>	Greater than operator
	
	>>> list1=[10,5]
	>>> list2=[5,15]
	>>> list1>list2
	True
	>>> list2>list1
	False
	>>> list3=[20,2]
	>>> list1>list2
	True
	>>> list1>list3
	False
	
	Note: these relational operators in application development not used with comparing list but these are used for comparing strings.

How to read or input more than one value inside list using input() function?

```
# write a program to input 5 values inside list

list1=[]
for i in range(5):
    value=int(input("enter any value"))
    list1.append(value)

print(list1)


list2=list(map(int,input("enter 5 values").split()))
print(list2)
```

Output:

```
enter any value10
enter any value20
enter any value30
enter any value40
enter any value50
```

```
[10, 20, 30, 40, 50]
enter 5 values10 20 30 40 50
[10, 20, 30, 40, 50]
```

https://www.hackerrank.com/challenges/find-second-maximum-number-in-a-list/problem?isFullScreen=false

```
if __name__ == '__main__':
   n=int(input())
   list1=list(map(int,input().split()))
   list1.sort()
   m=max(list1)
   c=list1.count(m)
   print(list1[len(list1)-(c+1)])
```

tuple

tuple is an immutable sequence. After creating tuple we cannot do any changes. tuple does not support mutable methods (append, insert, extend , remove, sort, reverse, pop , clear).

Tuples are immutable sequences, typically used to store collections of heterogeneous data (such as the 2-tuples produced by the enumerate() built-in). Tuples are also used for cases where an immutable sequence of homogeneous data is needed (such as allowing storage in a set or dict instance).

Where tuples are used?

1. Tuples are used to represent immutable list
2. Tuples are used to represent elements in set and dictionary
3. Tuple is used by enumerator for generating values.

“tuple” class or data type is used to represent tuple object.

How to create tuple?

Tuples may be constructed in a number of ways:

- Using a pair of parentheses to denote the empty tuple: ()

- Using a trailing comma for a singleton tuple: a, or (a,)
- Separating items with commas: a, b, c or (a, b, c)
- Using the tuple() built-in: tuple() or tuple(iterable)

Q: What is difference between list and tuple?

List	Tuple
List is mutable collection	Tuple is immutable collection
List is created using []	Tuple is created using ()
Mutable are not sharable	Immutables are sharable
Mutable are not hashable	Immutables are hashables
List required extra space for doing mutable operations	Tuple does not required extra space.
List comprehension is allowed	Tuple comprehension is not allowed
Not efficient in immutable operation	Efficient in immutable operations
“list” class or data type is used to represent list object	“tuple” class or data type is used for representing tuple object
List cannot used to represent data in collections like set and dictionary	Tuple can be used to represent data inside collections like set and dictionary

Example:

```
>>> t1=()
>>> print(t1)
()
>>> type(t1)
<class 'tuple'>
>>> t2=(10)
>>> type(t2)
<class 'int'>
>>> t2=(10,)
>>> print(type(t2))
<class 'tuple'>
>>> print(t2)
(10,)
>>> t3=10,
>>> type(t3)
<class 'tuple'>
>>> print(t3)
```

```
(10,)
>>> t4=(10,20,30,40,50)
>>> print(t4)
(10, 20, 30, 40, 50)
type(t4)
<class 'tuple'>
t5=10,20,30,40,50
type(t5)
<class 'tuple'>
print(t5)
(10, 20, 30, 40, 50)
t6=tuple()
print(t6)
()
>>> t7=tuple(range(10,110,10))
>>> t8=tuple([10,20,30,40,50])
>>> t8=tuple("PYTHON")
>>> t9=tuple((10,20,30,40,50))
>>> t10=tuple([10,20,30,40,50])
>>> print(t7,t8,t9,t10,sep="\n")
(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)
('P', 'Y', 'T', 'H', 'O', 'N')
(10, 20, 30, 40, 50)
(10, 20, 30, 40, 50)
```

Python program to find size of tuple?

```
t1=(10,20,30,40,50)
c=0
for value in t1:
   c=c+1

print(c)
print(len(t1))
```

Output:

```
5
```

5

Python Program to find maximum and minimum K elements in tuple

```
k=int(input("enter how many values"))
t1=tuple([int(input()) for i in range(k)])
print(t1)
print(max(t1))
print(min(t1))
```

Output:

```
enter how many values5
10
20
30
40
50
```