NAME - SANTOSH - [CB.EN.U4CCE20053]
DEPARTMENT - COMPUTER AND COMMUNICATION ENGINEERING

LAB TITLE AND CODE : SIGNAL PROCESSING LAB 19CCE281
EXPERIMENT NUMBER : 7
DATE : 07/10/2021

## TIME-SHIFTING PROPERTY OF DISCRETE-TIME FOURIER SERIES (DTFS)

* **AIM :**

Given an input sequence $x[n]$, verify the time-shifting property and evaluate Fourier coefficients for the resulting sequence $y[n]$ using DTFS.

* **SOFTWARE REQUIRED :**

Spyder IDE (Anaconda 3) - Python 3.8.8 (64-bit)

* **THEORY :**

① *Discrete-time signals* -

A discrete-time signal of fundamental period $N$ can consist of frequency components $f = \frac{1}{N}, \frac{2}{N}, \cdots, \frac{(N-1)}{N}$ besides $f=0$, the DC component. Therefore, the Fourier series representation of the discrete-time periodic signal contains only $N$ complex exponential basis functions.

② *Fourier series for discrete-time periodic signals* -

Given a periodic sequence $x[k]$ with period $N$, the Fourier series representation for $x[k]$ uses $N$ harmonically related exponential functions -

$$e^{j2\pi kn/N}, \quad k = 0, 1, \cdots, N-1$$

The Fourier series is expressed as -

$$x[k] = \sum_{n=0}^{N-1} C_n e^{j2\pi kn/N}$$

③ Fourier coefficients -

The Fourier coefficients $\{c_n\}$ are given by -

$$c_n = \frac{1}{N} \sum_{k=0}^{N-1} x[k] e^{-j 2\pi kn/N}$$

④ Time shifting -

Let $n_0$ be any integer. If $x[n]$ is a discrete-time signal to period $N$, then $y[n] = x[n-n_0]$. Therefore, the fourier series is -

$$x[n] \longleftrightarrow x[k]$$

which gives, $x[n-n_0] \xleftrightarrow{FS} e^{-j 2\pi k n_0/N} x[k]$,

the time-shifted Fourier coefficients.

✱ GRAPH PLOTTING ALGORITHM :

The following steps are followed -

① Define the x-axis and corresponding y-axis values as lists.
② Plot them on canvas using .plot() function.
③ Give a name to x-axis and y-axis using .xlabel() and .ylabel() functions.
④ Give a title to your plot using the .title() function.
⑤ Finally, to view your plot, we use the .show() function.

✱ THEORETICAL CALCULATION :

Given periodic sequence,

$x[n] = [\ldots\ 9\ 18\ 27\ 36\ 45\ \ldots]$

(Note : Boldface indicates $n=0$ index).

By keen observation, we find that fundamental period, $N = 5$

Angular frequency, $\Omega_0 = \frac{2\pi}{N} = \frac{2\pi}{5}$.

$$x[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-jk\frac{2\pi}{5}n} = \frac{1}{5} \sum_{n=0}^{4} x[n] e^{-jk\frac{2\pi}{5}n}$$

$$= \frac{1}{5}\left[x[0]e^{0} + x[1]e^{-jk\frac{2\pi}{5}} + x[2]e^{-jk\frac{4\pi}{5}} + x[3]e^{-jk\frac{6\pi}{5}} + x[4]e^{-jk\frac{8\pi}{5}}\right]$$

$$= \frac{1}{5}\left[9 + 18e^{-jk\frac{2\pi}{5}} + 27e^{-jk\frac{4\pi}{5}} + 36e^{-jk\frac{6\pi}{5}} + 45e^{-jk\frac{8\pi}{5}}\right]$$

Taking 9 common,

$$x[k] = \frac{9}{5}\left[1 + 2e^{-jk\frac{2\pi}{5}} + 3e^{-jk\frac{4\pi}{5}} + 4e^{-jk\frac{6\pi}{5}} + 5e^{-jk\frac{8\pi}{5}}\right]$$

Substituting the values of k in $x[k]$, we find that -

$x[0] = 21 + 0j$, for $k=0$

$x[1] = -4.5 + 6.2j$, for $k=1$

$x[2] = -4.5 + 1.5j$, for $k=2$

$x[3] = -4.5 - 1.5j$, for $k=3$

$x[4] = -4.5 - 6.2j$, for $k=4$

Upon computing the magnitude spectrum, we find that -

$|x[0]| = \sqrt{(21)^2 + (0)^2} = 21.0$

$|x[1]| = \sqrt{(-4.5)^2 + (6.2)^2} = 7.66$

$|x[2]| = \sqrt{(-4.5) + (1.5)^2} = 4.73$

$|x[3]| = \sqrt{(-4.5)^2 + (-1.5)^2} = 4.73$

$|x[4]| = \sqrt{(-4.5)^2 + (-6.2)^2} = 7.66$

Upon computing the phase spectrum, we find that -

$$(x[n]) = \tan^{-1}\left(\frac{y}{x}\right) = \_\_^{\circ} = \_\_ \times \frac{\pi}{180} \text{ radians}$$

Therefore,

$(x[0]) = 0$ radians

$(x[1]) = -0.94$ radians

$(x[2]) = -0.31$ radians

$(x[3]) = 0.31$ radians

$(x[4]) = 0.94$ radians

(4)

Given shifting factor, $n_0 = 2$.
Therefore, obtained resulting sequence,

$y[n] = [\ldots \mathbf{36}\ 45\ 9\ 18\ 27 \ldots]$

(Note: Boldface indicates $n=0$ index)

Again, fundamental period $(N) = 5$

Angular frequency $(\Omega_0) = \frac{2\pi}{N} = \frac{2\pi}{5}$

Therefore, Fourier coefficients of $y[n]$ using time shifting property of DTFS are -

$y[0] = e^{-j2\pi k n_0/N} \cdot x[0] = e^{-j2\pi(0)2/5} \cdot (27 + 0j) = (27 + 0j)$

$y[1] = e^{-j2\pi(1)2/5} \cdot x[1] = e^{-j2\pi(1)2/5} \cdot (-4.5 + 6.2j) = (7.28 - 2.37j)$

$y[2] = e^{-j2\pi(2)2/5} \cdot x[2] = e^{-j2\pi(2)2/5} \cdot (-4.5 + 1.5j) = (-2.78 - 3.83j)$

$y[3] = e^{-j2\pi(3)2/5} \cdot x[3] = e^{-j2\pi(3)2/5} \cdot (-4.5 - 1.5j) = (-2.78 + 3.83j)$

$y[4] = e^{-j2\pi(4)2/5} \cdot x[4] = e^{-j2\pi(4)2/5} \cdot (-4.5 - 6.2j) = (7.28 + 2.37j)$

Upon computing the magnitude spectrum, we find that -

$|y[0]| = \sqrt{(27)^2 + (0)^2} = 27.0$

$|y[1]| = \sqrt{(7.28)^2 + (-2.37)^2} = 7.66$

$|y[2]| = \sqrt{(-2.78)^2 + (-3.83)^2} = 4.73$

$|y[3]| = \sqrt{(-2.78)^2 + (3.83)^2} = 4.73$

$|y[4]| = \sqrt{(7.28)^2 + (2.37)^2} = 7.66$

Upon computing the phase spectrum, we find that -

$(y[n]) = \tan^{-1}\left(\frac{y}{x}\right) = \_\_^{\circ} = \_\_ \times \frac{\pi}{180}$ radians

$(y[0]) = 0.0$ radians

$(y[1]) = -0.31$ radians

$(y[2]) = 0.94$ radians

$(y[3]) = -0.94$ radians

$(y[4]) = 0.31$ radians

* PROGRAM WITH COMMENTS:

```
1  # import library source files
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  # Given Input Sequence x[n]
6  size_input = int (input ("enter the size of input x[n]: "))
7  user_defined_input = [0] * (size_input)
8  print ("Enter the elements of the input x[n] one-by-one
   as follows :- ")
9  for sample in range (0, size_input, 1):
10     user_defined_input [sample] = input ("Element " +
   str (sample + 1) + ": ")
11 print ("\n The entered input x[n] is : -" + str (user_defined_
   input) + "\n")
12
13 # Null Array Declaration for Input Sequence x[n]
14 x = []
15 fourier_series_x = []
16 magnitude_spectrum_x = []
17 phase_spectrum_x = []
18
19 # Null Array Declaration for Resulting sequence y[n]
20 y = []
21 fourier_series_y = []
22 magnitude_spectrum_y = []
23 phase_spectrum_y = []
24
25 # Compute Fundamental Time Period for x[n]
26 for sample in range (1, size_input - 1):
27     x.append (user_defined_input [sample-1])
28     if user_defined_input [0] == user_defined_input [sample]:
29     → breakpoint = sample - 2 # How many times does the sequence repeat?
30         break
```

```
31
32 # Obtain frequency domain representation
33 boldface = int (input ("Enter the element position that
   indicates n=0 index : "))
34 index = int ((boldface -1) / len (x))
35
36 # sketch Input sequence x[n]
37 plt. xlabel ('n')
38 plt. ylabel ('x[n]')
39 plt. title ("Input sequence : ... {} ... ". format (x))
40 plt. stem (np. arange (-(index * len(x)), len(x) *(breakpoint -index),
   a * breakpoint)
41 plt. show ()
42
43 # Compute Fourier series for x[n]
44 def summation (k):
45     sum = 0
46     for n in range (len(x)):
47         exponential = np. exp (-2j * np.pi * k * n / len (x))
48         sum = sum + float (x[n]) * exponential
49     return (sum)
50
51 # Compute Fourier coefficients for x[n]
52 for k in range (len(x)):
53     fourier_series_x. append (summation (k) / len(x))
54 print ("\n The Fourier coefficients for x[n] are as follows : {}
   ". format (fourier_series_x))
55
56 # sketch Fourier coefficients of x[n]
57 plt. xlabel ('n')
58 plt. ylabel ('x[n]')
59 plt. title ("Fourier coefficients of x[n]")
60 plt. stem (np. arange (0, len (fourier_series_x)). np. real (fourier-
   series_x))
```

```
plt.show()

# Compute Magnitude and Phase Spectrum for x[n]
for sample in range(len(fourier_series_x)):
    magnitude_spectrum_x.append((fourier_series_x[sample].real**2 + fourier_series_x[sample].imag**2)**0.5)
    phase = np.arctan(fourier_series_x[sample].imag/fourier_series_x[sample].real)
    phase_spectrum_x.append(phase)

# sketch Magnitude Spectrum of x[n]
plt.xlabel('k')
plt.ylabel('|x[k]|')
plt.title("Magnitude Spectrum of x[n]: ...{}...".format(magnitude_spectrum_x))
plt.stem(np.arange(0, len(magnitude_spectrum_x)), magnitude_spectrum_x)
plt.show()
print("\n The Magnitude Spectrum of x[n] is as follows: {}".format(magnitude_spectrum_x))

# sketch Phase Spectrum of x[n]
plt.xlabel('k')
plt.ylabel('Angle (in radians)')
plt.title("Phase Spectrum of x[n]: ...{}...".format(phase_spectrum_x))
plt.stem(np.arange(0, len(phase_spectrum_x)), phase_spectrum_x)
plt.show()
print("\n The Phase Spectrum of x[n] is as follows: {}".format(phase_spectrum_x))

Intermediate = []
```

```
# Copy Input sequence to Temporary sequence intermediate[n]
for sample in range (size_input):
    intermediate.append (user_defined_input [sample])

# Input Shifting Factor
shifting_factor = int (input ("\n Enter the shifting factor: "))

# shift Temporary sequence by "shifting_factor" of units
for i in range (shifting_factor):
    intermediate.append (0) # Adds required zeros at the end of list
    for j in range (len (intermediate) -1, 0, -1):
        intermediate [j] = intermediate [j-1] # shifts the zeros at the beginning of list
    intermediate [i] = 0 # Prepares for next iteration

# Place rest of elements in relevant position
temp = size_input - shifting_factor
for k in range (shifting_factor)
    intermediate [k] = user_defined_input [temp]
    temp = temp + 1

# Compute Fundamental Time Period for y[n]
for sample in range (1, size_input -1):
    y.append (intermediate [sample -1])
    if user_defined_input [0] == user_defined_input [sample]:
        break

print ("\n The resulting sequence y[n] is :-" + str(y) + "\n")

# sketch resulting sequence y[n]
plt.xlabel ('n')
plt.ylabel ('y[n]')
```

```
plt.title("Resulting sequence: ...{}... ".format(y))
plt.stem(np.arange(-index*len(y)), len(y)*(breakpoint-index),
y*breakpoint)
plt.show()

# Compute Fourier coefficients for y[n]
for k in range(len(x))
    fourier_series_y.append(np.exp(-2j*np.pi*k*
shifting_factor/len(x))*fourier_series_x[k])
print("\n The Fourier coefficients for y[n] are as follows: {}"
.format(fourier_series_y))

# sketch Fourier coefficients of y[n]
plt.xlabel('n')
plt.ylabel('y[n]')
plt.title("Fourier coefficients of y[n]")
plt.stem(np.arange(0, len(fourier_series_y)),
np.real(fourier_series_y))
plt.show()

# Compute magnitude and Phase Spectrum for y[n]
for sample in range(len(fourier_series_y)):
    magnitude_spectrum_y.append((fourier_series_y[sample]
.real**2 + fourier_series_y[sample].imag**2)**0.5)
    phase=np.arctan(fourier_series_y[sample].imag/
fourier_series_y[sample].real)
    phase_spectrum_y.append(phase)

# sketch magnitude Spectrum of y[n]
plt.xlabel('k')
plt.ylabel('|y[k]|')
plt.title("magnitude Spectrum of y[n]:...{}...".format
(magnitude_spectrum_y))
```

```
144 plt.stem(np.arange(0, len(magnitude_spectrum_y)),
    magnitude_spectrum_y)
145 plt.show()
146 print("\n The magnitude spectrum of y[n] is as follows: {}".
    format(magnitude_spectrum_y))
147
148 #sketch Phase Spectrum of y[n]
149 plt.xlabel('k')
150 plt.ylabel('Angle(in radians)')
151 plt.title("Phase Spectrum of y[n]: ... {} ...".format
    (phase_spectrum_y))
152 plt.stem(np.arange(0, len(phase_spectrum_y)), phase_spectrum
153 plt.show()
154 print("\n The Phase Spectrum of y[n] is as follows: {}"
    .format(phase_spectrum_y))
155
```

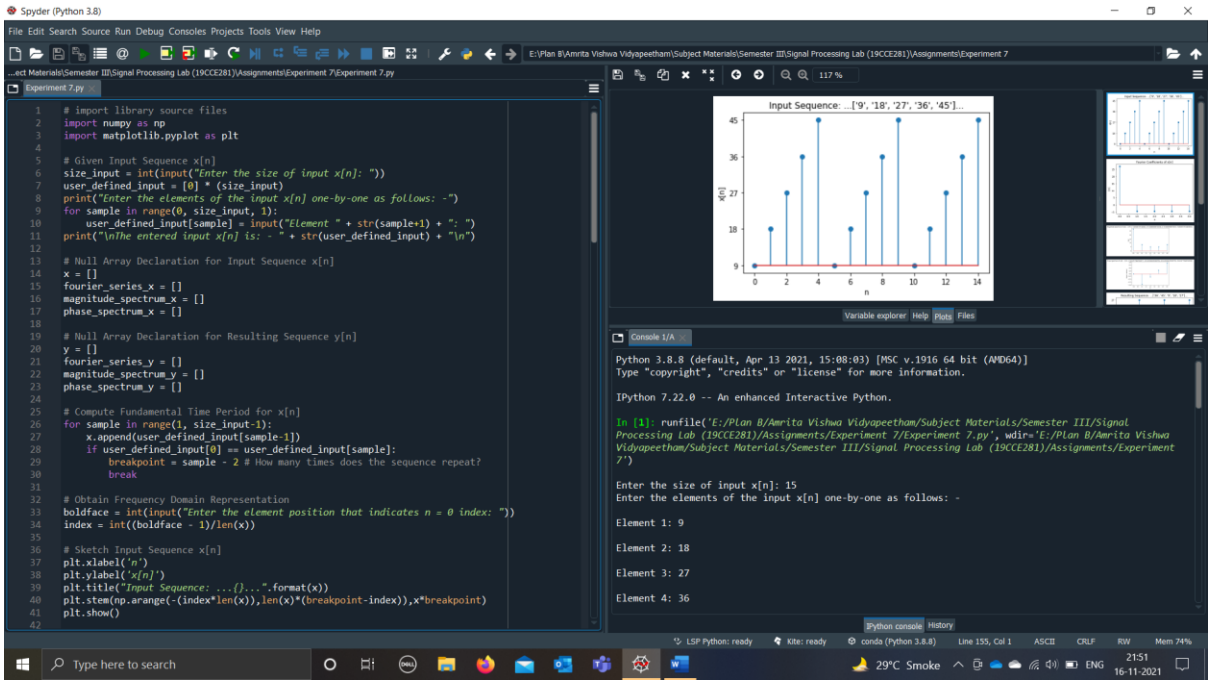
* **INFERENCE:**

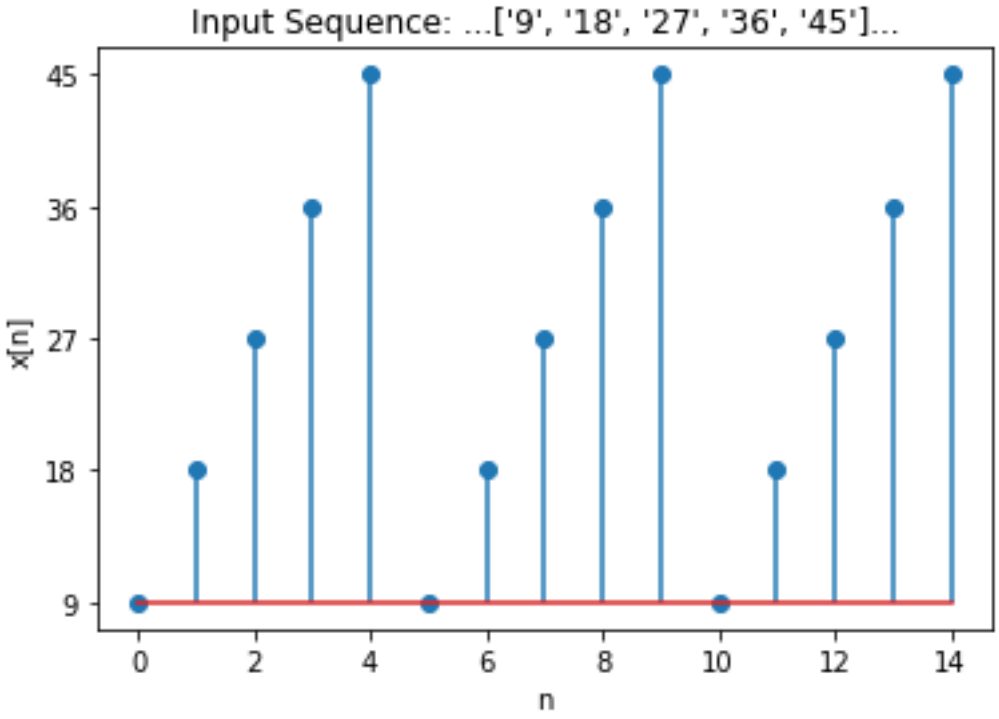
For the given input sequence x[n], demonstrated the time-shifting property and compute Fourier coefficients for the resulting sequence y[n] using DTFS.

RESULTS

VERIFIED.

## ***Given Input Sequence***




**Step 1**: Import library source files, numpy and matplotlib.pyplot.

**Step 2**: Enter the size of the input and declare an array with the number of elements equal to the value of size.

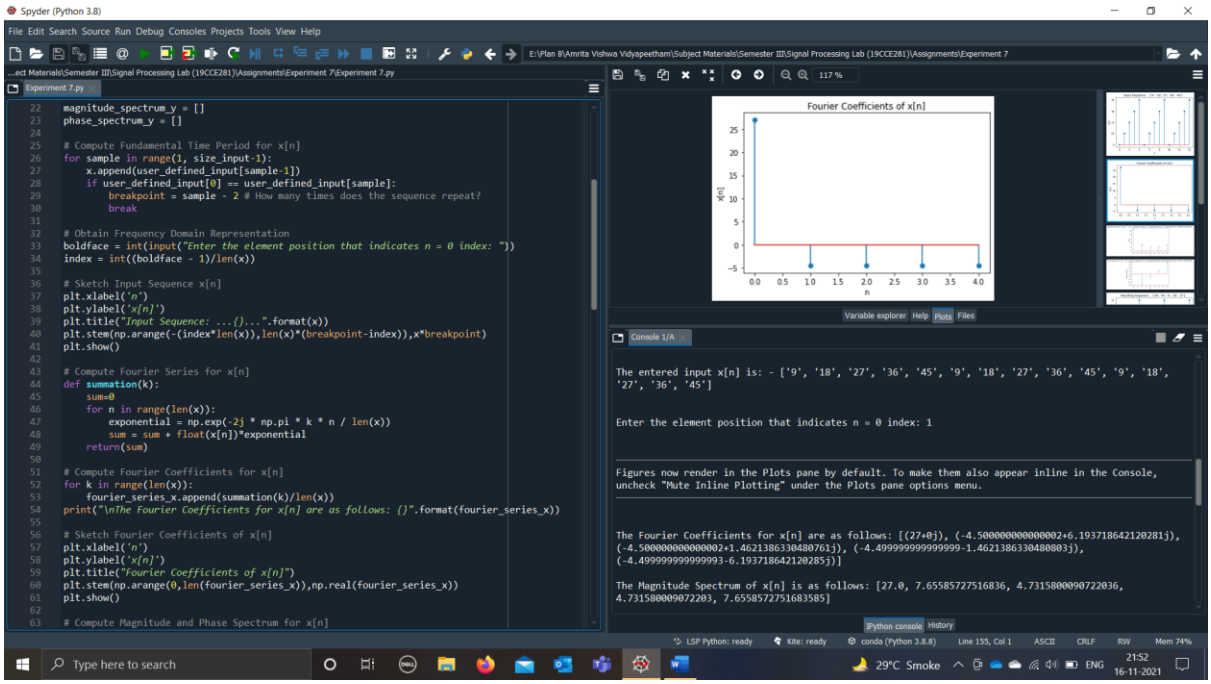
**Step 3**: Enter the elements of the input along with the element position that indicates n = 0 index.

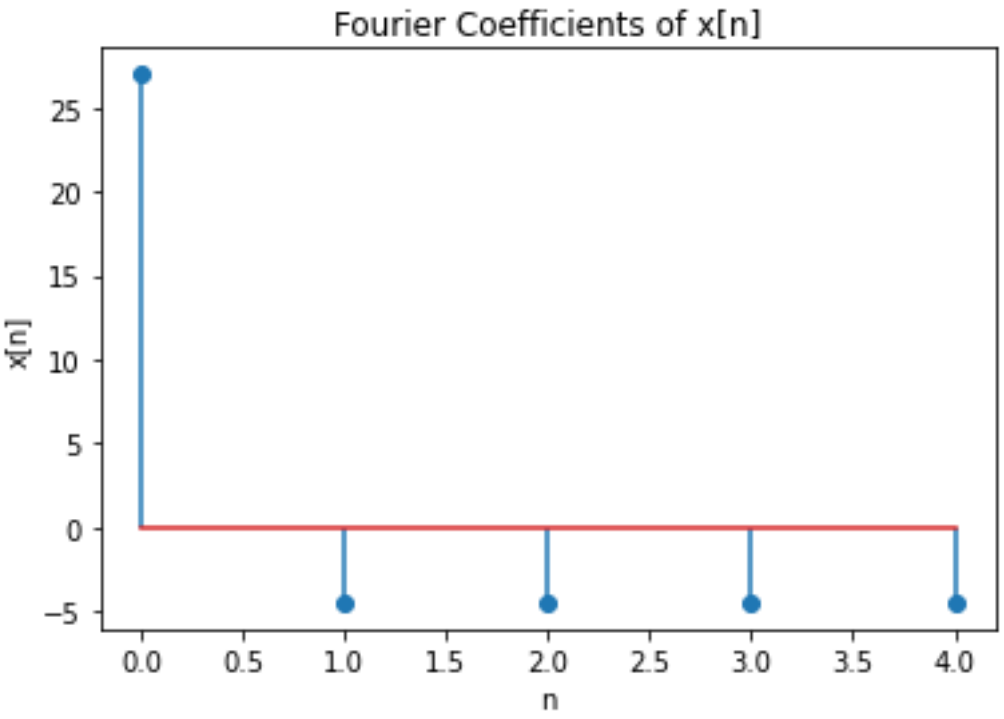
**Step 4**: Declare required null arrays required for input sequence x[n] and resulting sequence y[n].

**Step 5**: Compute fundamental time period and obtain frequency domain representation.

**Step 6**: Print the input sequence and show its labelled plot x[n].

## ***Plot Fourier Coefficients of x[n]***



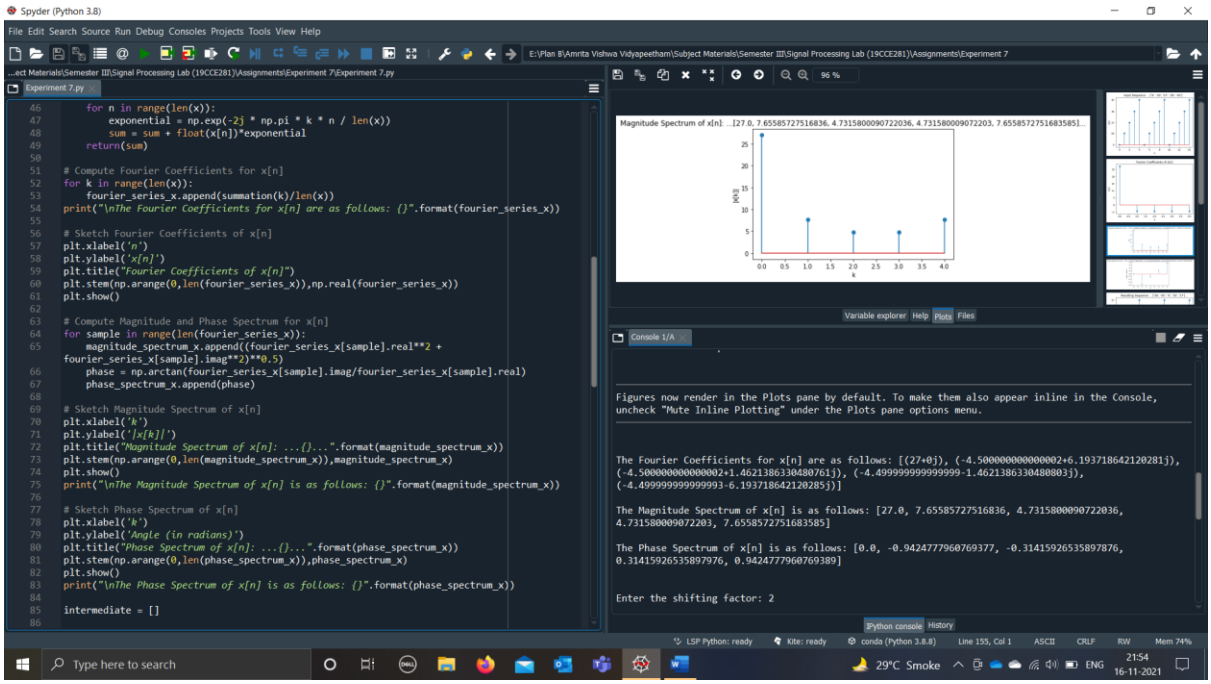




**Step 1**: Compute Fourier series for x[n].

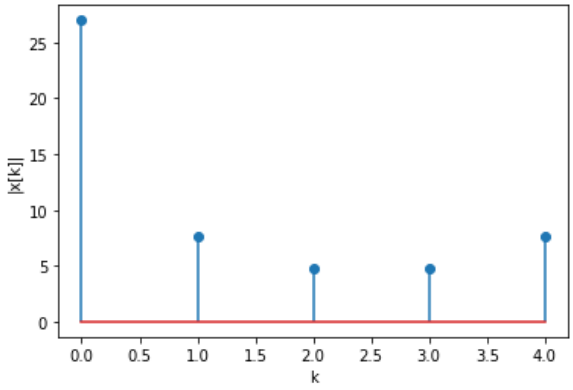
**Step 2**: Compute Fourier coefficients for x[n].

**Step 3**: Sketch Fourier coefficients of x[n].

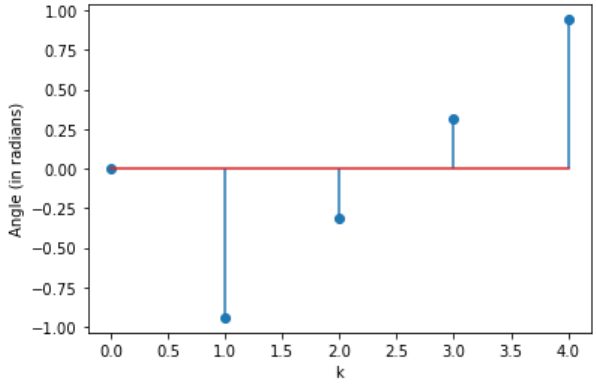
## ***Plot Magnitude and Phase Spectrum of x[n]***




Magnitude Spectrum of x[n]: ...[27.0, 7.65585727516836, 4.7315800090722036, 4.731580009072203, 7.6558572751683585]...




Phase Spectrum of x[n]: ...[0.0, -0.9424777960769377, -0.31415926535897876, 0.31415926535897976, 0.9424777960769389]...

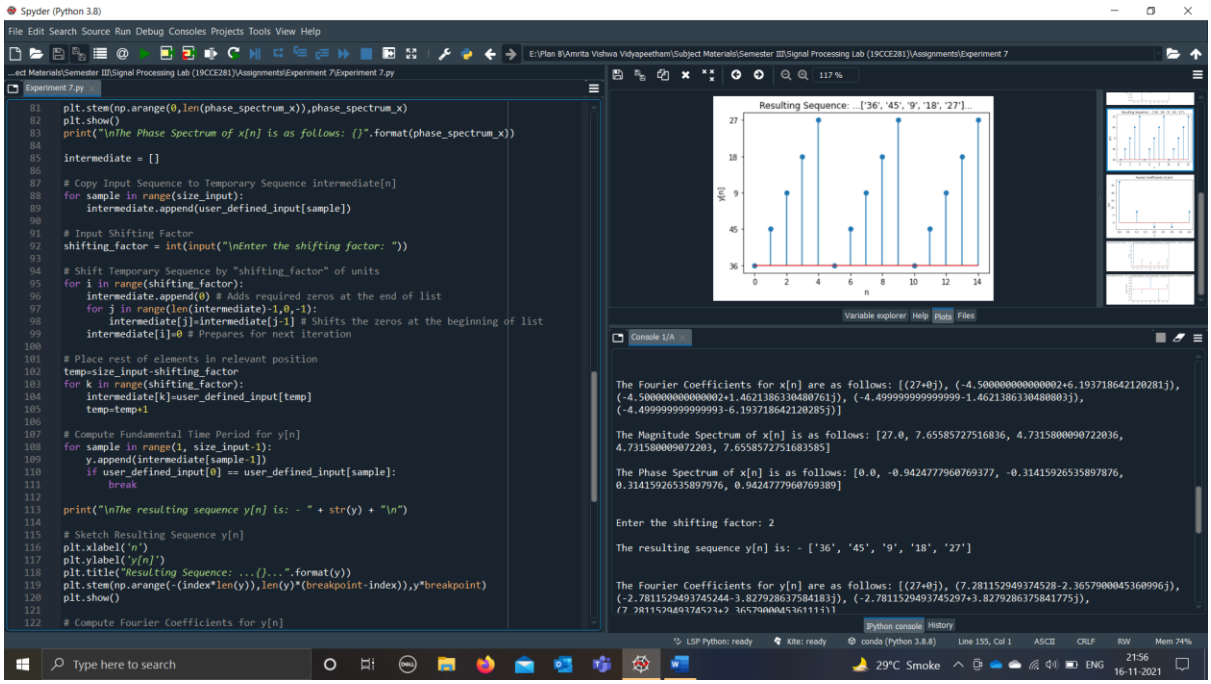


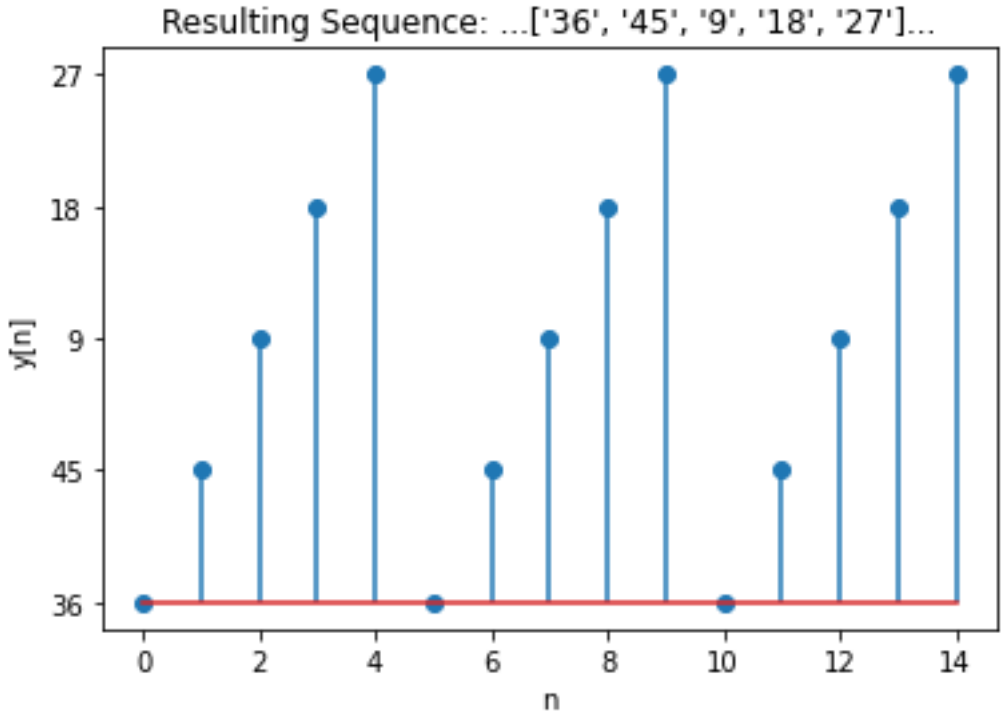

**Step 1**: Compute magnitude and phase spectrum for x[n].

**Step 2**: Sketch magnitude spectrum of x[n].

**Step 3**: Sketch phase spectrum of x[n].

## ***Resulting Sequence – y[n]***



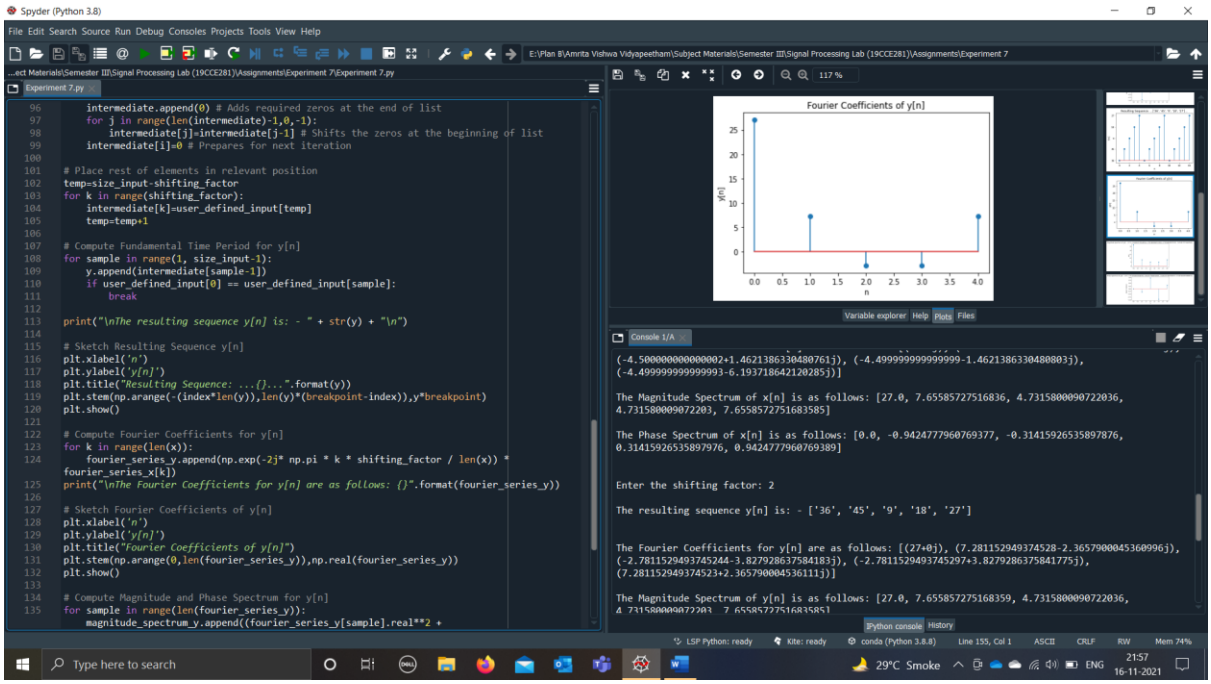

**Step 1**: Declare a NULL array “intermediate” and copy the input sequence to temporary sequence intermediate[n].

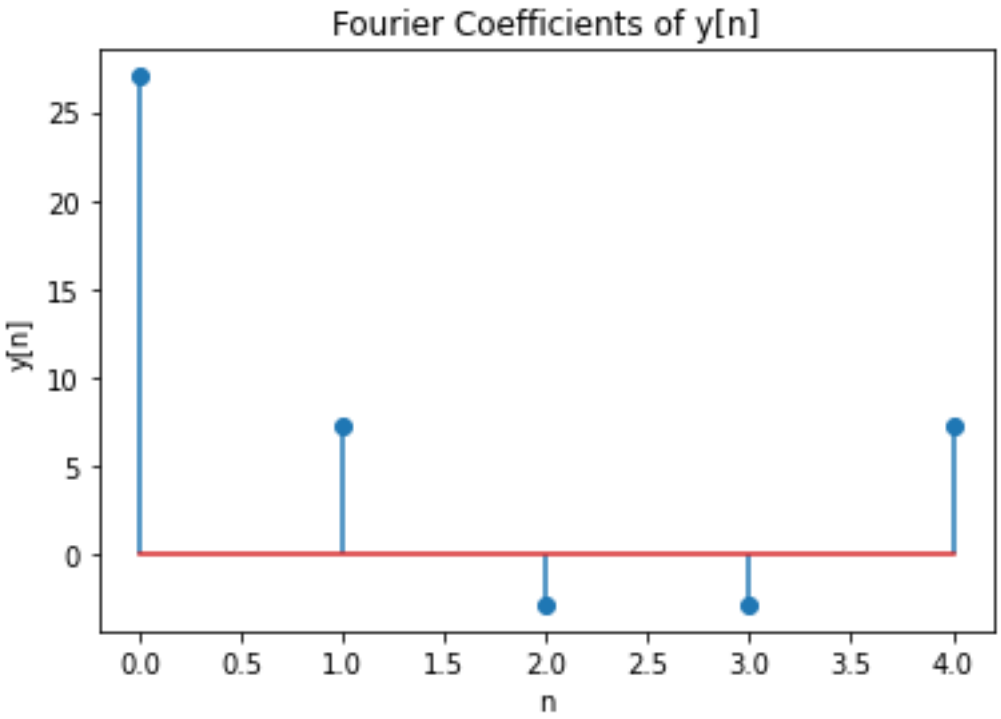
**Step 2**: Enter the shifting factor and shift temporary sequence by that many numbers of units.

**Step 3**: Place the rest of the elements in the relevant position and compute the fundamental time period for y[n].

**Step 4**: Print and sketch the resulting sequence, y[n].

## ***Plot Fourier Coefficients of y[n]***

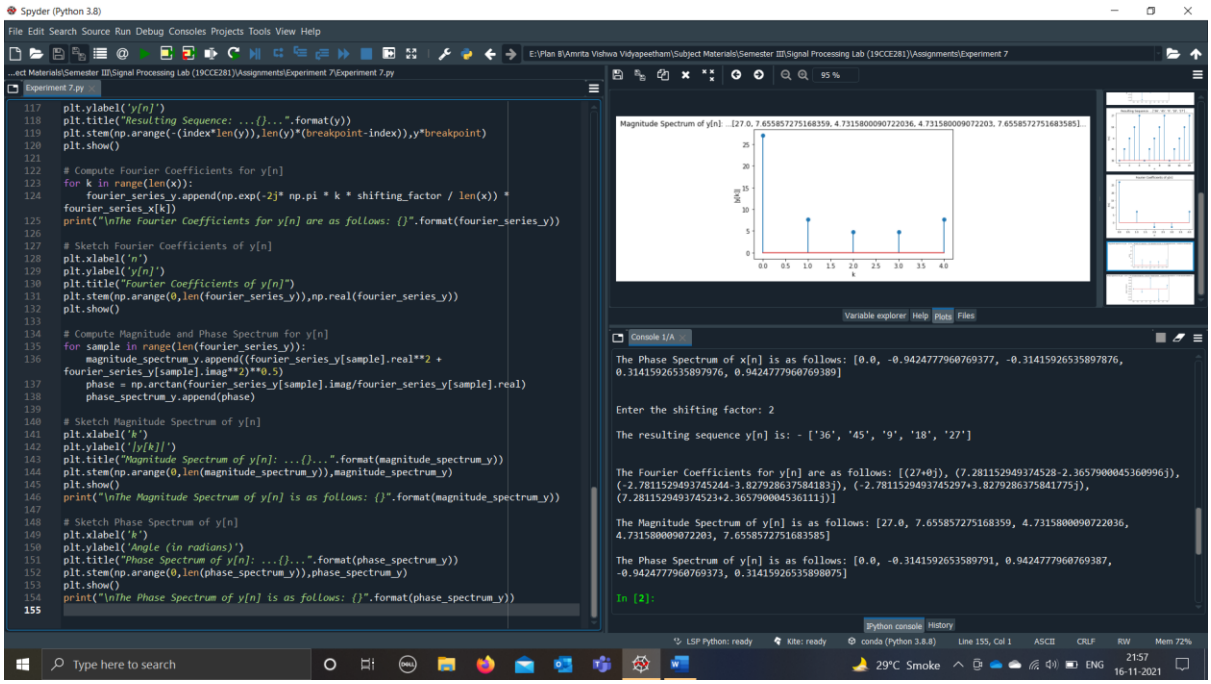


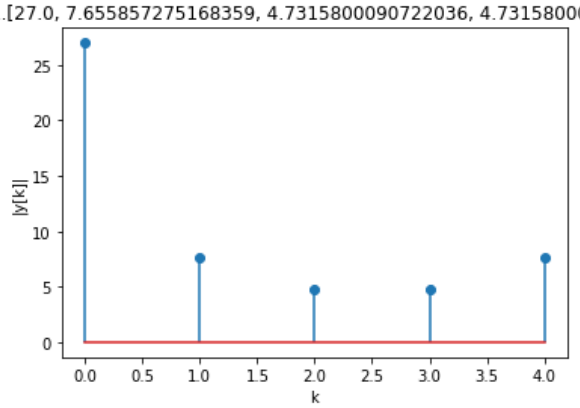

**Step 1**: Compute Fourier coefficients for y[n].

**Step 2**: Sketch Fourier coefficients of y[n].

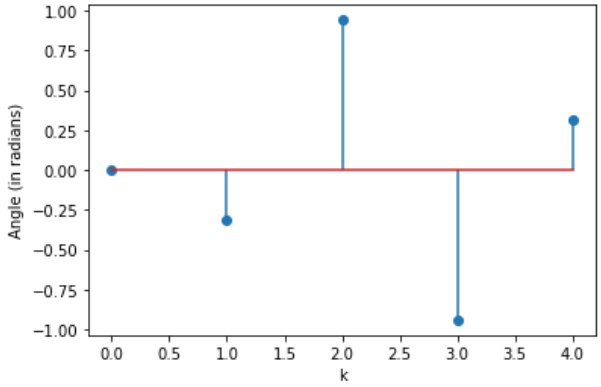
## ***Plot Magnitude and Phase Spectrum of y[n]***




Magnitude Spectrum of y[n]: ...[27.0, 7.655857275168359, 4.7315800090722036, 4.731580009072203, 7.6558572751683585]...




Phase Spectrum of y[n]: ...[0.0, -0.3141592653589791, 0.9424777960769387, -0.9424777960769373, 0.31415926535898075]...




**Step 1**: Compute magnitude and phase spectrum for y[n].

**Step 2**: Sketch magnitude spectrum of y[n].

**Step 3**: Sketch phase spectrum of y[n].

```
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

Restarting kernel...
```

---

```
In [1]:         'E:/Plan B/Amrita Vishwa Vidyapeetham/Subject Materials/Semester III/
Signal Processing Lab (19CCE281)/Assignments/Experiment 7/Experiment 7.py'      ='E:/Plan
B/Amrita Vishwa Vidyapeetham/Subject Materials/Semester III/Signal Processing Lab
(19CCE281)/Assignments/Experiment 7'

Enter the size of input x[n]: 15
Enter the elements of the input x[n] one-by-one as follows: -

Element 1: 9

Element 2: 18

Element 3: 27

Element 4: 36

Element 5: 45

Element 6: 9

Element 7: 18

Element 8: 27

Element 9: 36

Element 10: 45

Element 11: 9

Element 12: 18

Element 13: 27

Element 14: 36

Element 15: 45

The entered input x[n] is: - ['9', '18', '27', '36', '45', '9', '18', '27', '36', '45',
'9', '18', '27', '36', '45']


Enter the element position that indicates n = 0 index: 1

The Fourier Coefficients for x[n] are as follows: [(27+0j),
(-4.500000000000002+6.193718642120281j), (-4.500000000000002+1.4621386330480761j),
```

```
(-4.499999999999999-1.4621386330480803j), (-4.499999999999993-6.193718642120285j)]

The Magnitude Spectrum of x[n] is as follows: [27.0, 7.65585727516836, 4.7315800090722036,
4.731580009072203, 7.6558572751683585]

The Phase Spectrum of x[n] is as follows: [0.0, -0.9424777960769377, -0.31415926535897876,
0.31415926535897976, 0.9424777960769389]


Enter the shifting factor: 2

The resulting sequence y[n] is: - ['36', '45', '9', '18', '27']


The Fourier Coefficients for y[n] are as follows: [(27+0j),
(7.281152949374528-2.3657900045360996j), (-2.7811529493745244-3.827928637584183j),
(-2.7811529493745297+3.8279286375841775j), (7.281152949374523+2.365790004536111j)]

The Magnitude Spectrum of y[n] is as follows: [27.0, 7.655857275168359,
4.7315800090722036, 4.731580009072203, 7.6558572751683585]

The Phase Spectrum of y[n] is as follows: [0.0, -0.3141592653589791, 0.9424777960769387,
-0.9424777960769373, 0.31415926535898075]

In [2]:
```