

In [5]:

```
#question 1
Friends = ["Kashish","Puneet","Ashish","Ankit","Arvind"]      #List of friends
print(Friends)
```

['Kashish', 'Puneet', 'Ashish', 'Ankit', 'Arvind']

In [7]:

```
#question 2
Friends = ["Kashish","Puneet","Ashish","Ankit","Arvind"]      #List of friends
for friend in Friends:
    print(friend+" is good friend")
```

Kashish is good friend
Puneet is good friend
Ashish is good friend
Ankit is good friend
Arvind is good friend

In [8]:

```
#Question 3
vehicles = ['Bugati','Honda','Hero','Maruti']                  #List of cars
for vehicle in vehicles:
    print("I would like to to own a "+vehicle)
```

I would like to to own a Bugati
I would like to to own a Honda
I would like to to own a Hero
I would like to to own a Maruti

In [9]:

```
#Question 4
guests = ['Mukesh Ambani','Ratan Tata','Bill Gates']          #List of guests
for guest in guests:
    print("I would like to invite "+guest )
```

I would like to invite Mukesh Ambani
I would like to invite Ratan Tata
I would like to invite Bill Gates

In [14]:

```

#Question 5
guests = ['Mukesh Ambani', 'Ratan Tata', 'Bill Gates']
for guest in guests:
    print("I would like to invite "+guest )
print(guests[2]+" cannot make at party")
guests.remove('Bill Gates')
guests.append('Gautam Adani')
ist
print(guests)
for guest in guests:
    print("I would like to invite "+guest )

```

#remove the guest in list
#adding the guest at the end in the L

```

I would like to invite Mukesh Ambani
I would like to invite Ratan Tata
I would like to invite Bill Gates
Bill Gates cannot make at party
['Mukesh Ambani', 'Ratan Tata', 'Gautam Adani']
I would like to invite Mukesh Ambani
I would like to invite Ratan Tata
I would like to invite Gautam Adani

```

In [60]:

```

#Question 6
guests = ['Mukesh Ambani', 'Ratan Tata', 'Bill Gates']
print("I informing all guest that found bigger dinner tables ")
guests.insert(0, 'Jeff Bezos')
different position in the List
guests.insert(2, 'Warren Buffett')
guests.append('Sergey Brin')
for guest in guests:
    print("I would like to invite "+guest)
print("\nlist after adding guest")
print(guests)

```

#adding the guest in

```

I informing all guest that found bigger dinner tables
I would like to invite Jeff Bezos
I would like to invite Mukesh Ambani
I would like to invite Warren Buffett
I would like to invite Ratan Tata
I would like to invite Bill Gates
I would like to invite Sergey Brin

```

```

list after adding guest
['Jeff Bezos', 'Mukesh Ambani', 'Warren Buffett', 'Ratan Tata', 'Bill Gate
s', 'Sergey Brin']

```

In [7]:

```
#question 7
guests = ['Jeff Bezos', 'Mukesh Ambani', 'Warren Buffett', 'Ratan Tata', 'Bill Gates',
'Sergey Brin']
print("I can invite only two people for dinner")
pop_guest = guests.pop()
pop_guest = guests.pop()
pop_guest = guests.pop()
pop_guest = guests.pop()
for guest in guests:
    print("I would like to invite "+guest)
guests.remove('Jeff Bezos')
guests.remove('Mukesh Ambani')
print("list is empty")
print(guests)
```

```
I can invite only two people for dinner
I would like to invite Jeff Bezos
I would like to invite Mukesh Ambani
list is empty
[]
```

In [61]:

```

#question 8
places = ['New York', 'Mumbai', 'London', 'Goa', 'Los Angeles']
print(places)
print("\nsorted list")                #short the list temporary
print(sorted(places))
print("\noriginal list")
print(places)
print("\n reverse ordered list")
print(sorted(places,reverse=True))    #reverse the list temporary in aplhabetical
print("\noriginal list")
print(places)
print("\nreverse of the list")
places.reverse()
print(places)
print("\noriginal list")
places.reverse()                      #reverse the list temporary
print(places)
print("\nsort the list permanently")
places.sort()
print(places)
print("\nreverse of list in aplhabetical permanently")
places.sort(reverse = True)
print(places)

```

```
['New York', 'Mumbai', 'London', 'Goa', 'Los Angeles']
```

```
sorted list
```

```
['Goa', 'London', 'Los Angeles', 'Mumbai', 'New York']
```

```
original list
```

```
['New York', 'Mumbai', 'London', 'Goa', 'Los Angeles']
```

```
reverse ordered list
```

```
['New York', 'Mumbai', 'Los Angeles', 'London', 'Goa']
```

```
original list
```

```
['New York', 'Mumbai', 'London', 'Goa', 'Los Angeles']
```

```
reverse of the list
```

```
['Los Angeles', 'Goa', 'London', 'Mumbai', 'New York']
```

```
original list
```

```
['New York', 'Mumbai', 'London', 'Goa', 'Los Angeles']
```

```
sort the list permanently
```

```
['Goa', 'London', 'Los Angeles', 'Mumbai', 'New York']
```

```
reverse of list in aplhabetical permanently
```

```
['New York', 'Mumbai', 'Los Angeles', 'London', 'Goa']
```

In [31]:

```
#question 9
places = ['New York', 'Mumbai', 'London', 'Goa', 'Los Angles'] #List of places
print("length of the list")
len(places) #Len function for print
ing length of list
```

length of the list

Out[31]:

5

In [38]:

```
#question 10
pizzas = ['Pepperoni', 'Margherita', 'Greek']
for food in pizzas:
    print(food)
    print("I like "+food+" pizza")
print("\nI would like the chilli,paneer,potato pizza\n")
pizzas = ['chilli', 'potato', 'paneer']
for food in pizzas:
    print("I would like "+food)
    print("I really love pizza\n")
```

Pepperoni

I like Pepperoni pizza

Margherita

I like Margherita pizza

Greek

I like Greek pizza

I would like the chilli,paneer,potato pizza

I would like chilli

I really love pizza

I would like potato

I really love pizza

I would like paneer

I really love pizza

In [40]:

```
#question 11
animals = ['Dog', 'Cat', 'Horse']
for animal in animals:
    print(animal)
    print("A "+animal+" would make a great pet")
print("common of these animals is that, they are pet animal")
print("Any of these animals would make great pet")
```

```
Dog
A Dog would make a great pet
Cat
A Cat would make a great pet
Horse
A Horse would make a great pet
```

In [43]:

```
#question 12
print("number from 1 to 20\n")
for num in range(1,21,1):
    print(num)
```

number from 1 to 20

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

In [63]:

```
#question 13
million = []
for num in range(1,1000001):
    million.append(num)
print(million)
```

IOPub data rate exceeded.

The notebook server will temporarily stop sending output to the client in order to avoid crashing it.

To change this limit, set the config variable

`--NotebookApp.iopub_data_rate_limit`.

Current values:

NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)

NotebookApp.rate_limit_window=3.0 (secs)

In [51]:

```
#question 14
min(million)
```

Out[51]:

1

In [52]:

```
max(million)
```

Out[52]:

1000000

In [53]:

```
sum(million)
```

Out[53]:

500000500000

In [54]:

```
#question 15
print("Odd number from 1 to 20")
for num in range(1,20,2):           # using loop for printing number
    print(num)
```

odd number from 1 to 20

1
3
5
7
9
11
13
15
17
19

In [55]:

```
#question 16
three = list(range(3,31,3))
for multiple in three:             #using loop for printing numbers multiple of
    3
    print(multiple)
```

3
6
9
12
15
18
21
24
27
30

In [58]:

```
#question 17
cubes = []
for value in range(1,11):
    cube = value**3                #using loop for printing cube of numbers
    cubes.append(cube)
print(cubes)
```

[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]

In [59]:

```
#question 18
cubes = [value**3 for value in range(1,11)]    #using the comprehensions
print(cubes)
```

[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]

In []: