

Practicals\Q1\Q1.py

```
1  # Create a class SET. Create member functions to perform the following SET operations:
2  # 1) ismember: check whether an element belongs to the set or not and return value as
3  # true/false.
4  # 2) powerset: list all the elements of the power set of a set .
5  # 3) subset: Check whether one set is a subset of the other or not.
6  # 4) union and Intersection of two Sets.
7  # 5) complement: Assume Universal Set as per the input elements from the user.
8  # 6) set Difference and Symmetric Difference between two sets.
9  # 7) cartesian Product of Sets.
10 # Write a menu driven program to perform the above functions on an instance of the SET
11 # class.
12
13 class SET():
14     def __init__(self, lst) :
15         self.set = set(lst)
16
17     def isMember(self, element):
18         if element in self.set:
19             return True
20         return False
21
22     def powerSet(self):
23         pass
24
25     def isSubsetOf(self, otherset):
26         for ele in self.set :
27             if ele not in otherset.set:
28                 return False
29         return True
30
31     def setUnion(self, otherset):
32         unionSet = set()
33         for i in self.set:
34             unionSet.add(i)
35         for j in otherset.set:
36             unionSet.add(j)
37         return unionSet
38
39     def setIntersection(self, otherset):
40         intersect = set()
41         for i in self.set:
42             if i in otherset.set:
43                 intersect.add(i)
44         for j in otherset.set:
45             if j in self.set:
46                 intersect.add(j)
47         return intersect
48
49     def complement(self):
50         universalSet = eval(input("Enter Universal Set : "))
51         compl = set()
52         for i in universalSet:
53             if i not in self.set:
54                 compl.add(i)
55         return compl
56
```

```

57     def setDifference(self, otherset):
58         diff = self.set.copy()
59         intersection = self.setIntersection(otherset)
60         for ele in intersection:
61             diff.discard(ele)
62         return diff
63
64     def symmetricDifference(self, otherset):
65         union = otherset.setUnion(self)
66         intersection = otherset.setIntersection(self)
67         for i in intersection:
68             union.discard(i)
69         return union
70
71     def print(self):
72         print(self.set)
73
74
75
76
77
78
79
80
81
82 setA = SET([1,2,3,4,7,12])
83 setB = SET([4,3,2,1,7,11])
84
85 print("setA = ")
86 setA.print()
87 print("setB = ")
88 setB.print()
89 print("Checking if 11 is member of setA and setB")
90 print("setA", setA.isMember(11))
91 print("setB", setB.isMember(11))
92
93 print("Union of setA and setB : ")
94 print(setA.setUnion(setB))
95
96 print("Intersection of setA and setB : ")
97 print(setA.setIntersection(setB))
98
99 print("Cheking if setA is subset of setB : ")
100 print(setA.isSubsetOf(setB))
101
102 print("Complement of setA :")
103 # print(setA.complement())
104
105 print("Set Differnece of setA and setB : ")
106 print(setA.setDifference(setB))
107
108 print("Symmentric Difference of setA and setB : ")
109 print(setA.symmetricDifference(setB))
110

```