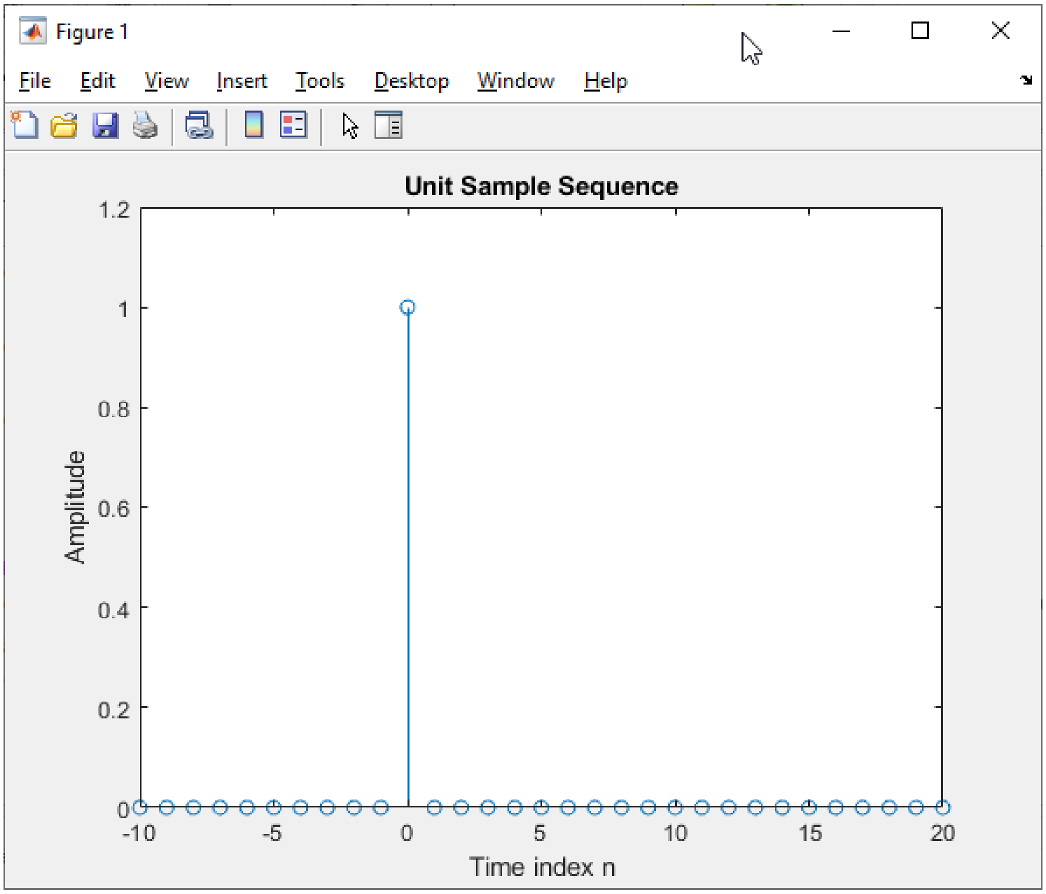
MDISCRETE-TIME SIGNALS: TIME-DOMAIN REPRESENTATION

GENERATION OF SEQUENCES

1.1 UNIT SAMPLE AND UNIT STEP SEQUENCES

Q1.1



Q1.2

```
% Program P1_1
% Generation of a Unit Sample Sequence
clf;
% Generate a vector from -10 to 20
n = -10:1:20;           % <start> : <step> : <stop>
% Generate the unit sample sequence
u = [zeros(1,10) 1 zeros(1,20)];
% Plot the unit sample sequence
stem(n,u);
% Plot the unit sample sequence
% xlabel('text') adds text beside the X-axis on the current axis.
xlabel('Time index n');
% ylabel('text') adds text beside the y-axis on the current axis.
ylabel('Amplitude');
```

xoá các figure trước đó

step=1

concat

10x[0] [1] [0]x20

vẽ các điểm rời rạc

zeros(___ , ___)
rows cols

ones(___ , ___)
rows cols

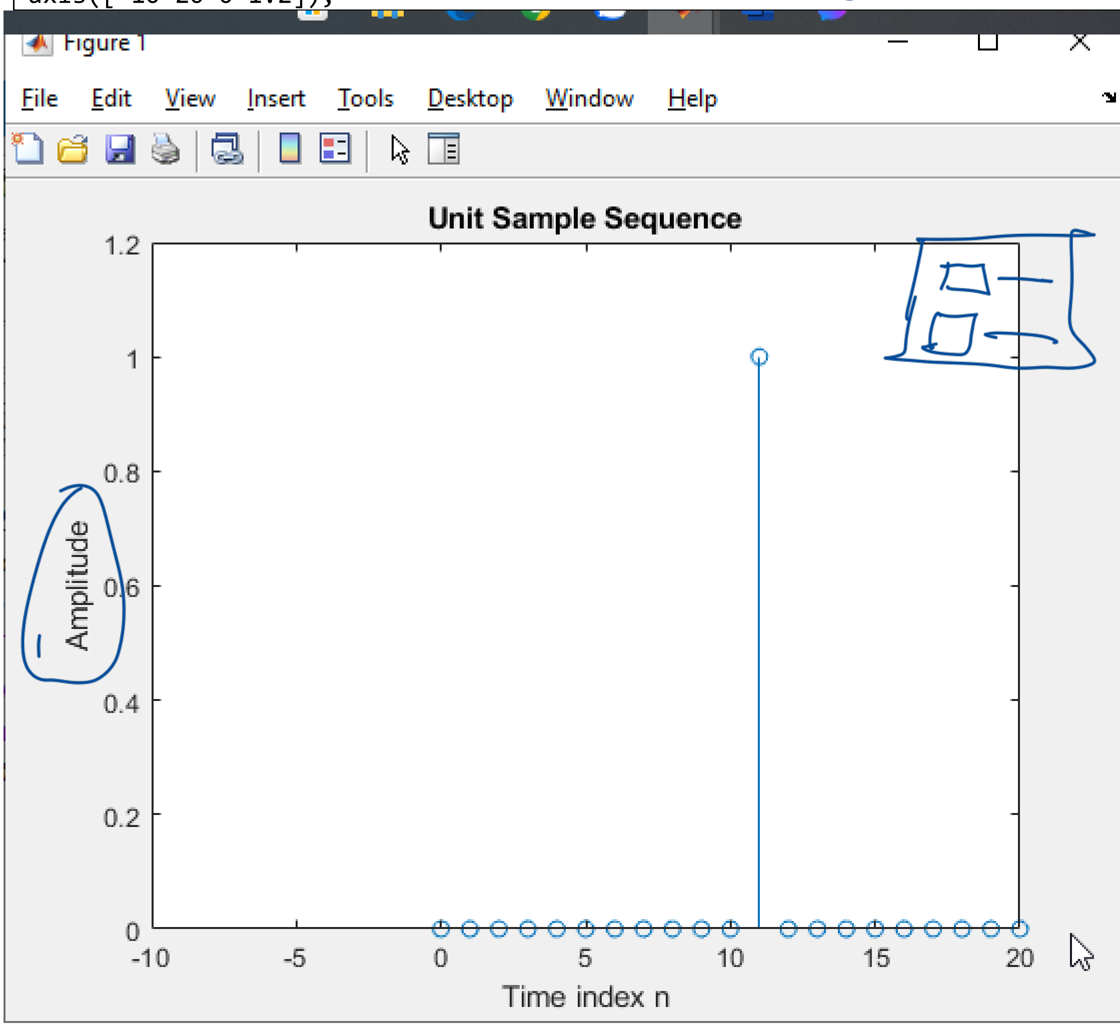
```
%    title('txt') adds the specified title to the axes or chart returned by
%    the gca command. Reissuing the title command causes the new title to
%    replace the old title.
title('Unit Sample Sequence');
% Control axis scaling and appearance.
axis([-10 20 0 1.2]);
```

range limit of x-axis → y

Q1.3

```
clf;
n = 0:1:20;
u = [zeros(1,11) 1 zeros(1,10)];
stem(n,u);
xlabel('Time index n');
ylabel('Amplitude');
title('Unit Sample Sequence');
axis([-10 20 0 1.2]);
```

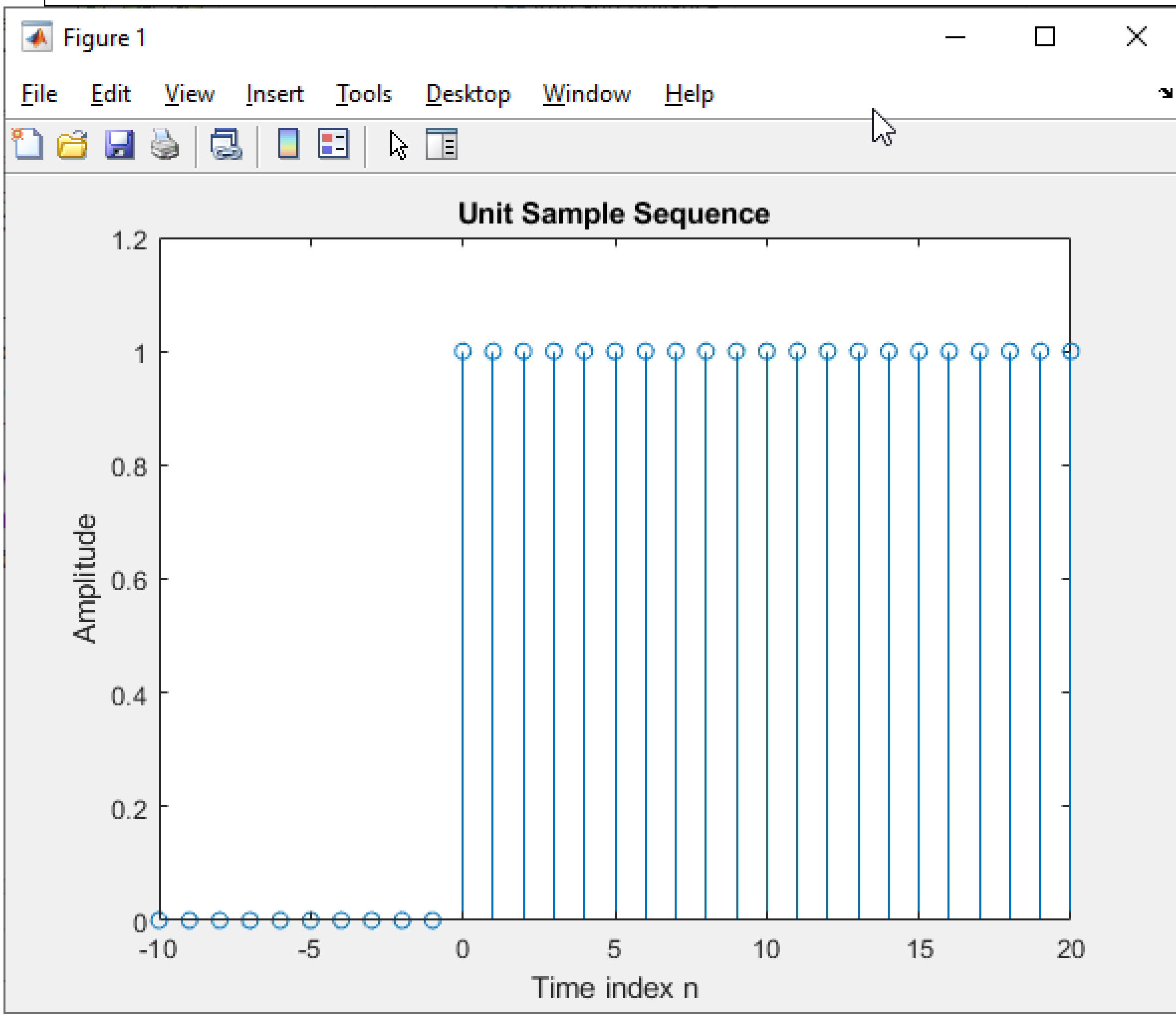
legend (   )



Q1.4

```
clf;
n = -10:1:20;
u = [zeros(1,10) 1 ones(1,20)];
```

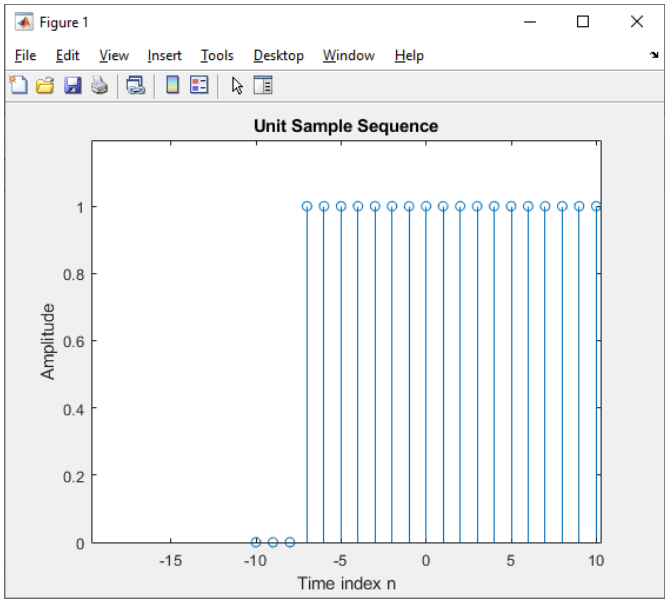
```
stem(n,u);
xlabel('Time index n');
ylabel('Amplitude');
title('Unit Sample Sequence');
axis([-10 20 0 1.2]);
```

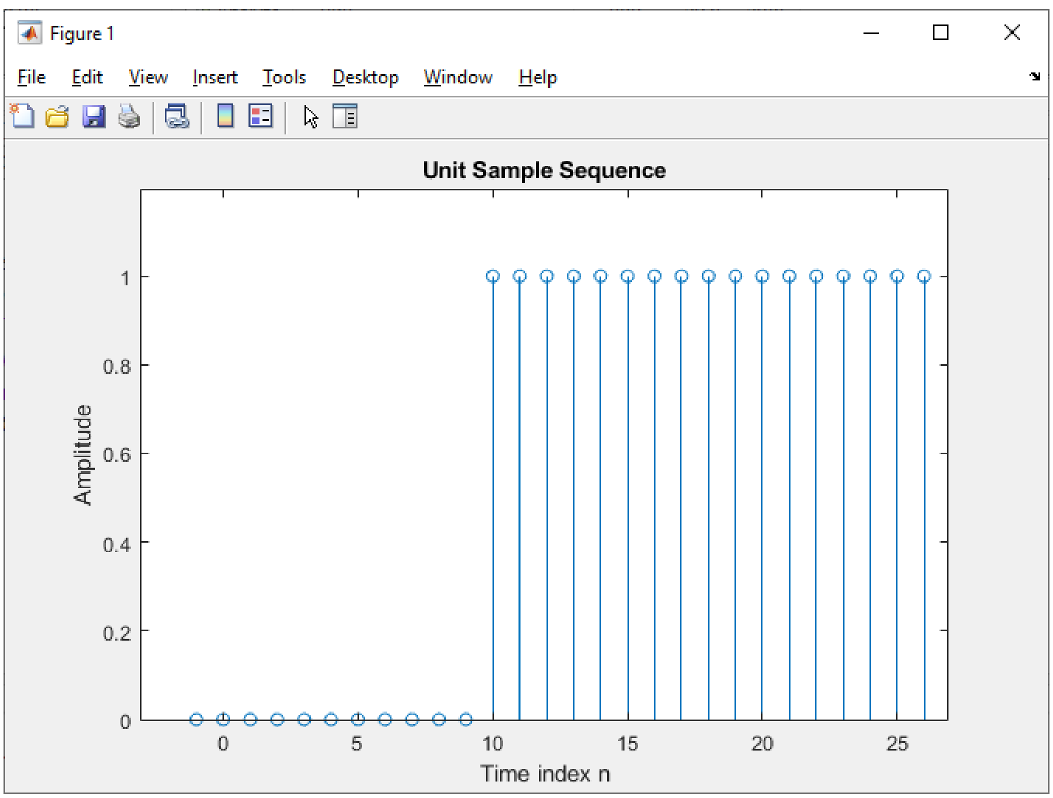


Q1.5

The modified Program P1_1 to generate a unit step sequence sd[n] with an advance of 7 samples is given below along with the sequence generated by running this program.

```
clf;
n = -10:1:10;
u = [zeros(1,3) ones(1,7) 1 ones(1,10)];
stem(n,u);
xlabel('Time index n');
ylabel('Amplitude');
title('Unit Sample Sequence');
axis([-10 20 0 1.2]);
```

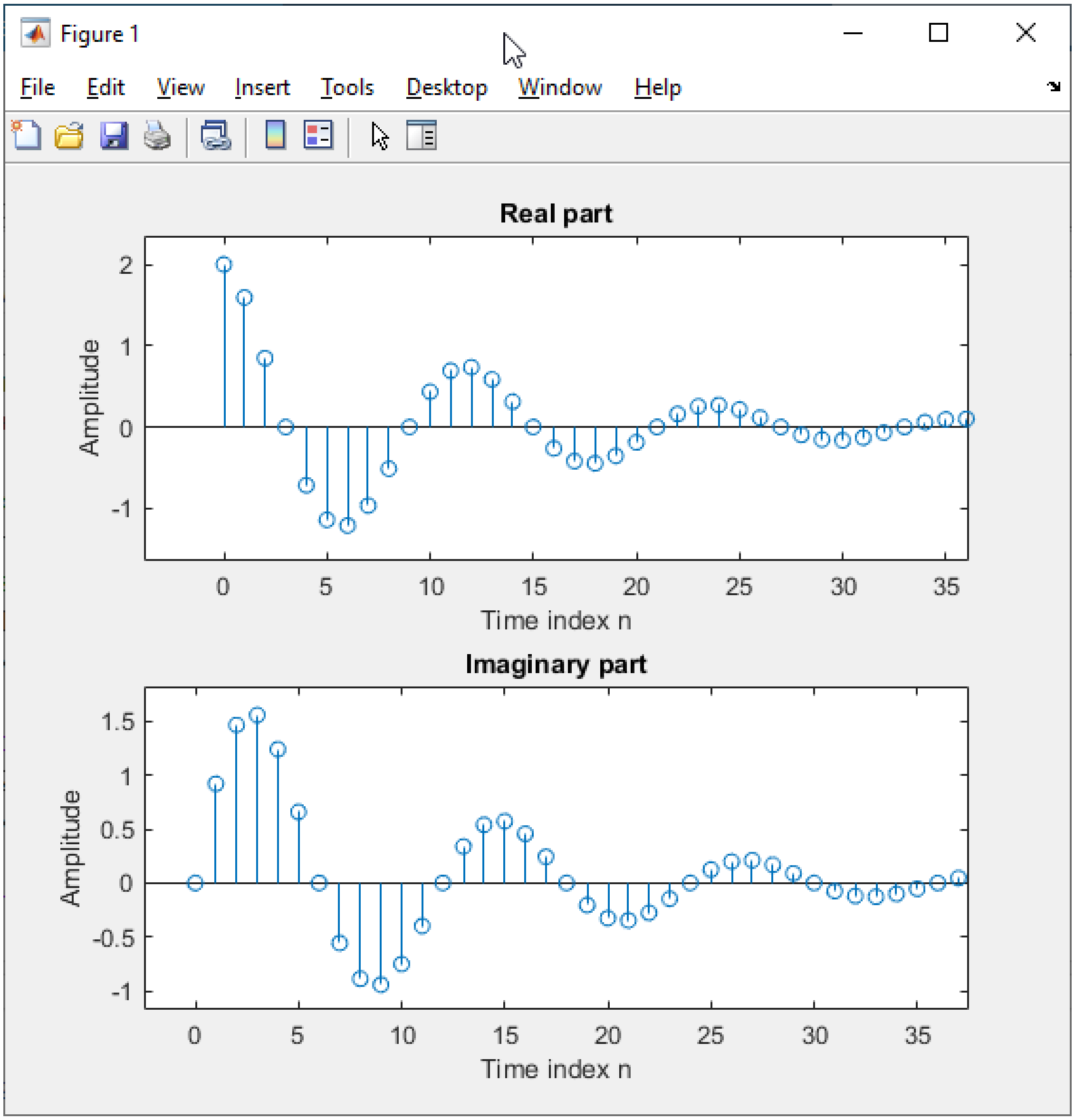




1.2 EXPONENTIAL SIGNALS

Q1.6

The complex-valued exponential sequence generated by running Program P1_2 is shown below:

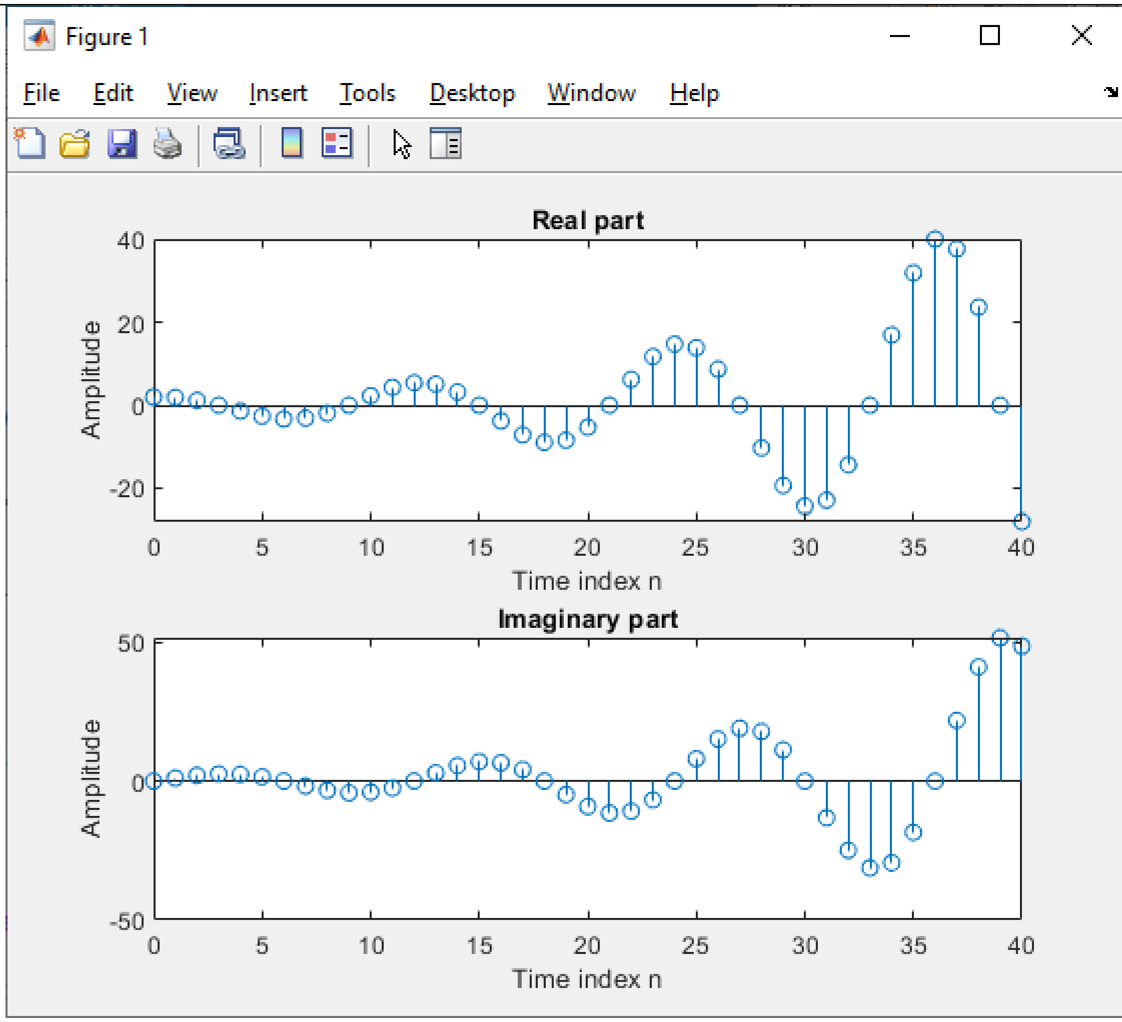


Q1.7

The parameter controlling the rate of growth or decay of this sequence is: c The parameter controlling the amplitude of this sequence is: K

Q1.8

The result of changing the parameter c to (1/12)+(pi/6)*i is:



The real part has changed from decay function to growth function since exp(1/12) has changed to exp(-1/12).

Q1.9

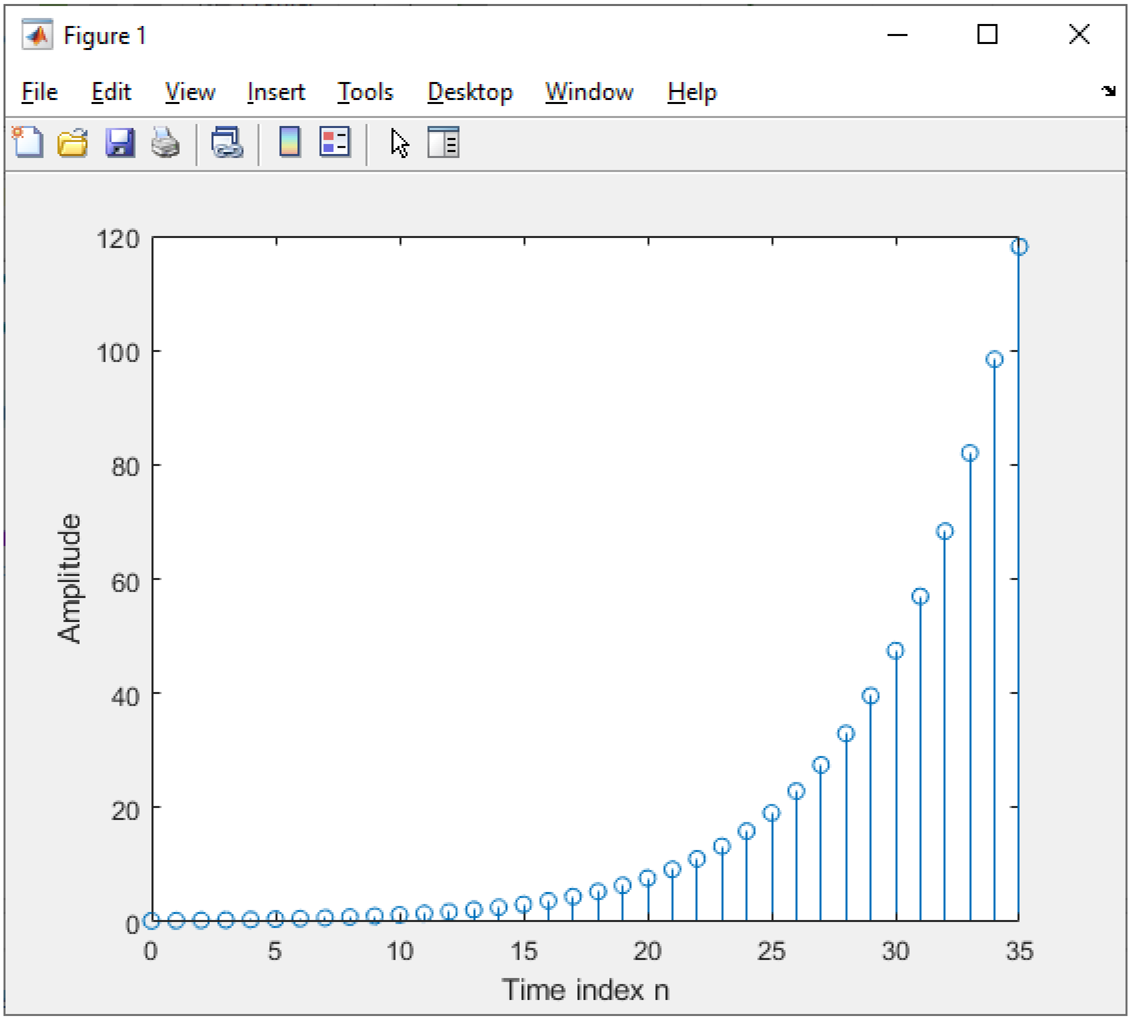
The purpose of the operator real is – to extract the real part of a Matlab vector.
The purpose of the operator imag is – to extract the imaginary part of a Matlab vector.

Q1.10

The purpose of the command subplot is – to plot more than one graph in the same Matlab figure.

Q1.11

The real-valued exponential sequence generated by running Program P1_3 is shown below:



Q1.12

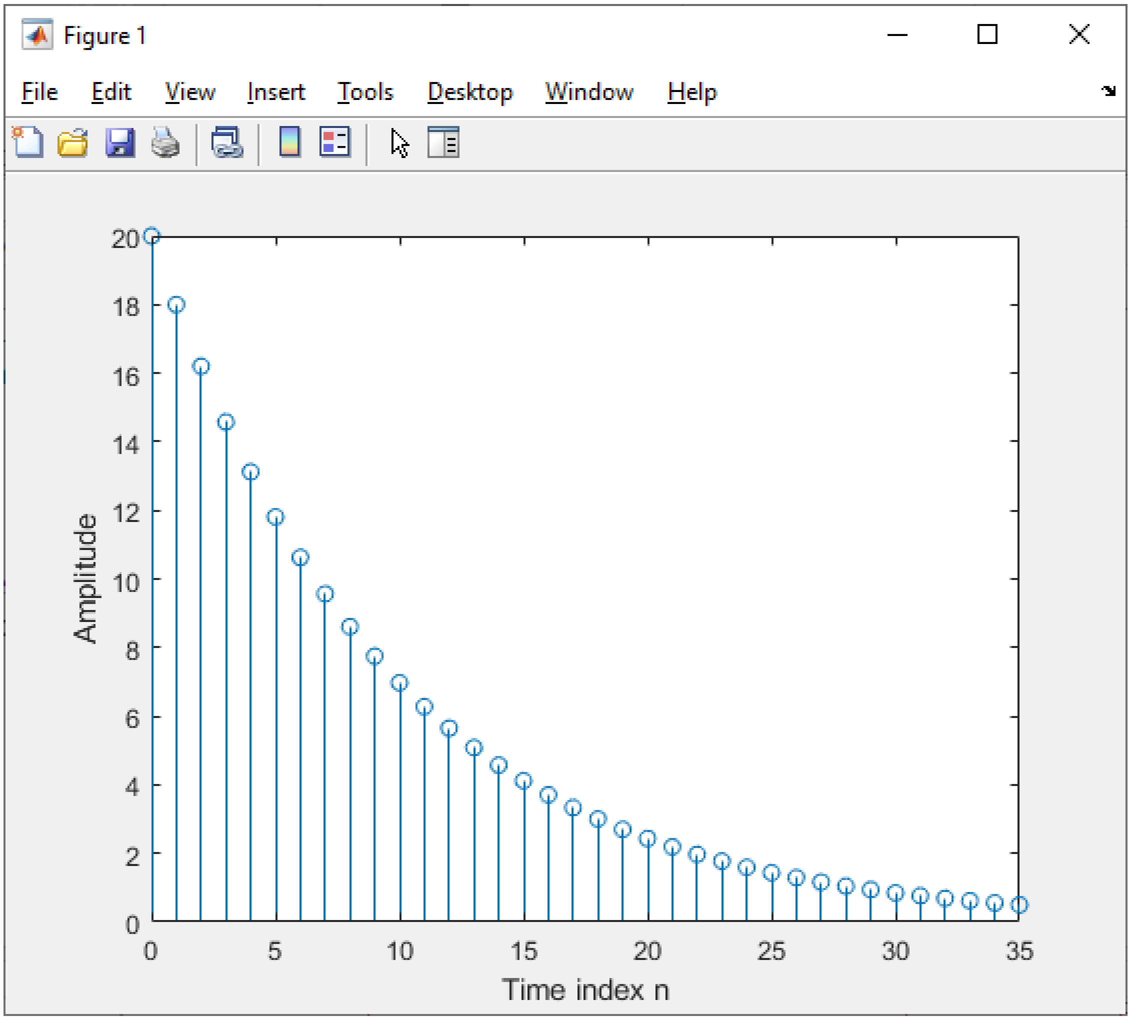
The parameter controlling the rate of growth or decay of this sequence is - a The parameter controlling the amplitude of this sequence is - K

Q1.13

The difference between the arithmetic operators ^ and .^ is – "^" raises a square matrix to a power using matrix multiplication. ".^" raises the elements of a matrix or vector to a power; this is a "pointwise" operation.

Q1.14

The sequence generated by running Program P1_3 with the parameter a changed to 0.9 and the parameter K changed to 20 is shown below:



Q1.15

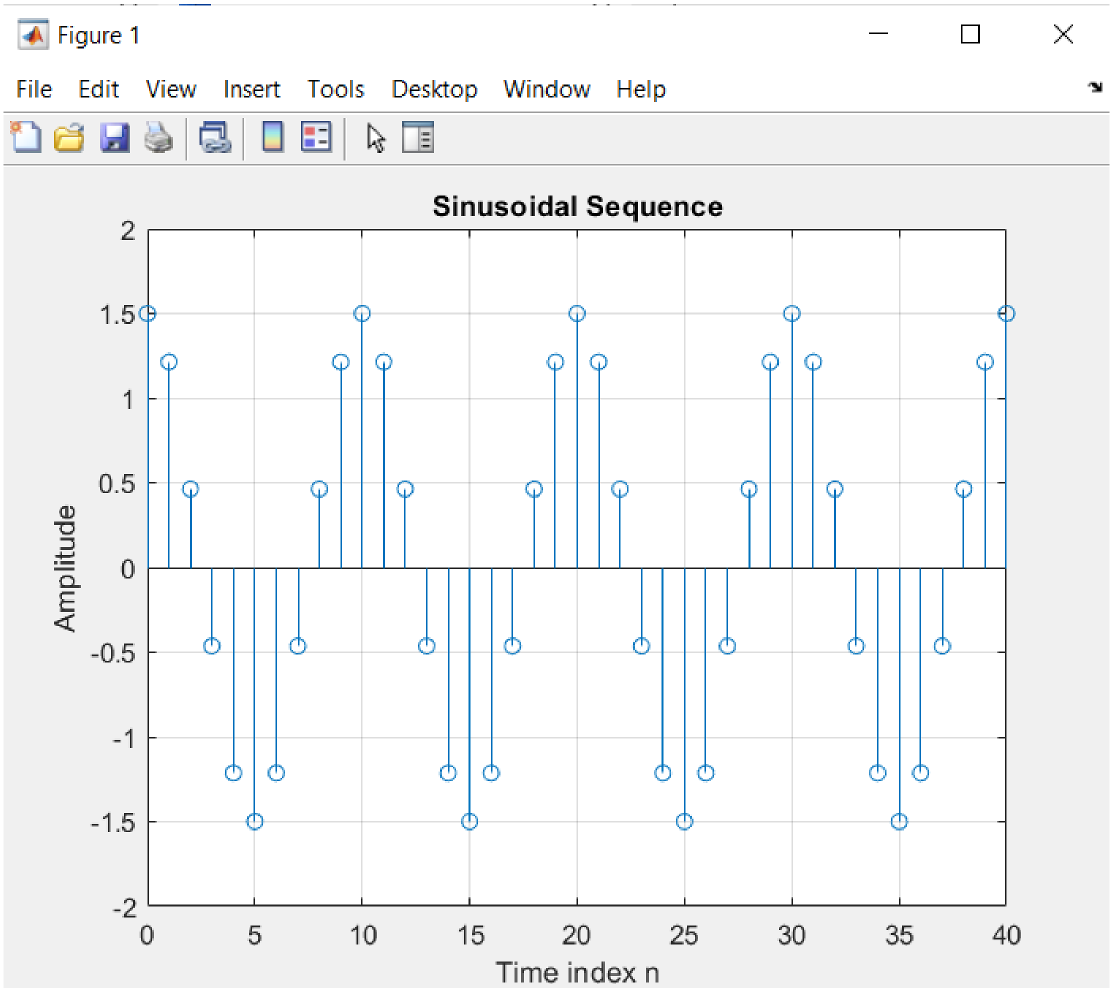
The length of this sequence is: 36 elemets – since n has 36 items. It is controlled by the following MATLAB command line: n = 0:35; 30 35 It can be changed to generate sequences with different lengths as follows (give an example command line and the corresponding length): n = 0:99; makes the length 100.

Q1.16

The energies of the real-valued exponential sequences x[n]generated in Q1.11 and Q1.14 and computed using the command sum are - 4.5673e+004 and 2.1042e+003.

1.3 SINUSOIDAL SEQUENCES

Q1.17



Q1.18

The frequency of this sequence is f = 0.1
It is controlled by following MATLAB command line : f = 0.1
A sequence with new length 0.2 can be generated by the following command line: f = 0.2
The parameter controlling the phase of this sequence is: phase
The parameter controlling the amplitude of this sequence is: A
The period of this sequence is 1/f = 10

Q1.19

The length of this sequence is 41 elemets
It is controlled by following MATLAB command line : n = 0:40
A sequence with new length 20 can be generated by the following command line: n = 0:19

Q1.20

The average power of the generated sinusoidal sequence is 1.152439024390244 (unit of power)

Q1.21

The purpose of *axis* command is to set the range of the x-axis to [0,40] and the range of the y-axis to [-2,2]. The purpose of *grid* command is to turn on the drawing of grid lines on the graph.

Q1.22

```
% Program P1_4
% Generation of a sinusoidal sequence
n = 0:40;
f = 0.9;
phase = 0;
A = 1.5;
arg = 2*pi*f*n - phase;
x = A*cos(arg);
clf;                % Clear old graph
stem(n,x);          % Plot the generated sequence
axis([0 40 -2 2]);
grid;
title('Sinusoidal Sequence');
xlabel('Time index n');
ylabel('Amplitude');
axis;

%n = 0:9;
y = x.^2;
P = (1/length(y))*sum(y);
```

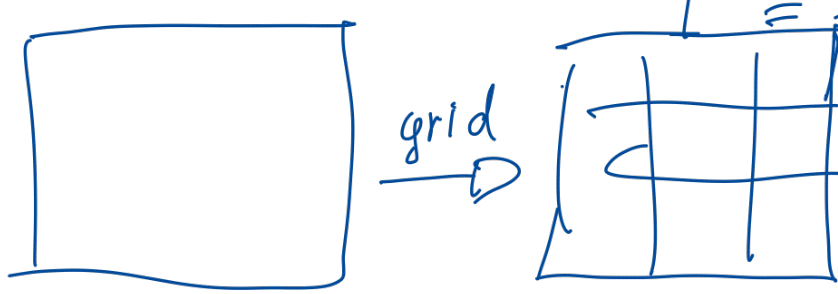
Scalar

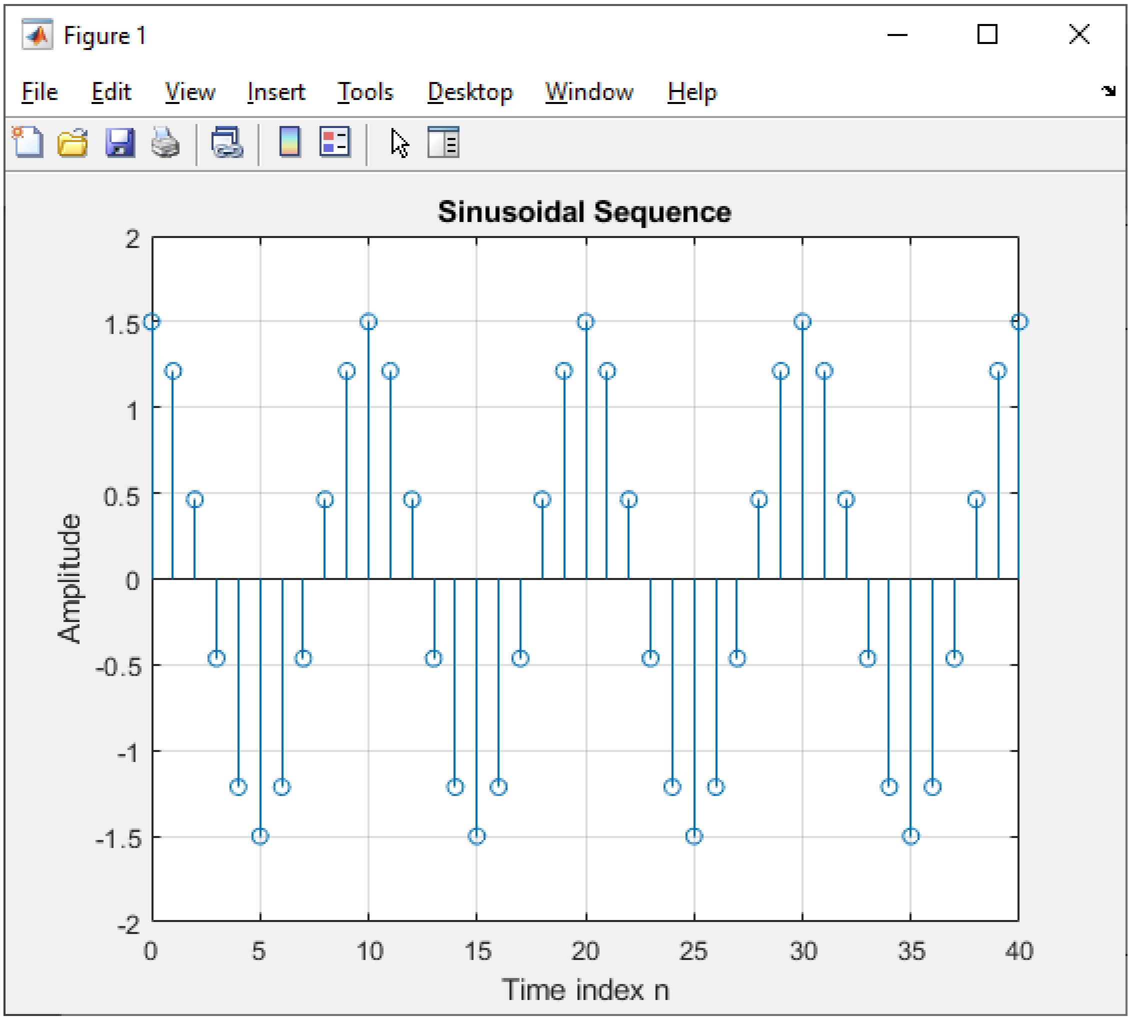
2*pi*f*n - phase

vector

$$P = \frac{1}{T}\int_0^T |x(t)|^2 dt$$

$$P = \frac{1}{N}\sum_{i=0}^{N-1} |x(n)|^2$$





arg

A comparison of this new sequence with the one generated in Question Q1.17 shows: the two graphs are identical. Since the arg in both case (f=0,1 and f=0,9) is opposite (cos(x) = cos(-x)).

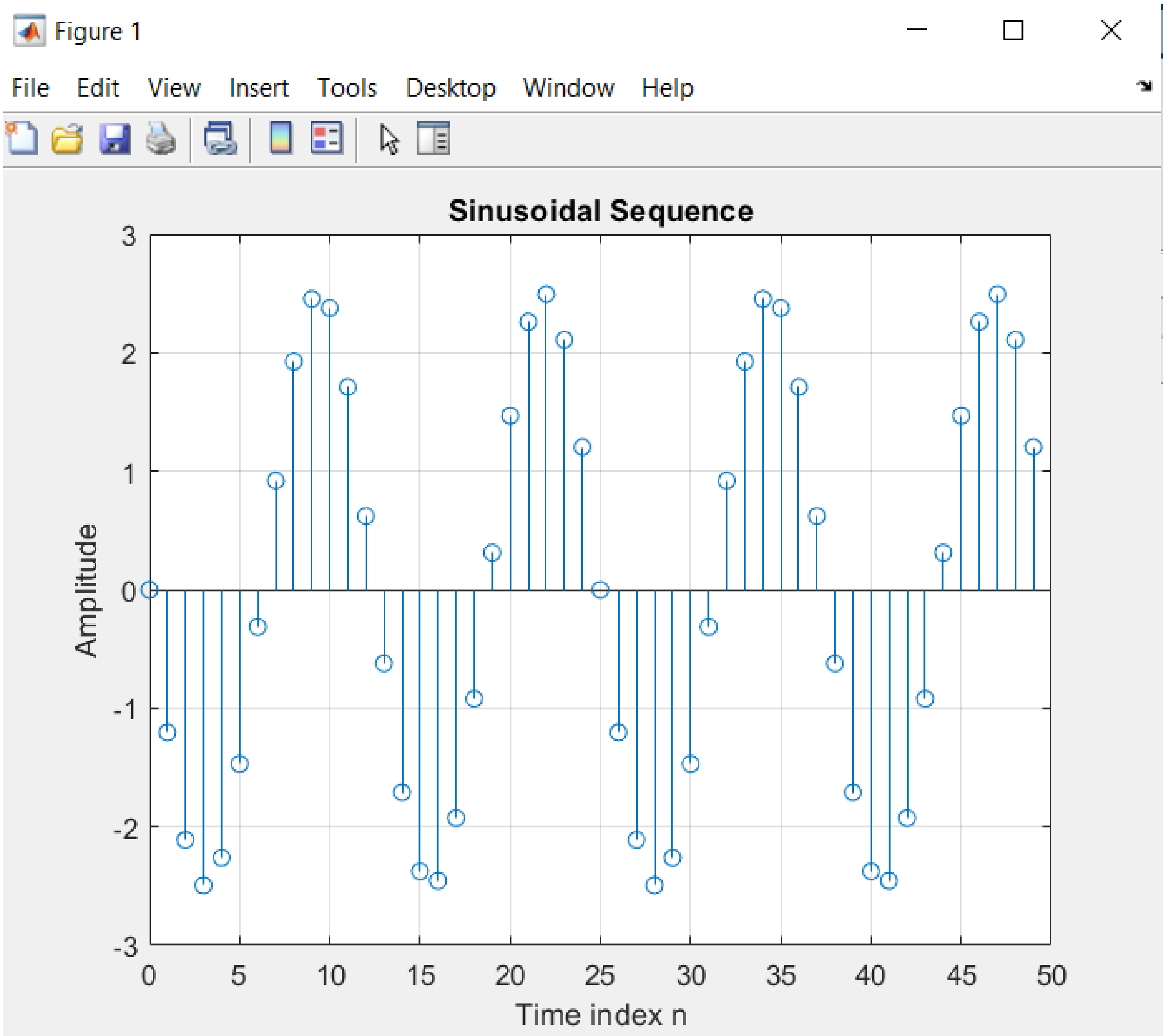
bf $\cos(2\pi \cdot 0,1 \cdot n)$

af $\cos(2\pi \cdot 0,9 \cdot n)$

$\Leftrightarrow \cos(-2\pi \cdot n + 2\pi \cdot 0,9 \cdot n) \Leftrightarrow \cos(2\pi n \underbrace{(-1 + 0,9)}_{-0,1})$

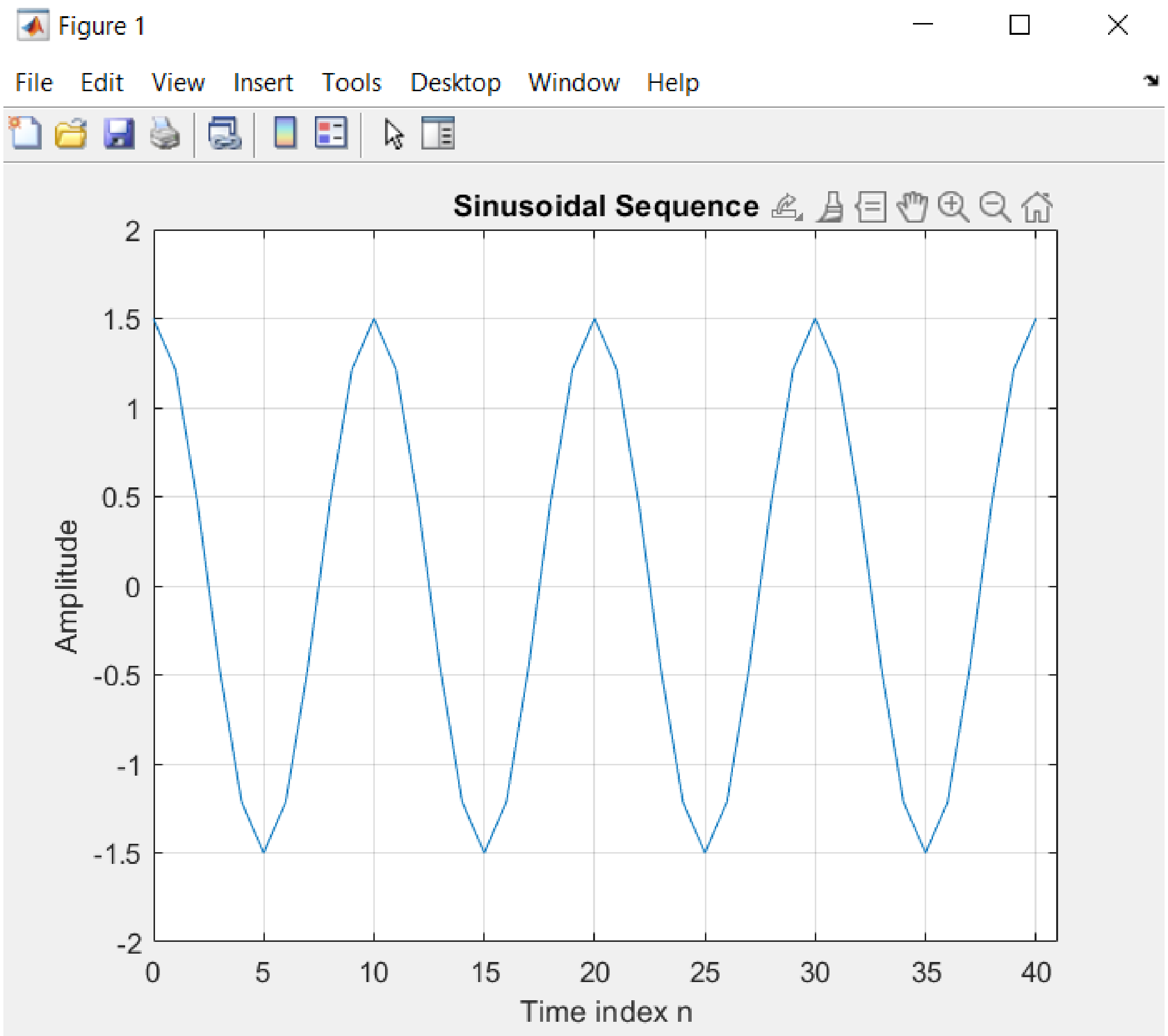
$\cos(2\pi n\, 0,1)$

Q1.23



The period of this sequence is 1/f = 12,5

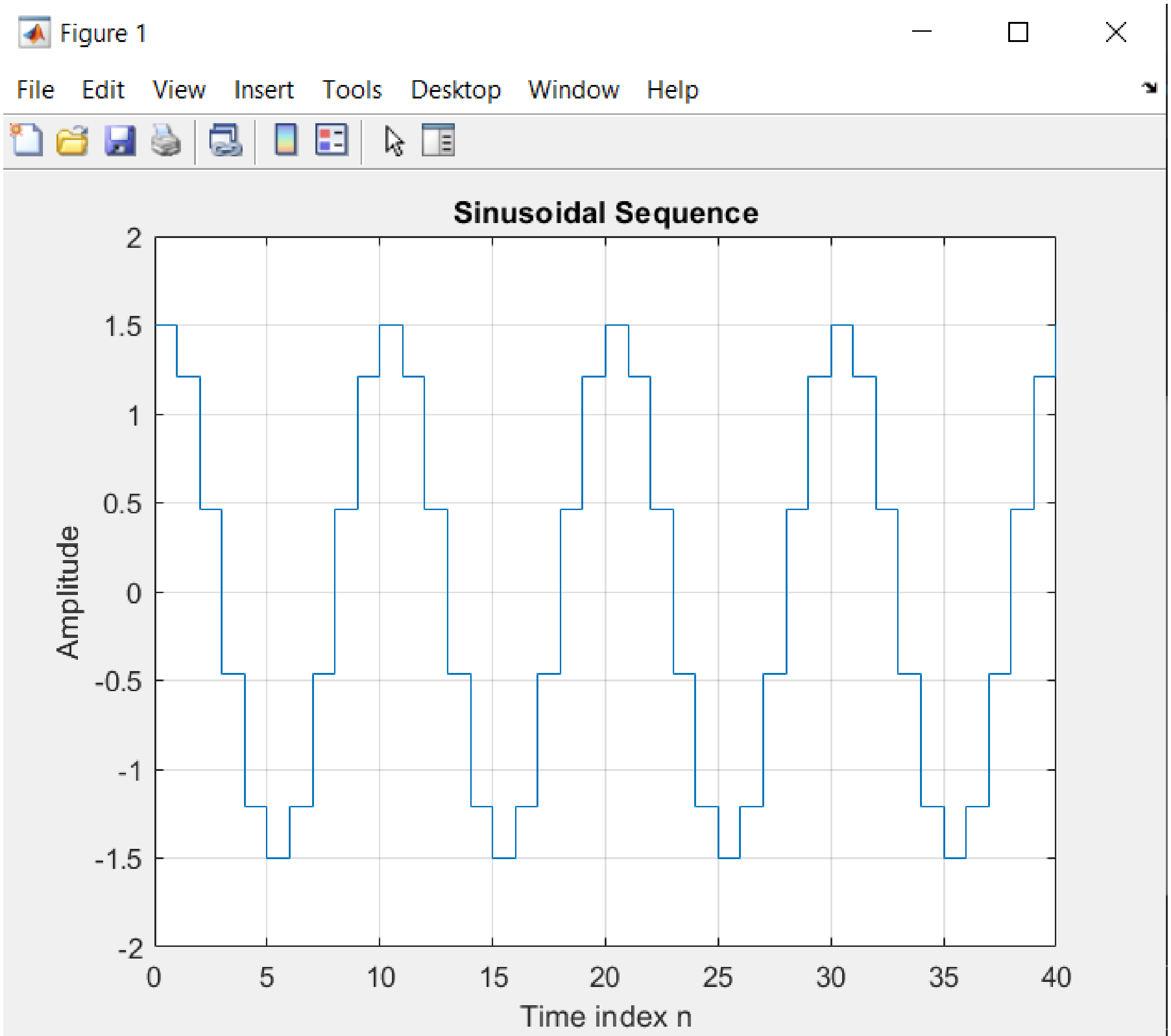
Q1.24



The difference between the new plot and those generated in Question Q1.17 and Q1.24 is discrete to continuous

Q1.25

By replacing the stem command in Program P1_4 with the *stairs* command the plot obtained is as shown below:



The difference between the new plot and those generated in Question Q1.17 and Q1.24 is from discrete to continuous

1.4 RANDOM SIGNALS

Q1.26

The MATLAB program to generate and display a random signal of length 100 with elements uniformly distributed in the interval [-2,2] is given below along with the plot of the random sequence generated by running the program:

```
% Xoá các figure cũ đã vẽ (clear figure)
clf;
% Xoá các lệnh cũ (clear command) trong cửa sổ lệnh
clc;
% Xoá các biến đã tạo
clear;
% Time index n
```

clear commad

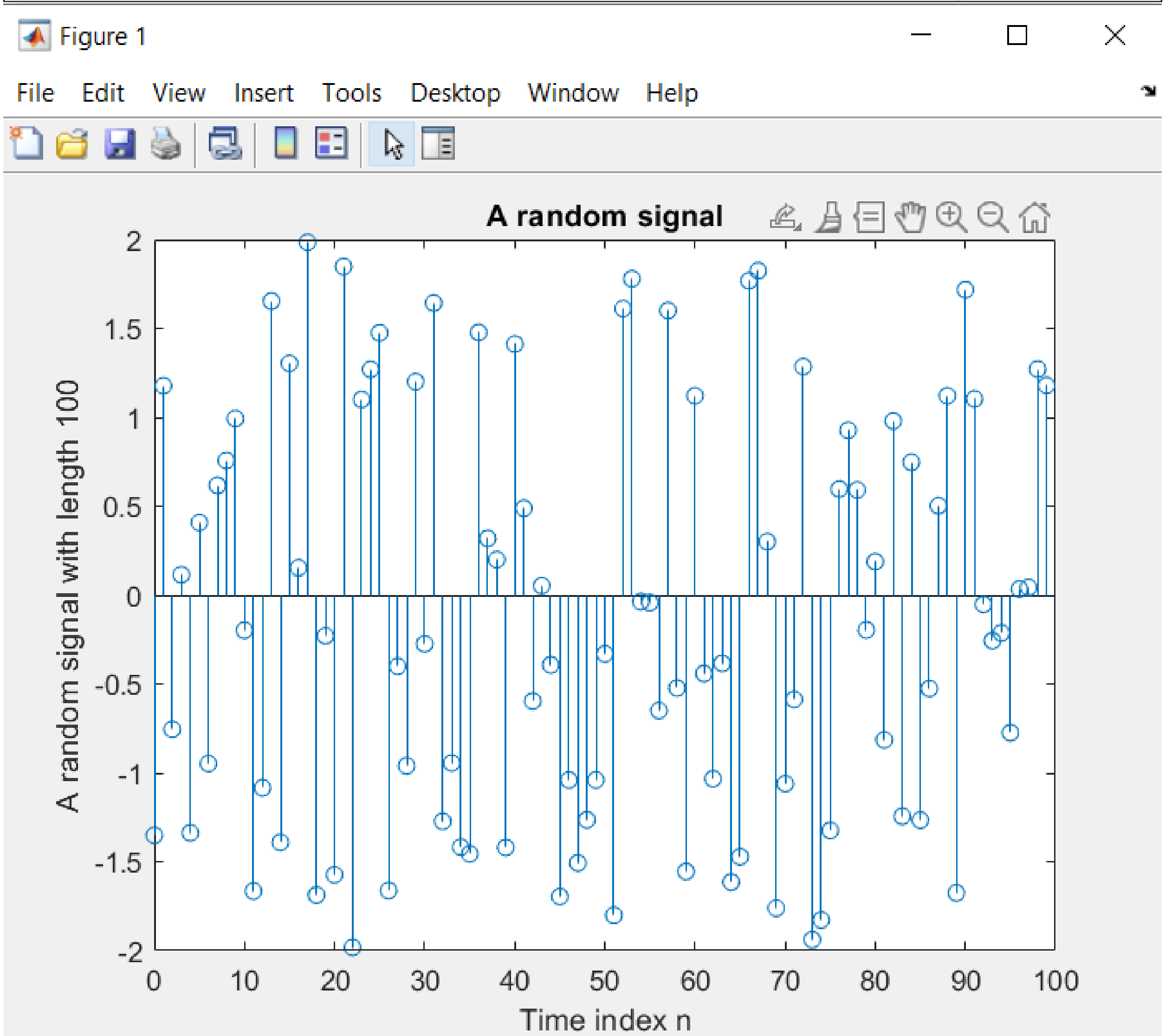
```
n = 0:99;
% A random signal of length 100 with elements uniformly distributed in the interval
[-2,2]
x = rand(1,100)*4-2;
% Vẽ các điểm rời rạc
stem(n, x);
title("A random signal");
xlabel("Time index n");
ylabel("A random signal with length 100");
```

rand(,)

rows cols

1

0

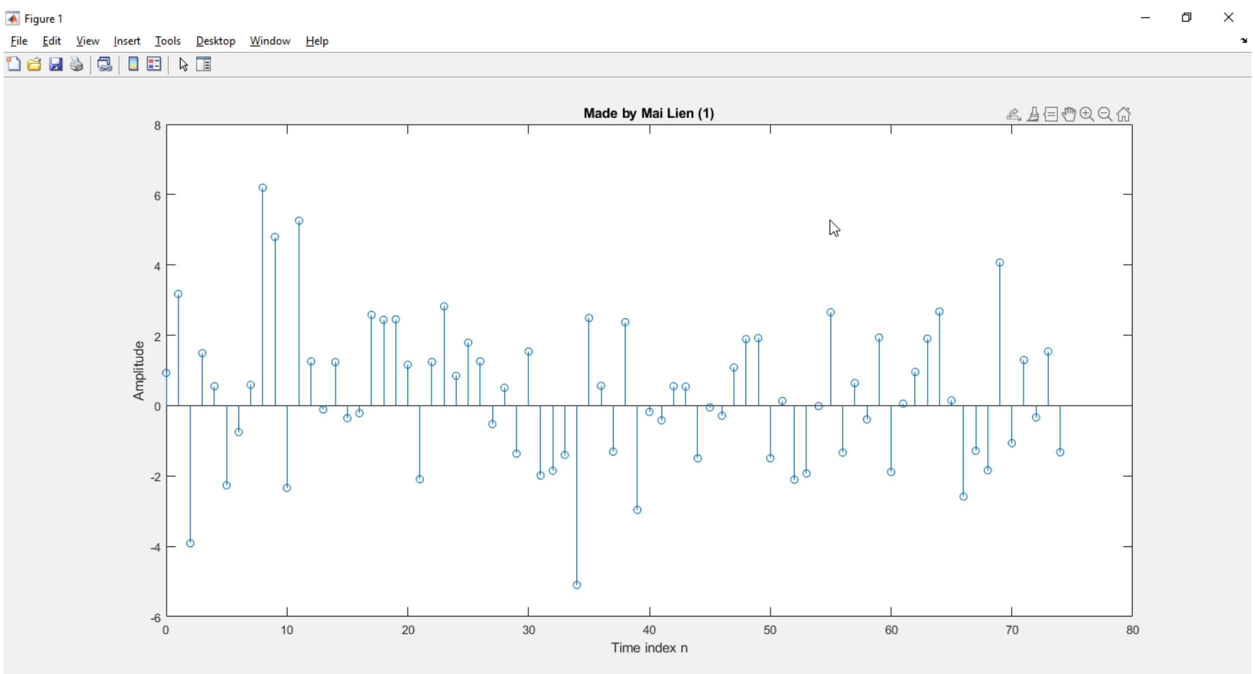


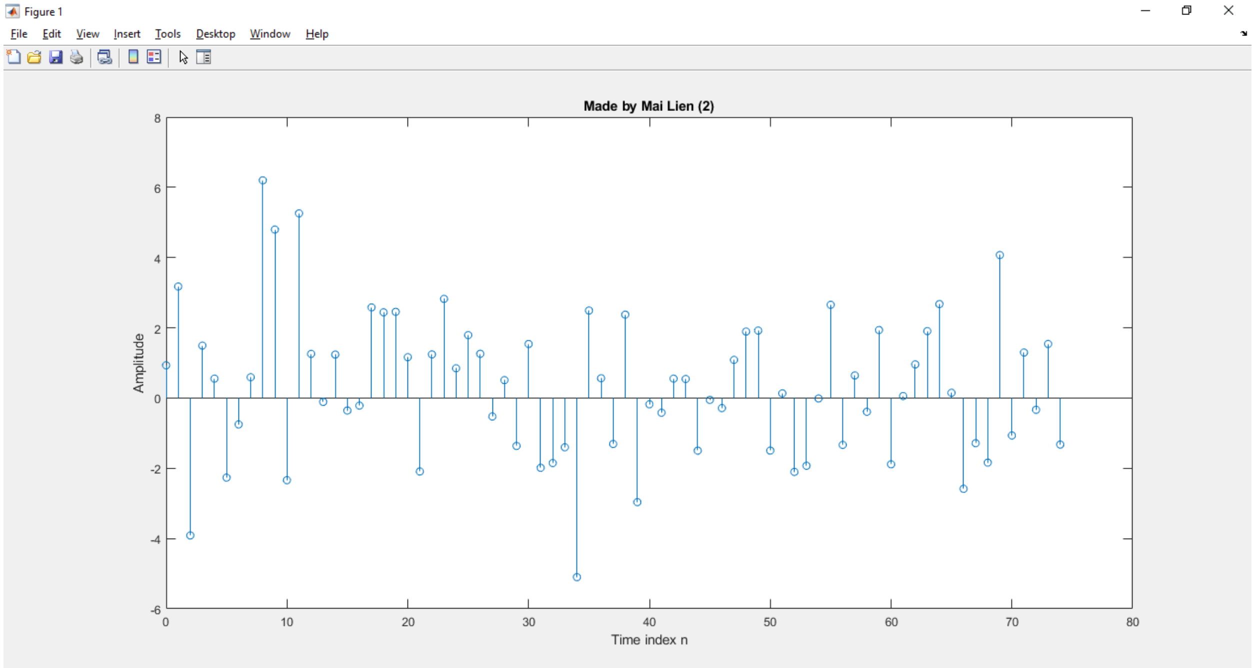
Q1.27

The MATLAB program to generate and display a Gaussian radom signal of length 75 with elements normally distributed with zero mean and variance of 3 is given below along with the plot of the random sequence generated by running the program:

n = 0:74;
% For reproducibility
rng('default');
	n = 0:74;
	% For reproducibility
	rng('default');

`mu = 0;`	`x = randn(1, 75).*sqrt(3) + 0;`
`sigma = sqrt(3);`	`stem(n, x);`
`x = random('Normal',mu, sigma, 1, 75);`	`title("Made by Mai Lien (2)");`
`stem(n, x);`	`xlabel("Time index n");`
`title("Made by Mai Lien (1)");`	`ylabel("Amplitude");`
`xlabel("Time index n");`	`% axis([XMIN XMAX YMIN YMAX]) sets scaling for the x- and y-axes`
`ylabel("Amplitude");`	
`XMIN = 0;`	`XMIN = 0;`
`XMAX = 74;`	`XMAX = 74;`
`YMIN = -6;`	`YMIN = -6;`
`YMAX = 6;`	`YMAX = 6;`
`xlim([XMIN XMAX]);`	`xlim([XMIN XMAX]);`
`ylim([YMIN YMAX]);`	`ylim([YMIN YMAX]);`





Q1.28

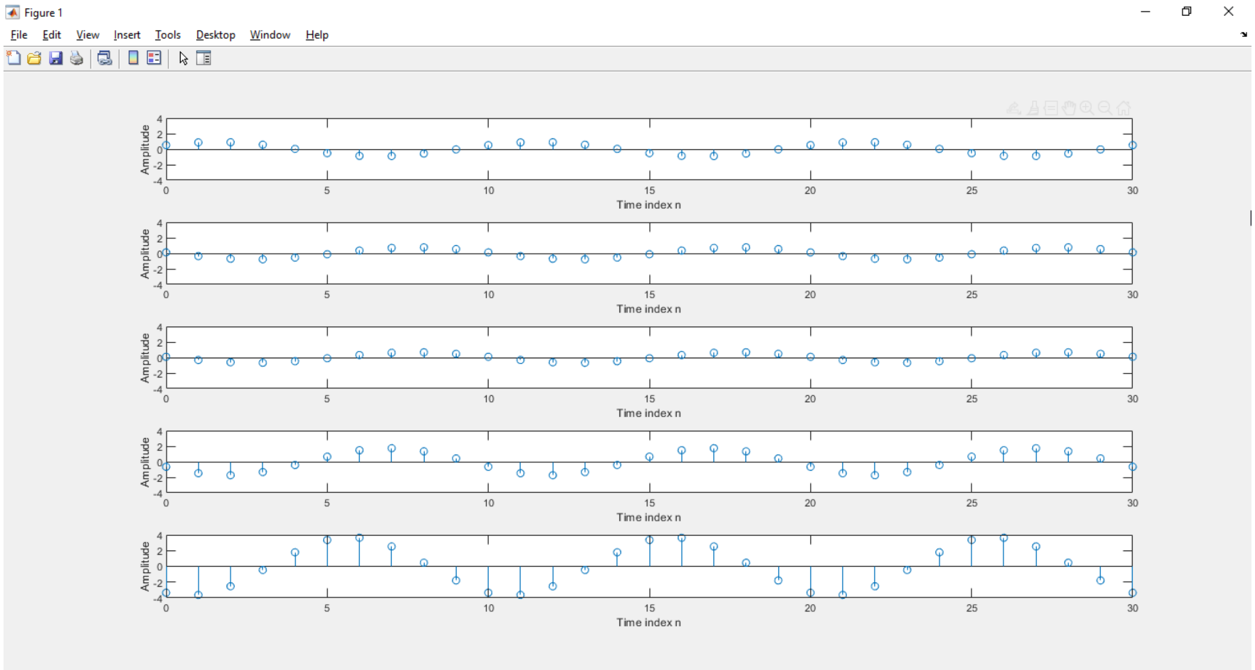
subplot (m, n, p)

rows cols p-th

subplot (5, 1, 1)

r c 1st

```
clf;clc;clear;
n = 0:30;

for var = 1:5
    subplot(5, 1, var);
    A = rand*4;
    phi = rand*2*pi;
    freq = 0.1;
    omegaZero = 2*pi*freq;
    arg = omegaZero * n + phi;
    x = A * cos(arg);
    stem(n, x)
    axis([0 30 -4 4])
    xlabel("Time index n")
    ylabel("Amplitude")
end
```



SIMPLE OPERATIONS ON SEQUENCES

1.5 SIGNAL SMOOTHING

Q1.29

```
% Program P1_5
% Signal Smoothing by Averaging
clf;
R = 51;
d = 0.8*(rand(R,1) - 0.5); % Generate random noise
m = 0:R-1;
s = 2*m.*(0.9.^m); % Generate uncorrupted signal
x = s + d'; % Generate noise corrupted signal
subplot(2,1,1);
plot(m,d','r-',m,s,'g--',m,x,'b-.');
xlabel('Time index n');ylabel('Amplitude');
legend('d[n] ','s[n] ','x[n] ');
x1 = [0 0 x];x2 = [0 x 0];x3 = [x 0 0];
y = (x1 + x2 + x3)/3;
subplot(2,1,2);
plot(m,y(2:R+1),'r-',m,x,'g--');
legend( 'y[n] ','x[n] ');
xlabel('Time index n');ylabel('Amplitude');
```

0

50

transpose

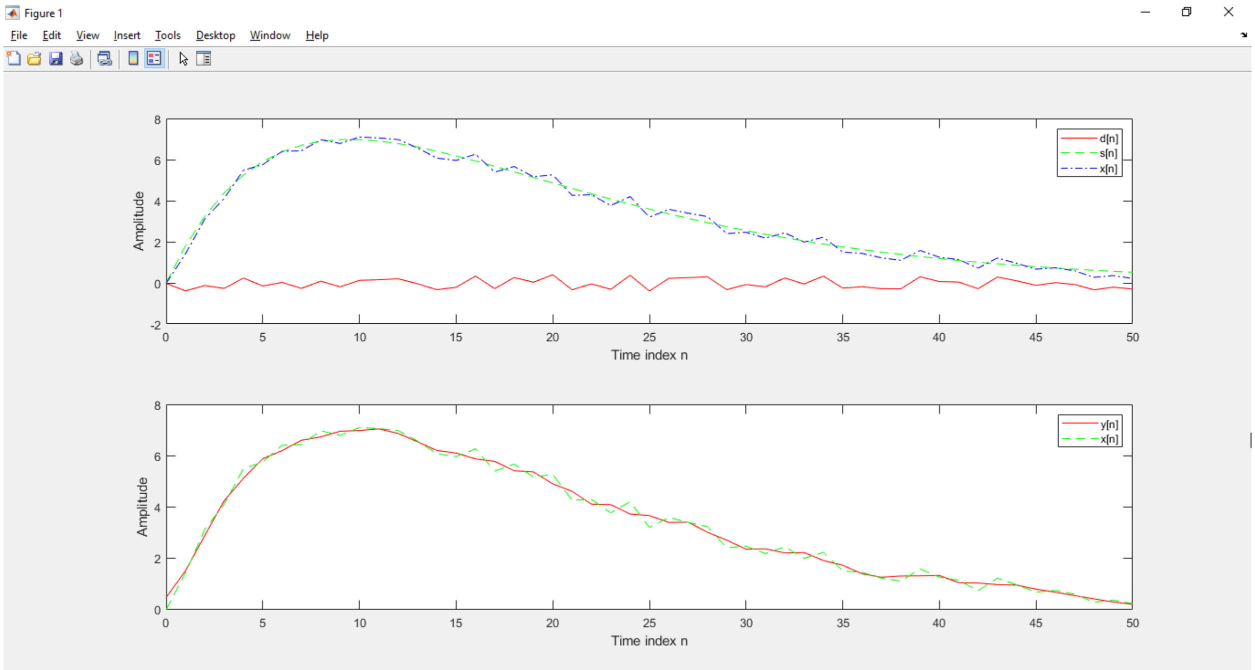
green + nét đứt

d[n]

legend

subplot

The signals generated by running Program P1_5 are displayed below:

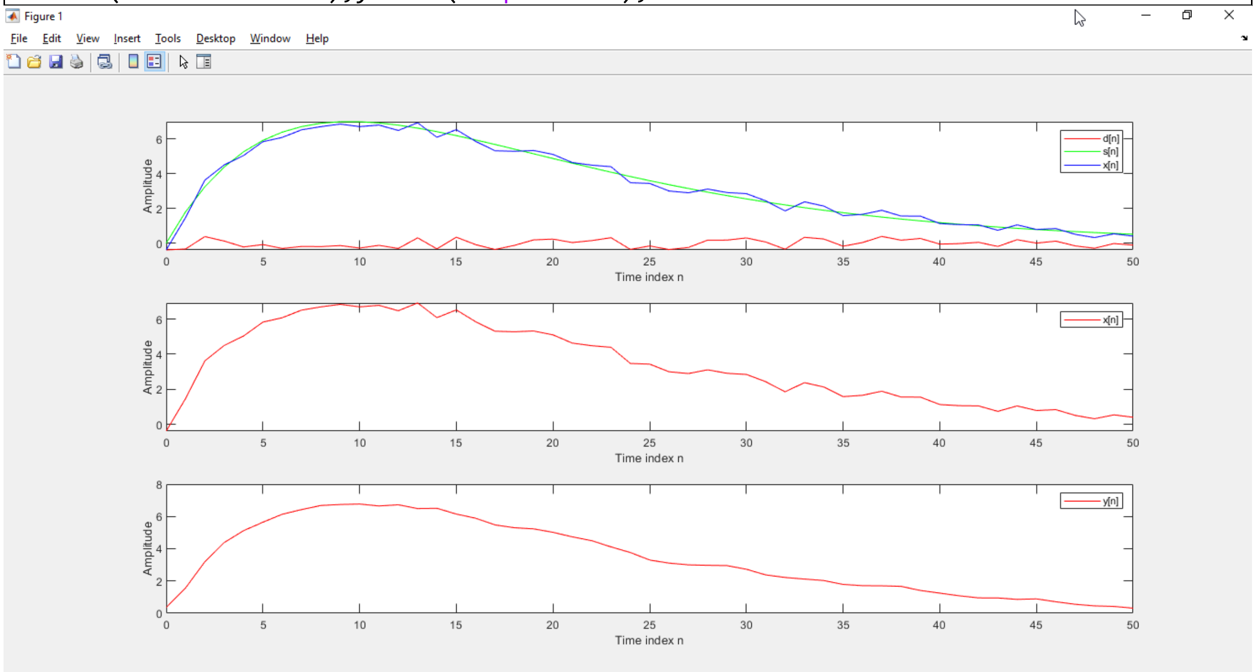


Modified

```
% Program P1_5
% Signal Smoothing by Averaging
clf;
R = 51;
% Generate random noise
d = 0.8*(rand(R,1) - 0.5);
m = 0:R-1;
% Generate uncorrupted signal
s = 2*m.*(0.9.^m);
% Generate noise corrupted signal
% d' equivalent transpose(d)

x = s + d';

subplot(3,1,1);
plot(m,d','r-',m,s,'g-',m,x,'b-');
xlabel('Time index n');ylabel('Amplitude');
legend('d[n] ','s[n] ','x[n] ');

x1 = [0 0 x];
x2 = [0 x 0];
x3 = [x 0 0];
% TO DE-NOISE
y = (x1+x2+x3)/3;

subplot(3,1,2);
title("Before Denoise")
plot(m, x, 'r-');
legend( 'x[n] ');
```

Mở rộng mả X

```
xlabel('Time index n');
ylabel('Amplitude');

subplot(3,1,3);
title("After Denoise")
plot(m,y(2:R+1),'r-');
legend( 'y[n] ');
xlabel('Time index n');ylabel('Amplitude');
```



Q1.30

The uncorrupted signal s[n] is: tích từng phần tử của hàm tăng tưởng m với hàm suy giảm (0.9.^m)
The additive noise d[n] is – là chuỗi nhiễu ngẫu nhiên được sinh từ hàm rand (phân phối xác suất đồng đều) biên độ từ -0,4 → 0.4.

Q1.31

The statement x = s + d CANNOT be used to generate the noise corrupted signal because – s là một vector hàng (kích thước 1x51) và d là một vector cột (kích thước 51x1).

Q1.32

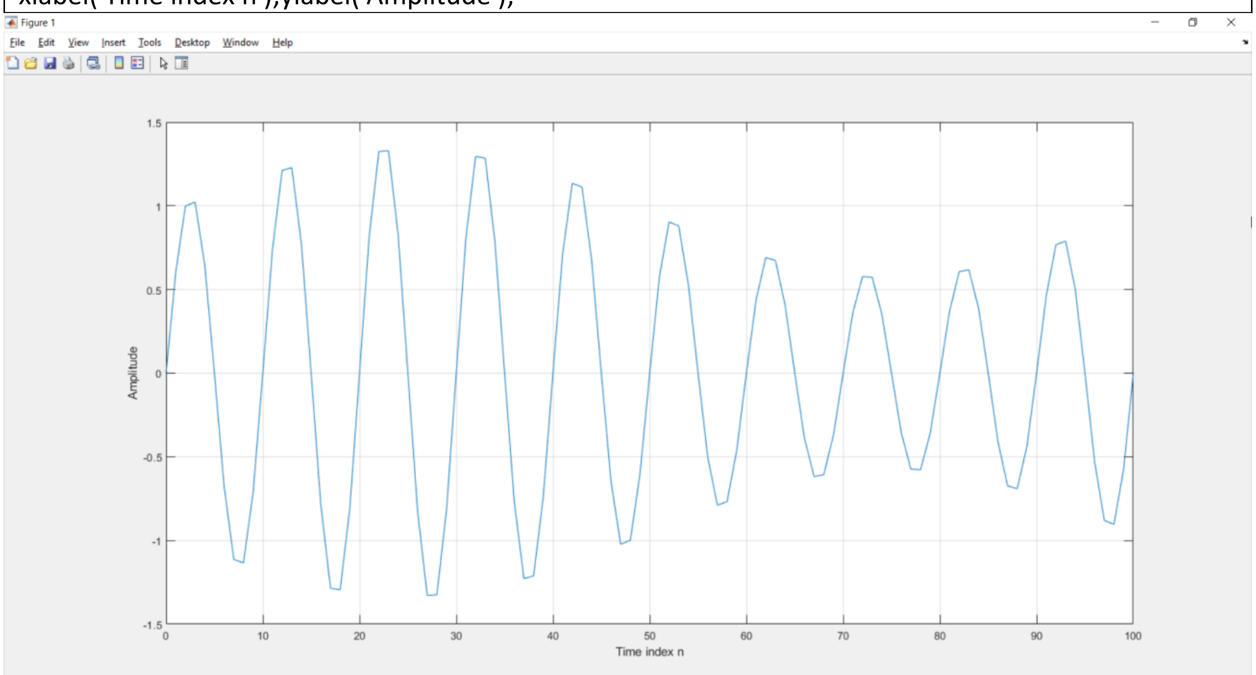
The relations between the signals x1, x2, and x3, and the signal x are –
x2 là mở rộng của x (thêm ‘0’ ở phía trước và phía sau)
X1 là trễ 1 mẫu của x2 (thêm ‘0’ ở phía trước x)
X3 là sớm 1 mẫu của x2(thêm ‘0’ ở phía sau x)

Q1.33

The purpose of the *legend* command is to create a legend for the graphs. In P1_5, the signals are plotted using different colors and line types; the legend provides information about which color and line type is associated with each signal.

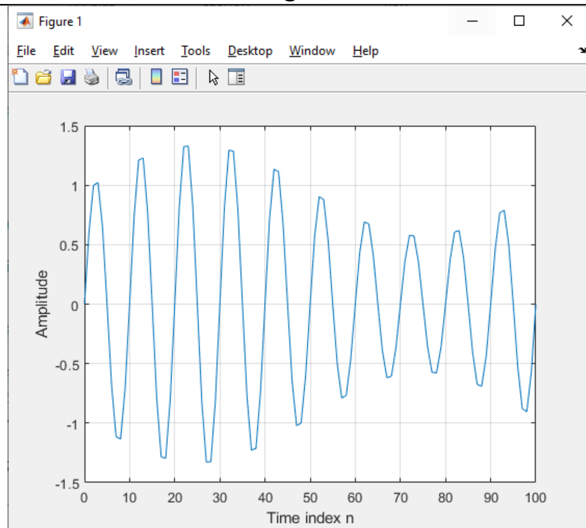
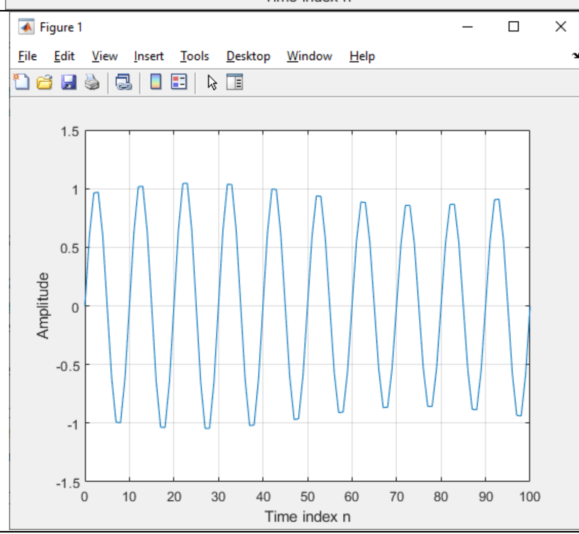
1.6 Generation of Complex Signals

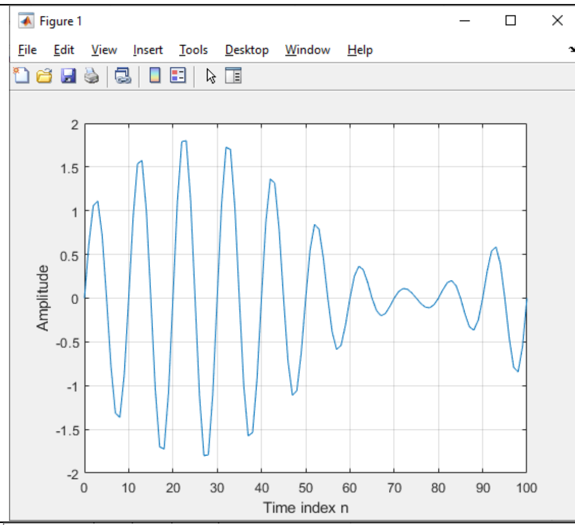
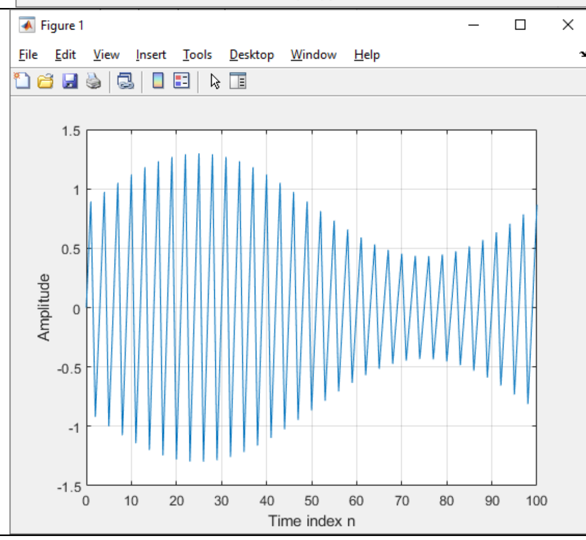
```
% Program P1_6
% Generation of amplitude modulated sequence
clf;
n = 0:100;
% ???
m = 0.4;
% Tần số sóng mang
fH = 0.1;
% Tần số tín hiệu
fL = 0.01;
% Sóng mang
xH = sin(2*pi*fH*n);
% Tín hiệu
xL = sin(2*pi*fL*n);
% Tín hiệu AM đã điều chế
y = (1+m*xL).*xH;
% Vẽ tín hiệu
plot(n,y);grid;
xlabel('Time index n');ylabel('Amplitude');
```



Q1.34

The amplitude modulated signals y[n] generated by running Program P1_6 for various values of the frequencies of the carrier signal xH[n] and the modulating signal xL[n], and various values of the modulation index **m** are shown below:

<table>
<tr><th>m</th><th>fH</th><th>fL</th><th>Figure</th></tr>
<tr><td>0.4</td><td>0.1</td><td>0.01</td><td>
</td></tr>
<tr><td>0.1</td><td>0.1</td><td>0.01</td><td>
</td></tr>
</table>

0.9	0.1	0.01	
0.5	1/3	0.01	

Q1.35

The difference between the arithmetic operators * and .* is:

* là phép nhân giữa hai ma trận .* là áp dụng phép nhân của các phần tử ở bên trong ma trận cùng kích thước($a_{i,j} * b_{i,j}$)

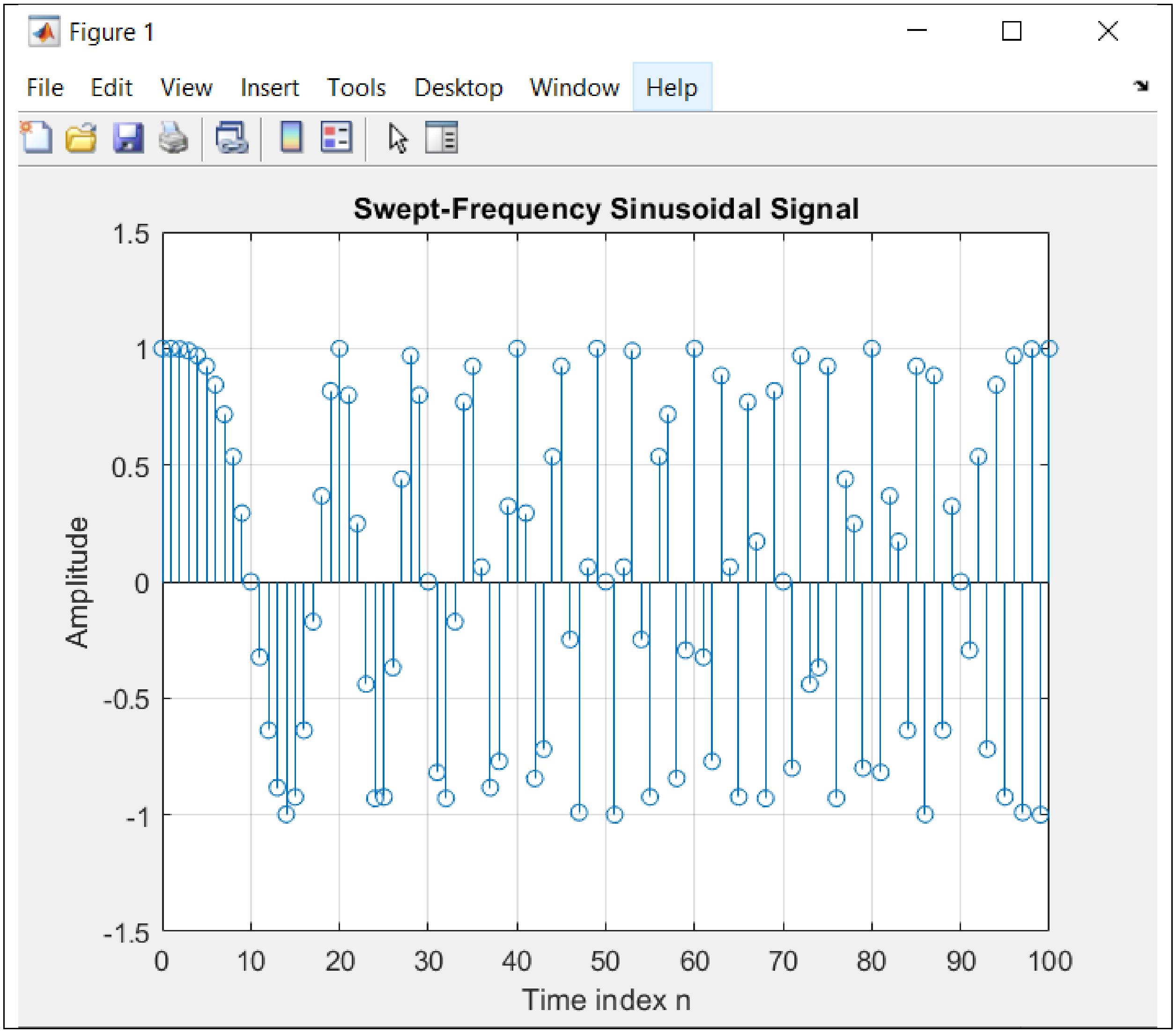
Program P1_7 is given below:

```
% Program P1_7
% Generation of a swept frequency sinusoidal sequence
n = 0:100;
a = pi/2/100;
b = 0;
arg = a*n.*n + b*n;
x = cos(arg);
clf;
stem(n, x);
```

```
axis([0,100,-1.5,1.5]);
title('Swept-Frequency Sinusoidal Signal');
xlabel('Time index n');
ylabel('Amplitude');
grid;
axis;
```

Q1.36

The swept-frequency sinusoidal sequence x[n] generated by running Program P1_7 is displayed below:



Q1.37

The minimum and maximum frequencies of this signal are

$$arg_i = \frac{\pi}{200} \cdot n_i^2 \rightarrow \omega_i = (arg_i)' = \frac{\pi}{100} n_i$$

$$\rightarrow f_i = \frac{\omega_i}{2\pi} = \frac{n_i}{200} \begin{cases} \rightarrow min = 0 \text{ at } n_0 = 0 \\ \rightarrow max = \frac{1}{2} \text{ at } n_{100} = 100 \end{cases}$$

Q1.38

The Program 1_7 modified to generate a swept sinusoidal signal with a minimum frequency of 0.1 and a maximum frequency of 0.3 is given below:

$$f_i : 0{,}1 \nearrow 0{,}3 : 0{,}2 \cdot \frac{n_i}{100} + 0{,}1 \rightarrow \omega_0 : \frac{\pi}{250} n_i + \frac{\pi}{5}$$

$$\rightarrow arg_i = \int \omega_i \, dn_i = \frac{\pi}{500} n_i^2 + \frac{\pi}{5} n_i + 0$$

```
% Program P1_7
% Generation of a swept frequency sinusoidal sequence
n = 0:01:100;
a = pi/500;
b = pi/5;
arg = a*n.*n + b*n;
x = cos(arg);
clf;
stem(n, x);
axis([0,100,-1.5,1.5]);
title('Swept-Frequency Sinusoidal Signal');
xlabel('Time index n');
ylabel('Amplitude');
grid; axis;
```

WORKSPACE INFORMATION

Q1.39

The information displayed in the command window as a result of the *who* command is:

Liệt kê tất cả các tên biến đang tồn tại trong bộ nhớ

Q1.40

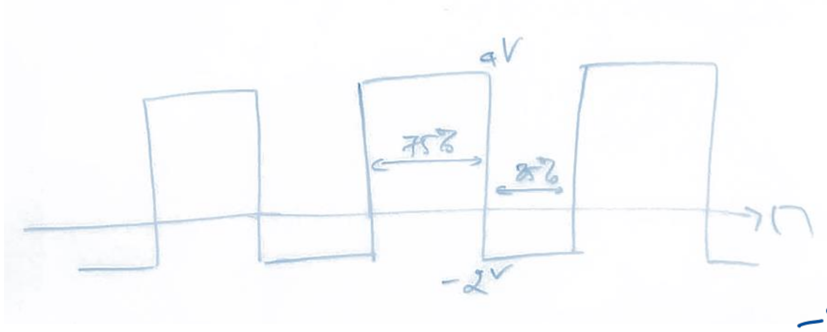
The information displayed in the command window as a result of the *whos* command is:

Liệt kê tên biến , kích thước, số phần tử và xét các phần ảo có khác 0 hay không.

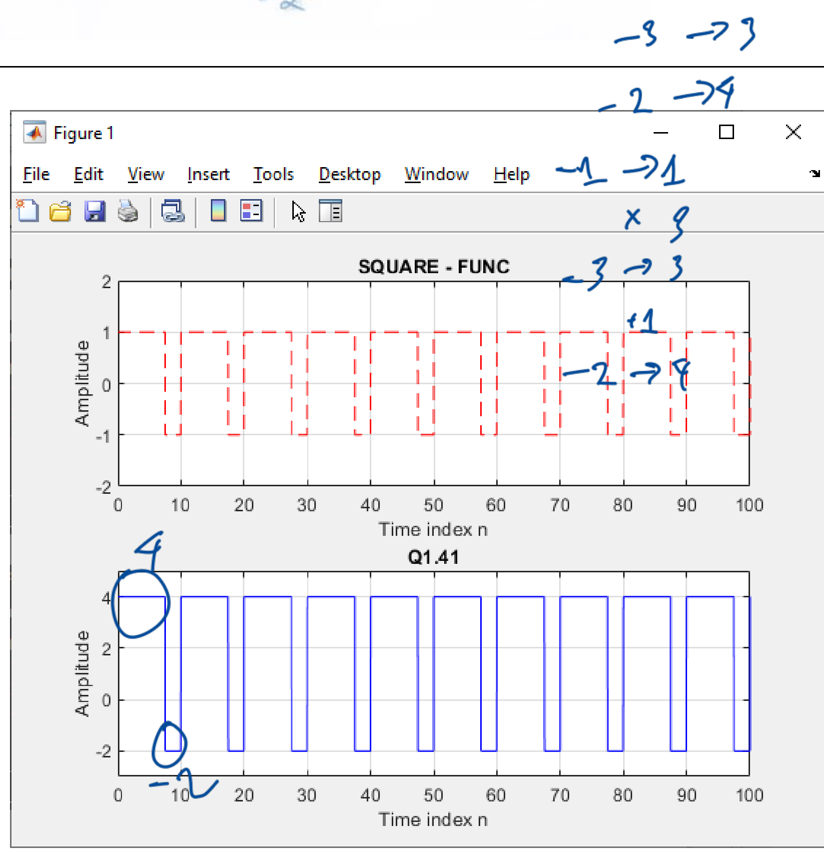
OTHER TYPES OF SIGNALS (Optional)

1.8 Squarewave and Sawtooth Signals

Q1.41




```
% by ngxx.fus
n = 0:0.001:100;

x = square( n * 2 * pi / 10, 75);

subplot(2,1,1);
plot(n, x,'r--');
axis([0 100 -2 2]);
title('SQUARE - FUNC');
grid;
xlabel('Time index n');
ylabel('Amplitude');

y = x*3+1;
subplot(2,1,2);
plot(n, y,'b-');
axis([0 100 -3 5]);
title('Q1.41');
grid;
xlabel('Time index n');
ylabel('Amplitude');
```



```
n = 0:0.01:30;
f = 0.1;
phase = 0;
duty=75;
Ahigh = 4;
Alow = -2;
arg = 2*pi*f*n + phase;
x =
Ahigh*square(arg,duty);
%bán kì dương thì gán
là Ahigh
x (x>0) = Ahigh;
%bán kì âm thì gán là
Alow
x (x<=0) = Alow;
clf; % Clear old graph
plot(n,x); % Plot the
generated sequence
axis([0 30 -5 5]);
grid;
title('Square Wave
Sequence of Fig.
1.1(a)');
xlabel('Time index n');
ylabel('Amplitude');
axis;
```