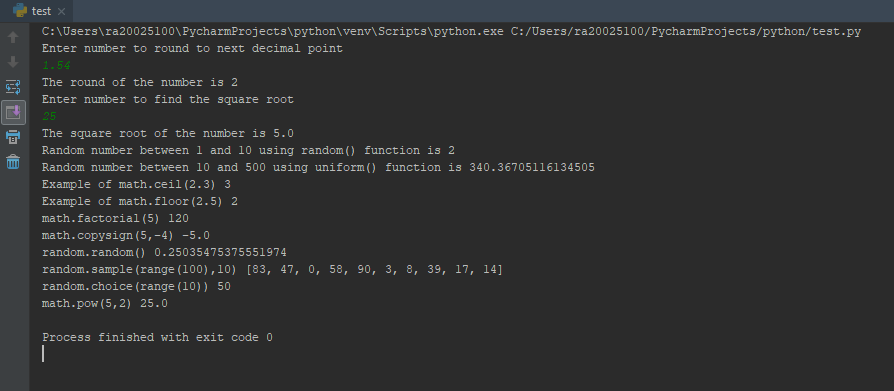
PYTHON ASSIGNMENT 2 (RA20025100)

21) Code

import random  
import math  
  
test\_array=[10,20,50,40,30]  
print("Enter number to round to next decimal point")  
print("The round of the number is",round(float(input())))  
print("Enter number to find the square root")  
print("The square root of the number is",math.sqrt(float(input())))  
print("Random number between 1 and 10 using random() function is",random.randint(1,10))  
print("Random number between 10 and 500 using uniform() function is",random.uniform(10,500))  
  
'''Math and Random module functions'''  
print("Some Math and Random modules functions")  
  
print("Example of math.ceil(2.3)",math.ceil(2.3))  
print("Example of math.floor(2.5)",math.floor(2.5))  
print("math.factorial(5)",math.factorial(5))  
print("math.copysign(5,-4)",math.copysign(5,-4))  
print("random.random()",random.random())  
print("random.sample(range(100),10)",random.sample(range(100),10))  
print("random.choice(range(10))",random.choice(test\_array))  
print("math.pow(5,2)",math.pow(5,2))

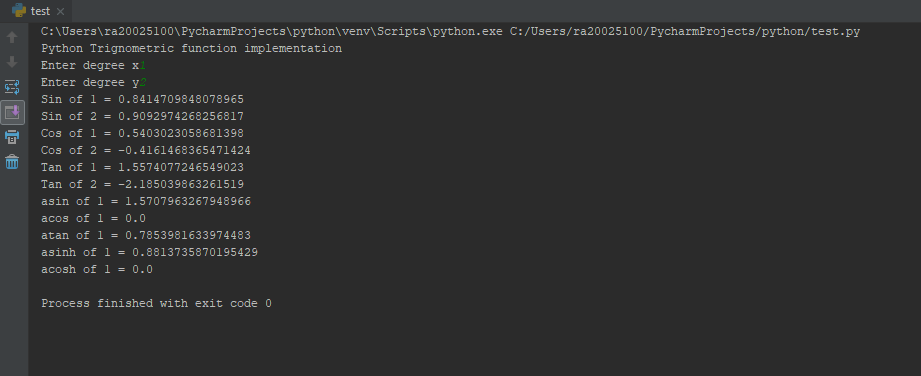
OUTPUT



22) CODE

import math  
  
print("Python Trignometric function implementation")  
x=int(input("Enter degree x"))  
y=int(input("Enter degree y"))  
print("Sin of",x,"=",math.sin(x))  
print("Sin of",y,"=",math.sin(y))  
print("Cos of",x,"=",math.cos(x))  
print("Cos of",y,"=",math.cos(y))  
print("Tan of",x,"=",math.tan(x))  
print("Tan of",y,"=",math.tan(y))  
  
print("asin of",x,"=",math.asin(x))  
print("acos of",x,"=",math.acos(x))  
print("atan of",x,"=",math.atan(x))  
  
print("asinh of",x,"=",math.asinh(x))  
print("acosh of",x,"=",math.acosh(x))

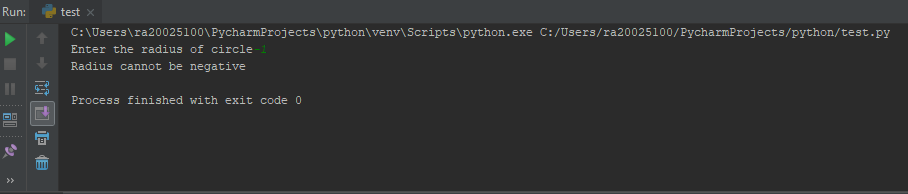
OUTPUT



23) CODE

import math  
  
try:  
 r=int(input("Enter the radius of circle"))  
except ValueError:  
 print("Enter Numeric values only")  
  
if r<=0:  
 print("Radius cannot be negative")  
 exit(0)  
else:  
 print("The area of circle is","=",math.pi\*r\*\*2)

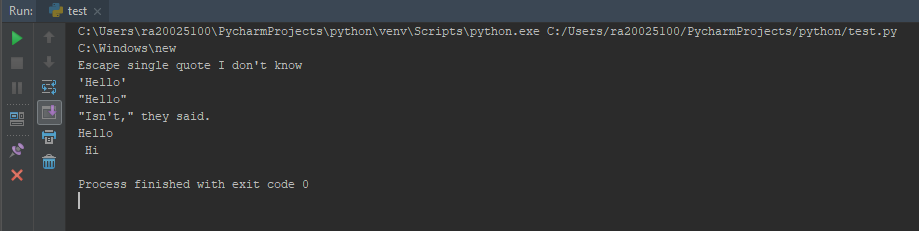
OUTPUT



24) CODE

print(r"C:\Windows\new")  
print("Escape single quote","I don't know")  
print("'Hello'")  
print('"Hello"')  
print('"Isn\'t," they said.')  
print("Hello \n Hi")

OUTPUT

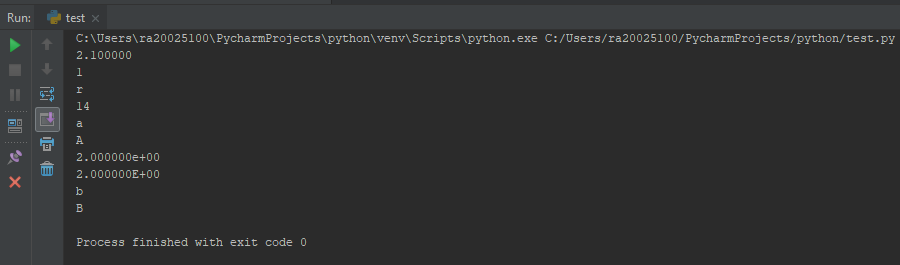


25) CODE

print("%f" % 2.1)  
print("%d"%1)  
print("%c"%'r')  
print("%o"%12)  
print("%x"%10)  
print("%X"%10)  
print("%e"%2)  
print("%E"%2)  
print("%x"%11)  
print("%X"%11)

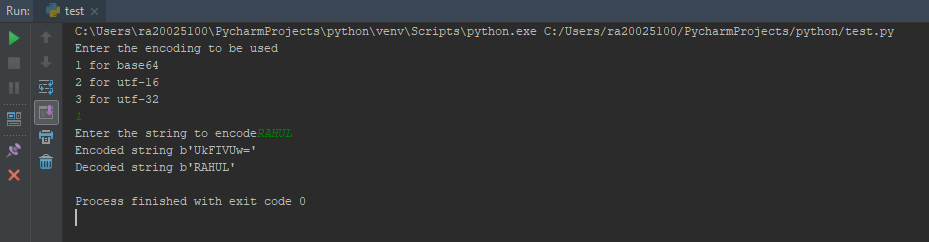
OUTPUT

26) CODE



import base64  
print("Enter the encoding to be used")  
print("1 for base64")  
print("2 for utf-16")  
print("3 for utf-32")  
en=int(input())  
str\_input=input("Enter the string to encode")  
  
if en==1:  
 str\_base64\_en = base64.b64encode(str\_input.encode())  
 print("Encoded string",str\_base64\_en)  
elif en==2:  
 str\_utf\_8=str\_input.encode('utf-16','strict')  
 print("Encoded string",str\_utf\_8)  
  
elif en==3:  
 str\_utf\_32=str\_input.encode('utf-32','strict')  
 print("Encoded string",str\_utf\_32)  
  
if en==1:  
 str\_base64\_de=base64.b64decode(str\_base64\_en)  
 print("Decoded string",str\_base64\_de)  
elif en==2:  
 str\_utf\_8\_de=str\_utf\_8.decode('utf-16','strict')  
 print("Decode string",str\_utf\_8\_de)  
elif en==3:  
 str\_utf\_32\_de=str\_utf\_32.decode('utf-32','strict')  
 print("Decode string",str\_utf\_32\_de)

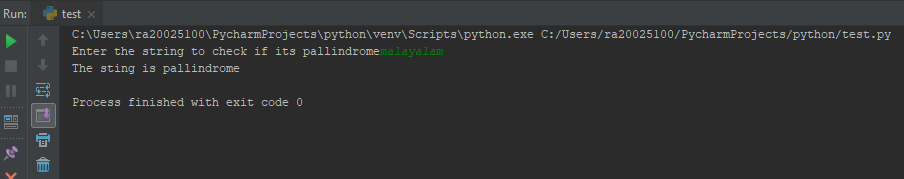
OUTPUT:



27) CODE

check=input("Enter the string to check if its pallindrome")  
rev="".join(reversed(check))  
if check==rev:  
 print("The sting is pallindrome")  
else:  
 print("The sting is not a pallindrome")

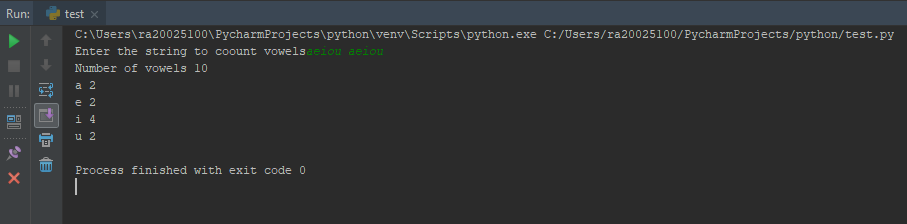
OUTPUT



28) CODE

counta=counte=counti=counto=countu=0  
total\_count=0  
s=input("Enter the string to coount vowels")  
a=s.split()  
s1=""  
for i in a:  
 s1+=i  
s1.lower()  
for i in s1:  
 if i=='a':  
 counta+=1  
 total\_count+=1  
 elif i=='e':  
 counte+=1  
 total\_count+=1  
 elif i=='i':  
 counti+=1  
 total\_count+=1  
 elif i=='o':  
 counti+=1  
 total\_count+=1  
 elif i=='u':  
 countu+=1  
 total\_count+=1  
print("Number of vowels",total\_count)  
if counta!=0:  
 print("a",counta)  
if counte!=0:  
 print("e",counte)  
if counti!=0:  
 print("i",counti)  
if counto!=0:  
 print("o",counto)  
if countu!=0:  
 print("u",countu)

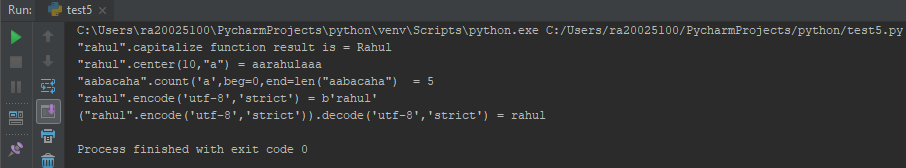
OUTPUT:



29) A) CODE

print("\"rahul\".capitalize function result is","=","rahul".capitalize())  
print("\"rahul\".center(10,\"a\")","=","rahul".center(10,"a"))  
print("\"aabacaha\".count('a',beg=0,end=len(\"aabacaha\") ","=","aabacaha".count('a',0,len("aabacaha")))  
print("\"rahul\".encode('utf-8','strict')","=","rahul".encode('utf-8','strict'))  
print("(\"rahul\".encode('utf-8','strict')).decode('utf-8','strict')",  
 "=",("rahul".encode('utf-8','strict')).decode('utf-8','strict'))

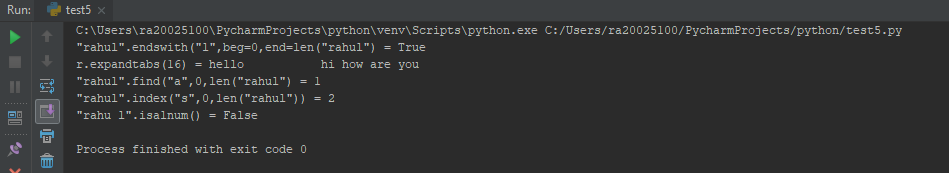
OUTPUT :29)A)



29) B) CODE

print("\"rahul\".endswith(\"l\",beg=0,end=len(\"rahul\")","=","rahul".endswith('l',0,len("rahul")))  
r="hello\thi how are you"  
print("r.expandtabs(16)","=",r.expandtabs(16))  
print("\"rahul\".find(\"a\",0,len(\"rahul\")","=","rahul".find("a",0,len("rahul")))  
print("\"rahul\".index(\"s\",0,len(\"rahul\"))","=","rahul".index("h",0,len("rahul")))  
print("\"rahu l\".isalnum()","=","rahu l".isalnum())

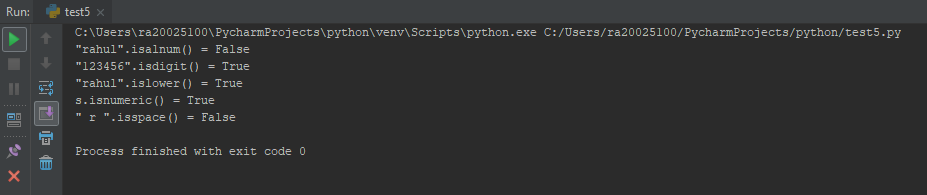
OUTPUT:29)B)



29) C) CODE

print("\"rahul\".isalnum()","=","rahul1".isalpha())  
print("\"123456\".isdigit()","=","123456".isdigit())  
print("\"rahul\".islower()","=","rahul".islower())  
s=u"1"  
print("s.isnumeric()","=", s.isnumeric())  
print("\" r \".isspace()","="," r ".isspace())

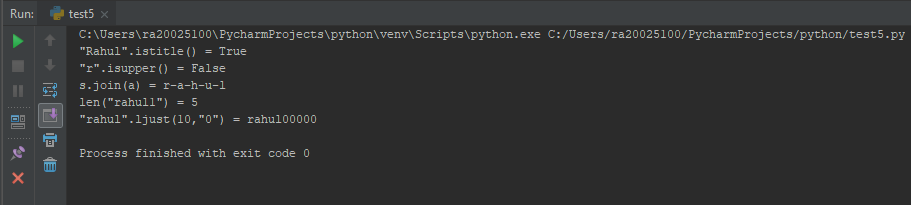
OUTPUT:29)C)



29)D) CODE

print("\"Rahul\".istitle()","=","Rahul".istitle())  
print("\"r\".isupper()","=","r".isupper())  
a=["r","a","h","u","l"]  
s="-"  
print("s.join(a)","=",s.join(a))  
print("len(\"rahul1\")","=",len("rahul"))  
print("\"rahul\".ljust(10,\"0\")","=","rahul".ljust(10,"0"))

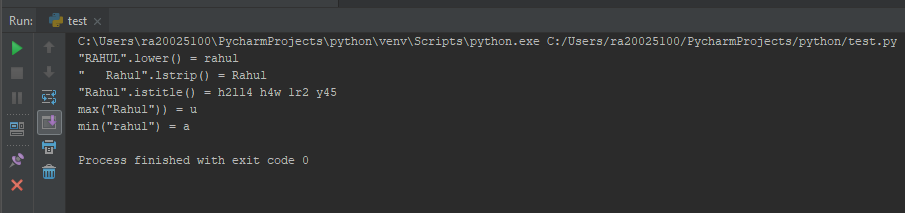
OUTPUT:29)D)



29)E)CODE

print("\"RAHUL\".lower()","=","RAHUL".lower())  
print("\" Rahul\".lstrip()","="," Rahul".lstrip())  
str="hello how are you"  
i="aeiou"  
o="12345"  
print("\"Rahul\".istitle()","=",str.translate(str.maketrans(i,o)))  
print("max(\"Rahul\"))","=",max("Rahul"))  
print("min(\"rahul\")","=",min("rahul"))

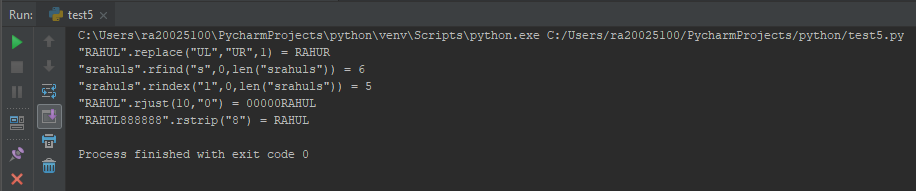
OUTPUT:29)E)



29)F)CODE

print("\"RAHUL\".replace(\"UL\",\"UR\",1)","=","RAHUL".replace("UL","UR",1))  
print("\"srahuls\".rfind(\"s\",0,len(\"srahuls\"))","=","srahuls".rfind("s",0,len("srahuls")))  
print("\"srahuls\".rindex(\"l\",0,len(\"srahuls\"))","=","srahuls".rindex("l",0,len("srahuls")))  
print("\"RAHUL\".rjust(10,\"0\")","=","RAHUL".rjust(10,"0"))  
print("\"RAHUL888888\".rstrip(\"8\")","=","RAHUL888888".rstrip("8"))

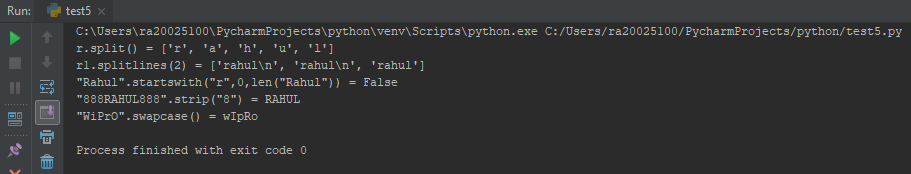
OUTPUT 29)F)



29)G)CODE

r="r a h u l"  
print("r.split()","=",r.split())  
r1="rahul\nrahul\nrahul"  
print("r1.splitlines(2)","=",r1.splitlines(2))  
print("\"Rahul\".startswith(\"r\",0,len(\"Rahul\"))","=","Rahul".startswith("r",0,len("Rahul")))  
print("\"888RAHUL888\".strip(\"8\")","=","888RAHUL888".strip("8"))  
print("\"WiPrO\".swapcase()","=","WiPrO".swapcase())

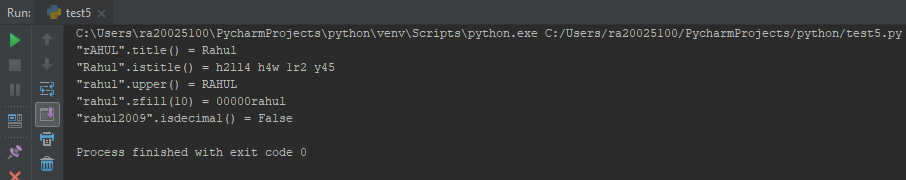
OUTPUT)29)G)



29)H)CODE

print("\"rAHUL\".title()","=","rAHUL".title())  
str="hello how are you"  
i="aeiou"  
o="12345"  
print("\"Rahul\".istitle()","=",str.translate(str.maketrans(i,o)))  
print("\"rahul\".upper()","=","rahul".upper())  
print("\"rahul\".zfill(10)","=","rahul".zfill(10))  
print("\"rahul2009\".isdecimal()","=","rahul2009".isdecimal())

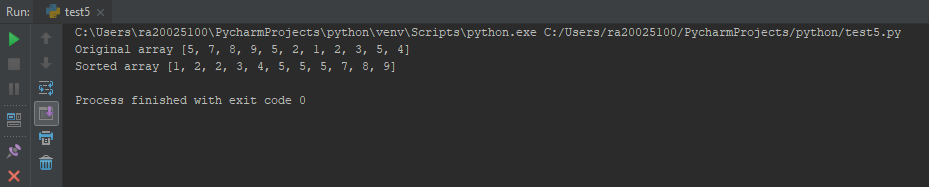
OUTPUT)29)H)



30)CODE

num=[5,7,8,9,5,2,1,2,3,5,4]  
print("Original array",num)  
for i in range(len(num)):  
 for j in range(i+1,len(num)):  
 if num[i]>num[j]:  
 num[i],num[j]=num[j],num[i]  
print("Sorted array",num)

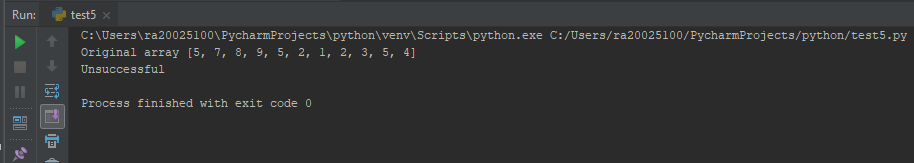
OUTPUT



31)CODE

key=0  
num=[5,7,8,9,5,2,1,2,3,5,4]  
print("Original array",num)  
for i in range(len(num)):  
 for j in range(i+1,len(num)):  
 if num[i]>num[j]:  
 num[i],num[j]=num[j],num[i]  
m=len(num)//2  
h=len(num)  
l=0  
if key>=num[m]:  
 l=m  
 for i in range(l,h):  
 if key==num[l]:  
 print("Successful")  
 break  
 else:  
 print("Unsuccessful")  
 break  
elif key<=num[m]:  
 h=m  
 for i in range(l,h):  
 if key==num[l]:  
 print("Successful")  
 break  
 else:  
 print("Unsuccessful")  
 break

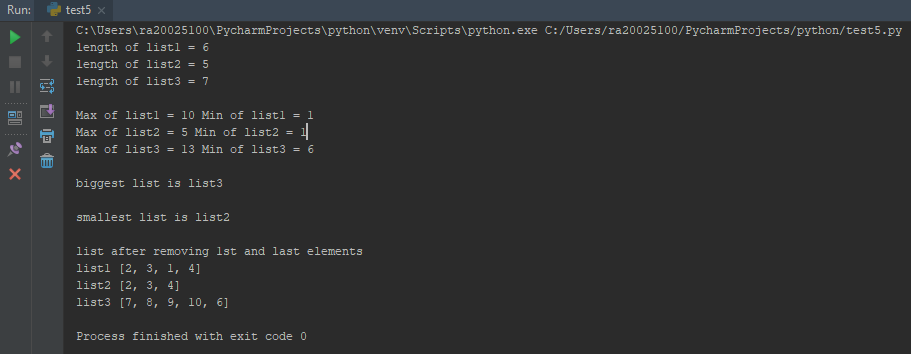
OUTPUT



32)CODE

list1=[10,2,3,1,4,5]  
list2=[1,2,3,4,5]  
list3=[11,7,8,9,10,6,13]  
  
print("length of list1","=",len(list1))  
print("length of list2","=",len(list2))  
print("length of list3","=",len(list3))  
print(end="\n")  
print("Max of list1","=",max(list1),"Min of list1","=",min(list1))  
print("Max of list2","=",max(list2),"Min of list2","=",min(list2))  
print("Max of list3","=",max(list3),"Min of list3","=",min(list3))  
print(end="\n")  
if list1>list2 and list1>list3:  
 print("biggest list is list1")  
elif list2>list1 and list2>list3:  
 print("biggest list is list2")  
elif list3>list1 and list3>list2:  
 print("biggest list is list3")  
print(end="\n")  
if list1<list2 and list1<list3:  
 print("smallest list is list1")  
elif list2 < list1 and list2 < list3:  
 print("smallest list is list2")  
elif list3<list1 and list3<list2:  
 print("smallest list is list3")  
print(end="\n")  
print("list after removing 1st and last elements")  
list1.remove(list1[0])  
list1.remove(list1[len(list1)-1])  
print("list1",list1)  
  
list2.remove(list2[0])  
list2.remove(list2[len(list2)-1])  
print("list2",list2)  
  
list3.remove(list3[0])  
list3.remove(list3[len(list3)-1])  
print("list3",list3)

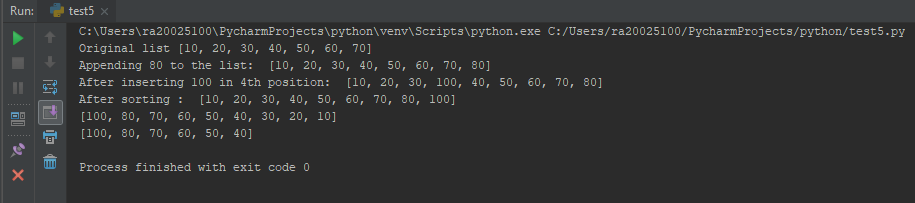
OUTPUT



33)CODE

list1 = [10,20,30,40,50,60,70]  
print("Original list", list1)  
list1.append(80)  
print("Appending 80 to the list: ", list1)  
  
list1.insert(3,100)  
print("After inserting 100 in 4th position: ", list1)  
  
list1.sort()  
print("After sorting : ", list1)  
  
list2 = sorted(list1,key = int, reverse = True)  
print(list2)  
  
list2.pop(len(list2)-1)  
list2.pop(len(list2)-1)  
list2.pop(len(list2)-1)  
print(list2)

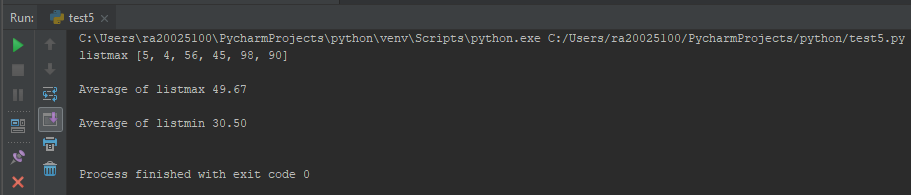
OUTPUT:



34) CODE

list1 = [1,1,2,3,4,5,5]  
list2 = [12,12,23,34,45,56,56]  
list3 = [67,67,78,89,90,98,98]  
listmax = []  
listmin=[]  
  
list1.sort()  
list2.sort()  
list3.sort()  
  
listmax.append(list1[len(list1)-1])  
i=len(list1)-2  
while list1[i]==listmax[len(listmax)-1]:  
 i-=1  
if i>=0:  
 listmax.append(list1[i])  
  
listmax.append(list2[len(list2)-1])  
i=len(list2)-2  
while list2[i]==listmax[len(listmax)-1]:  
 i-=1  
if i>=0:  
 listmax.append(list2[i])  
  
listmax.append(list3[len(list3)-1])  
i=len(list3)-2  
while list3[i]==listmax[len(listmax)-1]:  
 i-=1  
if i>=0:  
 listmax.append(list3[i])  
  
print("listmax",listmax)  
print(end="\n")  
avg=sum(listmax)/len(listmax)  
print("Average of listmax %.2f" %avg)  
print(end="\n")  
  
listmin.append(list1[0])  
i=1  
while list1[i]==listmin[len(listmin)-1]:  
 if i<=len(list1):  
 i+=1  
listmin.append(list1[i])  
  
listmin.append(list2[0])  
i=1  
while list2[i]==listmin[len(listmin)-1]:  
 if i<=len(list2):  
 i+=1  
listmin.append(list2[i])  
  
listmin.append(list3[0])  
i=1  
while list3[i]==listmin[len(listmin)-1]:  
 if i<=len(list3):  
 i+=1  
listmin.append(list3[i])  
  
avg=sum(listmin)/len(listmin)  
print("Average of listmin %.2f" %avg)  
print(end="\n")

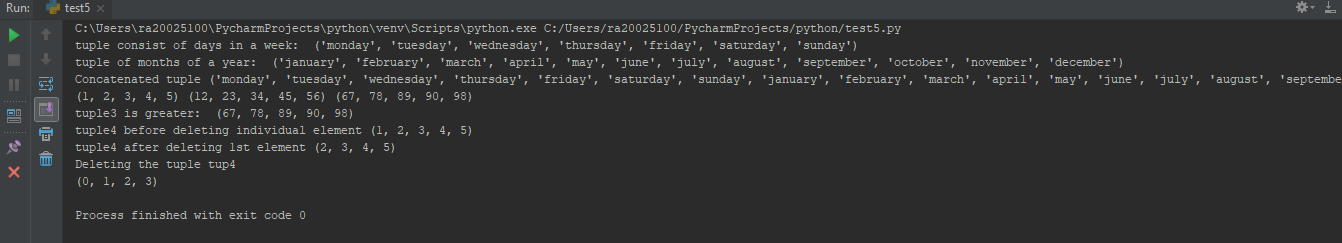
OUTPUT:



35) CODE

tup1=('monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday')  
print ("tuple consist of days in a week: ", tup1)  
  
tup2 = ('january', 'february', 'march', 'april', 'may', 'june', 'july', 'august', 'september', 'october', 'november', 'december')  
print ("tuple of months of a year: ", tup2)  
  
tup3 = tup1+tup2  
print("Concatenated tuple",tup3)  
  
tup1 = (1,2,3,4,5)  
tup2 = (12,23,34,45,56)  
tup3 = (67,78,89,90,98)  
  
print(tup1,tup2,tup3)  
  
if tup1 > tup2 and tup1>tup3:  
 print("tuple1 is greater: ", tup1)  
elif tup2 > tup3 and tup1>tup2:  
 print("tuple2 is greater: ", tup2)  
elif tup3>tup2 and tup3>tup1:  
 print("tuple3 is greater: ", tup3)  
  
tup4 = tup1  
print("tuple4 before deleting individual element",tup4)  
tup4=tup4[1:]  
print("tuple4 after deleting 1st element",tup4)  
  
print("Deleting the tuple tup4")  
del tup4  
  
tup4=()  
list1=list(tup4)  
for i in range(4):  
 list1.append(i)  
tup4=tuple(list1)  
print("Inserted to tuple after typecasting to list",tup4)

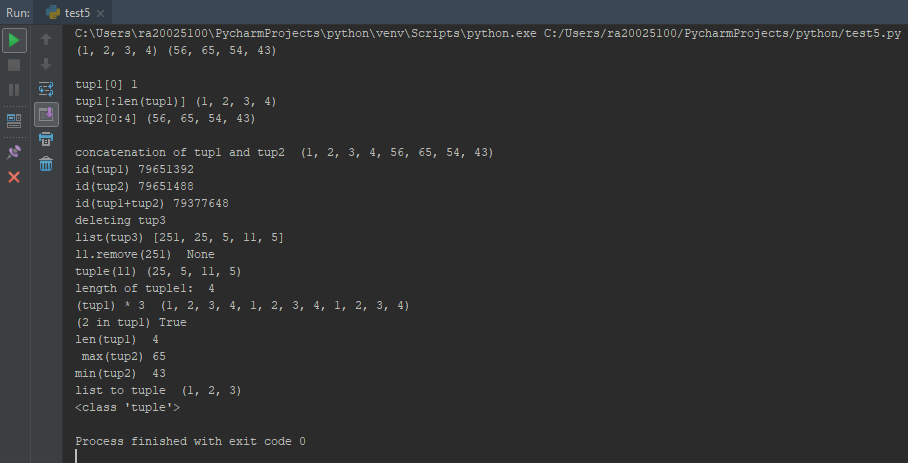
OUTPUT



36) CODE

tup1 = (1,2,3,4)  
tup2 = (56,65,54,43)  
tup3=(251,25,5,11,5)  
tup4=(251,25,5,11,5)  
print(tup1,tup2)  
  
print(end="\n")  
  
print("tup1[0]", tup1[0])  
print("tup1[:len(tup1)]" , tup1[:len(tup1)])  
print("tup2[0:4]", tup2[0:4])  
  
print(end="\n")  
  
  
print("concatenation of tup1 and tup2 ",tup1+tup2)  
  
print("id(tup1)",id(tup1))  
print("id(tup2)",id(tup2))  
print("id(tup1+tup2)",id(tup1+tup2))  
  
print("deleting tup3")  
del tup3  
  
print("list(tup3)",list(tup4))  
l1=list(tup4)  
print("l1.remove(251) ",l1.remove(251))  
  
print("tuple(l1)",tuple(l1))  
  
print("length of tuple1: ", len(tup1))  
  
print("(tup1) \* 3 ", (tup1) \* 3)  
print("(2 in tup1)", (2 in tup1))  
  
  
  
l = [1,2,3]  
print("len(tup1) ", len(tup1))  
print(" max(tup2)", max(tup2))  
print("min(tup2) ", min(tup2))  
tup6 = tuple(l)  
print("list to tuple ", tup6)  
print(type(tup6))

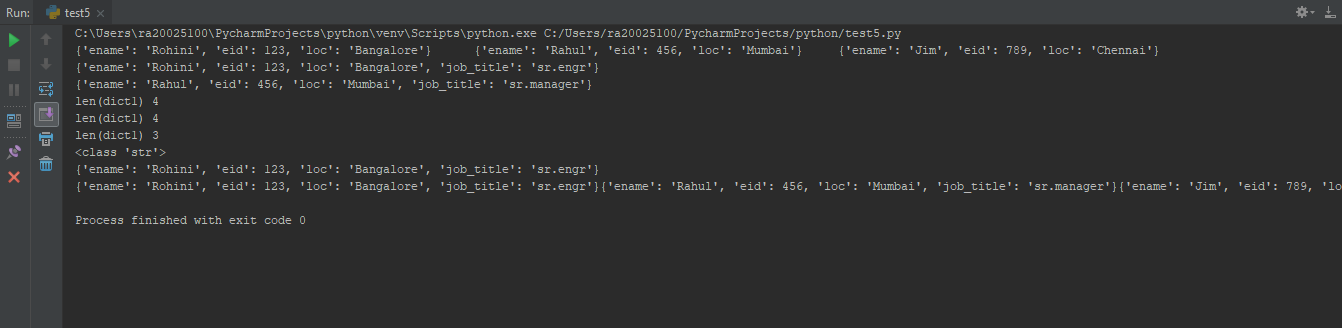
OUTPUT



37) CODE

dict1 = {"ename":"Rohini", "eid":123, "loc":"Bangalore" }  
dict2 = {"ename":"Rahul", "eid":456, "loc":"Mumbai" }  
dict3 = {"ename":"Jim", "eid":789, "loc":"Chennai" }  
  
print(dict1,"\t",dict2,"\t", dict3)  
  
dict1.update({"job\_title":"sr.engr"})  
print(dict1)  
dict2.update({"job\_title":"sr.manager"})  
print(dict2)  
  
print("len(dict1)",len(dict1))  
print("len(dict1)",len(dict2))  
print("len(dict1)",len(dict3))  
  
  
s1=str(dict1)  
print(type(s1))  
print(s1)  
s2=str(dict2)  
s3=str(dict3)  
s4 = s1+s2+s3  
print(s4)

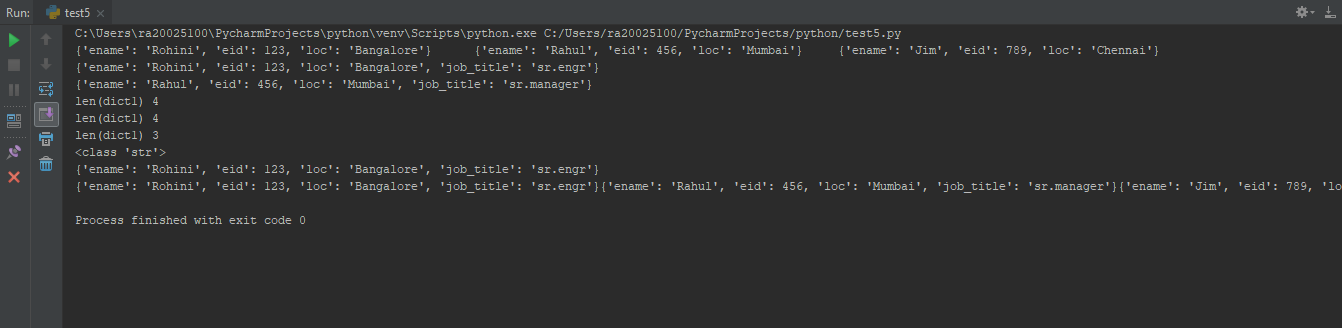
OUTPUT



38)CODE

dict1 = {'name':'RAHUL', 'age':22}  
dict2 = {'empid':100, 'salary':23000}  
dict3 = {}  
  
print(dict1, dict2)  
dict3.update(dict1)  
dict3.update(dict2)  
print(dict3)  
  
dict3['salary'] = ( dict3['salary'] + ( (dict3['salary'] \* 10) / 100) )  
dict3['age'] = 26  
print(dict3)  
  
dict3.update({'grade' :"B1"})  
print(dict3)  
  
print("dict3 keys",dict3.keys())  
print("dict3 values",dict3.values())  
  
del dict3['age']  
print("deleted key age")  
print(dict3)

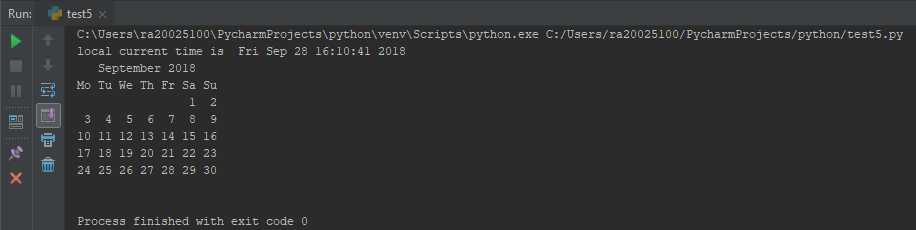
OUTPUT



39)CODE

import time  
import calendar  
localtime = time.asctime(time.localtime(time.time()))  
print("local current time is ", localtime)  
  
cal = calendar.month(2018,9)  
print(cal)

OUTPUT

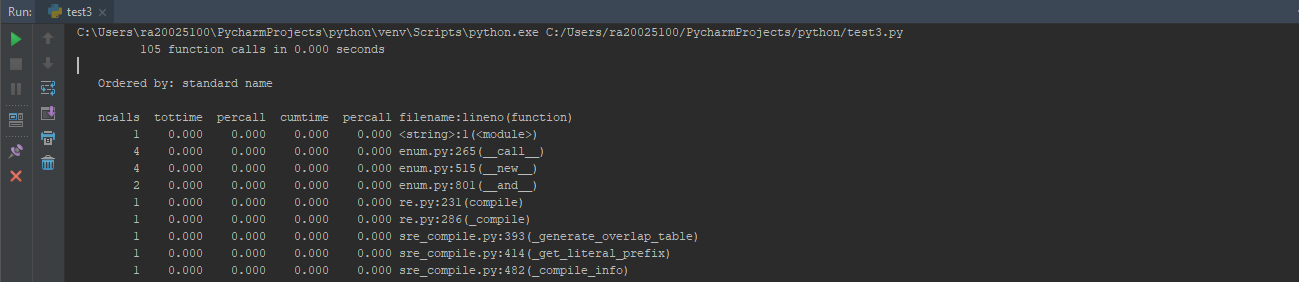
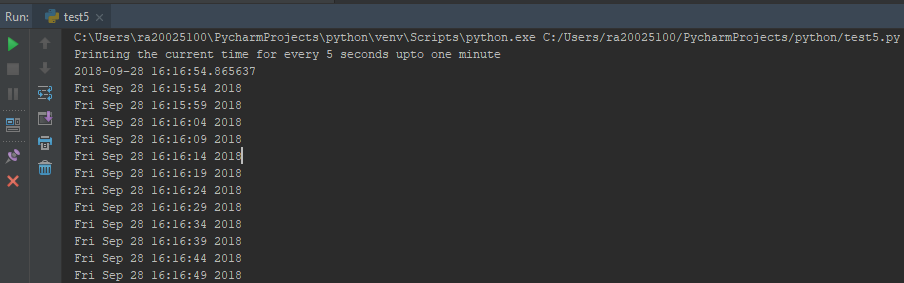


40) CODE

from datetime import \*  
import time  
  
print("Printing the current time for every 5 seconds upto one minute ",end="\n")  
res = datetime.now() + timedelta(seconds = 60)  
print(res, end="\n")  
  
while(datetime.now() < res):  
 print(time.asctime(time.localtime(time.time())))  
 time.sleep(5)

import cProfile  
import re  
cProfile.run('re.compile("32.py")')

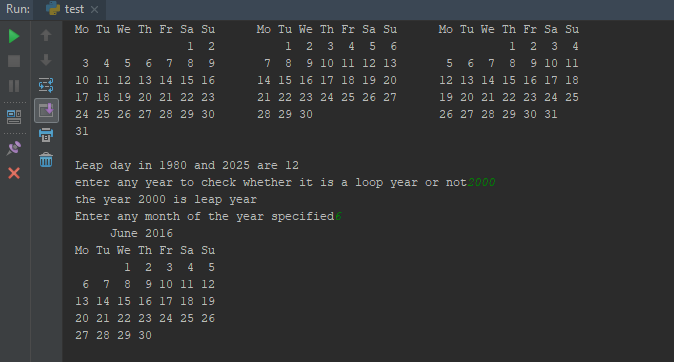
OUTPUT



41)CODE

import calendar  
print("The calendar year 2016 ")  
print (calendar.calendar(2016))  
  
st = 1980  
end = 2025  
count=0  
for i in range(1980,2025):  
 if i%4 == 0:  
 count+=1  
print("Leap day in 1980 and 2025 are",count)  
  
y = int(input("enter any year to check whether it is a loop year or not"))  
if y%4 == 0:  
 print("year %d is leap year"% y)  
else:  
 print("year %d is not a leap year" %y)  
  
m = int(input("Enter any month of the year specified"))  
  
print(calendar.month(2016,m))

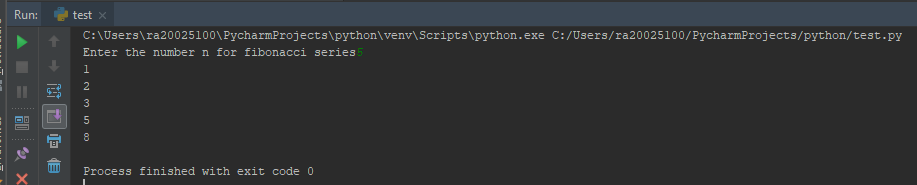
OUTPUT



42)CODE

def fib(n):  
 count =0  
 n1 = 0  
 n2 =1  
 if n< 0:  
 print("Not a positive integer")  
 if n ==1:  
 print(n1)  
 else:  
 while count < n:  
 n3 = n1+n2  
 print(n3)  
 n1 = n2  
 n2 = n3  
 count +=1  
  
fib(int(input("Enter the number n for fibonacci series")))

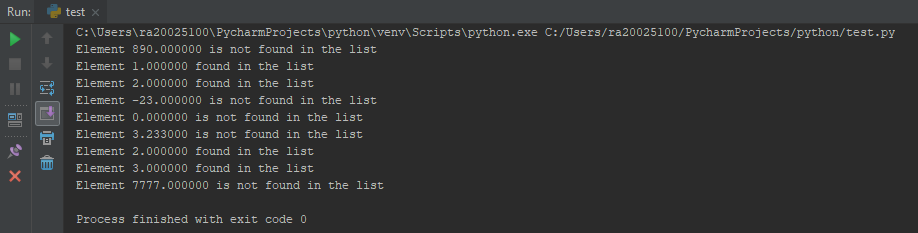
OUTPUT



43)CODE

def search(\*a):  
 list1 = [1 ,2 ,3 ,34 ,45 ,56 ,67 ,768 ,788 ,76 ,75]  
 for arg in a:  
 for i in list1:  
 if arg == i:  
 print("Element %f found in the list" % arg)  
 break  
 else:  
 print("Element %f is not found in the list" % arg)  
  
search(890)  
search(1 ,2)  
search(-23)  
search(00)  
search(3.233)  
search(2 ,3 ,7777)

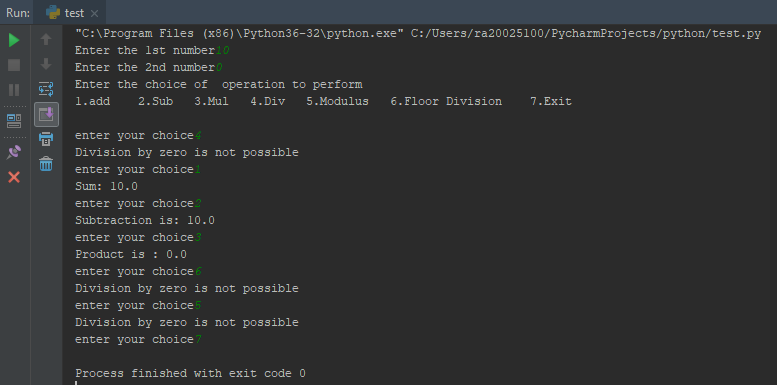
OUTPUT:



44) CODE

num1 = float(input("Enter the 1st number"))  
num2 = float(input("Enter the 2nd number"))  
  
print("Enter the choice of operation to perform ")  
print("1.add\t 2.Sub\t 3.Mul\t 4.Div\t 5.Modulus\t 6.Floor Division\t 7.Exit\n")  
  
while(1):  
 ip = int(input("enter your choice"))  
 if ip == 1:   
 add = lambda num1,num2 : num1+num2  
 print("Sum:", add(num1,num2))  
 elif ip == 2:  
 sub = lambda num1,num2 : num1-num2  
 print("Subtraction is:", sub(num1,num2))  
 elif ip == 3:  
 mul = lambda num1,num2 : num1\*num2  
 print("Product is :", mul(num1,num2))  
 elif ip == 4:  
 if not num2 == 0:  
 div = lambda num1,num2 : num1/num2  
 print("Division is:", div(num1,num2))  
 else:  
 print("Division by zero is not possible")  
 elif ip == 5:  
 if not num2 == 0:   
 modulus = lambda num1,num2 : num1%num2  
 print("Modulus:", modulus(num1,num2))  
 else:  
 print("Division by zero is not possible")  
 elif ip == 6:  
 if not num2 == 0:  
 floor\_div = lambda num1,num2 : num1//num2  
 print("Result of floor division:", floor\_div(num1,num2))  
 else:   
 print("Division by zero is not possible")  
 elif ip == 7:  
 break  
 else:  
 print("Enter the choice of operation to perform ")  
 break

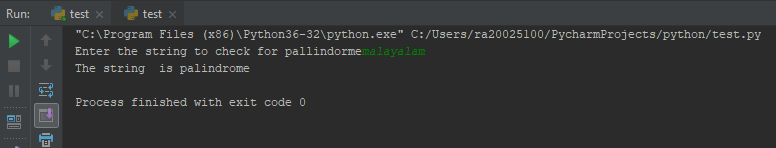
OUTPUT



45) CODE

ip=input("Enter the string to check for pallindorme")  
def palindrome(ip):  
 len1= len(ip)  
 if ip == ip[::-1]:  
 print("The string is palindrome" )  
 else:  
 print("The string is not palindrome")  
  
palindrome(ip)

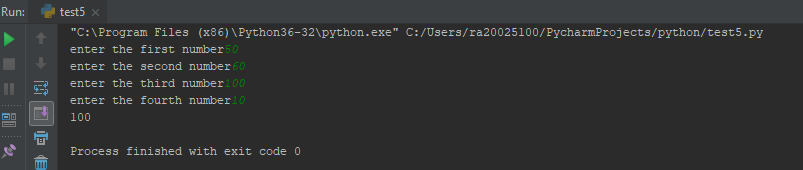
OUTPUT



46)CODE

num1 = int(input("enter the first number"))  
num2 = int(input("enter the second number"))  
num3 = int(input("enter the third number"))  
num4 = int(input("enter the fourth number"))  
  
def bignumber(num1=10,num2=20,num3=30,num4=40):  
 print(max(num1,num2,num3,num4))  
  
bignumber(num1,num2,num3,num4)

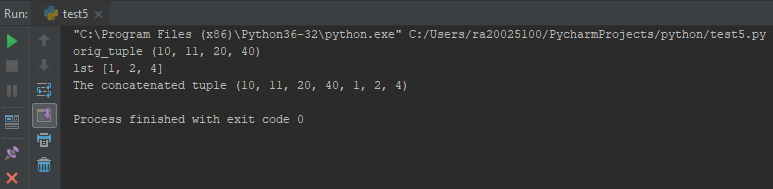
OUTPUT



47)CODE

orig\_tuple=(10,11,20,40)  
lst=[1,2,4]  
print("orig\_tuple",orig\_tuple)  
print("lst",lst)  
def concat\_tuple(a,b):  
 b=tuple(b)  
 c=a+b  
 print("The concatenated tuple",c)  
concat\_tuple(orig\_tuple,lst)

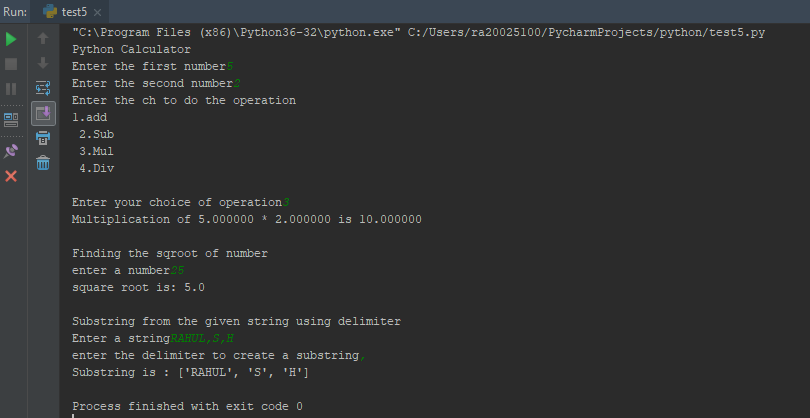
OUTPUT



48)CODE

import math  
  
print("Python Calculator")  
num1 = float(input("Enter the first number"))  
num2 = float(input("Enter the second number"))  
  
print("Enter the ch to do the operation ")  
print("1.add\n 2.Sub\n 3.Mul\n 4.Div\n")  
  
  
def add(num1, num2):  
 return num1 + num2  
  
  
def sub(num1, num2):  
 return num1 - num2  
  
  
def mul(num1, num2):  
 return num1 \* num2  
  
  
def div(num1, num2):  
 return num1 / num2  
  
  
def sqrt(num):  
 return math.sqrt(num)  
  
  
def substr(string, delim):  
 sub = string.split(delim)  
 return sub  
  
  
ch = int(input("Enter your choice of operation"))  
if ch == 1:  
 num3 = add(num1, num2)  
 print("Addition of %f + %f is %f" % (num1, num2, num3), "\n")  
elif ch == 2:  
 num3 = sub(num1, num2)  
 print("Subtraction of %f - %f is %f" % (num1, num2, num3), "\n")  
elif ch == 3:  
 num3 = mul(num1, num2)  
 print("Multiplication of %f \* %f is %f" % (num1, num2, num3), "\n")  
elif ch == 4:  
 if not num2 == 0:  
 num3 = div(num1, num2)  
 print("Division of %f / %f is %f" % (num1, num2, num3), "\n")  
 else:  
 print("Division by zero error\n")  
else:  
 print("Enter the input within the given chs \n")  
  
print("Finding the sqroot of number")  
sqnum = float(input("enter a number"))  
print("square root is:", sqrt(sqnum), "\n")  
  
print("Substring from the given string using delimiter")  
string = input("Enter a string")  
delim = input("enter the delimiter to create a substring")  
substring = substr(string, delim)  
print("Substring is :", substring)

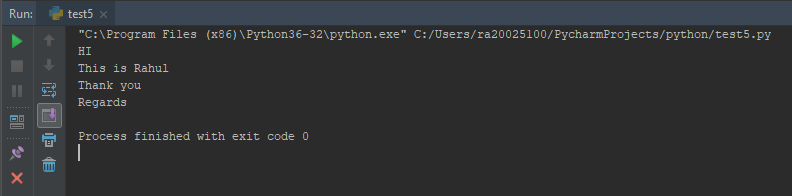
OUTPUT



49)CODE

f=open("rsh.txt", "r")  
print(f.read())  
f.close()  
  
f=open("rsh2.txt","w")  
f.write("Lorem Ipsum is simply dummy text of the printing and typesetting industry\n "  
 "Lorem Ipsum is simply dummy text of the printing and typesetting industry\n"  
 "Lorem Ipsum is simply dummy text of the printing and typesetting industry\n"  
 "Lorem Ipsum is simply dummy text of the printing and typesetting industry\n"  
 "Lorem Ipsum is simply dummy text of the printing and typesetting industry\n")  
f.close()  
  
f=open("rsh2.txt","a")  
f.write("Lorem Ipsum is simply dummy text of the printing and typesetting industry\n "  
 "Lorem Ipsum is simply dummy text of the printing and typesetting industry\n"  
 "Lorem Ipsum is simply dummy text of the printing and typesetting industry\n"  
 "Lorem Ipsum is simply dummy text of the printing and typesetting industry\n"  
 "Lorem Ipsum is simply dummy text of the printing and typesetting industry\n")  
f.close()

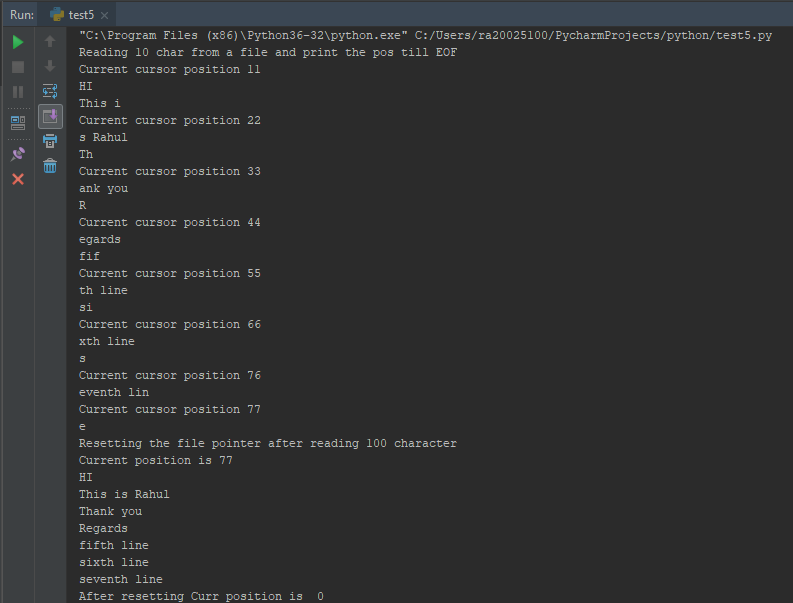
OUTPUT



50)CODE

print("Reading 10 char from a file and print the pos till EOF")  
f = open("rsh.txt", "r")  
while 1:  
 c=f.read(10)  
 if ""==c:  
 break  
 print("Current cursor position",f.tell())  
 print(c)  
  
print("Resetting the file pointer after reading 100 character")  
f.seek(0,0)  
c = f.read(100)  
print("Current position is",f.tell())  
print(c)  
f.seek(0,0)  
print("After resetting Curr position is " ,f.tell())  
  
print("Printing contents from 5th line onwards")  
cnt=0  
for l in f:  
 cnt+=1  
 if cnt>=5:  
 print(l)

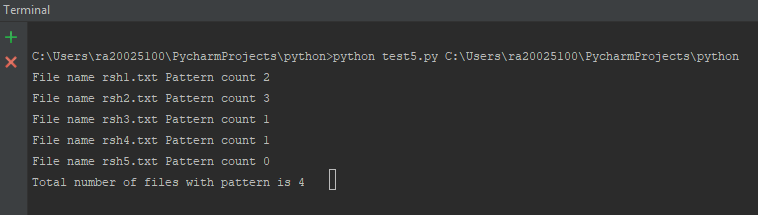
OUTPUT



51)CODE

import os  
import sys  
  
fil\_array=[]  
ldir=os.listdir(sys.argv[1])  
file\_count=0  
for i in ldir:  
 if i.endswith(".txt"):  
 fil\_array.append(os.path.join(sys.argv[1], i))  
for i in fil\_array:  
 pattern\_count=0  
 f=open(i,"r")  
 for l in f:  
 if "Treasure" in l:  
 pattern\_count+=1  
 if pattern\_count>0:  
 file\_count+=1  
 print("File name",os.path.basename(i),"Pattern count",pattern\_count)  
print("Total number of files with pattern is",file\_count)

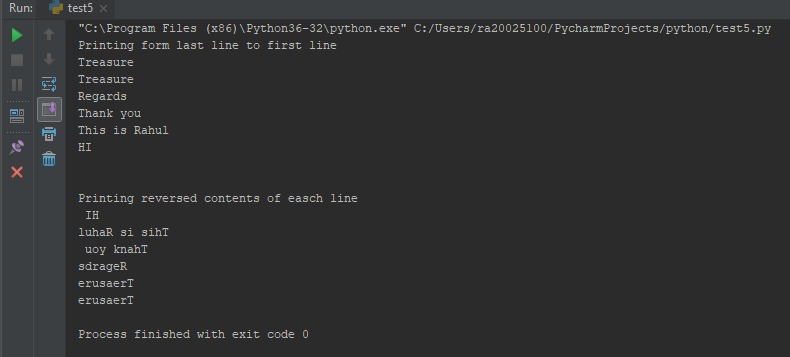
OUTPUT



52)CODE

strlist=[]  
f=open("rsh1.txt","r")  
for l in f:  
 a=l.split("\n")[0]  
 strlist.append(a)  
print("Printing form last line to first line")  
for i in range(len(strlist)-1,-1,-1):  
 print(strlist[i])  
print("\n")  
print("Printing reversed contents of easch line")  
for i in range(len(strlist)):  
 temp=strlist[i][::-1]  
 print(temp)

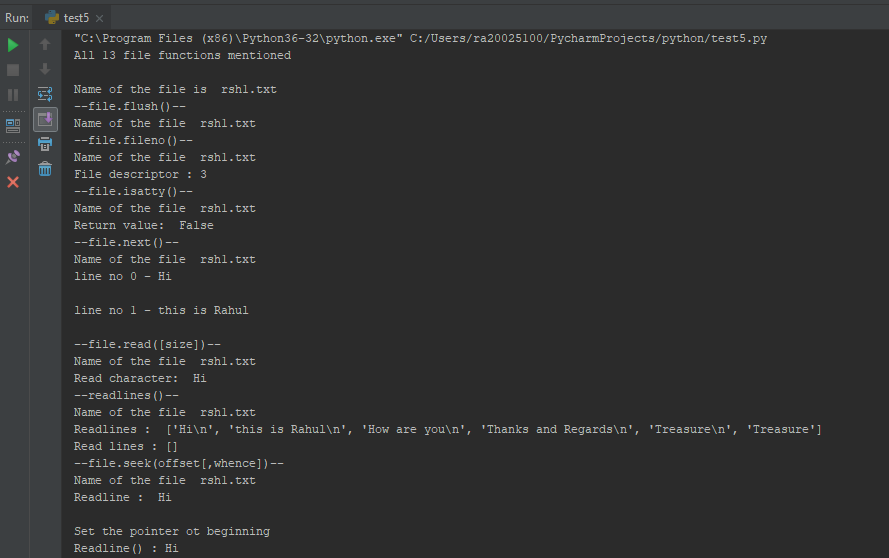
OUTPUT

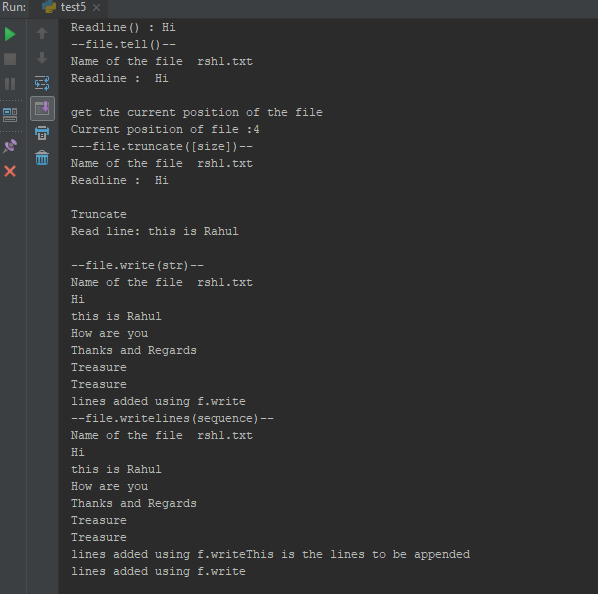


53)CODE

print ("All 13 file functions mentioned \n")  
  
f = open("rsh1.txt", "r+")  
print ("Name of the file is ", f.name)  
f.close()  
  
print ("--file.flush()--")  
f = open("rsh1.txt","r+")  
print ("Name of the file ", f.name)  
f.flush()  
f.close()  
  
  
print ("--file.fileno()--")  
f = open("rsh1.txt", "r+")  
print ( "Name of the file ", f.name)  
fid = f.fileno()  
print ("File descriptor :", fid)  
f.close()  
  
print("--file.isatty()--")  
f = open("rsh1.txt", "r+")  
print ("Name of the file ", f.name)  
ret = f.isatty()  
print ("Return value: ",ret)  
f.close()  
  
print("--file.next()--")  
f = open("rsh1.txt", "r+")  
print("Name of the file ", f.name)  
for index in range(2):  
 line = next(f)  
 print ("line no %d - %s" % (index, line))  
  
f.close()  
  
print("--file.read([size])--")  
f = open("rsh1.txt", "r+")  
print ("Name of the file ", f.name)  
line = f.read(2)  
print( "Read character: " ,line)  
f.close()  
  
print("--readlines()--")  
f = open("rsh1.txt", "r+")  
print ("Name of the file ", f.name)  
line = f.readlines()  
print ("Readlines : %s" %(line))  
line = f.readlines(4)  
print ("Read lines : %s" %(line))  
f.close()  
  
print("--file.seek(offset[,whence])--")  
f = open("rsh1.txt", "r+")  
print ("Name of the file ", f.name)  
line = f.readline()  
print ("Readline : %s" %(line))  
print ("Set the pointer ot beginning")  
f.seek(0,0)  
line=f.readline(2)  
print ("Readline() : %s " %(line))  
f.close()  
  
  
print("--file.tell()--")  
f = open("rsh1.txt", "r+")  
print ("Name of the file ", f.name)  
line = f.readline()  
print ("Readline : %s" %(line))  
print ("get the current position of the file")  
pos = f.tell()  
print( "Current position of file :%d" %(pos))  
f.close()  
  
print("---file.truncate([size])--")  
f = open("rsh1.txt", "r+")  
print ("Name of the file ", f.name)  
line = f.readline()  
print ("Readline : %s" %(line))  
print ("Truncate")  
f.truncate()  
line = f.readline()  
print( "Read line: %s" %(line))  
f.close()  
  
print("--file.write(str)--")  
f = open("rsh1.txt", "r+")  
print ("Name of the file ", f.name)  
string = "\nlines added using f.write"  
f.seek(0,2)  
line = f.write(string)  
  
f.seek(0,0)  
lines=f.read()  
print( lines)  
f.close()  
  
print("--file.writelines(sequence)--")  
f = open("rsh1.txt", "a+")  
print ("Name of the file ", f.name)  
stri= ["This is the lines to be appended", "\nlines added using f.write"]  
f.seek(0,2)  
line=f.writelines(stri)  
  
f.seek(0,0)  
lines=f.read()  
print (lines)  
f.close()

OUTPUT

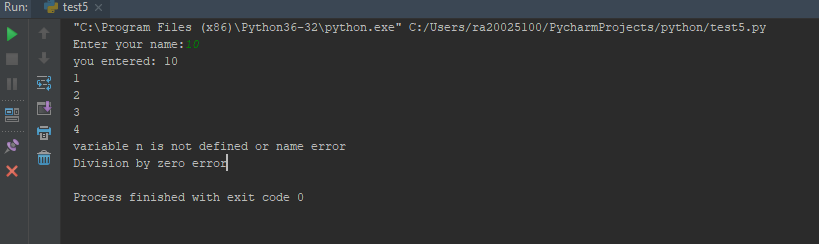




54)CODE

import time  
try:  
 name = input("Enter your name:")  
 print("you entered: " + name)  
except KeyboardInterrupt:  
 print( "Keyboard interrupt you hit ctrl-c")  
  
i=1  
try:  
 while(i<5):  
 time.sleep(1)  
 print (i)  
 i += 1  
except KeyboardInterrupt:  
 print( "Keyboard interrupt")  
  
  
try:  
 print ("Hi" +" "+ n)  
except NameError:  
 print ("variable n is not defined or name error")  
  
#arithmetic error  
try:  
 a=89/0  
except ArithmeticError:  
 print ("Division by zero error")

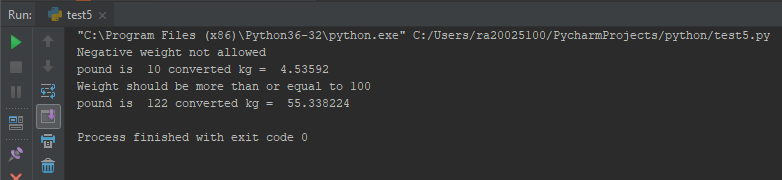
OUTPUT



55)CODE

def poundtokg(p):  
 try:  
 assert (p >= 0), "Negative weight not allowed"  
 print("pound is ", p, "converted kg = ", (p \* 0.453592))  
 except AssertionError as error:  
 print(error)  
  
poundtokg(-11)  
poundtokg(10)  
  
def pound100kg(p):  
 try:  
 assert(p>=100), "Weight should be more than or equal to 100"  
 print("pound is ", p, "converted kg = ",(p\*0.453592))  
 except AssertionError as e:  
 print(e)  
  
pound100kg(77)  
pound100kg(122)

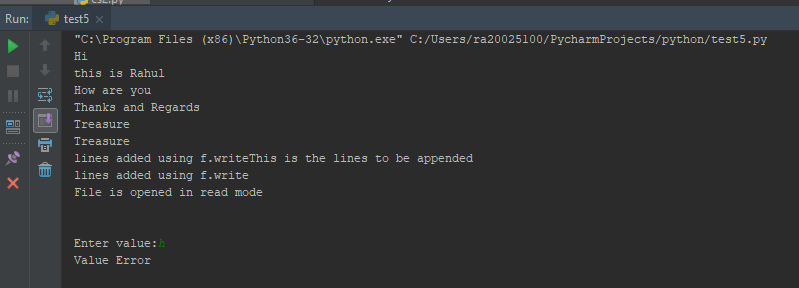
OUTPUT



56)CODE

try:  
 f = open("rsh1.txt", "r")  
 print(f.read())  
 f.write("Hello")  
except IOError:  
 print("File is opened in read mode")  
  
print("\n")  
try:  
 num = int(input("Enter value:"))  
 print("Printing the entered value", num)  
except ValueError:  
 print("Value Error")

OUTPUT



57)CODE

import math  
import sys  
  
try:  
 num = a/2  
 print ("num is ", num)  
except Exception:  
 print("Exception - base class for all exception")  
  
#Arithmetic errror  
try:  
 div = 2/0  
except ArithmeticError:  
 print( "Arithmetic error exception")  
  
#StopIteration  
try:  
 #z=[1,2]  
 f1=open("rsh1.txt")  
 for i in range(1,4):  
 print( next(f1))  
except StopIteration:  
 print("Stop Iteration Exception")  
f1.close()  
  
  
#SystemExit  
try:  
 sys.exit()  
except SystemExit:  
 print("System exit exception")  
  
#Overflow error  
try:  
 a=math.exp(3456/3456789\*2345670987)  
 print("a is ",a)  
except OverflowError:  
 print("Overflow error exception")  
  
#Value Error  
try:  
 a=math.sqrt(-44/5)  
 print("a is ",a)  
except ValueError:  
 print("Value error")  
  
#Zero division error  
try:  
 num = 56789/0  
 print("num is ", num)  
except ZeroDivisionError:  
 print("ZeroDivisionError - cannot divide by zero")  
  
#assertion error  
try:  
 val = 9  
 assert(val<9), "Value should be less than 9"  
except AssertionError as err:  
 print(err)  
  
#Attribute error  
try:  
 i = 9  
 i.append(4)  
except AttributeError:  
 print("AttributeError - integer has not attribute append")  
  
  
#EOF error  
try:  
 f2=open("rsh1.txt","r")  
 f2.read()  
except EOFError:  
 print("EOF error")  
  
#import error  
try:  
 import sample  
except ImportError:  
 print("import error exception")  
  
#look up error  
dict={'empid':'empname'}  
try:  
 print (dict['name'])  
except LookupError:  
 print('lookup error')  
  
#index error  
try:  
 l1=[1,2,3,4]  
 print("list l1 is ", l1)  
 print("list l1[9]", l1[9])  
except IndexError:  
 print("IndexoutofBound")  
  
#name error:  
try:  
 print( name\_n)  
except NameError:  
 print("name error")  
  
#keyboard interrupt  
try:  
 name = input("Enter your name:")  
 print("you entered: " + name)  
except KeyboardInterrupt:  
 print( "Keyboard interrupt you hit ctrl-c")  
  
#Key error  
try:  
 print("dict is ", dict['2'])  
except:  
 print("key error")  
  
#unbound local error  
counter=0  
def incre():  
 try:  
 counter+=1  
 except UnboundLocalError:  
 print("Unbound local error")  
  
incre()  
  
#IOError  
try:  
 f4=open('rsh1.txt',"r")  
 f4.write("Write at the end")  
except IOError:  
 print("Writnig in a file opened in read mode, IOError")  
  
#notimplemented error  
try:  
 class shape(object):  
 @property  
 def rect(self):  
 raise NotImplementedError("Notimplemented error - subclass should implement this method")  
  
 s=shape()  
 print(s.rect)  
except Exception as e:  
 print(e)  
  
#runtime error  
try:  
 raise RuntimeError  
except RuntimeError:  
 print("runtime error occured")  
  
#typeerror  
try:  
 stri="hello"  
 stri = stri-1  
except TypeError:  
 print("Type error")  
  
#value error:  
try:  
 stri= "hey"  
 print(int(stri))  
except ValueError:  
 print("value error")

OUTPUT

