INDEX

8.	Write a user defined function SWAP2CHANGE () which takes list as arguments in python to modify the content of the list in such a way that the elements, which are multiples of 10 swap with the value present in the very next position in the list. Invoke the function along with the list.	16-17	
9.	Write a function onedigit( ) that takes two numbers as input and returns the number that has minimum one’s digit. Read the numbers from the user and invoke the function . Also display the number that is returned having minimum one’s digit.	18-19	
10.	Write a program in python using user defined functions smallest() and largest() to find the smallest or largest among three numbers. Read the numbers from the user and invoke the function based on the user choice.	20-21	
11.	Write a program in python to write lines We work, we try to be better We work with full Zest But, why is that we just don’t know any letter in to the text file poem.txt. Also read the content of text file line by line and display each word separated by #.	22-23	
12.	Write a python program to add a string “ An Aeroplane is in the Sky 12@#45 flying to @ Chennai” to the file article.txt Also read the content of the file and count and display the number of alphabets, digits, uppercase letters, lowercase letters, spaces and other characters present in the text file article .txt.	24-26	
13.	Write a program in python to create binary file employee.dat with 4 records containing the following information employee no, employee name and salary. Also read the content of the binary file and display all those employees whose salary is above 25000 and also count and display the no of employees whose salary is above 25000. If not found then display appropriate message.	27-28	

14.	Write a function in python to update the starting point as Chennai, whose destination is “Cochin” from binary file “Bus.Dat”. Assuming the binary file is containing the following elements in the list: 1. Bus Number 2. Bus Starting Point 3. Bus Destination Write a complete program to create a file bus.dat with 4 records and invoke the function.	29-31	
15.	Write a program in python to create a csv file product.csv with 4 records containing the information product id, product name, price. Also read the content of the file and display all the records from product.csv whose price is more than 300.	32-33	
16.	Write a program to implement a stack for those book details (book no, book name). That is, now each item node of the stack contains two types of information – a book no and book name. Implement push and display operations.	34-35	
17.	Write a menu driven program to input a list N=[12,20,34,56,55,79,98,22,35,60] and perform the following operations as a separate user defined functions:舵 • Push the elements of the list into the stack which are divisible by 5. • Pop the content of the stack. • Display the content of the stack after insertion.	36-37	
18.	Write a menu driven program to input the below given dictionary details and perform the following operations on a stack as a separate user defined functions. R = {“OM”:76, “JAI”:45, “BOB”:89, “ALI”:65, “ANU”:90, “TOM”:82} • Push the keys (name of the student) of the dictionary into a stack, where the corresponding value (marks) is greater than 75. • Pop the content of the stack. • Display the content of the stack after insertion and deletion.	38-42	

19.	Write SQL commands for (a) to (f) and write output for (g): (a) To show all information about the students of History department. (b) To list the names of female students who are in Hindi department. (c) To list names of all students with their DOA in ascending order. (d) To display student’s Name, Fee, Age for male Students only. (e) To count the no. of student with Age<23. (f) To insert a new row in the STUDENT table with the following data: 9, “Zaheer”,36, “Computer”, {12/03/97}, 230, “M” (g) Give the output of following SQL statements: (i)Select COUNT (distinct department) from STUDENT; (ii)Select MAX (Age) from STUDENT where Sex = “F”; (iii) Select AVG (Fee) from STUDENT where Dateofadm < {01/01/98}; (iv) Select SUM (Fee) from STUDENT where Dateofadm < {01/01/98};	43-46	
20.	Write SQL queries for (a) to (f): (a) To show Book name, Author name and Price of books of First Publ. publishers. (b) To list the names from books of Text type. (c) To display the names and price from books in ascending order of their price. (d) To increase the price of all books of EPB Publishers by 50. (e) To display the Book_Id, Book_Name and Quantity_Issued for all books which have been issued. (The query will require contents from both the tables.) (f) To insert a new row in the table Issued having the following data: “F0003”,1 (g) Give the output of the following queries : (i)SELECT COUNT(*) FROM Books; (ii)SELECT MAX(Price) FROM Books WHERE Quantity >= 15;	47-49	

	(iii)SELECT Book_Name, Author_Name FROM Books WHERE Publishers = “EPB”; (iv)SELECT COUNT (DISTINCT Publishers) FROM Books WHERE Price >= 400;		

21.	Write SQL queries for (i) to (iv) and outputs for (v) to (viii):
	
	(i)To display Wno, Name, Gender from the table WORKER in descending order of Wno.
	(ii)To display the Wno and Name of all the FEMALE workers from the table WORKER.
	(iii)To display the Wno and Name of those workers from the table WORKER who are born between ‘1987-01-01’ and ‘1991-12-01’. (iv)To count and display MALE workers who have joined after ‘1986-01-01’.
	(v)SELECT COUNT(*), DCODE FROM WORKER GROUP BY DCODE HAVING COUNT(*)>1;
	(vi) SELECT DISTINCT DEPARTMENT FROM DEPT;
	(vii) SELECT NAME, DEPARTMENT, CITY FROM WORKER W, DEPT D WHERE W.DCODE = D.DCODE AND WNO<1003; (vii) SELECT MAX(DOJ), MIN(DOB) FROM WORKER;
		50-54	

22.	Write SQL Queries for the following: (i)To display employee ids, names of employees, job ids with corresponding job titles. (ii)To display names of employees, sales and corresponding job titles who have achieved sales more than 1300000. (iii)To display names and corresponding job titles of those employees who have ‘SINGH’ (anywhere) in their names. (iv)Identify foreign key in the table EMPLOYEE. (v) Write SQL command to change the JOBID to 104 of the EMPLOYEE with ID as E4 in the table ‘EMPLOYEE’.	55-57	
23.	Write SQL commands for the statements (i) to (iv) and outputs for SQL queries (v) to (vii): (i) To display the frequency of employees department wise. (ii) To list the names of those employees only whose name starts with ‘H’. (iii) To add a new column in salary table. The column name is Total_Sal. (iv) To store the corresponding values in the Total_Sal column. (v)	58-61	

	Select max(Basic) from Salary where Bonus>40;
	(vi) Select count(*) from Employee group by Sex;
	(vii) Select Distinct Depid from Employee;		

24.	Write a program to connect python with mysql database connectivity and perform the following operations on data in database. **Table name: Student**	62-65	
25.	Write a program to connect python with mysql database connectivity and perform the following operations on data in database. **Table name: Employee**	66-69	

1. Write a program that reads a line and print its statistics like

No of Uppercase letters

No of Lowercase letters

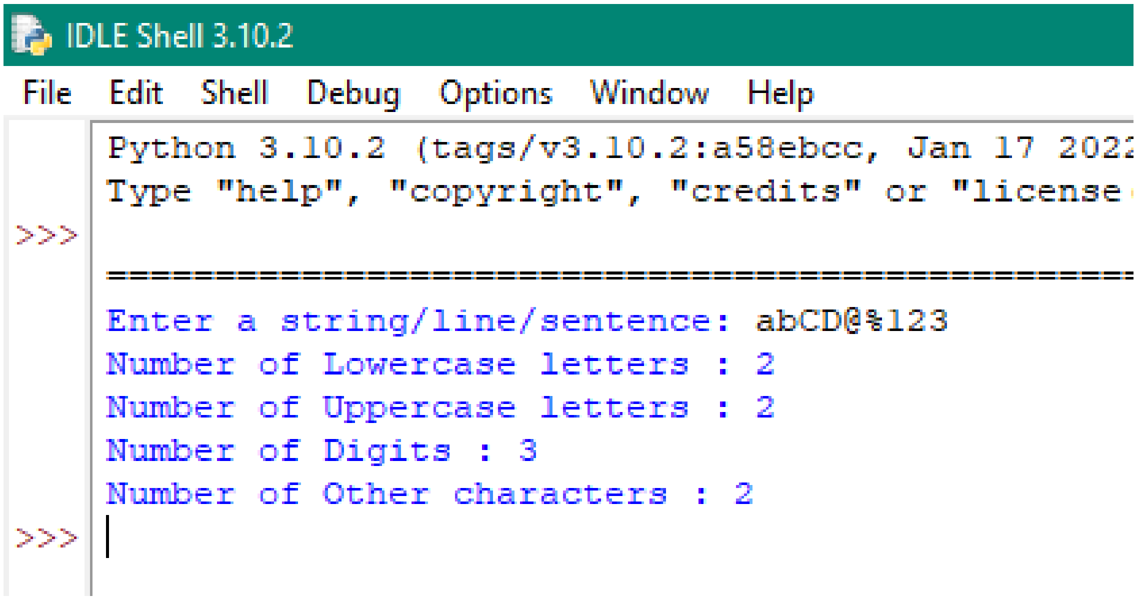
No of digits

Other Characters

Program code:

```
line = input("Enter a string/line/sentence: ")
lowercharcount = uppercharcount = 0
other = digitcount = 0
for a in line:
  if a.islower():
    lowercharcount += 1
  elif a.isupper():
    uppercharcount += 1
  elif a.isdigit():
    digitcount += 1
  else:
      other += 1
print("Number of Lowercase letters :",lowercharcount)
print("Number of Uppercase letters :",uppercharcount)
print("Number of Digits :", digitcount)
print("Number of Other characters :", other)
```

Output :-



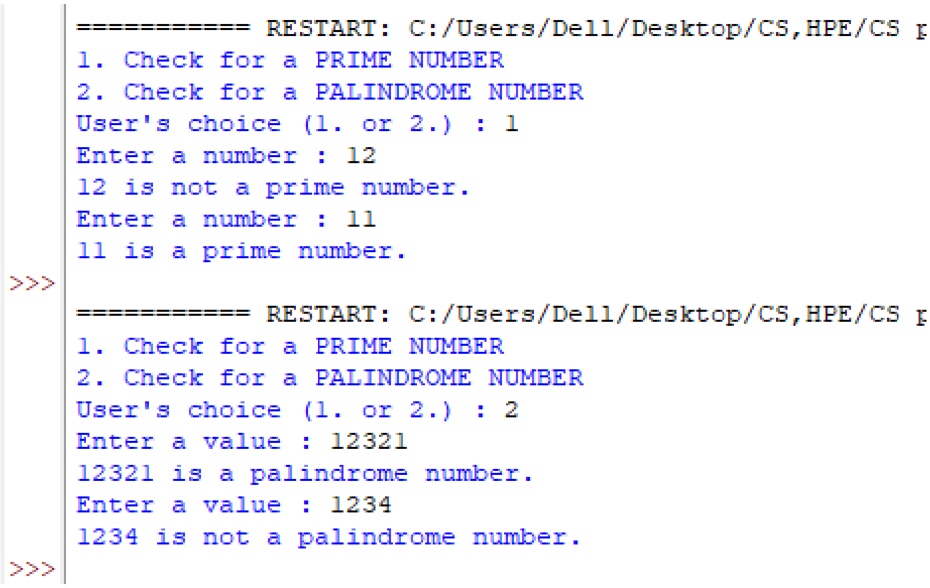
```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022
Type "help", "copyright", "credits" or "license
>>>
=================================================
Enter a string/line/sentence: abCD@%123
Number of Lowercase letters : 2
Number of Uppercase letters : 2
Number of Digits : 3
Number of Other characters : 2
>>>
```

2. Write a menu driven program in python which takes an integer number as input and as per the user choice checks whether the number is Prime or a Palindrome Number.

Program code:

```
print("1. Check for a PRIME NUMBER")

print("2. Check for a PALINDROME NUMBER")

ch = int(input("User's choice (1. or 2.) : "))

for i in range(2):

    if ch==1:

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

        if num%2==0:

            print(num, "is not a prime number.")

        else:

            print(num, "is a prime number.")


    else:

        val = input("Enter a value : ")

        if val == val[::-1]:

            print(val, "is a palindrome number.")

        else:

            print(val, "is not a palindrome number.")
```

Output :-



```
    ============ RESTART: C:/Users/Dell/Desktop/CS,HPE/CS p
    1. Check for a PRIME NUMBER
    2. Check for a PALINDROME NUMBER
    User's choice (1. or 2.) : 1
    Enter a number : 12
    12 is not a prime number.
    Enter a number : 11
    11 is a prime number.
>>>
    ============ RESTART: C:/Users/Dell/Desktop/CS,HPE/CS p
    1. Check for a PRIME NUMBER
    2. Check for a PALINDROME NUMBER
    User's choice (1. or 2.) : 2
    Enter a value : 12321
    12321 is a palindrome number.
    Enter a value : 1234
    1234 is not a palindrome number.
>>>
```

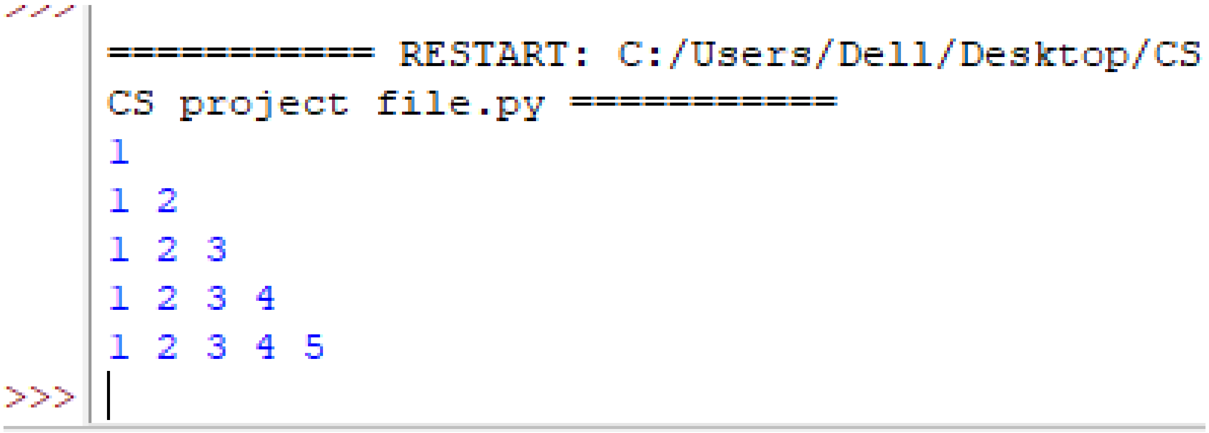
3. Write a program in python to display the given pattern.

```
1

1  2

1  2  3

1  2  3  4

1  2  3  4  5
```

Program code:

```
num = 6

for n in range(1, num):

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

        print(i, end=" ")

    print(" ")
```

Output :-



```
=========== RESTART: C:/Users/Dell/Desktop/CS
CS project file.py ===========
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
>>> 
```

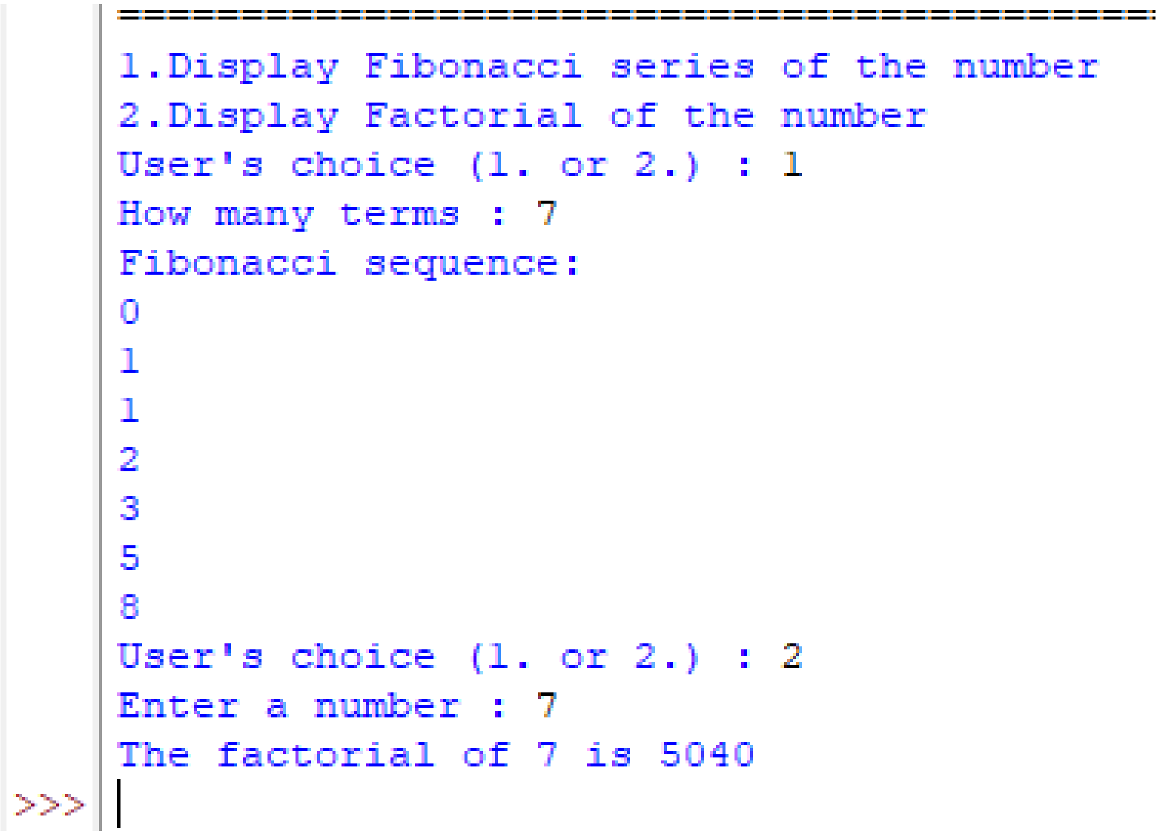
4. Write a menu driven program in python to display Fibonacci Series of a given number or a Factorial of a given number.

Program code:

```
print("1.Display Fibonacci series of the number")

print("2.Display Factorial of the number")

for i in range(2):

    ch = int(input("User's choice (1. or 2.) : "))

    if ch==1:

        nterms = int(input("How many terms : "))

        n1, n2 = 0, 1

        count = 0

        print("Fibonacci sequence:")

        while count < nterms:

            print(n1)

            nth = n1 + n2

            n1 = n2

            n2 = nth

            count += 1

    else:

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

        factorial = 1

        if num < 0:

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

```
elif num == 0:
    print("The factorial of 0 is 1")
else:
    for i in range(1,num + 1):
        factorial = factorial*i
    print("The factorial of",num,"is",factorial)
```

Output :-



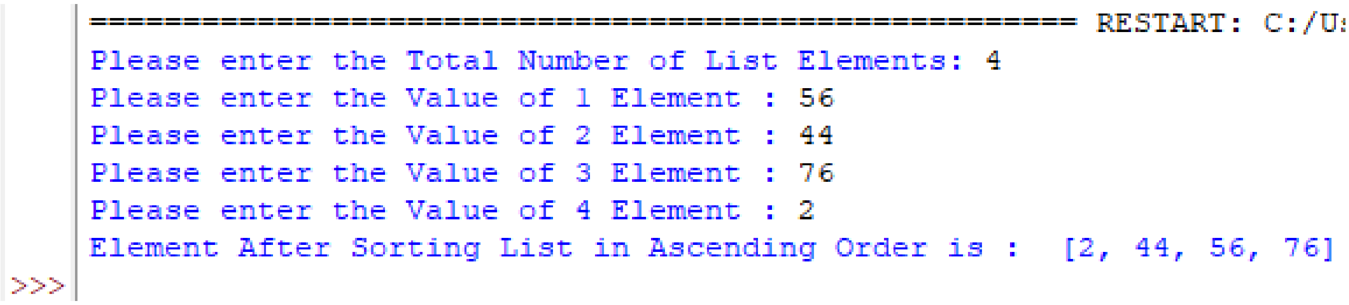
```
==========================================
1.Display Fibonacci series of the number
2.Display Factorial of the number
User's choice (1. or 2.) : 1
How many terms : 7
Fibonacci sequence:
0
1
1
2
3
5
8
User's choice (1. or 2.) : 2
Enter a number : 7
The factorial of 7 is 5040
>>> 
```

5. Write a program in python to read a list and sort the values of list without using built in functions.

Program code:

```
NumList = []
Number = int(input("Please enter the Total Number of List Elements: "))
for i in range(1, Number + 1):
    value = int(input("Please enter the Value of %d Element : " %i))
    NumList.append(value)


for i in range (Number):
    for j in range(i + 1, Number):
        if(NumList[i] > NumList[j]):
            temp = NumList[i]
            NumList[i] = NumList[j]
            NumList[j] = temp


print("Element After Sorting List in Ascending Order is : ", NumList)
```

Output :-



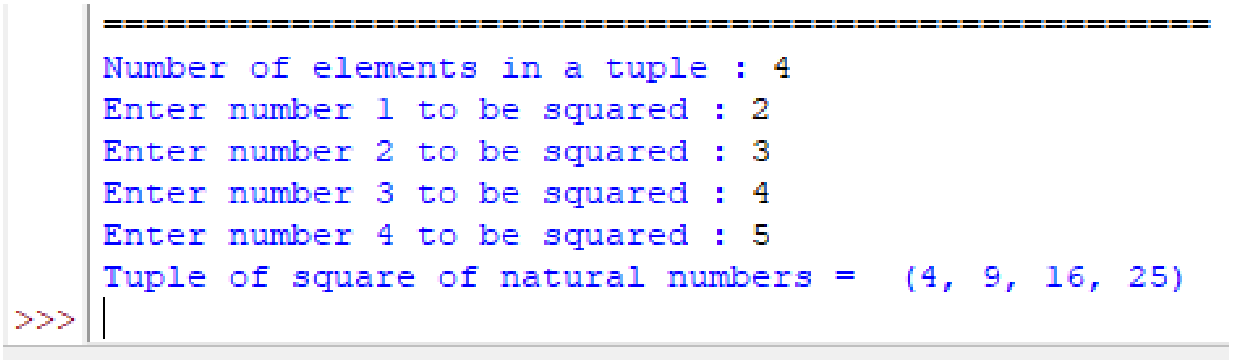
```
===================================================== RESTART: C:/U
Please enter the Total Number of List Elements: 4
Please enter the Value of 1 Element : 56
Please enter the Value of 2 Element : 44
Please enter the Value of 3 Element : 76
Please enter the Value of 4 Element : 2
Element After Sorting List in Ascending Order is :  [2, 44, 56, 76]
>>>
```

6. Write a python program to create a tuple storing the square of n natural numbers.

Program code:

```
tup = ()
n = int(input("Number of elements in a tuple : "))
for i in range(1, n+1):
    val = int(input("Enter number %d to be squared : "%i))
    tup += (val*val,)
print("Tuple of square of natural numbers = ",tup)
```

Output :-



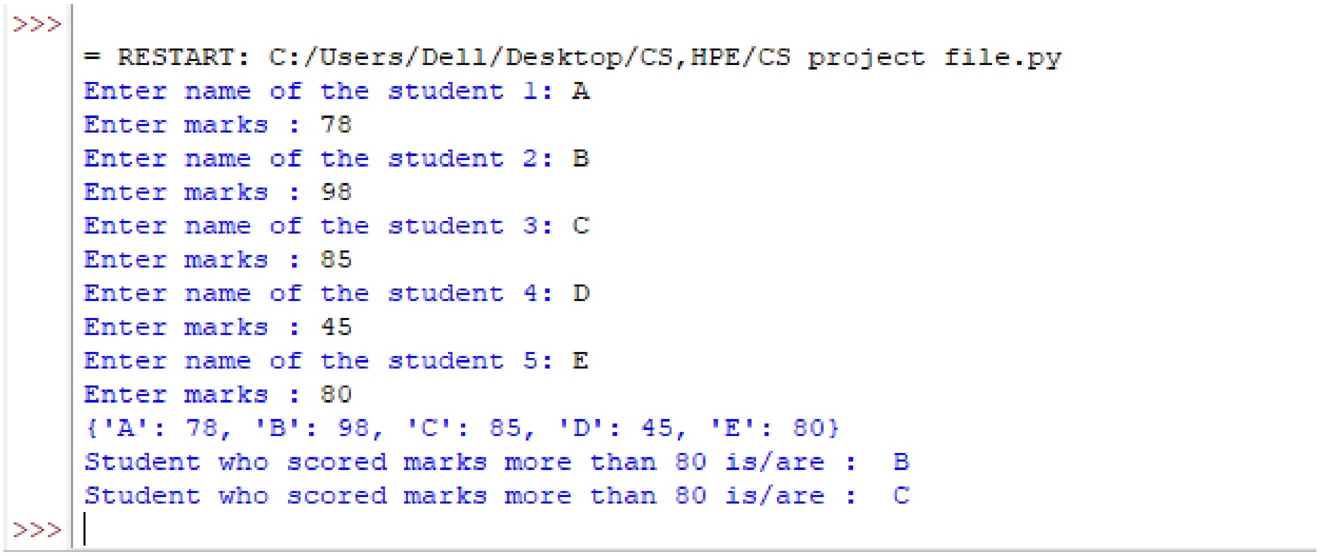
```
=====================================================
Number of elements in a tuple : 4
Enter number 1 to be squared : 2
Enter number 2 to be squared : 3
Enter number 3 to be squared : 4
Enter number 4 to be squared : 5
Tuple of square of natural numbers =  (4, 9, 16, 25)
>>> 
```

7. Write a program to create a student dictionary containing the name(key) and marks(value) of 5 students taken from the user. Also display the name of the student whose marks is greater than 80. If not display appropriate message.

Program code:

```
dic = {}

for i in range(1,6):

    name = input("Enter name of the student %d: "%i)

    marks = eval(input("Enter marks : "))

    dic[name]= marks

print(dic)

for j in dic:

    if dic[j]>80:

        print("Student who scored marks more than 80 is/are : ",j)
```

Output :-



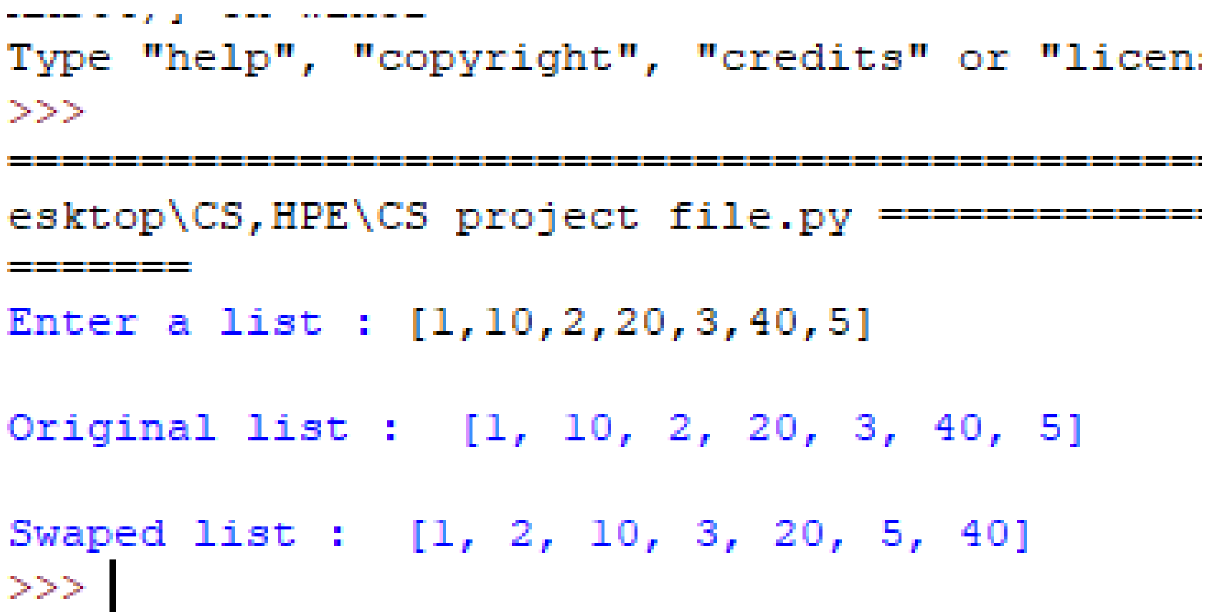
```
>>>
    = RESTART: C:/Users/Dell/Desktop/CS,HPE/CS project file.py
    Enter name of the student 1: A
    Enter marks : 78
    Enter name of the student 2: B
    Enter marks : 98
    Enter name of the student 3: C
    Enter marks : 85
    Enter name of the student 4: D
    Enter marks : 45
    Enter name of the student 5: E
    Enter marks : 80
    {'A': 78, 'B': 98, 'C': 85, 'D': 45, 'E': 80}
    Student who scored marks more than 80 is/are :  B
    Student who scored marks more than 80 is/are :  C
>>>
```

8. Write a user defined function SWAP2CHANGE () which takes list as arguments in python to modify the content of the list in such a way that the elements, which are multiples of 10 swap with the value present in the very next position in the list. Invoke the function along with the list.

Program code:

```
def SWAP2CHANGE(L):
    a = 1
    while a < len(L)-1:
        if L[a]%10==0:
            L[a],L[a+1] = L[a+1],L[a]
            a += 2
        else:
            a += 1
    return(L)
M = eval(input("Enter a list : "))
print()
print("Original list : ",M)
print()
N = SWAP2CHANGE(M)
print("Swaped list : ",N)
```

Output :-



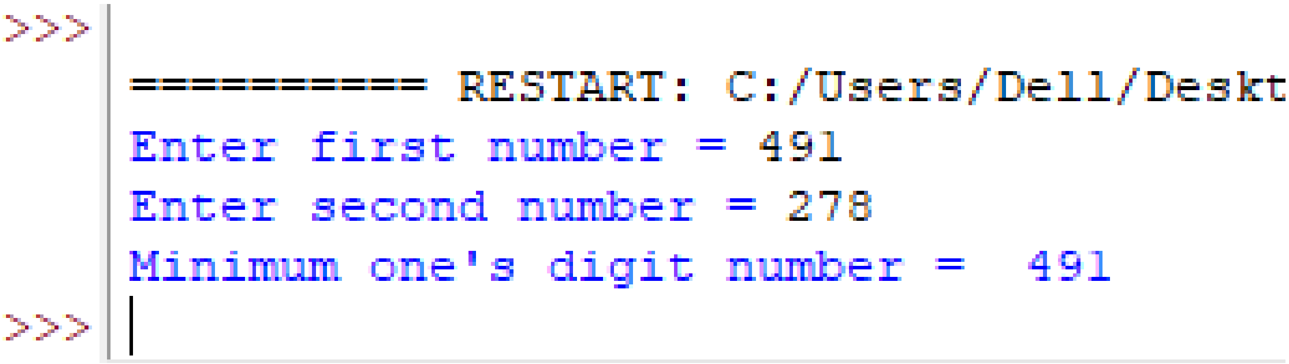
```
Type "help", "copyright", "credits" or "licen
>>>
==============================================
esktop\CS,HPE\CS project file.py ============
=======
Enter a list : [1,10,2,20,3,40,5]

Original list :  [1, 10, 2, 20, 3, 40, 5]

Swaped list :  [1, 2, 10, 3, 20, 5, 40]
>>> 
```

9. Write a function onedigit( ) that takes two numbers as input and returns the number that has minimum one’s digit. Read the numbers from the user and invoke the function . Also display the number that is returned having minimum one’s digit.

Program code:

```
def onedigit(x,y) :
    a = x % 10
    b = y % 10
    if a < b :
        return x
    else :
        return y
first = int(input("Enter first number = "))
second = int(input("Enter second number = "))
print ( "Minimum one's digit number = " , onedigit( first , second ) )
```

Output :-



```
>>> 
=========== RESTART: C:/Users/Dell/Deskt
Enter first number = 491
Enter second number = 278
Minimum one's digit number =  491
>>> 
```

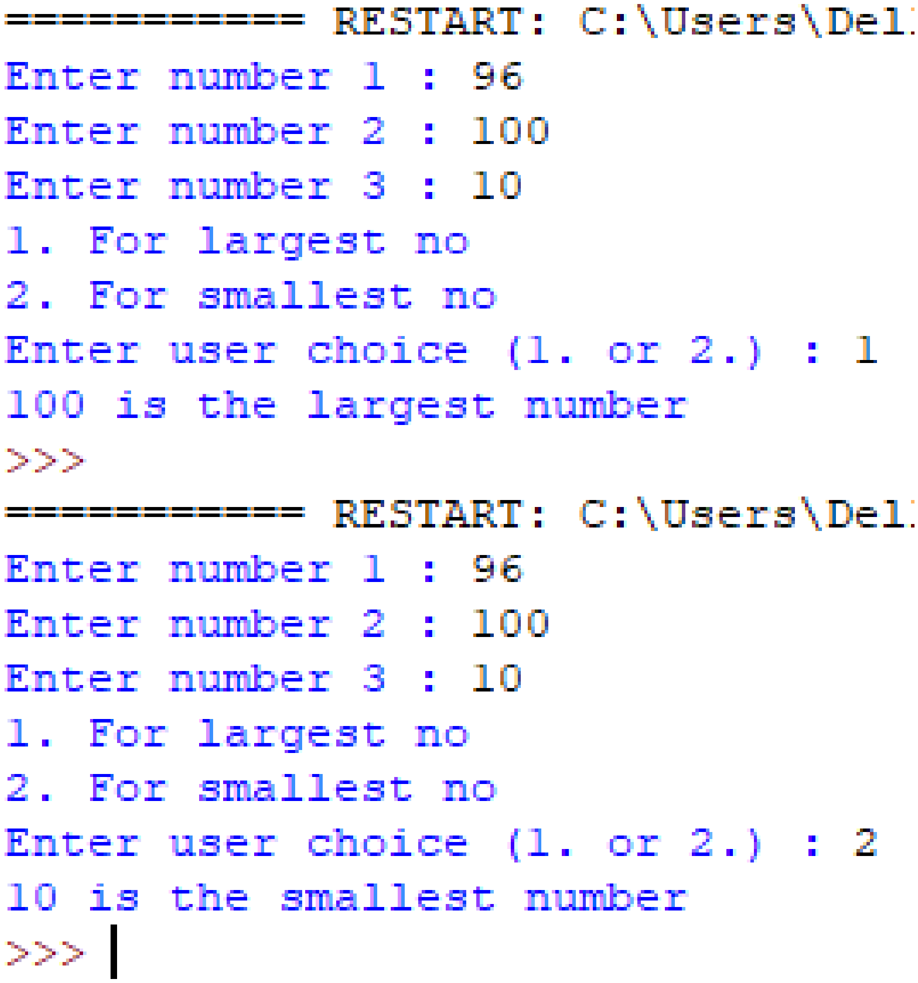
10. Write a program in python using user defined functions smallest() and largest() to find the smallest or largest among three numbers. Read the numbers from the user and invoke the function based on the user choice.

Program code:

```
def largest(a,b,c):
    if a>b and a>c:
        print(a,"is the largest number")
    elif b>a and b>c:
        print(b,"is the largest number")
    else:
        print(c,"is the largest number")
def smallest(a,b,c):
    if a<b and a<c:
        print(a,"is the smallest number")
    elif b<a and b<c:
        print(b,"is the smallest number")
    else:
        print(c,"is the smallest number")
a = int(input("Enter number 1 : "))
b = int(input("Enter number 2 : "))
c = int(input("Enter number 3 : "))
print('1. For largest no')
print('2. For smallest no')
```

```
ch = int(input('Enter user choice (1. or 2.) : '))

if ch==1:

   z=largest(a,b,c)

elif ch==2:

   f=smallest(a,b,c)
```

Output :-



```
=========== RESTART: C:\Users\Del
Enter number 1 : 96
Enter number 2 : 100
Enter number 3 : 10
1. For largest no
2. For smallest no
Enter user choice (1. or 2.) : 1
100 is the largest number
>>>
=========== RESTART: C:\Users\Del
Enter number 1 : 96
Enter number 2 : 100
Enter number 3 : 10
1. For largest no
2. For smallest no
Enter user choice (1. or 2.) : 2
10 is the smallest number
>>>
```

11. Write a program in python to write lines

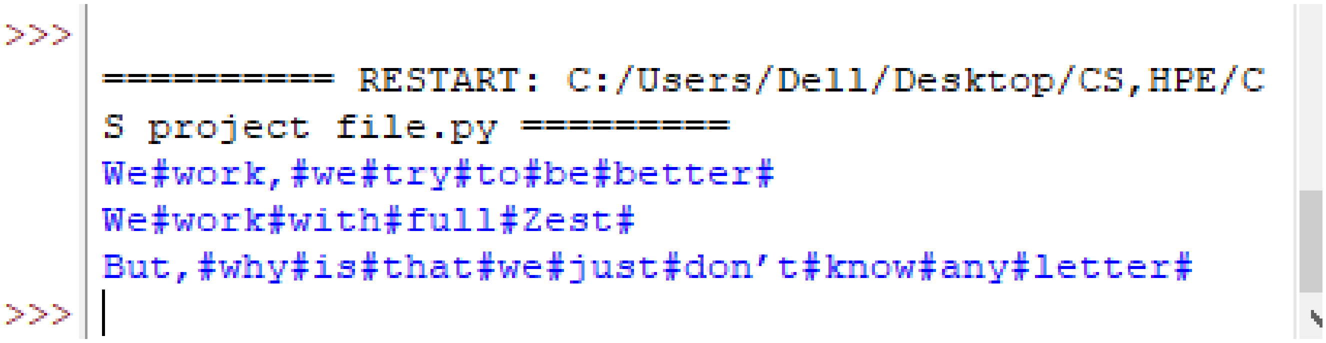
We work, we try to be better
We work with full Zest
But, why is that we just don’t know any letter

in to the text file poem.txt. Also read the content of text file line by line and display each word separated by #.

Program code:

```
f = open('poem.txt','w')

lines = ['We work, we try to be better\n','We work with full Zest\n','But, why is that we just don’t know any letter\n']

f.writelines(lines)

f.close()


k = open('poem.txt','r')

line = k.readlines()

for i in line:

    words = i.split()

    for j in words:

        print(j+'#',end='')

    print(' ')

k.close()
```

Output :-



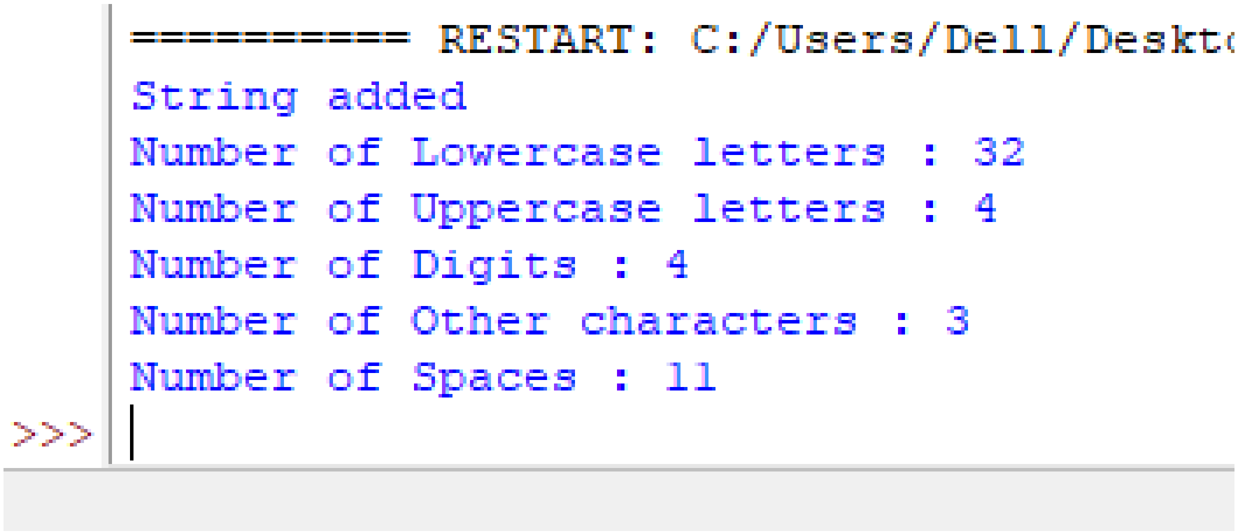
```
>>> 
========== RESTART: C:/Users/Dell/Desktop/CS,HPE/C
S project file.py =========
We#work,#we#try#to#be#better#
We#work#with#full#Zest#
But,#why#is#that#we#just#don't#know#any#letter#
>>> 
```

12. Write a python program to add a string “ An Aeroplane is in the Sky 12@#45 flying to @ Chennai” to the file article.txt Also read the content of the file and count and display the number of alphabets, digits, uppercase letters, lowercase letters, spaces and other characters present in the text file article .txt.

Program code:

```
f = open('article.txt','w')
a = f.write('An Aeroplane is in the Sky  12@#45 flying to @ Chennai')
print('String added')
f.close()

g = open('article.txt','r')
alpha=dig= 0
up=low=0
other=space=0
st = g.read()
for i in st:
    if i.islower():
        low += 1
    elif i.isupper():
        up += 1
    elif i.isdigit():
        dig += 1
    elif i.isalpha():
        alpha += 1
    elif i.isspace():
        space += 1
    else:
        other += 1
print("Number of Lowercase letters :",low)
print("Number of Uppercase letters :",up)
print("Number of Digits :",dig)
print("Number of Other characters :",other)
print("Number of Spaces :",space)
g.close()
```

Output :-



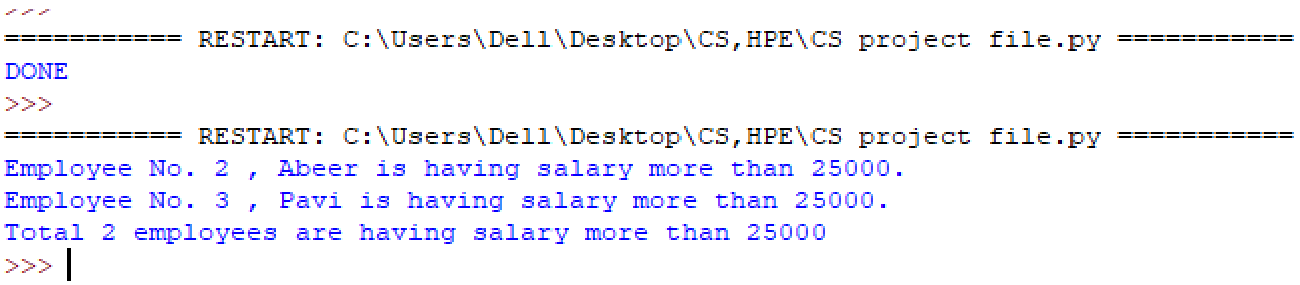
```
=========== RESTART: C:/Users/Dell/Deskt
String added
Number of Lowercase letters : 32
Number of Uppercase letters : 4
Number of Digits : 4
Number of Other characters : 3
Number of Spaces : 11
>>> 
```

13. Write a program in python to create binary file employee.dat with 4 records containing the following information employee no, employee name and salary. Also read the content of the binary file and display all those employees whose salary is above 25000 and also count and display the no of employees whose salary is above 25000. If not found then display appropriate message.

Program code:

```
import pickle

f = open('employee.dat','wb')

emp = [[1,'Tan',20000],[2,'Abeer',90000],[3,'Pavi',70000],[4,'Yashi',15000]]

pickle.dump(emp,f)

print('DONE')

f.close()


g = open('employee.dat','rb')

a = pickle.load(g)

count = 0

for i in a:

    if i[2]>25000:

        count += 1

        print('Employee No.',i[0],',',i[1],'is having salary more than 25000.')

print('Total',count,'employees are having salary more than 25000')

g.close()
```

Output :-



```
>>>
============ RESTART: C:\Users\Dell\Desktop\CS,HPE\CS project file.py ============
DONE
>>>
============ RESTART: C:\Users\Dell\Desktop\CS,HPE\CS project file.py ============
Employee No. 2 , Abeer is having salary more than 25000.
Employee No. 3 , Pavi is having salary more than 25000.
Total 2 employees are having salary more than 25000
>>> 
```

14. Write a function in python to update the starting point as Chennai, whose destination is “Cochin” from binary file “Bus.Dat”. Assuming the binary file is containing the following elements in the list:

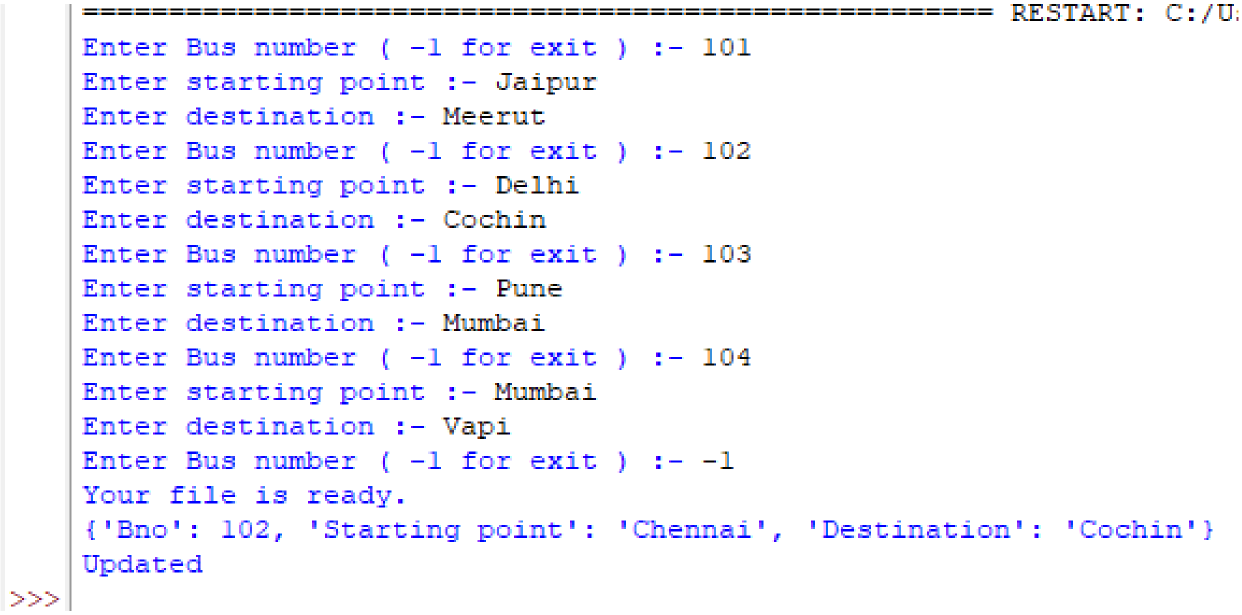
4. Bus Number
5. Bus Starting Point
6. Bus Destination

Write a complete program to create a file bus.dat with 4 records and invoke the function.

Program code:

```
import pickle

f = open("BUS.DAT","wb")

while True :

    bno = int(input("Enter Bus number ( -1 for exit ) :- "))

    if bno == -1 :

        break

    stpt = input("Enter starting point :- ")

    dest = input("Enter destination :- ")

    bus = { 'Bno': bno , 'Starting point': stpt , 'Destination' : dest }

    pickle.dump( bus , f)

print("Your file is ready.")

f.close()



def change():

    bus = {}

    f = False

    file = open("BUS.DAT","rb+")
    try:
        while True:
            rpos = file.tell()
            bus = pickle.load(file)
            if bus['Destination'] == 'Cochin':
                bus['Starting point'] = 'Chennai'
                file.seek(rpos)
                pickle.dump(bus,file)
                print(bus)
                f = True
    except EOFError:
        if f == False:
            print("Not found")
        else:
            print("Updated")
        file.close()
change()
```

Output :-



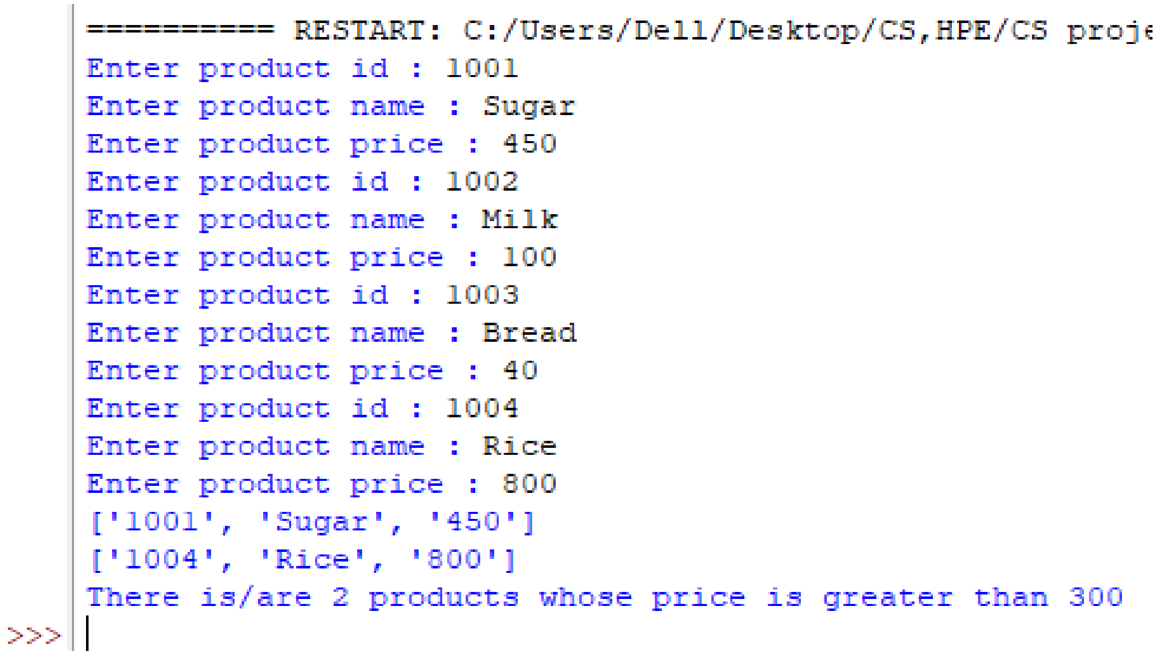
```
===================================================== RESTART: C:/U
Enter Bus number ( -1 for exit ) :- 101
Enter starting point :- Jaipur
Enter destination :- Meerut
Enter Bus number ( -1 for exit ) :- 102
Enter starting point :- Delhi
Enter destination :- Cochin
Enter Bus number ( -1 for exit ) :- 103
Enter starting point :- Pune
Enter destination :- Mumbai
Enter Bus number ( -1 for exit ) :- 104
Enter starting point :- Mumbai
Enter destination :- Vapi
Enter Bus number ( -1 for exit ) :- -1
Your file is ready.
{'Bno': 102, 'Starting point': 'Chennai', 'Destination': 'Cochin'}
Updated
>>>
```

15. Write a program in python to create a csv file product.csv with 4 records containing the information product id, product name, price. Also read the content of the file and display all the records from product.csv whose price is more than 300.

Program code:

```
import csv

fh = open('product.csv','w')

prodwriter = csv.writer(fh)

for i in range(4):

    prod_id = int(input("Enter product id : "))

    prod_name = input("Enter product name : ")

    prod_price = int(input("Enter product price : "))

    prorec = [prod_id, prod_name, prod_price]

    prodwriter.writerow(prorec)

fh.close()

count=0

with open('product.csv','r',newline='\r\n')as f:

    proreader = csv.reader(f)

    for i in proreader:

        if int(i[2]) > 300:

            count += 1

            print(i)

    print('There is/are',count,'products whose price is greater than 300')
```

Output :-



```
========== RESTART: C:/Users/Dell/Desktop/CS,HPE/CS proje
Enter product id : 1001
Enter product name : Sugar
Enter product price : 450
Enter product id : 1002
Enter product name : Milk
Enter product price : 100
Enter product id : 1003
Enter product name : Bread
Enter product price : 40
Enter product id : 1004
Enter product name : Rice
Enter product price : 800
['1001', 'Sugar', '450']
['1004', 'Rice', '800']
There is/are 2 products whose price is greater than 300
>>> 
```

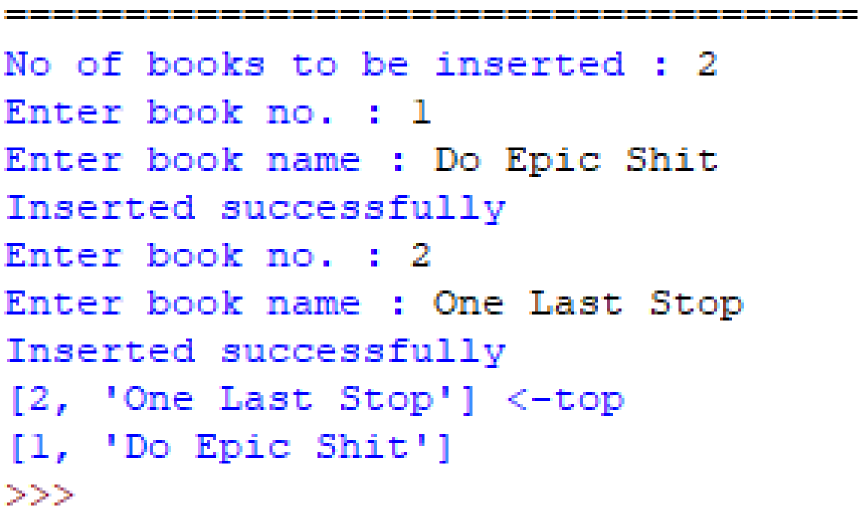
16. Write a program to implement a stack for those book details (book no, book name). That is, now each item node of the stack contains two types of information – a book no and book name. Implement push and display operations.

Program code:

```
stack=[]
def push(stack):
    n = int(input("No of books to be inserted : "))
    for i in range(n):
        bookno = int(input("Enter book no. : "))
        bookname = input("Enter book name : ")
        l = [bookno,bookname]
        stack.append(l)
        print("Inserted successfully")
def display(stack):
    if stack == []:
        print("Stack empty")
    else:
        top = len(stack)-1
        print(stack[top], "<-top")
        for a in range(top-1,-1,-1):
            print(stack[a])
push(stack)
```

display(stack)

<u>Output :-</u>



```
====================================
No of books to be inserted : 2
Enter book no. : 1
Enter book name : Do Epic Shit
Inserted successfully
Enter book no. : 2
Enter book name : One Last Stop
Inserted successfully
[2, 'One Last Stop'] <-top
[1, 'Do Epic Shit']
>>>
```

17. Write a menu driven program to input a list N=[12,20,34,56,55,79,98,22,35,60] and perform the following operations as a separate user defined functions:

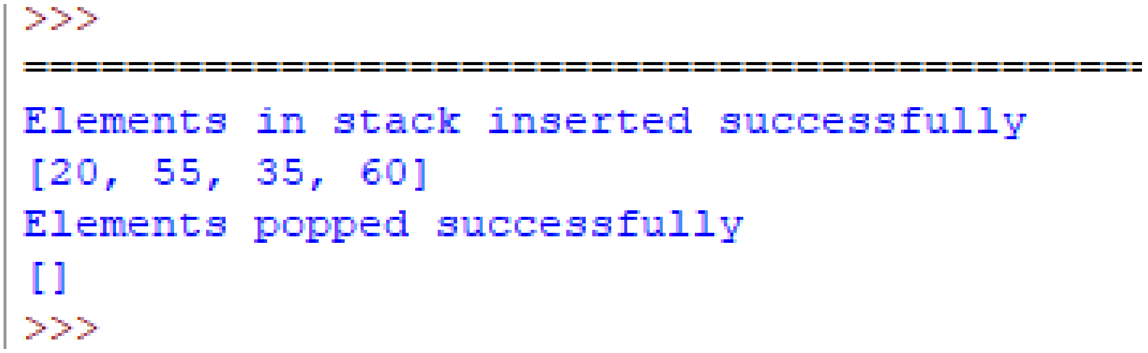
- Push the elements of the list into the stack which are divisible by 5.
- Pop the content of the stack.
- Display the content of the stack after insertion.

Program code:

```
stk=[]
def push(stk):
    N=[12,20,34,56,55,79,98,22,35,60]
    for i in range (len(N)):
        if N[i]%5==0:
            stk.append(N[i])
    print("Elements in stack inserted successfully")


def display(stk):
    if stk == []:
        print("Stack empty")
    else:
        print(stk)


def pop(stk):
    if stk == []:
        print("Underflow")
    else:
```

```
    for i in range(len(stk)):
        l = stk.pop()
    print("Elements popped successfully")
    print(stk)


push(stk)
display(stk)
pop(stk)
```

Output :



```
>>>
===========================================
Elements in stack inserted successfully
[20, 55, 35, 60]
Elements popped successfully
[]
>>>
```

18. Write a menu driven program to input the below given dictionary details and perform the following operations on a stack as a separate user defined functions.

R = {“OM”:76, “JAI”:45, “BOB”:89, “ALI”:65, “ANU”:90, “TOM”:82}

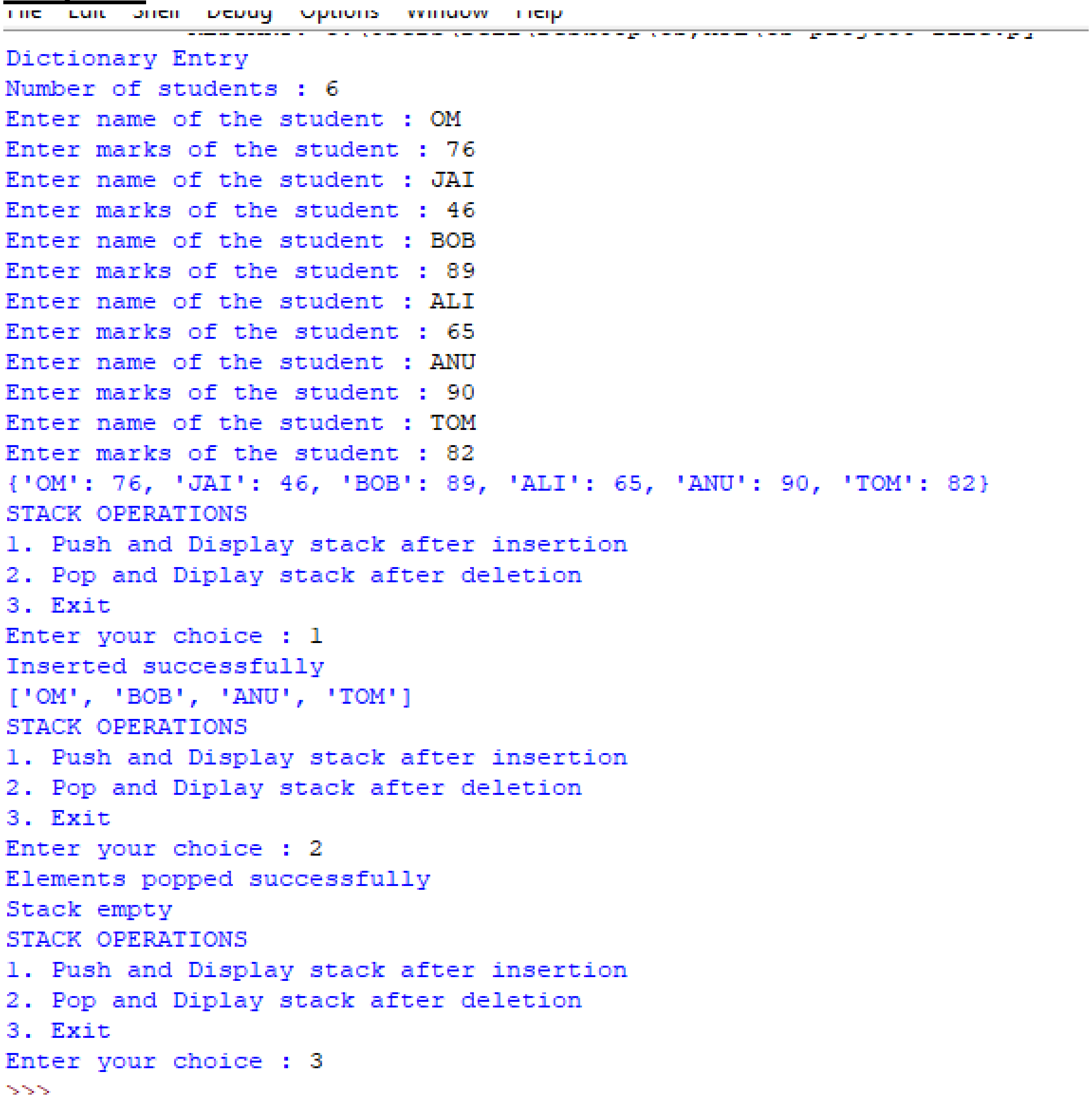
- Push the keys (name of the student) of the dictionary into a stack, where the corresponding value (marks) is greater than 75.
- Pop the content of the stack.
- Display the content of the stack after insertion and deletion.

Program code:

```
d = {}

def dic():

    n = int(input("Number of students : "))

    for i in range(n):

        nm = input('Enter name of the student : ')

        mrks = int(input('Enter marks of the student : '))

        d[nm] = mrks

    print(d)

stk = []

def push(stk,d):

    for i in d:

        if d[i]>75:

            stk.append(i)

            top = len(stk)-1

    print("Inserted successfully")

def pop(stk)
```

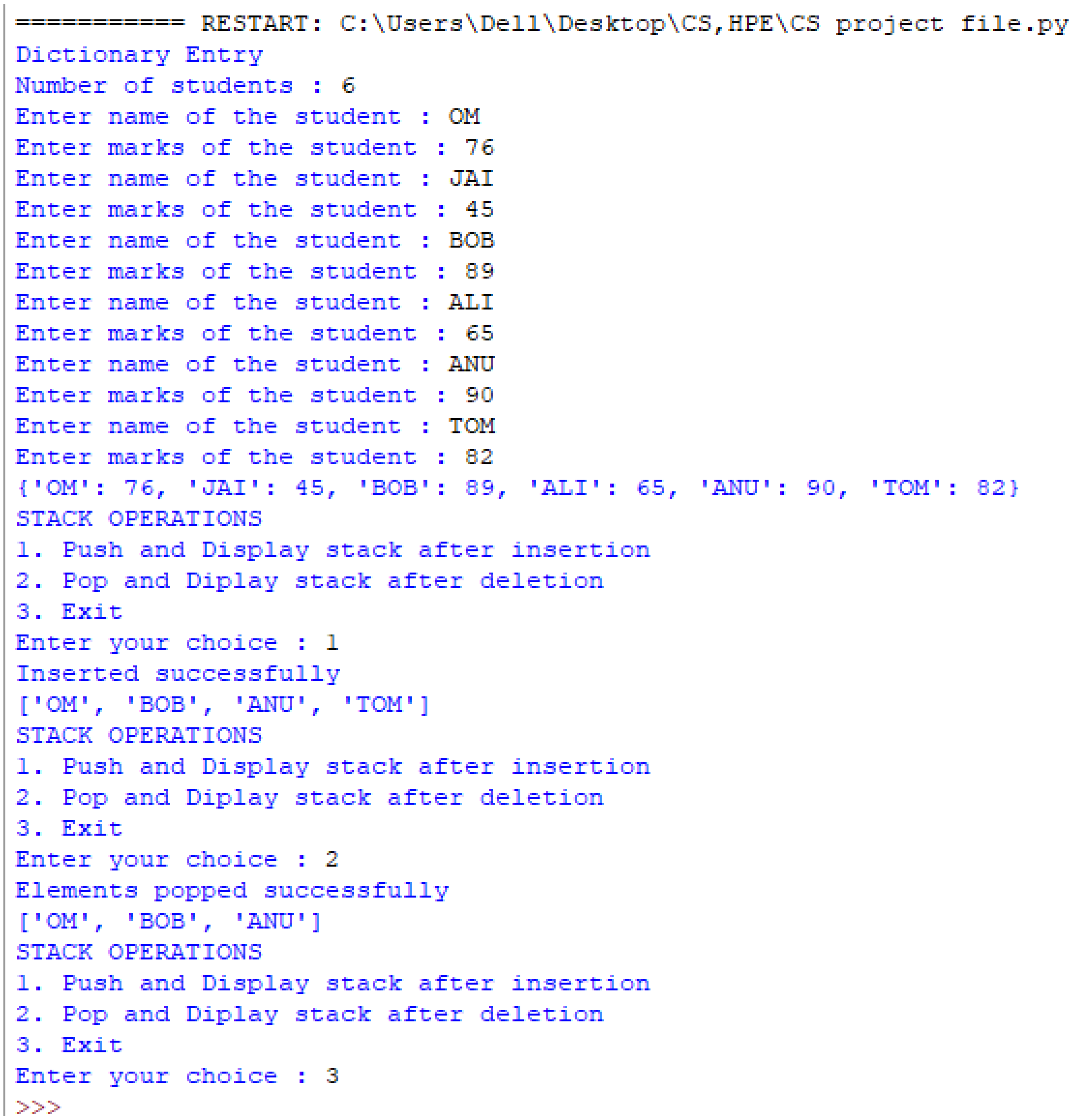
```
    if stk == []:
        print("Stack is empty")
    else:
        #TO POP SINGLE ELEMENT OF  THE STACK
        l = stk.pop()
        print("Elements popped successfully")
         #TO POP ENTIRE CONTENT OF THE STACKK
        '''for i in range(len(stk)):
            l = stk.pop()
        print("Elements popped successfully")'''
def display(stk):
    if stk == []:
        print("Stack empty")
    else:
        print(stk)


print('Dictionary Entry')
dic()
while True:
    print("STACK OPERATIONS")
    print("1. Push and Display stack after insertion")
    print("2. Pop and Diplay stack after deletion")
    print("3. Exit")
    ch=int(input("Enter your choice : "))
    if ch==1:
        push(stk,d)
        display(stk)
    elif ch==2:
        pop(stk)
        display(stk)
    elif ch==3:
        break
    else:
        print("Try again")
```

Output :-



```
Dictionary Entry
Number of students : 6
Enter name of the student : OM
Enter marks of the student : 76
Enter name of the student : JAI
Enter marks of the student : 46
Enter name of the student : BOB
Enter marks of the student : 89
Enter name of the student : ALI
Enter marks of the student : 65
Enter name of the student : ANU
Enter marks of the student : 90
Enter name of the student : TOM
Enter marks of the student : 82
{'OM': 76, 'JAI': 46, 'BOB': 89, 'ALI': 65, 'ANU': 90, 'TOM': 82}
STACK OPERATIONS
1. Push and Display stack after insertion
2. Pop and Diplay stack after deletion
3. Exit
Enter your choice : 1
Inserted successfully
['OM', 'BOB', 'ANU', 'TOM']
STACK OPERATIONS
1. Push and Display stack after insertion
2. Pop and Diplay stack after deletion
3. Exit
Enter your choice : 2
Elements popped successfully
Stack empty
STACK OPERATIONS
1. Push and Display stack after insertion
2. Pop and Diplay stack after deletion
3. Exit
Enter your choice : 3
>>>
```

OUTPUT WHEN ENTIRE STACK IS POPPED.



```
=========== RESTART: C:\Users\Dell\Desktop\CS,HPE\CS project file.py
Dictionary Entry
Number of students : 6
Enter name of the student : OM
Enter marks of the student : 76
Enter name of the student : JAI
Enter marks of the student : 45
Enter name of the student : BOB
Enter marks of the student : 89
Enter name of the student : ALI
Enter marks of the student : 65
Enter name of the student : ANU
Enter marks of the student : 90
Enter name of the student : TOM
Enter marks of the student : 82
{'OM': 76, 'JAI': 45, 'BOB': 89, 'ALI': 65, 'ANU': 90, 'TOM': 82}
STACK OPERATIONS
1. Push and Display stack after insertion
2. Pop and Diplay stack after deletion
3. Exit
Enter your choice : 1
Inserted successfully
['OM', 'BOB', 'ANU', 'TOM']
STACK OPERATIONS
1. Push and Display stack after insertion
2. Pop and Diplay stack after deletion
3. Exit
Enter your choice : 2
Elements popped successfully
['OM', 'BOB', 'ANU']
STACK OPERATIONS
1. Push and Display stack after insertion
2. Pop and Diplay stack after deletion
3. Exit
Enter your choice : 3
>>>
```

OUTPUT WHEN A SINGLE ELEMENT OF STACK IS POPPED.

19. Given the following student relation:

Relation Student

No.	Name	Age	Department	Dateofadm	Fee	Sex
1.	Pankaj	24	Computer	10/01/97	120	M
2.	Shalini	21	History	24/03/98	200	F
3.	Sanjay	22	Hindi	12/12/96	300	M
4.	Sudha	25	History	01/07/99	400	F
5.	Rakesh	22	Hindi	05/09/97	250	M
6.	Shakeel	30	History	27/06/98	300	M
7.	Surya	34	Computer	25/02/97	210	M
8.	Shikha	23	Hindi	31/07/97	200	F

Write SQL commands for (a) to (f) and write output for (g):

(a) To show all information about the students of History department.

➔ SELECT * FROM Student WHERE Department = ‘History’;

No.	Name	Age	Department	Dateofadm	Fee	Sex
2.	Shalini	21	History	24/03/98	200	F
4.	Sudha	25	History	01/07/99	400	F
6.	Shakeel	30	History	27/06/98	300	M

(b) To list the names of female students who are in Hindi department.

```
➔ SELECT Name FROM Student WHERE sex='F' and department='Hindi';
```

Name
Shikha

(c) To list names of all students with their DOA in ascending order.

```
➔ SELECT Name FROM Student ORDER BY Dateofadm;
```

Name
Sanjay
Pankaj
Surya
Shikha
Rakesh
Shalini
Shakeel
Sudha

(d) To display student’s Name, Fee, Age for male Students only.

```
➔ SELECT Name, Fee, Age FROM Student WHERE Sex=’M’;
```

Name	Fee	Age
Pankaj	120	24
Sanjay	300	22

Rakesh	250	22
Shakeel	300	30
Surya	210	34

(e) To count the no. of student with Age<23.

➔ SELECT COUNT (*) FROM Student WHERE Age<23;

Count(*)
3

(f) To insert a new row in the STUDENT table with the following data:

9, “Zaheer”,36, “Computer”, {12/03/97}, 230, “M”

➔ INSERT INTO Student VALUES (9, 'Zaheer',36,'Computer','97-03-12',230,'M');

No.	Name	Age	Department	Dateofadm	Fee	Sex
9	Zaheer	36	Computer	12/03/97	230	M

(g) Give the output of following SQL statements:

(i)Select COUNT (distinct department) from STUDENT;

→

COUNT(distinct department)
3

(ii)Select MAX (Age) from STUDENT where Sex = “F”;

→

MAX(Age)
25

(iii) Select AVG (Fee) from STUDENT where Dateofadm < {01/01/98};
→

AVG(Fee)
218.3333

(iv) Select SUM (Fee) from STUDENT where Dateofadm < {01/01/98};
→

SUM(Fee)
1310

20. Give the following table for database a LIBRARY.

TABLE: BOOKS

Book_Id	Book_Name	Author_name	Publishers	Price	Type	Qty
C0001	Fast Cook	Lata Kapoore	EPB	355	Cookery	8
F0001	The Tears	William Hopkins	First Publ	650	Fiction	10
T0001	My first C+ +	Brains & Brooke	EPB	350	Text	10
T0002	C++ Brain works	A.W.Rossaine	TDH	350	Text	15
F0002	Thunderbolts	Anna Roberts	First Publ.	750	Fiction	5

TABLE: ISSUED

BOOK_ID	QUANTITY_ISSUED
T0001	4
C0001	5
F0001	2

Write SQL queries for (a) to (f):

(a) To show Book name, Author name and Price of books of First Publ. publishers.

➔ Select Book_Name,Author_Name, price from books where Publisher =”First Publ”;

Book_Name	Author_name	Price
The Tears	William Hopkins	650
Thunderbolts	Anna Roberts	750

(b) To list the names from books of Text type.

➔ Select Book_Name from books where type=”Text”;

Book_Name
My first C+ +
C++ Brain works

(c) To display the names and price from books in ascending order of their price.

➔ Select Book_name, price from books order by price;

Book_Name	Price
My first C+ +	350
C++ Brain works	350
Fast Cook	355
The Tears	650
Thunderbolts	750

(d) To increase the price of all books of EPB Publishers by 50.

➔ Update books set price= price+50 where publishers = EPB"
Select * from books;

Book_Id	Book_Name	Author_name	Publishers	Price	Type	Qty
C0001	Fast Cook	Lata Kapoore	EPB	405	Cookery	8
F0001	The Tears	William Hopkins	First Publ	650	Fiction	10
T0001	My first C+ +	Brains & Brooke	EPB	400	Text	10
T0002	C++ Brain works	A.W.Rossaine	TDH	350	Text	15
F0002	Thunderbolts	Anna Roberts	First Publ.	750	Fiction	5

(e) To display the Book_Id, Book_Name and Quantity_Issued for all books which have been issued. (The query will require contents from both the tables.)

➔ Select Book_ID, Book_Name, Quantity_Issued from Books,Issued where Books.BookId=Issued.BookId;

Book_Id	Book_Name	QUANTITY_ISSUED
C0001	Fast Cook	5
F0001	The Tears	2
T0001	My first C+ +	4
T0002	C++ Brain works	
F0002	Thunderbolts	

(f) To insert a new row in the table Issued having the following data: “F0003”,1

➔ Insert into Issued values(“F0003”,1);

(g) Give the output of the following queries:

(i)SELECT COUNT (*) FROM Books;

→

Count(*)
5

(ii)SELECT MAX(Price) FROM Books WHERE Quantity >= 15;

→

Max(Price)
750

(iii)SELECT Book_Name, Author_Name FROM Books WHERE Publishers = “EPB”;

→

Book_Name	Author_name
Fast Cook	Lata Kapoore
My first C+ +	Brains & Brooke

(iv)SELECT COUNT (DISTINCT Publishers) FROM Books WHERE Price >=400;

→

Count(Distinct Publishers)
1

21. Consider the following DEPT and WORKER tables.

Table: DEPT

DCODE	DEPARTMENT	CITY
D01	MEDIA	DELHI
D02	MARKETING	DELHI
D03	INFRASTRUCTURE	MUMBAI
D05	FINANCE	KOLKATA
D04	HUMAN RESOURCE	MUMBAI

Table: WORKER

WNO	NAME	DOJ	DOB	GENDER	DCODE
1001	George K	2013-09-02	1991-09-01	MALE	D01
1002	Ryma Sen	2021-12-11	1990-12-15	FEMALE	D03
1003	Mohitesh	2013-02-03	1987-09-04	MALE	D05
1007	Anil Jha	2014-01-17	1984-10-19	MALE	D04
1004	Manila Sahai	2012-12-09	1986-11-14	FEMALE	D01
1005	R Sahay	2013-11-18	1987-03-31	MALE	D02
1006	Jaya Priya	2014-06-09	1985-06-23	FEMALE	D05

Write SQL queries for (i) to (iv) and outputs for (v) to (viii):

(i)To display Wno, Name, Gender from the table WORKER in descending order of Wno.

→ SELECT WNO, NAME, GENDER FROM WORKER ORDER BY WNO Desc;

WNO	NAME	GENDER
1007	ANIL JHA	MALE
1006	JAYA PRIYA	FEMALE
1005	R SAHAY	MALE
1004	MANILA SAHAI	FEMALE
1003	MOHITESH	MALE
1002	RYMA SEN	FEMALE
1001	GEORGE K	MALE

(ii)To display the Name of all the FEMALE workers from the table WORKER.

→ SELECT NAME FROM WORKER WHERE GENDER='FEMALE';

NAME
RYMA SEN
MANILA SAHAI
JAYA PRIYA

(iii)To display the Wno and Name of those workers from the table WORKER who are born between ‘1987-01-01’ and ‘1991-12-01’.

→ SELECT WNO, NAME FROM WORKER WHERE DOB BETWEEN ‘1987-01-01’ AND ‘1991-12-01’;

WNO	NAME
1001	GEORGE K
1002	RYMA SEN
1003	MOHITESH
1005	R SAHAY

(iv)To count and display MALE workers who have joined after ‘1986-01-01’.

→ SELECT Count (*) FROM WORKER WHERE GENDER=’MALE’ AND DOJ>’1986-01-01’;

Count (*)
3

(v)SELECT COUNT (*), DCODE FROM WORKER GROUP BY DCODE HAVING COUNT(*)>1;

→

COUNT (*)	DCODE
2	D01
2	D05

(vi) SELECT DISTINCT DEPARTMENT FROM DEPT;

→

DISTINCT DEPARTMENT
MEDIA
MARKETING
INFRASTRUCTURE
FINANACE
HUMAN RESOUCE

(vii) SELECT NAME, DEPARTMENT, CITY FROM WORKER W, DEPT D WHERE W.DCODE = D.DCODE AND WNO<1003;

→

NAME	DEPARTMENT	CITY
GEORGE K	MEDIA	DELHI
RYMA SEN	INFRASTRUCTURE	MUMBAI

(viii) SELECT MAX(DOJ), MIN(DOB) FROM WORKER;
→

MAX (DOJ)	MIN (DOB)
2014-06-09	1984-10-19

22. In a database, there are two tables given below:

Table: EMPLOYEE

EMPLOYEEID	NAME	SALES	JOBID
E1	SUMIT SINHA	1100000	102
E2	VIJAY SINGH TOMAR	1300000	101
E3	AJAY RAJPAL	1400000	103
E4	MOHIT RAMNANI	1250000	102
E5	SHAILJA SINGH	1450000	103

Table: JOB

JOBID	JOBTITLE	SALARY
101	President	200000
102	Vice President	125000
103	Administration Assistant	80000
104	Accounting Manager	700000
105	Accountant	65000
106	Sales Manager	80000

Write SQL Queries for the following:

(i)To display employee ids, names of employees, job ids with corresponding job titles.

→ SELECT EMPLOYEEID, NAME, E.JOBID, JOBTITLE FROM EMPLOYEE E, JOB J WHERE E.JOBID = J.JOBID ;

EMPLOYEEID	NAME	JOBID	JOBTITLE
E2	VIJAY TOMAR SINGH	101	PRESIDENT
E1	SUMIT SINHA	102	VICE PRESIDENT
E4	MOHIT RAMNANI	102	VICE PRESIDENT
E3	AJAY RAJPAL	103	ADMINISTRATION ASSISTANT
E5	SHAILJA SINGH	103	ADMINISTRATION ASSISTANT

(ii)To display names of employees, sales and corresponding job titles who have achieved sales more than 1300000.

→ SELECT NAME, SALES, JOBTITLE FROM EMPLOYEE, JOB WHERE EMPLOYEE.JOBID = JOB.JOBID AND SALES > 1300000;

NAME	SALES	JOBTITLE
AJAY RAJPAL	1400000	ADMINISTRATION ASSISTANT
SHAILJA SINGH	1450000	ADMINISTRATION ASSISTANT

(iii)To display names and corresponding job titles of those employees who have ‘SINGH’ (anywhere) in their names.

→ SELECT NAME, JOBTITLE FROM EMPLOYEE, JOB WHERE EMPLOYEE.JOBID = JOB.JOBID AND NAME LIKE “%SINGH%”;

NAME	JOBTITLE
VIJAY SINGH TOMAR	PRESIDENT
SHAILJA SINGH	ADMINISTRATION ASSISTANT

(iv)Identify foreign key in the table EMPLOYEE.

→ JOBID

(v) Write SQL command to change the JOBID to 104 of the EMPLOYEE with ID as E4 in the table ‘EMPLOYEE’.

→ UPDATE EMPLOYEE SET JOBID=104 WHERE EMPLOYEEID = ‘E4’;

EMPLOYEEID	NAME	SALES	JOBID
E1	SUMIT SINHA	1100000	102
E2	VIJAY SINGH TOMAR	1300000	101
E3	AJAY RAJPAL	1400000	103
E4	MOHIT RAMNANI	1250000	104
E5	SHAILJA SINGH	1450000	103

23. Consider the following tables Employee and Salary.

Table: EMPLOYEE

Eid	Name	Depid	Qualification	Sec
1	Deepali Gupta	101	MCA	F
2	Rajat Tyagi	101	BCA	M
3	Hari Mohan	102	B.A.	M
4	Harry	102	M.A.	M
5	Sumit Mittal	103	B.Tech.	M
6	Jyoti	101	M.Tech.	F

Table: SALARY

Eid	Basic	D.A.	HRA	Bonus
1	6000	2000	2300	200
2	2000	300	300	30
3	1000	300	300	40
4	1500	390	490	30
5	8000	900	900	80
6	10000	300	490	89

Write SQL commands for the statements (i) to (iv) and outputs for SQL queries (v) to (vii):

(i)To display the frequency of employees department wise.
→ SELECT DEPID, COUNT (*) FROM EMPLOYEE GROUP BY DEPID;

DEPID	COUNT(*)
101	3
102	2
101	1

(ii) To list the names of those employees only whose name starts with 'H'.
→ SELECT NAME FROM EMPLOYEE WHERE NAME LIKE 'H%';

Name
Hari Mohan
Harry

(iii) To add a new column in salary table. The column name is Total_Sal.
→ ALTER TABLE SALARY ADD COLUMN TOTAL_SAL INT;

Eid	Basic	D.A.	HRA	Bonus	Total_Sal
1	6000	2000	2300	200	NULL
2	2000	300	300	30	NULL
3	1000	300	300	40	NULL
4	1500	390	490	30	NULL
5	8000	900	900	80	NULL
6	10000	300	490	89	NULL

(iv) To store the corresponding values in the Total_Sal column.
→ UPDATE SALARY SET TOTAL_SAL = BASIC+DA+HRA+BONUS;

Eid	Basic	D.A.	HRA	Bonus	Total_Sal
1	6000	2000	2300	200	10500
2	2000	300	300	30	2630
3	1000	300	300	40	1640
4	1500	390	490	30	2410
5	8000	900	900	80	9880
6	10000	300	490	89	10879

(v) Select max(Basic) from Salary where Bonus>40;

→

MAX (BASIC)
10000

(vi) Select count(*) from Employee group by Sex;

→

COUNT (*)
2
4

(vii) Select Distinct Depid from Employee;

→

DISTINCT DEPID
101
102
103

24. Write a program to connect python with mysql database connectivity and perform the following operations on data in database:

(i) Create the table

(ii) Insert the data

(iii) Fetch the data

(iv) Update the data

Table name: Student

Field

Adm No – int Primary key

Student name – Varchar (30)

Gender – Char

Date of Birth – Date

Stream – Varchar (15)

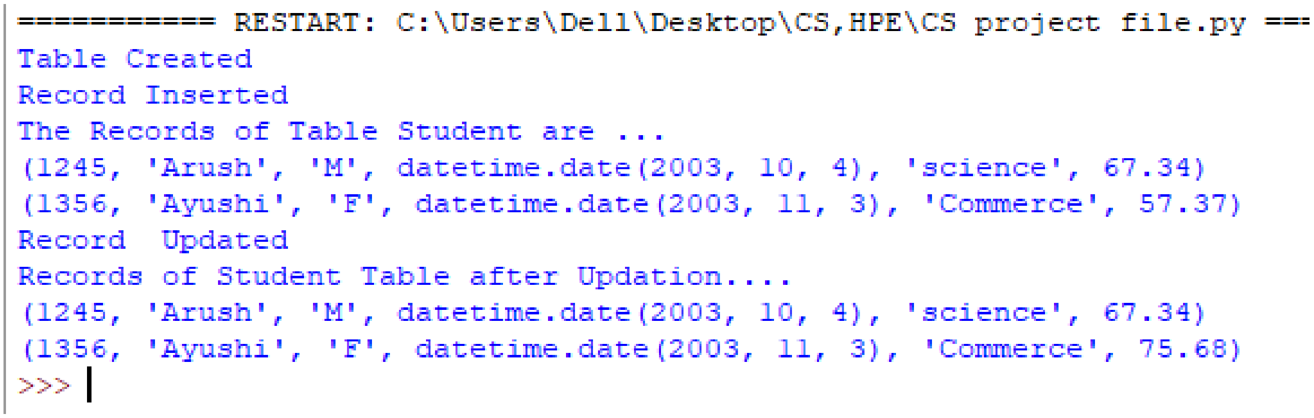
Marks – Float (5,2)

Program code:

```
#CREATE TABLE

import mysql.connector

demodb = mysql.connector.connect(host="localhost", user="root",passwd="root",
database="EDUCATION")

democursor=demodb.cursor()

democursor.execute('''CREATE TABLE STUDENT (admn_no int primary
key,sname varchar(30), gender char(1),
DOB date, stream varchar(15), marks float(4,2))"')

democursor.execute("commit;")

demodb.close()

print("Table Created")



#INSERT THE DATA

import mysql.connector

demodb = mysql.connector.connect(host="localhost", user="root",passwd="root",
database="EDUCATION")

democursor=demodb.cursor()

democursor.execute("insert into student values (%s, %s, %s, %s, %s, %s)",

(1245, 'Arush', 'M', '2003-10-04', 'science', 67.34))

democursor.execute("insert into student values (%s, %s, %s, %s, %s, %s)",

(1356, 'Ayushi', 'F', '2003-11-03', 'Commerce', 57.37))

demodb.commit()

print("Record Inserted")

demodb.close()



#FETCH THE DATA

import mysql.connector

demodb = mysql.connector.connect(host="localhost", user="root",passwd="root",
database="EDUCATION")
```

```
democursor=demodb.cursor( )
democursor.execute("select * from student")
print("The Records of Table Student are ...")
for i in democursor:
    print(i)
demodb.close()


#UPDATE THE DATA
import mysql.connector
demodb = mysql.connector.connect(host="localhost", user="root",passwd="root", database="EDUCATION")
democursor=demodb.cursor( )
democursor.execute("update student set marks=75.68 where admn_no=1356")
print("Record  Updated")


print("Records of Student Table after Updation....")
democursor.execute("select * from student")
for i in democursor:
    print(i)
demodb.commit( )
demodb.close()
```

Output:



```
============ RESTART: C:\Users\Dell\Desktop\CS,HPE\CS project file.py ===
Table Created
Record Inserted
The Records of Table Student are ...
(1245, 'Arush', 'M', datetime.date(2003, 10, 4), 'science', 67.34)
(1356, 'Ayushi', 'F', datetime.date(2003, 11, 3), 'Commerce', 57.37)
Record  Updated
Records of Student Table after Updation....
(1245, 'Arush', 'M', datetime.date(2003, 10, 4), 'science', 67.34)
(1356, 'Ayushi', 'F', datetime.date(2003, 11, 3), 'Commerce', 75.68)
>>> 
```

25. Write a program to connect python with mysql database connectivity and perform the following operations on data in database:

(i) Create the table

(ii) Insert the data

(iii) Fetch the data

(iv) Update the data

Table name: Employee

Field

Emp No – int Primary Key

Emp Name – Varchar (30)

Designation – Varchar (15)

Date of Joining – Date

Salary – Float (10,2)

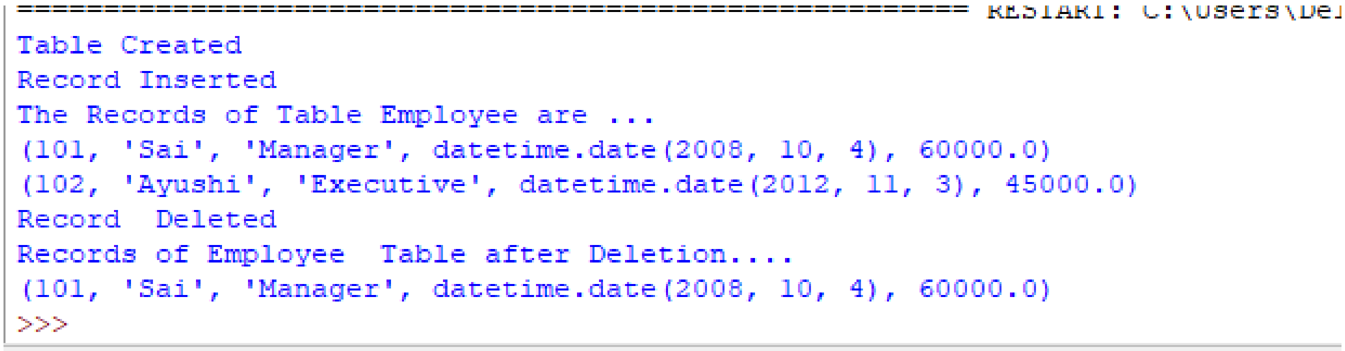
Program code:

```
# CREATE TABLE

import mysql.connector

demodb = mysql.connector.connect(host="localhost", user="root",

passwd="root", database="EDUCATION")

democursor=demodb.cursor( )

democursor.execute("CREATE TABLE Employee1 (emp_no int primary key,empname varchar(30), desig varchar(15), DOJ date, salary float(10,2));")

democursor.execute("commit;")
```

```
demodb.close()
print("Table Created")

#INSERT THE DATA
import mysql.connector
demodb = mysql.connector.connect(host="localhost", user="root",
passwd="root", database="EDUCATION")
democursor=demodb.cursor( )
democursor.execute("insert into employee values (%s, %s, %s, %s, %s )",
(101, 'Sai', 'Manager', '2008-10-04', 60000))
democursor.execute("insert into employee values (%s, %s, %s, %s, %s)",
(102, 'Ayushi', 'Executive', '2012-11-03', 45000.00))
demodb.commit( )
print("Record Inserted")
demodb.close()

#FETCH THE DATA
import mysql.connector
demodb = mysql.connector.connect(host="localhost", user="root",
passwd="root", database="EDUCATION")
democursor=demodb.cursor( )
democursor.execute("select * from employee")
print("The Records of Table Employee are ...")
```

```
for i in democursor:
 print(i)
demodb.close()

#DELETE THE DATA
import mysql.connector
demodb = mysql.connector.connect(host="localhost", user="root",
passwd="root", database="EDUCATION")
democursor=demodb.cursor( )
democursor.execute("Delete from Employee where emp_no=102")
print("Record  Deleted")
print("Records of Employee  Table after Deletion....")
democursor.execute("select * from Employee")
for i in democursor:
 print(i)
demodb.commit( )
demodb.close()
```

Output:



```
======================================================= RESTART: C:\Users\De
Table Created
Record Inserted
The Records of Table Employee are ...
(101, 'Sai', 'Manager', datetime.date(2008, 10, 4), 60000.0)
(102, 'Ayushi', 'Executive', datetime.date(2012, 11, 3), 45000.0)
Record  Deleted
Records of Employee  Table after Deletion....
(101, 'Sai', 'Manager', datetime.date(2008, 10, 4), 60000.0)
>>>
```