Name: Mostafa Mohamed Ahmed Elgendy	Sec: 2	B. N.: 25

Experiment 2 –Part 1 Results sheet:

1- Analytical solution to discover time invariance and linearity of the systems:

a) $y[n] = x[n] - x[n-1] - x[n-2]$

Linearity:

$x_1[n] \rightarrow y_1[n] = x_1[n] - x_1[n-1] - x_1[n-2]$

$x_2[n] \rightarrow y_2[n] = x_2[n] - x_2[n-1] - x_2[n-2]$

$Ay_1[n] + By_2[n] = A(x_1[n] - x_1[n-1] - x_1[n-2]) + B(x_2[n] - x_2[n-1] - x_2[n-2])$

$x_3[n] = Ax_1[n] + Bx_2[n]$

$y_3[n] = x_3[n] - x_3[n-1] - x_3[n-2]$

$y_3[n] = A(x_1[n] - x_1[n-1] - x_1[n-2]) + B(x_2[n] - x_2[n-1] - x_2[n-2])$

<u>The system is linear (because $y_3[n] = Ay_1[n] + By_2[n]$).</u>

Time Invariance:

$x_1[n] \rightarrow y_1[n] = x_1[n] - x_1[n-1] - x_1[n-2]$

$y_1[n-n_0] = x_1[n-n_0] - x_1[n-n_0-1] - x_1[n-n_0-2]$

$x_2[n] = x_1[n-n_0] \rightarrow y_2[n] = x_2[n] - x_2[n-1] - x_2[n-2]$

$y_2[n] = x_1[n-n_0] - x_1[n-n_0-1] - x_2[n-n_0-2]$

<u>The system is time invariant (because $y_1[n-n_0] = y_2[n]$).</u>

b) $y[n] = \cos(x[n])$

Linearity:

$x_1[n] \rightarrow y_1[n] = \cos(x_1[n])$

$x_2[n] \rightarrow y_2[n] = \cos(x_2[n])$

$Ay_1[n] + By_2[n] = A\cos(x_1[n]) + B\cos(x_2[n])$

$x_3[n] = Ax_1[n] + Bx_2[n]$

$y_3[n] = \cos(x_3[n]) = \cos(Ax_1[n] + Bx_2[n])$

The system is not linear (because $y_3[n] \neq Ay_1[n] + By_2[n]$).

Time Invariance:

$x_1[n] \rightarrow y_1[n] = \cos(x_1[n])$

$y_1[n-n_0] = \cos(x_1[n-n_0])$

$x_2[n] = x_1[n-n_0] \rightarrow y_2[n] = \cos(x_1[n-n_0])$

The system is time invariant system (because $y_1[n-n_0] = y_2[n]$).

c) $y[n] = nx[n]$

Linearity:

$x_1[n] \rightarrow y_1[n] = nx_1[n]$

$x_2[n] \rightarrow y_2[n] = nx_2[n]$

$Ay_1[n] + By_2[n] = Anx_1[n] + Bnx_2[n]$

$x_3[n] = Ax_1[n] + Bx_2[n]$

$y_3[n] = nx_3[n] = n(Ax_1[n] + Bx_2[n])$

<u>The system is linear system (because $y_3[n] = Ay_1[n] + By_2[n]$).</u>

Time Invariance:

$x_1[n] \rightarrow y_1[n] = nx_1[n]$

$y_1[n-n_0] = (n-n_0)x_1[n-n_0]$

$x_2[n] = x_1[n-n_0] \rightarrow y_2[n] = nx_1[n-n_0]$

<u>The system is time variant system (because $y_1[n-n_0] \neq y_2[n]$).</u>

Write the code used to input the first signal to the three systems

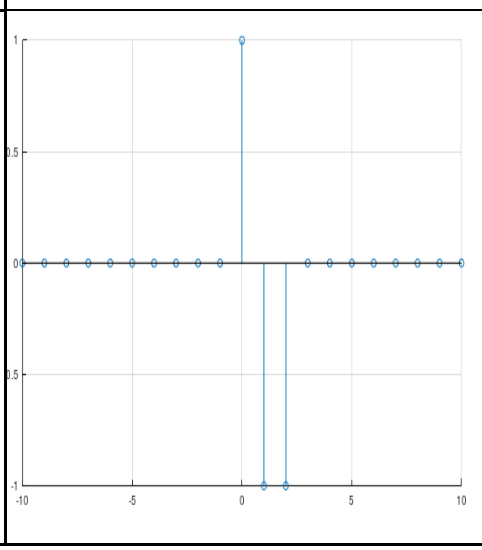
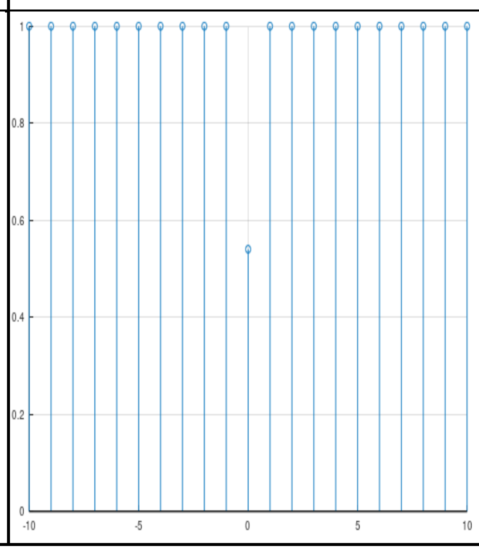
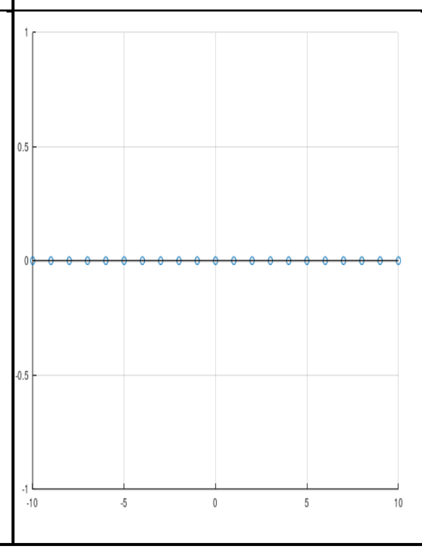
System	Code

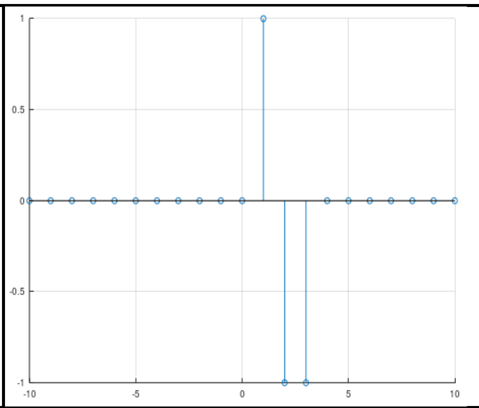
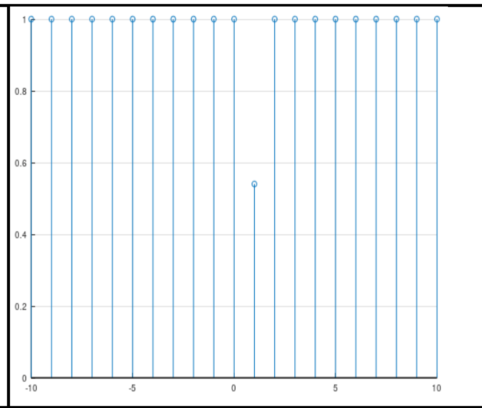
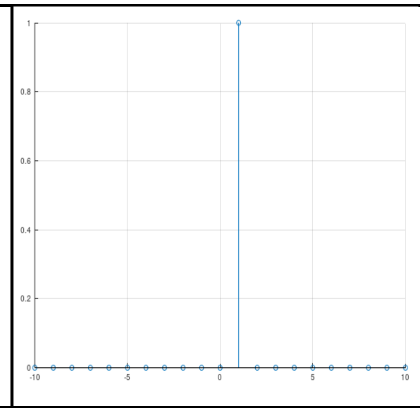
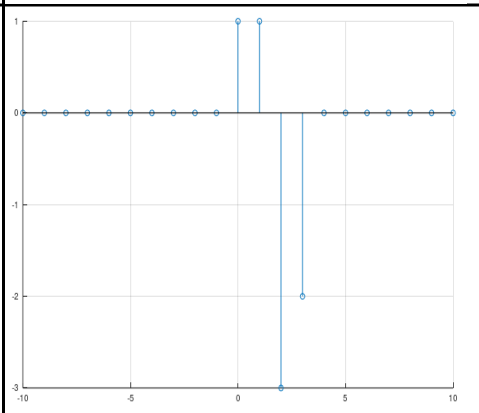
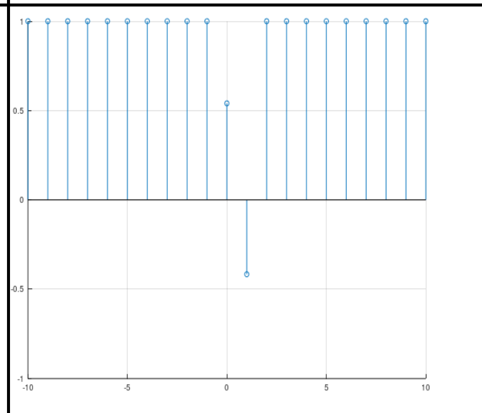
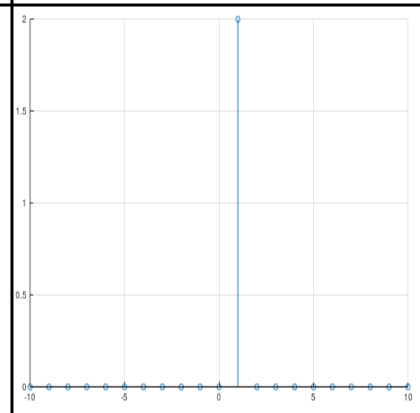
y[n] = x[n] – x[n-1] – x[n-2]	a = 10
	n = [-a:a];
	x = [zeros(1,a) 1 zeros(1,a)]; #delta(n)
	y = x - circshift(x', 1, 1)' - circshift(x', 2, 1)' ;
	stem(n, y);

y[n] = cos(x[n])	a = 10
	n = [-a:a];
	x = [zeros(1,a) 1 zeros(1,a)]; #delta(n)
	y = cos(x);
	stem(n, y);

y[n] = nx[n]	a = 10
	n = [-a:a];
	x = [zeros(1,a) 1 zeros(1,a)]; #delta(n)
	y = n.* x;
	stem(n, y);

Plotting for the responses of the systems to the three inputs:

	System (1)	System (2)	System (3)
Input(1)			

Input(2)			
Input(3)			
Comment	The output of the system in case of input(2) is a shifted version of the output of the system in case of input(1) which means that the system is time invariant system. The output of the system in case of input(3) is the sum of the output of the system in case of input(1) and double the output of the system in case of input(2). output(3) = output(1) + 2*output(2). The system is linear.	The output of the system in case of input(2) is a shifted version of the output of the system in case of input(1) which means that the system is time invariant system. The output of the system in case of input(3) is not equal to the sum of the output of the system in case of input(1) and double the output of the system in case of input(2). output(3) ≠ output(1) + 2*output(2). The system is not linear.	The output of the system in case of input(2) is not a shifted version of the output of the system in case of input(1) which means that the system is time variant system. The output of the system in case of input(3) is the sum of the output of the system in case of input(1) and double the output of the system in case of input(2). output(3) = output(1) + 2*output(2). The system is linear.

Experiment 2- Part 2 Results sheet:

a) Convolution complete code (1)

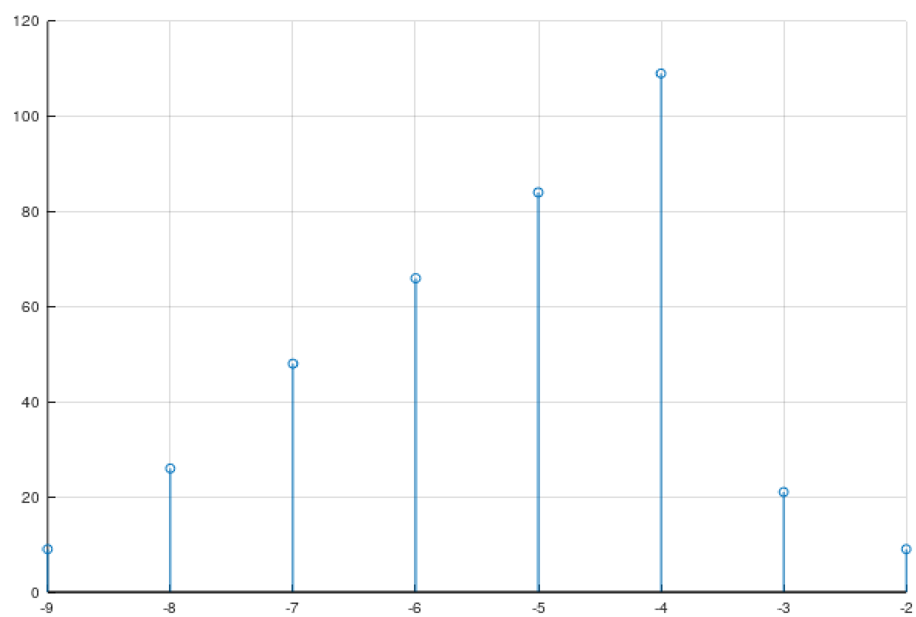
```
nx = [-3 -2 -1];
x = [1 2 3];
nh = [-6 -5 -4 -3 -2 -1];
h = [9 8 5 32 5 3];

M = length(x);
N = length(h);
ny = [(nx(1) + nh(1)) : (nx(M) + nh(N))];
y = zeros(1, M + N - 1);

for u = 1 : N
  x1 = h(u) * [zeros(1, u - 1) x zeros(1, length(y) - (u - 1) - M)];
  y = y + x1;
end

stem(ny, y);
```

Using stem function to plot the final output y.



c) The conv command:

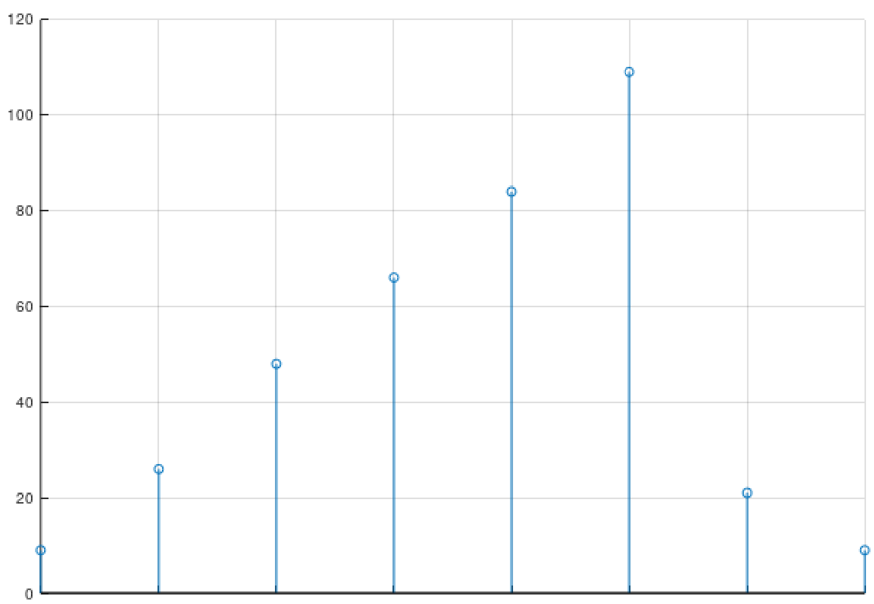
```
nx = [-3 -2 -1];
x = [1 2 3];
nh = [-6 -5 -4 -3 -2 -1];
h = [9 8 5 32 5 3];

M = length(x);
N = length(h);
ny = [(nx(1) + nh(1)) : (nx(M) + nh(N))];
y = zeros(1, M + N - 1);

y = conv(h, x);

stem(ny, y);
```

Using stem function to plot its final output y.



Experiment 2 Results sheet:

<table>
<tr><td>a)Inverse Fourier Series Code</td><td>

```
#inverse fourier series
function x = f_series_inverse(a)
  N = length(a);
  k = 0 : N - 1;
  for n = 0: N - 1
    x(n + 1) = sum(a .* exp(2 * pi * i * k * n / N));
  end
end
################
```

</td></tr>
<tr><td rowspan="3">b)Fourier series of the three signals:</td><td>ak of the signal x:
2.50000 + 0.00000i -0.50000 + 0.50000i -0.50000 - 0.00000i -0.50000 - 0.50000i</td></tr>
<tr><td>ak of the signal x:
1.50000 + 0.00000i -0.25000 - 0.25000i 0.00000 - 0.00000i -0.25000 + 0.25000i</td></tr>
<tr><td>ak of the signal x:
0.00000 + 0.00000i 0.00000 - 0.85065i -0.00000 + 0.52573i 0.00000 - 0.52573i
-0.00000 + 0.85065i</td></tr>
<tr><td rowspan="3">X signal for the previous Fourier series coefficients</td><td>The signal x:
1.0000 - 0.0000i 2.0000 - 0.0000i 3.0000 - 0.0000i 4.0000 + 0.0000i</td></tr>
<tr><td>The signal x:
1.0000 - 0.0000i 2.0000 - 0.0000i 2.0000 + 0.0000i 1.0000 + 0.0000i</td></tr>
<tr><td>The signal x:
-2.1094e-16 + 1.1102e-16i , 1.0000e+00 + 1.6653e-16i , 2.0000e+00 + 2.2204e-16i
-2.0000e+00 + 1.1102e-16i , -1.0000e+00 + 5.5511e-17i

Note that the output of the program in this case is not very accurate (it can be approximated to the exact input signal x by considering that e-16 is nearly equal to zero) due to floating point errors.</td></tr>
</table>

c) analytical solution of the three signals:	x[n] = cos(2 * pi * n * 3 / 7) w_0 = 2 * pi / 7 x[n] = cos(3w_0 * n) x[n] = 0.5*exp(3jw_0n) + 0.5*exp(-3jw_0n) ak = 0.5 k = 3 ak = 0.5 k = -3 ak = 0 otherwise
	x[n] = sin(2 * pi * n * 3 / 7) w_0 = 2 * pi / 7 x[n] = sin(3w_0 * n) x[n] = -0.5j*exp(3jw_0n) + 0.5j*exp(-3jw_0n) ak = -0.5j k = 3 ak = 0.5j k = -3 ak = 0 otherwise
	x[n] = exp(j * 2 * pi * n * 3 / 7) w_0 = 2 * pi / 7 x[n] = exp(3jw_0) ak = 1 k = 3 ak = 0 otherwise

Simulation output of the three signals:	n = [0:6];
	x = cos(2 * pi * n * 3 / 7);
	disp(f_series(x));
	
	program output:
	0.00000 + 0.00000i , -0.00000 + 0.00000i ,
	-0.00000 - 0.00000i , 0.50000 - 0.00000i , 0.50000 + 0.00000i
	-0.00000 + 0.00000i , 0.00000 - 0.00000i

	n = [0:6];
	x = sin(2 * pi * n * 3 / 7);
	disp(f_series(x));
	
	program output:
	-0.00000 + 0.00000i , 0.00000 - 0.00000i , 0.00000 + 0.00000i
	0.00000 - 0.50000i , -0.00000 + 0.50000i ,
	0.00000 + 0.00000i , 0.00000 + 0.00000i

	n = [0:6];
	x = exp(i * 2 * pi * n * 3 / 7);
	disp(f_series(x));
	
	program output:
	0.00000 - 0.00000i , 0.00000 + 0.00000i , -0.00000 + 0.00000i
	1.00000 + 0.00000i , 0.00000 + 0.00000i , -0.00000 + 0.00000i
	0.00000 + 0.00000i