In [1]:
```
# A list is a collection of values. Remember, it may contain different types of

# To define a list, you must put values separated with commas in square bracket
```

In [2]:
```
list1=[]
print(type(list1))
print(list1)
```

```
<class 'list'>
[]
```

In [3]:
```
print([12,23,"python"])
```

```
[12, 23, 'python']
```

In [4]:
```
print(type([12,23,"python"]))
```

```
<class 'list'>
```

In [5]:
```
# define list
list1=["apple", "mango","cherry","strawberry","pineapple"]
print(list1)
```

```
['apple', 'mango', 'cherry', 'strawberry', 'pineapple']
```

In [6]:
```
type(list1)
```

Out[6]:
```
list
```

In [7]:
```
# length of list
len(list1)
```

Out[7]:
```
5
```

In [8]:
```
list1=["apple", "mango","cherry","strawberry","pineapple","apple","papaya"]
print(list1)
```

```
['apple', 'mango', 'cherry', 'strawberry', 'pineapple', 'apple', 'papaya']
```

In [9]:
```
len(list1)
```

Out[9]:
```
7
```

In [10]:
```
# Access List Items
list1[1]
```

Out[10]:
```
'mango'
```

In [11]:
```
print(list1[-3:-2])
```

```
['pineapple']
```

In [12]:
```
print(list1[-4:-1])
```

```
['strawberry', 'pineapple', 'apple']
```

In [13]:
```
# Check if Item Exists
if "apple" in list1:
    print("apple is in list1" )
else:
    print("no")
```

```
apple is in list1
```

In [14]:
```
# Check if Item Exists
s=["orange","apple"]
for x in s:
    if x in list1:
        print(f"{x} is in list1" )
    else:
        print(f"{x} is not in list1")
```

```
orange is not in list1
apple is in list1
```

In [15]:
```
print(list1)
```

```
['apple', 'mango', 'cherry', 'strawberry', 'pineapple', 'apple', 'papaya']
```

In [16]:
```
list1[1]="Mango"
print(list1)
```

```
['apple', 'Mango', 'cherry', 'strawberry', 'pineapple', 'apple', 'papaya']
```

In [17]:
```
list1[-2:]=["Orange","blueberry"]
print(list1)
```

```
['apple', 'Mango', 'cherry', 'strawberry', 'pineapple', 'Orange', 'blueberr
y']
```

In [18]:
```
# add items in a list
list1.append("watermelon")
print(list1)  # add element to the end of the list
```

```
['apple', 'Mango', 'cherry', 'strawberry', 'pineapple', 'Orange', 'blueberr
y', 'watermelon']
```

In [19]:
```
list1.insert(1,"grapes")
print(list1)  # add elemement at specified index or position
```

```
['apple', 'grapes', 'Mango', 'cherry', 'strawberry', 'pineapple', 'Orange',
'blueberry', 'watermelon']
```

In [21]:
```
list2=["kiwi","Book","Lotus"]
```

In [24]:
```
list1.extend(list2)  # add list of elements at the end of the list
```

In [25]:
```
print(list1)
```

```
['apple', 'grapes', 'Mango', 'cherry', 'strawberry', 'pineapple', 'Orange',
'blueberry', 'watermelon', 'kiwi', 'Book', 'Lotus', 'kiwi', 'Book', 'Lotus']
```

In [27]:
```
n=int(input("enter the value of n: "))
l=[]
for i in range(n):
    l.append(i)
print(l)
```

```
enter the value of n: 10
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [29]:
```
n=int(input("enter the value of n: "))
l=[]
for i in range(1,n+1):
    l.append(i)
print(l)
```

```
enter the value of n: 10
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

In [32]:
```
list1.sort()  # sort list by order
print(list1)
```

```
['Book', 'Book', 'Lotus', 'Lotus', 'Mango', 'Orange', 'apple', 'blueberry',
'cherry', 'grapes', 'kiwi', 'kiwi', 'pineapple', 'strawberry', 'watermelon']
```

In [33]:
```
list1.reverse()  # reverse list by an order
print(list1)
```

```
['watermelon', 'strawberry', 'pineapple', 'kiwi', 'kiwi', 'grapes', 'cherry',
'blueberry', 'apple', 'Orange', 'Mango', 'Lotus', 'Lotus', 'Book', 'Book']
```

In [35]:
```
# sum of elements of list
n=int(input("enter the value of n: "))
l=[]
for i in range(1,n+1):
    l.append(i)
print(l)
print(sum(l))
```

```
enter the value of n: 10
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
55
```

In [36]:
```
type(l)
```

Out[36]:
```
list
```

In [37]:
```
n=int(input("enter the value of n: "))
l=[]
for i in range(1,n+1):
    l.append(i)
sum=0
for x in l:
    sum+=x
print(sum)
```

```
enter the value of n: 10
55
```

In [38]:
```
l.insert(20,400)
```

In [39]:
```
print(l)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 400]
```

In [40]:
```
#  if index is not available the insert add element at the end of the list
```

In [41]:
```
# delete the element from the list
```

In [42]:
```
print(list1)
```

```
['watermelon', 'strawberry', 'pineapple', 'kiwi', 'kiwi', 'grapes', 'cherry',
'blueberry', 'apple', 'Orange', 'Mango', 'Lotus', 'Lotus', 'Book', 'Book']
```

In [43]:
```
list1.pop()
```

Out[43]:
```
'Book'
```

In [45]:
```
print(list1)
```

```
['watermelon', 'strawberry', 'pineapple', 'kiwi', 'kiwi', 'grapes', 'cherry',
'blueberry', 'apple', 'Orange', 'Mango', 'Lotus', 'Lotus', 'Book']
```

In [46]:
```
list1.remove("pineapple")
print(list1)
```

```
['watermelon', 'strawberry', 'kiwi', 'kiwi', 'grapes', 'cherry', 'blueberry',
'apple', 'Orange', 'Mango', 'Lotus', 'Lotus', 'Book']
```

In [47]:
```
n=int(input("emter the limit: "))
l1=[]
for i in range(n):
    x=input()
    l1.append(x)
print(l1)
```

```
emter the limit: 5
23
er
45
67
er
['23', 'er', '45', '67', 'er']
```

In [48]:
```
l1.remove('er')
print(l1)
```

```
['23', '45', '67', 'er']
```

In [49]:
```
# length of elements from a list
length=[]
for x in list1:
    length.append(len(x))
print(length)
```

```
[10, 10, 4, 4, 6, 6, 9, 5, 6, 5, 5, 5, 4]
```

In [50]:
```
length.pop(2)
```

Out[50]:
```
4
```

In [51]:
```
print(length)
```

```
[10, 10, 4, 6, 6, 9, 5, 6, 5, 5, 5, 4]
```

In [52]:
```
# delete method
del length[0:2]
print(length)
```

```
[4, 6, 6, 9, 5, 6, 5, 5, 5, 4]
```

In [53]:
```
del length
```

In [54]:
```
print(length)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[54], line 1
----> 1 print(length)

NameError: name 'length' is not defined
```

In [55]:
```
# clear method
l1.clear()
```

In [56]:
```
print(l1.clear())
```

```
None
```

```
The del keyword can also delete the list completely.
The del keyword also removes the specified index
The clear() method empties the list.
The list still remains, but it has no content.
```

In [57]:
```
print(list1)
```

```
['watermelon', 'strawberry', 'kiwi', 'kiwi', 'grapes', 'cherry', 'blueberry',
'apple', 'Orange', 'Mango', 'Lotus', 'Lotus', 'Book']
```

In [58]:
```
l=[]
for i in list1:
    if "ry" in i:
        l.append(i)
print(l)
```

```
['strawberry', 'cherry', 'blueberry']
```

In [61]:
```
# list comprehension
l1=[]
l2=[l1.append(i) for i in list1 if "e" in i]

print(l1)
```

```
['watermelon', 'strawberry', 'grapes', 'cherry', 'blueberry', 'apple', 'Orang
e']
```

In [63]:
```
# Python program to interchange first and last elements in a list
n=int(input("enter the limit: "))
l1=[]
for i in range(n):
    x=input()
    l1.append(x)
print(l1)
l1[0],l1[-1]=l1[-1],l1[0]
print("after the interchanging: ")
print(l1)
```

```
enter the limit: 5
34
45
56
67
32
['34', '45', '56', '67', '32']
after the interchanging:
['32', '45', '56', '67', '34']
```

In [64]:
```
# Python program to swap two elements in a list
i1=int(input("enter index1: "))
i2=int(input("enter index2: "))
n=int(input("enter the limit: "))
l1=[]
for i in range(n):
    x=input()
    l1.append(x)
print(l1)
l1[i1],l1[i2]=l1[i2],l1[i1]
print("after swap elemets")
print(l1)
```

```
enter index1: 2
enter index2: 3
enter the limit: 5
34
45
54
32
12
['34', '45', '54', '32', '12']
after swap elemets
['34', '45', '32', '54', '12']
```

In [70]:
```
# reversing a list
l1.reverse()
print(l1)
```

```
['12', '54', '32', '45', '34']
```

In [71]:
```
l1.sort()
print(l1)
```

```
['12', '32', '34', '45', '54']
```

In [72]:
```
l1.sort(reverse=True)
```

In [73]:
```
print(l1)
```

```
['54', '45', '34', '32', '12']
```

In [76]:
```
l1=[23,34,43,23,13]
ls=l1[::-1]
print(ls)
```

```
[13, 23, 43, 34, 23]
```

In [77]:
```
# copy list
ls=[12,23,34,45,56,65,54,43,32]
ls1=ls.copy()
print(ls1)
```

```
[12, 23, 34, 45, 56, 65, 54, 43, 32]
```

In [95]:
```
# Count occurrences of an element in a list
ls=[12,23,34,45,56,65,54,43,32,12,12,23,23,23,45]
ls.count(12)
```

Out[95]:
```
3
```

In [101]:
```
# Find sum and average of List in Python
ls=[12,23,34,45,56,65,54,43,32,12,12,23,23,23,45]
count=0
for x in ls:
    count+=x
sum=count
avg=count/len(ls)
print(count)
print(avg)
```

```
502
33.46666666666667
```

In [ ]:
```

```

In [2]:
```
L = [4, 5, 1, 2, 9, 7, 10, 8]
counts = sum(L)

# finding average
avg = counts/len(L)

print("sum = ", counts)
print("average = ", avg)
```

```
sum =  46
average =  5.75
```

In [ ]:
```

```

In [10]:
```
# Sum of number digits in List
ls=[12,23,34,45,56,65,54,43,32,12,12,23,23,23,45]
s=[]
for i in ls:
    sum=0
    for x in str(i):
        sum+=int(x)
    s.append(sum)
print(s)
```

```
[3, 5, 7, 9, 11, 11, 9, 7, 5, 3, 3, 5, 5, 5, 9]
```

In [11]:
```
# Multiply all numbers in the list
l=[1,2,3,4,5]
prod=1
for x in l:
    prod*=x
print(prod)
```

```
120
```

In [12]:
```
# find smallest number in a list
ls=[12,23,34,45,56,65,54,43,32,12,12,23,23,23,45]
max(ls)
```

Out[12]:
```
65
```

In [13]:
```
min(ls)
```

Out[13]:
```
12
```

In [16]:
```
# ind second largest number in a list
list1 = [10, 20, 20, 4, 45, 45, 45, 99, 99]
l=list(set(list1))
l.sort()
print(l[-2])
```

```
45
```

In [18]:
```
# even number in list
list1 = [10, 20, 20, 4, 45, 45, 45, 99, 99]
for x in list1:
    if x%2==0:
        print(f"{x}",end=' ')
```

```
10 20 20 4
```

In [20]:
```
# odd number in list
list1 = [10, 20, 20, 4, 45, 45, 45, 99, 99]
for x in list1:
    if x%2!=0:
        print(f"{x}",end=' ')
```

```
45 45 45 99 99
```

In [23]:
```
#find count of even and odd number in list
ls=[12,23,34,45,56,65,54,43,32,12,12,23,23,23,45]
even=0
odd=0
for x in ls:
    if x%2==0:
        even+=1
    else:
        odd+=1
print(even,": even numbers in list")
print(odd,": odd numbers in list")
```

```
7 : even numbers in list
8 : odd numbers in list
```

In [26]:
```
# Remove multiple elements from a list
ls.remove(12)
```

In [27]:
```
ls
```

Out[27]:
```
[23, 34, 45, 56, 65, 54, 43, 32, 12, 12, 23, 23, 23, 45]
```

In [28]:
```
del ls[2:5]
ls
```

Out[28]:
```
[23, 34, 54, 43, 32, 12, 12, 23, 23, 23, 45]
```

In [35]:
```
# Multidimensional Lists
l=[[12,3,4],[2,3,4,5],[2,4,6]]
```

In [32]:

In [ ]:

In [ ]: