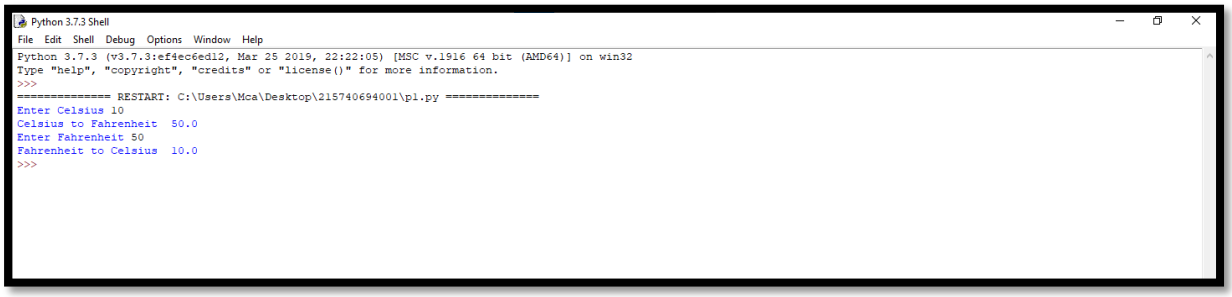
Practical:-1

1. Write a Python Program to Convert Celsius to Fahrenheit and vice –a-versa. 14/02/22.

```
c=int(input("Enter Celsius "))

f=(9/5)*c+32

print("Celsius to Fahrenheit ",f)


f1=int(input("Enter Fahrenheit "))

c=(5/9)*(f1-32)

print("Fahrenheit to Celsius ",c)
```

Output:-



Practical:-2

2 Write a program in python to swap two variables without using temporary variable. . 14/2/22

```
a=int(input("Enter a "))
b=int(input("Enter b "))
print("--------------------")
print("Before swap")
print("--------------------")
print("A is ",a)
print("B is ",b)
print("--------------------")
a,b=b,a
print("After swap")
print("--------------------")
print("A is ",a)
print("B is ",b)
```

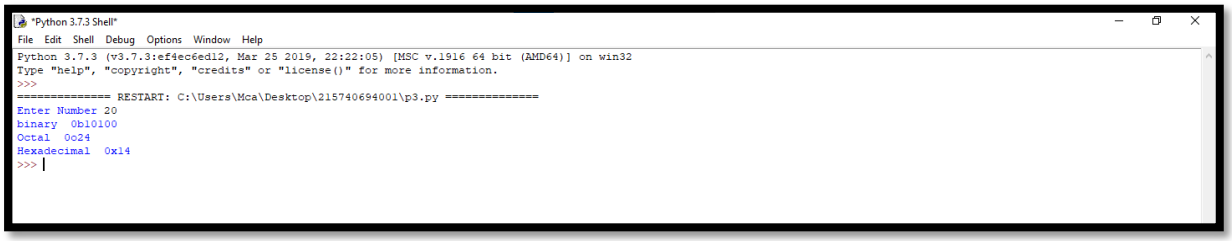
Output:-

Practical:-3

3. Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal. 14-2-22

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

print("binary ",bin(a))

print("Octal ",oct(a))

print("Hexadecimal ",hex(a))
```

Output:-



Practical:-4

4 Write a program to make a simple calculator. 14/2/22

4.1

```
a=int(input("Enter a "))
b=int(input("Enter b "))
c=a+b
print("sum is",c)
c=a-b
print("sub is",c)
c=a*b
print("mulis",c)
c=a/b
print("div is",c)
```

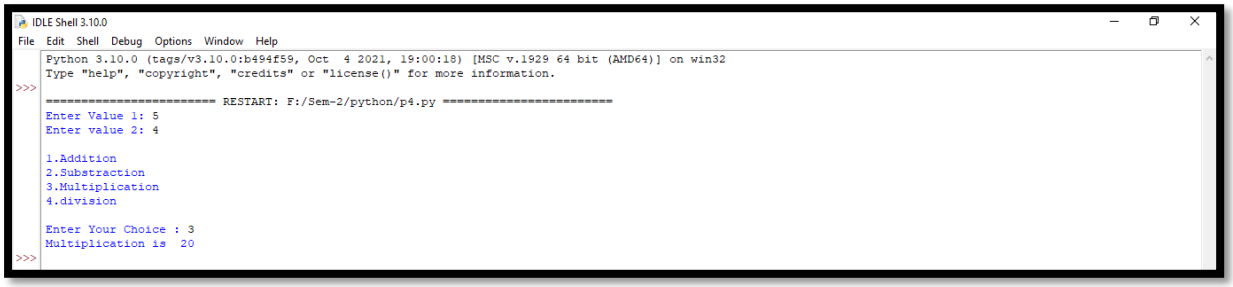
Output:-

4.2

```
a=int(input("Enter Value 1: "))
b=int(input("Enter value 2: "))
print("")
print("1.Addition")
print("2.Substraction")
print("3.Multiplication")
print("4.division")
print(" ")
n=int(input("Enter Your Choice : "))
```

```
if n==1:
    print("addition is ",a+b)
elif n==2:
    print("substraction is ",a-b)
elif n==3:
    print("Multiplication is ",a*b)
elif n==4:
    print("Division is ",a/b)
else:
    print("invalid Choice..!")
```

Output:-



4.3

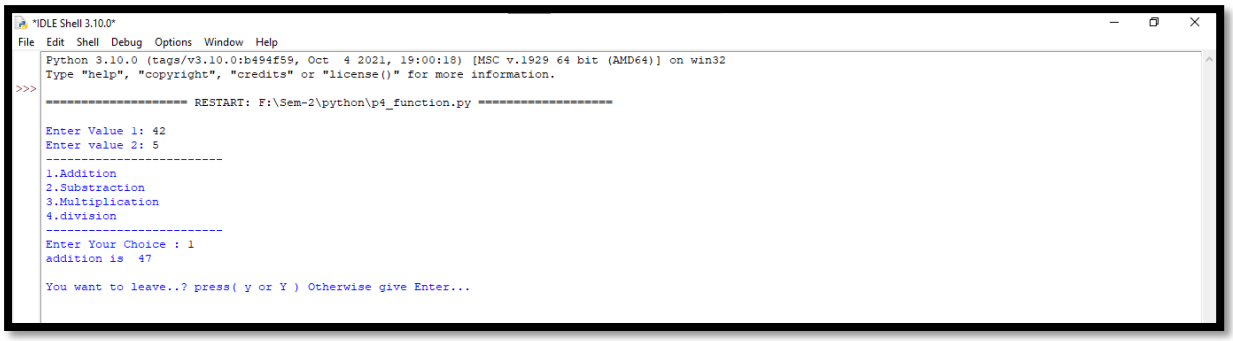
```
def add(a,b):
    return a+b
def sub(a,b):
    return a-b
def mul(a,b):
    return a*b
def div(a,b):
```

```
    return a//b


while("true"):
    print(" ")
    a=int(input("Enter Value 1: "))
    b=int(input("Enter value 2: "))
    print("-------------------------")
    print("1.Addition")
    print("2.Substraction")
    print("3.Multiplication")
    print("4.division")
    print("-------------------------")
    n=int(input("Enter Your Choice : "))
    if n==1:
        print("addition is ",add(a,b))
elif n==2:
        print("substraction is ",sub(a,b))
elif n==3:
        print("Multiplication is ",mul(a,b))
elif n==4:
        print("Division is ",div(a,b))
    else:
        print("invalid Choice..!")
    print(" ")
    y=input("You want to leave..? press( y or Y ) Otherwise give Enter... ")
    if y=='y' or y=='Y':
        break
```

else:

continue

Output:-

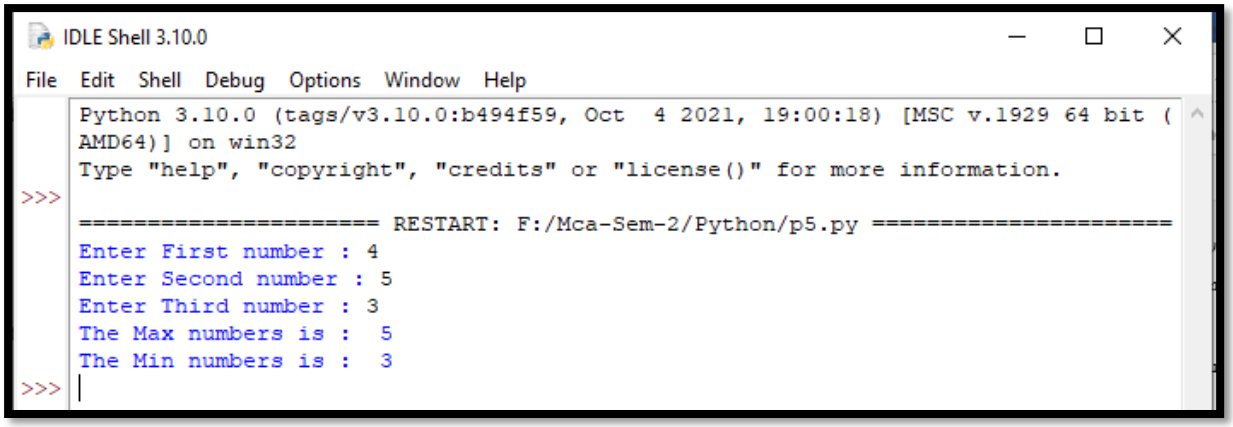


Practical:-5

5 Write a program in python to find out maximum and minimum number out of three user entered number.

```
def maxn(num1, num2, num3):
    if (num1 > num2) and (num1 > num3):
        maxn=num1
    elif (num2 > num1) and (num2 > num3):
        maxn=num2
    else:
        maxn = num3
    print("The Max numbers is : ", maxn)
def minn(num1, num2, num3):
    if (num1 < num2) and (num1 < num3):
        minn = num1
    elif (num2 < num1) and (num2 < num3):
        minn=num2
    else:
        minn=num3
    print("The Min numbers is : ", minn)


number1 = int(input('Enter First number : '))
number2 = int(input('Enter Second number : '))
number3 = int(input('Enter Third number : '))
maxn(number1, number2, number3)
minn(number1, number2, number3)
```

Output:-

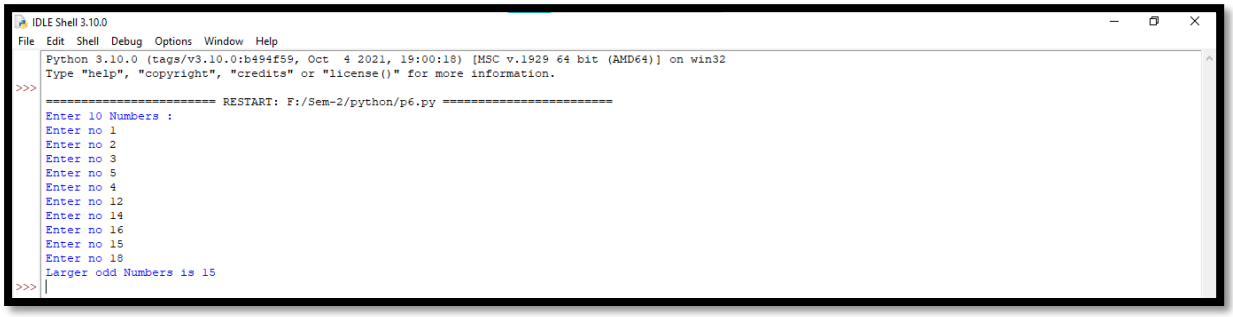

IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: F:/Mca-Sem-2/Python/p5.py ==========
Enter First number : 4
Enter Second number : 5
Enter Third number : 3
The Max numbers is : 5
The Min numbers is : 3
>>>

Practical:-6

6 Write a program which will allow user to enter 10 numbers and display largest odd number from them. It will display appropriate message in case if no odd number is found. 14/03/2022

```
print("Enter 10 Numbers : ")
modd=0
n=0
while n<10:
i=int(input("Enter no "))
    if i%2!=0:
        if i>modd:
modd = i
    n=n+1
print("Larger odd Numbers is",modd)
```

Output:-

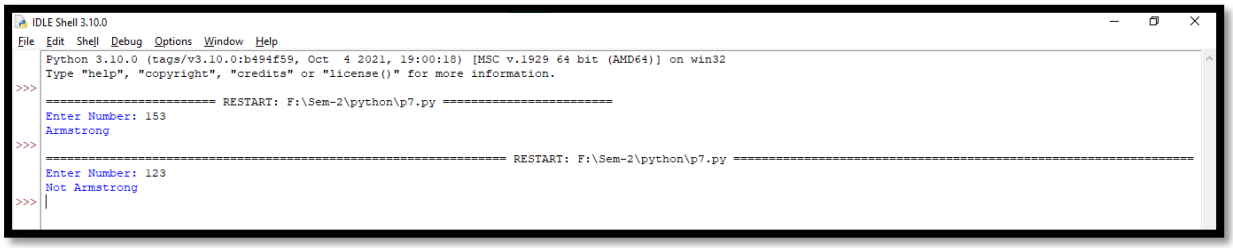


Practical:-7

7. Write a Python program to check if the number provided by the user is an Armstrong number.15/2/22

```
n=int(input("Enter Number: "))
temp=n
s=0
while n>0:
    d=n%10;
    s=s+d*d*d
    n=n//10
if temp==s:
    print("Armstrong")
else:
    print("Not Armstrong")
```

Output:-

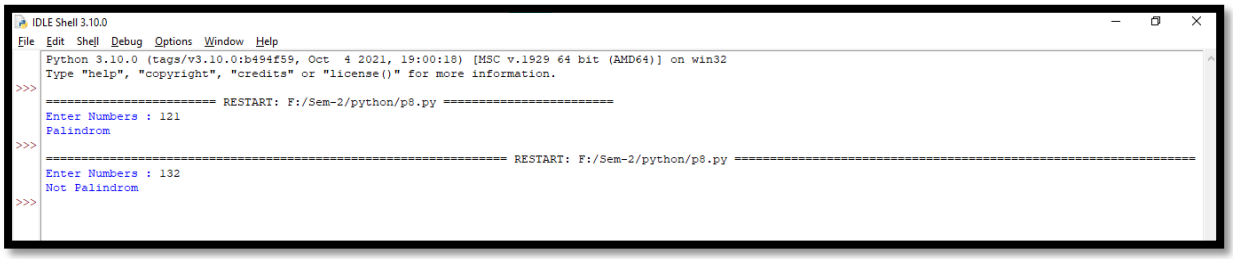


Practical:-8

8. Write a Python program to check if the number provided by the user is a palindrome or not. 15/2/22.

```
n=int(input("Enter Numbers : "))
temp=n
rev=0
while n>0:
    d=n%10
    rev=(rev*10)+d
    n=n//10
if temp==rev:
    print("Palindrom")
else:
    print("Not Palindrom")
```

Output:-



Practical:-9

9.Write a Python program to perform following operation on given string input:

a) Count Number of Vowel in given string

b) Count Length of string (do not use Len ())

c) Reverse string

d) Find and replace operation

e) check whether string entered is a palindrome or not . 1/3/22

```
def countVowel():
    p=input("Enter String ")
    a=e=i=o=u=c=0
for k in p:
        if k=='a':
            a=a+1


        if k=='e':
            e=e+1


        if k=='i':
i=i+1


        if k=='o':
            o=o+1


        if k=='u':
```

```
            u=u+1


        if k=='a' or k=='e' or k=='i' or k=='o' or k=='u':
            c=c+1


    print("A =",a)
    print("E =",e)
    print("I =",i)
    print("O =",o)
    print("U =",u)
    print("total vowels =",c)


def countLength():
i=input("Enter string ")
    s=0
    for p in i:
        s=s+1
    print("Length is ",s)


def reverseString():
i=input("Enter string ")
    print("Reverse string ",i[::-1])


def replaceString():
i=input("Enter string ")
    f=input("Enter Find String ")
    r=input("Enter Replace String ")
```

```
    print("Original string",i)
    print(i.replace(f,r))
def checkStringPaliorNot():
i=input("Enter string ")
    if i==i[::-1]:
        print("string is a palindrome")
    else:
        print("string is not a palindrome")


while("true"):
    print("a. Count Number of Vowel in given string")
    print("b. Count Length of string")
    print("c. Reverse string ")
    print("d. Find and replace operation")
    print("e. check whether string entered is a palindrome or not")
    print("Enter choice :")
    c=input()
    cl=c.lower()


    if cl=='a':
countVowel()
elif cl=='b':
countLength()
elif cl=='c':
reverseString()
elif cl=='d':
replaceString()
```

```
elif cl=='e':
checkStringPaliorNot()
    else:
        print("Invalid Choice")


    print("---------------------------------")
ch=input("Do you Want To Exit press(Y or y) ")
    print("---------------------------------")
    if ch=='Y' or ch=='y':
        break
```

Output:-

```
========== RESTART: F:\Mca-Sem-2\Python\p9.py ==========
a. Count Number of Vowel in given string
b. Count Length of string
c. Reverse string
d. Find and replace operation
e. check whether string entered is a palindrome or not
Enter choice :
c
Enter string keyur halpati
Reverse string  itaplah ruyek
----------------------------------
Do you Want To Exit press(Y or y)
----------------------------------
a. Count Number of Vowel in given string
b. Count Length of string
c. Reverse string
d. Find and replace operation
e. check whether string entered is a palindrome or not
Enter choice :
d
Enter string keyur halpati
Enter Find String keyur
Enter Replace String mayuri
Original String keyur halpati
Replaced String mayuri halpati
----------------------------------
Do you Want To Exit press(Y or y)
----------------------------------
a. Count Number of Vowel in given string
b. Count Length of string
c. Reverse string
d. Find and replace operation
e. check whether string entered is a palindrome or not
Enter choice :
e
Enter string kek
string is a palindrome
----------------------------------
Do you Want To Exit press(Y or y)
```

Practical:-10

10 Define a procedure histogram () that takes a list of integers and prints a histogram to the screen. For example, histogram ([4, 9, 7]) should print the following:

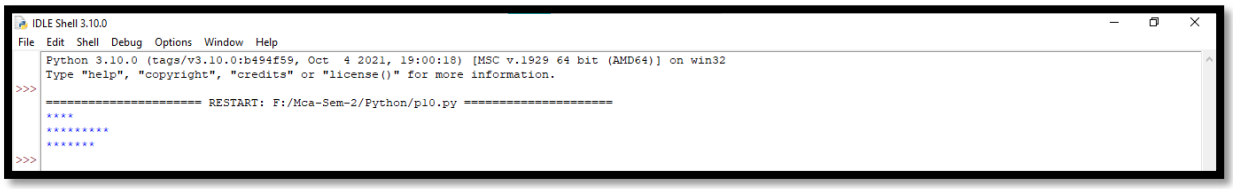
****

****

********* 21/03/2022**

```
def histogram( items ):
    for n in items:
        output = ''
        t = n
        while( t > 0 ):
          output += '*'
          t = t - 1
        print(output)

histogram([4,9,7])
```

Output:-

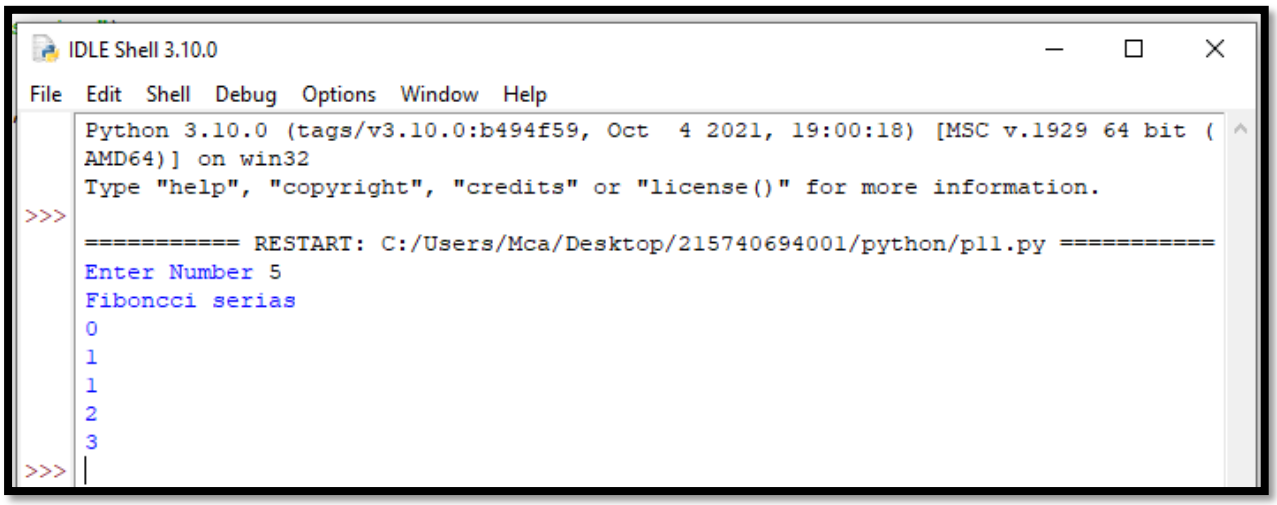


Practical:-11

#11 Write a program in python to implement Fibonacci series up to user entered number.(Use recursive Function) 28/03/2022

```
//normal

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

a=0

b=1

print("Fibonacci serias")

print(a)

print(b)

for i in range(0,n-2):

    c=a+b

    print(c)

    a=b

    b=c
```

Output:-



```
//  Recursive
def fibrec(n):
    if n==0:
        return n
    elif n==1:
        return n
    else:
        return (fibrec(n-1) + fibrec(n-2))

n=int(input("Enter Number "))
print("Fibonccci serias")
for f in range(0,n):
    print(fibrec(f))
```

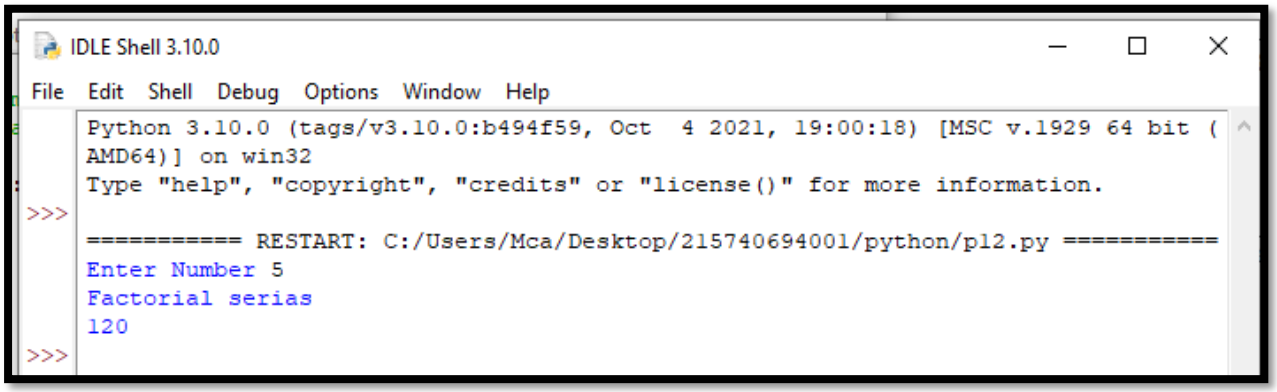
Output:-

Practical:-12

#12 Write a program in python to implement Factorial series up to user entered number. (Use recursive Function) 28/03/2022.

```
//normal

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

print("Factorial serias")

fact=1

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

    fact=fact*i

print(fact)
```

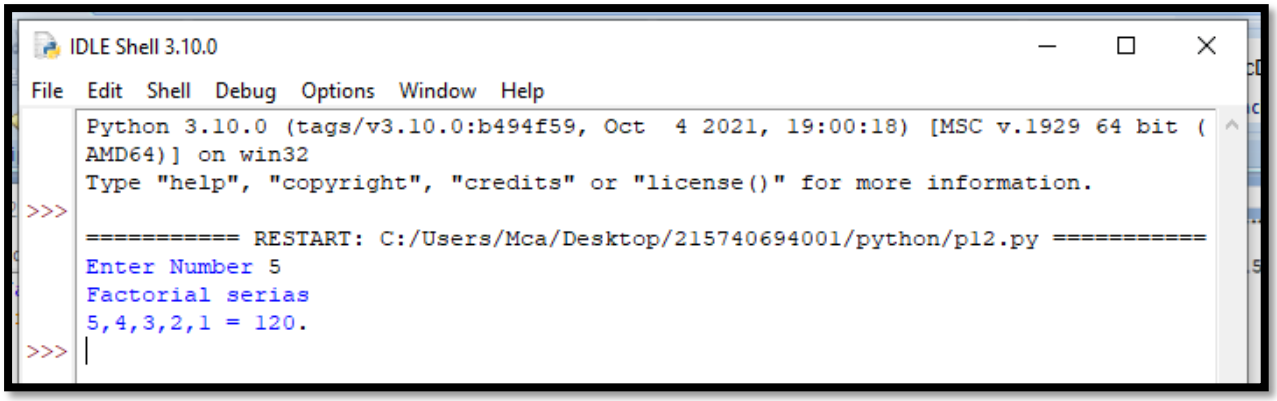
Output:-




```
//recursive

def factrec(n):

    if n==1:

        print(n,end=" = ")

        return n

    else:

        print(n,end=",")

        return (n * factrec(n-1))
```

```
n=int(input("Enter Number "))

print("Factorial serias")

print(factrec(n),end="")
```

Output:-



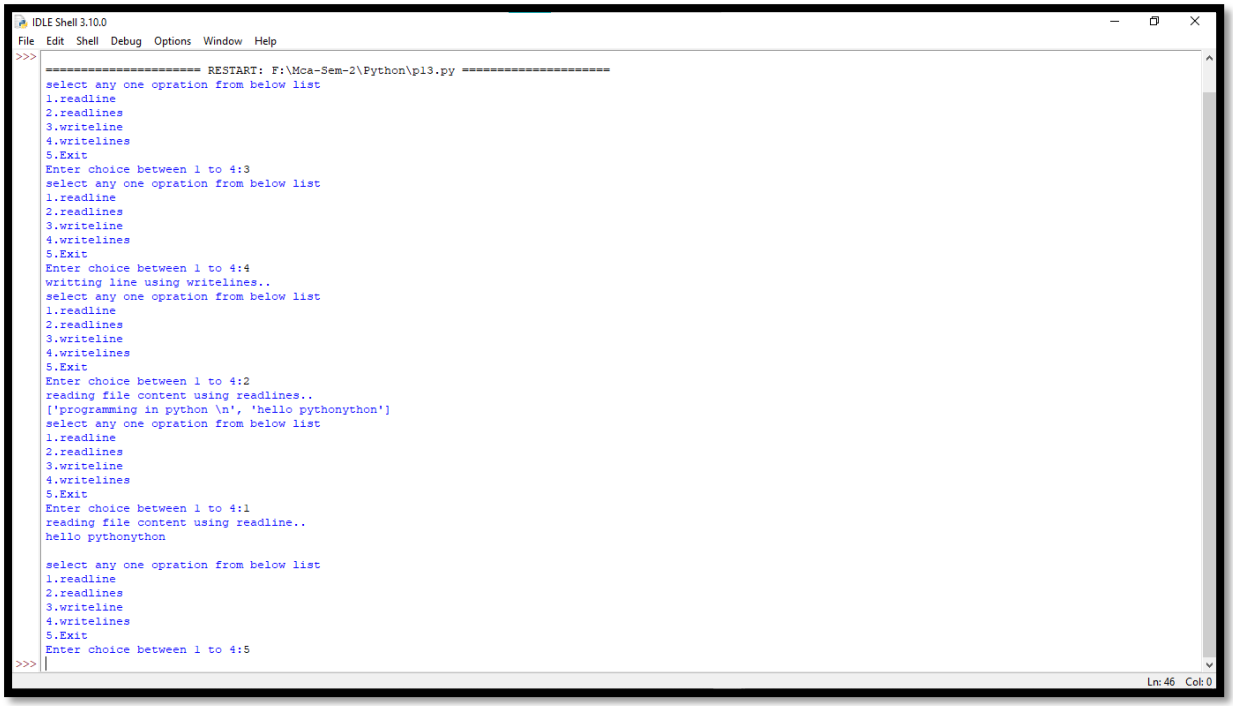
Practical:-13

#13 Write a program in Python to implement readline, readlines, write line and writelines file handling mechanisms.

```
while("True"):
    f=open('F:\Mca-Sem-2\Python\Files\data.txt','r+')
    print("select any one opration from below list")
    print("1.readline")
    print("2.readlines")
    print("3.writeline")
    print("4.writelines")
    print("5.Exit")
    choice = input("Enter choice between 1 to 4:")
    if choice == '1':
        print("reading file content using readline..")
        ln = f.readline()
        while ln != "":
            ln = f.readline()
            print(ln)

    elif choice == '2':
        print("reading file content using readlines..")
        print(f.readlines())

    elif choice == '3':
        f.write("file opration \n in programming in python")
```

```
    elif choice == '4':
        print("writting line using writelines..")
        f.writelines(["programming in ", "python"," \nhello"," python"])
    elif choice == '5':
        break;

f.close()
```

Output:-



Practical:-14

14. Write a program in python to implement Salary printing file read operation. (File format: Employee No, name, deptno, basic, DA, HRA, Conveyance) should perform below operations.

a) Print Salary Slip for given Employee Number

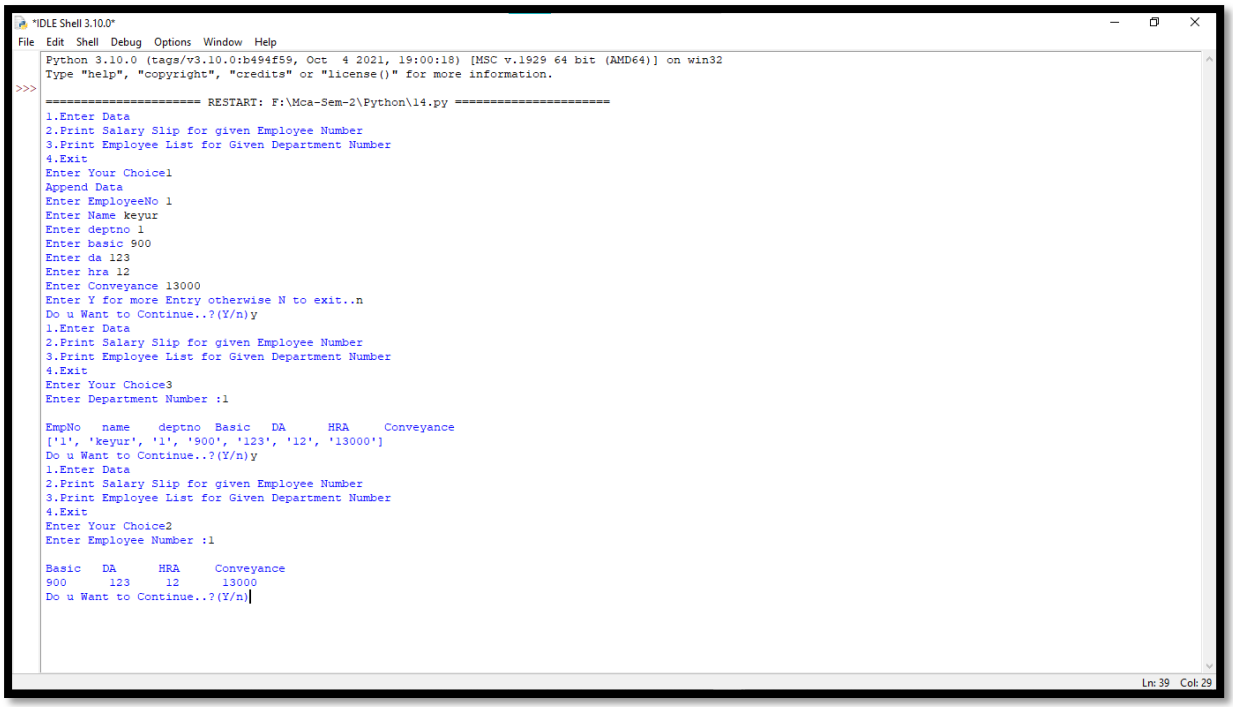
b) Print Employee List for Given Department Number.

```
import pickle
while("true"):
    print("1.Enter Data")
    print("2.Print Salary Slip for given Employee Number")
    print("3.Print Employee List for Given Department Number")
    print("4.Print All Employee")
    ch=int(input("Enter Your Choice"))
    if ch==1:
        f = open("F:\\Mca-Sem-2\\Python\\Files\\employee.txt","ab")
        data = []
        while("true"):
            print("Append Data")
            no=input("Enter EmployeeNo ")
            name=input("Enter Name ")
            deptno=input("Enter deptno ")
            basic=input("Enter basic ")
            da=input("Enter da ")
            hra=input("Enter hra ")
            Con=input("Enter Conveyance ")
            #list1=["\n"+no+"\t",name+"\t",deptno+"\t",basic+"\t",da+"\t",hra+"\t",Con+"\t"]
```

```
            list1=[no,name,deptno,basic,da,hra,Con+"\n"]
            data.append(list1)
            ch=input("Enter Y for more Entry otherwise N to exit..")
            if ch=='N' or ch=='n':
                break;
        pickle.dump(data,f)
        f.close()
    elif ch==2:
        f = open("F:\\Mca-Sem-2\\Python\\Files\\employee.txt","rb")
        empno=input("Enter Employee Number :")
        r=pickle.load(f)
        f=0
        print()
        print("Basic"+"\t"+"DA"+"\t"+"HRA"+"\t"+"Conveyance")
        for i in r:
            if i[0]==empno:
                print(i[3]+"\t",i[4]+"\t",i[5]+"\t",i[6])
                f=1
                break
        if f==0:
            print("Not")


    elif ch==3:
        f = open("F:\\Mca-Sem-2\\Python\\Files\\employee.txt","rb")
        deptno=input("Enter Department Number :")
        r=pickle.load(f)
        f=0
```

```
        print()

print("EmpNo"+"\t"+"name"+"\t"+"deptno"+"\t"+"Basic"+"\t"+"DA"+"\t"+"HRA"+"\t"+"Co
nveyance")
        for i in r:
            if i[2]==deptno:
                print(i)
                f=1
                break
        if f==0:
            print("Not")
    elif ch==4:
        f = open("F:\\Mca-Sem-2\\Python\\Files\\employee.txt","rb")
        r=pickle.load(f)
        f=0
        print()

print("EmpNo"+"\t"+"name"+"\t"+"deptno"+"\t"+"Basic"+"\t"+"DA"+"\t"+"HRA"+"\t"+"Co
nveyance")
        for i in r:
            print(i)
    else:
        print("Invalid Choices")
    ch=input("Do u Want to Continue..?(Y/n)")
    if ch=='N' or ch=='n':
        break
```

Output:-


IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: F:\Mca-Sem-2\Python\14.py ==========
1.Enter Data
2.Print Salary Slip for given Employee Number
3.Print Employee List for Given Department Number
4.Exit
Enter Your Choice1
Append Data
Enter EmployeeNo 1
Enter Name keyur
Enter deptno 1
Enter basic 900
Enter da 123
Enter hra 12
Enter Conveyance 13000
Enter Y for more Entry otherwise N to exit..n
Do u Want to Continue..?(Y/n)y
1.Enter Data
2.Print Salary Slip for given Employee Number
3.Print Employee List for Given Department Number
4.Exit
Enter Your Choice3
Enter Department Number :1
EmpNo name deptno Basic DA HRA Conveyance
['1', 'keyur', '1', '900', '123', '12', '13000']
Do u Want to Continue..?(Y/n)y
1.Enter Data
2.Print Salary Slip for given Employee Number
3.Print Employee List for Given Department Number
4.Exit
Enter Your Choice2
Enter Employee Number :1
Basic DA HRA Conveyance
900 123 12 13000
Do u Want to Continue..?(Y/n)
Ln: 39 Col: 29

Practical:-15

15 Write a program in python to implement Railway Reservation System using file handling technique. System should perform below operations.

a. Reserve a ticket for a passenger.

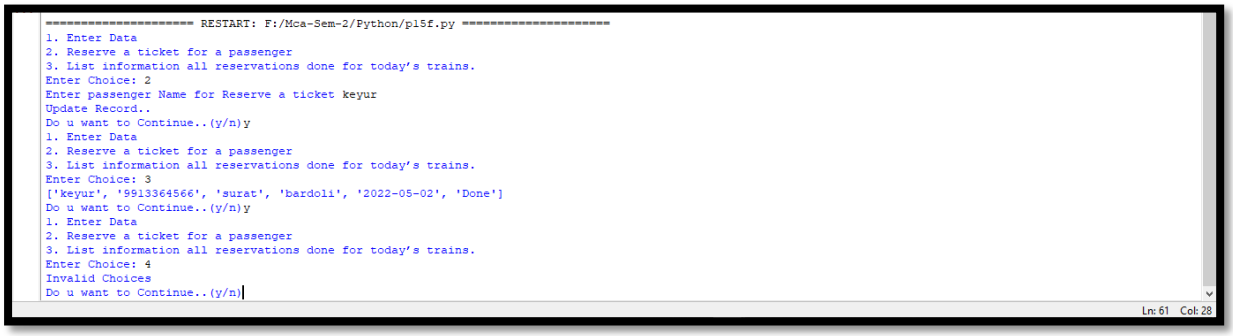
b. List information all reservations done for today's trains.

```
import pickle
from datetime import date
today=str(date.today())
while("true"):
    print("1. Enter Data")
    print("2. Reserve a ticket for a passenger")
    print("3. List information all reservations done for today's trains.")
    n=input("Enter Choice: ")
    if n=='1':
        f = open("F:\\Mca-Sem-2\\Python\\Files\\Railway_Reservation_System.txt","ab+")
        data = []
        while("true"):
            print("Append Data")
            name=input("Enter Name ")
            no=input("Enter phone ")
            s1=input("Enter Source ")
            s2=input("Enter Destination ")
            ticket=input("Enter Ticket ")
            list1=[name,no,s1,s2,today,ticket]
            data.append(list1)
```

```
        ch=input("Enter Y for more Entry otherwise N to exit..")
        if ch=='N' or ch=='n':
            break;
    pickle.dump(data,f)
    f.close()
elif n=='2':
    book=input("Enter passenger Name for Reserve a ticket ")
    f = open("F:\\Mca-Sem-2\\Python\\Files\\Railway_Reservation_System.txt","rb+")
    r = pickle.load(f)
    flag=0
    l1=[]
    for i in r:
        if i[0]==book:
            flag=1
            i[5]="Done"
            l1.append(i)
    if flag==1:
        f.seek(0)
        pickle.dump(l1,f)
        print("Update Record..")
    else:
        print("Not Update..")
    f.close()
elif n=='3':
    f = open("F:\\Mca-Sem-2\\Python\\Files\\Railway_Reservation_System.txt","rb+")
    r = pickle.load(f)
    for i in r:
```

```
            if i[4]=='2022-05-02':
                print(i)
    else:
        print("Invalid Choices")
    ch=input("Do u want to Continue..(y/n)")
    if ch=='N' or ch=='n':
        break;x
```

Output:-



Practical:-16

16. Write a Python program to implement module.

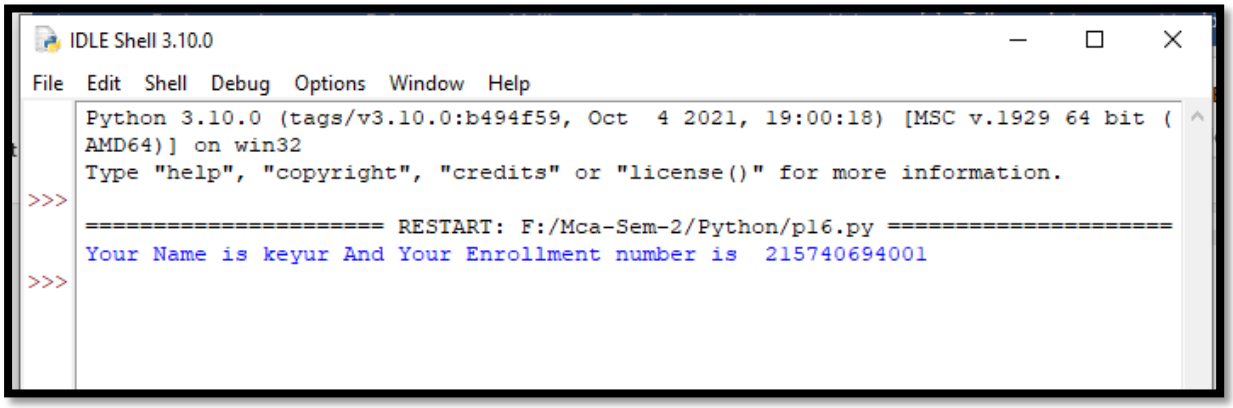
Module.py

```
def name(n):
    return "Your Name is "+n
tbl_student = {"sid":"1","st_enroll":"215740694001","st_code":"21mca01"}
```

UseModule.py

```
import module as m
p=m.name("keyur")
r=m.tbl_student["st_enroll"]
print(p,"And Your Enrollment number is ",r)
```

Output:-

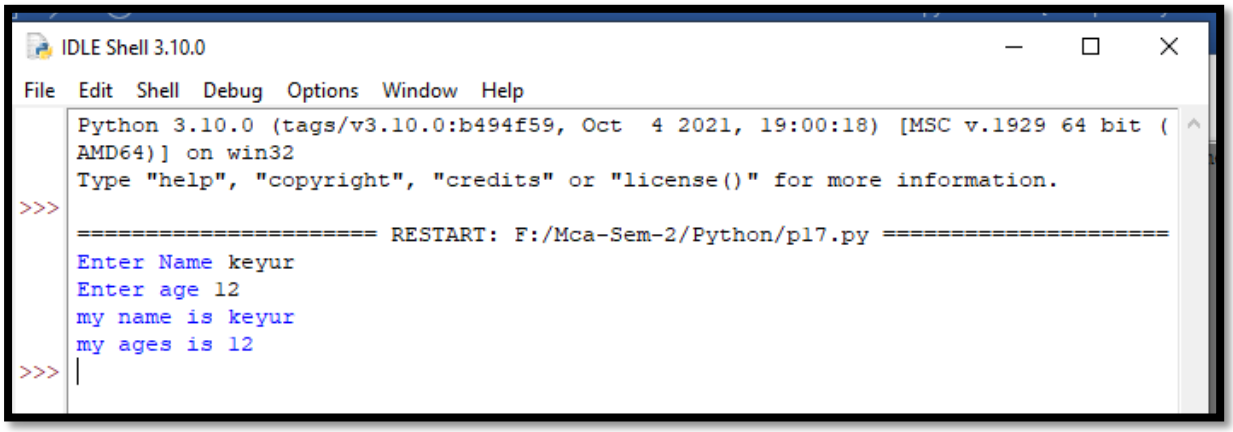


Practical:-17

17. Write a program which will implement decorators for functions and methods in python.

```
def deco(identity):
    def inner():
        a,b=identity()
        return "my name is "+a+"\n"+"my ages is "+b
    return inner
@deco
def demo():
    n=input("Enter Name ")
    age=input("Enter age ")
    return n,age
print(demo())
```

Output:-

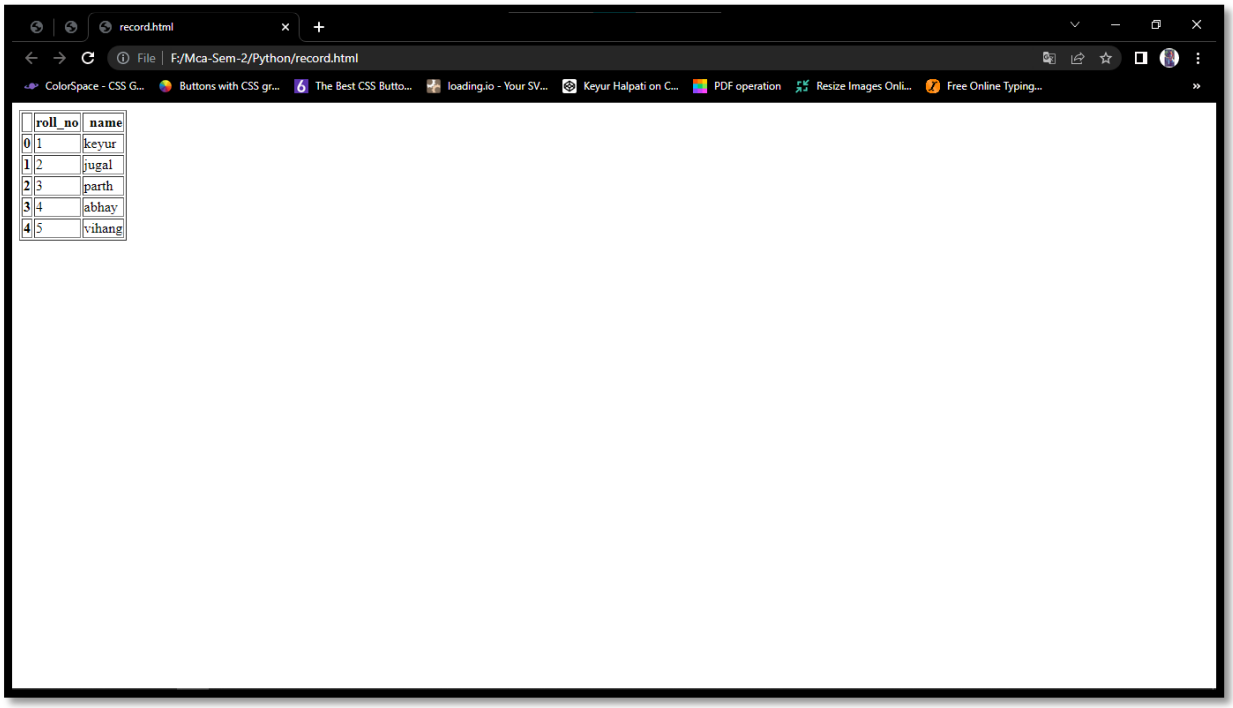


Practical:-18

18. Write a program to read CSV file and generate output using HTML table.

```
import pandas as p

d=p.read_csv("myfile_csv.csv")

d.to_html("record.html")
```

Output:-

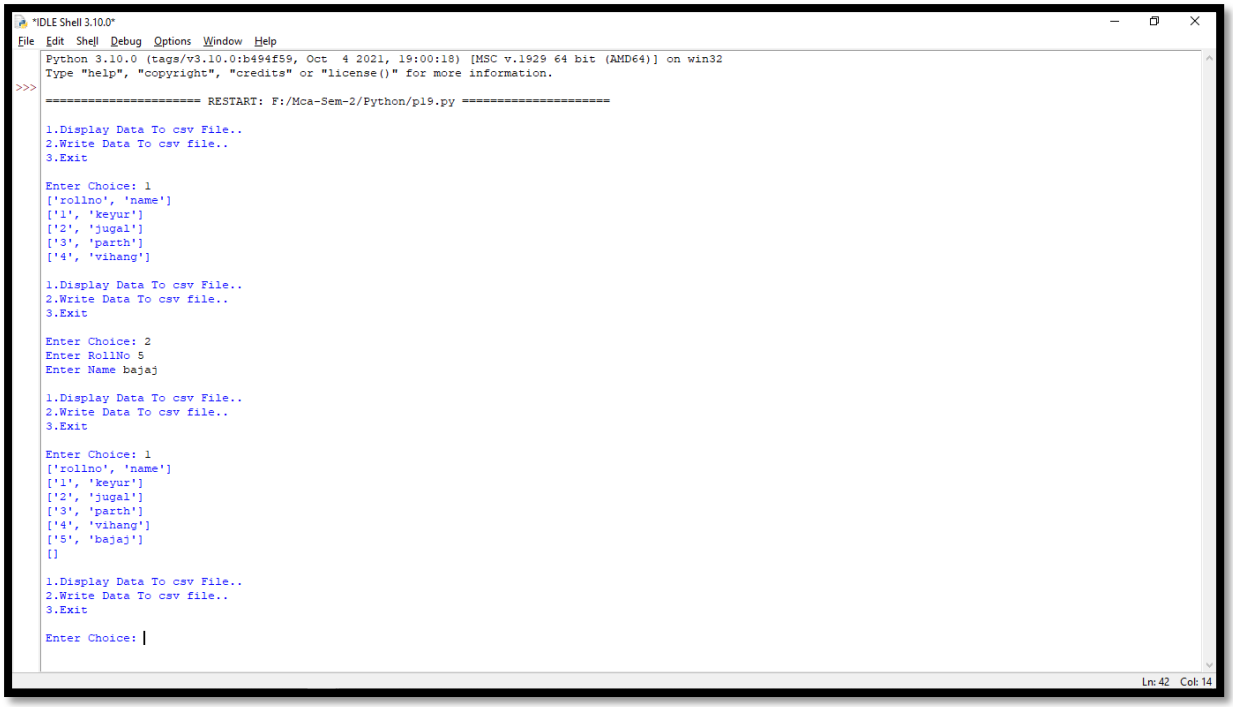


Practical:-19

19. Write a program to process CSV file using CSV module.

```
import csv as c
n=0
while(n!=3):
    print("")
    print("1.Display Data To csv File..")
    print("2.Write Data To csv file..")
    print("3.Exit")
    print("")
    n=int(input("Enter Choice: "))
    if n==1:
        with open('myfile_csv.csv','r') as f:
            r=c.reader(f)
            for i in r:
                print(i)
        f.close()
    elif n==2:

        role=input("Enter RollNo ")
        name=input("Enter Name ")
        list=[role,name]
        with open('myfile_csv.csv','a') as f:
            w=c.writer(f)
            w.writerow(list)
        f.close()
```

Output:-

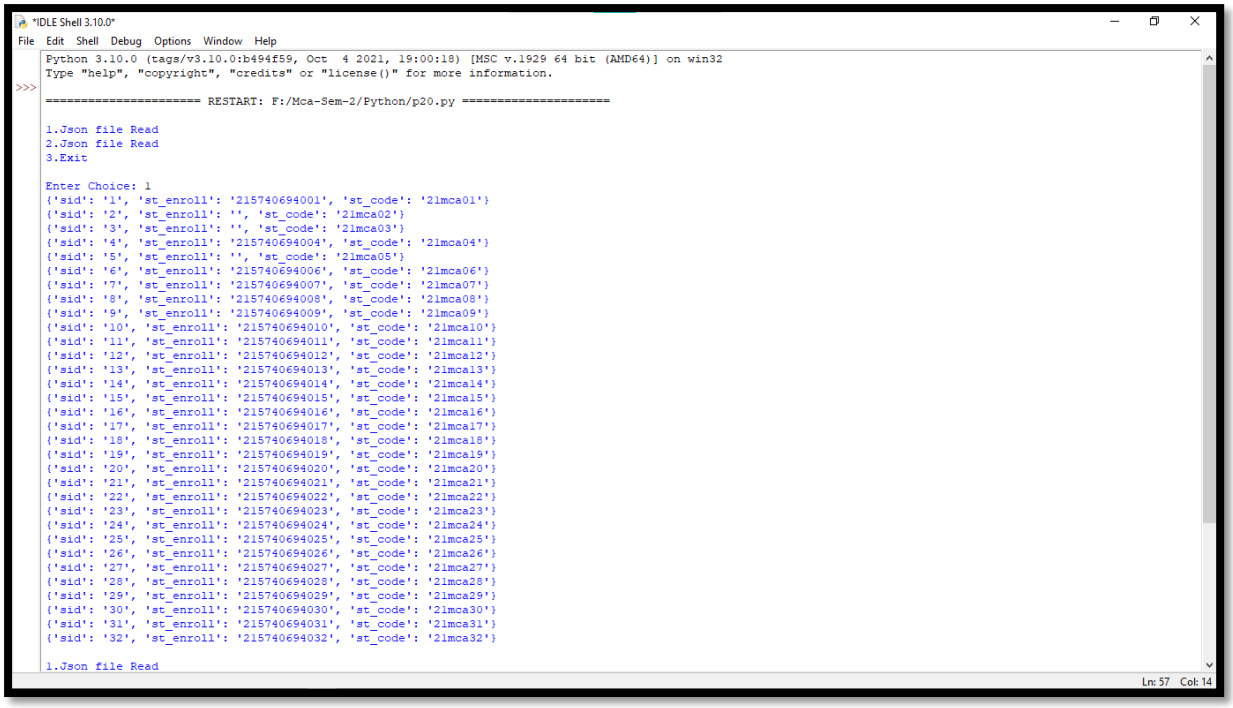

IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: F:/Mca-Sem-2/Python/p19.py ==========
1.Display Data To csv File..
2.Write Data To csv file..
3.Exit
Enter Choice: 1
['rollno', 'name']
['1', 'keyur']
['2', 'jugal']
['3', 'parth']
['4', 'vihang']
1.Display Data To csv File..
2.Write Data To csv file..
3.Exit
Enter Choice: 2
Enter RollNo 5
Enter Name bajaj
1.Display Data To csv File..
2.Write Data To csv file..
3.Exit
Enter Choice: 1
['rollno', 'name']
['1', 'keyur']
['2', 'jugal']
['3', 'parth']
['4', 'vihang']
['5', 'bajaj']
[]
1.Display Data To csv File..
2.Write Data To csv file..
3.Exit
Enter Choice:
Ln: 42 Col: 14

Practical:-20

20. Desirable: Write a program to process JSON and XML data.

```
import json as j
import xml.etree.ElementTree as x
n=0
while(n!=3):
    print("")
    print("1.Json file Read")
    print("2.Json file Read")
    print("3.Exit")
    print("")
    n=int(input("Enter Choice: "))
    if n==1:
        with open('data.json','r') as f:
            data=j.load(f)
            for i in data['tbl_student']:
                print(i)
        f.close()
    elif n==2:
        data=x.parse('data.xml')
        root=data.getroot()
        for i in root.findall('country'):
            year=i.find('year').text
            name=i.get('name')
            print(name,"->",year)
```

Output:-


IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: F:/Mca-Sem-2/Python/p20.py ==========
1.Json file Read
2.Json file Read
3.Exit
Enter Choice: 1
{'sid': '1', 'st_enroll': '215740694001', 'st_code': '21mca01'}
{'sid': '2', 'st_enroll': '', 'st_code': '21mca02'}
{'sid': '3', 'st_enroll': '', 'st_code': '21mca03'}
{'sid': '4', 'st_enroll': '215740694004', 'st_code': '21mca04'}
{'sid': '5', 'st_enroll': '', 'st_code': '21mca05'}
{'sid': '6', 'st_enroll': '215740694006', 'st_code': '21mca06'}
{'sid': '7', 'st_enroll': '215740694007', 'st_code': '21mca07'}
{'sid': '8', 'st_enroll': '215740694008', 'st_code': '21mca08'}
{'sid': '9', 'st_enroll': '215740694009', 'st_code': '21mca09'}
{'sid': '10', 'st_enroll': '215740694010', 'st_code': '21mca10'}
{'sid': '11', 'st_enroll': '215740694011', 'st_code': '21mca11'}
{'sid': '12', 'st_enroll': '215740694012', 'st_code': '21mca12'}
{'sid': '13', 'st_enroll': '215740694013', 'st_code': '21mca13'}
{'sid': '14', 'st_enroll': '215740694014', 'st_code': '21mca14'}
{'sid': '15', 'st_enroll': '215740694015', 'st_code': '21mca15'}
{'sid': '16', 'st_enroll': '215740694016', 'st_code': '21mca16'}
{'sid': '17', 'st_enroll': '215740694017', 'st_code': '21mca17'}
{'sid': '18', 'st_enroll': '215740694018', 'st_code': '21mca18'}
{'sid': '19', 'st_enroll': '215740694019', 'st_code': '21mca19'}
{'sid': '20', 'st_enroll': '215740694020', 'st_code': '21mca20'}
{'sid': '21', 'st_enroll': '215740694021', 'st_code': '21mca21'}
{'sid': '22', 'st_enroll': '215740694022', 'st_code': '21mca22'}
{'sid': '23', 'st_enroll': '215740694023', 'st_code': '21mca23'}
{'sid': '24', 'st_enroll': '215740694024', 'st_code': '21mca24'}
{'sid': '25', 'st_enroll': '215740694025', 'st_code': '21mca25'}
{'sid': '26', 'st_enroll': '215740694026', 'st_code': '21mca26'}
{'sid': '27', 'st_enroll': '215740694027', 'st_code': '21mca27'}
{'sid': '28', 'st_enroll': '215740694028', 'st_code': '21mca28'}
{'sid': '29', 'st_enroll': '215740694029', 'st_code': '21mca29'}
{'sid': '30', 'st_enroll': '215740694030', 'st_code': '21mca30'}
{'sid': '31', 'st_enroll': '215740694031', 'st_code': '21mca31'}
{'sid': '32', 'st_enroll': '215740694032', 'st_code': '21mca32'}
1.Json file Read
Ln: 57 Col: 14


1.Json file Read
2.Json file Read
3.Exit
Enter Choice: 2
Liechtenstein -> 2008
Singapore -> 2011
Panama -> 2011
1.Json file Read
2.Json file Read
3.Exit
Enter Choice:
Ln: 57 Col: 14

Practical:-21

21.Create Web Database Application “Address Book” with options to

a) add/ insert a record

b) modify a record

c) display a record

d) delete a record. 6-6-2022

```
import connection as con
def insert():
    bname=input("Enter BookName ")
    bpage=int(input("Enter Pages "))
    q=con.mycon.cursor()
    sql="INSERT INTO tbl_book(bname,bpage) VALUES (%s,%s)"
    val=(bname,bpage)
    try:
        v=q.execute(sql,val)
        con.mycon.commit()
        print("Data Inserted...")
    except:
        con.mycon.rollback()
        print("Data Not Inserted...")


def update():
    q=con.mycon.cursor()
    fetch="SELECT * FROM tbl_book"
```

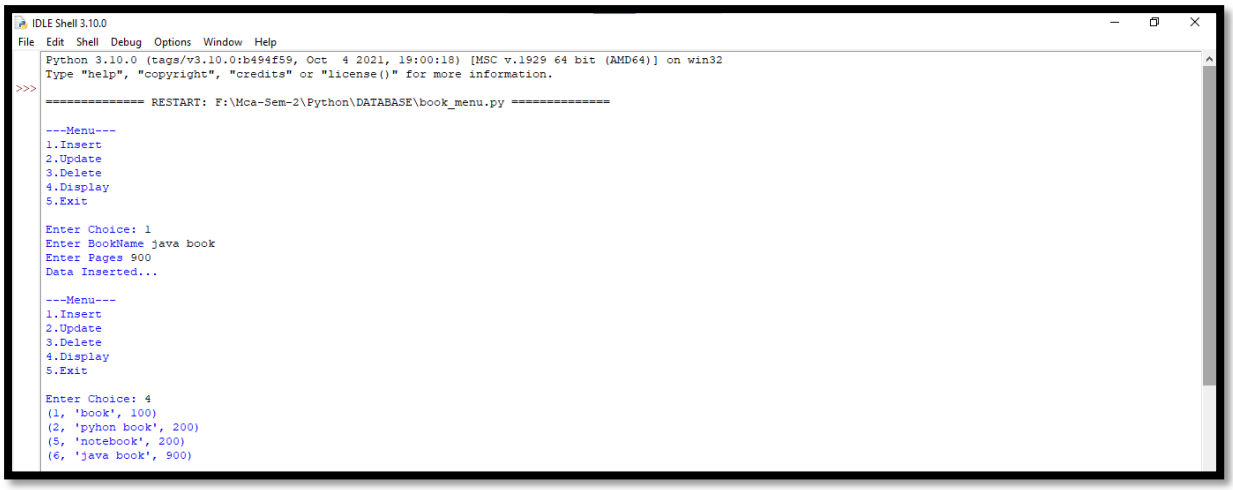
```
try:
    q.execute(fetch)
    data=q.fetchall()
    for i in data:
        print(i)
except:
    print("No Data")
print("")
bid=int(input("Enter BookId to Book Detail "))
sql="SELECT * FROM tbl_book where bid=%s"
upval=(bid,)
try:
    q.execute(sql,upval)
    data=q.fetchall()
    for i in data:
        bname=i[1]
        bpage=i[2]
    print("1. update bookname")
    print("2. update bookpages")
    ch=input("Enter choices ")
    if(ch=='1'):
        bname=input("Enter Bookname ")
    elif(ch=='2'):
        bpage=input("Enter Bookpage ")
    else:
        print("Invalid Input..")
    sql="update tbl_book set bname=%s,bpage=%s where bid=%s"
```

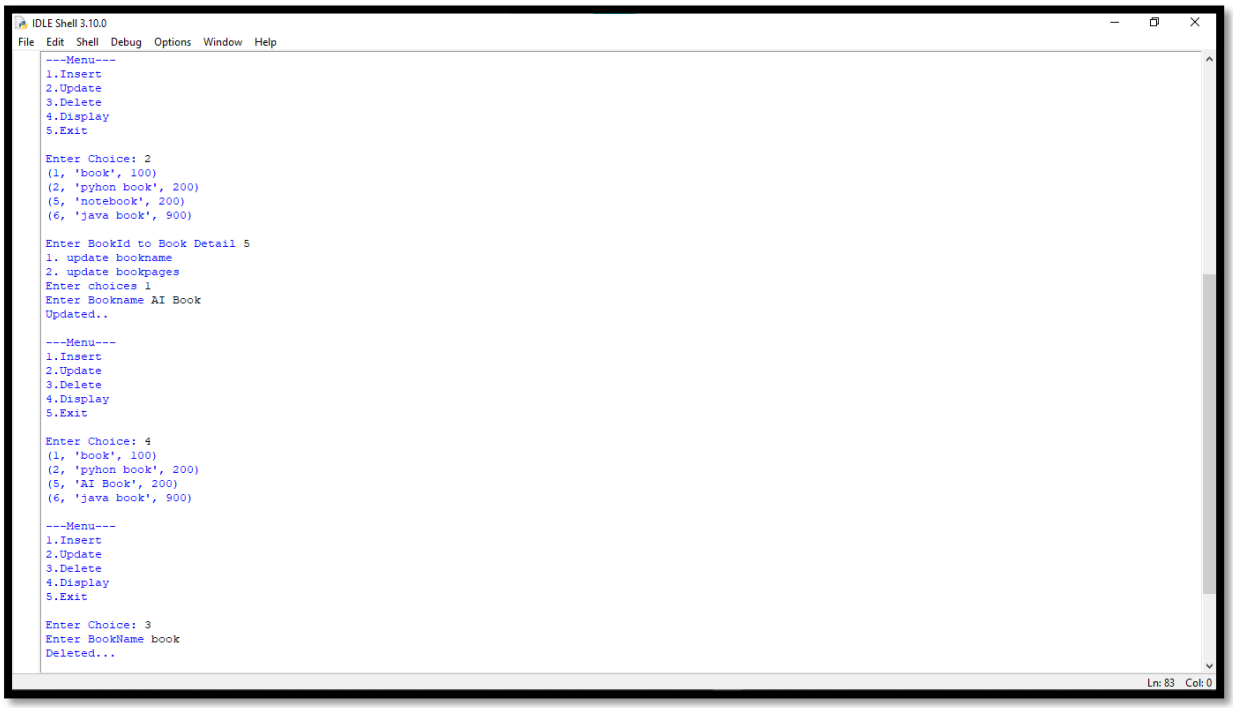
```
        val=(bname,bpage,bid)
        try:
            q.execute(sql,val)
            con.mycon.commit()
            print("Updated..")
        except:
            print("Not Updated..")
    except:
        print("No data available")


def delete():
    bname=input("Enter BookName ")
    q=con.mycon.cursor()
    sql="DELETE FROM `tbl_book` WHERE bname=%s"
    val=(bname,)
    try:
        v=q.execute(sql,val)
        con.mycon.commit()
        print("Deleted...")
    except:
        con.mycon.rollback()
        print("Not Deleted...")


def select():
    q=con.mycon.cursor()
    fetch="SELECT * FROM tbl_book"
    try:
```

```
        q.execute(fetch)
        data=q.fetchall()
        for i in data:
            print(i)
    except:
        print("No Data")



n=0
while n!=5:
    print("")
    print("---Menu---")
    print("1.Insert")
    print("2.Update")
    print("3.Delete")
    print("4.Display")
    print("5.Exit")
    print("")
    ch=int(input("Enter Choice: "))
    if ch==1:
        insert()
    elif ch==2:
        update()
    elif ch==3:
        delete()
    elif ch==4:
        select()
```

```
    else:
        break
```

Output:-





Practical:-22

22.Create Web Database Application “Event Registration” with options to
a) Event Registration

b) Cancel Registration

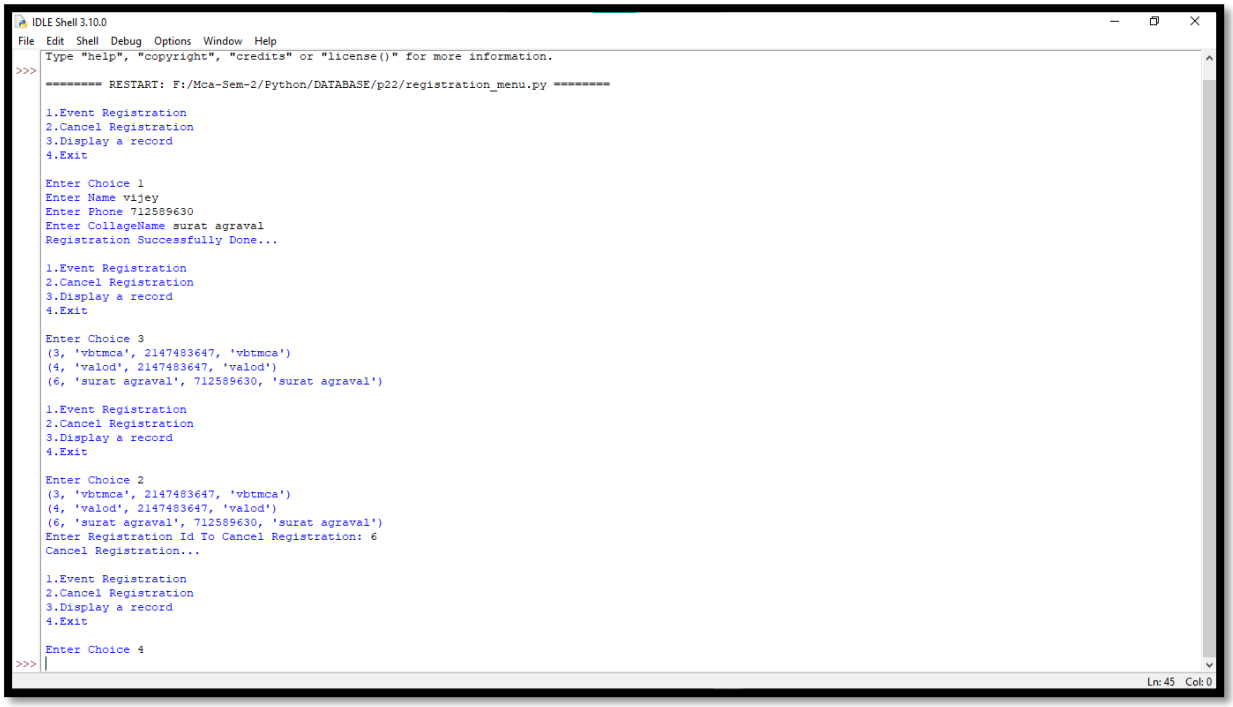
c) display a record

```
import connection as con
def registration():
    q=con.mycon.cursor()
    name=input("Enter Name ")
    phone=int(input("Enter Phone "))
    name=input("Enter CollageName ")
    data=(name,phone,name)
    sql="insert into tbl_registration(rname,rphone,clgname)values(%s,%s,%s)"
    try:
        q.execute(sql,data)
        con.mycon.commit()
        print("Registration Successfully Done...")
    except:
        con.mycon.rollback()
        print("Registration is Fail...")
def cancel():
    q=con.mycon.cursor()
    sql="select * from tbl_registration"
    try:
        q.execute(sql)
```

```
        data=q.fetchall()
        for i in data:
            print(i)
        rid=int(input("Enter Registration Id To Cancel Registration: "))
        val=(rid,)
        sql1="DELETE FROM tbl_registration where rid=%s"
        try:
            q.execute(sql1,val)
            con.mycon.commit()
            print("Cancel Registration...")
        except:
            con.mycon.rollback()
            print("Pendding To cancel Registration...")
    except:
        print("No Data Available")
def display():
    q=con.mycon.cursor()
    sql="select * from tbl_registration"
    try:
        q.execute(sql)
        data=q.fetchall()
        for i in data:
            print(i)
    except:
        print("No Data Available...")
n=0
while n!=4:
```

```
print("")
print("1.Event Registration")
print("2.Cancel Registration")
print("3.Display a record")
print("4.Exit")
print("")
n=int(input("Enter Choice "))
if n==1:
    registration()
elif n==2:
    cancel()
elif n==3:
    display()
else:
    break
```

Output:-



Part II: Advanced Topic: Data Analysis

1. Perform following operations on a CSV file

a. Create a data frame from csv file, dictionary, List of tuples

b. Operations on Data Frame Shape, head, tail

c. Retrieving rows / columns from data frame

d. Finding maximum and minimum values

e. Displaying statistical information

f. Performing queries

g. Data Analysis using groupby()

```
import pandas as p
n1=0
while n1!=8:
    print("1. Create a data frame from csv file, dictionary, List of tuples")
    print("2. Operations on Data Frame Shape, head, tail")
    print("3. Retrieving rows / columns from data frame")
    print("4. Finding maximum and minimum values")
    print("5. Displaying statistical information")
    print("6. Performing queries")
    print("7. Data Analysis using groupby()")
    n1=int(input("Enter choices "))
    if n1==1:
        n=0
        while n!=4:
            print("1.csv file")
            print("2.dictionary")
```

```
        print("3.List of tuples")
        print("4.Exit")
        n=int(input("Enter Choice "))
        if n==1:
            df=p.read_csv("myfile_csv.csv")
            print(df)
        elif n==2:
            di={"name":["keyur","Keyu","keyu4"],"surname":"Halpati"}
            df=p.DataFrame(di)
            print(df)
        elif n==3:
            tup=[(1,"keyur",13000),(2,"xyz",12000)]
            df=p.DataFrame(tup,columns=["id","name","salary"])
            print(df)
        else:
            break
    elif n1==2:
        n=0
        while n!=4:
            tup=[(1,"keyur",13000),(2,"xyz",12000),(3,"zzz",14000),(4,"xyy",15000)]
            print("1.Data Frame Shape")
            print("2.Data Frame head")
            print("3.Data Frame tail")
            print("4.Exit")
            n=int(input("Enter Choice "))
            if n==1:
                df=p.DataFrame(tup,columns=["id","name","salary"])
```

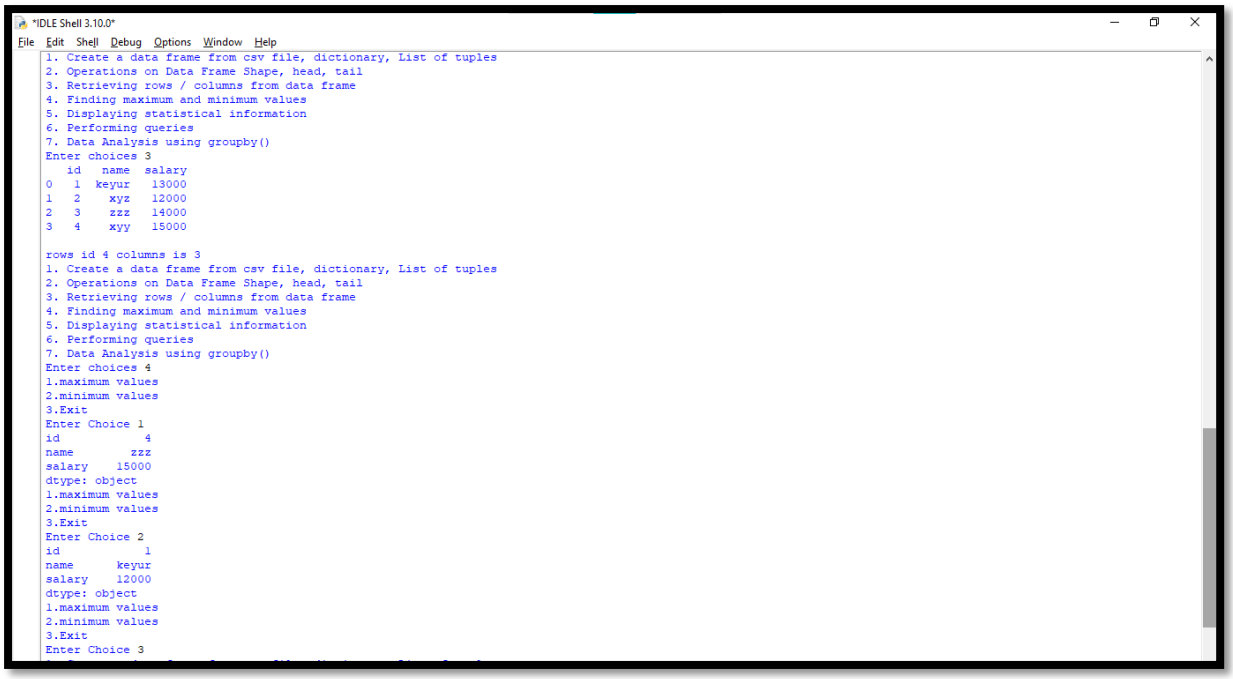
```
            print(df)
            r,c=df.shape
            print("")
            print("rows id",r,"columns is",c)
        elif n==2:
            df=p.DataFrame(tup,columns=["id","name","salary"])
            print(df.head(2))
        elif n==3:
            df=p.DataFrame(tup,columns=["id","name","salary"])
            print(df.tail(2))
        else:
            break

elif n1==3:
    tup=[(1,"keyur",13000),(2,"xyz",12000),(3,"zzz",14000),(4,"xyy",15000)]
    df=p.DataFrame(tup,columns=["id","name","salary"])
    print(df)
    r,c=df.shape
    print("")
    print("rows id",r,"columns is",c)
elif n1==4:
    n=0
    while n!=4:
        tup=[(1,"keyur",13000),(2,"xyz",12000),(3,"zzz",14000),(4,"xyy",15000)]
        print("1.maximum values")
        print("2.minimum values")
        print("3.Exit")
```

```
            n=int(input("Enter Choice "))
            if n==1:
                df=p.DataFrame(tup,columns=["id","name","salary"])
                print(df.max())
            elif n==2:
                df=p.DataFrame(tup,columns=["id","name","salary"])
                print(df.min())
            else:
                break
    elif n1==5:
        tup=[(1,"keyur",13000),(2,"xyz",12000),(3,"zzz",14000),(4,"xyy",15000)]
        df=p.DataFrame(tup,columns=["id","name","salary"])
        print(df.describe())
    elif n1==6:
        print("Inprogress...")
    elif n1==7:
        d=p.read_csv("myfile_csv.csv")
        df=p.DataFrame(d)
        print(df.groupby(['rollno','name']).mean())
    else:
        break
```

Output:-


IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: F:\Mca-Sem-2\Python\dataframe.py ==========
1. Create a data frame from csv file, dictionary, List of tuples
2. Operations on Data Frame Shape, head, tail
3. Retrieving rows / columns from data frame
4. Finding maximum and minimum values
5. Displaying statistical information
6. Performing queries
7. Data Analysis using groupby()
Enter choices 1
1.csv file
2.dictionary
3.List of tuples
4.Exit
Enter Choice 1
rollno name
0 1 keyur
1 1 keyur
2 1 keyur
3 4 vihang
4 5 bajaj
1.csv file
2.dictionary
3.List of tuples
4.Exit
Enter Choice 2
name surname
0 keyur Halpati
1 Keyu Halpati
2 keyu4 Halpati
1.csv file
2.dictionary
3.List of tuples
4.Exit
Enter Choice 3
id name salary
0 1 keyur 13000
1 2 xyz 12000
1.csv file
2.dictionary
3.List of tuples
4.Exit
Enter Choice 4
Ln: 53 Col: 14


IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
1. Create a data frame from csv file, dictionary, List of tuples
2. Operations on Data Frame Shape, head, tail
3. Retrieving rows / columns from data frame
4. Finding maximum and minimum values
5. Displaying statistical information
6. Performing queries
7. Data Analysis using groupby()
Enter choices 2
1.Data Frame Shape
2.Data Frame head
3.Data Frame tail
4.Exit
Enter Choice 1
id name salary
0 1 keyur 13000
1 2 xyz 12000
2 3 zzz 14000
3 4 xyy 15000
rows id 4 columns is 3
1.Data Frame Shape
2.Data Frame head
3.Data Frame tail
4.Exit
Enter Choice 2
id name salary
0 1 keyur 13000
1 2 xyz 12000
1.Data Frame Shape
2.Data Frame head
3.Data Frame tail
4.Exit
Enter Choice 3
id name salary
2 3 zzz 14000
3 4 xyy 15000
1.Data Frame Shape
2.Data Frame head
3.Data Frame tail
4.Exit
Enter Choice 4

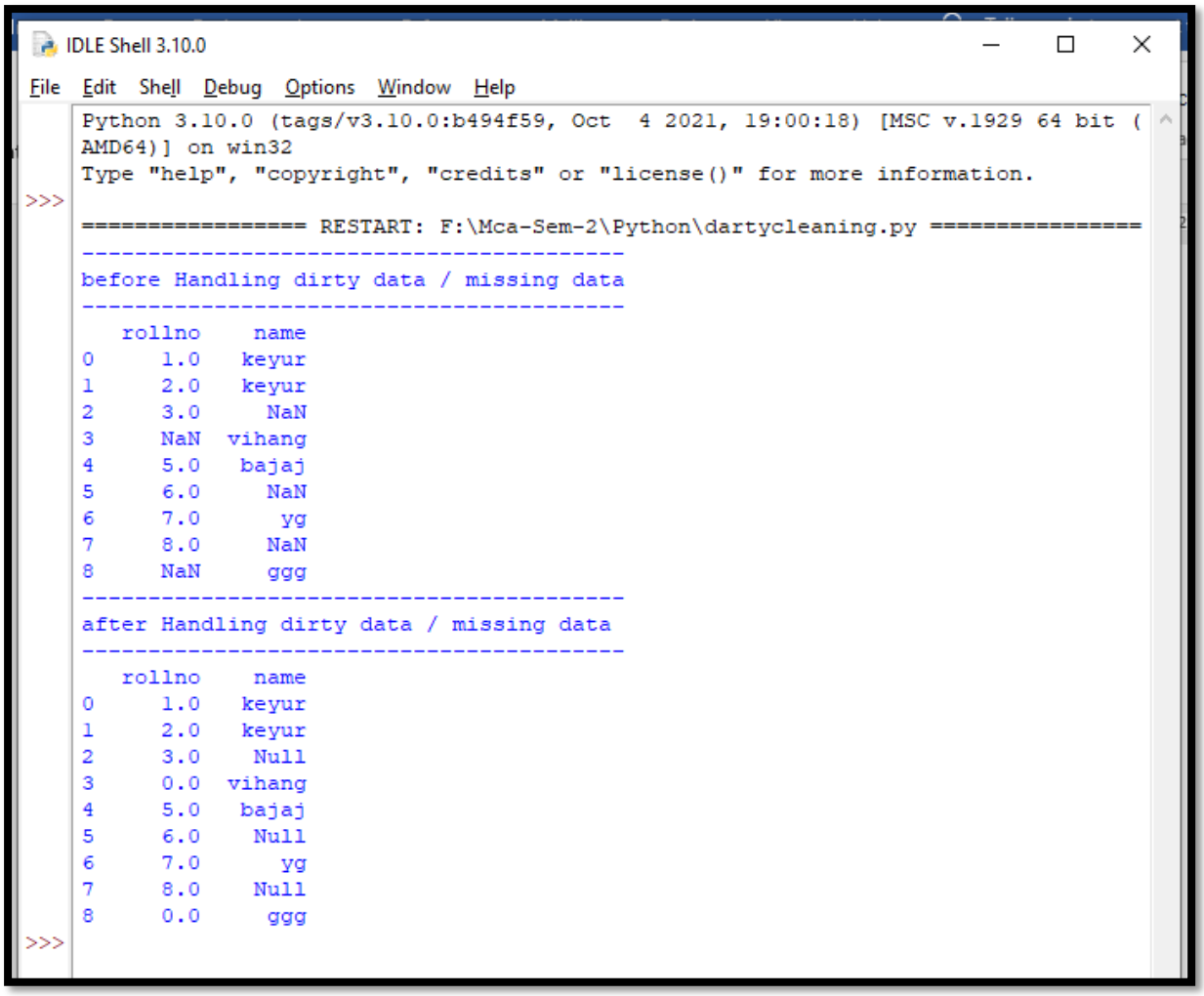

IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
1. Create a data frame from csv file, dictionary, List of tuples
2. Operations on Data Frame Shape, head, tail
3. Retrieving rows / columns from data frame
4. Finding maximum and minimum values
5. Displaying statistical information
6. Performing queries
7. Data Analysis using groupby()
Enter choices 3
id name salary
0 1 keyur 13000
1 2 xyz 12000
2 3 zzz 14000
3 4 xyy 15000
rows id 4 columns is 3
1. Create a data frame from csv file, dictionary, List of tuples
2. Operations on Data Frame Shape, head, tail
3. Retrieving rows / columns from data frame
4. Finding maximum and minimum values
5. Displaying statistical information
6. Performing queries
7. Data Analysis using groupby()
Enter choices 4
1.maximum values
2.minimum values
3.Exit
Enter Choice 1
id 4
name zzz
salary 15000
dtype: object
1.maximum values
2.minimum values
3.Exit
Enter Choice 2
id 1
name keyur
salary 12000
dtype: object
1.maximum values
2.minimum values
3.Exit
Enter Choice 3


1. Create a data frame from csv file, dictionary, List of tuples
2. Operations on Data Frame Shape, head, tail
3. Retrieving rows / columns from data frame
4. Finding maximum and minimum values
5. Displaying statistical information
6. Performing queries
7. Data Analysis using groupby()
Enter choices 5
id salary
count 4.000000 4.000000
mean 2.500000 13500.000000
std 1.290994 1290.994449
min 1.000000 12000.000000
25% 1.750000 12750.000000
50% 2.500000 13500.000000
75% 3.250000 14250.000000
max 4.000000 15000.000000
1. Create a data frame from csv file, dictionary, List of tuples
2. Operations on Data Frame Shape, head, tail
3. Retrieving rows / columns from data frame
4. Finding maximum and minimum values
5. Displaying statistical information
6. Performing queries
7. Data Analysis using groupby()
Enter choices 7
Empty DataFrame
Columns: []
Index: [(1, keyur), (4, vihang), (5, bajaj)]

Part III: Advanced Topic: Data cleaning

```
import pandas as p

df=p.read_csv("myfile_csv.csv")

print("----------------------------------------")

print("before Handling dirty data / missing data")

print("----------------------------------------")

print(df)

print("----------------------------------------")

print("after Handling dirty data / missing data")

print("----------------------------------------")

df1=df.fillna({'rollno':0,'name':'Null'})

print(df1)
```

Output:-


IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: F:\Mca-Sem-2\Python\dartycleaning.py ==========
before Handling dirty data / missing data
rollno name
0 1.0 keyur
1 2.0 keyur
2 3.0 NaN
3 NaN vihang
4 5.0 bajaj
5 6.0 NaN
6 7.0 yg
7 8.0 NaN
8 NaN ggg
after Handling dirty data / missing data
rollno name
0 1.0 keyur
1 2.0 keyur
2 3.0 Null
3 0.0 vihang
4 5.0 bajaj
5 6.0 Null
6 7.0 yg
7 8.0 Null
8 0.0 ggg
>>>

Part IV: Advanced Topic: Python for Data Visualization Library: pylab, matplotlib, seaborn.

Practical:-1

1.Write a program in python to implement simple interest and compound interest values on chart using PyLab. Show the difference between both. (Note: Use of object oriented paradigm is compulsory.)

Practical:-2

2.Using a data file, draw

a. Bar Graph

b. Histogram

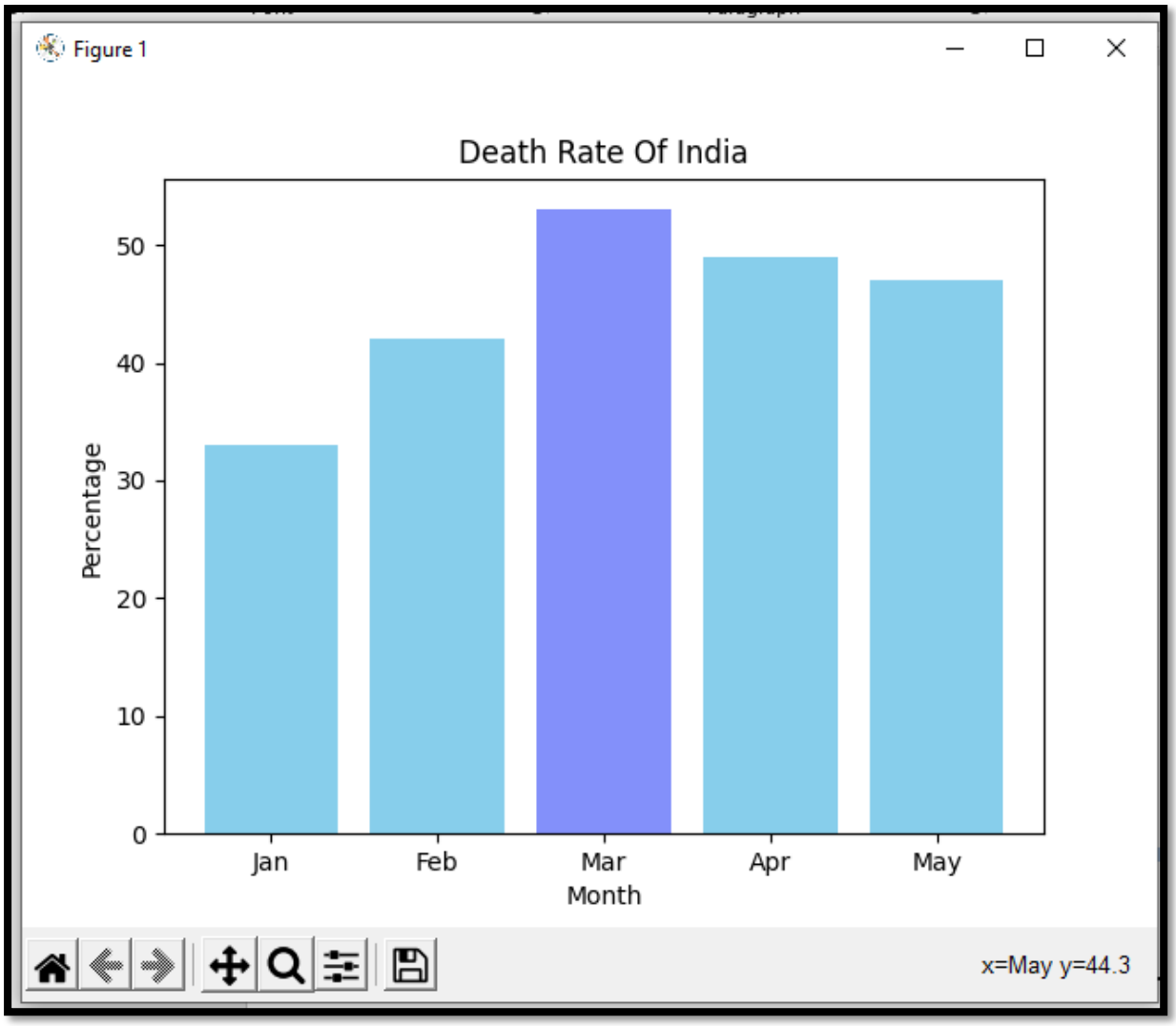
c. Pie Chart

d. Line Chart

a. Bar Graph

```
import matplotlib.pyplot as p
month = ['Jan','Feb','Mar','Apr','May']
Percentage = [33,42,53,49,47]
p.bar(month, Percentage, color=['skyblue', 'skyblue', '#8390FA', 'skyblue', 'skyblue'])
p.title('Death Rate Of India')
p.xlabel('Month')
p.ylabel('Percentage')
p.show()
```

Output:-

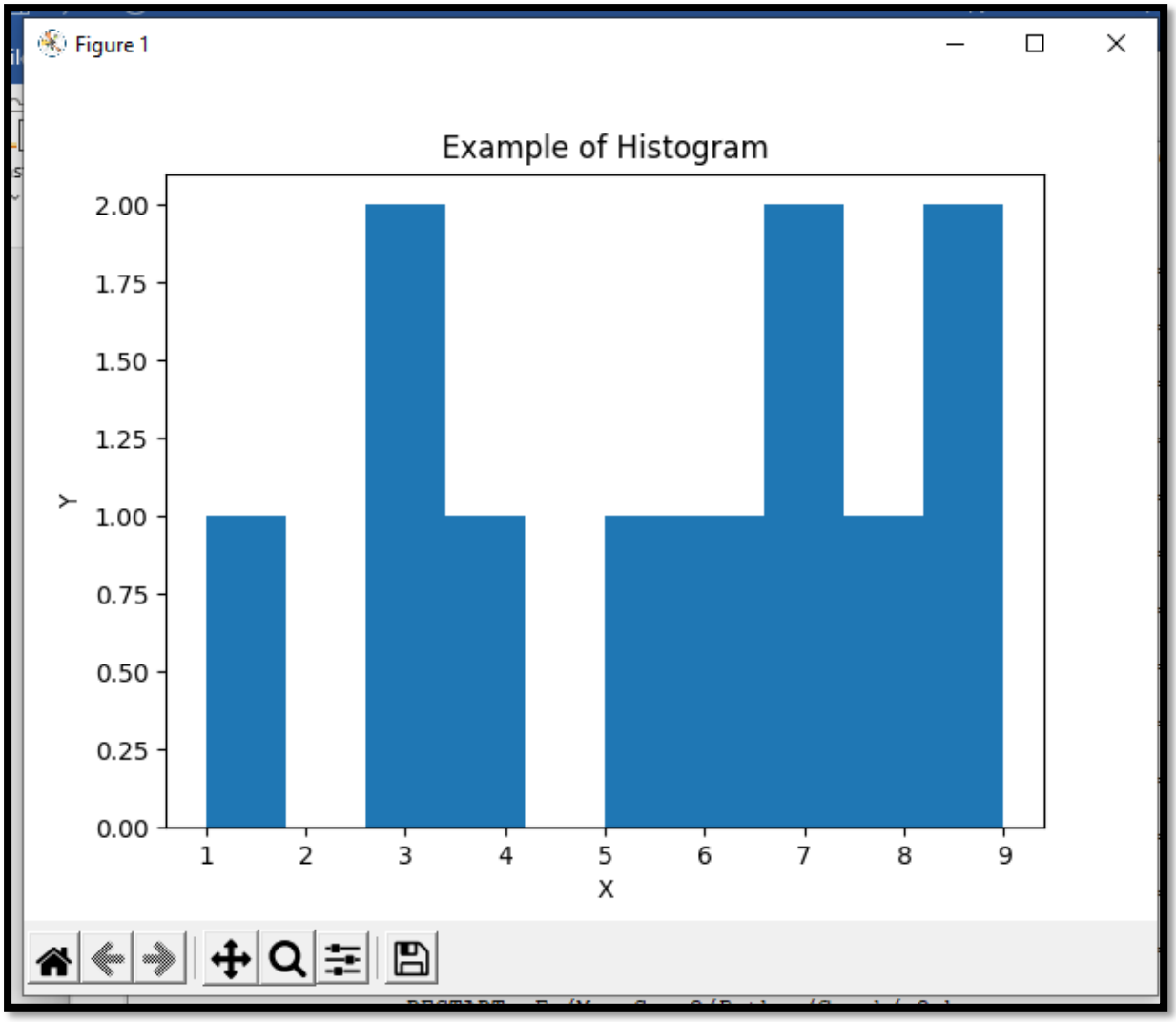


b. Histogram

```
import matplotlib.pyplot as p


x = [1,9,3,5,7,3,7,6,4,8,9]


p.hist(x, bins=10)
p.title("Example of Histogram")
p.xlabel("X")
p.ylabel("Y")
```

p.show()

Output:-

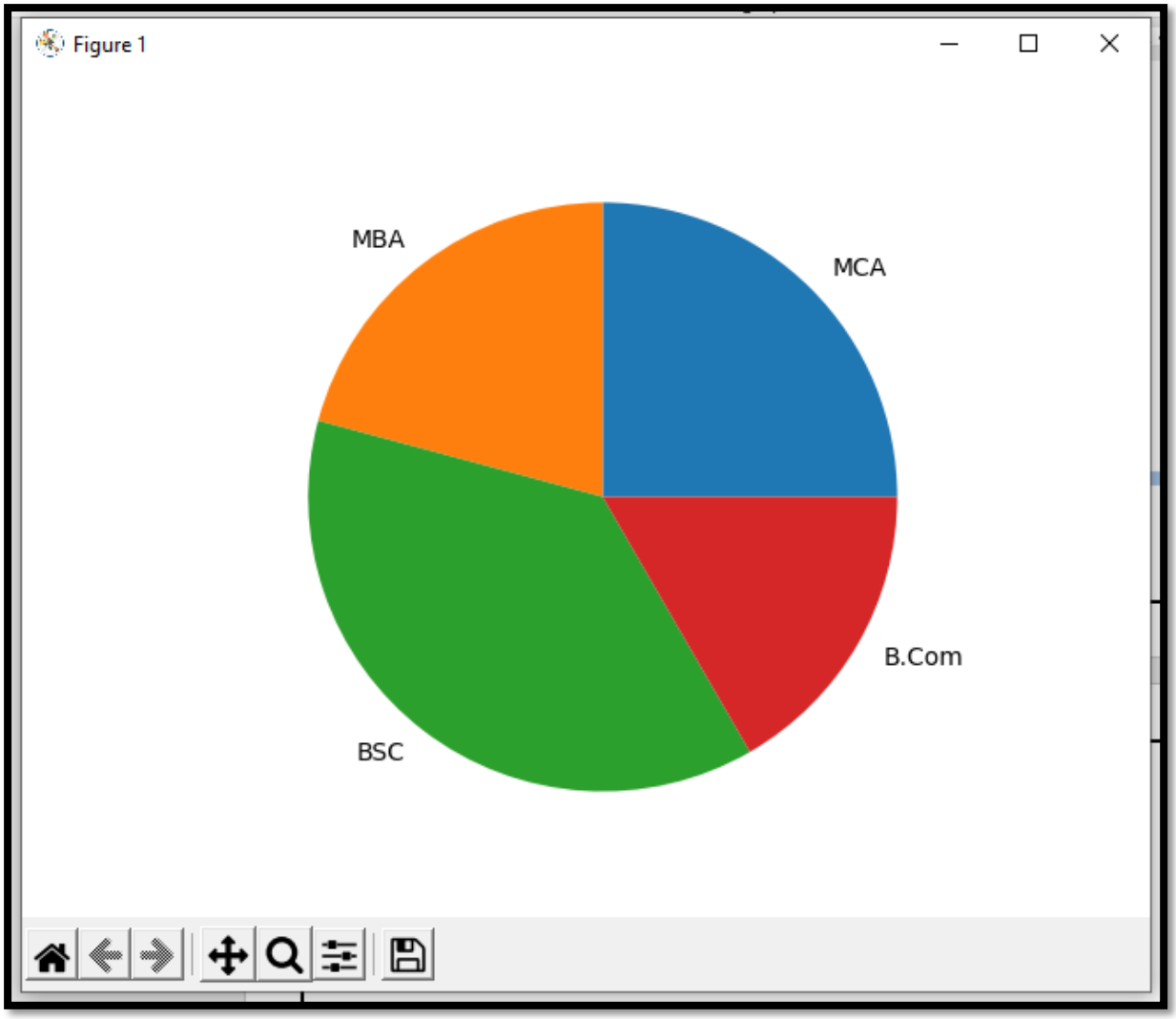


c. Pie Chart

```
import matplotlib.pyplot as p

labels = 'MCA', 'MBA', 'BSC', 'B.Com'

sizes = [30, 25, 45, 20]

fig1,a = p.subplots()

a.pie(sizes, labels=labels)

a.axis('equal')
```

p.show()

Output:-



d. Line Chart

```
import matplotlib.pyplot as p

Year = [2018,2019,2020,2021,2022]

Unemployment_Rate = [7.5,9,8,7.9,8.5]

p.plot(Year, Unemployment_Rate)

p.title('Student Result Rate vs Year')

p.xlabel('Year')

p.ylabel('Student Rate')

p.show()
```

Output:-

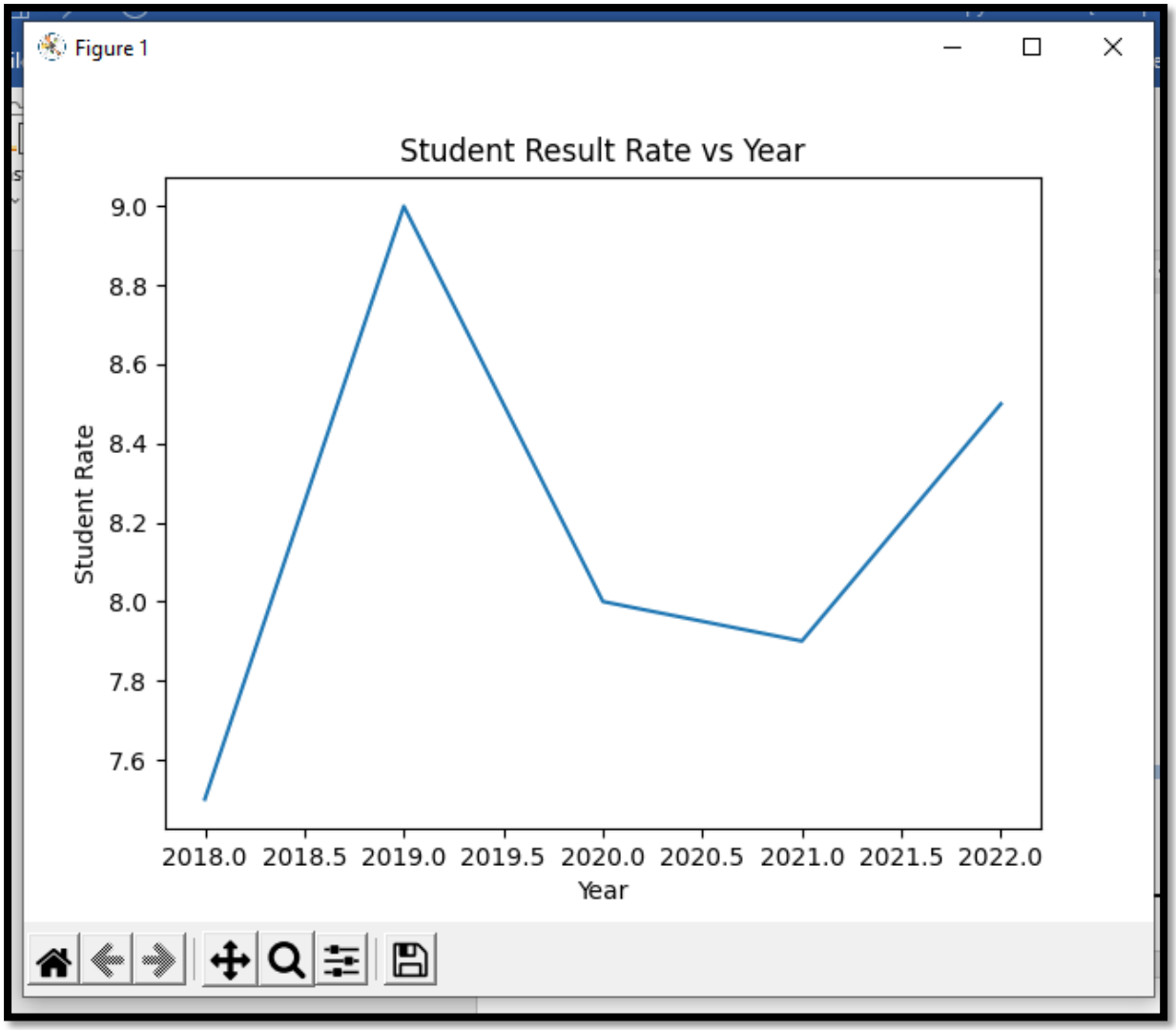


Figure 1
Student Result Rate vs Year
Student Rate
9.0
8.8
8.6
8.4
8.2
8.0
7.8
7.6
2018.0
2018.5
2019.0
2019.5
2020.0
2020.5
2021.0
2021.5
2022.0
Year

Practical:-3

Perform following operations on a CSV file

a. Create a data frame from csv file, dictionary, List of tuples

b. Operations on Data Frame Shape, head, tail

c. Retrieving rows / columns from data frame

d. Finding maximum and minimum values

e. Displaying statistical information

f. Performing queries

g. Handling missing data

```
import pandas as p
n1=0
while n1!=8:
    print("1. Create a data frame from csv file, dictionary, List of tuples")
    print("2. Operations on Data Frame Shape, head, tail")
    print("3. Retrieving rows / columns from data frame")
    print("4. Finding maximum and minimum values")
    print("5. Displaying statistical information")
    print("6. Performing queries")
    print("7. Handling missing data")
    print("8. Exit")
    n1=int(input("Enter choices "))
    if n1==1:
        n=0
        while n!=4:
            print("1.csv file")
            print("2.dictionary")
```

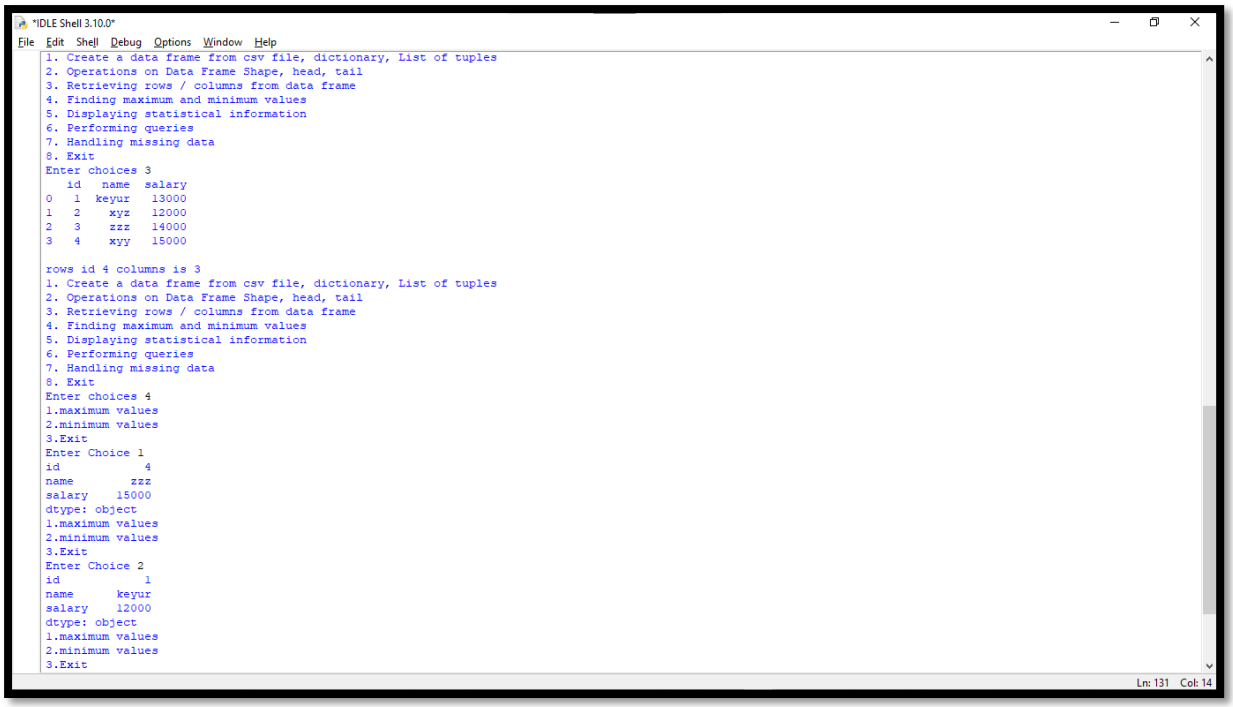
```
        print("3.List of tuples")
        print("4.Exit")
        n=int(input("Enter Choice "))
        if n==1:
            df=p.read_csv("myfile_csv.csv")
            print(df)
        elif n==2:
            di={"name":["keyur","Keyu","keyu4"],"surname":"Halpati"}
            df=p.DataFrame(di)
            print(df)
        elif n==3:
            tup=[(1,"keyur",13000),(2,"xyz",12000)]
            df=p.DataFrame(tup,columns=["id","name","salary"])
            print(df)
        else:
            break
    elif n1==2:
        n=0
        while n!=4:
            tup=[(1,"keyur",13000),(2,"xyz",12000),(3,"zzz",14000),(4,"xyy",15000)]
            print("1.Data Frame Shape")
            print("2.Data Frame head")
            print("3.Data Frame tail")
            print("4.Exit")
            n=int(input("Enter Choice "))
            if n==1:
                df=p.DataFrame(tup,columns=["id","name","salary"])
```

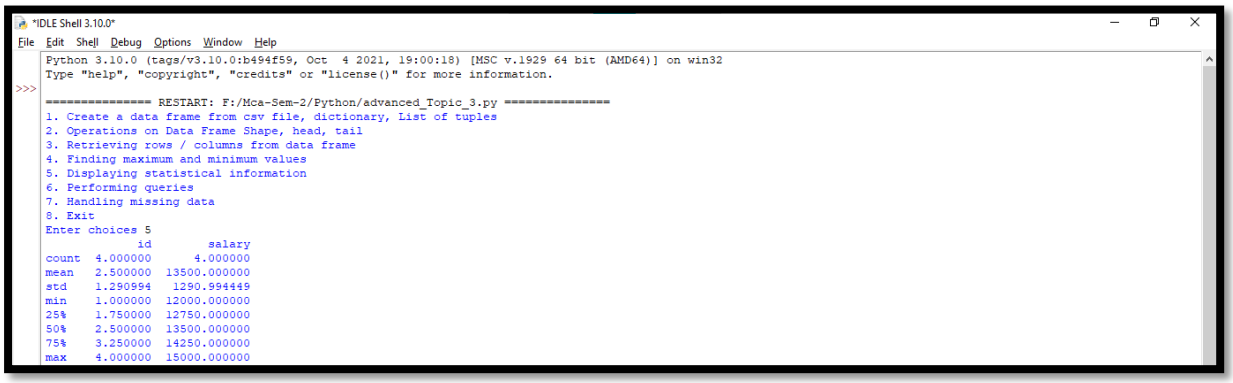
```
            print(df)
            r,c=df.shape
            print("")
            print("rows id",r,"columns is",c)
        elif n==2:
            df=p.DataFrame(tup,columns=["id","name","salary"])
            print(df.head(2))
        elif n==3:
            df=p.DataFrame(tup,columns=["id","name","salary"])
            print(df.tail(2))
        else:
            break

elif n1==3:
    tup=[(1,"keyur",13000),(2,"xyz",12000),(3,"zzz",14000),(4,"xyy",15000)]
    df=p.DataFrame(tup,columns=["id","name","salary"])
    print(df)
    r,c=df.shape
    print("")
    print("rows id",r,"columns is",c)
elif n1==4:
    n=0
    while n!=4:
        tup=[(1,"keyur",13000),(2,"xyz",12000),(3,"zzz",14000),(4,"xyy",15000)]
        print("1.maximum values")
        print("2.minimum values")
        print("3.Exit")
```

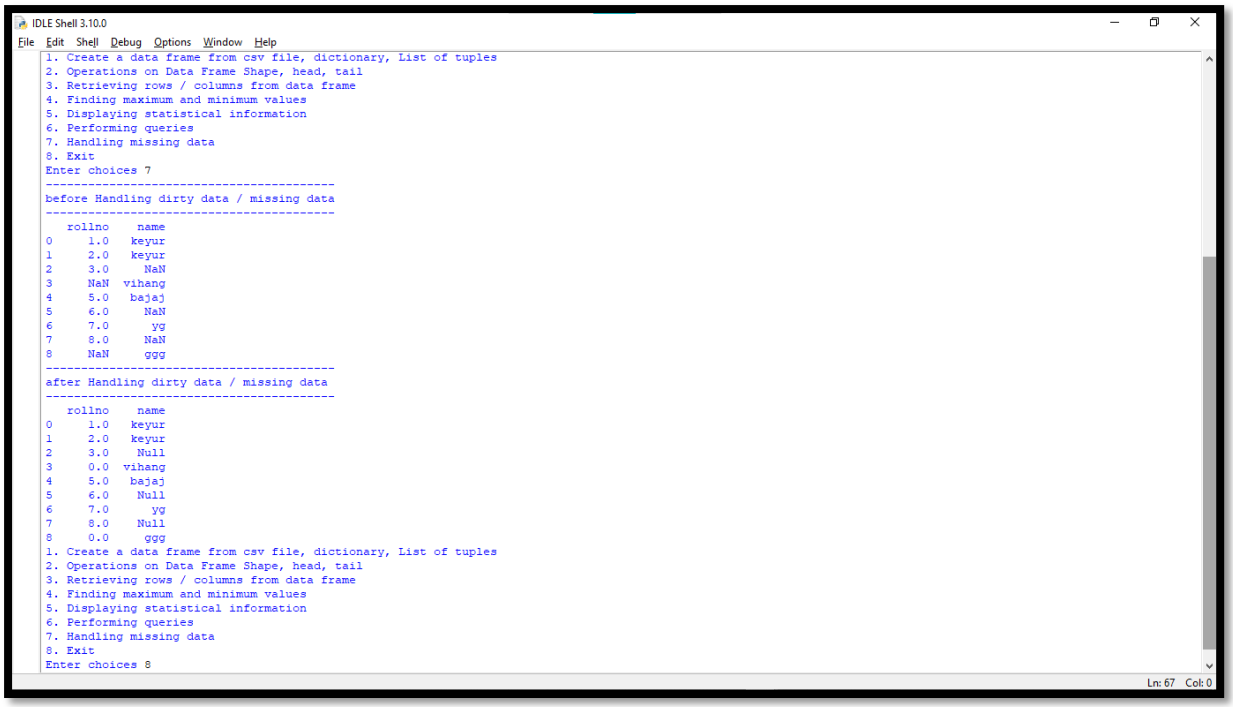
```
            n=int(input("Enter Choice "))
            if n==1:
                df=p.DataFrame(tup,columns=["id","name","salary"])
                print(df.max())
            elif n==2:
                df=p.DataFrame(tup,columns=["id","name","salary"])
                print(df.min())
            else:
                break
    elif n1==5:
        tup=[(1,"keyur",13000),(2,"xyz",12000),(3,"zzz",14000),(4,"xyy",15000)]
        df=p.DataFrame(tup,columns=["id","name","salary"])
        print(df.describe())
    elif n1==6:
        print("Inprogress...")
    elif n1==7:
        df=p.read_csv("myfile_csv.csv")
        print("----------------------------------------")
        print("before Handling dirty data / missing data")
        print("----------------------------------------")
        print(df)
        print("----------------------------------------")
        print("after Handling dirty data / missing data")
        print("----------------------------------------")
        df1=df.fillna({'rollno':0,'name':'Null'})
        print(df1)
    else:
```

break

Output:-


IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
1. Create a data frame from csv file, dictionary, List of tuples
2. Operations on Data Frame Shape, head, tail
3. Retrieving rows / columns from data frame
4. Finding maximum and minimum values
5. Displaying statistical information
6. Performing queries
7. Handling missing data
8. Exit
Enter choices 3
id name salary
0 1 keyur 13000
1 2 xyz 12000
2 3 zzz 14000
3 4 xyy 15000
rows id 4 columns is 3
1. Create a data frame from csv file, dictionary, List of tuples
2. Operations on Data Frame Shape, head, tail
3. Retrieving rows / columns from data frame
4. Finding maximum and minimum values
5. Displaying statistical information
6. Performing queries
7. Handling missing data
8. Exit
Enter choices 4
1.maximum values
2.minimum values
3.Exit
Enter Choice 1
id 4
name zzz
salary 15000
dtype: object
1.maximum values
2.minimum values
3.Exit
Enter Choice 2
id 1
name keyur
salary 12000
dtype: object
1.maximum values
2.minimum values
3.Exit
Ln: 131 Col: 14


IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: F:/Mca-Sem-2/Python/advanced_Topic_3.py ==========
1. Create a data frame from csv file, dictionary, List of tuples
2. Operations on Data Frame Shape, head, tail
3. Retrieving rows / columns from data frame
4. Finding maximum and minimum values
5. Displaying statistical information
6. Performing queries
7. Handling missing data
8. Exit
Enter choices 5
id salary
count 4.000000 4.000000
mean 2.500000 13500.000000
std 1.290994 1290.994449
min 1.000000 12000.000000
25% 1.750000 12750.000000
50% 2.500000 13500.000000
75% 3.250000 14250.000000
max 4.000000 15000.000000


IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
1. Create a data frame from csv file, dictionary, List of tuples
2. Operations on Data Frame Shape, head, tail
3. Retrieving rows / columns from data frame
4. Finding maximum and minimum values
5. Displaying statistical information
6. Performing queries
7. Handling missing data
8. Exit
Enter choices 7
before Handling dirty data / missing data
rollno name
0 1.0 keyur
1 2.0 keyur
2 3.0 NaN
3 NaN vihang
4 5.0 bajaj
5 6.0 NaN
6 7.0 yg
7 8.0 NaN
8 NaN ggg
after Handling dirty data / missing data
rollno name
0 1.0 keyur
1 2.0 keyur
2 3.0 Null
3 0.0 vihang
4 5.0 bajaj
5 6.0 Null
6 7.0 yg
7 8.0 Null
8 0.0 ggg
1. Create a data frame from csv file, dictionary, List of tuples
2. Operations on Data Frame Shape, head, tail
3. Retrieving rows / columns from data frame
4. Finding maximum and minimum values
5. Displaying statistical information
6. Performing queries
7. Handling missing data
8. Exit
Enter choices 8
Ln: 67 Col: 0