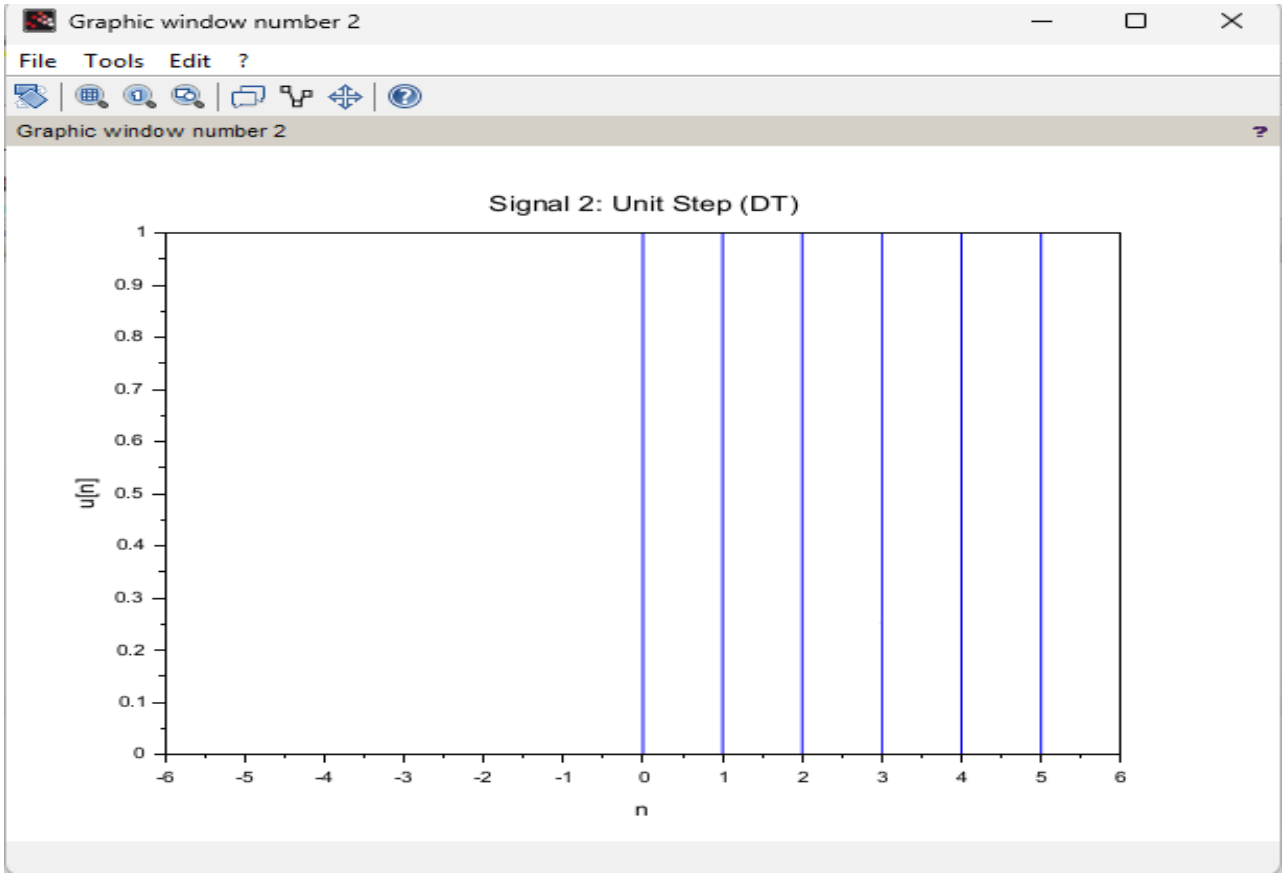
```
// EXERCISE 1: Basic CT and DT Signals
clc;
clear;
// -------- SIGNAL 1: Unit Impulse (DT) --------
scf(1); clf();
n =-5:5;
x = (n == 0);
// Draw stem manually
for i = 1:length(n)
   plot([n(i), n(i)], [0, x(i)]);
end
plot(n, zeros(1, length(n)), 'k');
xtitle('Signal 1: Unit Impulse (DT)', 'n', 'x[n]');
// -------- SIGNAL 2: Unit Step (DT) --------
scf(2); clf();
n =-5:5;
u = (n >= 0);
for i = 1:length(n)
   plot([n(i), n(i)], [0, u(i)]);
end
```

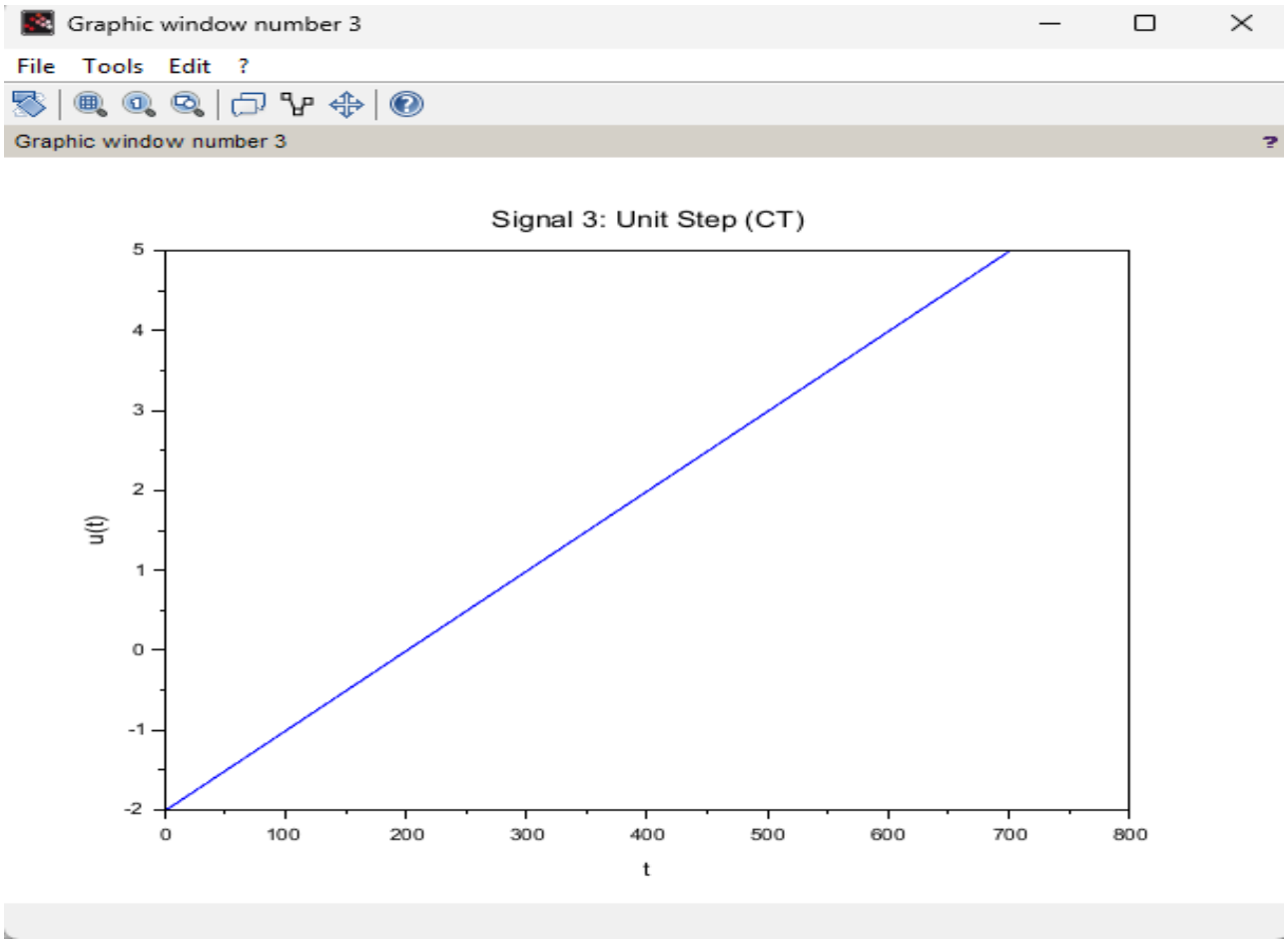
```
plot(n, zeros(1, length(n)), 'k');
xtitle('Signal 2: Unit Step (DT)', 'n', 'u[n]');
// -------- SIGNAL 3: Unit Step (CT) --------
scf(3); clf();
t =-2:0.01:5;
ct_step = (t >= 0);
plot(t, ct_step);
xtitle('Signal 3: Unit Step (CT)', 't', 'u(t)');
// -------- SIGNAL 4: Sinusoidal (CT) --------
scf(4); clf();
t = 0:0.01:2;
y = sin(2 * %pi * 5 * t);
plot(t, y);
xtitle('Signal 4: Sinusoidal Signal (CT)- 5Hz', 't', 'sin(2*pi*5*t)');
// -------- SIGNAL 5: Sinusoidal (DT) --------
scf(5); clf();
n = 0:20;
y_dt = sin(2 * %pi * 0.1 * n);
for i = 1:length(n)
   plot([n(i), n(i)], [0, y_dt(i)]);
end
plot(n, zeros(1, length(n)), 'k');
xtitle('Signal 5: Sinusoidal Signal (DT)', 'n', 'sin[n]');
// -------- SIGNAL 6: Exponential Decaying (CT) --------
scf(6); clf();
t = 0:0.01:3;
exp_ct = exp(-2 * t);
plot(t, exp_ct);
xtitle('Signal 6: Decaying Exponential (CT)', 't', 'e^(-2t)');
```

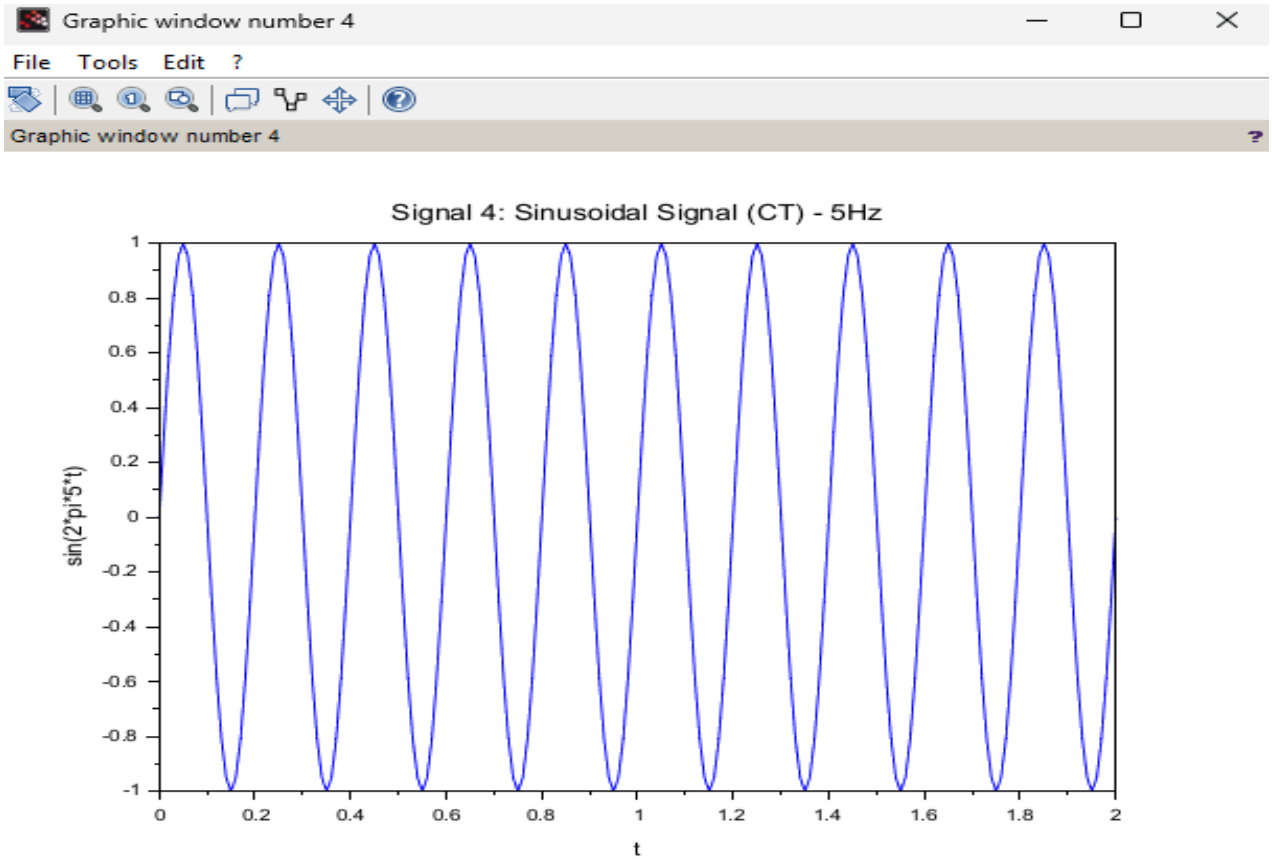
```
// -------- SIGNAL 7: Exponential (DT) --------
scf(7); clf();
n = 0:15;
exp_dt = (0.8).^n;
for i = 1:length(n)
   plot([n(i), n(i)], [0, exp_dt(i)]);
end
plot(n, zeros(1, length(n)), 'k');
xtitle('Signal 7: Exponential Signal (DT)', 'n', '0.8^n');
disp('Exercise 1 Complete');
```

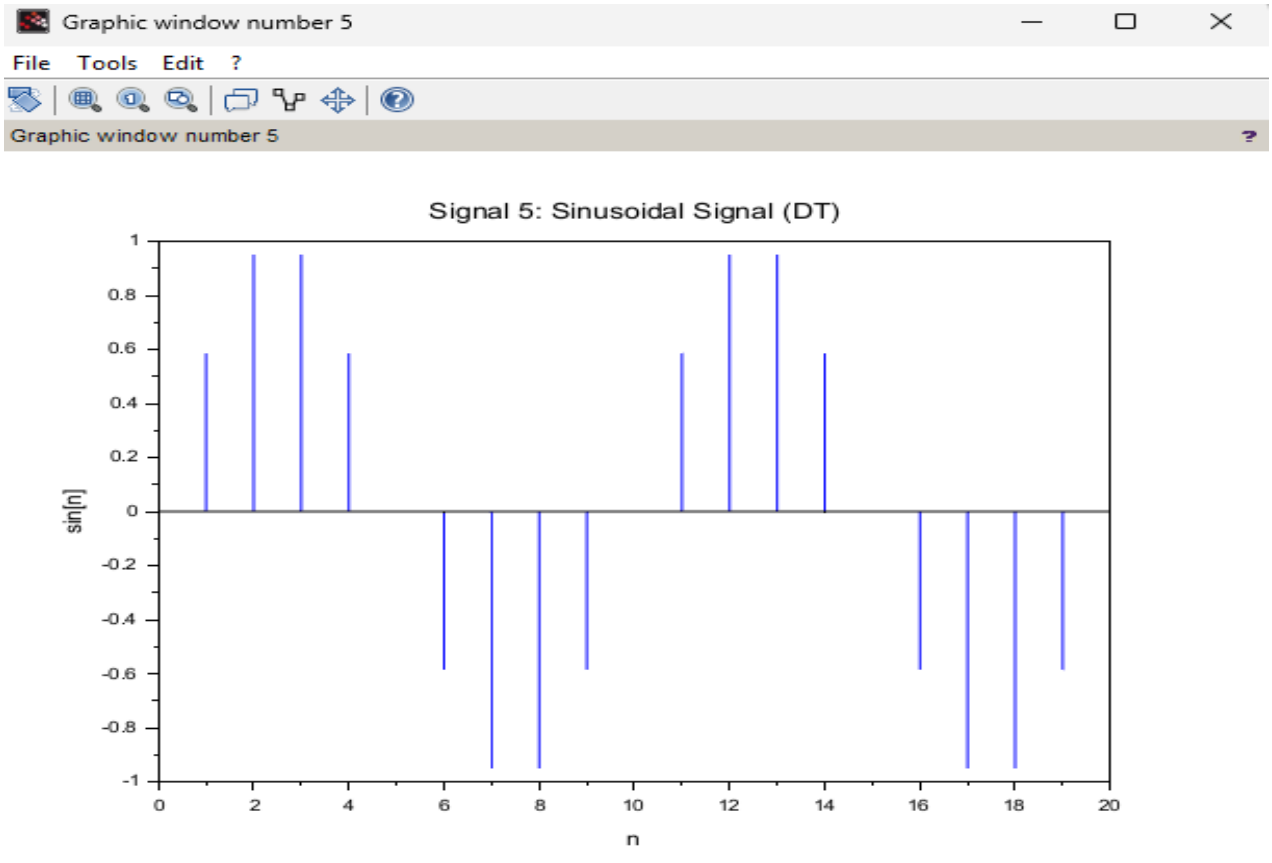
OUTPUT:

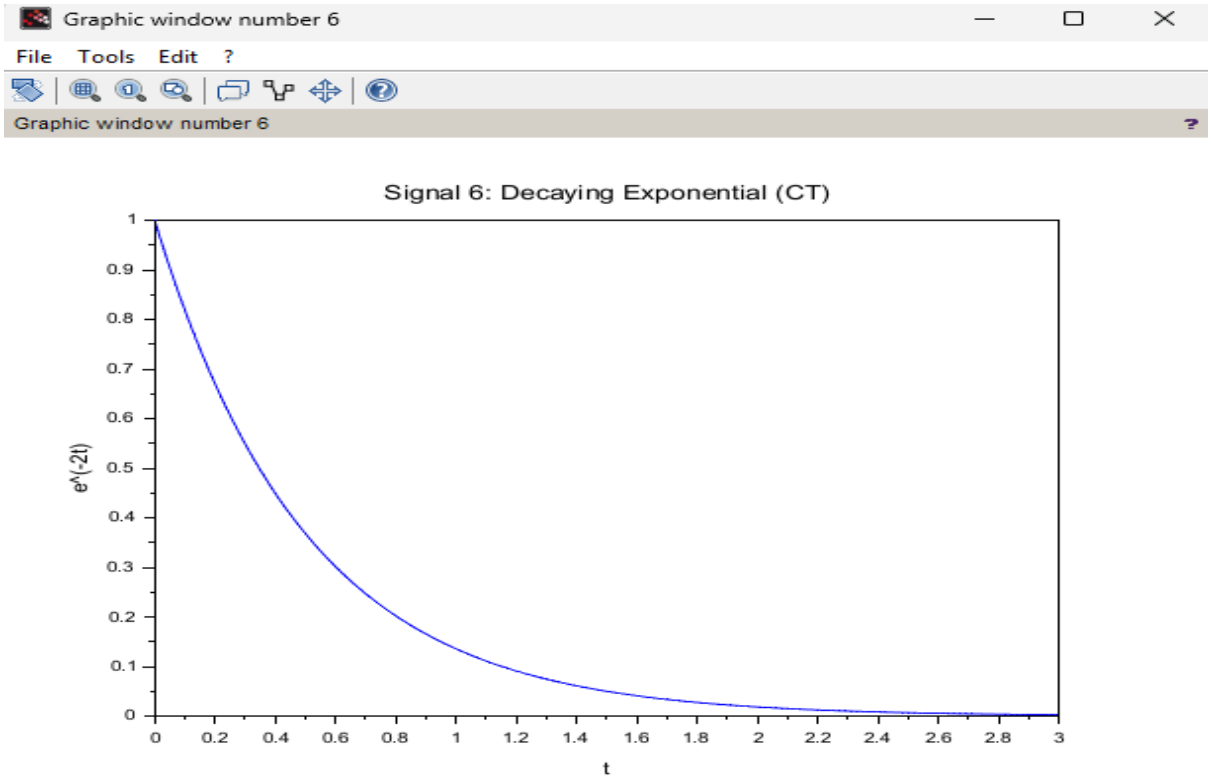


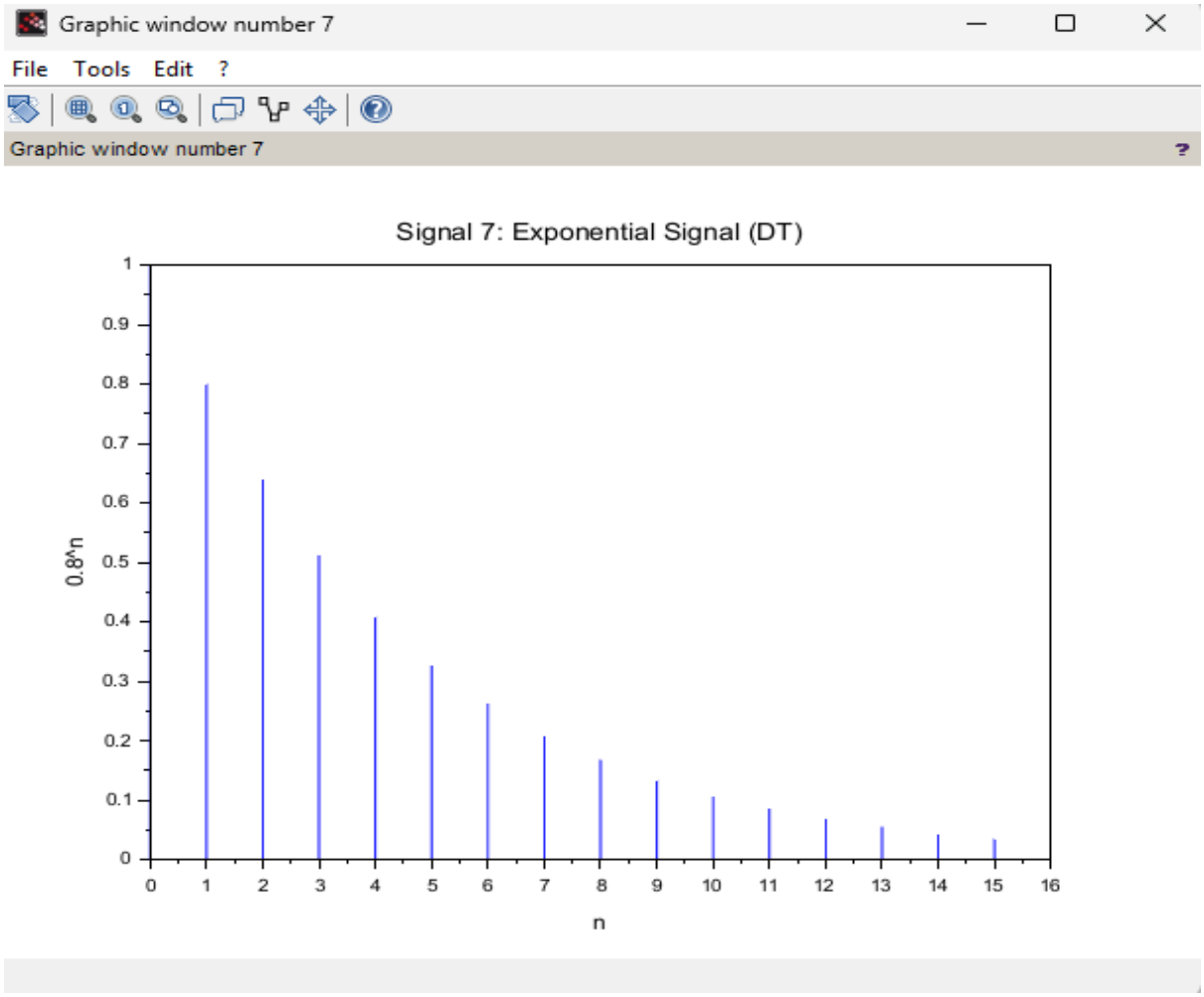

Graphic window number 2
File Tools Edit ?
Graphic window number 2
Signal 2: Unit Step (DT)
u[n]
n
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6


Graphic window number 3
File Tools Edit ?
Graphic window number 3
Signal 3: Unit Step (CT)
u(t)
t
5
4
3
2
1
0
-1
-2
0 100 200 300 400 500 600 700 800


Graphic window number 4
File Tools Edit ?
Graphic window number 4
Signal 4: Sinusoidal Signal (CT) - 5Hz
sin(2*pi*5*t)
t
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


Graphic window number 5
File Tools Edit ?
Graphic window number 5
Signal 5: Sinusoidal Signal (DT)
sin[n]
n
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2
4
6
8
10
12
14
16
18
20


Graphic window number 6
File Tools Edit ?
Graphic window number 6
Signal 6: Decaying Exponential (CT)
e^(-2t)
t


Graphic window number 7
File Tools Edit ?
Graphic window number 7
Signal 7: Exponential Signal (DT)
0.8^n
n

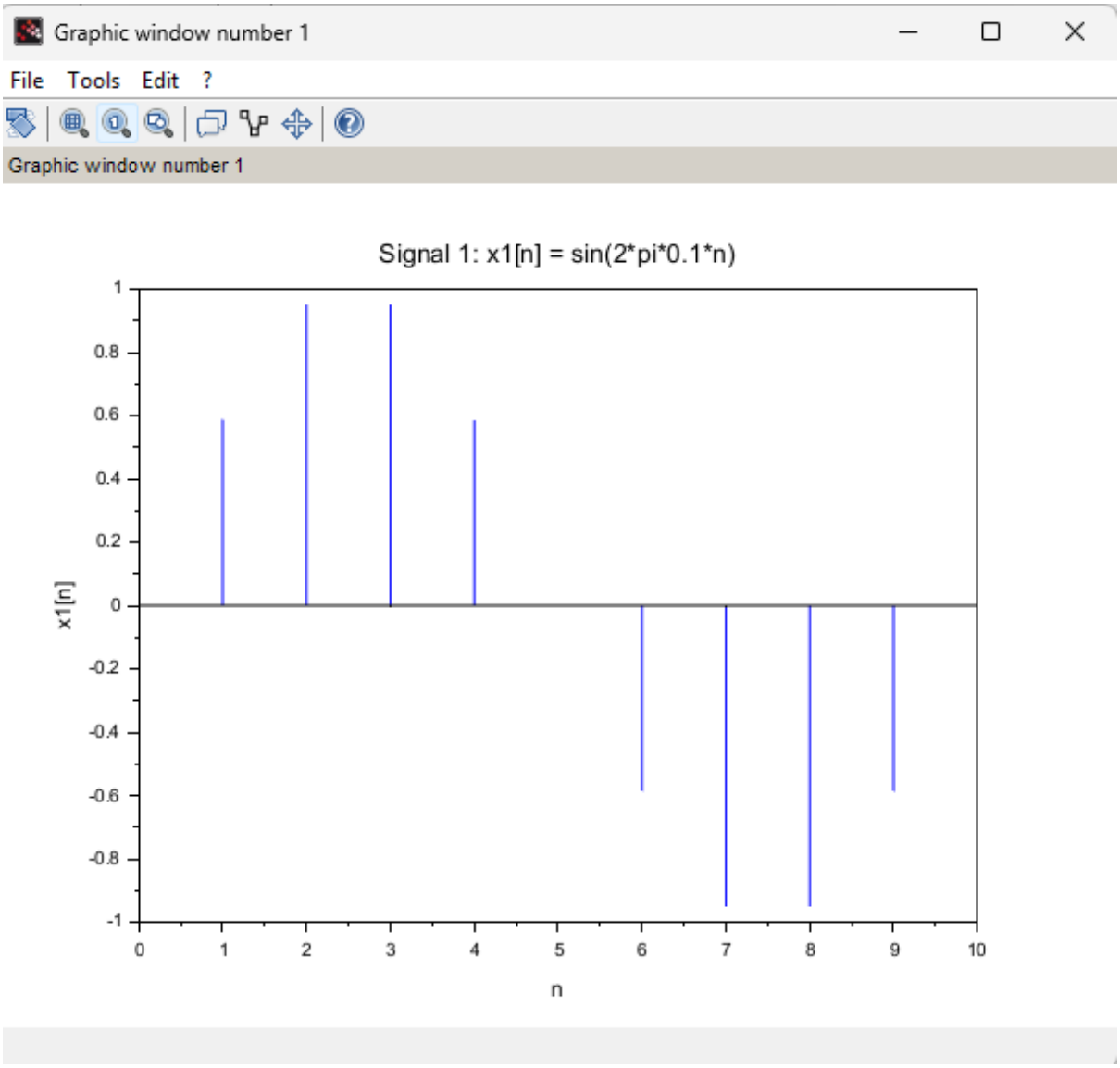
Exercise 2: Perform the basic operations such as addition , multiplication .

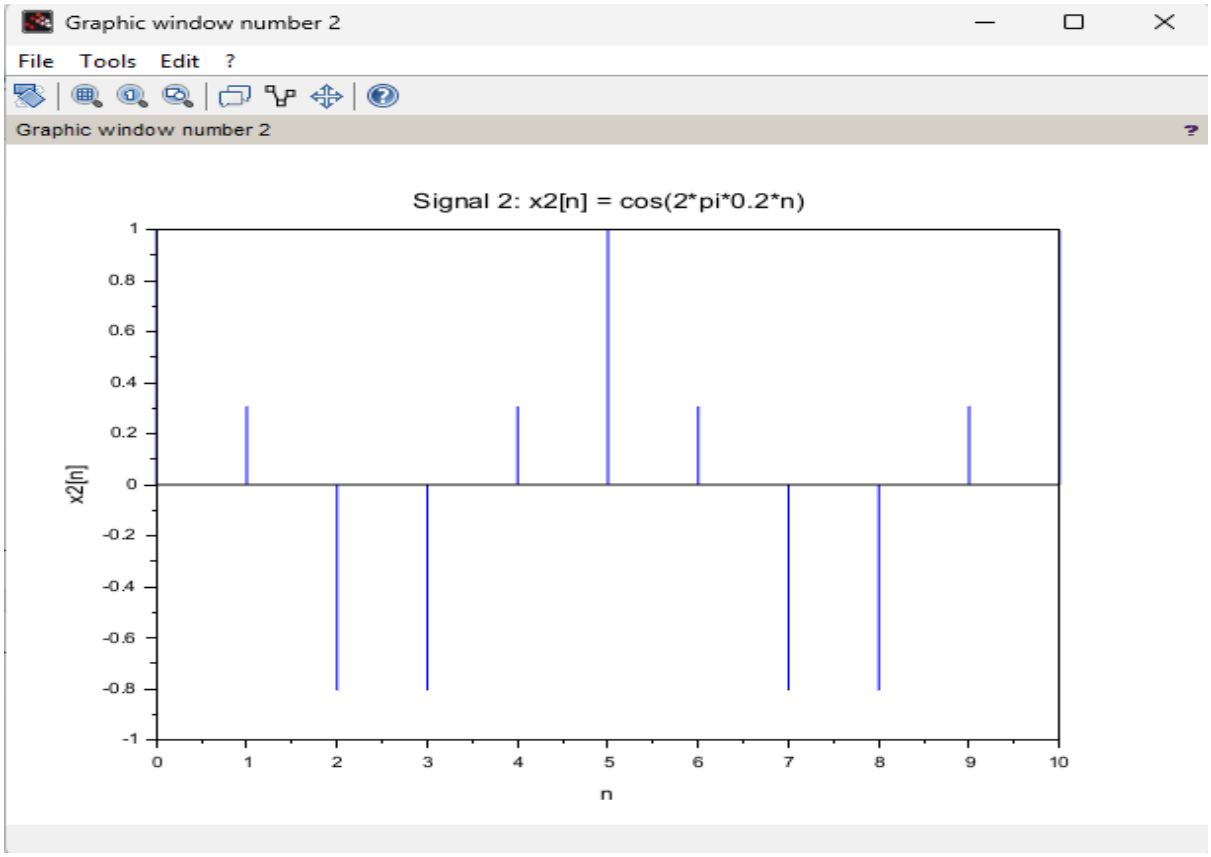
Code:

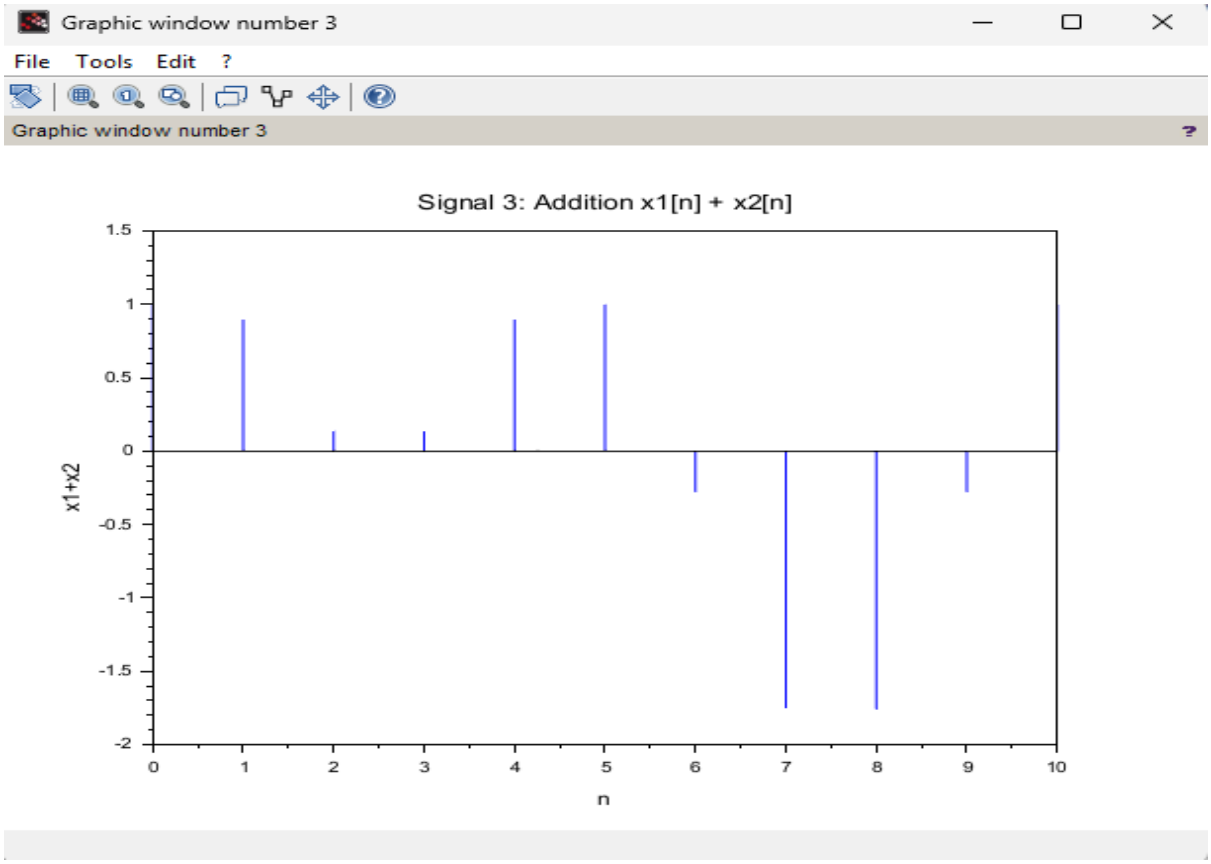
```
// EXERCISE 2: Basic Operations - Addition & Multiplication
clc;
clear;
n = 0:10;
x1 = sin(2 * %pi * 0.1 * n);
x2 = cos(2 * %pi * 0.2 * n);
// -------- SIGNAL 1: x1[n] --------
scf(1); clf();
for i = 1:length(n)
   plot([n(i), n(i)], [0, x1(i)]);
end
plot(n, zeros(1, length(n)), 'k');
xtitle('Signal 1: x1[n] = sin(2*pi*0.1*n)', 'n', 'x1[n]');
// -------- SIGNAL 2: x2[n] --------
scf(2); clf();
for i = 1:length(n)
   plot([n(i), n(i)], [0, x2(i)]);
end
plot(n, zeros(1, length(n)), 'k');
xtitle('Signal 2: x2[n] = cos(2*pi*0.2*n)', 'n', 'x2[n]');
// -------- SIGNAL 3: Addition --------
x_add = x1 + x2;
scf(3); clf();
for i = 1:length(n)
   plot([n(i), n(i)], [0, x_add(i)]);
end
plot(n, zeros(1, length(n)), 'k');
xtitle('Signal 3: Addition x1[n] + x2[n]', 'n', 'x1+x2');
```

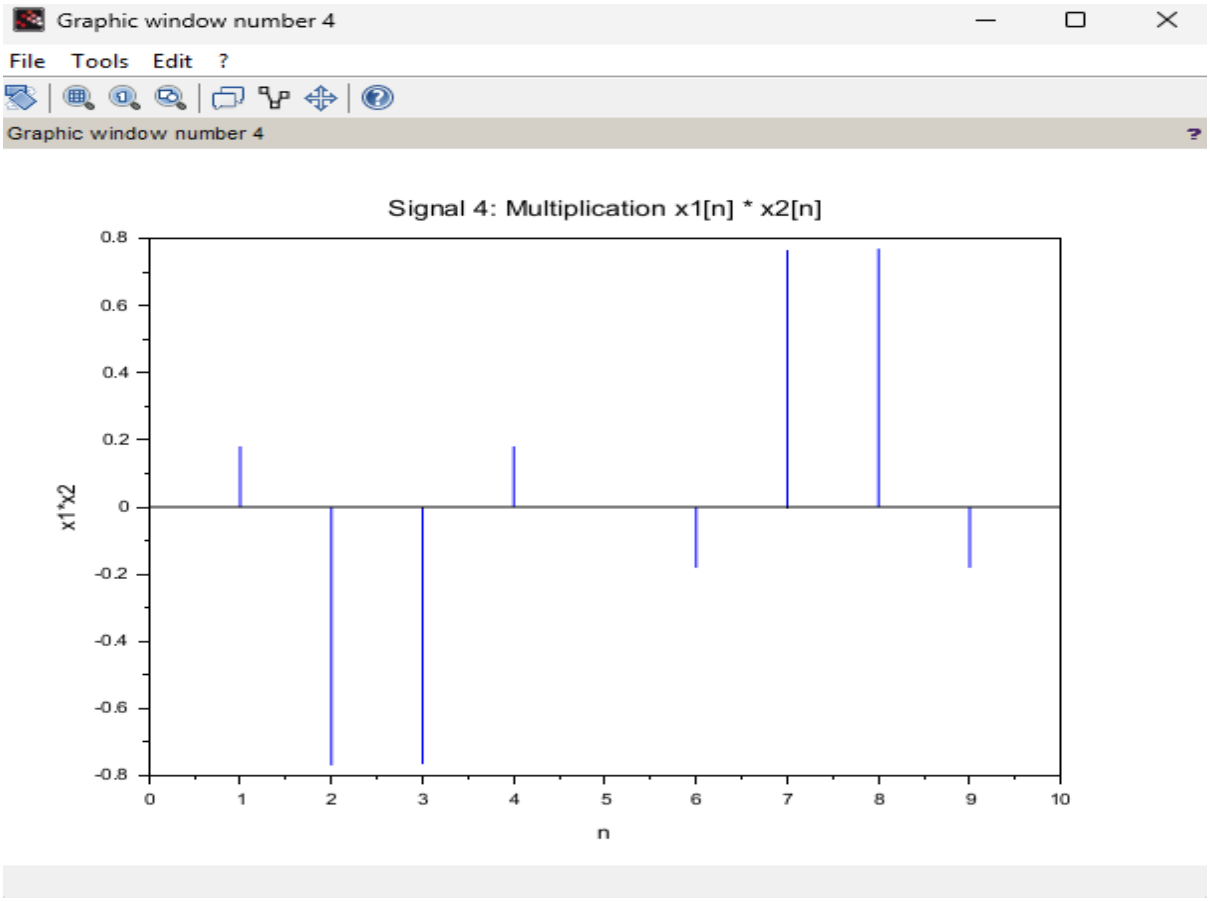
```
// -------- SIGNAL 4: Multiplication --------
x_mul = x1 .* x2;
scf(4); clf();
for i = 1:length(n)
   plot([n(i), n(i)], [0, x_mul(i)]);
end
plot(n, zeros(1, length(n)), 'k');
xtitle('Signal 4: Multiplication x1[n] * x2[n]', 'n', 'x1*x2');
disp('Exercise 2 Complete!');
```

OUTPUT:




Graphic window number 2
File Tools Edit ?
Graphic window number 2
Signal 2: x2[n] = cos(2*pi*0.2*n)
x2[n]
n
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0 1 2 3 4 5 6 7 8 9 10


Graphic window number 3
File Tools Edit ?
Graphic window number 3
Signal 3: Addition x1[n] + x2[n]
x1+x2
n
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0 1 2 3 4 5 6 7 8 9 10



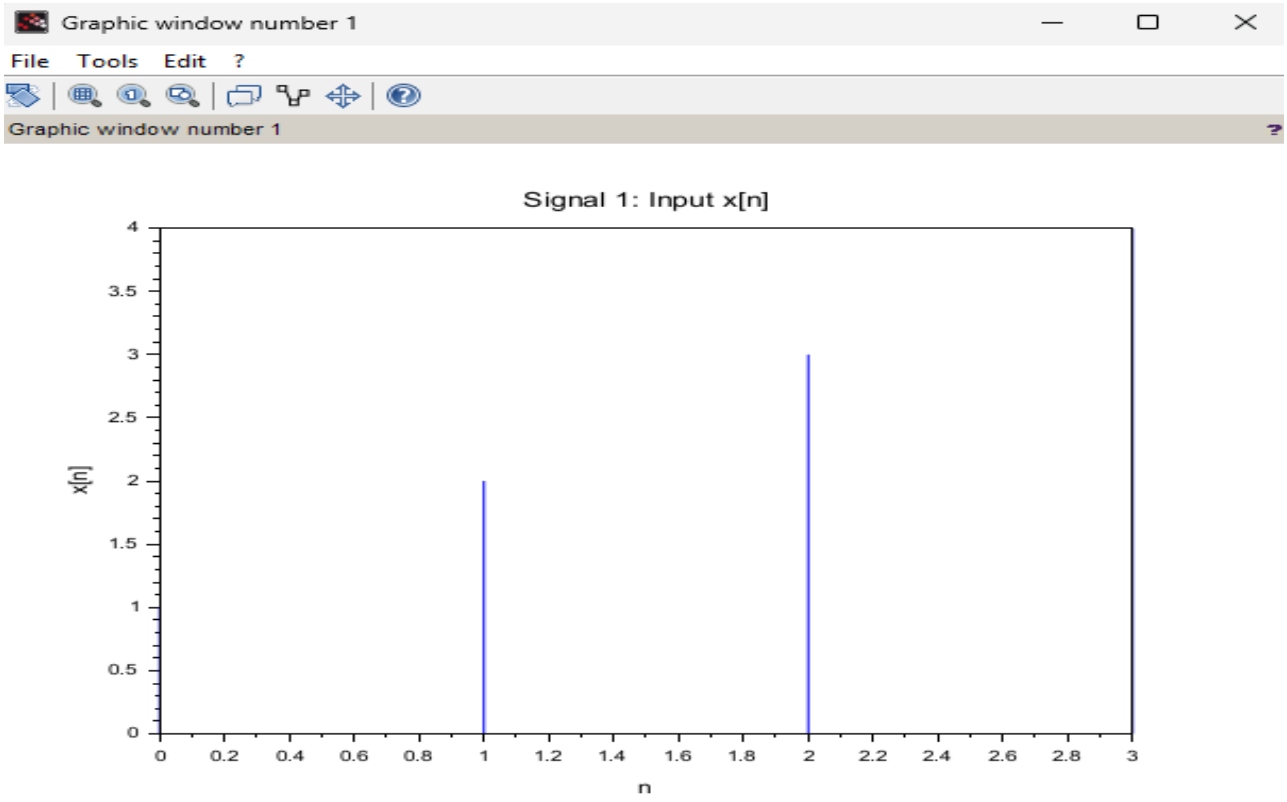
Exercise 3: Perform linear convolution of DT signals.

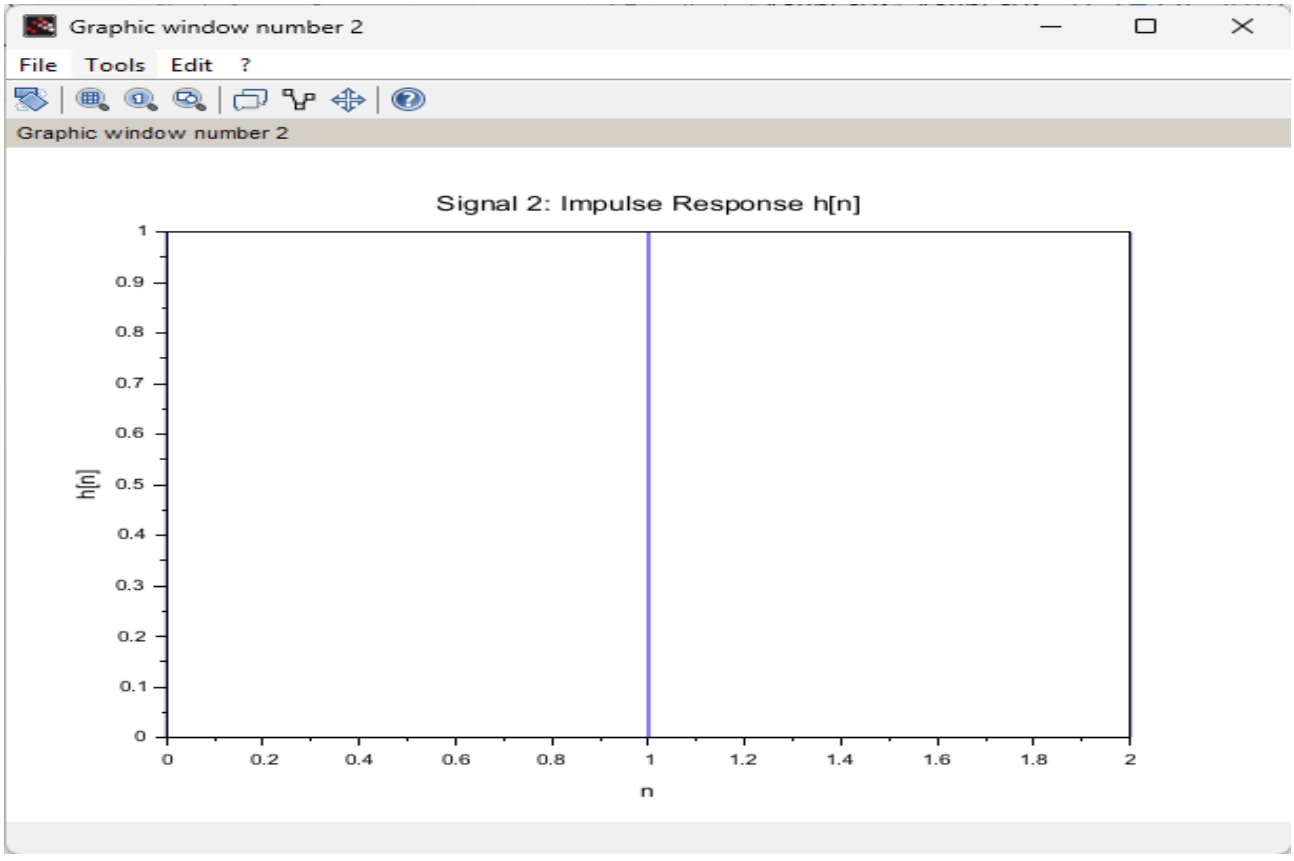
Code:

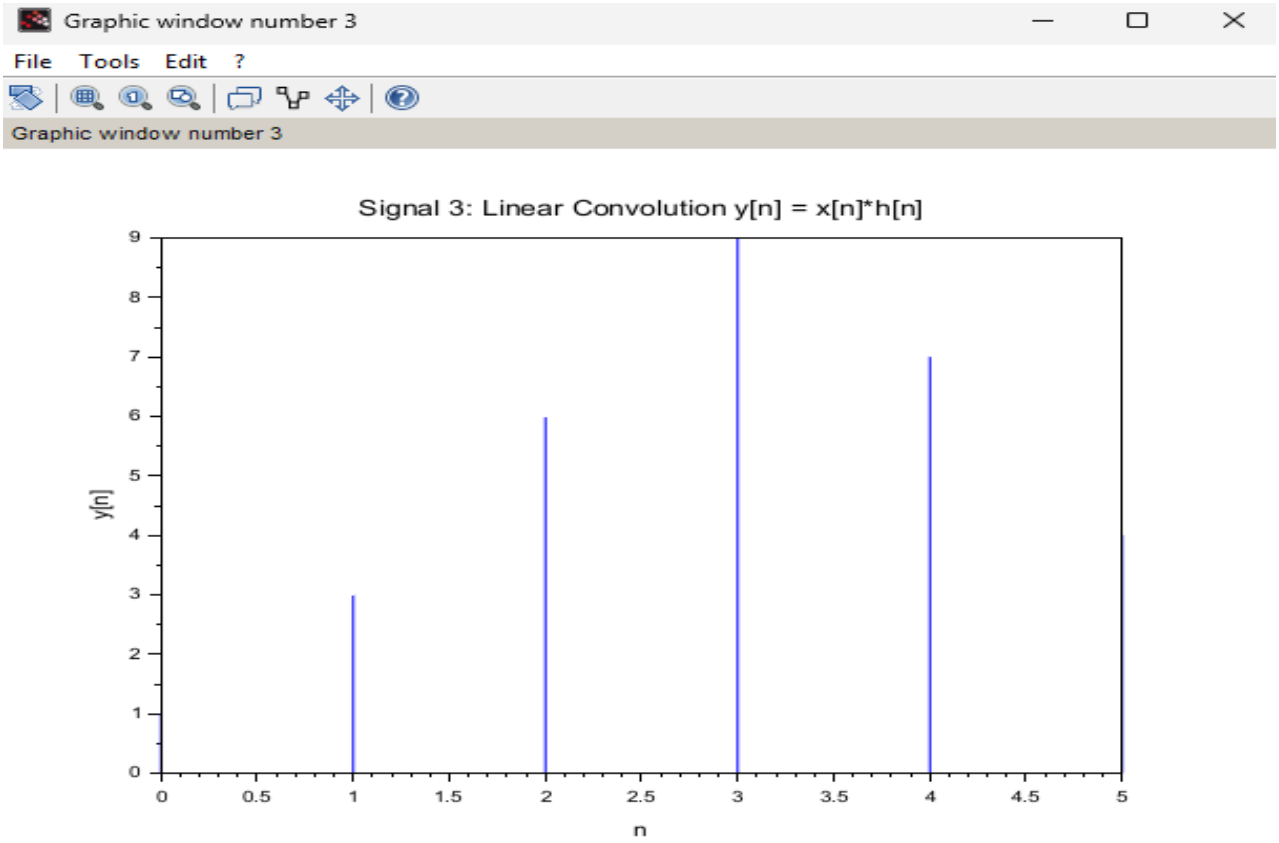
```
// EXERCISE 3: Linear Convolution of DT Signals
clc;
clear;
x = [1, 2, 3, 4];
h = [1, 1, 1];
y = convol(x, h);
nx = 0:length(x)-1;
nh = 0:length(h)-1;
ny = 0:length(y)-1;
// -------- Input Signal x[n] --------
scf(1); clf();
for i = 1:length(nx)
   plot([nx(i), nx(i)], [0, x(i)]);
```

```
end
plot(nx, zeros(1, length(nx)), 'k');
xtitle('Signal 1: Input x[n]', 'n', 'x[n]');
// -------- Impulse Response h[n] --------
scf(2); clf();
for i = 1:length(nh)
   plot([nh(i), nh(i)], [0, h(i)]);
end
plot(nh, zeros(1, length(nh)), 'k');
xtitle('Signal 2: Impulse Response h[n]', 'n', 'h[n]');
// -------- Linear Convolution y[n] --------
scf(3); clf();
for i = 1:length(ny)
   plot([ny(i), ny(i)], [0, y(i)]);
end
plot(ny, zeros(1, length(ny)), 'k');
xtitle('Signal 3: Linear Convolution y[n] = x[n]*h[n]', 'n', 'y[n]');
disp('Linear Convolution Result:');
disp(y);
disp('Exercise 3 Complete!');
```

OUTPUT:


Graphic window number 1
File Tools Edit ?
Graphic window number 1
Signal 1: Input x[n]
x[n]
n
0 0.5 1 1.5 2 2.5 3 3.5 4
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3


Graphic window number 2
File Tools Edit ?
Graphic window number 2
Signal 2: Impulse Response h[n]
h[n]
n
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2



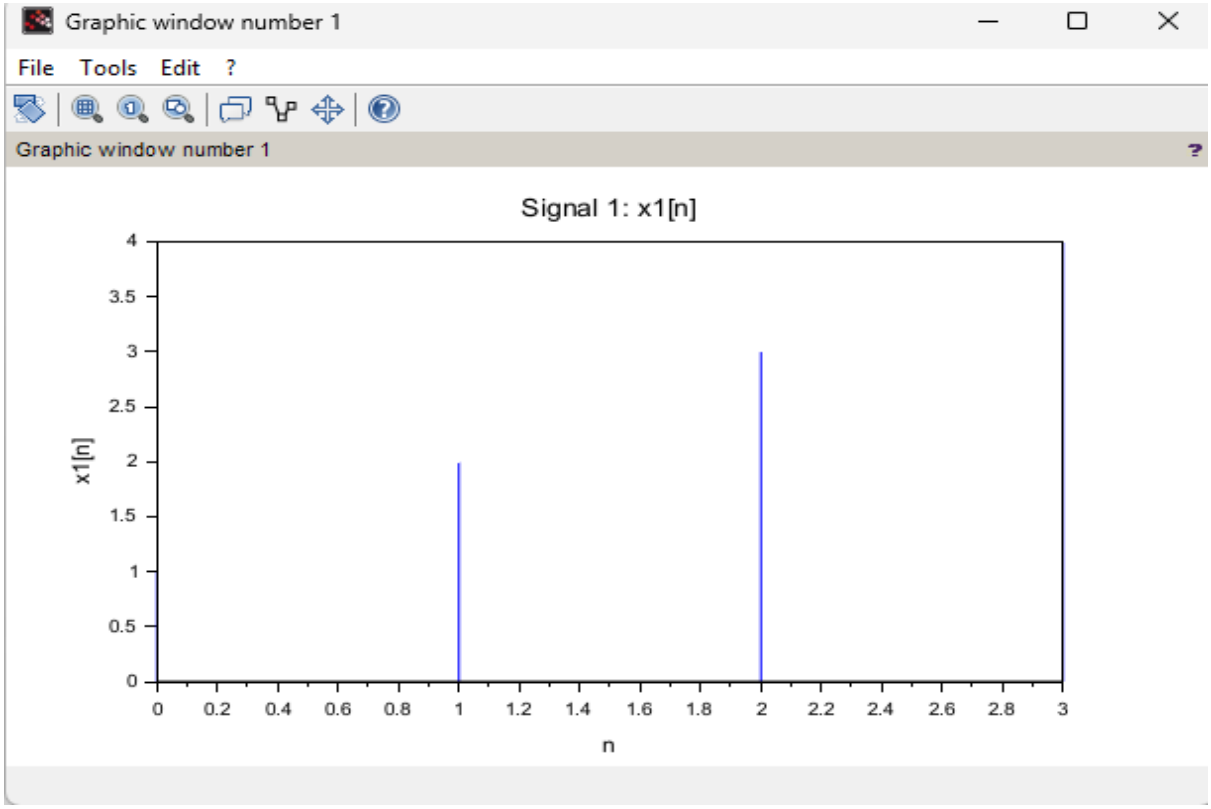
Exercise 4: Perform Circular convolution of DT signals.

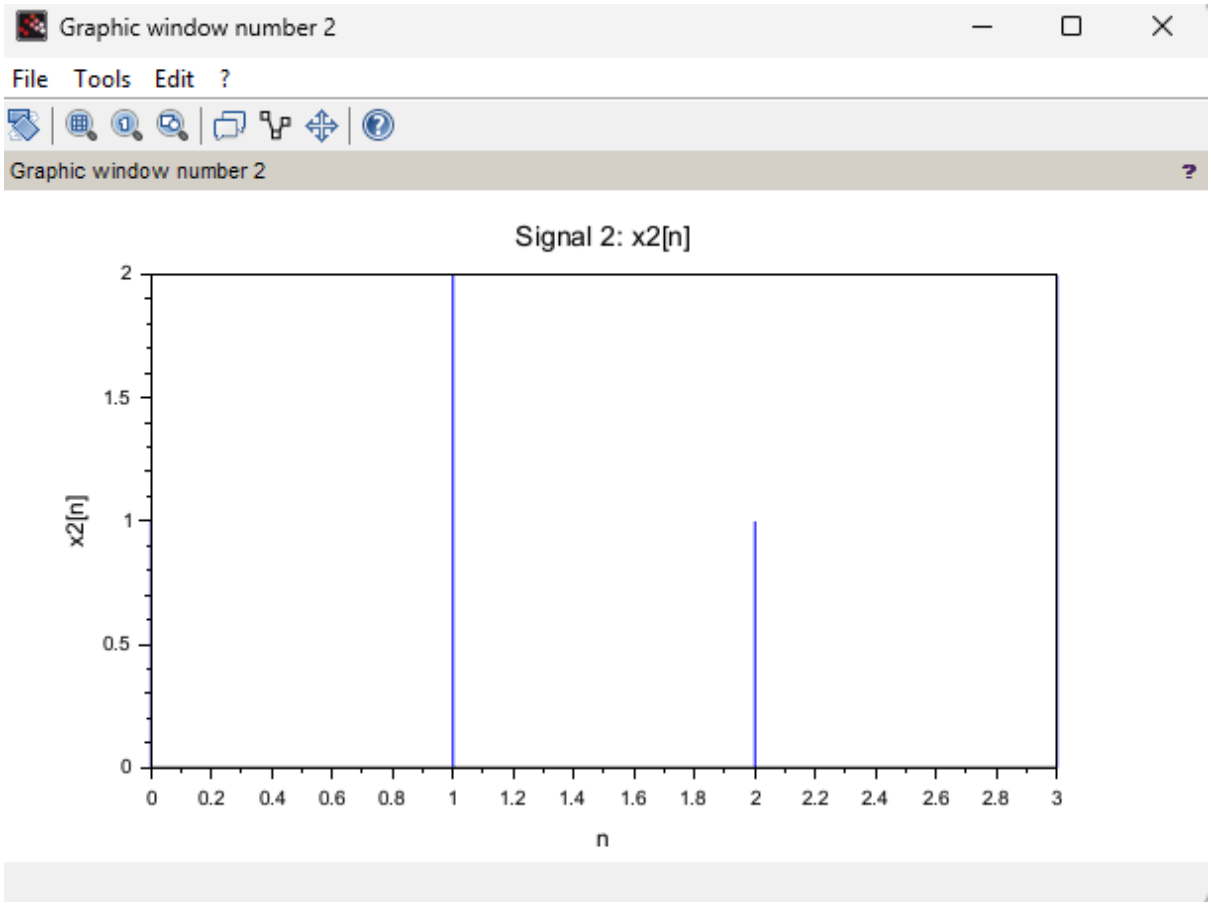
Code:

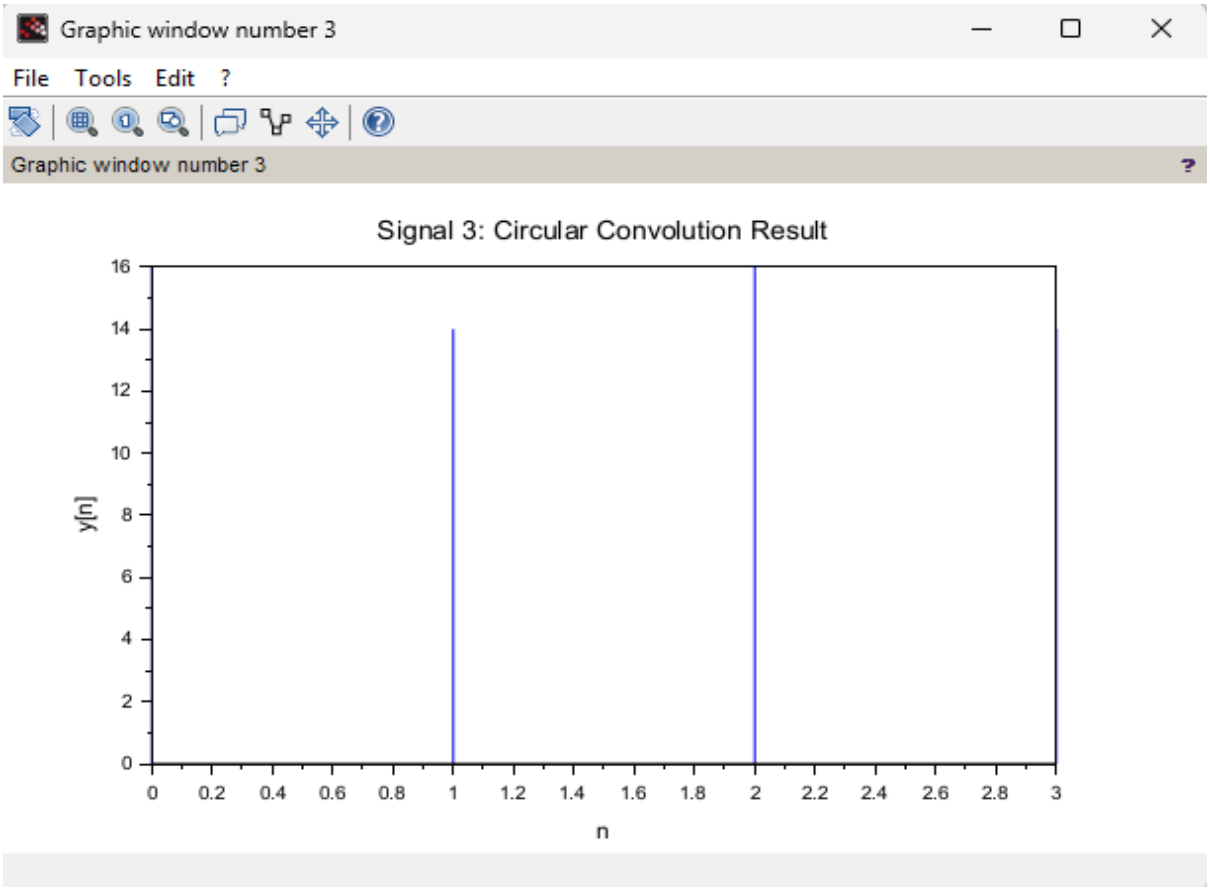
```
// EXERCISE 4: Circular Convolution of DT Signals
clc;
clear;
x1 = [1, 2, 3, 4];
x2 = [1, 2, 1, 2];
N = length(x1);
// Circular convolution using DFT
X1 = fft(x1);
X2 = fft(x2);
Y  = X1 .* X2;
y_circ = real(ifft(Y));
n = 0:N-1;
```

```
// -------- Signal x1[n] --------
scf(1); clf();
for i = 1:length(n)
   plot([n(i), n(i)], [0, x1(i)]);
end
plot(n, zeros(1, length(n)), 'k');
xtitle('Signal 1: x1[n]', 'n', 'x1[n]');
// -------- Signal x2[n] --------
scf(2); clf();
for i = 1:length(n)
   plot([n(i), n(i)], [0, x2(i)]);
end
plot(n, zeros(1, length(n)), 'k');
xtitle('Signal 2: x2[n]', 'n', 'x2[n]');
// -------- Circular Convolution Result --------
scf(3); clf();
for i = 1:length(n)
   plot([n(i), n(i)], [0, y_circ(i)]);
end
plot(n, zeros(1, length(n)), 'k');
xtitle('Signal 3: Circular Convolution Result', 'n', 'y[n]');
disp('Circular Convolution Result:');
disp(y_circ);
disp('Exercise 4 Complete!');
```

OUTPUT:


Graphic window number 1
File Tools Edit ?
Graphic window number 1
Signal 1: x1[n]
x1[n]
n
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3
0 0.5 1 1.5 2 2.5 3 3.5 4


Graphic window number 2
File Tools Edit ?
Graphic window number 2
Signal 2: x2[n]
x2[n]
n
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3
0 0.5 1 1.5 2



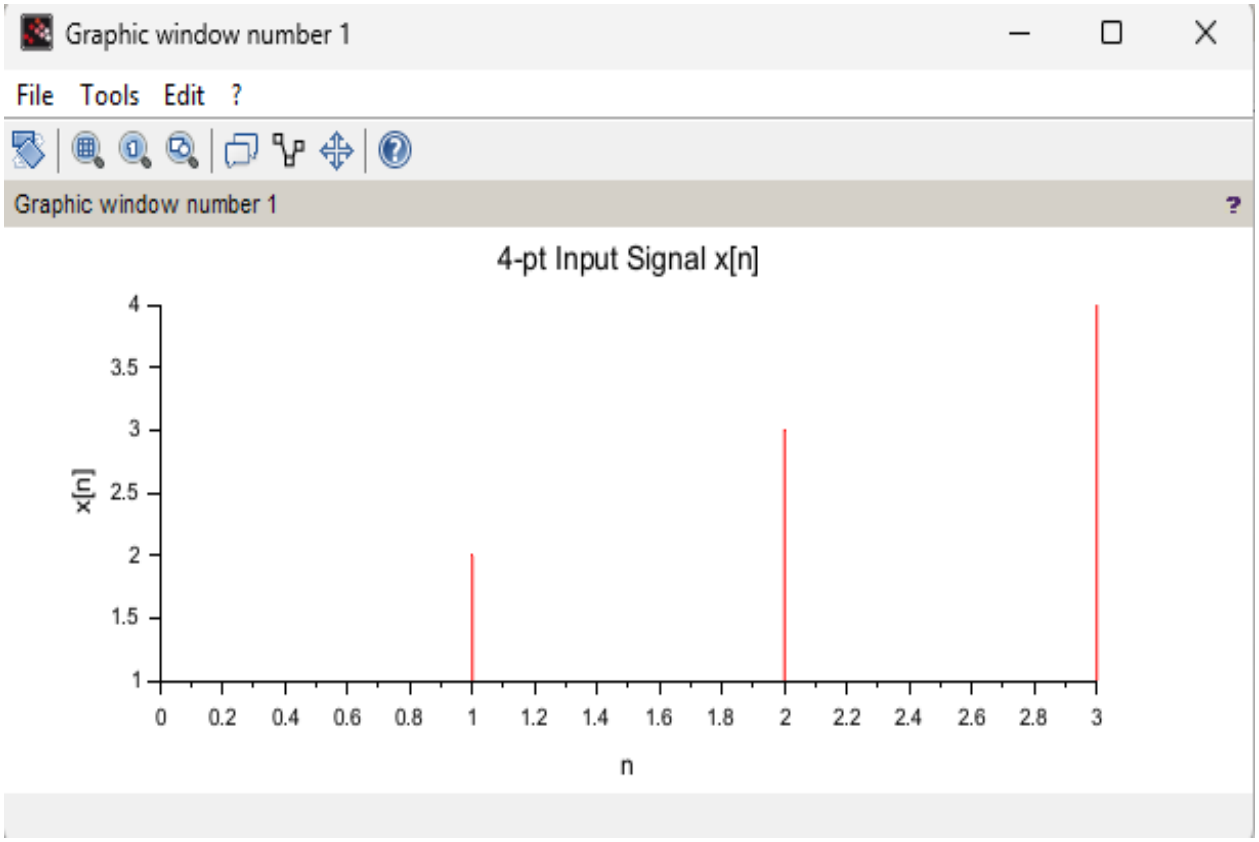
Exercise 5: Perform 4 point and 8 point DFT.

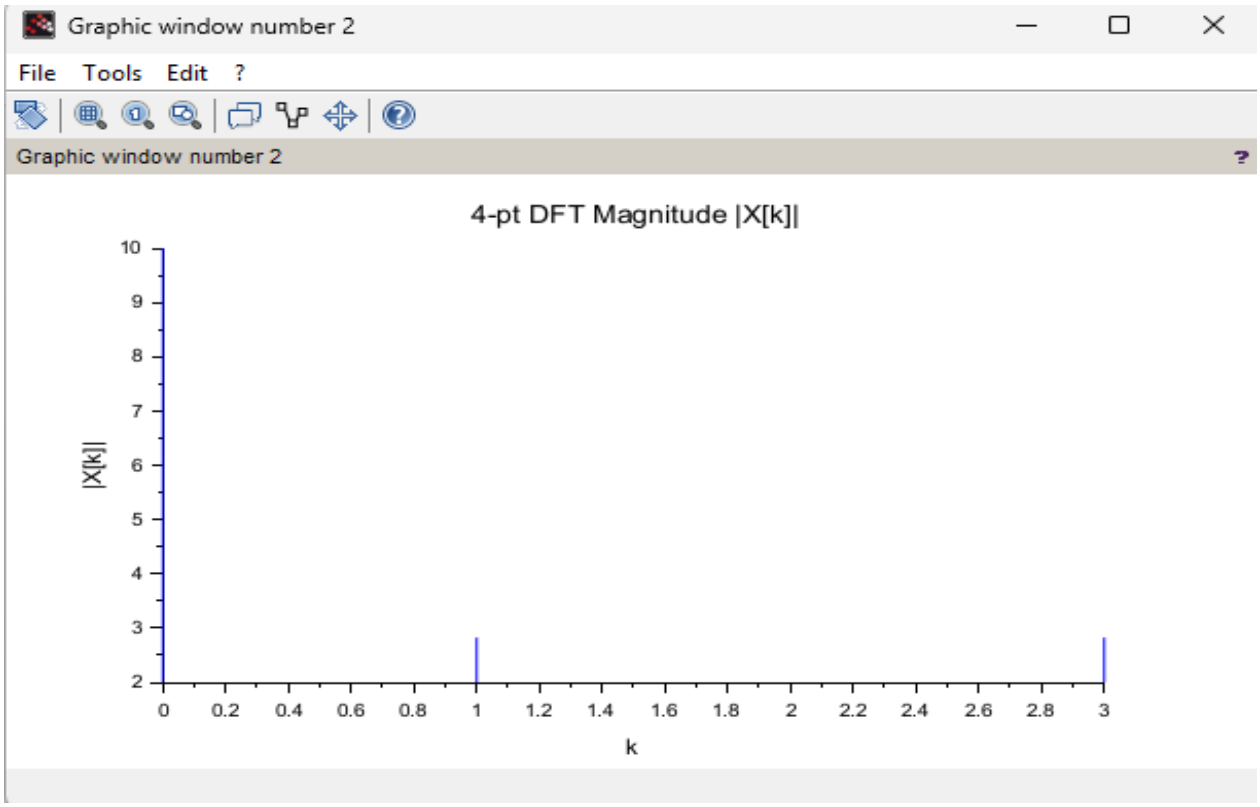
CODE:

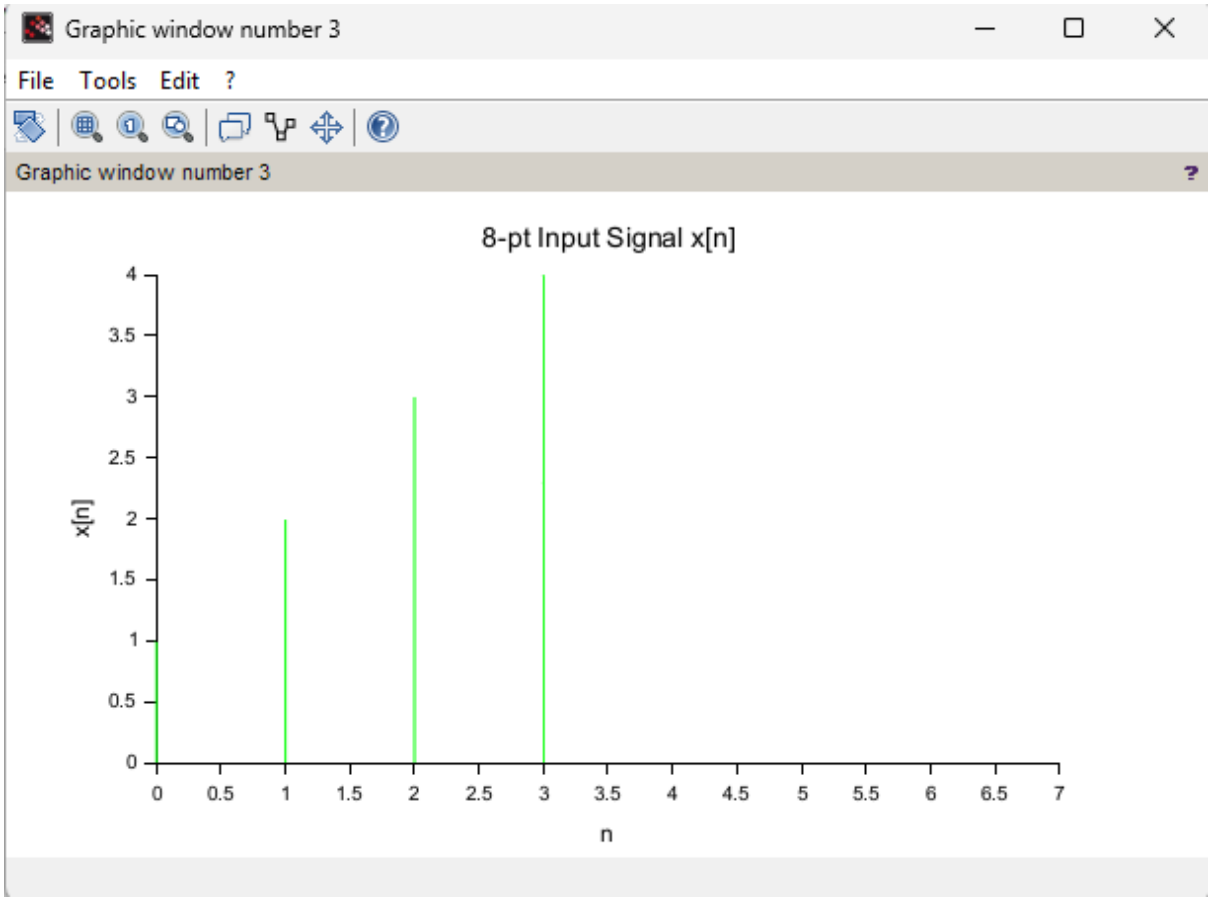
```
// EXERCISE 5: 4-Point and 8-Point DFT
clc;
clear;
// -------- 4-Point DFT --------
x4 = [1, 2, 3, 4];
N4 = 4;
X4 = fft(x4);          // safer version
mag4 = abs(X4);
k4 = 0:N4-1;
// -------- Plot: x4[n] --------
scf(1); clf();
plot2d3(k4, x4);
xtitle('4-pt Input Signal x[n]', 'n', 'x[n]');
```

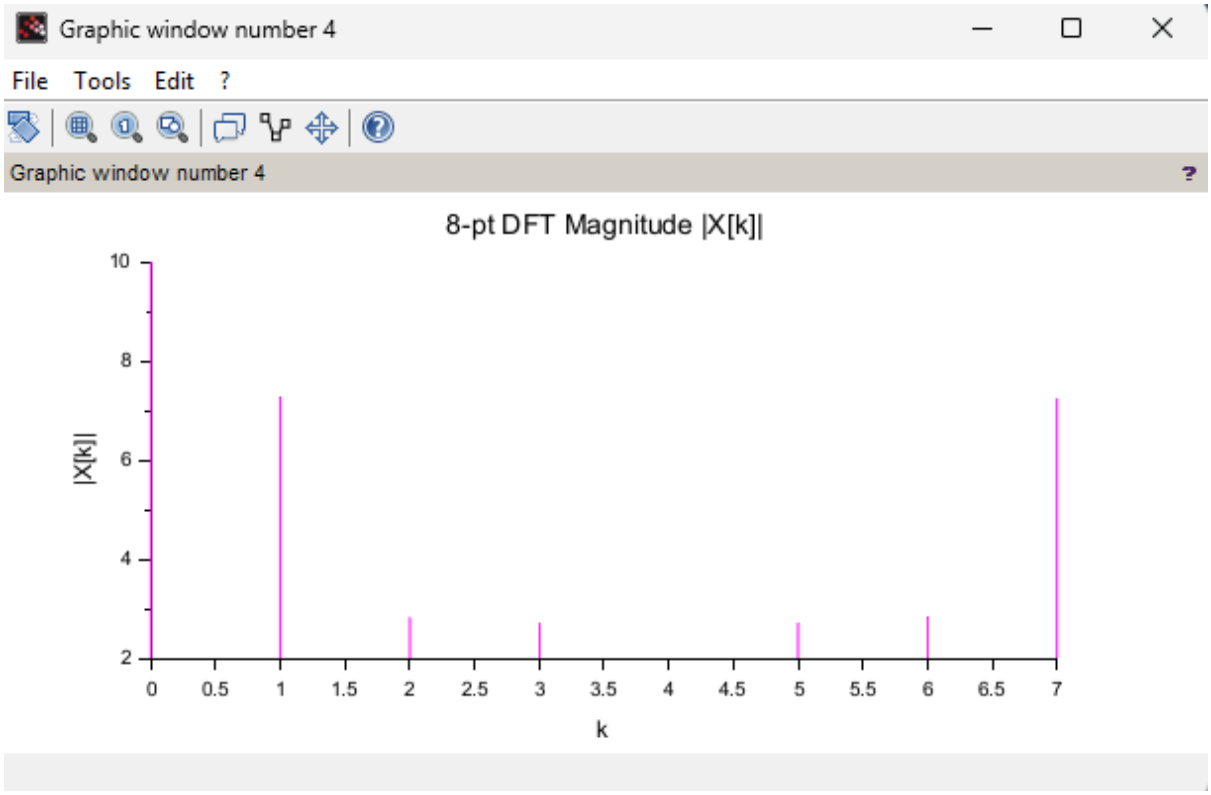
```
xgrid();
// -------- Plot: |X4[k]| --------
scf(2); clf();
plot2d3(k4, mag4);
xtitle('4-pt DFT Magnitude |X[k]|', 'k', '|X[k]|');
xgrid();
// -------- 8-Point DFT --------
x8 = [1, 2, 3, 4, 0, 0, 0, 0];
N8 = 8;
X8 = fft(x8);          // safer version
mag8 = abs(X8);
k8 = 0:N8-1;
// -------- Plot: x8[n] --------
scf(3); clf();
plot2d3(k8, x8);
xtitle('8-pt Input Signal x[n]', 'n', 'x[n]');
xgrid();
// -------- Plot: |X8[k]| --------
scf(4); clf();
plot2d3(k8, mag8);
xtitle('8-pt DFT Magnitude |X[k]|', 'k', '|X[k]|');
xgrid();
// -------- Display Results --------
disp('4-Point DFT Result:');
disp(X4);
disp('8-Point DFT Result:');
disp(X8);
disp('Exercise 5 Complete!');
```

OUTPUT:


Graphic window number 1
File Tools Edit ?
Graphic window number 1
4-pt Input Signal x[n]
x[n]
n
4
3.5
3
2.5
2
1.5
1
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3


Graphic window number 2
File Tools Edit ?
Graphic window number 2
4-pt DFT Magnitude |X[k]|
|X[k]|
k
10
9
8
7
6
5
4
3
2
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3


Graphic window number 3
File Tools Edit ?
Graphic window number 3
8-pt Input Signal x[n]
x[n]
n
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 6.5 7
0 0.5 1 1.5 2 2.5 3 3.5 4


Graphic window number 4
File Tools Edit ?
Graphic window number 4
8-pt DFT Magnitude |X[k]|
|X[k]|
k
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 6.5 7
2 4 6 8 10