COMPUTER SCIENCE PRACTICALS

By- Kushagra Lakhwani

PYTHON REVISION TOUR

COMPUTER PRACTIAL

1. Program to calculate series 1/1^2+ 1/3^2 +1/5^2+……1/n^2

```
n=int(input('enteranumber')) series=0
for i in range(1,n+1,2):
   series=series+(1/(i**2))

print("Theanswerof1/2^2+1/3^2+1/5^2+ ..... 1/n^2isforn=",n,"is",series)
```

```
enter a number 4
The answer of 1/2^2 +1/3^2 +1/5^2+……1/n^2 is for n = 4 is 1.1111111111111112
>>>
```

2. Program to calculate series 1+x1/2! +x2/3!+-- xn/(n+1)!

```
num=int(input('enter a number ')) x=int(input('enteranother
number'))

res=0
fact=1

for i in range(1, num+1):
   fact *= (i+1)
   res = res + (i/ fact)
print("Theanswerof1+x1/2!+x2/3!+ ----------xn/(n+1)!isforx
=",x,"is",(res*x + 1))
```

```
enter a number 4
enter another number 5
The answer of 1+x1/2! +x2/3!+----------xn/(n+1)! is for x = 5 is 5.958333333333333
>>>
```

3. Program to calculate series 1+(1+2) +(1+2+3)+ --- +N

```
n=int(input('enteranumber')) ans=(n*(n+1)*(2*n+4))/12
print("Theanswerof1+(1+2)+(1+2+3)+ ----------+nisforn=",n,"is",ans)
```

```
enter a number6
The answer of 1+(1+2) +(1+2+3)+----------+n is for n = 6 is 56.0
>>>
```

4. Program to calculate series $x - x^3/3 + x^5/5 - x^7/7 + x^9/9$

```
x=int(input('enter a number')) series1=0
series2=0
for i in range(5,10,4):
  series1=series1+((x**i)/i) for j in range(3,8,4):
  series2=series2+((x**j)/j)

print("Valueofx-x3/3+x5/5-x7/7+x9/9forx=",x,"is",x-(series2-series1))
```

```
enter a number5
Value of x -  x3/3 + x5/5 - x7/7+ x9/9 for x = 5 is 206441.50793650793
>>>
```

5. Program to find the greatest common divisor of 2 no.

```
x=int(input("enter a number")) y=int(input("enter another
number"))


if x > y:
   smaller=y else:
   smaller = x
for i in range(1, smaller+1):
   if((x % i == 0) and (y % i == 0)):
      gcd = i

print("The greatest common divisor of",x,"and",y,"is",gcd)
```

```
enter a number7
enter another number5
The greatest common divisor of 7 and 5 is 1
>>>
```

6.Write a program to find LCM and HCF of 2 numbers

```
x=int(input("enter a number")) y=int(input("enter another
number"))


if x > y:
   smaller=y else:
   smaller = x
for i in range(1, smaller+1):
   if((x % i == 0) and (y % i == 0)):
      hcf = i if x > y:
     greater = x else:
   greater = y while(True):
     if((greater%x==0)and(greater%y==0)): lcm =greater
        break greater += 1

print("LCM is :",lcm,"HCF is :",hcf)
```

```
enter a number5
enter another number7
LCM is : 35 HCF is : 1
>>>
```

7. Program to find sum of
1+1/2^2+1/3^2+1/5^2+..... 1/n^2

```
n=int(input('enteranumber')) series=0
for i in range(1,n+1,2):
   series=series+(1/(i**2))

print("The answer of 1+1/2^2+1/3^2+1/5^2+... ..... 1/n^2 is for n=",n,"is",1+
series)
```

```
enter a number 6
The answer of 1 + 1/2^2 +1/3^2 +1/5^2+.....1/n^2 is for n = 6 is 2.151111111111111
>>>
```

8. Program to calculate series 1+4+7 +10+ N

```
n=int(input('enter a number')) sum=0

for i in range(1,(n*3)+1,3):
   sum=sum+i

print("The answer to 1+4+7+11+. ..... N for N=",n,"is",sum)
```

```
enter a number7
The answer to 1+4+7 +11+....N for N = 7 is 70
>>>
```

9. Write a program to sum of 10 odd numbers.

```
sum=0

for i in range(1,11,2):
   sum+=i

print("sum of the first 10 odd numbers is",sum)
```

```
sum of the first 10 odd numbers is 25
>>>
```

---

10. Write a program to find largest of 3 numbers

```
num1=int(input("ënter1stno.")) num2=
int(input("ënter2ndno.")) num3=int(input("ënter
3rdno."))


if (num1 >= num2) and (num1 >= num3):
  largest = num1
elif (num2 >= num1) and (num2 >= num3):
  largest = num2 else:
  largest = num3

print("The largest number between",num1,",",num2,"and",num3,"is",largest)
```

```
ënter 1st no.67
ënter 2nd no.34
ënter 3rd no.123
The largest number between 67 , 34 and 123 is 123
>>>
```

---

11. Program a program to show use of break and continue statement

```
import random

#Programtomakeagametogetanumberfromtheuserandcompareittoa randomnumberandruntheprogramuntiluserwins

a=int(input('enteranumberbetween1and10')) while0<a<11:
 n=random.randint(1,11) if a==n:
       print("You Win") break
   elif a!=n: print("You lose") continue
```

```
enter a number between 1 and 10 5
You lose
You lose
You lose
You Win
>>>
```

12.Program to print sum of prime numbers 1 to N.

```
n=int(input('enter a number '))

sum=0

for i in range(2,n+1):
   q=True
   for j in range(2,i):
      if i%j==0:
         q=False
         break
   if q==True:
      sum+=i

print(sum)
```

```
========== RESTART: D:/Videos/kkk.py ==========
enter a number18
58
>>>
========== RESTART: D:/Videos/kkk.py ==========
enter a number 59
440
>>>
```

13.To print sum of the digits present in a string

```
x=input('enter a string')
sum=0
for i in x:
   if i.isdigit():
      sum+=int(i)

print(sum)
```

```
enter a stringorigami123
6
>>>
```

14. To capitalize each and every first letter of a worn in a string

```
x=input("enter a string")
a=x.capitalize()
b=a[0]+""

for i in range(0,len(a)-1):
   if a[i]==" ":
      b+=" "
      b+=a[i+1].capitalize()

   else:
      b+=a[i+1]

print(b)
```

```
enter a stringsono chi no sadame jojo
Sono  Chi  No  Sadame  Jojo
>>>
```

15.To swap each element of a list with the immediate nest element

```
l=list(input('enter a list'))

for i in range(0,len(l)-1,2):
    a=l[i]
    l[i]=l[i+1]
    l[i+1]=a

print(l)
```

```
enter a listorigami
f['r', 'o', 'g', 'i', 'm', 'a', 'i']
>>>
```

16.To input a list and rotate the items in the list

```
l=eval(input('enter a list'))
last=l[len(l)-1]

for i in range(len(l),-1,-1):
    if i<len(l)-1:
        l[i+1]=l[i]

l[0]=last
print(l)
```

```
enter a list[1,2,3,4,5,6]
[6, 1, 2, 3, 4, 5]
>>>
```

17.Write a program to create a dictionary of marks of a student and calculate average marks

```
key=0
value=0
a={}
n=int(input('enter number of students'))
avg=0
for i in range(n):
    key=input('enter name')
    value=int(input('marks'))
    avg+=value
    a[key]=value

print('the average marks are ',avg/n)
```

```
enter number of students3
enter nameorigami
marks75
enter nameman
marks58
enter namepaper
marks67
the average marks are  66.66666666666667
>>>
```

---

18. To find the sum of main diagonal of a list

```
l=eval(input('enter a 2-D list'))
sum=0

for i in range(len(l)):
   for j in range(len(l)):
      if j==i:
         sum+=l[i][j]

print(sum)
```

```
enter a 2-D list[[1,2,3],[4,5,6],[6,7,8]]
14
>>>
```

19.To show Bubble sort

```
a=eval(input("enter a list"))
n=len(a)
for i in range(len(a)):
   for j in range (0,n-i-1):
      if a[j]>a[j+1]:
         a[j],a[j+1]=a[j+1],a[j]
print(a)
#ascending
```

```
enter a list[13,3,7,29,19,27,2]
[2, 3, 7, 13, 19, 27, 29]
>>>
```

Functions & Search Algorithms

Passing a Tuple for modification – change not reflected in original tuple

```
def modify_tuple(T):
   T = T + (3, 5)
   print("Contents of tuple inside function : ",
T) T = (9, 10, 7, 16)
print("Contents of tuple before function call : ", T)
modify_tuple(T)
print("Contents of tuple after function call : ", T)
```

Output :

```
Contents of tuple before function call :  (9, 10, 7, 16)
Contents of tuple inside function :  (9, 10, 7, 16, 3, 5)
Contents of tuple after function call : (9, 10, 7, 16)
```

Passing a List for modification – modifies original list

```
def modify_list(L):
   L.extend([3, 5])
   print("Contents of list inside function : ",
L) L = [9, 10, 7, 16]
print("Contents of list before function call : ",
L) modify_list(L)
print("Contents of list after function call : ", L)
```

Output:

```
Contents of list before function call :  [9, 10, 7, 16]
Contents of list inside function :  [9, 10, 7, 16, 3, 5]
Contents of list after function call : [9, 10, 7, 16, 3, 5]
```

Program : Write a Program to enter the numbers and find Linear Search, Binary Search, Lowest Number and Selection Sort using list/array code.

Solution:

```
arr=[]

def array_operation():

  ch=1

  while ch!=10:

    print('Various Array operation\n')

    print('1 Create and Enter value\n')

    print('2 Print Array\n')

    print('3 Reverse Array\n')
    print('4 Linear Search\n')

    print('5 Binary Search\n')

    print('6 Lowest Number \n')

    print('7 Selection Sort\n')

    print('10 Exit\n')

    ch=int(input('Enter Choice '))

    if ch==1 :

      appendarray() elif

    ch==2 :

      print_array() elif
```

```
    ch==3 :
      reverse_array() elif
    ch==4 :
      linear_search() elif
    ch==5 :
      binary_search() elif
    ch==6 :
      min_number() elif
    ch==7 :
      selection_sort()


def appendarray(): for i
  in range(0,10):
    x=int(input('Enter Number : '))
    arr.insert(i,x)
#.....

def print_array():
  for i in range(0,10):
    print(arr[i]),
#.....

def reverse_array(): for i
```

```
  in range(1,11):
    print(arr[-i]),
#.....
def lsearch(): try:
    x=int(input('Enter the Number You want to search
    : ')) n=arr.index(x)
    print ('Number Found at %d location'%
  (i+1)) except:
    print('Number Not Exist in list')
#.....
def linear_search():
  x=int(input('Enter the Number you want to
  search : ')) fl=0
  for i in range(0,10): if
    arr[i]==x :
      fl=1
      print ('Number Found at %d location'%
      (i+1)) break
  if fl==0 :
    print ('Number Not Found')
#.....
```

```
def binary_search():
  x=int(input('Enter the Number you want to
  search : ')) fl=0
  low=0 heigh=len(arr)
  while low<=heigh :
    mid=int((low+heigh)/2) if
    arr[mid]==x :
      fl=1
      print ('Number Found at %d location'%
      (mid+1)) break
    elif arr[mid]>x :
      low=mid+1
    else :
      heigh=mid-1

  if fl==0 :
    print ('Number Not Found')

#.....

def min_number():
  n=arr[0] k=0
```

```
   for i in range(0,10): if
      arr[i]<n :
         n=arr[i] k=i
   print('The Lowest number is %d '%(n))


#.....

def selection_sort(): for i
   in range(0,10):
      n=arr[i] k=i
      for j in range(i+1,10): if
         arr[j]<n :
            n=arr[j] k=j
      arr[k]=arr[i] arr[i]=n
array_operation()
```

```
on37-32/prog_array_oprtn.py
Various Array operation

1 Create and Enter value

2 Print Array

3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit

Ln: 192  Col: 4
```

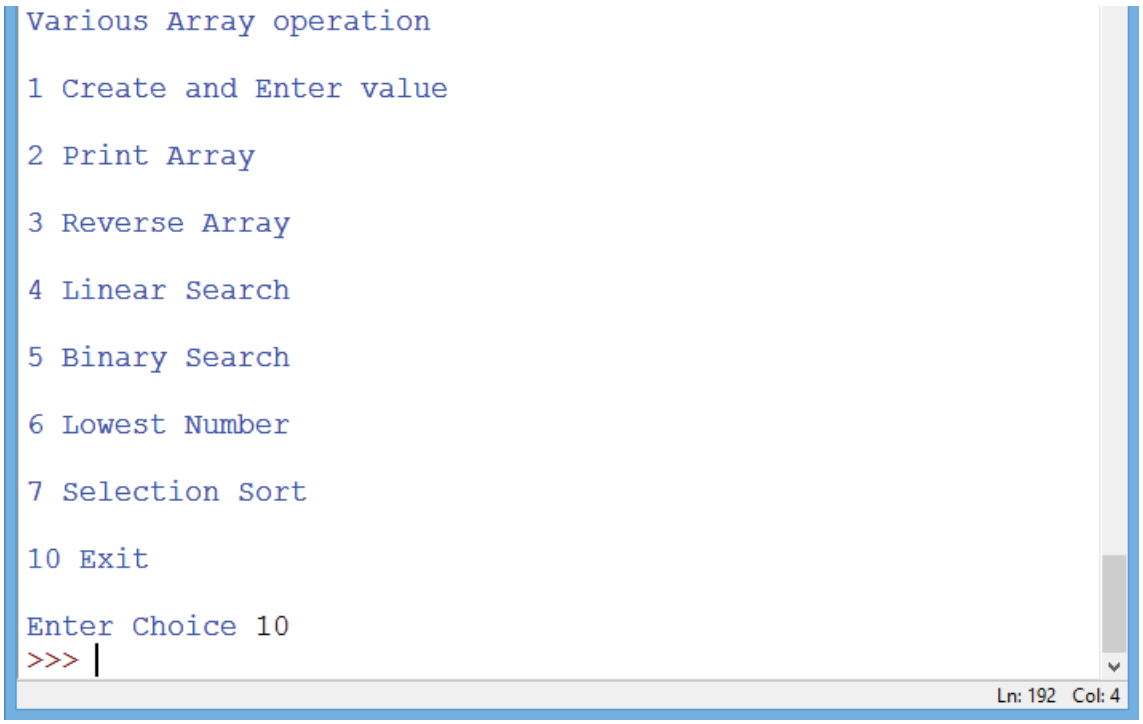
```
Enter Choice 1
Enter Number : 50
Enter Number : 20
Enter Number : 10
Enter Number : 22
Enter Number : 55
Enter Number : 33
Enter Number : 67
Enter Number : 56
Enter Number : 78
Enter Number : 90
Various Array operation

1 Create and Enter value

2 Print Array

3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit

Enter Choice 2
```

```
Enter Choice 2
50
20
10
22
55
33
67
56
78
90
Enter Choice 3
90
78
56
67
33
55
22
10
20
50
Various Array operation

1 Create and Enter value

2 Print Array
```

```
3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit

Enter Choice 4
Enter the Number you want to search : 56
Number Found at 8 location
```

```
Various Array operation

1 Create and Enter value

2 Print Array

3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit

Enter Choice 5
Enter the Number you want to search : 50
Number Found at 1 location
```

```
Various Array operation

1 Create and Enter value

2 Print Array

3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit

Enter Choice 6
The Lowest number is 10
```


Various Array operation
1 Create and Enter value
2 Print Array
3 Reverse Array
4 Linear Search
5 Binary Search
6 Lowest Number
7 Selection Sort
10 Exit
Enter Choice 10
>>>
Ln: 192 Col: 4

File Handling

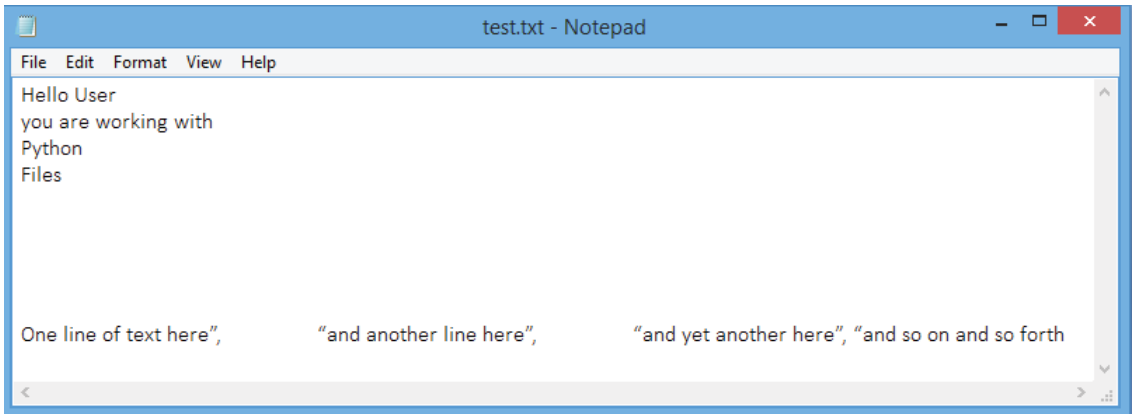
Program : Write a Program to read data from data file and show Data File Handling related functions utility in python.

```
f=open("test.txt",'r')
print(f.name)
f_contents=f.read()
print(f_contents)
f_contents=f.readlines()
print(f_contents)
f_contents=f.readline()
print(f_contents)
for line in f: print(line,
  end='')
f_contents=f.read(50)
print(f_contents)
size_to_read=10
f_contents=f.read(size_to_read)
while len(f_contents)>0:
  print(f_contents)
  print(f.tell())
  f_contents=f.read(size_to_read)
```

```
test.txt
Hello User
you are working with
Python
Files




[]


>>>
```

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

```
#Program to read data from data file in append

af=open("test.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()
```



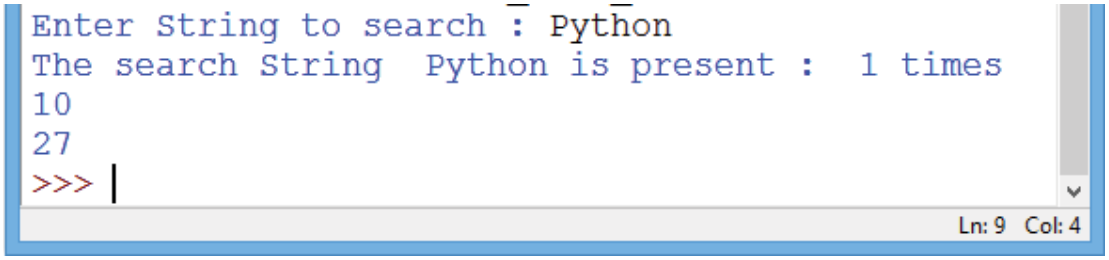
Program : Write a Program to read data from data file in read mode and count the particular word occurrences in given string, number of times in python.

Solution:

```
#Program to read data from data file in read mode

and #count the particular word occurrences in

given string, #number of times in python.

f=open("test.txt",'r')

read=f.readlines()

f.close()

times=0 #the variable has been created to show the number of times the loop

runs times2=0 #the variable has been created to show the number of times the

loop runs chk=input("Enter String to search : ")

count=0

for sentence in read:

  line=sentence.split()

  times+=1

  for each in line:

    line2=each times2+=1

    if chk==line2:

      count+=1

print("The search String ", chk, "is present : ", count,
```

"times") print(times)

print(times2)



Program : Write a Program to read data from data file in read mode and append the words starting with letter 'T' in a given file in python.

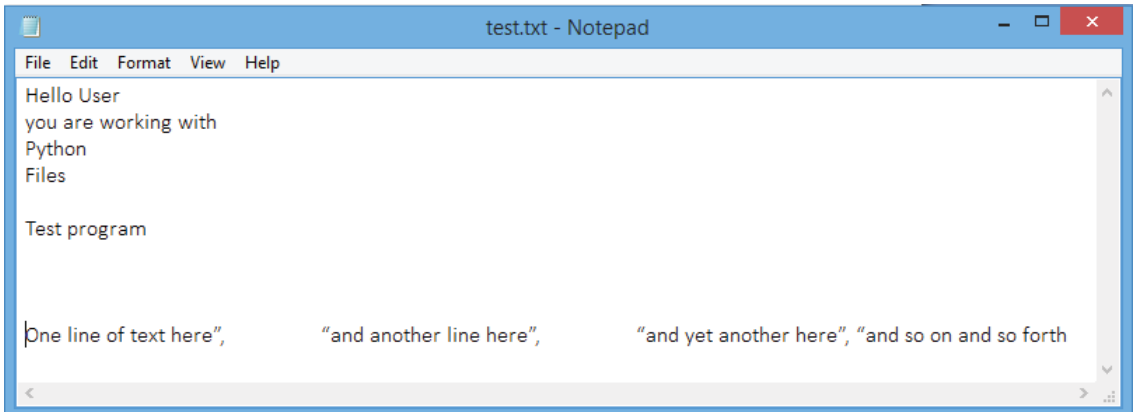
Solution:

```
#Program to read data from data file in read mode

#append the words starting with letter 'T'

#in a given file in python

f=open("test.txt",'r')

read=f.readlines() f.close()

id=[]

for ln in read:

  if ln.startswith("T"):

    id.append(ln)
```

print(id)

```
ams/Python/Python37-32/prog_file_handl4.py
['Test program\n']
>>>
```

Ln: 9 Col: 4



MySql, Python Connectivity and Joins

Stack and Queue

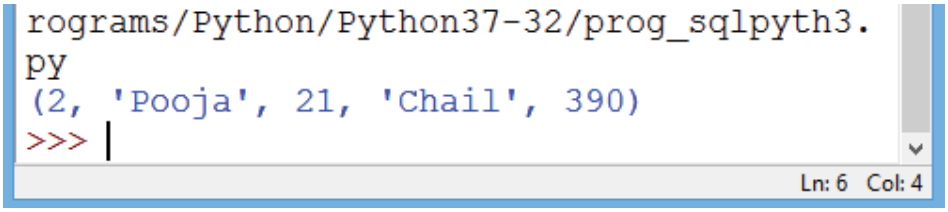
Program : Write a Program to show MySQL database connectivity in python.

Solution:

```
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)
```

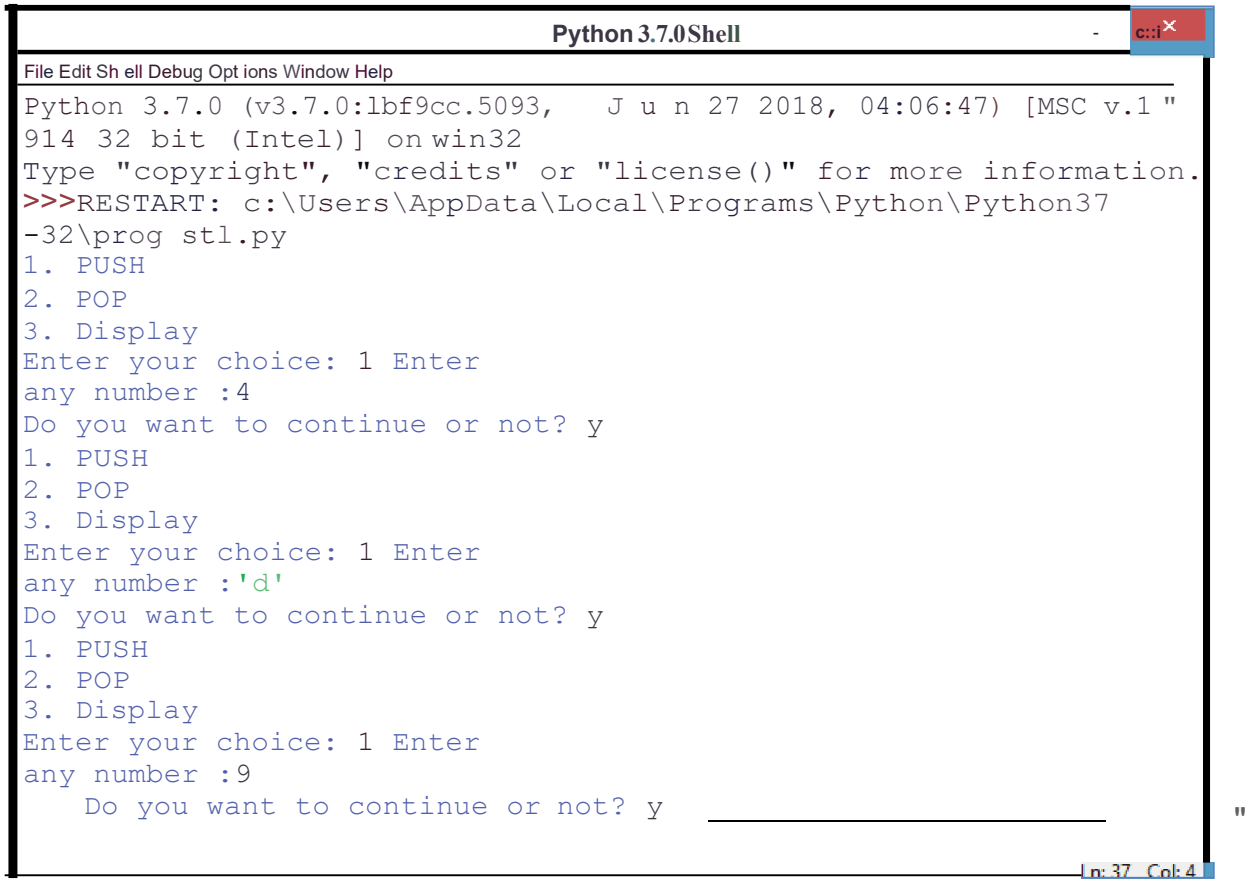
Program : Write a Python program to implement all basic operations of a stack, such as adding element (PUSH operation), removing element (POP operation) and displaying the stack elements (Traversal operation) using lists.

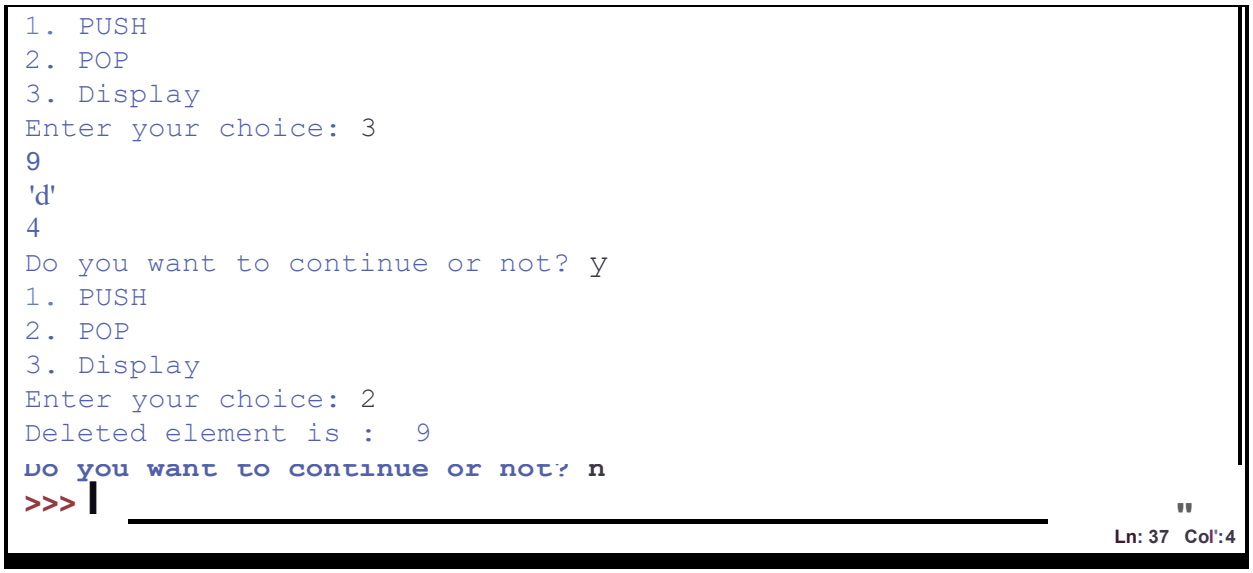
#Implementation of List as stack




```
s=[]

c="y"

while (c=="y"):

  print ("1. PUSH")
  print ("2. POP ") print

  ("3. Display")
```

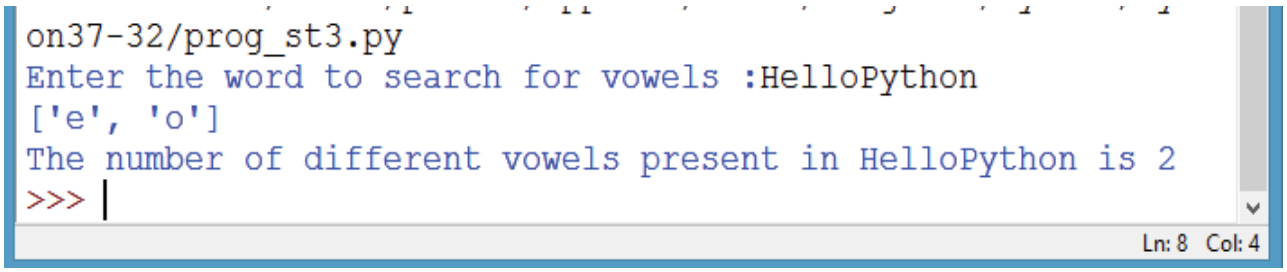
```
choice=int(input("Enter your choice:
")) if (choice==1):
   a=input("Enter any number :")
   s.append(a)
elif (choice==2): if
   (s==[]):
      print ("Stack Empty")
   else:
      print ("Deleted element is : ,s.pop())
elif (choice==3):
   l=len(s)
   for i in range(l-1,-1,-1): #To display elements from last element to
      first print (s[i])
else:
   print("Wrong Input")
c=input("Do you want to continue or not? ")
```


Python 3.7.0 Shell
File Edit Sh ell Debug Opt ions Window Help
Python 3.7.0 (v3.7.0:1bf9cc.5093, J u n 27 2018, 04:06:47) [MSC v.1 "
914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>RESTART: c:\Users\AppData\Local\Programs\Python\Python37
-32\prog stl.py
1. PUSH
2. POP
3. Display
Enter your choice: 1 Enter
any number :4
Do you want to continue or not? y
1. PUSH
2. POP
3. Display
Enter your choice: 1 Enter
any number :'d'
Do you want to continue or not? y
1. PUSH
2. POP
3. Display
Enter your choice: 1 Enter
any number :9
Do you want to continue or not? y
Ln: 37 Col: 4


1. PUSH
2. POP
3. Display
Enter your choice: 3
9
'd'
4
Do you want to continue or not? y
1. PUSH
2. POP
3. Display
Enter your choice: 2
Deleted element is : 9
Do you want to continue or not? n
>>>
Ln: 37 Col: 4

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

```
#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))
```



Program: Write a program in Python to add, delete and display elements from a queue using list.

Solution:

```
#Implementing List as a Queue - using function append()
and pop() a=[]
c='y'
while (c=='y'):
  print ("1. INSERT")
  print ("2. DELETE ")
  print ("3. Display")
  choice=int(input("Enter your choice:
  ")) if (choice==1):
    b=int(input("Enter new number: "))
    a.append(b)
  elif (choice==2): if
    (a==[]):
      print("Queue Empty") else:
      print ("Deleted element is:",a[0])
      a.pop(0)
  elif (choice==3): l=len(a)
    for i in range(0,l):
      print (a[i])
  else:
```

```
print("wrong input")
c=input("Do you want to continue or not: ")
```

```
\Python37-32\prog_qu1.py
1. INSERT
2. DELETE
3. Display
enter your choice 1
enter new number 5
do you want to continue or not y
1. INSERT
2. DELETE
3. Display
enter your choice 1
enter new number 8
do you want to continue or not y
1. INSERT
2. DELETE
3. Display
enter your choice 1
enter new number 44
do you want to continue or not y
```

Ln: 37 Col: 4

```
1. INSERT
2. DELETE
3. Display
enter your choice 3
5
8
44
do you want to continue or not y
1. INSERT
2. DELETE
3. Display
enter your choice 2
deleted element is: 5
do you want to continue or not n
>>>
```

Ln: 37 Col: 4

Program : 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

```
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(sql)

  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>
```

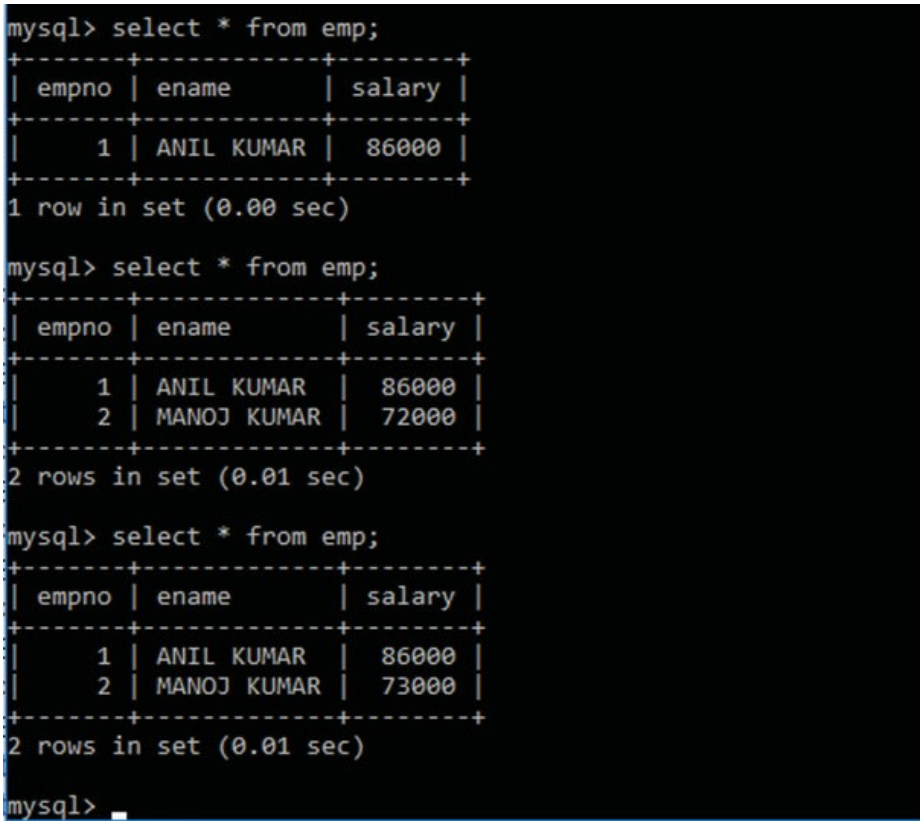
Fetching all the records from EMP table having salary more than 70000.

```
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()
```


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

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;
+--•----+--•--•-------+--•·-···+
I empno | ename       | salary |
+--••--•+--•--••--•--•+--••--·-+
      1 | ANIL KUMAR  |  86000 |
      2 | HANO) KUMAR |  72080 |
+--·····+---------·---+·-····-·+
2 rows in set (0.01 sec)

nysql> select • from emp;
+--·····+-------------+--·····+
I empno I enome       | solory |
+-·-····+-·--·-------+-···---·-+
      1 | ANIL KUMAR  |  86000|
      2 |  HANO) l(UIAR |  73080 |
+·-·•--·+--•---•---·-·+·-··---·+
2 rows in set (0.01 sec)

nysql> select • from emp;
+--·····+···---·--··-+·····+
I  empno| ename      1 salary |
+·····+·----------+--·····+
      1 | ANIL KUMAR  |  86800|
+·-····+-··---------+·····+
1 row in set (0.00 sec)
```

TABLE MEMBER

MID	NAME	DCODE	ISSUEDATE
101	AGAM SINGH	R102	2017-11-30
103	ARTH JOSEPH	F102	2016-12-13
102	NISHAHANS C101		2017-07-24

i. To display all details from the table MEMBER in descending order of ISSUEDATE.
Select * from member order by issuedate desc

ii. To display the DCODE and DTITLE of all Folk Type DVDs from the table DVD
Select dcode, dtitle from dvd where dtype="folk"

iii. To display the DTYPE and number of DVDs in each DTYPE from the table DVD
Select dtype, count(dtype) from dvd group by dtype

iv. To display all NAME and ISSUEDATE of those members from the table MEMBER who have DVDs issued (i.e ISSUEDATE) in the year 2017
select name, issuedate from member where issuedate ="2017%"

TABLE SALESPERSON

TABLE : SALESPERSON

Code	NAME	SALARY	ITCODE
1001	TANDEEP JHA	60000	I2
1002	YOGRAJ SINHA	70000	I5
1003	TENZIN JACK	45000	I2
1005	ANOKHI RAJ	50000	I7
1004	TARANA SEN	55000	I7

TABLE : ITEM

ITCODE	ITEMTYPE	TURNOVER
I5	STATIONARY	3400000
I7	HOSIERY	6500000
I2	BAKERY	10090000

(i) To display the CODE and NAME of all SALESPERSON having "I7" ITCode from the table SALESPERSON.

(ii) To display all details from table SALESPERSON in descending order of SALARY.

(iii) To display the number of SALESPERSON dealing in each TYPE of ITEM.(Use ITCODE for the same)

(iv) To display NAME of all the salespersons from the SALES PERSON table along with their corresponding ITEMTYPE from the ITEM table.

i) Select CODE, NAME from SALESPERSON where ITCode=17;

ii) Select * from salesperson order by salary desc

iii) Select count(*),itcode from salesperson group by itcode

iv) Select name, itemtype from salesperson, item where salesperson.code=item.itcode