Q.5) Suppose there are 2 fuzzy sets A & B where A & B both contains 10 elements ,

i)fit the triangular membership function on both 2 sets,

ii)combine the 2 sets based on fuzzy union, such that the membership value of both 2 sets will also combine,

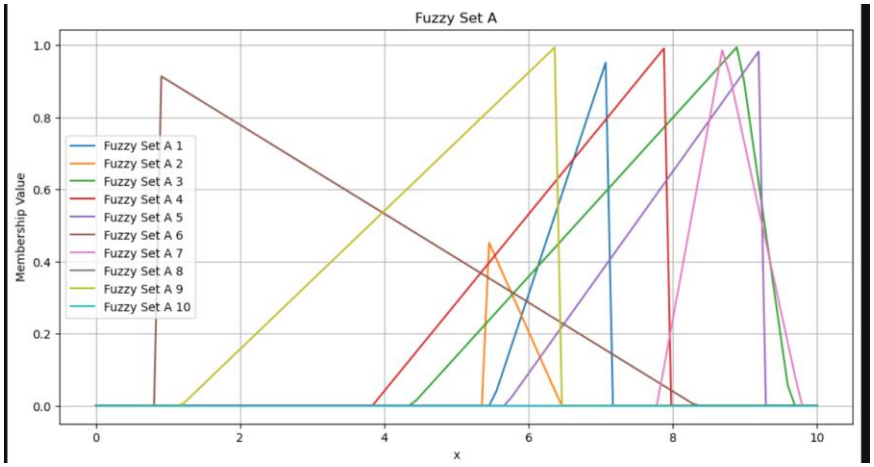
iii)draw the graph of fuzzy set A,B & A union B

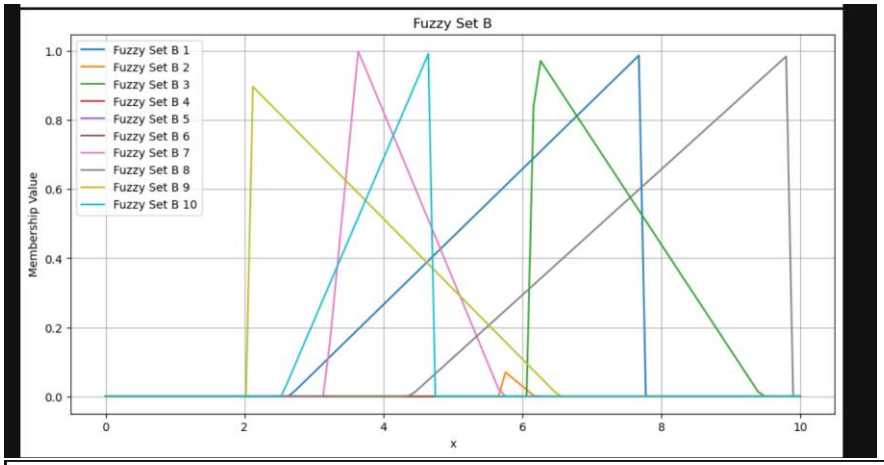
```
import numpy as np
import matplotlib.pyplot as plt

#i)
# Define the triangular membership function
def triangular_membership(x, a, b, c):
    return np.where(x <= a, 0,
                    np.where(x < b, (x - a) / (b - a),
                             np.where(x < c, (c - x) / (c - b), 0)))

# Number of elements in each fuzzy set
num_elements = 10

# Generate random parameters (a, b, c) for 10 elements for set A and set B
np.random.seed(0)  # For reproducibility

# Randomly define parameters for 10 elements for set A
params_A = np.random.uniform(low=0, high=10, size=(num_elements, 3))  #10 rows 3 cols, 3 specifies that each fuzzy set requires three parameters: a, b, and c for the triangular membership function
# Randomly define parameters for 10 elements for set B
params_B = np.random.uniform(low=0, high=10, size=(num_elements, 3))
```

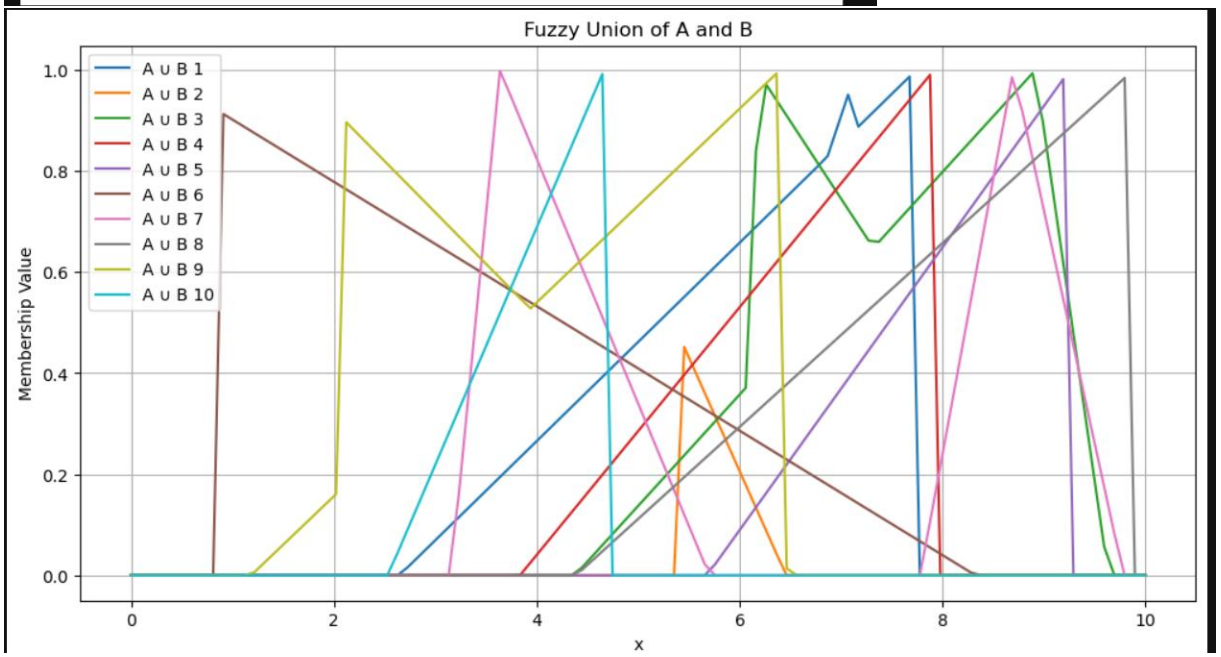
```
#user input of A & B
# params_A = []
# params_B = []
# for i in range(num_elements):
#     a1 = float(input(f"Enter the value for a1 of set A element {i+1}: "))
#     b1 = float(input(f"Enter the value for b1 of set A element {i+1}: "))
#     c1 = float(input(f"Enter the value for c1 of set A element {i+1}: "))
#     params_A.append((a1, b1, c1))
#     a2 = float(input(f"Enter the value for a2 of set B element {i+1}: "))
#     b2 = float(input(f"Enter the value for b2 of set B element {i+1}: "))
#     c2 = float(input(f"Enter the value for c2 of set B element {i+1}: "))
#     params_B.append((a2, b2, c2))
# params_A = np.array(params_A)
# params_B = np.array(params_B)

# Define the range of x values
x = np.linspace(0, 10, 100)



#ii)
# Initialize lists to hold membership values
mv_A_list = []
mv_B_list = []
mv_union_list = []

# Calculate membership values for each element
for (a1, b1, c1), (a2, b2, c2) in zip(params_A, params_B):
    mv_A = triangular_membership(x, a1, b1, c1)
    mv_B = triangular_membership(x, a2, b2, c2)
    mv_union = np.maximum(mv_A, mv_B)
```

```
    mv_A_list.append(mv_A)
    mv_B_list.append(mv_B)
    mv_union_list.append(mv_union)


#eg of for (a1, b1, c1), (a2, b2, c2) in zip(params_A, params_B):
# params_A = [
#     (1, 3, 5),
#     (2, 4, 6),
#     (1.5, 3.5, 5.5),
#     # ... 7 more sets of parameters
# ]

# params_B = [
#     (2, 4, 6),
#     (1, 3, 5),
#     (2.5, 4.5, 6.5),
#     # ... 7 more sets of parameters
# ]

# [
#     ((1, 3, 5), (2, 4, 6)),
#     ((2, 4, 6), (1, 3, 5)),
#     ((1.5, 3.5, 5.5), (2.5, 4.5, 6.5)),
#     # ... 7 more pairs
# ]

# Plot Fuzzy Set A
plt.figure(figsize=(12, 6))
for i, mv_A in enumerate(mv_A_list):
```

```
    plt.plot(x, mv_A, label=f'Fuzzy Set A {i+1}')   #used to plot the membership functions for all 10 elements of fuzzy set
plt.title('Fuzzy Set A')
plt.xlabel('x')
plt.ylabel('Membership Value')
plt.grid(True)
plt.legend()
plt.show()

# Plot Fuzzy Set B
plt.figure(figsize=(12, 6))
for i, mv_B in enumerate(mv_B_list):
    plt.plot(x, mv_B, label=f'Fuzzy Set B {i+1}')
plt.title('Fuzzy Set B')
plt.xlabel('x')
plt.ylabel('Membership Value')
plt.grid(True)
plt.legend()
plt.show()

# Plot Fuzzy Union of A and B
plt.figure(figsize=(12, 6))
for i, mv_union in enumerate(mv_union_list):
    plt.plot(x, mv_union, label=f'A ∪ B {i+1}')
plt.title('Fuzzy Union of A and B')
plt.xlabel('x')
plt.ylabel('Membership Value')
plt.grid(True)
plt.legend()
plt.show()
```


Fuzzy Set A
Membership Value
x
Fuzzy Set A 1
Fuzzy Set A 2
Fuzzy Set A 3
Fuzzy Set A 4
Fuzzy Set A 5
Fuzzy Set A 6
Fuzzy Set A 7
Fuzzy Set A 8
Fuzzy Set A 9
Fuzzy Set A 10


Fuzzy Set B
Membership Value
x
Fuzzy Set B 1
Fuzzy Set B 2
Fuzzy Set B 3
Fuzzy Set B 4
Fuzzy Set B 5
Fuzzy Set B 6
Fuzzy Set B 7
Fuzzy Set B 8
Fuzzy Set B 9
Fuzzy Set B 10


Fuzzy Union of A and B
Membership Value
x
A ∪ B 1
A ∪ B 2
A ∪ B 3
A ∪ B 4
A ∪ B 5
A ∪ B 6
A ∪ B 7
A ∪ B 8
A ∪ B 9
A ∪ B 10