In [1]:
```
#1) Write a Python program to sum all the items in a list.
total = 0

list = [17, 9, 5, 9, 17]

for item in range(0, len(list)):
        total = total + list[item]

print("Sum of all elements in given list:",total)
```

```
Sum of all elements in given list: 57
```

In [2]:
```
#2) Write a Python program to multiplies all the items in a list.
def mult_list(list):

    product = 1
    for i in list:
        product = product * i
    return product


list1 = [17, 9, 8, 1]
print(list1)
print("product: ", mult_list(list1))
```

```
[17, 9, 8, 1]
product:  1224
```

In [6]:
```
#3) Write a Python program to get a list, sorted in increasing order by the last element in each tuple
#from a given list of non-empty tuples.
def last(n):
    return n[-1]

def sort(tuples):
    return sorted(tuples, key=last)

a=[(2, 3), (1, 2), (2, 2)]
print("Sorted:")
print(sort(a))
```

```
Sorted:
[(1, 2), (2, 2), (2, 3)]
```

In [7]:
```
#1) Write a Python program to create a tuple.
x = (10, 20, 30, 40, 50)
print(x)
print("Datatype of y= ", type(x))
```

```
(10, 20, 30, 40, 50)
Datatype of y=  <class 'tuple'>
```

In [8]:
```
#2) Write a Python program to create a tuple with different data types.
t1 = ("tuple", False, "3.2", 17)
print(t1)
```

```
('tuple', False, '3.2', 17)
```

In [9]:
```
#3) Write a Python program to check whether an element exists within a tuple
t1 = ("p", "y", "t", "h", "o", "n", "p", "r", "o","g","r","a","m","e")
print("p" in t1)
print("H" in t1)
print(5 in t1)
```

```
True
False
False
```

In [10]:
```
#1) Write a Python program to create a set
```

```
#1) Write a Python program to create a set.

x=set(['wlecome','tybcs','in','python','practical'])
print(x)
print(type(x))
```

```
{'python', 'practical', 'tybcs', 'wlecome', 'in'}
<class 'set'>
```

In [11]:

```
#2) Write a Python program to iterate over sets.
num_set = set([0, 1, 2, 3, 4, 5])
for n in num_set:
  print(n, end=' ')

print("\n\nCreating a set using string:")
char_set = set("Python")

for val in char_set:
    print(val, end=' ')
```

```
0 1 2 3 4 5

Creating a set using string:
o P t n h y
```

In [12]:

```
#3) Write a Python program to create set difference.
set1 = set([1, 1, 2, 3, 4, 5])
set2 = set([1, 5, 6, 7, 8, 9])

print("\nOriginal sets:")
print(set1)
print(set2)

r1 = set1.difference(set2)
print("\nDifference of set1 - set2:")
print(r1)

r2 = set2.difference(set1)
print("\nDifference of set2 - set1:")
print(r2)
```

```
Original sets:
{1, 2, 3, 4, 5}
{1, 5, 6, 7, 8, 9}

Difference of set1 - set2:
{2, 3, 4}

Difference of set2 - set1:
{8, 9, 6, 7}
```

In [13]:

```
#1) Write a Python script to sort (ascending and descending) a dictionary by value.
import operator
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
print('Original dictionary : ',d)

Sort_dict = dict( sorted(d.items(), key=operator.itemgetter(1)))
print('Ascending order by value : ',Sort_dict)

Sort_dict = dict( sorted(d.items(), key=operator.itemgetter(1),reverse=True))
print('Descending order by value : ',Sort_dict)
```

```
Original dictionary :  {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
Ascending order by value :  {0: 0, 2: 1, 1: 2, 4: 3, 3: 4}
Descending order by value :  {3: 4, 4: 3, 1: 2, 2: 1, 0: 0}
```

In [14]:

```
#2) Write a Python script to add a key to a dictionary.
d = {0:10, 1:20}
print(d)
d.update({2:30})
print("Updated Dictionary with key :")
print(d)
```

```
{0: 10, 1: 20}
Updated Dictionary with key :
{0: 10, 1: 20, 2: 30}
```

In [15]:
```
#3) Write a Python program to iterate over dictionaries using for loops.
d = {'Red': 5, 'Green': 2, 'Blue': 3}
for color_key, value in d.items():
     print(color_key, 'corresponds to ', d[color_key])
```

```
Red corresponds to  5
Green corresponds to  2
Blue corresponds to  3
```

In [16]:
```
#1. Write a Python program to remove duplicates from a list.
list1 = [1, 2, 3, 1, 2, 4, 5, 4 ,6, 2,5,8,8]
print("List Before removing duplicates :\n", list1)
list2 = [] #Temporary List

for i in list1:
    if i not in list2:
        list2.append(i)

list1 = list2

print("List After removing duplicates :\n", list1)
```

```
List Before removing duplicates :
 [1, 2, 3, 1, 2, 4, 5, 4, 6, 2, 5, 8, 8]
List After removing duplicates :
 [1, 2, 3, 4, 5, 6, 8]
```

In [17]:
```
#2. Write a Python program to check a list is empty or not.
def Enquiry(lis1):
    if len(lis1) == 0:
        return 0
    else:
        return 1

# Driver Code
lis1 = [5,6]
if Enquiry(lis1):
    print ("The list is not empty")
else:
    print("Empty List")
```

```
The list is not empty
```

In [18]:
```
#1. Write a Python program to convert a list to a tuple.
def convert(list):
    return tuple(list)

list = [1, 2, 3, 4]
print(convert(list))
```

```
(1, 2, 3, 4)
```

In [19]:
```
#2. Write a Python program to remove an item from a tuple.
tuple = [(1,2), (2.25, 9.9), ("Python", "practical")]
tuple.pop(1)
print(tuple)
```

```
[(1, 2), ('Python', 'practical')]
```

In [20]:
```
#3. Write a Python program to slice a tuple.
numTuple = (11, 22, 33, 44, 55, 66, 77, 88, 99, 100)
print("Tuple Items = ", numTuple)

slice1 = numTuple[1:7]
print("Tuple Items from 2 to 6 = ", slice1)
```

```
Tuple Items =  (11, 22, 33, 44, 55, 66, 77, 88, 99, 100)
Tuple Items from 2 to 6 =  (22, 33, 44, 55, 66, 77)
```

In [21]:

```
#4. Write a Python program to find the length of a tuple.
tuple1 = (10, 20, 30, 40, 50,60)
print("Tuple Items = ", tuple1)

print("Tuple Length = ", len(tuple1))
```

```
Tuple Items =  (10, 20, 30, 40, 50, 60)
Tuple Length =  6
```

In [22]:

```
#1. Write a Python program to check if a set is a subset of another set.
A = {1, 2, 3,4}
B = {1, 2, 3, 4, 6}
C = {1, 2, 4,4}

print("A is SubSet B :",A.issubset(B))

print("B is SubSet A :",B.issubset(A))

print("A is SubSet C :",A.issubset(C))

print("C is SubSet B :",C.issubset(B))
```

```
A is SubSet B : True
B is SubSet A : False
A is SubSet C : False
C is SubSet B : True
```

In [23]:

```
#2. Write a Python program to find maximum and the minimum value in a set.
setn = {5, 10, 3, 15, 2, 20}
print("Original set elements:")
print(setn)
print(type(setn))

print("\nMaximum value of the said set:")
print(max(setn))

print("\nMinimum value of the said set:")
print(min(setn))
```

```
Original set elements:
{2, 3, 20, 5, 10, 15}
<class 'set'>

Maximum value of the said set:
20

Minimum value of the said set:
2
```

In [24]:

```
#3. Write a Python program to find the length of a set.
setn = {5, 10, 3, 15, 2, 20}
print("\nOriginal set elements:")
print(setn)
print(type(setn))
print("Length of the set:")
print(len(setn))

setn = {5, 5, 5, 5, 5, 5}
print("\nOriginal set elements:")
print(setn)
print("Length of the set:")
print(len(setn))

setn = {5, 5, 5, 5, 5, 5, 7}
print("\nOriginal set elements:")
print(setn)
print("Length of the set:")
print(len(setn))
```

```
Original set elements:
{2, 3, 20, 5, 10, 15}
<class 'set'>
Length of the set:
6
```

```
Original set elements:
{5}
Length of the set:
1

Original set elements:
{5, 7}
Length of the set:
2
```

In [26]:

```
#1. Write a Python script to generate and print a dictionary that contains a number (between 1 and n)
#in the form (x, x*x).
n=int(input("Input a number :"))
d = dict()

for x in range(1,n+1):
    d[x]=x*x

print("A number (between 1 and n) in the form (x, x*x) :\n ",d)
```

```
Input a number :5
A number (between 1 and n) in the form (x, x*x) :
  {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

In [27]:

```
#2. Write a Python script to merge two Python dictionaries.
d1 = {'a': 100, 'b': 200}
print("Dictionary 1:",d1)
d2 = {'x': 300, 'y': 200}
print("\nDictionary 2:",d2)
d = d1.copy()
d.update(d2)
print("\nMerged Dictionary :\n",d)
```

```
Dictionary 1: {'a': 100, 'b': 200}

Dictionary 2: {'x': 300, 'y': 200}

Merged Dictionary :
 {'a': 100, 'b': 200, 'x': 300, 'y': 200}
```

In [28]:

```
#3. Write a Python program to get a dictionary from an object's fields.
class dictObj(object):
     def __init__(self):
         self.x = 'red'
         self.y = 'Yellow'
         self.z = 'Green'
     def do_nothing(self):
         pass
test = dictObj()
print(test.__dict__)
```

```
{'x': 'red', 'y': 'Yellow', 'z': 'Green'}
```

In [29]:

```
#1. Write a Python program to get the largest number from a list.
list1 = [10, 20, 4, 45, 99,105]
list1.sort()
print("Largest element is:", list1[-1])
```

```
Largest element is: 105
```

In [30]:

```
#2. Write a Python program to get the smallest number from a list.
list1 = [10, 20, 4, 45, 99,0]
list1.sort()
print("Smallest element is:", *list1[:1])
```

```
Smallest element is: 0
```

In [31]:

```
#3. Write a Python program to count the number of strings where the string length is 2
#or more and the first and last character are same from a given list of strings
```

```
def match_words(words):
  ctr = 0

  for word in words:
    if len(word) > 1 and word[0] == word[-1]:
      ctr += 1
  return ctr

print(match_words(['abc', 'xyz', 'aba', '1221','121','xyxab']))
```

```
3
```

In [32]:

```
#4. Write a Python program to add an item in a tuple.
#create a tuple
intTuple = (10, 20, 30, 40, 50)
print("Tuple Items = ", intTuple)

intTuple = intTuple + (70,)
print("Tuple Items = ", intTuple)

intTuple = intTuple + (80, 90)
print("Tuple Items = ", intTuple)

intTuple = intTuple[2:5] + (11, 22, 33, 44) + intTuple[7:]
print("Tuple Items = ", intTuple)
```

```
Tuple Items =  (10, 20, 30, 40, 50)
Tuple Items =  (10, 20, 30, 40, 50, 70)
Tuple Items =  (10, 20, 30, 40, 50, 70, 80, 90)
Tuple Items =  (30, 40, 50, 11, 22, 33, 44, 90)
```

In [33]:

```
#5. Write a Python program to convert a tuple to a string.
def convertTuple(tup):
        # initialize an empty string
    str = ''
    for item in tup:
        str = str + item
    return str

tuple = ('p', 'y', 't', 'h', 'o','n')
str = convertTuple(tuple)
print(tuple)
print(str)
```

```
('p', 'y', 't', 'h', 'o', 'n')
python
```

In [34]:

```
#6. Write a Python program to create the colon of a tuple.
from copy import deepcopy
#create a tuple
tuplex = ("HELLO", 5, [], True)
print(tuplex)
#make a copy of a tuple using deepcopy() function
tuplex_colon = deepcopy(tuplex)
tuplex_colon[2].append(50)
print(tuplex_colon)
print(tuplex)
```

```
('HELLO', 5, [], True)
('HELLO', 5, [50], True)
('HELLO', 5, [], True)
```

In [35]:

```
#7. Write a Python program to unpack a tuple in several variables
tuplex = 4, 8, 3,5
print(tuplex)
n1, n2, n3,n4 = tuplex
#unpack a tuple in variables
print(n1 + n2 + n3+n4)
#the number of variables must be equal to the number of items of the tuple
n1, n2, n3, n4= tuplex
```

```
(4, 8, 3, 5)
20
```

In [36]:
```
#8. Write a Python program to add member(s) in a set.
color_set = set()
print(color_set)
print("\nAdd single element:")
color_set.add("Red")
print(color_set)
print("\nAdd multiple items:")
color_set.update(["Blue", "Green"])
print(color_set)
```

```
set()

Add single element:
{'Red'}

Add multiple items:
{'Red', 'Green', 'Blue'}
```

In [37]:
```
#9. Write a Python program to remove item(s) from set
num_set = set([0, 1, 3, 4, 5])
print("Original set:")
print(num_set)
num_set.pop()
print("\nAfter removing the element from the set:")
print(num_set)
```

```
Original set:
{0, 1, 3, 4, 5}

After removing the element from the set:
{1, 3, 4, 5}
```

In [38]:
```
#10. Write a Python program to create an intersection of sets
A = {2, 3, 5, 4}
B = {2, 5, 100}
C = {2, 3, 8, 9, 10}

print(B.intersection(A))
print(B.intersection(C))
print(A.intersection(C))

print(C.intersection(A, B))
```

```
{2, 5}
{2}
{2, 3}
{2}
```

In [39]:
```
#11. Write a Python program to create a union of sets.
# Python3 program for union() function

set1 = {2, 4, 5, 6}
set2 = {4, 6, 7, 8}
set3 = {7, 8, 9, 10}

# union of two sets
print("set1 U set2 : ", set1.union(set2))

# union of three sets
print("set1 U set2 U set3 :", set1.union(set2, set3))
```

```
set1 U set2 :  {2, 4, 5, 6, 7, 8}
set1 U set2 U set3 : {2, 4, 5, 6, 7, 8, 9, 10}
```

In [40]:
```
#12. Write a Python script to concatenate following dictionaries to create a new one.
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
dic4 = {2.20:8.50}
for d in (dic1, dic2, dic3): dic4.update(d)
print(dic4)
```

```
{2.2: 8.5, 1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

In [41]:
```
#13. Write a Python program to map two lists into a dictionary.
keys = ['red', 'green', 'blue']
values = ['#FF0000','#008000', '#0000FF']
color_dictionary = dict(zip(keys, values))
print(color_dictionary)
```

```
{'red': '#FF0000', 'green': '#008000', 'blue': '#0000FF'}
```

In [42]:
```
#14. Write a Python program to sort a dictionary by key.
color_dict = {'red':'#FF0000',
          'green':'#008000',
          'black':'#000000',
          'white':'#FFFFFF'}

for key in sorted(color_dict):
    print("%s: %s" % (key, color_dict[key]))
```

```
black: #000000
green: #008000
red: #FF0000
white: #FFFFFF
```

In [43]:
```
#15. Write a Python program to get the maximum and minimum value in a dictionary
my_dict = {'x':500, 'y':5874, 'z': 560}

key_max = max(my_dict.keys(), key=(lambda k: my_dict[k]))
key_min = min(my_dict.keys(), key=(lambda k: my_dict[k]))

print('Maximum Value: ',my_dict[key_max])
print('Minimum Value: ',my_dict[key_min])
```

```
Maximum Value:  5874
Minimum Value:  500
```

In [44]:
```
#16. Write a Python program to clone or copy a list.
def Cloning(li1):
    li_copy = li1[:]
    return li_copy

# Driver Code
li1 = [4, 8, 2, 10, 15, 18]
li2 = Cloning(li1)
print("Original List:", li1)
print("After Cloning:", li2)
```

```
Original List: [4, 8, 2, 10, 15, 18]
After Cloning: [4, 8, 2, 10, 15, 18]
```

In [45]:
```
#17. Write a Python program to find the list of words that are longer than n from a given list of words.
def long_words(n, str):
    word_len = []
    txt = str.split(" ")
    for x in txt:
        if len(x) > n:
            word_len.append(x)
    return word_len
print(long_words(5, "Hello Tybcs Welcome to Python Practical"))
```

```
['Welcome', 'Python', 'Practical']
```

In [46]:
```
#18. Write a Python program to unzip a list of tuples into individual lists.
test_list = [('pranjal', 1), ('humera', 2), ('sakshi', 3), ('supriya', 4)]
print ("Original list is : "  )
print(test_list)
res = map(None, *test_list)
print ("Modified list is : " )
print(res)
```

```
Original list is :
```

```
[('pranjal', 1), ('humera', 2), ('sakshi', 3), ('supriya', 4)]
Modified list is :
<map object at 0x06ADF190>
```

In [47]:

```python
# Python code to demonstrate
# Unzip a list of tuples
# using map()

# initializing list of tuples
test_list = [(5, 1), (4, 2), (2, 3), (7, 4)]

# Printing original list
print ("Original list is : ")
print(test_list)

# using map() to
# perform Unzipping
res = map(None, *test_list)

# Printing modified list
print ("Modified list is : " )
print(res)
```

```
Original list is :
[(5, 1), (4, 2), (2, 3), (7, 4)]
Modified list is :
<map object at 0x06ADF2B0>
```

In [48]:

```python
#19. Write a Python program to reverse a tuple.
def Reverse(tuples):
    new_tup = tuples[::-1]
    return new_tup
tuples = ('python')
print(Reverse(tuples))
```

```
nohtyp
```

In [49]:

```python
#20. Write a Python program to convert a list of tuples into a dictionary.
def Convert(tup, di):
    for a, b in tup:
        di.setdefault(a, []).append(b)
    return di

# Driver Code
tups = [('pranjal', 10), ("sakshi", 12), ("humera", 14)]
dictionary = {}
print (Convert(tups, dictionary))
```

```
{'pranjal': [10], 'sakshi': [12], 'humera': [14]}
```

In [50]:

```python
#21. Write a Python program to print a tuple with string formatting.
t = (100, 200, 300,'kalpita')
print('This is a tuple {0}'.format(t))
```

```
This is a tuple (100, 200, 300, 'kalpita')
```

In [51]:

```python
#22. Write a Python program to create a symmetric difference.
set1 = set(["green", "blue","pink"])
set2 = set(["blue", "yellow","purple","green"])
print("Original sets:")
print(set1)
print(set2)
r1 = set1.symmetric_difference(set2)
print("\nSymmetric difference of set1 - set2:")
print(r1)
r2 = set2.symmetric_difference(set1)
print("\nSymmetric difference of set2 - set1:")
print(r2)
```

```
Original sets:
{'pink', 'blue', 'green'}
{'purple', 'yellow', 'blue', 'green'}
```

```
Symmetric difference of set1 - set2:
{'purple', 'pink', 'yellow'}

Symmetric difference of set2 - set1:
{'purple', 'pink', 'yellow'}
```

In [52]:

```python
#23. Write a Python program to check if a given value is present in a set or not.
nums = {1, 3, 5, 7, 9, 11}
print("Original sets(nums): ",nums,"\n")
print("Test if 6 exists in nums:")
print(6 in nums)
print("\nTest if 7 exists in nums:")
print(7 in nums)
print("\nTest if 15 exists in nums:")
print(15 in nums)
```

```
Original sets(nums):  {1, 3, 5, 7, 9, 11}

Test if 6 exists in nums:
False

Test if 7 exists in nums:
True

Test if 15 exists in nums:
False
```

In [53]:

```python
#24. Write a Python program to check if a given set is superset of itself and superset of another given set
A = {4, 1, 3, 5}
B = {6, 0, 4, 1, 5, 0, 3, 5}
print("A.issuperset(B) : ", A.issuperset(B))
print("B.issuperset(A) : ", B.issuperset(A))
```

```
A.issuperset(B) :  False
B.issuperset(A) :  True
```

In [54]:

```python
#25. Write a Python program to check a given set has no elements in common with other given set.
sn1 = {1,2,3}
sn2 = {4,5,6}
sn3 = {3}
print("Original sets:")
print(sn1)
print(sn2)
print(sn3)
print("Check sn1 set has no elements in common with sn2 set:")
print(sn1.isdisjoint(sn2))
print("Check sn1 set has no elements in common with sn3 set:")
print(sn1.isdisjoint(sn3))
```

```
Original sets:
{1, 2, 3}
{4, 5, 6}
{3}
Check sn1 set has no elements in common with sn2 set:
True
Check sn1 set has no elements in common with sn3 set:
False
```

In [55]:

```python
#26. Write a Python program to remove the intersection of a 2nd set from the 1st set.
sn1 = {1,2,3,4,5}
sn2 = {4,5,6,7,8}
print("Original sets:")
print(sn1)
print(sn2)
print("\nRemove the intersection of a 2nd set from the 1st set using difference_update():")
sn1.difference_update(sn2)
print("sn1: ",sn1)
print("sn2: ",sn2)
```

```
Original sets:
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}

Remove the intersection of a 2nd set from the 1st set using difference_update():
```

```
sn1:  {1, 2, 3}
sn2:  {4, 5, 6, 7, 8}
```

In [56]:

```python
#27. Write a Python program to remove duplicates from Dictionary
# Python3 code to demonstrate working of
# Remove duplicate values in dictionary
# Using loop

# initializing dictionary
test_dict = { 5 : 10, 4 : 15, 6 : 20,5:10,4:10}

# printing original dictionary
print("The original dictionary is : ")
print(test_dict)

# Remove duplicate values in dictionary
# Using loop
temp = []
res = dict()
for key, val in test_dict.items():
        if val not in temp:
                temp.append(val)
                res[key] = val

# printing result
print("The dictionary after values removal : " )
print(res)
```

```
The original dictionary is :
{5: 10, 4: 10, 6: 20}
The dictionary after values removal :
{5: 10, 6: 20}
```

In [57]:

```python
#28. Write a Python script to check whether a given key already exists in a dictionary
d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
def is_key_present(x):
  if x in d:
      print('Key is present in the dictionary')
  else:
      print('Key is not present in the dictionary')
is_key_present(5)
```

```
Key is present in the dictionary
```

In [58]:

```python
#29. Write a Python program to sum all the items in a dictionary
def returnSum(myDict):
    list = []
    for i in myDict:
        list.append(myDict[i])
    final = sum(list)
    return final
dict = {'a': 10, 'b':20, 'c':30}
print("Sum :", returnSum(dict))
```

```
Sum : 60
```

In [59]:

```python
#30. Write a Python program to multiply all the items in a dictionary
d = {'a': 2,'b': 3,'c': 6,}
answer = 1
for i in d:
    answer = answer*d[i]
print(answer)
```

```
36
```

In [60]:

```python
#31. Write a Python program to remove a key from a dictionary
myDict = {'a':1,'b':2,'c':3,'d':4}
print(myDict)
if 'a' in myDict:
    del myDict['a']
print(myDict)
```

```
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

{'b': 2, 'c': 3, 'd': 4}

In [ ]: