## Sorting

Sorting is organization of elements/values in ascending or descending order. This sorting is done using two methods

1. Sort method of list
2. Sorted function

**sort(*, key=None, reverse=False)**

This method sorts the list in place, using only < comparisons between item. Sort method of list is mutable; sorting is done in same list.
Sort method of list by default organizes elements in ascending order.

```
>>> list1=[4,8,1,2,6,3,9,1,4,3]
>>> print(list1)
[4, 8, 1, 2, 6, 3, 9, 1, 4, 3]
>>> list1.sort()
>>> print(list1)
[1, 1, 2, 3, 3, 4, 4, 6, 8, 9]
>>> list1.sort(reverse=True)
>>> print(list1)
[9, 8, 6, 4, 4, 3, 3, 2, 1, 1]
```

**sorted(iterable,key=None,reverse=False)**

Return a new sorted list from the items in iterable.

```
>>> list1=[3,8,1,4,2,9,7,4]
>>> print(list1)
[3, 8, 1, 4, 2, 9, 7, 4]
>>> list2=sorted(list1)
>>> print(list2)
[1, 2, 3, 4, 4, 7, 8, 9]
>>> print(list1)
[3, 8, 1, 4, 2, 9, 7, 4]
>>> list3=sorted(list1,reverse=True)
>>> print(list3)
[9, 8, 7, 4, 4, 3, 2, 1]
```

```
>>> str1="JAVA"
>>> str2=sorted(str1)
>>> print(str1)
JAVA
>>> print(str2)
['A', 'A', 'J', 'V']
>>> list3=["naresh","suresh","amar","ramesh","kishore"]
>>> list4=sorted(list3)
>>> print(list3)
['naresh', 'suresh', 'amar', 'ramesh', 'kishore']
>>> print(list4)
['amar', 'kishore', 'naresh', 'ramesh', 'suresh']
>>> list5=sorted(list3,reverse=True)
>>> print(list5)
['suresh', 'ramesh', 'naresh', 'kishore', 'amar']
>>>
list6=["naresh","suresh","AMAR","ramesh","NARESH","SURESH","RAMESH","
kishore"]
>>> list7=sorted(list6)
>>> print(list6)
['naresh', 'suresh', 'AMAR', 'ramesh', 'NARESH', 'SURESH', 'RAMESH',
'kishore']
>>> print(list7)
['AMAR', 'NARESH', 'RAMESH', 'SURESH', 'kishore', 'naresh', 'ramesh',
'suresh']
>>> list8=sorted(list6,key=str.upper)
>>> print(list8)
['AMAR', 'kishore', 'naresh', 'NARESH', 'ramesh', 'RAMESH', 'suresh',
'SURESH']
```

**Example:**

```
# Write a program to input n elements list and organize elements in
ascending order with using
# sort,sorted



n=int(input("Enter how many values?"))
```

```
list1=[]

for i in range(n):
    value=int(input("Enter value "))
    list1.append(value)

print(f'Before Sorting {list1}')

# Bubble Sorting
for i in range(n):
    for j in range(n-1):
        if list1[j]>list1[j+1]:
            list1[j],list1[j+1]=list1[j+1],list1[j]


print(f'After Sorting {list1}')
```

**Output**

```
Enter how many values?5
Enter value 3
Enter value 5
Enter value 1
Enter value 4
Enter value 2
Before Sorting [3, 5, 1, 4, 2]
After Sorting [1, 2, 3, 4, 5]
```

**Example**

```
# Write a program to input n elements inside list and find 2 maximum
element

n=int(input("Enter how many values?"))

list1=[]
for i in range(n):
    value=int(input("Enter Value "))
    list1.append(value)

list1.sort()
first_max=list1[-1]
c=list1.count(first_max)

print(f'After Sorting {list1}')

print(f'Second Maximum {list1[-(c+1)]}')
```

**Output**

```
Enter how many values?5
Enter Value 4
Enter Value 2
Enter Value 3
Enter Value 5
Enter Value 5
After Sorting [2, 3, 4, 5, 5]
Second Maximum 4
```

**sequence.count(value)**

This returns count of given value (OR) this method returns a given value exist how many times/count

**Example:**

```
>>> list1=[1,2,3,4,7,1,2,1,2,5,3,4,7,8]
>>> list1.count(1)
3
>>> list1.count(5)
1
>>> list1.count(4)
2
>>> list1.count(9)
0
```

**Example:**

```
list1=[1,2,3,4,7,1,2,1,2,5,3,4,7,8]
list2=[]

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

print(list1)
print(list2)

for value in list2:
    c=list1.count(value)
    print(f'{value}--{c}')
```

**Output**

```
[1, 2, 3, 4, 7, 1, 2, 1, 2, 5, 3, 4, 7, 8]
[1, 2, 3, 4, 7, 5, 8]
1--3
2--3
3--2
4--2
7--2
5--1
8--1
```

**Common Operations applied on any sequence**

| Operation | Result |
|---|---|
| x in s | True if an item of *s* is equal to x, else False |
| x not in s | False if an item of *s* is equal to x, else True |
| s + t | the concatenation of *s* and *t* |
| s * n or n * s | equivalent to adding *s* to itself *n* times |
| s[i] | *i*th item of *s*, origin 0 |

| Operation | Result |
|---|---|
| s[i:j] | slice of *s* from *i* to *j* |
| s[i:j:k] | slice of *s* from *i* to *j* with step *k* |
| len(s) | length of *s* |
| min(s) | smallest item of *s* |
| max(s) | largest item of *s* |
| s.index(x[, i[, j]]) | index of the first occurrence of x in *s* (at or after index *i* and before index *j*) |
| s.count(x) | total number of occurrences of x in *s* |

| | |
|---|---|
| **x in s** | **True if an item of *s* is equal to *x*, else False** |

```
>>> list1=[10,20,30,40,50]
>>> 10 in list1
True
>>> 100 in list1
False
```

| | |
|---|---|
| **x not in s** | **False if an item of *s* is equal to *x*, else True** |

```
>>> list2=[100,200,300,400,500]
>>> 100 not in list2
False
>>> 1000 not in list2
True
```

| | |
|---|---|
| **s + t** | **the concatenation of *s* and *t*** |

```
>>> list1=[10,20,30,40,50]
>>> list2=[60,70,80,90,100]
>>> list3=list1+list2
>>> print(list1)
[10, 20, 30, 40, 50]
>>> print(list2)
[60, 70, 80, 90, 100]
p
>>> print(list3)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list4=[*list1,*list2]
>>> print(list4)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

**s * n or n * s   equivalent to adding *s* to itself *n* times**

```
>>> list1=[0]*10
>>> print(list1)
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
>>> list2=[5]*10
>>> print(list2)
[5, 5, 5, 5, 5, 5, 5, 5, 5, 5]
>>> list3=5*[10]
>>> print(list3)
```

len(s)   length of *s*

```
>>> list1=[10,20,30,40,50]
>>> len(list1)
5
```

**Example:**

```
list1=[10,20,30,40,50]

c=0
for value in list1:
    c=c+1
```

```
print(f'Length is {c}')
```

**Output**

```
Length is 5
```

**Example:**

```
# Python | Program to print duplicates from a list of integers
# Given a list of integers with duplicate elements in it.
# The task is to generate another list, which contains only the
duplicate elements.
# In simple words, the new list should contain elements that appear
as more than one.

'''
Examples:

Input : list = [10, 20, 30, 20, 20, 30, 40, 50, -20, 60, 60, -20, -20]
Output : output_list = [20, 30, -20, 60]

'''


list1=[10, 20, 30, 20, 20, 30, 40, 50, -20, 60, 60, -20, -20]
list2=[]
list3=[]

# Storing unique elements
for value in list1:
    if value not in list2:
        list2.append(value)

for value in list2:
    c=list1.count(value)
    if c>=2:
        list3.append(value)

print(list1)
print(list2)
print(list3)
```

**Output**

```
[10, 20, 30, 20, 20, 30, 40, 50, -20, 60, 60, -20, -20]
[10, 20, 30, 40, 50, -20, 60]
[20, 30, -20, 60]
```

**Example:**

```
'''
Python program to find Cumulative sum of a list
The problem statement asks to produce a
new list whose i^{th} element will be equal to the sum of the (i + 1)
elements.

Input : list = [10, 20, 30, 40, 50]
Output : [10, 30, 60, 100, 150]

Input : list = [4, 10, 15, 18, 20]
Output : [4, 14, 29, 47, 67]
'''

list1=[10, 20, 30, 40, 50]
list2=[]
s=0

for value in list1:
    s=s+value
    list2.append(s)

print(list1)
print(list2)
```

**Output**

```
[10, 20, 30, 40, 50]
[10, 30, 60, 100, 150]
```

```
'''
Break a list into chunks of size N in Python

Example
my_list = ['geeks', 'for', 'geeks', 'like',
          'geeky','nerdy', 'geek', 'love',
             'questions','words', 'life']


n=5
[['geeks', 'for', 'geeks', 'like', 'geeky'],
 ['nerdy', 'geek', 'love', 'questions', 'words'],
 ['life']]


'''


my_list = ['geeks', 'for', 'geeks', 'like',
          'geeky','nerdy', 'geek', 'love',
             'questions','words', 'life']
n=5
output_list=[]

start=0
stop=n

while stop<len(my_list):
   t=my_list[start:stop]
   output_list.append(t)
   start=stop
   stop+=n
else:
   output_list.append(my_list[start:])


print(my_list)
print(output_list)
```