

#1.ILLUSTRATION OF FUNCTION PROGRAMMING

Develop a menu-driven program in Python using **user defined functions** to find the area of different shapes (Circle, Square and Rectangle).

SOURCE CODE

```
import math

def circle(r):
    return r**2*22/7

def square(s):
    return s**2                                #defining functions

def rectangle(l,b):
    return l*b

print("*****MENU*****")                    #designing of menu
print('1.Area of circle:')
print('2.Area of square')
print('3.Area of rectangle')

while True:
    ch=int(input('Enter your choice:'))
    if ch==1:
        rc=int(input('Enter radius:'))
        a=circle(rc)                          #Calling Function1
        print('Area of circle is :', a)
    elif ch==2:
        sc=int(input('Enter the side'))
        b=square(sc)                          #Calling Function2
        print('Area of square is :',b)
    elif ch==3:
        lc=int(input('Enter the length:'))
```

```
bc=int(input('Enter the breadth:'))
c=rectangle(lc,bc)           #Calling Function3
print('The area of rectangle is:',c)
ans=input('Do you wish to continue?:')
if ans in 'NOno':
    print('*****THANK YOU*****')
    break
```

OUTPUT

*****MENU*****

1. Area of circle:

2. Area of square

3. Area of rectangle

Enter your choice:1

Enter radius:2

Area of circle is : 12.571428571428571

Do you wish to continue?:y

Enter your choice:2

Enter the side4

Area of square is : 16

Do you wish to continue?:y

Enter your choice:3

Enter the length:56

Enter the breadth:7

The area of rectangle is: 392

Do you wish to continue?:n

*****THANK YOU*****

#2.ILLUSTRATION OF FUNCTION PROGRAMMING (USING LIST)

Develop a Python Program which includes two user defined functions namely,

~ create()- to create a list of roll numbers(integers)

~ findroll()- to search for a particular roll in a list

SOURCE CODE

```
l=[ ]
```

```
def create():                                #Creating a list of roll numbers
```

```
    global l
```

```
    while True:
```

```
        rno=int(input('Enter the roll no:'))
```

```
        l.append(rno)
```

```
        ans=input('Do you wish to continue:')
```

```
        if ans in "NOnO":
```

```
            print("Roll nos are:",l)
```

```
            break
```

```
def findroll():                              #Search for a file
```

```
    key = int(input("Enter the roll no:"))
```

```
    f=0
```

```
    for i in l:
```

```
        if key==i:
```

```
            f=1
```

```
            print('Roll no: found:',i)
```

```
            break
```

```
    if f==0:
```

```
        print("Record Not found")
```

```
#main begins
```

```

print("*****MENU*****")          #Designing of menu
print('1.To enter a roll no:')
print('2.To search for a roll no:')
print('3.Exit')
print("*****MENU*****")
while True:
    ch=int(input('Enter your choice:'))
    if ch==1:                        #Calling Function 1
        create()
    elif ch==2:                      #Calling Function2
        findroll()
    elif ch==3:                      #Calling Function3
        print("*****THANK YOU*****")
        break

```

OUTPUT

*****MENU*****

1.To enter a roll no:

2.To search for a roll no:

3.Exit

*****MENU*****

Enter your choice:1

Enter the roll no:11

Do you wish to continue:y

Enter the roll no:12

Do you wish to continue:y

Enter the roll no:13

Do you wish to continue:y

Enter the roll no:78

Do you wish to continue:n

Roll nos are: [11, 12, 13, 78]

Enter your choice:2

Enter the roll no:11

Roll no: found: 11

Enter your choice:2

Enter the roll no:79

Roll not Found

Enter your choice:3

*****THANK YOU*****

#3. ILLUSTRATION OF MODULE PROGRAMMING

#Create a package named "mymodule" containing three modules cube, sphere and cylinder to find the surface area and volume of these shapes. Later, import these modules into the main program and invoke the functions as a menu-driven program.

SOURCE CODE

mymodule.py

```
import math

def cube(side):\
    surfacearea=6*side**2
    vol=side**3
    print("The surface area of cube is:",surfacearea)
    print("The volume of cube is:",vol)

def sphere(r):
    surfacearea= 4*math.pi*r**2
    vol=(4/3)*math.pi*r**3
    print("The surface area of sphere is:",surfacearea)
    print("The volume of sphere is:",vol)

def cylinder(r,h):
    surfacearea=2*math.pi*r*h+2*math.pi*r**2
    vol=math.pi*r**2*h
    print("The surface area of cylinder is:",surfacearea)
    print("The volume of cylinder is:",vol)

import mymodule as m                # Importing module package
print("*****MENU*****")           #designing of menu
print('1.The surface area and volume of cube is:')
print('2.The surface area and volume of sphere is')
```

```

print('3.The surface area of cylinder is')
print('4.Exit')
print("*****MENU*****")
while True:
    ch=int(input('Enter your choice:'))
    if ch==1:
        side=int(input('Enter side of cube:'))
        m.cube(side)                #Calling Function1 from mymodule
    elif ch==2:
        r=int(input('Enter the radius of the sphere'))
        m.sphere(r)                #Calling Function2 from mymodule
    elif ch==3:
        r=int(input('Enter radius of cylinder:'))
        h=int(input('Enter height of cylinder:'))
        m.cylinder(r,h)            #Calling Function3 from mymodule
    elif ch==4:
        ans=input('Do you wish to continue?:')
        if ans in 'NOno':
            print('*****THANK YOU*****')
            break

```


OUTPUT

*****MENU*****

- 1.The surface area and volume of cube is:
- 2.The surface area and volume of sphere is
- 3.The surface area of cylinder is
- 4.Exit

*****MENU*****

Enter your choice:1

Enter side of cube:3

The surface area of cube is: 54

The volume of cube is: 27

Enter your choice:2

Enter the radius of the sphere6

The surface area of sphere is: 2714.336052701581

The volume of sphere is: 904.7786842338603

Enter your choice:3

Enter radius of cylinder:5

Enter height of cylinder:8

The surface area of cylinder is: 408.4070449666731

The volume of cylinder is: 15707.963267948966

Enter your choice:4

Do you wish to continue?:n

*****THANK YOU*****'''

#4. ILLUSTRATION OF TEXT FILE PROGRAMMING - I

Develop a python program to do the following tasks:

- # Create a text file with multiple lines of text in it.
- # Read the text File and display the number of vowels, consonants, lowercase, uppercase, spaces and digits in the file.

SOURCE CODE

```
def create():  
    f=open("lines.txt","w")           #creating the text File  
    f.write("Python is a programming language.\n")  
    f.write("It is easy to understand\n")  
    f.write("Python was created by GUIDO VAN ROSSUM in early 1990s\n")  
    f.close()  
create()  
  
def reading():  
    f=open("lines.txt","r")    #opening file in read mode  
    data=f.read()  
    up=0  
    lo=0  
    vow=0  
    con=0  
    dig=0  
    spc=0  
    for i in data :  
        if i.isupper():    #in built function  
            up+=1
```

```
if i.islower():  
    lo+=1  
if i in "AEIOUaeiou":  
    vow+=1  
if i.isalpha() and i not in 'aeiouAEIOU':  
    con+=1  
if i.isdigit():  
    dig+=1  
if i.isspace():  
    spc+=1
```

```
print("THE BELOW IS THE FILE CONTENT\n",s)  
print("No of upper characters are",up)  
print("No of lower characters are",lo)  
print("No of vowels are",vow)  
print("No of digits are",dig)  
print("No of consonants are",con)  
print("No of spaces are",spc)
```

```
reading()
```

OUTPUT

THE BELOW IS THE FILE CONTENT

Python is a programming language.

It is easy to understand

Python was created by GUIDO VAN ROSSUM in early 1990s

No of upper characters are 17

No of lower characters are 71

No of vowels are 32

No of digits are 4

No of consonants are 56

No of spaces are 20

#5.ILLUSTRATION OF TEXT FILE PROGRAMMING-II

Develop a python program to do the following task:

- ~ Create a text file with multiple lines of text in it and print the same, to the standard output device
- ~ Copy all those lines that contain the word 'the' to a new file.

SOURCE CODE

```
print("*****")
print("MENU")
print("1-Create a File")
print("2-Copy all those lines that contain the word 'the' to a new file")
print("3-exit")
print("*****")
def create():
    f=open("multi.txt",'w')
    while True:
        a=input("Enter a line")
        f.write(a+'\n')          #Writing Lines to the files
        ans=input("Do you wish to continue entering lines")
        if ans in 'Nn':
            break
    print("The content of the File is \n")
    f=open("multi.txt",'r')
    print(f.read())
    f.close()
```

```

def new():
    f1=open("multi.txt",'r')
    f2=open("multi2.txt",'w')
    lines=f1.readlines()
    for i in lines:
        words=i.split()
        for j in words:
            if j=='the':
                f2.write(i)

    f1.close()

    f2=open("multi2.txt",'r') #opening new file in read mode printing the contents
    print(f2.read())

# Main begins
while True:
    ch=int(input("Enter choice"))
    if ch==1:
        create()    # Function call
    elif ch==2:
        new()
    elif ch==3:
        print("*****THANK YOU*****")
        break

    ans=input("Do you wish to continue to go to next choice")
    if ans in 'Nn':
        break

```

OUTPUT

menu

1-Create a File

2-Copy all those lines that contain the word 'the' to a new file

3-exit

Enter choice1

Enter a lineThis is a line

do you wish to continue entering lines y

Enter a lineThis is the new line

do you wish to continue entering lines y

Enter a lineThe sun rises in the east

do you wish to continue entering lines y

Enter a lineEnd line

do you wish to continue entering lines n

The content of the File is

This is a line

This is the new line

The sun rises in the east

End line

do you wish to continue to go to next choice y

enter choice2

This is the new line

The sun rises in the east

#6.ILLUSTRATION OF TEXT FILE PROGRAMING-III

Develop a program to create a text file to with a story in it and do the following tasks:

~ To count the frequency of an inputted word in the file

~ To read a random line and display it.

SOURCE CODE

```
def create():                                #function definition
    f=open("story.txt","w")
    f.write("once upon a time there lived a ghost\n")
    f.write("he was known to be the killer and was feared the most\n")
    f.write("he was strong and scary\n")
    f.close()

def read():
    f1=open("story.txt","r")
    print("THE FILE CONTENT IS BELOW:\n\n")
    data=f1.read()                          #reading the content
    print(data)

def freq():
    f1=open("story.txt","r")
    content=f1.read()
    s=content.split()                       #splitting the content into words
    a=input("input a word to find the frequency")
    c=0
    for i in s:
        if i==a:
            c+=1
    print("The count of ",a, "is ",c)
```



```

import random

def random1():
    f2=open("story.txt","r")
    p=f2.readlines()
    s=random.randrange(0,len(p))
    print(p[s])

#Main begins.

print("*****MENU*****")

print("1: Create a file with a story")
print("2: Read and print the content")
print("3: Count the frequency of the inputted word")
print("4: Read and print the random file")

while True:
    ch=int(input("enter choice"))
    if ch==1:
        create()
    elif ch==2:
        read()
    elif ch==3:
        freq()
    elif ch==4:
        print("The random line is")
        random1()
    ans=input("Do you wish to continue")
    if ans in 'Nn':
        break

```

OUTPUT

*****MENU*****

1: Create a file with a story

2: Read and print the content

3: Count the frequency of the inputted word

4: Read and print the random file

enter choice1

Do you wish to continue y

enter choice2

THE FILE CONTENT IS BELOW:

once upon a time there lived a ghost

he was known to be the killer and was feared the most

he was strong and scary

Do you wish to continue y

enter choice3

input a word to find the frequency he

The count of he is 2

Do you wish to continue y

enter choice4

The random line is

once upon a time there lived a ghost

Do you wish to continue n

#7.ILLUSTRATION OF BINARY FILE PROGRAMING-I

A binary file named "movies.dat" contain certain records of certain movies (movieid, movieName, rating).Write a menu driven python program to do the following tasks:

1. Append a movie
2. Search for a movie based on the Movie ID
3. Read and display all movies

SOURCE CODE

```
import pickle
rec=[]
print('1.Append a record:\n2.Search a record:\n3.Read and display all:\n4.EXIT')
while True:
    choice=int(input('Enter your choice(1-4):'))
    if choice==1:
        movieid=int(input('Enter Movie ID:'))
        moviename=input('Enter Movie Name:')
        rating=float(input('Enter the Movie Rating'))
        data=[movieid,moviename,rating]          #Appending object
        rec.append(data)
        print('Record Appended Successfully')
        file=open('movies.dat','wb')
        pickle.dump(rec,file)                  #For writing record in a file
        file.close()
    elif choice==2:
        search=int(input('Enter Movie ID to be searched:'))
        file=open('movies.dat','rb')
        rec=pickle.load(file)
```

```

found=0
for i in rec:
    if i[0]==search:                #For equality check
        found=1
        print('Movie found:',i)
        break
if found==0:
    print("Movie Not Found")
file.close()
elif choice==3:
    file=open('movies.dat','rb')
    print("ID\tMOVIE\tRATING")
    rec=pickle.load(file)
    for i in rec:
        print(i[0],"\t",i[1],"\t",i[2])
elif choice==4:
    print('*****THANK YOU*****')
    break
else:
    print('Input out of range')
    break

```

OUTPUT

1.Append a record:

2.Search a record:

3.Read and display all:

4.EXIT

Enter your choice(1-4):1

Enter Movie ID: 101

Enter Movie Name: ABCD

Enter the Movie Rating4

Record Appended Successfully

Enter your choice(1-4):1

Enter Movie ID: 304

Enter Movie Name: DABBANG

Enter the Movie Rating2.5

Record Appended Successfully

Enter your choice(1-4):1

Enter Movie ID: 566

Enter Movie Name: DRISHYAM

Enter the Movie Rating4.5

Record Appended Successfully

Enter your choice(1-4): 3

ID	MOVIE	RATING
----	-------	--------

101	ABCD	4.0
-----	------	-----

304	DABBANG	2.5
-----	---------	-----

566	DRISHYAM	4.5
-----	----------	-----

Enter your choice(1-4): 2

Enter Movie ID to be searched: 304

Movie found: [304, 'DABBANG', 2.5]

#8.ILLUSTRATION OF BINARY FILE PROGRAMING-II

A binary file named "flight.dat" which will contain certain records of flight (flightid,flightname and number of passengers). Write a menu driven program to do the following task:

1. Append a record
2. Delete a record
3. Read and display all.

SOURCE CODE

```
import pickle
```

```
a=[]
```

```
def create():
```

```
    f=open("flight.dat","wb")
```

```
    rec1=[101,"jet airways",30]
```

```
    a.append(rec1)
```

```
    f=open("flight.dat","rb")
```

```
    h=f.read()
```

```
    print(h)
```

```
create()
```

```
def appending():
```

```
    f=open("flight.dat","rb+")          #Opening the file in read mode
```

```
    while True:
```

```
        e=int(input("Enter the flightid:"))
```

```
        n=input("Enter the flightname:")
```

```
        h=input("Enter the number of passangers:")
```

```
        rec=[e,n,h]
```

```
        a.append(rec)
```

```
        k=input("Do you wish to continue entering:")
```

```

        if k in 'Nn':
            break

    f.seek(0)
    pickle.dump(a,f)
    print("Records successfully appended")
    f.close()

def deleting():
    f=open("flight.dat","rb+")
    cv=pickle.load(f)
    c=int(input("enter the id of the record to be deleted:"))
    flag=0
    for i in cv:
        if i[0]==c:
            cv.remove(i)                # Removing the record
            print("Records successfully deleted")
            flag=0
        if flag==1:
            print("record found")
        else:
            f.seek(0)
            pickle.dump(cv,f)

    f.close()

def reading():                        # Loading the function
    f=open("flight.dat","rb")
    data=pickle.load(f)
    print("FLIGHTID\t FLIGHT\tNOOFPASSENGERS")
    for i in data:

```

```

        print(i[0],"\t",i[1],"\t",i[2])

#Main begins
print("*****MENU*****")
print("1.To Append a record.")
print("2.To Delete a record.")
print("3.To Read and display all.")
print("4.Exit")
print("*****MENU*****")
while True:
    ch=int(input("Enter your choice:"))
    if ch==1:
        appending()
    elif ch==2:
        deleting()
    elif ch==3:
        reading()
    elif ch==4:
        print("*****Thank You*****")
        break

```


OUTPUT

*****MENU*****

- 1.To Append a record.
- 2.To Delete a record.
- 3.To Read and display all.
- 4.Exit

*****MENU*****

Enter your choice: 1

Enter the flightid: 1089

Enter the flightname: AIR INDIA

Enter the number of passangers: 567

Do you wish to continue entering:y

Enter the flightid: 2345

Enter the flightname: SINGAPORE AIRLINES

Enter the number of passangers: 800

Do you wish to continue entering: n

Records successfully appended

Enter your choice:3

FLIGHTID	FLIGHT	NOOFPASSENGERS
101	jet airways	30
1089	AIR INDIA	567
2345	SINGAPORE AIRLINES	800

Enter your choice:2

Enter the id of the record to be deleted:1089

Records successfully deleted

Enter your choice:3

FLIGHTID	FLIGHT	NOOFPASSENGERS
101	jet airways	30
2345	SINGAPORE AIRLINES	800

#9.ILLUSTRATION OF BINARY FILE PROGRAMING-III

A binary file named inventory.dat contain certain records of stock (product id, product name, quantity and price).

Write a menu driven python program to do the following task:

1. Append a product record
2. Update a product based on the product id
3. Read and display all products

SOURCE CODE

```
print("*****MENU*****")

print("1.Append a record:")
print("2.Update a record based on product id:")
print("3.Read and display all:")
print("4.EXIT:")

import pickle

def create():

    f=open("inventory.dat",'wb') #creating the binary file
    l=[ ]

    while True:

        ID=int(input("Enter product id:"))
        name=input("Product name:")
        qty=input("Enter quantity:")
        price=int(input("Enter price:"))
        rec=[ID,name,qty,price]
        l.append(rec)
        ans=input("Do you wish to continue(y/n):")
        if ans in 'Nn':
            break
```

```

        pickle.dump(l,f)
        f.close()
create()
def appending():
    myfile=open("inventory.dat",'rb+')
    data=pickle.load(myfile)
    ID=int(input("Enter product id:"))
    name=input("Enter product name:")
    qty=int(input("Enter quantity:"))
    price=int(input("Enter price:"))
    rec=[ID,name,qty,price]
    data.append(rec)
    print(data)
    print("New record appended sucessfully")#appending to the file
    myfile.seek(0)      #seeking the pointer to the beginning of the file
    pickle.dump(s,myfile)
    myfile.close()
def updating():
    myfile=open("inventory.dat",'rb+')
    key=int(input("Enter the product id to be updated:"))#updating the file
    data=pickle.load(myfile)
    for i in data:
        if i[0]==key:
            print("The located product is",i)
            print("The current quantity is:",i[2])
            i[2]=int(input("Enter quantity:"))
            print("The current price is:",i[3])

```

```

        i[3]=int(input("Enter price:"))
    else:
        myfile.seek(0)
        pickle.dump(data,myfile)
        break

def reading():
    myfile=open("inventory.dat",'rb')
    data=pickle.load(myfile)
    print("Product ID\tProduct Name\tQuantity\tPrice")#displaying
    for i in data:
        print(i[0],'\t',i[1],'\t',i[2],'\t',i[3])
    myfile.close()

while True:
    ch=int(input("Enter choice:"))
    if ch==1:
        appending()
    elif ch==2:
        updating()
    elif ch==3:
        reading()
    elif ch==4:
        break
    ans=input("Do you wish to continue:")
    if ans in 'NOnO':
        break

```

OUTPUT

*****MENU*****

1.Append a record:

2.Update a record based on product id:

3.Read and display all:

4.EXIT:

Enter product id:1

Product name:Bag

Enter quantity:56

Enter price:89000

Do you wish to continue(y/n):y

Enter product id:2

Product name:Orange

Enter quantity:67

Enter price:1200

Do you wish to continue(y/n):n

Enter choice:3

Product ID	Product Name	Quantity	Price
1	Bag	56	89000
2	Orange	67	1200

Do you wish to continue: y

Enter choice:2

Enter the product id to be updated:1

The located product is [1, 'Bag', '56', 89000]

The current quantity is: 56

Enter quantity:50

The current price is: 89000

Enter price:12000

#10.SEARCH OPERATION USING DICTIONARY OBJECT

A binary file named "emp.dat" contain certain records of employees (empid, empname and salary).Write a menu driven python program to do the following tasks:

1. Append a record
2. Search a record
3. Read and display all

SOURCE CODE

```
import pickle
rec={}
print('1.Append a record:\n2.Search a record:\n3.Read and display all:\n4.EXIT')
while True:
    choice=int(input('Enter your choice(1-4):'))
    if choice==1:
        empid=int(input('Enter Employee ID:'))
        empname=input('Enter Employee Name:')
        salary=int(input('Enter Employee Salary:'))
        dvalue=[empname,salary]          #Appending dictionary object
        rec[empid]=dvalue
        print('Record Appended SUccessfully')
        file=open('emp.dat','wb')
        pickle.dump(rec,file)           #For dumping a file
        file.close()
    elif choice==2:
        file=open('emp.dat','rb')
        rec=pickle.load(file)
        found=0
        print("The Employees drawing salary above 45000 are:")
```

```

for i in rec:
    if rec[i][1]>45000:                #For equality check
        found=1
        print('Employee Record found:',rec[i])
if found==0:
    print("Employee Record not Found")
    file.close()
elif choice==3:
    file=open('emp.dat','rb')
    rec=pickle.load(file)
    print("EMPID\tNAME\tSALARY")
    for i in rec:
        print(i,"\t",rec[i][0],"\t",rec[i][1])
elif choice==4:
    print('THANK YOU')
    break

```

OUTPUT:

1.Append a record:

2.Search a record:

3.Read and display all:

4.EXIT

Enter your choice(1-4):1

Enter Employee ID: 101

Enter Employee Name: MANJUSHA

Enter Employee Salary: 56000

Record Appended SUccessfully

Enter your choice(1-4): 1

Enter Employee ID: 506

Enter Employee Name: ARUN

Enter Employee Salary: 25000

Record Appended SUccessfully

Enter your choice(1-4):1

Enter Employee ID: 809

Enter Employee Name: JISHA

Enter Employee Salary :80000

Record Appended SUccessfully

Enter your choice(1-4):3

EMPID	NAME	SALARY
101	MANJUSHA	56000
506	ARUN	25000
809	JISHA	80000

Enter your choice(1-4): 2

The Employees drawing salary above 45000 are:

Employee Record found: ['MANJUSHA', 56000]

Employee Record found: ['JISHA', 80000]

11. ILLUSTRATION OF CSV FILE PROGRAMMING– I

Develop a menu driven program implementing the user defined functions to perform different functions on a csv file named mobile.csv(modelid, modelname, modelprice)

1. Append a record
2. Updating a record based on modelid
3. Display all
- 4.Exit

SOURCE CODE

```
import csv

def write():
    f=open('mobile.csv','w',newline='')
    vh=csv.writer(f)      #enabling writer function for writing onto file
    rec=[]
    print('Enter the details one by one-')
    while True:
        id=int(input('Enter the model ID: '))
        n=input('Enter the model name: ')
        p=int(input('Enter the model price: '))
        l=[id,n,p]
        rec.append(l)
        ans=input('Do you wish to continue? ')
        if ans in 'Nn':
            break
        vh.writerows(rec)

def update():
    with open('mobile.csv','r') as f:
        l=[ ]
        data=csv.reader(f)
        for i in data:
```

```

        l.append(i)          #appending records to the list
f.close()
fh=open('mobile.csv','w',newline='')
key=input('enter the model id to be updated: ')
found=0
for i in l:
    if key==i[0]:
        print('Record before updation')
        print(i[0],'\t',i[1],'\t',i[2])
        i[1]=input('enter new model name: ')
        i[2]=int(input('enter new model price: '))
        found=1
        print('Record after updation-')
        print(i[0],'\t',i[1],'\t',i[2])
if found==0:
    print('Record not found')
fh.close()
with open ('mobile.csv','w',newline='') as f:
    rt=csv.writer(f)
    rt.writerows(l)
def display():
    with open('mobile.csv','r')as f:
        print('modelid ','modelName ','modelprice')
        rec=csv.reader(f)
        for i in rec:
            print(i[0],'\t',i[1],'\t',i[2])

#main begins
print('THE MENU')

```

```
print('*****')
print('1. Append a record')
print('2. Update a record based on modelid')
print('3. Display all')
print('4. Exit')
while True:          #main program begins
    ch=int(input('Enter choice: '))
    if ch==1:
        write()
    elif ch==2:
        update()
    elif ch==3:
        display()
    elif ch==4:
        break
    else:
        print('choice is not valid')
```

OUTPUT

THE MENU

1. Append a record
2. Update a record based on modelid
3. Display all
4. Exit

Enter choice: 1

Enter the details one by one-

Enter the model ID: 1001

Enter the model name: SAMSUNG

Enter the model price: 13000

Do you wish to continue? y

Enter the model ID: 1009

Enter the model name: XIAMI

Enter the model price: 40000

Do you wish to continue? y

Enter the model ID: 103

Enter the model name: REDMI

Enter the model price: 10000

Do you wish to continue? n

Enter choice: 3

Modelid	Modelname	Modelprice
1001	SAMSUNG	13000
1009	XIAMI	40000
103	REDMI	10000

Enter choice: 2

Enter the model id to be updated: 1009

Record before updation

1009 XIAMI 40000

Enter new model name: XIOMI

Enter new model price: 50000

Record after updation-

1009 XIOMI 50000

#12. ILLUSTRATION OF CSV FILE PROGRAMMING– II

Develop a menu driven program implementing the user defined functions to perform different functions on a csv file named library.csv(bookid, bookname, noofcopies)

1. Append a record
2. Searching a record based on bookid
3. Display all
4. Exit

SOURCE CODE

```
import csv
def write():
    f=open('library.csv','w',newline='')
    vh=csv.writer(f)
    rec=[]
    print('Enter the details one by one-')
    while True:
        id=int(input('Enter the book ID: '))    #inputting fields
        n=input('Enter the book name: ')
        p=int(input('Enter the no of copies: '))
        l=[id,n,p]
        rec.append(l)
        ans=input('Do you wish to continue? ')
        if ans in 'Nn':
            break
    vh.writerows(rec)                        # appending onto file
def search():
    with open('library.csv','r') as f:        # opening file in read mode
        data=csv.reader(f)                  #enabling reader function for reading
    found=0
    key=input('enter the book id to be searched: ')
```

```

print('BookID\tBookname\tno of copies')
for i in data:
    if i[0]==key:
        print(i[0],'\t',i[1],'\t',i[2])
        found=1
if found==0:
    print('Record not found')
def display():
    with open('library.csv','r')as f:
        print('BookID\tBookname\tNo of copies')
        rec=csv.reader(f)
        for i in rec:
            print(i[0],'\t',i[1],'\t',i[2])
#Main Begins
print('THE MENU')
print('1. Append a record')
print('2. Searching a record based on bookid')
print('3. Display all')
print('4.Exit')
while True:
    ch=int(input('enter choice: '))
    if ch==1:
        write()
    elif ch==2:
        search()
    elif ch==3:
        display()
    elif ch==4:
        break

```

OUTPUT

THE MENU

1. Append a record
2. Searching a record based on bookid
3. Display all
- 4.Exit

Enter choice: 1

Enter the details one by one-

Enter the book ID: 34

Enter the book name: The Wondergirl

Enter the no of copies: 645

Do you wish to continue? y

Enter the book ID: 54

Enter the book name: Inferno

Enter the no of copies: 764

Do you wish to continue? n

Enter choice: 2

Enter the book id to be searched: 34

BookID	Bookname	No of copies
34	The Wondergirl	645

Enter choice: 3

BookID	Bookname	No of copies
34	The Wondergirl	645
54	Inferno	764

#13. ILLUSTRATION OF CSV FILE PROGRAMMING– III

Develop a menu driven program implementing the user defined functions to perform different functions on a csv file named contacts.csv(name, phone)

1. Append a contact
2. Count the number of contacts in the file
3. Display all contacts
3. Exit

SOURCE CODE

```
import csv

def append():
    f=open('contacts.csv','w',newline='')
    vh=csv.writer(f)
    l=[]
    print('Enter the details one by one-')
    while True:
        name=input('enter the name: ')
        phone=int(input('enter the phone number: '))
        rec=[name,phone]
        l.append(rec)
        ans=input('Do you wish to continue? ')
        if ans in 'nN':
            break
    vh.writerows(l)

def count():
    f=open('contacts.csv','r')
    data=csv.reader(f)
    c=0
    for i in data:
        c+=1
```



```

        print('Count of records is', c)
        f.close()
def display():
    with open('contacts.csv','r')as f:
        print('name\tnumber')
        rec=csv.reader(f)
        for i in rec:
            print(i[0],'\t',i[1])            # printing in tabular format
#Main begins
print('THE MENU')
print('*****')
print('1. Append a record')
print('2. Count the number of records')
print('3. Display all')
print('4. Exit')
while True:
    ch=int(input('enter choice: '))
    if ch==1:
        append()
    elif ch==2:
        count()
    elif ch==3:
        display()
    elif ch==4:
        break

```

OUTPUT

THE MENU

1. Append a record
2. Count the number of records
3. Display all
4. Exit

Enter choice: 1

Enter the details one by one-

Enter the name: Bessy

Enter the phone number: 9539233233

Do you wish to continue? y

Enter the name: Mani

Enter the phone number: 9946645456

Do you wish to continue? n

Enter choice: 3

Name	Number
Bessy	9539233233
Mani	9946645456

Enter choice: 2

Count of records is 2

#14. ILLUSTRATION OF STACK PROGRAMMING USING LIST OF INTEGERS – I

Develop a program to implement the following stack operation in python using list of integers according to the user's choice

1. Push an integer to the stack
2. Pop integer from the stack
3. Display the stack
4. Exit

SOURCE CODE

```
stack=[]
def isEmpty(stk):
    if stk==[]:
        return True        #function for empty stack
    else:
        return False
def PUSH(stk,elm):
    stk.append(elm)
    print("Element inserted")
def POP(stk):
    print("Element deleted is",stk.pop())
#Main Begins
print("1.Push")
print("2.Pop")           #menu
print("3.Display")
print("4.Exit")
while True:
    ch=int(input("Enter your choice:"))
    if ch==1:
        itemname=input("Enter element to be inserted:")
        PUSH(stack,itemname)
```

```

        print("The stack after insertion is:",stack)
        print("*****")
elif ch==2:
    if isEmpty(stack):
        print("STACK EMPTY/Underflow Case")
    else:
        POP(stack)      #function call
        print("Stack after deletion is",stack)
        print("*****")
elif ch==3:
    print("Stack is:")
    print("*****")
    for i in range(len(stack)-1,-1,-1):      #printing stack
        print(stack[i])
elif ch==4:
    print("End of code")
    print("*****THANK YOU*****")
    break

```

OUTPUT

1.Push

2.Pop

3.Display

4.Exit

Enter your choice:1

Enter element to be inserted:11

Element inserted

The stack after insertion is: ['11']

Enter your choice:1

Enter element to be inserted:22

Element inserted

The stack after insertion is: ['11', '22']

Enter your choice:1

Enter element to be inserted:33

Element inserted

The stack after insertion is: ['11', '22', '33']

Enter your choice:3

Stack is:

33

22

11

Enter your choice:2

Element deleted is 33

Stack after deletion is ['11', '22']

15. # ILLUSTRATION OF STACK PROGRAMMING USING LIST OF VOTERS – II

Develop a python program to implement the following operations on a stack containing voters list details (ID,NAME,AGE)as per user's choice.

1. Push a voter record to the stack
2. Pop a voter record from the stack
3. Display the list of voters
4. Exit

SOURCE CODE

```
def isEmpty(stk):
    if stk==[]:
        return True
    else:
        return False

def push(stk,elm):
    stk.append(elm)
    print("Voters details inserted")

def pop(stk):
    print("The deleted voter is ",stk.pop())

stack=[ ]

print("1.Push a Voter detail")           #menu
print("2.Pop a voter")
print("3.Display all Voters ")
print("4.exit")

while True:
    ch=int(input("Enter the choice"))
    if ch==1:
        name=input("Enter voter name")
        vid=int(input("Enter voter id"))           #entering details
        age=int(input("Enter voter age"))
```

```

        data=[name,vid,age]
        push(stack,data)
        print("The stack after insertion is", stack)
        print("*****")
elif ch==2:
    if isEmpty(stack):
        print("Underflow case")
    else:
        pop(stack)                                #function call
        print("The stack after deletion is",stack)
        print("*****")
elif ch==3:
    print("The stack is:")
    for i in range(len(stack)-1,-1,-1):
        print(stack[i])
    print("*****")
elif ch==4:
    print("End of code")
    print("*****THANK YOU*****")
    break

```

OUTPUT

1.Push a Voter Detail

2.Pop a voter

3.Display all Voters

4.exit

Enter the choice1

Enter voter nameRaj

Enter voter id12

Enter voter age56

Voters details inserted

The stack after insertion is [['Raj', 12, 56]]

Enter the choice1

Enter voter nameMinsa

Enter voter id45

Enter voter age34

Voters details inserted

The stack after insertion is [['Raj', 12, 56], ['Minsa', 45, 34]]

Enter the choice3

Stack is:

['Minsa', 45, 34]

['Raj', 12, 56]

Enter the choice2

The deleted voter is ['Minsa', 45, 34]

The stack after deletion is [['Raj', 12, 56]]

16. ILLUSTRATION OF STACK PROGRAMMING USING LIST OF PRODUCTS – III

Develop a program to implement the following stack operations in python using list of product names.

1. Push an item name to the stack(the products whose name starts with b/B)
2. Pop item name from the stack
3. Display the stack
4. Exit

SOURCE CODE

```
def isEmpty(stk):
    if stk==[]:
        return True #checking for stack empty
    else:
        return False

def push (stk,elm):
    stk.append(elm)
    print("The product added successfully")

def pop(stk):
    print("Element deleted is ",stk.pop())

#Main Begins
stack=[]
print("1.Push a Product detail(as per criteria) whose name start with b/B")
print("2.Pop a Product")
print("3.Display all products")
print("4.EXIT")
while True:
    ch=int(input("Enter your choice"))
    if ch==1:
        n=int(input("Enter the number of elements to be pushed"))
```

```

for i in range(n):
    itemname=input("Enter the name to be inserted")
    if itemname[0] in "Bb":
        push(stack,itemname)    #fn call
    print("The stack after insertion is ",stack)
    print("*****")
elif ch==2:
    if isEmpty(stack):
        print("underflow case")
    else:
        pop(stack)    #fn call
    print("The stack after deletion is",stack)
    print("*****")
elif ch==3:
    print("The Stack is:")
    for i in range(len(stack)-1,-1,-1):
        print(stack[i])
    print("*****")
elif ch==4:
    print("end of code")
    print("*****THANKK YOU*****")
    break

```

OUTPUT

1. Push a Product detail(as per criteria) whose name start with b/B
2. Pop a Product
3. Display all products
4. EXIT

Enter your choice1

Enter the number of elements to be pushed5

Enter the name to be inserted Bag

The product added successfully

Enter the name to be inserted Can

Enter the name to be inserted tin

Enter the name to be inserted bin

The product added successfully

Enter the name to be inserted bread

The product added successfully

The stack after insertion is ['Bag', 'bin', 'bread']

Enter your choice3

The stack is:

bread

bin

Bag

Enter your choice2

Element deleted is bread

The stack after deletion is ['Bag', 'bin']

17. ILLUSTRATION OF CONNECTIVITY PROGRAMMING-I

Integrate SQL with Python by importing the MYSQL module and to implement the DML command (SELECT).

Create a table STUDENT (Roll, Name, Stream, Section).Populate the table with 4 records of your choice. Display all the records of student table.

SOURCE CODE

```
import mysql.connector as mc
try:

    con=mc.connect(host='localhost',user='root',password='password',database='recordwork1')
    if (con.is_connected()):    #checking if the connection is established
        print('success')
    else:
        print('connector not established')
    cur=con.cursor()
    cur.execute('select * from student')
    rows=cur.fetchall()        #retrieving data from the resultset
    print('Data from the student table is as follows:')
    for i in rows:              #displaying the table
        print(i[0],'\t',i[1],'\t',i[2],'\t',i[3])
    cur.close()
    con.close()

except Exception as e:
    print(e)
```

OUTPUT

success

Data from the table student is as follows:

1	AARADHYA	BIO INFO	A
2	BENJAMIN	COMPSCIENCE	C
3	CHRISTY	HUMANITIES	E
4	DHAYA MUHAMMED	COMMERCE INFO	D

18. ILLUSTRATION OF CONNECTIVITY PROGRAMMING-II

Integrate SQL with python by importing the MYSQL module and to implement the DML commands(INSERT and SELECT).

Populate the STUDENT(Roll, Name, Stream, Section) table with 4 records of your choice using INSERT query and display all the records by using the appropriate Query

SOURCE CODE

import mysql.connector as a

try:

```
con=a.connect(host="localhost",user="root",password="password",database="recordwork")
```

```
if con.is_connected():          #checking whether connected or not
    print("Connected Successfully")
```

```
else:
```

```
    print("Error")
```

```
cur=con.cursor()                #creating Cursor Instance
```

```
while True:
```

```
    roll=int(input("Enter the roll no of the student:"))
```

```
    name=input("Enter the name of the student:")
```

```
    stream=input("Enter the stream of the student:")
```

```
    section=input("Enter the section of the student:")
```

```
    cur.execute("insert into student
values({},'{}','{}','{}')".format(roll,name,stream,section))
```

```
    con.commit()                #Saving to database
```

```
    print("Record Inserted Successfully")
```

```
    ans=input("Do you wish to enter more?")
```

```
    if ans in "Nn":
```

```
        break
```

```
print()
```

```

print("The data in table student is as follows:")
cur.execute("select * from student")
data=cur.fetchall() # retrieving data from resultset

print("ROLL\tNAME\tSTREAM\tSECTION")
for i in data:                                     #displaying the table
    print(i[0],"\t",i[1],"\t",i[2],"\t",i[3])
except Exception as e:
    print(e)

```

OUTPUT

Connected Successfully

Enter the roll no of the student: 5

Enter the name of the student: Nakul

Enter the stream of the student: Biomath

Enter the section of the student: A

Recorded Inserted Successfully

Do you wish to enter more? n

The data in table student is as follows:

ROLL	NAME	STREAM	SECTION
1	AARADHYA	BIO INFO	A
2	BENJAMIN	COMPSCIENCE	C
3	CHRISTY	HUMANITIES	E
4	DHAYA MUHAMMED	COMMERCE INFO	D
5	Nakul	Biomath	A

19. ILLUSTRATION OF CONNECTIVITY PROGRAMMING-III

Integrate SQL with python by importing the MYSQL module and to implement the DML commands(UPDATE and SELECT).

Populate the STUDENT(Roll, Name, Stream, Section)table with 4 records of your choice and do the following tasks:

~ Accept the roll no: of the student to be Updated.

~ Update the Record

~ Display all the records (After Updation)

SOURCE CODE

```
import mysql.connector as a

con=a.connect(host="localhost",user="root",password="password",database="recordwork")

if con.is_connected():          #checking whether connected or not
    print("Connected Successfully")
else:
    print("Error")

try:
    cur=con.cursor()            #Creating Cursor Instance
    while True:
        roll=int(input("Enter the roll no of the student to be updated:"))
        name=input("Enter the new name-") #updating the data
        cur.execute("Update student set name='{}' where roll={}".format(name,roll))
        print("Record updated successfully")
        con.commit()
        ans=input("Do you wish to update more?")
        if ans in "Nn":
            break
    print("*****")
```

```

print()
print("The data in table student is as follows:")
cur.execute("Select * from Student")
data=cur.fetchall()      #retrieving data from resultset
print("ROLL\tNAME\tSTREAM\tSECTION")
for i in range(0,len(data)):      #displaying the table
    print(i[0],"\t",i[1],"\t",i[2],"\t",i[3])
except Exception as e:
    print(e)

```

OUTPUT

Connected Successfully

Enter the roll no of the student to be updated:5

Enter the new name - NEEMA

Record updated successfully

Do you wish to update more? y

Enter the roll no of the student to be updated:2

Enter the new name-BEN

Record updated successfully

Do you wish to update more? n

The data in table student is as follows:

ROLL	NAME	STREAM	SECTION
1	AARADHYA	BIO INFO	A
2	BEN	COMPSCIENCE	C
3	CHRISTY	HUMANITIES	E
4	DHAYA MUHAMMED	COMMERCE INFO	D
5	NEEMA	Biomath	A

20. ILLUSTRATION OF CONNECTIVITY PROGRAMMING-IV

Integrate SQL with python by importing the MYSQL module and to implement the DML commands(DELETE and SELECT).

Populate the STUDENT(Roll, Name, Stream, Section)table with 4 records of your choice and do the following tasks:

~ Accept the roll no: of the student to be deleted.

~ Delete the Record

~ Display all the records (After deletion)

SOURCE CODE

```
import mysql.connector as a

con=a.connect(host="localhost",user="root",password="password",database="recordwork")

if con.is_connected():          #checking whether connected or not
    print("Connected Successfully")
else:
    print("Error")

try:
    cur=con.cursor()            #creating Cursor Instance
    while True:
        roll=int(input("Enter the roll no of the student to be deleted:"))
        cur.execute("Delete from student where roll={}".format(roll))
        print("Record deleted successfully")
        con.commit()
        ans=input("Do you wish to delete more?")
        if ans in "Nn":
            break
    print("*****")
    print()
```

```

        print("The data in table student is as follows:")
        cur.execute("select * from student")
        data=cur.fetchall()           #retrieving data from resultset
        print("ROLL\tNAME\tSTREAM\tSECTION")
        for i in data:                #displaying the table
            print(i[0],"\t",i[1],"\t",i[2],"\t",i[3])
except Exception as e:
    print(e)

```

OUTPUT

Connected Successfully

Enter the roll no of the student to be deleted:4

Record deleted successfully

Do you wish to delete more?n

The data in table student is as follows:

ROLL	NAME	STREAM	SECTION
1	AARADHYA	BIO INFO	A
2	BEN	COMPSCIENCE	C
3	CHRISTY	HUMANITIES	E
5	NEEMA	Biomath	A