

# INDEX

S.No	Program
1	<i>Input and display matrix</i>
2	<i>Sum of rows and columns in matrix</i>
3	<i>Transpose matrix</i>
4	<i>Sum of diagonal elements of matrix</i>
5	<i>Displaying elements below and above major diagonal</i>
6	<i>Border elements of matrix</i>
7	<i>Sorting list in ascending order</i>
8	<i>Sum and average of elements in list</i>
9	<i>Swapping first and second half elements in list</i>
10	<i>Sum of elements having 2 in one's place</i>
11	<i>Class and teacher using dictionary</i>
12	<i>Search a contact using dictionary</i>
13	<i>Delete a contact using dictionary</i>
14	<i>Counting vowels and consonants in a string</i>
15	<i>Print words of a string in separate lines</i>
16	<i>Check whether a number is Armstrong</i>
17	<i>Fibonacci Series</i>
18	<i>Prime between 2 given numbers</i>
19	<i>Check whether a number is Palindrome</i>
20	<i>Reversing a string</i>
21	<i>Class Bus</i>
22	<i>Class Photo</i>
23	<i>Class City</i>
24	<i>Class Tourist</i>
25	<i>Class Candidate</i>
26	<i>Example of Inheritance</i>

27	<i>Multiple Inheritance</i>
28	<i>Insertion Sort</i>
29	<i>Appending element in a list using traditional method</i>
30	<i>Appending element using bisect method</i>
31	<i>Selection Sort</i>
32	<i>Bubble Sort</i>
33	<i>Deleting an element using traditional method</i>
34	<i>Deleting an element using remove method</i>
35	<i>Binary Search</i>
36	<i>Linear Search</i>
37	<i>List to Matrix</i>
38	<i>Count 2 alphabets in a file</i>
39	<i>Count 2 words in a file</i>
40	<i>Each character of a file in separate lines</i>
41	<i>Display file backwards</i>
42	<i>Stack using class</i>
43	<i>Stack without class</i>
44	<i>Queue without class</i>
45	<i>Queue using class</i>
46	<i>Append and insert a record in binary file Item.dat</i>
47	<i>Modify, search and delete a record in binary file Item.dat</i>
48	<i>Count uppercase and lowercase characters in a file</i>
49	<i>Count no. of lines starting with A, in a file</i>
50	<i>Copy of file, wherein copy has words starting with vowels</i>
51	<i>Raising ZeroDivisionError</i>
52	<i>User defined exception</i>
53	<i>Try-finally clause program</i>
54	<i>Check whether an integer is entered (Exception Handling)</i>

55	5 subject names displayed as tuple
56	Search a number in the list
57	Count words of a sentence, and find average length of a word
58	Check whether a number is magic number
59	Changing numbers that are multiples of 10 to 10, rest to 1
60	Copy of binary file sports.dat
61	First line, last line and total no. of lines in a file
62	Creating, displaying and appending a binary file
63	Creating object used for both reading and writing (file)
64	File to print certain roman numerals
65	Seek and tell
66	Delete a year from dictionary, displaying year and population
67	Class Society
68	Class STUDENT
69	Fibonacci series using generator function
70	Cubes of 1-5 using generator function
71	Doubling even numbers, tripling odd numbers
72	Reversing a list
73	Reversing a word in a string using generator function
74	Check whether the user entered correct time (Exception Handling)
75	Factorial of first 10 numbers using generator function
76	Prime numbers till a given number using generator function
77	Tuple of form $(2^n) - 1$ using generator function
78	$2^n$ using generator function
79	Generate 7 random numbers using generator function
80	Return True if string starts with a digit
81	Armstrong between 2 numbers
82	Check whether substring present in string (Meta search)

<i>83</i>	<i>Delete an element using pop()</i>
<i>84</i>	<i>Generator function to check whether file exists or not</i>
<i>85</i>	<i>Display employee name and ID in ascending order</i>
<i>86</i>	<i>Swap values of 2 tuples</i>
<i>87</i>	<i>Maximum and minimum value in tuple</i>
<i>88</i>	<i>Decimal to Binary</i>
<i>89</i>	<i>Binary to Decimal</i>
<i>90</i>	<i>Number raised to another number</i>
<i>91</i>	<i>SQL Program 1</i>
<i>92</i>	<i>SQL Program 2</i>
<i>93</i>	<i>SQL Program 3</i>
<i>94</i>	<i>SQL Program 4</i>
<i>95</i>	<i>SQL Program 5</i>
<i>96</i>	<i>SQL Program 6</i>
<i>97</i>	<i>SQL Program 7</i>
<i>98</i>	<i>SQL Program 8</i>
<i>99</i>	<i>SQL Program 9</i>
<i>100</i>	<i>SQL Program 10</i>

# Python Programs:

- 1) Input and display a matrix

```
#Program to input and display a matrix
def inputmatrix(a,m,n):
    for i in range(m):
        for j in range(n):
            a[i][j]=input()
def display(a,m,n):
    print "Displaying the matrix: "
    for i in range(m):
        for j in range(n):
            print a[i][j], '\t',
    print
import random
m=input("No. of rows: ")
n=input("No. of columns: ")
a=[[random.random() for col in range(n)] for row in range(m)]
print "Input",m*n,"Elements:"
inputmatrix(a,m,n)
display(a,m,n)
```

Output:

```
No. of rows: 3
No. of columns: 3
Input 9 Elements:
1
2
3
4
5
6
7
8
9
Displaying the matrix:
1      2      3
4      5      6
7      8      9
>>> |
```

- 2) Display sum of rows and columns in a matrix

```
import random
m=input ("No. of rows: ")
n=input ("No. of columns: ")

a=[[random.random() for col in range(n)] for row in range(m)]
```

```

print "Enter",m*n,"elements:"
for i in range(m):
    for j in range(n):
        a[i][j]=input()

print "The matrix is"
for i in range(m):
    for j in range(n):
        print a[i][j],'\t',
    print

for i in range(m):
    row_sum=0
    for j in range(n):
        row_sum+=a[i][j]
    print "Sum of row ",i+1," = ", row_sum

for j in range(n):
    col_sum=0
    for i in range(m):
        col_sum+=a[i][j]
    print "Sum of column ",j+1," = ", col_sum

```

Output:

```

No. of rows: 3
No. of columns: 3
Enter 9 elements:
1
3
5
6
7
8
9
2
4
The matrix is
1      3      5
6      7      8
9      2      4
Sum of row  1   =   9
Sum of row  2   =  21
Sum of row  3   =  15
Sum of column 1   =  16
Sum of column 2   =  12
Sum of column 3   =  17

```

### 3) Display matrix and its transpose

```
import random
m=input ("No. of rows: ")
n=input ("No. of columns: ")
a=[[random.random() for col in range(n)] for row in range(m)]
t=[[random.random() for col in range(m)] for row in range(n)]

print "Enter",m*n,"elements:"
for i in range(m):
    for j in range(n):
        a[i][j]=input()

print "The given matrix is"
for i in range(m):
    for j in range(n):
        print a[i][j],'\t',
    print
for i in range(n):
    for j in range(m):
        t[i][j]=a[j][i]
    print

print "\n The Tranpose Matrix is"
for i in range(n):
    for j in range(m):
        print t[i][j],'\t',
    print
```

Output:

```
No. of rows: 3
No. of columns: 3
Enter 9 elements:
1
2
3
4
5
6
7
8
9
The given matrix is
1      2      3
4      5      6
7      8      9
|
The Tranpose Matrix is
1      4      7
2      5      8
3      6      9
>>>
```

4) Display matrix and sum of diagonal elements

```
import random
m=input ("No. of rows: ")
n=input ("No. of columns: ")
a=[[random.random() for col in range(n)] for row in range(m)]

print "Input",m*n,"elements: "
for i in range(m):
    for j in range(n):
        a[i][j]=input()

print "Displaying matrix"
for i in range(m):
    for j in range(n):
        print a[i][j], '\t',
    print

print "Sum of diagonal elements:"
s=0
for i in range(m):
    for j in range(n):
        if i==j or i+j==m-1:
            s+=a[i][j]
print "Sum:",s
```

Output:

```
No. of rows: 3
No. of columns: 3
Input 9 elements:
2
3
4
5
6
7
8
7
6
Displaying matrix
2      3      4
5      6      7
8      7      6
Sum of diagonal elements:
Sum: 26
>>> |
```

5) Display Matrix, and elements below and above the major diagonal

```
import random
m=input ("No. of rows: ")
n=input ("No. of columns: ")
a=[[random.random() for col in range(n)] for row in range(m)]

print "Enter ",m*n, " elements:"
for i in range(m):
    for j in range(n):
        a[i][j]=input()

print "\n The given matrix is"
for i in range(m):
    for j in range(n):
        print a[i][j],'\t',
    print

print "Displaying elements below major diagonal:"
for i in range(m):
    for j in range(n):
        if i>j:
            print a[i][j], '\t',
    print

print "Displaying elements above major diagonal:"
for i in range(m):
    for j in range(n):
        if i<j:
            print a[i][j], '\t',
        else:
            print '\t',
    print
```

Output:

```
No. of rows: 3
No. of columns: 3
Enter 9 elements:
1
2
3
4
5
6
7
8
9
```

```

The given matrix is
1      2      3
4      5      6
7      8      9
Displaying elements below major diagonal:

4
7      8
Displaying elements above major diagonal:
2      3
      6

>>>

```

## 6) Displaying border elements of a matrix

---

```

#Display Border elements of the matrix

import random
m=input("No. of rows: ")
n=input("No. of columns: ")
a=[[random.random() for col in range(n)]for row in range(m)]
print "Input",m*n,"elements:"
for i in range(m):
    for j in range(n):
        a[i][j]=input()

print "Displaying the matrix..."
for i in range(m):
    for j in range(n):
        print a[i][j], '\t',
    print

print "Displaying border elements"
for i in range(m):
    for j in range(n):
        if i==0 or j==0 or i==m-1 or j==n-1:
            print a[i][j], '\t',
        else:
            print '\t',
    print

```

Output:

```
No. of rows: 3
No. of columns: 3
Input 9 elements:
1
2
3
4
5
6
7
8
9
Displaying the matrix...
1      2      3
4      5      6
7      8      9
Displaying border elements
1      2      3
4          6
7      8      9
>>> |
```

7) Sorting List in ascending order

```
n=input("Enter size of list: ")
a=range(n)
print "Enter",n,"numbers: "
for i in range(n):
    a[i]=input()

print "Displaying the original list: "
for i in range(n):
    print a[i],

for i in range(0,n-1,1):
    for j in range(i+1,n,1):
        if a[i]>a[j]:
            temp=a[i]
            a[i]=a[j]
            a[j]=temp

print "\nDisplaying the new list in ascending order : "
for i in range(n):
    print a[i],
```

Output:

```
Enter size of list: 5
Enter 5 numbers:
23
12
9
30
24
Displaying the original list:
23 12 9 30 24
Displaying the new list in ascending order :
9 12 23 24 30
>>> |
```

8) Sum and Average of elements in a list

```
#sum and average of numbers in a list
n=input("Size of list: ")
a=range(n)
print "Enter",n,"elements:"
for i in range(n):
    a[i]=input()

print "Displaying list: "
for i in range(n):
    print a[i],

print "\nDisplaying average of numbers"
s=0.0
for i in range(n):
    s+=a[i]
print "Sum:", s
print "Average:", s/n
```

Output:

```
Size of list: 5
Enter 5 elements:
2
4
5
6
7
Displaying list:
2 4 5 6 7
Displaying average of numbers
Sum: 24.0
Average: 4.8
>>> |
```

9) Swapping first half and second half elements in a List

```
n=input("Size of list: ")
a=range(n)
print "Enter",n,"numbers : "
for i in range(n):
    a[i]=input()

print "Displaying the original list: "
for i in range(n):
    print a[i], 

j=n/2
for i in range(0,n/2,1):
    a[i],a[j]=a[j],a[i]
    j=j+1

print "\nList after swapping first half and second half elements"
for i in range(n):
    print a[i],
```

Output:

```
Size of list: 6
Enter 6 numbers :
2
3
4
5
6
7
Displaying the original list:
2 3 4 5 6 7
List after swapping first half and second half elements
5 6 7 2 3 4
>>> |
```

10) Sum of elements in a list having 2 in its unit's place

```
def sum_num(a,size):
    s=0
    print "\nDisplaying the elements having 2 in units place : "
    for i in range(0,len(a)):
        if a[i] % 10==2:
            s=s+a[i]
            print a[i],
    print
    print "Sum of elements having 2 in units place ", s, "\n"
```

```

def main():
    n=input("Enter size: ")
    a=range(n)
    print "Input",n,"elements:"
    for i in range(n):
        a[i]=input()
    print "Displaying the list: "
    for i in range(n):
        print a[i],
        sum_num(a,n)
main()

```

Output:

```

Enter size: 5
Input 5 elements:
2
12
34
54
32
Displaying the list:
2 12 34 54 32
Displaying the elements having 2 in units place :
2 12 32
Sum of elements having 2 in units place  46
>>>

```

11) Input sections along with their respective class teacher using dictionary

```

classxi=dict()

n=input("Enter no. of Sections in Class XI:")

i=1

while i<=n:
    a=raw_input("Section:")

    b=raw_input ("Class Teacher:")
    classxi[a]=b
    i=i+1

print "Class","\t","Section","\t","Class Teacher"
for i in classxi:
    print "XI","\t",i,"\t\t",classxi[i]

```

Output:

```
Enter no. of Sections in Class XI:4
Section:A
Class Teacher:Mr. ABC
Section:B
Class Teacher:Mr. DEF
Section:C
Class Teacher:Mrs. PQR
Section:D
Class Teacher:Mrs. XYZ
Class Section          Class Teacher
XI    A                Mr. ABC
XI    C                Mrs. PQR
XI    B                Mr. DEF
XI    D                Mrs. XYZ
>>> |
```

12) Search a contact in a phonebook using dictionary

```
phonebook=dict()
n=input("No. of contacts: ")
i=1

while i<=n:
    a=raw_input("Name: ")
    b=raw_input("Phone Number: ")
    phonebook[a]=b
    i=i+1

print "\nPhonebook Information : "
print "\nName",'\t',"Phone Number"

l=phonebook.keys()
for i in l:
    print i,'\t',phonebook[i]
name=raw_input("Enter name for searching : ")
if phonebook.has_key(name):
    print " Telephone number found"
    print name, ":",phonebook[name]
else:
    print "Not found"
```

Output:

```
No. of contacts: 3
Name: Leo
Phone Number: 3930029
Name: Abhi
Phone Number: 2100323
Name: Mahrez
Phone Number: 3920312
```

```
Phonebook Information :
```

```
Phonebook Information :
```

```
Name      Phone Number
Mahrez   3920312
Abhi     2100323
Leo      3930029
Enter name for searching : Abhi
Telephone number found
Abhi : 2100323
>>> |
```

13) Delete a contact and display updated information (Phonebook) using dictionary

```
phonebook=dict()
n=input("Enter no. of contacts: ")
i=1
while i<=n:
    a=raw_input("Name: ")
    b=raw_input("Phone Number: ")
    phonebook[a]=b
    i=i+1
print "Phonebook Information : "
print "\n Name",'\t','Phone Number"
l=phonebook.keys()
for i in l:
    print i,' \t',phonebook[i]
name=raw_input("Enter name for deletion: ")
del phonebook[name]
l=phonebook.keys()
print "\n Phonebook Information after deletion : "
print "Name",'\t','Phone number"
for i in l:
    print i,' \t',phonebook[i]
```

Output:

```
Enter no. of contacts: 3
Name: ABC
Phone Number: 123456
Name: DEF
Phone Number: 234567
Name: GHI
Phone Number: 345678
Phonebook Information :

Name      Phone Number
ABC      123456
GHI      345678
DEF      234567
Enter name for deletion: DEF

Phonebook Information after deletion :
Name      Phone number
ABC      123456
GHI      345678
>>> |
```

14) Count number of vowels and consonants in a string

```
str=raw_input("Enter a string: ")
vow=con=0

for ch in str:
    if ch in 'AaEeIiOoUu':
        vow+=1
    else:
        if ch.isalpha():
            con+=1
print "Number of vowels =",vow
print "Number of consonants=",con
```

Output:

```
Enter a string: Messi will win Ballon Dor.
Number of vowels = 7
Number of consonants= 14
>>> |
```

15) Print words of a string in separate lines

```
import string
sentence=raw_input("Enter a sentence : ")
word_count=0

for w in sentence.split():
    print w
    word_count=word_count+1
print "\nThe number of words in the sentence = ", word_count
```

Output:

```
Enter a sentence : Dubai has horrible summers
Dubai
has
horrible
summers

The number of words in the sentence =  4
>>> |
```

16) Checking whether a number is Armstrong

```
n=input("Enter a number: ")
m=n
sm=0
while (n>0):
    digit=n%10
    sm=sm+digit**3
    n=n/10
print "Sum of cube of digits:",sm
if (m==sm):
    print "It is Armstrong number"
else:
    print"it is not Armstrong number"
```

Output:

```
Enter a number: 371
Sum of cube of digits: 371
It is Armstrong number
>>>

Enter a number: 234
Sum of cube of digits: 99
it is not Armstrong number
>>> |
```

### 17) Fibonacci Series

```
n=input("No. of terms of Fibonacci: ")
if n>2:
    "Fibonacci Series till",n,"terms:"
    a=0
    b=1
    print a
    print b
    for i in range(2,n):
        c=a+b
        print c
        a=b
        b=c
elif n==2:
    print "Fibonacci Series:", 0,1
else:
    print "Enter 2 or more no. of terms."
```

Output:

```
No. of terms of Fibonacci: 8
0
1
1
2
3
5
8
13
>>> |
```

### 18) Prime numbers between 2 given numbers

```
a=input("Enter lower limit of range: ")
b=input("Enter upper limit of range: ")
for i in range(a,b+1):
    prime=True
    for j in range(2,i):
        if i%j==0:
            prime=False
    if prime==True:
        print i,
```

Output:

```
Enter lower limit of range: 12
Enter upper limit of range: 22
13 17 19
>>> |
```

19) Check whether a number is Palindrome

```
n=input("Enter a number: ")
m=n
reverse=0
while(n>0):
    digit=n%10
    reverse=reverse*10+digit
    n=n/10
print "The Reversed Number : ",reverse
if (m==reverse):
    print "It is Palindrome"
else:
    print "It is not Palindrome"
```

Output:

```
Enter a number: 232
The Reversed Number :  232
It is Palindrome
>>>

Enter a number: 334
The Reversed Number :  433
It is not Palindrome
>>> |
```

20) Reversing a string

```
print """*30
str="Green Revolution"
print "The Original String : ", str
rev=""
for c in str:
    rev=c+rev

print "The reversed string using for loop : ", rev
print """*30
```

Output:

```
*****
The Original String :  Green Revolution
The reversed string using for loop :  noituloveR neerG
*****
>>> |
```

21) Define a class Bus in Python with the following specifications:

Data Members:

- \* Busno - to store Bus No.
- \* From - to store Place name of origin
- \* To - to store Place name of destination
- \* Type - to store Bus Type such as 'O' for Ordinary
- \* Distance - to store the Distance in Kilometers
- \* Fare - to store the bus fare

Member Functions:

- \* A Constructor to initialize Type as 'O' and Fare as 500
- \* A function CalcFare() to calculate Fare as per the following criteria:

Type	Fare
'O'	15*Distance
'E'	20*Distance
'L'	24*Distance

- \* A functions Allocate() allow user to enter value for Busno, From, To, Type and Distance. Also, this function should call CalcFare() to calculate Fare.
- \* A function Show() to display the content of all the data members on the screen.

```

class Bus:
    def __init__(self):
        self.Busno=0
        self.From=""
        self.To=""
        self.Type="O"
        self.Distance=0
        self.Fare=500
    def Calcfare(self):
        if self.Type=="O":
            self.Fare=15*self.Distance
        if self.Type=="E":
            self.Fare=20*self.Distance
        if self.Type=="L":
            self.Fare=24*self.Distance
    def Allocate(self):
        self.Busno=input("Enter Bus Number : ")
        self.From=raw_input("Enter Origin : ")
        self.To=raw_input("Enter Destinnnation : ")
        self.Type=raw_input("Enter Bus Type O/E/L : ")
        self.Distance=input(" Enter Distance : ")
        self.Calcfare()
    def Show(self):
        print "Bus No : ",self.Busno
        print "From : ",self.From
        print "To : ",self.To
        print "Type : ",self.Type
        print "Distance : ",self.Distance
        print "Fare : ",self.Fare
b=Bus()

b.Allocate()
b.Show()

```

Output:

```

Enter Bus Number : 32
Enter Origin : Karama
Enter Destination : Qusais
Enter Bus Type O/E/L : L
Enter Distance : 25
Bus No : 32
From : Karama
To : Qusais
Type : L
Distance : 25
Fare : 600
>>> |

```

22) Write a class PHOTO in Python with following specifications:

Instance Attributes

- Pno # Numeric value
- Category # String Value
- Exhibit # Exhibition Gallery with String value

Methods:

-FixExhibit() # A method to assign Exhibition Gallery as per Category

# as shown in the following table:

Category	Exhibit
Antique	Zaveri
Modern	Johnson
Classic	Terenida

-Register() # A function to allows user to enter values

# Pno, Category and call FixExhibit() method

-ViewAll() # A function to display all the data members.

```
class PHOTO:  
    def __init__(self,P,Cat, Ex):  
        self.Pno=P  
        self.Category=Cat  
        self.Exhibit=Ex  
  
    def FixExhibit(self):  
        if self.Category=="Antique":  
            self.Exhibit="Zaveri"  
        if self.Category=="Modern":  
            self.Exhibit="Johnsen"  
        if self.Category=="Classic":  
            self.Exhibit="Terenida"  
  
    def Register(self):  
        self.Pno=input(" Enter Photo Number : ")  
        self.Category=raw_input(" Enter PHOTO Category ( Antique/ Modern/ Classic ) : ")  
        self.FixExhibit()  
  
    def ViewAll(self):  
        print "Photo Number : ", self.Pno  
        print "PHOTO Category : ", self.Category  
        print "Exhibition Gallery : ", self.Exhibit  
  
s=PHOTO(001,"Antique","Zaveri")  
s.ViewAll()  
s.Register()  
s.ViewAll()
```

Output:

```
Photo Number : 1
PHOTO Category : Antique
Exhibition Gallery : Zaveri
Enter Photo Number : 2
Enter PHOTO Category ( Antique/ Modern/ Classic ) : Modern
Photo Number : 2
PHOTO Category : Modern
Exhibition Gallery : Johnsen
>>> |
```

23) Write a class CITY in Python with the following specifications.

Instance Attributes:

- Ccode # Numeric Value
- CName # String value
- Pop # Numeric value for Population
- KM # Numeric value
- Density # Numeric value for Population Density

Methods:

- DenCal() # Method to calculate Density as Pop/KM
- Record() # Method to allow user to enter values CCode, CName, Pop, KM and call  
DenCal() method
- View() # Method to display all the members also display a message  
“Highly Populated City” if the Density is more than 10000.

```
class CITY:
    def __init__(self):
        self.Ccode=0
        self.CName=""
        self.Pop=0
        self.KM=0
        self.Density=0
    def DenCal(self):
        self.Density=self.Pop/self.KM
    def Record(self):
        self.Ccode=input("Enter City code : ")
        self.CName=raw_input("Enter City Name : ")
        self.Pop=input("Enter City Population : ")
        self.KM=input("Enter area (KM) : ")
        self.DenCal()
```

```

def View(self):
    print "City Code : ",self.Ccode
    print "Enter City Name : ",self.CName
    print "City Population : ", self.Pop
    print "KM : ", self.KM
    print "Density : ", self.Density
    if self.Density>10000:
        print "Highly Populated "
    else:
        print "Not highly populated "

c=CITY()
c.Record()
c.View()

```

Output:

```

Enter City code : 10001
Enter City Name : New York
Enter City Population : 12500000
Enter area (KM) : 6000
City Code : 10001
Enter City Name : New York
City Population : 12500000
KM : 6000
Density : 2083
Not highly populated
>>> |

```

24) Define a class Tourist in Python with the following specification:

Data Members:

- \* CNo - to store Cab No
- \* CType - to store a character 'A', 'B', or 'C' as City Type
- \* PerKM – to store per Kilometer charges
- \* Distance – to store Distance travelled ( in KM)

Member Functions

- A constructor function to initialize CType as 'A' and CNo as '0000'
- A function CityCharges() to assign PerKM as per the following table

CType PerKM

A 20

B 18

C 15

- A function RegisterCab() to allows administrator to enter the values for CNo and CType. Also this function should call CityCharges() to assign PerKM Charges
- A function Display() to allows user to enter the value of Distance and display CNo, CType, PerKM, PerKM\*Distance (as Amount) on the screen.

```

class Tourist:
    def __init__(self):
        self.CType="A"
        self.Cno="0000"
        self.PerKM=0.0
        self.Distance=0.0
    def CityCharges(self):
        if self.CType=="A":
            self.PerKM=20
        if self.CType=="B":
            self.PerKM=18
        if self.CType=="C":
            self.PerKM=15
    def RegisterCab(self):
        self.CType=raw_input("Enter city type A/B/C : ")
        self.Cno=raw_input("Enter cab no. : ")
        self.CityCharges()
    def Display(self):
        self.Distance= input("Enter distance : ")
        print " Cab No. : ",self.Cno
        print " City Type : ",self.CType
        print " Per KM rate : ", self.PerKM
        print "Amount = ", self.PerKM*self.Distance
tour=Tourist()
tour.RegisterCab()
tour.Display()

```

Output:

```

Enter city type A/B/C : B
Enter cab no. : 2931
Enter distance : 50
Cab No. : 2931
City Type : B
Per KM rate : 18
Amount = 900
>>> |

Enter city type A/B/C : A
Enter cab no. : 2033
Enter distance : 35
Cab No. : 2033
City Type : A
Per KM rate : 20
Amount = 700
>>> |

```

25) Define a class Candidate in Python with following description:

#### Private Members

- \* A data member RNo (Registration Number) of type long
- \* A data member Name of type string
- \* A data member Score of type float
- \* A data member Remarks of type string
- \* A member function AssignRem() to assign Remarks as per the Score obtained by a candidate. Score range and the respective Remarks are shown as follows:

Score	Remarks
<hr/>	
>=50	Selected
less than 50	Not selected

#### Public Members

- \* A function ENTER() to allow user to enter values for RNo, Name, Score & call function AssignRem() to assign the remarks.
- \* A function DISPLAY () to allow user to view the content of all the data members.

```
class Candidate:  
    def __init__(self,RNo,Name,Score,Remark):  
        self.RNo=RNo  
        self.Name=Name  
        self.Score=Score  
        self.Remark=Remark  
    def AssignRem(self):  
        if self.Score>=50:  
            self.Remark="Selected"  
        if self.Score<50:  
            self.Remark="Not Selected"  
    def ENTER(self):  
        self.RNo=input(" Enter Registration Number : ")  
        self.Name=raw_input(" Enter Candidate Name : ")  
        self.Score=float(raw_input(" Enter Score : "))  
        self.AssignRem()  
    def DISPLAY(self):  
        print "Registration Number : ", self.RNo  
        print "Candidate Name : ", self.Name  
        print "Score : ", self.Score  
        print "Remark : ", self.Remark
```

```
s=Candidate(11,"Jas",90,"Very Good")
s.DISPLAY()
s.ENTER()
s.DISPLAY()
```

Output:

```
Registration Number : 11
Candidate Name : Jas
Score : 90
Remark : Very Good
Enter Registration Number : 12
Enter Candidate Name : Kev
Enter Score : 93
Registration Number : 12
Candidate Name : Kev
Score : 93.0
Remark : Selected
>>> |
```

## 26) Example of Inheritance

```
class person(object):
    def __init__(self,n,a):
        self.name=n
        self.age=a
    def getName(self):
        return self.name
    def getAge(self):
        return self.age

class student(person):
    def __init__(self,n, a, r, m):
        super(student,self).__init__(n, a)
        self.rollno=r
        self.marks=m
    def getRoll(self):
        return self.rollno
    def getMarks(self):
        return self.marks

p=person("Ram",15)
print "Person's name ",p.getName()
print "Age of person = ",p.getAge()

s=student("Sita",15, 1, 99)

print "Student names ",s.getName()
print "Student age ",s.getRoll()

print "Students marks ",s.getMarks()
```

Output:

```
Person's name Ram
Age of person = 15
Student names: Sita
Student age: 1
Students marks: 99
>>> |
```

## 27) Multiple Inheritance Example

```
class student(object):
    def __init__(self,Id,name):
        self.Id=Id
        self.name=name
    def getName(self):
        return self.name
    def getId(self):
        return self.Id
    def show(self):
        print self.name
        print self.Id
class Teacher(object):
    def __init__(self,tec_Id,tec_name, subject):
        self.tec_Id=tec_Id
        self.tec_name=tec_name
        self.subject=subject
    def getName(self):
        return self.tec_name
    def getId(self):
        return self.tec_Id
    def getSubject(self):
        return self.subject
    def show(self):
        print self.tec_name
        print self.tec_Id
        print self.subject

class school(student,Teacher): #class school inherits class student and teacher
    def __init__(self, Id, name, tec_Id, tec_name, subject, sch_Id):
        student.__init__(self,Id,name)
        Teacher.__init__(self,tec_Id,tec_name, subject)
        self.sch_Id= sch_Id
    def getId(self):
        return self.sch_Id
    def display(self):
        print "Student Id: ",self.Id
        print "Student Name: ",self.name
        print "Teacher Id : ",self.tec_Id
        print "Teacher : ",self.tec_name
        print "Subject : ",self.subject
        print "School Id : ",self.sch_Id

s1=school(3,"Sham",56,"Ram","FIT",530)
s1.display()
s1.getId()
print "Show "
s1.show()
```

Output:

```
Student Id: 3
Student Name: Sham
Teacher Id : 56
Teacher : Ram
Subject : FIT
School Id : 530
Show
Sham
3
>>> |
```

## 28) Insertion Sort

```
def insertion(L):

    for i in range(len(L)):
        V=L[i]
        j=i
        while L[j-1]>V and j>=1: #shifting the elements
            L[j]=L[j-1]
            j-=1
        L[j]=V
    print'After sorting - ', L

def main():
    n=input('Enter the no. of elements: ')
    L=range(n)
    for i in range(n):
        L[i]=input('Enter element :')
    print 'Before sorting - ',L
    insertion(L)

main()
```

Output:

```
Enter the no. of elements: 4
Enter element :12
Enter element :9
Enter element :30
Enter element :32
Before sorting - [12, 9, 30, 32]
After sorting - [9, 12, 30, 32]
```

## 29) Appending element in a list (Traditional)

```
import string
def insert(l,item):
    n=len(l)
    if item<l[0]:
        p=0
    else:
        for i in range(n-1): #finding position
            if l[i]<=item and l[i+1]>item:
                p=i+1
                break
    if item>l[n-1]:
        p=n
    l.append(None) #Extra space at end
    i=n
    while i>p: #shifting
        l[i]=l[i-1]
        i=i-1
    l[p]=item #inserting element

def main():

    n=input('Enter the no. of elements : ')
    l=range(n)
    for i in range(n):
        l[i]=input('Enter list element : ')

    ch='Y'
    while ch.upper()=='Y':
        item=input('Enter the number to inserted :')
        insert(l, item)
        print 'List after inserting element -' , l
        ch= raw_input("continue?")
main()
```

Output:

```
Enter the no. of elements : 4
Enter list element : 4
Enter list element : 6
Enter list element : 7
Enter list element : 8
Enter the number to inserted :9
List after inserting element - [4, 6, 7, 8, 9]
continue?Y
Enter the number to inserted :5
List after inserting element - [4, 5, 6, 7, 8, 9]
continue?N
>>> |
```

### 30) Inserting element in a list using Bisect method

```
import bisect
n=input('Enter the no. of elements : ')
l=range(n)
for i in range(n):
    l[i]=input('Enter element : ')
ch='Y'
while ch.upper()=='Y':
    item=input('Enter the number to be inserted : ')
    p=bisect.bisect(l,item) #get position
    print 'Element will be inserted at position: ',p+1
    bisect.insort(l,item)   #inserts at proper place
    print 'List after inserting element : ', l
    ch= raw_input("Continue?")
```

Output:

```
Enter the no. of elements : 3
Enter element : 10
Enter element : 12
Enter element : 14
Enter the number to be inserted : 13
Element will be inserted at position: 3
List after inserting element : [10, 12, 13, 14]
Continue?Y
Enter the number to be inserted : 11
Element will be inserted at position: 2
List after inserting element : [10, 11, 12, 13, 14]
Continue?N
>>> |
```

### 31) Selection Sort

```
def selection_sort(L):
    for i in range(len(L)):
        min_pos=i
        for j in range(i+1,len(L)):
            if L[j]<L[min_pos]:
                min_pos=j
        temp=L[i]
        L[i]=L[min_pos]
        L[min_pos]=temp
        print L #to see each step of sorting
print 'After sorting...'
print L
def main():
    n=input('Enter no. of elements: ')
    L=range(n)
    for i in range(n):
        L[i]=input('Enter element: ')
    print 'Before sorting...'
    print L
    selection_sort(L)
main()
```

Output:

```
Enter no. of elements:4
Enter element: 12
Enter element: 3
Enter element: 5
Enter element: 7
Before sorting...
[12, 3, 5, 7]
[3, 12, 5, 7]
[3, 5, 12, 7]
[3, 5, 7, 12]
[3, 5, 7, 12]
After sorting...
[3, 5, 7, 12]
>>> |
```

### 32) Bubble Sort

```
def bubble_sort(L,n):
    for i in range (n):
        for j in range(0,n-1-i):
            if L[j]>L[j+1]:
                temp=L[j]
                L[j]=L[j+1]
                L[j+1]=temp
                print L #to see each step of sorting

def main():
    n=input('enter no. of elements: ')
    L=range(n)
    for i in range(n):
        L[i]=input('enter element: ')
    print 'before sorting...',L
    bubble_sort(L,n)
    print 'After sorting...', L
main()
```

Output:

```
enter no. of elements:4
enter element: 9
enter element: 3
enter element: 1
enter element: 4
before sorting... [9, 3, 1, 4]
[3, 9, 1, 4]
[3, 1, 9, 4]
[3, 1, 4, 9]
[1, 3, 4, 9]
After sorting... [1, 3, 4, 9]
>>> |
```

33) Deletion of an element in a list using traditional method

```
def delete(l,item):
    #finding position using linear search
    n=len(l)
    for i in range(n):
        if l[i]==item :
            pos=i
            break
    else:
        print "Element not found"
        exit(0)
    #shifting elements
    i=pos
    while i<n-1:
        l[i]=l[i+1]
        i=i+1
    l[i]=None
    n=n-1    # Reduce the size of list by 1 after deletion
print 'List after deleting element : '
for i in range(n):
    print l[i]

def main():
    n=input('Enter the no. of elements -')
    l=range(n)
    for i in range(n):
        l[i]=input('Enter element-')
    ch='Y'
    while ch.upper()=='Y':
        item=input('Enter the item for deletion : ')
        print 'List before deleting element : ', l
        delete(l, item)
        ch= raw_input("Continue?")
main()
```

Output:

```
Enter the no. of elements -4
Enter element-3
Enter element-5
Enter element-6
Enter element-7
Enter the item for deletion : 6
List before deleting element : [3, 5, 6, 7]
List after deleting element :
3 5 7 Continue?N
>>> |
```

### 34) Deletion of an element using remove

```
n=input('Enter no. of elements -')
l=range(n)
for i in range(n):
    l[i]=input('Enter an element:')

print 'Current List contents ',l
item=input('Enter the item to be deleted : ')
if item in l:
    l.remove(item)
    print 'After deletion',l
else:
    print 'Item not found'
```

Output:

```
Enter no. of elements -4
Enter an element:2
Enter an element:3
Enter an element:4
Enter an element:5
Current List contents  [2, 3, 4, 5]
Enter the item to be deleted : 3
After deletion [2, 4, 5]
>>> |
```

### 35) Binary Search

```
def binary_search(L, item):
    start=0
    end=len(L)-1
    f=0
    while start<=end:
        mid = (start+end)/2
        if item==L[mid]:
            return(mid)
            f=1
        elif item>L[mid]:
            start =mid+1
        else:
            end=mid-1
    return(-1)

def main():
    n=input('Enter no. of items : ')
    L=range(n)
    for i in range(n):
        L[i]=input('Enter element: ')
    item=input('Enter the item to search : ')
    p=binary_search(L,item)
    if p==-1:
        print 'Element not found'
    else:
        print 'Element found at ',p+1
main()
```

Output:

```
Enter no. of items : 4
Enter element: 2
Enter element: 3
Enter element: 4
Enter element: 5
Enter the item to search : 3
Element found at 2
>>> |
```

36) Linear Search

```
def linear_search(L, item):
    for i in range(len(L)):
        if L[i]==item:
            print " Element found at ",i +1
            break
    else:
        print "Element not found"

def main():
    n=input('Enter no. of items-')
    L=range(n)
    for i in range(n):
        L[i]=input('Enter element: ')
    item=input('Enter the item to search : ')
    linear_search(L,item)
main()
```

Output:

```
Enter no. of items-4
Enter element: 2
Enter element: 3
Enter element: 4
Enter element: 5
Enter the item to search : 4
Element found at 3
>>> |
```

37) Function which accepts a list and its size as arguments and assigns the elements in a matrix of integers...

If a list= 1,2,3,4,5,6

Resultant Matrix: 1 2 3 4 5 6

```
    1 2 3 4 5 0
    1 2 3 4 0 0
    1 2 3 0 0 0
    1 2 0 0 0 0
    1 0 0 0 0 0
```

```

def List_to_Matrix(A,n):
    import random
    mat=[[random.random() for col in range(n)] for row in range(n)]
    for i in range(n):
        for j in range (n):
            if (j<n-i):
                mat[i][j] = A[j]
            else:
                mat[i][j] = 0
    print "\n\n The given list : "
    for i in range(n):
        print A[i],|
    print "\n\n The matrix is"
    for i in range(n):
        for j in range(n):
            print mat[i][j],'\t',
        print

def main():
    print "Creating a list A having n elements "
    n= input("Enter a number (size of the list): ")
    A=range(n)      # Creeates a list A of size n
    print "\nInput", n, "elements   "

    for i in range(n):
        A[i]=input()
    List_to_Matrix(A,n)
main()

```

Output:

```

Creating a list A having n elements
Enter a number (size of the list): 4

Input 4 elements
1
2
3
4

The given list :
1 2 3 4

The matrix is
1      2      3      4
1      2      3      0
1      2      0      0
1      0      0      0
>>> |

```

38) Menu driven program to create, display, and count 2 alphabets in a file

```
def create():
    ofile = open("story.txt", "w+")
    choice = True
    while True:
        line = raw_input("Enter sentence :")
        ofile.write(line)
        ofile.write('\n')
        choice = raw_input("Want to enter more data in file Y / N")
        if choice.upper() == 'N' : break
    ofile.close()

def display():
    fo = open("story.txt", "r")
    print "\n Displaying the text file contents using readlines() "
    lines = fo.readlines()
    print lines
    print type(lines)
    fo.close()
    for l in lines:
        print l

def count_alphabets():
    fo = open("story.txt", "r")
    print "\n Displaying each character of the the text file in separate lines  "
    lines = fo.readlines()
    alpha1=raw_input("Enter first alphabet to be counted : ")
    alpha2=raw_input("Enter second alphabet to be counted : ")
    fo.close()
    count1=0
    count2=0
    for l in lines:
        for w in l.split():
            for c in w:
                if c.upper()==alpha1.upper():
                    count1=count1+1
                if c.upper()==alpha2.upper():
                    count2=count2+1
    print alpha1," is repeated ",count1," times"
    print alpha2," is repeated ",count2," times"

def main():

    while True:
        print "\n\n\n"
        print "===== MAIN MENU ====="
        print "1. Create Text File "
        print "2. Display the file "
        print "3. Count two alphabets given by the user"
        print "4. Exit "
        ch = input('Enter your choice :')
        if ch==1:
            print "Creating the text file "
            create()
        elif ch==2:
            print "Displaying the file "
```

```
        elif ch==3:  
            print "Counting two alphabets in the file : "  
            count_alpha()  
        elif ch==4:  
            exit()  
        else:  
            print' Wrong choice entered'  
main()
```

Output:

```
=====  
          MAIN MENU  
=====  
1. Create Text File  
2. Display the file  
3. Count two alphabets given by the user  
4. Exit  
Enter your choice :1  
Creating the text file  
Enter sentence :Messi is the greatest  
Want to enter more data in file Y / NY  
Enter sentence :Sharjah is a horrible place  
Want to enter more data in file Y / NN
```

```
=====  
          MAIN MENU  
=====  
1. Create Text File  
2. Display the file  
3. Count two alphabets given by the user  
4. Exit  
Enter your choice :2  
Displaying the file  
  
Displaying the text file contents using readlines()  
['Messi is the greatest\n', 'Sharjah is a horrible place\n']  
<type 'list'>  
Messi is the greatest  
  
Sharjah is a horrible place
```

```
=====
      MAIN MENU
=====
1. Create Text File
2. Display the file
3. Count two alphabets given by the user
4. Exit
Enter your choice :3
Counting two alphabets in the file :

Displaying each character of the the text file in separate lines
Enter first alphabet to be counted : e
Enter second alphabet to be counted : f
e is repeated 6 times
f is repeated 0 times
```

```
=====
      MAIN MENU
=====
1. Create Text File
2. Display the file
3. Count two alphabets given by the user
4. Exit
Enter your choice :4
>>> |
```

39) Program to create, display and count 2 words in a file

```
def create():
    ofile = open("story.txt", "w+")
    choice = True
    while True:
        line = raw_input("enter sentence :")
        ofile.write(line)
        ofile.write('\n')
        choice = raw_input("want to enter more data in file Y / N")
        if choice.upper() == 'N' : break
    ofile.close()
def display():
    fo = open("story.txt", "r")
    print "\n Displaying the text file contents using readlines() "
    lines = fo.readlines()
    fo.close()
    for l in lines:
        print l
```

```
def count_word():
    fo = open("story.txt", "r")
    print "Counting 2 words in the file"
    lines = fo.readlines()
    word1=raw_input("Enter first word to be counted : ")
    word2=raw_input("Enter second word to be counted : ")
    fo.close()
    count1=0
    count2=0
    for l in lines:
        for w in l.split():
            if w.upper()==word1.upper():
                count1=count1+1
            if w.upper()==word2.upper():
                count2=count2+1
    print word1," is repeated ",count1," times"
    print word2," is repeated ",count2," times"

def main():

    while True:
        print "\n\n\n"
        print "===== MAIN MENU ====="
        print "1. Create Text File "
        print "2. Display the file "
        print "3. Counting two words given by the user "
        print "4. Exit "
        ch = input('Enter your choice :')
        if ch==1:
            print "Creating the text file "
            create()
        elif ch==2:
            print "Displaying the file "
            display()
        elif ch==3:
            print "Counting two words in the file : "
            count_word()

        elif ch==4:
            exit()
        else:
            print' Wrong choice entered'
main()
```

Output:

```
=====
      MAIN MENU
=====
1. Create Text File
2. Display the file
3. Counting two words given by the user
4. Exit
Enter your choice :1
Creating the text file
enter sentence :when he goes to bed, he makes sure everything is in order
want to enter more data in file Y / Ny
enter sentence :he gets up in the morning
want to enter more data in file Y / Nn
```

```
=====
      MAIN MENU
=====
1. Create Text File
2. Display the file
3. Counting two words given by the user
4. Exit
Enter your choice :2
Displaying the file

Displaying the text file contents using readlines()
when he goes to bed, he makes sure everything is in order

he gets up in the morning
```

```
=====
      MAIN MENU
=====
1. Create Text File
2. Display the file
3. Counting two words given by the user
4. Exit
Enter your choice :3
Counting two words in the file :
Counting 2 words in the file
Enter first word to be counted : in
Enter second word to be counted : he
in is repeated 2 times
he is repeated 3 times
```

40) Display each character of file in separate lines

```
def fileCreation():
    ofile = open("story.txt", "w+")
    choice = True
    while True:
        line = raw_input("Enter sentence:")
        ofile.write(line)
        ofile.write('\n')
        choice = raw_input("More? Y / N")
        if choice.upper() == 'N' : break
    ofile.close()
def display_text_file():
    fo = open("story.txt", "r")
    print "\nDisplaying the text file contents using readlines() "
    lines = fo.readlines()
    fo.close()
    for l in lines:
        print l

def display_each_char_in_separate_lines():
    fo = open("story.txt", "r")
    print "\nDisplaying each character of the the text file in separate lines"
    lines = fo.readlines()
    fo.close()
    for l in lines:
        for w in l.split():
            for ch in w:
                print ch

print "Creating the text file "
fileCreation()
print "Displaying the file "
display_text_file()
display_each_char_in_separate_lines()
```

Output:

```
Creating the text file
Enter sentence:Lionel Messi
More? Y / NN
Displaying the file
\nDisplaying the text file contents using readlines()
Lionel Messi
```

```
Displaying each character of the the text file in separate lines
L
i
o
n
e
l
M
e
s
s
i
>>> |
```

41) Display contents of file backwards

```
def fileCreation():
    ofile = open("story.txt", "w+")
    choice = True
    while True:
        line = raw_input("Enter sentence :")
        ofile.write(line)
        ofile.write('\n')
        choice = raw_input("More? Y / N")
        if choice.upper() == 'N' : break
    ofile.close()

def display_text_file():
    fo = open("story.txt", "r")
    print "\n Displaying the text file contents using readlines() "
    lines = fo.readlines()
    fo.close()
    for l in lines:
        print l

def read_display_file_backwards():
    fo = open("story.txt", "r")
    print "\n Displaying the text file contents backwards "
    fo.seek(0,2)
    pos=fo.tell()
    print "The number of characters in file = ", pos-2
    i=3
    while i<=pos:
        fo.seek(-i,2)
        str=fo.read(1)
        print str,
        i=i+1
    print "Creating the text file "
    fileCreation()
    print "Displaying the file "
    display_text_file()
    read_display_file_backwards()
```

Output:

```
Creating the text file
Enter sentence :Lionel Messi
More? Y / NN
Displaying the file

Displaying the text file contents using readlines()
Lionel Messi

Displaying the text file contents backwards
The number of characters in file = 12
i s s e M   l e n o i L
>>> |
```

#### 42) Stack using class

```
class stack:
    s=[]
    def push(self):
        a=input("Enter any number :")
        stack.s.append(a)
    def display(self):
        l=len(stack.s)
        if l==0:
            print "Stack is empty "
        else:
            for i in range(l-1,-1,-1):
                print stack.s[i],
a=stack()
c="y"
while (c=="y"):
    print "1. PUSH"
    print "2. POP "
    print "3. Display"
    print "4. Exit "
    choice=input("Enter your choice: ")
    if (choice==1):
        a.push()
    elif (choice==2):
        if (a.s==[]):
            print "Stack Empty"
        else:
            print "Deleted element is : ",a.s.pop()
    elif (choice==3):
        a.display()
    elif choice==4:
        print "Exiting the program."
    else:
        print("Wrong Input")
    c=raw_input("Do you want to continue or not? (y/n) ")
```

Output:

```
1. PUSH
2. POP
3. Display
4. Exit
Enter your choice: 1
Enter any number :2
Do you want to continue or not? (y/n) y
1. PUSH
2. POP
3. Display
4. Exit
Enter your choice: 1
Enter any number :3
Do you want to continue or not? (y/n) y
1. PUSH
2. POP
3. Display
4. Exit
Enter your choice: 1
Enter any number :4
Do you want to continue or not? (y/n) y
1. PUSH
2. POP
3. Display
4. Exit
Enter your choice: 3
4 3 2 Do you want to continue or not? (y/n) y
1. PUSH
2. POP
3. Display
4. Exit
Enter your choice: 2
Deleted element is : 4
Do you want to continue or not? (y/n) n
```

#### 43) Stack without class

```
s=[]
def push():
    a=input("Enter a number: ")
    s.append(a)
def display():
    l=len(s)
    if l==0:
        print "Empty stack"
    else:
        for i in range(l-1,-1,-1):
            print s[i],
```

```
while True:
    print "\n1. Push"
    print "2. Pop"
    print "3. Display"
    print "4. Exit"
    ch=input("Choose: ")
    if ch==1:
        push()
    elif ch==2:
        if s==[]:
            print "Empty stack"
        else:
            print "Element deleted:",s.pop()
    elif ch==3:
        display()
    elif ch==4:
        exit()
    else:
        print "Wrong choice"
```

Output:

```
1. Push
2. Pop
3. Display
4. Exit
Choose: 1
Enter a number: 1
```

```
1. Push
2. Pop
3. Display
4. Exit
Choose: 1
Enter a number: 2
```

```
1. Push
2. Pop
3. Display
4. Exit
Choose: 1
Enter a number: 3
```

```
1. Push
2. Pop
3. Display
4. Exit
Choose: 3
3 2 1
1. Push
2. Pop
3. Display
4. Exit
Choose: 2
Element deleted: 3
```

```
1. Push  
2. Pop  
3. Display  
4. Exit  
Choose: 3  
2 1
```

#### 44) Queue without class

```
a=[]  
c='y'  
while c=='y':  
    print "1. INSERT"  
    print "2. DELETE "  
    print "3. Display"  
    ch=input("Enter your choice: ")  
    if (ch==1):  
        b=input("Enter new number ")  
        a.append(b)  
    elif (ch==2):  
        if (a==[]):  
            print("Queue Empty")  
        else:  
            print "Deleted element is:",a[0]  
            del a[0]  
    elif (ch==3):  
        l=len(a)  
        if l==0:  
            print "Queue is Empty "  
        else:  
            for i in range(0,l):  
                print a[i],  
            print|  
    else:  
        print("Wrong input")  
c=raw_input("Do you want to continue or not(y/n): ")
```

Output:

```
1. INSERT  
2. DELETE  
3. Display  
Enter your choice: 1  
Enter new number 1  
Do you want to continue or not(y/n): y  
1. INSERT  
2. DELETE  
3. Display  
Enter your choice: 1  
Enter new number 2  
Do you want to continue or not(y/n): y  
1. INSERT  
2. DELETE  
3. Display  
Enter your choice: 1  
Enter new number 3  
Do you want to continue or not(y/n): y
```

```

1. INSERT
2. DELETE
3. Display
Enter your choice: 3
1 2 3
Do you want to continue or not(y/n): y
1. INSERT
2. DELETE
3. Display
Enter your choice: 2
Deleted element is: 1
Do you want to continue or not(y/n): y
1. INSERT
2. DELETE
3. Display
Enter your choice: 3
2 3
Do you want to continue or not(y/n): n

```

#### 45) Queue using class

```

class queue:
    q=[]
    def insertion(self):
        a=input("Enter any number: ")
        queue.q.append(a)
    def deletion(self):
        if (queue.q==[]):
            print "Queue Empty"
        else:
            print "Deleted element is: ",queue.q[0]
            del queue.q[0]
    def display(self):
        l=len(queue.q)
        if l==0:
            print "Queue Empty "
        else:
            for i in range(0,l):
                print queue.q[i],
            print|
a=queue()
c="y"
while (c=="y"):
    print "1. INSERTION"
    print "2. DELETION "
    print "3. DISPLAY"
    choice=input("Enter your choice: ")
    if (choice==1):
        a.insertion()
    elif (choice==2):
        a.deletion()
    elif (choice==3):
        a.display()
    else:
        print("Wrong input")
    c=raw_input("Do you want to continue or not (y/n):")

```

Output:

```
1. INSERTION
2. DELETION
3. DISPLAY
Enter your choice: 1
Enter any number: 1
Do you want to continue or not (y/n):y
1. INSERTION
2. DELETION
3. DISPLAY
Enter your choice: 1
Enter any number: 2
Do you want to continue or not (y/n):y
1. INSERTION
2. DELETION
3. DISPLAY
Enter your choice: 3
1 2
Do you want to continue or not (y/n):y
1. INSERTION
2. DELETION
3. DISPLAY
Enter your choice: 2
Deleted element is: 1
Do you want to continue or not (y/n):y
1. INSERTION
2. DELETION
3. DISPLAY
Enter your choice: 3
2
Do you want to continue or not (y/n):n
>>> |
```

46) Given class:

```
class Item:
    def __init__(self):
        self.lno=0
        self.name=""
        self.Quantity=""
        self.Rate=""
        self.Amount=0
    def storedata(self):
        self.lno=input("Enter Item Number : ")
        self.name=raw_input('Enter name: ')
        self.Quantity= raw_input('Enter Quantity: ')
        self.Rate=raw_input("Enter Rate : ")
        self.Amount=input("Enter Amount : ")
```

```
def display(self):
    print self.Ino,"\\t\\t",self.name, "\\t\\t", self.Quantity,"\\t\\t",self.Rate,"\\t\\t",self.Amount
```

Function to create, append and insert a record in the binary file Item.dat

```
import pickle,os
def create():
    f=open("Item.dat","wb")

    n=input("Enter the no. of Items :")
    s=Item()
    for i in range(n):
        s.storedata()
        pickle.dump(s,f)
    f.close()

def append():
    f=open("Item.dat","ab")

    n=input("Enter the no. of Items :")
    s=Item()
    for i in range(n):
        s.storedata()
        pickle.dump(s,f)
    f.close()
def insert():

    f1=open("Item.dat","rb")
    f2=open("newfile.dat","wb")

    s1=Item()
    s1.storedata()
    end=0
    try:
        while True:
            s= pickle.load(f1)
            if s.name<=s1.name:
                pickle.dump(s,f2)
            else :
                end=1
                break

        pickle.dump(s1,f2)
        f1.seek(0)
        while True:
            s= pickle.load(f1)
            if s.name>s1.name:
                pickle.dump(s,f2)

    except EOFError:
        if end==0:
            pickle.dump(s1,f2)
    f1.close()
    f2.close()

os.remove("Item.dat")
os.rename("newfile.dat","Item.dat")
```

47) Using same class Item, write functions to modify, search and delete a record in file Item.dat

```
def delete():
    Ino=input("Enter the Item Number for deletion: ")

    f1=open("Item.dat","rb")
    f2=open("newfile.dat","wb")
    status=0
    try:
        while True:
            s= pickle.load(f1)
            if s.Ino!=Ino:
                pickle.dump(s,f2)
            else :
                status=1
    except EOFError:
        f1.close()
        f2.close()
    os.remove("Item.dat")
    os.rename("newfile.dat","Item.dat")
    if status==1:
        print 'record deleted'
    else:
        print 'record not found'

def modify():
    Ino=input("Enter the Item Number whose details need to be modified : ")
    f1=open("Item.dat","rb")
    f2=open("newfile.dat","wb")
    s=Item()
    status=0

    try:
        while True:
            s= pickle.load(f1)
            if s.Ino==Ino:
                status=1
                s.storedata()
            pickle.dump(s,f2)

    except EOFError:
        f1.close()
        f2.close()
    os.remove("Item.dat")
    os.rename("newfile.dat","Item.dat")
    if status ==1:
        print 'file modified'
    else:
        print 'record not found '
```

```

def search():
    Ino=input("Enter the Item Number whose details to be searched : ")
    f1=open("Item.dat","rb")
    status=0
    try:
        while True:
            s= pickle.load(f1)
            if s.Ino==Ino:
                status=1
                print "Item Number : ",s.Ino
                print 'Item name : ', s.name
                print 'Quantity : ',s.Quantity
                print 'Rate : ', s.Rate
                print "Amount : ", s.Amount

    except EOFError:
        f1.close()
    if status==0:
        print 'Record not found'

```

48) Count uppercase and lowercase characters in a text file

```

def create():
    f=open("text.txt","w+")
    ch=True
    while True:
        line=raw_input("Enter something: ")
        f.write(line)
        f.write('\n')
        ch=input("More? 1 for Yes:")
        if ch!=1: break
    f.close()
def display():
    f=open("text.txt","r")
    print "Displaying file."
    lines=f.readlines()
    f.close()
    for l in lines:
        print l
def upplow():
    f=open("text.txt","r")
    print "Counting upper and lowercase characters"
    lines=f.readlines()
    f.close()
    up=lo=0
    for l in lines:
        for i in range(len(l)-1):
            if l[i].isupper():
                up+=1
            elif l[i].islower():
                lo+=1
    print "Uppercase characters:",up
    print "Lowercase characters:",lo
create()
display()
upplow()

```

Output:

```
Enter something: i love you
More? 1 for Yes:1
Enter something: YOU LOVE ME
More? 1 for Yes:2
Displaying file.
i love you

YOU LOVE ME

Counting upper and lowercase characters
Uppercase characters: 9
Lowercase characters: 8
>>>
```

49) Count no. of lines in a text file starting with A

```
def create():
    f=open("text.txt","w+")
    ch=True
    while True:
        line=raw_input("Enter something: ")
        f.write(line)
        f.write('\n')
        ch=input("More? 1 for Yes:")
        if ch!=1: break
    f.close()
def display():
    f=open("text.txt","r")
    print "Displaying file."
    lines=f.readlines()
    f.close()
    for l in lines:
        print l
def countA():
    f=open("text.txt","r")
    print "Displaying lines starting with A/a"
    lines=f.readlines()
    f.close()
    count=0
    for l in lines:
        if l[0]=='a' or l[0]=='A':
            count+=1
    print "No. of lines starting with A/a:",count

create()
display()
countA()
```

Output:

```
Enter something: already
More? 1 for Yes:1
Enter something: all gone
More? 1 for Yes:1
Enter something: elelele
More? 1 for Yes:2
Displaying file.
already

all gone

elelele

Displaying lines starting with A/a
No. of lines starting with A/a: 2
>>> |
```

50) Create a copy of a text file, wherein the copy has words starting with vowels

```
def create():
    ofile = open("story.txt", "w+")
    choice = True
    while True:
        line = raw_input("enter sentence :")
        ofile.write(line)
        choice = raw_input("want to enter more data in file Y / N")
        if choice.upper() == 'N' : break
    ofile.close()

def Copy_vowels():
    ifile = open("story.txt", "r")
    #ifile.seek(0)
    ofile = open("newstory.txt", "w")
    l=ifile.read()
    s=l.split(' ')
    for i in s:
        if i[0]=='a' or i[0]=='e' or i[0]=='i' or i[0]=='o' or i[0]=='u':
            ofile.write(i)
            ofile.write(" ")
    ifile.close()
    ofile.close()
def display_original_file():
    for l in open("story.txt", "r").readlines():
        print l,

def display_copy_file():
    for l in open("newstory.txt", "r").readlines():
        print l
```

```

def main():

    while True:
        print "\n"
        print "1. Create Text File "
        print "2. Display the file "
        print "3. Create copy with the words starting with vowels "
        print "4. Display new file "
        print "5. Exit "
        ch = input('Enter your choice :')
        if ch==1:
            print "Creating Original Text File "
            create()
        elif ch==2:
            print "Displaying the Original File"
            display_original_file()
        elif ch==3:
            print "Making Copy of the Text File"
            Copy_vowels()
        elif ch==4:
            print "Displaying the Copy of the Text File with words starting whith vowels"
            display_copy_file()
        elif ch==5:
            exit()
        else:
            print' Wrong choice entered'
main()

```

Output:

```

1. Create Text File
2. Display the file
3. Create copy with the words starting with vowels
4. Display new file
5. Exit
Enter your choice :1
Creating Original Text File
enter sentence :i love you
want to enter more data in file Y / Ny
enter sentence :messi is the best ever
want to enter more data in file Y / Nn
1. Create Text File
2. Display the file
3. Create copy with the words starting with vowels
4. Display new file
5. Exit
Enter your choice :2
Displaying the Original File
i love youmessi is the best ever 1. Create Text File
2. Display the file
3. Create copy with the words starting with vowels
4. Display new file
5. Exit
Enter your choice :3
Making Copy of the Text File
1. Create Text File
2. Display the file
3. Create copy with the words starting with vowels
4. Display new file
5. Exit
Enter your choice :4
Displaying the Copy of the Text File with words starting whith vowels
i is ever

```

51) Program to raise Zero Division Error

```
try:  
    a=int(raw_input("enter numerator:"))  
    b=int(raw_input("enter denominator:"))  
    if b==0:  
        raise ZeroDivisionError,str(a)+" /0 not possible "  
    print "Quotient= ",a/b  
except ZeroDivisionError,e:  
    print "Exception:",e.message
```

Output:

```
enter numerator:6  
enter denominator:0  
Exception: 6 /0 not possible  
>>> |
```

52) Program to raise user-defined exception

```
class myError(Exception):  
    pass  
MyError1=myError("Negative Marking not allowed ")  
MyError2=myError("Maximum marks are 100")  
  
try:  
    marks=input("Enter marks : ")  
    if marks<0:  
        raise MyError1  
    elif marks>100:  
        raise MyError2  
    elif marks>40:  
        print "Promoted to next class "  
    else:  
        print "Your re-examination is scheduled"  
except myError,e:  
    print e.message, " Marks entered : ", marks
```

Output:

```
Enter marks : 44  
Promoted to next class  
>>>  
  
Enter marks : 23  
Your re-examination is scheduled  
>>>  
  
Enter marks : 200  
Maximum marks are 100  Marks entered :  200  
>>>  
  
Enter marks : -200  
Negative Marking not allowed  Marks entered :  -200  
>>> |
```

53) A try-finally clause program where the inverse of a number x has to be found out.

```
try:
    x = float(raw_input("Your number: "))
    inverse = 1.0 / x
except ValueError:
    print "You should have given either an int or a float"
except ZeroDivisionError:
    print " Division by Zero, Infinity"

finally:
    print "There may or may not have been an exception."
    print "The inverse: ", inverse
```

Output:

```
Your number: -3.4
There may or may not have been an exception.
The inverse: -0.294117647059
>>>
RESTART: C:\Users\DELL\Desktop\KF\PYTHON STUFF\Grade 12 Python and SQL Programs\Exception Handling\Try_Finally_Clause.py
Your number: 0
Division by Zero, Infinity
There may or may not have been an exception.
The inverse:

Traceback (most recent call last):
  File "C:\Users\DELL\Desktop\KF\PYTHON STUFF\Grade 12 Python and SQL Programs\Exception Handling\Try_Finally_Clause.py", line 20, in <module>
    print "The inverse: ", inverse
NameError: name 'inverse' is not defined
>>> |
```

54) Program to check whether an integer is entered. If not, display a message that it is not an integer.

```
ok= False
while not ok:
    try:
        numStr=int(raw_input("Enter an integer : "))
        ok=True
        print "It is an integer"
    except:
        print "Error ! Not a valid integer."
```

Output:

```
- Enter an integer : 3
It is an integer
>>>

Enter an integer : -2.3
Error ! Not a valid integer.
```

55) Input 5 subject names and display them as a tuple

```
t=tuple()
print " Enter 5 subjects one after other : ";
for i in range(5):
    a=raw_input("Enter subject : ")
    t=t+(a,)

print "\n\n Output is :"
print t
```

Output:

```
Enter 5 subjects one after other :
Enter subject : math
Enter subject : physics
Enter subject : chem
Enter subject : computer science
Enter subject : bio

Output is :
('math', 'physics', 'chem ', 'computer science', 'bio')
>>> |
```

56) Find a number in a list entered by a user

```
n=input("No. of elements: ")
a=range(n)
print "Enter",n,"elements for the list: "
for i in range(n):
    a[i]=input()
print "Displaying the list: "
for i in range(n):
    print a[i]

s=input("Enter a number: ")
for i in range(n):
    if s==a[i]:
        print "The number is found", "at position:",i+1
    else:
        continue
```

Output:

```
No. of elements: 5
Enter 5 elements for the list:
3
4
5
6
7
Displaying the list:
3
4
5
6
7
Enter a number: 5
The number is found at position: 3
>>>
```

57) Count the number of words of a sentence in a file, and also display average length of a word

```
def filecreate():
    ofile=open("sent.txt","w+")
    line=raw_input("Enter a sentence: ")
    ofile.write(line)
    ofile.close()

def display():
    ofile=open("sent.txt","r")
    print "\nDisplaying text file: "
    lines=ofile.readlines()
    ofile.close()
    for l in lines:
        print l

def wordcount_avglength():
    ofile=open("sent.txt","r")
    print "Counting words in the sentence..."
    lines=ofile.readlines()
    ofile.close()
    count=1
    for l in lines:
        for i in l:
            if i.isspace()==True:
                count+=1
    print "No. of words:",count
    ofile=open("sent.txt","r")
    ca=0
    lines=ofile.readlines()
    ofile.close()
    for l in lines:
        for i in l:
            if i.isspace()==False:
                ca+=1
    avg=ca/(count*1.0)
    print "Average length of words:",avg
```

```
def main():
    ch=0
    while ch<=4:
        print "-"*20,"MAIN MENU","-*20
        print "1. Create and display file"
        print "2. Count no. of words, and display avg length of words"
        print "3. Exit"
        print "-*49
        ch=input("Enter choice: ")
        if ch==1:
            filecreate()
            display()
        elif ch==2:
            wordcount_avglength()
        elif ch==3:
            exit()
        else:
            print "Invalid choice!"
main()
```

Output:

```
----- MAIN MENU -----
1. Create and display file
2. Count no. of words, and display avg length of words
3. Exit
-----
Enter choice: 1
Enter a sentence: You are a nice friend

Displaying text file:
You are a nice friend
----- MAIN MENU -----
1. Create and display file
2. Count no. of words, and display avg length of words
3. Exit
-----
Enter choice: 2
Counting words in the sentence...
No. of words: 5
Average length of words: 3.4
----- MAIN MENU -----
1. Create and display file
2. Count no. of words, and display avg length of words
3. Exit
-----
Enter choice: 3
>>> |
```

58) Check whether a given number is a magic number

```
''' Magic number is a number wherein the sum of the digits gives a number such
that the number obtained has its sum of digits 1'''

n=input("Enter a number: ")
sum=0
num=n
while num>9:
    sum=num
    s=0
    while sum!=0:
        s=s+(sum%10)
        sum=sum/10
        num=s
    if num==1:
        print "It is a magic number"
    else:
        print "It is not a magic number."
```

Output:

```
Enter a number: 2341
It is a magic number
>>>

Enter a number: 2342
It is not a magic number.
>>> |
```

59) Input a list, and change numbers that are multiples of 10, to 10, and the rest to 1.

```
def input_dis_list(P,N):
    print "Input",N,"elements:"
    for i in range(N):
        P[i]=input()
    print "\nDisplaying the list: "
    for i in range(N):
        print P[i],
    print

def change(P,N):
    print "Changing elements that are multiples of 10, to 10 \
and changing the rest to 1"
    for i in range(N):
        if P[i]%10==0:
            P[i]=10
        else:
            P[i]=1
    for i in range(N):
        print P[i],
    print
```

```

N=input("Enter no. of elements: ")
P=range(N)
input_dis_list(P,N)
change(P,N)

Output:

Enter no. of elements: 4
Input 4 elements:
20
12
11
10

Displaying the list:
20 12 11 10
Changing elements that are multiples of 10, to 10 and changing the rest to 1
10 1 1 10
>>> |

```

60) Given a binary file SPORTS.DAT, containing records of the following

structure type :

class Sports:

    Event

    Participant

Write a function in Python that would read contents from the file SPORTS.DAT

and creates a file named ATHLETIC.DAT copying only those records from SPORTS.DAT where the event name is 'Athletics'.

```

import pickle
import os

class Sports:
    def __init__(self):
        self.Event=""
        self.Participant=""
    def storedata(self):
        self.Event=raw_input ('Enter Sports Event: ')
        self.Participant= raw_input('Enter Participant : ')
    def display(self):
        print self.Event, '\t\t', self.Participant
def create():
    f=open("Sports.dat","wb")
    n=input("Enter the no. of Events :")
    s=Sports()
    for i in range(n):
        s.storedata()
        pickle.dump(s,f)
    f.close()

```

```

def Copy_Sports_Athletics():
    f1=open("Sports.dat","rb")
    f2=open("Athletics.dat","wb")
    s=Sports()
    Found=0

    try:
        while True:
            s= pickle.load(f1)
            if s.Event=='Athletics':
                Found=1
                pickle.dump(s,f2)

    except EOFError:
        f1.close()
        f2.close()

    if Found:
        print "Athletics copied "
    else:
        print "Athletics not found "

def display_sports():

    print 'Event\tParticipant '
    s=Sports()
    f=open("Sports.dat","rb")

    try:
        while True:
            s= pickle.load(f)
            s.display()
    except EOFError:
        f.close()

def display_athletics():

    print 'Event\tParticipant '
    s=Sports()
    f=open("Athletics.dat","rb")

    try:
        while True:
            s= pickle.load(f)
            s.display()
    except EOFError:
        f.close()

```

```

def main():
    while True:
        print '1. Create file'
        print '2. Modify record'
        print '3. Display Sports file details'
        print '4. Display Athletics File file '
        print '5-Exit\n'
        ch = input('Enter your choice ::')
        if ch==1:
            create()
        elif ch==2:
            Copy_Sports_Athletics()
        elif ch==3:
            display_sports()
        elif ch==4:
            display_athletics()
        elif ch==5:
            exit()
        else:
            print' Wrong choice entered'
main()

```

Output:

```

1. Create file
2. Modify record
3. Display Sports file details
4. Display Athletics File file
5-Exit

```

```

Enter your choice :1
Enter the no. of Events :3
Enter Sports Event: Athletics
Enter Participant : ABC
Enter Sports Event: Football
Enter Participant : DEF
Enter Sports Event: Rugby
Enter Participant : PQR
1. Create file
2. Modify record
3. Display Sports file details
4. Display Athletics File file
5-Exit

```

```

Enter your choice :2
Athletics copied
1. Create file
2. Modify record
3. Display Sports file details
4. Display Athletics File file
5-Exit

```

```

Enter your choice :3
Event          Participant
Athletics      ABC
Football       DEF
Rugby         PQR

```

```
1. Create file  
2. Modify record  
3. Display Sports file details  
4. Display Athletics File file  
5-Exit
```

```
Enter your choice :4  
Event           Participant  
Athletics       ABC  
1. Create file  
2. Modify record  
3. Display Sports file details  
4. Display Athletics File file  
5-Exit
```

```
Enter your choice :5  
>>> |
```

61) Print first line, last line and total no. of lines in a file

```
def fileCreation():  
    ofile = open("data.txt", "w+")  
    choice = True  
    while True:  
        line = raw_input("enter sentence :")  
        ofile.write(line)  
        ofile.write('\n')  
        choice = raw_input("want to enter more data in file Y / N")  
        if choice.upper() == 'N' : break  
    ofile.close()  
def display_text_file1():  
    fo = open("data.txt", "r")  
    print "\n Displaying the source file contents using readlines() "  
    lines = fo.readlines()  
    fo.close()  
    for l in lines:  
        print l  
def first_last_line_and_total_lines():  
    fin = open("data.txt", "r")  
    lineList = fin.readlines()  
    fin.close()  
    print "First Line = ", lineList[0]  
    print "Last line = ", lineList[-1]  
    print "Total lines = ", len(lineList)  
fileCreation()  
display_text_file1()  
first_last_line_and_total_lines()
```

Output:

```
enter sentence :today is hot
want to enter more data in file Y / NY
enter sentence :tomorrow is also hot
want to enter more data in file Y / NY
enter sentence :exams are a headache
want to enter more data in file Y / Ny
enter sentence :i have a headache
want to enter more data in file Y / Ny
enter sentence :so i am an exam :3
want to enter more data in file Y / Nn

    Displaying the source file contents using readlines()
today is hot

tomorrow is also hot

exams are a headache

i have a headache

so i am an exam :3

First Line = today is hot

Last line = so i am an exam :3

Total lines = 5
>>> |
```

## 62) Appending data in a binary file

```
import pickle

def CreateBinFile():

    file = open('data.dat','wb')
    while True:
        x = raw_input("Enter the integers for the Binary File : ")
        pickle.dump(x,file)
        ans = raw_input('want to enter more data Y / N')
        if ans.upper()=='N' : break
    file.close()

def insertAtPosition():
    file = open('data.dat','ab')
    while True:
        y = raw_input("Enter Integer Data for appending in binary file ")
        pickle.dump(y,file)
        ans = raw_input('want to enter more data Y / N')
        if ans.upper()=='N' : break
    file.close()
```

```

def copyFile():
    ofile = open('copydata.dat','wb')
    ifile = open('data.dat','rb')
    try:
        while True:
            y=pickle.load(ifile)
            pickle.dump(y,ofile)
    except EOFError:
        pass
    ifile.close()
    ofile.close()
def displayFile1():
    file = open('data.dat','rb')
    try :
        while True :
            y = pickle.load(file)
            print y,
    except EOFError :
        pass
    file.close()
def displayFile2():
    file = open('copydata.dat','rb')
    try :
        while True :
            y = pickle.load(file)
            print y,
    except EOFError :
        pass
    file.close()
print "\nCreating Original Binary File : \n";
CreateBinFile()
print "\nDisplaying the Original Binary File  \n";
displayFile1()
print "\nAppending data to the Binary File "
insertAtPosition()
copyFile()
print "\nDisplaying the Binary File after appending data  \n";
displayFile2()

```

Output:

Creating Original Binary File :

Enter the integers for the Binary File : 4  
want to enter more data Y / Ny  
Enter the integers for the Binary File : 5  
want to enter more data Y / Ny  
Enter the integers for the Binary File : 6  
want to enter more data Y / Ny  
Enter the integers for the Binary File : 7  
want to enter more data Y / Nn

Displaying the Original Binary File

```
4 5 6 7
Appending data to the Binary File
Enter Integer Data for appending in binary file 8
want to enter more data Y / Ny
Enter Integer Data for appending in binary file 9
want to enter more data Y / Nn
```

```
Displaying the Binary File after appending data
```

```
4 5 6 7 8 9
>>>
```

### 63) Creating an object both for writing and reading (File)

```
def fileHandling():
    print "\n Creating text File with multiple lines : "
    file = open("story.txt","w+") # both reading & writing can be done
    choice = True
    while True:
        line = raw_input("Enter sentence : ")
        file.write(line) # creation of file
        choice = raw_input("Want to enter more data in file Y / N : ")
        if choice.upper() == 'N' : break
    print "\n Displaying the contents of the Text File : \n"
    file.seek(0) # transferring file object to beginning of the file
    lines = file.readlines()
    file.close()
    for l in lines:
        print l

fileHandling()
```

Output:

```
Creating text File with multiple lines :
Enter sentence : today is a warm day
Want to enter more data in file Y / N : y
Enter sentence : so won't be going out
Want to enter more data in file Y / N : n
```

```
Displaying the contents of the Text File :
```

```
today is a warm dayso won't be going out
>>> |
```

#### 64) File to display certain roman numerals

```
import pickle
numerals={1:"I",4:"IV",5:"V",9:"IX",10:"X",40:"XL",50:"L",90:"XC",100:"C",400:"CD",500:"D",900:"CM",1000:"M"}
file1=open("roman.log","wb")
pickle.dump(numerals,file1)

file1.close()

file2 = open ("roman.log","rb")
num = pickle.load(file2)
file2.close

n = 0

while n != -1:
    print "Enter 1 / 4 / 5 / 9 / 10 / 40 / 50 / 90 / 100 / 400 / 500 / 900 / 1000"
    print "...Enter -1 to exit..."
    n = int(raw_input("Enter the number :"))
    if n != -1:
        print "Equivalent roman number of this numeral is :", num[n]
    else :
        print "Thank You"
```

Output:

```
Enter 1 / 4 / 5 / 9 / 10 / 40 / 50 / 90 / 100 / 400 / 500 / 900 / 1000
...Enter -1 to exit...
Enter the number :9
Equivalent roman number of this numeral is : IX
Enter 1 / 4 / 5 / 9 / 10 / 40 / 50 / 90 / 100 / 400 / 500 / 900 / 1000
...Enter -1 to exit...
Enter the number :40
Equivalent roman number of this numeral is : XL
Enter 1 / 4 / 5 / 9 / 10 / 40 / 50 / 90 / 100 / 400 / 500 / 900 / 1000
...Enter -1 to exit...
Enter the number :-1
Thank You
>>> |
```

#### 65) Seek and tell program

```
TextFile = open("assessemnt3.txt", "r+")
str = TextFile.read(10)
print str
pos = TextFile.tell()
print pos
position = TextFile.seek(0, 0)
str = TextFile.read(20)
print str
TextFile.close()
```

Output:

```
The progra
10
The program writes t
>>>
```

66) Code to input no. of years, and enter population. Then, delete a year from the dictionary.

```
population=dict()

n=input("Enter total number of years : ")
i=1

while i<=n:
    a=input("Enter year: ")
    b=raw_input("Enter population: ")
    population[a]=b
    i=i+1
print "\n Population Information : "
print "\n Year",'\t',"City"
l=population.keys()
print l
for i in l:
    print i,' \t',population[i]

year=input("Enter year for deletion : ")
del population[year]
l=population.keys()
print "\n Population Information after deletion : "
print "\n Year",'\t',"City"
for i in l:
    print i,' \t',population[i]
```

Output:

```
Enter total number of years : 4
Enter year: 2003
Enter population: 40000
Enter year: 2004
Enter population: 50000
Enter year: 2007
Enter population: 90000
Enter year: 2012
Enter population: 125000
```

```
Population Information :
```

```
Year      City
[2012, 2003, 2004, 2007]
2012     125000
2003     40000
2004     50000
2007     90000
Enter year for deletion : 2004
```

```
Population Information after deletion :
```

```
Year      City
2012     125000
2003     40000
2007     90000
>>> |
```

67) Create a class SOCIETY with following info:

Data Members:

society\_name

house\_no

no\_of\_members

flat

income

Methods:

\* An `__init__` method to assign initial values of society\_name as "Surya Apartments", flat as "A Type", house\_no as 20, no\_of\_members as 3, income as 25000.

\* Inputdata() - to read data members (society,house\_no,no\_of\_members&income) and call `allocate_flat()`.

\* `allocate_flat()` - To allocate flat according to income

Income	Flat
-----	-----
>=25000	A Type
>=20000 and <25000	B Type
<20000	C Type

\* Showdata() - to display the details of the entire class.

```
class society:
    def __init__(self):
        self.society_name='Surya Apartments'
        self.house_no=20
        self.no_of_members=3
        self.income=25000
        self.flat='A Type'
    def input_data(self):
        self.society_name=raw_input("Enter Society Name: ")
        self.house_no=input('Enter House Number: ')
        self.no_of_members=input('Enter No. of members: ')
        self.income=input('Enter income: ')
        self.allocate_flat()
    def allocate_flat(self):
        if self.income<20000:
            self.flat='C'
        elif self.income>=20000 and self.income<25000:
            self.flat='B'
        elif self.income>=25000:
            self.flat='A'
    def showdata(self):
        print self.flat, "\t\t", self.house_no, "\t\t", self.no_of_members, "\t\t", self.society_name
```

```

n=input('Enter the number of users : ')
l=range(n)
for i in range(n):
    print 'User:',i+1
    l[i]=society()
    l[i].input_data()
print "Flat Type \t House No \t Members \t Society"
for i in range(n):
    l[i].showdata()
    print

```

Output:

```

Enter the number of users : 2
User: 1
Enter Society Name: ABC
Enter House Number: 10
Enter No. of members: 4
Enter income: 20000
User: 2
Enter Society Name: XYZ
Enter House Number: 33
Enter No. of members: 3
Enter income: 30000
Flat Type      House No      Members      Society
B              10            4            ABC
A              33            3            XYZ

```

>>> |

- 68) Enter a class STUDENT, and enter n number of users, display the result, and also display it in ascending order

```

class Student:
    def __init__(self):
        self.rno=0
        self.name=''
        self.marks=0
    def input_data(self):

        self.rno=input('Enter Roll Number : ')
        self.name=raw_input("Enter Name : ")
        self.marks=input('Average Marks : ')

    def showdata(self):
        print "{0:<15} {1:<35} {2:>10.2f}".format(self.rno, self.name, self.marks)

n=input('Enter the number of students: ')

```

```

l=range(n)
print "Enter details of ",n," students "
for i in range(n):

    l[i]=Student()
    l[i].input_data()

print "\t\t Student Details "
print "{0:<15} {1:<35} {2:>10}".format("Roll No.", "Name", "Avg. Marks")
print '-'*80
for i in range(n):
    l[i].showdata()
print '-'*80

for i in range(n-1):
    for j in range(i+1,n,1):
        if l[i].marks>l[j].marks:
            l[i],l[j]=l[j],l[i]

print "\t\t Student Details with marks in ascending order "
print "{0:<15} {1:<35} {2:>10}".format("Roll No.", "Name", "Avg. Marks")
print '-'*80
for i in range(n):
    l[i].showdata()
print '-'*80

```

Output:

```

Enter the number of students: 2
Enter details of  2  students
Enter Roll Number : 1
Enter Name : ABC
Average Marks : 95
Enter Roll Number : 2
Enter Name : DEF
Average Marks : 85
                Student Details
    Roll No.      Name           Avg. Marks
-----
1              ABC             95.00
2              DEF             85.00
-----
                Student Details with marks in ascending order
    Roll No.      Name           Avg. Marks
-----
2              DEF             85.00
1              ABC             95.00
-----
>>>

```

69) Fibonacci series using generator function

```
def fibo(max):
    a,b=0,1
    while a< max:
        yield a
        a,b= b, a+b
un=raw_input("Fibonacci values upto number:")
n=int(un)
for i in fibo(n):
    print i,
```

Output:

```
Fibonacci values upto number:200
0 1 1 2 3 5 8 13 21 34 55 89 144
>>>
```

70) Printing cubes of numbers 1-5 using generator function

```
def cubic_number(n):
    for i in range(1,n+1):
        yield i**3

print " Printing cube of numbers 1-5 : "
for a in cubic_number(5):
    print a,
```

Output:

```
Printing cube of numbers 1-5 :
1 8 27 64 125
>>> |
```

71) Double even numbers, and triple odd numbers in a list

```
def Even_Double_Odd_Thrice(A, N):
    for i in range(N):
        if A[i]% 2==0:
            A[i]=A[i]*2
        else:
            A[i]=A[i]*3

    print
    print "List after doubling even and tripling odd numbers..."
    for j in range(N):
        print A[j],

def main():
    print "Creating a list A having n elements "
    n= input("Enter size of list: ")
    A=range(n)
    print "\n input", n, "elements: "

    for i in range(n):
        A[i]=input()

    print "\nDisplaying the list elements  "
    for i in range (n):
        print A[i],

    Even_Double_Odd_Thrice(A, n)
main()
```

Output:

```
Creating a list A having n elements
Enter size of list: 4

    input 4 elements:
2
3
4
5

Displaying the list elements
2 3 4 5
List after doubling even and tripling odd numbers...
4 9 8 15
>>> |
```

## 72) Reversing a list

```
def Reversing(A, N):
    j=N-1
    for i in range(N/2):
        A[i],A[j]=A[j],A[i]
        j=j-1

    print
    print "The new list after reversing list elements : "
    for j in range(N):
        print A[j],

def main():
    print "Creating a list A having n elements "
    n= input("Enter size of list: ")
    A=range(n)
    print "Input",n,"elements:"

    for i in range(n):
        A[i]=input()

    print "\nDisplaying the list elements..."
    for i in range (n):
        print A[i],

    Reversing(A,n)

main()
```

Output:

```
Creating a list A having n elements
Enter size of list: 4
Input 4 elements:
2
4
6
8

Displaying the list elements...
2 4 6 8
The new list after reversing list elements :
8 6 4 2
>>>
```

73) Reversing a word in string using generator function

```
def reverse(data):
    for index in range(len(data)-1, -1, -1):
        yield data[index]

def main():
    print " Calling the generator reverse() to print the reversed string golf : "
    for char in reverse('golf'):
        print char

main()
```

Output:

```
Calling the generator reverse() to print the reversed string golf :
f
l
o
g
>>>
```

74) Program to check whether the user has entered the correct time or not (Exception Handling)

```
class TimeError(Exception):
    pass
HourError=TimeError("Hours should be between 0..23 : ")
MinuteError=TimeError("Minutes Should be between 0..59 : ")
class Time:
    hours=0
    minutes=0
    def disp_time(self):
        print "The correct Time is : ",self.hours,":",self.minutes
t=Time()
ok=False
while not ok:
    try:
        t.hours=input("Enter hours (0..23) : ")
        if t.hours<0 or t.hours>23:
            raise HourError
        else:
            print "Hours: ",t.hours
        t.minutes=input("Enter minutes (0..59) : ")
        if t.minutes<0 or t.minutes>59:
            raise MinuteError
        else:
            print " Minutes: " ,t.minutes
        ok=True
    except TimeError ,e:
        print e.message, " Time entered is : ",t.hours,":",t.minutes
t.disp_time()
```

Output:

```
Enter hours (0..23) : 25
Hours should be between 0..23 :   Time entered is :  25 : 0
Enter hours (0..23) : 10
Hours:  10
Enter minutes (0..59) : 65
Minutes Should be between 0..59 :   Time entered is :  10 : 65
Enter hours (0..23) : 12
Hours:  12
Enter minutes (0..59) : 30
Minutes:  30
The correct Time is :  12 : 30
>>> |
```

75) Generator function that displays factorial of first 10 numbers

```
def Fact_Gen(a,n):  
  
    f=1  
    for i in range (a,n+1):  
        f=f*i  
        yield f  
print "Factorial of numbers in the range 1-10 : "  
  
for i in Fact_Gen(1,10):  
    print i,
```

Output:

```
Factorial of numbers in the range 1-10 :  
1 2 6 24 120 720 5040 40320 362880 3628800  
>>> |
```

76) Prime numbers till a given number, using generator function

```
def Prime(num):  
    a=2  
    while a<num:  
        pr=1  
        for i in range(2,a,1):  
            if a%i==0:  
                pr=0  
        if pr==1:  
            yield a  
        a=a+1  
un=raw_input("Prime numbers till the given number:")  
n=int(un)  
for i in Prime(n):  
    print i,
```

Output:

```
Prime numbers till the given number:20  
2 3 5 7 11 13 17 19  
>>> |
```

77) Tuple in form  $(n, (2^{**}n)-1)$ , using generator function

```
def tuple_in_form_n_2_raised_to_n_minus_1(n):  
  
    t=()  
  
    for i in range (n+1):  
        x=(2**i)-1  
        t=i,x,  
        yield t  
  
n=int(raw_input("Number : "))  
  
print "To generate values in the following tuple form : (n, (2**n)-1) "  
  
for i in tuple_in_form_n_2_raised_to_n_minus_1(n):  
    print i,
```

Output:

```
Number      : 5
To generate values in the following tuple form : (n, (2**n)-1)
(0, 0) (1, 1) (2, 3) (3, 7) (4, 15) (5, 31)
>>> |
```

78) Generator function that displays 2 raised to a power upto a number n

```
def Series_2_raised_to_n(n):

    f=1
    for i in range (n+1):
        f=2**i
        yield f

n=int(raw_input("Number      : "))

print "A generator function that generates values of series 2**n"

for i in Series_2_raised_to_n(n):
    print i,
```

Output:

```
Number      : 10
A generator function that generates values of series 2**n
1 2 4 8 16 32 64 128 256 512 1024
>>> |
```

79) Generator function to generate 7 random numbers

```
import random
def lottery():
    for i in range(7):
        yield random.randint(1000,9999)
for random_no in lottery():
    print 'and the next number is..',random_no
```

Output:

```
and the next number is.. 9462
and the next number is.. 9756
and the next number is.. 3537
and the next number is.. 6403
and the next number is.. 6859
and the next number is.. 6496
and the next number is.. 4274
>>> |
```

80) Program that returns True if a string starts with a digit

```
def startsWithDigit(word):
    for ch in word:
        if ch.isdigit():
            return True
        else:
            return False

def main():
    str=raw_input("Enter a string : ")
    print "The string has a digit in the string in the beginning : ", startsWithDigit(str)

main()
```

Output:

```
Enter a string : 5forle
The string has a digit in the string in the beginning :  True
>>>

Enter a string : forlan
The string has a digit in the string in the beginning :  False
>>> |
```

81) Armstrong number between 2 given numbers

```
a=input("Enter First number : ")
b=input("Enter Second number : ")

print "Armstrong numbers between ",a," and ", b

for i in range(a,b+1,1):
    n=i
    m=n
    sm=0

    while (n>0):
        digit=n%10
        sm=sm+digit**3
        n=n/10
    if (m==sm):
        print m,
```

Output:

```
Enter First number : 200
Enter Second number : 600
Armstrong numbers between  200  and  600
370 371 407
>>> |
```

82) Program to determine whether a given substring is present in the string or not (meta search)

```
def metasearch():
    import re
    p=re.compile('singer')
    search1=re.search(p, 'Some singers sing well')
    if search1:
        match=search1.group()
        index=search1.start()
        lindex=search1.end()
        print "matched", match, "at index", index , "ending at" ,lindex
    else:
        print "No match found"

metasearch()
```

Output:

```
matched singer at index 5 ending at 11
>>> |
```

83) Delete an element from a list using pop()

```
L1 = [1, 2, 5, 4, 70, 10, 90, 80, 50]
print "\nPrinting the original list : "
print L1

print "\nStoring the deleted element value in variable a : \n",L1[1]
a= L1.pop(1) # here the element deleted will be returned to 'a'

print "\nDisplaying the new list after deletion of the 2nd element : "
print L1

print "\nDisplaying the deleted : "
print a
```

Output:

```
Printing the original list :
[1, 2, 5, 4, 70, 10, 90, 80, 50]

Storing the deleted element value in variable a :
2

Displaying the new list after deletion of the 2nd element :
[1, 5, 4, 70, 10, 90, 80, 50]

Displaying the deleted :
2
>>>
```

84) Generator function to check whether file exists or not

```
try:
    f = open('integers.txt')
    s = f.readline()
    i = int(s.strip())
except (IOError, ValueError):
    print "An I/O error or a ValueError occurred".format(IOError,ValueError)
except:
    print "An unexpected error occurred"
    raise
```

Output:

```
An I/O error or a ValueError occurred
>>> |
```

85) Program to input and display employee name and ID, and sort them in ascending order of their ID's.

```
empinfo = dict()

n=input("Enter total number of employees : ")
i=1

while i<=n:
    a=raw_input("enter number : ")
    b=raw_input("enter name : ")
    empinfo[a]=b
    i=i+1

L=empinfo.keys()
print "Before Sorting "
print "Employee Number",'\'t','Employee Name"
|
for i in L:
    print i,"\'t\'t\'t",empinfo[i]

M=empinfo.keys()
M.sort()
print "Employee Information after sorting on Keys : "
print "Employee Number",'\'t','Employee Name "

for i in M:
    print i,"\'t\'t\'t",empinfo[i]
```

Output:

```
Enter total number of employees : 3
enter number : 302
enter name : Abhi
enter number : 001
enter name : Kevin
enter number : 293
enter name : Darzene
Before Sorting
Employee Number      Employee Name
302                  Abhi
001                  Kevin
293                  Darzene
Employee Information after sorting on Keys :
Employee Number      Employee Name
001                  Kevin
293                  Darzene
302                  Abhi
>>> |
```

86) Input two tuples, and swap their values

```
t1=tuple()
n=input("Total number of values in first tuple : ")

for i in range(n):
    a=input("Enter elements : ")
    t1=t1+(a,)

t2=tuple()
m=input("Total number of values in second tuple : ")
for i in range(m):
    a=input("Enter elements : ")
    t2=t2+(a,)

print "\n First Tuple :"
print t1
print "\n Second Tuple : "
print t2
t1,t2=t2,t1
print "\n AFTER SWAPPING : "
print "\n First Tuple : "
print t1
print "\n Second Tuple : "
print t2
```

Output:

```
Total number of values in first tuple : 3
Enter elements : 12
Enter elements : 23
Enter elements : 34
Total number of values in second tuple : 3
Enter elements : 1
Enter elements : 2
Enter elements : 3

First Tuple :
(12, 23, 34)

Second Tuple :
(1, 2, 3)

AFTER SWAPPING :

First Tuple :
(1, 2, 3)

Second Tuple :
(12, 23, 34)
>>> |
```

87) Maximum and minimum values in a tuple

```
t=tuple()
n=input("Enter size: ")
print "Enter all numbers one after other: "
for i in range(n):
    a=raw_input()
    t=t+(a,)
print "Output is "
for i in range(n):
    print t[i],\n

mx=t[0]
mn=t[0]

for i in range(n):
    if mx<t[i]:
        mx=t[i]
    if mn>t[i]:
        mn=t[i]

print "\nMaximum value in the tuple = ",mx
print "Minimum value in the tuple = ",mn
```

Output:

```
Enter size: 4
Enter all numbers one after other:
2
3
4
5
Output is
2 3 4 5
Maximum value in the tuple = 5
Minimum value in the tuple = 2
>>> |
```

#### 88) Decimal to Binary

```
def binary(n):
    """Function to print binary number for the input decimal """
    if n > 1:
        binary(n//2)
    print n % 2, #end = ''

# Take decimal number from user
dec = int(input("Enter an integer: "))
print "The number = ",
binary(dec)
```

Output:

```
Enter an integer: 53
The number = 1 1 0 1 0 1
>>> |
```

#### 89) Binary to Decimal

```
print"Program to input a Binary number and print its Decimal equivalent"
n=input("Enter a binary number : ")
x=0
decimal=0
while(n>0):
    digit=n%10
    decimal=decimal+digit*2**x
    n=n/10
    x=x+1
print"The Decimal equivalent of the Binary number is : ", decimal
```

Output:

```
Program to input a Binary number and print its Decimal equivalent
Enter a binary number : 1000101
The Decimal equivalent of the Binary number is : 69
>>> |
```

90) A number raised to another number

```
a=input("Enter a number base : ")
b=input("Enter power : ")
print "after raising ",a, "to power", "b:", a**b
```

Output:

```
Enter a number base : 5
Enter power : 4
  after raising  5   to the power  4 = 625
>>> |
```

# SQL Programs:

## 1) Database: XII-B (Classlist and marks tables)

```
mysql> use xiib;
Database changed
mysql> create table classlist(roll int(2),name varchar(14),dob date,house varchar(8),mobile char(10),emirate varchar(15));
Query OK, 0 rows affected (0.05 sec)

mysql> insert into classlist values(1,'Aaditya','1999-12-19','Jupiter','0559029312','Dubai');
Query OK, 1 row affected (0.03 sec)

mysql> insert into classlist values(2,'Abhishek','1999-01-06','Saturn','0563964501','Dubai');
Query OK, 1 row affected (0.00 sec)

mysql> insert into classlist values(3,'Kevin','1999-09-13','Jupiter','0526902870','Abu Dhabi');
Query OK, 1 row affected (0.00 sec)

mysql> insert into classlist values(4,'Rohit','1999-07-21','Mars','0566969404','Sharjah');
Query OK, 1 row affected (0.08 sec)

mysql> insert into classlist values(5,'Vivek','1999-04-02','Neptune','0529209312','Sharjah');
Query OK, 1 row affected (0.00 sec)

mysql> create table marks(roll int(2),math int(3),phy int(3),chem int(3));
ERROR 1050 (42S01): Table 'marks' already exists
mysql> drop table marks;
Query OK, 0 rows affected (0.11 sec)

mysql> create table marks(roll int(2),math int(3),phy int(3),chem int(3));
Query OK, 0 rows affected (0.02 sec)

mysql> insert into marks values(1,90,90,90);
Query OK, 1 row affected (0.03 sec)

mysql> insert into marks values(2,93,92,94);
Query OK, 1 row affected (0.02 sec)

mysql> insert into marks values(3,95,89,93);
Query OK, 1 row affected (0.00 sec)

mysql> insert into marks values(4,91,83,90);
Query OK, 1 row affected (0.00 sec)

mysql> insert into marks values(5,94,93,89);
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from classlist;
+----+-----+-----+-----+-----+-----+
| roll | name      | dob        | house     | mobile    | emirate   |
+----+-----+-----+-----+-----+-----+
| 1   | Aaditya   | 1999-12-19 | Jupiter   | 0559029312 | Dubai     |
| 2   | Abhishek  | 1999-01-06 | Saturn    | 0563964501 | Dubai     |
| 3   | Kevin      | 1999-09-13 | Jupiter   | 0526902870 | Abu Dhabi |
| 4   | Rohit     | 1999-07-21 | Mars      | 0566969404 | Sharjah   |
| 5   | Vivek     | 1999-04-02 | Neptune   | 0529209312 | Sharjah   |
+----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select * from marks;
+----+-----+-----+-----+
| roll | math | phy | chem |
+----+-----+-----+-----+
| 1   | 90   | 90  | 90  |
| 2   | 93   | 92  | 94  |
| 3   | 95   | 89  | 93  |
| 4   | 91   | 83  | 90  |
| 5   | 94   | 93  | 89  |
+----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select name from classlist where emirate='Dubai';
+-----+
| name |
+-----+
| Aaditya |
| Abhishek |
+-----+
2 rows in set (0.00 sec)
```

```
mysql> select name from classlist where name like '%k';
+-----+
| name |
+-----+
| Abhishek |
| Vivek |
+-----+
2 rows in set (0.00 sec)
```

```
mysql> select name,math,phy,chem from classlist c,marks m where m.roll=c.roll;
+-----+-----+-----+-----+
| name   | math | phy | chem |
+-----+-----+-----+-----+
| Aaditya | 90  | 90  | 90  |
| Abhishek | 93  | 92  | 94  |
| Kevin   | 95  | 89  | 93  |
| Rohit   | 91  | 83  | 90  |
| Vivek   | 94  | 93  | 89  |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> alter table marks;
Query OK, 0 rows affected (0.01 sec)
```

2) Database Mall- Shop and shop type table

```
mysql> use mall;
Database changed
mysql> create table shop(SID int(4),Name varchar(15),Area varchar(15));
Query OK, 0 rows affected (0.05 sec)

mysql> insert into shop values(1001,'Max','Jumeirah');
Query OK, 1 row affected (0.06 sec)

mysql> insert into shop values(1443,'McDonalds','Box Park');
Query OK, 1 row affected (0.08 sec)

mysql> insert into shop values(1234,'Geekay Games','Box Park');
Query OK, 1 row affected (0.00 sec)

mysql> insert into shop values(1922,'Starbucks','Wasl Road');
Query OK, 1 row affected (0.09 sec)

mysql> create table shoptype(SID int(4), Type varchar(15), contact char(10), Offers varchar(3));
Query OK, 0 rows affected (0.14 sec)

mysql> insert into shoptype values(1001,'Fashion','0522903332','YES');
Query OK, 1 row affected (0.08 sec)

mysql> insert into shoptype values(1443,'Restaurant','0559023999','NO');
Query OK, 1 row affected (0.01 sec)

mysql> insert into shoptype values(1234,'Gaming','0569220123','YES');
Query OK, 1 row affected (0.00 sec)

mysql> insert into shoptype values(1922,'Restaurant','0504490022','YES');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from shop;
+----+-----+-----+
| SID | Name      | Area   |
+----+-----+-----+
| 1001 | Max       | Jumeirah|
| 1443 | McDonalds | Box Park|
| 1234 | Geekay Games | Box Park|
| 1922 | Starbucks  | Wasl Road|
+----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> select * from shoptype;
+----+-----+-----+-----+
| SID | Type      | contact | Offers |
+----+-----+-----+-----+
| 1001 | Fashion   | 0522903332 | YES    |
| 1443 | Restaurant | 0559023999 | NO     |
| 1234 | Gaming    | 0569220123 | YES    |
| 1922 | Restaurant | 0504490022 | YES    |
+----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> select * from shop order by name;
+----+-----+-----+
| SID | Name      | Area   |
+----+-----+-----+
| 1234 | Geekay Games | Box Park|
| 1001 | Max       | Jumeirah|
| 1443 | McDonalds  | Box Park|
| 1922 | Starbucks  | Wasl Road|
+----+-----+-----+
4 rows in set (0.09 sec)
```

```
mysql> select Name,contact from shop s,shoptype t where t.SID=s.SID;
+-----+-----+
| Name      | contact |
+-----+-----+
| Max       | 0522903332 |
| McDonalds | 0559023999 |
| Geekay Games | 0569220123 |
| Starbucks  | 0504490022 |
+-----+-----+
4 rows in set (0.00 sec)
```

### 3) Office database (Employee and Income Tables)

```
mysql> use office;
Database changed
mysql> create table employee(id int(3), name varchar(15), age int(2), hiredate date, job varchar(15), mobile char(10));
Query OK, 0 rows affected (0.09 sec)

mysql> insert into employee values(134,'Harry',32,'2010-12-21','Accountant','6782349012');
Query OK, 1 row affected (0.07 sec)

mysql> insert into employee values(103,'Kevin',30,'2007-02-22','Executive','6943312334');
Query OK, 1 row affected (0.00 sec)

mysql> insert into employee values(101,'Bendtner',32,'2004-01-01','CEO','6849209102');
Query OK, 1 row affected (0.08 sec)

mysql> insert into employee values(182,'Catherine',24,'2015-11-23','Secretary','6783490912');
Query OK, 1 row affected (0.00 sec)

mysql> insert into employee values(114,'Abhishek',29,'2009-11-12','Sales Manager','6889709666');
Query OK, 1 row affected (0.09 sec)

mysql> insert into employee values(153,'Neill',30,'2011-03-20','Sales Manager','6909029902');
Query OK, 1 row affected (0.03 sec)

mysql> create table income(id int(3),sal int(5),comm int(4));
Query OK, 0 rows affected (0.04 sec)

mysql> insert into income values(134,8000,500);
Query OK, 1 row affected (0.00 sec)

mysql> insert into income values(103,12000,1000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into income values(101,20000,4000);
Query OK, 1 row affected (0.08 sec)

mysql> insert into income values(182,5000,200);
Query OK, 1 row affected (0.08 sec)

mysql> insert into income values(114,7500,1500);
Query OK, 1 row affected (0.00 sec)

mysql> insert into income values(153,8000,1000);
Query OK, 1 row affected (0.00 sec)

mysql> select * from employee;
+----+-----+-----+-----+-----+-----+
| id | name   | age  | hiredate | job    | mobile |
+----+-----+-----+-----+-----+-----+
| 134 | Harry  | 32  | 2010-12-21 | Accountant | 6782349012 |
| 103 | Kevin   | 30  | 2007-02-22 | Executive  | 6943312334 |
| 101 | Bendtner | 32  | 2004-01-01 | CEO        | 6849209102 |
| 182 | Catherine | 24  | 2015-11-23 | Secretary  | 6783490912 |
| 114 | Abhishek | 29  | 2009-11-12 | Sales Manager | 6889709666 |
| 153 | Neill   | 30  | 2011-03-20 | Sales Manager | 6909029902 |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)
```

```
mysql> select * from income;
```

id	sal	comm
134	8000	500
103	12000	1000
101	20000	4000
182	5000	200
114	7500	1500
153	8000	1000

6 rows in set (0.00 sec)

```
mysql> select * from employee order by name desc;
```

id	name	age	hiredate	job	mobile
153	Neill	30	2011-03-20	Sales Manager	6909029902
103	Kevin	30	2007-02-22	Executive	6943312334
134	Harry	32	2010-12-21	Accountant	6782349012
182	Catherine	24	2015-11-23	Secretary	6783490912
101	Bendtner	32	2004-01-01	CEO	6849209102
114	Abhishek	29	2009-11-12	Sales Manager	6889709666

6 rows in set (0.00 sec)

```
mysql> select name,age from employee where age=30;
```

name	age
Kevin	30
Neill	30

2 rows in set (0.00 sec)

```
mysql> select name,sal+comm from employee e, income i where i.id=e.id;
```

name	sal+comm
Harry	8500
Kevin	13000
Bendtner	24000
Catherine	5200
Abhishek	9000
Neill	9000

6 rows in set (0.00 sec)

```
mysql> select name,sal+comm from employee e,income i where i.sal+comm<1000;  
Empty set (0.00 sec)
```

4) Database Sport: Event (athletics) and coach table

```
mysql> use sport;
Database changed
mysql> create table event(code int(1), event varchar(15), date date, no_of_participants int(2), venue varchar(15));
Query OK, 0 rows affected (0.12 sec)

mysql> desc event;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| code | int(1) | YES | | NULL | |
| event | varchar(15) | YES | | NULL | |
| date | date | YES | | NULL | |
| no_of_participants | int(2) | YES | | NULL | |
| venue | varchar(15) | YES | | NULL | |
+-----+-----+-----+-----+-----+
5 rows in set (0.04 sec)

mysql> insert into event values(1,'4x100 relay','2016-11-04',28,'Police Stadium');
Query OK, 1 row affected (0.09 sec)

mysql> insert into event values(2,'100m hurdles','2016-10-31',8,'Police Stadium');
Query OK, 1 row affected (0.07 sec)

mysql> insert into event values(3,'Long Jump','2016-11-01',12,'Estadio Kevin');
Query OK, 1 row affected (0.08 sec)

mysql> insert into event values(4,'Shot Put','2016-11-05',10,'Estadio Kevin');
Query OK, 1 row affected (0.08 sec)

mysql> insert into event values(5,'Pole Vault','2016-10-30',14,'HH Rohit Arena');
Query OK, 1 row affected (0.00 sec)

mysql> create table coach(code int(1), coach varchar(15), id int(3), contact char(10));
Query OK, 0 rows affected (0.11 sec)

mysql> insert into coach values(1,'Denis',213,'0553489000');
Query OK, 1 row affected (0.01 sec)

mysql> insert into coach values(2,'Larry',193,'0524912568');
Query OK, 1 row affected (0.00 sec)

mysql> insert into coach values(3,'Marc',101,'0564278391');
Query OK, 1 row affected (0.00 sec)

mysql> insert into coach values(2,'Niall',111,'0559012232');
Query OK, 1 row affected (0.00 sec)

mysql> insert into coach values(4,'Mathieu',192,'0558829303');
Query OK, 1 row affected (0.00 sec)

mysql> select * from event;
+-----+-----+-----+-----+-----+
| code | event | date | no_of_participants | venue |
+-----+-----+-----+-----+-----+
| 1 | 4x100 relay | 2016-11-04 | 28 | Police Stadium |
| 2 | 100m hurdles | 2016-10-31 | 8 | Police Stadium |
| 3 | Long Jump | 2016-11-01 | 12 | Estadio Kevin |
| 4 | Shot Put | 2016-11-05 | 10 | Estadio Kevin |
| 5 | Pole Vault | 2016-10-30 | 14 | HH Rohit Arena |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```

mysql> select * from coach;
+---+---+---+-----+
| code | coach | id | contact |
+---+---+---+-----+
| 1 | Denis | 213 | 0553489000 |
| 2 | Larry | 193 | 0524912568 |
| 3 | Marc | 101 | 0564278391 |
| 2 | Niall | 111 | 0559012232 |
| 4 | Mathieu | 192 | 0558829303 |
+---+---+---+-----+
5 rows in set (0.00 sec)

mysql> select * from event where no_of_participants between 10 and 20;
+---+---+---+-----+-----+
| code | event | date | no_of_participants | venue |
+---+---+---+-----+-----+
| 3 | Long Jump | 2016-11-01 | 12 | Estadio Kevin |
| 4 | Shot Put | 2016-11-05 | 10 | Estadio Kevin |
| 5 | Pole Vault | 2016-10-30 | 14 | HH Rohit Arena |
+---+---+---+-----+-----+
3 rows in set (0.07 sec)

mysql> select event,venue from event where venue in('Police Stadium','HH Rohit Arena');
+---+-----+
| event | venue |
+---+-----+
| 4x100 relay | Police Stadium |
| 100m hurdles | Police Stadium |
| Pole Vault | HH Rohit Arena |
+---+-----+
3 rows in set (0.00 sec)

mysql> select event,coach from event e, coach c where c.code=e.code;
+---+---+
| event | coach |
+---+---+
| 4x100 relay | Denis |
| 100m hurdles | Larry |
| Long Jump | Marc |
| 100m hurdles | Niall |
| Shot Put | Mathieu |
+---+---+
5 rows in set (0.00 sec)

mysql> select event,date from event where date between '2016-11-01' and '2016-11-05';
+---+-----+
| event | date |
+---+-----+
| 4x100 relay | 2016-11-04 |
| Long Jump | 2016-11-01 |
| Shot Put | 2016-11-05 |
+---+-----+
3 rows in set (0.07 sec)

```

##### 5) Database taxi (Cab and Driver tables)

```
mysql> use taxi;
Database changed
mysql> create table cab(carno int(5),model varchar(10),brand varchar(10),driver varchar(10),contact char(10));
Query OK, 0 rows affected (0.04 sec)

mysql> insert into cab values(32933,'Innova','Toyota','Heisenberg','0505902333');
Query OK, 1 row affected (0.16 sec)

mysql> insert into cab values(20190,'IS300','Lexus','Kevin','0526902870');
Query OK, 1 row affected (0.00 sec)

mysql> insert into cab values(10191,'C300','Mercedes','Abhishek','0569099121');
Query OK, 1 row affected (0.01 sec)

mysql> insert into cab values(59049,'Altima','Nissan','Heskey','0551999302');
Query OK, 1 row affected (0.00 sec)

mysql> create table fare(carno int(5), startfee int(2), feeperkm int(1), salik int(1));
Query OK, 0 rows affected (0.02 sec)

mysql> insert into fare values(32933,6,2,3);
Query OK, 1 row affected (0.02 sec)

mysql> insert into fare values(20190,10,3,4);
Query OK, 1 row affected (0.09 sec)

mysql> insert into fare values(10191,8,3,3);
Query OK, 1 row affected (0.00 sec)

mysql> insert into fare values(59049,7,2,2);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from cab;
```

carno	model	brand	driver	contact
32933	Innova	Toyota	Heisenberg	0505902333
20190	IS300	Lexus	Kevin	0526902870
10191	C300	Mercedes	Abhishek	0569099121
59049	Altima	Nissan	Heskey	0551999302

4 rows in set (0.00 sec)

```
mysql> select * from fare;
```

carno	startfee	feeperkm	salik
32933	6	2	3
20190	10	3	4
10191	8	3	3
59049	7	2	2

4 rows in set (0.00 sec)

```
mysql> select brand,model from cab;
+-----+-----+
| brand | model |
+-----+-----+
| Toyota | Innova |
| Lexus | IS300 |
| Mercedes | C300 |
| Nissan | Altima |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select brand,model,startfee from cab c, fare f where f.carno=c.carno;
+-----+-----+-----+
| brand | model | startfee |
+-----+-----+-----+
| Toyota | Innova | 6 |
| Lexus | IS300 | 10 |
| Mercedes | C300 | 8 |
| Nissan | Altima | 7 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from cab order by driver asc;
+-----+-----+-----+-----+-----+
| carno | model | brand | driver | contact |
+-----+-----+-----+-----+-----+
| 10191 | C300 | Mercedes | Abhishek | 0569099121 |
| 32933 | Innova | Toyota | Heisenberg | 0505902333 |
| 59049 | Altima | Nissan | Heskey | 0551999302 |
| 20190 | IS300 | Lexus | Kevin | 0526902870 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

6) Vacation database (Vehicle and travel tables)

```
mysql> use vacation;
Database changed
mysql> create table vehicle(code int(3), vtype varchar(18), perkm int(3));
Query OK, 0 rows affected (0.02 sec)

mysql> insert into vehicle values(101,'volvo bus',160);
Query OK, 1 row affected (0.00 sec)

mysql> insert into vehicle values(102,'ac bus',150);
Query OK, 1 row affected (0.10 sec)

mysql> insert into vehicle values(103,'ordinary bus',90);
Query OK, 1 row affected (0.01 sec)

mysql> insert into vehicle values(104,'suv',40)
      -> ;
Query OK, 1 row affected (0.00 sec)

mysql> insert into vehicle values(105,'car',20);
Query OK, 1 row affected (0.00 sec)

mysql> create table travel(no int(3),name varchar(15),tdate date,km int(3),code int(3), nop int(2));
Query OK, 0 rows affected (0.03 sec)

mysql> insert into travel values(101,'Janish','2015-11-13',200,101,32);
Query OK, 1 row affected (0.02 sec)

mysql> insert into travel values(103,'Vedika','2016-04-21',100,103,45);
Query OK, 1 row affected (0.02 sec)

mysql> insert into travel values(105,'Tarun','2016-03-23',350,102,42);
Query OK, 1 row affected (0.02 sec)

mysql> insert into travel values(102,'John','2016-02-13',90,102,40);
Query OK, 1 row affected (0.00 sec)

mysql> insert into travel values(107,'Ahmed','2015-01-10',75,104,2);
Query OK, 1 row affected (0.00 sec)

mysql> insert into travel values(104,'Raveena','2016-05-28',80,105,4);
Query OK, 1 row affected (0.00 sec)

mysql> insert into travel values(106,'Kripal','2016-02-06',200,101,25);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from vehicle;
+---+-----+-----+
| code | vtype      | perkm |
+---+-----+-----+
| 101 | volvo bus  | 160  |
| 102 | ac bus     | 150  |
| 103 | ordinary bus| 90   |
| 104 | suv         | 40   |
| 105 | car         | 20   |
+---+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select * from travel;
+-----+-----+-----+-----+-----+-----+
| no   | name    | tdate   | km    | code   | nop   |
+-----+-----+-----+-----+-----+-----+
| 101  | Janish  | 2015-11-13| 200  | 101   | 32   |
| 103  | Vedika  | 2016-04-21 | 100  | 103   | 45   |
| 105  | Tarun   | 2016-03-23 | 350  | 102   | 42   |
| 102  | John    | 2016-02-13 | 90   | 102   | 40   |
| 107  | Ahmed   | 2015-01-10 | 75   | 104   | 2    |
| 104  | Raveena | 2016-05-28 | 80   | 105   | 4    |
| 106  | Kripal  | 2016-02-06 | 200  | 101   | 25   |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

```
mysql> select count(*) from travel group by code having count(*)>1;
+-----+
| count(*) |
+-----+
|      2   |
|      2   |
+-----+
2 rows in set (0.10 sec)
```

```
mysql> select distinct code from travel;
+---+
| code |
+---+
| 101 |
| 103 |
| 102 |
| 104 |
| 105 |
+---+
5 rows in set (0.00 sec)
```

```
mysql> select a.code,name,vtype from travel a, vehicle b where a.code=b.code and km<90;
+---+-----+-----+
| code | name    | vtype  |
+---+-----+-----+
| 104 | Ahmed   | suv   |
| 105 | Raveena | car   |
+---+-----+-----+
2 rows in set (0.06 sec)
```

```

mysql> select name,km*perkm from travel a, vehicle b where a.code=b.code and a.code='105';
+-----+-----+
| name    | km*perkm |
+-----+-----+
| Raveena |      1600 |
+-----+-----+
1 row in set (0.01 sec)

mysql> select no,name,tdate from travel order by no desc;
+-----+-----+-----+
| no   | name  | tdate   |
+-----+-----+-----+
| 107  | Ahmed  | 2015-01-10 |
| 106  | Kripal | 2016-02-06 |
| 105  | Tarun  | 2016-03-23 |
| 104  | Raveena | 2016-05-28 |
| 103  | Vedika | 2016-04-21 |
| 102  | John   | 2016-02-13 |
| 101  | Janish | 2015-11-13 |
+-----+-----+-----+
7 rows in set (0.00 sec)

```

### 7) Dept database (Department and Employee tables (2015 Board qn)

```

mysql> create database dept;
Query OK, 1 row affected (0.00 sec)

mysql> use dept;
Database changed
mysql> create table dept(dcode char(3), department varchar(20), location varchar(10));
Query OK, 0 rows affected (0.15 sec)

mysql> insert into dept values('D01','infrastructure','delhi');
Query OK, 1 row affected (0.00 sec)

mysql> insert into dept values('D02','marketing','delhi');
Query OK, 1 row affected (0.00 sec)

mysql> insert into dept values('D03','media','mumbai');
Query OK, 1 row affected (0.00 sec)

mysql> insert into dept values('D04','finance','kolkota');
Query OK, 1 row affected (0.15 sec)

mysql> insert into dept values('D05','human resource','mumbai');
Query OK, 1 row affected (0.05 sec)

mysql> create table employee(eno int(4),name varchar(15),doj date, dob date, gender varchar(6), dcode char(3));
Query OK, 0 rows affected (0.20 sec)

mysql> insert into employee values(1001,'George','2013-09-02','1991-09-01','male','D01');
Query OK, 1 row affected (0.00 sec)

mysql> insert into employee values(1002,'Ryma','2012-12-11
      '> ',1990-12-15','female','D03');
Query OK, 1 row affected (0.09 sec)

```

```

mysql> insert into employee values(1003,'Mohitesh','2013-02-03','1987-09-04','male','D05');
Query OK, 1 row affected (0.00 sec)

mysql> insert into employee values(1007,'Anil','2014-01-17','1984-10-19','male','D05');
Query OK, 1 row affected (0.02 sec)

mysql> insert into employee values(1004,'Manila','2012-12-09','1986-11-14','female','D01');
Query OK, 1 row affected (0.00 sec)

mysql> insert into employee values(1005,'Sahay','2013-11-18','1987-03-31','male','D02');
Query OK, 1 row affected (0.07 sec)

mysql> insert into employee values(1006,'Jaya','2014-06-09','1985-06-23','female','D05');
Query OK, 1 row affected (0.08 sec)

mysql> select * from dept;
+-----+-----+-----+
| dcode | department | location |
+-----+-----+-----+
| D01  | infrastructure | delhi |
| D02  | marketing | delhi |
| D03  | media | mumbai |
| D04  | finance | kolkota |
| D05  | human resource | mumbai |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+-----+
| eno | name | doj | dob | gender | dcode |
+-----+-----+-----+-----+-----+-----+
| 1001 | George | 2013-09-02 | 1991-09-01 | male | D01 |
| 1002 | Ryma | 2012-12-11 | 1990-12-15 | female | D03 |
| 1003 | Mohitesh | 2013-02-03 | 1987-09-04 | male | D05 |
| 1007 | Anil | 2014-01-17 | 1984-10-19 | male | D05 |
| 1004 | Manila | 2012-12-09 | 1986-11-14 | female | D01 |
| 1005 | Sahay | 2013-11-18 | 1987-03-31 | male | D02 |
| 1006 | Jaya | 2014-06-09 | 1985-06-23 | female | D05 |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> select eno,name,gender from employee order by eno asc;
+-----+-----+-----+
| eno | name | gender |
+-----+-----+-----+
| 1001 | George | male |
| 1002 | Ryma | female |
| 1003 | Mohitesh | male |
| 1004 | Manila | female |
| 1005 | Sahay | male |
| 1006 | Jaya | female |
| 1007 | Anil | male |
+-----+-----+-----+
7 rows in set (0.00 sec)

```

```
mysql> select * from employee where gender like 'male';
+----+-----+-----+-----+-----+-----+
| eno | name   | doj    | dob     | gender | dcode |
+----+-----+-----+-----+-----+-----+
| 1001 | George | 2013-09-02 | 1991-09-01 | male   | D01   |
| 1003 | Mohitesh | 2013-02-03 | 1987-09-04 | male   | D05   |
| 1007 | Anil   | 2014-01-17 | 1984-10-19 | male   | D05   |
| 1005 | Sahay  | 2013-11-18 | 1987-03-31 | male   | D02   |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select eno,name from employee where dob between '1987-01-01' and '1991-12-01';
+----+-----+
| eno | name   |
+----+-----+
| 1001 | George |
| 1002 | Ryma   |
| 1003 | Mohitesh|
| 1005 | Sahay  |
+----+-----+
4 rows in set (0.00 sec)

mysql> select count(*) from employee where gender='female' and dob > '1986-01-01';
+-----+
| count(*) |
+-----+
|      2 |
+-----+
1 row in set (0.00 sec)

mysql> select max(doj), min(dob) from employee;
+-----+-----+
| max(doj) | min(dob) |
+-----+-----+
| 2014-06-09 | 1984-10-19 |
+-----+-----+
1 row in set (0.06 sec)
```

8) Tech database (Shop and Accessory table) (2014 Board qn)

```
mysql> create table accessory(no char(3), name varchar(20), price int(5), id char(3));
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> insert into accessory values('A01','Motherboard',12000,'S01');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into accessory values('A02','Hard Disk',5000,'S01');
Query OK, 1 row affected (0.04 sec)
```

```
mysql> insert into accessory values('A03','Keyboard',500,'S02');
Query OK, 1 row affected (0.19 sec)
```

```
mysql> insert into accessory values('A04','Mouse',300,'S01');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into accessory values('A05','Motherboard',13000,'S02');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into accessory values('A06','Keyboard',400,'S03');
Query OK, 1 row affected (0.12 sec)
```

```
mysql> insert into accessory values('A07','LCD',6000,'S04');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into accessory values('T08','LCD',5500,'S05');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into accessory values('T09','Mouse',350,'S05');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into accessory values('T10','Hard Disk',4500,'S03');
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from accessory;
```

no	name	price	id
A01	Motherboard	12000	S01
A02	Hard Disk	5000	S01
A03	Keyboard	500	S02
A04	Mouse	300	S01
A05	Motherboard	13000	S02
A06	Keyboard	400	S03
A07	LCD	6000	S04
T08	LCD	5500	S05
T09	Mouse	350	S05
T10	Hard Disk	4500	S03

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

```
mysql> drop table shop;
Query OK, 0 rows affected (0.10 sec)
```

```
mysql> create table shop(id char(3),sname varchar(20), area varchar(13));
Query OK, 0 rows affected (0.03 sec)
```

```

mysql> insert into shop values('S01','ABC Computronics','CP');
Query OK, 1 row affected (0.00 sec)

mysql> insert into shop values('S02','All Infotech Media','GK II');
Query OK, 1 row affected (0.08 sec)

mysql> insert into shop values('S03','Tech Shoppe','CP');
Query OK, 1 row affected (0.02 sec)

mysql> insert into shop values('S04','Geeks Tecno Soft','Nehru Place');
Query OK, 1 row affected (0.08 sec)

mysql> insert into shop values('S05','Hitech techstore','Nehru Place');
Query OK, 1 row affected (0.00 sec)

mysql> select * from shop;
+----+-----+-----+
| id | sname          | area |
+----+-----+-----+
| S01 | ABC Computronics | CP   |
| S02 | All Infotech Media | GK II |
| S03 | Tech Shoppe      | CP   |
| S04 | Geeks Tecno Soft | Nehru Place |
| S05 | Hitech techstore | Nehru Place |
+----+-----+-----+
5 rows in set (0.00 sec)

mysql> select name,price from accessory order by price;
+-----+-----+
| name        | price |
+-----+-----+
| Mouse       | 300  |
| Mouse       | 350  |
| Keyboard    | 400  |
| Keyboard    | 500  |
| Hard Disk   | 4500 |
| Hard Disk   | 5000 |
| LCD         | 5500 |
| LCD         | 6000 |
| Motherboard | 12000|
| Motherboard | 13000|
+-----+-----+
10 rows in set (0.00 sec)

mysql> select id,sname from shop where area='Nehru Place';
+----+-----+
| id | sname          |
+----+-----+
| S04 | Geeks Tecno Soft |
| S05 | Hitech techstore |
+----+-----+
2 rows in set (0.00 sec)

mysql> select min(price), max(price) from accessory;
+-----+-----+
| min(price) | max(price) |
+-----+-----+
|      300   |     13000  |
+-----+-----+

```

```
1 row in set (0.00 sec)
```

```
mysql> select name,min(price),max(price) from accessory;
```

name	min(price)	max(price)
Motherboard	300	13000

```
1 row in set (0.00 sec)
```

```
mysql> select sname,name,price from shop s, accessory a where s.id=a.id;
```

sname	name	price
ABC Computronics	Motherboard	12000
ABC Computronics	Hard Disk	5000
All Infotech Media	Keyboard	500
ABC Computronics	Mouse	300
All Infotech Media	Motherboard	13000
Tech Shoppe	Keyboard	400
Geeks Tecno Soft	LCD	6000
Hitech techstore	LCD	5500
Hitech techstore	Mouse	350
Tech Shoppe	Hard Disk	4500

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

```
mysql> select distinct name from accessory where price>=5000;
```

name
Motherboard
Hard Disk
LCD

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

```
mysql> select area,count(*) from shop group by area;
```

area	count(*)
CP	2
GK II	1
Nehru Place	2

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

```
mysql>
```

9) Trading database(items and traders table 2013 board qn)

```
mysql> create database tradin;
Query OK, 1 row affected (0.00 sec)

mysql> use tradin;
Database changed
mysql> create table items(code int(4), iname varchar(20), qty int(3), price int(5), company varchar(10), tcode char(3));
Query OK, 0 rows affected (0.20 sec)

mysql> insert into items values(1001,'digital pad 12i',120,11000,'xenita','t01');
Query OK, 1 row affected (0.01 sec)

mysql> insert into items values(1006,'led screen 40',70,38000,'santora','t02');
Query OK, 1 row affected (0.10 sec)

mysql> insert into items values(1004,'car gps system',50,21500,'geoknow','t01');
Query OK, 1 row affected (0.00 sec)

mysql> insert into items values(1003,'digital camera 12x',160,8000,'digiclick','t02');
Query OK, 1 row affected (0.08 sec)

mysql> insert into items values(1005,'USB 32gb',600,1200,'storehome','t03');
Query OK, 1 row affected (0.00 sec)

mysql> create table traders(tcode char(3),tname varchar(20),city varchar(8));
Query OK, 0 rows affected (0.12 sec)

mysql> insert into traders values('t01','electronic sales','mumbai');
Query OK, 1 row affected (0.02 sec)

mysql> insert into traders values('t03','busy store coop','delhi');
Query OK, 1 row affected (0.08 sec)

mysql> insert into traders values('t02','disp house inc','chennai');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from items;
+-----+-----+-----+-----+-----+-----+
| code | iname           | qty   | price | company | tcode |
+-----+-----+-----+-----+-----+-----+
| 1001 | digital pad 12i | 120   | 11000 | xenita   | t01  |
| 1006 | led screen 40   | 70    | 38000 | santora   | t02  |
| 1004 | car gps system  | 50    | 21500 | geoknow   | t01  |
| 1003 | digital camera 12x | 160   | 8000  | digiclick | t02  |
| 1005 | USB 32gb         | 600   | 1200  | storehome | t03  |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select * from traders;
+-----+-----+-----+
| tcode | tname           | city   |
+-----+-----+-----+
| t01  | electronic sales | mumbai |
| t03  | busy store coop   | delhi  |
| t02  | disp house inc   | chennai |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select * from items order by iname;
+-----+-----+-----+-----+-----+-----+
| code | iname           | qty   | price | company | tcode |
+-----+-----+-----+-----+-----+-----+
| 1004 | car gps system  | 50    | 21500 | geoknow   | t01  |
| 1003 | digital camera 12x | 160   | 8000  | digiclick | t02  |
| 1001 | digital pad 12i   | 120   | 11000 | xenita   | t01  |
| 1006 | led screen 40   | 70    | 38000 | santora   | t02  |
| 1005 | USB 32gb         | 600   | 1200  | storehome | t03  |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select iname,price from items where price between 10000 and 22000;
+-----+-----+
| iname       | price |
+-----+-----+
| digital pad 12i | 11000 |
| car gps system | 21500 |
+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> select tcode,count(*) from items;
+-----+-----+
| tcode | count(*) |
+-----+-----+
| t01  |      5 |
+-----+-----+
1 row in set (0.00 sec)
```

```

mysql> select tcode,count(*) from items group by tcode;
+-----+-----+
| tcode | count(*) |
+-----+-----+
| t01  |      2 |
| t02  |      2 |
| t03  |      1 |
+-----+-----+
3 rows in set (0.00 sec)

mysql> select iname,qty,price from items where qty>150;
+-----+-----+-----+
| iname        | qty   | price  |
+-----+-----+-----+
| digital camera 12x | 160 | 8000 |
| USB 32gb       | 600  | 1200  |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select tname,iname from traders t,items i where i.tcode=t.tcode;
+-----+-----+
| tname    | iname        |
+-----+-----+
| electronic sales | digital pad 12i |
| disp house inc   | led screen 40  |
| electronic sales | car gps system |
| disp house inc   | digital camera 12x|
| busy store coop  | USB 32gb      |
+-----+-----+
5 rows in set (0.00 sec)

```

```

mysql> select tname,iname from traders t,items i where i.tcode=t.tcode order by tname desc;
+-----+-----+
| tname    | iname        |
+-----+-----+
| electronic sales | digital pad 12i |
| electronic sales | car gps system |
| disp house inc   | led screen 40  |
| disp house inc   | digital camera 12x|
| busy store coop  | USB 32gb      |
+-----+-----+
5 rows in set (0.00 sec)

```

10) Workplace database (Staff and Salary tables) 2006 board qn

```
mysql> create database workplace;
Query OK, 1 row affected (0.00 sec)

mysql> use workplace;
Database changed
mysql> create table staff(id int(3), name varchar(10), dept varchar(15), gender char(1), yearsexp int(2));
Query OK, 0 rows affected (0.03 sec)

mysql> insert into staff values(101,'Siddharth','SALES','M',12);
Query OK, 1 row affected (0.00 sec)

mysql> insert into staff values(107,'Naman','RESEARCH','M',5);
Query OK, 1 row affected (0.00 sec)

mysql> insert into staff values(109,'Janvi','FINANCE','F',9);
Query OK, 1 row affected (0.00 sec)

mysql> insert into staff values(117,'James','SALES','M',3);
Query OK, 1 row affected (0.07 sec)

mysql> create table salary(id int(3), basic int(5), allowance int(4), comm_percent int(2));
Query OK, 0 rows affected (0.03 sec)

mysql> insert into salary values(101,12000,1000,3);
Query OK, 1 row affected (0.00 sec)

mysql> insert into salary values(107,32000,4000,5);
Query OK, 1 row affected (0.00 sec)

mysql> insert into salary values(109,42000,1700,20);
Query OK, 1 row affected (0.00 sec)

mysql> insert into salary values(117,21700,2600,30);
Query OK, 1 row affected (0.00 sec)

mysql> select * from staff;
+----+-----+-----+-----+
| id | name   | dept    | gender | yearsexp |
+----+-----+-----+-----+
| 101 | Siddharth | SALES | M     |      12 |
| 107 | Naman    | RESEARCH | M     |       5 |
| 109 | Janvi    | FINANCE | F     |       9 |
| 117 | James    | SALES | M     |       3 |
+----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from salary;
+----+-----+-----+-----+
| id | basic | allowance | comm_percent |
+----+-----+-----+-----+
| 101 | 12000 | 1000     | 3          |
| 107 | 32000 | 4000     | 5          |
| 109 | 42000 | 1700     | 20         |
| 117 | 21700 | 2600     | 30         |
+----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> select * from staff where dept='SALES' and yearsexp>10;
+----+-----+-----+-----+
| id | name      | dept   | gender | yearsexp |
+----+-----+-----+-----+
| 101 | Siddharth | SALES | M       |      12 |
+----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select name,basic+allowance from staff s, salary t where t.id=s.id;
+-----+-----+
| name      | basic+allowance |
+-----+-----+
| Siddharth |        13000 |
| Naman     |        36000 |
| Janvi     |        43700 |
| James     |        24300 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select name,comm_percent from staff s, salary t where t.id=s.id and comm_percent>15;
+-----+-----+
| name      | comm_percent |
+-----+-----+
| Janvi     |          20 |
| James     |          30 |
+-----+-----+
2 rows in set (0.00 sec)

mysql> select count(*) from staff where gender='M';
+-----+
| count(*) |
+-----+
|      3 |
+-----+
1 row in set (0.00 sec)
```