NAME - SANTOSH (CB.EN.U4CCE20053)

DEPARTMENT - COMPUTER AND COMMUNICATION ENGINEERING (CCE) A

LAB TITLE AND CODE : SIGNAL PROCESSING LAB 19CCE281

EXPERIMENT NO: 2

DATE : 21/09/2021

GENERATION OF ELEMENTARY SIGNALS

★ AIMS :

Plot continuous-time and discrete-time sequence for the generation of sinusoidal, exponential, impulse, unit step and ramp functions using Python code.

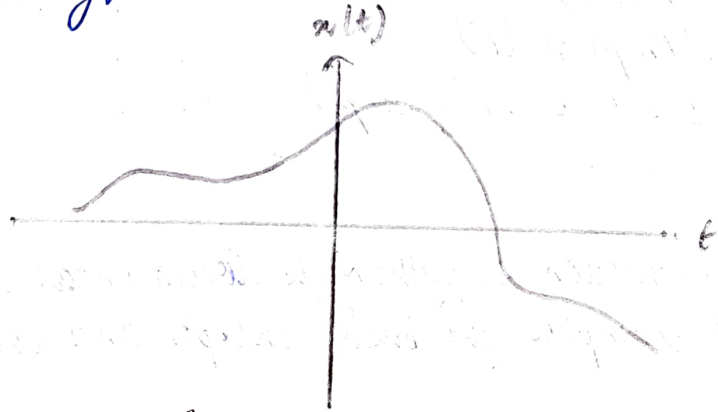
★ SOFTWARE REQUIRED :

Spyder IDE (Anaconda 3) - Python 3.9.7 (64-bit)

★ THEORY :

① Continuous-Time Signal - It is a varying quantity (a signal) whose domain, which is often time, is a continuum (e.g.: a connected interval of the reals). That is, the function's domain is an uncountable set. The function itself need not be continuous.

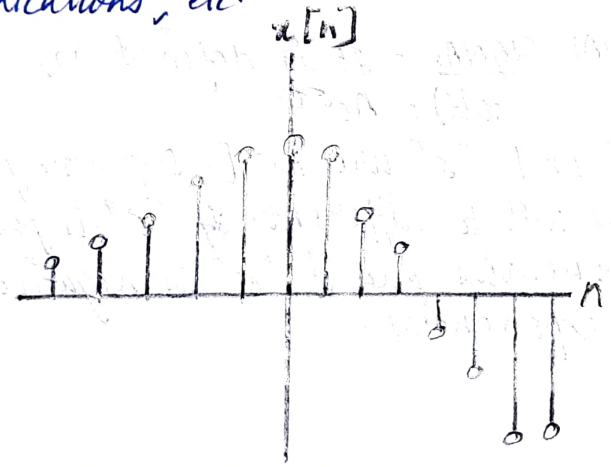
Defined for all time instants and not at specific time instants. The electrical signals derived in proportion with the physical quantities such as temperature, pressure, sound etc. are generally continuous. Applications of these signals include Public Switched Telephone Network (PSTN), FM Radio Broadcasting, Image Processing, etc.



Continuous-Time Signal

(2) Discrete-Time Signal – It is a time series consisting of a sequence of a sequence of quantities. Discrete-time views values of variables as occurring at distinct, seperate "points in time", or equivalently as being unchanged throughout each non-zero region of time ("time period") - that is, time is viewed as a discrete variable.

Unlike a continuous-time signal, a discrete-time signal is not a function of a continuous argument; however, it may have been obtained by sampling from a continuous-time signal. For example, if you were monitoring the temperature of a room, you would be able to take a measured value of temperature at any time. Applications of these signals include speech processing, SONAR, communications, etc.

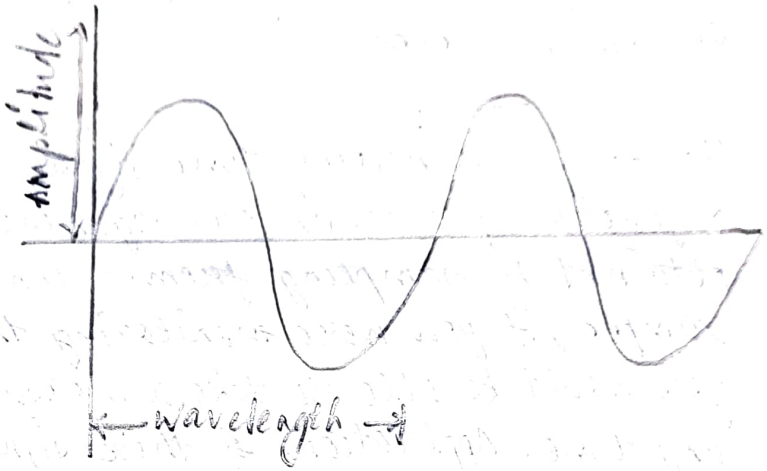


Discrete-Time Signal

(3) Sinusoidal signal - These are periodic functions that are based on the sine or cosine function from trigonometry. When the sine wave starts from zero and covers positive values, reaches zero; and again covers negative values, reaches zero, it is said to have completed one cycle or single cycle.

The continuous-time version of a sinusoidal signal, in its most general form, may be written as $x(t) = A\cos(\omega t + \phi)$ where A is the amplitude, ω is the frequency in radians per second, and ϕ is the angle in radians.

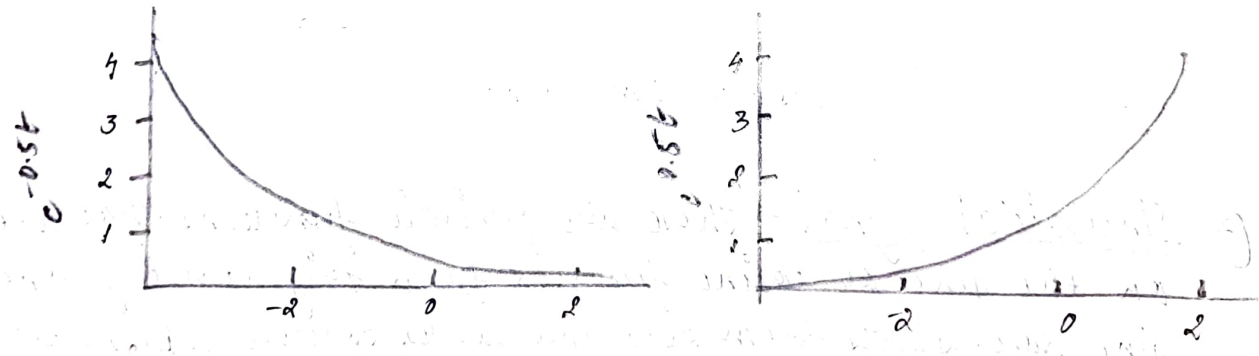
The discrete-time version of a sinusoidal signal, in its most general form, may be written as $x[n] = A\cos(\Omega n + \phi)$, where the angular frequency Ω must be rational multiple of 2π for the signal to be periodic. Here, the period is measured in samples.



(4) REAL EXPONENTIAL SIGNAL - It is defined as,

$$x(t) = Ae^{\sigma t}$$

where both "A" and "σ" are real. Depending on the value of the "σ" the signals will be different. If "σ" is positive, the signal $x(t)$ is growing exponential and if "σ" is negative, then the signal $x(t)$ is a decaying exponential.



(a) Decaying exponential ($\sigma < 0$) (b) Growing Exponential ($\sigma > 0$)

(5) COMPLEX EXPONENTIAL SIGNAL - It is defined as,

$$x(t) = Ae^{st}$$

where "s" is a complex variable and it is defined as,

$$s = \sigma + j\omega$$

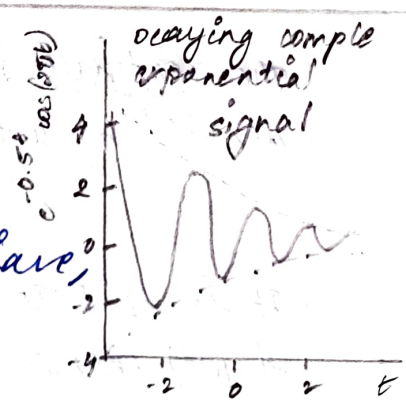
Therefore, $x(t) = e^{st} = e^{(\sigma + j\omega)t} = e^{\sigma t} e^{j\omega t}$ — (1)

Using Euler's identity,

$$e^{j\omega t} = e^{j\theta} = \cos\theta + j\sin\theta \quad \text{— (2)}$$

Substituting equation (2) in equation (1), we have,

$$x(t) = e^{\sigma t}[\cos\theta + j\sin\theta]$$



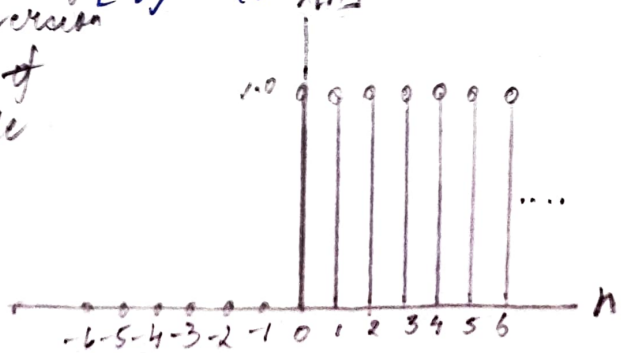
(5) CONTINUOUS-TIME AND DISCRETE-TIME COMPLEX EXPONENTIALS - In both cases, the complex exponential can be expressed through Euler's relation in the form of a real and an imaginary part, both of which are sinusoidal with a phase difference of $\pi/2$ and with an envelope that is a real exponential. When the magnitude of the complex exponential is a constant, then the real and imaginary parts neither grow nor decay with time; in other words, they are purely sinusoidal. In this case for continuous-time, the complex exponential is periodic. For discrete-time, the complex exponential may or may not be periodic depending on whether the sinusoidal real and imaginary components are periodic.

(7) UNIT STEP FUNCTION - It is the value of which is zero for negative arguments and one for positive arguments. It is an example of the general class of step functions, all of which can be represented as linear combinations of translation of this one.

The discrete-time version of the step function, commonly denoted by $u[n]$, is defined by -

$$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

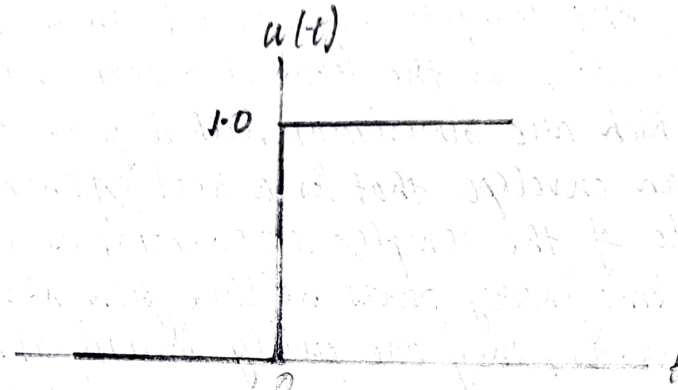
Discrete-time version of step function of unit amplitude



The continuous-time version of the step function, commonly denoted by $u(t)$, is defined by -

$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

It is said to exhibit a discontinuity at $t=0$, since the value of $u(t)$ changes instantaneously from 0 to 1 when $t=0$.

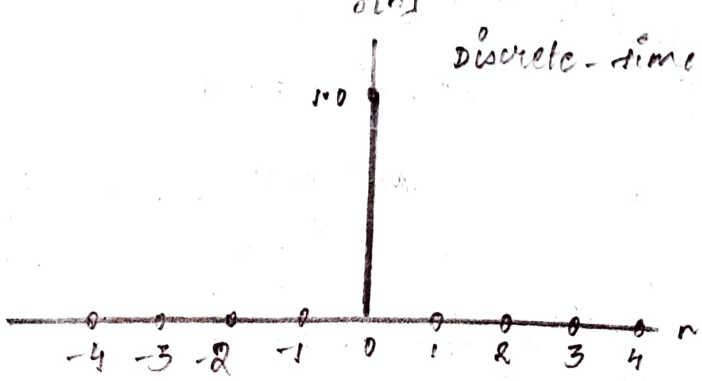


continuous-time version of step function of unit impulse

⑧ UNIT IMPULSE FUNCTION - It is a generalized function or distribution over the real numbers, whose value is zero everywhere except at zero, and whose integral over the entire real line is equal to one. It is used as a tool for the normalization of state vectors and also, has uses in probability theory and signal processing.

The discrete-time version of the impulse, commonly denoted by $\delta[n]$, is defined by -

$$\delta[n] = \begin{cases} 1, & n=0 \\ 0, & n \neq 0 \end{cases}$$

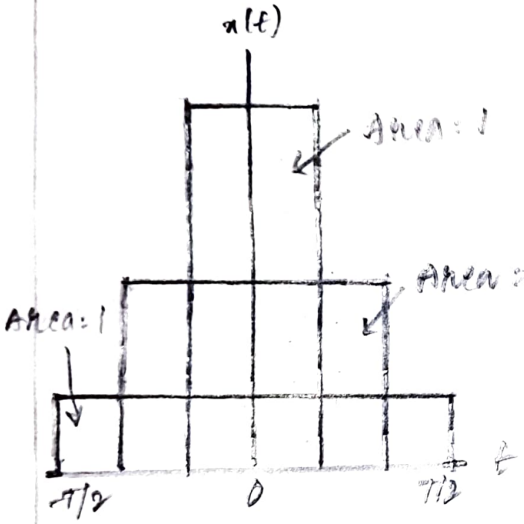


The continuous-time version of the unit impulse, commonly denoted by $\delta(t)$, is defined by the following pair of relations –

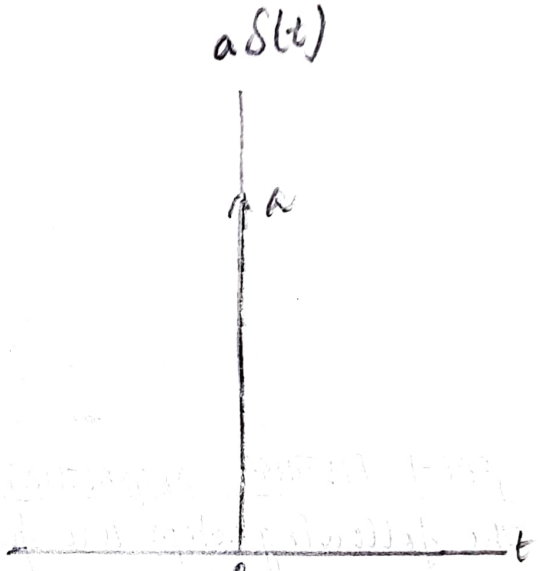
$$\delta(t) = 0 \quad \text{for } t \neq 0$$

and

$$\int_{-\infty}^{\infty} \delta(t)\, dt = 1$$



Evolution of a rectangular pulse of unit area into an impulse of unit strength

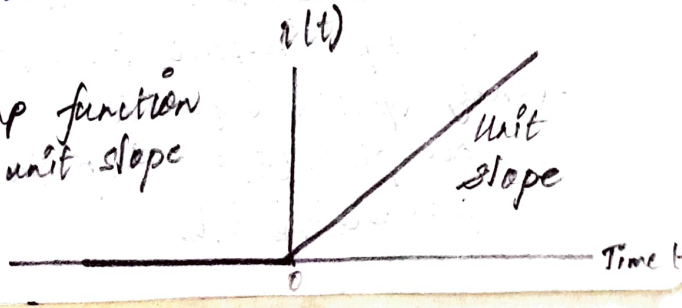


Graphical symbol for an impulse of strength a

(5) RAMP FUNCTION - It is a unary real function, whose graph is shaped like a ramp. It can be expressed by numerous definitions for example, "0 for negative inputs, output equals input for non-negative inputs." The term "ramp" can also be used for other functions obtained by scaling and shifting, and the function is is the unit ramp function (slope 1, starting at 0).

The continuous-time version of the ramp function, commonly denoted by $r(t)$, is defined by –

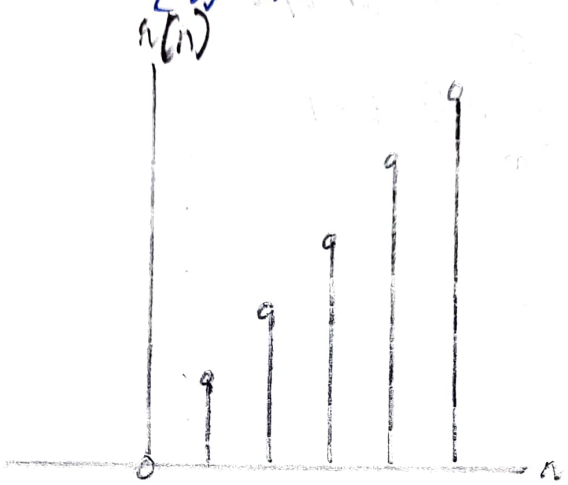
$$r(t) = \begin{cases} t, & t \geq 0 \\ 0, & t < 0 \end{cases}$$



Ramp function of unit slope

The discrete-time version of the impulse, commonly denoted by $r[n]$, is defined by-

$$r[n] = \begin{cases} n, & n > 0 \\ 0, & n < 0 \end{cases}$$



Discrete-time version of the ramp function

+ **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.

+ **PROGRAM WITH COMMENTS:**

```
import numpy as np
import matplotlib.pyplot as plt

def sinusoidal_c(): # Sinusoidal continuous-Time Signal
    t = np.linspace(-0.02, 0.05, 1000) # linspace() creates
evenly spaced sequences
    plt.plot(t, 325*np.sin(2*np.pi*50*t))
    plt.xlabel('t')
```

```
    plt.ylabel ['x(t)']
    plt.title ['plot of CT signal : x(t) = 325*sin (100*pi*t)']
    plt.xlim [[-0.02, 0.05]) # set the x-limits of the current axes
    plt.show() # To view your plot

def sinusoidal_d(): # Sinusoidal Discrete-Time Signal
    n = np.arange (50) # Get evenly spaced values within a given interval
    dt = 0.07/50 # Fixes Signal Space Resolution
    x = 325 * np.sin (2* np.pi * 50 * n * dt)
    plt.xlabel ['n']
    plt.ylabel ['x[n]']
    plt.title ['plot of DT signal : x[n]= 325* sin (100*pi*n)']
    plt.stem (n, x) # stem plot plots vertical lines at each x position covered under the graph from the baseline to y. and places a marker there.
    plt.show() # To view your plot.

def exponential_c(): # Exponential Continuous-Time Signal
    t = np.linspace [0.02, 0.05, 1000] # linspace() creates evenly spaced sequencies
    plt.subplot [3,1,1) # subplot (rows, columns, index) describes the figure layout
    plt.plot (t, np.exp [2j * np.pi * 50 * t).real) # Return the real part of the complex argument
    plt.xlabel ['t']
    plt.ylabel ['Re x(t)']
    plt.title ['Real part of x(t) = exp (j*100*pi*t)']
    plt.xlim [[-0.02, 0.05]) # set the x-limits of the current axes
    plt.subplot (3, 1, 3) # subplot (rows, columns, index) describes the figure layout
    plt.plot (t, np.exp (2j * np.pi * 50 * t).imag) # Return the imaginary part of the complex argument
```

```
    plt.xlabel('t')
    plt.ylabel('Im x(t)')
    plt.title('Imaginary part of x(t) = exp(j*100*pi*t)')
    plt.xlim([-0.02, 0.05]) # Set the x-limits of the current axes.
    plt.show() # To view your plot

def exponential_d(): # Exponential Discrete-Time Signal
    n = np.arange(-2, 5) # Get evenly spaced values within a given interval
    arr=[] # Null Array Declaration
    for sample in range(len(n)): # Returns the length of the array
        if n[sample] < 0:
            temp = 1/(2**(-(n[sample]))) # conversion of negative power to fractions (2^-1 = 1/2)
        else:
            temp = (2**(n[sample]))
        arr.append(temp) # Adds a single item to the existing list
    plt.xlabel('n')
    plt.ylabel('x[n]')
    plt.title('Plot of DT signal : x[n] = 2^n')
    plt.stem(n, arr) # Stem plot plots vertical lines at each x position covered under the graph from the baseline to y, and places a marker there.
    plt.show() # To view your plot

def step_c(): # Unit Step Continuous-Time Signal
    t = np.linspace(-2, 10, 1000) # linspace() creates evenly spaced sequences
    arr = [] # Null Array Declaration
    for sample in range(len(t)): # Returns the length of array
        if t[sample] >= 0:
            temp = 1
```

```
        else
            temp=0
        arr.append(temp) # Adds a single item to the existing list
    plt.xlabel('t')
    plt.ylabel('x(t)')
    plt.title('Unit Step Continuous-Time Function')
    plt.step(t,arr) # Line forming a series of steps between data points
    plt.show() # To view your plot

def step_d(): # Unit Step Discrete-Time Signal
    n=np.arrange[-2,10) # Get evenly spaced values within a given interval
    arr=[] # Null Array Declaration
    for sample in range(len(n)): # Returns the length of array
        if n[sample]>=0:
            temp=1
        else
            temp=0
        arr.append(temp) # Adds a single item to the existing list
    plt.xlabel('n')
    plt.ylabel('x(n)')
    plt.title('Unit Step Discrete-Time Function for 0≤n≤9')
    plt.stem(n, arr) # stem plot plots vertical lines at each x position under the graph from the baseline to y, and places a marker there.
    plt.show() # To view your plot

def impulse_c(): # Unit Impulse Continuous-Time Signal
    t=np.arrange(-2,10) # Get evenly spaced values within a given interval
    arr=[] # Null Array Declaration
```

```
for sample in range (len(t)): # Returns the length of array
    if t[sample] >=0 :
      temp =1
    else
      temp =0
    arr. append (temp) # Adds a single item to the existing list
  plt. xlabel ('t')
  plt. ylabel ('x(t)')
  plt. title ('Unit Impulse Continuous-Time Function')
  plt. plot (t, arr) # Line forming a series of steps within a
given interval
  plt show () To view your plot

def impulse_d() : # Unit Impulse Discrete-Time signal
    n= np. arrange (-2, 10) # Get evenly spaced values within a
given interval
    arr = [] # Null Array Declaration
    for sample in range (len(n)) : # Returns the length of array
      if n[sample] >= 0:
        temp =1
      else
        temp =0
      arr. append (temp) # Adds a single item to the existing l
    plt. xlabel ('n')
    plt. ylabel ('x(n)')
    plt. title ('Unit Impulse Discrete-Time Function for 0≤n≤9')
    plt. stem (n, arr) # stem plot plots vertical lines at each
position covered under the graph from the baseline to y. and
places a marker there.
    plt. show () # To view your plot
```

```
def ramp_c(): # Ramp Function Continuous-Time Signal
    t = np.linspace(0, 10, 1000) # linspace() creates evenly spaced sequences
    plt.plot(t, t)
    plt.xlabel('t')
    plt.ylabel('x(t)')
    plt.title('Ramp Continuous-Time Function')
    plt.xlim([0, 10]) # Set the x-limits of the current axes
    plt.show() # To view your plot

def ramp_d(): # Ramp Function Discrete-Time signal
    n = np.arange(0, 10) # linspace() creates evenly spaced sequences
    plt.xlabel('n')
    plt.ylabel('n(t)')
    plt.title('Ramp Discrete-Time Function for 0≤n≤9')
    plt.stem(n, n) # stem plot plots vertical lines at each x position covered under the graph from the baseline to y, and places a marker there
    plt.show() # To view your plot

# Main
while True: # This simulates a Do Loop
    choice = input(
        "MENU: \n 1. Sinusoidal Continuous-Time Signal. \n 2. Sinusoidal Discrete-Time Signal. \n 3. Exponential Continuous-Time Signal. \n 4. Exponential Discrete-Time signal. \n 5. Unit step Continuous-Time Signal \n 6. Unit Step Discrete-Step Signal. \n 7. Unit Impulse Continuous-Time Signal. \n 8. Unit Impulse Discrete-Time Signal. \n 9. Ramp Function Continuous-Time Signal. \n 10. Ramp Function Discrete-Time signal. \n 11. Exit \n Enter the number corresponding to the menu to implement the choice: ") # Menu Driven Implementation
```

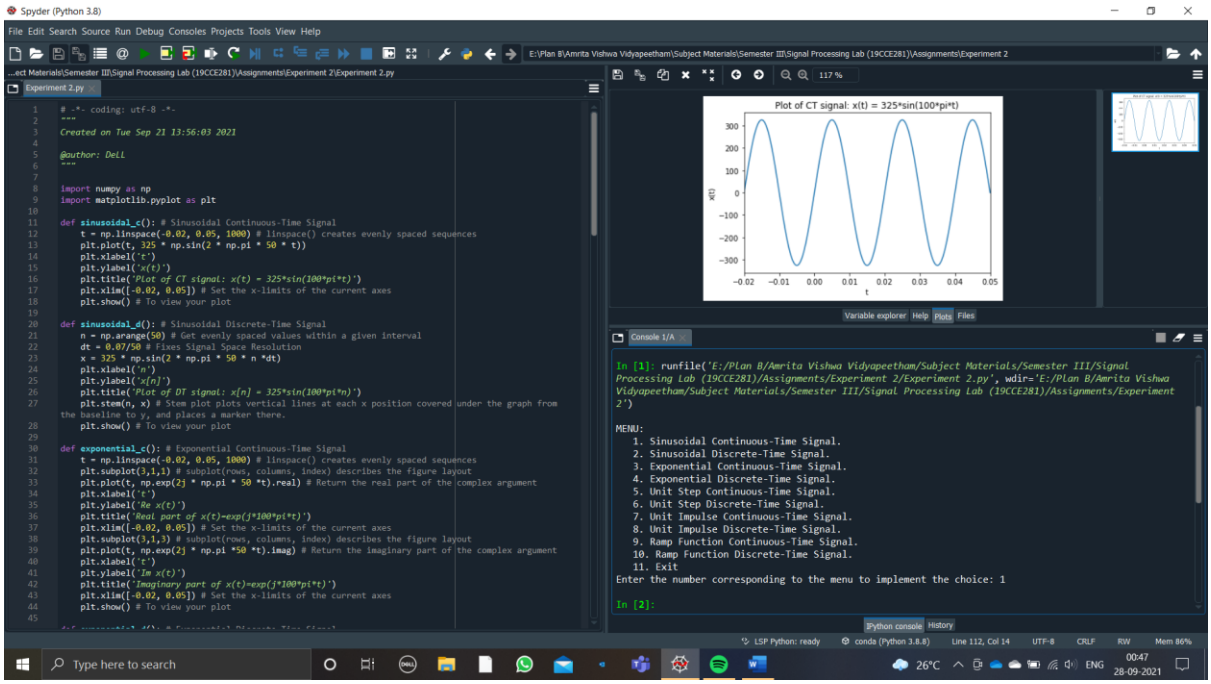
```
if choice == str(1): # str() returns the string version of
the variable "choice"
    sinusoidal_c() # Sinusoidal Continuous-Time Signal
    break
elif choice == str(2):
    sinusoidal_d() # Sinusoidal Discrete-Time signal
    break
elif choice == str(3):
    exponential_c() # Exponential Continuous-Time signal
    break
elif choice == str(4):
    exponential_d() # Exponential Discrete-Time Signal
    break
elif choice == str(5):
    step_c() # Unit Step Continuous-Time Signal
    break
elif choice == str(6):
    step_d() # Unit step Discrete-Time Signal
    break
elif choice == str(7):
    step impulse_c() # Unit step Impulse Continuous-Time Signal
    break
elif choice == str(8):
    impulse_d() # Unit Impulse Discrete-Time Signal
    break
elif choice == str(9):
    ramp_c() # Ramp Function Continuous-Time Signal
    break
elif choice == str(10):
    ramp_d() # Ramp Function Discrete-Time signal
    break
elif choice == str(11):
    break # Exit Loop
```

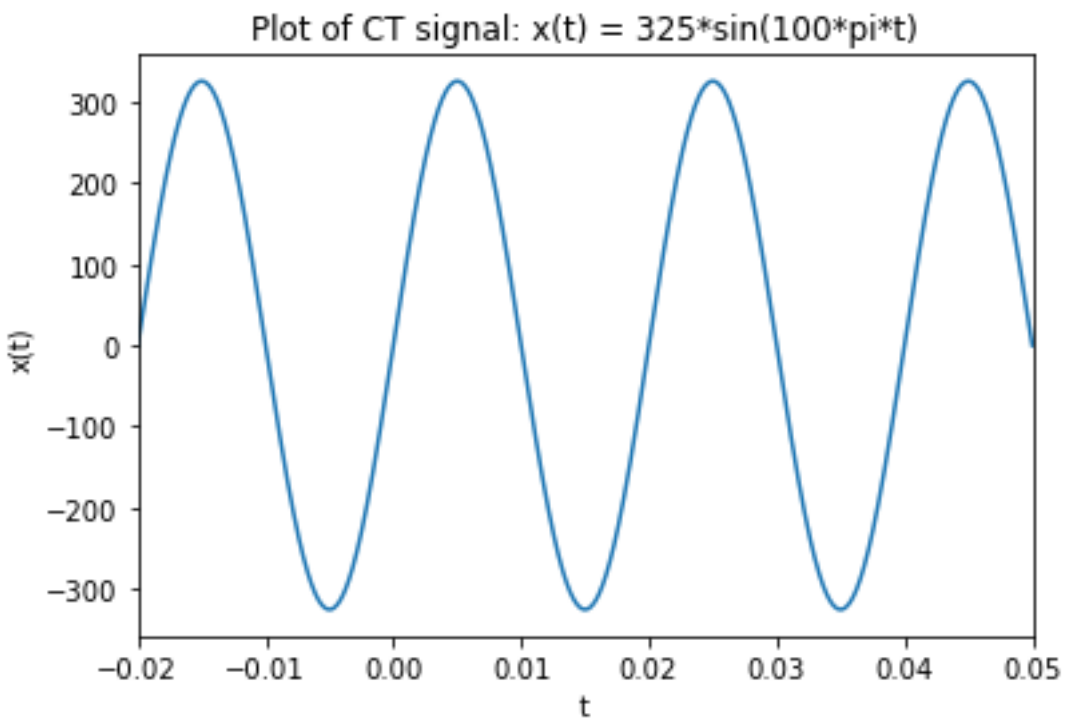
```
else :
    print ("Error : Invalid Input! Please try again. \n")
```

+ INFERENCE :

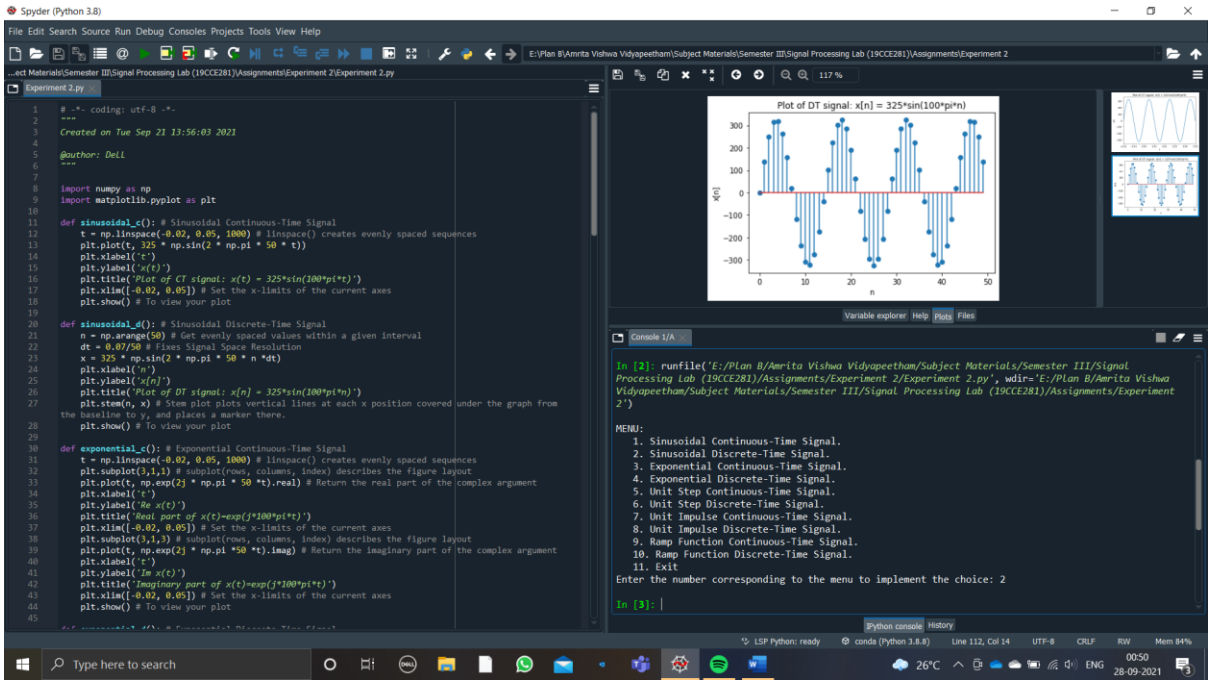
Visualize the complex exponential signal and real sinusoids and demonstrate other simple elementary signals and results verified.

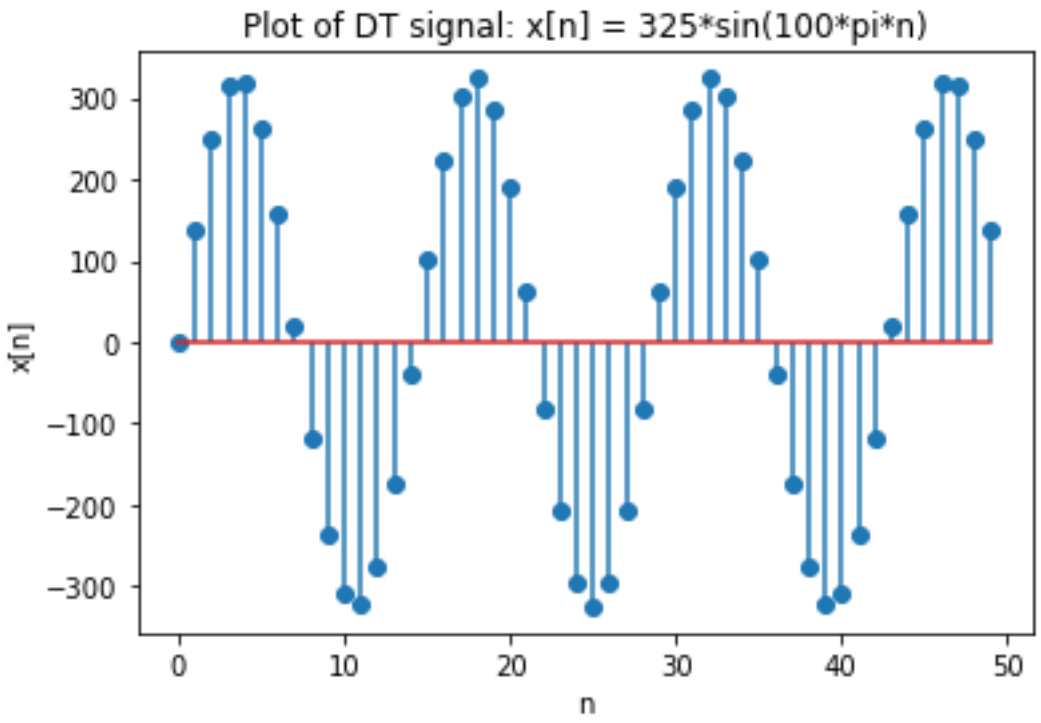
Sinusoidal Continuous-Time Signal



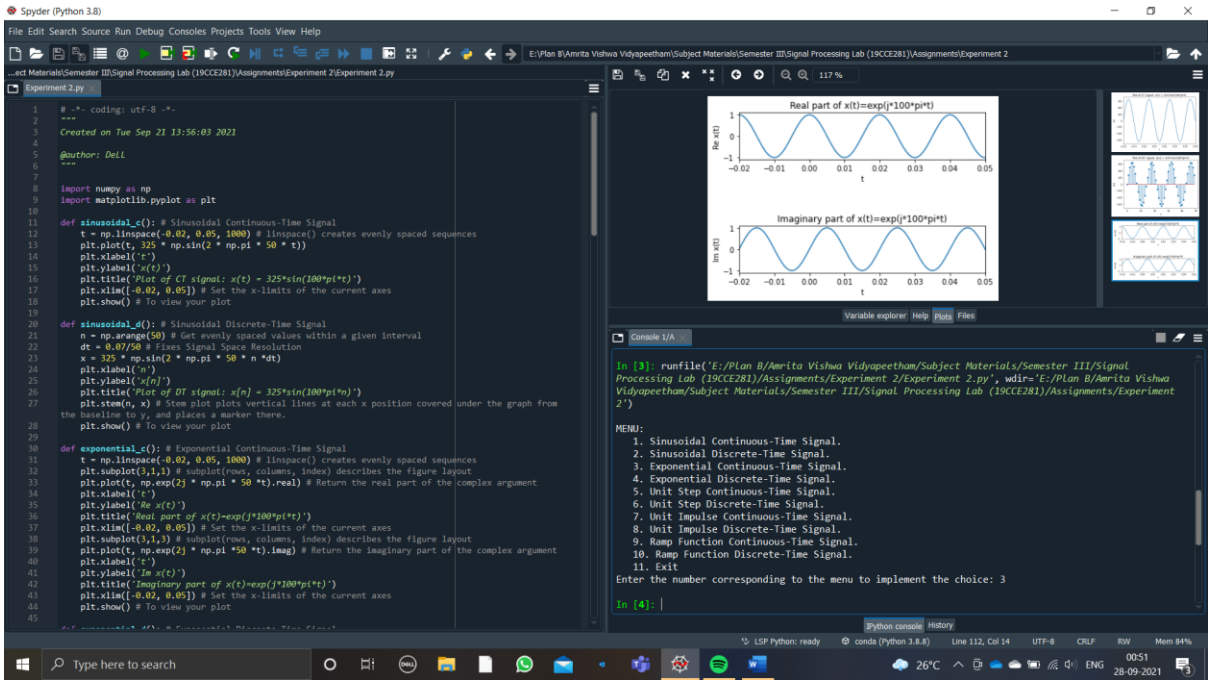


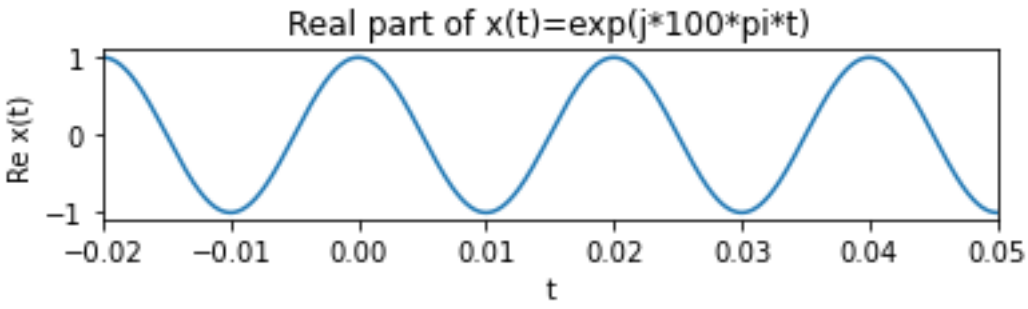
Sinusoidal Discrete-Time Signal

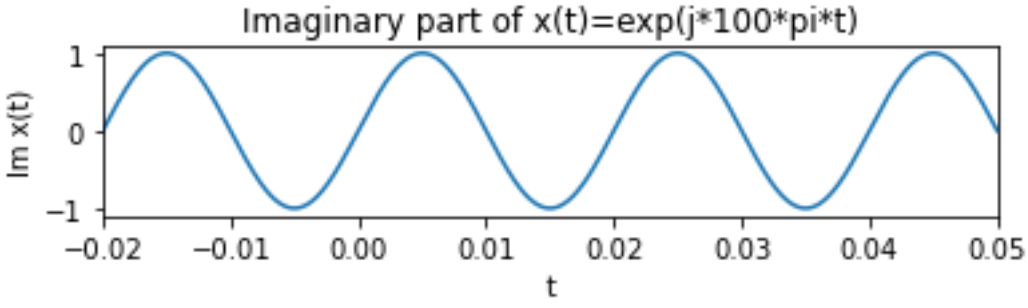




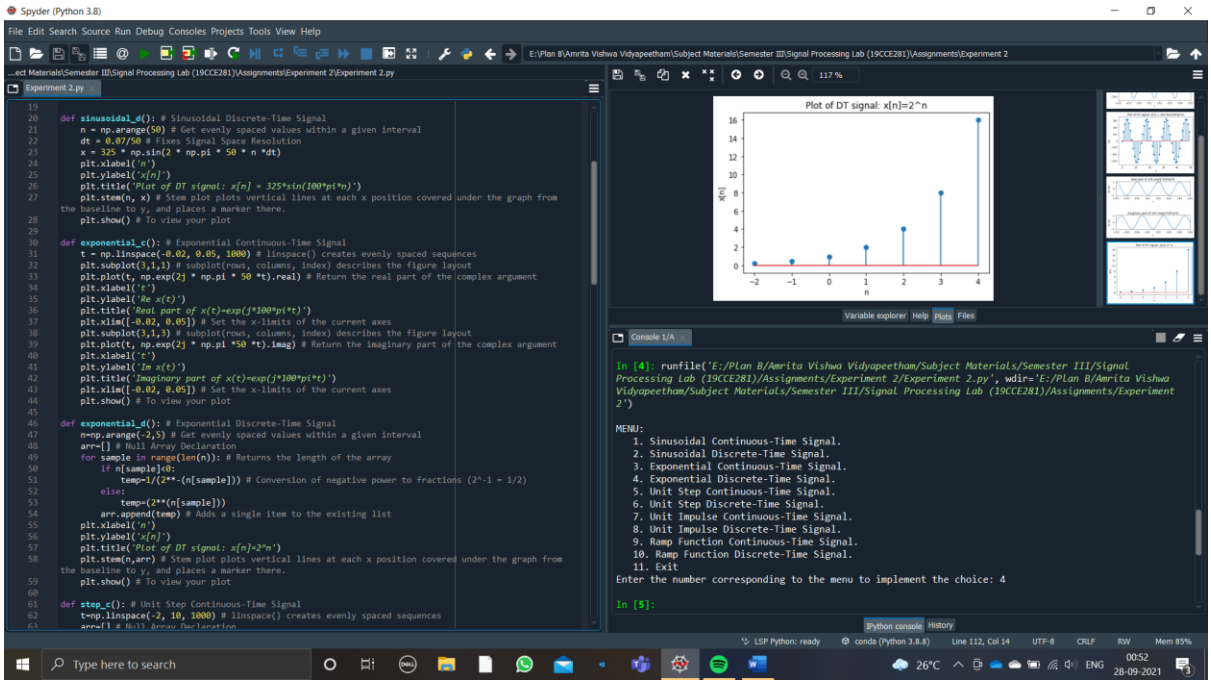
Exponential Continuous-Time Signal

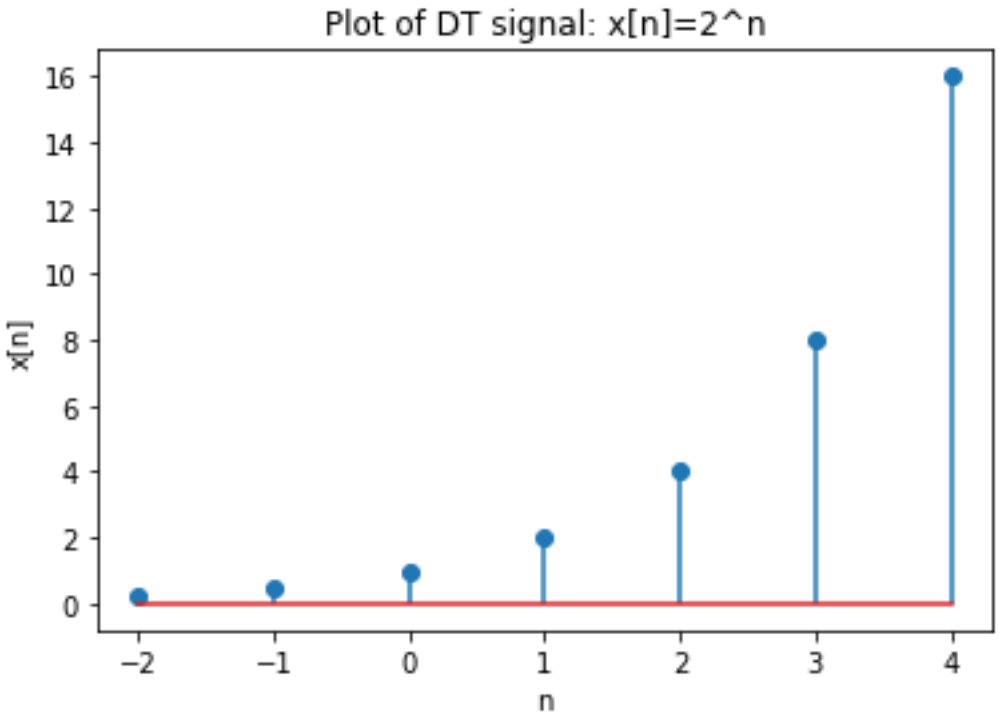




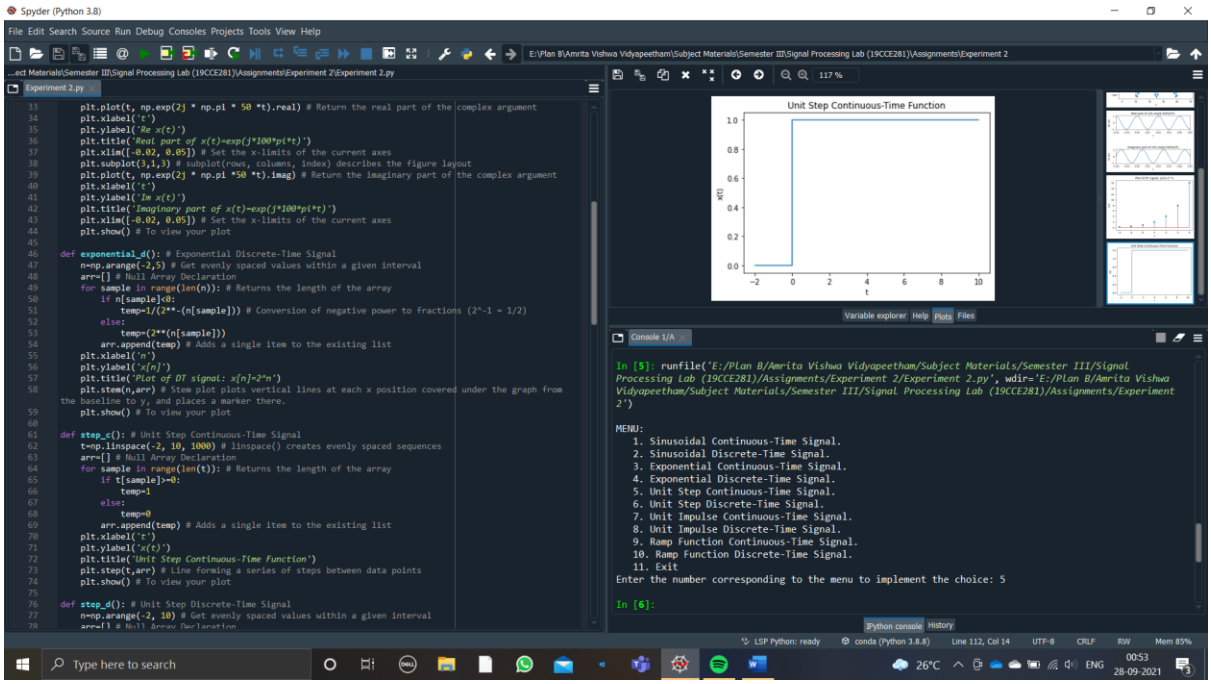


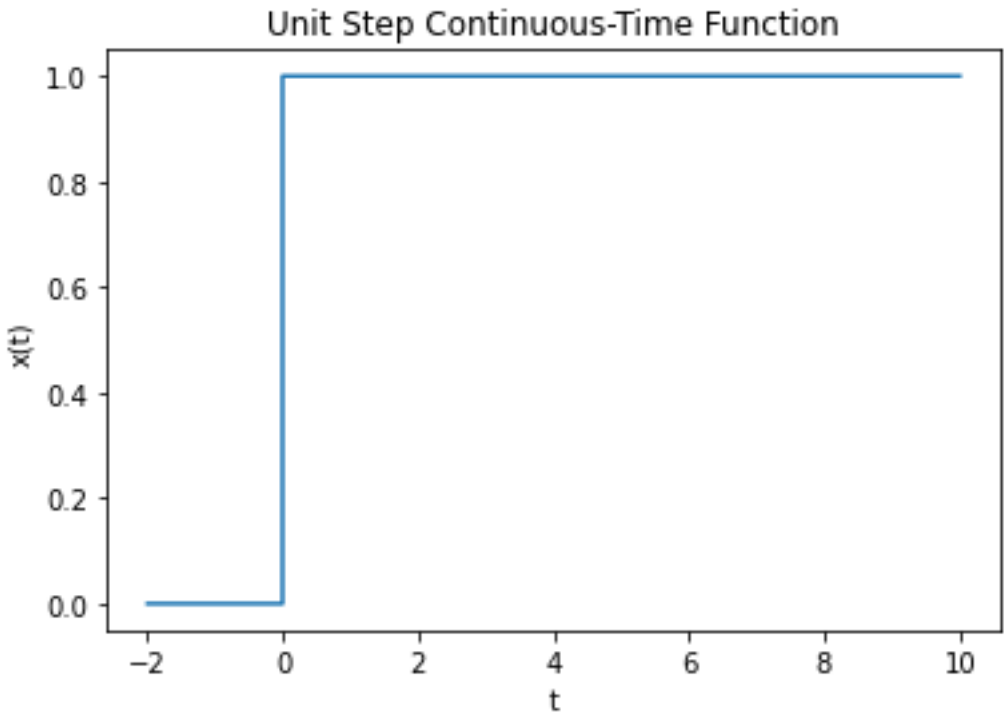
Exponential Discrete-Time Signal



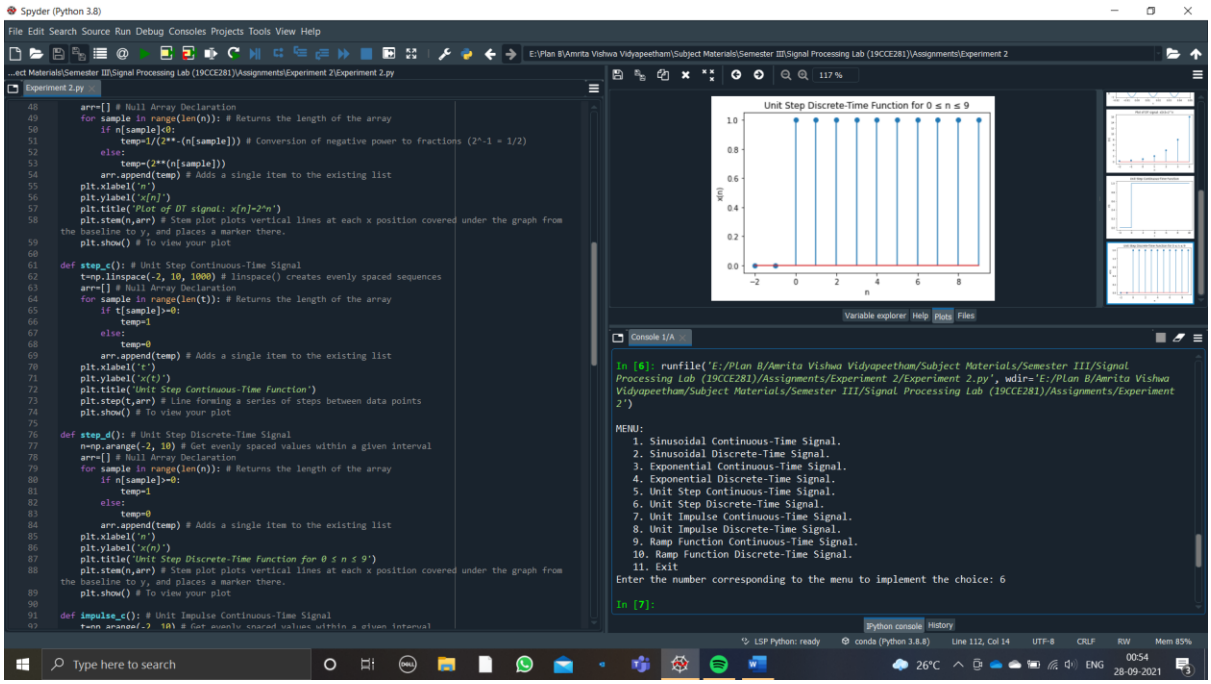


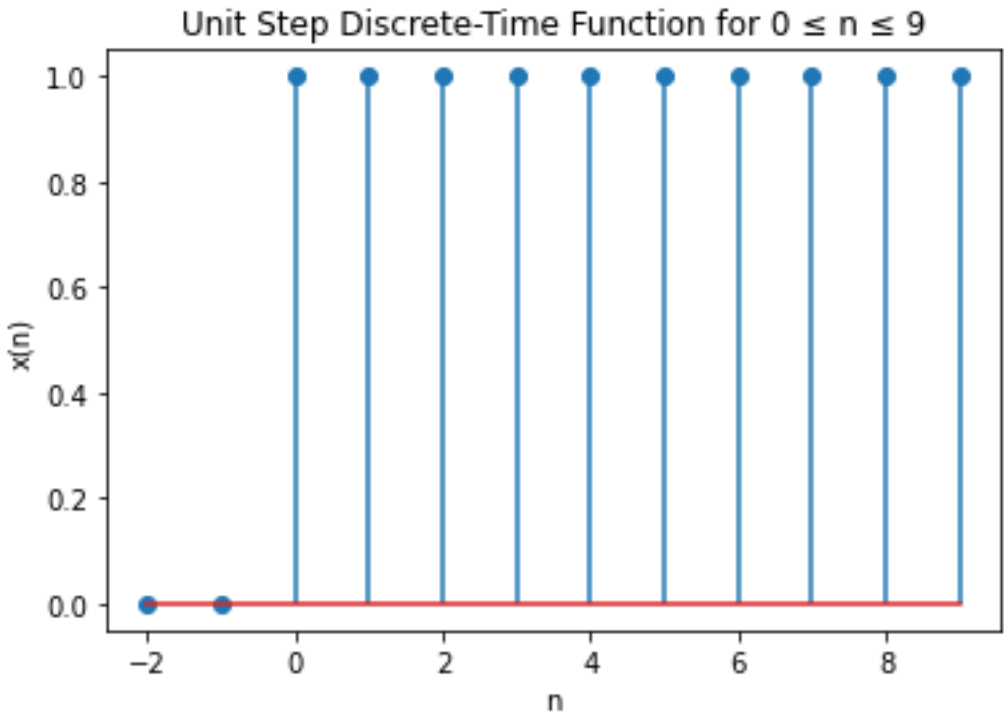
Unit Step Continuous-Time Signal



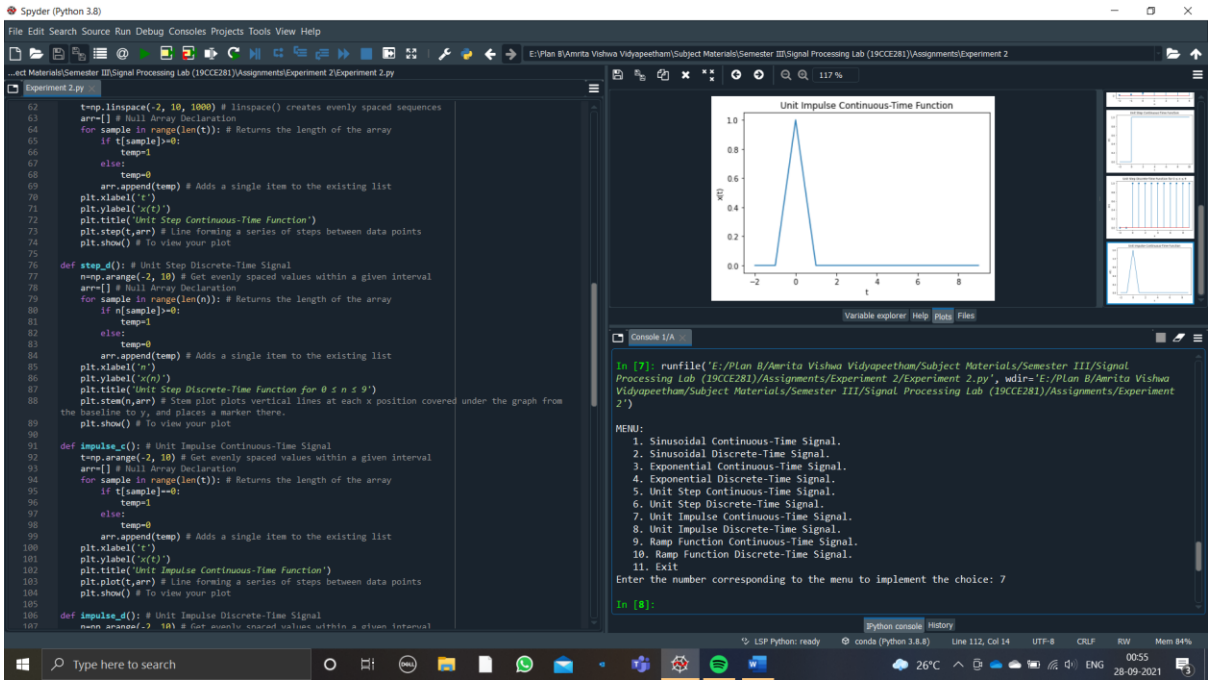


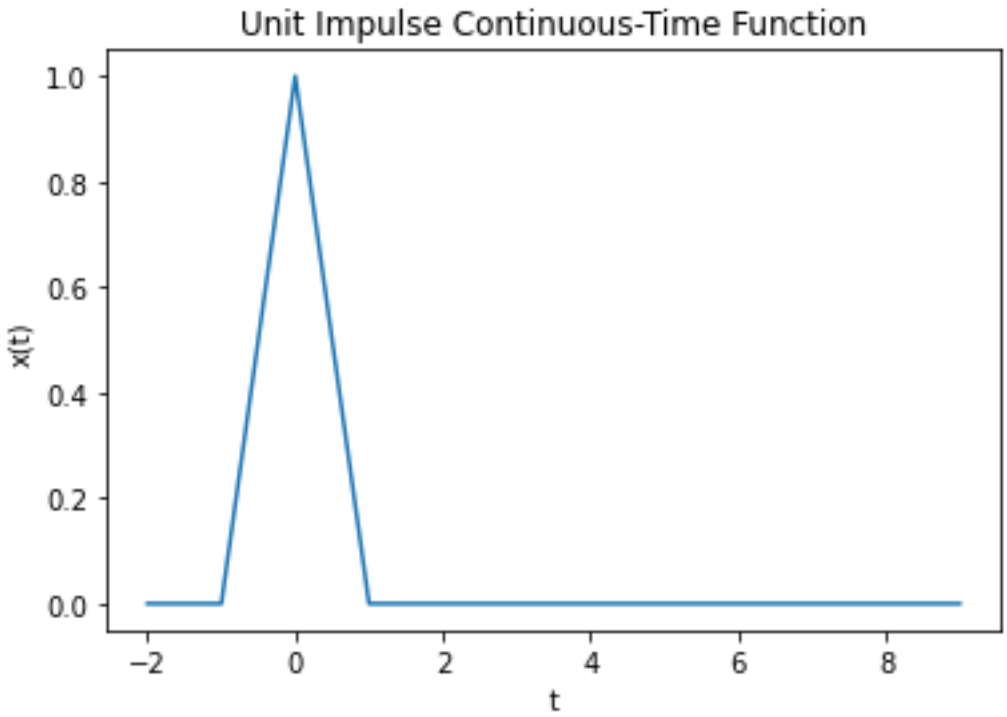
Unit Step Discrete-Time Signal



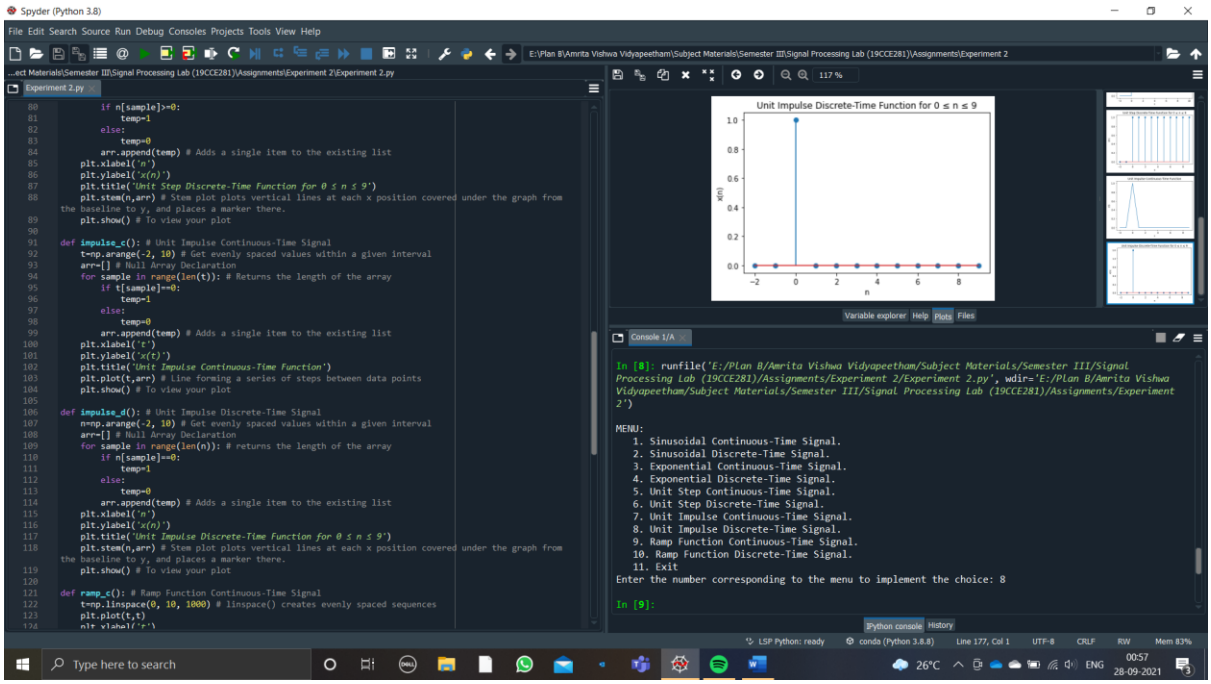


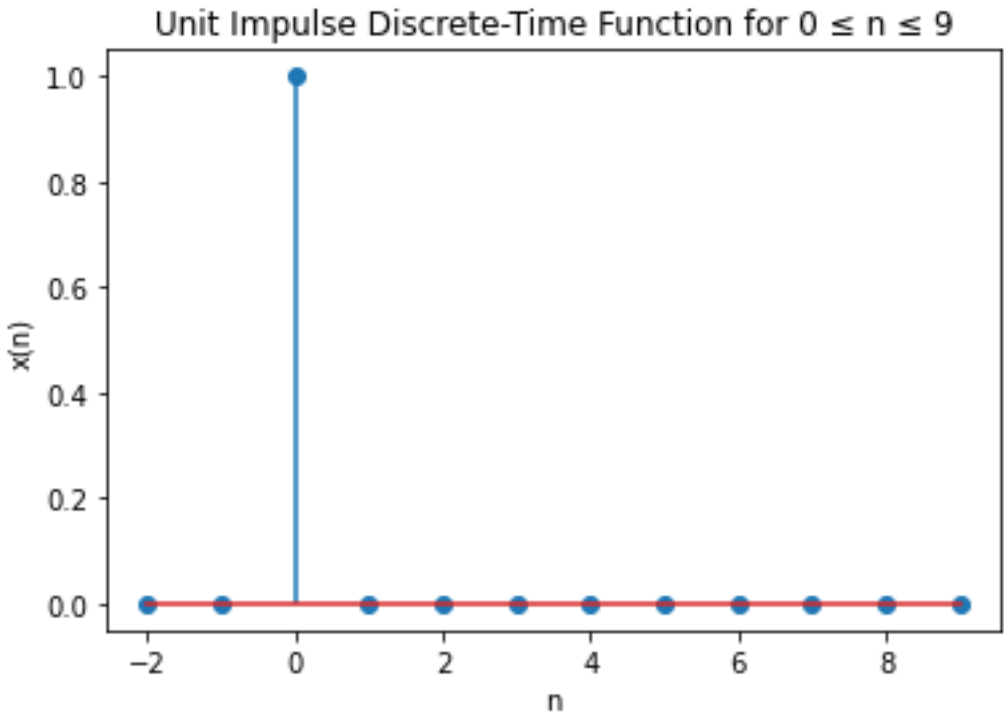
Unit Impulse Continuous-Time Signal



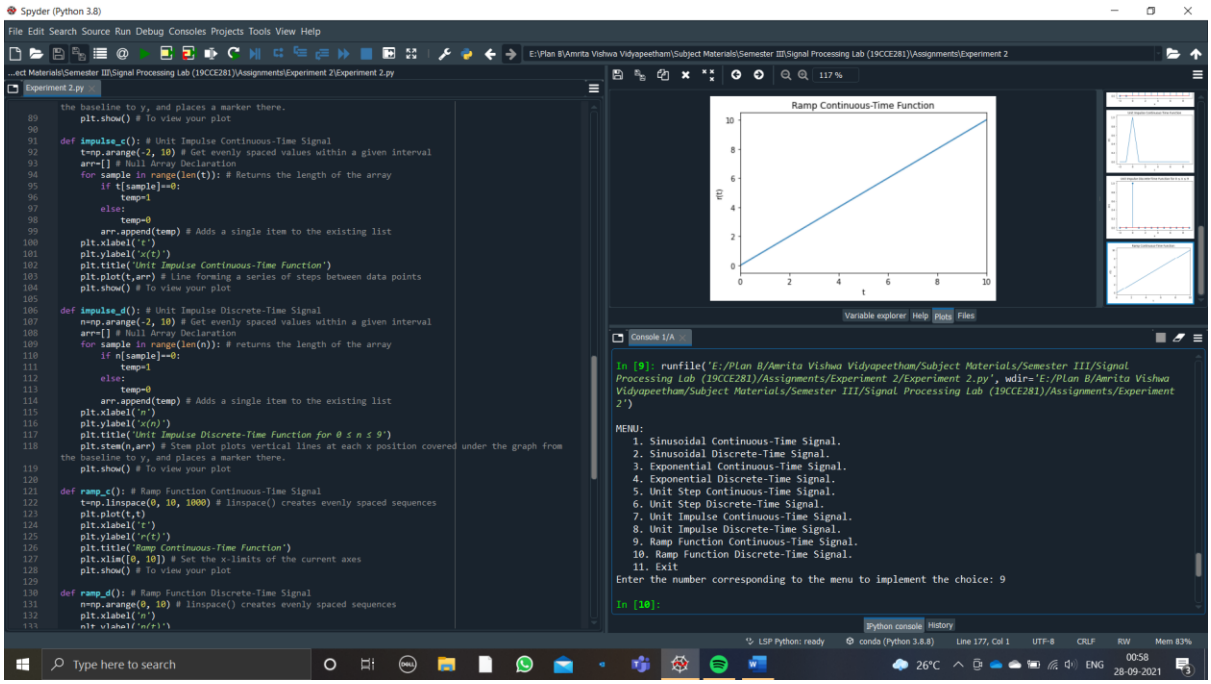


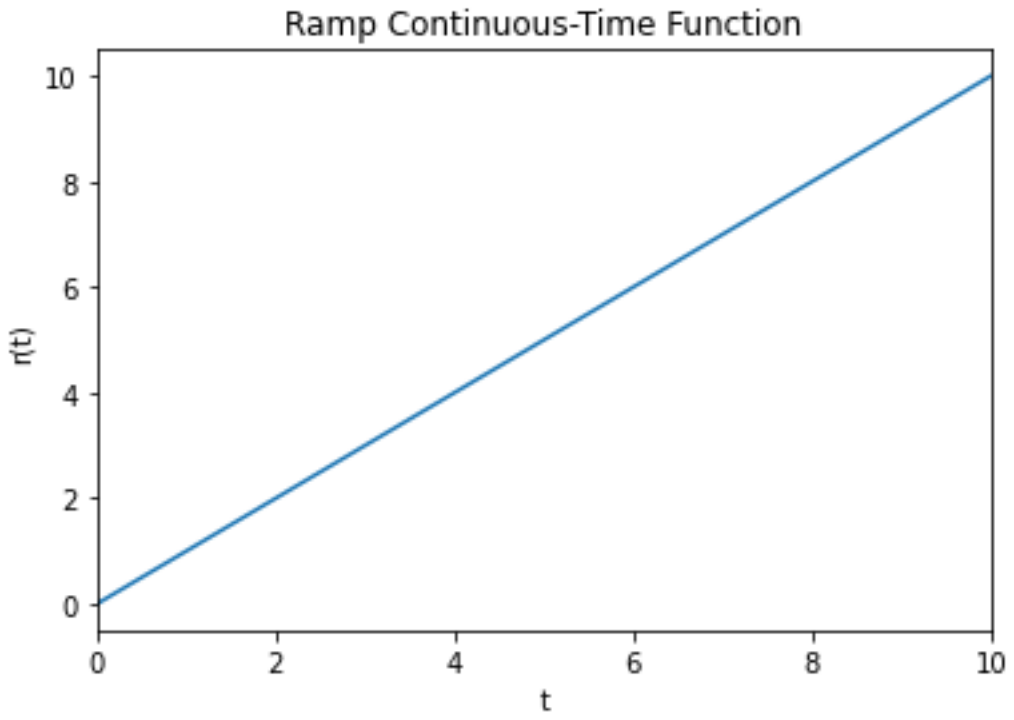
Unit Impulse Discrete-Time Signal



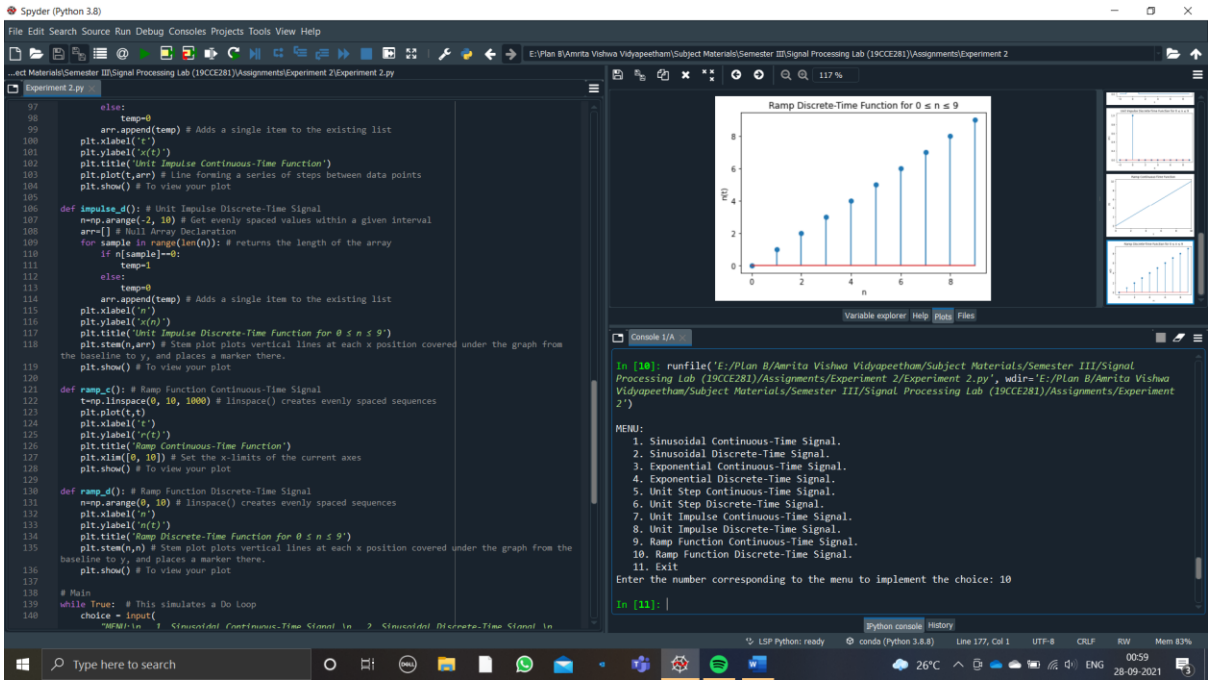


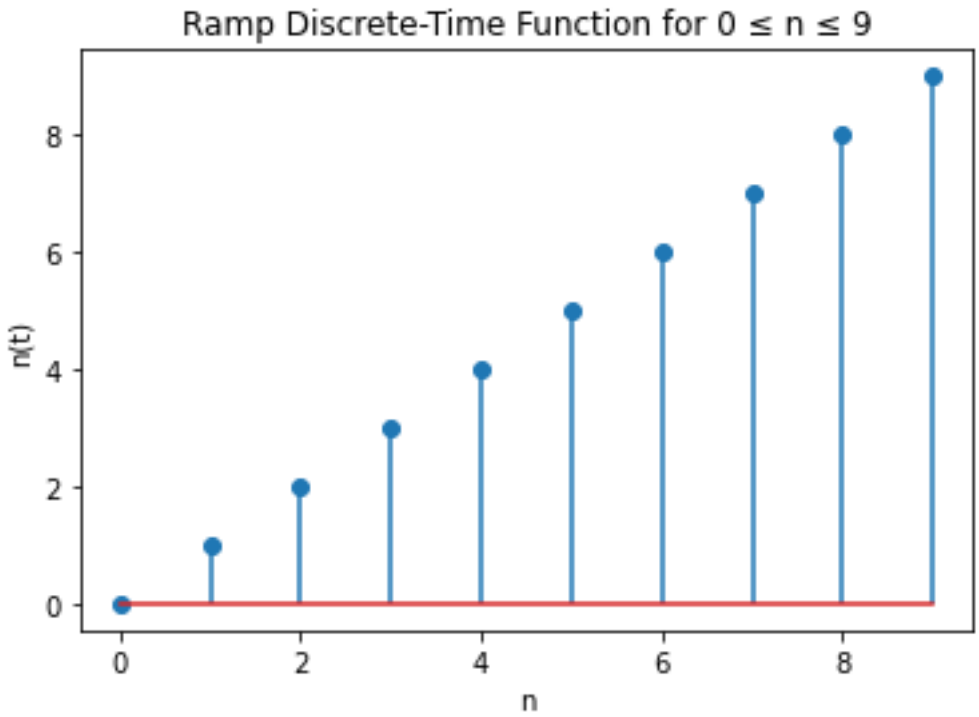
Ramp Function Continuous-Time Signal



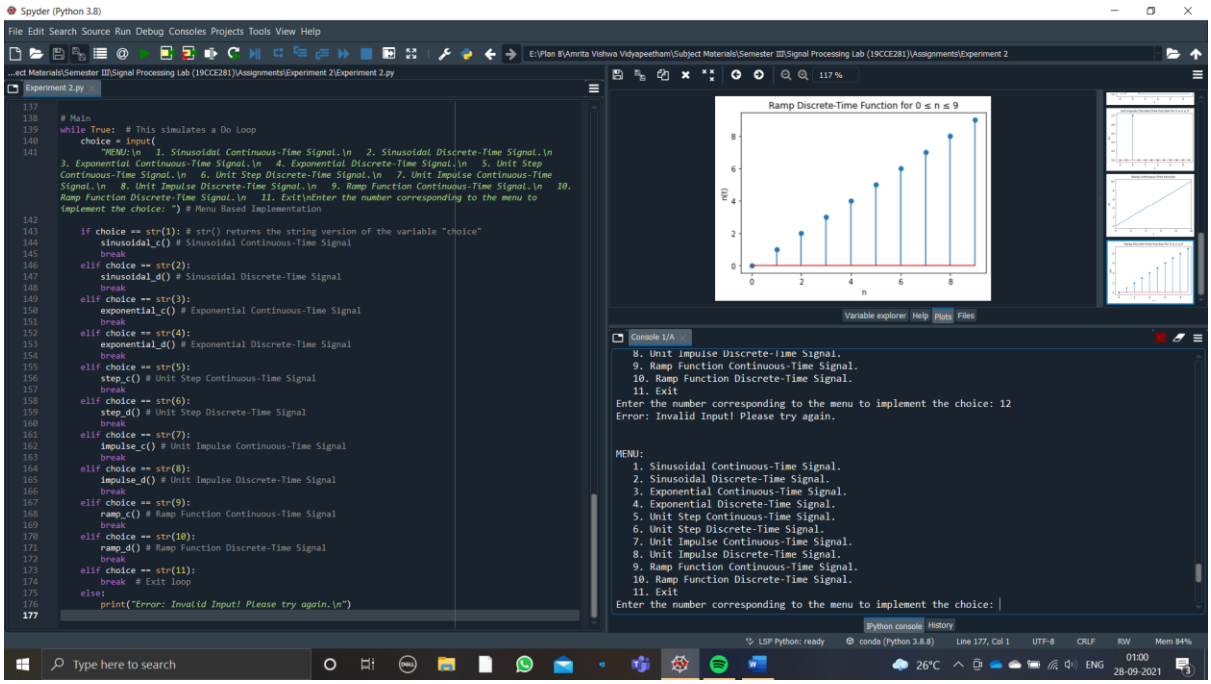


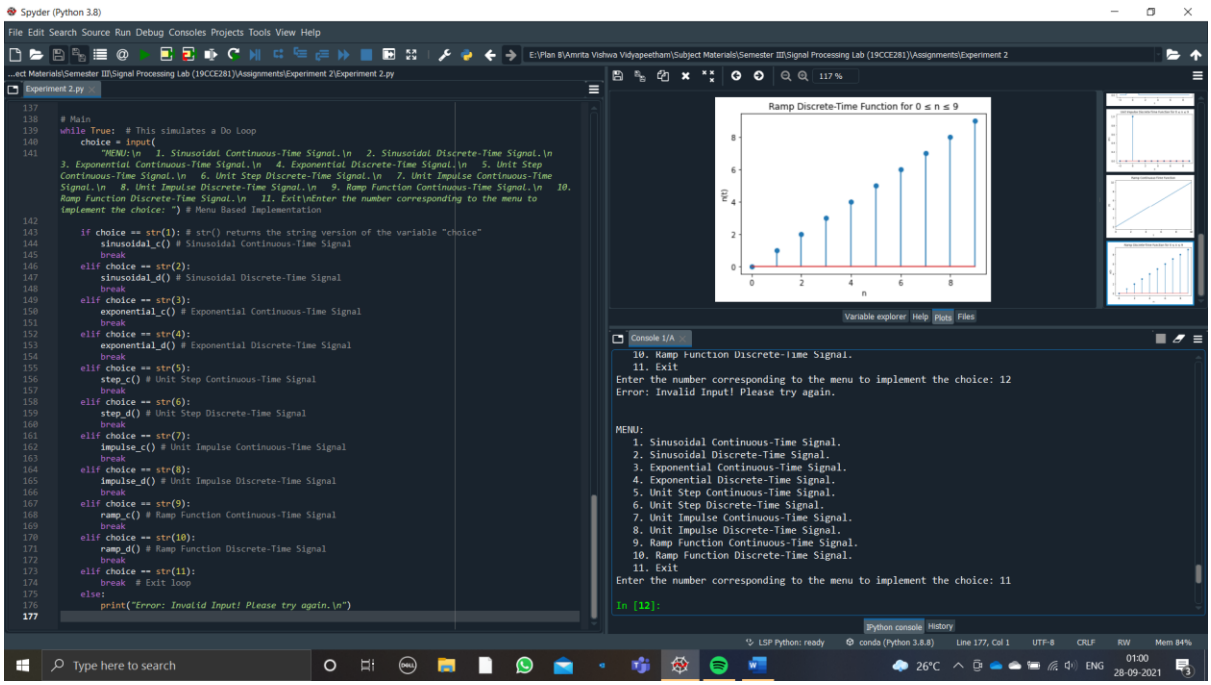
Ramp Function Discrete-Time Signal





Exit





Thank You!

```
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 2/Experiment 2.py'      ='E:/Plan
B/Amrita Vishwa Vidyapeetham/Subject Materials/Semester III/Signal Processing Lab
(19CCE281)/Assignments/Experiment 2'

MENU:
   1. Sinusoidal Continuous-Time Signal.
   2. Sinusoidal Discrete-Time Signal.
   3. Exponential Continuous-Time Signal.
   4. Exponential Discrete-Time Signal.
   5. Unit Step Continuous-Time Signal.
   6. Unit Step Discrete-Time Signal.
   7. Unit Impulse Continuous-Time Signal.
   8. Unit Impulse Discrete-Time Signal.
   9. Ramp Function Continuous-Time Signal.
   10. Ramp Function Discrete-Time Signal.
   11. Exit
Enter the number corresponding to the menu to implement the choice: 1

MENU:
   1. Sinusoidal Continuous-Time Signal.
   2. Sinusoidal Discrete-Time Signal.
   3. Exponential Continuous-Time Signal.
   4. Exponential Discrete-Time Signal.
   5. Unit Step Continuous-Time Signal.
   6. Unit Step Discrete-Time Signal.
   7. Unit Impulse Continuous-Time Signal.
   8. Unit Impulse Discrete-Time Signal.
   9. Ramp Function Continuous-Time Signal.
   10. Ramp Function Discrete-Time Signal.
   11. Exit
Enter the number corresponding to the menu to implement the choice: 2

MENU:
   1. Sinusoidal Continuous-Time Signal.
   2. Sinusoidal Discrete-Time Signal.
   3. Exponential Continuous-Time Signal.
   4. Exponential Discrete-Time Signal.
   5. Unit Step Continuous-Time Signal.
   6. Unit Step Discrete-Time Signal.
   7. Unit Impulse Continuous-Time Signal.
   8. Unit Impulse Discrete-Time Signal.
   9. Ramp Function Continuous-Time Signal.
   10. Ramp Function Discrete-Time Signal.
   11. Exit
Enter the number corresponding to the menu to implement the choice: 3
```

```
MENU:
   1. Sinusoidal Continuous-Time Signal.
   2. Sinusoidal Discrete-Time Signal.
   3. Exponential Continuous-Time Signal.
   4. Exponential Discrete-Time Signal.
   5. Unit Step Continuous-Time Signal.
   6. Unit Step Discrete-Time Signal.
   7. Unit Impulse Continuous-Time Signal.
   8. Unit Impulse Discrete-Time Signal.
   9. Ramp Function Continuous-Time Signal.
   10. Ramp Function Discrete-Time Signal.
   11. Exit
Enter the number corresponding to the menu to implement the choice: 4

MENU:
   1. Sinusoidal Continuous-Time Signal.
   2. Sinusoidal Discrete-Time Signal.
   3. Exponential Continuous-Time Signal.
   4. Exponential Discrete-Time Signal.
   5. Unit Step Continuous-Time Signal.
   6. Unit Step Discrete-Time Signal.
   7. Unit Impulse Continuous-Time Signal.
   8. Unit Impulse Discrete-Time Signal.
   9. Ramp Function Continuous-Time Signal.
   10. Ramp Function Discrete-Time Signal.
   11. Exit
Enter the number corresponding to the menu to implement the choice: 5

MENU:
   1. Sinusoidal Continuous-Time Signal.
   2. Sinusoidal Discrete-Time Signal.
   3. Exponential Continuous-Time Signal.
   4. Exponential Discrete-Time Signal.
   5. Unit Step Continuous-Time Signal.
   6. Unit Step Discrete-Time Signal.
   7. Unit Impulse Continuous-Time Signal.
   8. Unit Impulse Discrete-Time Signal.
   9. Ramp Function Continuous-Time Signal.
   10. Ramp Function Discrete-Time Signal.
   11. Exit
Enter the number corresponding to the menu to implement the choice: 6

MENU:
   1. Sinusoidal Continuous-Time Signal.
   2. Sinusoidal Discrete-Time Signal.
   3. Exponential Continuous-Time Signal.
   4. Exponential Discrete-Time Signal.
   5. Unit Step Continuous-Time Signal.
   6. Unit Step Discrete-Time Signal.
   7. Unit Impulse Continuous-Time Signal.
   8. Unit Impulse Discrete-Time Signal.
   9. Ramp Function Continuous-Time Signal.
   10. Ramp Function Discrete-Time Signal.
   11. Exit
```

```
Enter the number corresponding to the menu to implement the choice: 7

MENU:
   1. Sinusoidal Continuous-Time Signal.
   2. Sinusoidal Discrete-Time Signal.
   3. Exponential Continuous-Time Signal.
   4. Exponential Discrete-Time Signal.
   5. Unit Step Continuous-Time Signal.
   6. Unit Step Discrete-Time Signal.
   7. Unit Impulse Continuous-Time Signal.
   8. Unit Impulse Discrete-Time Signal.
   9. Ramp Function Continuous-Time Signal.
   10. Ramp Function Discrete-Time Signal.
   11. Exit
Enter the number corresponding to the menu to implement the choice: 8

MENU:
   1. Sinusoidal Continuous-Time Signal.
   2. Sinusoidal Discrete-Time Signal.
   3. Exponential Continuous-Time Signal.
   4. Exponential Discrete-Time Signal.
   5. Unit Step Continuous-Time Signal.
   6. Unit Step Discrete-Time Signal.
   7. Unit Impulse Continuous-Time Signal.
   8. Unit Impulse Discrete-Time Signal.
   9. Ramp Function Continuous-Time Signal.
   10. Ramp Function Discrete-Time Signal.
   11. Exit
Enter the number corresponding to the menu to implement the choice: 9

MENU:
   1. Sinusoidal Continuous-Time Signal.
   2. Sinusoidal Discrete-Time Signal.
   3. Exponential Continuous-Time Signal.
   4. Exponential Discrete-Time Signal.
   5. Unit Step Continuous-Time Signal.
   6. Unit Step Discrete-Time Signal.
   7. Unit Impulse Continuous-Time Signal.
   8. Unit Impulse Discrete-Time Signal.
   9. Ramp Function Continuous-Time Signal.
   10. Ramp Function Discrete-Time Signal.
   11. Exit
Enter the number corresponding to the menu to implement the choice: 10

MENU:
   1. Sinusoidal Continuous-Time Signal.
   2. Sinusoidal Discrete-Time Signal.
   3. Exponential Continuous-Time Signal.
   4. Exponential Discrete-Time Signal.
   5. Unit Step Continuous-Time Signal.
   6. Unit Step Discrete-Time Signal.
   7. Unit Impulse Continuous-Time Signal.
   8. Unit Impulse Discrete-Time Signal.
   9. Ramp Function Continuous-Time Signal.
   10. Ramp Function Discrete-Time Signal.
```

```
   11. Exit
Enter the number corresponding to the menu to implement the choice: 12
Error: Invalid Input! Please try again.

MENU:
   1. Sinusoidal Continuous-Time Signal.
   2. Sinusoidal Discrete-Time Signal.
   3. Exponential Continuous-Time Signal.
   4. Exponential Discrete-Time Signal.
   5. Unit Step Continuous-Time Signal.
   6. Unit Step Discrete-Time Signal.
   7. Unit Impulse Continuous-Time Signal.
   8. Unit Impulse Discrete-Time Signal.
   9. Ramp Function Continuous-Time Signal.
   10. Ramp Function Discrete-Time Signal.
   11. Exit
Enter the number corresponding to the menu to implement the choice: 11

In [2]:
```