```
## collections
### 1.List
### 2.Tuple
### 3.Dictionary
### 4.Sets
```

```
### 1.List
####   1.It is used to store the collection of data
####   2.It is ordered and changable
####   3.List is denoted by square brakets i.e   [_]
####   4.List is similiar to arrays
####   5.List index starts from size-1
```

```
* 1.How to create a empty list
*    syntax:
*    variable_name ==[_]
```

In [2]:

```
l1=[]
type(l1)
```

Out[2]:

```
list
```

```
#### How to assign values to the list
####  variable_name=[value1,value2...]
```

In [4]:

```
list1=[10,20,"welcome",10.2,"srit"]
print(list1)
```

```
[10, 20, 'welcome', 10.2, 'srit']
```

```
#### How to access values from the list
#### Syntax: List_name[index value]
```

In [7]:

```
print(list1[4])
```

```
srit
```

```
* how to access with in the range of values
* syntax:list_name[lowerbound,upperbound]
```

In [10]:

```
list1[0:3]
```

```
[10, 20, 'welcome']
```

In [21]:

```
list1[-4:]
```

Out[21]:

```
[20, 'welcome', 10.2, 'srit']
```

In [80]:

```
# to access the values from the list using loop
# for variable in list_name:
    #print(variable)
for i in list1:
    print(i)
```

```
10
20
welcome
10.2
srit
```

```
* len() -> To find the length of the list
* Syntax : len(list_name)
```

In [24]:

```
len(list1)  #length of the list
```

Out[24]:

```
5
```

In [25]:

```
list2=[10,20,30,2,18,50]
max(list2)    # maximum of the list
```

Out[25]:

```
50
```

In [27]:

```
print(dir(list),end=" ") # directory for the list
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__di
r__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__ge
titem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__init_
subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__',
'__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmu
l__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook
__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop',
'remove', 'reverse', 'sort']
```

In [29]:

```
min(list2) # minimum in the list
```

Out[29]:

```
2
```

In [30]:

```
sum(list2)  #sum of the elements in the list
```

Out[30]:

```
130
```

- append: To add the element at the end of the list
- syntax: list_name.append(value)

In [37]:

```
l1=[10,20,15,4,25,80]
l1.append(100)
print(l1)
```

```
[10, 20, 15, 4, 25, 80, 100]
```

In [39]:

```
# insert : It is used to insert an element in specific position
# syntax: list_nam.insert(position,element)

l1.insert(2,101)
print(l1)
```

```
[10, 20, 101, 15, 4, 25, 80, 100]
```

In [40]:

```
# count : To count the occurance of a element
# syntax : list_name.count(value)


l1.count(101)
```

Out[40]:

```
1
```

In [81]:

```
#extend : to add mutiple values to the list
#syntax : list_name.extend([val1,val2,......])
l1.extend([5,6,7])
print(l1,end=" ")
```

```
[20, 5, 6, 7]
```

In [52]:

```
# index: it returns the index of the given value
# syntax: list_name,index(element)

l1.index(300)
```

Out[52]:

```
12
```

In [68]:

```
# pop:To delete an element from the list
#syntax: list_name.pop() -> this deletes the last element by default
#synntax: list_name.pop(index_position) -> this delete the last element

l1.pop()
l1
l1.pop(2)
l1
```

Out[68]:

```
[10, 20, 4]
```

In [71]:

```
#remove : to remove the element from the list
#syntax : list_name.remove(element_name)

l1.remove(10)
l1
```

Out[71]:

```
[20]
```

In [73]:

```
# to print the reversal of the list elements
# syntax : list_name.reverse()

l3=['sai','charan','karnatakam']
print(l3)
l3.reverse()
print(l3)
```

```
['sai', 'charan', 'karnatakam']
['karnatakam', 'charan', 'sai']
```

In [76]:

```
# sort: to print the elements of list in ascending or decending
# syntax: list_name.sort() -> it returs the elements in ascending
# syttax: list_name.sort(reverse=True) -> it retunrs the elements in descending order

l4=[1,3,5,4,2]
l4.sort()
print(l4)
l4.sort(reverse=True)
print(l4)
```

```
[1, 2, 3, 4, 5]
[5, 4, 3, 2, 1]
```

In [77]:

```
#copy : It is used to copy the data from one list to another list
#syntax :new_listname=old_listname.copy()
l5=l4.copy()
print(l5)
```

```
[5, 4, 3, 2, 1]
```

In [79]:

```
# clear: To clear the data from the list
# syntax: list_name.clear()

l4.clear()
l4          #list => l4 is cleared
```

Out[79]:

```
[]
```

In [87]:

```
key=int(input("enter the key"))
li=[10,20,30,10,20,40]
print(li.count(key))
```

```
enter the key20
2
```

In [1]:

```
# to create a list by taking input from the user

l=input().split()
```

```
10 20 30 40 50 60
```

In [ ]:

```
# Tasks:
# 1.Create a function to find the largest element in the list
# 2.create a function to find the lowest element in the list
# 3.create a function to find the second largest in the list
          #l =[20,30,40,50,80,10,90]
# 4.create a function to find thhe third largest in list
          #l=[10,20,30,10,10,20,30,40]
# 5.create a function to find the kth largest of the list
          #l=[10,50,20,30,60,80,100,500,500,89,47]
          #key=2
# 6.create a function to find the kth lowest of the list
```

In [9]:

```
# 1.Create a function to find the largest element in the list
li=[20,30,40,50,80,10,90]
def findinglargest(li):
    result=max(li)
    return result
findinglargest(li)
```

Out[9]:

```
90
```

In [10]:

```
# 2.create a function to find the lowest element in the list

def findingsmallest(li):
    result=min(li)
    return result
findingsmallest(li)
```

Out[10]:

```
10
```

In [11]:

```
# 3.create a function to find the second largest in the list
         #l =[20,30,40,50,80,10,90]

def secondlarge(l):
    l.sort()
    return l[-2]
secondlarge(li)

```

Out[11]:

```
80
```

In [12]:

```
# 4.create a function to find thhe third largest in list
         #l=[10,20,30,10,10,20,30,40]

def thirdlarge(l):
    l.sort()
    return l[-3]
thirdlarge(li)

```

Out[12]:

```
50
```

In [5]:

```
# 5.create a function to find the kth largest of the list
lis=[10,50,20,30,60,80,100,500,500,89,47]
lt=[]
for i in lis:
    if i not in lt:
        lt.append(i)
key=int(input())
def nthlarge(lt,key):
    lt.sort()
    return lt[-key]
nthlarge(lt,key)
```

```
2
```

Out[5]:

```
100
```

In [7]:

```
# 6.create a function to find the kth lowest of the list
key=int(input())
def nthsmall(lt,key):
    lt.sort()
    return lt[key-1]
nthsmall(lt,key)
```

```
1
```

Out[7]:

```
10
```

In [ ]:

```

```