Q1.

Q1.py > ...

```python
# even numbers List
list_ex = list([x for x in range(11) if x%2==0])
# Custom Tuple
tuple_ex = (2,3,'Hello',5,['Sourabh',8351,234],8)
# Odd number set
set_ex = set([x for x in range(11) if x%2!=0])
#
print('Traversing The list: ', end=" ")
for i in list_ex:
    print(i,end=" ")
print('\n')
#
print('Traversing The tuple: ', end=" ")
for i in tuple_ex:
    print(i,end=" ")
print('\n')
#
print('Traversing The set: ', end=" ")
for i in set_ex:
    print(i, end=" ")
print('\n')
```

TERMINAL

```
PS E:\Study Material\CDAC\Python Learning\31-03-2022> python
Traversing The list:  0 2 4 6 8 10

Traversing The tuple:  2 3 Hello 5 ['Sourabh', 8351, 234] 8

Traversing The set:  1 3 5 7 9
```

Q2.Script:

Q2.py > ...

```python
S1 = int(input('Enter the Number of elements in the First set: '))
s1_list = []
for i in range(S1):
    se = int(input("Enter the Element: "))
    s1_list.append(se)
sum_s1 = sum(s1_list)
minimum_s1 = min(s1_list)
maximum_s1 = max(s1_list)
S1 = set(s1_list)
# print(type(S1))
print(S1)

S2 = int(input('Enter the Number of elements in the Second set: '))
s2_list = []
for i in range(S2):
    se = int(input("Enter the Element: "))
    s2_list.append(se)
sum_s2 = sum(s1_list)
minimum_s2 = min(s2_list)
maximum_s2 = max(s2_list)
S2 = set(s2_list)
# print(type(S1))
print(S2)
# len
print("The len of the First set is",len(S1))
print("The len of the Second set is",len(S2))
# min
print("The Minimum element in First set is",minimum_s1)
print("The Maximum element in First set is",maximum_s2)
# set 2
print("The minimum element in Second set is",minimum_s2)
print("The minimum element in Second set is",maximum_s2)
# sum
print("The sum of all elements in the First set is",sum_s1)
print("The sum of all elements in the Second set is",sum_s2)
print("The sum of both the set is",sum_s1+sum_s2)
# union
un = S1|S2
print('The Union is')
print(un)
# difference
diff = S1.difference(S2)
```

```python
print('The difference is ',diff,sep="\n")
# intersection
inter = S1.intersection(S2)
print('The intersection is ',inter,sep="\n")
print()
# Symmetric Difference
SD = S1.symmetric_difference(S2)
# issuperset
if S1.issuperset(S2):
    print('S1 is Superset of S2')
elif S2.issuperset(S1):
    print("No, Second Set is the SuperSet of First Set")
else:
    print('No SuperSet')
#  issubset
if S1.issubset(S2):
    print('S1 is Subset of S2')
elif S2.issubset(S1):
    print("No, Second Set is the Subset of First Set")
else:
    print('No SubSet')
```

Output:

```
PS E:\Study Material\CDAC\Python Learning\31-03-2022> python -u "e:\Study Material\CDAC\Python Learning\31-03-2022\Q2.py"
Enter the Number of elements in the First set: 2
Enter the Element: 1
Enter the Element: 3
{1, 3}
Enter the Number of elements in the Second set: 4
Enter the Element: 7
Enter the Element: 8
Enter the Element: 9
Enter the Element: 6
{8, 9, 6, 7}
The len of the First set is 2
The len of the Second set is 4
The Minimum element in First set is 1
The Maximum element in First set is 9
The minimum element in Second set is 6
The minimum element in Second set is 9
The sum of all elements in the First set is 4
The sum of all elements in the Second set is 4
The sum of both the set is 8
The Union is
{1, 3, 6, 7, 8, 9}
The difference is
{1, 3}
The intersection is
set()

No SuperSet
No SubSet
```

Q3.

```
Q3.py > ...
1   t1_list = []
2   for i in range(5):
3       T = int(input("Enter number {} :".format(i+1)))
4       t1_list.append(T)
5
6   minimum_t1 = min(t1_list)
7   maximum_t1 = max(t1_list)
8   sum_t1 = sum(t1_list)
9   t1_list.sort()
10  T = tuple(t1_list)
11  print('The Minimum element is:',minimum_t1)
12  print('The Maximum element is:',maximum_t1)
13  print('The Sum of all the elements is:',sum_t1)
14  print('Printing Tuple:',T)
```

TERMINAL

```
PS E:\Study Material\CDAC\Python Learning\31-03-2022> python -u "e:\Study Material\CDAC\Python Learning\31-03-2022\Q3.py"
Enter number 1 :1
Enter number 2 :5
Enter number 3 :7
Enter number 4 :9
Enter number 5 :6
The Minimum element is: 1
The Maximum element is: 9
The Sum of all the elements is: 28
Printing Tuple: (1, 5, 6, 7, 9)
```

Q4.

```
Q4.py > ...
1   T = tuple([1,2,5,6])
2   T_concatenated = T+T
3   print(T_concatenated)
4   T_repitation = T*3
5   print(T_repitation)
```

TERMINAL

```
PS E:\Study Material\CDAC\Python Learning\31-03-2022> python -u "e:\Study Material\CDAC\Python Learning\31-03-2022\Q4.py"
(1, 2, 5, 6, 1, 2, 5, 6)
(1, 2, 5, 6, 1, 2, 5, 6, 1, 2, 5, 6)
```