Assignment No. 1

Subject : Computational Intelligence Lab
Division : C
Roll No : 68
Batch : C4

Source Code :

```
setU = [1,2,3,4,5,6,7,8,9,10]
setA = []
setB = []
# limit = input("Enter the range : ")
# print ("Enter the elements : ")
# for i in range(0,limit):
#     setU.append(input())
limit = len(setU)
# condition = raw_input("Enter the condition for set A : ")
condition = "2"
if(condition == "2"):
   for i in range(0,limit):
       if setU[i] % 2 == 0:
           setA.append([setU[i],1])
       else:
           setA.append([setU[i],0])

condition = "3"
if(condition == "3"):
   for i in range(0,limit):
       if setU[i] % 3 == 0:
           setB.append([setU[i],1])
       else:
           setB.append([setU[i],0])

setUnion = []
for i in range(0,len(setU)):
   if setA[i][1] == 0 and setA[i][1] == setB[i][1]:
       setUnion.append([setA[i][0],0])
   else:
       setUnion.append([setA[i][0],1])

setIntersection = []
for i in range(0,len(setU)):
```

```
    if setA[i][0] == setB[i][0] and setA[i][1] == setB[i][1]:
        if setA[i][1] == 1 and setB[i][1] == 1:
            setIntersection.append([setA[i][0],1])
        else:
            setIntersection.append([setA[i][0],0])
    else:
        setIntersection.append([setA[i][0],0])

setDifference = []
for i in range(0,len(setU)):
    if setA[i][0] == setB[i][0]:
        if setA[i][1] == 0 and setB[i][1] == 0:
            setDifference.append([setA[i][0],1])
        else:
            setDifference.append([setA[i][0],0])
    else:
        setDifference.append([setA[i][0],0])

setACompliment = []
for i in range(0,len(setU)):
    if setA[i][1] == 1:
        setACompliment.append([setA[i][0],0])
    else:
        setACompliment.append([setA[i][0],1])

# For Symmetric Difference = (A intersection B') Union (B intersection A')

setBCompliment = []
for i in range(0,len(setU)):
    if setB[i][1] == 1:
        setBCompliment.append([setB[i][0],0])
    else:
        setBCompliment.append([setB[i][0],1])

# Intersection A and B'
setIntersectionABdash = []
for i in range(0,len(setU)):
    if setA[i][0] == setBCompliment[i][0] and setA[i][1] == setBCompliment[i][1]:
        if setA[i][1] == 1 and setBCompliment[i][1] == 1:
            setIntersectionABdash.append([setA[i][0],1])
        else:
            setIntersectionABdash.append([setA[i][0],0])
```

```
    else:
        setIntersectionABdash.append([setA[i][0],0])

# Intersection A and B'
setIntersectionBAdash = []
for i in range(0,len(setU)):
    if setB[i][0] == setACompliment[i][0] and setB[i][1] == setACompliment[i][1]:
        if setB[i][1] == 1 and setACompliment[i][1] == 1:
            setIntersectionBAdash.append([setB[i][0],1])
        else:
            setIntersectionBAdash.append([setB[i][0],0])
    else:
        setIntersectionBAdash.append([setB[i][0],0])

# (A intersection B') Union (B intersection A')
# Set symmetric Difference
setSy = []
for i in range(0,len(setU)):
    if setIntersectionABdash[i][1] == 0 and setIntersectionABdash[i][1] ==
setIntersectionBAdash[i][1]:
        setSy.append([setIntersectionABdash[i][0],0])
    else:
        setSy.append([setIntersectionABdash[i][0],1])

#equality
ctr = 0
for i in range(0,len(setU)):
    if setA[i][0] == setB[i][0] and setA[i][1] == setB[i][1]:
        pass
    else:
        ctr = ctr + 1

# Inclusion of A
ctrnclusion = 0
setInclusion = []
for i in range(0,len(setU)):
    if setA[i][1] <= setB[i][1]:
        pass
    else:
        ctrnclusion = ctrnclusion + 1

# Inclusion of B
```

```
ctrnclusion2 = 0
setInclusion2 = []
for i in range(0,len(setU)):
   if setB[i][1] <= setA[i][1]:
       pass
   else:
       ctrnclusion2 = ctrnclusion2 + 1


#
# Now  printing everything
#

print
("-------------------------------------------------------------------------------
-----------------")

print("Universal Set U = [ ",end='')
for i in range(0,limit):
   if i == limit - 1:
       print(setU[i],end='')
   else:
       print(setU[i],",",end='')
print(" ]")

print("\nsetModBy2 = Fuzzyset [{ ",end='')
for i in range(0,len(setA)):
   if i == len(setA) - 1:
       print(setA[i],end='')
   else:
       print(setA[i],",",end='')
print(" },UniversalSpace -> {",setU[0],",","1,",setU[len(setU)-1],"}]\n")

print("\nsetModBy3 = Fuzzyset [{ ",end='')
for i in range(0,len(setB)):
   if i == len(setB) - 1:
       print(setB[i],end='')
   else:
       print(setB[i],",",end='')
print(" },UniversalSpace -> {",setU[0],",","1,",setU[len(setU)-1],"}]\n")
```

```
print
("-------------------------------------------------------------------------------
-----------------")
if ctr != 0:
   print ("SetA and setB are not equal.")
print
("-------------------------------------------------------------------------------
-----------------")

if ctrnclusion == 0:
   print ("SetA is included in setB.")
else:
   print ("SetA is not included in setB.")
print
("-------------------------------------------------------------------------------
-----------------")

if ctrnclusion == 0 and ctrnclusion2 == 0:
   print ("SetA and setB are 'Comparable'.")
else:
   print ("SetA and setB are 'Not Comparable'.")
print
("-------------------------------------------------------------------------------
-----------------")

print("\nsetUnion = Fuzzyset [{ ",end='')
for i in range(0,len(setUnion)):
   if i == len(setUnion) - 1:
       print(setUnion[i],end='')
   else:
       print(setUnion[i],",",end='')
print(" },UniversalSpace -> {",setU[0],",","1,",setU[len(setU)-1],"}]\n")

print
("-------------------------------------------------------------------------------
-----------------")

print("\nsetIntersection = Fuzzyset [{ ",end='')
for i in range(0,len(setIntersection)):
   if i == len(setIntersection) - 1:
       print(setIntersection[i],end='')
   else:
```

```
        print(setIntersection[i],",",end='')
print(" },UniversalSpace -> {",setU[0],",","1,",setU[len(setU)-1],"}]\n")

print
("-------------------------------------------------------------------------------
-----------------")

print("\nsetDifference = Fuzzyset [{ ",end='')
for i in range(0,len(setDifference)):
   if i == len(setDifference) - 1:
       print(setDifference[i],end='')
   else:
       print(setDifference[i],",",end='')
print(" },UniversalSpace -> {",setU[0],",","1,",setU[len(setU)-1],"}]\n")

print
("-------------------------------------------------------------------------------
-----------------")

print("\nsetACompliment = Fuzzyset [{ ",end='')
for i in range(0,len(setACompliment)):
   if i == len(setACompliment) - 1:
       print(setACompliment[i],end='')
   else:
       print(setACompliment[i],",",end='')
print(" },UniversalSpace -> {",setU[0],",","1,",setU[len(setU)-1],"}]\n")

print
("-------------------------------------------------------------------------------
-----------------")

print("\nsetBCompliment = Fuzzyset [{ ",end='')
for i in range(0,len(setBCompliment)):
   if i == len(setBCompliment) - 1:
       print(setBCompliment[i],end='')
   else:
       print(setBCompliment[i],",",end='')
print(" },UniversalSpace -> {",setU[0],",","1,",setU[len(setU)-1],"}]\n")

print
("-------------------------------------------------------------------------------
-----------------")
```

```
print("\nsetIntersectionABdash = Fuzzyset [{ ",end='')
for i in range(0,len(setIntersectionABdash)):
   if i == len(setIntersectionABdash) - 1:
       print(setIntersectionABdash[i],end='')
   else:
       print(setIntersectionABdash[i],",",end='')
print(" },UniversalSpace -> {",setU[0],",","1,",setU[len(setU)-1],"}]\n")

print
("-------------------------------------------------------------------------------
-----------------")

print("\nsetIntersectionBAdash = Fuzzyset [{ ",end='')
for i in range(0,len(setIntersectionBAdash)):
   if i == len(setIntersectionBAdash) - 1:
       print(setIntersectionBAdash[i],end='')
   else:
       print(setIntersectionBAdash[i],",",end='')
print(" },UniversalSpace -> {",setU[0],",","1,",setU[len(setU)-1],"}]\n")

print
("-------------------------------------------------------------------------------
-----------------")

print("\nsetSy = Fuzzyset [{ ",end='')
for i in range(0,len(setSy)):
   if i == len(setSy) - 1:
       print(setSy[i],end='')
   else:
       print(setSy[i],",",end='')
print(" },UniversalSpace -> {",setU[0],",","1,",setU[len(setU)-1],"}]\n")

print
("-------------------------------------------------------------------------------
-----------------")
```

Output of Program :

```
# ---------------------- OUTPUT OF PROGRAM -----------------------------------#

# Dhaneshs-MacBook-Air:Computational-Intelligence dhaneshpawar$ python3 ass1.py
#
-------------------------------------------------------------------------------
---------------
# Universal Set U = [ 1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9 ,10 ]

# setModBy2 = Fuzzyset [{ [1, 0] ,[2, 1] ,[3, 0] ,[4, 1] ,[5, 0] ,[6, 1] ,[7, 0]
,[8, 1] ,[9, 0] ,[10, 1] },UniversalSpace -> { 1 , 1, 10 }]



# setModBy3 = Fuzzyset [{ [1, 0] ,[2, 0] ,[3, 1] ,[4, 0] ,[5, 0] ,[6, 1] ,[7, 0]
,[8, 0] ,[9, 1] ,[10, 0] },UniversalSpace -> { 1 , 1, 10 }]

#
-------------------------------------------------------------------------------
---------------
# SetA and setB are not equal.
#
-------------------------------------------------------------------------------
---------------
# SetA is not included in setB.
#
-------------------------------------------------------------------------------
---------------
# SetA and setB are 'Not Comparable'.
#
-------------------------------------------------------------------------------
---------------

# setUnion = Fuzzyset [{ [1, 0] ,[2, 1] ,[3, 1] ,[4, 1] ,[5, 0] ,[6, 1] ,[7, 0]
,[8, 1] ,[9, 1] ,[10, 1] },UniversalSpace -> { 1 , 1, 10 }]

#
-------------------------------------------------------------------------------
---------------

# setIntersection = Fuzzyset [{ [1, 0] ,[2, 0] ,[3, 0] ,[4, 0] ,[5, 0] ,[6, 1] ,[7,
0] ,[8, 0] ,[9, 0] ,[10, 0] },UniversalSpace -> { 1 , 1, 10 }]
```

```
#
-------------------------------------------------------------------------------
---------------

# setDifference = Fuzzyset [{ [1, 1] ,[2, 0] ,[3, 0] ,[4, 0] ,[5, 1] ,[6, 0] ,[7,
1] ,[8, 0] ,[9, 0] ,[10, 0] },UniversalSpace -> { 1 , 1, 10 }]

#
-------------------------------------------------------------------------------
---------------

# setACompliment = Fuzzyset [{ [1, 1] ,[2, 0] ,[3, 1] ,[4, 0] ,[5, 1] ,[6, 0] ,[7,
1] ,[8, 0] ,[9, 1] ,[10, 0] },UniversalSpace -> { 1 , 1, 10 }]

#
-------------------------------------------------------------------------------
---------------

# setBCompliment = Fuzzyset [{ [1, 1] ,[2, 1] ,[3, 0] ,[4, 1] ,[5, 1] ,[6, 0] ,[7,
1] ,[8, 1] ,[9, 0] ,[10, 1] },UniversalSpace -> { 1 , 1, 10 }]

#
-------------------------------------------------------------------------------
---------------

# setIntersectionABdash = Fuzzyset [{ [1, 0] ,[2, 1] ,[3, 0] ,[4, 1] ,[5, 0] ,[6,
0] ,[7, 0] ,[8, 1] ,[9, 0] ,[10, 1] },UniversalSpace -> { 1 , 1, 10 }]

#
-------------------------------------------------------------------------------
---------------

# setIntersectionBAdash = Fuzzyset [{ [1, 0] ,[2, 0] ,[3, 1] ,[4, 0] ,[5, 0] ,[6,
0] ,[7, 0] ,[8, 0] ,[9, 1] ,[10, 0] },UniversalSpace -> { 1 , 1, 10 }]

#
-------------------------------------------------------------------------------
---------------

# setSy = Fuzzyset [{ [1, 0] ,[2, 1] ,[3, 1] ,[4, 1] ,[5, 0] ,[6, 0] ,[7, 0] ,[8,
1] ,[9, 1] ,[10, 1] },UniversalSpace -> { 1 , 1, 10 }]
```

```
#
--------------------------------------------------------------------------------
---------------
# Dhaneshs-MacBook-Air:Computational-Intelligence dhaneshpawar$
```