


```

In [ ]: 1 from itertools import combinations
2 def ADD(l,s):
3     if(l not in s):
4         s.add(l)
5     else:
6         print("Element already exist")
7 def REMOVE(l,s):
8     if(l in s):
9         s.remove(l)
10    else:
11        print("Element doesn't exist ")
12
13 def UNION(s1,s2):
14     if(len(s1)!=0 and len(s2)!=0):
15         print(s1.union(s2))
16 def INTERSECTION(s1,s2):
17     if(len(s1)!=0 and len(s2)!=0):
18         print(s1.intersection(s2))
19
20 def DIFFERENCE(s1,s2):
21     if(len(s1)!=0 and len(s2)!=0):
22         print(s1-s2)
23
24 def SUBSET(s1,s2):
25     if(len(s1)!=0 and len(s2)!=0):
26         print(s1.issubset(s2))
27
28 def LEN(s1):
29     l=0
30     for x in s1:
31         l=l+1
32     print(l)
33
34 def SYMMETRIC_DIFFERENCE(s1,s2):
35     if(len(s1)!=0 and len(s2)!=0):
36         print(s1.symmetric_difference(s2))
37
38 def Power_Set(s1,):
39     """Returns the power set of the given set s."""
40     power_set_list = []
41     for r in range(len(s1) + 1):
42         power_set_list.extend(combinations(s1, r))
43     print(power_set_list)
44
45 def unique_subsets(s):
46     """Returns all unique subsets of the given set s."""
47     # Convert the input to a list to support indexing
48     s = list(s)
49     subsets = []
50     # Generate all subsets
51     for r in range(len(s) + 1):
52         subsets.extend(combinations(s, r))
53     print(subsets)
54
55

```

