In [ ]:

```
#Task:

9 7 20 raju kiran somu mahesh123 som2342u raj234esh$

2079
```

In [6]:

```
s = ['9','7','20']
s.reverse()
print("".join(s))
```

```
2079
```

In [7]:

```
d = ['ramu','somu']
k = []
for i in range(len(d)):
    k.append(d[i].title())
print(k)
```

```
['Ramu', 'Somu']
```

In [16]:

```
g = ['mahes34h','s23een546u']
l = []
mm = []
for j in g:
    l = []
    for m in j:
        if m.isalpha():
            l.append(m)
#     print(l)
    mm.append("".join(l))
print(mm)

```

```
['mahesh', 'seenu']
```

In [27]:

```
a=['shnsjm','jgdhh','dfhkj']
print(a)
a.reverse()
print(a)
a.append('aa')
a.sort()
print(a)
print(a.count('jgdhh'))
```

```
['shnsjm', 'jgdhh', 'dfhkj']
['dfhkj', 'jgdhh', 'shnsjm']
['aa', 'dfhkj', 'jgdhh', 'shnsjm']
1
```

In [ ]:

```
import re
p='[a-z]{1,}'
l=['9','7','20','raju','kiran','somu','mahesh123','som2342u','
if re.match(p,l)

```

In [28]:

```
g={3,3,5,'somu','rajesh'}
g
```

Out[28]:

```
{3, 5, 'rajesh', 'somu'}
```

In [31]:

```
g={5,3,10,'omkar','akash'}
(g)
```

Out[31]:

```
{10, 3, 5, 'akash', 'omkar'}
```

In [32]:

```
print(dir(set))
```

```
['__and__', '__class__', '__contains__', '__delattr_
_', '__dir__', '__doc__', '__eq__', '__format__', '_
_ge__', '__getattribute__', '__gt__', '__hash__', '_
_iand__', '__init__', '__init_subclass__', '__ior_
_', '__isub__', '__iter__', '__ixor__', '__le__', '_
_len__', '__lt__', '__ne__', '__new__', '__or__', '_
_rand__', '__reduce__', '__reduce_ex__', '__repr__',
'__ror__', '__rsub__', '__rxor__', '__setattr__', '_
_sizeof__', '__str__', '__sub__', '__subclasshook_
_', '__xor__', 'add', 'clear', 'copy', 'difference',
'difference_update', 'discard', 'intersection', 'int
ersection_update', 'isdisjoint', 'issubset', 'issupe
rset', 'pop', 'remove', 'symmetric_difference', 'sym
metric_difference_update', 'union', 'update']
```

In [38]:

```
s={3,5,5,3,3,3,}
s.add(45)
print(s)
print(type(s))
```

```
{45, 3, 5}
<class 'set'>
```

In [39]:

```
s1 = s.copy()
print(s)
print(s1)
```

```
{45, 3, 5}
{45, 3, 5}
```

In [40]:

```
s={4,2,1,5,7,8}
s1={100,145,123,101,4,2,1}
print(s.difference(s1))
print(s1.difference(s))
print(s)
print(s1)
```

```
{8, 5, 7}
{145, 123, 100, 101}
{1, 2, 4, 5, 7, 8}
{1, 2, 4, 100, 101, 145, 123}
```

In [55]:

```
s={4,2,1,5,7,8}
s1={100,145,123,101,4,2,1}
s=s.difference_update(s1)
print(s1)
print(s)
```

```
{1, 2, 4, 100, 101, 145, 123}
None
```

In [56]:

```
s={4,2,1,5,7,8}
s1={100,145,123,101,4,2,1}
s1=s1.difference_update(s)
print(s)
print(s1)
```

```
{1, 2, 4, 5, 7, 8}
None
```

In [57]:

```
s={4,2,1,5,7,8}
s1={100,145,123,101,4,2,1}
print(s.intersection(s1))
print(s1.intersection(s))
print(s)
print(s1)
```

```
{1, 2, 4}
{1, 2, 4}
{1, 2, 4, 5, 7, 8}
{1, 2, 4, 100, 101, 145, 123}
```

In [64]:

```
s={4,2,1,5,7,8}
s1={100,145,123,101,4,2,1}
s1=s.intersection_update(s1)
print(s)
print(s1)
```

```
{1, 2, 4}
None
```

In [65]:

```
s={4,2,1,5,7,8}
s1={100,145,123,101,4,2,1}
print(s.union(s1))
print(s)
print(s1)
```

```
{1, 2, 4, 5, 100, 7, 8, 101, 145, 123}
{1, 2, 4, 5, 7, 8}
{1, 2, 4, 100, 101, 145, 123}
```

In [66]:

```
s={4,2,1,5,7,8}
s1={100,145,123,101,4,2,1}
s.update(s1)
print(s)
print(s1)
```

```
{1, 2, 4, 5, 100, 7, 8, 101, 145, 123}
{1, 2, 4, 100, 101, 145, 123}
```

In [67]:

```
s={4,2,1,5,7,8}
s1={100,145,123,101,4,2,1}
print(s.symmetric_difference(s1))
print(s)
print(s1)
```

```
{100, 101, 5, 7, 8, 145, 123}
{1, 2, 4, 5, 7, 8}
{1, 2, 4, 100, 101, 145, 123}
```

In [68]:

```
s={4,2,1,5,7,8}
s1={100,145,123,101,4,2,1}
s=s.symmetric_difference(s1)
print(s)
print(s1)
```

```
{100, 101, 5, 7, 8, 145, 123}
{1, 2, 4, 100, 101, 145, 123}
```

Dictionary:

- Collection of key and value pairs
- It can be define as {} and dict()
- It can changes the value by key, Slicing can be done only on key and values
- Ordered type of data is to be stored but it is in key va

lue pairs
- It doesn't allows the duplicate keys

In [72]:

```
h = {'name':['rajesh','somu'],'age':34,'year':'2 year'}
print(h)
print(type(h))
```

```
{'name': ['rajesh', 'somu'], 'age': 34, 'year': '2 y
ear'}
<class 'dict'>
```

In [73]:

```
for i in h.items():
    print(i,end=" ")
```

```
('name', ['rajesh', 'somu']) ('age', 34) ('year', '2
year')
```

In [75]:

```
for j in h.keys():
    print(j,end=" ")
```

```
name age year
```

In [76]:

```
for k in h.values():
    print(k,end=" ")
```

```
['rajesh', 'somu'] 34 2 year
```

In [77]:

```
1 print(dir(dict))
```

```
['__class__', '__contains__', '__delattr__', '__deli
tem__', '__dir__', '__doc__', '__eq__', '__format_
_', '__ge__', '__getattribute__', '__getitem__', '__
gt__', '__hash__', '__init__', '__init_subclass__',
'__iter__', '__le__', '__len__', '__lt__', '__ne__',
'__new__', '__reduce__', '__reduce_ex__', '__repr_
_', '__setattr__', '__setitem__', '__sizeof__', '__s
tr__', '__subclasshook__', 'clear', 'copy', 'fromkey
s', 'get', 'items', 'keys', 'pop', 'popitem', 'setde
fault', 'update', 'values']
```

In [79]:

```
1 print(h.get('name'))
```

```
['rajesh', 'somu']
```

In [80]:

```
1 print(h.get('age'))
```

```
34
```

In [88]:

```
1 print(h)
2 h.pop('age')
3 print(h)
```

```
{'name': ['rajesh', 'somu'], 'age': 34, 'year': '2 y
ear'}
{'name': ['rajesh', 'somu'], 'year': '2 year'}
```

In [91]:

```
print(h)
h.popitem()
print(h)
```

```
{'name': ['rajesh', 'somu'], 'year': '2 year'}
{'name': ['rajesh', 'somu']}
```

In [95]:

```
h['surname']='kiran'
print(h)
```

```
{'name': 'kiran', 'surname': 'kiran'}
```

In [97]:

```
h.setdefault('io')
print(h)
```

```
{'name': 'kiran', 'surname': 'kiran', 'io': None}
```

In [100]:

```
h.fromkeys('name')
```

Out[100]:

```
{'n': None, 'a': None, 'm': None, 'e': None}
```

In [102]:

```
h.update({'io':'raj'})
```

In [106]:

```
print(h)
h.update({'surname':'Surya'})
print(h)
```

```
{'name': 'kiran', 'surname': 'raju', 'io': 'raj'}
{'name': 'kiran', 'surname': 'Surya', 'io': 'raj'}
```

Modules and Packages

In [109]:

```
# print the random values between the given range
import random
def pack(lb,ub,n):
    for i in range(n):
        print(random.randint(lb,ub),end=" ")
pack(1,100,10)
```

```
73 45 73 65 5 14 84 36 72 48
```

File Handling

```
Open File
Doing Operations on File
Closing File
```

In [113]:

```
f = open('Data/file.txt')# By default it is in read mode
print(f.read())
f.close()
print(f.read())
```

...

```
read mode
write mode
append mode
```

In [116]:

```
f = open('Data/file.txt','r')
print(f.read())
f.close()
```

```
I am in anits college
For workshop on Python
For the department of CSE
```

In [121]:

```
f = open('Data/file.txt','w')
f.write('i am an apssdc trainer')
f.close()
```

In [122]:

```
f = open('Data/file.txt','a')
f.write(' Training in the Anits College \n')
f.close()
```

In [124]:

```
with open('Data/file.txt','r') as f:
    f.read()
    print(f.tell())
    f.close()
```

```
80
```

In [126]:

```
with open('Data/file.txt','r') as f:
    f.seek(0)
    print(f.read(9))
```

```
I am an a
```

In [127]:

```
with open('Data/file.txt','r') as f:
    r = f.read()
    print(r.split())
```

```
['I', 'am', 'an', 'apssdc', 'trainer.', 'Training',
'in', 'the', 'Anits', 'College.', 'For', 'the', 'dep
artment', 'of', 'CSE']
```

In [128]:

```
with open('Data/file.txt','r') as f:
    print(f.readlines())
```

```
['I am an apssdc trainer.\n', 'Training in the Anits
College.\n', 'For the department of CSE']
```

In [130]:

```
with open('Data/file.txt','r') as f:
    w = f.readlines()
    for i in w:
        print(i,end="")
```

```
I am an apssdc trainer.
Training in the Anits College.
For the department of CSE
```

In [132]:

```
with open('Data/file.txt','r') as f:
    c = 0
    w = f.readlines()
    for i in w:
        c+=1
    print(c,end="")
```

```
3
```

Functional Programming

In [134]:

```
## List Comprehension
l = []
for i in range(1,10):
    l.append(i)
print(l)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [141]:

```
print([i for i in range(1,10)])
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [145]:

```
print([i for i in (input().split())])
```

```
1 2 3 45 6
['1', '2', '3', '45', '6']
```

In [149]:

```
para = "This is anits college.We are having the Python Worksho
l=[]
for j in para.split():
    l.append(j)
print(l)
```

```
['This', 'is', 'anits', 'college.We', 'are', 'havin
g', 'the', 'Python', 'Workshop']
```

In [151]:

```
print([i for i in para.split()])
```

```
['This', 'is', 'anits', 'college.We', 'are', 'havin
g', 'the', 'Python', 'Workshop']
```

Maps

In [1]:

```
def add(a):
    return a+a
l = list(map(add,range(1,5)))
print(l)
```

```
[2, 4, 6, 8]
```

In [2]:

```
def cube(a):
    return a ** 3
print(set(map(cube,range(1,5))))
```

```
{8, 1, 27, 64}
```

In [ ]:

```

```