*#program*def even(list1):  
 list2=[]  
 for i in list1:  
 if i % 2 == 0:  
 list2.append(i)  
 return list2  
  
list2 = even([1,5,7,4,6,8,10,12])  
print("Even numbers are" ,list2)  
  
*#program*def add(\*numbers):  
 total=0  
 for n in numbers:  
 total=total+n  
 print("Sum is ",total)  
  
add()  
add(4,5,6,7)  
add(20,30,40,50)  
  
*# #Program*def fact(n):  
 if n ==0:  
 return 1  
 else:  
 return n\*fact(n-1)  
print("factorial of a number ",fact(7))  
  
 *#Program -lambda function*l=[2,5,6,10,12,14,18]  
even\_no= list(filter(lambda x: x % 2 ==0,l))  
print("Even numbers are ",even\_no)  
  
*#program - ATM program*amount1=100000  
while True:  
 print("1 Deposit")  
 print("2 Withdrawal")  
 print("3 Balance")  
 a = input("Enter the option")  
 if a =="1":  
 s=int(input("Enter the Deposit amount"))  
 amount1 = amount1+s  
 print("total balance",amount1)  
 elif a =="2":  
 f = int(input("Enter the Withdrawal amount"))  
 amount1 = amount1 - f  
 print("Total balance ",amount1)  
 elif a =="3":  
 print("Existing balance", amount1)  
  
 b=input("Do you want to continue ?(yes/no)")  
 if b == "no":  
 print("Thank you")  
 break  
 elif b =="yes":  
 continue

*#Program - SWap*a=int(input("Enter the first value "))  
b=int(input("Enter the second value "))  
print("Original value of a and b is :",a,"",b)  
a = a+b  
b = a-b  
a = a-b  
print(" After swapping a is :",a,"b is :" ,b)  
  
*#Program - occurrences of each character*str1="Hello world"  
print("occurrences of H :", str1.count("H"))  
print("occurrences of e :", str1.count("e"))  
print("occurrences of l :", str1.count("l"))  
print("occurrences of o :", str1.count("o"))  
print("occurrences of w :", str1.count("w"))  
print("occurrences of r :", str1.count("r"))  
print("occurrences of d :", str1.count("d"))  
  
*#program- Anagram strings*b=("LisTen AcT")  
b1=b.lower()  
a=("SiLeNt CAT")  
a1 = a.lower()  
a1 = [a1[i] for i in range(0,len(a1))]  
a1.sort()  
b1 = [b1[i] for i in range(0,len(b1))]  
b1.sort()  
if a1 == b1:  
 print("Given strings are Anagram")  
else:  
 print("Given strings are not Anagram")  
  
  
*#program -4. Write a python program to reverse each word of a given string*def sentence(s1):  
 words=s1.split(" ")  
 newwords=[word[::-1] for word in words]  
 newsentence=" ".join(newwords)  
 return newsentence  
  
s1="Python Concept Of The Day"  
print(sentence(s1))  
  
*#program-substring*def substr(str1,n):  
 for i in range(n):  
 temp= " "  
 for j in range(i,n):  
 temp += str1[j]  
 print(temp)  
str1="test"  
substr(str1,len(str1))  
print(str1)  
  
*#rpogram -print pattern*for i in range(1,6):  
 for j in range(5,0,-1):  
 if(i ==j):  
 print("\*",end="")  
 else:  
 print(j,end="")  
 print()  
  
*#Program for diamond pattern*rows = 5  
k = 2 \* rows - 2  
for i in range(0, rows):  
 for j in range(0, k):  
 print(end=" ")  
 k = k - 1  
 for j in range(0, i + 1):  
 print("\* ", end="")  
 print("")  
k = rows - 2  
  
for i in range(rows, -1, -1):  
 for j in range(k, 0, -1):  
 print(end=" ")  
 k = k + 1  
 for j in range(0, i + 1):  
 print("\* ", end="")  
 print("")

*#Program - print pattern E*for row in range(7):  
 for col in range(5):  
 if col ==0 or ((row==0 or row ==3 or row ==6 and col>0)):  
 print("\*", end="")  
 else:  
 print(end=" ")  
 print()  
print("\n")  
print("\n")  
*#program - print pattern O*for row in range(7):  
 for col in range(5):  
 if ((col ==0 or col ==4) and (row!=0 and row!=6))or ((row ==0 or row==6) and (col>0 and col<4)):  
 print("\*",end="")  
 else:  
 print(end=" ")  
 print()  
print("\n")  
print("\n")  
*#program -print pattern F*for row in range(7):  
 for col in range(5):  
 if (col ==0) or ((row ==0 or row ==3) and col>0):  
 print("\*",end="")  
 else:  
 print(end = " ")  
 print()  
print("\n")  
print("\n")  
  
*#program - pattern triangle*rows=int(input("Enter number of rows: "))  
k=0  
count=0  
count1=0  
for i in range(1,rows+1):  
 for j in range(1,(rows-i)+1):  
 print(" ",end="")  
 count+= 1  
 while k!=((2\*i)-1):  
 if count<=rows-1:  
 print(i+k,end=" ")  
 count+=1  
 else:  
 count1+=1  
 print(i+k-(2\*count1),end=" ")  
 k+=1  
  
 count1=count=k= 0  
 print()  
print("\n")  
print("\n")  
  
*#program*a=int((input("Enter the first number")))  
b=(int(input("Enter the second number")))  
def add(a,b):  
 print("addition value - ", a+b)  
def sub(a,b):  
 print("subtraction value - ", a-b)  
def mul(a,b):  
 print("Multiplied value - ",a\*b)  
def div(a,b):  
 print(" Divided value - " , a/b)  
def pow(a,b):  
 print("power of a is - " , a\*\*b)  
  
add(a,b)  
sub(a,b)  
mul(a,b)  
div(a,b)  
pow(a,b)

*#program -print class name*class car:  
 def parts(self):  
 pass  
class bus:  
 def route(self):  
 pass  
b=bus()  
print(b.\_\_class\_\_)  
b1=b.\_\_class\_\_  
print(b1.\_\_name\_\_)  
c=car()  
print(c.\_\_class\_\_)  
classes=c.\_\_class\_\_  
print(classes.\_\_name\_\_)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
*#Program - Encapsulation*class student:  
 def \_\_init\_\_(self,name,salary,project):  
 self.name=name  
 self.salary=salary  
 self.project=project  
  
 def show(self):  
 print("Name is ",self.name, "salary = ",self.salary)  
  
 def work(self):  
 print("Name is ", self.name,",I am working at ",self.project)  
  
s1=student("Mary",12000,"NLP")  
s1.show()  
s1.work()  
s2 = student("Anderson" ,14000,"HIG")  
s2.show()  
s2.work()  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program - public members*class student:  
 def \_\_init\_\_(self,name,salary):  
 self.name=name  
 self.salary=salary  
  
 def printmethod(self):  
 print("Name is ",self.name,"and my salary is ",self.salary)  
  
g=student("Leo ", 10000)  
g.printmethod()  
  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
*#Program-private members*class detail:  
 def \_\_init\_\_(self,name,age,salary):  
 self.name=name  
 self.\_\_age=age  
 self.salary=salary  
  
d=detail("Anamica ",34,23000)  
print("Name is ",d.name,"salary ",d.salary)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program- access private member*class employee:  
 def \_\_init\_\_(self,name,salary):  
 self.name=name  
 self.\_\_salary=salary  
  
 def show(self):  
 print("Name is ",self.name,"and my salary is ",self.\_\_salary)  
  
emp = employee("lara",30000)  
emp.show()  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program- Name Mangling*class Employee:  
 def \_\_init\_\_(self, name, salary):  
 self.name = name  
 self.\_\_salary = salary  
  
emp = Employee('Jessa', 10000)  
  
print('Name:', emp.name)  
print('Salary:', emp.\_Employee\_\_salary)  
  
*#program*print(4+3%5)  
  
*#program*class vehicle:  
 def \_\_init\_\_(self,price):  
 self.price=price  
 def display(self):  
 print("price = ", self.price)  
  
class category(vehicle):  
 def \_\_init\_\_(self,price,name):  
 vehicle.\_\_init\_\_(self,price)  
 self.name=name  
 def disp\_name(self):  
 print("vehicle=",self.name)  
  
obj= category(20000,'BMW')  
obj.disp\_name()  
obj.display()  
  
*#program*print(issubclass(category,vehicle))  
  
*#Program*print(isinstance(obj,vehicle))  
  
*#program*class A:  
 def display(self):  
 print("This is base class")  
  
class B(A):  
 def display(self):  
 print("This is derived class")  
  
obj = B()  
obj.display()  
  
*#Program -*class vehicle:  
 def vehicle\_info(self):  
 print("Inside vehicle class")  
class car(vehicle):  
 def car\_info(self):  
 print("Inside car class")  
  
c = car()  
c.vehicle\_info()  
c.car\_info()

*#program using elif statement*a=30  
b=30  
  
if b>a :  
 print("Bi greater",b)  
elif b == a:  
 print("a and b are equal")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
*#program using else statement*a = 90  
b = 34  
  
if b > a :  
 print("b is greater")  
elif b == a:  
 print("a and b are equal")  
else:  
 print("a is greater")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
*#program using oneline ifelse statement*a = 2  
b = 330, print("A") if a > b else print("B")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program using and operator*a = 200  
b = 33  
c = 500  
if a > b and c > a :  
 print("Both conditions are true")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program using or operator*a = 200  
b = 33  
c = 500  
if a > b or c > a:  
 print ("one of the condition satisfied")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program using nested if statement*x = 41  
if x > 10:  
 print("number greater than 10: ",x)  
if x > 20:  
 print("number greater than 20: ",x)  
else:  
 print ("but not above than 20: ",x)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program*x = 19  
if x > 10:  
 print("number greater than 10: ",x)  
if x > 20:  
 print("number greater than 20: ",x)  
else:  
 print ("but not above than 20: ",x)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program for pass statement*"""a - 10  
b = 13  
if b > a:  
 pass  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")"""  
  
*#program for while loop*i = 1  
print("While condition")  
while i < 6:  
 print(i)  
 i += 1  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")

*# program-1 while loop*i = 1  
while i < 6:  
 print(i)  
 if i == 3:  
 break  
 i+= 1  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*# program-2 while loop*j = 1  
while j < 8:  
 print(j)  
 print("Welcome to python world")  
 if j == 7 :  
 break  
 j+=1  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program -3 while loop*k = 0  
while k < 5:  
 k += 1  
 if k == 3:  
 continue  
 print(k)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program -4*c = 1  
while c < 5:  
 print(c)  
 c += 1  
else:  
 print("c is no longer than 6")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program -5 for loop*fruits = ["grapes","banana","Mango","Apple"]  
for x in fruits:  
 print("Given Element: ",x)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program-6 for loop(strings)*for x in "banana":  
 print("Result: ",x)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program-7*city = ["chennai","Banglore","mumbai","Kolkatta","goa"]  
for y in city:  
 print("cities list: ",y)  
 if y == "Kolkatta":  
 break  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program-8*city = ["chennai","Banglore","mumbai","Kolkatta","goa"]  
for y in city:  
 if y == "Kolkatta":  
 break  
 print("Result after break statement: ",y)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program\_9 for & continue statement*city = ["chennai","Banglore","mumbai","Kolkatta","goa"]  
for c in city:  
 if c == "mumbai":  
 continue  
 print ("list after if checking: ",c)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program-10 range*for x in range(6):  
 print(x)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program\_11*for x in range(2,6):  
 print(x)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program - 12*for x in range(1,6,1):  
 print(x)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program - 13*for x in range(6):  
 print(x)  
else:  
 print("finally finished")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program-14*for x in range(6):  
 if x == 3: break  
 print(x)  
else:  
 print("Finally finished")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program-15*x = [1,3,5,7,9]  
y = [2,4,6,8]  
for h in x:  
 for k in y:  
 print(h,k)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program - 15- function*def my\_function():  
 print("Hello from a function")  
  
my\_function()  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program - 16 - function*def family(name):  
 print(name+ "jones")  
  
family("amenda ")  
family("emili ")  
family("margrette ")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program- 17*def func1(sname,hname):  
 print(sname+" "+hname)  
  
func1("Geetha","Latha")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program -18*def joy(\*name):  
 print("The youngest child is "+name[2])  
  
joy("Latha","Geetha","uma","Rama","Ruba")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program -19*def my\_family(child1,child2,child3):  
 print("The youngest child is: "+child2)  
  
my\_family("Ruban","Altaf","Ram")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program - 20*def my\_function(\*\*kid ):  
 print("His last name is "+kid["lname"])  
  
my\_function(fname="Guru",lname="muthu")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program - 21 -Assignment*def fact(x):  
 print("The factor of ", x ,"are: ")  
 for i in range(1,x+1):  
 if x % i == 0:  
 print(i)  
  
j=420  
fact(j)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program -22 -Assignment to print 10 even numbers*print("Natural numbers are ")  
for i in range(0,11):  
 print(i)  
  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program -23 Assignment*for i in range(1,24):  
 if i % 2 == 0:  
 print("Even numbers are ",i)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program-24 Assignment*for i in range(1,24):  
 if i % 2 != 0:  
 print("Odd numbers are ",i)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program -25 Assignment*for i in range(1,11):  
 print("Whole numbers are ",i)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program -26 Assignment -loop statement to print the following series:  
#10, 20, 30 … … 300*print("Multipels of 10 numbers ",x)  
for i in range(10,301):  
 if i % 10 == 0:  
 print (i)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program -27 Assignment - print first 10 integers and their squares*for i in range(1,11):  
 print (i, " ",i \* i)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program - 28 Multipels of 7*for i in reversed(range (7,106)):  
 if i % 7 == 0:  
 print ("multipels of 7",i)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#rprogram-29 to display all even numbers that falls between two numbers*a = int(input("Enter the first number: "))  
b = int(input("Enter the second number: "))  
for i in range(a+1,b):  
 if i % 2 ==0:  
 print ("List of even numbers are: ",i)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program -30 to find the sum of the digits of a number accepted from the user.*num=int(input("enter a number:"))  
sum=0  
for i in str(num):  
 sum=sum+int(i)  
print(sum)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program -31 using function*def sum():  
 a=int(input("Enter the first number"))  
 b=int(input("Enter the second number"))  
 c = a + b  
 print("Result: ",c)  
  
sum()  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program -32 using function*def mult(a,b):  
 c = a \* b  
 print("Result of multiplication is: ",c)  
  
mult(6,9)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program - 33 using arbitary arguments*def play(\*name):  
 print("The youngest child is ",name[3])  
  
play("guru","Ram","siva","vishnu")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program -34 using function default parameter value*def country(name="Norway"):  
 print("My country name is ",name)  
  
country("USA")  
country("Europe")  
country("Brazil")  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program-35 passing list as an argument*def myfamily(member):  
 for x in member:  
 print ("Members list are ",x)  
  
list1=["Anuj","Uma","Raja","Rama","Akash"]  
myfamily(list1)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program - 36 function using return statement*def myfunc(x):  
 return 3 \* x  
print("The answer is")  
print(myfunc(2))  
print(myfunc(5))  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program -37 function to find maximum of 3 numbers*def max():  
 a=int(input("Enter the 1 number "))  
 b=int(input("Enter the 2 number "))  
 c=int(input("Enter the 3 number "))  
 if a > b and a > c:  
 print("Biggest number is ",a)  
 elif b > c:  
 print("Bigest number is ",b)  
 else:  
 print("Biggest number is ",c)  
  
max()  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program - 38 function to sum all the numbers in a list*def display(list1):  
 sum = 0  
 for i in list1:  
 sum = sum + i  
 print(sum)  
  
list2 = [8,2,3,0,7]  
print("Result of the addition is ")  
display(list2)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#Program - 39 function to multiply all the numbers in a list*def display(list1):  
 sum = 1  
 for i in list1:  
 sum = sum \* i  
 print(sum)  
  
list2 = [8, 2, 3, -1, 7]  
print("Result of the multiplication process is ")  
display(list2)  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program -40*a = "1234abcd"  
print("Reversed string is ", a[-1:])  
  
  
  
str=input("Enter the string ")  
n=int(input("Enter the number of characters "))  
b = []  
b=str.split(" ")  
for x in str:  
 if len(x) > n:  
 b.append(x)  
print(b)  
  
  
*#program  
# samplelist=[0,1,True]  
# print("All values" , all(samplelist))*nestedlist = [[2,4,6,8,10],[1,3,5,7,9]]  
for i in nestedlist:  
 print(i)  
 for j in i:  
 print(j)  
  
*#program*list1=[1,2,3,4,5]  
list2 = list1.reverse()  
print(list1)  
  
*#program*list1 = ["M", "na", "i", "Ke"]  
list2 = ["y", "me", "s", "lly"]  
list3=[i+j for i,j in zip(list1,list2)]  
print(list3)  
  
*#programs*list1 = ["Hello ", "take "]  
list2 = ["Dear", "Sir"]  
newlist = [x+y for x in list1 for y in list2]  
print(newlist)  
  
*#programs*list1 = [10, 20, 30, 40]  
list2 = [100, 200, 300, 400]  
for x,y in zip(list1,list2[::-1]):  
 print(x,y)  
  
*#program*list1=["george","","shyam","madan","kumar",""]  
*#res = []*res = list(filter(None,list1))  
print(res)  
  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
list1 = ["Mike", "", "Emma", "Kelly", "", "Brad"]  
  
*# remove None from list1 and convert result into list*res = list(filter(None, list1))  
print(res)  
  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*# program*tuple1 = ('P', 'Y', 'T', 'H', 'O', 'N')  
print(tuple1[-1])  
  
*#program*tuple1= ('P', 'Y', 'T', 'H', 'O', 'N')  
print(tuple1[-3])  
  
*#program*tuple1 = (0,1,2,3,4,5)  
print("Existing list ",tuple1)  
list1 = list(tuple1)  
list1.remove(2)  
tuple1 = tuple(list1)  
print("changed list ",tuple1)  
  
*#program*nested\_tuple = ((20, 40, 60), (10, 30, 50), "Python")  
print(nested\_tuple[2][0])  
for i in nested\_tuple:  
 print("tuple",i,"Elements")  
 for j in i:  
 print(j ,end=",")  
 print("\n")  
  
*# program*tuple1 = (10, 20, 30, 40, 50)  
tuple1 = tuple1[::-1]  
print(tuple1)  
  
*#program*tuple1 = ("Orange", [10, 20, 30], (5, 15, 25))  
print(tuple1[1][1])  
  
*#program*tuple1=tuple("hello",)  
print(tuple1)  
tuple3 = tuple(50,)  
print(tuple3)"""  
  
#program  
tuple1=(10,20,30,40)  
a,b,c,d = tuple1  
print("value of a ",a)  
print("value of b ",b)  
print("value of c ",c)  
print("value of d ",d)  
  
#program  
tuple1 = (11, 22)  
tuple2 = (99, 88)  
tuple3=tuple2  
tuple2=tuple1  
tuple1=tuple3  
print("tuple1 ",tuple1)  
print("tuple2 ",tuple2)  
  
#program-Copy specific elements from one tuple to a new tuple  
tuple1 = (11, 22, 33, 44, 55, 66)  
print(tuple1[3:-1])  
  
#program-Modify the tuple  
tuple1 = (11, [22, 33], 44, 55)  
print("Given list ",tuple1)  
tuple1[1][0]=222  
print("After modification ",tuple1)  
  
#Program-Counts the number of occurrences of item 50 from a tuple  
tuple1=(1,4,3,2,4,50,6,4,7,9)  
print("No.of appearances of 4: ",tuple1.count(4))  
  
#Program-Check if all items in the tuple are the same  
tuple1=(34,34,34,34)  
list1=list(tuple1)  
print(len(list1))  
  
#program- List Comprehension  
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]  
newlist = []  
for x in fruits:  
 if "a" in x:  
 newlist.append(x)  
print(newlist)  
  
#program-list Comprehension  
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]  
newlist= [x for x in fruits if"a" in x]  
print(newlist)  
  
#Program- list comprehension  
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]  
newlist = [x if x != "banana" else "orange" for x in fruits]  
print(newlist)  
  
#program- print any one of the color,get input from user  
colors=("Red","Green","Violet","blue")  
print("List of color: ",colors)  
value=(input("Enter any one of the color: "))  
for i in colors:  
 if value in i:  
 print("Color existed")  
 else:  
 print("Not existed")  
 break  
  
#program- Dictionary  
person = {"Name": "james", "age": 25, "salary": 20000}  
print(person)  
  
person = {"name": "Jessa", "country": "USA", "telephone": 1178}  
print(person)  
print(person.keys())  
for key in person:  
 print(key ,":",person[key])  
print("Length of the dict: ",len(person))

*#program - Reverse of string (g ni rtS toNmAI)*def preserve\_and\_reverse (s1):  
 list1 = s1.split()  
 print (list1)  
 char1 = ""  
 for i in range (0, len(list1)):  
 split\_list = list[0:(i + 1)]  
 print (split\_list)  
 for j in split\_list:  
 char1 =char1 + split\_list[j]  
 list1 = list1.append(char1)  
 output = list1.replace("", " ")  
 print (output)  
s1= input("Enter a string: ")  
result = preserve\_and\_reverse (s1)  
print (result)  
  
  
rows=int(input("Enter number of rows: "))  
k=0  
count=0  
count1=0  
for i in range(1,rows+1):  
 for j in range(1,(rows-i)+1):  
 print(" ",end="")  
 count+= 1  
 while k!=((2\*i)-1):  
 if count<=rows-1:  
 print(i+k,end=" ")  
 count+=1  
 else:  
 count1+=1  
 print(i+k-(2\*count1),end=" ")  
 k+=1  
  
 count1=count=k= 0  
 print()  
print("\n")  
print("\n")  
  
*#program - print pattern E*for row in range(7):  
 for col in range(5):  
 if col ==0 or ((row==0 or row ==3 or row ==6 and col>0)):  
 print("\*", end="")  
 else:  
 print(end=" ")  
 print()  
  
*#program -print pattern F*for row in range(7):  
 for col in range(5):  
 if (col ==0) or ((row ==0 or row ==3) and col>0):  
 print("\*",end="")  
 else:  
 print(end = " ")  
 print()  
*#program - print pattern O*for row in range(7):  
 for col in range(5):  
 if ((col ==0 or col ==4) and (row!=0 and row!=6))or ((row ==0 or row==6) and (col>0 and col<4)):  
 print("\*",end="")  
 else:  
 print(end=" ")  
 print()  
  
*#program*a=int((input("Enter the first number")))  
b=(int(input("Enter the second number")))  
def add(a,b):  
 print("addition value - ", a+b)  
def sub(a,b):  
 print("subtraction value - ", a-b)  
def mul(a,b):  
 print("Multiplied value - ",a\*b)  
def div(a,b):  
 print(" Divided value - " , a/b)  
def pow(a,b):  
 print("power of a is - " , a\*\*b)  
  
add(a,b)  
sub(a,b)  
mul(a,b)  
div(a,b)  
pow(a,b)  
program  
a=int((input("Enter the first number")))  
b=(int(input("Enter the second number")))  
def add(a,b):  
 print("addition value - ", a+b)  
def sub(a,b):  
 print("subtraction value - ", a-b)  
def mul(a,b):  
 print("Multiplied value - ",a\*b)  
def div(a,b):  
 print(" Divided value - " , a/b)  
def pow(a,b):  
 print("power of a is - " , a\*\*b)  
  
add(a,b)  
sub(a,b)  
mul(a,b)  
div(a,b)  
pow(a,b)

*#program- accessing private members from public method*class employee:  
 def \_\_init\_\_(self,name,salary):  
 self.name=name  
 self.\_\_salary=salary  
  
 def show(self):  
 print("Name- ",self.name,",Salary- ",self.\_\_salary)  
  
emp=employee("Mary",20000)  
emp.show()  
  
*#program- Name Mangling*class employee:  
 def \_\_init\_\_(self,name,salary):  
 self.name=name  
 self.\_\_salary=salary  
  
emp=employee("Ram",30000)  
print("Name- ",emp.name)  
print("Salary- ",emp.\_employee\_\_salary)  
  
*#Program-Protected Member*class company():  
 def \_\_init\_\_(self):  
 self.\_project="NLC"  
class employee(company):  
 def \_\_init\_\_(self,name):  
 self.name=name  
 company.\_\_init\_\_(self)  
  
 def show(self):  
 print("Employee name : ",self.name)  
 print("Working on project : ",self.\_project)  
  
emp=employee("Amenda")  
emp.show()  
  
*#Program-getters and setters*class student:  
 def \_\_init\_\_(self,name,age):  
 self.name=name  
 self.\_\_age=age  
 def get\_age(self):  
 return self.\_\_age  
 def set\_age(self,age):  
 self.\_\_age=age  
  
stud = student("Gourav",12)  
print("Name- ",stud.name,",Age- ",stud.get\_age())  
print("After setting age")  
stud.set\_age(15)  
print("Name- ",stud.name,",Age- ",stud.get\_age())  
  
*#Program-Information Hiding and conditional logic for setting an object attributes*class student:  
 def \_\_init\_\_(self,name,rollno,age):  
 self.name=name  
 self.\_\_rollno=rollno  
 self.\_\_age=age  
 def show(self):  
 print("Name- ",self.name,",Roll.no- ",self.\_\_rollno,",Age- ",self.\_\_age)  
 def get\_rollno(self):  
 return self.\_\_rollno  
 def set\_rollno(self,number):  
 if number >50:  
 print("Please enter correct roll number")  
 else:  
 self.\_\_rollno=number  
  
s1=student("Mary",123,15)  
s1.show()  
print("After modify")  
s1.set\_rollno(23)  
s1.show()  
  
*#Program- classes and objects*class vehicle():  
 def \_\_init\_\_(self,maxspeed,mileage):  
 self.maxspeed=maxspeed  
 self.mileage=mileage  
  
v=vehicle(140,120)  
print("maxspeed ",v.maxspeed,"mileage ",v.mileage)  
  
*#Program- Bus object that will inherit all of the variables and methods of the parent Vehicle class and display it.*class Vehicle:  
  
 def \_\_init\_\_(self, name, max\_speed, mileage):  
 self.name = name  
 self.max\_speed = max\_speed  
 self.mileage = mileage  
  
class bus(Vehicle):  
 pass  
  
volvo=bus("School volvo",130,150)  
print("Bus name-",volvo.name,",Speed-",volvo.max\_speed,",Mileage-",volvo.mileage)  
  
  
*#Program*x=int(input("Enter the value to be checked as odd or even - "))  
def check():  
 if x & 1:  
 return 'odd'  
 else:  
 return 'even'  
  
*# print("Result is ",check())  
  
#Program*class Mystore:  
 \_\_prod\_code=[]  
 \_\_prod\_name=[]  
 \_\_prod\_price=[]  
 \_\_prod\_quant=[]  
  
 def getdata(self):  
 self.p=int(input("Enter no.of products you need to store: "))  
 for x in range(self.p):  
 self.\_\_prod\_code.append(int(input("Enter product code: ")))  
 self.\_\_prod\_name.append(input("Enter product name: "))  
 self.\_\_prod\_price.append(int(input("Enter cost price: ")))  
  
 def display(self):  
 print(" Stock in stores")  
 print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
 print("Product code\t Product name\tCost price")  
 print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
  
 for x in range(self.p):  
 print(self.\_\_prod\_code[x], "\t\t\t",self.\_\_prod\_name[x],"\t\t\t",self.\_\_prod\_price[x])  
 print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
 def print\_bill(self):  
 total\_price=0  
 for x in range(self.p):  
 print("Enter the Quantity of product code")  
 q=int(input(self.\_\_prod\_code[x] ))  
 self.\_\_prod\_quant.append(q)  
 total\_price=total\_price+self.\_\_prod\_price[x]\*self.\_\_prod\_quant[x]  
 print(" INVOICE RECEIPT ")  
 print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
 print("Product code\t\tProduct name\t\tCost price\t\tQuantity\t\tTotal Amount")  
 for x in range(self.p):  
 print(self.\_\_prod\_code[x],"\t\t\t\t",self.\_\_prod\_name[x],"\t\t\t\t",self.\_\_prod\_price[x],"\t\t\t\t", self.\_\_prod\_quant[x],"\t\t\t\t",self.\_\_prod\_quant[x]\*self.\_\_prod\_price[x])  
 print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
 print(" Total Amount=", total\_price)  
 Amount=int(input("Enter the amount given by consumer "))  
 Total= Amount -total\_price  
 print("Balance amount= ",Total)  
  
s=Mystore()  
s.getdata()  
s.display()  
s.print\_bill()  
  
  
*#Program -Create a Bus class that inherits from the Vehicle class. Give the capacity argument of Bus.seating\_capacity() a default value of 50.*class Vehicle:  
 def \_\_init\_\_(self, name, max\_speed, mileage):  
 self.name = name  
 self.max\_speed = max\_speed  
 self.mileage = mileage  
  
 def seating\_capacity(self, capacity):  
 return f" The Seating Capacity of a {self.name} is {capacity}"  
  
class bus(Vehicle):  
 def seating\_capacity(self,capacity=50):  
 return super().seating\_capacity(capacity=50)  
  
b = bus("Omni",150,250)  
print(b.seating\_capacity())  
  
*#program-Define a class attribute”color” with a default value white. I.e., Every Vehicle should be white.*class Vehicle:  
  
 def \_\_init\_\_(self, name, max\_speed, mileage):  
 self.name = name  
 self.max\_speed = max\_speed  
 self.mileage = mileage  
  
class Bus(Vehicle):  
 color = "white"  
  
class Car(Vehicle):  
 color = "white"  
  
c= Car("SWift ",170,280)  
print("Name- ",c.name,",Max-speed- ",c.max\_speed,",Mileage- ",c.mileage)  
c1= Bus("Volvo" , 180,230)  
print("Name- ",c1.name,",Max-speed- ",c1.max\_speed,",Mileage- ",c1.mileage)  
  
*#Program - Create a Bus child class that inherits from the Vehicle class. The default fare charge of any vehicle is seating capacity \* 100.  
# If Vehicle is Bus instance, we need to add an extra 10% on full fare as a maintenance charge.  
# So total fare for bus instance will become the final amount = total fare + 10% of the total fare.*class vehicle():  
 def \_\_init\_\_(self,name,maxspeed,mileage,capacity):  
 self.name=name  
 self.maxspeed=maxspeed  
 self.mileage=mileage  
 self.capacity=capacity  
 def fare(self):  
 print("Name- ",self.name,"Maxspeed- ",self.maxspeed,"Mileage- ",self.mileage,"Seating capacity- ",self.capacity)  
 return self.capacity\*100  
class bus(vehicle):  
 pass  
schoolbus= bus("Omni",120,260,50)  
print("Amount= ",schoolbus.fare())  
  
*#program- Write a program to determine which class a given Bus object belongs to.*class vehicle():  
 def \_\_init\_\_(self,name,maxspeed,mileage,capacity):  
 self.name=name  
 self.maxspeed=maxspeed  
 self.mileage=mileage  
 self.capacity=capacity  
class bus(vehicle):  
 pass  
  
schoolbus = bus("Maruthi",150,250,30)  
print("Name- ",schoolbus.name,",Maxspeed- ",schoolbus.maxspeed,",Mileage- ",schoolbus.mileage,",Seating capacity- ",schoolbus.capacity)  
print(type(schoolbus))

*# #program  
# var=3  
#  
# var1=0  
#  
# var2=var1+1  
#  
# while(var<7):  
#  
# var2=var2+var1  
#  
# if(var1%2==0):  
#  
# var=var+1  
# var1=var1+1  
#  
# else:  
# var2=var2\*var1  
# var1=var1+3  
#  
# print("value v1",var1)  
# print("Value v2",var2)*import django  
print(django)  
  
*#Program - Python function that takes a sequence of numbers and  
# determines whether all the numbers are different from each other*def test\_distinct(data):  
 if len(data)==len(set(data)):  
 return True  
 else:  
 return False;  
  
print(test\_distinct([1,5,7,9]))  
print(test\_distinct([2,4,5,5,7,9]))  
  
*# Python3 program explaining work  
# of randint() function  
  
# import random module*import random  
  
r1 = random.randint(5, 15)  
print("Random number between 5 and 15 is " ,(r1))  
  
r2 = random.randint(-10, -2)  
print("Random number between -10 and -2 is " ,(r2))  
  
*#program-program to demonstrate the use of choice() method*import random  
list1=[1,2,3,4,5,6]  
print(random.choice(list1))  
string = "Geeks"  
print(random.choice(string))  
tuple1=(2,4,6,8,10)  
print(random.choice(tuple1))

*#next program*h=list(range(30,100,10))  
print("given Range ", h)  
j=list(range(10))  
print("Arrived range = ", j)  
print("After slicing-1", j[2:6:2])  
a=list(range(10))  
print("Arrived range = ", a)  
print("After slicing-2 ", a[0:8:3])  
a=list(range(10))  
print("After slicing-3 ", a[:-2])  
a=list(range(10))  
print("The Element are ", a)  
print("After slicing-4 ", a[:-2:2])  
a=list(range(10))  
print("The Element are ", a)  
print("After slicing-5 ", a[::4])  
a=list(range(10))  
print("The Element are ", a)  
print("After slicing-6 ", a[2:-2])  
a="Stuttgart"  
print("Given String: ",a)  
print("After slicing -7 " , a[2:-2])  
a="Stuttgart"  
print("Given String: ",a)  
print("After slicing-8 " , a[-2:])  
a=list(range(10))  
print("The Element are ", a)  
print("After slicing-9 ", a[2:3])  
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")  
  
*#program -40*a = "1234abcd"  
b=a[::-1]  
print("Reversed string is ",b)  
  
  
*#Program -42*for i in range(6):  
 for j in range(i):  
 print("\*",end=" ")  
 print(" ")  
print("~~~~~~~~~~~~~~~~~~~")  
for i in range(5,0,-1):  
 for j in range(i):  
 print("\*", end=" ")  
 print(" ")  
  
*#program for using loop comprehension*fruits = ["Banana","Mango","Apple","Avakoda"]  
newlist=[]  
for x in fruits:  
 if "a" in x:  
 newlist.append(x)  
print(newlist)  
  
  
*#program to sum all items in the list*x = [10,11,12,13]  
print("The item to be added", x)  
result = sum(x)  
print("The sum of all items are" , result)  
  
*#program- test*x=50  
def fun1():  
 x=25  
 print(x)  
fun1()  
print(x)  
  
*#program -test*x=75  
def myfunc():  
 x = x+1  
 print(x)  
myfunc()  
print(x)  
  
*#program - test*print(bool(0),bool(3.14159),bool(-3),bool(1.0+1j))  
  
*# program - test*def func1():  
 x=50  
 return x  
func1()  
print(x)  
  
*#Program -test*a=[10,20]  
b=a  
b+=[30,40]  
print(a)  
print(b)  
  
*#program - test*x=10  
y=50  
if x\*\*2<100 and y<100:  
 print(x,y)  
  
  
*#program -strings*a = " Rithika ,sen "  
print("length before strip method",len(a))  
x= a.strip(" ")  
print(x)  
print("length after strip method",len(x))  
print("splitting method ")  
c=a.split(",")  
print(c)  
print(list("Rithika sen"))  
c="python"  
print(list(c))  
for i in c:  
 print(i)  
  
*#program -strings*c=" WELCOME TO PYTHON WORLD "  
print(len(c))  
*#print(c.strip("Welcome "))*d = c.strip(" ")  
print(len(d))  
e="welcome to python world"  
print(e.capitalize())  
print(c.casefold())  
print(c.center(60))  
print(e.count("e"))  
print(e.count("o"))  
print(e.format())  
  
*# program -strings*name="kumar"  
age = 26  
print ("My name is ",name + " ","Age =",format(age))  
x="My name is varshaa"  
print(x.encode())  
*# program*x="H\te\tl\tl\to"  
print(x.expandtabs(2))  
print(x.find("e"))  
  
  
*#program -16-assignment*"""a Python program to generate and print a list of first and last 5 elements   
where the values are square of numbers between 1 and 30 (both included)."""  
def values():  
 l = list()  
 for i in range(1,30):  
 l.append(i\*\*2)  
 print(l[:5])  
 print(l[-5:])  
values()  
  
  
*#program-17 -assignment*"""a Python program to generate and print a list except for the first 5 elements,   
where the values are square of numbers between 1 and 30 (both included)"""  
def printvalues():  
 l = list()  
 for i in range(1,30):  
 l.append(i\*\*2)  
 print(l[5:15])  
printvalues()  
  
*#program-18  
#Write a Python program to sum all the items in a list*newlist=[]  
k=int(input("enter the number of elements in list"))  
for i in range(1,k):  
 newlist1= int(input("enter the elements one by one"))  
 newlist.append(newlist1)  
print(sum(newlist))  
  
*#program-19  
# Write a Python program to get the largest number from a list.*newlist=[]  
k=int(input("enter the number of elements in list"))  
for i in range(0,k):  
 newlist1= int(input("enter the elements one by one"))  
 newlist.append(newlist1)  
print("given elements" ,newlist)  
print(max(newlist))  
  
*#program\_20  
#Write a Python program to get the smallest number from a list*newlist=[]  
k=int(input("enter the number of elements in list"))  
for i in range(0,k):  
 newlist1= int(input("enter the elements one by one"))  
 newlist.append(newlist1)  
print("given elements" ,newlist)  
print("Smallest element in the list" ,min(newlist))  
  
*#program=21  
# a Python program to count the number of strings  
# where the string length is 2 or more and the first and last character are same from a given list of strings*"""a Python program to count the number of strings  
where the string length is 2 or more and the first and last character are same from a given  
list of strings"""  
str=input("enter a string ")  
k=len(str)  
print("length of the string", k)  
for i in range(1,k):  
 if (k >=2 and str[0]==str[k-1]):  
 break  
print("string is ", str)  
  
*#progrma -22*"""a Python program to get a list, sorted in increasing order by the   
last element in each tuple from a given list of non-empty tuples"""  
  
newlist = [(2,1),(3,3),(1,2),(2,4)]  
print("GIven list ", newlist)  
def printitem(n):return n[-1]  
tuple1= newlist  
print("sorted list" , sorted(tuple1,key=printitem))  
  
*#Write a Python program to remove duplicates from a list.*newlist = [1,3,3,7,74,5,1,9,9]  
print("Given list", newlist)  
newset=set()  
list2=list()  
for x in newlist:  
 if x not in newset:  
 newlist.append(x)  
 newset.add(x)  
print("List without duplicates", newset)  
  
*#Write a Python program to check a list is empty or not*x=input("Enter the elements separated by comma: ")  
  
if x:  
 print("list is full")  
else:  
 print("empty")  
  
*#Write a Python program to clone or copy a list.*list1=[1,2,3,4,5]  
list2=[list1]  
print("original list", list1)  
print("cloning list",list2)  
  
*#Write a Python program to find the list of words that are  
# longer than n from a given list of words.*str=input("Enter the string ")  
n=int(input("Enter the number of characters "))  
a = []  
b=str.split(" ")  
for x in b:  
 if (len(x) > n):  
 a.append(x)  
print(a)