Pratical1A==========================================================================

PI = 3.14

r = float(input("Enter the radius of the circle : "))

area = PI \* r \* r

print("Area of the circle is : %.2f" %area)

l = int(input("Enter the length of the rectangle : "))

b = int(input("Enter the breadth of the rectangle : "))

area = l \* b

print("Area of the rectangle is : %d" %area)

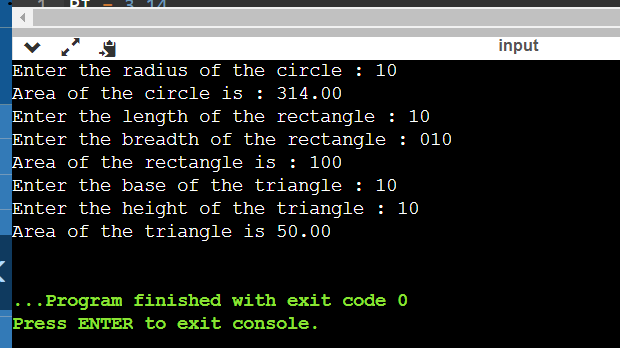
b = float(input("Enter the base of the triangle : "))

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

area = 0.5 \* b \* h

print("Area of the triangle is %.2f" %area)

O/P:



Pratical 1B:

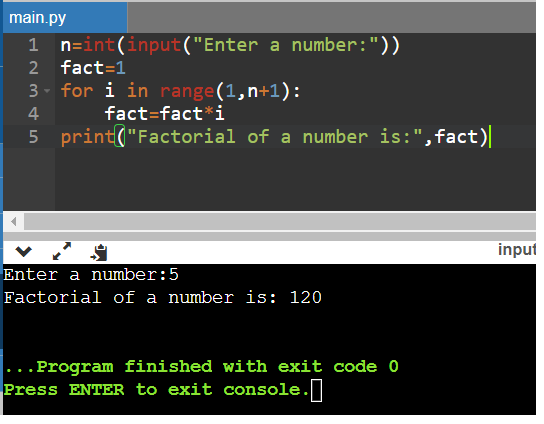
n=int(input("Enter a number:"))

fact=1

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

fact=fact\*i

print("Factorial of a number is:",fact)



Pratical 2A&B======================================================================

l1 = []

num1 = int(input('Enter size of list 1: '))

for n in range(num1):

numbers1 = int(input('Enter any number:'))

l1.append(numbers1)

l2 = []

num2 = int(input('Enter size of list 2:'))

for n in range(num2):

numbers2 = int(input('Enter any number:'))

l2.append(numbers2)

#A

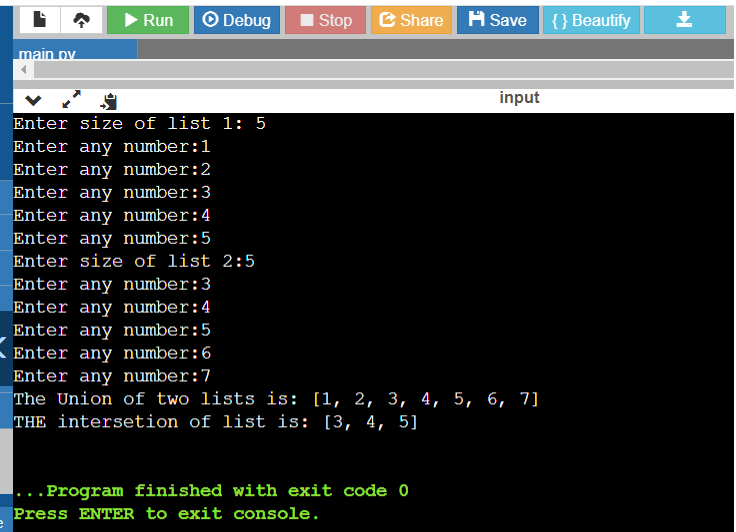
union = list(set().union(l1,l2))

print('The Union of two lists is:',union)

#B

intersection = list(set(l1) & set(l2))

print("THE intersetion of list is:",intersection)

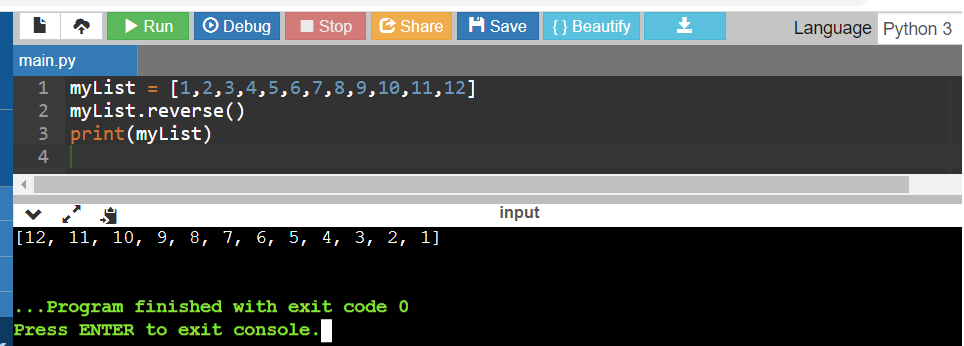


Pratical 2C: :---------------------------------------------------------------------------------------------------------------------

myList = [1,2,3,4,5,6,7,8,9,10,11,12]

myList.reverse()

print(myList)



Pratical 3a========================================================================

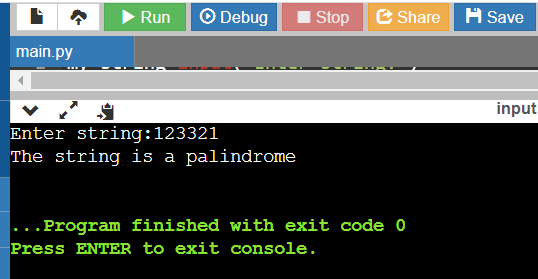
my\_string=input("Enter string:")

if(my\_string==my\_string[::-1]):

print("The string is a palindrome")

else:

print("The string isn't a palindrome")



Pratical 3b: :---------------------------------------------------------------------------------------------------------------------

string="bajajinstitutional"

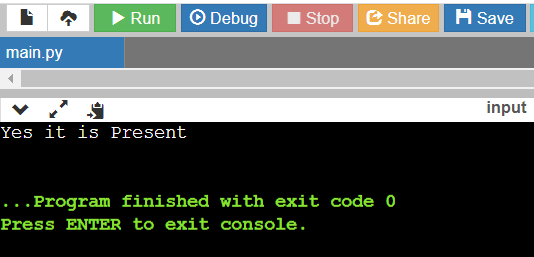
substring="bajaj"

if substring in string:

print("Yes it is Present")

else:

print("No it is not Present")



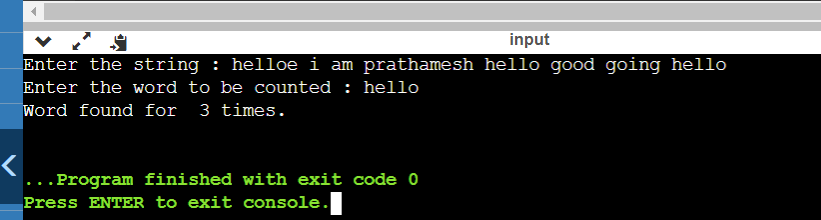
Pratical 3c:-----------------------------------------------------------------------------------------------------------------------

string = input("Enter the string : ")

word = input("Enter the word to be counted : ")

n = string.count(word)

print('Word found for ', n, 'times.')



Pratical 3d:-----------------------------------------------------------------------------------------------------------------------

list = []

n1= int(input("Enter no. of element in list : "))

for n in range(0, n1):

s = input('enter element : ' )

list.append(s)

new\_list = []

count = 0

remove = input("Enter word to remove : ")

n = int(input("Enter occurence to remove : "))

for i in list:

if(i == remove):

count = count+1

if(count == n ):

pass

else:

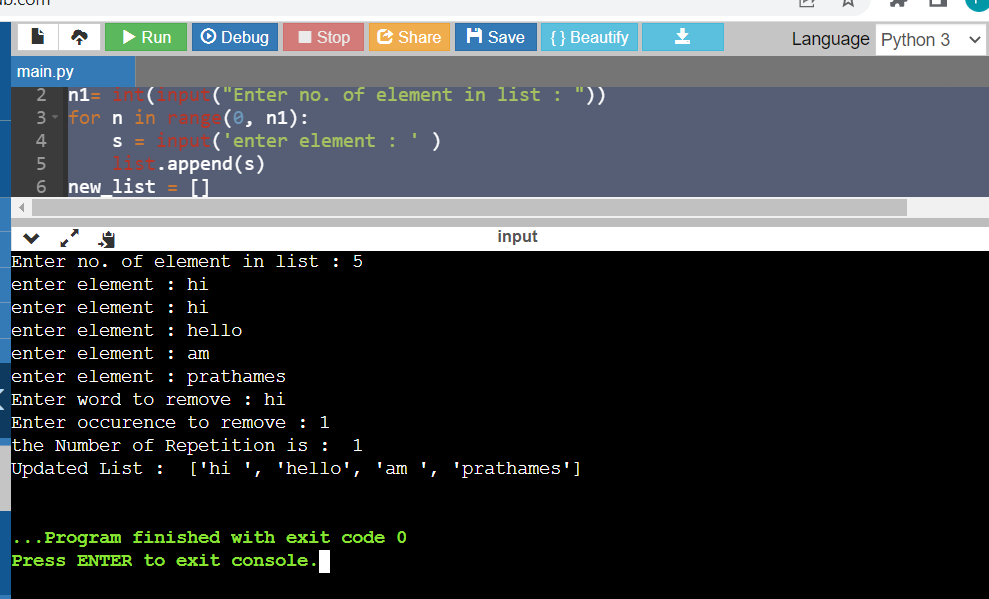
new\_list.append(i)

else:

new\_list.append(i)

print("the Number of Repetition is : " , count )

print("Updated List : " , new\_list )



Pratical 4 A=========================================================================

def add1(a,b):

sum = a+b

return sum

a = int(input("Enter a Number = "))

b = int(input("Enter a another Number = "))

result = add1(a,b)

print(result)

def add2():

num1 = int(input("Enter a Number = "))

num2 = int(input("Enter a another Number = "))

print(num1+num2)

print("Without Arguments and return type \n")

add2()

def add3():

num1 = int(input("Enter a Number = "))

num2 = int(input("Enter a another Number = "))

result = a+b

return (result)

print("Without Arguments ")

print(add3())

Pratical 4B:-----------------------------------------------------------------------------------------------------------------------

def recur\_factorial(n):

if n == 1:

return n

else:

return n\*recur\_factorial(n-1)

num = int(input("Enter a Number :"))

if num < 0:

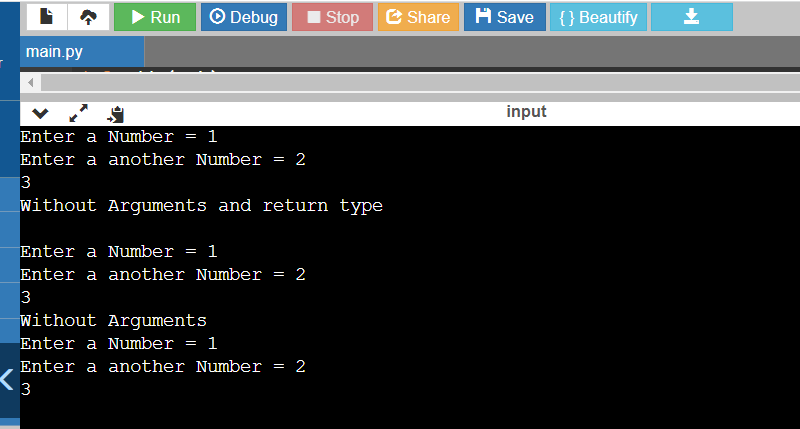
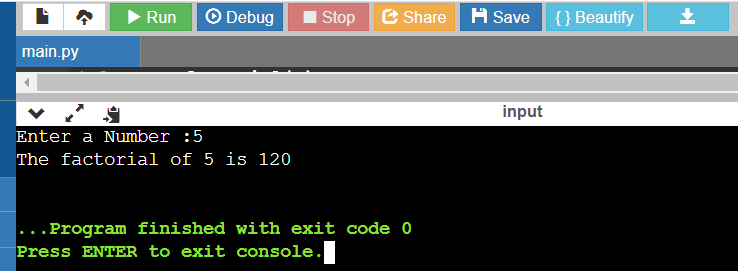
print("Sorry, factorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

print("The factorial of", num, "is", recur\_factorial(num))

Pratical5A =========================================================================

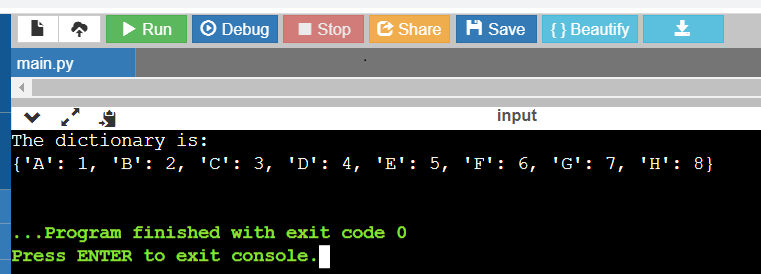
keys=['A','B','C','D','E','F','G','H']

values=[1,2,3,4,5,6,7,8]

d=dict(zip(keys,values))

print("The dictionary is:")

print(d)



Pratical 5B:----------------------------------------------------------------------------------------------------------------------

string = input("Enter the String : ")

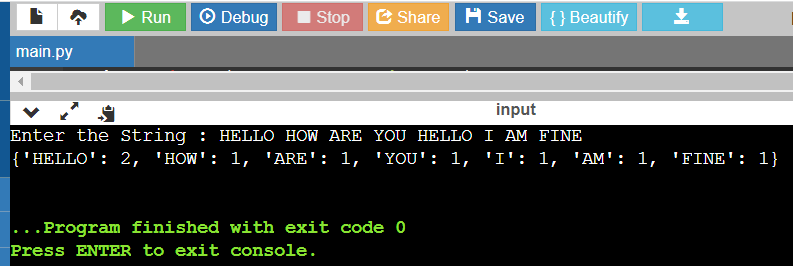
list = string.split()

newdict = dict()

for i in list:

newdict.update({i :list.count(i)})

print(newdict)



Pratical 5C:-----------------------------------------------------------------------------------------------------------------------

str = input("Enter a string : ")

x = str.split()

dict = {}

for word in x:

if (word[0] not in dict.keys()):

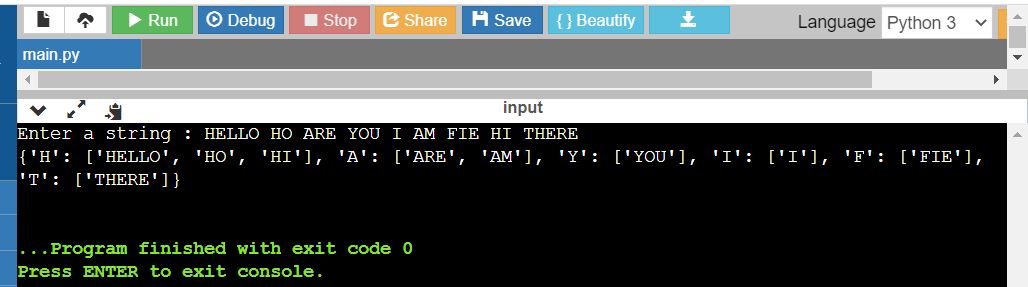
dict[word[0]] = [word]

else:

if( word not in dict[word[0]]):

dict[word[0]].append(word)

print(dict)



Pratical 6A=========================================================================

f = open("Demo.txt" , "w")

lines = [ "Hello i am Prathamesh\n" , "232\n"]

f.writelines(lines)

f.close()

f = open("Demo.txt" , "r")

print(f.readlines())

f.close()

Pratical 6B-----------------------------------------------------------------------------------------------------------------------

f= open('demo.txt','r')

txt=f.read()

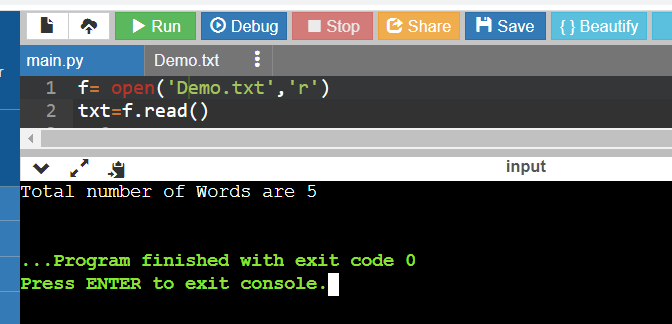
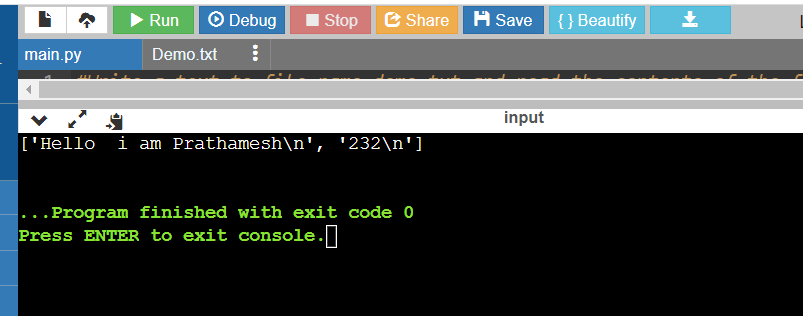
c=0

for i in txt.split():

c=c+1

print('Total number of Words are',c)

O/P:-

Pratical 6C-----------------------------------------------------------------------------------------------------------------------

f = open("Demo.txt" , "r")

word = "python"

wordData = f.read()

w = wordData.split()

wordCount = 0

for i in w:

if i == word :

wordCount = wordCount + 1

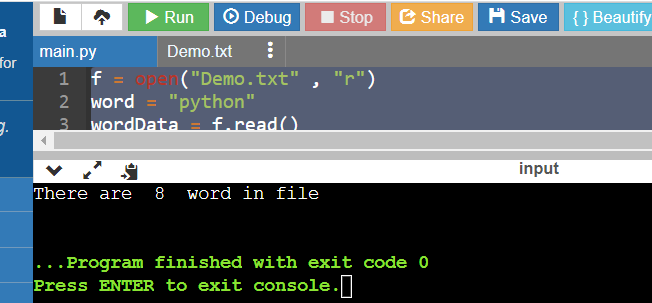
if wordCount == 0 :

print("Word not Found")

else:

print("There are " , wordCount , " word in file")

f.close()



Pratical 7 A=========================================================================

import pickle

F = open("Student.dat", "wb")

while True:

op = int(input("Enter 1 to add data and 0 to quit = "))

if (op == 1):

name = input("Enter Your Name : ")

rollno = int(input("Enter Your Roll Number : "))

pickle.dump([name, rollno], F)

elif (op == 0):

break

F = open("Student.dat", "rb")

print("\nContents in File \n")

while True:

try:

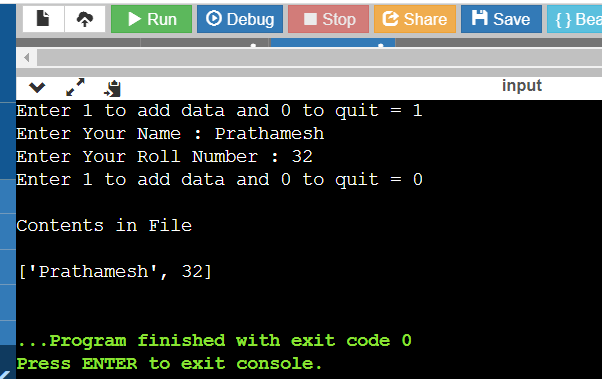
l = pickle.load(F)

print(l)

except EOFError:

break

F.close()

.

Pratical N07 B-------------------------------------------------------------------------------------------------------------------

import pickle

with open ("customer.dat", "wb") as F1:

while True:

op = input("Enter y to add data and n to quit = ")

if (op == 'y'):

name = input ("Enter Custmer Name : ")

city = input ("Enter Your City : ")

amount = int (input("Enter Amount : "))

pickle.dump([city,name,amount], F1)

else :

break

F1 = open ("customer.dat","rb")

while True:

try:

l = pickle.load(F1)

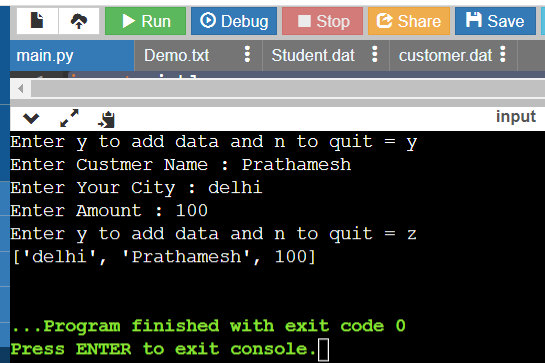
if (l[0].lower()=="delhi"):

print(l)

except EOFError:

break

F1.close()



Pratical 7 C-----------------------------------------------------------------------------------------------------------------------

import pickle

with open ("sports.dat", "wb") as F1:

while True:

op = input("Enter y to add data and n to quit = ")

if (op == 'y'):

event = input ("Enter Event Name : ")

location = input ("Enter Location : ")

participantNo = int (input("Enter Participant No. : "))

pickle.dump([event,location,participantNo], F1)

else:

break

F1 = open ("sports.dat","rb")

dict = {}

while True:

try:

l = pickle.load(F1)

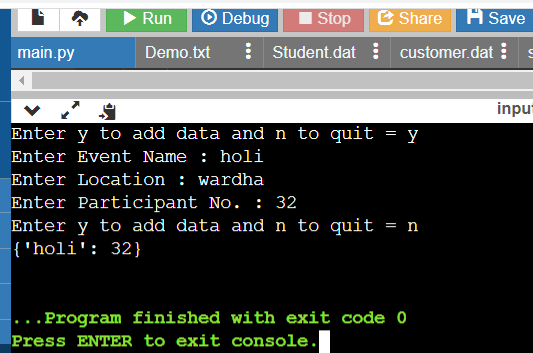
dict.update({l[0]:l[2]})

print(dict)

except EOFError:

break

F1.close()



Pratical 8A-=========================================================================

import csv

F = open("Student.csv", "w")

csvwriter = csv.writer(F)

while True:

op = input("Enter y to add data and n to quit = ")

if (op == 'y'):

bookno = input("Enter Your Book No : ")

bname = input("Enter Your Name : ")

bauthor = input("Enter Author Name : ")

bpublisher = input("Enter Publisher Name : ")

bprice = int(input("Enter Price : "))

csvwriter.writerow([bookno, bname , bauthor , bpublisher , bprice])

else :

break

F.close()

F = open("Student.csv", "r")

print("\nContents in File \n")

csvreader = csv.reader(F)

while True:

try:

for row in csvreader:

print(row)

except EOFError:

break

F.close()

Pratical 8 B-----------------------------------------------------------------------------------------------------------------------

import csv

with open ("employee.csv", "w" , newline='') as F1:

writer = csv.writer(F1)

while True:

op = input("Enter y to add data and n to quit = ")

if (op == 'y'):

emp = input("Enter Emp = ")

name = input ("Enter Customer Name : ")

salary = int (input("Enter Salary : "))

writer.writerow([emp,name,salary])

else:

break

F1.close()

F1 = open ("employee.csv","r")

reader = csv.reader(F1 , delimiter=',')

while True:

try:

for row in reader:

if( int(row[2]) > 800000 ):

print(row)

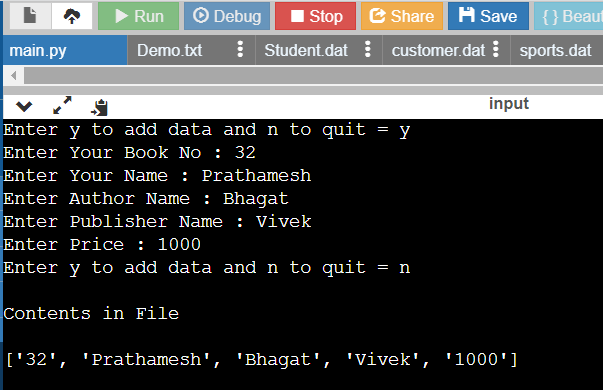
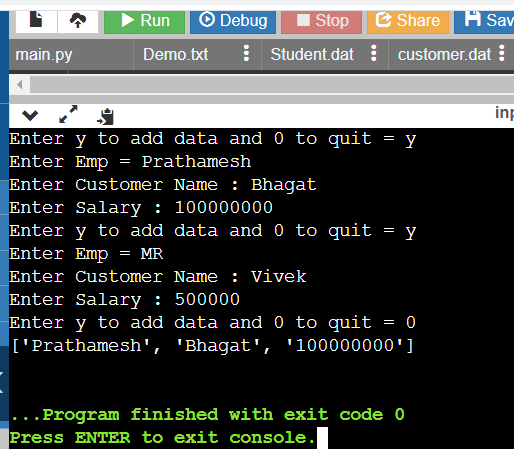
break

except EOFError:

print("No Entry Found")

break

F1.close() A: B:

Pratical 8C-----------------------------------------------------------------------------------------------------------------------

import csv

with open ("Books.csv", "w" , newline='') as F1:

csvwriter = csv.writer(F1)

while True:

op = input("Enter y to add data and n to quit = ")

if (op == 'y'):

bookno = input("Enter Your Book No : ")

bname = input("Enter Your Name : ")

bauthor = input("Enter Author Name : ")

bpublisher = input("Enter Publisher Name : ")

bprice = int(input("Enter Price : "))

csvwriter.writerow([bookno, bname , bauthor , bpublisher , bprice])

else:

break

F1.close()

F1 = open ("Books.csv","r")

csvreader = csv.reader(F1 , delimiter='@')

count=0

while True:

try:

for row in csvreader:

count+=1

print(row)

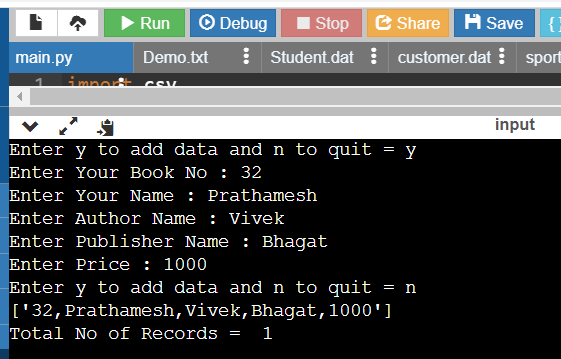
break

except EOFError:

break

print("Total No of Records = ", count)

F1.close()



Pratical 9A-----------------------------------------------------------------------------------------------------------------------

import math

class Sphere:

PI = math.pi

def \_\_init\_\_(self , radius):

self.radius = radius

def diameter( self ):

return 2\*self.radius

def surafceArea( self ):

return 4\*self.PI\*(self.radius\*\*2)

def volume( self ):

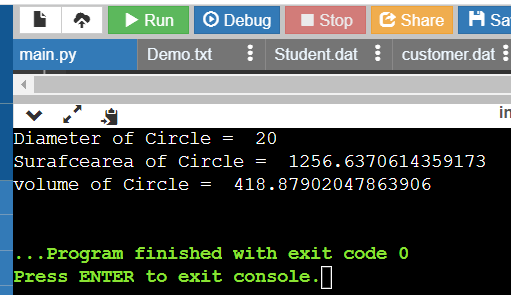
return (4/3)\*self.PI\*(self.radius\*\*2)

obj = Sphere(10)

print("Diameter of Circle = " , obj.diameter())

print("Surafcearea of Circle = " , obj.surafceArea())

print("volume of Circle = " , obj.volume())



Pratical 9 B-----------------------------------------------------------------------------------------------------------------------

class Account:

balance = 0

def \_\_init\_\_(self):

self.name = input("Enter your Name = ")

self.age = int(input("Enter your Age = "))

self.accountNo = input("Enter your Account No = ")

def deposit( self ):

amount = int(input("Enter Amount you want to deposit = "))

self.balance += amount

print("Amount Deposited")

self.viewBalance()

def withdraw( self ):

amount = int(input("Enter Amount you want to withdraw = "))

if( amount > self.balance ):

print(" cannot Withdraw ")

self.viewBalance()

else:

self.balance -= amount

print("not Suffient Balance")

self.viewBalance()

def viewBalance( self ):

print("\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n")

print("Name = " , self.name)

print("Age = " , self.age)

print("Account Number = " , self.accountNo)

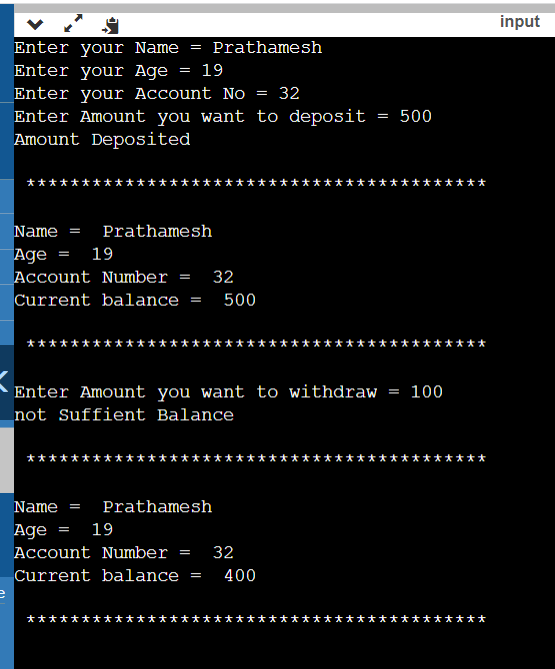
print("Current balance = " , self.balance)

print("\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n")

pvb = Account()

pvb.deposit()

pvb.withdraw()



Pratical 9C-----------------------------------------------------------------------------------------------------------------------

class Person:

def \_\_init\_\_(self , name , city ):

self.name = name

self.city = city

def display(self):

print(f"Name = {self.name}")

print(f"City = {self.city}")

class Student(Person):

def \_\_init\_\_(self , name , city , roll , dept ):

super().\_\_init\_\_(name , city)

self.roll = roll

self.dept = dept

def display(self):

super().display()

print(f"Roll = {self.roll}")

print(f"Deptartment = {self.dept}")

class Teacher(Person):

def \_\_init\_\_(self , name , city , desig , dept ):

super().\_\_init\_\_(name , city)

self.desig = desig

self.dept = dept

def display(self):

super().display()

print(f"Designation = {self.desig}")

print(f"Dept = {self.dept}")

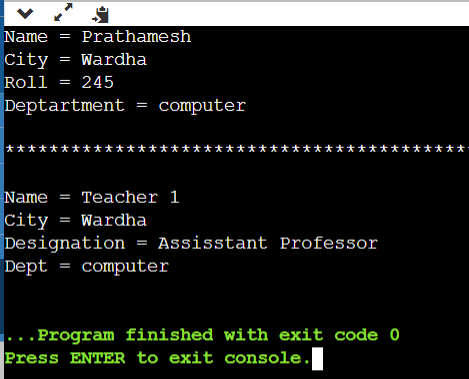
PVB= Student("Prathamesh " , "Wardha" , 245 , "computer")

PVB.display()

print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

teacher = Teacher("Teacher 1" , "Wardha" , "Assisstant Professor" , "computer")

teacher.display()



Pratical 10A=======================================================================

import sqlite3

conn = sqlite3.connect('employees.sqlite3')

cur = conn.cursor()

cur.execute('DROP TABLE IF EXISTS Employee ')

cur.execute('''CREATE TABLE Employee( EmpId integer, FirstName varchar(20), LastName varchar(20), Email varchar(25), PhoneNo varchar(25), Salary integer)''')

sqlInsert = "INSERT INTO Employee VALUES(?,?,?,?,?,?)"

val = (1,'James','Lion','james.lion@def.com','123.456.1834',55000)

cur.execute(sqlInsert , val)

emp = [ (4,'Lex','De Haan','lex@test.com','123.456.4569',15000), (3,'Neena','Kochhar','neena@test.com','123.456.4568',17000)]

print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Original\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

cur.executemany("INSERT INTO Employee VALUES(?,?,?,?,?,?)" , emp)

item = cur.execute("SELECT \* FROM Employee")

for i in item:

print(i)

print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*After Updating\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

cur.execute('''UPDATE Employee SET email = 'jking@test.com' WHERE EmpId = 1;''')

item = cur.execute("SELECT \* FROM Employee")

for i in item:

print(i)

print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*After Deleting\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

cur.execute("DELETE FROM Employee WHERE EmpId = 4")

item = cur.execute("SELECT \* FROM Employee")

for i in item:

print(i)

print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Salary Between 5000 and 20000\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

item = cur.execute("SELECT \* FROM Employee Where Salary Between 5000 and 20000")

for i in item:

print(i)

print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*After Ordering\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

item = cur.execute("SELECT \* FROM Employee ORDER BY FirstName")

for i in item:

print(i)

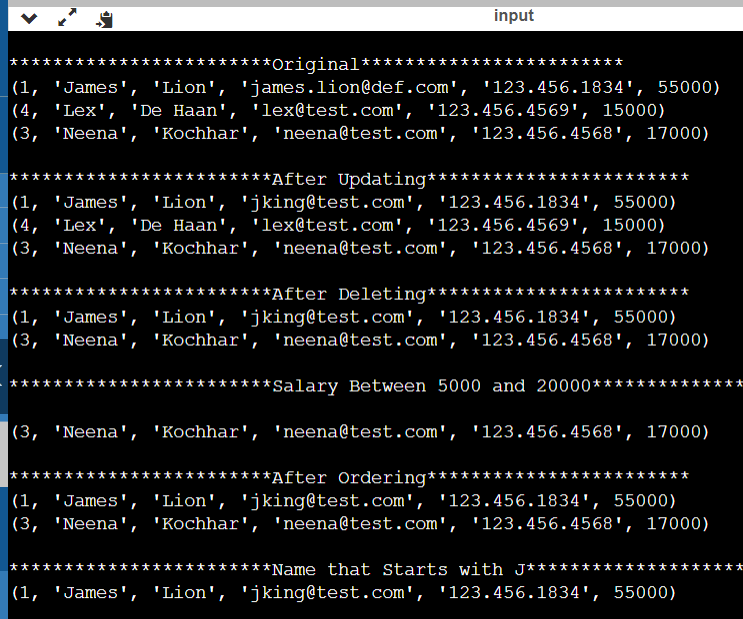
print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Name that Starts with J\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

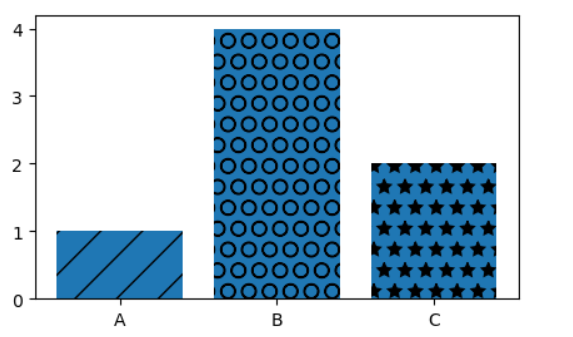
item = cur.execute("SELECT \* FROM Employee WHERE FirstName LIKE 'J%' ")

for i in item:

print(i)

conn.close()





Pratical 11 D---------------------------------------------------------------------------------------------------------------------

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

**import** pandas **as** pd

bins = [40,50,60,70,80,90,100]

plt.figure(figsize=(8,5))

plt.hist(fifa.Overall, bins=bins, color='#abcdef')

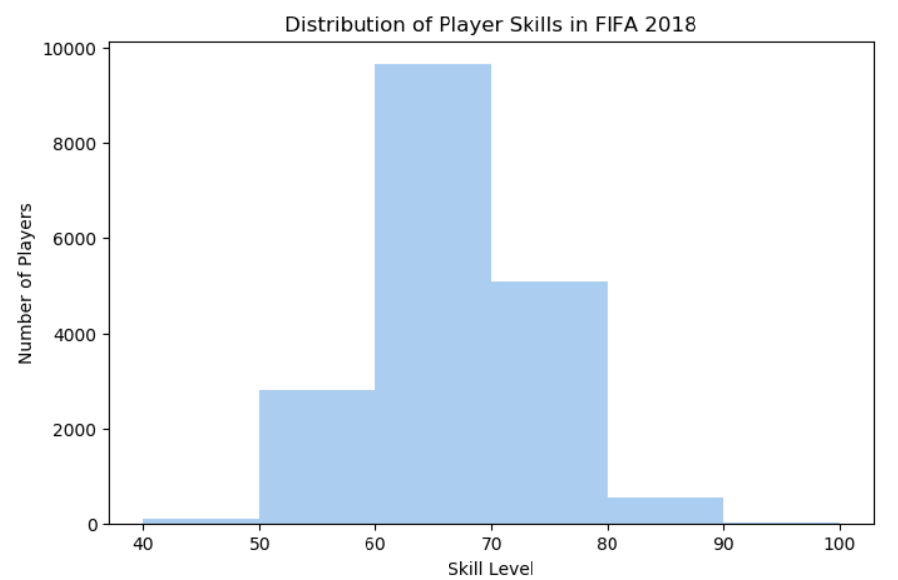
plt.xticks(bins) plt.ylabel('Number of Players')

plt.xlabel('Skill Level')

plt.title('Distribution of Player Skills in FIFA 2018')

plt.savefig('histogram.png', dpi=300)

plt.show()



Prtical 11e-----------------------------------------------------------------------------------------------------------------------

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

**import** pandas **as** pd

left **=** fifa**.**loc[fifa['Preferred Foot'] **==** 'Left']**.**count()[0]

right **=** fifa**.**loc[fifa['Preferred Foot'] **==** 'Right']**.**count()[0]

plt**.**figure(figsize**=**(8,5))

labels **=** ['Left', 'Right']

colors **=** ['#abcdef', '#aabbcc']

plt**.**pie([left, right], labels **=** labels, colors**=**colors, autopct**=**'%.2f %%')

plt**.**title('Foot Preference of FIFA Players')

plt**.**show()

