SUBMITTED TO: Mr. MUHAMMAD ZAHID SIR (CS Faculty)

SUBMITTED BY:

**A Project Report File for Partial Fulfillment of All India Senior School Certificate (AISSC) Practical Examination 2020-21**

**BACK – END**

MYSQL

**FRONT – END**

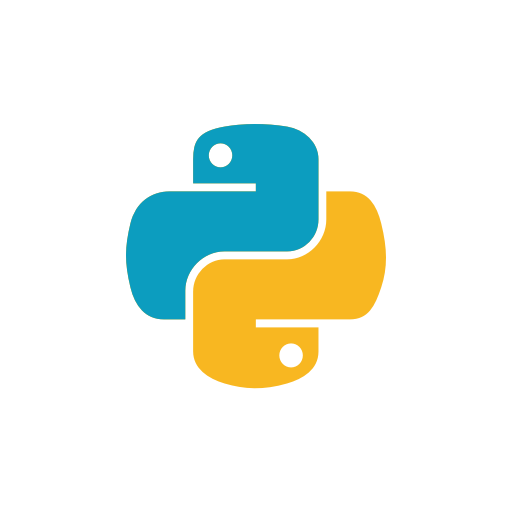
PYTHON

**A PYTHON BASED PROJECT**

**COMPUTER SCIENCE PROJECT REPORT FILE**

COLUMBIA CONVENT







***Divyansh***

***Sharma***

**CERTIFICATE**

This is certify that **Master DIVYANSH SHARMA** studying in XI has successfully completed the **COMPUTER SCIENCE NEW (083) PRACTICAL FILE** based on the syllabus and given a satisfactory account of it. This report contains record of practical program work (on Python Programming).

Signature Signature

Internal Examiner External Examiner

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Principal’s Sign & Seal

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGEMENT**

Without acknowledging the contribution and suggestion received from numerous sources, this project work could not have been completed.

It is our great pleasure and privilege to express profound indebtedness to our **Principal Mam** **Mrs. Nalini Pal** and **Subject Faculty Mr. Muhammad Zahid Sheikh** for his kind support, cooperation and guidance during the course of the project.

We would also like to thank, each and every faculty member of the institute from bottom of our heart for their kind and humble support during the tenure of the project.

Our effort is an outcome of blessing and able guidance and timely teaching extended to us by our respective teachers and valuable guidance provided by them in successful completion of this project.

Many people have directly or indirectly contributed to the development of this project. We wish to thank them all.

**Contents**

* *5*
* *16*
* *27*
* *31*
* *37*
* *44*
* *54*
* *63*
* *Program based on ‘USER DEFINED FUNCTIONS’*
* *Program based on ‘BUILT IN FUNCTIONS’*
* *‘RECURSION’*
* *‘LIBRARY & MODULES’*
* *‘FILE HANDLING’*
* *‘LINEAR LIST’*
* *‘Python – MySQL CONNECTIVITY’*
* *My SQL Queries*

User-Defined

FUNCTIONS

*ARITHMETIC OPERATIONS*

*SIMPLE INTEREST*

*TEMPERATURE CONVERSION*

*CALCULATING ROOTS*

*PRIME CHECKER*

*FACTORIAL CALCULATER*

TAKE 2 VALUES AND RETURN ALL ARITHMETIC OPERATIONS

def Arith(a,b):

return(a+b,a-b,a\*b,a/b,a\*\*b,a%b)

a=int(input("\tEnter the First Number --> "))

b=int(input("\n\tEnter the Second number --> "))

add,sub,mul,div,power,rem=Arith(a,b)

print('\n\tAddition of ',a,'and',b,'is',add)

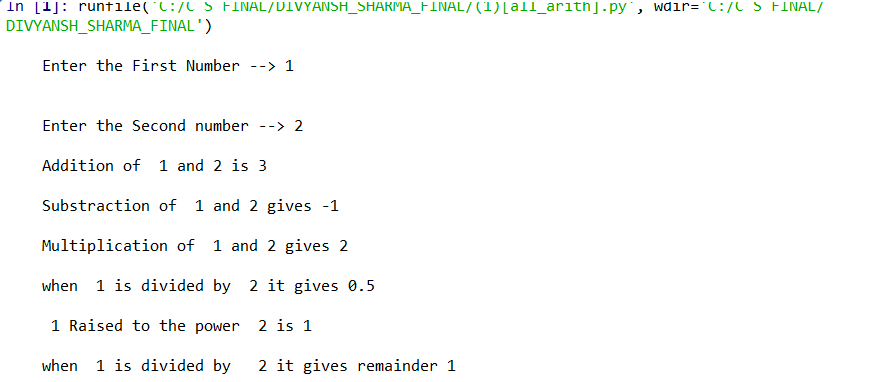
print('\n\tSubstraction of ',a,'and',b,'gives',sub)

print('\n\tMultiplication of ',a,'and',b,'gives',mul)

print('\n\twhen ',a,'is divided by ',b,'it gives',div)

print('\n\t',a,'Raised to the power ',b,'is',power)

print('\n\twhen ',a,'is divided by ',b,'it gives remainder',rem)



FUNCTION FOR CALCULATING SIMPLE INTREST

# Function for simple intrest

def si(p,r,t):

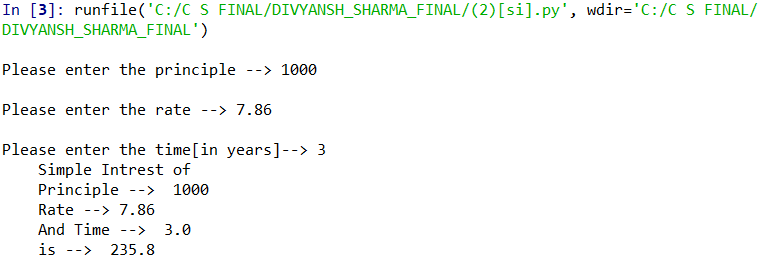
return((p\*r\*t)/(100))

p=int(input('Please enter the principle --> '))

r=float(input('Please enter the rate --> '))

t=float(input('Please enter the time[in years]--> '))

print('\tSimple Intrest of \n\tPrinciple --> ',p,'\n\tRate -->',r,'\n\tAnd Time --> ',t,'\n\tis --> ',si(p,r,t))

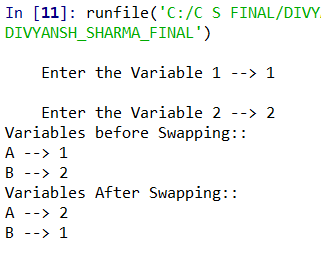


SWAPPING USING TEMP VARIABLE

def Swap(a,b):

temp=a

a=b



b=temp

return(a,b)

a=input("\tEnter the Variable 1 --> ")

b=input("\tEnter the Variable 2 --> ")

print('Variables before Swapping::')

print('A -->',a)

print('B -->',b)

a,b=Swap(a,b)

print('Variables After Swapping::')

print('A -->',a)

print('B -->',b)

TEMPERATURE CONVERSION FUNCTION

## Temperature conversion program

def c\_to\_f(c):

return(((c)\*(9/5)+32))

def f\_to\_c(f):

return((f-32)\*(5/9))

print("\t\t##### TEMPERATURE CONVERSION ######")

print('1--> To give the temperature value in Celcius and get its Fahrenheit value --> ')

print('2--> To give the temperature value in Fahrenheit and get its Celcius value --> ')

i=int(input("Enter the Choice --> "))

if i==1:

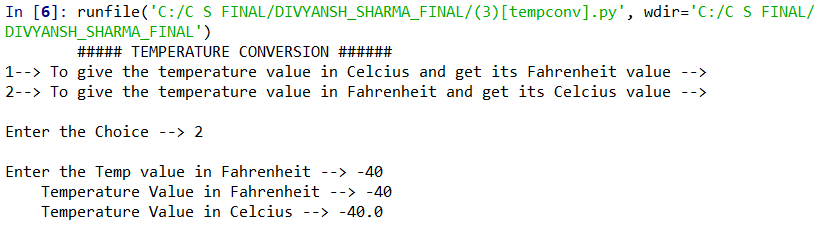
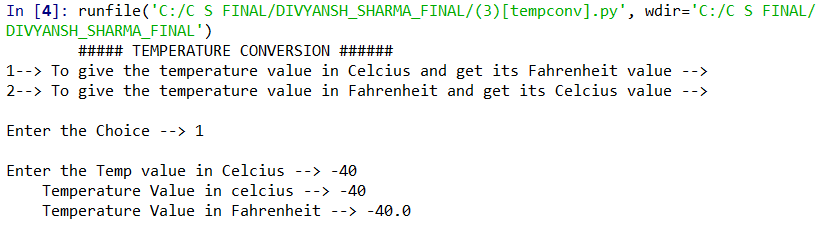
c=int(input("Enter the Temp value in Celcius --> "))

print('\tTemperature Value in celcius -->',c,'\n\tTemperature Value in Fahrenheit -->',c\_to\_f(c))

if i==2:

f=int(input("Enter the Temp value in Fahrenheit --> "))

print('\tTemperature Value in Fahrenheit -->',f,'\n\tTemperature Value in Celcius -->',f\_to\_c(c))



PERIMETER & AREAS OF DIFFERENT MATHEMATICAL FIGURES

[Square, Circle, Rectangle & Triangle]

c=(input("Enter the figure \n 1->SQUARE \n 2->CRICLE \n 3->RECTANGLE \n 4->TRIANGLE \n --> "))

def square(a):

ps=4\*a

pa=a\*\*2

return(ps,pa)

def circle(r):

pc=2\*(3.14)\*r

ac=(3.14)\*(r\*\*2)

return(pc,ac)

def rect(l,b):

pr=2\*(l+b)

ar=l\*b

return(pr,ar)

def tri(b,h):

pt=b+2\*(((b/2)\*\*2)+(h\*\*2))

at=0.5\*b\*h

return(pt,at)

if c=="1":

a=float(input("Enter the side of the square -> "))

t=square(a)

print("\n Side of square is ->",a)

print("\n Perimeter of square is->",t[0])

print("\n Area of square is->",t[1])

elif c=="2":

r=float(input("Enter the radius of the CIRCLE -> "))

t=circle(r)

print("Radius of the cicle is ->",r)

print("Perimeter of circle is->",t[0])

print("Area of circle is ->",t[1])

elif c=="3":

l=float(input("Enter the length of rectangle -> "))

b=float(input("Enter the breadth of rectangle -> "))

t=rect(l,b)

print("Length is ->",l,"\nBreadth is -> ",b)

print("Perimeter of rectangle is ->",t[0])

print("Area of rectangle is ->",t[1])

elif c=="4":

b=float(input("Enter the base length of the triangle -> "))

h=float(input("Enter the height of the triangle ->"))

t=tri(b,h)

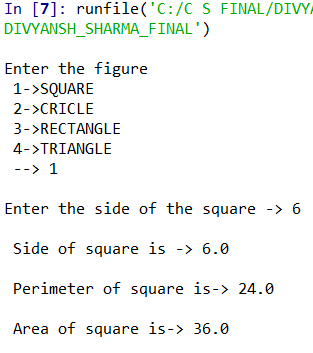
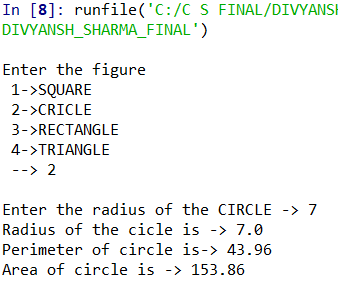
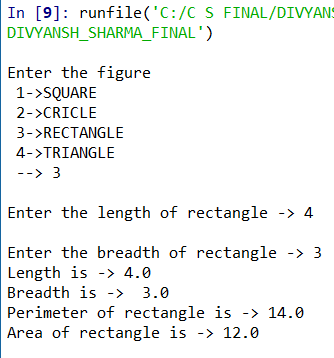
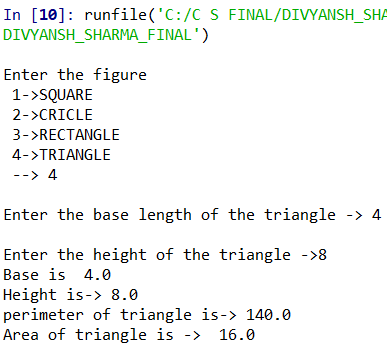
print("Base is ",b,"\nHeight is->",h)

print("perimeter of triangle is->",t[0])

print("Area of triangle is -> ",t[1])

else:

print("INVALID INPUT !!!")

reversing a number

def revno(n):

i=10

revno=0

while n!=0:

cou=1

k=n%10

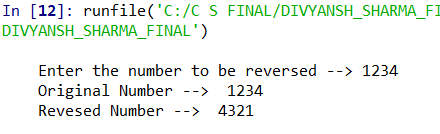
revno+=k\*(i\*\*((len(str(n))-cou)))

n//=10

return(revno)

n=int(input('\tEnter the number to be reversed --> '))

print('\tOriginal Number --> ',n,"\n\tRevesed Number --> ",revno(n))



squaring an d cubing using functions

def power(n):

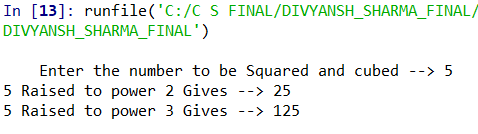
return(n\*\*2,n\*\*3)

i=int(input("\tEnter the number to be Squared and cubed --> "))

a,b=power(i)

print(i,'Raised to power 2 Gives -->',a)

print(i,'Raised to power 3 Gives -->',b)



Simple intrest using default argument values

def si(p=100000,r=6.7,t=1):

return((p\*t\*r)/100)

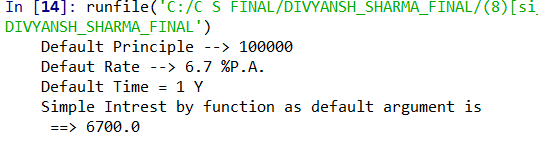
print("\tDefault Principle --> 100000")

print("\tDefaut Rate --> 6.7 %P.A.")

print("\tDefault Time = 1 Y")

print("\tSimple Intrest by function as default argument is")

print('\t ==>',si())



quadratic equation – (roots & their types)

def quadratic(a,b,c):

d=b\*\*2-(4\*a\*c)

if d==0:

return(d,(-b+(d)/2\*a),(-b-(d)/2\*a))

elif d<0:

return(d,False,False)

elif d>0:

return(d,(-b+(d)/2\*a),(-b-(d)/2\*a))

print('Equation of the Type a (x\*\*2) + b x + c is called Quadratic Equation \n\t(Where a,b,c are constants and can be negative) ')

print("Please Enter the Value of A,B,C -->")

a=int(input("A -->"))

b=int(input("B -->"))

c=int(input("C -->"))

d,r1,r2=quadratic(a,b,c)

if r1==False:

print("Sorry This equation contain no real roots")

elif d>0:

print("This Equation has Two distinct real roots ")

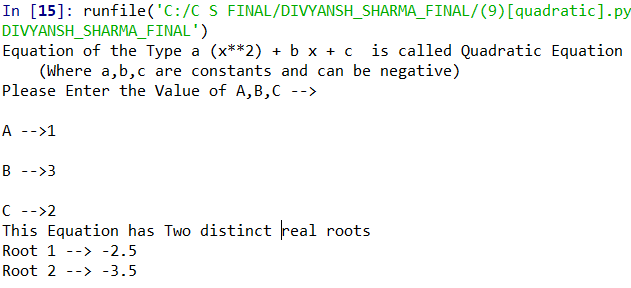
print('Root 1 -->',r1)

print('Root 2 -->',r2)

elif d==0:

print("Thia is a special Quadratic where both thhe roots are same i.e.")

print('Root 1 = Root 2 ==>',r1)



Function for factorial

def fact(n):

if n==0:

return(1)

else:

fact=1

for i in range(2,n+1):

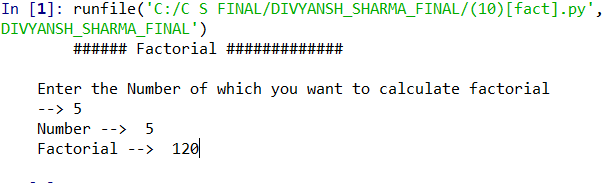
fact\*=i

return(fact)

print("\t\t###### Factorial #############")

n=int(input("\tEnter the Number of which you want to calculate factorial \n\t--> "))

print("\tNumber --> ",n,"\n\tFactorial --> ",fact(n))



Prime Number Checking

def prime(n):

i=2

is\_prime=True

while i<n:

rem=n%i

if rem==0:

is\_prime=False

break

else:

i+=1

return(is\_prime)

n=int(input("Enter the number of which you want to check is prime or not -->"))

if n==1:

print("It is not a prime number it is a special number")

elif n==0:

print("Sorry not concludable")

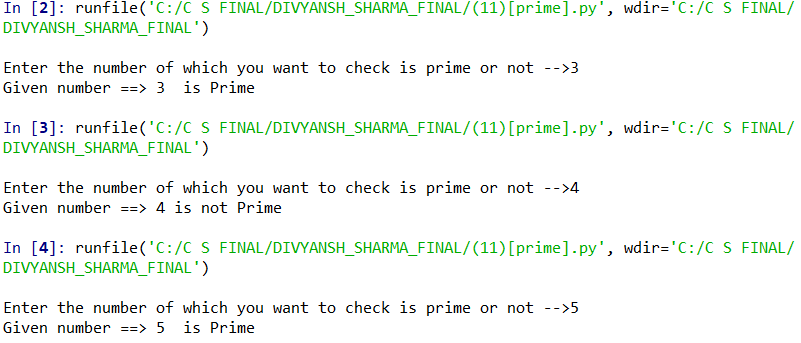
else:

if prime(n) == True:

print('Given number ==>',n,' is Prime')

else:

print("Given number ==>",n,"is not Prime")



string counter

[ vowel , consonants, digit , spaces & special symbol ]

def strchk(sk):

alphabet=consonents=vowel=digit=space=special=0

for f in sk:

isapl=f.isalpha()

isdigit=f.isdigit()

if isapl==True:

alphabet+=1

if f==str("a") or f==str("e") or f==str("i") or f==str("o") or f==str("u") :

vowel+=1

else:

consonents+=1

elif isdigit==True:

digit+=1

elif f==' ':

space+=1

else:

special+=1

return(alphabet,consonents,vowel,digit,space,special)

s=input("Please enter the String --> ").lower().strip()

a,c,v,d,sp,s=strchk(s)

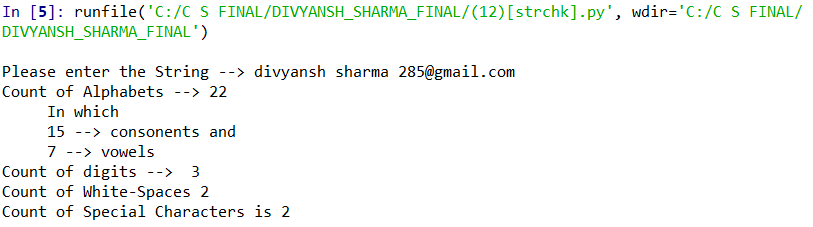
print("Count of Alphabets -->",a)

print('\t In which\n\t',c,'--> consonents and \n\t',v,'--> vowels')

print('Count of digits --> ',d)

print('Count of White-Spaces',sp)

print('Count of Special Characters is',s)



Built-In

FUNCTIONS

*MATH FUNCTION*

*CONVERSION FUNCTION*

*STRING FUNCTION*

conversion functions

print("\t\t#####Conversion Functions######")

i=int(input("Enter An Integer --> "))

print("Currently",i," is Stored as",type(i) )

i=float(i)

print("variable converted to float ")

print("Currently",i," is Stored as",type(i) )

i=str(i)

print("variable converted to String")

print("Currently",i," is Stored as",type(i) )

fl=float(input("Please enter value for float --> "))

print("Currently",fl," is Stored as",type(fl) )

print("Float value converted to integer")

fl=int(fl)

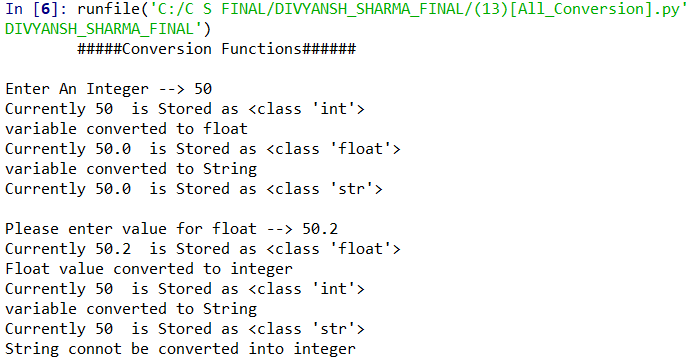
print("Currently",fl," is Stored as",type(fl) )

fl=str(fl)

print("variable converted to String")

print("Currently",fl," is Stored as",type(fl))

print("String connot be converted into integer")



Math functions

import math

print("PROGRAM TO DEMONSTRATE USE OF ALL MATH FUNCTIONS \n")

print("FUNCTION 1 - CEIL")

print("Return the ceiling of x as an Integral")

print("This is the smallest integer >= x.")

print("ex. ceil(13.1) ==> Gives 14")

y=str(input(" DO YOU WANT TO USE CEIL FUNCTION ? ( y/n) --> ")).lower()

if y=="y":

n=float(input("enter the value to be used with ceil function --> "))

print("RESULT ==>" , math.ceil(n))

print("FUNCTION 2 - FLOOR",)

print('Return the floor of x as an Integral')

print('This is the largest integer <= x.')

print(' EX. floor(13.1) ==> Gives 13 ')

y=str(input(" DO YOU WANT TO USE FLOOR FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value to be used with floor function : "))

print("RESULT =" , math.floor(n))

print("FUNCTION 3 - POWER")

print('Return x\*\*y (x to the power of y)')

print("EX. pow(2,3) ==> 2\*\*3 ==> 8")

y=str(input(" DO YOU WANT TO USE POW FUNCTION?(y/n)"))

if y=="y":

base=int(input("enter the base value : "))

exponent=int(input("enter the exponent value : "))

print("RESULT =" , math.pow(base,exponent))

print("FUNCTION 4 - SQUARE ROOT ")

print("It returns the square root of the given number ")

print("Ex. sqrt(4) ==> 2")

y=str(input(" DO YOU WANT TO USE SQRT FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value to rooted : "))

print("RESULT =" , math.sqrt(n))

print("FUNCTION 5 - Exponential ")

print("Return e raised to the power of x.")

print("EX. exp(1) ==> 2.718281828459045")

y=str(input(" DO YOU WANT TO USE EXPONENTAIL FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of argument : "))

print("RESULT =" , math.exp(n))

print("FUNCTION 6 - LOG")

print("It returns the natural loarithm for the num with given base ")

print('EX. log(8,2) ==> 3')

y=str(input(" DO YOU WANT TO USE LOG FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of num : "))

base=int(input("enter the base value of log : "))

print("RESULT =" , math.log(n,base))

print("FUNCTION 7 - LOG10")

print("It returns the natural loarithm for the num with base 10 ")

y=str(input(" DO YOU WANT TO USE LOG10 FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of num : "))

print("RESULT =" , math.log10(n))

print("FUNCTION 8 - FABS")

print("It returns the absolute value of num ")

y=str(input(" DO YOU WANT TO USE FABS FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of num : "))

print("RESULT =" , math.fabs(n))

print("FUNCTION 9 - SIN")

print("It returns the sin of argument ")

print("Ex. sin(1.5707963267948966) ==> 1")

y=str(input(" DO YOU WANT TO USE SIN FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of arg : "))

print("RESULT =" , math.sin(n))

print("FUNCTION 10 - COS")

print("It returns the cos of argument ")

print("Ex. cos(1.5707963267948966) ==> 0")

y=str(input(" DO YOU WANT TO USE COS FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of arg : "))

print("RESULT =" , math.cos(n))

print("FUNCTION 11 - TAN")

print("It returns the tan of argument ")

print('tan(0.7853981633974483) ==> 1')

y=str(input(" DO YOU WANT TO USE TAN FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of arg : "))

print("RESULT =" , math.tan(n))

print("FUNCTION 12 - DEGREES")

print("It converts radian to degrees ")

print("math.degrees(0.7853981633974483) ==> 45.0")

y=str(input(" DO YOU WANT TO USE DEGREES FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the angle : "))

print("RESULT =" , math.degrees(n))

print("FUNCTION 13 - RADIANS")

print("It converts degrees to radians ")

print("radians(45.0) ==> 0.7853981633974483")

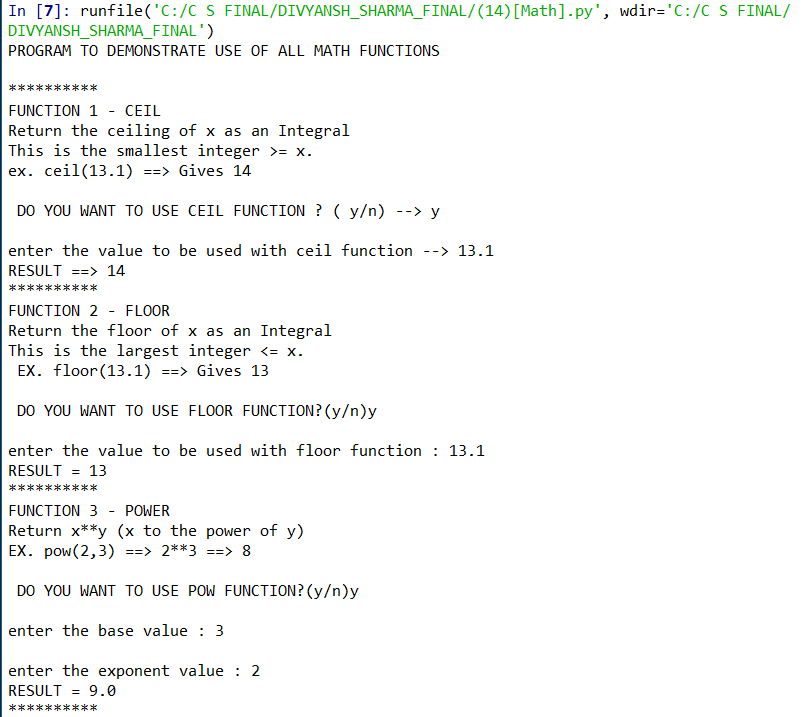
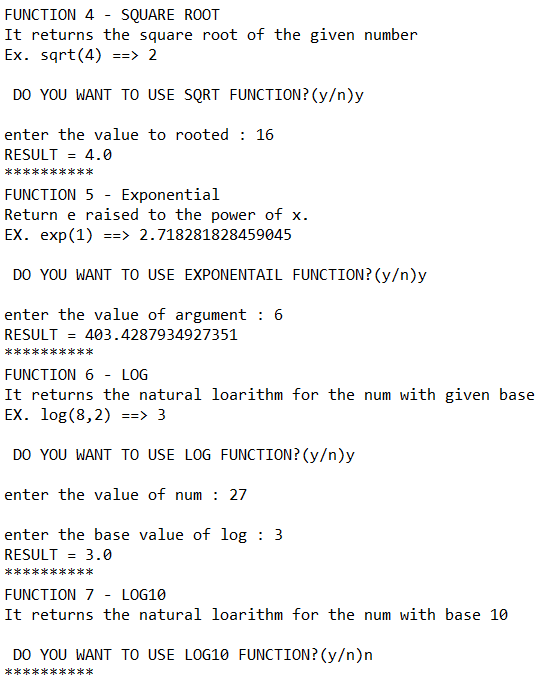
y=str(input(" DO YOU WANT TO USE RADIANS FUNCTION?(y/n)"))

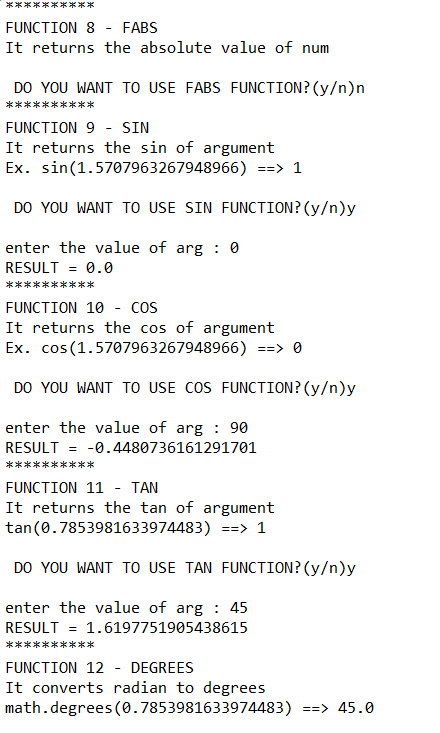
if y=="y":

n=float(input("enter the value of arg : "))

print("RESULT =" , math.radians(n))

output



STRING FUNCTIONS

import math

print("PROGRAM TO DEMONSTRATE USE OF ALL MATH FUNCTIONS \n")

print('\*'\*10)

print("FUNCTION 1 - CEIL")

print("Return the ceiling of x as an Integral")

print("This is the smallest integer >= x.")

print("ex. ceil(13.1) ==> Gives 14")

y=str(input(" DO YOU WANT TO USE CEIL FUNCTION ? ( y/n) --> ")).lower()

if y=="y":

n=float(input("enter the value to be used with ceil function --> "))

print("RESULT ==>" , math.ceil(n))

print('\*'\*10)

print("FUNCTION 2 - FLOOR",)

print('Return the floor of x as an Integral')

print('This is the largest integer <= x.')

print(' EX. floor(13.1) ==> Gives 13 ')

y=str(input(" DO YOU WANT TO USE FLOOR FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value to be used with floor function : "))

print("RESULT =" , math.floor(n))

print('\*'\*10)

print("FUNCTION 3 - POWER")

print('Return x\*\*y (x to the power of y)')

print("EX. pow(2,3) ==> 2\*\*3 ==> 8")

y=str(input(" DO YOU WANT TO USE POW FUNCTION?(y/n)"))

if y=="y":

base=int(input("enter the base value : "))

exponent=int(input("enter the exponent value : "))

print("RESULT =" , math.pow(base,exponent))

print('\*'\*10)

print("FUNCTION 4 - SQUARE ROOT ")

print("It returns the square root of the given number ")

print("Ex. sqrt(4) ==> 2")

y=str(input(" DO YOU WANT TO USE SQRT FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value to rooted : "))

print("RESULT =" , math.sqrt(n))

print('\*'\*10)

print("FUNCTION 5 - Exponential ")

print("Return e raised to the power of x.")

print("EX. exp(1) ==> 2.718281828459045")

y=str(input(" DO YOU WANT TO USE EXPONENTAIL FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of argument : "))

print("RESULT =" , math.exp(n))

print('\*'\*10)

print("FUNCTION 6 - LOG")

print("It returns the natural loarithm for the num with given base ")

print('EX. log(8,2) ==> 3')

y=str(input(" DO YOU WANT TO USE LOG FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of num : "))

base=int(input("enter the base value of log : "))

print("RESULT =" , math.log(n,base))

print('\*'\*10)

print("FUNCTION 7 - LOG10")

print("It returns the natural loarithm for the num with base 10 ")

y=str(input(" DO YOU WANT TO USE LOG10 FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of num : "))

print("RESULT =" , math.log10(n))

print('\*'\*10)

print("FUNCTION 8 - FABS")

print("It returns the absolute value of num ")

y=str(input(" DO YOU WANT TO USE FABS FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of num : "))

print("RESULT =" , math.fabs(n))

print('\*'\*10)

print("FUNCTION 9 - SIN")

print("It returns the sin of argument ")

print("Ex. sin(1.5707963267948966) ==> 1")

y=str(input(" DO YOU WANT TO USE SIN FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of arg : "))

print("RESULT =" , math.sin(n))

print('\*'\*10)

print("FUNCTION 10 - COS")

print("It returns the cos of argument ")

print("Ex. cos(1.5707963267948966) ==> 0")

y=str(input(" DO YOU WANT TO USE COS FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of arg : "))

print("RESULT =" , math.cos(n))

print('\*'\*10)

print("FUNCTION 11 - TAN")

print("It returns the tan of argument ")

print('tan(0.7853981633974483) ==> 1')

y=str(input(" DO YOU WANT TO USE TAN FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of arg : "))

print("RESULT =" , math.tan(n))

print('\*'\*10)

print("FUNCTION 12 - DEGREES")

print("It converts radian to degrees ")

print("math.degrees(0.7853981633974483) ==> 45.0")

y=str(input(" DO YOU WANT TO USE DEGREES FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the angle : "))

print("RESULT =" , math.degrees(n))

print('\*'\*10)

print("FUNCTION 13 - RADIANS")

print("It converts degrees to radians ")

print("radians(45.0) ==> 0.7853981633974483")

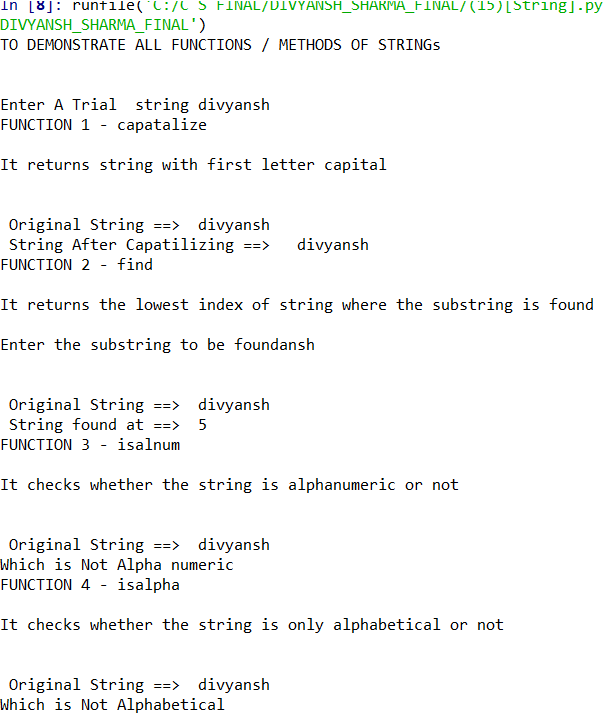
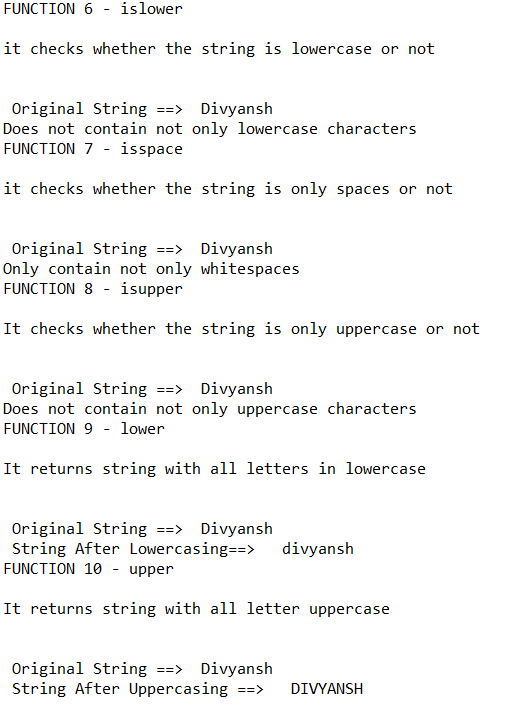
y=str(input(" DO YOU WANT TO USE RADIANS FUNCTION?(y/n)"))

if y=="y":

n=float(input("enter the value of arg : "))

print("RESULT =" , math.radians(n))

output

Recursion

FUNCTIONS

*SUM OF N No.’s*

*FACTORIAL OF N*

*FIBBONACCI SERIES*

*BINARY SEARCH*

SUM ON N NATURAL NUMBER USING RECURSION

def add(n):

if n==1:

return (1)

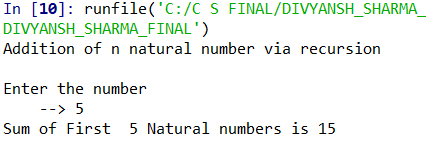
else:

return (n+add(n-1))

print("Addition of n natural number via recursion")

i=int(input("Enter the number \n\t--> "))

print("Sum of First ",i,"Natural numbers is",add(i))



Factorial using recursion

def fact(n):

if n==1:

return (1)

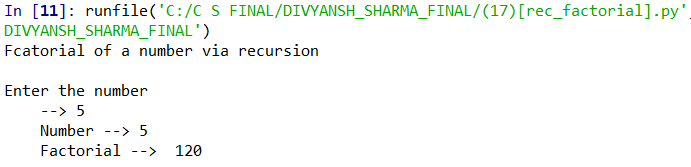
else:

return (n\*fact(n-1))

print("Fcatorial of a number via recursion")

i=int(input("Enter the number \n\t--> "))

print("\tNumber -->",i,"\n\tFactorial --> ",fact(i))



fibbonacci series using recursion

def fibo(n):

if n<=1:

return n

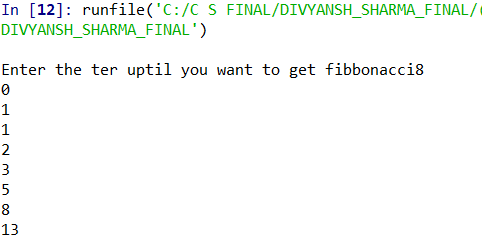
else:

return(fibo(n-1) + fibo(n-2))

n=int(input("Enter the ter uptil you want to get fibbonacci"))

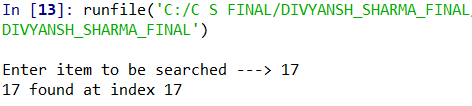
for i in range(n):

print(fibo(i))



binary search using recursion

def bins(lst,key):



low = 0

high=len(lst)-1

while low<=high:

mid=int((high+low)/2)

if key==lst[mid]:

return (mid)

elif key<lst[mid]:

high=mid-1

else:

low=mid+1

else:

return -999

lst=[f for f in range(0,20)]

item=int(input("Enter item to be searched ---> "))

res=bins(lst,item)

if res>0:

print(item,"found at index",res)

else:

print("Sorry! Item not found")

Library & Modules

FUNCTIONS

*MATH FUNCTION USING ‘MATH ’ MODULE*

*RANDOM LIBRARY*

*MODULE FOR TEMPERATURE CONVERSION*

*AERA & PERIMETER CALCULATION MODULE*

MATH FUNCTION BY INPORTING MATH LIBRARY

import math

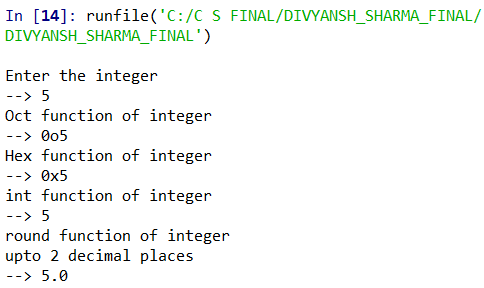
i=float(input("Enter the integer \n--> "))

print("Oct function of integer \n-->",oct(int(i)))

print("Hex function of integer \n-->",hex(int(i)))

print("int function of integer \n-->",int(i))

print("round function of integer \nupto 2 decimal places \n-->",round(i,2))



RANDOM LIBRARY

import random

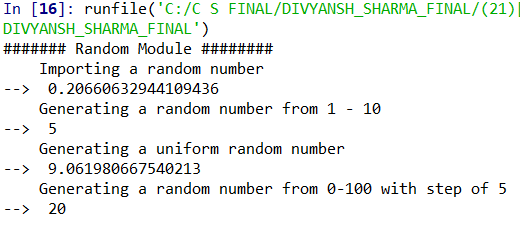
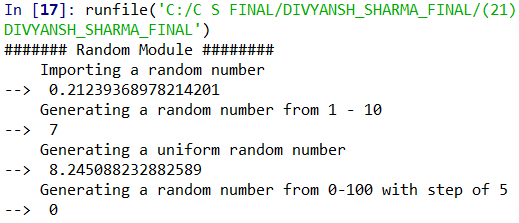
print("####### Random Module ########")

print("\tImporting a random number \n--> ",random.random())

print("\tGenerating a random number from 1 - 10 \n--> ", random.randint(1,10))

print("\tGenerating a uniform random number \n--> ",random.uniform(1,10))

print("\tGenerating a random number from 0-100 with step of 5 \n--> ",random.randrange(0,100,5))

USE OF STRING FUNCTIONS

# program to calculate join, split, replace

print("#BASIC STRING FUNCTIONS#")

print("\n")

s=input("Enter the string --> ")

j=input("Enter the string to join -->")

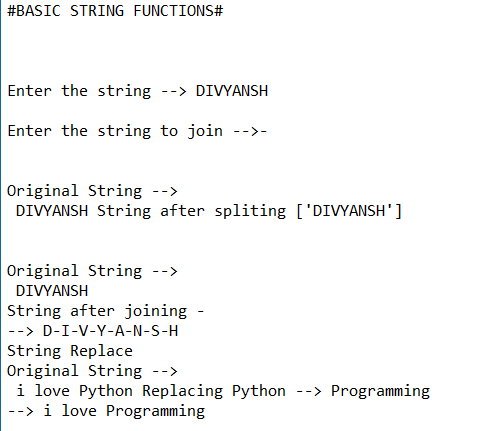
print("\n\nOriginal String -->\n",s,"String after spliting",s.split())

print("\n\nOriginal String -->\n",s,"\nString after joining",j,"\n-->",j.join(s))

s="i love Python"

print("String Replace")

print("Original String -->\n",s,"Replacing Python --> Programming \n-->",s.replace("Python","Programming"))



MODULE FOR TEMPERATURE CONVERSION

import module\_23 as temp

print("\t\t##### TEMPERATURE CONVERSION ######")

print('1--> To give the temperature value in Celcius and get its Fahrenheit value --> ')

print('2--> To give the temperature value in Fahrenheit and get its Celcius value --> ')

i=int(input("Enter the Choice --> "))

if i==1:

c=int(input("Enter the Temp value in Celcius --> "))

print('\tTemperature Value in celcius -->',c)

print('\n\tTemperature Value in Fahrenheit -->',temp.c\_to\_f(c))

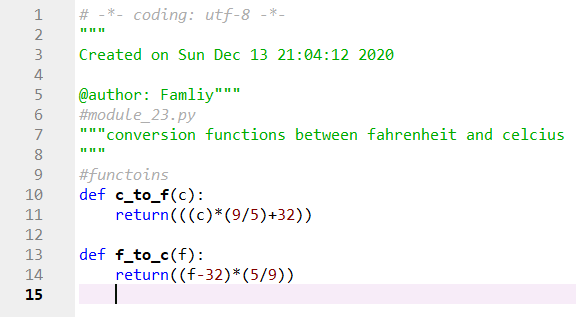
if i==2:

f=int(input("Enter the Temp value in Fahrenheit --> "))

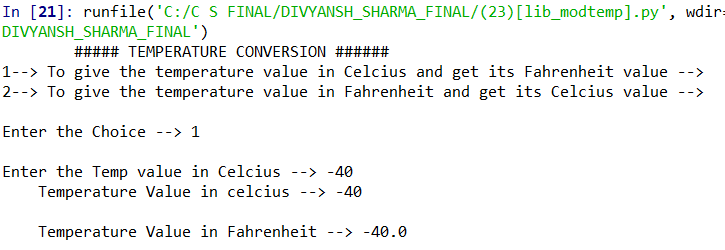
print('\tTemperature Value in Fahrenheit -->',f)

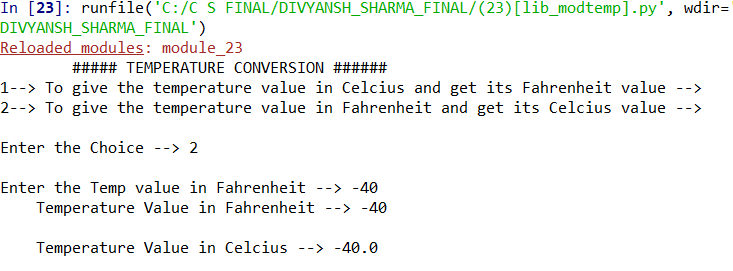
print('\n\tTemperature Value in Celcius -->',temp.f\_to\_c(f))

MODULE



OUTPUT





MODULES FOR AREA AND PERIMETER CALCULATION

import module\_24 as area

print("Enter the choice")

print("1 --> Square")

print("2 --> Circle")

print("3 --> Rectangle")

print("4 --> Triangle")

c=str(input("Please Enter your choice -->"))

if c=="1":

a=float(input("Enter the side of the square -> "))

ps,pa=area.square(a)

print("\n Side of square is ->",a)

print("\n Perimeter of square is->",ps)

print("\n Area of square is->",pa)

elif c=="2":

r=float(input("Enter the radius of the CIRCLE -> "))

pc,ac=area.circle(r)

print("Radius of the cicle is ->",r)

print("Perimeter of circle is->",pc)

print("Area of circle is ->",ac)

elif c=="3":

l=float(input("Enter the length of rectangle -> "))

b=float(input("Enter the breadth of rectangle -> "))

pr,ar=area.rect(l,b)

print("Length is ->",l,"\nBreadth is -> ",b)

print("Perimeter of rectangle is ->",pr)

print("Area of rectangle is ->",ar)

elif c=="4":

b=float(input("Enter the base length of the triangle -> "))

h=float(input("Enter the height of the triangle ->"))

pt,at=area.tri(b,h)

print("Base is ",b,"\nHeight is->",h)

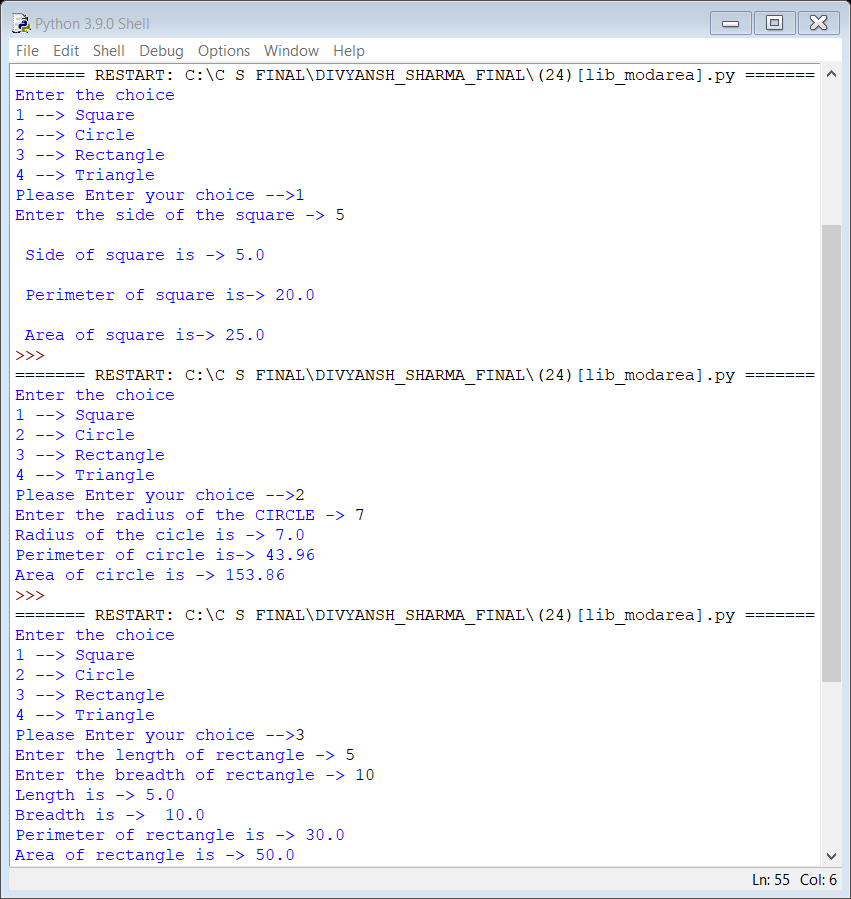
print("perimeter of triangle is->",pt)

print("Area of triangle is -> ",at)

else:

print("INVALID INPUT !!!")

MODULE



# -\*- coding: utf-8 -\*-

"""

Created on Sun Dec 13 21:10:12 2020

@author: Famliy

"""

""" MODULE FOR AREA AND PERIMETER

"""

''' figures.py'''

def square(a):

ps=4\*a

pa=a\*\*2

return(ps,pa)

def circle(r):

pc=2\*(3.14)\*r

ac=(3.14)\*(r\*\*2)

return(pc,ac)

def rect(l,b):

pr=2\*(l+b)

ar=l\*b

return(pr,ar)

def tri(b,h):

pt=b+2\*(((b/2)\*\*2)+(h\*\*2))

at=0.5\*b\*h

return(pt,at)

OUTPUT

File Handling

FUNCTIONS

*Writing & Reading FILES*

*BINARY FILE READING & WRITING*

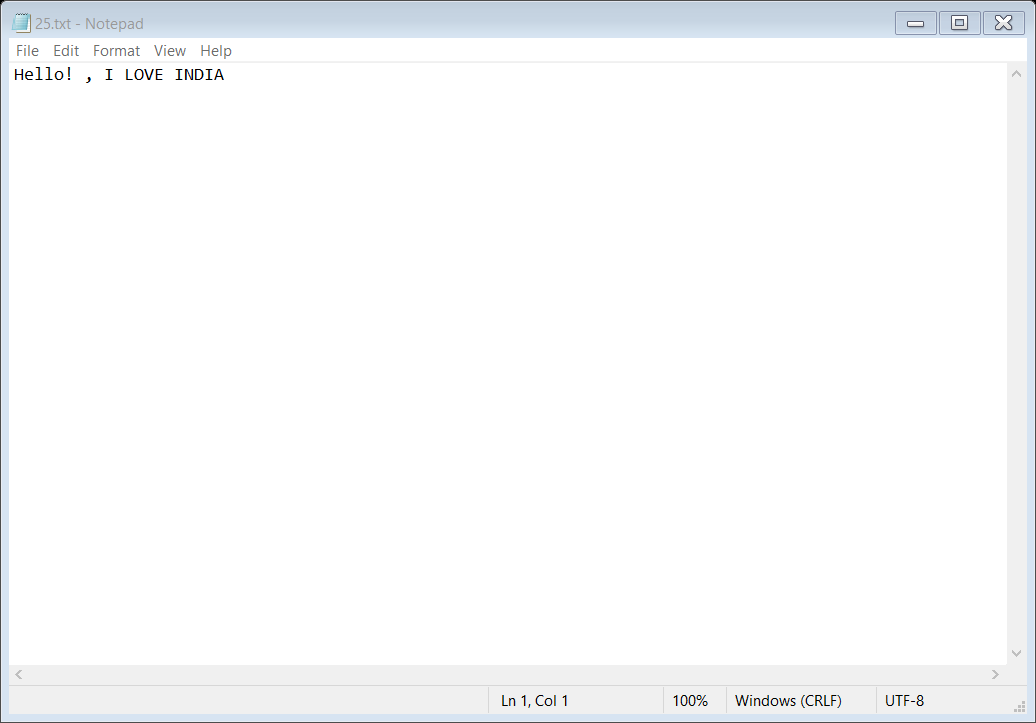
*CSV FILE READING AND WRITING*

WRITING A TEXT FILE IN ‘W’ MODE

print("To write a single line in a file")

with open("./25.txt","w") as fl:

fl.writelines("Hello! , I LOVE INDIA")

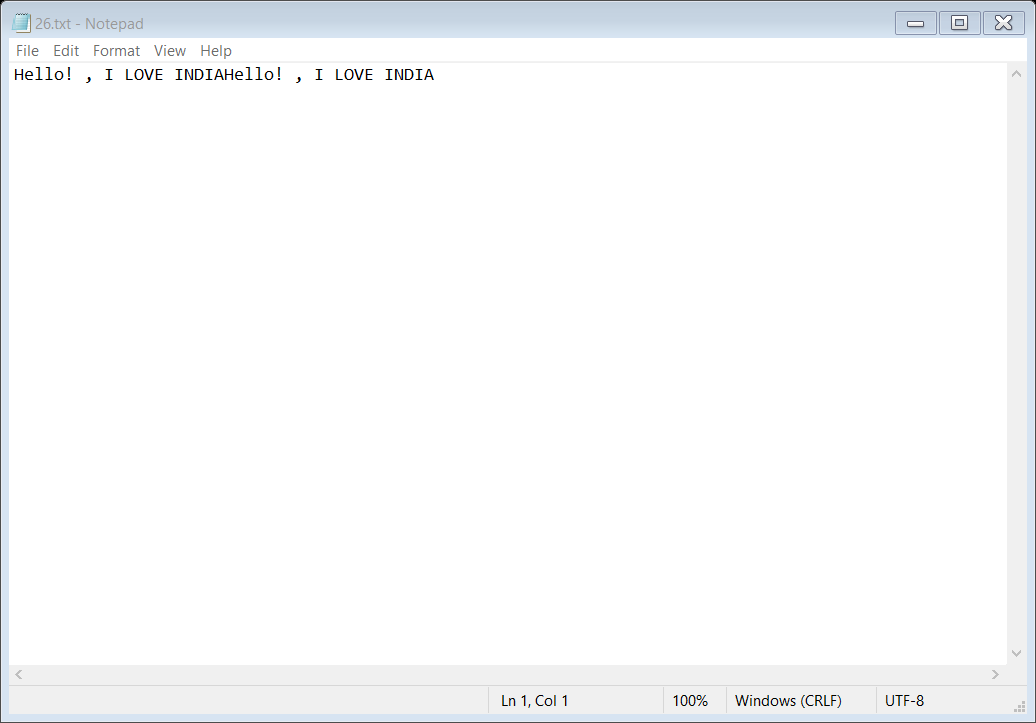


WRITING A TEXT FILE IN ‘A’ MODE

print("To write a single line in a file")

with open("../26.txt","a") as fl:

fl.writelines("Hello! , I LOVE INDIA")



READ USER SPECIFIED CHARACTER FROM TEXT FILE

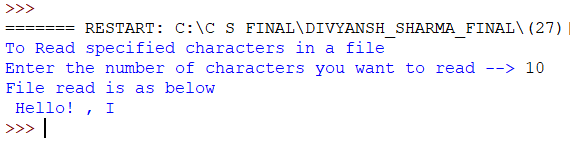
print("To Read specified characters in a file")

i=int(input("Enter the number of characters you want to read --> "))

with open("./26.txt","r") as fl:

r=fl.read(i)

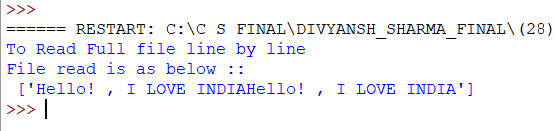
print("File read is as below\n",r)



READING ALL DATA FROM TEXT FILE

print("To Read Full file line by line")

with open("./26.txt","r") as fl:



r=fl.readlines()

print("File read is as below ::\n",r)

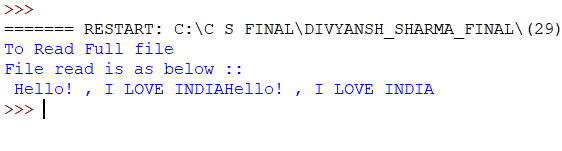
READING FULL TEXT FILE

print("To Read Full file")

with open("E://CS 12/FINAL/26.txt","r") as fl:

r=fl.read()

print("File read is as below ::\n",r)



READING PARAGRAPH LINE STARTING FROM ‘A’

with open("./30.txt","w+") as fl:

fl.writelines("India is one of the youngest superpowers in the world.\n ")

fl.writelines("The National bird of India is the peacock, which has a very colourful and beautiful tail.\n")

fl.writelines(" The national flower of India is Lotus.\n")

fl.writelines(" Lotus comes in many colours, white and pink being prominent.\n")

fl.writelines(" The National animal of India is the Royal Bengal tiger.\n")

fl.writelines("And It is very powerful and majestic to look at.\n")

fl.writelines("The national anthem of India is ‘Jana Gana Mana.’\n")

fl.writelines(" Its approximate playing time is 52 seconds.\n")

fl.writelines(" The national song of India is Vande Mataram, written by Bankimchandra Chatterji and was sung in the 1896 session of the Indian National Congress. \n")

fl.writelines("The national insects of India are the Papilionidae or swallowtails.\n")

fl.writelines(" The national aquatic animal of India is South Asian Dolphin.\n")

fl.writelines(" The national reptile of India is King Cobra, also known as the ‘king of snakes. \n")

fl.writelines("The national fruit of India is Mango, also called the ‘King of Fruits.’\n")

with open("./30.txt","r") as fl:

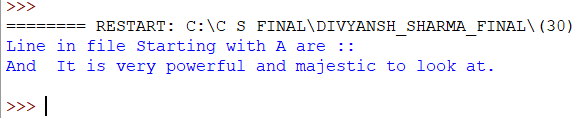
r=fl.readlines()

print("Line in file Starting with A are ::")

for i in r:

if i.startswith("A"):

print(i)



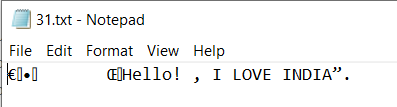
CREATING BINARY FILE AND WRITING DATA IN IT

import pickle

print("To write a BINARY line in a file")

with open("./31.txt","wb") as fl:

pickle.dump("Hello! , I LOVE INDIA",fl)



create binary file and read data from it

import pickle

print("To Read a BINARY line in a file")

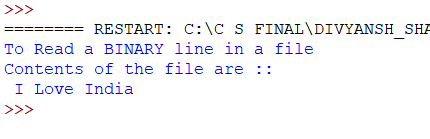
with open("./32.txt","wb") as fl:

pickle.dump("I Love India",fl)

with open("./32.txt","rb") as fl:

r=pickle.load(fl)

print("Contents of the file are :: \n",r)



creating csv file and writing in it using writer

import csv

i=[r for f in range(1,100) for r in range(1,11) ]

fl=open("./33.txt","w")

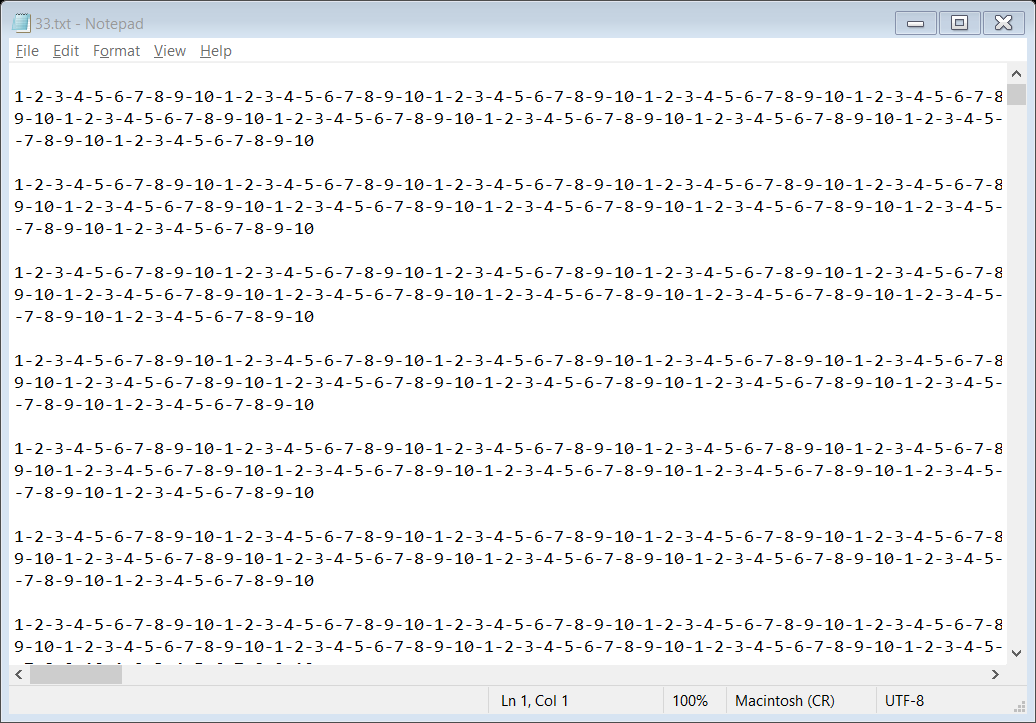
r=csv.writer(fl,delimiter="-",)

for k in i:

r.writerow(i)

fl.close()

print("Written SUccessfully")



creating csv file and writing it using writerows function

import csv

print("######### CSV WRITER USING WRITEROWS ########")

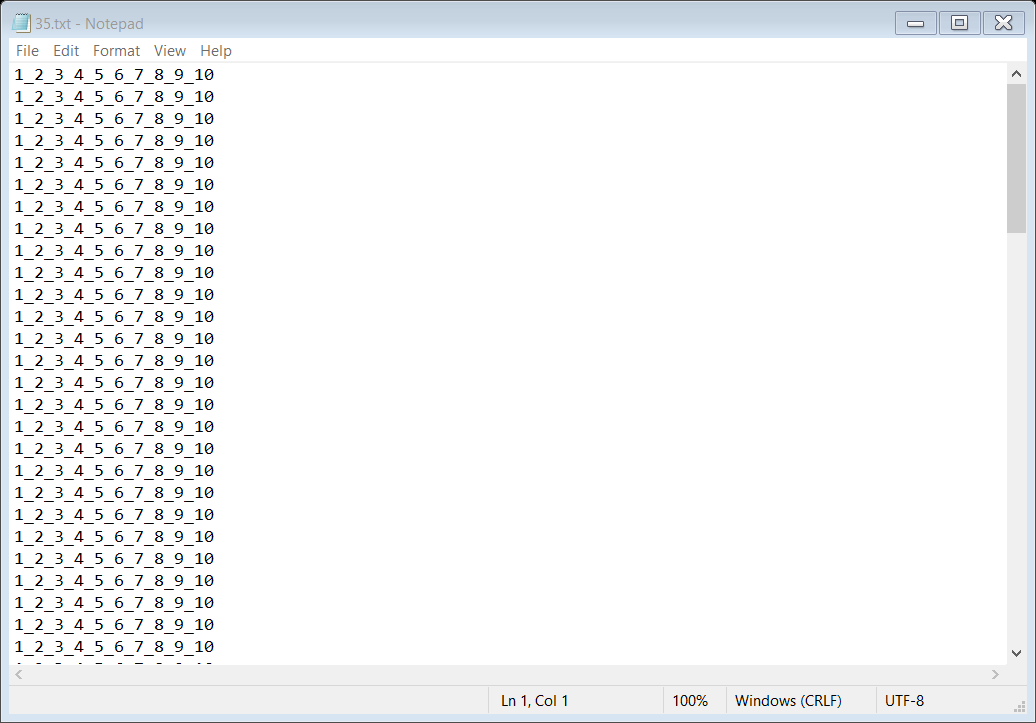
g=[l for l in range(1,11)]

i=[ g for i in range(1,101) ]

with open("./35.txt","w") as fl:

r=csv.writer(fl,delimiter="\_",lineterminator="\n")

r.writerows(i)



creating csv file to append data in it

import csv

print("######### CSV WRITER IN APPEND MODE ########")

g=[l for l in range(1,11)]

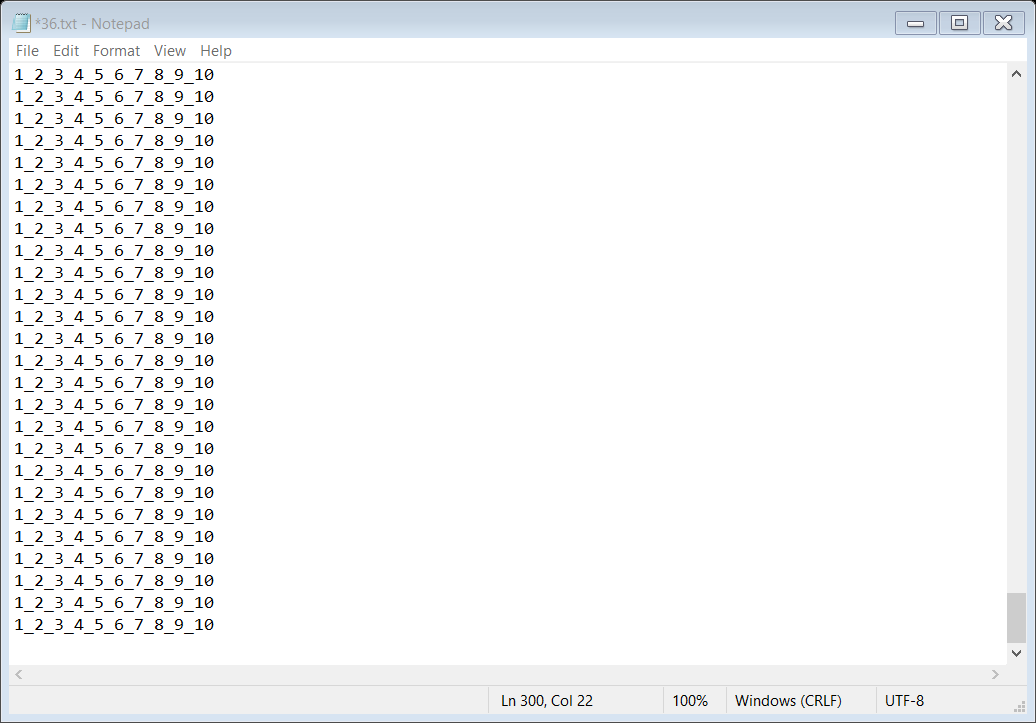
i=[ g for i in range(1,101) ]

with open("./36.txt","a") as fl:

r=csv.writer(fl,delimiter="\_",lineterminator="\n")

r.writerows(i)

print("DATA APPENDED SUCCESSFULLY")



reading contents of csv file

import csv

print("\t######### CSV file reading ########")

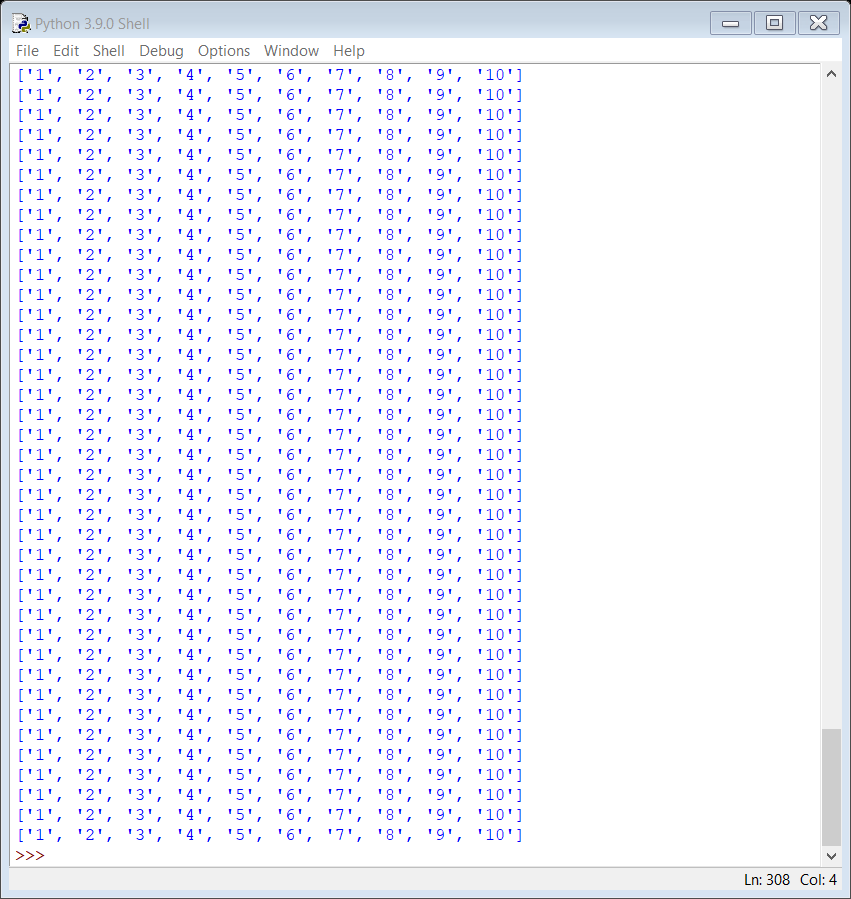
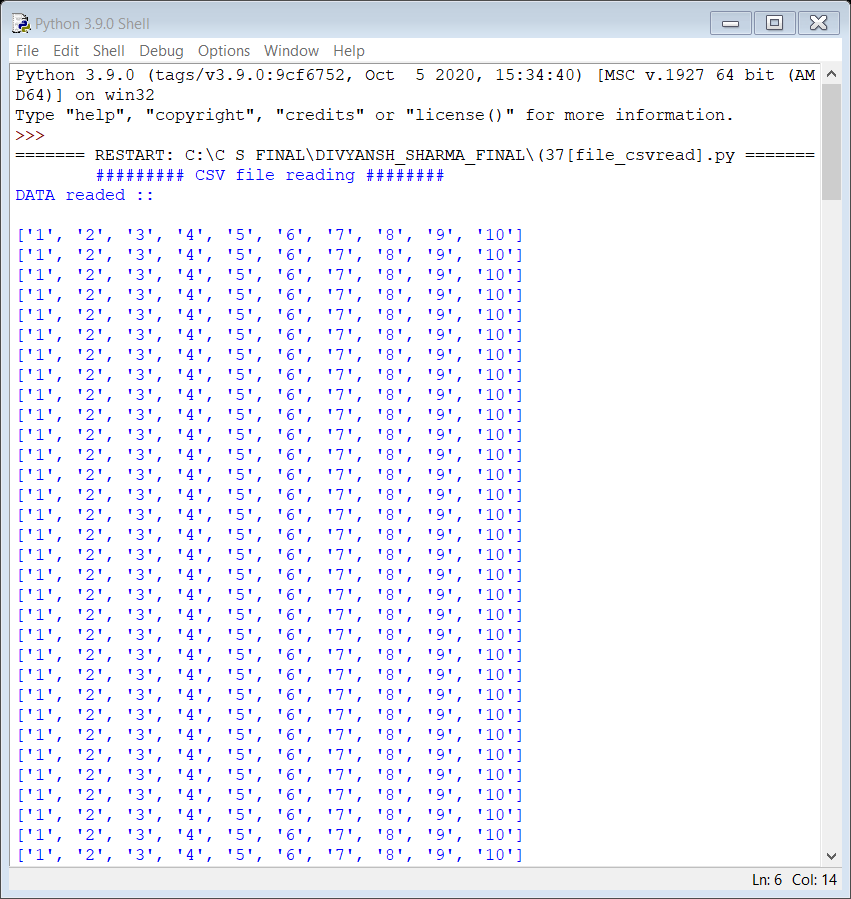
with open("./36.txt","r") as fl:

r=csv.reader(fl,delimiter="\_",lineterminator="\n")

print("DATA readed ::\n ")

for i in r:

print(i)

Linear List

FUNCTIONS

*LINEAR SEARCH*

*BINARY SEARCH*

*INSETING BY BISECT MODULE*

*LIST COMPREHENSION*

*STACKS &*

*QUEUES*

LINEAR SEARCH USING TRADITIONAL SEARCH ALGORITHM

lst=[i for i in range(1,101)]

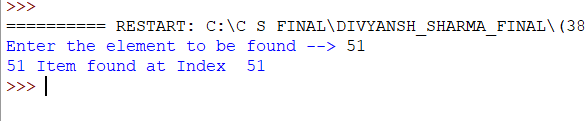
req=int(input("Enter the element to be found --> "))

for k in range(0,len(lst)):

if req==lst[k]:

break

print(req,"Item found at Index ",k+1)



INSERTING USING BISECT MODULE

import bisect

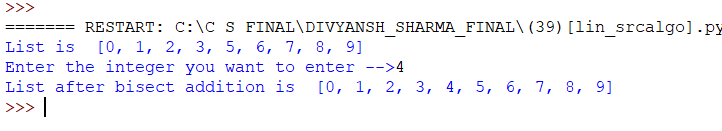
i=[0,1,2,3,5,6,7,8,9]

print("List is ",i)

k=int(input("Enter the integer you want to enter -->"))

bisect.insort(i,4)

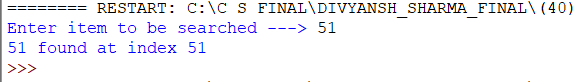
print("List after bisect addition is ",i)



BINARY SEARCH IN LINEAR LIST

### binary search

OUTPUT



def bins(lst,key):

low = 0

high=len(lst)-1

while low<=high:

mid=int((high+low)/2)

if key==lst[mid]:

return (mid)

elif key<lst[mid]:

high=mid-1

else:

low=mid+1

else:

return -999

lst=[i for i in range(0,101)]

item=int(input("Enter item to be searched ---> "))

res=bins(lst,item)

if res>0:

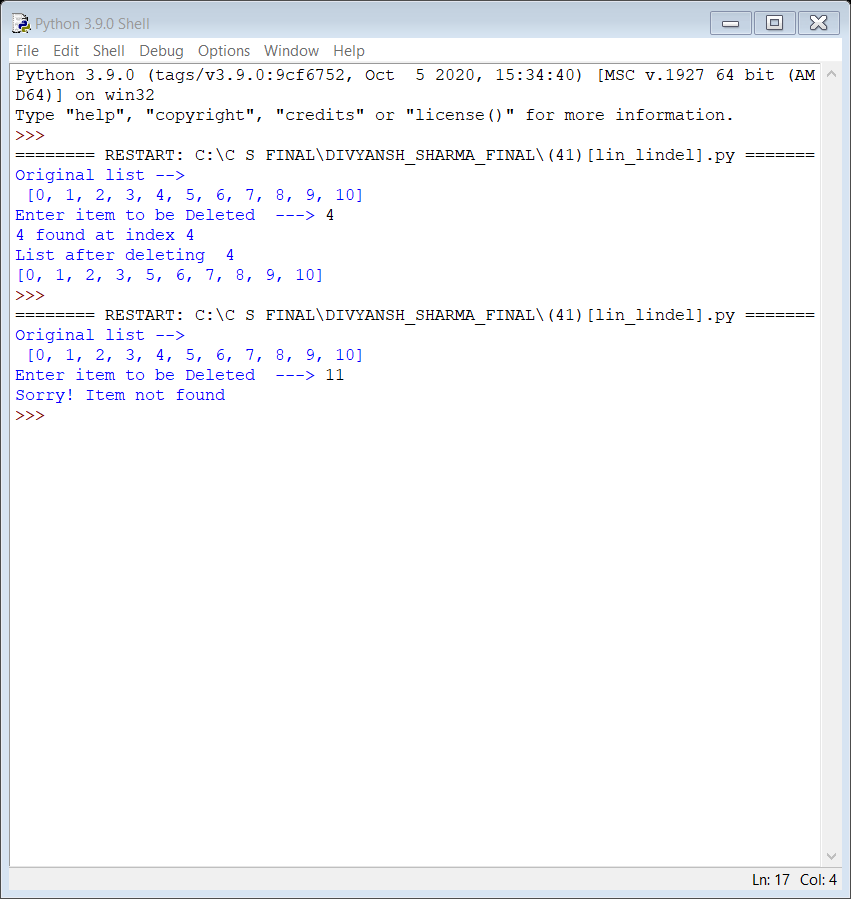
print(item,"found at index",res)

else:

print("Sorry! Item not found")

deleting an item using linear list

##linear list delete item



def bins(lst,key):

low = 0

high=len(lst)-1

while low<=high:

mid=int((high+low)/2)

if key==lst[mid]:

return (mid)

elif key<lst[mid]:

high=mid-1

else:

low=mid+1

else:

return -999

lst=[i for i in range(0,101)]

print("Original list -->\n",lst)

item=int(input("Enter item to be Deleted ---> "))

res=bins(lst,item)

if res>0:

print(item,"found at index",res)

lst.pop(res)

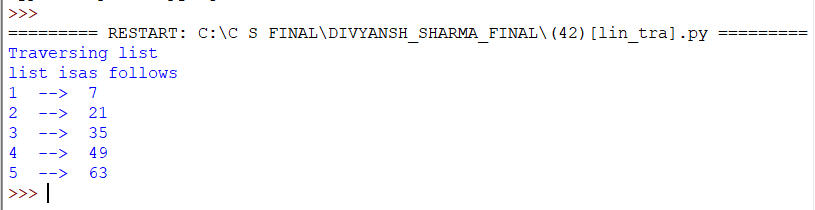
print("List after deleting ",item)

print(lst)

else:

print("Sorry! Item not found")

traversing all items in a linear list



print("Traversing list")

l=[i\*7 for i in range(1,11,2)]

print("list is as follows")

for i in range(0,len(l)):

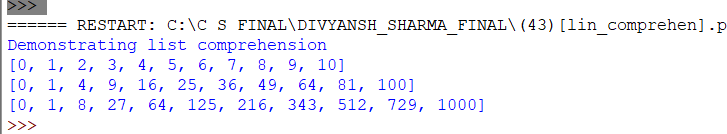
print(i+1,' --> ',l[i])

list comprehension

print("Demonstrating list comprehension")

l1=[i for i in range(0,11) ]

output



l2=[i\*\*2 for i in l1]

l3=[i\*\*3 for i in l1]

print(l1)

print(l2)

print(l3)

stack operations

print(" # # S T A C K # # ")

def e(stk):

if stk ==[]:

return(True)

else:

return(False)

def push(stk,arr):

stk.append(arr)

def pop(stk):

if e(stk) == True:

return("\n\tUnderflow")

else:

stk.pop(-1)

def peek(stk):

if e(stk)==True:

return("\n\t--S T A C K - I S - E M P T Y --")

else:

return(stk[-1])

def disp(stk):

if e(stk)== True:

return("\n\t--S T A C K - I S - E M P T Y --")

elif len(stk)==1:

return(stk[0])

else:

k=[]

k=stk

k.reverse()

for i in k:

print(i)

done=False

stk=[]

while done== False:

print("\n\nSatck implementations")

print("1. --> P U S H ")

print("2. --> P O P ")

print("3. --> P E E K ")

print("4. --> D I S P L A Y ")

print("5. --> E X I T ")

ch=int(input("Enter your Choice --> "))

if ch==1:

arr=int(input("\nEnter the element --> "))

push(stk,arr)

elif ch==2:

if e(stk)==True:

print(pop(stk))

else:

pop(stk)

elif ch==3:

if e(stk)==True:

print('\n',peek(stk))

else:

print("Peek of Stack at position --> ",len(stk),"\n element at peek is",peek(stk))

elif ch== 4:

print('\n\t')

if e(stk)== True:

print(disp(stk))

elif len(stk) == 1:

print(disp(stk))

else:

disp(stk)

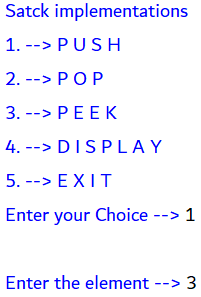
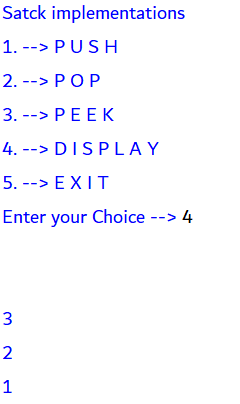
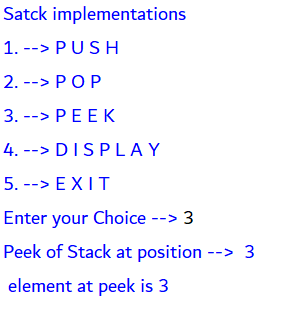
elif ch==5 :

done = True

print("\n\t-- T H A N K - Y O U --")

else:

print("## I N V A L I D - I N P U T ##")

queue operations

print(" # # Q U E U E # # ")

def empty(que):

if que ==[]:

return(True)

else:

return(False)

def EN(que,arr):

que.append(arr)

def DQ(que):

if empty(que) == True:

return("\n\tUnderflow")

else:

que.pop(0)

def peek(que):

if empty(que)==True:

return("\n\t-- Q U E U E - I S - E M P T Y --")

else:

return(que[-1])

def disp(que):

if empty(que)== True:

return("\n\t-- Q U E U E - I S - E M P T Y --")

elif len(que)==0:

return (que[0])

else:

for i in que:

print(i)

done=False

que=[]

while done== False:

print("\n\nQueue implementations")

print("1. --> P U S H ")

print("2. --> P O P ")

print("3. --> P E E K ")

print("4. --> D I S P L A Y ")

print("5. --> E X I T ")

ch=int(input("Enter your Choice --> "))

if ch==1:

arr=int(input("\nEnter the element --> "))

EN(que,arr)

elif ch==2:

if empty(que)==True:

print(DQ(que))

print("Queue is empty")

else:

DQ(que)

elif ch==3:

if empty(que)==True:

print('\n',peek(que))

else:

print("Peek of Queue at position --> ",len(que),"\n element at peek is",peek(que))

elif ch== 4:

print('\n\t')

if empty(que)==True:

print(disp(que))

elif len(que)==0:

print(disp(que))

else:

disp(que)

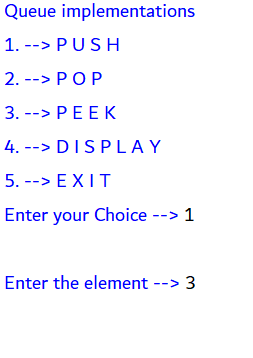
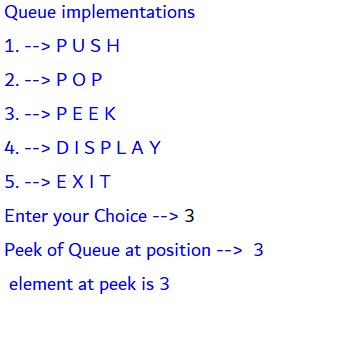
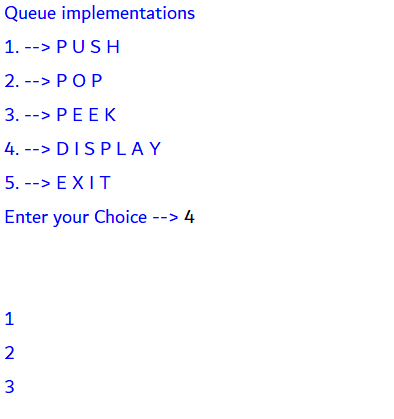
elif ch==5 :

done = True

print("\n\t-- T H A N K - Y O U --")

else:

print("## I N V A L I D - I N P U T ##")

Python-MySQL

CONNECTIVITY

FUNCTIONS

*TABLE CREATION*

*INSERT RECORDS*

*DISPLAY RECORDS*

*FETCH RECORD USING - PARAMETERISED QUERY’S*

*UPDATE RECORD*

creating database and table

import mysql.connector

q=mysql.connector.connect(host="localhost",user="root",passwd="")

cur=q.cursor()

print("\*\* Database and Table Creation \*\* ")

dat=input("Enter the name of the Database -->")

tab=input("Enter the name of the Table -->")

qry=("CREATE DATABASE`{}`").format(dat,)

cur.execute(qry)

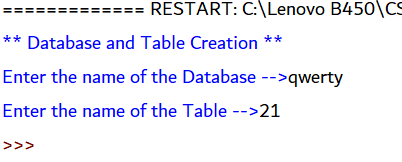
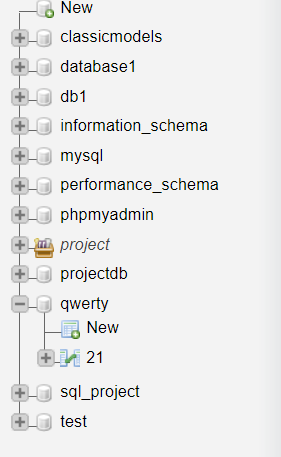
q.commit()

qry=("CREATE TABLE `{}`.`{}`(no int(5),sname varchar(30),scho int(5))").format(dat,tab)

cur.execute(qry)

q.commit()

q.close()

inserting record in predefined table

import mysql.connector

q=mysql.connector.connect(host="localhost",user="root",passwd="")

cur=q.cursor()

print("\t\*\* RECORD INSERTION IN STUDENT TABLE \*\* ")

rec\_no=int(input("Please Enter the number of records you want to enter -->"))

cur.execute("CREATE TABLE IF NOT EXISTS `test`.`student`(no int(5),sname varchar(30),scho int(5))")

q.commit()

for i in range(1,rec\_no+1):

no=int(input (f"Please enter the no of student {i}-->"))

nm=(input (f"Please enter the Student name of student {i}-->"))

sc=int(input (f"Please enter the scholar number of the student {i}-->"))

qry=("INSERT INTO `test`.`student`(`no`,`sname`,`scho`) VALUES ({},'{}',{})").format(no,nm,sc)

cur.execute(qry)

print("Final data is added as follows -->>")

qry="SELECT \* FROM `test`.`student`"

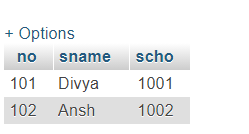
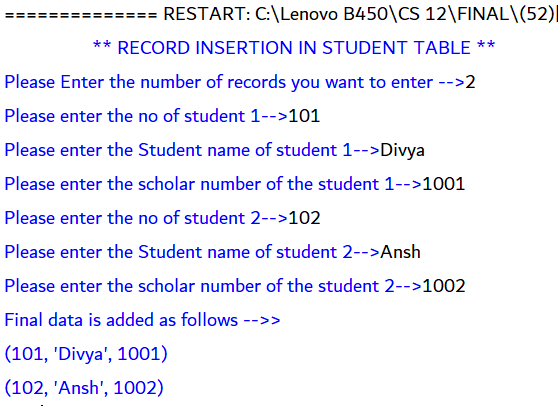
cur.execute(qry)

r=cur.fetchall()

for i in r:

print(i)

q.close()



displaying all records of the table

print("Program to display records of table")

db=input("Please enter the Name of the Database -->")

tb=input("Please enter the Name of the Table -->")

import mysql.connector

q=mysql.connector.connect(host="localhost",user="root",passwd="",database="{}".format(db,))

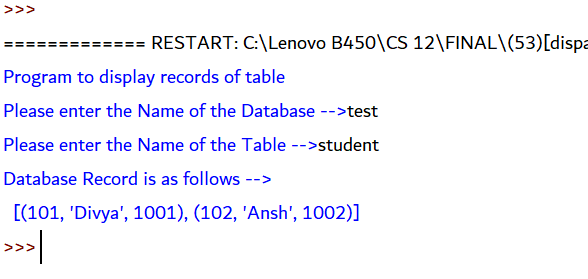
cur=q.cursor()

qry="SELECT \* FROM `{}`".format(tb,)

cur.execute(qry)

r=cur.fetchall()

print("Database Record is as follows -->\n ",r)



displaying records one-by-one

print("Program to display records of table")

db=input("Please enter the Name of the Database -->")

tb=input("Please enter the Name of the Table -->")

import mysql.connector

q=mysql.connector.connect(host="localhost",user="root",passwd="",database="{}".format(db,))

cur=q.cursor()

qry="SELECT \* FROM `{}`".format(tb,)

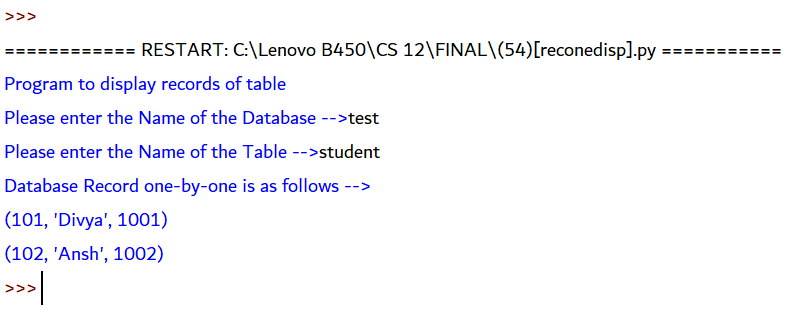
cur.execute(qry)

r=cur.fetchall()

print("Database Record one-by-one is as follows --> ")

for i in r :

print(i)



displaying user specified number of records

print("Program to display records of table")

db=input("Please enter the Name of the Database -->")

tb=input("Please enter the Name of the Table -->")

import mysql.connector

q=mysql.connector.connect(host="localhost",user="root",passwd="",database="{}".format(db,))

cur=q.cursor()

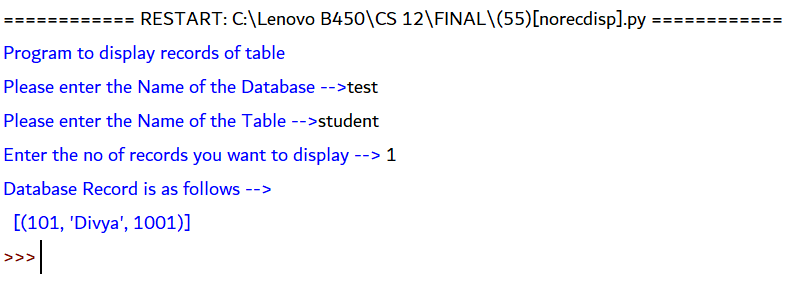
qry="SELECT \* FROM `{}`".format(tb,)

cur.execute(qry)

no=int(input("Enter the no of records you want to display --> "))

r=cur.fetchmany(no)

print("Database Record is as follows -->\n ",r)



fetching records using parametrised query (OLD STYLE)

print("Program to display records of table")

db=input("Please enter the Name of the Database -->")

tb=input("Please enter the Name of the Table -->")

import mysql.connector

q=mysql.connector.connect(host="localhost",user="root",passwd="",database="%s"%(db,))

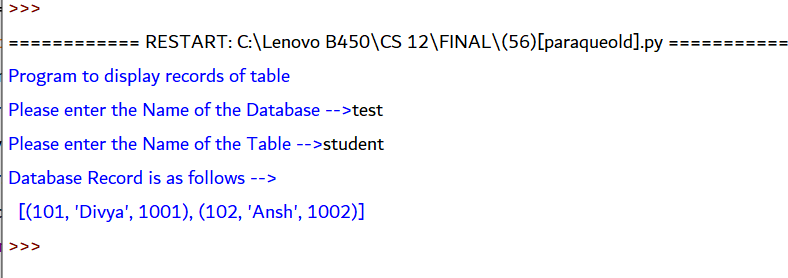
cur=q.cursor()

qry="SELECT \* FROM `%s`"%(tb,)

cur.execute(qry)

r=cur.fetchall()

print("Database Record is as follows -->\n ",r)



fetching records using parametrised query (new STYLE)

print("Program to display records of table")

db=input("Please enter the Name of the Database -->")

tb=input("Please enter the Name of the Table -->")

import mysql.connector

q=mysql.connector.connect(host="localhost",user="root",passwd="",database="{}".format(db,))

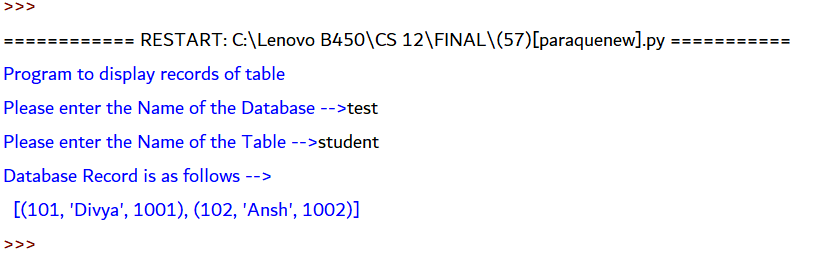
cur=q.cursor()

qry="SELECT \* FROM `{}`".format(tb,)

cur.execute(qry)

r=cur.fetchall()

print("Database Record is as follows -->\n ",r)



Updating record of user choice

import mysql.connector

q=mysql.connector.connect(host="localhost",user="root",passwd="",database="test")

cur=q.cursor()

qry="SELECT \* FROM `student`"

cur.execute(qry)

r=cur.fetchall()

print("Database is as follows")

for i in r:

print(i)

us=int(input("Please enter the Scholar number of the student whos data you want to update \n-->"))

print("Enter your field for your updation -->")

print("1--> Roll no")

print("2--> Name")

print("3--> Scholar Number")

i=int(input("--> "))

if i==1:

ur=int(input("Please enter the updated roll no.-->"))

qry="UPDATE `student` SET `sno`={} where scho={}'".format(ur,us)

cur.execute(qry)

q.commit()

qry="select \* from `student` where `scho`={}".format(us,)

cur.execute(qry)

r=cur.fetchall()

print("Updated record\n",r)

elif i==2:

un=(input("Please enter the updated Name -->"))

qry="UPDATE `student` SET `sname`='{}' where scho={}".format(un,us)

cur.execute(qry)

q.commit()

qry="select \* from `student` where `scho`={}".format(us,)

cur.execute(qry)

r=cur.fetchall()

print("Updated record\n",r)

elif i==3:

usc=int(input("Please Enter the updated scholar number -->"))

qry="UPDATE `student` SET scho={} WHERE scho={}".format(usc,us)

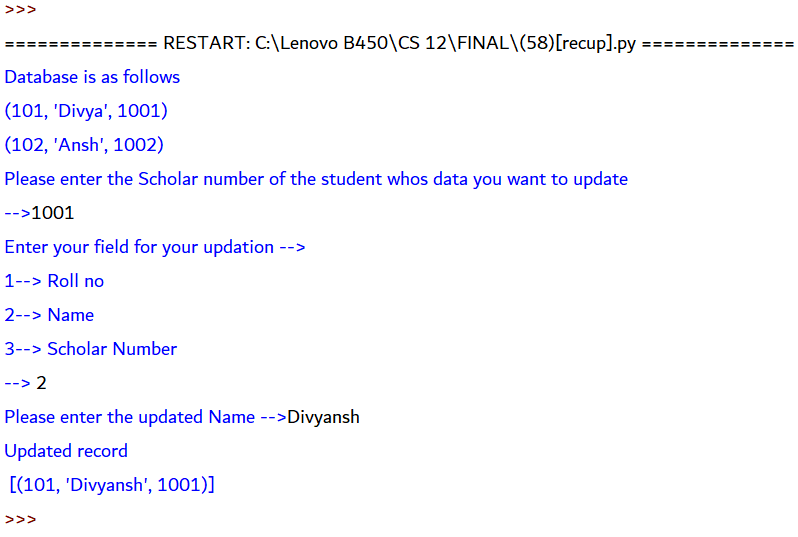
cur.execute(qry)

q.commit()

qry="select \* from `student` where `scho`={}".format(us,)

cur.execute(qry)

r=cur.fetchall()



print("Updated record\n",r)

else:

print("Invalid Input")

deleting record of user choice

import mysql.connector

q=mysql.connector.connect(host="localhost",user="root",passwd="",database="test")

cur=q.cursor()

sc=int(input("Enter the scho of the student whose record yowant to delete \n-->"))

cur.execute("DELETE FROM `test`.`student` WHERE `student`.`scho`={}".format(sc,))

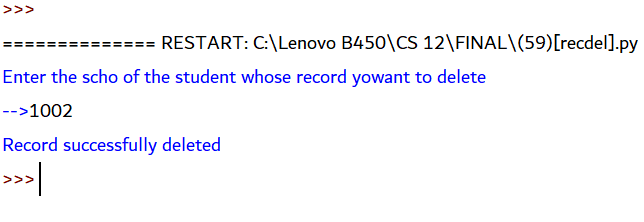
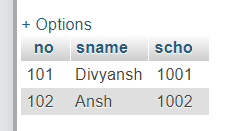
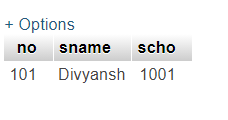
q.commit()

print("Record successfully deleted")

before

after

query

MySQL

QUERIES

*ORDER BY*

*GROUP BY*

*AGREGATE FUNCTIONS*

*COUNT*

Query for data

CREATE DATABASE sql\_project

CREATE TABLE student(rollno int(5),name varchar(30),marks float(2),grade CHARACTER(1), section CHARACTER(1))

INSERT INTO `student`(`rollno`, `name`, `marks`, `grade`, `section`) VALUES

(101,'Ruhani',76.80,'A','A'),

(102,'George',71.20,'B','A'),

(103,'Simran',81.20,'A','C'),

(104,'Ali',61.20,'B','C'),

(105,'Kushal',51.60,'C','C'),

(106,'Arsiya',91.60,'A+','B'),

(107,'Raunaq',32.50,'F','B'),

(108,'Meera',97.20,'A+','B'),

(109,'Amaal',57.20,'C','B'),

(111,'Simran',66.00,'B','A'),

(112,'Adam',74.20,'B','C'),

(113,'Gurnoor',93.50,'A+','B'),

(115,'Rabiya',72.50,'B','B'),

(117,'Rahil',32.00,'F','C'),

(118,'Neha',59.50,'C','A')

CREATE TABLE empl (empno int(5) PRIMARY KEY,ename varchar(50),job varchar(50),mgr int(4) DEFAULT NULL ,hiredate DATE ,sal float(2),comm float(2) DEFAULT NULL , Deptno int(2) )

INSERT INTO `empl`(`empno`, `ename`, `job`, `mgr`, `hiredate`, `sal`, `comm`, `Deptno`) VALUES (8396,'SMITH','CLERK',8902,19901218,800,NULL,20),

(8499,'ANYA','SALESMAN',8968,19910220,1600,300,30),

(8521,'SETH','SALESMAN',8698,19910222,1250,500,30),

(8566,'MAHADEVAN','MANAGER',8839,19910402,2985,NULL,20),

(8654,'MOMIN','SALESMAN',8698,19910928,1250,1400,30),

(8698,'BINA','MANAGER',8839,19910501,2850,NULL,30),

(8839,'AMIR','PRESIDENT',NULL,19911118,5000,NULL,10),

(8844,'KULDEEP','SALESMAN',8698,19910908,1500,0,30),

(8882,'SHIVANSH','MANAGER',8839,19910609,2450,NULL,10),

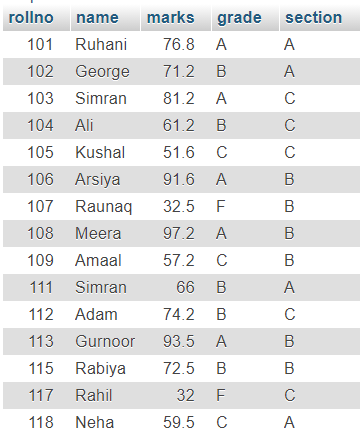
(8886,'ANOOP','CLERK',8888,19930112,1100,NULL,20),

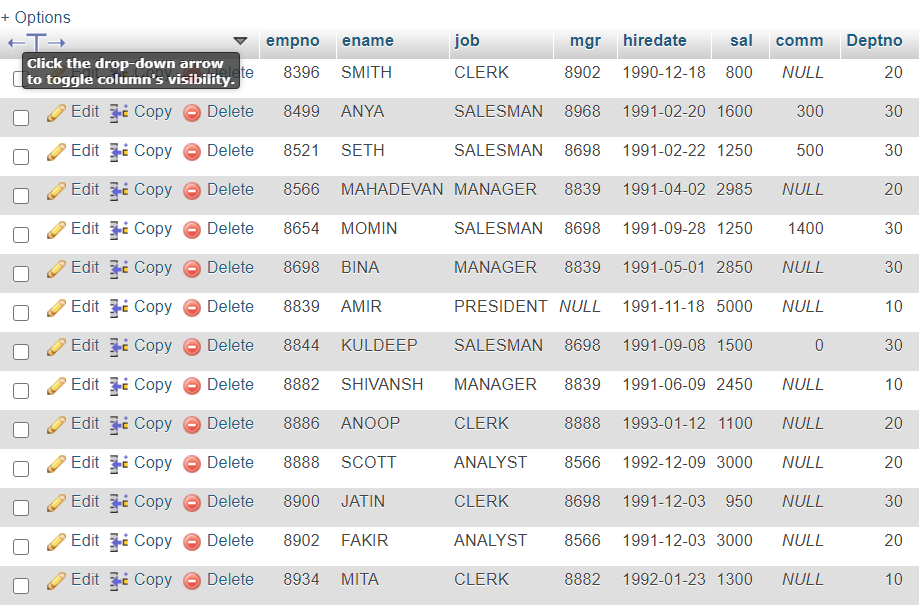
(8888,'SCOTT','ANALYST',8566,19921209,3000,NULL,20),

(8900,'JATIN','CLERK',8698,19911203,950,NULL,30),

(8902,'FAKIR','ANALYST',8566,19911203,3000,NULL,20),

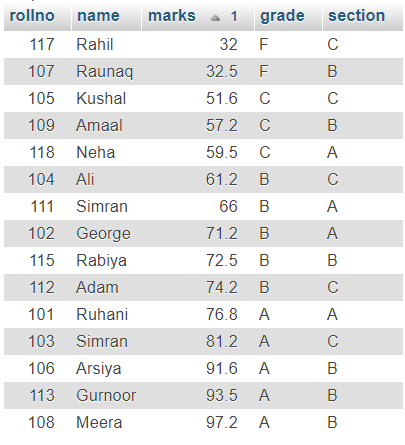
(8934,'MITA','CLERK',8882,19920123,1300,NULL,10)



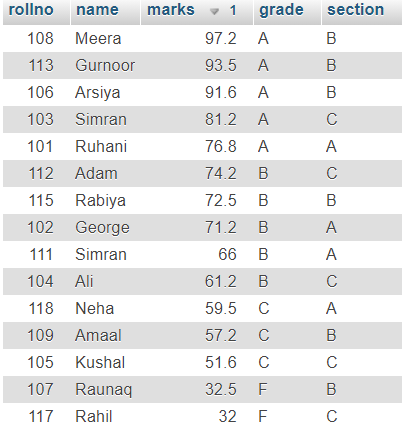
******

sql – queries

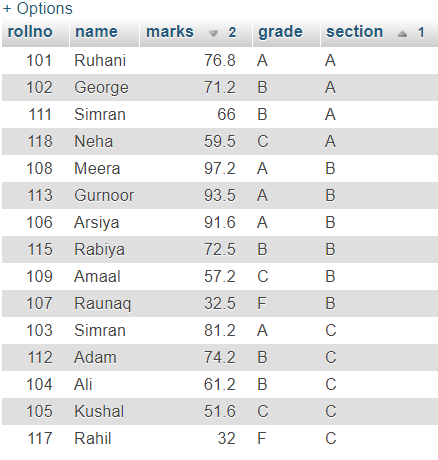
1 -> SELECT \* FROM `student`ORDER BY `marks` ASC



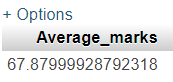
2 -> SELECT \* FROM `student` ORDER BY `marks` DESC



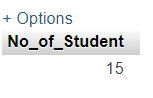
SELECT \* FROM `student` ORDER BY `section`ASC , `marks` DESC



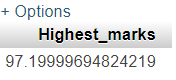
SELECT AVG(`marks`) Average\_marks from `student`



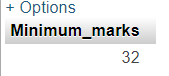
SELECT COUNT(`name`) No\_of\_Student FROM `student`



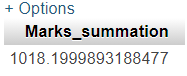
SELECT MAX(`marks`) Highest\_marks FROM `student`



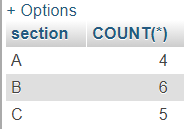
SELECT MIN(`marks`) Minimum\_marks FROM `student`



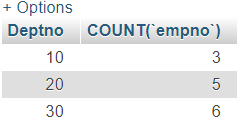
SELECT SUM(`marks`) Marks\_summation FROM `student`



SELECT `section` , COUNT(\*) FROM `student` GROUP BY `section`



SELECT `Deptno`,COUNT(`empno`) FROM empl GROUP BY `Deptno`



SELECT `job`, COUNT(\*) count FROM `empl` GROUP BY `job` HAVING COUNT(\*)>3

