In [ ]:

```
from itertools import combinations
def ADD(l,s):
    if(l not in s):
        s.add(l)
    else:
        print("Element already exist")
def REMOVE(l,s):
    if(l in s):
        s.remove(l)
    else:
        print("Element doesn't exist ")

def UNION(s1,s2):
    if(len(s1)!=0 and len(s2)!=0):
        print(s1.union(s2))
def INTERSECTION(s1,s2):
    if(len(s1)!=0 and len(s2)!=0):
        print(s1.intersection(s2))

def DIFFERENCE(s1,s2):
    if(len(s1)!=0 and len(s2)!=0):
        print(s1-s2)

def SUBSET(s1,s2):
    if(len(s1)!=0 and len(s2)!=0):
        print(s1.issubset(s2))

def LEN(s1):
    l=0
    for x in s1:
        l=l+1
    print(l)

def SYMMETRIC_DIFFERENCE(s1,s2):
    if(len(s1)!=0 and len(s2)!=0):
        print(s1.symmetric_difference(s2))

def Power_Set(s1,):
    """Returns the power set of the given set s."""
    power_set_list = []
    for r in range(len(s1) + 1):
        power_set_list.extend(combinations(s1, r))
    print(power_set_list)

def unique_subsets(s):
    """Returns all unique subsets of the given set s."""
    # Convert the input to a list to support indexing
    s = list(s)
    subsets = []
    # Generate all subsets
    for r in range(len(s) + 1):
        subsets.extend(combinations(s, r))
    print(subsets)


```

```
56
```