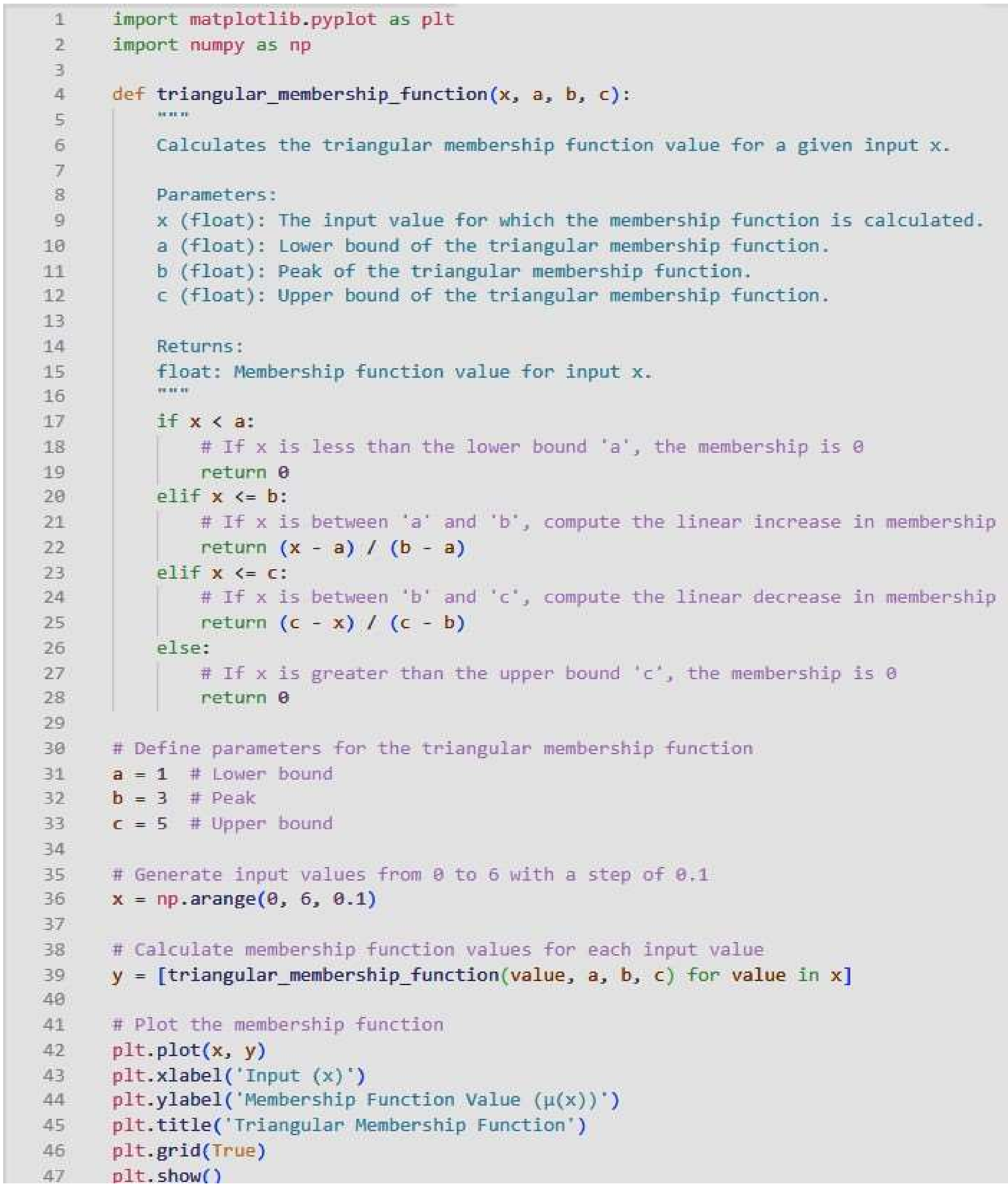
Program 10

Aim : WAP in Python to draw Membership Function curve in Fuzzy Relations of the following function:

a) Triangular function
b) Gaussian function
c) Trapezoid function

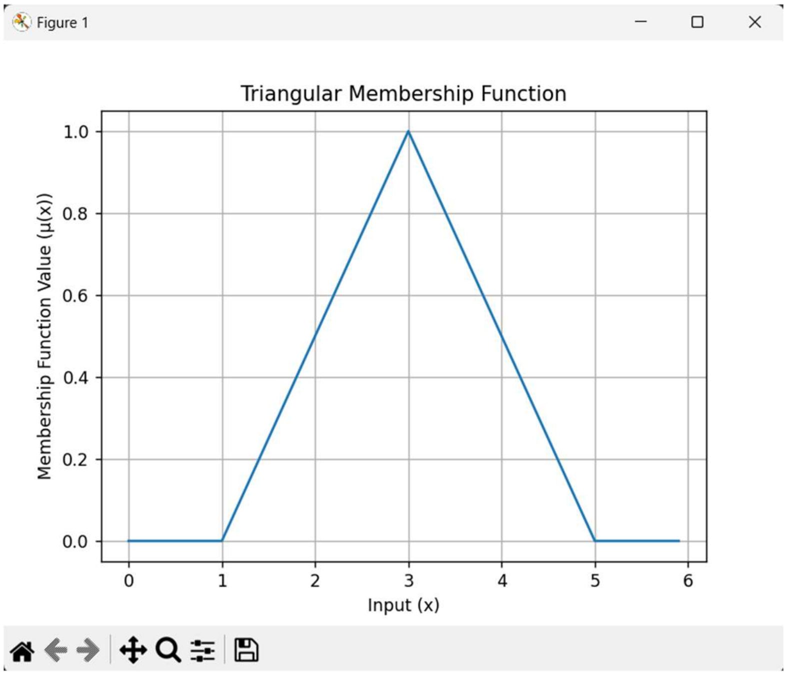
a) Triangular function

Code:



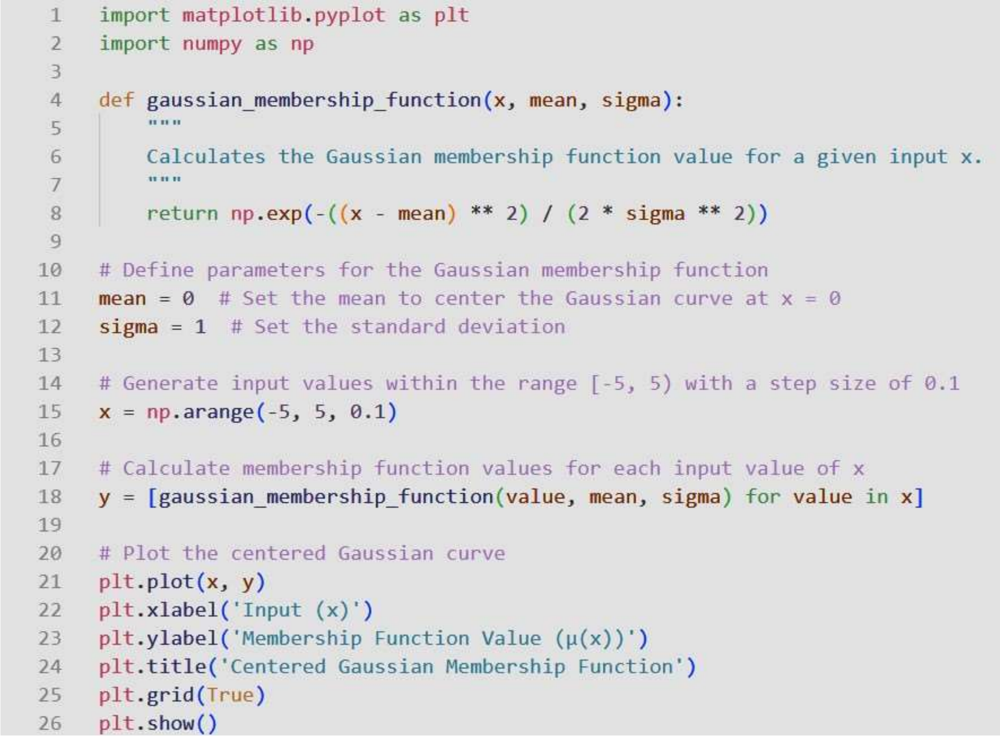
```python
import matplotlib.pyplot as plt
import numpy as np

def triangular_membership_function(x, a, b, c):
    """
    Calculates the triangular membership function value for a given input x.

    Parameters:
    x (float): The input value for which the membership function is calculated.
    a (float): Lower bound of the triangular membership function.
    b (float): Peak of the triangular membership function.
    c (float): Upper bound of the triangular membership function.

    Returns:
    float: Membership function value for input x.
    """
    if x < a:
        # If x is less than the lower bound 'a', the membership is 0
        return 0
    elif x <= b:
        # If x is between 'a' and 'b', compute the linear increase in membership
        return (x - a) / (b - a)
    elif x <= c:
        # If x is between 'b' and 'c', compute the linear decrease in membership
        return (c - x) / (c - b)
    else:
        # If x is greater than the upper bound 'c', the membership is 0
        return 0

# Define parameters for the triangular membership function
a = 1  # Lower bound
b = 3  # Peak
c = 5  # Upper bound

# Generate input values from 0 to 6 with a step of 0.1
x = np.arange(0, 6, 0.1)

# Calculate membership function values for each input value
y = [triangular_membership_function(value, a, b, c) for value in x]

# Plot the membership function
plt.plot(x, y)
plt.xlabel('Input (x)')
plt.ylabel('Membership Function Value (μ(x))')
plt.title('Triangular Membership Function')
plt.grid(True)
plt.show()
```

Output:



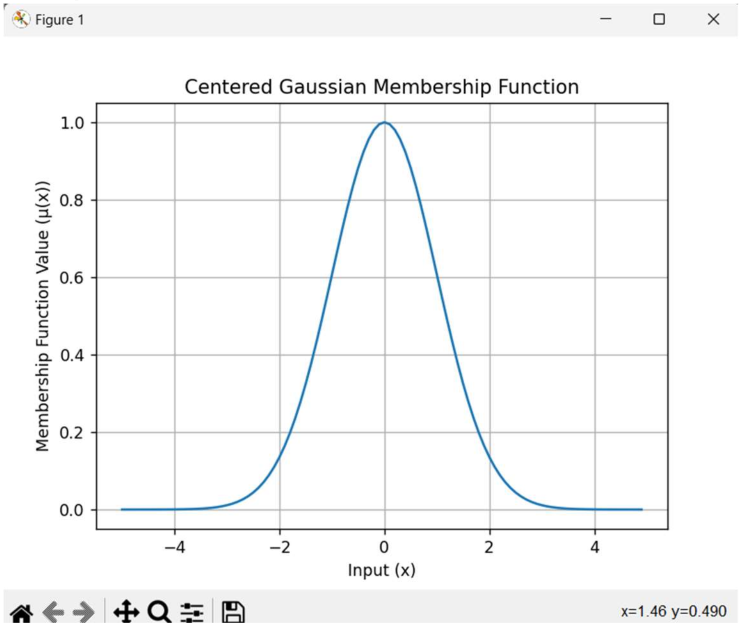
b) Gaussian function

Code:



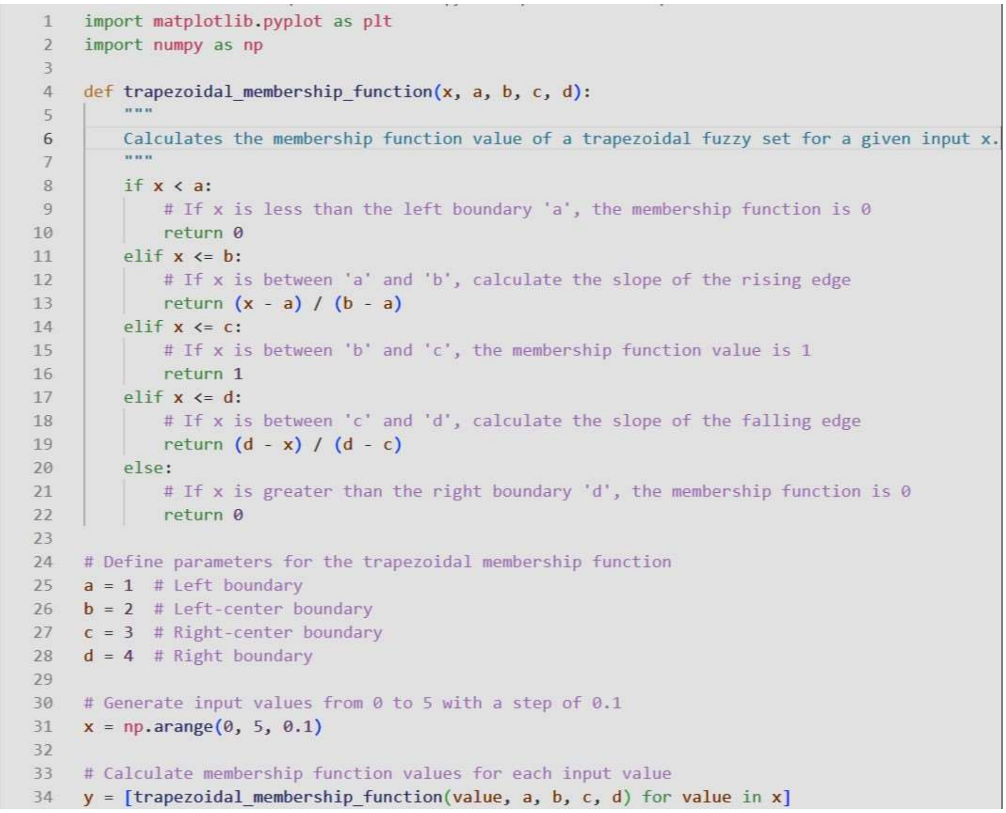
```python
import matplotlib.pyplot as plt
import numpy as np

def gaussian_membership_function(x, mean, sigma):
    """
    Calculates the Gaussian membership function value for a given input x.
    """
    return np.exp(-((x - mean) ** 2) / (2 * sigma ** 2))

# Define parameters for the Gaussian membership function
mean = 0  # Set the mean to center the Gaussian curve at x = 0
sigma = 1  # Set the standard deviation

# Generate input values within the range [-5, 5) with a step size of 0.1
x = np.arange(-5, 5, 0.1)

# Calculate membership function values for each input value of x
y = [gaussian_membership_function(value, mean, sigma) for value in x]

# Plot the centered Gaussian curve
plt.plot(x, y)
plt.xlabel('Input (x)')
plt.ylabel('Membership Function Value (μ(x))')
plt.title('Centered Gaussian Membership Function')
plt.grid(True)
plt.show()
```

Output:



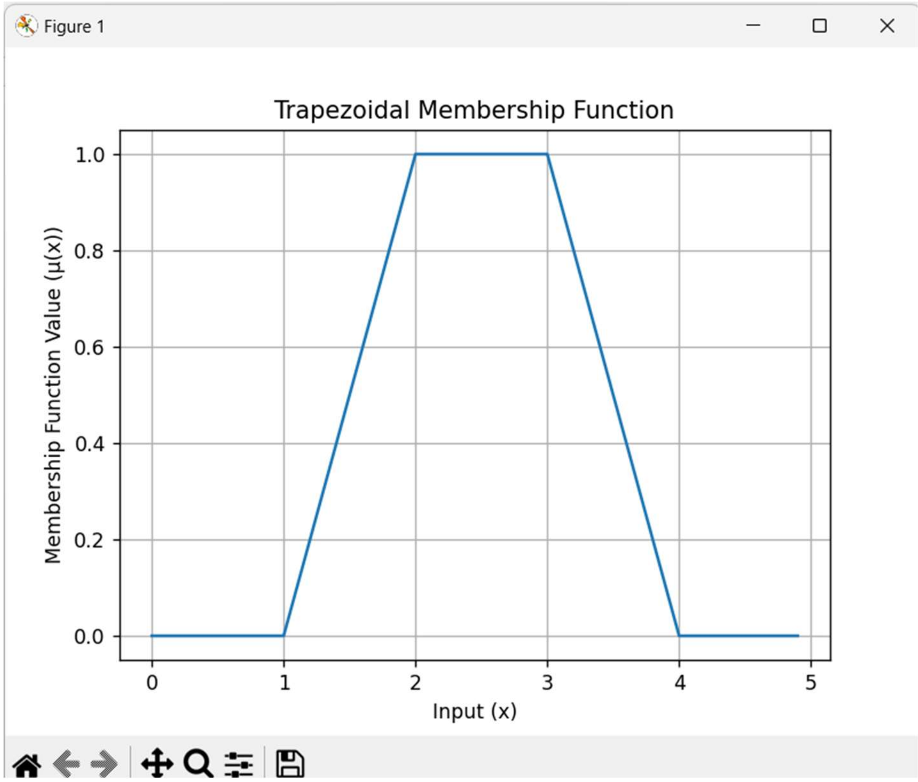
c) Trapezoid function

Code:



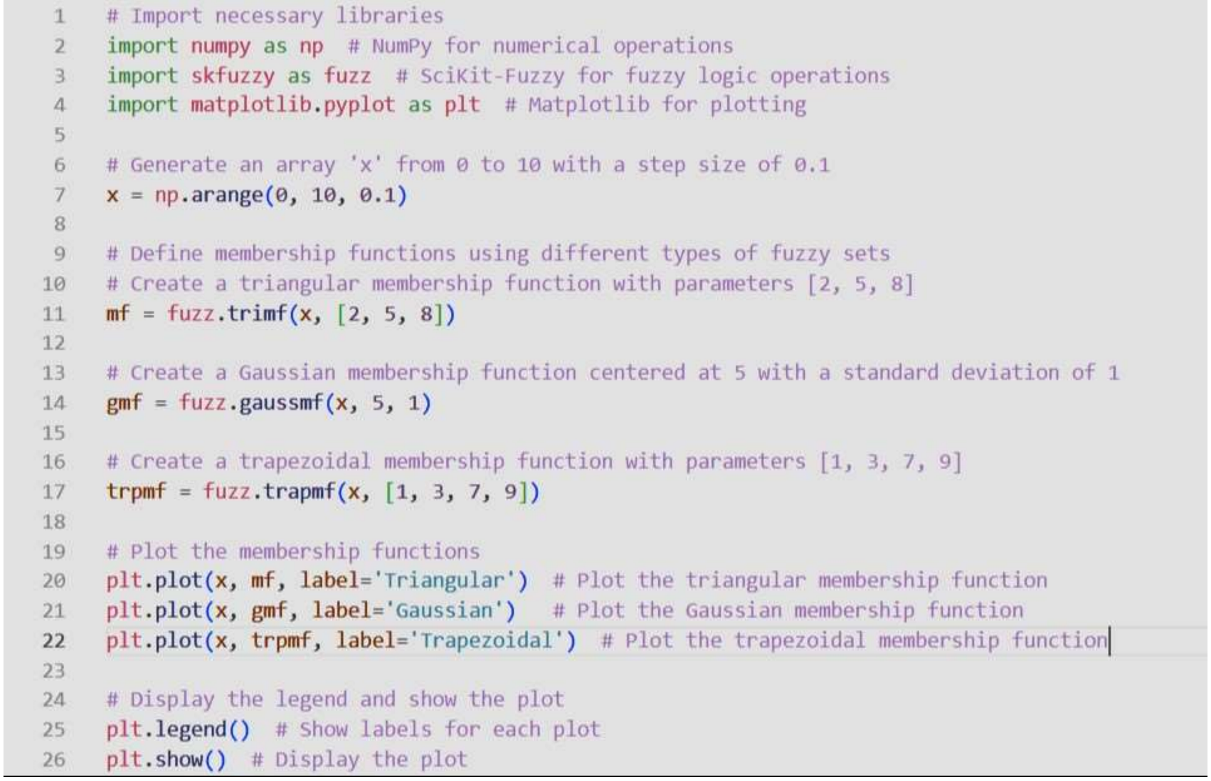
```python
import matplotlib.pyplot as plt
import numpy as np

def trapezoidal_membership_function(x, a, b, c, d):
    """
    Calculates the membership function value of a trapezoidal fuzzy set for a given input x.
    """
    if x < a:
        # If x is less than the left boundary 'a', the membership function is 0
        return 0
    elif x <= b:
        # If x is between 'a' and 'b', calculate the slope of the rising edge
        return (x - a) / (b - a)
    elif x <= c:
        # If x is between 'b' and 'c', the membership function value is 1
        return 1
    elif x <= d:
        # If x is between 'c' and 'd', calculate the slope of the falling edge
        return (d - x) / (d - c)
    else:
        # If x is greater than the right boundary 'd', the membership function is 0
        return 0

# Define parameters for the trapezoidal membership function
a = 1  # Left boundary
b = 2  # Left-center boundary
c = 3  # Right-center boundary
d = 4  # Right boundary

# Generate input values from 0 to 5 with a step of 0.1
x = np.arange(0, 5, 0.1)

# Calculate membership function values for each input value
y = [trapezoidal_membership_function(value, a, b, c, d) for value in x]
```

Output:



d) All membership functions in one graph

Code:



```
# Import necessary libraries
import numpy as np  # NumPy for numerical operations
import skfuzzy as fuzz  # SciKit-Fuzzy for fuzzy logic operations
import matplotlib.pyplot as plt  # Matplotlib for plotting

# Generate an array 'x' from 0 to 10 with a step size of 0.1
x = np.arange(0, 10, 0.1)

# Define membership functions using different types of fuzzy sets
# Create a triangular membership function with parameters [2, 5, 8]
mf = fuzz.trimf(x, [2, 5, 8])

# Create a Gaussian membership function centered at 5 with a standard deviation of 1
gmf = fuzz.gaussmf(x, 5, 1)

# Create a trapezoidal membership function with parameters [1, 3, 7, 9]
trpmf = fuzz.trapmf(x, [1, 3, 7, 9])

# Plot the membership functions
plt.plot(x, mf, label='Triangular')  # Plot the triangular membership function
plt.plot(x, gmf, label='Gaussian')   # Plot the Gaussian membership function
plt.plot(x, trpmf, label='Trapezoidal')  # Plot the trapezoidal membership function

# Display the legend and show the plot
plt.legend()  # Show labels for each plot
plt.show()  # Display the plot
```

Output:

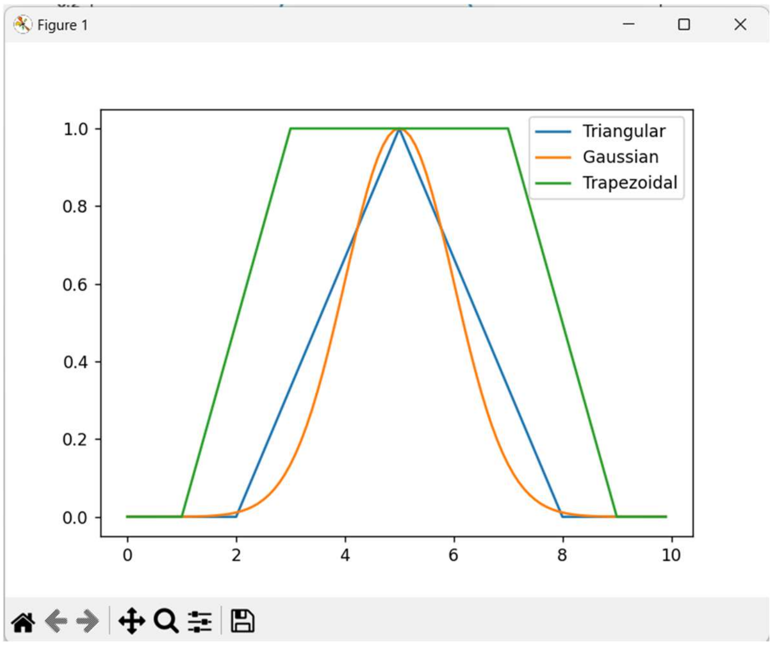


Figure 1
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
Triangular
Gaussian
Trapezoidal