

COMPUTER PRACTICAL FILE

**KONICAA SHARMA
XII-D
2020-21
Board Roll No. :26631915**

INDEX

S.No.	Program Code	Program Details	Date of execution	Signature
Sno-3	PRINT-1	To calculate and display total marks and percentage of a student from a given list storing marks of student	15/4/20	
Sno-4	PRINT-2	To multiply an element by 2 if it's at odd index for a given list containing both numbers and strings	15/4/20	
Sno-7	PRINT-3	To swap the elements of a list containing numbers such that if the second element of two consecutive elements is divisible by 5 is swapped with the first element	20/4/20	
Sno-10	PRINT-4	To input 'n' classes and names of their class teachers to store them in a dictionary and display the same. Also accept a particular class from the user and display the name of the class teacher	25/4/20	
Sno-11	PRINT-5	To store student names and their percentage in a dictionary and delete a particular student name from the dictionary. Also display the dictionary after deletion	28/4/20	
Sno-16	PRINT-6	To find the second most repeated word in string	10/5/20	
Sno-24	PRINT-7	To concatenate 2 dictionaries to create a new one	15/5/20	
Sno-25	PRINT-8	To check if given key already exists in dictionary	18/5/20	
Sno-28	PRINT-9	To sort a dictionary by a key	20/5/20	
Sno-29	PRINT-10	To combine two dictionaries adding values for common keys	20/5/20	
Sno-31	PRINT-11	To sort a list alphabetically in a dictionary	25/5/20	
Sno-43	PRINT-12	To calculate income tax as per the following table	25/5/20	
Sno-49	PRINT-13	To find age of a person using date and time library	10/6/20	
Sno-50	PRINT-14	To compare ages of two persons and find who is elder using datetime library	15/6/20	

Sno-51	PRINT-15	To create a package in python calculating area, perimeter, volume of different geometric shapes	20/6/20	
Sno-54	PRINT-16	To calculate volume and area of a sphere inside separate modules and import them in one complete package	20/6/20	
Sno-62	PRINT-17	To check number of lines starting with an uppercase letter in text1.txt	5/7/20	
Sno-63	PRINT-18	To count number of vowels in a text file	5/7/20	
Sno-65	PRINT-19	To write sequence of data like strings ,tuples ,list in test1.txt and append using writelines () and open file using ‘with’ statement	6/7/20	
Sno-66	PRINT-20	To copy contents of one text file to another	8/7/20	
Sno-78	PRINT-21	To write and read records of binary file containing string	10/7/20	
Sno-79	PRINT-22	To write and read records of employees in binary file using tuples	15/7/20	
Sno-80	PRINT-23	To write and read records of student in binary file containing lists	20/7/20	
Sno-83	PRINT-24	To write, read, modify and delete records of employee in ‘emp.dat’ and to find employee with highest salary using dictionary	29/7/20	
Sno-84	PRINT-25	Library management using list	8/8/20	
Sno-86	PRINT-26	To read contents from file sports.dat and create a file named Athletic.dat copying only those records from sports.dat where the event name is ‘Athletics’	15/8/20	
Sno-91	PRINT-27	To read characters from keyboard one by one .All lower case characters get stored inside the file LOWER,all uppercase characters get stored inside the file UPPER and all other characters get stored inside the file OTHERS	23/8/20	
Sno-107	PRINT-28	Menu driven program working on Batch.csv	25/8/20	
Sno-109	PRINT-29	To read records of employees in myfile.csv and display those who have salary more than 70000	28/8/20	
Sno-110	PRINT-30	To read contacts.csv containing name and phone number of people and to search phone number of a specific person	28/8/20	

Sno-116	PRINT-31	To read employee records from emp.csv file and copy in binary file Emp2.dat and read all records from Emp2.dat	1/9/20	
Sno-117	PRINT-32	To sort emp.csv's records according to salary column in descending order or name column in ascending order	5/9/20	
Sno-118	PRINT-33	Implementing Stack using list, create a stack, push an element to a Stack, checking for empty Stack (Underflow), pop operation/deleting elements from a stack and traversal/displaying a Stack	10/9/20	
Sno-119	PRINT-34	Implementing queue using list, create a queue, checking for empty queue, insert an element to the queue, deleting element from queue, traversal/displaying queue	10/9/20	
Sno-122	PRINT-35	To write add(books) and delete(books) methods in Python to add and remove books considering them to act as append() and pop() operations in Stack	25/9/20	
Sno-123	PRINT-36	To write AddClient() and DeleteClient() methods to add a new client and delete a client from client name list of Queue data structure	27/9/20	
Sno-126	PRINT-37	A stack STK and a queue QUE are being maintained. Check whether they have the same size, check for equality and stop and report if elements are different	1/10/20	
Sno-129	PRINT-38	To insert or delete an element from a queue depending on user's choice	5/10/20	
Sno-133	PRINT-39	To convert the first letter of each word to uppercase and remaining to lowercase	10/10/20	
Sno-134	PRINT-40	To replace a character by another maintaining the case	13/10/20	
	PRINT-41	SAMPLE PROJECT ON CONNECTIVITY OF MYSQL AND PYTHON	14/10/20	
		ADVANCED SQL QUERIES(Q17 , Q18 & Q19 of chap10 Preeti Arora)	14/10/20	

PRINT-1: S.No.-3

To calculate and display total marks and percentage of a student from a given list storing marks of student

```
##### PROGRAM PRINT-1 #####
#SNO-03:To calculate and display total marks and percentage
#of a student from a given list storing marks of student
#AUTHOR-KONICAA SHARMA

s=int(input("Enter the number of subjects="))
total=t=0;a=[]
for i in range(s):
    m1=int(input("Enter marks in subject="))
    o=int(input("Out Of?="))
    a.append(m1)
    total+=o
for k in a:
    t=t+k
print("List containing marks of student is:",a)
print("Total marks out of",total,"are:",t)
print("Percentage=", (t/total)*100)
```

```
===== RESTART: C:\Users\Hp\Desktop\cs exam practice\q17.py =====
Enter the number of subjects=5
Enter marks in subject=100
Out Of?=100
Enter marks in subject=98
Out Of?=100
Enter marks in subject=100
Out Of?=100
Enter marks in subject=99
Out Of?=100
Enter marks in subject=99
Out Of?=100
List containing marks of student is: [100, 98, 100, 99, 99]
Total marks out of 500 are: 496
Percentage= 99.2
>>>
```

PRINT-2: S.No.-4

To multiply an element by 2 if it's at odd index for a given list containing both numbers and strings

```
##### PROGRAM PRINT-2 #####
#SNo-4:To multiply an element by 2 if it's at odd index
#for a given list containing both numbers and strings
#AUTHOR-KONICAA SHARMA

a=eval(input('Enter a list='))
size=len(a)
for i in range(0,size,2):
    a[i]*=2
print("List after modification:",a)
```

```
===== RESTART: C:\Users\Hp\Desktop\konicaahcsclass12\q18ch1.py =====
Enter a list=[12,'14','Red2',80,'Honesty',2,'8']
List after modification: [24, '14', 'Red2Red2', 80, 'HonestyHonesty', 2, '88']
>>>
```

PRINT-3: S.No.-7

To swap the elements of a list containing numbers such that if the second element of two consecutive elements is divisible by 5 is swapped with the first element

```
#####
# PROGRAM PRINT-3 #####
#SNo-7:To swap the elements of a list containing numbers such
#that if the second element of two consecutive elements
#is divisible by 5 is swapped with the first element
#KONICAA SHARMA

a=[3,25,13,6,35,8,14,45]
print("Original list:",a)
for j in range(len(a)-1):
    if a[j+1]%5==0:
        a[j],a[j+1]=a[j+1],a[j]
print("Modified list:",a)
```

```
>>>
===== RESTART: C:\Users\Hp\Desktop\konicaahcsclass12\q21 ch1.py ======
Original list: [3, 25, 13, 6, 35, 8, 14, 45]
Modified list: [25, 3, 13, 35, 6, 8, 45, 14]
>>>
```

PRINT-4: S.No.-10

To input 'n' classes and names of their class teachers to store them in a dictionary and display the same. Also accept a particular class from the user and display the name of the class teacher

```
##### PROGRAM PRINT-4 #####
#SNo-10:To input 'n' classes and names of their class teachers to store
#them in a dictionary and display the same. Also accept a particular
#class from the user and display the name of the class teacher
#AUTHOR-KONICAA SHARMA

CLASS=dict()
a=[]
n=int(input('Enter the number of classes you want to enter='))
for i in range(n):
    clas=input('Enter class=')
    cl=input('Enter name of class teacher=')
    CLASS[clas]=cl
    a.append(clas)
print("The records are-")
print(CLASS)
c=input('Enter the class whose class teacher is to be found:')
if c in a:
    print(CLASS[c])
else:
    print("The following class is not mentioned in the records")
```

```
===== RESTART: C:\Users\Hp\Desktop\konicaahcsclass12\q24.py =====
Enter the number of classes you want to enter=3
Enter class=XII-A
Enter name of class teacher=Pooja
Enter class=XII-B
Enter name of class teacher=Barkha
Enter class=XII-C
Enter name of class teacher=Preeta
The records are-
{'XII-A': 'Pooja', 'XII-B': 'Barkha', 'XII-C': 'Preeta'}
Enter the class whose class teacher is to be found:XII-A
Pooja
>>>
===== RESTART: C:\Users\Hp\Desktop\konicaahcsclass12\q24.py =====
Enter the number of classes you want to enter=2
Enter class=XI-A
Enter name of class teacher=Samita
Enter class=XI-B
Enter name of class teacher=Vanita
The records are-
{'XI-A': 'Samita', 'XI-B': 'Vanita'}
Enter the class whose class teacher is to be found:XI-C
The following class is not mentioned in the records
```

PRINT-5: S.No.-11

To store student names and their percentage in a dictionary and delete a particular student name from the dictionary. Also display the dictionary after deletion.

```
#####
# PROGRAM PRINT-5 #####
#SNo-11:To store student names and their percentage in a dictionary
#and delete a particular student name from the dictionary
#Also display dictionary after deletion
#AUTHOR-KONICAA SHARMA

CLASS=dict()
a=[]
n=int(input('Enter the no. of students='))
for i in range(n):
    nm=input('Enter name of student=')
    per=input('Enter percentage=')
    CLASS[nm]=per
    a.append(nm)
print("Original dictionary is-")
print(CLASS)
dl=input('Enter the student name whose record is to be deleted=')
if dl in a:
    del(CLASS[dl])
    print("Modified list:",CLASS)
else:
    print("Given name does not exist in dictionary")
```

```
=====
RESTART: C:\Users\Hp\Desktop\konicaahcsclass12\q25.py =====
Enter the no. of students=3
Enter name of student=ARCHANA
Enter percentage=98
Enter name of student=LATA
Enter percentage=89
Enter name of student=KAVYA
Enter percentage=90
Original dictionary is-
{'ARCHANA': '98', 'LATA': '89', 'KAVYA': '90'}
Enter the student name whose record is to be deleted=KAVYA
Modified list: {'ARCHANA': '98', 'LATA': '89'}
>>>
=====
RESTART: C:\Users\Hp\Desktop\konicaahcsclass12\q25.py =====
Enter the no. of students=2
Enter name of student=RITU
Enter percentage=78
Enter name of student=ANEESHA
Enter percentage=90
Original dictionary is-
{'RITU': '78', 'ANEESHA': '90'}
Enter the student name whose record is to be deleted=RAHUL
Given name does not exist in dictionary
>>>
```

PRINT-6: S.No.-16
To find the second most repeated word in string

```
##### PROGRAM PRINT-6 #####
#SNo-16:To find second most repeated words in string
#AUTHOR-KONICAA SHARMA

s=input('Enter string=')
s2=s.lower()
l=s2.split()
d={}
q={}
for i in l:
    c=l.count(i)
    d[i]=c
#print(d)
m=max(d.values())
for i in d:
    if d!=m:
        q[i]=d[i]
print(q)
m=max(q.values())
for i in q:
    if q[i]==m:
        print('Second most repeated word is:-',i)
```

```
===== RESTART: C:\Users\Hp\Desktop\konicaahcsclass12\q30.py =====
Enter string=Nory was Catholic because her mother was a Catholic, and Nory's mother was a Catholic because her father was a Catholic, and her father was a Catholic because his mother was a Catholic
{'nory': 1, 'was': 6, 'catholic': 4, 'because': 3, 'her': 3, 'mother': 3, 'a': 5, 'catholic,'': 2, 'and': 2, 'nory's': 1, 'father': 2, 'his': 1}
Second most repeated word is:- was
>>> |
```

PRINT-7: S.No.-24
To concatenate 2 dictionaries to create a new one

```
#####
# PROGRAM PRINT-7 #####
#SNo29-To concatenate 2 dictionaries to create a new one
#AUTHOR-KONICAA SHARMA

d1=dict()
size=int(input('Mention the number of entries:'))
for i in range(size):
    a=input('enter a letter=')
    b=input('enter number=')
    d1[a]=b
print(d1)
d2=dict()
print()
size=int(input('Mention the number of entries:'))
for i in range(size):
    a=input('enter a letter=')
    b=input('enter number=')
    d2[a]=b
print(d2)
d1.update(d2)
print("The concatenated dictionary:")
print(d1)
```

```
>>>
=====
RESTART: C:\Users\Hp\Desktop\konicaahcsclass12\q38
Mention the number of entries:3
enter a letter=A
enter number=1
enter a letter=B
enter number=2
enter a letter=C
enter number=3
{'A': '1', 'B': '2', 'C': '3'}

Mention the number of entries:2
enter a letter=D
enter number=4
enter a letter=E
enter number=5
{'D': '4', 'E': '5'}
The concatenated dictionary:
{'A': '1', 'B': '2', 'C': '3', 'D': '4', 'E': '5'}
>>> |
```

PRINT-8: S.No.-25
To check if given key already exists in dictionary

```
##### PROGRAM PRINT-8 #####
#SNo-25:To check if given key already exists in dictionary
#AUTHOR-KONICAA SHARMA

d1=dict()
size=int(input('Mention the number of entries you want to make='))
for i in range(size):
    a=input('Enter a letter=')
    b=input('Enter no.')
    d1[a]=b
print(d1)
d2=dict()
key=input('Enter key to check=')
if key in d1:
    print('Value of the given key is',d1[key])
    print('Given key exists')
else:
    print('Given key not exists')
```

```
>>>
=====
RESTART: C:\Users\Hp\Desktop\konicaahcsclass12\q
Mention the number of entries you want to make=3
Enter a letter=A
Enter no.1
Enter a letter=B
Enter no.2
Enter a letter=C
Enter no.3
{'A': '1', 'B': '2', 'C': '3'}
Enter key to check=B
Value of the given key is 2
Given key exists
>>>
=====
RESTART: C:\Users\Hp\Desktop\konicaahcsclass12\q
Mention the number of entries you want to make=2
Enter a letter=D
Enter no.4
Enter a letter=E
Enter no.5
{'D': '4', 'E': '5'}
Enter key to check=K
Given key not exists
>>>
```

PRINT-9: S.No.-28:
To sort a dictionary by a key

```
##### PROGRAM PRINT-8 #####
#SNo-28:To sort a dictionary by key
#AUTHOR-KONICAA SHARMA

d1=dict();d2={};l=[]
size=int(input('Mention the number of entries you want to make='))
print()
for i in range(size):
    a=input('Enter key=')
    b=input('Enter value=')
    d1[a]=b
    l.append(a)
print("Original dictionary:",d1)
l.sort()
for i in l:
    d2[i]=d1[i]
print("Modified dictionary:",d2)

>>>
===== RESTART: C:\Users\Hp\Desktop\konicaahcsclass12\q42.py =====
Mention the number of entries you want to make=5

Enter key=J
Enter value=3
Enter key=F
Enter value=7
Enter key=A
Enter value=0
Enter key=N
Enter value=1
Enter key=V
Enter value=4
Original dictionary: {'J': '3', 'F': '7', 'A': '0', 'N': '1', 'V': '4'}
Modified dictionary: {'A': '0', 'F': '7', 'J': '3', 'N': '1', 'V': '4'}
>>>
===== RESTART: C:\Users\Hp\Desktop\konicaahcsclass12\q42.py =====
Mention the number of entries you want to make=3

Enter key=6
Enter value=G
Enter key=2
Enter value=C
Enter key=4
Enter value=G
Original dictionary: {'6': 'G', '2': 'C', '4': 'G'}
Modified dictionary: {'2': 'C', '4': 'G', '6': 'G'}
>>> |
```

PRINT-10: S.No.-29

To combine two dictionaries adding values for common keys

```
#####
# PROGRAM PRINT-10 #####
#SNo-29:: To combine two dictionaries adding values for common keys
#AUTHOR-KONICAA SHARMA

lis=[]
for i in range(2):
    d={}
    print('*****')
    print('For dictionary',i+1)
    n=int(input('Enter length of the dictionary='))
    for i in range(n):
        print('---')
        key=input('Enter key=')
        val=int(input('Enter numerical value of the key='))
        d[key]=val
    lis.append(d)
d1=lis[0]
d2=lis[1]
for key in d1:
    if key in d2:
        d2[key]=d2[key]+d1[key]
    else:
        d2.update({key:d1[key]})

print('XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX')
print('\nConcatenated dictionary is:\n',d2)
*****  

For dictionary 1  

Enter length of the dictionary=3  

-----  

Enter key=a  

Enter numerical value of the key=100  

-----  

Enter key=b  

Enter numerical value of the key=200  

-----  

Enter key=c  

Enter numerical value of the key=300  

*****  

For dictionary 2  

Enter length of the dictionary=3  

-----  

Enter key=a  

Enter numerical value of the key=300  

-----  

Enter key=b  

Enter numerical value of the key=200  

-----  

Enter key=d  

Enter numerical value of the key=400  

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  

Concatenated dictionary is:  

{'a': 400, 'b': 400, 'd': 400, 'c': 300}
```

PRINT-11:S.No.-31
To sort a list alphabetically in a dictionary

```
#####
# PROGRAM PRINT-11 #####
#SNo-31:To sort a list alphabetically in a dictionary
#AUTHOR-KONICAA SHARMA
list0=eval(input('Enter your list:'))
list1=[]
for i in list0:
    i=str(i)
    list1.append(i)
list1=sorted(list1)
d={}
for i in list1:
    i=str(i)
    i=i.title()
    key=i[0]
    interlist=[]
    for j in list1:
        j=str(j)
        j=j.title()
        key2=j[0]
        if key2==key:
            interlist.append(j)
        else:
            pass
    d[key]=sorted(interlist)
print('\nThe dictionary after sorting the list is:\n',d)
print('*'*50)
print('Initial Object:\tWords or associated numbers sorted')
for i in d:
    print(i,':\t\t',d[i])
>>>
=====
RESTART: C:\Users\Hp\Documents\STACK\sno31.py =====
Enter your list:[1,231,11123,'You','see','yak','yell','sea','salmon','yes','2312']

The dictionary after sorting the list is:
{'1': ['1', '11123'], '2': ['231', '2312'], 'Y': ['Yak', 'Yell', 'Yes', 'You'],
 'S': ['Salmon', 'Sea', 'See']}

*****
Initial Object: Words or associated numbers sorted
1 : ['1', '11123']
2 : ['231', '2312']
Y : ['Yak', 'Yell', 'Yes', 'You']
S : ['Salmon', 'Sea', 'See']
>>>
```

PRINT-12: S.No.-43
To calculate income tax as per the following table

INCOME	SLAB(in %)
200,000	5
200,000-500,000	10
Above 500,000	20

<pre>##### PROGRAM PRINT-12 ##### #SNo-43:To calculate income tax as per the given table #AUTHOR-KONICAA SHARMA i=int(input("Enter Income=")) a=10000#5 % of 200000 b=30000#10% of 300000 if i<=200000: tax=(i/100)*5 elif i>200000 and i<=500000: amt=i-200000 tax=a+(amt/100)*10 elif i>500000: am2=i-500000 tax=a+b+(am2/100)*20 print("Income Tax=",tax)</pre>	<pre>Python 3.7.3 (v3.7.3:ef4ec 4)] on win32 Type "help", "copyright", >>> RESTART: C:\Users\Hp\Docu Enter Income=150000 Income Tax= 7500.0 >>> RESTART: C:\Users\Hp\Docu Enter Income=400000 Income Tax= 30000.0 >>> RESTART: C:\Users\Hp\Docu Enter Income=600000 Income Tax= 60000.0 >>> </pre>
---	--

PRINT-13: S.No.-49
To find age of a person using date and time library

```
##### PROGRAM PRINT 13 #####
#SNO49-To find age using datetime library
#AUTHOR-KONICAA SHARMA

import datetime
dob=eval(input('Enter your date of birth in yyyy-mm-dd format='))
curdate=datetime.date.today()

bd=dob[2]
bm=dob[1]
by=dob[0]

cd=curdate.day
cm=curdate.month
cy=curdate.year

if cd<bd:
    cd=cd+30
    cm=cm-1
if cm<bm:
    cm=cm+12
    cy=cy-1
y=cy-by
m=cm-bm
d=cd-bd
print('Your age is',y,'years',m,'months',d,'days')
```

```
>>>
RESTART: C:\Users\Hp\Desktop\CLASS XII COMPUTER SCIENCE\PYTHON
.py
Enter your date of birth in yyyy-mm-dd format=2003,10,18
Your age is 17 years 0 months 21 days
>>>
```

PRINT-14: S.No.-50

To compare ages of two persons and find who is elder using datetime library

```
##### PROGRAM PRINT-14 #####
#SNo-50:To compare ages of two persons and find who is elder using datetime library
#AUTHOR-KONICAA SHARMA
from age_module import calculate_age
import datetime
print('\nFormat[day,month,year]\n')
dob=eval(input('Enter date of birth of first person:'))
by=dob[2];bm=dob[1];bd=dob[0]
ay1,am1,ad1=calculate_age(bd,bm,by)
print()
dob=eval(input('Enter date of birth of second person:'))
by=dob[2];bm=dob[1];bd=dob[0]
ay2,am2,ad2=calculate_age(bd,bm,by)
print()
s1='First person is elder'
s2='Second person is elder'
```

```
if ay1>ay2:
    print(s1)
elif ay1==ay2:
    if am1>am2:
        print(s1)
    elif am1==am2:
        if ad1>ad2:
            print(s1)
        elif ad1==ad2:
            print("They are of the same age")
        else:
            print(s2)
    else:
        print(s2)
else:
    print(s2)
```

```

#AGE MODULE
import datetime
def calculate_age(bday,bmonth,byear):
    curdate=datetime.date.today()
    cday=curdate.day
    cmonth=curdate.month
    cyear=curdate.year
    if cday<bday:
        cday+=30
        cmonth-=1
    if cmonth<bmonth:
        cmonth+=12
        cyear-=1
    year=cyear-byear
    month=cmonth-bmonth
    day=cday-bday
    print("Age is",year,"years",month,"months",day,"days")
    return(year,month,day)
===== RESTART: C:\Users\Hp\Documents\PRACTICE OF COMPUTER SCIENCE\sno50.p
Format[day,month,year]

Enter date of birth of first person:[18,10,2003]
Age is 17 years 0 months 22 days

Enter date of birth of second person:[6,8,2003]
Age is 17 years 3 months 4 days

Second person is elder
>>>
===== RESTART: C:\Users\Hp\Documents\PRACTICE OF COMPUTER SCIENCE\sno50.p
Format[day,month,year]

Enter date of birth of first person:[18,10,2003]
Age is 17 years 0 months 22 days

Enter date of birth of second person:[19,10,2003]
Age is 17 years 0 months 21 days

First person is elder

```

```
===== RESTART: C:\Users\Hp\Documents\PRACTICE OF COMPUTER SCIENCE\sno5
```

```
Format [day,month,year]
```

```
Enter date of birth of first person:[18,10,2003]  
Age is 17 years 0 months 22 days
```

```
Enter date of birth of second person:[18,10,2003]  
Age is 17 years 0 months 22 days
```

```
They are of the same age
```

PRINT-15: S.No.-51

To create a package in python calculating area, perimeter, volume of different geometric shapes

```
#Area Module
from math import pi
def area_rectangle(length,breadth):
    return length*breadth
def area_square(side):
    return side*side
def area_circle(radius):
    return pi*radius**2
def area_triangle(base,height):
    return (base*height)/2
def area_parallellogram(base,height):
    return (base*height)
def area_cube(s):
    return 6*s**2
def area_sphere(radius):
    return 4*pi*radius**2
```

```
#Perimeter module
from math import pi
def peri_rectangle(length,breadth):
    return 2*(length+breadth)
def peri_square(side):
    return 4*side
def peri_circle(radius):
    return 2*pi*radius
def peri_triangle(base,sidel,side2):
    return base+sidel+side2
def peri_cuboid(sid1,side2):
    return 2*(sid1+side2)
def peri_cube(s):
    return 6*s
```

```
#Volume module
from math import pi
def volume_sphere(radius):
    return (4*pi*radius**3)/3
def volume_cube(side):
    return side**3
def volume_cuboid(l,b,h):
    return l*b*h
```

```
#__init__.py
from Area_module import*
from Perimeter_module import*
from Volume_module import*
```

```

#####
# PROGRAM PRINT-15 #####
#SNo-51:To find Area,Perimeter and volume of different Geometric shapes
#AUTHOR-KONICAA SHARMA

from Geometry import area,perimeter,volume
def area_shapes():
    while True:
        print('Select shape whose area you need to calculate:')
        print("1.Area of Rectangle")
        print("2.Area of Square")
        print("3.Area of Triangle")
        print("4.Area of Circle")
        print("5.Area of Cube")
        print("6.Area of sphere")
        print("7.Back to Main menu")
        print("*****")
        ch=int(input("\nEnter your choice(1-7)?"))
        if ch==1:
            length=int(input("Enter Length="))
            breadth=int(input("Enter Breadth="))
            print(area.area_rectangle(length,breadth))
            print("=====")
        elif ch==2:
            s=int(input("Enter side of square="))
            print(area.area_square(s))
            print("=====")
        elif ch==3:
            base=int(input("Enter base="))
            height=int(input("Enter height="))
            print(area.area_triangle(base,height))
            print("=====")
        elif ch==4:
            r=int(input("Enter radius of circle="))
            print(area.area_circle(r))
            print("=====")
        elif ch==5:
            s=int(input("Enter side of cube="))
            print(area.area_cube(s))
            print("=====")
        elif ch==6:
            r=int(input("Enter radius of sphere="))
            print(area.area_sphere(r))
            print("=====")
        elif ch==7:
            return
        else:
            print("WRONG INPUT")

```

```

def peri_shapes():
    while True:
        print('Select shape whose perimeter you need to calculate:')
        print("1.Perimeter of Rectangle")
        print("2.Perimeter of Square")
        print("3.Perimeter of Triangle")
        print("4.Perimeter of Circle")
        print("5.Perimeter of Cube")
        print("6.Back to Main menu")
        print("#####")

        ch=int(input("\nEnter your choice(1-6)?"))
        if ch==1:
            length=int(input("Enter Length="))
            breadth=int(input("Enter Breadth="))
            print(perimeter.peri_rectangle(length,breadth))
            print("=====")
        elif ch==2:
            s=int(input("Enter side of square="))
            print(perimeter.peri_square(s))
            print("=====")

        elif ch==3:
            sidel=int(input("Enter sidel="))
            side2=int(input("Enter side2="))
            side3=int(input("Enter side3="))
            print(perimeter.peri_triangle(sidel,side2,side3))
            print("=====")
        elif ch==4:
            r=int(input("Enter radius of circle="))
            print(perimeter.peri_circle(r))
            print("=====")
        elif ch==5:
            s=int(input("Enter side of cube="))
            print(perimeter.peri_cube(s))
            print("=====")
        elif ch==6:
            return
        else:
            print("Wrong Choice")
    |
def volume_shapes():
    while True:
        print('Select shape whose volume you need to calculate:')
        print("1.Volume of Cuboid")
        print("2.Volume of Cube")

```

```

print("3.Volume of sphere")
print("4.Back to Main menu")
ch=int(input("\nEnter your choice(1-4)?"))

if ch==1:
    length=int(input("Enter Length="))
    breadth=int(input("Enter Breadth="))
    height=int(input("Enter Height="))
    print(volume.volume_cuboid(length,breadth,height))
    print("=====")
elif ch==2:
    s=int(input("Enter side of cube="))
    print(volume.volume_cube(s))
    print("=====")
elif ch==3:
    r=int(input("Enter radius of sphere="))
    print(volume.volume_sphere(r))
    print("=====")
elif ch==4:
    return
else:
    print("WRONG INPUT")

while True:
    print("*****")
    print('Select your choice:\n1.Area\n2.Perimeter\n3.Volume\n4.EXIT')
    ch=int(input("Enter your choice:"))

    if ch==1:
        area_shapes()
    elif ch==2:
        peri_shapes()
    elif ch==3:
        volume_shapes()
    elif ch==4:
        print("TERMINATING")
        break
    else:
        print("WRONG INPUT")

```

```
>>>
RESTART: C:\Users\Hp\Documents\PRACTICE OF COMPUTER SCIENCE\sno51 prac
*****
Select your choice:
1.Area
2.Perimeter
3.Volume
4.EXIT
Enter your choice:1
Select shape whose area you need to calculate:
1.Area of Rectangle
2.Area of Square
3.Area of Triangle
4.Area of Circle
5.Area of Cube
6.Area of sphere
7.Back to Main menu
*****
Enter your choice(1-7) ?1
Enter Length=1
Enter Breadth=4
4
=====
Select shape whose area you need to calculate:
1.Area of Rectangle
2.Area of Square
3.Area of Triangle
4.Area of Circle
5.Area of Cube
6.Area of sphere
7.Back to Main menu
*****
Enter your choice(1-7) ?2
Enter side of square=12
144
=====
Select shape whose area you need to calculate:
1.Area of Rectangle
2.Area of Square
3.Area of Triangle
4.Area of Circle
5.Area of Cube
6.Area of sphere
7.Back to Main menu
*****
Enter your choice(1-7) ?3
Enter base=5
Enter height=3
7.5
```

```
=====
Select shape whose area you need to calculate:  
1.Area of Rectangle  
2.Area of Square  
3.Area of Triangle  
4.Area of Circle  
5.Area of Cube  
6.Area of sphere  
7.Back to Main menu  
*****
```

```
Enter your choice(1-7)?4  
Enter radius of circle=2  
12.566370614359172  
=====
```

```
Select shape whose area you need to calculate:  
1.Area of Rectangle  
2.Area of Square  
3.Area of Triangle  
4.Area of Circle  
5.Area of Cube  
6.Area of sphere  
7.Back to Main menu  
*****
```

```
Enter your choice(1-7)?5  
Enter side of cube=6  
216  
=====
```

```
Select shape whose area you need to calculate:  
1.Area of Rectangle  
2.Area of Square  
3.Area of Triangle  
4.Area of Circle  
5.Area of Cube  
6.Area of sphere  
7.Back to Main menu  
*****
```

```
Enter your choice(1-7)?6  
Enter radius of sphere=4  
201.06192982974676  
=====
```

```
Select shape whose area you need to calculate:  
1.Area of Rectangle  
2.Area of Square  
3.Area of Triangle  
4.Area of Circle  
5.Area of Cube  
6.Area of sphere  
7.Back to Main menu  
*****
```

```
Enter your choice(1-7)?7
```

```
*****
Select your choice:
1.Area
2.Perimeter
3.Volume
4.EXIT
Enter your choice:2
Select shape whose perimeter you need to calculate:
1.Perimeter of Rectangle
2.Perimeter of Square
3.Perimeter of Triangle
4.Perimeter of Circle
5.Perimeter of Cube
6.Back to Main menu
#####
Enter your choice(1-6)?1
Enter Length=5
Enter Breadth=6
22
=====
Select shape whose perimeter you need to calculate:
1.Perimeter of Rectangle
2.Perimeter of Square
3.Perimeter of Triangle
4.Perimeter of Circle
5.Perimeter of Cube
6.Back to Main menu
#####
Enter your choice(1-6)?2
Enter side of square=4
16
=====
Select shape whose perimeter you need to calculate:
1.Perimeter of Rectangle
2.Perimeter of Square
3.Perimeter of Triangle
4.Perimeter of Circle
5.Perimeter of Cube
6.Back to Main menu
#####
Enter your choice(1-6)?3
Enter sidel=2
Enter side2=3
Enter side3=4
9
=====
Select shape whose perimeter you need to calculate:
```

```

1.Perimeter of Rectangle
2.Perimeter of Square
3.Perimeter of Triangle
4.Perimeter of Circle
5.Perimeter of Cube
6.Back to Main menu
#####
Enter your choice(1-6) ?4
Enter radius of circle=5
31.41592653589793
=====
Select shape whose perimeter you need to calculate:
1.Perimeter of Rectangle
2.Perimeter of Square
3.Perimeter of Triangle
4.Perimeter of Circle
5.Perimeter of Cube
6.Back to Main menu
#####

Enter your choice(1-6) ?5
Enter side of cube=5
30
=====
Select shape whose perimeter you need to calculate:
1.Perimeter of Rectangle
2.Perimeter of Square
3.Perimeter of Triangle
4.Perimeter of Circle
5.Perimeter of Cube
6.Back to Main menu
#####

Enter your choice(1-6) ?6
*****
Select your choice:
1.Area
2.Perimeter
3.Volume
4.EXIT
Enter your choice:3
Select shape whose volume you need to calculate:
1.Volume of Cuboid
2.Volume of Cube
3.Volume of sphere
4.Back to Main menu

Enter your choice(1-4) ?1
Enter Length=3
Enter Breadth=4
Enter Height=5
60
=====
```

```
Select shape whose volume you need to calculate:
```

- 1.Volume of Cuboid
- 2.Volume of Cube
- 3.Volume of sphere
- 4.Back to Main menu

```
Enter your choice(1-4)?2
```

```
Enter side of cube=2
```

```
8
```

```
=====
```

```
Select shape whose volume you need to calculate:
```

- 1.Volume of Cuboid
- 2.Volume of Cube
- 3.Volume of sphere
- 4.Back to Main menu

```
Enter your choice(1-4)?3
```

```
Enter radius of sphere=3
```

```
113.09733552923255
```

```
=====
```

```
Select shape whose volume you need to calculate:
```

- 1.Volume of Cuboid
- 2.Volume of Cube
- 3.Volume of sphere
- 4.Back to Main menu

```
Enter your choice(1-4)?4
```

```
*****
```

```
Select your choice:
```

- 1.Area
- 2.Perimeter
- 3.Volume
- 4.EXIT

```
Enter your choice:5
```

```
WRONG INPUT
```

```
*****
```

```
Select your choice:
```

- 1.Area
- 2.Perimeter
- 3.Volume
- 4.EXIT

```
Enter your choice:4
```

```
TERMINATING
```

```
>>> |
```

PRINT-16: S.No.-54

To calculate volume and area of a sphere inside separate modules and import them in one complete package

```
#AREA
from math import pi
def area(r):
    return 4*pi*r**2
```

```
#VOLUME
from math import pi
def volume(r):
    return (4*pi*r**3)/3
```

```
#__init__.py
from volume_module import *
from area_module import *
```

```
##### PROGRAM PRINT-16 #####
#SNo-54:To calculate volume and area of a sphere inside
#separate modules and import them in one complete package
#AUTHOR-KONICAA SHARMA
```

```
from SNO54 import volume_module,area_module
while True:
    print("====")
    print("\n1.To calculate Area of Sphere")
    print("\n2.To calculate Volume of Sphere")
    print("\n3.EXIT")
    choice=int(input("Enter your choice(1-3):"))
    if choice==1:
        r=int(input("Enter radius="))
        print('AREA is',area_module.area(r))
    elif choice==2:
        r=int(input("Enter radius="))
        print('VOLUME is',volume_module.volume(r))
    elif choice==3:
        print("TERMINATING")
        break
    else:
        print("WRONG CHOICE")
```

```
>>>
===== RESTART: C:\Users\Hp\Documents\PRACTICE OF COMPUTER SCIENCE\sno5
=====
1.To calculate Area of Sphere
2.To calculate Volume of Sphere
3.EXIT
Enter your choice(1-3):1
Enter radius=4
AREA is 201.06192982974676
=====
1.To calculate Area of Sphere
2.To calculate Volume of Sphere
3.EXIT
Enter your choice(1-3):2
Enter radius=4
VOLUME is 268.082573106329
=====

1.To calculate Area of Sphere
2.To calculate Volume of Sphere
3.EXIT
Enter your choice(1-3):4
WRONG CHOICE
=====
1.To calculate Area of Sphere
2.To calculate Volume of Sphere
3.EXIT
Enter your choice(1-3):3
TERMINATING
>>> |
```

PRINT-17: S.No.-62

To check number of lines starting with an uppercase letter in text1.txt

```
##### PROGRAM PRINT-17 #####
#SNo-62:To check number of lines starting with an uppercase letter in test1.txt
#AUTHOR-KONICAA SHARMA

myfile=open("Test1.txt","r")
n=0
ch=' '
while ch:
    ch=myfile.read(1)
    if (ch.isupper()):
        n=n+1
    str1 myfile.readline()
print("Total number of lines starting with uppercase letters are=",n)
myfile.close()
```

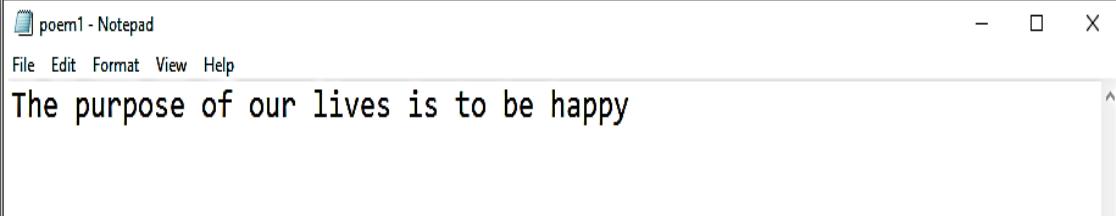


```
>>>
===== RESTART: C:\Users\Hp\Documents\DATA FILE HANDLING\sno-62.py =====
Total number of lines starting with uppercase letters are= 3
>>>
```

PRINT-18: S.No.-63
To count number of vowels in a text file

```
##### PROGRAM PRINT-18 #####
#SNo-63:To count the number of vowels in a text file
#AUTHOR-KONICAA SHARMA

fnm=input("Enter the name of the text file in which you wish to count number of vowels=")
myfile=open(fnm)
n=0
vowels=['a','e','i','o','u','A','E','I','O','U']
for str1 in myfile:
    for ch in str1:
        if ch in vowels:
            n=n+1
print("Total vowels in ",fnm,"=",n)
myfile.close()
```



```
>>>
===== RESTART: C:\Users\Hp\Documents\DATA FILE HANDLING\sno63.py ======
Enter the name of the text file in which you wish to count number of vowels=poem1.txt
Total vowels in poem1.txt = 13
>>> |
```

PRINT-19: S.No.-65

To write sequence of data like strings ,tuples ,list in test1.txt and append using writelines () and open file using ‘with’ statement

```
##### PROGRAM PRINT-19 #####
#SNo-65:To write sequence of data like strings,tuples
#list in test1.txt and append using writelines ()
#and open file using 'with' statement
#AUTHOR-KONICAA SHARMA

with open("Test1.txt","a") as myfile:
    while True:
        s1=input("Enter your favourite cities in India:")
        myfile.writelines(s1+'\n')
        ch=input("Do you want to enter more(y/n)?")
        myfile.flush()
        if ch in 'nN':
            break
myfile.close()
file=open("test1.txt")
print("\nDATA OF TEXT FILE....\n")
for i in file:
    print(i)
file.close()
```

```
>>>
===== RESTART: C:\Users\Hp\Documents\DATA FILE HANDLING\sno-65.py ======
Enter your favourite cities in India:'MUMBAI'
Do you want to enter more(y/n)?Y
Enter your favourite cities in India:['DELHI']
Do you want to enter more(y/n)?Y
Enter your favourite cities in India:('KANPUR')
Do you want to enter more(y/n)?N

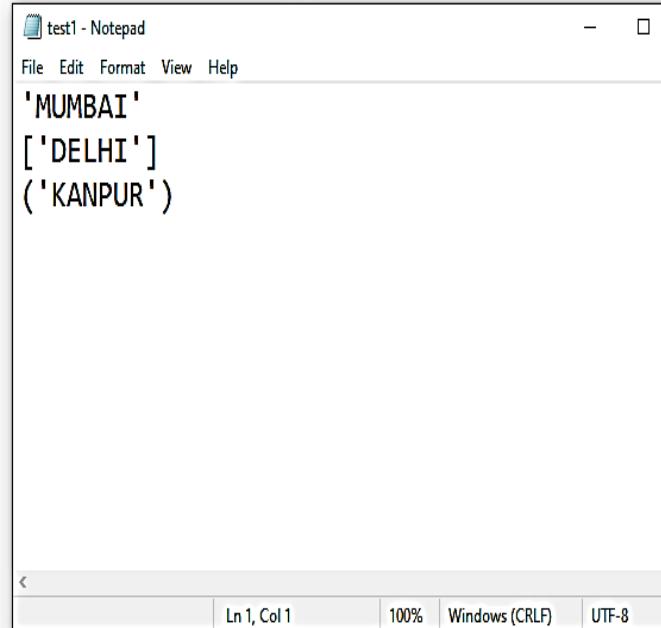
DATA OF TEXT FILE.....

'MUMBAI'

['DELHI']

('KANPUR')

>>>
```



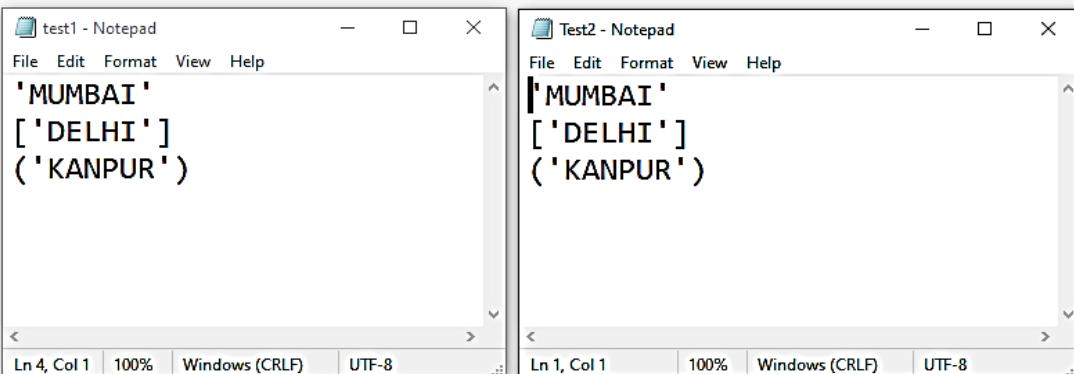
PRINT-20: S.No. -66
To copy contents of one text file to another

```
##### PROGRAM PRINT-20 #####
#SNo66-To copy contents of one text file to another
#AUTHOR-KONICAA SHARMA

myfile1=open("Test1.txt")
myfile2=open("test2.txt","w+")
str1=" "
while(str1):
    str1=myfile1.readline()
    myfile2.write(str1)
myfile1.close()
myfile2.seek(0)
str1=myfile2.read()
print("Content copied in another text file.....")
print(str1)
myfile2.close()
```

```
>>>
===== RESTART: C:\Users\Hp\Documents\DATA FILE HANDLING\sno66.py ======
Content copied in another text file.....
'MUMBAI'
['DELHI']
('KANPUR')
```

```
>>>
```



PRINT-21: S.No.-78
To write and read records of binary file containing string

```
##### PROGRAM PRINT-21 #####
#SNo-78:To write and read records of binary file containing string
#AUTHOR-KONICAA SHARMA

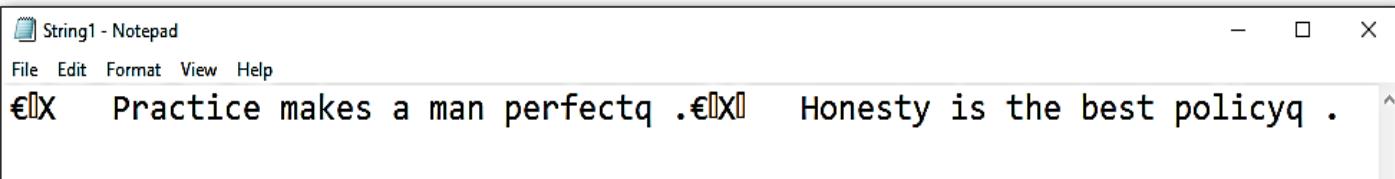
import pickle
ans='y'
myfile=open('String1.dat','wb+')
while ans in 'yY':
    s=input("Enter string=")
    pickle.dump(s myfile)
    ans=input("Do you want to enter more strings(y/n)?")
myfile.seek(0)#0 means beginning,1 means current position,2 means end
print("\nAll records of String1.dat....\n")
try:
    while True:
        st=pickle.load(myfile)
        print(st)
except EOFError:
    print("\nFile Operation Successful")
myfile.close()
```

```
>>>
=====
RESTART: C:\Users\Hp\Documents\DATA FILE HANDLING\sno78.py =====
Enter string=Practice makes a man perfect
Do you want to enter more strings(y/n)?y
Enter string=Honesty is the best policy
Do you want to enter more strings(y/n)?n

All records of String1.dat.....

Practice makes a man perfect
Honesty is the best policy

File Operation Successful
>>>
```



The screenshot shows a Notepad window titled "String1 - Notepad". The window contains two lines of text: "Practice makes a man perfect" and "Honesty is the best policy". The window has standard operating system window controls (minimize, maximize, close) at the top right.

PRINT-22: S.No.-79

To write and read records of employees in binary file using tuples

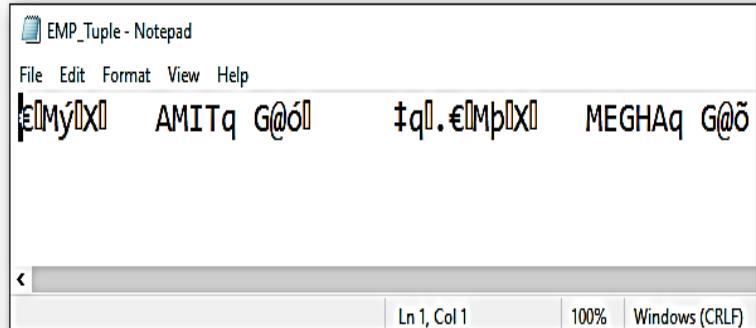
```
##### PROGRAM PRINT-22 #####
#SNo-79:To write and read records of employee
#in binary file using tuples
#AUTHOR-KONICAA SHARMA

import pickle
myfile=open("EMP_Tuple.dat","wb+")
ans='yY'
while ans in 'yY':
    empcode=int(input("Enter Employee code="))
    name=input("Enter employee name=")
    basic=float(input("Enter basic salary="))
    data=(empcode,name,basic)
    pickle.dump(data,myfile)
    ans=input("Do yo want to enter more records(y/n)=")
myfile.seek(0)
print("\nSuccessfully written records in Employee.....\n")
try:
    while True:
        st=pickle.load(myfile)
        print(st)
except EOFError:
    print("\nFile Operation successful\n")

myfile.close()
```

```
===== RESTART: C:/Users/Hp/Documents/DATA FILE HANDLING/sno79.py =====
```

```
Enter Employee code=1021
Enter employee name=AMIT
Enter basic salary=78000
Do yo want to enter more records(y/n)=Y
Enter Employee code=1022
Enter employee name=MEGHA
Enter basic salary=88000
Do yo want to enter more records(y/n)=N
```



```
Successfully written records in Employee.....
```

```
(1021, 'AMIT', 78000.0)
(1022, 'MEGHA', 88000.0)
```

```
File Operation successful
```

PRINT-23: S.No.-80

To write and read records of student in binary file containing lists

```
##### PROGRAM PRINT-23 #####
#SNo-80:To write and read records of student in binary file containing list
#AUTHOR-KONICAA SHARMA

import pickle
ans="y"
myfile=open('List1.dat','wb+')
while ans in 'yY':
    rno=int(input("Enter roll no="))
    name=input("Enter name=")
    tmarks=int(input("Enter total marks="))
    data=[rno,name,tmarks]
    pickle.dump(data myfile)
    ans=input("Enter do you want to enter more data(y/n)?")
myfile.seek(0)
print("\nAll records of List1.dat...\n")
try:
    while True:
        st=pickle.load(myfile)
        print(st)
except EOFError:
    print("\nFile Operation Successful")
myfile.close()
```

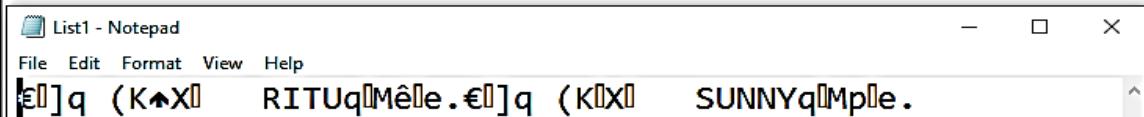
```
>>>
===== RESTART: C:\Users\Hp\Documents\DATA FILE HANDLING\sno80.py =
Enter roll no=12
Enter name=RITU
Enter total marks=490
Enter do you want to enter more data(y/n)?Y
Enter roll no=18
Enter name=SUNNY
Enter total marks=368
Enter do you want to enter more data(y/n)?N

All records of List1.dat...

[12, 'RITU', 490]
[18, 'SUNNY', 368]
```

File Operation Successful

>>>



PRINT-24: S.No.-83

To write, read, modify and delete records of employee in ‘emp.dat’ and to find employee with highest salary using dictionary

```
##### PROGRAM PRINT-24 #####
#SNo-83:To add,display,modify and delete records of employee in
#emp.dat and to find Employee with highest salary using dictionary
#AUTHOR-KONICAA SHARMA

import pickle
import os
def Create_append():
    myfile=open('emp.dat', 'ab')
    em={}
    ans='y'
    while ans in 'yY':
        eno=int(input("Enter Emp code="))
        name=input("Enter name of employee=")
        age=int(input("Enter age="))
        sal=int(input("Enter salary of employee"))

        em['Empno']=eno
        em['Ename']=name
        em['Age']=age
        em['Salary']=sal

        pickle.dump(em,myfile)
        ans=input("Do you want to append more records(y/n)=")
        print("Successfully written records")
    myfile.close()
```

```
def Display():
    try:
        myfile=open("emp.dat", "rb")
        try:
            while True:
                data=pickle.load(myfile)
                print(data)
            print("That's all")
        except EOFError:
            myfile.close()
    except FileNotFoundError:
        print("File does not exist")

def Delete():
    eid=int(input("Enter Employee id you want to delete="))
    flag=0
    try:
        myfile=open("emp.dat", "rb")
        myfile2=open("temporary.dat", "wb")
```

```

eid=int(input("Enter Employee id you want to modify"))
flag=0
try:
    while True:
        data=pickle.load(myfile)
        if data['Empno']==eid:
            flag=1
            print(data)
            print("Enter new details of this employee=")
            name=input("Enter Employee name=")
            age=int(input("Enter age="))
            sal=int(input("Enter salary="))

            data['Ename']=name
            data['Age']=age
            data['Salary']=sal
            pickle.dump(data,myfile2)

except EOFError:
    myfile.close()
    myfile2.close()
    if flag==0:
        print(eid,"not found in emp.dat")
        os.remove("emp.dat")
        os.rename("temp.dat","emp.dat")
except FileNotFoundError:
    print("File does not exist")

```

```

def Delete():
    eid=int(input("Enter Employee id you want to delete"))
    flag=0
    try:
        myfile=open("emp.dat","rb")
        myfile2=open("temporary.dat","wb")
        try:
            while True:
                data=pickle.load(myfile)
                if data['Empno']==eid:
                    flag=1
                    print("The record has been successfully deleted")
                else:
                    pickle.dump(data,myfile2)
        except EOFError:
            if flag==0:
                print(eid,"not found in emp.dat")

```

```

        myfile.close()
        myfile2.close()
        os.remove("emp.dat")
        os.rename("temporary.dat","emp.dat")
    except FileNotFoundError:
        print("File does not exist")

def maxsalary():
    try:
        myfile=open("emp.dat","rb")
        try:
            data=pickle.load(myfile)
            max=data.copy()
            while True:
                data=pickle.load(myfile)
                if max['Salary']<data['Salary']:
                    max=data.copy()
        except EOFError:
            print('Employee with highest salary....')
            print(max)
            myfile.close()
    except FileNotFoundError:
        print('File does not exist...so please create it first')

#####MAIN PROGRAM#####
while True:
    print("*****")
    print("\n1.To create/append records of employee")
    print("\n2.To display records of employee")
    print("\n3.To modify records of employee")
    print("\n4.To delete records of employee")
    print("\n5.To display employee with highest salary")
    print("\n6.Exit")
    ch=int(input("Enter your choice(1-6)="))
    if ch==1:
        Create_append()
    elif ch==2:
        Display()
    elif ch==3:
        Modify()
    elif ch==4:
        Delete()
    elif ch==5:
        maxsalary()
    elif ch==6:
        print("Terminating")
        break
    else:
        print("WRONG INPUT")

```

```
===== RESTART: C:\Users\Hp\Documents\DATA FILE HANDLING\sno83.py =====
*****
```

- 1.To create/append records of employee
- 2.To display records of employee
- 3.To modify records of employee
- 4.To delete records of employee
- 5.To display employee with highest salary

6.Exit

Enter your choice(1-6)=2

```
{'Empno': 1202, 'Ename': 'Anushka', 'Age': 30, 'Salary': 48000}
{'Empno': 1211, 'Ename': 'Simarjeet', 'Age': 35, 'Salary': 49000}
{'Empno': 1222, 'Ename': 'Alex', 'Age': 29, 'Salary': 50000}
{'Empno': 1291, 'Ename': 'Rajat', 'Age': 42, 'Salary': 68000}
{'Empno': 1092, 'Ename': 'Harshita', 'Age': 28, 'Salary': 76000}
{'Empno': 1221, 'Ename': 'Megha', 'Age': 41, 'Salary': 90000}
*****
```

- 1.To create/append records of employee

- 2.To display records of employee

- 3.To modify records of employee

- 4.To delete records of employee

- 5.To display employee with highest salary

6.Exit

Enter your choice(1-6)=1

Enter Emp code=1099

Enter name of employee=Amita

Enter age=39

Enter salary of employee=50000

Do you want to append more records(y/n)=n

Succesfully written records

- 1.To create/append records of employee

- 2.To display records of employee

- 3.To modify records of employee

- 4.To delete records of employee

```
5.To display employee with highest salary
```

```
6.Exit
```

```
Enter your choice(1-6)=2
```

```
{'Empno': 1202, 'Ename': 'Anushka', 'Age': 30, 'Salary': 48000}  
{'Empno': 1211, 'Ename': 'Simarjeet', 'Age': 35, 'Salary': 49000}  
{'Empno': 1222, 'Ename': 'Alex', 'Age': 29, 'Salary': 50000}  
{'Empno': 1291, 'Ename': 'Rajat', 'Age': 42, 'Salary': 68000}  
{'Empno': 1092, 'Ename': 'Harshita', 'Age': 28, 'Salary': 76000}  
{'Empno': 1221, 'Ename': 'Megha', 'Age': 41, 'Salary': 90000}  
{'Empno': 1099, 'Ename': 'Amita', 'Age': 39, 'Salary': 50000}  
*****
```

```
1.To create/append records of employee
```

```
2.To display records of employee
```

```
3.To modify records of employee
```

```
4.To delete records of employee
```

```
5.To display employee with highest salary
```

```
6.Exit
```

```
Enter your choice(1-6)=3
```

```
Enter Employee id you want to modify=8389
```

```
8389 not found in emp.dat
```

```
1.To create/append records of employee
```

```
2.To display records of employee
```

```
3.To modify records of employee
```

```
4.To delete records of employee
```

```
5.To display employee with highest salary
```

```
6.Exit
```

```
Enter your choice(1-6)=3
```

```
Enter Employee id you want to modify=1221
```

```
{'Empno': 1221, 'Ename': 'Megha', 'Age': 41, 'Salary': 90000}
```

```
Enter new details of this employee=
```

```
Enter Employee name=Mansha
```

```
Enter age=34
```

```
Enter salary=78000
```

```
*****
```

- 1.To create/append records of employee
- 2.To display records of employee
- 3.To modify records of employee
- 4.To delete records of employee
- 5.To display employee with highest salary
- 6.Exit

Enter your choice(1-6)=2

```
{'Empno': 1202, 'Ename': 'Anushka', 'Age': 30, 'Salary': 48000}  
{'Empno': 1211, 'Ename': 'Simarjeet', 'Age': 35, 'Salary': 49000}  
{'Empno': 1222, 'Ename': 'Alex', 'Age': 29, 'Salary': 50000}  
{'Empno': 1291, 'Ename': 'Rajat', 'Age': 42, 'Salary': 68000}  
{'Empno': 1092, 'Ename': 'Harshita', 'Age': 28, 'Salary': 76000}  
{'Empno': 1221, 'Ename': 'Mansha', 'Age': 34, 'Salary': 78000}  
{'Empno': 1099, 'Ename': 'Amita', 'Age': 39, 'Salary': 50000}
```

```
*****
```

```
*****
```

- 1.To create/append records of employee
- 2.To display records of employee
- 3.To modify records of employee
- 4.To delete records of employee
- 5.To display employee with highest salary

6.Exit

Enter your choice(1-6)=4

Enter Employee id you want to delete=1099

The record has been successfully deleted

```
*****
```

```
*****
```

- 1.To create/append records of employee
- 2.To display records of employee
- 3.To modify records of employee
- 4.To delete records of employee
- 5.To display employee with highest salary

6.Exit

Enter your choice(1-6)=2

```
{'Empno': 1202, 'Ename': 'Anushka', 'Age': 30, 'Salary': 48000}  
{'Empno': 1211, 'Ename': 'Simarjeet', 'Age': 35, 'Salary': 49000}  
{'Empno': 1222, 'Ename': 'Alex', 'Age': 29, 'Salary': 50000}  
{'Empno': 1291, 'Ename': 'Rajat', 'Age': 42, 'Salary': 68000}  
{'Empno': 1092, 'Ename': 'Harshita', 'Age': 28, 'Salary': 76000}  
{'Empno': 1221, 'Ename': 'Mansha', 'Age': 34, 'Salary': 78000}
```

```
*****
```

```
*****
```

- 1.To create/append records of employee
- 2.To display records of employee
- 3.To modify records of employee
- 4.To delete records of employee
- 5.To display employee with highest salary

6.Exit

Enter your choice(1-6)=4

Enter Employee id you want to delete=1029

1029 not found in emp.dat

```
*****
```

```
*****
```

- 1.To create/append records of employee
- 2.To display records of employee
- 3.To modify records of employee
- 4.To delete records of employee
- 5.To display employee with highest salary

6.Exit

Enter your choice(1-6)=5

Employee with highest salary....

```
{'Empno': 1221, 'Ename': 'Mansha', 'Age': 34, 'Salary': 78000}
```

```
*****
```

- 1.To create/append records of employee
- 2.To display records of employee
- 3.To modify records of employee
- 4.To delete records of employee
- 5.To display employee with highest salary

6.Exit

Enter your choice(1-6)=8

WRONG INPUT

```
*****
```

- 1.To create/append records of employee
- 2.To display records of employee
- 3.To modify records of employee
- 4.To delete records of employee
- 5.To display employee with highest salary

6.Exit

Enter your choice(1-6)=6

Terminating

>>>

PRINT-25: S.No.-84
Library management using list

```
##### PROGRAM PRINT-25 #####
#SNo-84-LIBRARY MANAGEMENT USING LIST
#AUTHOR-KONICAA SHARMA

import pickle
import os
def Add():
    libfile=open("Library.dat",'ab')
    while True:
        bno=int(input("Enter book accession number="))
        bnm=input("Enter book name=")
        author=input("Enter author name=")
        pub=input("Enter publisher=")
        gen=input("Enter genre of book=")
        price=int(input("Enter price="))
        copies=int(input("Enter copies="))
        issued=0
        lib=[bno, bnm, author, pub, gen, price, copies, issued]
        pickle.dump(lib, libfile)
        ans=input("Do you want to add more(y/n)?")
        if ans not in 'yY':
            break
    libfile.close()

def display_author():
    author=input("Enter author name whose books you are looking for=")
    found=0
    try:
        libfile=open('Library.dat','rb')
        try:
            while True:
                lib=pickle.load(libfile)
                if (lib[2].lower()==author.lower()):
                    found=1
                    print("Book No\t Book Name\t Book Author\t"
                          "Book Publisher\t Genre\t\tPrice\tCopies\tIssued")
                    print('\n',lib[0],'\t',lib[1],'\t',lib[2],'\t',lib[3],'\t',lib[4],'\t',
                          lib[5],'\t',lib[6],'\t',lib[7])
        except EOFError:
            if found==0:
                print("Author not found")
        libfile.close()
    except FileNotFoundError:
        print("File does not exist so create it first and add records in it")
```

```

def display_pub():
    pub=input("Enter publisher whose book you are looking for=")
    found=0
    try:
        libfile=open('Library.dat','rb')
        try:
            while True:
                lib=pickle.load(libfile)
                if (lib[3].lower()==pub.lower()):
                    found=1
                print('''Book No\t Book Name\t Book Author\t
                      Book Publisher\t Genre\t\tPrice\tCopies\tIssued''')
                print('\n',lib[0],'\t',lib[1],'\t',lib[2],'\t',lib[3],'\t',lib[4],'\t',
                      lib[5],'\t',lib[6],'\t',lib[7])
        except EOFError:
            if found==0:
                print("Publisher not found")
        libfile.close()
    except FileNotFoundError:
        print("File does not exist so create it first and add records in it")

```

```

def display_all():
    try:
        libfile=open("Library.dat","rb")
        try:
            print('''Book No\t Book Name\t Book Author\t
                  Book Publisher\t Genre\t\tPrice\tCopies\tIssued''')
            while True:
                lib=pickle.load(libfile)
                print(lib[0],'\t',lib[1],'\t',lib[2],'\t',lib[3],'\t',lib[4],'\t',
                      lib[5],'\t',lib[6],'\t',lib[7])
        except EOFError:
            libfile.close()
    except FileNotFoundError:
        print("File does not exist so create and add records in it ")

```

```

def display_accno():
    acno=int(input("Enter accession number you are looking for"))
    try:
        libfile=open("Library.dat",'rb')
        try:

```

```

        while True:
            lib=pickle.load(libfile)
            if (lib[0]==acno):
                print('''Book No\t Book Name\t Book Author\t
                      Book Publisher\t Genre\t\tPrice\tCopies\tIssued''')
                print(lib[0],'\t',lib[1],'\t',lib[2],'\t',lib[3],'\t',lib[4],'\t',
                      lib[5],'\t',lib[6],'\t',lib[7])
                break
            except EOFError:
                print("Book not found")
            libfile.close()
        except FileNotFoundError:
            print("File does not exist so create and add records in it")

def display_genre():
    genre=input("Enter genre you are looking for=")
    found=0
    try:
        libfile=open("Library.dat",'rb')
        try:
            while True:
                lib=pickle.load(libfile)
                if (lib[4].lower()==genre.lower()):
                    found=1
                print('''Book No\t Book Name\t Book Author\t
                      Book Publisher\t Genre\t\tPrice\tCopies\tIssued''')
                print(lib[0],'\t',lib[1],'\t',lib[2],'\t',lib[3],'\t',
                      lib[4],'\t',lib[5],'\t',lib[6],'\t',lib[7])
        except EOFError:
            if found==0:
                print("Book with Genre",genre," not found")
            libfile.close()
    except FileNotFoundError:
        print("File does not exist so create and add records in it")

def Issue():
    print()
    Display_menu()
    print()
    flag=0
    acno=int(input("Enter book accession number="))
    try:
        libfile=open("Library.dat","rb")
        libfile2=open("temp.dat","wb")
        try:
            while True:
                lib=pickle.load(libfile)
                if lib[0]==acno:
                    flag=1

```

```

        if lib[6]>lib[7]:
            lib[7]=lib[7]+1
            lib[6]=lib[6]-1
            print("Book is issued")
        else:
            print("All copies of this book are already issued")
            pickle.dump(lib,libfile2)
    except EOFError:
        if flag==0:
            print("Wrong accession number entered")
    libfile.close()
    libfile2.close()
    os.remove("Library.dat")
    os.rename("temp.dat","Library.dat")
except FileNotFoundError:
    print("File does not exist so create and add records in it")

```

```

def Return():
    print()
    flag=0
    acno=int(input("Enter book accession number you want to return="))
    try:
        libfile=open("Library.dat","rb")
        libfile2=open("temp.dat", "wb")
        try:
            while True:
                lib=pickle.load(libfile)
                if lib[0]==acno:
                    flag=1
                    if lib[7]>0:
                        lib[7]=lib[7]-1
                        lib[6]=lib[6]+1
                        print("Book is returned")
                    else:
                        print("No copy of this was issued")

                pickle.dump(lib,libfile2)
        except EOFError:
            if flag==0:
                print("Wrong accession number entered")
        libfile.close()
        libfile2.close()
        os.remove("Library.dat")
        os.rename("temp.dat","Library.dat")
    except FileNotFoundError:
        print("File does not exist so create and add records in it")

```

```

def Delete():
    print()
    Display_menu()
    print()
    flag=0
    acno=int(input("Enter book accession number you want to delete="))
    try:
        libfile=open("Library.dat", "rb")
        libfile2=open("temp.dat", "wb")
        try:
            while True:
                lib=pickle.load(libfile)
                if lib[0]==acno:
                    flag=1
                    print("Book with Accession no", acno, "is deleted")
                else:
                    pickle.dump(lib, libfile2)
        except EOFError:
            if flag==0:
                print("Wrong accession number entered")
        libfile.close()
        libfile2.close()
        os.remove("Library.dat")
        os.rename("temp.dat", "Library.dat")
    except FileNotFoundError:
        print("File does not exist so create and add records in it")

```

```

def Update():
    flag=0;ans='y'
    print()
    Display_menu()
    print()
    acno=int(input("Enter book accession number="))
    try:
        libfile=open("Library.dat", "rb")
        libfile2=open("temp.dat", "wb")
        try:
            while True:
                lib=pickle.load(libfile)
                if lib[0]==acno:
                    flag=1
                    print(lib)

```

```

while True:
    ch=int(input("You want to 1.Add or 2.Remove copies of the book or 3.EXIT-"))
    if ch==1:
        while ans in 'yY':
            n=int(input("Enter number of copies you want to add="))
            lib[6]+=n
            ans=input("Do you want to add more books(y/n):")

    elif ch==2:
        while ans in 'yY':
            n=int(input("Enter number of copies you want to remove="))
            if (n<=lib[6] and n<=lib[6]-lib[7]):
                lib[6]-=n
            else:
                print("Your number of copies are more than available right now,i.e.",lib[6]>lib[7])
                break
            ans=input("Do you want to remove more books(y/n):")
    elif ch==3:
        break
    else:
        print("WRONG CHOICE")
pickle.dump(lib,libfile2)

```

```

except EOFError:
    if flag==0:
        print("Wrong accession number entered")
    libfile.close()
    libfile2.close()
    os.remove("Library.dat")
    os.rename("temp.dat","Library.dat")
except FileNotFoundError:
    print("File does not exist so create and add records in it")

def display_title():
    title=input("Enter title of the book you are looking for=")
    found=0
    try:
        libfile=open("Library.dat",'rb')
        try:
            while True:
                lib=pickle.load(libfile)
                if (lib[1].lower()==title.lower()):
                    found=1
                    print('''Book No\t Book Name\t Book Author\t Book Publisher
                           \t Genre\t\tPrice\tCopies\tIssued''')
                    print(lib[0],'\t',lib[1],'\t',lib[2],'\t',lib[3]
                          ,'\t',lib[4],'\t',lib[5],'\t',lib[6])
        except EOFError:
            if found==0:
                print(title," not found in library")
        libfile.close()
    except FileNotFoundError:
        print("File does not exist so create and add records in it")

```

```

def Display_menu():
    while True:
        print("*****")
        print("1.All books")
        print("2.Particular accession number")
        print("3.Particular genre")
        print("4.Particular author")
        print("5.Particular publisher")
        print("6.Specific title")
        print("7.Go back")
        choice=int(input("Enter your choice(1-7)="))
        if choice==1:
            display_all()
        elif choice==2:
            display_accno()
        elif choice==3:
            display_genre()
        elif choice==4:
            display_author()
        elif choice==5:
            display_pub()
        elif choice==6:
            display_title()
        elif choice==7:
            return

```

```

##### MAIN PROGRAM #####
while True:
    print("\n1.Add book in library")
    print("2.Display books")
    print("3.Issue book")
    print("4.Return book")
    print("5.Delete book")
    print("6.Update book")
    print("7.Exit")
    ch=int(input("Enter your choice(1-7)="))
    if ch==1:
        Add()
    elif ch==2:
        Display_menu()
    elif ch==3:
        Issue()
    elif ch==4:
        Return()
    elif ch==5:
        Delete()
    elif ch==6:
        Update()
    elif ch==7:
        print("Okay byee!!!")
        break
    else:
        print("Invalid choice")

```

```

1.Add book in library
2.Display books
3.Issue book
4.Return book
5.Delete book
6.Update book
7.Exit
Enter your choice(1-7)=1
Enter book accession number=1054
Enter book name=Cindrella
Enter author name=Daisy Fisher
Enter publisher=Scribner
Enter genre of book=fiction
Enter price=500
Enter copies=10
Do you want to add more(y/n)?n

```

```

1.Add book in library
2.Display books
3.Issue book
4.Return book
5.Delete book
6.Update book
7.Exit
Enter your choice(1-7)=2
*****

```

1.All books

2.Particular accession number

3.Particular genre

4.Particular author

5.Particular publisher

6.Specific title

7.Go back

Enter your choice(1-7)=1

Book No	Book Name	Book Author	Book Publisher	Genre	Price	Copies	Issued
1021	Python	Preeti Arora	Sultan Chand	academics	400	2	0
1092	R.D.Sharma	R.D.Sharma	Dhanpat Rai	academics	800	8	0
1026	Dracula	Bram Stoker	Robinson	horror	500	18	0
1056	Pradeep	Pradeep	RK.Sharma	academics	900	12	0
1054	Cindrella	Daisy Fisher	Scribner	fiction	500	10	0

1.All books

2.Particular accession number

3.Particular genre

4.Particular author

5.Particular publisher

6.Specific title

7.Go back

Enter your choice(1-7)=2

Enter accession number you are looking for=1021

Book No	Book Name	Book Author	Book Publisher	Genre	Price	Copies	Issued
1021	Python	Preeti Arora	Sultan Chand	academics	400	2	0

```
*****
1.All books
2.Particular accession number
3.Particular genre
4.Particular author
5.Particular publisher
6.Specific title
7.Go back
Enter your choice(1-7)=2
Enter accession number you are looking for=1009
Book not found
*****
1.All books
2.Particular accession number
3.Particular genre
4.Particular author
5.Particular publisher
6.Specific title
7.Go back
Enter your choice(1-7)=3
Enter genre you are looking for=horror
Book No Book Name Book Author Book Publisher Genre Price Copies Issued
1026 Dracula Bram Stoker Robinson horror 500 18 0
*****
1.All books
2.Particular accession number
3.Particular genre
4.Particular author
5.Particular publisher
6.Specific title
7.Go back
Enter your choice(1-7)=3
Enter genre you are looking for=Thriller
Book with Genre Thriller not found
*****
1.All books
2.Particular accession number
3.Particular genre
4.Particular author
5.Particular publisher
6.Specific title
7.Go back
Enter your choice(1-7)=4
Enter author name whose books you are looking for=Preeti Arora
Book No Book Name Book Author Book Publisher Genre Price Copies Issued
1021 Python Preeti Arora Sultan Chand academics 400 2 0
*****
1.All books
2.Particular accession number
3.Particular genre
4.Particular author
5.Particular publisher
6.Specific title
7.Go back
Enter your choice(1-7)=4
Enter author name whose books you are looking for=Benjamin Franklin
Author not found
```

```
1.All books
2.Particular accession number
3.Particular genre
4.Particular author
5.Particular publisher
6.Specific title
7.Go back
```

Enter your choice(1-7)=5

Enter publisher whose book you are looking for=Robine
Publisher not found

```
1.All books
2.Particular accession number
3.Particular genre
4.Particular author
5.Particular publisher
6.Specific title
7.Go back
```

Enter your choice(1-7)=6

Enter title of the book you are looking for=S.L.Arora
S.L.Arora not found in library

```
1.All books
2.Particular accession number
3.Particular genre
4.Particular author
5.Particular publisher
6.Specific title
7.Go back
```

Enter your choice(1-7)=6

Enter title of the book you are looking for=Pradeep

Book No	Book Name	Book Author	Book Publisher	Genre	Price	Copies	Issued
1056	Pradeep	Pradeep	RK.Sharma	academics	900	12	

```
1.All books
2.Particular accession number
3.Particular genre
4.Particular author
5.Particular publisher
6.Specific title
7.Go back
```

Enter your choice(1-7)=7

```
1.Add book in library
2.Display books
3.Issue book
4.Return book
5.Delete book
6.Update book
7.Exit
```

Enter your choice(1-7)=3

- 1.All books
- 2.Particular accession number
- 3.Particular genre
- 4.Particular author
- 5.Particular publisher
- 6.Specific title
- 7.Go back

Enter your choice(1-7)=3

Enter genre you are looking for=fiction

Book No	Book Name	Book Author	Book Publisher	Genre	Price	Copies	Issued
1054	Cindrella	Daisy Fisher	Scribner	fiction	500	10	0

- 1.All books
- 2.Particular accession number
- 3.Particular genre
- 4.Particular author
- 5.Particular publisher
- 6.Specific title
- 7.Go back

Enter your choice(1-7)=7

Enter book accession number=1054

Book is issued

- 1.Add book in library
- 2.Display books
- 3.Issue book
- 4.Return book
- 5.Delete book
- 6.Update book
- 7.Exit

Enter your choice(1-7)=2

- 1.All books
- 2.Particular accession number
- 3.Particular genre
- 4.Particular author
- 5.Particular publisher
- 6.Specific title
- 7.Go back

Enter your choice(1-7)=2

Enter accession number you are looking for=1054

Book No	Book Name	Book Author	Book Publisher	Genre	Price	Copies	Issued
1054	Cindrella	Daisy Fisher	Scribner	fiction	500	9	1

- 1.All books
- 2.Particular accession number
- 3.Particular genre
- 4.Particular author
- 5.Particular publisher
- 6.Specific title
- 7.Go back

Enter your choice(1-7)=7

1.Add book in library

2.Display books

3.Issue book

4.Return book

5.Delete book

6.Update book

7.Exit

Enter your choice(1-7)=4

Enter book accession number you want to return=1045

Wrong accession number entered

1.Add book in library

2.Display books

3.Issue book

4.Return book

5.Delete book

Enter your choice(1-7)=4

Enter book accession number you want to return=1054

Book is returned

1.Add book in library

2.Display books

3.Issue book

4.Return book

5.Delete book

6.Update book

7.Exit

Enter your choice(1-7)=2

1.All books

2.Particular accession number

3.Particular genre

4.Particular author

5.Particular publisher

6.Specific title

7.Go back

Enter your choice(1-7)=2

Enter accession number you are looking for=1054

Book No	Book Name	Book Author	Book Publisher	Genre	Price	Copies	Issued
1054	Cindrella	Daisy Fisher	Scribner	fiction	500	10	0

1.All books

2.Particular accession number

3.Particular genre

4.Particular author

5.Particular publisher

6.Specific title

Enter your choice(1-7)=7

- 1.Add book in library
- 2.Display books
- 3.Issue book
- 4.Return book
- 5.Delete book
- 6.Update book
- 7.Exit

Enter your choice(1-7)=5

- 1.All books
- 2.Particular accession number
- 3.Particular genre
- 4.Particular author
- 5.Particular publisher
- 6.Specific title
- 7.Go back

Enter your choice(1-7)=1

Book No	Book Name	Book Author	Book Publisher	Genre	Price	Copies	Issue
1021	Python	Preeti Arora	Sultan Chand	academics	400	2	0
1092	R.D.Sharma	R.D.Sharma	Dhanpat Rai	academics	800	8	0
1026	Dracula	Bram Stoker	Robinson	horror	500	18	0
1056	Pradeep	Pradeep	RK.Sharma	academics	900	12	0
1054	Cindrella	Daisy Fisher	Scribner	fiction	500	10	0

- 1.All books
- 2.Particular accession number
- 3.Particular genre
- 4.Particular author
- 5.Particular publisher
- 6.Specific title
- 7.Go back

Enter your choice(1-7)=7

Enter book accession number you want to delete=1021

Book with Accession no 1021 is deleted

- 1.Add book in library
- 2.Display books
- 3.Issue book
- 4.Return book
- 5.Delete book
- 6.Update book
- 7.Exit

Enter your choice(1-7)=2

- 1.All books
- 2.Particular accession number
- 3.Particular genre
- 4.Particular author
- 5.Particular publisher
- 6.Specific title
- 7.Go back

```
Enter your choice(1-7)=2
Enter accession number you are looking for=1021
Book not found
*****
1.All books
2.Particular accession number
3.Particular genre
4.Particular author
5.Particular publisher
6.Specific title
7.Go back
Enter your choice(1-7)=7
```

```
1.Add book in library
2.Display books
3.Issue book
4.Return book
5.Delete book
6.Update book
7.Exit
```

```
Enter your choice(1-7)=6
```

```
*****
1.All books
2.Particular accession number
3.Particular genre
4.Particular author
5.Particular publisher
6.Specific title
7.Go back
Enter your choice(1-7)=7
```

```
Enter book accession number=1054
[1054, 'Cindrella', 'Daisy Fisher', 'Scribner', 'fiction', 500, 10, 0]
You want to 1.Add or 2.Remove copies of the book or 3.EXIT-1
Enter number of copies you want to add=10
Do you want to add more books(y/n):n
You want to 1.Add or 2.Remove copies of the book or 3.EXIT-3
```

```
1.Add book in library
2.Display books
3.Issue book
4.Return book
5.Delete book
6.Update book
7.Exit
```

```
Enter your choice(1-7)=2
*****
```

- 1.All books
- 2.Particular accession number
- 3.Particular genre
- 4.Particular author
- 5.Particular publisher
- 6.Specific title
- 7.Go back

Enter your choice(1-7)=2

Enter accession number you are looking for=1054

Book No	Book Name	Book Author	Book Publisher	Genre	Price	Copies	Issued
1054	Cindrella	Daisy Fisher	Scribner	fiction	500	20	0

- 1.All books
- 2.Particular accession number
- 3.Particular genre
- 4.Particular author

Enter your choice(1-7)=7

- 1.Add book in library
- 2.Display books
- 3.Issue book
- 4.Return book
- 5.Delete book
- 6.Update book
- 7.Exit

Enter your choice(1-7)=6

- 1.All books
- 2.Particular accession number
- 3.Particular genre
- 4.Particular author
- 5.Particular publisher
- 6.Specific title
- 7.Go back

Enter your choice(1-7)=7

Enter book accession number=1034

Wrong accession number entered

- 1.Add book in library
- 2.Display books
- 3.Issue book
- 4.Return book
- 5.Delete book
- 6.Update book
- 7.Exit

Enter your choice(1-7)=6

```
2.Particular accession number  
3.Particular genre  
4.Particular author  
5.Particular publisher  
6.Specific title  
7.Go back  
Enter your choice(1-7)=7
```

```
Enter book accession number=1054  
[1054, 'Cindrella', 'Daisy Fisher', 'Scribner', 'fiction', 500, 20, 0]  
You want to 1.Add or 2.Remove copies of the book or 3.EXIT-2  
Enter number of copies you want to remove=12  
Do you want to remove more books(y/n):n  
You want to 1.Add or 2.Remove copies of the book or 3.EXIT-3
```

```
1.Add book in library  
2.Display books  
3.Issue book  
4.Return book  
5.Delete book  
6.Update book  
7.Exit
```

```
Enter your choice(1-7)=2
```

```
*****  
1.All books  
2.Particular accession number  
3.Particular genre  
4.Particular author  
5.Particular publisher  
6.Specific title  
7.Go back  
Enter your choice(1-7)=2
```

```
Enter accession number you are looking for=1054  
Book No Book Name Book Author Book Publisher Genre Price Copies Issued  
1054 Cindrella Daisy Fisher Scribner fiction 500 8 0  
*****
```

```
1.All books  
2.Particular accession number  
3.Particular genre  
4.Particular author  
5.Particular publisher  
6.Specific title  
7.Go back  
Enter your choice(1-7)=7
```

```
1.Add book in library  
2.Display books  
3.Issue book  
4.Return book  
5.Delete book  
6.Update book  
7.Exit  
Enter your choice(1-7)=71  
Invalid choice
```

```
1.Add book in library  
2.Display books  
3.Issue book  
4.Return book  
5.Delete book  
6.Update book  
7.Exit  
Enter your choice(1-7)=7  
Okay here!!!
```

PRINT-26: S.No.-86

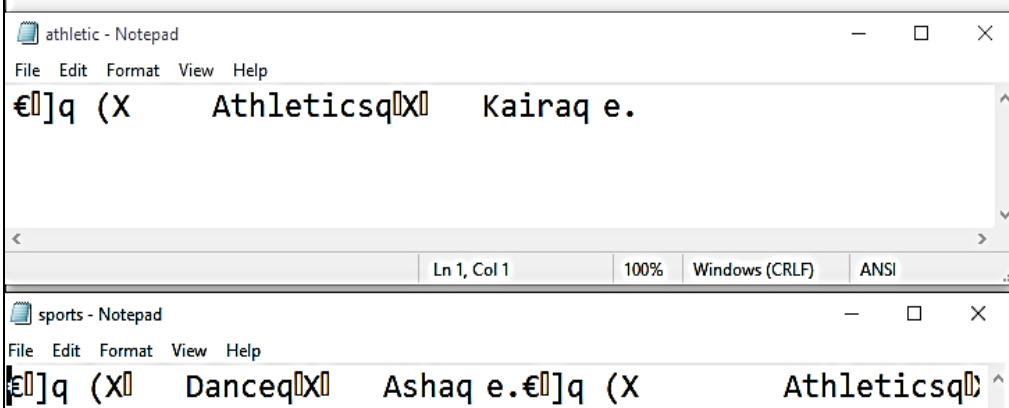
To read contents from file sports.dat and create a file named Athletic.dat copying only those records from sports.dat where the event name is ‘Athletics’ (sports.dat contains information in the format: Event - Participant)

```
##### PROGRAM PRINT-26 #####
#Sno-86:To read contents from file sports.dat and |create a file
#named Athletic.dat copying only those records from sports.dat
#where the event name is 'Athletics'
#AUTHOR-KONICAA SHARMA

import pickle
myfile=open("sports.dat","rb+")
myfile2=open("athletic.dat","wb+")
try:
    while True:
        sport=pickle.load(myfile)
        if sport[0]=='Athletics':
            a=sport
            pickle.dump(a myfile2)
except EOFError:
    print("File operation successful..")

myfile2.seek(0)
try:
    while True:
        data=pickle.load(myfile2)
        print(data)
except EOFError:
    print("Successfully copied records in Athletic.dat...")
myfile.close()
myfile2.close()
```

```
===== RESTART: C:\Users\Hp\Documents\DATA FILE HANDLING\sno-86.py =====
File operation successful..
['Athletics', 'Kaira']
Successfully copied records in Athletic.dat...
>>>
```



PRINT-27: S.No.-91

To read characters from keyboard one by one .All lower case characters get stored inside the file LOWER, all uppercase characters get stored inside the file UPPER and all other characters get stored inside the file OTHERS

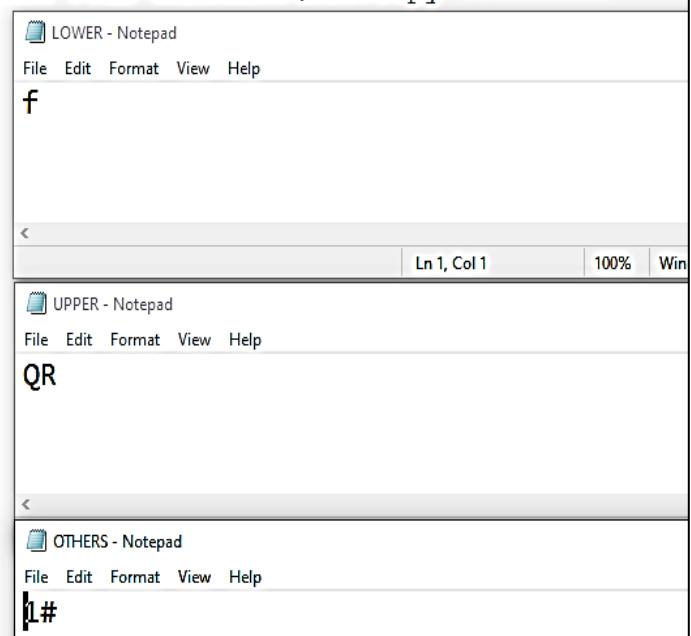
```
##### PROGRAM PRINT-27 #####
#SNO-91:To read characters from keyboard one by one.
#All lower case characters get stored inside the file LOWER,
#all uppercase characters get stored inside the file UPPER
#and all other characters get stored inside the file OTHERS
#AUTHOR-KONICAA SHARMA

myfile1=open('LOWER','w+')
myfile2=open('UPPER','w+')
myfile3=open('OTHERS','w+')
ans='y'
while ans in 'yY':
    rec=input("Enter a character:")
    if rec.islower():
        myfile1.write(rec)
    elif rec.isupper():
        myfile2.write(rec)
    else:
        myfile3.write(rec)
    ans=input("Do you want to enter more records(y/n):")
myfile1.close()
myfile2.close()
myfile3.close()
```

```
===== RESTART: C:\Users\Hp\Documents\DATA FILE HANDLING\sno91.py =====
```

```
Enter a character:l
Do you want to enter more records(y/n):y
Enter a character:f
Do you want to enter more records(y/n):y
Enter a character:Q
Do you want to enter more records(y/n):y
Enter a character:R
Do you want to enter more records(y/n):y
Enter a character:#
```

```
Do you want to enter more records(y/n):n
>>>
```



PRINT-28: S.No.107
Menu driven program working on Batch.csv

```
##### PROGRAM PRINT-28 #####
#SNo-107: To write records of student in a csv file
#and find the topper
#AUTHOR-KONICAA SHARMA

import csv
def add_rec():
    ans='y'
    fields=['NAME','CLASS','MARKS']
    student=[]
    st=[]
    while ans in 'yY':
        nm=input("Enter name of the student:")
        cl=input("Enter class:")
        m=int(input("Enter marks:"))
        student=[nm,cl,m]
        st.append(student)
        ans=input("Do you want to enter more records:")
    with open ('topper.csv','w',newline='') as myfile:
        csv_writer=csv.writer(myfile,delimiter=',')
        csv_writer.writerow(fields)
        for row in st:
            csv_writer.writerow(row)
    print("File is created")

def cl_topper():
    q=[]
    with open('topper.csv','r',newline='') as myfile:
        csv_reader=csv.reader(myfile)
        topper=0
        st=next(csv_reader)
        for row in csv_reader:
            if int(row[2])>topper:
                topper=int(row[2])
                name=row[0]
                cl=row[1]
                q=[topper,name,cl]
    print("Topper is",name)
    print(q)

def display():
    with open('topper.csv','r',newline='') as myfile:
        csv_reader=csv.reader(myfile)
        st=next(csv_reader)
        print(st)
        for row in csv_reader:
            print(row)
```

```
while True:  
    print("=====  
    print("##### MAIN MENU #####")  
    print("\n1.Add student's records")  
    print("\n2.Display all records")  
    print("\n3.Display topper")  
    print("\n4.Exit")  
    choice=int(input("Enter your choice="))  
    if choice==1:  
        add_rec()  
    elif choice==2:  
        display()  
    elif choice==3:  
        cl_topper()  
    elif choice==4:  
        break
```

```
>>>  
===== RESTART: C:\Users\Hp\Documents\CSV FILE\sno107.py =====  
=====  
##### MAIN MENU #####  
  
1.Add student's records  
  
2.Display all records  
  
3.Display topper  
  
4.Exit  
Enter your choice=1  
Enter name of the student:KONICAA  
Enter class:XII-D  
Enter marks:499  
Do you want to enter more records:Y  
Enter name of the student:MANSHA  
Enter class:XII-A  
Enter marks:490  
Do you want to enter more records:Y  
Enter name of the student:DEVANSH  
Enter class:XII-B  
Enter marks:409  
Do you want to enter more records:N  
File is created
```

```
1.Add student's records  
2.Display all records  
3.Display topper  
4.Exit  
Enter your choice=2  
['NAME', 'CLASS', 'MARKS']  
['KONICAA', 'XII-D', '499']  
['MANSHA', 'XII-A', '490']  
['DEVANSH', 'XII-B', '409']
```

```
=====  
##### MAIN MENU #####  
1.Add student's records  
2.Display all records  
3.Display topper  
4.Exit  
Enter your choice=3  
Topper is KONICAA  
[499, 'KONICAA', 'XII-D']
```

```
=====  
##### MAIN MENU #####  
1.Add student's records  
2.Display all records  
3.Display topper  
4.Exit  
Enter your choice=4  
>>>
```

PRINT-29: S.No.-109

To read records of employees in myfile.csv and display those who have salary more than 70000

```
#####
# PROGRAM PRINT-29 #####
#SNO-109:Sum of salary and counting employee
#getting more than 70000
#AUTHOR-KONICAA SHARMA

import csv
with open ('myfile.csv') as csvfile:
    myreader=csv.reader(csvfile,delimiter=',')
    count=0
    sum=0
    print ("%10s"% "EMPNO", "%20s"%"EMP NAME", "%10s"%"SALARY")
    print ("=====")
    for row in myreader:
        print ("%10s"%row[0], "%20s"%row[1], "%10s"%row[2])
        sum+=int(row[2])
        if int(row[2])>70000:
            count+=1
    print ("=====")
    print ("%30s"% "SUM OF SALARY : ",sum)
    print ("%40s"% "#EMPLOYEE GETTING SALARY>70000 : ",count)
    print ("=====")
```

A	B	C
1	1 Amit	6000
2	2 Suresh Ku	8000
3	3 Gabbar	75000
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

```
>>>
===== RESTART: C:\Users\Hp\Documents\CSV FILE\sno109.py =====
      EMPNO          EMP NAME      SALARY
=====
      1              Amit       6000
      2          Suresh Kumar     8000
      3           Gabbar       75000
=====
      SUM OF SALARY : 89000
 #EMPLOYEE GETTING SALARY>70000 : 1
=====

>>>
```

PRINT-30: S.No.-110

To read contacts.csv containing name and phone number of people and to search phone number of a specific person

```
##### PROGRAM PRINT-30 #####
#SNo-110:Read contacts.csv containing name and phone number of a
#person and to search phone number of a specified person
#display all
#AUTHOR-KONICAA SHARMA

import csv
def add_rec():
    ans='y'
    fields=['NAME', 'PHNO']
    contact=[]
    ct=[]
    while ans in 'yY':
        nm=input("Enter name:")
        ph=int(input("Enter phone number:"))
        contact=[nm,ph]
        ct.append(contact)
        ans=input("Do you want to enter more records:")
    with open ('contacts.csv','w+',newline='') as myfile:
        csv_writer=csv.writer(myfile,delimiter=',')
        csv_writer.writerow(fields)
        for row in ct:
            csv_writer.writerow(row)
        print("File is created")

def req_contact():
    with open('contacts.csv','r',newline='') as myfile:
        csv_reader=csv.reader(myfile)
        contact=0
        st=next(csv_reader)
        name=input("Enter the name of the contact you are searching for=")
        for row in csv_reader:
            if row[0]==name:
                print(row)

def display():
    with open('contacts.csv','r',newline='') as myfile:
        csv_reader=csv.reader(myfile)
        st=next(csv_reader)
        print(st)

        for row in csv_reader:
            print(row)
```

```

#####
# MAIN PROGRAM #####
#####

while True:
    print("1.Add contacts")
    print("2.Display all the contacts")
    print("3.Search for a particular contact number")
    print("4.EXIT")
    ch=int(input("Enter your choice="))
    if ch==1:
        add_rec()
    elif ch==2:
        display()
    elif ch==3:
        req_contact()
    elif ch==4:
        print("TERMINATING")
        break
    else:
        print("WRONG CHOICE")

```

```

>>>
=====
RESTART: C:\Users\Hp\Documents\CSV FILE\sno110.py =====
1.Add contacts
2.Display all the contacts
3.Search for a particular contact number
4.EXIT
Enter your choice=1
Enter name:RAVI
Enter phone number:9808472987
Do you want to enter more records:Y
Enter name:MAYA
Enter phone number:7809254387
Do you want to enter more records:Y
Enter name:DIVYA
Enter phone number:8093542987
Do you want to enter more records:N
File is created
1.Add contacts
2.Display all the contacts
3.Search for a particular contact number
4.EXIT
Enter your choice=2
['NAME', 'PHNO']
['RAVI', '9808472987']
['MAYA', '7809254387']
['DIVYA', '8093542987']

```

```
1.Add contacts
2.Display all the contacts
3.Search for a particular contact number
4.EXIT
Enter your choice=3
Enter the name of the contact you are searching for=RAVI
['RAVI', '9808472987']
1.Add contacts
2.Display all the contacts
3.Search for a particular contact number
4.EXIT
Enter your choice=4
TERMINATING
>>>
```

PRINT-31: S.No.-116

To read employee records from emp.csv file and copy in binary file Emp2.dat and read all records from Emp2.dat

```
##### PROGRAM PRINT-31 #####
#Sno116-To read employee records from 'emp.csv' file and
#copy in binary file 'EMP2.dat', read all records from "EMP2.dat"
#AUTHOR-KONICAA SHARMA

import pickle
import csv

def read_copy():
    try:
        myfile=open("emp.csv","r")
        myfile2=open("Emp2.dat","wb+")
        emp=[]
        mycsv=csv.reader(myfile)
        next(mycsv)
        for rec in mycsv:
            ecode=int(rec[0])
            ename=rec[1]
            basic=float(rec[2])
            emp=[ecode,ename,basic]
            pickle.dump(emp,myfile2)
            myfile2.flush()
        print("Data of emp.csv(csv file) copied to Emp2.dat(binary file)")
        myfile.close()
        myfile2.close()
    except FileNotFoundError:
        print("File does not exist")

def read_all():
    try:
        myfile=open("Emp2.dat","rb")
        try:
            while True:
                e=pickle.load(myfile)
                print(e)
        except EOFError:
            myfile.close()
    except FileNotFoundError:
        print("File Does Not Exist")
#####
##### MAIN PROGRAM #####
#####
```

```

while True:
    print("\n1.Copy records from emp.csv to Emp2.dat")
    print("\n2.Read all records from Emp2.dat")
    print("\n3.Exit")
    ch=int(input("Enter your choice(1-3)?"))
    if ch==1:
        read_copy()
    elif ch==2:
        read_all()
    elif ch==3:
        print("Enter terminating")
        break
    else:
        print("Wrong Choice")

```

```

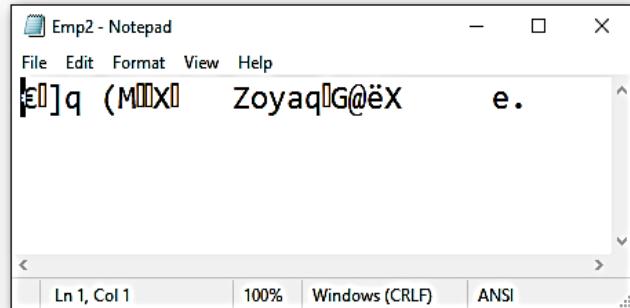
>>>
===== RESTART: C:\Users\Hp\Documents\CSV FILE\sno116.py =====

1.Copy records from emp.csv to Emp2.dat
2.Read all records from Emp2.dat
3.Exit
Enter your choice(1-3)?2
[1215, "'Rohit Kapoor'", 280000.0]
[1021, "'Soham Kapoor'", 27000.0]

1.Copy records from emp.csv to Emp2.dat
2.Read all records from Emp2.dat
3.Exit
Enter your choice(1-3)?1
Data of emp.csv(csv file)copied to Emp2.dat(binary file)

1.Copy records from emp.csv to Emp2.dat
2.Read all records from Emp2.dat
3.Exit
Enter your choice(1-3)?3
Enter terminating
>>>

```



PRINT-32: S.No.-117

To sort emp.csv's records according to salary column in descending order or name column in ascending order

```
##### PROGRAM PRINT-32 #####
#SNo-117:To sort Employee records according to salary column in descending order in 'emp.csv' or according to name column in ascending order
#KONICAA SHARMA

import csv
def write_add():
    myfile=open('emp.csv','a',newline='')
    mycsv=csv.writer(myfile)
    e=[]
    n=myfile.tell()
    if n==0:
        e=["Empcode","EmpName","Basic"]
        mycsv.writerow(e)
        myfile.flush()
    else:
        while True:
            empcode=int(input("Enter Employee code="))
            empname=input("Enter Employee name=")
            basic=float(input("Enter Basic Salary="))
            e=[empcode,empname,basic]
            mycsv.writerow(e)
            myfile.flush()
            ans=input("Do you want to enter more records of employee (y/n) :")
            if ans in 'nN':
                break
    myfile.close()

def read():
    try:
        myfile=open('emp.csv','r')
        emp=[]
        mycsv=csv.reader(myfile)
        heading=next(mycsv)
        print(heading)
        for rec in mycsv:
            print(rec)
    except FileNotFoundError:
        print("File does not exist")
```

```

def sort():
    try:
        myfile=open('emp.csv','r')
        emp=[]
        mycsv=csv.reader(myfile)
        heading=next(mycsv)
        for rec in mycsv:
            emp.append(rec)
        size=len(emp)
        for i in range(size):
            for j in range(i+1,size):
                if emp[i][2]<emp[j][2]:
                    emp[i],emp[j]=emp[j],emp[i]
        myfile.close()
        myfile2=open('emp.csv','w',newline='')
        mycsv2=csv.writer(myfile2)
        mycsv2.writerow(heading)
        for row in emp:
            mycsv2.writerow(row)
        myfile2.close()
        print("Sorting of salaries successful")
    except FileNotFoundError:
        print("File does not exist")

```

```

def sort_name():
    try:
        myfile=open('emp.csv','r')
        emp=[]
        mycsv=csv.reader(myfile)
        heading=next(mycsv)
        for rec in mycsv:
            emp.append(rec)
        size=len(emp)
        for i in range(size):
            for j in range(i+1,size):
                if emp[i][1]>emp[j][1]:
                    emp[i],emp[j]=emp[j],emp[i]
        myfile.close()
        myfile2=open('emp.csv','w',newline='')
        mycsv2=csv.writer(myfile2)
        mycsv2.writerow(heading)
        for row in emp:
            mycsv2.writerow(row)
        myfile2.close()
        print("Sorting of names is successful")
    except FileNotFoundError:
        print("File does not exist")

```

```

##### MAIN PROGRAM #####
while True:
    print("\n1.Write/Add employee records in emp.csv")
    print("\n2.Read all records of emp.csv")
    print("\n3.Sort records of emp.csv by salary column in descending order")
    print("\n4.Sort records of emp.csv by name column in ascending order")
    print("\n5.Exit")
    ch=int(input("Enter your choice(1-5):"))
    if ch==1:
        write_add()
    elif ch==2:
        read()
    elif ch==3:
        sort()
    elif ch==4:
        sort_name()
    elif ch==5:
        print("Terminating")
        break
    else:
        print("Wrong Choice")

```

```

1.Write/Add employee records in emp.csv

2.Read all records of emp.csv

3.Sort records of emp.csv by salary column in descending order

4.Sort records of emp.csv by name column in ascending order

5.Exit
Enter your choice(1-5):1
Enter Employee code=1021
Enter Employee name=RITU
Enter Basic Salary=50000
Do you want to enter more recordsof employee(y/n):Y
Enter Employee code=1022
Enter Employee name=NEHA
Enter Basic Salary=80000
Do you want to enter more recordsof employee(y/n):Y
Enter Employee code=1023
Enter Employee name=MEGHA
Enter Basic Salary=40000
Do you want to enter more recordsof employee(y/n):N

```

```
1.Write/Add employee records in emp.csv  
2.Read all records of emp.csv  
3.Sort records of emp.csv by salary column in descending order  
4.Sort records of emp.csv by name column in ascending order  
5.Exit  
Enter your choice(1-5):2  
['Empcode', 'EmpName', 'Basic']  
['1022', 'Amit', '45000.0']  
['1021', 'Jitendra', '50000.0']  
['1023', 'Raghbir', '90000.0']  
['1021', 'RITU', '50000.0']  
['1022', 'NEHA', '80000.0']  
['1023', 'MEGHA', '40000.0']  
  
1.Write/Add employee records in emp.csv  
2.Read all records of emp.csv  
3.Sort records of emp.csv by salary column in descending order  
4.Sort records of emp.csv by name column in ascending order  
5.Exit
```

```
Enter your choice(1-5):3  
Sorting of salaries successful  
  
1.Write/Add employee records in emp.csv  
2.Read all records of emp.csv  
3.Sort records of emp.csv by salary column in descending order  
4.Sort records of emp.csv by name column in ascending order  
5.Exit  
Enter your choice(1-5):4  
Sorting of names is successful  
  
1.Write/Add employee records in emp.csv  
2.Read all records of emp.csv  
3.Sort records of emp.csv by salary column in descending order  
4.Sort records of emp.csv by name column in ascending order  
5.Exit  
Enter your choice(1-5):5  
Terminating
```

PRINT-33: S.No.-118

Implementing Stack using list, create a stack, push an element to a Stack, checking for empty Stack (Underflow), pop operation/deleting elements from a stack and traversal/displaying a Stack

```
##### PROGRAM PRINT-33 #####
#SNo-118:IMPLEMENTING STACK USING LIST
#Create a stack,Push an element to a Stack,Checking for empty Stack(Underflow)
#POP operation/Deleting elements from a stack,Traversal/Displaying a Stack
#AUTHOR-KONICAA SHARMA

Stack=list() # or Stack=[]-Creating a Stack
top=-1
def push_stack():
    global top
    num=int(input("Enter element to push="))
    Stack.append(num)
    print(num,"is pushed in Stack")
    top=len(Stack)-1

def pop_stack():
    global top
    if len(Stack)==0:#or Stack==[]
        print("Memory Underflow/Stack is empty")
    else:
        num=Stack.pop()
        print("Element deleted from the top of the Stack i.e.",num)
        top=len(Stack)-1

def display():
    if Stack==[]:
        print("Stack is Empty or Memory underflow so nothing to display/traverse")
    else:
        top=len(Stack)-1
        print("TOP->",Stack[top])
        for i in range(top-1,-1,-1):
            print(Stack[i])

#####
# MAIN PROGRAM #####
while True:
    print("*****")
    print("1.PUSH in Stack")
    print("2.POP in Stack")
    print("3.Traverse/Display Stack")
    print("4.Exit")
```

```

choice=int(input("Enter your choice(1-4):"))
print("*****")
if choice==1:
    push_Stack()
    print("Top is at",top)
elif choice==2:
    pop_Stack()
    print("Top is at",top)
elif choice==3:
    display()
elif choice==4:
    print("Terminating")
    break
else:
    print("Wrong Choice")

```

```

>>>
===== RESTART: C:\Users\Hp\Documents\STACK\sno118.py =====
*****
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4):1
*****
Enter element to push=10
10 is pushed in Stack
Top is at 0
*****
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4):1
*****
Enter element to push=20
20 is pushed in Stack
Top is at 1
*****
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4):1
*****

```

```
*****
Enter element to push=30
30 is pushed in Stack
Top is at 2
*****
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4):3
*****
TOP-> 30
20
10
*****
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4):2
*****
Element deleted from the top of the Stack i.e. 30
Top is at 1
*****
```

```
*****
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4):2
*****
Element deleted from the top of the Stack i.e. 20
Top is at 0
*****
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4):3
*****
TOP-> 10
*****
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4):2
*****
Element deleted from the top of the Stack i.e. 10
Top is at -1
*****
```

```
*****  
1.PUSH in Stack  
2.POP in Stack  
3.Traverse/Display Stack  
4.Exit  
Enter your choice(1-4):3  
*****  
Stack is Empty or Memory underflow so nothing to display/traverse  
*****  
1.PUSH in Stack  
2.POP in Stack  
3.Traverse/Display Stack  
4.Exit  
Enter your choice(1-4):2  
*****  
Memory Underflow/Stack is empty  
Top is at -1  
*****  
1.PUSH in Stack  
2.POP in Stack  
3.Traverse/Display Stack  
4.Exit  
Enter your choice(1-4):4  
*****  
Terminating  
>>> |
```

PRINT-34: S.No.-119

Implementing queue using list, create a queue, checking for empty queue, insert an element to the queue, deleting element from queue, traversal/displaying queue

```
##### PROGRAM PRINT-34 #####
#SNo-119:IMPLEMENTING QUEUE USING LIST
#Create a queue,Push an element to a queue,Checking for empty Queue(Underflow)
#Delete operation/Deleting elements from a QUEUE,Traversal/Displaying a Queue
#AUTHOR-KONICAA SHARMA

Queue=list() # or Queue=[]-Creating a Queue
Front=-1
Rear=-1

def Insert_Queue():
    global Front,Rear
    num=int(input("Enter element to insert="))
    if Queue==[]:
        Front=Rear=0
    else:
        Rear=Rear+1
    Queue.append(num)

def Delete_Queue():
    global Front,Rear,Queue
    if len(Queue)==0:#or Stack==[]
        print("Memory Underflow/Queue is empty")
    else:
        num=Queue[0]
        Queue.pop(0)
        if Front==Rear:
            print("The only element in Queue is deleted i.e.",num)
            Front=Rear=-1
        else:
            print("Element deleted from the Front of the Queue i.e.",num)
            Rear-=1

def display():
    global Front,Rear
    if Queue==[]:
        print("Queue is Empty or Memory underflow so nothing to display/traverse")
    else:
        print("Front->",end="")
        for i in range(Front,Rear+1):
            print(Queue[i],end="-")
        print("Rear")
```

```

##### MAIN PROGRAM #####
while True:
    print("*****")
    print("1.INSERTION in Queue")
    print("2.Deletion in Queue")
    print("3.Traverse/Display Queue")
    print("4.Exit")
    choice=int(input("Enter your choice(1-4):"))
    print("*****")
    if choice==1:
        Insert_Queue()
        print("Position of Front is ",Front,"and Rear is",Rear)
    elif choice==2:
        Delete_Queue()
        print("Position of Front is ",Front,"and Rear is",Rear)
    elif choice==3:
        display()
    elif choice==4:
        print("Terminating")
        break
    else:
        print("Wrong Choice")

```

```

>>>
=====
      RESTART: C:\Users\Hp\Documents\STACK\sno119.py =====
*****
1.INSERTION in Queue
2.Deletion in Queue
3.Traverse/Display Queue
4.Exit
Enter your choice(1-4):1
*****
Enter element to insert=18
Position of Front is  0 and Rear is 0
*****
1.INSERTION in Queue
2.Deletion in Queue
3.Traverse/Display Queue
4.Exit
Enter your choice(1-4):1
*****
Enter element to insert=36
Position of Front is  0 and Rear is 1
*****
1.INSERTION in Queue
2.Deletion in Queue
3.Traverse/Display Queue
4.Exit
Enter your choice(1-4):1
*****
Enter element to insert=72
Position of Front is  0 and Rear is 2

```

```
*****  
1.INSERTION in Queue  
2.Deletion in Queue  
3.Traverse/Display Queue  
4.Exit  
Enter your choice(1-4):3  
*****  
Front->18<-36<-72<-Rear  
*****  
1.INSERTION in Queue  
2.Deletion in Queue  
3.Traverse/Display Queue  
4.Exit  
Enter your choice(1-4):2  
*****  
Element deleted from the Front of the Queue i.e. 18  
Position of Front is 0 and Rear is 1  
*****  
1.INSERTION in Queue  
2.Deletion in Queue  
3.Traverse/Display Queue  
4.Exit  
Enter your choice(1-4):3  
*****  
Front->36<-72<-Rear
```

```
*****  
1.INSERTION in Queue  
2.Deletion in Queue  
3.Traverse/Display Queue  
4.Exit  
Enter your choice(1-4):2  
*****  
Element deleted from the Front of the Queue i.e. 36  
Position of Front is 0 and Rear is 0  
*****  
1.INSERTION in Queue  
2.Deletion in Queue  
3.Traverse/Display Queue  
4.Exit  
Enter your choice(1-4):3  
*****  
Front->72<-Rear  
*****  
1.INSERTION in Queue  
2.Deletion in Queue  
3.Traverse/Display Queue  
4.Exit  
Enter your choice(1-4):2  
*****  
The only element in Queue is deleted i.e. 72  
Position of Front is -1 and Rear is -1
```

```
1.INSERTION in Queue
2.Deletion in Queue
3.Traverse/Display Queue
4.Exit
Enter your choice(1-4):3
*****
Queue is Empty or Memory underflow so nothing to display/traverse
*****
1.INSERTION in Queue
2.Deletion in Queue
3.Traverse/Display Queue
4.Exit
Enter your choice(1-4):4
*****
Terminating
>>>
```

PRINT-35: S.No.-122

**To write add(books) and delete(books) methods in Python to add and remove books
considering them to act as append() and pop() operations in Stack**

```
##### PROGRAM PRINT-35 #####
#SNo-122:To write add(books) and delete(books) methods in Python to add and
#remove books considering them to act as append() and pop() operations in Stack
#AUTHOR-KONICAA SHARMA

books=list() # or books=[]-Creating a books
top=-1
def add_books():
    global top
    nm=input("Enter book name=")
    books.append(nm)
    print(nm,"is pushed in books")
    top=len(books)-1

def delete_books():
    global top
    if len(books)==0:#or books==[]
        print("Memory Underflow/books is empty")
    else:
        nm=books.pop()
        print("Element deleted from the top of the books i.e.",nm)
        top=len(books)-1

def display():
    if books==[]:
        print("books is Empty or Memory underflow so nothing to display/traverse")
    else:
        top=len(books)-1
        print("TOP->",books[top])
        for i in range(top-1,-1,-1):
            print(books[i])

##### MAIN PROGRAM #####
while True:
    print("*****")
    print("1.ADD in books")
    print("2.DELETE in books")
    print("3.Traverse/Display books")
    print("4.Exit")
```

```

choice=int(input("Enter your choice(1-4):"))
print("*****")
if choice==1:
    add_books()
    print("Top is at",top)
elif choice==2:
    delete_books()
    print("Top is at",top)
elif choice==3:
    display()
elif choice==4:
    print("Terminating")
    break
else:
    print("Wrong Choice")

```

```

>>>
===== RESTART: C:\Users\Hp\Documents\STACK\sno122.py =====
*****
1.ADD in books
2.DELETE in books
3.Traverse/Display books
4.Exit
Enter your choice(1-4):1
*****
Enter book name=MALGUDI_TALES
MALGUDI_TALES is pushed in books
Top is at 0
*****
1.ADD in books
2.DELETE in books
3.Traverse/Display books
4.Exit
Enter your choice(1-4):1
*****
Enter book name=HARRY POTTER
HARRY POTTER is pushed in books
Top is at 1
*****
1.ADD in books
2.DELETE in books
3.Traverse/Display books
4.Exit
Enter your choice(1-4):3
*****

```

```
*****
TOP-> HARRY POTTER
MALGUDI TALES
*****
1.ADD in books
2.DELETE in books
3.Traverse/Display books
4.Exit
Enter your choice(1-4):1
*****
Enter book name=EXEMPLAR
EXEMPLAR is pushed in books
Top is at 2
*****
1.ADD in books
2.DELETE in books
3.Traverse/Display books
4.Exit
Enter your choice(1-4):3
*****
TOP-> EXEMPLAR
HARRY POTTER
MALGUDI TALES
*****
```

```
*****
1.ADD in books
2.DELETE in books
3.Traverse/Display books
4.Exit
Enter your choice(1-4):2
*****
Element deleted from the top of the books i.e. EXEMPLAR
Top is at 1
*****
1.ADD in books
2.DELETE in books
3.Traverse/Display books
4.Exit
Enter your choice(1-4):2
*****
Element deleted from the top of the books i.e. HARRY POTTER
Top is at 0
*****
1.ADD in books
2.DELETE in books
3.Traverse/Display books
4.Exit
Enter your choice(1-4):3
*****
TOP-> MALGUDI TALES
*****
```

```
*****  
1.ADD in books  
2.DELETE in books  
3.Traverse/Display books  
4.Exit  
Enter your choice(1-4):2  
*****  
Element deleted from the top of the books i.e. MALGUDI_TALES  
Top is at -1  
*****  
1.ADD in books  
2.DELETE in books  
3.Traverse/Display books  
4.Exit  
Enter your choice(1-4):2  
*****  
Memory Underflow/books is empty  
Top is at -1  
*****
```

```
*****  
1.ADD in books  
2.DELETE in books  
3.Traverse/Display books  
4.Exit  
Enter your choice(1-4):3  
*****  
books is Empty or Memory underflow so nothing to display/traverse  
*****  
1.ADD in books  
2.DELETE in books  
3.Traverse/Display books  
4.Exit  
Enter your choice(1-4):4  
*****  
Terminating  
>>>
```

PRINT-36: S.No.-123

To write AddClient() and DeleteClient() methods to add a new client and delete a client from client name list of Queue data structure

```
##### PROGRAM PRINT-36 #####
#SNo-123:To write AddClient() and DeleteClient() methods to add a new client
#and delete a client from client name list of Queue data structure
#AUTHOR-KONICAA SHARMA

Client=list() # or Client=[]-Creating a Client
Front=-1
Rear=-1
def AddClient():
    l=[]
    global Front,Rear
    code=int(input("Enter client code="))
    nm=input("Enter name of the client=")
    sal=int(input("Enter salary="))
    l=[code,nm,sal]
    if Client==[]:
        Front=Rear=0
    else:
        Rear=Rear+1
    Client.append(l)

def DeleteClient():
    global Front,Rear,Client
    if len(Client)==0:
        print("Memory Underflow/Client is empty")
    else:
        num=Client[0]
        Client.pop(0)
        if Front==Rear:
            print("The only element in Client is deleted i.e.",num)
            Front=Rear=-1
        else:
            print("Element deleted from the Front of the Client i.e.",num)
            Rear-=1

def display():
    global Front,Rear
    if Client==[]:
        print("Client is Empty or Memory underflow so nothing to display/traverse")
    else:
        print("Front->",end="")
        for i in range(Front,Rear+1):
            print(Client[i],end="-")
        print("Rear")
```

```

##### MAIN PROGRAM #####
while True:
    print("*****")
    print("1.INSERTION in Client")
    print("2.Deletion in Client")
    print("3.Traverse/Display Client")
    print("4.Exit")
    choice=int(input("Enter your choice(1-4) :"))
    print("*****")
    if choice==1:
        AddClient()
        print("Position of Front is ",Front,"and Rear is",Rear)
    elif choice==2:
        DeleteClient()
        print("Position of Front is ",Front,"and Rear is",Rear)
    elif choice==3:
        display()
    elif choice==4:
        print("Terminating")
        break
    else:
        print("Wrong Choice")

```

```

=====
RESTART: C:\Users\Hp\Documents\STACK\sno123.py =====
*****
1.INSERTION in Client
2.Deletion in Client
3.Traverse/Display Client
4.Exit
Enter your choice(1-4):1
*****
Enter client code=1021
Enter name of the client=RAVI
Enter salary=100000
Position of Front is 0 and Rear is 0
*****
1.INSERTION in Client
2.Deletion in Client
3.Traverse/Display Client
4.Exit
Enter your choice(1-4):1
*****
Enter client code=1022
Enter name of the client=RAHUL
Enter salary=80000
Position of Front is 0 and Rear is 1
*****

```

```
1.INSERTION in Client
2.Deletion in Client
3.Traverse/Display Client
4.Exit
Enter your choice(1-4):1
*****
Enter client code=1023
Enter name of the client=BABITA
Enter salary=90000
Position of Front is 0 and Rear is 2
*****
1.INSERTION in Client
2.Deletion in Client
3.Traverse/Display Client
4.Exit
Enter your choice(1-4):3
*****
Front->[1021, 'RAVI', 100000]<-[1022, 'RAHUL', 80000]<-[1023, 'BABITA', 90000]<-Rear
*****
1.INSERTION in Client
2.Deletion in Client
3.Traverse/Display Client
4.Exit
Enter your choice(1-4):2
*****
Element deleted from the Front of the Client i.e. [1021, 'RAVI', 100000]
Position of Front is 0 and Rear is 1
*****
```

```
1.INSERTION in Client
2.Deletion in Client
3.Traverse/Display Client
4.Exit
Enter your choice(1-4):3
*****
Front->[1022, 'RAHUL', 80000]<-[1023, 'BABITA', 90000]<-Rear
*****
1.INSERTION in Client
2.Deletion in Client
3.Traverse/Display Client
4.Exit
Enter your choice(1-4):2
*****
Element deleted from the Front of the Client i.e. [1022, 'RAHUL', 80000]
Position of Front is 0 and Rear is 0
*****
1.INSERTION in Client
2.Deletion in Client
3.Traverse/Display Client
4.Exit
Enter your choice(1-4):3
*****
Front->[1023, 'BABITA', 90000]<-Rear
*****
```

```
1.INSERTION in Client
2.Deletion in Client
3.Traverse/Display Client
4.Exit
Enter your choice(1-4):2
*****
The only element in Client is deleted i.e. [1023, 'BABITA', 90000]
Position of Front is -1 and Rear is -1
*****
1.INSERTION in Client
2.Deletion in Client
3.Traverse/Display Client
4.Exit
Enter your choice(1-4):3
*****
Client is Empty or Memory underflow so nothing to display/traverse
*****
1.INSERTION in Client
2.Deletion in Client
3.Traverse/Display Client
4.Exit
Enter your choice(1-4):4
*****
Terminating
>>>
```

PRINT-37: S.No.-126

A stack STK and a queue QUE are being maintained. Check whether they have the same size, check for equality and stop and report if elements are different

```
##### PROGRAM PRINT-37 #####
#SNO-126:A stack STK and a queue QUE are being maintained.
#To Check whether they have the same size, check for equality
#and stop and report if elements are different
#AUTHOR-KONICAA SHARMA

STK=[]
QUE=[]
a=len(STK)
b=len(QUE)
a=b
def stack():
    def push_Stack():
        num=int(input("Enter element to push="))
        STK.append(num)
        print(num,"is pushed in Stack")
    def pop_Stack():
        if len(STK)==0:
            print("Memory Underflow/Stack is empty")
        else:
            num=STK.pop()
            print("Element deleted from the top of the Stack i.e.",num)
    def display():
        if STK==[]:
            print("Stack is Empty or Memory underflow so nothing to display/traverse")
        else:
            print(STK)

while True:
    print("====")
    print("1.PUSH in Stack")
    print("2.POP in Stack")
    print("3.Traverse/Display Stack")
    print("4.Exit")
    choice=int(input("Enter your choice(1-4)"))
    print("-----")
    if choice==1:
        push_Stack()
    elif choice==2:
        pop_Stack()
    elif choice==3:
        display()
    elif choice==4:
        print("Terminating")
        break
    else:
        print("Wrong Choice")
```

```

def queue():
    def Insert_Queue():
        num=int(input("Enter element to insert="))
        QUE.append(num)
    def Delete_Queue():
        if len(QUE)==0:
            print("Memory Underflow/Queue is empty")
        else:
            num=QUE[0]
            QUE.pop(0)
    def display():
        if QUE==[]:
            print("Queue is Empty or Memory underflow so nothing to display/traverser")
        else:
            print(QUE)

while True:
    print("====")
    print("1.INSERTION in Queue")
    print("2.Deletion in Queue")
    print("3.Traverse/Display Queue")
    print("4.Exit")
    choice=int(input("Enter your choice(1-4)"))
    print("-----")
    if choice==1:
        Insert_Queue()
    elif choice==2:
        Delete_Queue()
    elif choice==3:
        display()
    elif choice==4:
        print("Terminating")
        break
    else:
        print("Wrong Choice")
def Compare():
    def check_size():
        if STK==[] and QUE==[]:
            print("Insert records first")
        else:
            if len(STK)==len(QUE):
                print("Both stack and queue have the same size")
            else:
                print("Stack and Queue do not have the same size")
    def check_equality():
        if STK==[] or QUE==[]:
            print("Insert records in both stack as well as queue first")

```

```

else:
    while len(STK)!=0 and len(QUE)!=0:
        s=STK.pop()
        q=QUE.pop(0)
        flag=1
        if s==q:
            continue
        else:
            flag=0
    if flag==1:
        print("Stack and Queue are equal")
    elif flag==0:
        print("Stack and Queue are not equal")
while True:
    print("====")
    print("1.Check for same size")
    print("2.Check for equality of elements of stack and queue")
    print("3.EXIT")
    ch=int(input("Enter your choice(1-3):"))
    print("-----")
    if ch==1:
        check_size()
    elif ch==2:
        check_equality()
    elif ch==3:
        return
    else:
        print("Wrong Choice")

```

```

##### MAIN PROGRAM #####
while True:
    print("====")
    print("1.Operation on Stack")
    print("2.Operation on Queue")
    print("3.Check for the equality of stack and queue")
    print("4.EXIT")
    c=int(input("Enter your choice(1-4):"))
    if c==1:
        stack()
    elif c==2:
        queue()
    elif c==3:
        Compare()
    elif c==4:
        print("TERMINATING")
        break
    else:
        print("Wrong Choice")

```

```
===== RESTART: C:\Users\Hp\Documents\STACK\sno 126.py =====
=====
1.Operation on Stack
2.Operation on Queue
3.Check for the equality of stack and queue
4.EXIT
Enter your choice(1-4):1
=====
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4)1
-----
Enter element to push=10
10 is pushed in Stack
=====
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4)4
-----
Terminating
```

```
1.Operation on Stack
2.Operation on Queue
3.Check for the equality of stack and queue
4.EXIT
Enter your choice(1-4):3
=====
1.Check for same size
2.Check for equality of elements of stack and queue
3.EXIT
Enter your choice(1-3):1
-----
Stack and Queue do not have the same size
=====
1.Check for same size
2.Check for equality of elements of stack and queue
3.EXIT
Enter your choice(1-3):2
-----
Insert records in both stack as well as queue first
=====
1.Check for same size
2.Check for equality of elements of stack and queue
3.EXIT
Enter your choice(1-3):3
```

```
1.Operation on Stack
2.Operation on Queue
3.Check for the equality of stack and queue
4.EXIT
Enter your choice(1-4):1
=====
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4)1
-----
Enter element to push=20
20 is pushed in Stack
=====
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4)3
-----
[10, 20]
=====
1.PUSH in Stack
2.POP in Stack
3.Traverse/Display Stack
4.Exit
Enter your choice(1-4)4
-----
Terminating
=====
1.Operation on Stack
2.Operation on Queue
3.Check for the equality of stack and queue
4.EXIT
Enter your choice(1-4):2
=====
1.INSERTION in Queue
2.Deletion in Queue
3.Traverse/Display Queue
4.Exit
Enter your choice(1-4)1
-----
Enter element to insert=20
=====
1.INSERTION in Queue
2.Deletion in Queue
3.Traverse/Display Queue
4.Exit
```

```
Enter your choice(1-4)1
-----
Enter element to insert=10
=====
1.INSERTION in Queue
2.Deletion in Queue
3.Traverse/Display Queue
4.Exit
Enter your choice(1-4)3
-----
[20, 10]
=====
1.INSERTION in Queue
2.Deletion in Queue
3.Traverse/Display Queue
4.Exit
Enter your choice(1-4)4
-----
Terminating
=====
1.Operation on Stack
2.Operation on Queue
3.Check for the equality of stack and queue
4.EXIT
Enter your choice(1-4)•3
```

```
2.Check for equality of elements of stack and queue
3.EXIT
Enter your choice(1-3):1
-----
Both stack and queue have the same size
=====
1.Check for same size
2.Check for equality of elements of stack and queue
3.EXIT
Enter your choice(1-3):2
-----
Stack and Queue are equal
=====
1.Check for same size
2.Check for equality of elements of stack and queue
3.EXIT
Enter your choice(1-3):3
-----
=====
1.Operation on Stack
2.Operation on Queue
3.Check for the equality of stack and queue
4.EXIT
Enter your choice(1-4):4
TERMINATING
```

PRINT-38: S.No.-129

To insert or delete an element from a queue depending on user's choice

```
##### PROGRAM PRINT-38 #####
#SNo-129:To insert or delete an element from queue
#depending upon the user's choice
#AUTHOR-KONICAA SHARMA

Queue=list()
Front=-1
Rear=-1

def Insert_Queue():
    global Front,Rear
    num=int(input("Enter element to insert="))
    if Queue==[]:
        Front=Rear=0
    else:
        Rear=Rear+1
    Queue.append(num)

def Delete_Queue():
    global Front,Rear,Queue
    if len(Queue)==0:#or Stack==[]
        print("Memory Underflow/Queue is empty")
    else:
        num=Queue[0]
        Queue.pop(0)
        if Front==Rear:
            print("The only element in Queue is deleted i.e.",num)
            Front=Rear=-1
        else:
            print("Element deleted from the Front of the Queue i.e.",num)
            Rear-=1

def display():
    global Front,Rear
    if Queue==[]:
        print("Queue is Empty or Memory underflow so nothing to display/traverse")
    else:
        print("Front->",end="")
        for i in range(Front,Rear+1):
            print(Queue[i],end="-")
        print("Rear")
```

```

##### MAIN PROGRAM #####
while True:
    print("*****")
    print("1.INSERTION in Queue")
    print("2.Deletion in Queue")
    print("3.Traverse/Display Queue")
    print("4.Exit")
    choice=int(input("Enter your choice(1-4)"))
    print("*****")
    if choice==1:
        Insert_Queue()
        print("Position of Front is ",Front,"and Rear is",Rear)
    elif choice==2:
        Delete_Queue()
        print("Position of Front is ",Front,"and Rear is",Rear)
    elif choice==3:
        display()
    elif choice==4:
        print("Terminating")
        break
    else:
        print("Wrong Choice")

```

```

>>>
=====
RESTART: C:\Users\Hp\Documents\STACK\sno129.py =====
*****
1.INSERTION in Queue
2.Deletion in Queue
3.Traverse/Display Queue
4.Exit
Enter your choice(1-4)1
*****
Enter element to insert=5
Position of Front is 0 and Rear is 0
*****
1.INSERTION in Queue
2.Deletion in Queue
3.Traverse/Display Queue
4.Exit
Enter your choice(1-4)1
*****
Enter element to insert=3
Position of Front is 0 and Rear is 1
*****
1.INSERTION in Queue
2.Deletion in Queue
3.Traverse/Display Queue
4.Exit
Enter your choice(1-4)1
*****

```

```
*****  
Enter element to insert=2  
Position of Front is 0 and Rear is 2  
*****  
1.INSERTION in Queue  
2.Deletion in Queue  
3.Traverse/Display Queue  
4.Exit  
Enter your choice(1-4)3  
*****  
Front->5<-3<-2<-Rear  
*****  
1.INSERTION in Queue  
2.Deletion in Queue  
3.Traverse/Display Queue  
4.Exit  
Enter your choice(1-4)2  
*****  
Element deleted from the Front of the Queue i.e. 5  
Position of Front is 0 and Rear is 1  
*****  
1.INSERTION in Queue  
2.Deletion in Queue  
3.Traverse/Display Queue  
4.Exit  
Enter your choice(1-4)3  
*****  
Front->3<-2<-Rear  
*****
```

```
1.INSERTION in Queue  
2.Deletion in Queue  
3.Traverse/Display Queue  
4.Exit  
Enter your choice(1-4)2  
*****  
Element deleted from the Front of the Queue i.e. 3  
Position of Front is 0 and Rear is 0  
*****  
1.INSERTION in Queue  
2.Deletion in Queue  
3.Traverse/Display Queue  
4.Exit  
Enter your choice(1-4)2  
*****  
The only element in Queue is deleted i.e. 2  
Position of Front is -1 and Rear is -1  
*****  
1.INSERTION in Queue  
2.Deletion in Queue  
3.Traverse/Display Queue  
4.Exit  
Enter your choice(1-4)3  
*****  
Queue is Empty or Memory underflow so nothing to display/traverse  
*****
```

PRINT-39: S.No.-133

To convert the first letter of each word to uppercase and remaining to lowercase

```
##### PROGRAM PRINT-39 #####
#SNO133-To convert first letter of each word in capital and rest all in small
#AUTHOR-KONICAA SHARMA
try:
    f=open("file132.txt","r+")
    print ("BEFORE CHANGE")
    whole=f.read()
    print (whole)
    f.seek(0)
    flag=1
    while True:
        pos=f.tell()
        ch=f.read(1)
        if ch!='$':
            if ch==' ' or ch=='\n' or ch=='. ' or ch==', ':
                flag=1
            else:
                if flag==1:
                    ch=ch.upper()
                    flag=0
                elif flag==0:
                    ch=ch.lower()
        f.seek(pos)
        f.write(ch)
    else:
        break
    f.flush()
    print("File After Change")
    f.seek(0)
    whole=f.read()
    print(whole)
    f.close()
except:
    print("FILE NOT FOUND")
```

```
>>>
=====
RESTART: C:\Users\Hp\Documents\STACK\sno133.py =====
BEFORE CHANGE
Work Is Worship$

File After Change
Work Is Worship$
```

PRINT-40: S.No.-134
To replace a character by another maintaining the case

```
##### PROGRAM PRINT-40 #####
#SNo-134:- To replace a character by another maintaining the case
#AUTHOR-KONICAA SHARMA
try:
    f=open('poem1.txt','r+')
    print('File Before Change:')
    whole=f.read()
    print(whole)
    final=f.tell()
    f.seek(0)
    ch_old=input('Enter character to change=')
    ch_new=input('Enter new character to replace with=')
    for i in range(0,final):
        ch=f.read(1)
        if ch==ch_old:
            if ch_old.isupper():
                ch=ch_new.upper()
            else:
                ch=ch_new.lower()
        if ch=='\n':
            pass
        else:
            f.seek(i)
            f.write(ch)
        f.flush()
    f.seek(0)
    print('File after change:')
    whole=f.read()
    print(whole)
    f.close()
except:
    print('File not found')
```

```
===== RESTART: C:/Users/Hp/Documents/DATA FILE HANDLING/sno134.py ===
File Before Change:
The purpose of our lives is to be happy

Enter character to change=p
Enter new character to replace with=o
File after change:
The ouroose of our lives is to be haooy
>>> |
```

PRINT 41
SAMPLE PROJECT ON CONNECTIVITY OF MYSQL AND PYTHON

```
import mysql.connector as my
def create_database(): # To create a database
    con=my.connect(host="localhost",user="root",passwd="admin")
    if con.is_connected():
        print("Connection is established")
        c=con.cursor()
        sql='create database sample'
        c.execute(sql)
def create_table(): # To create table
    con=my.connect(host="localhost",user="root",passwd="admin")
    if con.is_connected():
        print("Connection is established")
        c=con.cursor()
        sql='use sample'
        c.execute(sql)
        sql='Create table fastfood \
            (fno integer primary key, \
            fname varchar(20) not null ,\
            type varchar(20),\
            price integer)'
        c.execute(sql)
        sql='show tables'
        c.execute(sql)
        data=c.fetchall()
        print(data)
        c.execute('describe fastfood')
        data=c.fetchall()
        for i in data:
            print(i)

def insert_rows(): # insert rows in existing tables
    try:
        con=my.connect(host="localhost",user="root",passwd="admin",database='sample')
        c=con.cursor()
        while True:
            fno=int(input('Enter Food No='))
            fname=input('Enter Food Name=')
            t=input('Enter Food Type (North Indian/Chinese/Continental/South Indian)=')
            price=int(input('Enter price='))
```

```

#           sql="insert into fastfood(fno,fname,type,price)values({},'{}','{}',{})".format(fno,fname,t,price)
#           sql="insert into fastfood(fno,fname,type,price)values(%d,'%s','%s',%d)%(fno,fname,t,price)
c.execute(sql)
more=input("You want to enter more(y/n)?")
if more.upper() !='Y':
    break
con.commit()
except Exception as e:
    print(e)

def retrieveall():
try:
    con=my.connect(host="localhost",user="root",passwd="admin",database='sample')
    c=con.cursor()
    sql='select * from fastfood'
    c.execute(sql)
    data=c.fetchall()
    for i in data :
        print(i)
except Exception as e:
    print(e)

```

```

def retrieve_price() : # Retrieve according to price condition
try:
    con=my.connect(host="localhost",user="root",passwd="admin",database='sample')
    c=con.cursor()
    p=int(input("Enter Price ="))
    sql='select * from fastfood where price>{}'.format(p)
    c.execute(sql)
    data=c.fetchall()
    for i in data :
        print(i)
except Exception as e:
    print(e)

def retrieve_type(): # Retrieve according to type
try:
    con=my.connect(host="localhost",user="root",passwd="admin",database='sample')
    c=con.cursor()
    t=input("Enter type of food=")
    t='%'+t
    sql="select * from fastfood where type like '{}'".format(t)
#    sql="select * from fast food where type like '%{}'".format(t)

    c.execute(sql)
    data=c.fetchall()
    for i in data :
        print(i)
except Exception as e:
    print(e)

```

```
===== RESTART: C:\Users\Hp\Desktop\fast food practice.py
Enter Price =150
(101, 'Paneer Makhani', 'North Indian', 200)
(103, 'Dosa', 'South Indian', 250)
(104, 'Batter Fried Fish', 'Continental', 500)
Enter type of food=North Indian
(101, 'Paneer Makhani', 'North Indian', 200)
Enter Food Number=102
(102, 'Chowmein', 'Chinese', 100)
Enter type of Food =Continental
1
deleted 1 rows
Enter type of Food =North Indian
Enter % of price to be increased=10
1
Updated 1 rows
[(101, 'Paneer Makhani', 'North Indian', 220)]
```

MYSQL QUERIES

Q17.

```
mysql> desc worker;
```

Field	Type	Null	Key	Default	Extra
ecode	int(11)	YES		NULL	
name	varchar(20)	YES		NULL	
desig	varchar(18)	YES		NULL	
plevel	char(4)	YES		NULL	
doj	date	YES		NULL	
dob	date	YES		NULL	

6 rows in set (0.01 sec)

```
mysql> select * from worker;
```

ecode	name	desig	plevel	doj	dob
11	sachin patel	supervisor	P001	2004-09-13	1985-06-23
12	chander nath	operator	P003	2010-02-22	1987-05-14
13	fizza	operator	P003	2009-06-14	1983-10-14
15	ameen ahmed	mechanic	P002	2006-08-21	1984-03-13
18	sanya	clerk	P002	2005-12-19	1983-06-09

5 rows in set (0.01 sec)

```
mysql> desc paylevel;
```

Field	Type	Null	Key	Default	Extra
plevel	char(4)	YES		NULL	
pay	char(5)	YES		NULL	
allowance	int(11)	YES		NULL	

3 rows in set (0.01 sec)

```
mysql> select * from paylevel;
```

plevel	pay	allowance
P001	26000	12000
P002	22000	10000
P003	12000	6000

3 rows in set (0.17 sec)

(a)To display details of all workers in descending order of DOB

```
mysql> SELECT * FROM WORKER ORDER BY DOB DESC;
+-----+-----+-----+-----+-----+
| ecode | name      | desig    | plevel | doj      | dob      |
+-----+-----+-----+-----+-----+
|   12  | chander nath | operator | P003   | 2010-02-22 | 1987-05-14 |
|   11  | sachin patel | supervisor | P001   | 2004-09-13 | 1985-06-23 |
|   15  | ameen ahmed  | mechanic  | P002   | 2006-08-21 | 1984-03-13 |
|   13  | fizza        | operator  | P003   | 2009-06-14 | 1983-10-14 |
|   18  | sanya        | clerk     | P002   | 2005-12-19 | 1983-06-09 |
+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

(b)To display the PLEVEL and number of workers in that PLEVEL

```
mysql> SELECT PLEVEL,COUNT(*) FROM WORKER GROUP BY PLEVEL;
+-----+-----+
| PLEVEL | COUNT(*) |
+-----+-----+
| P001   |      1 |
| P002   |      2 |
| P003   |      2 |
+-----+-----+
3 rows in set (0.18 sec)
```

(c)To display the PLEVEL and number of workers in that PLEVEL whose pay is greater than 15000

```
mysql> SELECT WORKER.PLEVEL,COUNT(*) FROM WORKER,PAYLEVEL WHERE PAYLEVEL.PAY>15000
      -> AND PAYLEVEL.PLEVEL=WORKER.PLEVEL GROUP BY PLEVEL;
+-----+-----+
| PLEVEL | COUNT(*) |
+-----+-----+
| P001   |      1 |
| P002   |      2 |
+-----+-----+
2 rows in set (0.03 sec)
```

(d)To display NAME and DESIG of those Workers whose PLEVEL is either P001 or P002

```
mysql> SELECT NAME,DESIG FROM WORKER WHERE PLEVEL!='P003';
+-----+-----+
| NAME      | DESIG    |
+-----+-----+
| sachin patel | supervisor |
| ameen ahmed  | mechanic   |
| sanya       | clerk     |
+-----+-----+
3 rows in set (0.01 sec)
```

Q18.

```
mysql> desc event;
```

Field	Type	Null	Key	Default	Extra
eventid	char(3)	YES		NULL	
event	varchar(20)	YES		NULL	
numofperformers	char(2)	YES		NULL	
celebrityid	char(4)	YES		NULL	

4 rows in set (0.01 sec)

```
mysql> select * from event;
```

eventid	event	numofperformers	celebrityid
101	birthday	10	C102
102	PROMOTION PARTY	20	C103
103	engagement	12	C102
104	wedding	15	C104

4 rows in set (0.00 sec)

```
mysql> desc celebrity;
```

Field	Type	Null	Key	Default	Extra
celebrityid	char(4)	YES		NULL	
name	varchar(30)	YES		NULL	
phone	decimal(10,0)	YES		NULL	
feecharged	decimal(10,0)	YES		NULL	

4 rows in set (0.00 sec)

```
mysql> select * from celebrity;
```

celebrityid	name	phone	feecharged
C101	Faiz khan	9912745382	200000
C102	sanjay kumar	893466448	250000
C103	neera khan kapoor	981166568	300000
C104	reena bhatia	912745382	100000

4 rows in set (0.00 sec)

(a) To display EventId, Event name, CelebrityID, and names of celebrities for only those events that have more than 10 performers

```
mysql> select eventid,event,celebrity.celebrityid,name from event,celebrity  
-> where event.celebrityid=celebrity.celebrityid and numofperformers>10;  
+-----+-----+-----+  
| eventid | event | celebrityid | name |  
+-----+-----+-----+  
| 103 | engagement | C102 | sanjay kumar |  
| 102 | PROMOTION PARTY | C103 | neera khan kapoor |  
| 104 | wedding | C104 | reena bhatia |  
+-----+-----+-----+  
3 rows in set (0.00 sec)
```

(b) To display Event name ,celebrity id and names of celebrities who have ‘Khan’ anywhere in their names

```
mysql> select event,celebrity.celebrityid,name from event,celebrity  
-> WHERE event.celebrityid=celebrity.celebrityid  
-> and name like '%KHAN%';  
+-----+-----+-----+  
| event | celebrityid | name |  
+-----+-----+-----+  
| PROMOTION PARTY | C103 | neera khan kapoor |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

(c) To display Event name, names of celebrities and number of performers who charge more than 200000

```
mysql> select event,name,numofperformers from event,celebrity  
-> WHERE event.celebrityid=celebrity.celebrityid  
-> and feecharged>200000;  
+-----+-----+-----+  
| event | name | numofperformers |  
+-----+-----+-----+  
| birthday | sanjay kumar | 10 |  
| PROMOTION PARTY | neera khan kapoor | 20 |  
| engagement | sanjay kumar | 12 |  
+-----+-----+-----+  
3 rows in set (0.00 sec)
```

Q19.

```
mysql> desc teacher;
```

Field	Type	Null	Key	Default	Extra
TeacherID	char(4)	YES		NULL	
Tname	varchar(20)	YES		NULL	
City	varchar(20)	YES		NULL	
Subject	varchar(20)	YES		NULL	
Qualification	varchar(20)	YES		NULL	
Designation	varchar(20)	YES		NULL	
Pay	int(11)	YES		NULL	

```
7 rows in set (0.16 sec)
```

```
mysql> desc student;
```

Field	Type	Null	Key	Default	Extra
StdID	char(4)	YES		NULL	
Name	varchar(20)	YES		NULL	
FName	varchar(20)	YES		NULL	
Stream	varchar(20)	YES		NULL	
TeacherID	char(4)	YES		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> select * from teacher;
```

TeacherID	Tname	City	Subject	Qualification	Designation	Pay
1021	Anand Mathur	Mumbai	Maths	PGT	HOD	75000
1022	Neha	Delhi	English	MBA	Academic Advisor	100000
1023	Preeti	Pune	Maths	PGT	Administrator	100000
1024	Samita	Delhi	History	PGT	HOD	150000
1022	Kavya	Delhi	English	MBA	Academic Advisor	150000

```
5 rows in set (0.00 sec)
```

```
mysql> select * from student;
```

StdID	Name	FName	Stream	TeacherID
2021	Amit	Raj Kishore	Commerce	1021
2022	Amesh Sharma	Kishor Chand	Science	1022
2023	Rahul	Rajeev Sharma	Science	1023

```
3 rows in set (0.00 sec)
```

(a) Show the name of students enrolled in Science stream

```
mysql> select name from student where stream='Science';
+-----+
| name |
+-----+
| Amesh Sharma |
| Rahul |
+-----+
```

(b) Count the number of students in Commerce students

```
mysql> select count(name) from student where stream='Commerce';
+-----+
| count(name) |
+-----+
|      1      |
+-----+
1 row in set (0.00 sec)
```

(c) Count the number of teachers in each designation

```
mysql> select designation,count(*) from teacher group by designation;
+-----+-----+
| designation | count(*) |
+-----+-----+
| Academic Advisor |      2 |
| Administrator     |      1 |
| HOD              |      2 |
+-----+-----+
3 rows in set (0.00 sec)
```

(d) Display the maximum pay of teacher who is teaching English

```
mysql> select max(pay) from teacher where subject='English';
+-----+
| max(pay) |
+-----+
| 150000  |
+-----+
1 row in set (0.00 sec)
```

(e) Display the name of students who are taught by “Anand Mathur”

```
mysql> select name from student where teacherid=(select teacherid
   -> from teacher where tname='Anand Mathur');
+-----+
| name |
+-----+
| Amit |
+-----+
1 row in set (0.00 sec)
```

(f)Display the name and designation of teachers who are teaching a student named “Amit”

```
mysql> select tname,designation from teacher where teacherid=(select
-> teacherid from student where name='Amit');
+-----+-----+
| tname | designation |
+-----+-----+
| Anand Mathur | HOD |
+-----+-----+
1 row in set (0.00 sec)
```

(g)Find out the name of teacher who is getting the highest pay

```
mysql> select tname from teacher where pay=(select max(pay)
-> from teacher);
+-----+
| tname |
+-----+
| Samita |
| Kavya |
+-----+
2 rows in set (0.01 sec)
```

(h)Find out the city of teachers who are teaching math subject

```
mysql> select city from teacher where subject='maths';
+-----+
| city |
+-----+
| Mumbai |
| Pune |
+-----+
2 rows in set (0.00 sec)
```

(i)Find out the name of teacher who is getting the lowest salary among PGTs

```
mysql> select tname from teacher where pay=(select min(pay) from
-> teacher where qualification='PGT');
+-----+
| tname |
+-----+
| Anand Mathur |
+-----+
1 row in set (0.00 sec)
```

(j)Display the list of students who are taught by PGTs only

```
mysql> select name from student,teacher where
-> student.teacherid=teacher.teacherid and qualification='PGT';
+-----+
| name |
+-----+
| Amit |
| Rahul |
+-----+
2 rows in set (0.00 sec)
```