Operator

```
-----------
a=50
b=30
print("a+b",a+b)
print("a-b",a-b)
print("a*b",a*b)
print("a/b",a/b)
print("a//b",a//b)
print("a**b",a**b)
print("a%b",a%b)
```

Additional (concatanation "+")(+,-,/,*,//,%)

```
print("Java"+"Python")
#print("java"+2)  #TypeError
#Repeation(*)
print("java"*5)
print(5*"java")
print((7*5)*"java")
```

#Relational(>,<,>=,<=)

```
------------------
print(10>5)
print(10<5)
print("python">"java")
print(ord('p'))
print(ord('j'))
print(True>2)
print(ord('a'))
print(ord('P'))
print(ord('J'))
print(ord('A'))
```

 # Nesting atlest one condition is false result is false otherwise is true

```
-----------------------------------------------------------------
---------
print(5>2>1)
print(5>2>8)
```

Equality

```
print(10==10)
print(10!=100)
#Logical
# and
print(20 and 30 )
print(50 and 10)
# or
print(20 or 30 )
print(50 or 10)
```

#not Boolean

```
print(10 and 20)
print(30 and 10)
```

#membership operator (in , not in) : check the availabity

```
a=10,20,30
print(10 in a )
a=10,20,30
print(100 not in a )
```

refrence operator (is ,is not) a : if both ref variable pointing to same obj give true othervise false

```
a=(10,20,30)
b=(10,20,30)
c=100
d=100
x=[10,20,30]
y=[10,20,30] # mutable
e={10,20,30}
f={10,20,30}
print(a is b)
print(a is b)
print(a is not b)
print(e is not f)
print(id(a)) ------------------id of a
```

conditional statement----------

```
if--------------------
Syntax:
    if condition:
        true body
Syntax:
    if condition:
        true body
Syntax: if- else
    if condition:
        true body
    else:
        false body
Syntax: if- elif-else
    if condition:
        action-1
    elif:
        action-2
    else:
        default
 Ex.------------ to take input from user---------------
user1=int(input("Enter your number :"))
user2=int(input("Enter your number :"))
```

```
if user1<user2:
    print("Small number",user1)
else:
    print(user2)
Ex. Two number from user positive or negitive----------------------------
---
Ex-----------------------------------
user1=int(input("Enter your number :"))
if user1<0:
    print("Negitive Number: ",user1)
else:
    print("positive number :",user1)
Ex.cost and sell--------------------------------
-------------------------------------------------
a=int(input("Enter Your Cost:"))
b=int(input("Enter Your Sell:"))
if a>b:
    print("You have loss",a)
else :
    print("You have profit")
Ex.login ------------------------------------
user="Ajay"
password="ajay123"
user_name=input("Enter your username :")
password_name=input("Enter your password :")
if user_name==user and password_name==password:
    print("Login Succesfull")
else:
    print("incorrect username and password"
-------------------------------------------
user=int(input("Enter Your Number : "))
if user==0:
    print("Zero")
elif user==1:
    print("one")
elif user==2:
    print("two")
elif user==3:
    print("Three")
elif user==4:
    print("Four")
elif user==5:
    print("Five")
elif user==6:
    print("Six")
elif user==7:
    print("Seven")
elif user==8:
    print("Eight")
elif user==9:
    print("Nine")
else:
    print("please enter between 0-9")
-----------------------------------------------------
user=int(input("Enter Your number : "))
if user % 3==0 and user % 5==0 :
    print("hii hello")
elif user % 5==0 :
```

```
    print("hello")
elif user % 3==0 :
    print("hii")
-----------------------------------------------
nested if else-----------------------------------------<>
syntax:
age=int(input("Enter Your Age:"))
gen=input("Enter Gender:")
if gen=="male":
    if age>=18:
        print("Eligible for voting")
    else:
        print("Not Eligible ")
else:
    if age>=18:
        print("Eligible for voting")
    else:
        print("Not Eligible")
-------------------------------------------------------------------
```

for loop--<>

```
# for loop
for i in range(10):
    print(i)
print("**************************")
for i in range(20,40):
    print(i)
print("**************************")
for i in range(2,21,2):
    print(i)
for i in range(20):
    print("Ms.Dhoni")
-----------------------------------
# for loop
for i in range(1,51,2):
    print(i)
print("******************************")
user =int(input("Enter Number :"))
for i in range(1,11):
    print(i*user)
print("******************************")
total=0
for i in range(1,11):
    total=total+i
print("Addition is",+total)
-------------------------------------------------------------------
```

while loop --

```
import time
print("lets start")
run=0
while run!=6 :
    time.sleep(2)
    print("Running........")
    un=input("Do you continue Yes or No :")
    if un=="yes":
        print("You completed   all level",run)
        run=run+1
```

```
    else:
        if un=="no":
            print("Ok")
            break
if run == run:
    print("You congrased Achieved this level ",run)
-----------------------------------
'''
n=int(input("Enter Number : "))
for user in range(n):
   print(" * "*n)
n=int(input("Enter Number : "))
for i in range(n):
   print(" A "*n)
n=int(input("Enter Number : "))
for i in range(n):
   print(" 1 "*n)
n=int(input("Enter Number : "))
for i in range(n):
   print(" 1 2 3 "*n)
n=int(input("Enter Number : "))
for i in range(n):
   print(" A B C "*n)
n=int(input("Enter Number : "))
for i in range(n):
   print((str("1 2 3")+" "))
n=int(input("Enter Number : "))
for i in range(n):
   print((chr(n-i+64)+" ")*n)
n=int(input("Enter Number : "))
for i in range(n):
   for j in range(n):
      print(chr(j+65),end=" ")
   print()
n=int(input("Enter Number : "))
for i in range(n):
   print((chr(n-i+64)+" ")*n)
print()
'''
n=int(input("Enter Number : "))
for i in range(n,0,-1):
   for j in range(n,0,-1):
      print(chr(j+64),end=" ")
   print()
ex.
while True:
    print("1.addition")
    print("2.sub")
    print("3.multiplication")
    choice =(input("Select Option 1/2/3: "))
    if choice in ["1","2","3"]:
        num1=int(input("Enter number :"))
        num2 = int(input("Enter number :"))
        if choice=="1":
            print(num1 + num2)
        elif choice=="2":
            print(num1-num2)
        elif choice == "3":
```

```
            print(num1 * num2)
    else:
        print("Inavalid")
----------------------<>
'''
n=int(input("Enter Number : "))
for user in range(n):
   print(" * "*n)
n=int(input("Enter Number : "))
for i in range(n):
   print(" A "*n)
n=int(input("Enter Number : "))
for i in range(n):
   print(" 1 "*n)
n=int(input("Enter Number : "))
for i in range(n):
   print(" 1 2 3 "*n)
n=int(input("Enter Number : "))
for i in range(n):
   print(" A B C "*n)
n=int(input("Enter Number : "))
for i in range(n):
   print((str("1 2 3")+" "))
n=int(input("Enter Number : "))
for i in range(n):
   print((chr(n-i+64)+" ")*n)
n=int(input("Enter Number : "))
for i in range(n):
   for j in range(n):
      print(chr(j+65),end=" ")
   print()
n=int(input("Enter Number : "))
for i in range(n):
   print((chr(n-i+64)+" ")*n)
print()
n=int(input("Enter Number : "))
for i in range(n,0,-1):
   for j in range(n,0,-1):
      print(chr(j+64),end=" ")
   print()
n=int(input("Enter Number : "))
for i in range(n):
   for j in range(i):
      print(j+1,end=" ")
   print()
'''
--------------------------------------------------------------------------
------------------
```

#byte : represet as number in array and it cannot change

```
value(immutable)
num = [10,20,30]
b = bytes(num)
print(b[0])
print(type(b))
num = [10,20,255]
b = bytes(num)
print(b[0])
print(type(b))
```

bytearray :it is similar to the byte only diffrence is it can change value

```
num = [10,20,30]
b = bytearray(num)
print(b[0])
print(type(b))
num = [10,20,30]
b = bytearray(num)
print(b[0])
b[0]=111
print(b[0])
print(type(b))
```

==========

String:

```
s="Python"
print(s[0])
print(s[1])
print(s[2])
print(s[3])
print(s[4])
print("**********************")
print(s[-1])
print(s[-2])
print(s[-3])
print(s[-4])
```

```
name=str(input(" Enter Character Here :"))
i=0
for c in name:
    print(f"{c} positive index{i} and negitive index{i-len(name)}")
    i=i+1
```

---

string :accesing characters

```
#slice operator
# s="python is simple to learn"
# print(s[0:6])
# print(s[6:10])
# print(s[10:16])
# print(s[:9])
# print(s[0:])
# print(s[0:4:2])
# print(s[-4:-1])
user=input("Enter Name : ")
print(user[-1::-1])
#function of string
#len():
s="python"
print(len(s))
#index():
s="pythontont"
print(s.index("t"))
print(s.index("t",3))
print(s.index("t",6,8))
print(s.index("o",5))
```

```
#print(s.index("d")) ValueError: substring not found
#find():similar to index only diffrence is string is not found to return
negitive value
s="python is easy  to learn"
print(s.find("t"))
print(s.find("t",3))
print(s.find("t",6,8))
print(s.find("o",5))
print(s.find("z"))
#count():
s="misshappiness"
print(s.count("i"))
print(s.count("i",3))
print(s.count("s",3))
print(s.count("s",1,8))
#enumerate():
s="python is easy  to learn"
s=enumerate(s)
print(tuple(enumerate(s)))
-------------------------
s1="java"
s2="python"
s3=""
for i in range(len(s1)):
    s3+=s1[i]+s2[i]
s3 +=s2[len(s1):]
print(s3)
------------------------------------------------------------
#strip():removing space
s= "  python  "
print(s.lstrip())
print(s.rstrip())
print(s.strip())
print("**********************************")
s= "@@@python###"
print(s.lstrip("@"))
print(s.rstrip("#"))
print(s.lstrip("@").rstrip("#"))
print(s.strip())
print("**********************************")
s= "python"
print(s.upper())
print(s.lower())
print(s.title())
print(s.title())
print(s.capitalize())
print(s.swapcase())
#split(): spliting string into token
s="python is easy to learn"
l=s.split()
print(l)
s="pythoniseasyoflearn"
l=s.split()
print(l)
s="python*is*easy*of*learn"
l=s.split("*")
print(l)
dob="14-08-2024"
```

```
l=dob.split("-")
print(l)
url="www.google.com"
l=url.split(".")
print(l)
print("************************************")
#join(): joining token  into string
s=["hello","how","are","you"]
l= " ".join(s)
print(l)
print(("************************************"))
#replace(): replacing character from anothr string
s=" Ms Dhoni Plays Football"
l=s.replace("Football","Cricket")
print(l)
print("************************")
#isalnum(): checking case of string like a-z,A-Z,0-9
s="python123"
l=s.isalnum()
print(l)
#isalpha(): checking case of string like a-z,A-Z
s="python"
l=s.isalnum()
print(l)
#isdigit():checking case of string like 0-9
s="123"
l=s.isalnum()
print(l)
s="python and PYTHON or Python"
l=s.lower()
print(l)
s="python and PYTHON or Python"
l=s.upper()
print(l)
s="python and PYTHON or Python"
l=s.title()
print(l)
s=" "
l=s.isspace()
print(l)
s= "a2b3c4"
l=s.replace("a2b3c4","abc123")
print(l)
s= "a2b3c4"
print(s[::2],end='')
print(s[1::2])
-----------------------------------<>
without using slice operator to reverse programme
# # s=input("enter:")
# # l=""
# # for i in range(len(s)-1,-1,-1):
# #     l +=s[i]
# # print(l)
-------------------------------------------------------------------------
-----
-----------------------------------------------------------------------
```

----------List()----------

```
it is mutable
we can create multiple type data to to store in single variable
# list
l=[10,20,30,10.4,"Raj",True]
print(l)
print(type(l))
print("****************************************")
```

Way of create list <> ----------

```
l=[10,20,30]
print(l)
print(type(l))
print("****************************************")
```

#Empty list

```
l=[]
print(l)
print(type(l))
print("****************************************")
```

#using list() function

```
l=list()
print(l)
print(type(l))
print("****************************************")
```

#using split() function

```
s="python is simple to learn"
l=s.split()
print(l)
print(type(l))
print("****************************************")
```

 # using eval() function

```
s=eval(input("Enter Expression : "))
print(s)
print(type(s))
output:
        Enter Expression : [10,29,36]
        [10, 29, 36]
        <class 'list'>
```

 # using while loop

```
l=[10,20,30]
i=0
while i<len(l):
    print(l[i])
    i+=1
```

 # Accessing element using list

```
l=[10,20,30,40,50]
print(l[0])
print(l[1])
print(l[2])
print(l[3])
print(l[4])
```

Accessing element using slice operator

```
l=[10,20,30,40,50]
print(l[0:3])
print(l[-1:])
print(l[0:-1])
```

#adding element inside list using append() function

```
l=[10,20,30,40]
print(l)
l.append(50)
print(l)
```

 # add elment at soecified index number using insert() function

```
l=[10,20,30,40]
print(l)
l.insert(1,15)
print(l)
l=[10,20,30,40]
print(l)
l.insert(-1,35)
print(l)
```

 # Add two list in one list using extend() function

```
l=["Ms dhoni","Messi","Ronaldo","virat Kohli"]
l2=["Cricket","Football","Football","Cricket"]
l.extend(l2)
print(l)
```

#********** **ADD** ------> append() , insert() , extend()

 # Remove element at end of index number and also remove as using index number

```
l=[10,20,30,40,50]
print(l)
print(l.pop(0))
print(l.pop())
print(l.pop(-1))
#l.pop(9) IndexError: pop index out of range
print(l)
```

#element remove at specific number using remove() function

```
l=[10,20,30,40,50]
print(l)
l.remove(30)
l.remove(50)
```

```
#l.remove(89)  ValueError: list.remove(x): x not in list
print(l)
```

all elment remove from list using clear() function

```
l=[10,20,30,40,50]
print(l)
l.clear()
print(l)
```

#********** Remove Element From List ----------> pop() ,remove() ,clear()

#********** ADD Element From List ----------> append() , insert() , extend()

```
# l=[]
# for i in range(31):
#     if "2" in str(i):
#         print(f"number is {i}")
#         i+=i
# print(i)
```

#index(): return index number of first occurrence element present inside list

 # if specified element is not present will get value error

```
l=[10,20,30,40,50,20]
print(l.index(20))
print(l.index(20,3))
print(l.index(20,3,6))
#print(l.index("raj")) #ValueError: 'raj' is not in list
```

#count(): return total number of occurrence of specified element present inside list

```
l=[10,20,30,40,34,34,34,34,20,50,20]
print(l)
print(l.count(20))
```

#sort(): data are in sequence manner

```
l=[8,3,5,72,4,6]
l.sort()
print(l)
fruits=["banana","apple","cherry","orange","mango"]
fruits.sort()
print(fruits)
```

#reverse list of sort() function

```
fruits=["banana","apple","cherry","orange","mango"]
```

```
fruits.sort(reverse=True) # reverse DNSO
print(fruits)
l=[8,3,5,72,4,6]
l.sort(reverse=True)
print(l)
s="kal Chutti hai"
l=s.split()
l_s=" "
for i in range(len(l)-1,-1,-1):
    l_s +=l[i]+" "
b=l_s.strip()
print(l_s)
s="kal Chutti hai"
l=s.split()
out=" ".join(l[::-1])
print(out)
-------------------------
l=[1,2,3,2,4,5,6,3,2,3,8,2,8,2,2,6,2]
a=[]
for i in l:
    if i not in a:
        a.append(i)
print(a)
l=["Raj","Ajay","Raj","Ajay","Ravi","Suresh","Ravi"]
a=[]
for i in l:
    if i not in a:
        a.append(i)
print(a)
------------------------------------------------------------
# agent
while True:
    user=input("Enter Last Name :")
    if "d" in str(user):
        print("ok")
        break
    else:
        print("Your not agent")
    user2=input("Enter Fav Actor : ")
    if "r" in str(user2):
         print("ok")
    else:
        print("Your not agent")
    user3=input("Enter Lcky Number :")
    if "7" in str(user3):
        print("ok")
    else:
        print("Your not agent")
    user4=input("Enter your fav dish :")
    if len(user4)>7:
        print("ok")
        print("Welcome Your Are Agent")
        break
    else:
        print("Your not agent")
```

```
----------------------------
```
 # cloning we are using two type
1) using slice
2) copy() function

#using slice oprerator

```
l=[10,20,30]
print(l)
l2=l[::]
l2[0]=777
print(l2)
print(l)
```

 # using copy() function

```
l=[10,20,30]
print(l)
l2=l.copy()
l2[0]=888
print(l2)
print(l)
print("*************************")
```

 # list aliasing : giving another name to the existing list object is called LA

```
l=[10,20,30]
print(l)
l2=l
l2[0]=999
print(l2)
print(l)
l3=l2.copy()
l3[2]=1000
print(l3)
print(l2)
l4=l[::]
print(l4)
print(l)
============================
```
 # List Comparing

```
l=[10,20,30]
l2=[10,20,30]
l3=["Ajay","Ram","Sanjay"]
l4=["Ajay","Ram","Sanjay"]
print(l==l2)
print(l3==l4)
```

#nested list

```
l=[10,20,["Ajay","Ram","Sanjay"],30]
print(l)
```

```
print(l[0])
print(l[2][0])
-------------------------------------------------------------------------
---------
name="Krishna"
even=[]
odd=[]
for i in range(len(name)):
    if i % 2 == 0:
        even.append(name[i])
    else:
        odd.append(name[i])
print("even :","".join(even))
print("odd :","".join(odd))
l="krishnakrisaahhhhnaaa"
a=[]
for i in l:
    if i not in a:
        a.append(i)
print(a)
# list comparision: it is compact way to create list from any iterable
object based on condition
out=[]
for i in range(10):
    out.append(i)
print(out)
# compact  code
double_num=[i*i for i in range(10) ]
print(double_num)
name=["Krishna","Ajay","Ram","Sanjay","Sumit"]
out=[i[0] for i in name]
print(out)
-----------------------------------------
s="python is simple to learn and easy"
out=[item for word in s.split() for item in (word.upper(),len(word))]
print(out)
```

---

tuple():----------

```
it similar to list
# l=(10,20,30)
# a,b,c=l
# print(a)
# print(b)
# print(c)
# print(10 in l)
#
#
# print(l[:])
# print(l[1:])
# print(l[:-1])
```

---

set {} ----------

```
#insertion is not preserved
```

```
#duplcate value not allowed
#indexing and slicing is not allowed
#set is used to represet {}

s={1,2,5,7,2,4}
print(s)
print(type(s))

#empty set
a=set()
print(a)
print(type(a))
l={1,3,3,4,5,36,4,7,2,1,7,9,3,4,6}
p=list(set(s))
print("duplicate remove list : ",p)
print(type(p))

#add(): adding element into set and not iterable objet allowed
s={10,20,30}
print(s)
s.add("AAA")
s.add(1)
s.add(10.4)
print(s)
#update (x,y,z) : add element to set and iterable obj is allowed
s={10,20,30}
s.update(["AAA","BBB","CCC"],(10,20,30),"Ajay")
print(s)
# remove(): remove element from set
s={10,20,30}
print(s)
s.remove(10)
print(s)
# s.remove(100) #KeyError: 100
print(s)
# pop() : remove element from list
s={10,20,30}
print(s)
s.pop()
print(s)
s=set()
s.pop()
print(s)
#discard() : remove specified element from set but not give error
s={10,20,30}
print(s)
s.discard(10)
s.discard(1000)
print(s)
#clear(): remove all element from set
s={10,20,30}
print(s)
s.clear()
print(s)
#mathmatical opration on set
#-------------------------------------------
#union(): return all element present in both set and denoted by " | " pie
symbol
```

```
s1={10,20,30,40}
s2={30,40,50,60}
print(s1.union(s2))
print(s1 | s2)
print("``````````````````````````")
#intersection(): return common element from both set and denoted by " & "
s1={10,20,30,40}
s2={30,40,50,60}
print(s1.intersection(s2))
print(s1 & s2)
print("``````````````````````````")
# diffrence(): return elment which present in set but not in set2 and
denoted by " - "
s1={10,20,30,40}
s2={30,40,50,60}
print(s1.difference(s2))
print(s1 - s2)
print("``````````````````````````")
# write program common name present in name
# name1=str(input("Enter name : "))
# name2=str(input("Enter another name : "))
#
# out=set(name1) & set(name2)
# print(out)
# write program vowels present in name
# vowels=("a","e","i","o","u")
# name=input("Enter name : ")
# l=set(name)
# print("vowels in name is : ", l.intersection(vowels) )
user=input("Enter name : ")
if len(user)==len(set(user)):
    print("heterogram")
else:
    print("not heterogram")
-----------------------------------------
# frozenset(): it is read only version of set
s1={10,20,30,40}
fs=frozenset(s1)
print(fs)
#fs.add  AttributeError: 'frozenset' object has no attribute 'add'
#-----------------------------------------------------------------
# dict() :
# how to create dictionary
d={"apple":100,
   "banana":200,
   "orange":300
}
print(d)
print(type(d))
# empty dict ():
d={}
print(d)
# empty dict (): using dict() constructer
d=dict()
print(d)
# how to add value pairs inside dictionary
#syntax: value=dict_name[key]
d={}
```

```
d["apple"]=200
d["cherry"]=300
print(d)
d["apple"]=500 # if the key is duplicate then old value will be replace
with new value
# how to read  value from dictionary
# syntax : value=dict_name[key]
d={}
d["apple"]=200
d["mango"]=300
print(d)
v=d["apple"]
print(v)
print(d["mango"])
#print(d["banana"]) KeyError: 'banana'
students={}
user=int(input("Enter number of student: "))
for i in range(user):
   name=input("enter the student name".format(i+1))
   marks=float(input("enter the student marks".format(name)))
   students[name]=marks
print("\n student data")
print(students)
print(type(students))
------------------------------------------------------------------------
----------
user=["hEllo","how","aRe","you"]
for i in user:
    if i[1].isupper():
       print(i)
print("``````````````````````````````````````")
input="abc123DEF456ghiJKLaa222D"
lower=""
upper=""
digit=""
for i in input:
    if i.islower() :
        lower=lower+i
    elif i.isupper():
        upper=upper+i
    elif i.isdigit():
        digit=digit+i
print(lower," & length is: ",len(lower))
print(upper," & length is: ",len(upper))
print(digit," & length is: ",len(digit))
freinds=["Ajay","Ravi","Sanjay","Rahul"]
for i in freinds:
    if len(i) % 2==0:
        print("Freind is : ",i,": it is divisible by  :",len(i))
================================================================
# print("``````````````````````")
# # update() : it is update value in dic
#
# l={100:"toys",200:"cloth",600:"fruits"}
# l[100]="Shoes"
# print(l)
#
# #or
```

```
# l={100:"toys",200:"cloth",600:"fruits"}
# l2={300:"one",900:"two",500:"three"}
#
# l.update(l2)
# print(l)
#
# print("```````````````")
# #delete() : to deleate key-value pairs in dict
#
# l={100:"toys",200:"cloth",600:"fruits"}
# del l[100]
# print(l)
#
# # function of dictionary:
# #-----------------------------------------------------------------
#
# #get(): return specified key  but if key is not present it give none
#
# l={100:"toys",200:"cloth",600:"fruits"}
# print(l.get(100))
# print(l.get(700)) # none
# print(l.get(600))
#
# #get(key,default): return specified key  but if key is not present it
give none but it return defaul value
# l={100:"toys",200:"cloth",600:"fruits"}
# print(l.get(100))
# print(l.get(700,"hello")) # none but give default value show default
value
# print(l.get(600))
#
# print("```````````````````````")
# #keys(): return list of keys present in dict
# l={100:"toys",200:"cloth",600:"fruits"}
# print(l.keys())
# #or
# for k in l.keys():
#     print(k)
#
# print("```````````````````````")
#
# #values(): return list of values present in dict
# l={100:"toys",200:"cloth",600:"fruits"}
# print(l.values())
# #or
# for k in l.values():
#     print(k)
#
# print("```````````````````````")
# #items(): return list of key-values present in dict
#
# l={100:"toys",200:"cloth",600:"fruits"}
# print(l.items())
# #or
# for k in l.items():
#     print(k)
#
# print("```````````````````")
```

```
#
# #clear(): remove all key-value from dict
# l={100:"toys",200:"cloth",600:"fruits"}
# l.clear()
# print(l)
#
# print("``````````````````````")
# #pop[key]: remove specified key value pairs and return associated that
with key
# l={100:"toys",200:"cloth",600:"fruits"}
# l.pop(100)
# #l.pop(700) #KeyError: 700
# print(l)
#
# #popitem(): remove rendom key-value pairs from dict
# l={100:"toys",200:"cloth",600:"fruits"}
# l.popitem()
# print(l)
#
#
# print("``````````````````````````")
#
# #copy(): copy of an list
# l={100:"toys",200:"cloth",600:"fruits"}
# v=l.copy()
# print(v)
#
# patients = {}
#
# while True:
#     choice = input("\n1. Add\n2. Update\n3. Delete\n4. Clear\n5.
Display\n6. Exit\nEnter choice: ")
#
#     if choice == '1':
#         name = input("Name: ")
#         patients[name] = {'Age': input("Age: "), 'Weight':
input("Weight: ")}
#         print("Patient added!")
#
#     elif choice == '2':
#         name = input("Enter Name to update: ")
#         if name in patients:
#             patients[name]['Age'] = input("New Age: ")
#             patients[name]['Weight'] = input("New Weight: ")
#             print("Patient updated!")
#         else:
#             print("Patient not found!")
#
#     elif choice == '3':
#         name = input("Enter Name to delete: ")
#         if name in patients:
#             del patients[name]
#             print(f"Patient {name} deleted!")
#         else:
#             print("Patient not found!")
#
#     elif choice == '4':
#         patients.clear()
```

```
#         print("All patients cleared!")
#
#     elif choice == '5':
#         if not patients:
#             print("No patients found!")
#         else:
#             for name, info in patients.items():
#                 print(f"Name: {name}, Age: {info['Age']}, Weight:
{info['Weight']}")
#
#     elif choice == '6':
#         print("Exiting...")
#         break
#
#     else:
#         print("Invalid choice!")
# user=eval(input("Enter Value :"))
# sum_values=sum(user.values())
# print(user.values())
# print(sum_values)
# name=input("Enter name : ")
# vowels=("aeiou")
#
# z={}
# for i in name:
#     if i in vowels:
#         z[i]=z.get(i,0)+1
# print(z)
------------------------------Function()---------------------------------
-------------------------------------------------------------------->
--------------------------------------------------------------------------
-------------------------------------------------------------------->
Function(): function is group of  predefine statements
Type of Function :
1) Pre define function : which comes with python libraries
 EX.--- print(), type(), id() , input(), int()
2) User define Function : function which is define by programmers based
on business requirement is called user defiend
We can define function using 2 keywords
1) def-------------> man
2) return-----------.optional
EX.
def sayhello():
    print("Hello Freinds..... Good Afternoon")
sayhello()
sayhello()
def add(num1,num2):
    print("addition is :",num1+num2)
add(10,10)
# def f1(name):
#     return name.replace("-","_")
# user=input("Enter string:")
# print(f1(user))
# def KababToSnake(string):
#     s=string.replace("-","_")
#     print(s)
#
# KababToSnake("python-is-simple")
```

```
#
# def KababToSnake(string):
#     s=string.split("-")
#     l="_".join(s)
#     print(l)
#
# KababToSnake("python-is-simple")
-------------------------- 23-09-2024
# def square(num):
#     return num*num
#
# user=int(input("Enter :"))
# print(square(user))
print("positional arguments:")

#positional arguments(): no of argument and position of argument must be
same,if we change position result may change
def add(num1,num2):
    print(num1)
    print(num2)
add(10,20)
print("*****************")
add(20,10)
print("************************")
#keyword argument(): passing value of parameter iusing variable name
print("keyword argument():")
def resister(name,fees,subject):
    print("name of student: ",name)
    print("fees of student: ", fees)
    print("subject of student: ", subject)
resister("jay",5000,"python")
print("**********************")
resister(name="sanjay",subject="java",fees=6000)
print("**************************************")
#mixing argument(): mixing with positional argument and keyword argument
def f1(a,b,c):
    print(a)
    print(b)
    print(c)
f1(10,20,30) # position
print("**********")
f1(a=20,c=20,b=60) # keyword
print("*********")
# if we mix positional argument with keyword argument then first start
with positional arguments after that keyword argumnt
print("mixing Element")
f1(10,20,c=45) #mixing
# f1(a=23,23) #invalid
# Default argument (): if we provide value will be consider otherwise
take default value
def f1(a="jay"):
    print("good afternoon",a)
f1()
f1("ram")
# if we are passing mix default and non-default argument then first non
default then default argumnet
# def f1(name,a=1,b=0,c=9) # valid start with non-default argumnt
#     f1()
```

```
#
# def f2(a=1,b=2,c) # not valid
print("*****************")
def add(n1=0,n2=0,n3=0,n4=0):
    total=n1+n2+n3+n4
    print("Total addition is :",total)
add(1,2,3)
-------------------------------------------------------------------------23-
09-2024
#return() : fumction can take argument and process business logic and
return result by using return keyword
# def f1(a):
#     return a*a
# print(f1(5))
def monkey_troble(smile1,smile2):
    return smile1==smile2
print(f"{monkey_troble(True,True)} : In Trouble Both are smiling")
print(f"{monkey_troble(False,False)} : In Trouble Both are not smiling")
print(f"{monkey_troble(True,False)} : Not Trouble one smile and one not
smile")
# def sleep_in(weakdays,vacation):
#     return not weakdays or vacation
# print(sleep_in(True,True))
# print(sleep_in(True,False))
# print(sleep_in(False,False))
# print(sleep_in(False,True))
def speed_cough(speed,birthday):
    if speed>85 and birthday==True or speed>80 and birthday!=True:
        return 2
    elif speed<=65 and birthday==True or speed<=60 and birthday!=True:
        return 0
    elif speed>60 and birthday==True or speed>60 and birthday!=True:
        return 2
print(speed_cough(65,True))
print(speed_cough(85,False))
print(speed_cough(45,True))
-----------------------------------------------------
#varible number of  argument() : calling the function passing 0 and to
any number of argument
# if we take first b then *a result will get when having emty passed then
give type error :poisitional argument require
# def f1(b,*a):
#     print("f1 is exe")
# f1(10,20,30)
# f1()
#if we take first b then *a result will get when having emty passed then
give type error :keyword argument require
# def f1(*a,b):
#     print("f1 is exe")
# f1(10,20,30)
# f1()
#Keword-varible argumet (kargs)
# def f1(**a):
#     print(a)
#
# f1(a=10,b=20,c=30)
# def add(*n):
#     v=0
```

```
#
#     for i in n:
#         v=v+i
#     print(f"Total  is :{v}")
#     print("******************")
#
# add()
# add(1)
# add(1,2)
# add(1,2,3)
# add(1,2,3,4)
#
#
#
# def add(*n):
#     v=0
#     for i in  n:
#         if i == 15:
#             break
#         v=v+i
#
#
#     print(f"Total  is :{v}")
#     print("******************")
#
# add(5,15,4)
# add(1)
# add(1,2)
# add(1,15,2,3)
# add(1,2,15,4)
#return() : funtion can take argument and process business logic and
return result by using return keyword
# def f1(a):
#     return a*a
# print(f1(5))
# def monkey_troble(smile1,smile2):
#     return smile1==smile2
#
# print(f"{monkey_troble(True,True)} : In Trouble Both are smiling")
# print(f"{monkey_troble(False,False)} : In Trouble Both are not
smiling")
# print(f"{monkey_troble(True,False)} : Not Trouble one smile and one not
smile")
#
# def sleep_in(weakdays,vacation):
#     return not weakdays or vacation
# print(sleep_in(True,True))
# print(sleep_in(True,False))
# print(sleep_in(False,False))
# print(sleep_in(False,True))
# def speed_cough(speed,birthday):
#
#     if speed>85 and birthday==True or speed>80 and birthday!=True:
#         return 2
#     elif speed<=65 and birthday==True or speed<=60 and birthday!=True:
#         return 0
#     elif speed>60 and birthday==True or speed>60 and birthday!=True:
#         return 2
```

```
#
#
# print(speed_cough(65,True))
# print(speed_cough(85,False))
# print(speed_cough(45,True))
# def speed_cought(speed,birhday):
#
#     if birhday:
#         speed=speed-5
#
#     if speed <= 60:
#         return ("You get one ticket")
#     elif 61 <= speed <= 80:
#         return ("You get two ticket")
#     else:
#         return ("You get Big Ticket")
#
# print(speed_cought(61,True))
# print(speed_cought(61,False))
# print(speed_cought(81,True))
# print(speed_cought(81,False))
# def f1(name,n):
#     a=""
#     for i in range(n):
#         a=a+name
#     return a
#
#
# print(f1("Ajay",3))
200
def f1():
    a=10  # Local Variable
    b=20
    print(a+b)
    print(globals()['a']+globals()['b'])
f1()
# Annonymous  Function : function wihout name called as annonymouus
function just use for instance and
# declare Annonymous  Function using lamda function
#syntax : lamda input_list:expression
# def f1(n):
#     return n*n
# print("squre of 5 is :",f1(5))
#
#
# rv=lambda a:a*a
# print("squre of 6 is :",f1(6))
# v=lambda name:len(name)
# print("length of name: ",v("Ajay"))
-------------------------------------01-10-2024-----------
#map(): if we want perform some common operation to each and every
element present inside seq then go for map
#syntax: map(f1,seq) where fun is to define logic and seq is
list,tuple,string......
# l=["Ajay","Jay","Roy","Duregesh"]
#
# def f1(name):
#     return name+" Devloper"
```

```
#
# rv=list(map(f1,l))
# print(rv)
# l=["Ajay","Jay","Roy","Duregesh"]
#
# v=lambda name:name+" Devloper"
# rv=list(map(v,l))
# print(rv)
l=[2,3,4,5,6]
s=lambda n:n*n
rv=list(map(s,l))
print(rv)
l=[2,3,4,5,6,7,8,9,10,11,12,13,14]
def f1(name):
    if name%2==0:
        return name
rv=list(filter(f1,l))
print(rv)
rv=tuple(filter(lambda n:n%2!=0,l))
print(rv)
rv=set(filter(lambda n:n>5,l))
print(rv)
to print even no.char
l=["Ajay","Jay","Roy","Duregesh"]
rv=list(filter(lambda l:len(l)%2==0,l))
print(rv)
#to print having end with "a" char
l=["Ajay","Jay","Roy","Duregesha","oja"]
rv=list(filter(lambda l:l[-1]=="a",l))
print(rv)
#to print length should be greter than 5
l=["Ajay","Jay","Roy","Durgesh,ViratKohli"]
rv=list(filter(lambda l:len(l)>5,l))
print(rv)
------------------------------------------------------------------------
------
def decore(func):
    def inner(name):
        if name=="krishana":
            print(f"hello {name} Bad Morning")
        else:
            func(name)
    return inner
@decore
def sayhello(name):
    print(f"hello {name} Good Morning")
sayhello("Ajay")
sayhello("krishana")
sayhello("Durgesh")
------------------------------------------------------------------------
---------- 03-10-2024----------------
#decorator : it can take function as argument and return same function
with extended function
# def decore(func):
#     def inner(name):
#         if name=="BBB":
#             print(f"hello {name} Bad Morning")
#         else:
```

```
#             func(name)
#     return inner
#
#
# @decore
# def sayhello(name):
#     print(f"hello {name} Good Morning")
#
# sayhello("AAA")
# sayhello("BBB")
# sayhello("CCC")
# def decore(fun):
#     def smartdiv(a,b):
#         if b==0:
#             print("Zero is Not Divisible")
#         else:
#             fun(a,b)
#     return smartdiv
#
#
# @decore
# def division(a,b):
#     print(a/b)
#
# division(10,2)
# division(10,0)
# division(50,5)
l=[]
num=int(input("Enter No:"))
for i in range(num,0,-1):
        l.append(i)
print(l)
```