PRACTICAL FILE

Class XII - Computer Science with Python(083)

Suggested Practical list

Term-1

Python Programming

● Read a text file line by line and display each word separated by#.
● Read a text file and display the number of vowels /consonants /uppercase/ lowercase characters in the file.
● Remove all the lines that contain the character 'a' in a file and write it to another file.
● Create a binary file with name and roll number. Search for a given roll number and display the name, if not found display appropriate message.
● Create a binary file with roll number, name and marks. Input a roll number and update the marks.
● Write a random number generator that generates random numbers between 1 and 6 (simulates a dice).
● Create a CSV file by entering user-id and password, read and search the password for given userid.

Term-2

Python Programming

● Write a Python program to implement a stack using list.

Database Management

- Create a student table and insert data. Implement the following SQL commands on the student table:
- ALTER table to add new attributes / modify data type / drop attribute
- UPDATE table to modify data
- ORDER By to display data in ascending / descending order
- DELETE to remove tuple(s)
- GROUP BY and find the min, max, sum, count and average Joining of two tables.
- Similar exercise may be framed for other cases.
- Integrate SQL with Python by importing suitable module.

Program 1: **Write a program to read and display file content line by line with each word separated by #.**

```
#TO READ AND DISPLAY FILE CONTENT LINE BY LINE WITH EACH WORD SEPARATED BY #
a= open (r"C:\Users\Jammy\Documents\Abhishek\Python\TEXT.txt","r")
lines=a.readlines()
for line in lines:
    x=line.split()
    for y in x:
        print(y+" # " , end=" ")
    print(" ")
```

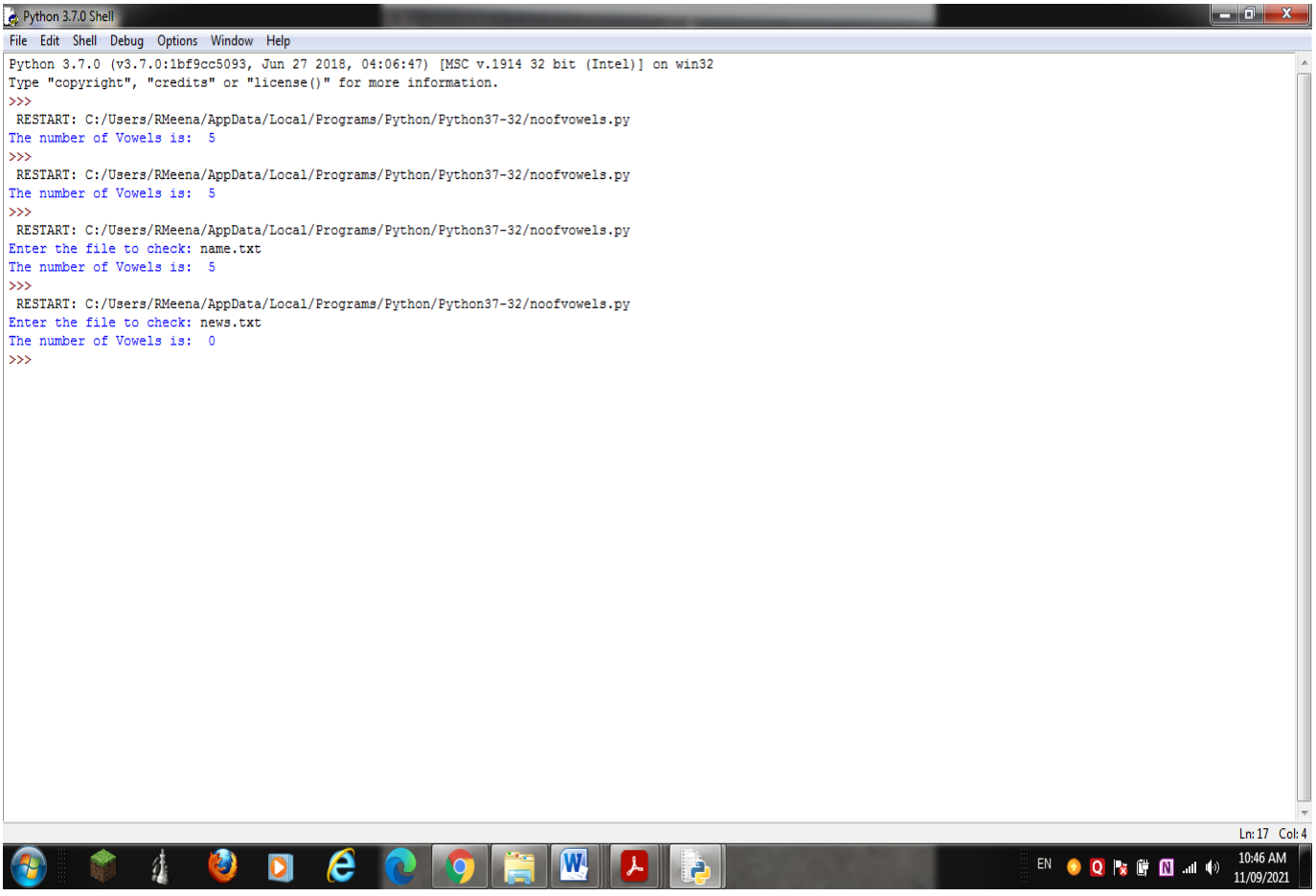
OUTPUT

```
Hello #  World #
Welcome #  to #  python #
Enjoy #  programing #
```

Program: 2 Read a text file and display the number of vowels in the file.

```
#TO READ TEXT FILE AND DISPLAY NO. OF VOWELS
fileName = input("Enter the file to check: ").strip()

infile = open(fileName,"r")

vowels =["a","e","i","o","u"]

text = infile.readline()

countV = 0

for v in text:

    if v in vowels:

        countV += 1
print("The number of Vowels is: ",countV)
```

OUT PUT:



Program: 3 Read a text file and display the number of consonants in the file.

```
#TO READ TEXT FILE AND DISPLAY NO. OF CONSONANTS
fileName = input("Enter the file to check: ").strip()

infile = open(fileName,"r")

cons =set("BCDFGHJKLMNPQRSTVWXYZbcdfghjklmnpqrstvwxyz")

text = infile.readline()

countC = 0

for c in text:

   if c in cons:

       countC += 1
print("The number of Consonant is: ",countC)
```

OUTPUT

```
Enter the file to check: name.txt
The number of Consonent is:  7
>>>
```

Program 4: Read a text file and display the number of lowercase letters in the file.

```
File  Edit  Format  Run  Options  Window  Help
#TO READ TEXT FILE AND DISPLAY NO. OF LOWERCASE LETTERS.
fileName = input("Enter the file to check: ").strip()

infile = open(fileName,"r")

text = infile.readline()

countL = 0

for l in text:

   if l.islower():

       countL += 1
print("The number of Lowercase letters is: ",countL)
```

OUTPUT

```
>>>
 RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/noofconsonents.py
Enter the file to check: name.txt
The number of Consonent is:  7
>>>
 RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/nooflower.py
Enter the file to check: name.txt
The number of Lowercase letters is:  10
>>>
```

Program 5: Read a text file and display the number of uppercase letters in the file.

File Edit Format Run Options Window Help

```
#TO READ TEXT FILE AND DISPLAY NO. OF UPPERCASE LETTERS.
fileName = input("Enter the file to check: ").strip()

infile = open(fileName,"r")

text = infile.readline()

countU = 0

for u in text:

   if u.isupper():

       countU += 1
print("The number of Uppercase letters is: ",countU)
```

OUTPUT

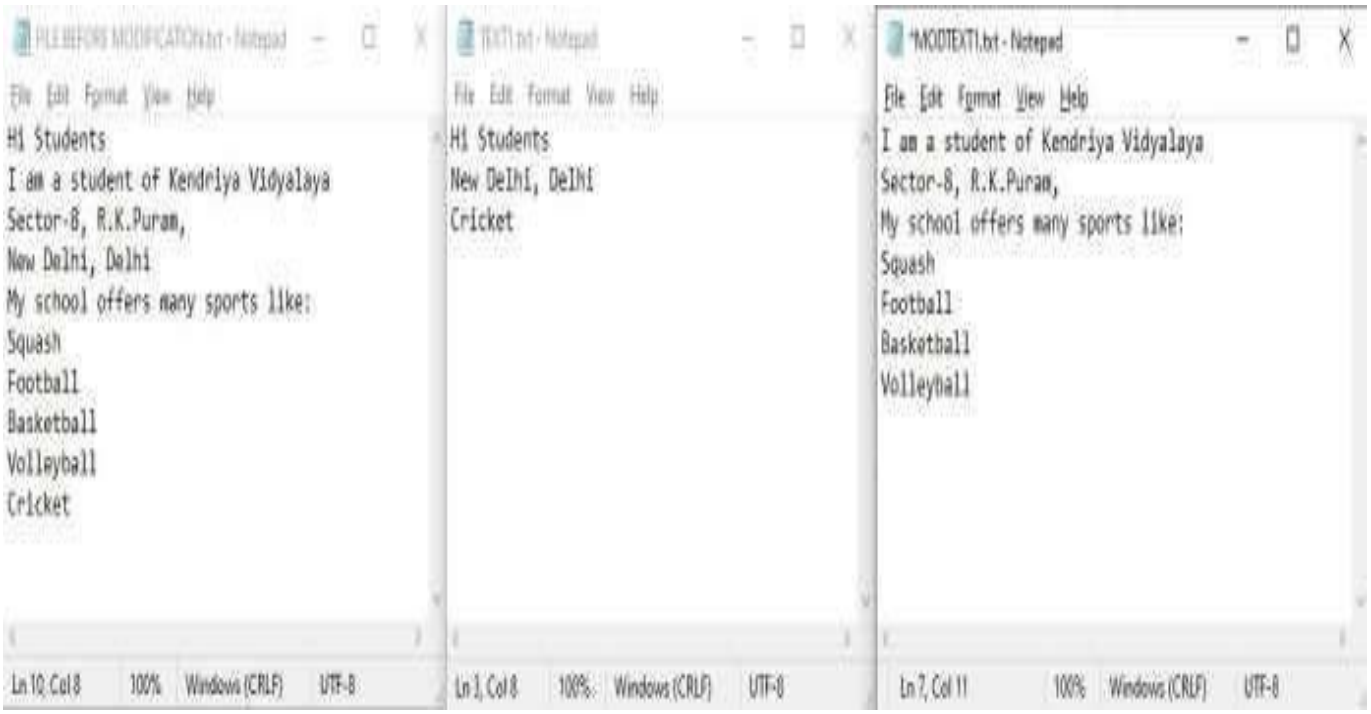
```
RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/nooflower.py
nter the file to check: name.txt
he number of Lowercase letters is:  10
>>
RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/noofupper.py
nter the file to check: name.txt
he number of Uppercase letters is:  2
>>
```

PROGRAM -6: Remove all the lines that contain the character 'a' in a file and write it to another file.

```
#TO REMOVE ALL THE LINES THAT CONTAIN THE CHARACTER 'A' IN A FILE & WRITE IT TO ANOTHER FILE
file=open(r"C:\Users\Jammy\Documents\Abhishek\Python\TEXT1.txt","r")
lines=file.readlines()
file.close()
file=open(r"C:\Users\Jammy\Documents\Abhishek\Python\TEXT1.txt","w")
file1=open(r"C:\Users\Jammy\Documents\Abhishek\Python\MODTEXT1.txt","w")
for line in lines:
    if 'a' in line :
        file1.write(line)
    else:
        file.write(line)
print("Lines that contain a character are removed from TEXT1")
print("Lines that contain a character are added in MODTEXT1 ")
file.close()
file1.close()
```

OUTPUT

```
Lines that contain a character are removed from TEXT1
Lines that contain a character are added in MODTEXT1
```



Program 7: Write a Program to read data from data file in append mode and use write Lines function utility in python.

writelines.py - C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/writelines.py (3.7.0)

File Edit Format Run Options Window Help

```
#Program to read data from data file in append mode
af=open("name.txt",'a')
lines_of_text = ("One line of text here",\
"and another line here",\
"and yet another here", "and so on and so forth")

af.writelines('\n' + lines_of_text)
af.close()
```

OUTPUT:

name - Notepad

File Edit Format View Help

```
Randhir Meena
One line of text here","and another line here","and yet another here",
"and so on and so forth
```

PROGRAM 8: Create a binary file with name and roll number. Search for a given roll number and display the name, if not found display appropriate message.

```
#CREATE A BINARY FILE WITH NAME & ROLL NUMBER. SEARCH FOR A GIVEN ROLL NUMBER AND DISPLAY NAME, IF NOT FOUND DISPLAY
#APPROPRIATE MESSAGE
import pickle
sdata= {}
slist= []
totals=int(input("Enter number of students :"))
for i in range(totals):
    sdata["Roll no."]=int(input("Enter roll no. :"))
    sdata["Name"]=input("Enter name :")
    slist.append(sdata)
    sdata={}
a=open(r"C:\Users\Jammy\Documents\Abhishek\Python\TEXT2.dat", "wb")
pickle.dump(slist,a)
a.close()

x=open(r"C:\Users\Jammy\Documents\Abhishek\Python\TEXT2.dat", "rb")
slist=pickle.load(x)
b=int(input("Enter the roll number of student to be searched :"))
y=False
for TEXT2 in slist:
    if (TEXT2["Roll no."]==b):
        y=True
        print(TEXT2["Name"],"Found in file")
if (y==False):
    print("Data of student not found.")
```

OUTPUT

```
Enter number of students :3
Enter roll no. :25
Enter name :Anshul
Enter roll no. :41
Enter name :Hari
Enter roll no. :39
Enter name :Abhishek
Enter the roll number of student to be searched :39
Abhishek Found in file
```

PROGRAM 9: Create a binary file with roll number, name and marks. Input a roll number and update the marks.

```
#TO CREATE A BINARY FILE WITH ROLL NUMBER, NAME & MARKS, INPUT A ROLL NUMBER & UPDATE THE MARKS.
import pickle

sdata={}
totals=int(input("Enter number of students :"))
a=open(r"C:\Users\Jammy\Documents\Abhishek\Python\TEXT3.dat", "wb")
for i in range(totals):
    sdata["Roll no"]=int(input("Enter roll no. :"))
    sdata["Name"]=input("Enter name :")
    sdata["Marks"]=float(input("Enter marks :"))
    pickle.dump(sdata,a)
    sdata={}
a.close()

#UPDATING MARKS

found=False
rollno=int(input("Enter roll number :"))
a=open(r"C:\Users\Jammy\Documents\Abhishek\Python\TEXT3.dat", "rb+")
try:
    while True:
        pos=a.tell()
        sdata=pickle.load(a)
        if (sdata["Roll no"]==rollno):
            sdata["Marks"]=float(input("Enter new marks :"))

            a.seek(pos)
            pickle.dump(sdata,a)
            found=True
except EOFError:
    if found==False:
        print("Roll number not found.")
    else:
        print("Students marks updated successfully.")
    a.close()
```

OUTPUT

```
Enter number of students :3
Enter roll no. :25
Enter name :Anshul
Enter marks :94
Enter roll no. :41
Enter name :Hari
Enter marks :95
Enter roll no. :39
Enter name :Abhishek
Enter marks :93
Enter roll number :39
Enter new marks :96
Students marks updated successfully.
```

PROGRAM 10: Write a random number generator that generates random numbers between 1 and 6 using user defined function (simulates a dice).

```
#TO GENERATE RANDOM NUMBERS BETWEEN 1 AND 6 USING USER DEFINED FUNCTION
def fun():
    import random
    r=random.randint(1,6)
    print("Random number generated between 1 to 6 :",r)
fun()
```

OUTPUT

```
Random number generated between 1 to 6 : 6
>>>
```

PROGRAM-11: Write a program to create a CSV file with empid, name and mobile no. and search empid, update the record and display the records.

```
#Input Data

import csv
with open(r'C:\Users\Jammy\Documents\Abhishek\Python\employee.txt','a') as csvfile:
     mywriter = csv.writer(csvfile,delimiter=',')
     ans='y'
     while ans.lower()=='y':
          eno=int(input("Enter Employee ID :"))
          name=input("Enter Employee Name :")
          mobno=int(input("Enter Employee Mobile No :"))
          mywriter.writerow([eno,name,mobno])
          ans=input("Do You Want To Enter More Data? (y/n):")
ans='y'

#Search Empid And Display The Records

with open(r'C:\Users\Jammy\Documents\Abhishek\Python\employee.txt','r') as csvfile:
    myreader = csv.reader(csvfile,delimiter=',')
    ans='y'
    while ans.lower()=='y':
         found=False
         e = int(input("Enter Employee ID to search :"))
         for row in myreader:
              if len(row)!=0:
                   if int(row[0])==e:
                        print("Name :",row[1])
                        print("Mobile no. :",row[2])
                        found=True
                        break
                   if not found:
                        print("Employee ID not found.")
         ans = input("Do You Want To Search More? (y/n):")
ans='y'

with open(r'C:\Users\Jammy\Documents\Abhishek\Python\employee.txt','a') as csvfile:
    x=csv.writer(csvfile)
    x.writerow([eno,name,mobno ])
```

OUTPUT:

```
Enter Employee ID :1
Enter Employee Name :Anshul
Enter Employee Mobile No :919864124
Do You Want To Enter More Data? (y/n):y
Enter Employee ID :2
Enter Employee Name :Hari
Enter Employee Mobile No :941827498
Do You Want To Enter More Data? (y/n):n
Enter Employee ID to search :1
Name : Anshul
Mobile no. : 919864124
Do You Want To Search More? (y/n):y
Enter Employee ID to search :2
Name : Hari
```

PROGRAM-12: Program to enter two numbers and print the arithmetic operations like +,-,*,/, // and %.

```
#Program for Arithmetic Calculator
result = 0
val1 = float(input("Enter the first value :"))
val2 = float(input("Enter the second value :"))
op = input("Enter any one of the operator (+,-,*,/,//,%)")
if op == "+":
    result = val1 +val2
elif op == "-":
 result = val1 - val2
elif op == "*":
 result = val1 * val2
elif op == "/":
 if val2 == 0:
     print("Please enter a value other than 0")
 else:
     result = val1 / val2
elif op == "//":
    result = val1 // val2
else:
    result = val1 % val2
print("The result is :",result)
```

OUTPUT:

```
Enter the first value :10
Enter the second value :5
Enter any one of the operator (+,-,*,/,//,%)+
The result is : 15.0
>>>
 RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/calculator.py
Enter the first value :10
Enter the second value :5
Enter any one of the operator (+,-,*,/,//,%)-
The result is : 5.0
>>>
 RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/calculator.py
Enter the first value :10
Enter the second value :5
Enter any one of the operator (+,-,*,/,//,%)*
The result is : 50.0
>>>
 RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/calculator.py
Enter the first value :10
Enter the second value :5
Enter any one of the operator (+,-,*,/,//,%)%
The result is : 0.0
>>>
```

PROGRAM-13: Write a Program to check if the entered number is Armstrong or not.

*armastrong.py - C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/armastrong.py (3...

File Edit Format Run Options Window Help

```
# Program to check if the entered number is Armstrong or not.

#An Armstrong number has sum of the cubes of its digits is equal to the numbe
no=int(input("Enter any number to check : "))
no1 = no
sum = 0
while(no>0):
    ans = no % 10;
    sum = sum + (ans * ans * ans)
    no = int (no / 10)
if sum == no1:
    print("Armstrong Number")
else:
    print("Not an Armstrong Number")
```

OUTPUT:

```
 RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/calculator.py
Enter the first value :10
Enter the second value :5
Enter any one of the operator (+,-,*,/,//,%)*
The result is : 50.0
>>>
 RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/calculator.py
Enter the first value :10
Enter the second value :5
Enter any one of the operator (+,-,*,/,//,%)%
The result is : 0.0
>>>
 RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/armastrong.py
Enter any number to check : 121
Not an Armstrong Number
>>>
 RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/armastrong.py
Enter any number to check : 371
Armstrong Number
>>>
```

Program 14: Write a Program to find factorial of the entered number.

```
File  Edit  Format  Run  Options  Window  Help
#Program to calculate the factorial of an inputted number (using while loop)
num = int(input("Enter the number for calculating its factorial : "))
fact = 1
i = 1

while i<=num:
    fact = fact*i
    i = i + 1
print("The factorial of ",num,"=",fact)
```

OUTPUT:

```
 RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/factorial1.py
Enter the number for calculating its factorial : 6
The factorial of  6 = 720
>>>
 RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/factorial1.py
Enter the number for calculating its factorial : 7
The factorial of  7 = 5040
>>>
```

Program 15: Write a Program to enter the number of terms and to print the Fibonacci Series.

```
#fibonacci

i =int(input("enter the limit:"))
x = 0

y = 1

z = 1

print("Fibonacci series \n")
print(x, y,end= " ")
while(z<= i):
    print(z, end=" ")
    x = y
    y = z
    z = x + y
```

OUTPUT:

```
 RESTART: C:/Users/RMeena/AppData/Local/Programs/Python/Python37-32/fibonacci.py
enter the limit:5
Fibonacci series

0 1 1 2 3 5
>>>
```

PROGRAM 16: Write a program to create Lpush() and Lpop() function to do push and pop operation on a stack using a list e.g. take a student information and push and pop the details.

```
#Write a programme to create Lpush( ) and Lpop( ) function to do push and pop operation
#on a stack using a list e.g. take a student information and push and pop the details
def isEmpty(S):
    if len(S)==0:
        return True
    else:
        return False
def Push(S,item):
    S.append(item)
    top=len(S)-1
def Pop(S):
    if isEmpty(S):
        return "UNDERFLOW"
    else:
        val=S.pop()
        if len(S)==0:
            top=None
        else:
            top=len(S)-1
        return val
#main function begins here
S=[]
top=None
while True:
    print("***STACK DEMONSTRATION***")
    print("1. PUSH")
    print("2. POP")
    print("0. EXIT")
    ch=int(input("Enter your choice :"))
    if ch==1:
        val=int(input("Enter the item of Push :"))
        Push(S,val)
    elif ch==2:
        val=Pop(S)
        if val=="UNDERFLOW":
            print("STACK IS EMPTY")
        else:
            print("Deleted Item was:", val)
    elif ch==0:
        print("THANK YOU")
        break
```

OUTPUT:

```
***STACK DEMONSTRATION***
1. PUSH
2. POP
0. EXIT
Enter your choice :1
Enter the item of Push :10
***STACK DEMONSTRATION***
1. PUSH
2. POP
0. EXIT
Enter your choice :2
Deleted Item was: 10
***STACK DEMONSTRATION***
1. PUSH
2. POP
0. EXIT
Enter your choice :0
THANK YOU
```

PROGRAM 17: Write a program to create a student table and insert data. Implement the following SQL commands on the student table:

ALTER table to add new attributes / modify data type / drop attribute

UPDATE table to modify data ORDER By to display data in ascending / descending order

DELETE to remove tuple(s) GROUP BY and find the min, max, sum, count and average

use studentdb;

Database changed

CREATING TABLE

create table student (roll int not null, studentname varchar(30) not null, class

char(5) not null, section char(1) not null, classstream char(20) not null);

desc student;

Field	Type	Null	Key	Default	Extra
roll	int	NO		NULL	
studentname	varchar(30)	NO		NULL	
class	char(5)	NO		NULL	
section	char(1)	NO		NULL	
classstream	char(20)	NO		NULL	

INSERTING VALUE

insert into student values

```
(1, "Akhil", "XII", "A", "Science"),
(2, "Satya", "XII", "C", "Science"),
(3, "Antony", "XII", "D", "Commerce"),
(4, "Vishal", "XII", "E", "Humanities"),
(5, "Deepak", "XII", "A", "Science")
(6, "Brij", "XII", "B", "Science");
select * from student;
```

roll	studentname	class	section	classstream
1	Akhil	XII	A	Science
2	Satya	XII	C	Science
3	Antony	XII	D	Commerce
4	Vishal	XII	E	Humanities
5	Deepak	XII	A	Science
6	Brij	XII	B	Science

ADDING COLUMN IN TABLE USING ALTER COMMAND

```
alter table student
add(Substream char(20) not null), (Percentage float not
                    null);
```

desc student;

Field	Type	Null	Key	Default	Extra
roll	int	NO		NULL	
studentname	varchar(30)	NO		NULL	
class	char(5)	NO		NULL	
section	char(1)	NO		NULL	
classstream	char(20)	NO		NULL	
Substream	char(20)	NO		NULL	
Percentage	float	NO		NULL	

UPDATING VALUE

```
update student set Substream = "Computer Science" where roll = 1;

update student set Substream = "Biology" where roll = 2;
update student set Substream = "Maths" where roll = 3;
update student set Substream = "NA" where roll = 4;
update student set Substream = "Computer Science" where roll = 5;
update student set Substream = "Computer Science" where roll = 6;
```

roll	studentname	class	section	classstream	Substream	Percentage
1	Akhil	XII	A	Science	Computer Science	0
2	Satya	XII	C	Science	Biology	0
3	Antony	XII	D	Commerce	Maths	0
4	Vishal	XII	E	Humanities	NA	0
5	Deepak	XII	A	Science	Computer Science	0
6	Brij	XII	B	Science	Computer Science	0

```
update student set Percentage = 81 where roll = 6;
update student set Percentage = 69 where roll = 6;
update student set Percentage = 92 where roll = 6;
update student set Percentage = 55 where roll = 6;
update student set Percentage = 85 where roll = 6;
update student set Percentage = 72 where roll = 6; select * from student;
```

roll	studentname	class	section	classstream	Substream	Percentage
1	Akhil	XII	A	Science	Computer Science	81

roll	studentname	class	section	classstream	Substream	Percentage
1	Akhil	XII	A	Science	Computer Science	81
2	Satya	XII	C	Science	Biology	69
3	Antony	XII	D	Commerce	Maths	92
4	Vishal	XII	E	Humanities	NA	55
5	Deepak	XII	A	Science	Computer Science	85
6	Brij	XII	B	Science	Computer Science	72

alter table student add(Adhaar_card bigint not null);

desc student;

Field	Type	Null	Key	Default	Extra
roll	int	NO		NULL	

Field	Type	Null	Key	Default	Extra
roll	int	NO		NULL	
studentname	varchar(30)	NO		NULL	
class	char(5)	NO		NULL	
section	char(1)	NO		NULL	
classstream	char(20)	NO		NULL	
Substream	char(20)	NO		NULL	
Percentage	float	NO		NULL	
Aadhaar_card	bigint	NO		NULL	

alter table student
drop column Adhaar_card;

desc student;

Field	Type	Null	Key	Default	Extra
roll	int	NO		NULL	

Field	Type	Null	Key	Default	Extra
roll	int	NO		NULL	
studentname	varchar(30)	NO		NULL	
class	char(5)	NO		NULL	
section	char(1)	NO		NULL	
classstream	char(20)	NO		NULL	
Substream	char(20)	NO		NULL	
Percentage	float	NO		NULL	

AVERAGE COMMAND

select avg(Percentage) from student;

avg(Percentage)
75.66666666666667

COUNT COMMAND

select count(*) from student;

count(*)
6

MIN, MAX AND SUM COMMAND

select min(Percentage) from student;

min(Percentage)
55

select max(Percentage) from student;

max(Percentage)
92

select sum(Percentage)/count(*) from student where substream = "Computer Science";

sum(Percentage)/count(*)
79.33333333333333

DELETE A TUPLE FROM TABLE

delete from student where

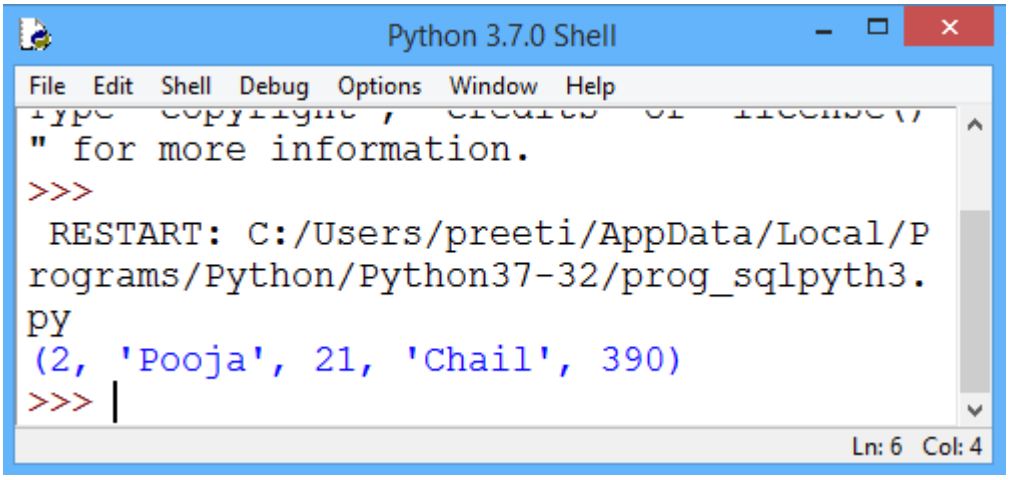
classstream = "Science"; select * from

student;

roll	studentname	class	section	classstream	Substream	Percentage
3	Antony	XII	D	Commerce	Maths	92
4	Vishal	XII	E	Humanities	NA	55

Program 18: Write a Program to show MySQL database connectivity in python.

```
import mysql.connector

con=mysql.connector.connect(host='localhost',user='root',password='', db='school')

        stmt=con.cursor()

        query='select * from student;'

        stmt.execute(query)

        data=stmt.fetchone()

        print(data)
```



Program 19: Perform all the operations with reference to table 'Employee' through MySQL-Python connectivity.

```
import MySQLdb

# Using connect method to connect database

db1 =MySQLdb.connect("localhost","root","","TESTDB" )

# using cursor() method for preparing cursor

cursor = db1.cursor()

# Preparing SQL statement to create EMP table

sql = "CREATE TABLE EMP(empno integer primary key,ename varchar(25) not null,salary
float);"

cursor.execute(sql)

# disconnect from server

db1.close()
```

```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.1.73-community MySQL Community Server (GPL)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use testdb
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql>
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use testdb
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> show tables;
+------------------+
| Tables_in_testdb |
+------------------+
| emp              |
+------------------+
1 row in set (0.00 sec)

mysql>
```

Inserting a record in ‘emp’

```
import MySQLdb

db1 = MySQLdb.connect("localhost","root","","TESTDB" )

cursor = db1.cursor()

# Prepareing SQL statement to insert one record with the

given values sql = "INSERT INTO EMP VALUES (1,'ANIL

KUMAR',86000);"

try:

  cursor.execute(s

  ql)

  db1.commit()
```

except:

```
  db1.rollback()

  db1.close()
```

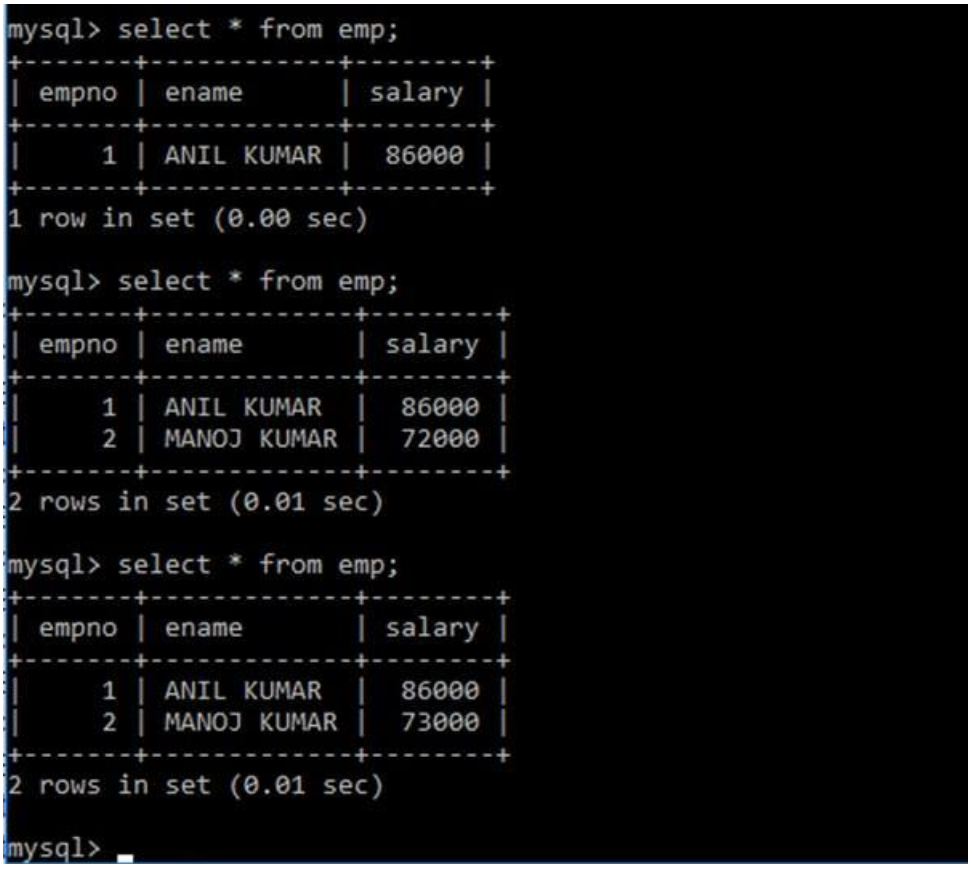
```
mysql> show tables;
+------------------+
| Tables_in_testdb |
+------------------+
| emp              |
+------------------+
1 row in set (0.00 sec)

mysql> select * from emp;
+-------+------------+--------+
| empno | ename      | salary |
+-------+------------+--------+
|     1 | ANIL KUMAR |  86000 |
+-------+------------+--------+
1 row in set (0.00 sec)

mysql>
```

Fetching all the records from EMP table having salary more than 70000.

```
import MySQLdb

db1 =

MySQLdb.connect("localhost","root","","TESTDB" )

cursor = db1.cursor()

sql = "SELECT * FROM EMP WHERE SALARY > 70000;"

try:

    cursor.execute(sql)

    #using fetchall() function to fetch all records from the table EMP
and store in  resultset

resultset = cursor.fetchall()


    for row in resultset:

       print (row)

  except:

      print ("Error: unable to fetch data")

  db1.close()
```

Updating record(s) of the table using UPDATE

```
import MySQLdb

db1 = MySQLdb.connect("localhost","root","","TESTDB" )

cursor = db1.cursor()

#Preparing SQL statement to increase salary of all employees whose salary is less than 80000

sql = "UPDATE EMP SET salary = salary +1000 WHERE salary<80000;"

 try:

    cursor.execute(sql)


    db1.commit()

except:


   db1.rollback()

db1.close()
```

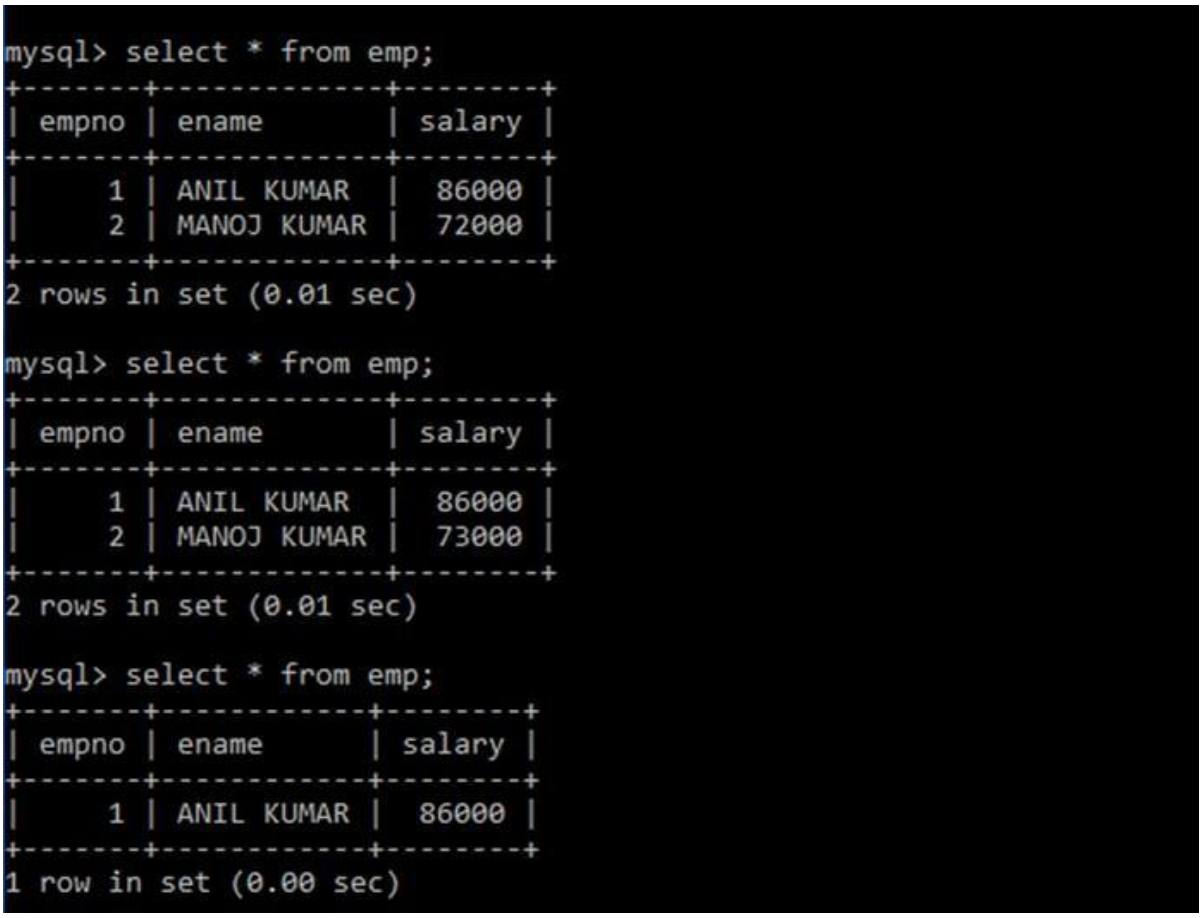


Deleting record(s) from table using DELETE

```
import MySQLdb
db1 = MySQLdb.connect("localhost","root","","TESTDB" )
 cursor = db1.cursor()
sal=int(input("Enter salary whose record to be deleted : "))
#Preparing SQL statement to delete records as per given
condition sql = "DELETE FROM EMP WHERE salary =sal”
try:
    cursor.execute(sql)
    print(cursor.rowcount, end=" record(s) deleted ")
     db1.commit()
except:
   db1.rollback()
  db1.close()
```

Output:

```
>>> Enter salary whose record to be
deleted: 80000 1 record(s) deleted
>>>
```


mysql> select * from emp;
+-------+-------------+--------+
| empno | ename | salary |
+-------+-------------+--------+
| 1 | ANIL KUMAR | 86000 |
| 2 | MANOJ KUMAR | 72000 |
+-------+-------------+--------+
2 rows in set (0.01 sec)
mysql> select * from emp;
+-------+-------------+--------+
| empno | ename | salary |
+-------+-------------+--------+
| 1 | ANIL KUMAR | 86000 |
| 2 | MANOJ KUMAR | 73000 |
+-------+-------------+--------+
2 rows in set (0.01 sec)
mysql> select * from emp;
+-------+------------+--------+
| empno | ename | salary |
+-------+------------+--------+
| 1 | ANIL KUMAR | 86000 |
+-------+------------+--------+
1 row in set (0.00 sec)

Program 20: Write a program to display unique vowels present in the given word using Stack.

Solution:

```
#Program to display unique vowels present in the given

word #using Stack


vowels =['a','e','i','o','u']

word = input("Enter the word to search for

vowels :") Stack = []

for letter in word:

   if letter in vowels:

      if letter not in Stack:

         Stack.append(letter)

print(Stack)

print("The number of different vowels present in",word,"is",len(Stack))
```