PYTHON PROGRAMMING

INDEX

Program – 1

Write a program using function that returns sum and product of all elements of a list

Program

```
#Function to find sum
def Sum(x):
    s=0
    for i in x:
        s=s+i
    return s

#Function to find product
def product(x):
    p=1

    for i in x:
        p=p*i
    return p

#Main program
n=int(input("Enter the limit : "))
l=[]
for i in range(n):
          e=int(input("Input elements : "))
           l.append(e)
print("Sum of elements in the list : ",Sum(l))
print("Product of elements in the list : ",product(l))
```

Output

```
Enter the limit : 5
Input elements : 4
Input elements : 2
Input elements : 1
Input elements : 3
Input elements : 6
Sum of elements in the list : 16
Product of elements in the list : 144
```

Program – 2

Write a program using function to display n terms of Fibonacci series

```
#Function to fibonacci series
def fib(n):
    f=0
    s=1
    l=[]
    l.append(f)
    l.append(s)
    for i in range(3,n+1):
        t=f+s
        l.append(t)
        f=s
        s=t
    print("Fibonacci series")
    print(l)

#Main program
n=int(input("Enter the limit : "))
fib(n)
```

Output

```
Enter the limit : 5
Fibonacci series
[0, 1, 1, 2, 3]
```

Program – 3

Write a menu driven program to accept a line of text and to

1. **Print the palindromic words**
2. **Display the longest word**

```
#Function to display palindromic words
def pal(x):
    print("Palindromic words are")
    for i in x:
        if i==i[::-1]:
            print(i)

#Function to display longest word
def long(x):
    big=0
    n=""
    for i in x:
        if len(i)>big:
            big=len(i)
            n=i
    print("Longest word is :",n)

#Main program
s=input("Enter a text : ")
t=s.split()
while True:
    print("1. Palindromic words")
    print("2. Longest word")
    print("3. Exit")
    ch=int(input("Enter choice : "))
    if ch==1:
        pal(t)
    elif ch==2:
        long(t)
    elif ch==3
          break
    else:
          print(“wrong choice”)
```

Output

```
Enter a text : I love my dad and mom
1. Palindromic words
2. Longest word
3. Exit
Enter choice : 1
Palindromic words are

I
dad
mom
1. Palindromic words
2. Longest word
3. Exit
Enter choice : 2
Longest word is : love
1. Palindromic words
2. Longest word
3. Exit
Enter choice : 3
```

Program – 4

Define a function which takes an integer as argument and returns 0 if the number is prime and 1 if it is not prime. Using this function, write a program to copy the prime numbers of a list into a new list

```
#Function to display prime numbers
def prime(x):
    if x!=1:
              for j in range(2,x):
                        if x%j==0:
                              return 1
             else:
                              return 0
#Main program

n=int(input("Enter limit : "))
l=[]
t=[]
for i in range(n):
    e= int(input("Enter elements : "))
    l.append(e)
for i in l:
    p=prime(i)
    if p==0:
        t.append(i)
print("Prime numbers are")
print(t)
```

Output

```
Enter limit : 6
Enter elements : 1
Enter elements : 2
Enter elements : 3
Enter elements : 4
Enter elements : 5
Enter elements : 6
Prime numbers are
[ 2, 3, 5]
```

Program – 5

Write program using function which takes a list as argument and swaps the adjacent elements

```
#Function to swap adjacent elements
def swap(x,n):
    print("Original list is : ",x)
    if n%2==0:
        for i in range(0,n,2):
            x[i],x[i+1]=x[i+1],x[i]
    else :
        for i in range(0,n-1,2):
            x[i],x[i+1]=x[i+1],x[i]
    print("After swapping, list is : ",x)

#Main program
n=int(input("Enter limit : "))
l=[]
for i in range(n):
    e=int(input("Enter elements : "))
    l.append(e)
swap(l,n)
```

Output

```
Enter limit : 6
Enter elements : 11
Enter elements : 22
Enter elements : 33
Enter elements : 44
Enter elements : 55
Enter elements : 66
Original list is : [11, 22, 33, 44, 55, 66]
After swapping, list is :  [22, 11, 44, 33, 66, 55]
```

Program – 6

Create a dictionary to store your friends' names and their birthday. Write a program using 3 functions which takes the dictionary as argument and implements the following

- **Display the birthday of a particular person**
- **Add /modify a friend's birthday**
- **Delete the data of a friend**
- **Exit**

```
#Function to Display
def disp(d):
    n=input("Enter the name of a person whose birthday has to be displayed")
    print(d[n],"is the birthday of ",n)

#Function to Modify
def modify(d):
    n=input("Enter name whose birthday is to be modified: ")
    b=input("Enter birthday : ")
    d[n]=b
    print(d)

#Function to Delete
def dele(d):
    n=input("Enter name to be deleted: ")
    del d[n]
    print(d)

#Main program
n=int(input("Enter limit : "))
d={}
for i in range(n):
    n=input("Enter name : ")
    b=input("Enter birthday : ")
    d[n]=b
while True:
    print("1. Display birth day")
    print("2. Modify birth day")
    print("3. Delete friend")
    print("4. Exit")
    ch=int(input("Enter choice : "))
    if ch==1:
        disp(d)
    elif ch==2:
        modify(d)
    elif ch==3:
       dele(d)
    elif ch==4:
         break
    else:
        print('Wrong Input')
```

Output

```
Enter limit : 2
Enter name : Ramu
Enter birthday : 25/5/1980
Enter name : Sanju
Enter birthday : 30/6/1981
1. Display birth day
2. Modify birth day
3. Delete friend
4. Exit
Enter choice :  1
Name            Birthday
Ramu            25/5/1980
Sanju 30/6/1981
1. Display birth day
2. Modify birth day
3. Delete friend
4. Exit
Enter choice : 2
Enter name whose birthday is to be modified: Sanju
Enter birthday : 12/8/1981
Name            Birthday
Ramu            25/5/1980
Sanju 12/8/1981
1. Display birth day
2. Modify birth day
3. Delete friend
4. Exit
Enter choice : 3
Enter name to be deleted: Ramu
Name            Birthday
Sanju 12/8/1981
```

Program – 7

Write a python program to read a text file line by line and display each word separated by #.

```
#python program to read a text file line by line and display
#each word separated by  #.
f=open("sample.txt","w")
str="Python is a programming Language.Python is used for writing AI program."
f.write(str)
f.close()
print("Displaying the contents of the file")
f=open("sample.txt","r")
ch=f.readlines()
for i in ch:
    print(i)
f.close()
print("Displaying each word in the file separated by '#'\n")
f=open("sample.txt","r")
ch=f.read()
ch=ch.replace(" ",'#')
print(ch)
f.close()
```

Output

```
Displaying the contents of the file
Python is a programming Language.Python is used for writing AI program.

Displaying each word in the file separated by '#'

Python#is#a#programming#Language.Python#is#used#for#writing#AI#program.
```

Program – 8

Write a program to create a text file. Display the contents of the text file. Remove all the lines that contain the character `a' in a file and write it to another file. Display the new file.

```
f=open("sample.txt","w")
str="Python is a programming language.\nIt is so simple"
f.write(str)
f.close()
print("Displaying the contents of the file\n")
f=open("sample.txt","r")
ch=f.readlines()
for i in ch:
    print(i)
f.close()
f1=open('New.txt','w')
f=open("sample.txt","r")
ch=f.readlines()
for i in ch:
    if 'a' not in i :
        f1.write(i)
f.close()
f1.close()
print("\nDisplaying the contents of the file which do not contain 'a' ")
f1=open('New.txt','r')
ch=f1.read()
print(ch)
f1.close()
```

Output

```
Displaying the contents of the file
Python is a programming language.
It is so simple
Displaying the sentence which do
not contain 'a'It is so simple
```

Program – 9

Write a program to create a text file. Read the text file and display the number of vowels/ consonants / uppercase/ lowercase characters in the file

```
#Read a text file and display the number of vowels, consonants,
#uppercase, lowercase characters in the file.
    #creating a text file
    f=open("sample.txt","w")
    str=input("Enter text : ")
    f.write(str)
    f.close()
    #displaying text file
    f=open("sample.txt","r")
    ch=f.read()
    print("The Contents of the file:")
    print(ch)
    f.close()
    f=open("sample.txt","r")
    l=f.read()
    vow=0
    con=0
    up=0
    lp=0
    for j in l:
        if j in ['a','e','i','o','u','A','E','I','O','U']:
            vow=vow+1
        elif j.isalpha():
            con=con+1
        if j.isupper():
            up=up+1
        elif j.islower():
            lp=lp+1
    print("Number of vowels : ",vow)
    print("Number of consonants : ",con)
    print("Number of upper case : ",up)
    print("Number of lower case : ",lp)
```

Output

```
Enter text : Python is a simple language
Number of vowels :  9
Number of consonants : 14
Number of upper case : 1
Number of lower case :  22
```

Program – 10

Write a program to create a text file. Display the contents of the text file. Read lines from a text file and display number of words which are less than 4 characters

```
#creating a text file
f=open("sample.txt","w")
str=input("Enter text : ")
f.write(str)
f.close()
#displaying a text file
f=open("sample.txt","r")
ch=f.read()
print("The Contents of the file:")
print(ch)
f.close()
f=open("sample.txt","r")
l=f.read().split()
n=0
for i in l:
    if len(i)<4:
        n=n+1
print("No. of words with less than 4 characters : ",n)
```

Output

```
Enter text : Python is a simple language
The Contents of the file:
Python is a simple language
No. of words with less than 4 characters :  2
```

Program – 11

Create a binary file to store details of n employees in a binary file and display it on the screen

```
#program to read n employees employ no,name and salary
#and create a binary file and read and display the
#contents of the file on the screen
import pickle
def create():
    fw=open("employ.dat","wb")
    n=int(input("Enter n"))
    for i in range(n):
        eno=int(input("Enter Employee Number"))
        ename=input("Enter Employee Name")
        salary=int(input("Enter Salary"))
        l=[eno,ename,salary]
        pickle.dump(l,fw)
    fw.close()
def display():
    fr=open("employ.dat","rb")
    try:
        while True:
            l=pickle.load(fr)
            print(l[0],'\t',l[1],'\t',l[2])
    except EOFError:
        fr.close()
create()
display()
```

Output

```
Enter no of Employees : 3
Enter Employee no. : 101
Enter Employee name : Dhanya
Enter Salary : 45000
Enter Employee no. : 102
Enter Employee name : Rahul
Enter Salary : 87000
Enter Employee no. : 104
Enter Employee name : Anand
Enter Salary : 90000
The Contents of the File:
101     Dhanya       45000
102     Rahul        87000
104     Anand        90000
```

Program – 12

Create a binary file with book name and book number. Search for a given book number and display the name of the book. If not found, display an appropriate message

```
import pickle
#creating a binary file
def create():
    fw=open("book.dat","wb")
    n=int(input("Enter Number of books"))
    for i in range(n):
        bno=int(input("Enter Book Number"))
        bname=input("Enter Book Name")
        l=[bno,bname]
        pickle.dump(l,fw)
    fw.close()
#displaying the contents of the binary file
def display():
    import pickle
    fr=open("book.dat","rb")
    try:
        while True:
            l=pickle.load(fr)
            print(l[0],'\t',l[1])
    except EOFError:
        fr.close()

#searching the contents of the file
def search():
    fr=open("book.dat","rb")
    sbno=int(input("Enter the bookno of the book to be searched"))
    found=0
    try:
        while True:
            l=pickle.load(fr)
            if l[0]==sbno:
                print("The Book Name of Book Number ",l[0]," is ",l[1])
                found=1
                break
    except EOFError:
        if found==0:
            print("Search not found")
    fr.close()
#main program
create()
print("The contents of the file:")
display()
search()
```

<u>Output</u>

```
Enter Number of books2
Enter Book Number1001
Enter Book NamePython
Enter Book Number1002
Enter Book NameC++
The contents of the file:
1001   Python
1002   C++
Enter the bookno of the book to be searched1002
The Book Name of Book Number  1002  is  C++

Enter the bookno of the book to be searched1004
Search not found
```

Program – 13

Create a binary file with roll number, name and marks. Input a roll number and update the marks

```
import pickle
import os
#creating the file
f=open("stud.dat","wb")
n=int(input("Enter no of Students : "))
for i in range(n):
    rollno=int(input("Enter Roll no. : "))
    sname=input("Enter Student name : ")
    marks=int(input("Enter Marks : "))
    l=[rollno,sname,marks]
    pickle.dump(l,f)
f.close()
#displaying the contents of the file
print("Displaying original data")
f=open("stud.dat","rb")
print("Rollno\tName\tMark")
try:
    while  True:
        e=pickle.load(f)
        print(e[0],'\t',e[1],'\t',e[2])
except EOFError:
        f.close()
#modify the content of the file
f=open("stud.dat","rb")
f1=open("temp.dat","wb")
found=0
no=int(input("Enter the rollno of the student whose data has to be modified"))
try:
    while True:
        e=pickle.load(f)
        if e[0]==no:
            e[1]=input("Enter Name:")
            e[2]=int(input("Enter Mark:"))
            found=1
        pickle.dump(e,f1)
except EOFError:
    if found==0:
        print("REcord to be modified not present")
        f1.close()
        f.close()
    else:
        print("Record modified")
        f1.close()
        f.close()
        os.remove("stud.dat")
        os.rename("temp.dat","stud.dat")
#displaying the contents of the file after modification
print("Displaying contents of the file after modification")
f=open("stud.dat","rb")
print("Rollno\tName\tMark")
try:
    while  True:
        e=pickle.load(f)
        print(e[0],'\t',e[1],'\t',e[2])
except EOFError:
        f.close()
```

Output

```
Enter no of Students : 2
Enter Roll no. : 1
Enter Student name : Anand
Enter Marks : 99
Enter Roll no. : 2
Enter Student name : Adish
Enter Marks : 100
Displaying original data
Rollno         Name          Mark
1               Anand         99
2               Adish         100
Enter the rollno of the student whose data has to be modified1
Enter Name:Anand
Enter Mark:100
Record modified
Displaying contents of the file after modification
Rollno         Name          Mark
1              Anand          100
2              Adish          100
```

Program 14

create a binary file student.dat with contents rollno and name.Delete a record by entering the rollno.Display the contents of the file before and after deletion

```
#program to create a binary file student.dat with contents rollno and name
#Delete a record by entering the rollno.Display the contents of the file before and after deletion
def create():
    import pickle
    fw=open("student.dat","wb")
    n=int(input("Enter number of records"))
    for i in range(n):
        rollno=int(input("Enter Rollno"))
        name=input("Enter Name")
        l=[rollno,name]
        pickle.dump(l,fw)
    fw.close()
#displaying the contents of the binary file
def display():
    import pickle
    fr=open("student.dat","rb")
    try:
        while True:
            l=pickle.load(fr)
            print(l[0],'\t',l[1])
    except EOFError:
        fr.close()
#Deleting the contents from the binary file
def delete():
    import pickle
    import os
    fr=open("student.dat","rb")
    fw=open("temp","wb")
    found=0
    r=int(input("Enter the rollno of the record to be deleted"))
    try:
        while True:
            l=pickle.load(fr)
            if l[0]==r:
                    found=1
            else:
                pickle.dump(l,fw)
    except EOFError:
```

```
    fr.close()
    fw.close()
    if found==1:
        print("Record deleted")
        os.remove("student.dat")
        os.rename("temp","student.dat")
    else:
        print("Search Record not found")
#main program
create()
print("The Student Details:")
display()
delete()
print("The Student Details after Deletion:")
display()
```

Program - 15

Write a function in python, MakePush(Package) and MakePop(Package) to add a new Package and delete a Package from a List of Package Description, considering them to act as push and pop operations of the Stack data structure.

Program:

```
def Makepush(Package):
    id= input("Enter Package number: ")
    name=input("Enter Package name: ")
    p=[id,name]
    Package.append(p)

def Makepop(Package):
     if Package==[]:
        print("Empty stack")
     else:
        print("Deleted value: ",Package.pop())

def display(Package):
    n=len(Package)
    if Package==[]:
        print("Empty stack")
    else:
        for i in range(n-1,-1,-1):
            print (Package[i])

#main
Package=[]

while (True):
    print("1. PUSH")
    print("2. POP")
    print("3. DISPLAY")
    print("4. EXIT")
    choice= int(input("Enter a choice: "))
    if ( choice==1):
       Makepush(Package)
    elif (choice==2):
        Makepop(Package)
    elif choice==3:
        display(Package)
    else:
        print("End of program")
        break
```

Output:

```
1. PUSH
2. POP
3. DISPLAY
4. EXIT
Enter a choice: 1
Enter Package number: 1
Enter Package name: A
1. PUSH
2. POP
3. DISPLAY
4. EXIT
Enter a choice: 1
Enter Package number: 2
Enter Package name: B
1. PUSH
2. POP
3. DISPLAY
4. EXIT
Enter a choice: 1
Enter Package number: 3
Enter Package name: C
1. PUSH
2. POP
3. DISPLAY
4. EXIT
Enter a choice: 3
['3', 'C']
['2', 'B']
['1', 'A']
1. PUSH
2. POP
3. DISPLAY
4. EXIT
Enter a choice: 2
Deleted value:  ['3', 'C']
1. PUSH
2. POP
3. DISPLAY
4. EXIT
```

```
Enter a choice: 2
Deleted value:  ['2', 'B']
1. PUSH
2. POP
3. DISPLAY
4. EXIT
Enter a choice: 2
Deleted value:  ['1', 'A']
1. PUSH
2. POP
3. DISPLAY
4. EXIT
Enter a choice: 3
empty stack
1. PUSH
2. POP
3. DISPLAY
4. EXIT
Enter a choice: 4
End of program
```

Program 16

Create a dictionary containing names and marks as key value pairs of n students. Write a program, with separate user defined functions to perform the following operations:

- **Push the keys (name of the student) of the dictionary into a stack, where the corresponding value (marks) is greater than 75.**
- **Pop and display the content of the stack.**

Program

```
def push(student,a):
    student.append(i)

def pop(S):
 if student==[]:
   print("Student list is empty")
 else:
     print("Deleted name is..",student.pop())

def display(student):
   if student==[]:
     print("Student list is empty")
   else:
    print("The stack is...")
    n=len(student)
    for i in range(n-1,-1,-1):
        print (student[i])

student=[]
a={}

n=int(input("Enter the no of students: "))
for i in range(0,n):
    name=input("Enter name of the student: ")
    marks=int( input("Enter marks: "))
    a[name]=marks

print("The dictionary is")
print(a)
```

```
for i in a:
    if a[i]>75:
        push(student, i)
display(student)
print("After calling pop()...")
pop(student)
display(student)
```

Output

```
Enter the no of students: 4
Enter name of the student: Anoop
Enter marks: 67
Enter name of the student: Gokul
Enter marks: 90
Enter name of the student: Adithya
Enter marks: 99
Enter name of the student: Geethu
Enter marks: 74
The dictionary is
{'Anoop': 67, 'Gokul': 90, 'Adithya': 99, 'Geethu': 74}
The stack is...
Adithya
Gokul
After calling pop()...
Deleted name is.. Adithya
The stack is...
Gokul
```

Program 17

```
#program to create a CSV file with employee Number,
#employee name and salary.Display the contents of the file.
#search for the employee by entering the employee number.
import csv
#create function
def create():
    f=open("t1.csv","w",newline='')
    w=csv.writer(f)
    n=int(input(“Enter number of records”))
    for i in range(n):
        eno=input("Enter Employee No")
        ename=input("Enter Ename")
        s=int(input("Enter salary"))
        l=[eno,ename,s]
        w.writerow(l)
    f.close()

#search function
def search():
    f=open("t1.csv","r")
    l=[]
    r=csv.reader(f)
    n=input("Enter employee no to be searched")
    for i in r:
        if i[0]==n:
            print("employ no",i[0],"name",i[1],"salary",i[2])
            break
    else:
        print("Record not found")
    f.close()

#display function
def display():
    f=open("t1.csv","r")
    r=csv.reader(f)
    for i in r:
        print(i)
    f.close()
```

```
#main program
create()
display()
search()

OUTPUT
Enter Employee No1001
Enter EnameAnand
Enter salary9000
Enter do you want to continue?y
Enter Employee No1002
Enter EnameBincy
Enter salary8000
Enter do you want to continue?n
['1001', 'Anand', '9000']
['1002', 'Bincy', '8000']
Enter employee no to be searched1002
employ no 1002 name Bincy salary 8000
```

Program – 18

Write a menu driven python program to implement database connectivity with MySQL to :

- **Modify a record with new values for salary when employee number is given by the user**
- **Display all records in the table**

Program:

```
import mysql.connector as c
con=c.connect(host='localhost',user='root',passwd='npol',database='exam')
cur=con.cursor()
while True:
    print("1. Modify Salary")
    print("2. Display Employees")
    print("3. Exit")
    ch=int(input("Enter choice : "))
    if ch==1:
        no=int(input("Enter Employee Number : "))
        s=int(input("Enter new Salary : "))
        cur.execute("update emp set sal={} where empno={}".format(s,no))
        con.commit()
    elif ch==2:
        cur.execute("select * from emp")
        t=cur.fetchall()
        for i in t:
            print(i)
    else:
        break
```

Output

```
 1. Modify Salary
 2. Display Employees
 3. Exit
 Enter choice : 1
 Enter Employee Number : 1234
 Enter new Salary : 7000
 1. Modify Salary
 2. Display Employees
 3. Exit
 Enter choice : 2
 (1234, 'GEORGE', 'MANAGER', 7698, datetime.date(2013, 4, 12),
    Decimal('7000.00'), None, 20)
 (7369, 'SMITH', 'CLERK', 7902, datetime.date(2010, 12, 17), Decimal('5000.00'),
    Decimal('800.00'), 20)
 (7499, 'ALLEN', 'SALESMAN', 7698, datetime.date(2012, 2, 10), Decimal('6000.00'),
    Decimal('500.00'), 30)
 (7521, 'WARD', 'SALESMAN', 7698, datetime.date(2015, 7, 10), Decimal('4000.00'),
    Decimal('900.00'), 30)
 (7566, 'JONES', 'MANAGER', 7698, datetime.date(2010, 8, 12), Decimal('8000.00'),
    None, 20)
 (7934, 'JAMES', 'CLERK', 7698, datetime.date(2011, 12, 12), Decimal('3000.00'),
    None, 10)
 1. Modify Salary
 2. Display Employees
 3. Exit
Enter choice : 3
```

Program – 19

Write a menu driven python program to implement database connectivity with MySQL to :

- **Insert a record**
- **Display all records in the table**

Program

```
import mysql.connector as c
con=c.connect(host='localhost',user='root',passwd='npol',database='exam')
cur=con.cursor()
while True:
    print("1. Insert Employee")
    print("2. Display Employees")
    print("3. Exit")
    ch=int(input("Enter choice :"))
    if ch==1:
        query="insert into emp values(1234,'GEORGE','MANAGER',7698,'2013-04-12',6000,NULL,20)"
        cur.execute(query)
        con.commit()
        print("Record inserted")
    elif ch==2:
        cur.execute("select * from emp")
        t=cur.fetchall()
        for i in t:
            print(i)
    else:
        break
```

Output

```
  1. Insert Employee
  2. Display Employees
  3. Exit

  Enter choice : 1
  Record inserted
  1. Insert Employee
  2. Display Employees
  3. Exit
  Enter choice :  2
  (1234, 'GEORGE', 'MANAGER', 7698, datetime.date(2013, 4, 12),
     Decimal('6000.00'), None, 20)
  (7369, 'SMITH', 'CLERK', 7902, datetime.date(2010, 12, 17), Decimal('5000.00'),
     Decimal('800.00'), 20)
  (7499, 'ALLEN', 'SALESMAN', 7698, datetime.date(2012, 2, 10), Decimal('6000.00'),
     Decimal('500.00'), 30)
  (7521, 'WARD', 'SALESMAN', 7698, datetime.date(2015, 7, 10), Decimal('4000.00'),
     Decimal('900.00'), 30)
  (7566, 'JONES', 'MANAGER', 7698, datetime.date(2010, 8, 12), Decimal('8000.00'),
     None, 20)
(7934, 'JAMES', 'CLERK', 7698, datetime.date(2011, 12, 12), Decimal('3000.00'),
   None, 10)

  1. Insert Employee
  2. Display Employees
  3. Exit
Enter choice :  3
```

<u>Program-20</u>

Write a menu driven python program to implement database connectivity with MySQL to :

•Delete a record whose employee number is given by the user

•Display all records in the table

```
import mysql.connector as c
con=c.connect(host='localhost',user='root',passwd='npol',database='exam1')
cur=con.cursor()
while True:
    print("1. Delete Record")
    print("2. Display Employees")
    print("3. Exit")
    ch=int(input("Enter choice : "))
    if ch==1:
        no=int(input("Enter Employee Number : "))
        cur.execute("delete from emp where empno={}".format(no))
        con.commit()
    elif ch==2:
        cur.execute("select * from emp")
        t=cur.fetchall()
        for i in t:
            print(i)
    else:
        break
```

<u>output</u>

```
 1. Delete Record
2. Display Employees
3. Exit
Enter choice : 1
Enter Employee Number : 7369
 1. Delete Record
2. Display Employees
3. Exit
Enter choice : 2
(7499, 'ALLEN', 'SALESMAN', 7698, datetime.date(2012, 2, 10),
Decimal('345.60'), Decimal('500.00'), 30)
(7521, 'WARD', 'SALESMAN', 7698, datetime.date(2015, 7, 10),
Decimal('230.40'), Decimal('900.00'), 30)
(7566, 'JONES', 'MANAGER', 7698, datetime.date(2010, 8, 12),
Decimal('460.80'), None, 20)
```

Program-21

Write a menu driven python program to implement database connectivity with MySQL to :

• Increase the salary of all employees by 20%

• Display all records in the table

```
import mysql.connector as c
con=c.connect(host='localhost',user='root',passwd='npol',database='exam1')
cur=con.cursor()
while True:
     print("1. Modify Salary")
     print("2. Display Employees")
     print("3. Exit")
     ch=int(input("Enter choice : "))
     if ch==1:
           cur.execute("update emp set sal=sal+sal*20/100")
           con.commit()
    elif ch==2:
           cur.execute("select * from emp")
           t=cur.fetchall()
          for i in t:
               print(i)
    else:
         break
```

Output

```
 1. Modify Salary
2. Display Employees
3. Exit
Enter choice : 1
 1. Modify Salary
2. Display Employees
3. Exit
Enter choice : 2
(7369, 'SMITH', 'CLERK', 7902, datetime.date(2010, 12, 17),
Decimal('288.00'), Decimal('800.00'), 20)
(7499, 'ALLEN', 'SALESMAN', 7698, datetime.date(2012, 2, 10),
Decimal('345.60'), Decimal('500.00'), 30)
(7521, 'WARD', 'SALESMAN', 7698, datetime.date(2015, 7, 10),
Decimal('230.40'), Decimal('900.00'), 30)
(7566, 'JONES', 'MANAGER', 7698, datetime.date(2010, 8, 12),
Decimal('460.80'), None, 20)
```