Prg.no: 01 **Add line to the text file**

Program:

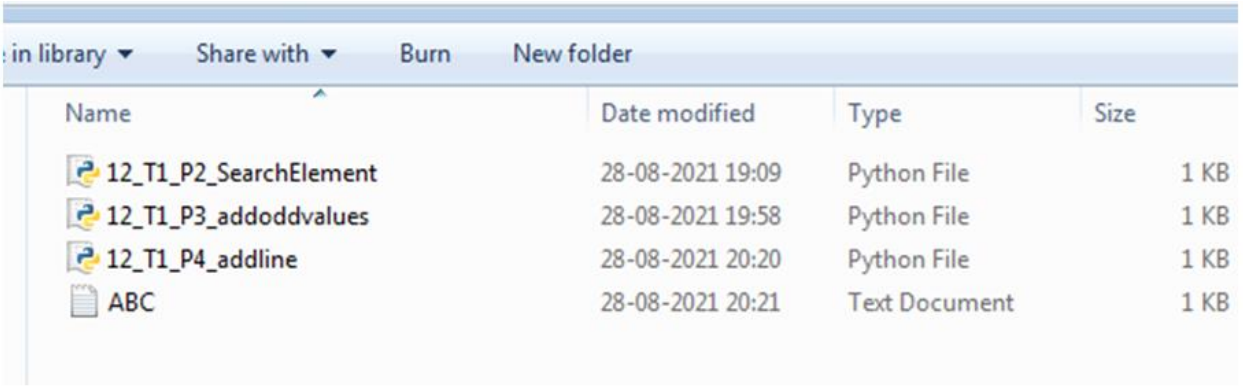
```
def program():
        f = open("ABC.txt","w")
        text=input("Enter the text:")
        f.write(text)
        f.close()
program()
```

Sample Input and Output:

ADD SINGLE LINE BY THE USER

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Enter the text: Have a nice Day!


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

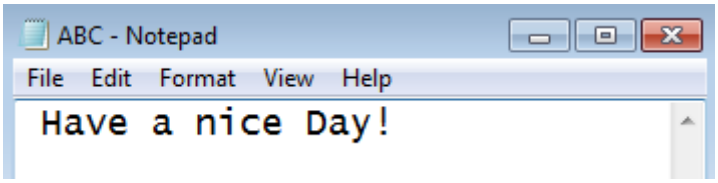
Prg.no: 02 Display the number of vowels/Consonants/Uppercase/Lowercase letters

Program:

```
#Read a text file and display the number of vowels/consonants/uppercase/lowercase characters in the file
print("Display the number of vowels/Consonants/Uppercase/Lowercase letters ")
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ")

f=open("ABC.txt","r")
rd=f.read()

v=0
c=0
lc=0
uc=0

for ch in rd:
   if (ch.islower()):
      lc+=1
   elif(ch.isupper()):
      uc+=1

   ch=ch.lower()

   if( ch in ['a','e','i','o','u']):
      v+=1
   elif (ch in ['b','c','d','f','g',
                'h','j','k','l','m',
                'n','p','q','r','s',
                't','v','w','x','y','z']):
       c+=1
f.close()
print("Vowels are : ",v)
print("Consonants are : ",c)
print("Lower case letters are : ",lc)
print("Upper case letters are : ",uc)
```

Sample Input and Output:

```
Display the number of vowels/Consonants/Uppercase/Lowercase letters
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Vowels are :  6
Consonants are :  6
Lower case letters are :  10
Upper case letters are :  2
```



Prg.no: 03 **Random Numbers Between 1 And 6**

Program:

```
#To generate random numbers between 1 and 6 using user defined function

print(" RANDOM NUMBERS BETWEEN 1 AND 6")
print(" ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")

def fun():
   import random
   r = random.randint(1,6)
   print("Random number generated between 1 to 6 : ",r)
fun()
```

Sample Input and Output:

```
RANDOM NUMBERS BETWEEN 1 AND 6
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Random number generated between 1 to 6 :  4
```

Prg.no: 04 **String Functions**

Program:

```
# To implement Python string functions
print (" STRING FUNCTIONS ")
print (" ~~~~~~~~~~~~~~~~ ")

a = str(input("Enter Sentence : "))
b = input("Enter the spacing :")
print("The string entered is a word : ",a.isalpha())
print("The string entered in lowercase : ",a.lower())
print("The string entered is in lowercase : ",a.islower())
print("The string entered in uppercase : ",a.upper())
print("The string entered is in uppercase : ",a.isupper())
print("The string entered after removing the space from left side : ",a.lstrip())
print("The string entered after removing the space from right side : ",a.rstrip())
print("The string entered contains whitespace : ",a.isspace())
print("The string entered is titlecased : ",a.istitle())
print("The string entered after joining with ",b," : ",b.join(a))
print("The string entered after swaping case : ",a.swapcase())
```

Sample Input and Output:

```
STRING FUNCTIONS
 ~~~~~~~~~~~~~~~~
Enter Sentence : Welcome
Enter the spacing :#
The string entered is a word :  True
The string entered in lowercase :  welcome
The string entered is in lowercase :  False
The string entered in uppercase :  WELCOME
The string entered is in uppercase :  False
The string entered after removing the space from left side :  Welcome
The string entered after removing the space from right side :  Welcome
The string entered contains whitespace :  False
The string entered is titlecased :  True
The string entered after joining with  #  :  W#e#l#c#o#m#e
The string entered after swaping case :  wELCOME
```

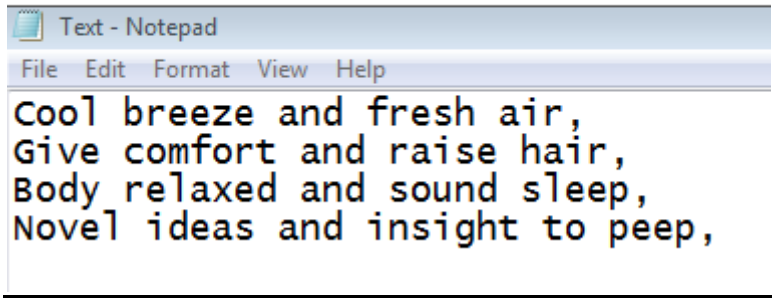
Prg.no: 05 Display File Content Line By Line With Each Word Separated By #

Program:

```
# To read & Display file content line by line with each word separated by #
print ("DISPLAY FILE CONTENT LINE BY LINE WITH EACH WORD SEPARATED BY #")
print ("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")

a = open("Text.txt","r")
l = a.readlines()
for line in l:
   x = line.split()
   for y in x:
      print(y+" # ",end = " ")
   print(" ")
```

Sample Input and Output:

```
DISPLAY FILE CONTENT LINE BY LINE WITH EACH WORD SEPARATED BY #
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Cool #  breeze #  and #  fresh #  air, #
Give #  comfort #  and #  raise #  hair, #
Body #  relaxed #  and #  sound #  sleep, #
Novel #  ideas #  and #  insight #  to #  peep, #
```



Prg.no: 06 **Count & Display Odd & Even Numbers**

Program:

```
# To input n numbers in tuple & count how many even & odd numbers are entered

def fun(t):
   e=0
   o=0
   for i in range(0,len(t)):
      if t[i] % 2 == 0:
         e+=1
      else:
         o+=1
   print("\nNumber of even numbers : ",e,"\nNumber of odd numbers : ",o)



#main
print("COUNT & DISPLAY EVEN & ODD NUMBERS")
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")

x=eval(input("\nEnter a tuple : "))
fun(x)
```

Sample Input and Output:

```
COUNT & DISPLAY EVEN & ODD NUMBERS
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Enter a tuple : (1,2,3,4,5,6,7)

Number of even numbers :  3
Number of odd numbers :  4
```

Prg.no: 07 **Text File Manipulation**

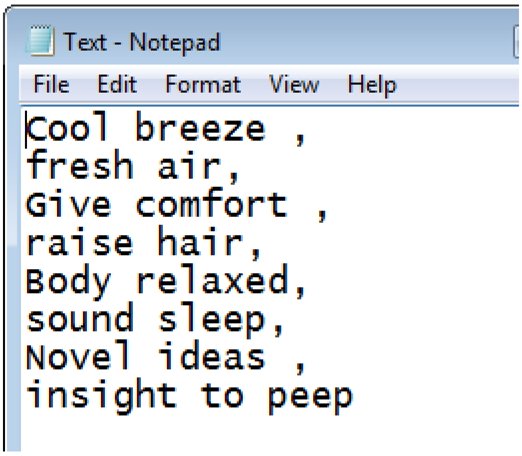
Program:

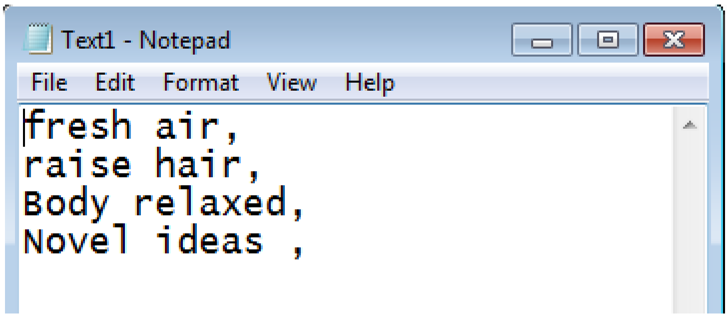
```
# To remove all the lines that contain the character 'a'in a file & write it into another file

print( " Write the lines in the new file that contain the character 'a' ")
print( " ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ")

f = open("Text.txt","r")
f1 = open("Text1.txt","w")
l= f.readlines()

for i in l:
    if 'a' in i:
        f1.write(i)

f.close()
f1.close()

f1 = open("Text1.txt","r")
b = f1.read()
print(b)
f1.close()
```

Sample Input and Output:

Write the lines in the new file that contain the character 'a'
~~~~~~

fresh air,
raise hair,
Body relaxed,
Novel ideas ,




~~~~~~

Prg.no: 08 **Binary File Manipulation**

Program:

```
#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.

import pickle
def write():
   f= open("Birec.dat","wb")
   record=[]
   while True:
      rno = int(input("\nEnter roll no : "))
      name = input("Enter name : ")
      marks = int(input("Enter marks : "))
      data=[rno,name,marks]
      record.append(data)
      ch=input("\nDo you want to enter more record( y/n) ?")
      if ch == 'n':
         break
   pickle.dump(record,f)

def read():
   f = open("Birec.dat","rb")
   s= pickle.load(f)
   print("Roll no\t\tName\t\tMarks")
   print("~~~~~~~\t\t~~~~\t\t~~~~~\n")
   for i in s:
      rno=i[0]
      name=i[1]
      marks=i[2]
      print(rno,"\t\t",name,"\t\t",marks)

def search():
   f=open("Birec.dat","rb")
   s=pickle.load(f)
   found = 0
   rno = int(input("\nEnter roll no : \n"))
   for i in s: vc
      if i[0] == rno:
         print("Record found ....\n")
         print("Roll No = ",i[0])
         print("Name = ", i[1])
         print("Marks = ", i[2])
         found = 1
   if found == 0:
         print("\nRecord not found.....")

#main

print("WRITE, READ AND SEARCH RECORD IN A BINARY FILE")
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
```

```
while True:
    print("\n1.Write a Record")
    print("2.Display")
    print("3.Search")
    print("4.Exit")
    ch = int(input("\nEnter the choice : "))
    if ch == 1:
        write()
    elif ch == 2:
        read()
    elif ch == 3:
        search()
    elif ch == 4:
        break
    else:
        print("\nInvalid Choice..!")
        break
```

Sample Input and Output:

```
WRITE, READ AND SEARCH RECORD IN A BINARY FILE
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

1.Write a Record
2.Display
3.Search
4.Exit

Enter the choice : 1

Enter roll no : 1
Enter name : Geetha
Enter marks : 98

Do you want to enter more record( y/n) ?y

Enter roll no : 2
Enter name : Manu
Enter marks : 92

Do you want to enter more record( y/n) ?n

1.Write a Record
2.Display
3.Search
4.Exit

Enter the choice : 2
Roll no         Name          Marks
~~~~~~~         ~~~~          ~~~~~

1               Geetha        98
2               Manu           92
```

Prg.no: 09 **Update Record in Binary file**

Program:

```
# Create a binary file with roll number, name and marks , input a roll number and
#update the marks.
import pickle
def write():
   f= open("Birec.dat","wb")
   record=[]
   while True:
      rno = int(input("\nEnter roll no : "))
      name = input("Enter name : ")
      marks = int(input("Enter marks : "))
      data=[rno,name,marks]
      record.append(data)
      ch=input("\nDo you want to enter more record( y/n) ?")
      if ch == 'n':
         break
   pickle.dump(record,f)

def read():
   f = open("Birec.dat","rb")
   s= pickle.load(f)
   print("Roll no\t\tName\t\tMarks")
   print("~~~~~~~\t\t~~~~\t\t~~~~~\n")
   for i in s:
      rno=i[0]
      name=i[1]
      marks=i[2]
      print(rno,"\t\t",name,"\t\t",marks)

def search():
   f=open("Birec.dat","rb")
   s=pickle.load(f)
   found = 0
   rno = int(input("\nEnter roll no : \n"))
   for i in s:
      if i[0] == rno:
         print("Record found ....\n")
         print("Roll No = ",i[0])
         print("Name = ", i[1])
         print("Marks = ", i[2])
         found = 1
   if found == 0:
         print("\nRecord not found.....")

def update():
   f=open("Birec.dat","rb+")
   s=pickle.load(f)
   final=0
   n=int(input("Enter Roll.No : "))
   for i in s:
```

```
        if n ==i[0]:
            print("Current Marks : ",i[2])
            i[2] = int(input("Enter new marks : "))
            print("\n Marks updated successfully...")
            final=1
            break
    if final == 0:
        print("No records found")
    else:
        f.seek(0)
        pickle.dump(s,f)
        print(s)
    f.close()

#main

print("WRITE, READ AND SEARCH RECORD IN A BINARY FILE")
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")

while True:
    print("\n1.Write a Record")
    print("2.Display")
    print("3.Search")
    print("4.Update")
    print("5.Exit")
    ch = int(input("\nEnter the choice : "))
    if ch == 1:
        write()
    elif ch == 2:
        read()
    elif ch == 3:
        search()
    elif ch == 4:
        update()
    elif ch == 5:
        break
    else:
        print("\nInvalid Choice..!")
        break
```

Sample Input and Output:

```
WRITE, READ AND SEARCH RECORD IN A BINARY FILE
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

1.Write a Record
2.Display
3.Search
4.Update
5.Exit

Enter the choice : 1

Enter roll no : 1
Enter name : Banu
Enter marks : 45

Do you want to enter more record( y/n) ?y

Enter roll no : 2
Enter name : Reenu
Enter marks : 67

Do you want to enter more record( y/n) ?y

Enter roll no : 3
Enter name : Rathi
Enter marks : 78

Do you want to enter more record( y/n) ?n

Enter the choice : 2
Roll no         Name          Marks
~~~~~~~         ~~~~          ~~~~~

1               Banu           45
2               Reenu          67
3               Rathi          78

1.Write a Record
2.Display
3.Search
4.Update
5.Exit

Enter the choice : 4
Enter Roll.No : 1
Current Marks :  45
Enter new marks : 78

 Marks updated successfully...
[[1, 'Banu', 78], [2, 'Reenu', 67], [3, 'Rathi', 78]]
```

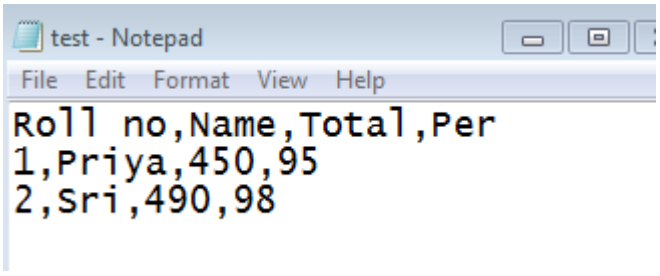
Prg.no: 10 **Read Data from CSV file**

Program:

```
#CSV file reader

from csv import reader
def prg():
   f=open("test.csv","r")
   d = reader(f,delimiter=",")
   r = next(d)
   e = list(d)
   f.close()
   for i in e:
      for j in i:
         print(j,"\t", end = " ")
      print()

# main

print(" CSV file reader")
print(" ~~~~~~~~~~~~~~~~")
prg()
```

Sample Input and Output:

```
CSV file reader
 ~~~~~~~~~~~~
1        Priya   450     95
2        Sri     490     98
```



Prg.no: 11 **Write data into CSV file**

Program:

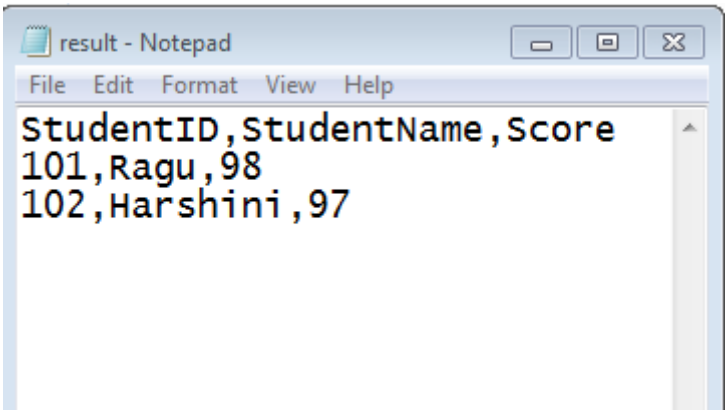
```
#CSV file Writer1

from csv import writer
def prg():

   #Create Header

   f=open("result.csv","w",newline ="\n")
   d = writer(f)
   d.writerow(['StudentID','StudentName','Score'])
   f.close()

   #Insert data

   f=open("result.csv","a",newline ="\n")

   while True:
      st_id=int(input("Enter Student ID : "))
      st_name = input("Enter Student name : ")
      st_score = input("Enter Score :")
      d = writer(f)
      d.writerow([st_id,st_name,st_score])
      ch = input("Do you want to insert another record ? (y/n)")
      ch=ch.lower()
      if ch != "y":
         break
   print("\n Record has been added..")
   f.close()

#main
print("WRITE DATA INTO CSV FILE")
print("~~~~~~~~~~~~~~~~~~~~~~~~~")

prg()
```

Sample Input and Output:

WRITE DATA INTO CSV FILE
~~~~~~~~~~~~~~~~~~~~~~~~

Enter Student ID : 101
Enter Student name : Ragu
Enter Score :98
Do you want to insert another record ? (y/n)y
Enter Student ID : 102
Enter Student name : Harshini
Enter Score :97
Do you want to insert another record ? (y/n)n

Record has been added..


~~~~~~~~~~~~~~~~~~~~~~~~

Prg.no: 12 **Search Results in CSV FIle**

Program:

```
import csv

def prg():
   with open("users.csv","a",newline="") as f:
      w=csv.writer(f,delimiter=",")
      while True:
         email=input("\nEnter emailID  : ")
         password = input("Enter your password : ")
         password2 = input("Retype your password : ")
         if password==password2:
            w.writerow([email,password])
            print("\nData inserted")
         else:
            print("\nPlease try again...")
         ch=input("\nDo you want to enter more record( y/n) ?")
         if ch == 'n':
            break

def login():
   password=input("\nEnter password : ")
   with open("users.csv","r") as f:
      r=csv.reader(f,delimiter=",")

      for i in r:
         if password == i[1]:
            print("\nEmail : ",i[0])
            print("Password : ",i[1])
            return True

   print("\nTry again!")
   return False
# Main

print("SEARCH RESULTS IN CSV FILE")
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~")

while True:
   print("\n1.Insert Record")
   print("2.Search")
   print("3.Exit")
   ch = int(input("\nEnter the choice : "))
   if ch == 1:
      prg()
   elif ch == 2:
      login()
   elif ch == 3:
      break
   else:
      print("\nInvalid Choice..!")
```

```
    break
```

Sample Input and Output:

```
SEARCH RESULTS IN CSV FILE
~~~~~~~~~~~~~~~~~~~~~~~~~~

1.Insert Record
2.Search
3.Exit

Enter the choice : 1

Enter emailID  : happy@gmail.com
Enter your password : WelCOme@9!
Retype your password : WelCOme@9!

Data inserted

Do you want to enter more record( y/n) ?y

1.Insert Record
2.Search
3.Exit

Enter the choice : 1

Enter emailID  : smile@gmail.com
Enter your password : smiLE@4
Retype your password : smiLE@4

Data inserted
Do you want to enter more record( y/n) ?n

1.Insert Record
2.Search
3.Exit

Enter the choice : 2
Enter password : smiLE@4
Email :  smile@gmail.com
Password :  smiLE@4
```

Prg.no: 13 **STACK OPERATION**

Write a Python program to implement a stack using a list data-structure.

PROGRAM:

```
stack=[ ]
def view( ):
    for x in range(len(stack)):
        print(stack[x])
def push( ):
    item=int(input("Enter integer value"))
    stack.append(item)
def pop( ):
   if(stack= =[ ]):
        print("Stack is empty")
    else:
        item=stack.pop(-1)
        print("Deleted element:",item)
def peek( ):
    item=stack[-1]
    print("Peeked element:",item)
print("Stack operation")
print("**********")
print("1.view")
print("2.push")
print("3.pop")
print("4.peek)
while True:
    choice=int(input("Enter your choice"))if
    choice= =1:
         view( )
    elif choice= =2:
         push( )
    elif choice==3:
         pop( )
    elif choice= =4:
```

```
        peek( )
    else:
        print("Wrong choice")
        break
```

OUTPUT:

```
Stack operation
***********
1.view
2.Push
3.pop
4.peek
Enter your choice 2
Enter integer value 56
Enter your choice 2
Enter integer value 78
Enter your choice 2
Enter integer value 90
Enter your choice 1
56
78
90
Enter your choice 4
Peeked element: 90
Enter your choice 3
Deleted element: 90
Enter your choice 1
56
78
Enter your choice 4
Peeked element: 78
Enter your choice 5
Wrong choice
```

Prg.no: 14 **STACK OPERATION (EMPLOYEE DETAILS)**

PROGRAM:

```
#stack implementation using functions
#program to create a stack of employee(empno,name,sal).

def line():
   print('~'*75)
employee=[]
def push():
      empno=input("Enter empno  ")
      name=input("Enter name ")
      sal=input("Enter sal ")
      emp=(empno,name,sal)
      employee.append(emp)
def pop():
      if(employee==[]):
            print("Underflow / Employee Stack in empty")
      else:
            empno,name,sal=employee.pop()
            print("poped element is ")
            print("empno ",empno," name ",name," salary ",sal)
def traverse():
      if not (employee==[]):
         n=len(employee)
         for i in range(n-1,-1,-1):
            print(employee[i])
         else:
            print("Empty , No employee to display")
while True:
   line()
   print("1. Push")
   print("2. Pop")
   print("3. Traversal")
   print("4. Exit")
   ch=int(input("Enter your choice "))
   if(ch==1):
      push()
   elif(ch==2):
      pop()
   elif(ch==3):
      traverse()
   elif(ch==4):
      print("End")
      break
```

```
    else:
        print("Invalid choice")
```

OUTPUT:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1. Push
2. Pop
3. Traversal
4. Exit
Enter your choice 1
Enter empno  1001
Enter name Sanjay
Enter sal 50000
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1. Push
2. Pop
3. Traversal
4. Exit
Enter your choice 1
Enter empno  1002
Enter name Arun
Enter sal 37000
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1. Push
2. Pop
3. Traversal
4. Exit
Enter your choice 3
('1002', 'Arun', '37000')
('1001', 'Sanjay', '50000')
```

Prg.no: 15 **PALINDROME USING STACK**

PROGRAM:

```
stack = []
top = -1

# push function
def push(ele: str):
    global top
    top += 1
    stack[top] = ele

# pop function
def pop():
    global top
    ele = stack[top]
    top -= 1
    return ele

def isPalindrome(string: str) -> bool:
    global stack
    length = len(string)

    stack = ['0'] * (length + 1)
    mid = length // 2
    i = 0
    while i < mid:
        push(string[i])
        i += 1
    if length % 2 != 0:
        i += 1
     while i < length:
        ele = pop()
         if ele != string[i]:
            return False
        i += 1
    return True
if __name__ == "__main__":
    print ( " Palindrome ")
    print ( " ~~~~~~~~~~ ")
    string = input ("Enter the String :")
    if isPalindrome(string):
        print("The given string '",string,"' is a palindrome")
    else:
        print("The given string '",string,"'is not a palindrome")
```

OUTPUT:

Palindrome
~~~~~~~~~~
Enter the String :malayalam

The given string ' malayalam ' is a palindrome
~~~~~~~~~~

Prg.no: 16 **STATIONARY AND CONSUMER**

AIM: To create two tables for stationary and consumer and execute the given commands usingSQL.

TABLE:STATIONARY

S_ID	StationaryName	Company	Price
DP01	Dot Pen	ABC	10
PL02	Pencil	XYZ	6
ER05	Eraser	XYZ	7
PL01	Pencil	CAM	5
GP02	Gel Pen	ABC	15

TABLE: CONSUMER

C_ID	ConsumerName	Address	S_ID
01	Good Learner	Delhi	PL01
06	Write Well	Mumbai	GP02
12	Topper	Delhi	DP01
15	Write & Draw	Delhi	PL02
16	Motivation	Bangalore	PL01

i) To display the details of those Consumers whose Address is Delhi

ii) To display the details of Stationary whose Price is in the range of 8 to 15(Both values included)

iii) To display the ConsumerName , Address from table Consumer and Company and Price from table Stationery with their corresponding matching S_ID

iv) To increase the Price of all Stationary by 2.

v) To display distinct Company from STATIONARY .

```
CREATE TABLE STATIONARY (S_ID char(5)  NOT NULL PRIMARY KEY, StationaryName char(25), Company
char(5), Price int);

    INSERT INTO STATIONARY VALUES(“DP01” , “Dot Pen”, “ABC”, 10);

    INSERT INTO STATIONERY VALUES(“PL02” , “Pencil”, “XYZ”, 6)

   CREATE TABLE CONSUMER (C_ID int , ConsumerName char(25)Address char(25), S_ID char(5));

    INSERT INTO CONSUMER VALUES(01, “Good Learner”, “Delhi”, “PL01”);

    INSERT INTO CONSUMER VALUES(06,”Write Well”,”Mumbai”,”GP02”);
```

OUTPUT:

i) Select * from consumer where address=”delhi”;

c_id	Consumername	address	S_id
1	good learner	delhi	PL01
12	Topper	delhi	DP02
15	write & draw	delhi	PL02

ii) select * from stationary where price between 8 and 15;

S_id	stationary	company	Price
Dp01	dot pen	ABC	10
GP02	gel pen	ABC	15

iii) select consumername, address, company, price from stationery, consumer

where stationery.s_id=consumer.s_id;

consumername	Address	company	Price
good learner	Delhi	CAM	5
write well	Mumbai	ABC	15
Topper	Delhi	ABC	10
write&draw	Delhi	XYZ	6
motivation	Bangalore	CAM	5

iv) update stationery set price=price+2;

select * from stationery;

S_id	Stationary	company	Price
DP01	Dot pen	ABC	12
PL02	Pencil	XYZ	8
ER05	Eraser	XYZ	9
PL01	Pencil	CAM	7
GP02	Gel pen	ABC	17

v) select distinct(company) from stationery;

Company
ABC
XYZ
CAM

Prg.no: 17 **ITEM AND TRADERS**

AIM: To create two tables for item and traders and execute the given commands usingSQL.

TABLE:ITEM

Code	IName	Qty	Price	Company	TCode
1001	DIGITAL PAD 121	120	11000	XENTIA	T01
1006	LED SCREEN 40	70	38000	SANTORA	T02
1004	CAR GPS SYSTEM	50	2150	GEOKNOW	T01
1003	DIGITAL CAMERA 12X	160	8000	DIGICLICK	T02
1005	PEN DRIVE 32GB	600	1200	STOREHOME	T03

TABLE:TRADERS

TCode	TName	City
T01	ELECTRONICS SALES	MUMBAI
T03	BUSY STORE CORP	DELHI
T02	DISP HOUSE INC	CHENNAI

i) To display the details of all the items in ascending order of item names (i.e IName)

ii) To display item name and price of all those items, whose price is in the range of 10000 and 22000 (both values inclusive)

iii) To display the number of items , which are traded by each trader. The expected output of this query should be

T01 2
T02 2
T03 1

iv) To display the Price , item name(i.e IName) and quantity(i.e Qty) of those items which have quantity more than 150.

v) To display the names of those traders, who are either from DELHI or from MUMBAI.

CREATE TABLE ITEM(Code int , IName char(25) , Qty int , Price int , Company char(25), TCode char(5));

INSERT INTO ITEM VALUES(1001,”DIGITAL PAD 121”,120, 11000,”XENTIA”, “T01”);INSERT INTO ITEM VALUES(1006,”LED SCREEN 40”,70, 38000,”SANTORA”, “T02”);

CREATE TABLE TRADERS(TCode char(5) , TName char(25), City char(20));

INSERT INTO TRADERS VALUES(“T01”,”ELECTRONICS SALES”,”MUMBAI”);INSERT INTO TRADERS VALUES(“T03”,”BUSY STORE CORP”,”DELHI”);

OUTPUT:

i) select * from ITEM order by IName;

Code	IName	Qty	Price	Company	TCode
1004	CAR GPS SYSTEM	50	2150	GEOKNOW	T01
1003	DIGITAL CAMERA 12X	160	8000	DIGICLICK	T02
1001	DIGITAL PAD 121	120	11000	XENTIA	T01
1006	LED SCREEN	70	38000	SANTORA	T02
1005	PEN DRIVE 32GB	600	1200	STORE HOME	T03

ii) select IName , Price from ITEM where Price between 10000 and 22000;

IName	Price
DIGITAL PAD 121	11000

iii) select TCode , count(*) from ITEM group by TCode;

Tcode	Count(*)
T01	2
T02	2
T03	1

iv) select Price , IName , Qty from ITEM where Qty>150;

Price	IName	Qty
8000	DIGITAL CAMERA 12X	160
1200	PEN DRIVE 32GB	600

v) select TName from TRADERS where City in (“DELHI”,”MUMBAI”);

TName
ELECTRONICS SALES
BUSY STORE CORP

Prg.no: 18

DOCTOR AND SALARY

AIM: To create two tables for doctor and salary and execute the given commandsusing SQL.

TABLE:DOCTOR

ID	NAME	DEPT	SEX	EXPERIENCE
101	John	ENT	M	12
104	Smith	ORTHOPEDIC	M	5
107	George	CARDIOLOGY	M	10
114	Lara	SKIN	F	3
109	K George	MEDICINE	F	9
105	Johnson	ORTHOPEDIC	M	10
117	Lucy	ENT	F	3
111	Bill	MEDICINE	F	12
130	Morphy	ORTHOPEDIC	M	15

TABLE: SALARY

ID	BASIC	ALLOWANCE	CONSULTATION
101	12000	1000	300
104	23000	2300	500
107	32000	4000	500
114	12000	5200	100
109	42000	1700	200
105	18900	1690	300
130	21700	2600	300

i) Display NAME of all doctors who are in "MEDICINE" having more than 10 years experience from table DOCTOR

ii) Display the average salary of all doctors working in "ENT" department using the tables DOCTOR and SALARY. (Salary=BASIC+ALLOWANCE)

iii) Display minimum ALLOWANCE of female doctors.

iv) Display DOCTOR.ID , NAME from the table DOCTOR and BASIC , ALLOWANCE from the table SALARY with their corresponding matching ID.

v) To display distinct department from the table doctor.

```
CREATE TABLE DOCTOR(ID int NOT NULL PRIMARY KEY, NAME char(25) , DEPT char(25) , SEX
    char , EXPERIENCE int);

INSERT INTO DOCTOR VALUES(101,"John", "ENT",'M',12);

INSERT INTO DOCTOR VALUES(104,"Smith", "ORTHOPEDIC",'M',5);

CREATE TABLE SALARY(ID int, BASIC int, ALLOWANCE int, CONSULTATION int);

INSERT INTO SALARY VLAUES(101, 12000,1000,300);

INSERT INTO SALARY VLAUES(104, 23000,2300,500);
```

RESULT: The above program has been executed successfully.

OUTPUT:

i) select NAME from DOCTOR where DEPT=”MEDICINE” and EXPERIENCE >10;

NAME
Bill

ii) select avg(BASIC+ALLOWANCE) “avg salary” from DOCTOR , SALARYwhere

DOCTOR.ID=SALARY.ID and DEPT=”ENT”;

Avg salary
13000.00

iii) select min(ALLOWANCE) from SALARY, DOCTORwhere

SEX=’F’ and DOCTOR.ID=SALARY.ID;

min(ALLOWANCE)
1700

iv) select DOCTOR.ID, NAME, BASIC ,ALLOWANCE from DOCTOR,SALARYwhere DOCTOR.ID=SALARY.ID;

ID	NAME	BASIC	ALLOWANCE
101	John	12000	1000
104	Smith	23000	2300
107	George	32000	4000
109	K George	42000	1700
114	Lara	12000	5200
130	Morphy	21700	2600

v) select distinct(DEPT) from DOCTOR;

DEPT
ENT
ORTHOPEDIC
CARDIOLOGY
SKIN
MEDICINE

Prg.no: 19 **INTEGRATE PYTHON WITH SQL - FETCHING RECORDS FROM TABLE**

PROGRAM:

```
import mysql.connector as sqltor
mycon=sqltor.connect(host=”localhost”, user=”root”, password=”root”, databse=”trinity”)if
mycon.is_connected( ) = = False:
    print(“Error connecting to MySQL  database”)
cursor=mycon.cursor( )
cursor.execute(“select * from student”)
data=cursor.rowcount(3)
count=cursor.rowcount
for row in data:
     print(row)
mycon.close( )
```

OUTPUT:

(1001, “Vinusha”, 50,70, 80 , “Namakkal”)

(1001, “Aswin”, 54,82, 85 , “Erode”)

(1001, “Bheem”, 90,73, 78 , “Salem”)

Prg.no: 20 **INTEGRATE PYTHON WITH SQL - COUNTING RECORDS FROM TABLE**

PROGRAM:

```
import mysql.connector as sqltor
mycon=sqltor.connect(host=”localhost”, user=”root”, password=”root”, databse=”trinity”)if
mycon.is_connected( ) = = False:
    print(“Error connecting to MySQL  database”)
cursor=mycon.cursor( )
cursor.execute(“select * from student”)
data=cursor.fetchone( )
count=cursor.rowcount
print(“Total number of rows retrieved from resultset :”, count)
data=cursor.fetchone( )
count=cursor.rowcount
print(“Total number of rows retrieved from resultset :”, count)
data=cursor.fetchmany(3)
count=cursor.rowcount
print(“Total number of rows retrieved from resultset :”, count)
```

OUTPUT:

```
Total number of rows retrieved from resultset : 1
Total number of rows retrieved from resultset : 2
Total number of rows retrieved from resultset : 5
```

Prg.no: 21 INTEGRATE SQL WITH PYTHON - SEARCHING A RECORD FROM TABLE

PROGRAM:

```
import mysql.connector as mc
mycon=mc.connect(host='localhost',user='root',password='root1',data base='db12')
if mycon.is_connected( ):
   print("Py->Sql connected")
eno=int(input("Enter num:"))
mcursor=mycon.cursor( )
```

```
mcursor.execute("select * from emp")
allrow=mcursor.fetchall( )
for row in allrow:if
  row[0]==eno:
      print(row)
mycon.commit( )
mycon.close( )
```

OUTPUT:

```
Py-> sql is connected
Enter num : 103
 (103,'Cinu , 43, 'Namakkal')
```

Prg.no: 22 INTEGRATE SQL WITH PYTHON - DELETING A RECORD FROM TABLE

PROGRAM :

```
import mysql.connector as mc
mycon=mc.connect(host='localhost',user='root',password='root1',database='db12')
if mycon.is_connected():
    print("Py->Sql connected")
eno=int(input("Enter num:"))
mcursor=mycon.cursor()
mcursor.execute("select * from mp")
allrow=mcursor.fetchall()
for row in allrow:
      if row[0]==eno:
           mcursor.execute("delete from emp where eno={}".format(eno))
mcursor.execute("select * from emp")
print(mcursor.fetchall())
mycon.commit() mycon.close()
```

OUTPUT:

```
Py -> sql is connected
Enter num : 102
(101,'Anu',23,'Salem')
(103,'Cinu',43,'Namakkal')
(104, 'Nishanth', 46,'Chennai')
(105, 'Nanda', 56, 'Erode')
```