Q: Implement fuzzy relations as discussed in the class. Your program should allow to create any two fuzzy sets. The program should generate fuzzy relation of the two sets. Also write functions to implement Alpha Cut of the relation. The function corresponding to alpha cut should ask for the alpha value.

Solution:

Code available at:

https://colab.research.google.com/drive/1mD-p2chsABSRE8gTLf1jvWNCvRtUAfJX?usp=sharing

```
class FuzzySet:
    def __init__(self):
        self.elements = []
        self.memberships = []

    def add_element(self, element, membership):
        if membership < 0 or membership > 1:
            print("Invalid membership value. Membership value should be between 0 and 1.")
            again_membership = float(input(f"Enter membership value of {element} AGAIN! (between 0 and 1): "))
            self.elements.append(element)
            self.memberships.append(again_membership)
        else:
            self.elements.append(element)
            self.memberships.append(membership)

    def union(self, other_set):
        union_set = FuzzySet()
```

```
        for i in range(len(self.elements)):
            union_set.add_element(self.elements[i], self.memberships[i])
        for i in range(len(other_set.elements)):
            if other_set.elements[i] not in union_set.elements:
                union_set.add_element(other_set.elements[i], other_set.memberships[i])
            else:
                index = union_set.elements.index(other_set.elements[i])
                if other_set.memberships[i] > union_set.memberships[index]:
                    union_set.memberships[index] = other_set.memberships[i]
        return union_set

    def cartesianProduct(self, other_set):
        cartesian_product = []
        mem_cartesian_product = []
        for element1 in self.elements:
            for element2 in other_set.elements:
                cartesian_product.append((element1, element2))

        for i in range(len(self.elements)):
            for j in range(len(other_set.elements)):
                mem_cartesian_product.append(((self.elements[i],
other_set.elements[j]),min(self.memberships[i], other_set.memberships[j])))

        print("Cartesian Product: ", cartesian_product)
        print("Membership of Cartesian Product: ", mem_cartesian_product)

        return mem_cartesian_product

    def alphaCut(self, alpha_value, cp):
```

```
    result = [f"{x[0]} - {x[1]}" for x in cp if x[1] >= alpha_value]
    print(result)



  def print_set(self):
    for i in range(len(self.elements)):
      print(self.elements[i], self.memberships[i])

set1 = FuzzySet()

n = int(input("Enter the number of elements in set A: "))

for i in range(n):
  element = input(f"Enter element {i+1} in set A: ")
  membership = float(input(f"Enter membership value of {element} in set A (between 0 and 1): "))
  set1.add_element(element, membership)

set2 = FuzzySet()

n = int(input("\nEnter the number of elements in set B: "))

for i in range(n):
  element = input(f"Enter element {i+1} in set B: ")
  membership = float(input(f"Enter membership value of {element} in set B (between 0 and 1): "))
  set2.add_element(element, membership)

union_set = set1.union(set2)
```

```
print("\nSet A:")
set1.print_set()
print("\nSet B:")
set2.print_set()
print("\nUnion Set:")
union_set.print_set()

print("\nCartesian Product of Set A and Set B")
cp = set1.cartesianProduct(set2)

alpha_cut = FuzzySet()

alpha_value = float(input(f"\nEnter the alpha cut value : "))
print("\nAlpha Cut:")
alpha_cut.alphaCut(alpha_value, cp)
```

OUTPUT

Enter the number of elements in set A: 2

Enter element 1 in set A: 1

Enter membership value of 1 in set A (between 0 and 1): 0.8

Enter element 2 in set A: 2

Enter membership value of 2 in set A (between 0 and 1): 0.5

Enter the number of elements in set B: 2

Enter element 1 in set B: 3

Enter membership value of 3 in set B (between 0 and 1): 0.7

Enter element 2 in set B: 4

Enter membership value of 4 in set B (between 0 and 1): 0.7

Set A:

1 0.8

2 0.5

Set B:

3 0.7

4 0.7

Union Set:

1 0.8

2 0.5

3 0.7

4 0.7

Cartesian Product of Set A and Set B

Cartesian Product: [('1', '3'), ('1', '4'), ('2', '3'), ('2', '4')]

Membership of Cartesian Product: [(('1', '3'), 0.7), (('1', '4'), 0.7), (('2', '3'), 0.5), (('2', '4'), 0.5)]

Enter the alpha cut value : 0.5

Alpha Cut:

["('1', '3') - 0.7", "('1', '4') - 0.7", "('2', '3') - 0.5", "('2', '4') - 0.5"]