SHARJAH INDIAN SCHOOL , Juwaiza

Tel: 06 5665775, P.O. Box – 2324, Sharjah ; e-mail: boys@sissharjah.com Website: www.sisjuwaiza.com

OUR VISION

Educate Enlighten Empower

COMPUTER SCIENCE

PRACTICAL RECORD FILE

NAME :	ALBIN BINU MATHEW
ROLL NO:	
CLASS:	XII C
ACADEMIC YEAR :	2020 - 2021

INDEX

SHARJAH INDIAN SCHOOL

(AFFILIATED TO CENTRAL BOARD OF SECONDARY EDUCATION, NEW DELHI)

Certificate

AISSCE Roll No : _______________

Certified that

Master Albin Binu Mathew

of Grade XII Section C has carried out the Practical Work in

Subject Code : 083 Subject : Computer Science

prescribed by the Central Board of Secondary Education, New Delhi

during the academic year 2020 – 2021.

Jihan Shareen	12-02-2021
…………………	..…………………...
Teacher in Charge	Date

…………………….	…………………….
Internal Examiner	External Examiner

RECORD FILE –A.1

Objective: Write a menu driven program to :

. Find factorial of a number.

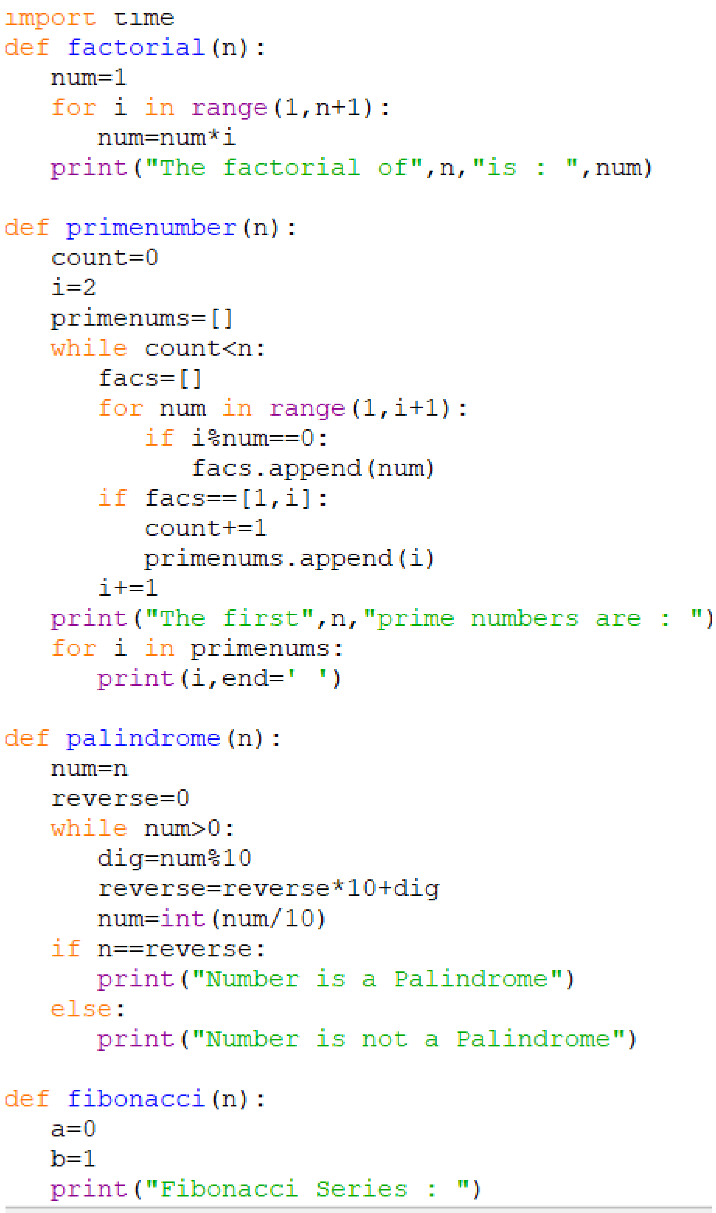
. Print first ‘n’ Prime numbers.

. To check if a number is a Palindrome or not.

. Display Fibonacci Series up to range ‘n’.

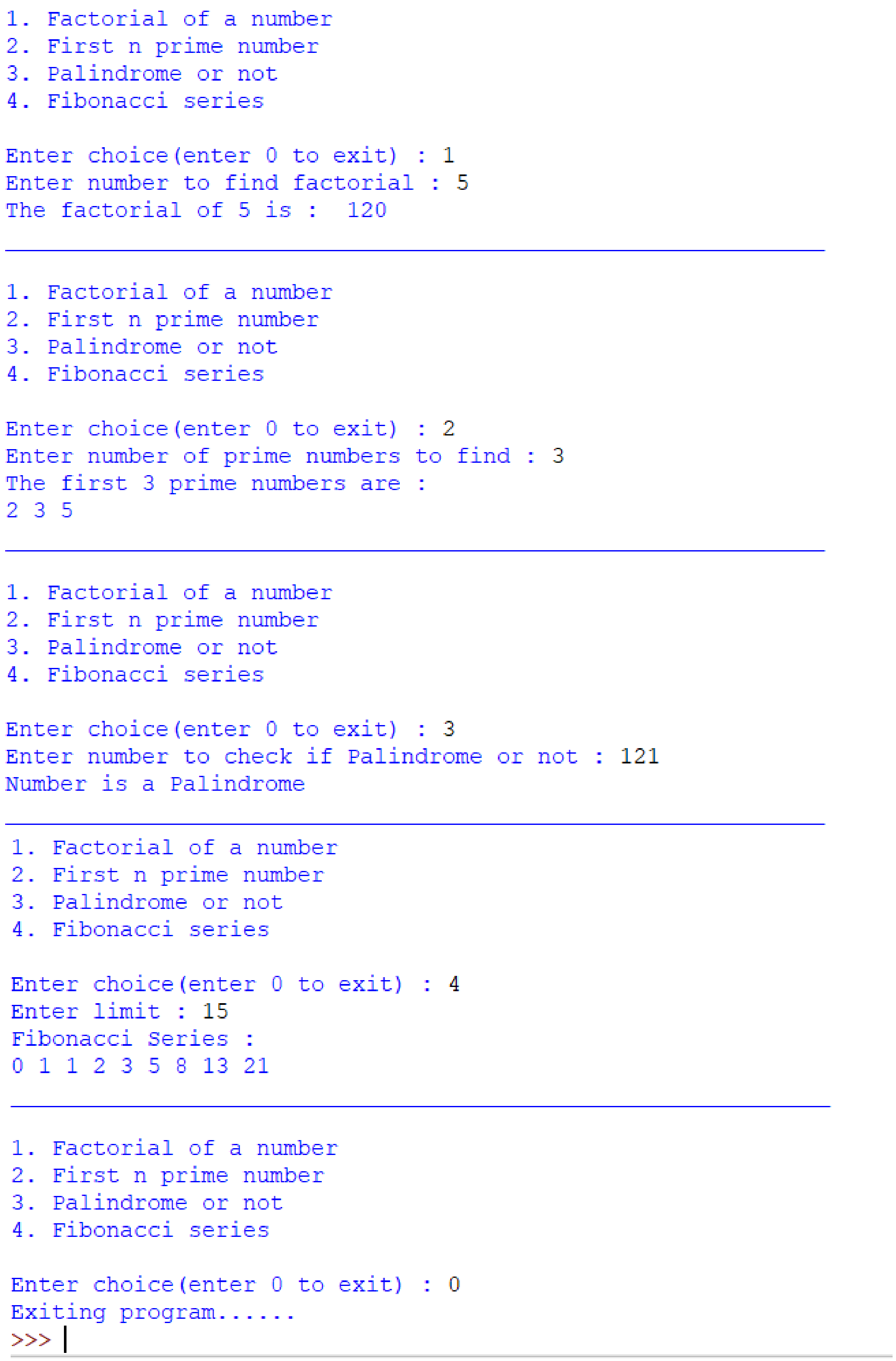
Concepts used: user-defined functions.

Source code:



```
import time
def factorial(n):
    num=1
    for i in range(1,n+1):
        num=num*i
    print("The factorial of",n,"is : ",num)

def primenumber(n):
    count=0
    i=2
    primenums=[]
    while count<n:
        facs=[]
        for num in range(1,i+1):
            if i%num==0:
                facs.append(num)
        if facs==[1,i]:
            count+=1
            primenums.append(i)
        i+=1
    print("The first",n,"prime numbers are : ")
    for i in primenums:
        print(i,end=' ')

def palindrome(n):
    num=n
    reverse=0
    while num>0:
        dig=num%10
        reverse=reverse*10+dig
        num=int(num/10)
    if n==reverse:
        print("Number is a Palindrome")
    else:
        print("Number is not a Palindrome")

def fibonacci(n):
    a=0
    b=1
    print("Fibonacci Series : ")
    print(a,b,end=' ')
    while b<n:
        a,b=b,a+b
        print(b,end=' ')

while True:
    print("\n1. Factorial of a number")
    print("2. First n prime number")
    print("3. Palindrome or not")
    print("4. Fibonacci series")
    ch=int(input("\nEnter choice(enter 0 to exit) : "))
    if ch==1:
        num=int(input("Enter number to find factorial : "))
        factorial(num)
        print("____________________________________________________________")

    elif ch==2:
        num=int(input("Enter number of prime numbers to find : "))
        primenumber(num)
        print("\n____________________________________________________________")

    elif ch==3:
        num=int(input("Enter number to check if Palindrome or not : "))
        palindrome(num)
        print("____________________________________________________________")

    elif ch==4:
        num=int(input("Enter limit : "))
        fibonacci(num)
        print("\n____________________________________________________________")

    elif ch==0:
        print("Exiting program......")
        time.sleep(1)
        exit()

    else:
        print("invalid")
        print("____________________________________________________________")
```

Output:



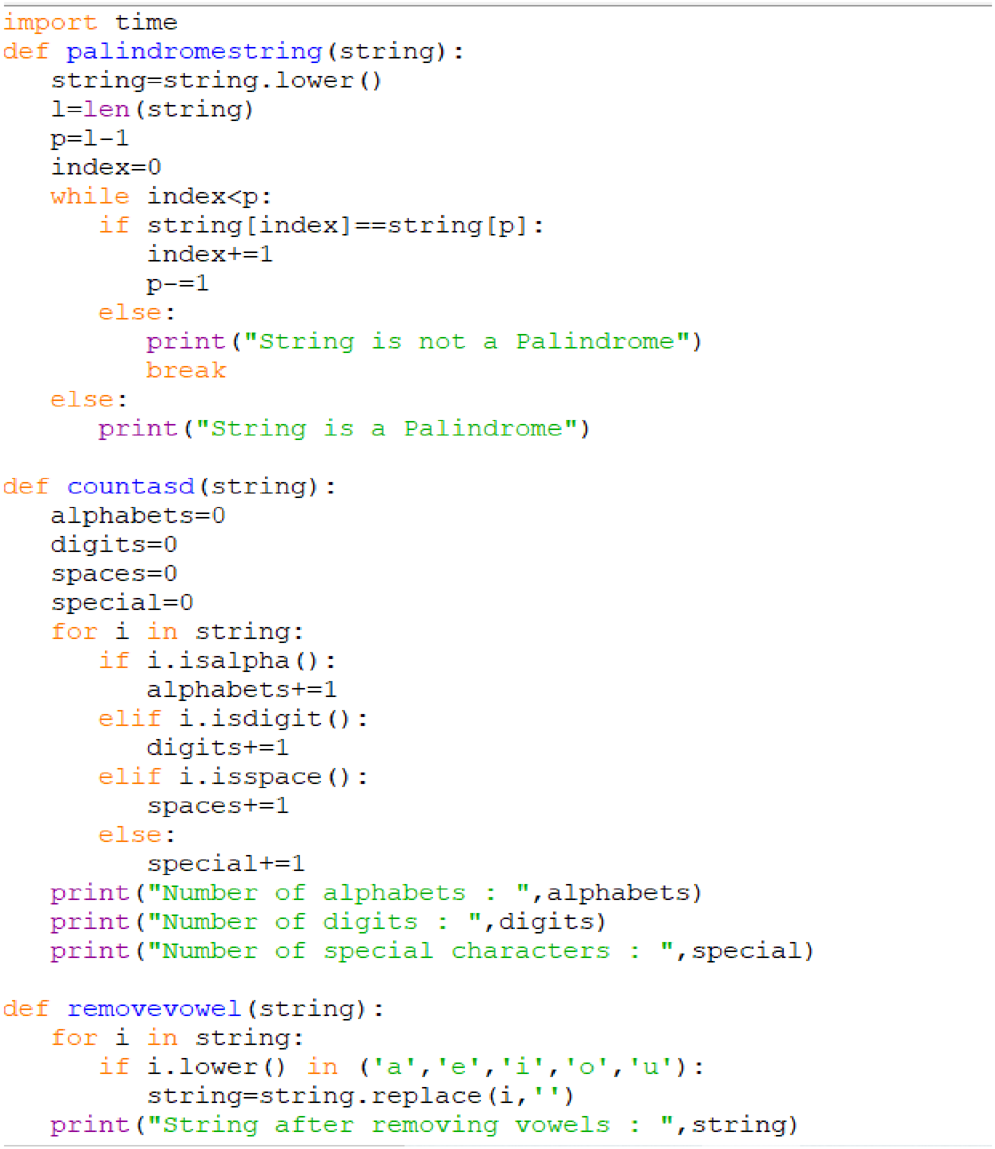
```
1. Factorial of a number
2. First n prime number
3. Palindrome or not
4. Fibonacci series

Enter choice(enter 0 to exit) : 1
Enter number to find factorial : 5
The factorial of 5 is :  120

____________________________________________________________

1. Factorial of a number
2. First n prime number
3. Palindrome or not
4. Fibonacci series

Enter choice(enter 0 to exit) : 2
Enter number of prime numbers to find : 3
The first 3 prime numbers are :
2 3 5

____________________________________________________________

1. Factorial of a number
2. First n prime number
3. Palindrome or not
4. Fibonacci series

Enter choice(enter 0 to exit) : 3
Enter number to check if Palindrome or not : 121
Number is a Palindrome

____________________________________________________________
1. Factorial of a number
2. First n prime number
3. Palindrome or not
4. Fibonacci series

Enter choice(enter 0 to exit) : 4
Enter limit : 15
Fibonacci Series :
0 1 1 2 3 5 8 13 21

____________________________________________________________

1. Factorial of a number
2. First n prime number
3. Palindrome or not
4. Fibonacci series

Enter choice(enter 0 to exit) : 0
Exiting program......
>>> |
```

RECORD FILE –A.2

Objective: To Write a menu driven program to:
. Check if a given string is a Palindrome
. Count the number of alphabets , special characters , digits in a String
. Remove all vowels from a string.
. Reverse a string

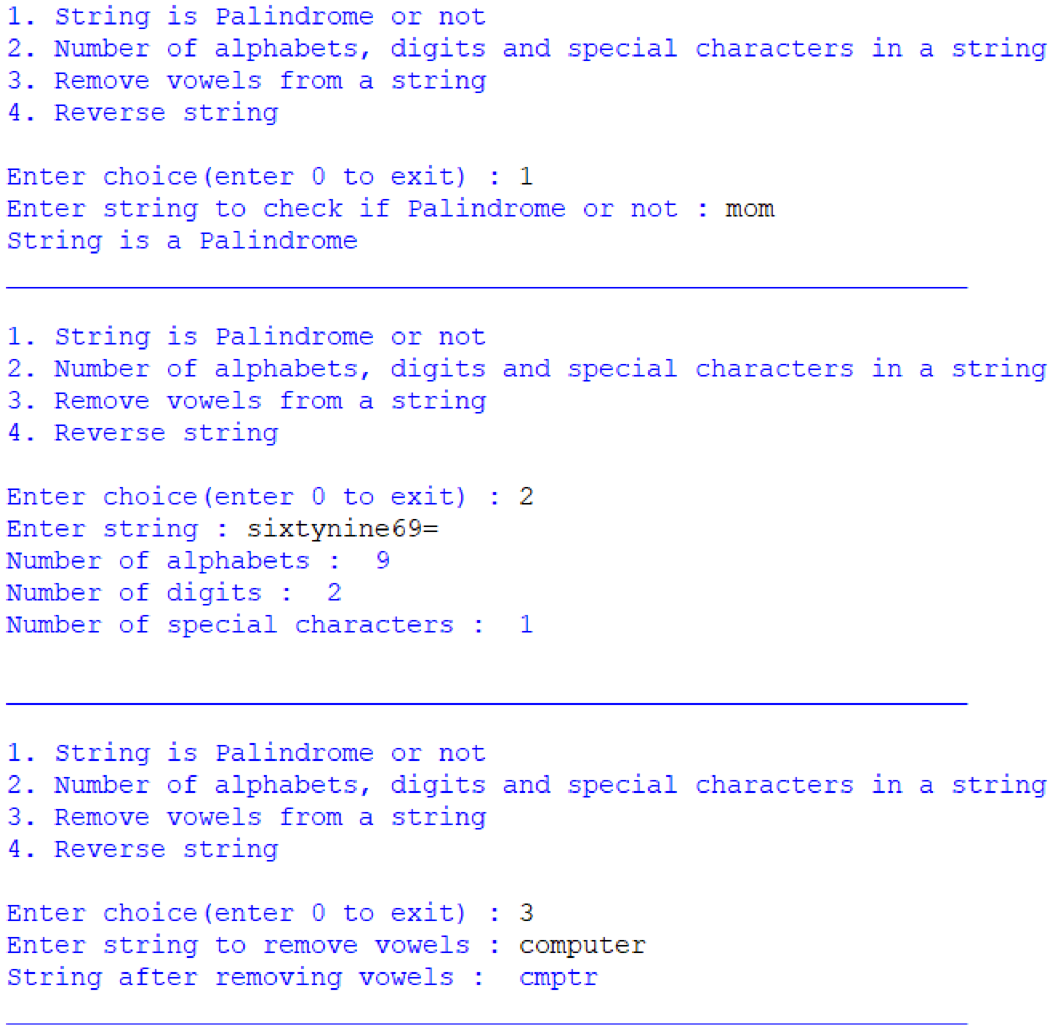
Concepts used: Built In & User defined Functions, Strings

Source code:

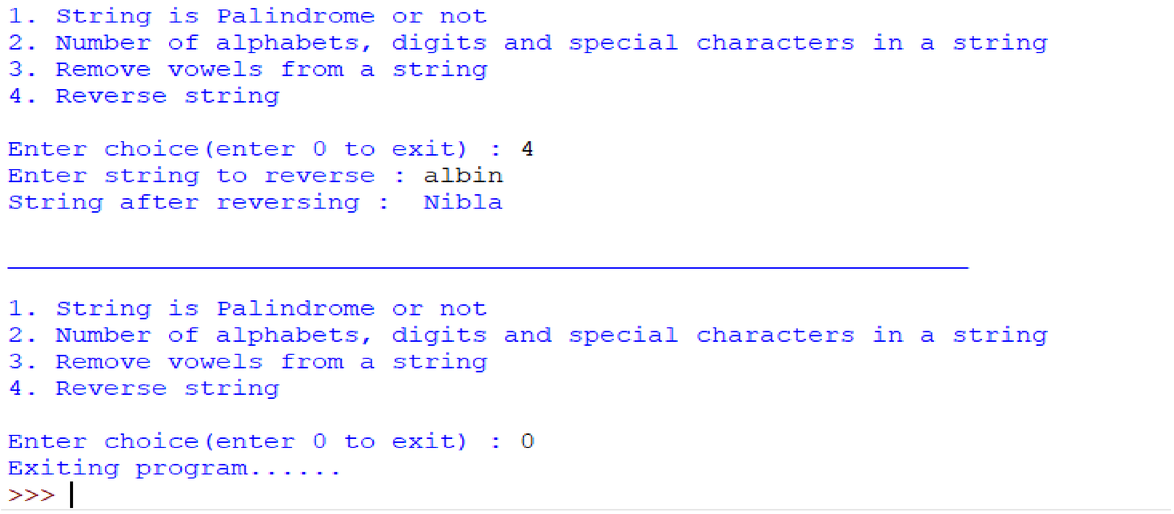


```python
import time
def palindromestring(string):
    string=string.lower()
    l=len(string)
    p=l-1
    index=0
    while index<p:
        if string[index]==string[p]:
            index+=1
            p-=1
        else:
            print("String is not a Palindrome")
            break
    else:
        print("String is a Palindrome")

def countasd(string):
    alphabets=0
    digits=0
    spaces=0
    special=0
    for i in string:
        if i.isalpha():
            alphabets+=1
        elif i.isdigit():
            digits+=1
        elif i.isspace():
            spaces+=1
        else:
            special+=1
    print("Number of alphabets : ",alphabets)
    print("Number of digits : ",digits)
    print("Number of special characters : ",special)

def removevowel(string):
    for i in string:
        if i.lower() in ('a','e','i','o','u'):
            string=string.replace(i,'')
    print("String after removing vowels : ",string)
    l=len(string)
    p=l-1
    string1=string[p::-1]
    string1=string1.capitalize()
    print("String after reversing : ",string1)

while True:
    print("\n1. String is Palindrome or not")
    print("2. Number of alphabets, digits and special characters in a string")
    print("3. Remove vowels from a string")
    print("4. Reverse string")
    ch=int(input("\nEnter choice(enter 0 to exit) : "))
    if ch==1:
        string=input("Enter string to check if Palindrome or not : ")
        palindromestring(string)
        print("____________________________________________________________")

    elif ch==2:
        string=input("Enter string : ")
        countasd(string)
        print("\n____________________________________________________________")

    elif ch==3:
        string=input("Enter string to remove vowels : ")
        removevowel(string)
        print("____________________________________________________________")

    elif ch==4:
        string=input("Enter string to reverse : ")
        reversestring(string)
        print("\n____________________________________________________________")

    elif ch==0:
        print("Exiting program......")
        time.sleep(1)
        exit()

    else:
        print("invalid")
```

Output:



```
1. String is Palindrome or not
2. Number of alphabets, digits and special characters in a string
3. Remove vowels from a string
4. Reverse string

Enter choice(enter 0 to exit) : 1
Enter string to check if Palindrome or not : mom
String is a Palindrome
______________________________________________________________

1. String is Palindrome or not
2. Number of alphabets, digits and special characters in a string
3. Remove vowels from a string
4. Reverse string

Enter choice(enter 0 to exit) : 2
Enter string : sixtynine69=
Number of alphabets :  9
Number of digits :  2
Number of special characters :  1


______________________________________________________________

1. String is Palindrome or not
2. Number of alphabets, digits and special characters in a string
3. Remove vowels from a string
4. Reverse string

Enter choice(enter 0 to exit) : 3
Enter string to remove vowels : computer
String after removing vowels :  cmptr
______________________________________________________________
```



```
1. String is Palindrome or not
2. Number of alphabets, digits and special characters in a string
3. Remove vowels from a string
4. Reverse string

Enter choice(enter 0 to exit) : 4
Enter string to reverse : albin
String after reversing :  Nibla

______________________________________________________________

1. String is Palindrome or not
2. Number of alphabets, digits and special characters in a string
3. Remove vowels from a string
4. Reverse string

Enter choice(enter 0 to exit) : 0
Exiting program......
>>> |
```

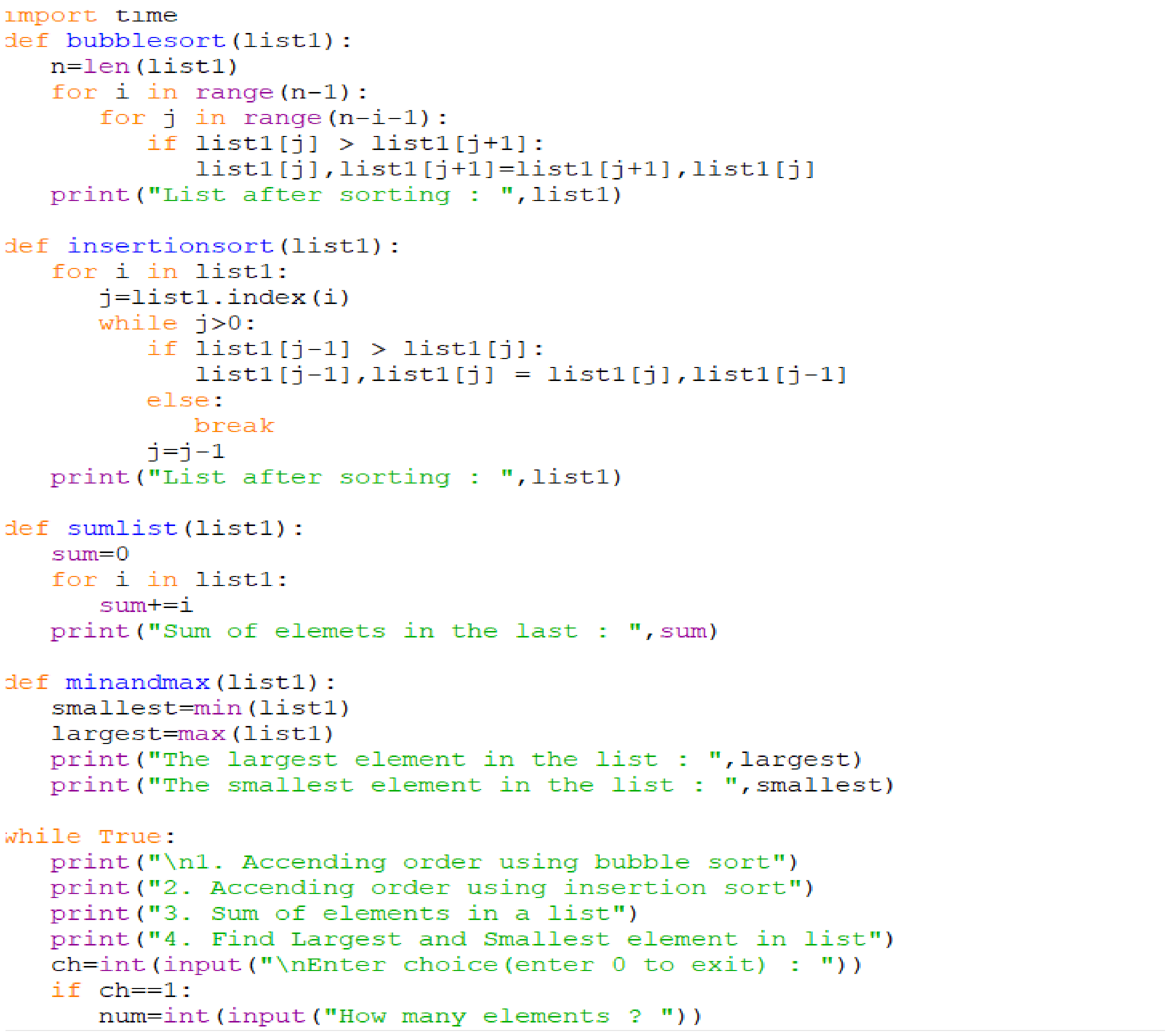
RECORD FILE –A.3

Objective: To Write a menu driven Python program that accepts a list (integers) and:

. Sort in ascending order using Bubble Sort.
. Sort in ascending order using Insertion Sort.
. Calculates sum of elements in the list.
. Largest & Smallest Element in the list.

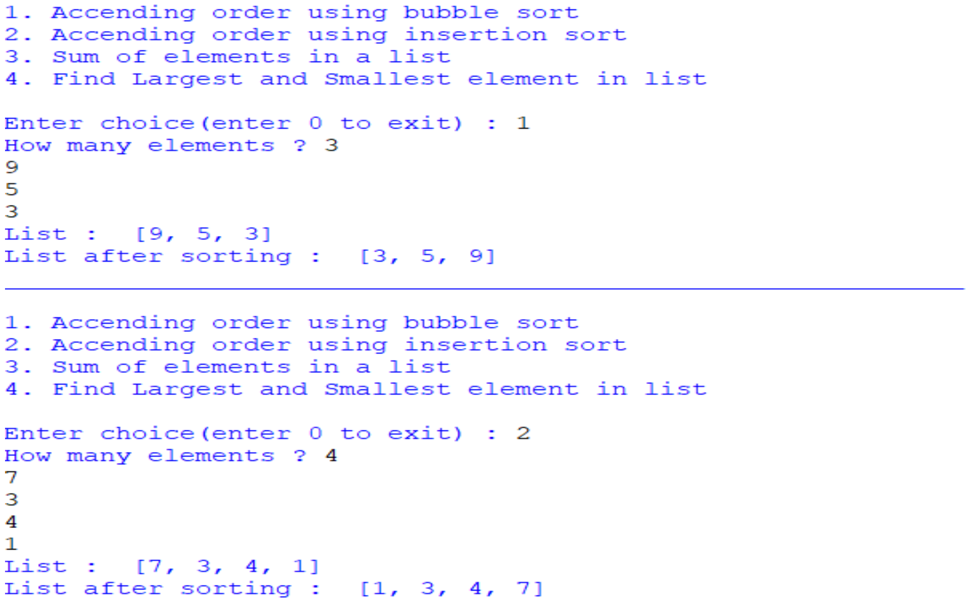
Concepts used: functions, list.

Source code:

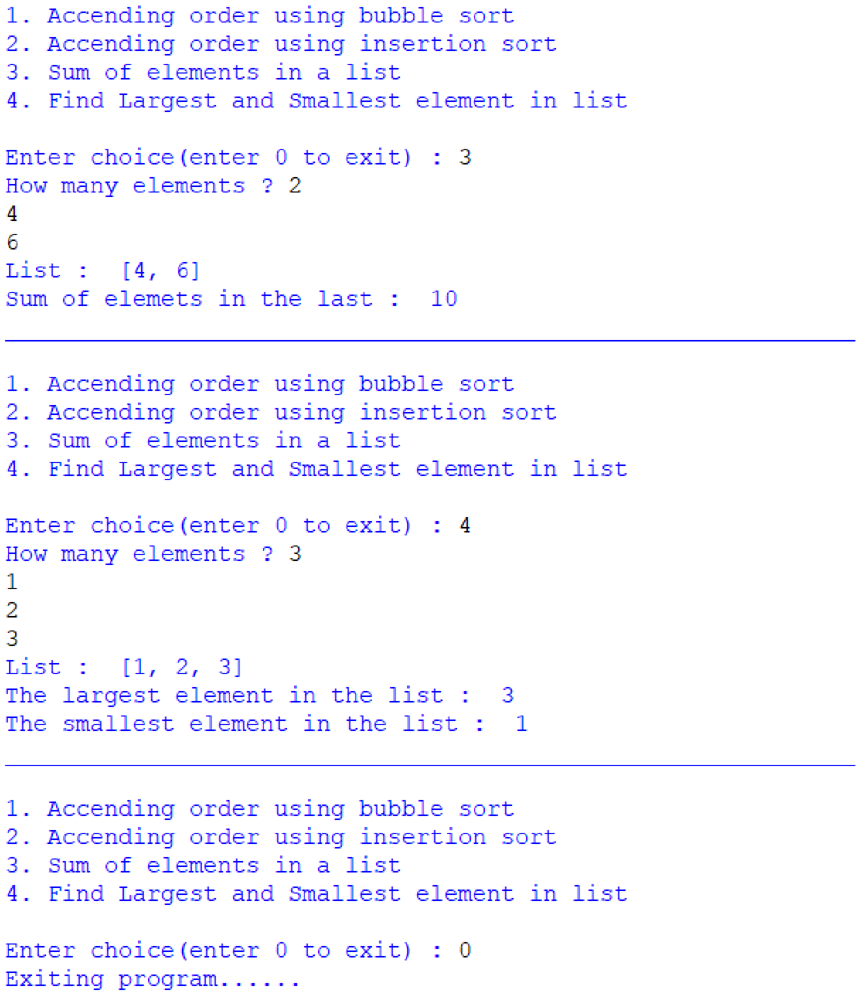


```
import time
def bubblesort(list1):
    n=len(list1)
    for i in range(n-1):
        for j in range(n-i-1):
            if list1[j] > list1[j+1]:
                list1[j],list1[j+1]=list1[j+1],list1[j]
    print("List after sorting : ",list1)

def insertionsort(list1):
    for i in list1:
        j=list1.index(i)
        while j>0:
            if list1[j-1] > list1[j]:
                list1[j-1],list1[j] = list1[j],list1[j-1]
            else:
                break
            j=j-1
    print("List after sorting : ",list1)

def sumlist(list1):
    sum=0
    for i in list1:
        sum+=i
    print("Sum of elemets in the last : ",sum)

def minandmax(list1):
    smallest=min(list1)
    largest=max(list1)
    print("The largest element in the list : ",largest)
    print("The smallest element in the list : ",smallest)

while True:
    print("\n1. Accending order using bubble sort")
    print("2. Accending order using insertion sort")
    print("3. Sum of elements in a list")
    print("4. Find Largest and Smallest element in list")
    ch=int(input("\nEnter choice(enter 0 to exit) : "))
    if ch==1:
        num=int(input("How many elements ? "))
        list1=[int(input())for i in range(num)]
        print("List : ",list1)
        bubblesort(list1)
        print("____________________________________________________________")

    elif ch==2:
        num=int(input("How many elements ? "))
        list1=[int(input())for i in range(num)]
        print("List : ",list1)
        insertionsort(list1)
        print("____________________________________________________________")

    elif ch==3:
        num=int(input("How many elements ? "))
        list1=[int(input())for i in range(num)]
        print("List : ",list1)
        sumlist(list1)
        print("____________________________________________________________")

    elif ch==4:
        num=int(input("How many elements ? "))
        list1=[int(input())for i in range(num)]
        print("List : ",list1)
        minandmax(list1)
        print("____________________________________________________________")

    elif ch==0:
        print("Exiting program......")
        time.sleep(1)
        exit()

    else:
        print("invalid")
        print("____________________________________________________________")
```

Output:



```
1. Accending order using bubble sort
2. Accending order using insertion sort
3. Sum of elements in a list
4. Find Largest and Smallest element in list

Enter choice(enter 0 to exit) : 1
How many elements ? 3
9
5
3
List :  [9, 5, 3]
List after sorting :  [3, 5, 9]
____________________________________________________________

1. Accending order using bubble sort
2. Accending order using insertion sort
3. Sum of elements in a list
4. Find Largest and Smallest element in list

Enter choice(enter 0 to exit) : 2
How many elements ? 4
7
3
4
1
List :  [7, 3, 4, 1]
List after sorting :  [1, 3, 4, 7]
```



```
1. Accending order using bubble sort
2. Accending order using insertion sort
3. Sum of elements in a list
4. Find Largest and Smallest element in list

Enter choice(enter 0 to exit) : 3
How many elements ? 2
4
6
List :  [4, 6]
Sum of elemets in the last :  10
____________________________________________________________

1. Accending order using bubble sort
2. Accending order using insertion sort
3. Sum of elements in a list
4. Find Largest and Smallest element in list

Enter choice(enter 0 to exit) : 4
How many elements ? 3
1
2
3
List :  [1, 2, 3]
The largest element in the list :  3
The smallest element in the list :  1
____________________________________________________________

1. Accending order using bubble sort
2. Accending order using insertion sort
3. Sum of elements in a list
4. Find Largest and Smallest element in list

Enter choice(enter 0 to exit) : 0
Exiting program......
```

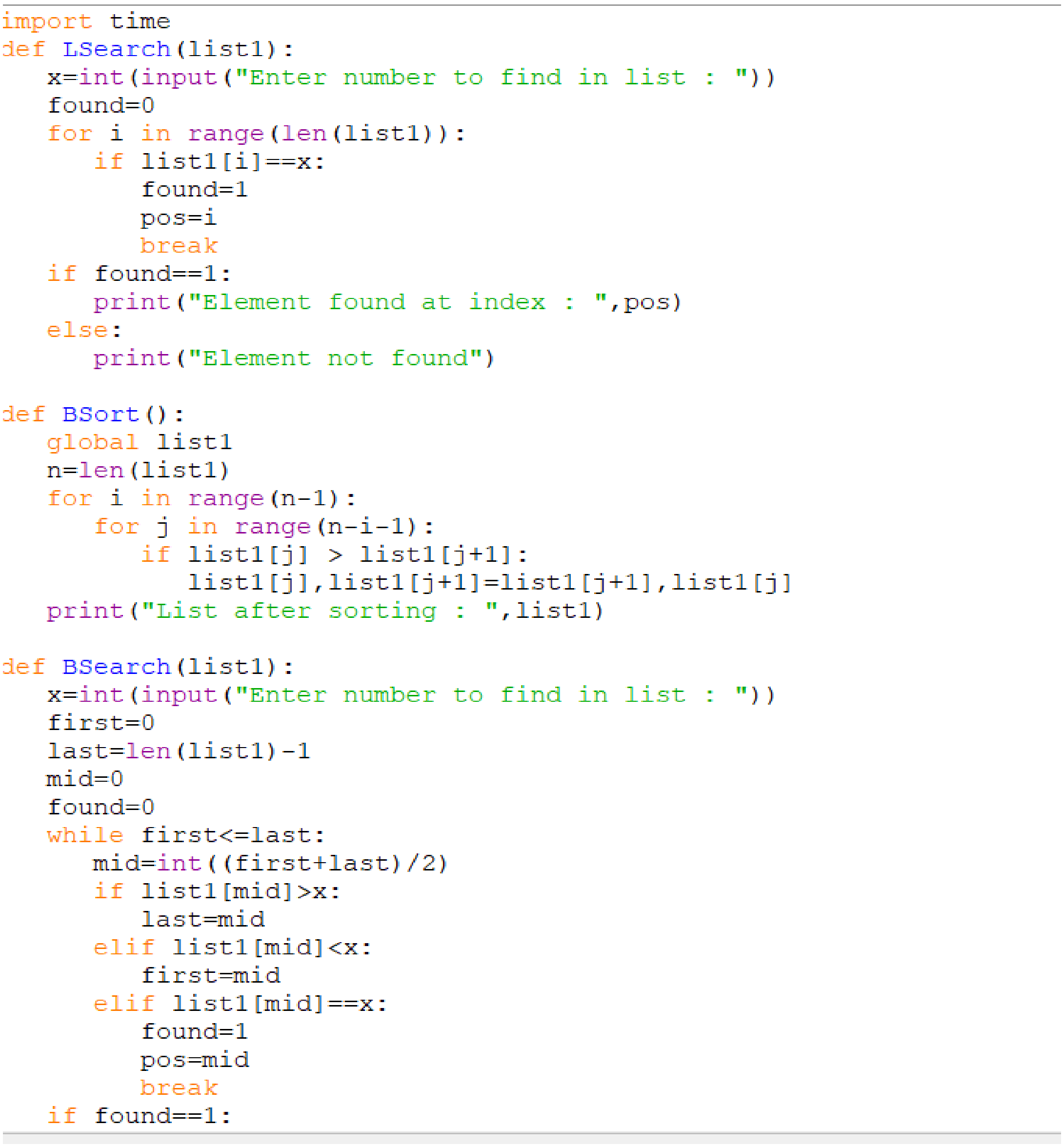
RECORD FILE –A.4

Objective: to Write a Python program to accept a list of integers from the user. Search for an element 'x' using:

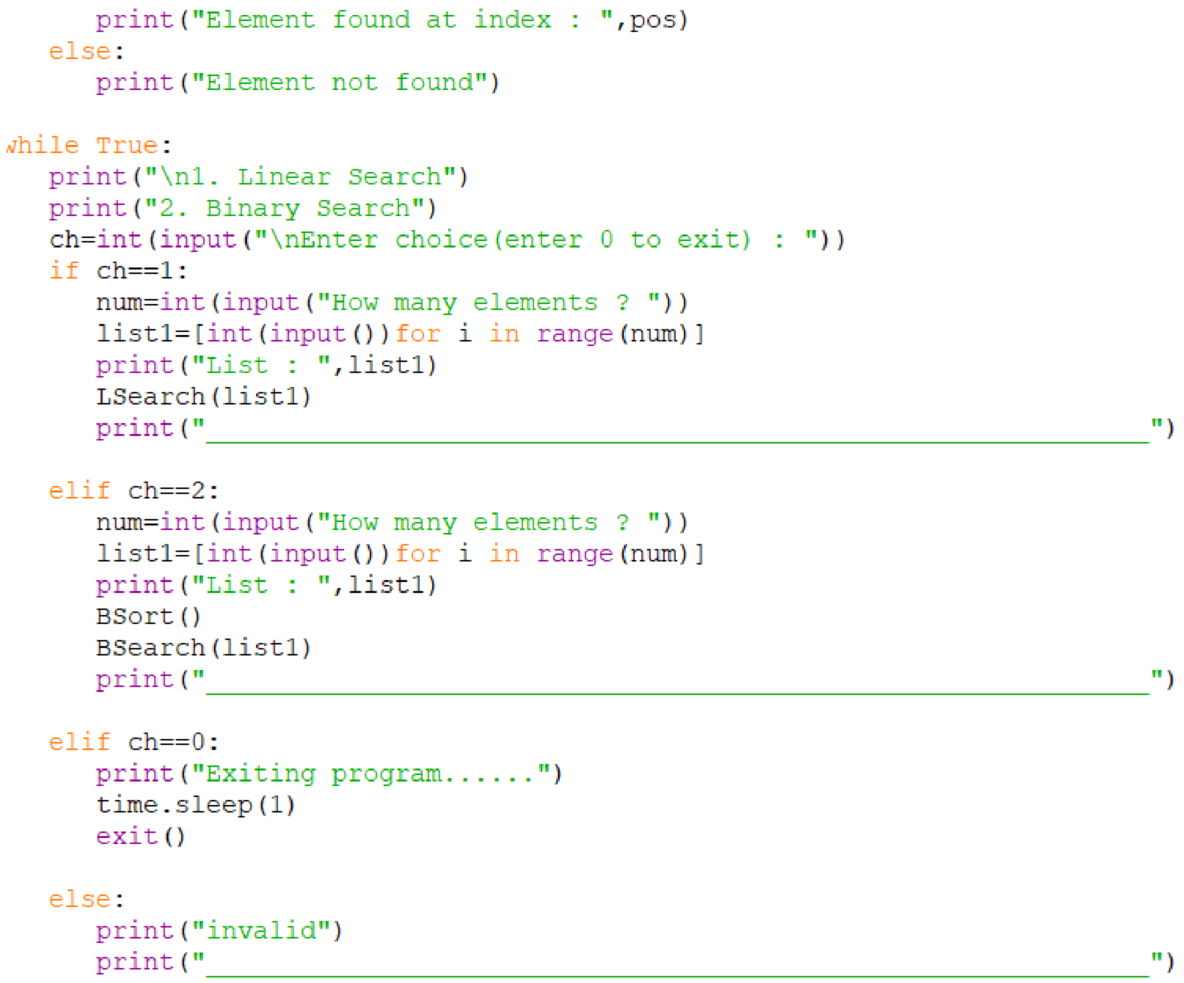
a) Linear Search
b) Binary Search

Concepts used: functions, list

Source code:

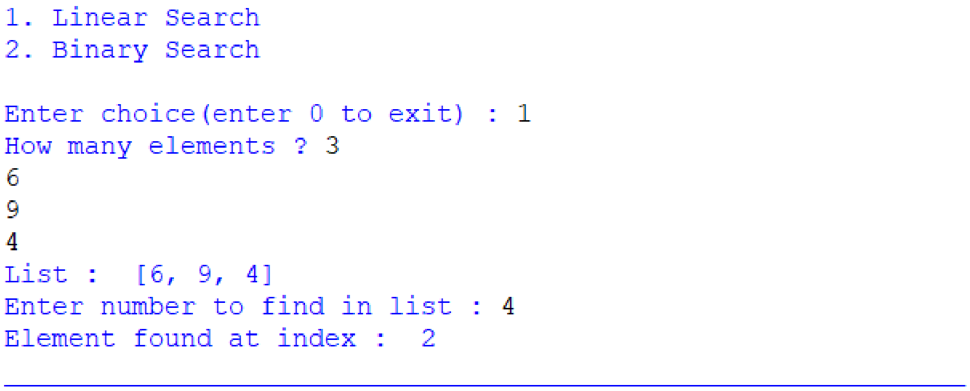


```
import time
def LSearch(list1):
   x=int(input("Enter number to find in list : "))
   found=0
   for i in range(len(list1)):
      if list1[i]==x:
         found=1
         pos=i
         break
   if found==1:
      print("Element found at index : ",pos)
   else:
      print("Element not found")

def BSort():
   global list1
   n=len(list1)
   for i in range(n-1):
      for j in range(n-i-1):
         if list1[j] > list1[j+1]:
            list1[j],list1[j+1]=list1[j+1],list1[j]
   print("List after sorting : ",list1)

def BSearch(list1):
   x=int(input("Enter number to find in list : "))
   first=0
   last=len(list1)-1
   mid=0
   found=0
   while first<=last:
      mid=int((first+last)/2)
      if list1[mid]>x:
         last=mid
      elif list1[mid]<x:
         first=mid
      elif list1[mid]==x:
         found=1
         pos=mid
         break
   if found==1:
```

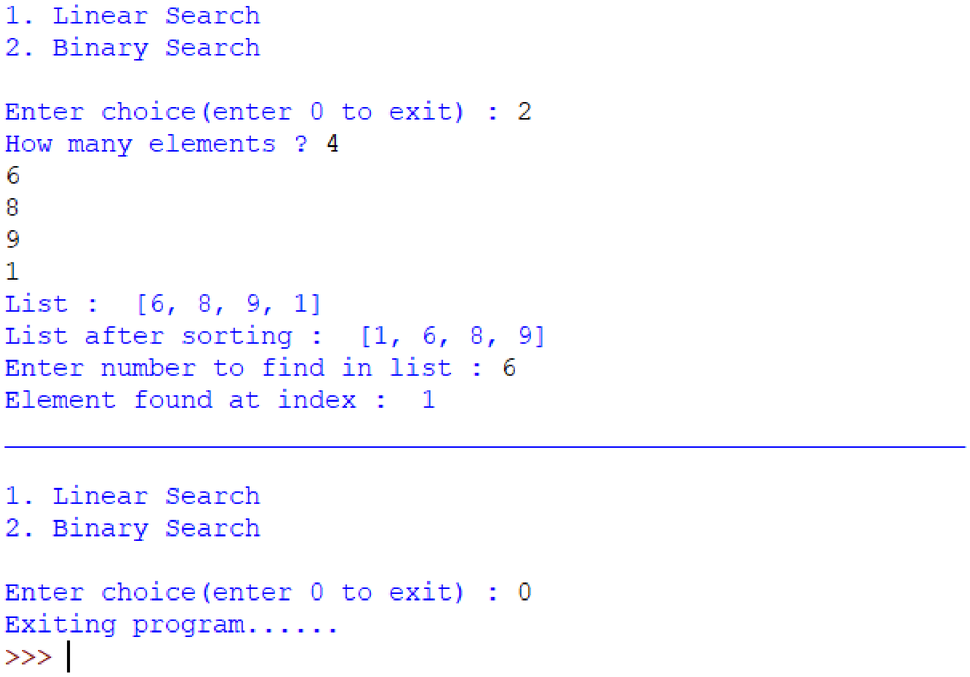


```
        print("Element found at index : ",pos)
    else:
        print("Element not found")

while True:
    print("\n1. Linear Search")
    print("2. Binary Search")
    ch=int(input("\nEnter choice(enter 0 to exit) : "))
    if ch==1:
        num=int(input("How many elements ? "))
        list1=[int(input())for i in range(num)]
        print("List : ",list1)
        LSearch(list1)
        print("____________________________________________________________")

    elif ch==2:
        num=int(input("How many elements ? "))
        list1=[int(input())for i in range(num)]
        print("List : ",list1)
        BSort()
        BSearch(list1)
        print("____________________________________________________________")

    elif ch==0:
        print("Exiting program......")
        time.sleep(1)
        exit()

    else:
        print("invalid")
        print("____________________________________________________________")
```

Output:



```
1. Linear Search
2. Binary Search

Enter choice(enter 0 to exit) : 1
How many elements ? 3
6
9
4
List :  [6, 9, 4]
Enter number to find in list : 4
Element found at index :  2
```



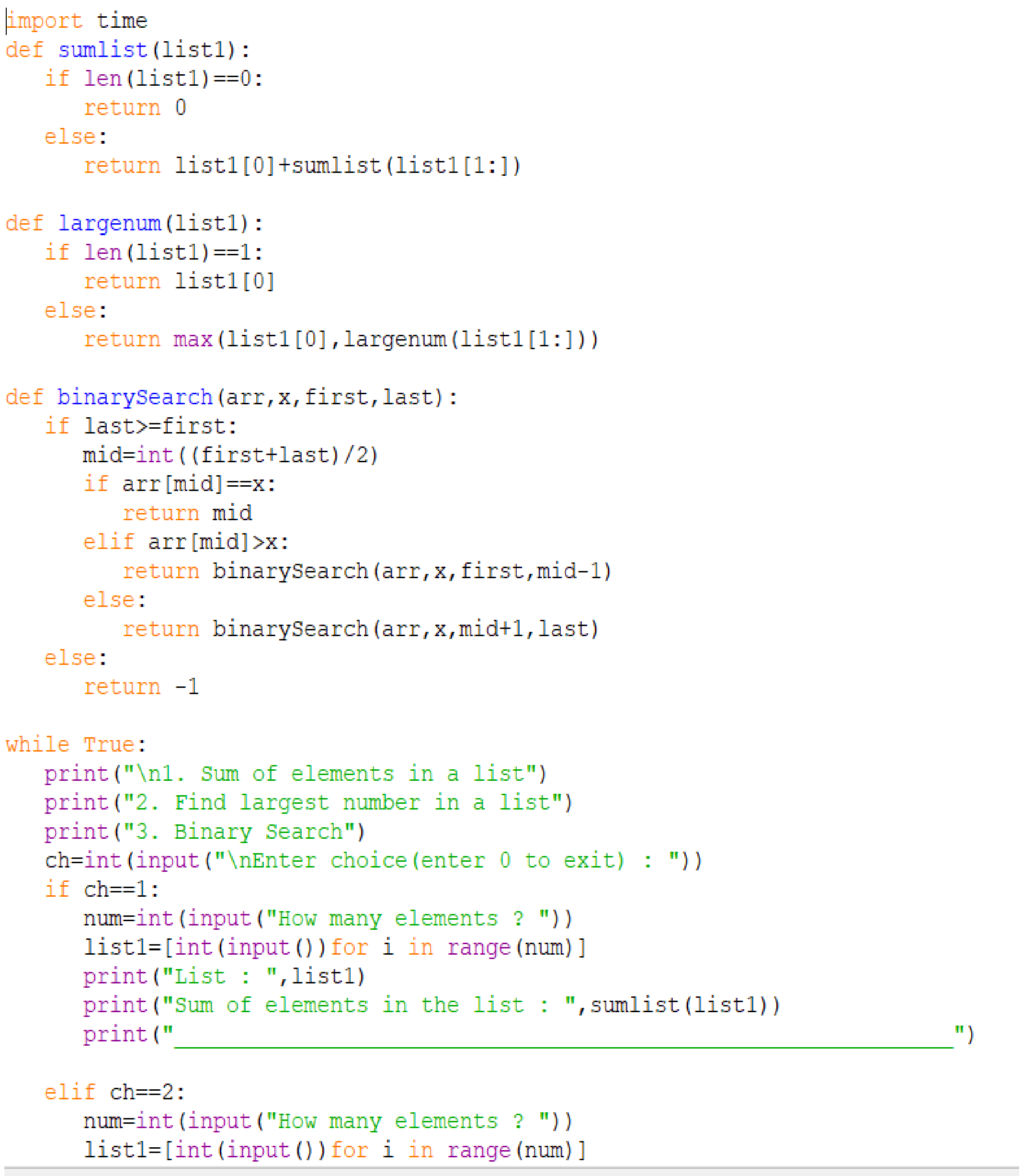
```
1. Linear Search
2. Binary Search

Enter choice(enter 0 to exit) : 2
How many elements ? 4
6
8
9
1
List :  [6, 8, 9, 1]
List after sorting :  [1, 6, 8, 9]
Enter number to find in list : 6
Element found at index :  1

1. Linear Search
2. Binary Search

Enter choice(enter 0 to exit) : 0
Exiting program......
>>> 
```

RECORD FILE –A.5

Objective: Using the concept of recursion , design a menu driven program to:
  . Find the sum of all elements of a list.
  . Find the largest element in a list
  . Search for an element in a sorted list, using Binary Search.

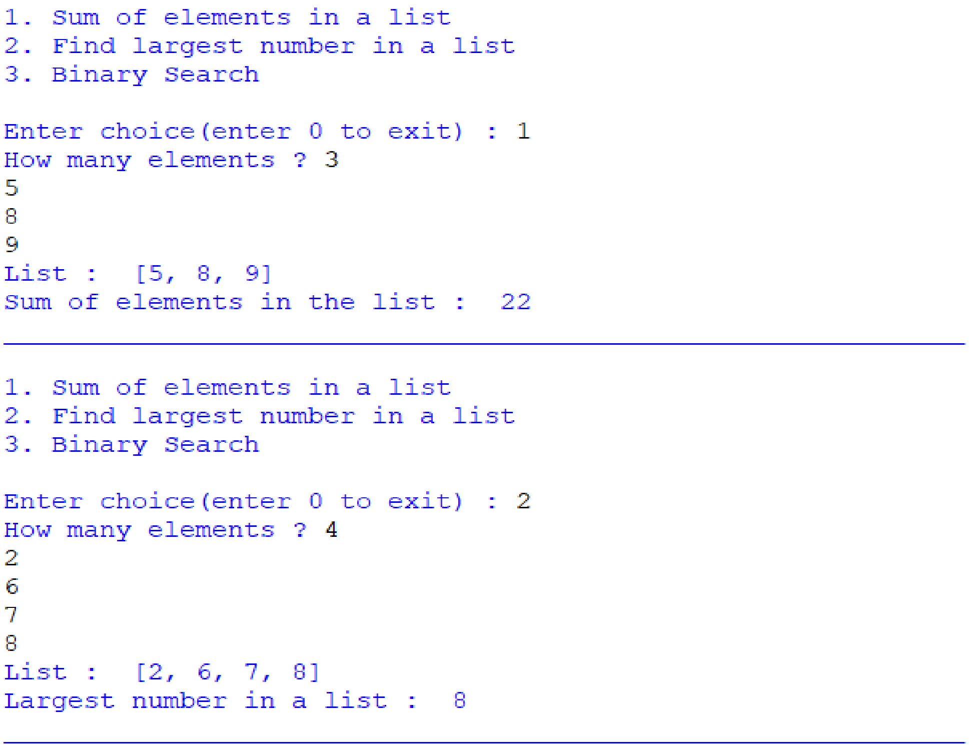
Concepts used: recursive functions.

Source code;

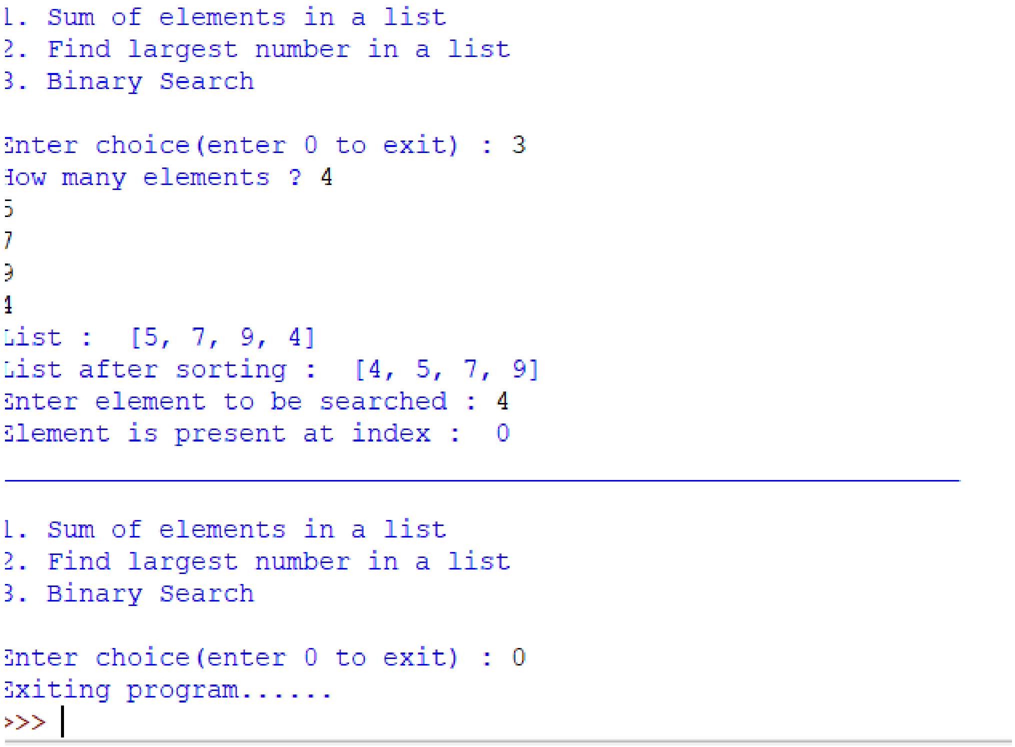


```
import time
def sumlist(list1):
   if len(list1)==0:
      return 0
   else:
      return list1[0]+sumlist(list1[1:])

def largenum(list1):
   if len(list1)==1:
      return list1[0]
   else:
      return max(list1[0],largenum(list1[1:]))

def binarySearch(arr,x,first,last):
   if last>=first:
      mid=int((first+last)/2)
      if arr[mid]==x:
         return mid
      elif arr[mid]>x:
         return binarySearch(arr,x,first,mid-1)
      else:
         return binarySearch(arr,x,mid+1,last)
   else:
      return -1

while True:
   print("\n1. Sum of elements in a list")
   print("2. Find largest number in a list")
   print("3. Binary Search")
   ch=int(input("\nEnter choice(enter 0 to exit) : "))
   if ch==1:
      num=int(input("How many elements ? "))
      list1=[int(input())for i in range(num)]
      print("List : ",list1)
      print("Sum of elements in the list : ",sumlist(list1))
      print("____________________________________________________________")

   elif ch==2:
      num=int(input("How many elements ? "))
      list1=[int(input())for i in range(num)]
      print("List : ",list1)
      print("Largest number in a list : ",largenum(list1))
      print("____________________________________________________________")

   elif ch==3:
      num=int(input("How many elements ? "))
      list1=[int(input())for i in range(num)]
      print("List : ",list1)
      list1.sort()
      print("List after sorting : ",list1)
      item=int(input("Enter element to be searched : "))
      result=binarySearch(list1,item,0,len(list1)-1)
      if result!=-1:
         print("Element is present at index : ",result)
      else:
         print("Element is not present in array")
      print("____________________________________________________________")

   elif ch==0:
      print("Exiting program......")
      time.sleep(1)
      exit()

   else:
      print("invalid")
      print("____________________________________________________________")
```

Output:



```
1. Sum of elements in a list
2. Find largest number in a list
3. Binary Search

Enter choice(enter 0 to exit) : 1
How many elements ? 3
5
8
9
List :  [5, 8, 9]
Sum of elements in the list :  22
______________________________________________________________

1. Sum of elements in a list
2. Find largest number in a list
3. Binary Search

Enter choice(enter 0 to exit) : 2
How many elements ? 4
2
6
7
8
List :  [2, 6, 7, 8]
Largest number in a list :  8
______________________________________________________________
```



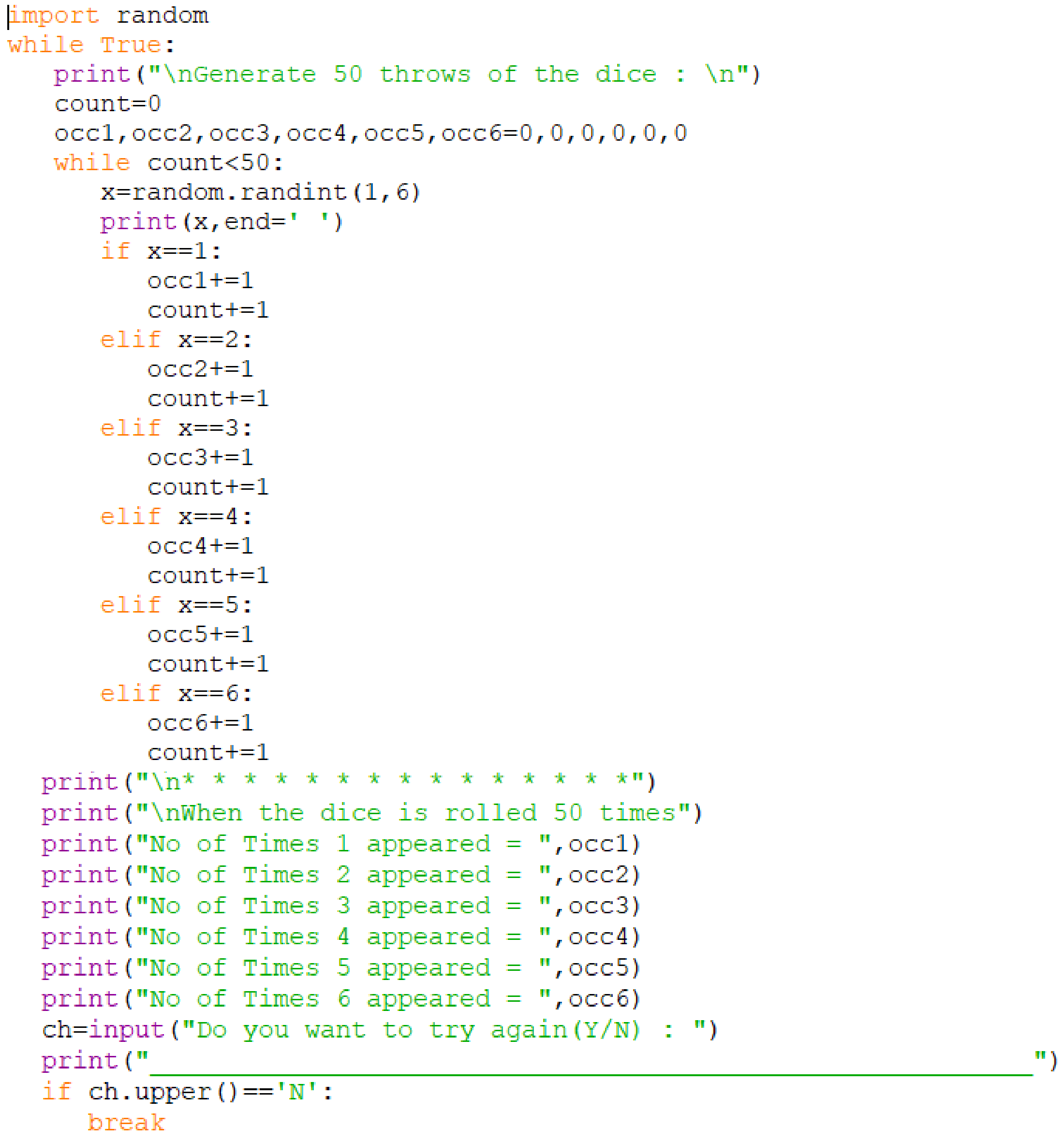
```
1. Sum of elements in a list
2. Find largest number in a list
3. Binary Search

Enter choice(enter 0 to exit) : 3
How many elements ? 4
5
7
9
4
List :  [5, 7, 9, 4]
List after sorting :  [4, 5, 7, 9]
Enter element to be searched : 4
Element is present at index :  0
______________________________________________________________

1. Sum of elements in a list
2. Find largest number in a list
3. Binary Search

Enter choice(enter 0 to exit) : 0
Exiting program......
>>> |
```

RECORD FILE- B.1

Objective: To Write a random number generator that generates random numbers between 1 and 6. (simulates a dice). If the dice throw is simulated 50 times, display the occurrences of each face of dice '1/2/3/4/5/6'.

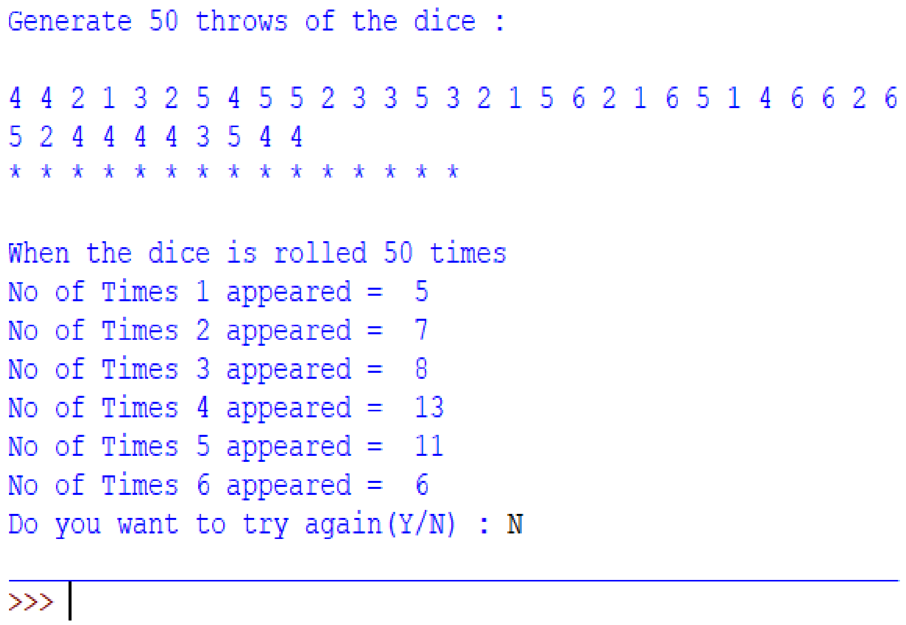
Concepts used: In built module–random.

Source code:



```
import random
while True:
    print("\nGenerate 50 throws of the dice : \n")
    count=0
    occ1,occ2,occ3,occ4,occ5,occ6=0,0,0,0,0,0
    while count<50:
        x=random.randint(1,6)
        print(x,end=' ')
        if x==1:
            occ1+=1
            count+=1
        elif x==2:
            occ2+=1
            count+=1
        elif x==3:
            occ3+=1
            count+=1
        elif x==4:
            occ4+=1
            count+=1
        elif x==5:
            occ5+=1
            count+=1
        elif x==6:
            occ6+=1
            count+=1
    print("\n* * * * * * * * * * * * * * *")
    print("\nWhen the dice is rolled 50 times")
    print("No of Times 1 appeared = ",occ1)
    print("No of Times 2 appeared = ",occ2)
    print("No of Times 3 appeared = ",occ3)
    print("No of Times 4 appeared = ",occ4)
    print("No of Times 5 appeared = ",occ5)
    print("No of Times 6 appeared = ",occ6)
    ch=input("Do you want to try again(Y/N) : ")
    print("_________________________________________________________")
    if ch.upper()=='N':
        break
```

Output:



```
Generate 50 throws of the dice :

4 4 2 1 3 2 5 4 5 5 2 3 3 5 3 2 1 5 6 2 1 6 5 1 4 6 6 2 6 4 4 3 3 6 3 1 5 5 4 5
5 2 4 4 4 4 3 5 4 4
* * * * * * * * * * * * * * *

When the dice is rolled 50 times
No of Times 1 appeared =  5
No of Times 2 appeared =  7
No of Times 3 appeared =  8
No of Times 4 appeared =  13
No of Times 5 appeared =  11
No of Times 6 appeared =  6
Do you want to try again(Y/N) : N
>>> 
```

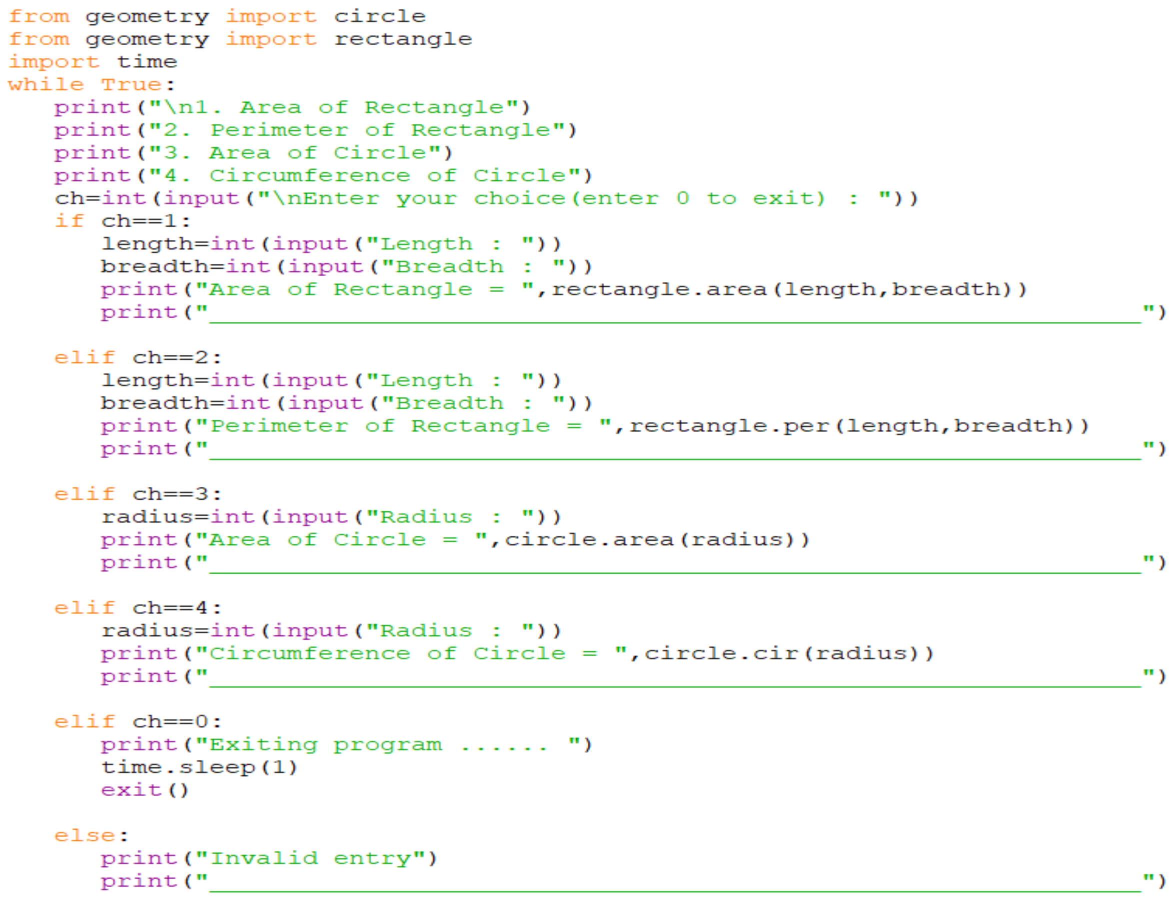
RECORD FILE- B.2

Objective: To Create a package ‘Geometry’ which contains user defined modules ‘Rectangle’ and ‘Circle’. Module ‘Rectangle’ contains functions to calculate area & perimeter of Rectangle. Module ‘Circle’ contains functions to calculate area & circumference of Circle. Write a Python program that calls these functions to display the:

. Area of Rectangle
. Perimeter of Rectangle
. Area of Circle
. Circumference of Circle.

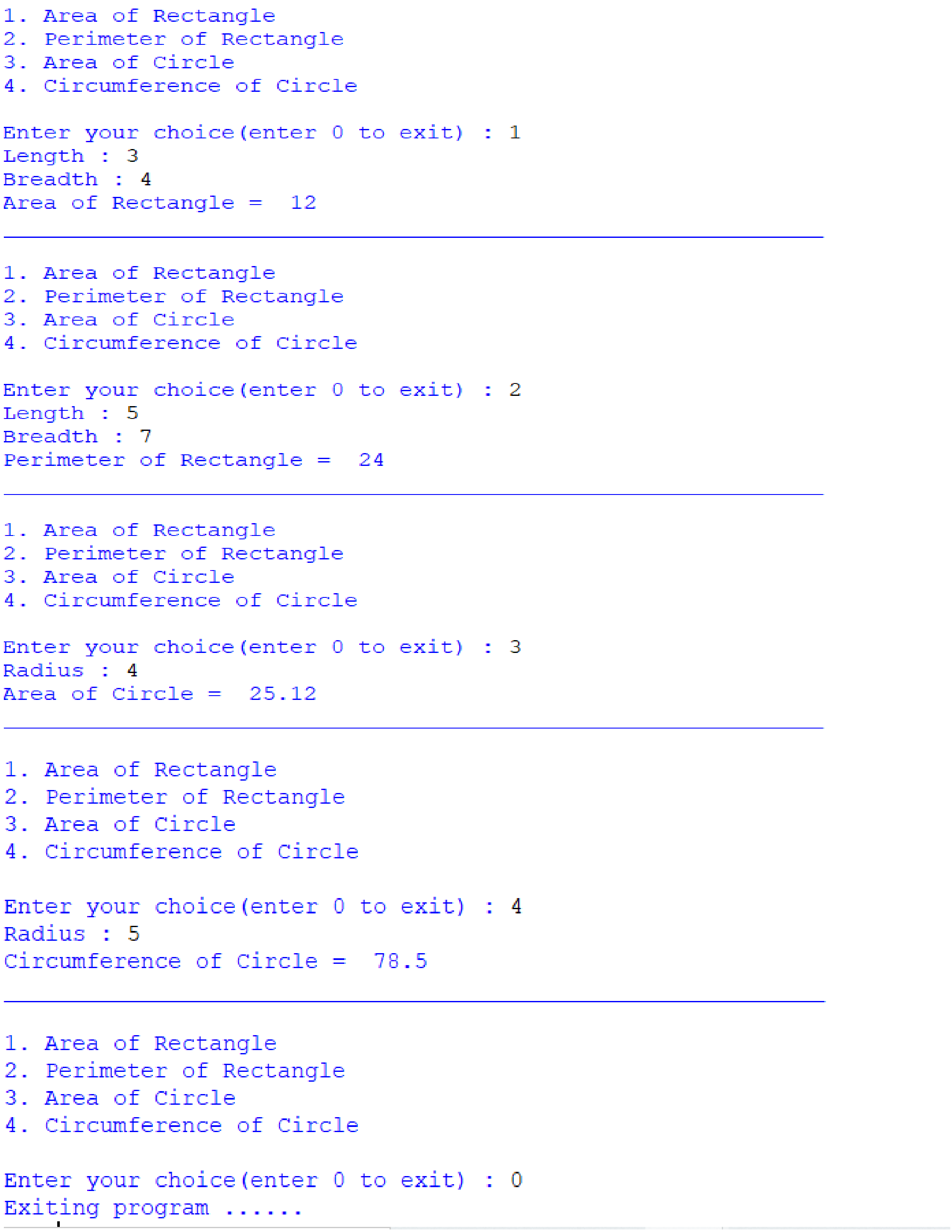
Concepts used: User defined module & package, Inbuilt module –math.

Source code:



```
from geometry import circle
from geometry import rectangle
import time
while True:
   print("\n1. Area of Rectangle")
   print("2. Perimeter of Rectangle")
   print("3. Area of Circle")
   print("4. Circumference of Circle")
   ch=int(input("\nEnter your choice(enter 0 to exit) : "))
   if ch==1:
      length=int(input("Length : "))
      breadth=int(input("Breadth : "))
      print("Area of Rectangle = ",rectangle.area(length,breadth))
      print("____________________________________________________________")

   elif ch==2:
      length=int(input("Length : "))
      breadth=int(input("Breadth : "))
      print("Perimeter of Rectangle = ",rectangle.per(length,breadth))
      print("____________________________________________________________")

   elif ch==3:
      radius=int(input("Radius : "))
      print("Area of Circle = ",circle.area(radius))
      print("____________________________________________________________")

   elif ch==4:
      radius=int(input("Radius : "))
      print("Circumference of Circle = ",circle.cir(radius))
      print("____________________________________________________________")

   elif ch==0:
      print("Exiting program ...... ")
      time.sleep(1)
      exit()

   else:
      print("Invalid entry")
      print("____________________________________________________________")
```

Output:



```
1. Area of Rectangle
2. Perimeter of Rectangle
3. Area of Circle
4. Circumference of Circle

Enter your choice(enter 0 to exit) : 1
Length : 3
Breadth : 4
Area of Rectangle =  12
____________________________________________________________

1. Area of Rectangle
2. Perimeter of Rectangle
3. Area of Circle
4. Circumference of Circle

Enter your choice(enter 0 to exit) : 2
Length : 5
Breadth : 7
Perimeter of Rectangle =  24
____________________________________________________________

1. Area of Rectangle
2. Perimeter of Rectangle
3. Area of Circle
4. Circumference of Circle

Enter your choice(enter 0 to exit) : 3
Radius : 4
Area of Circle =  25.12
____________________________________________________________

1. Area of Rectangle
2. Perimeter of Rectangle
3. Area of Circle
4. Circumference of Circle

Enter your choice(enter 0 to exit) : 4
Radius : 5
Circumference of Circle =  78.5
____________________________________________________________

1. Area of Rectangle
2. Perimeter of Rectangle
3. Area of Circle
4. Circumference of Circle

Enter your choice(enter 0 to exit) : 0
Exiting program ......
```

RECORD FILE- B.3

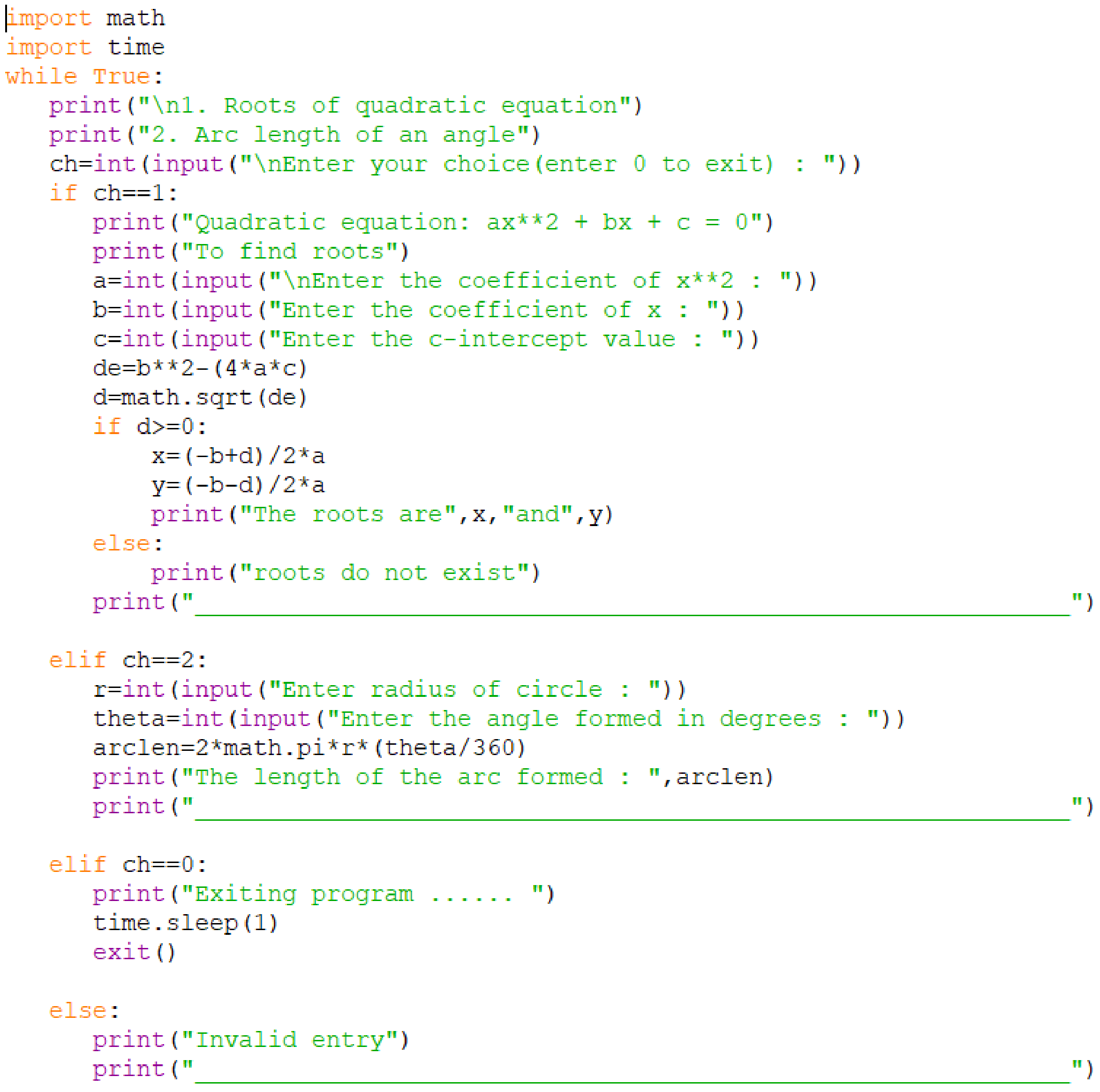
Objective: to Write a menu driven program that:

. Finds the roots of a quadratic function, ax2 + bx + c = 0

. Calculates arc length of an angle , given the angle & radius of Circle.

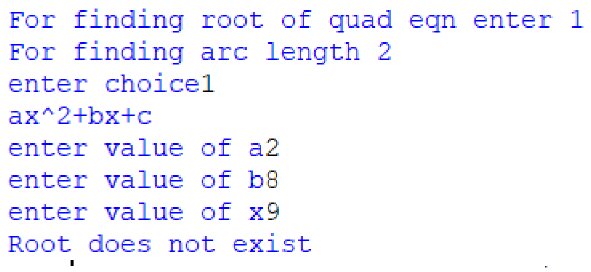
Concepts used: User defined Function, Inbuilt module – math.

Source code:

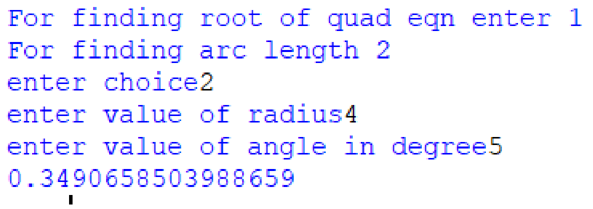


```python
import math
import time
while True:
   print("\n1. Roots of quadratic equation")
   print("2. Arc length of an angle")
   ch=int(input("\nEnter your choice(enter 0 to exit) : "))
   if ch==1:
      print("Quadratic equation: ax**2 + bx + c = 0")
      print("To find roots")
      a=int(input("\nEnter the coefficient of x**2 : "))
      b=int(input("Enter the coefficient of x : "))
      c=int(input("Enter the c-intercept value : "))
      de=b**2-(4*a*c)
      d=math.sqrt(de)
      if d>=0:
          x=(-b+d)/2*a
          y=(-b-d)/2*a
          print("The roots are",x,"and",y)
      else:
          print("roots do not exist")
      print("____________________________________________________________")

   elif ch==2:
      r=int(input("Enter radius of circle : "))
      theta=int(input("Enter the angle formed in degrees : "))
      arclen=2*math.pi*r*(theta/360)
      print("The length of the arc formed : ",arclen)
      print("____________________________________________________________")

   elif ch==0:
      print("Exiting program ...... ")
      time.sleep(1)
      exit()

   else:
      print("Invalid entry")
      print("____________________________________________________________")
```

Output:



```
For finding root of quad eqn enter 1
For finding arc length 2
enter choice1
ax^2+bx+c
enter value of a2
enter value of b8
enter value of x9
Root does not exist
```



```
For finding root of quad eqn enter 1
For finding arc length 2
enter choice2
enter value of radius4
enter value of angle in degree5
0.3490658503988659
```

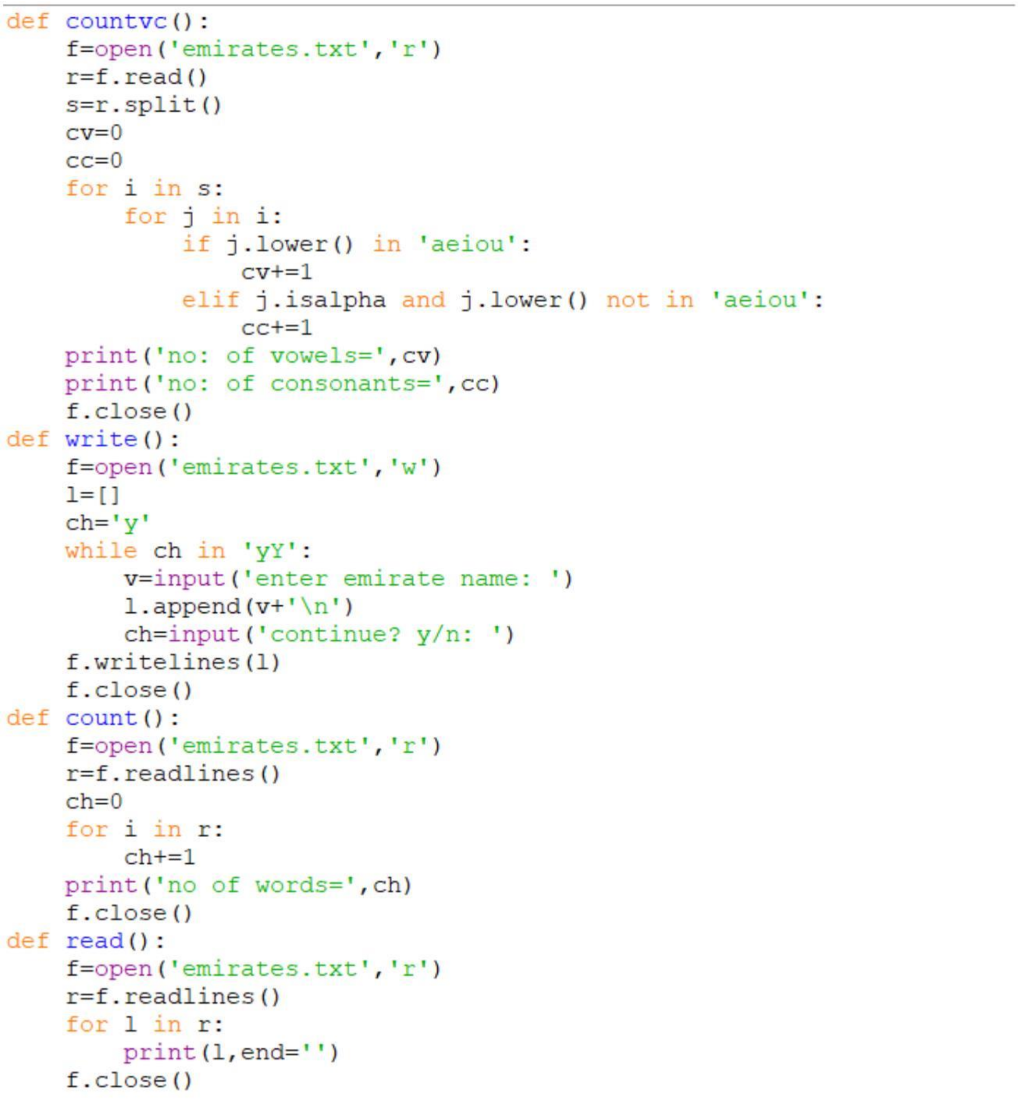
RECORD FILE–C.1

Objective : Write a menu driven program to:

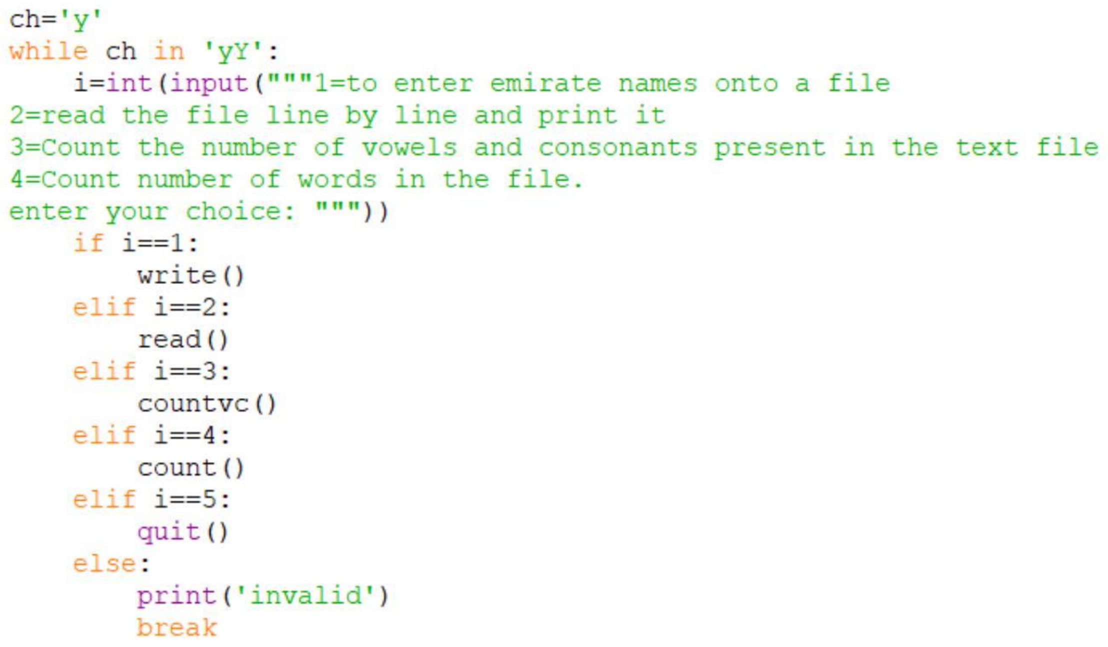
. Create a Text File ‘Emirates’ and enter the names of 7 Emirates.
. Read the file line by line and print it.
. Count the number of vowels and consonants present in the text file.
. Count number of words in the file.

Concept Used : File Handling – Text Files.

Source code:

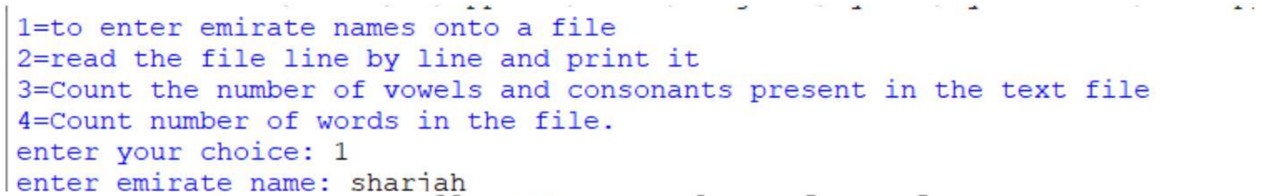


```
def countvc():
    f=open('emirates.txt','r')
    r=f.read()
    s=r.split()
    cv=0
    cc=0
    for i in s:
        for j in i:
            if j.lower() in 'aeiou':
                cv+=1
            elif j.isalpha and j.lower() not in 'aeiou':
                cc+=1
    print('no: of vowels=',cv)
    print('no: of consonants=',cc)
    f.close()
def write():
    f=open('emirates.txt','w')
    l=[]
    ch='y'
    while ch in 'yY':
        v=input('enter emirate name: ')
        l.append(v+'\n')
        ch=input('continue? y/n: ')
    f.writelines(l)
    f.close()
def count():
    f=open('emirates.txt','r')
    r=f.readlines()
    ch=0
    for i in r:
        ch+=1
    print('no of words=',ch)
    f.close()
def read():
    f=open('emirates.txt','r')
    r=f.readlines()
    for l in r:
        print(l,end='')
    f.close()
```

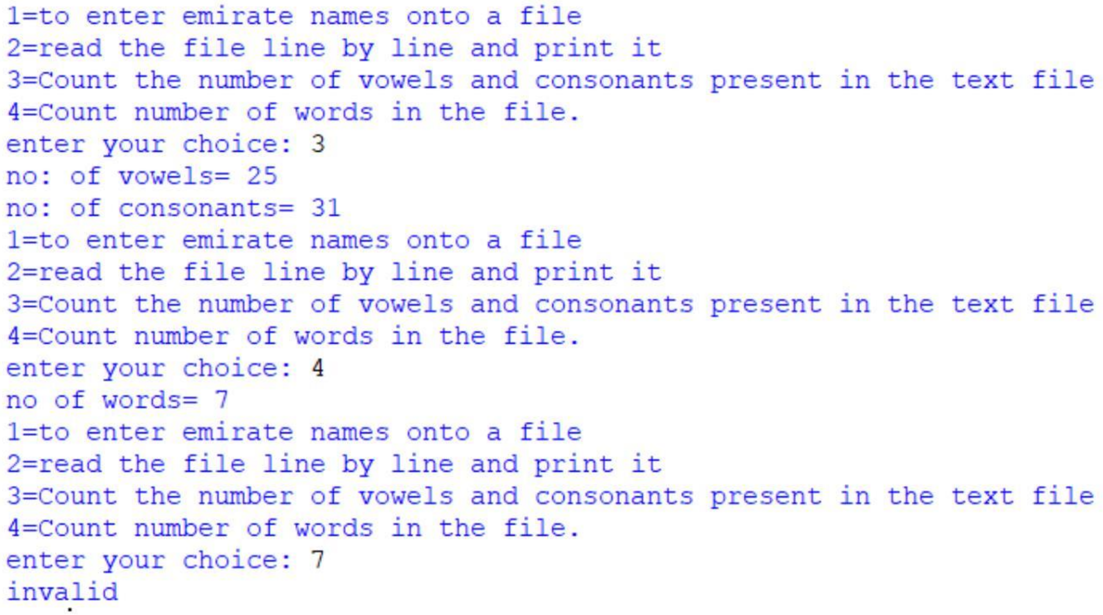


```
ch='y'
while ch in 'yY':
    i=int(input("""1=to enter emirate names onto a file
2=read the file line by line and print it
3=Count the number of vowels and consonants present in the text file
4=Count number of words in the file.
enter your choice: """))
    if i==1:
        write()
    elif i==2:
        read()
    elif i==3:
        countvc()
    elif i==4:
        count()
    elif i==5:
        quit()
    else:
        print('invalid')
        break
```

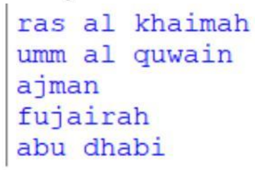
Output:



```
1=to enter emirate names onto a file
2=read the file line by line and print it
3=Count the number of vowels and consonants present in the text file
4=Count number of words in the file.
enter your choice: 1
enter emirate name: shariah
```

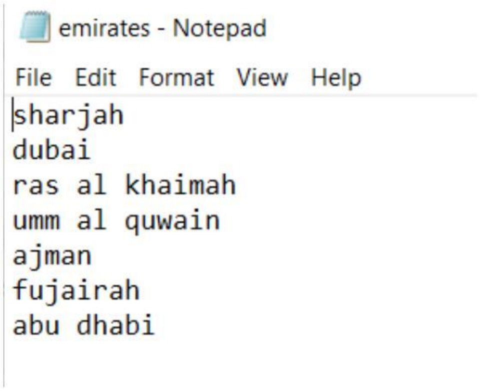


```
1=to enter emirate names onto a file
2=read the file line by line and print it
3=Count the number of vowels and consonants present in the text file
4=Count number of words in the file.
enter your choice: 3
no: of vowels= 25
no: of consonants= 31
1=to enter emirate names onto a file
2=read the file line by line and print it
3=Count the number of vowels and consonants present in the text file
4=Count number of words in the file.
enter your choice: 4
no of words= 7
1=to enter emirate names onto a file
2=read the file line by line and print it
3=Count the number of vowels and consonants present in the text file
4=Count number of words in the file.
enter your choice: 7
invalid
```



```
ras al khaimah
umm al quwain
ajman
fujairah
abu dhabi
```

Notepad:



```
sharjah
dubai
ras al khaimah
umm al quwain
ajman
fujairah
abu dhabi
```

RECORD FILE - C.2

Objective: Write a menu driven program to:

. Count the no of occurrence of word ‘I’

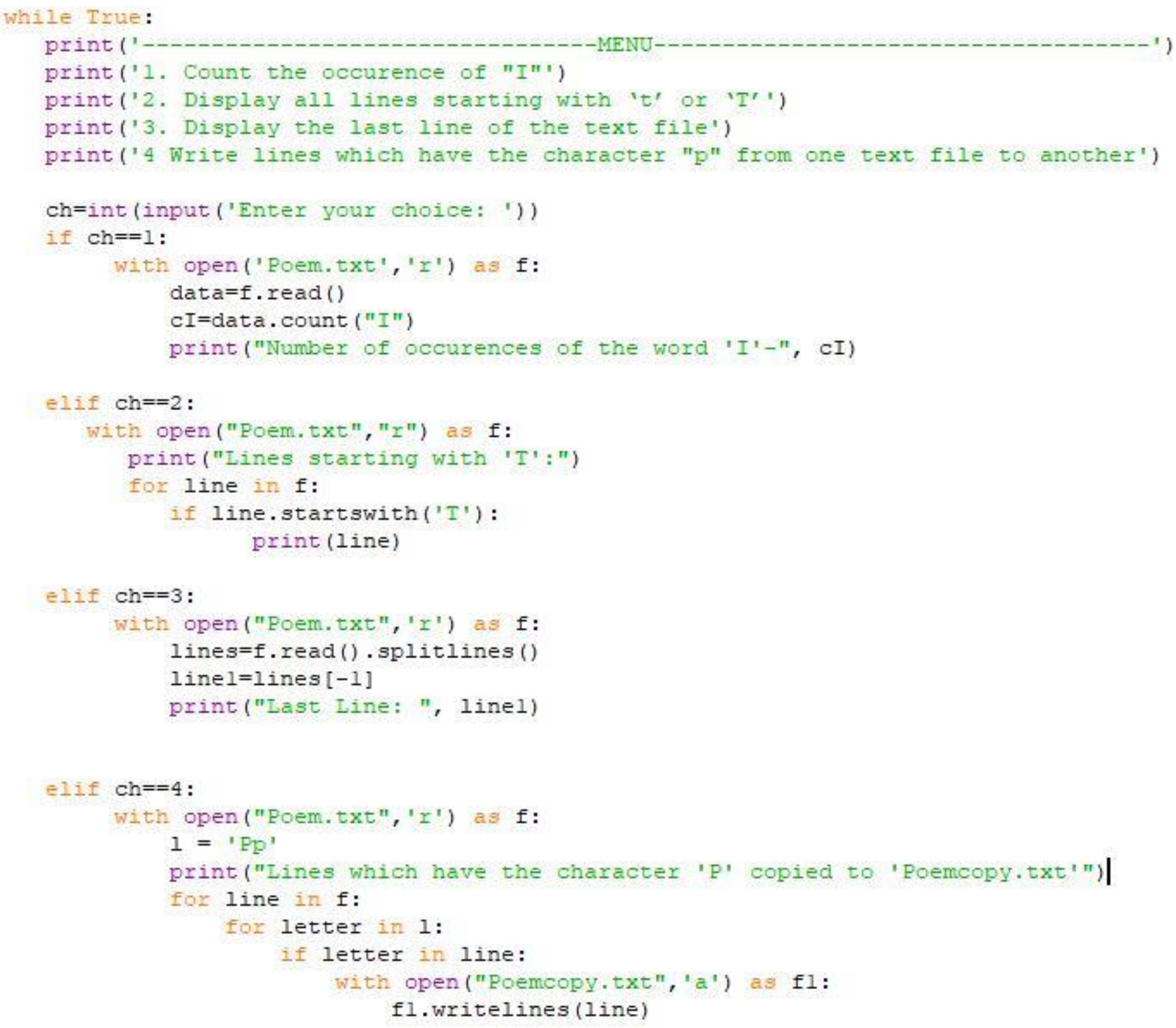
. Display all line starting with ‘t’ or ‘T’.

. Display the last line of the text file.

. Write those lines which have the character 'p' from one text file to another text file.

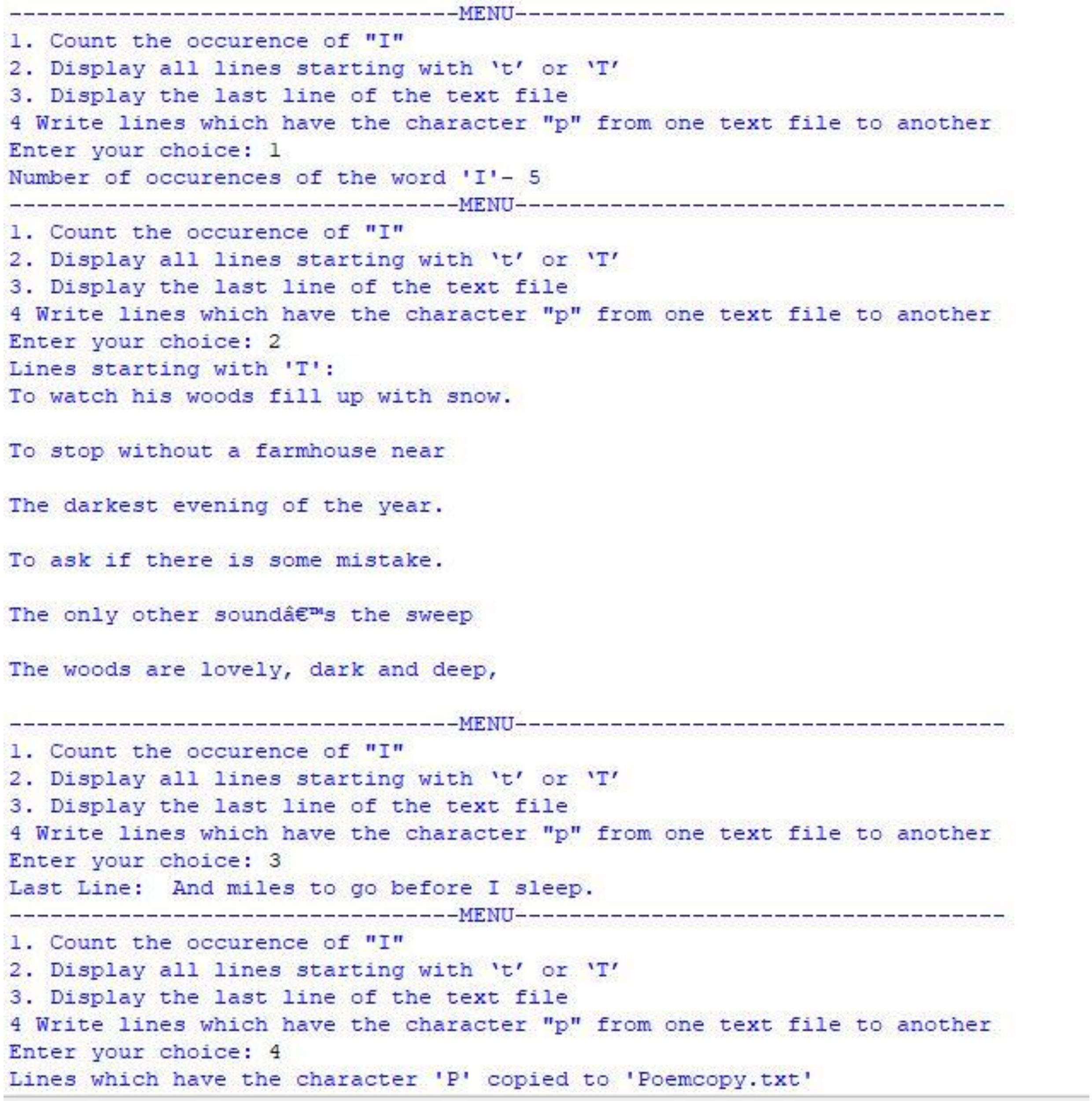
Concept Used: File Handling – Text Files.

Source code:



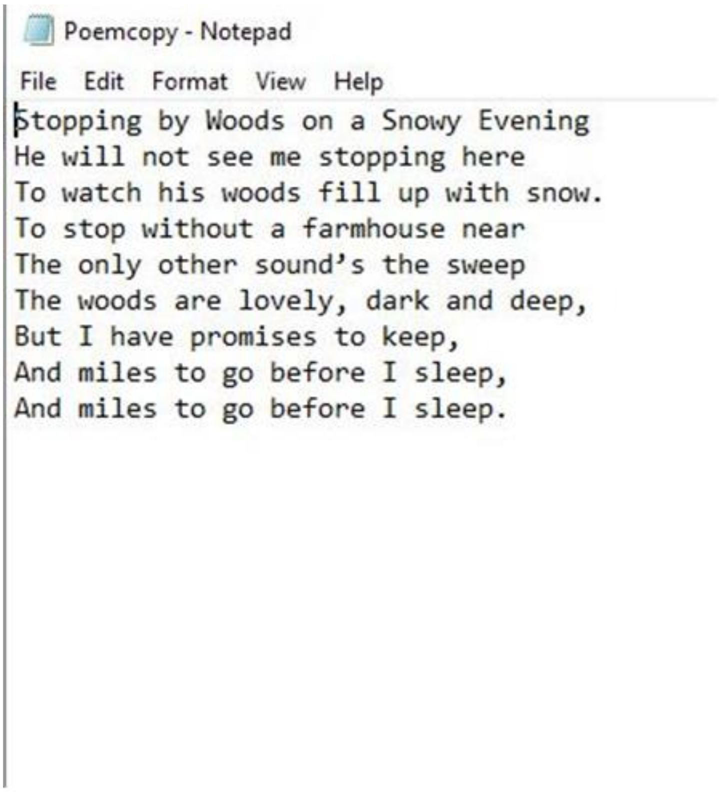
```
while True:
   print('---------------------------------MENU------------------------------------')
   print('1. Count the occurence of "I"')
   print('2. Display all lines starting with ‘t’ or ‘T’')
   print('3. Display the last line of the text file')
   print('4 Write lines which have the character "p" from one text file to another')

   ch=int(input('Enter your choice: '))
   if ch==1:
        with open('Poem.txt','r') as f:
            data=f.read()
            cI=data.count("I")
            print("Number of occurences of the word 'I'-", cI)

   elif ch==2:
      with open("Poem.txt","r") as f:
         print("Lines starting with 'T':")
         for line in f:
            if line.startswith('T'):
                  print(line)

   elif ch==3:
        with open("Poem.txt",'r') as f:
            lines=f.read().splitlines()
            line1=lines[-1]
            print("Last Line: ", line1)


   elif ch==4:
        with open("Poem.txt",'r') as f:
            l = 'Pp'
            print("Lines which have the character 'P' copied to 'Poemcopy.txt'")
            for line in f:
                for letter in l:
                    if letter in line:
                        with open("Poemcopy.txt",'a') as f1:
                            f1.writelines(line)
```

Output:



```
---------------------------------MENU-------------------------------------
1. Count the occurence of "I"
2. Display all lines starting with ‘t’ or ‘T’
3. Display the last line of the text file
4 Write lines which have the character "p" from one text file to another
Enter your choice: 1
Number of occurences of the word 'I'- 5
---------------------------------MENU-------------------------------------
1. Count the occurence of "I"
2. Display all lines starting with ‘t’ or ‘T’
3. Display the last line of the text file
4 Write lines which have the character "p" from one text file to another
Enter your choice: 2
Lines starting with 'T':
To watch his woods fill up with snow.

To stop without a farmhouse near

The darkest evening of the year.

To ask if there is some mistake.

The only other soundâ€™s the sweep

The woods are lovely, dark and deep,

---------------------------------MENU-------------------------------------
1. Count the occurence of "I"
2. Display all lines starting with ‘t’ or ‘T’
3. Display the last line of the text file
4 Write lines which have the character "p" from one text file to another
Enter your choice: 3
Last Line:  And miles to go before I sleep.
---------------------------------MENU-------------------------------------
1. Count the occurence of "I"
2. Display all lines starting with ‘t’ or ‘T’
3. Display the last line of the text file
4 Write lines which have the character "p" from one text file to another
Enter your choice: 4
Lines which have the character 'P' copied to 'Poemcopy.txt'
```

Text:



```
Stopping by Woods on a Snowy Evening
He will not see me stopping here
To watch his woods fill up with snow.
To stop without a farmhouse near
The only other sound's the sweep
The woods are lovely, dark and deep,
But I have promises to keep,
And miles to go before I sleep,
And miles to go before I sleep.
```

RECORD FILE– C.3

Objective: Write a menu driven program to make the following changes in a Text File ‘Notes.txt’:

. Replace the occurrence of a character with ‘*’.

. Count the number of a particular word.

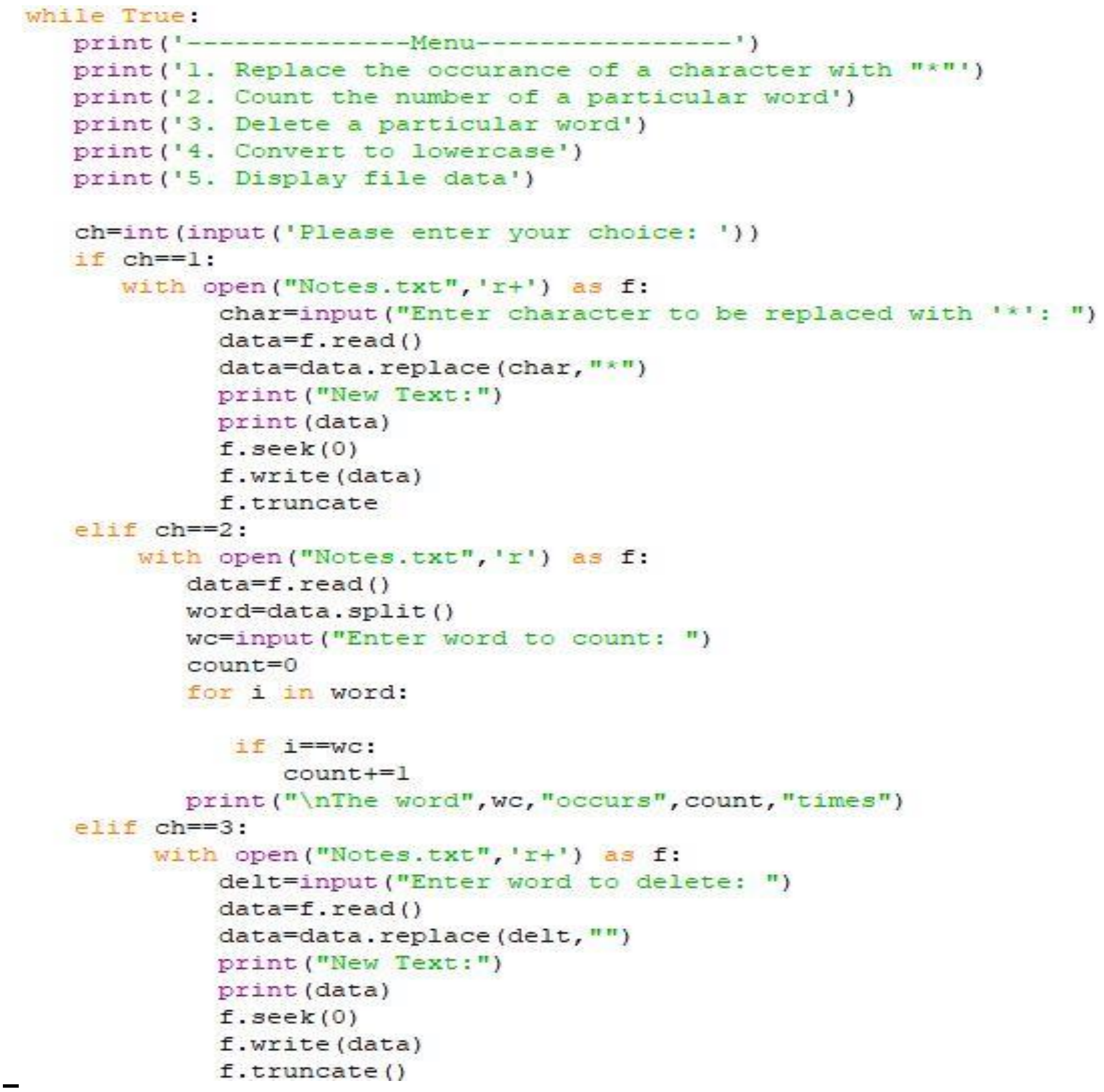
. Delete a particular word.

. Convert to lowercase.

. Display File Data

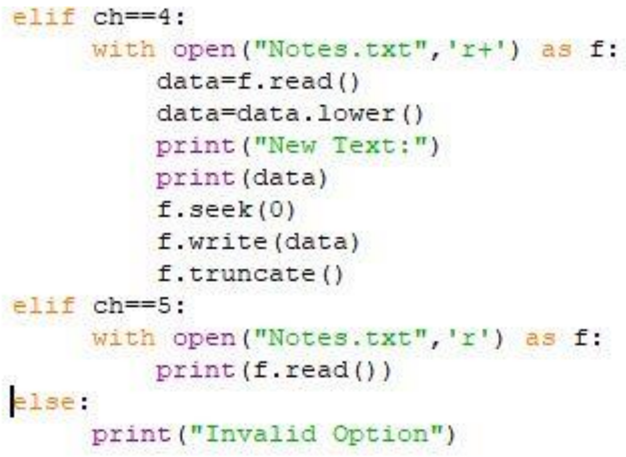
Concept Used: File Handling – Text Files

Source code:



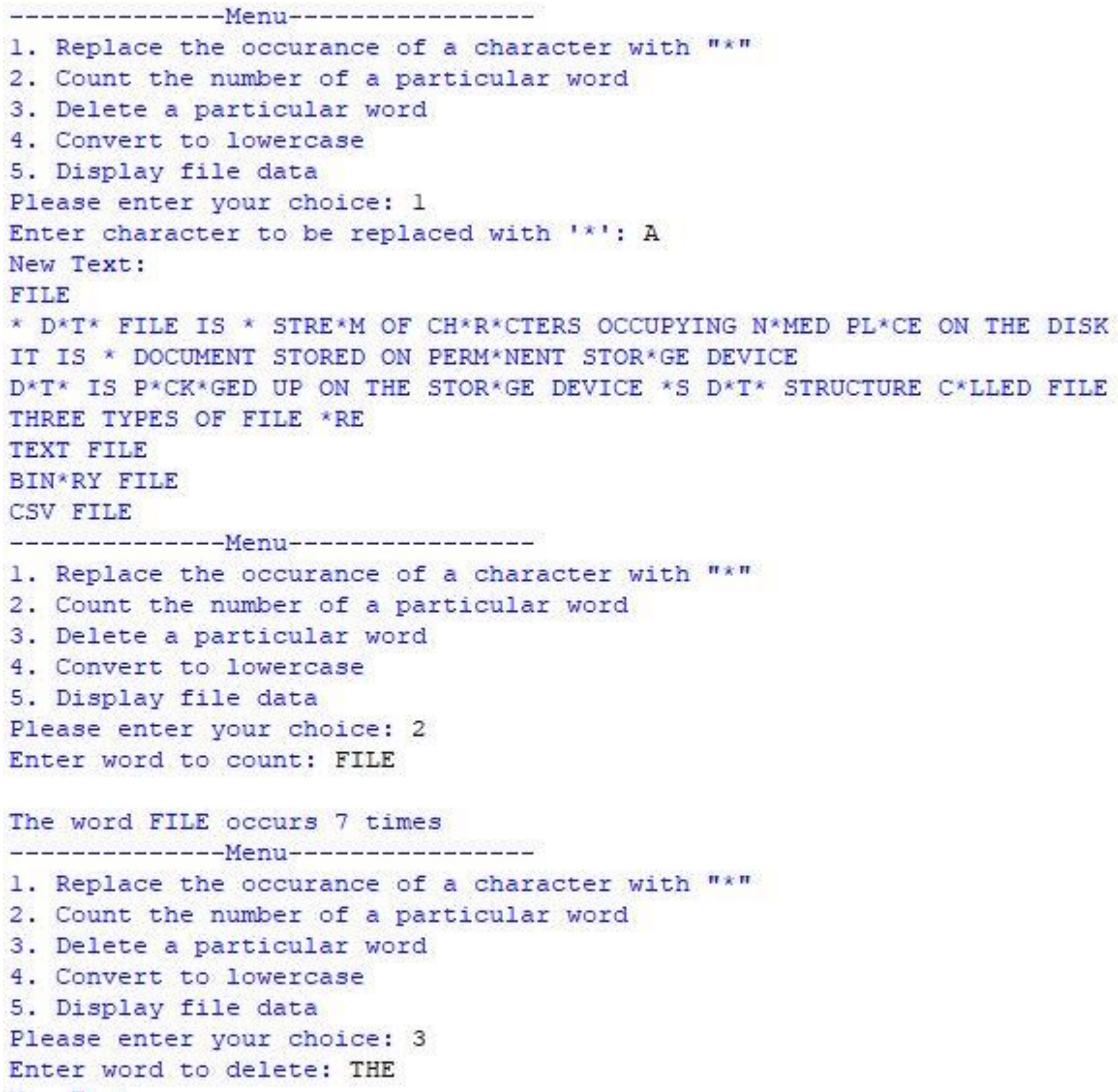
```
while True:
    print('---------------Menu------------------')
    print('1. Replace the occurance of a character with "*"')
    print('2. Count the number of a particular word')
    print('3. Delete a particular word')
    print('4. Convert to lowercase')
    print('5. Display file data')

    ch=int(input('Please enter your choice: '))
    if ch==1:
        with open("Notes.txt",'r+') as f:
                char=input("Enter character to be replaced with '*': ")
                data=f.read()
                data=data.replace(char,"*")
                print("New Text:")
                print(data)
                f.seek(0)
                f.write(data)
                f.truncate
    elif ch==2:
         with open("Notes.txt",'r') as f:
             data=f.read()
             word=data.split()
             wc=input("Enter word to count: ")
             count=0
             for i in word:

                 if i==wc:
                     count+=1
             print("\nThe word",wc,"occurs",count,"times")
    elif ch==3:
          with open("Notes.txt",'r+') as f:
                delt=input("Enter word to delete: ")
                data=f.read()
                data=data.replace(delt,"")
                print("New Text:")
                print(data)
                f.seek(0)
                f.write(data)
                f.truncate()
```

–

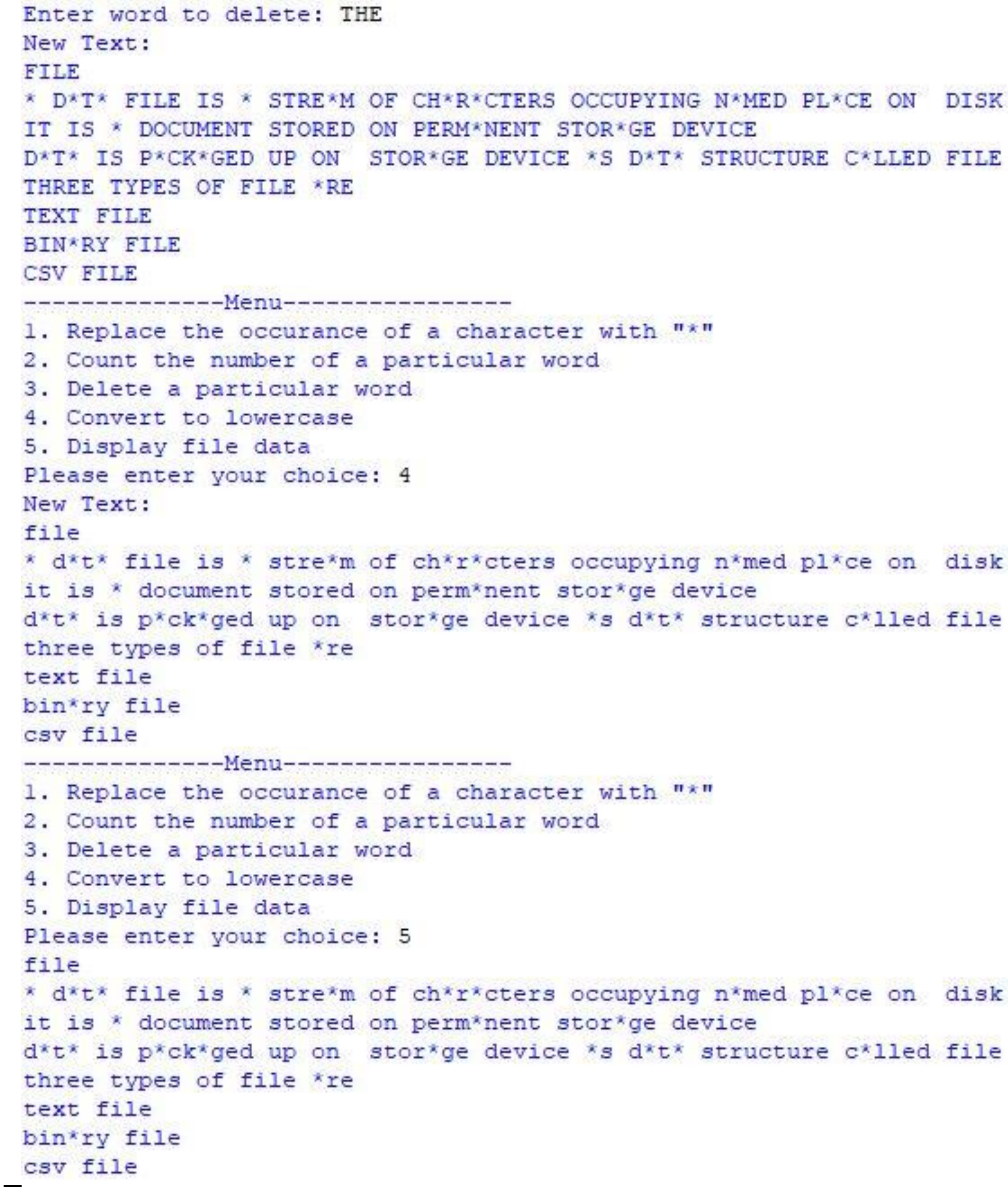


```
elif ch==4:
     with open("Notes.txt",'r+') as f:
         data=f.read()
         data=data.lower()
         print("New Text:")
         print(data)
         f.seek(0)
         f.write(data)
         f.truncate()
elif ch==5:
     with open("Notes.txt",'r') as f:
         print(f.read())
else:
     print("Invalid Option")
```

Output:



```
--------------Menu----------------
1. Replace the occurance of a character with "*"
2. Count the number of a particular word
3. Delete a particular word
4. Convert to lowercase
5. Display file data
Please enter your choice: 1
Enter character to be replaced with '*': A
New Text:
FILE
* D*T* FILE IS * STRE*M OF CH*R*CTERS OCCUPYING N*MED PL*CE ON THE DISK
IT IS * DOCUMENT STORED ON PERM*NENT STOR*GE DEVICE
D*T* IS P*CK*GED UP ON THE STOR*GE DEVICE *S D*T* STRUCTURE C*LLED FILE
THREE TYPES OF FILE *RE
TEXT FILE
BIN*RY FILE
CSV FILE
--------------Menu----------------
1. Replace the occurance of a character with "*"
2. Count the number of a particular word
3. Delete a particular word
4. Convert to lowercase
5. Display file data
Please enter your choice: 2
Enter word to count: FILE

The word FILE occurs 7 times
--------------Menu----------------
1. Replace the occurance of a character with "*"
2. Count the number of a particular word
3. Delete a particular word
4. Convert to lowercase
5. Display file data
Please enter your choice: 3
Enter word to delete: THE
```



```
Enter word to delete: THE
New Text:
FILE
* D*T* FILE IS * STRE*M OF CH*R*CTERS OCCUPYING N*MED PL*CE ON  DISK
IT IS * DOCUMENT STORED ON PERM*NENT STOR*GE DEVICE
D*T* IS P*CK*GED UP ON  STOR*GE DEVICE *S D*T* STRUCTURE C*LLED FILE
THREE TYPES OF FILE *RE
TEXT FILE
BIN*RY FILE
CSV FILE
--------------Menu----------------
1. Replace the occurance of a character with "*"
2. Count the number of a particular word
3. Delete a particular word
4. Convert to lowercase
5. Display file data
Please enter your choice: 4
New Text:
file
* d*t* file is * stre*m of ch*r*cters occupying n*med pl*ce on  disk
it is * document stored on perm*nent stor*ge device
d*t* is p*ck*ged up on  stor*ge device *s d*t* structure c*lled file
three types of file *re
text file
bin*ry file
csv file
--------------Menu----------------
1. Replace the occurance of a character with "*"
2. Count the number of a particular word
3. Delete a particular word
4. Convert to lowercase
5. Display file data
Please enter your choice: 5
file
* d*t* file is * stre*m of ch*r*cters occupying n*med pl*ce on  disk
it is * document stored on perm*nent stor*ge device
d*t* is p*ck*ged up on  stor*ge device *s d*t* structure c*lled file
three types of file *re
text file
bin*ry file
csv file
```

RECORD FILE – C.4

Objective: The file ‘Student’ contains the following information : Roll number, Name and Mark.

. Display all Details

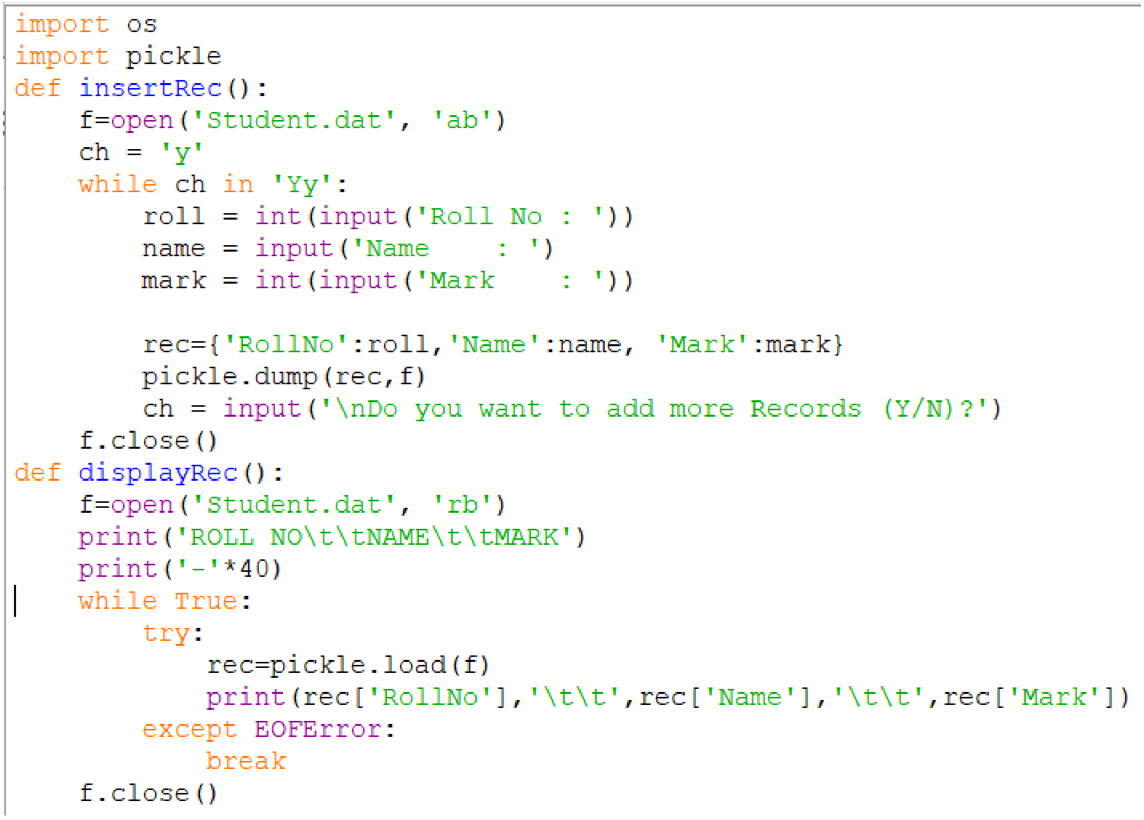
. Append records to the file.

. Search by Roll No and display details.

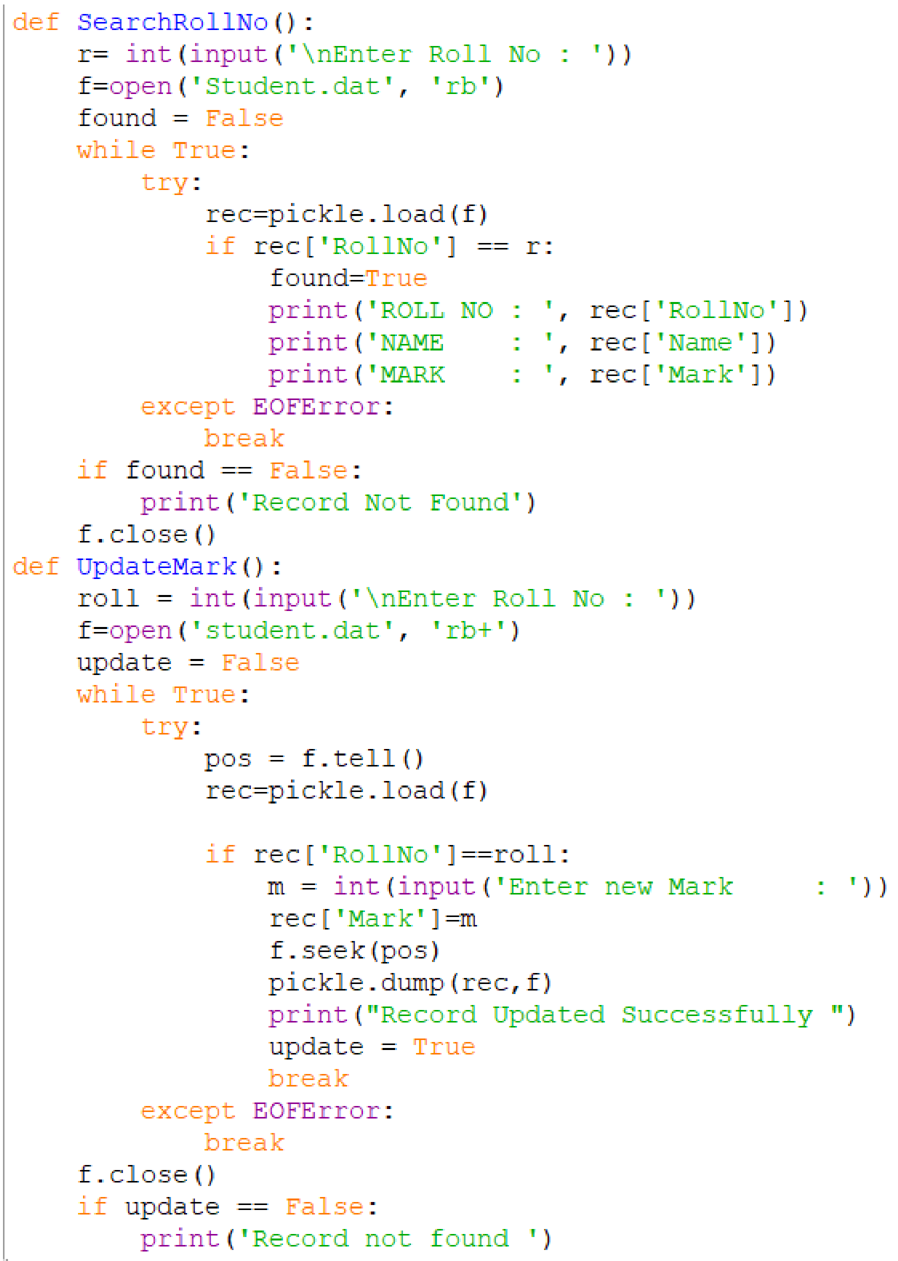
. Search by Roll No and Update Marks.

Concept Used: File Handling – Binary Files ; Dictionary.

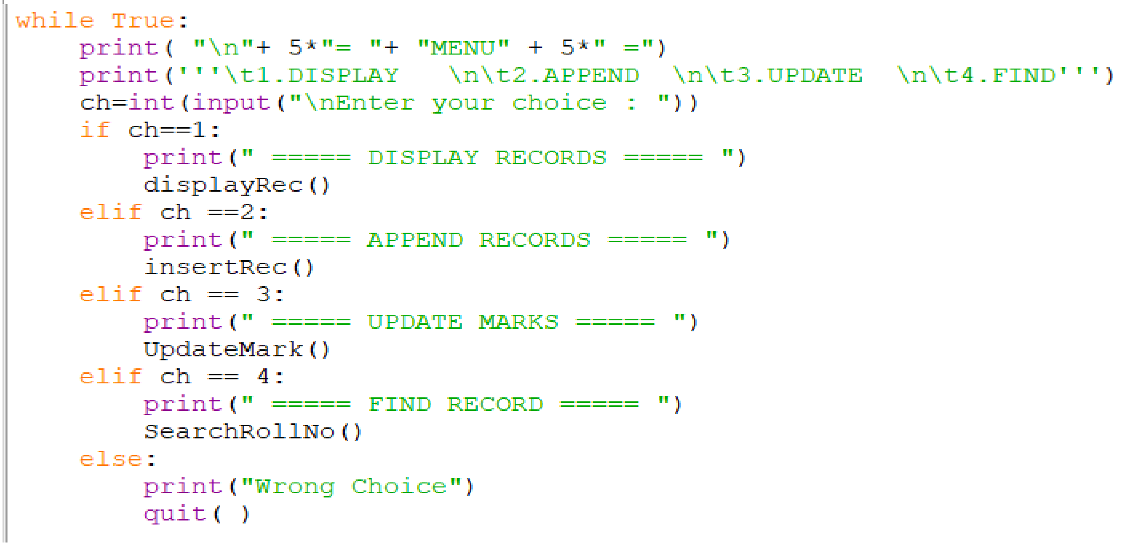
Source code:



```
import os
import pickle
def insertRec():
    f=open('Student.dat', 'ab')
    ch = 'y'
    while ch in 'Yy':
        roll = int(input('Roll No : '))
        name = input('Name    : ')
        mark = int(input('Mark    : '))

        rec={'RollNo':roll,'Name':name, 'Mark':mark}
        pickle.dump(rec,f)
        ch = input('\nDo you want to add more Records (Y/N)?')
    f.close()
def displayRec():
    f=open('Student.dat', 'rb')
    print('ROLL NO\t\tNAME\t\tMARK')
    print('-'*40)
    while True:
        try:
            rec=pickle.load(f)
            print(rec['RollNo'],'\t\t',rec['Name'],'\t\t',rec['Mark'])
        except EOFError:
            break
    f.close()
```

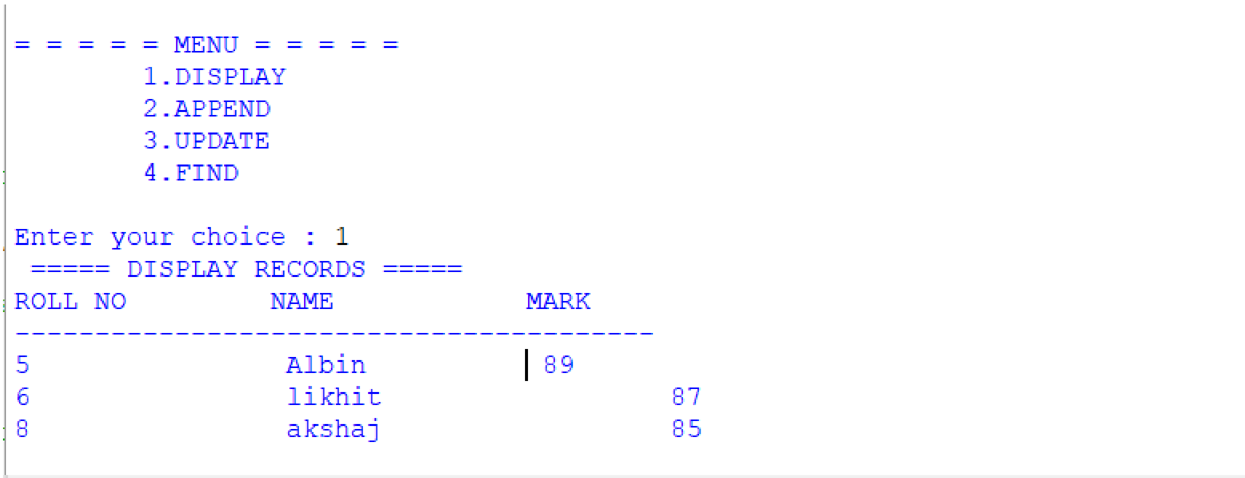


```
def SearchRollNo():
    r= int(input('\nEnter Roll No : '))
    f=open('Student.dat', 'rb')
    found = False
    while True:
        try:
            rec=pickle.load(f)
            if rec['RollNo'] == r:
                found=True
                print('ROLL NO : ', rec['RollNo'])
                print('NAME    : ', rec['Name'])
                print('MARK    : ', rec['Mark'])
        except EOFError:
            break
    if found == False:
        print('Record Not Found')
    f.close()
def UpdateMark():
    roll = int(input('\nEnter Roll No : '))
    f=open('student.dat', 'rb+')
    update = False
    while True:
        try:
            pos = f.tell()
            rec=pickle.load(f)

            if rec['RollNo']==roll:
                m = int(input('Enter new Mark     : '))
                rec['Mark']=m
                f.seek(pos)
                pickle.dump(rec,f)
                print("Record Updated Successfully ")
                update = True
                break
        except EOFError:
            break
    f.close()
    if update == False:
        print('Record not found ')
```

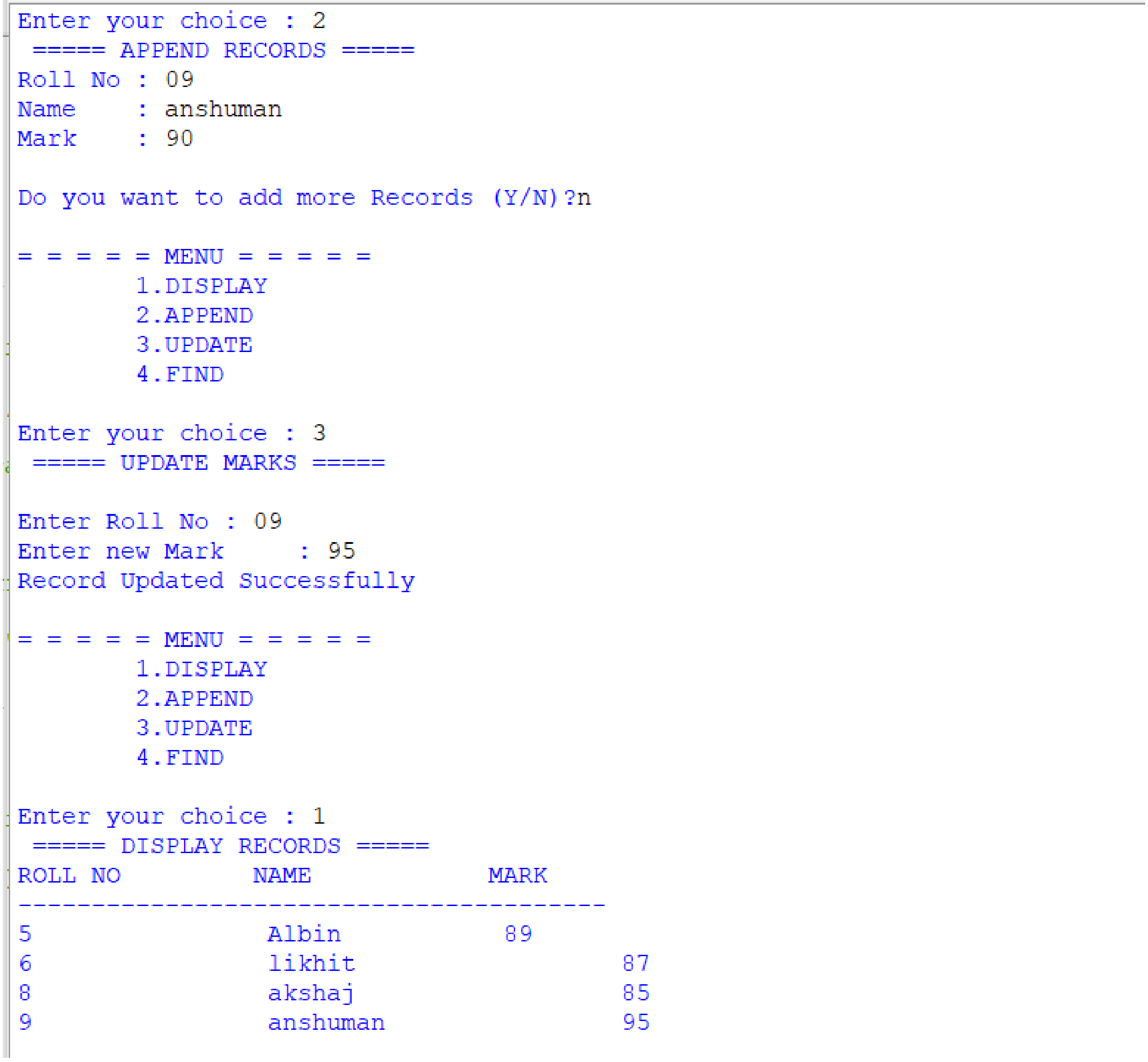


```
while True:
    print( "\n"+ 5*"= "+ "MENU" + 5*" =")
    print('''\t1.DISPLAY   \n\t2.APPEND   \n\t3.UPDATE   \n\t4.FIND''')
    ch=int(input("\nEnter your choice : "))
    if ch==1:
        print(" ===== DISPLAY RECORDS ===== ")
        displayRec()
    elif ch ==2:
        print(" ===== APPEND RECORDS ===== ")
        insertRec()
    elif ch == 3:
        print(" ===== UPDATE MARKS ===== ")
        UpdateMark()
    elif ch == 4:
        print(" ===== FIND RECORD ===== ")
        SearchRollNo()
    else:
        print("Wrong Choice")
        quit( )
```

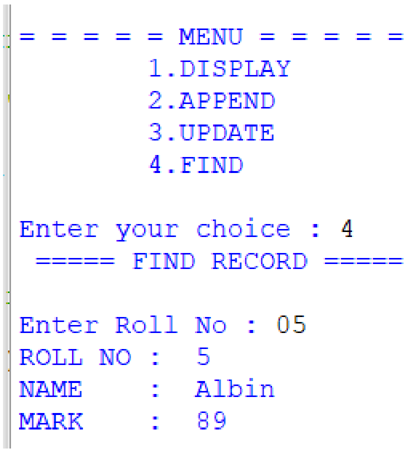
Output:



```
= = = = = MENU = = = = =
        1.DISPLAY
        2.APPEND
        3.UPDATE
        4.FIND

Enter your choice : 1
 ===== DISPLAY RECORDS =====
ROLL NO         NAME            MARK
----------------------------------------
5                Albin          89
6                likhit                  87
8                akshaj                  85
```



```
Enter your choice : 2
 ===== APPEND RECORDS =====
Roll No : 09
Name    : anshuman
Mark    : 90

Do you want to add more Records (Y/N)?n

= = = = = MENU = = = = =
        1.DISPLAY
        2.APPEND
        3.UPDATE
        4.FIND

Enter your choice : 3
 ===== UPDATE MARKS =====

Enter Roll No : 09
Enter new Mark     : 95
Record Updated Successfully

= = = = = MENU = = = = =
        1.DISPLAY
        2.APPEND
        3.UPDATE
        4.FIND

Enter your choice : 1
 ===== DISPLAY RECORDS =====
ROLL NO         NAME            MARK
----------------------------------------
5                Albin           89
6                likhit                  87
8                akshaj                  85
9                anshuman                95
```



```
= = = = = MENU = = = = =
        1.DISPLAY
        2.APPEND
        3.UPDATE
        4.FIND

Enter your choice : 4
 ===== FIND RECORD =====

Enter Roll No : 05
ROLL NO :  5
NAME    :  Albin
MARK    :  89
```

RECORD FILE – C.5

Objective: The file ‘Book’ contains the following information: Book Number, Title of the Book, Cost

Write a menu-driven program in Python to perform the following tasks:

. Append records to the file.

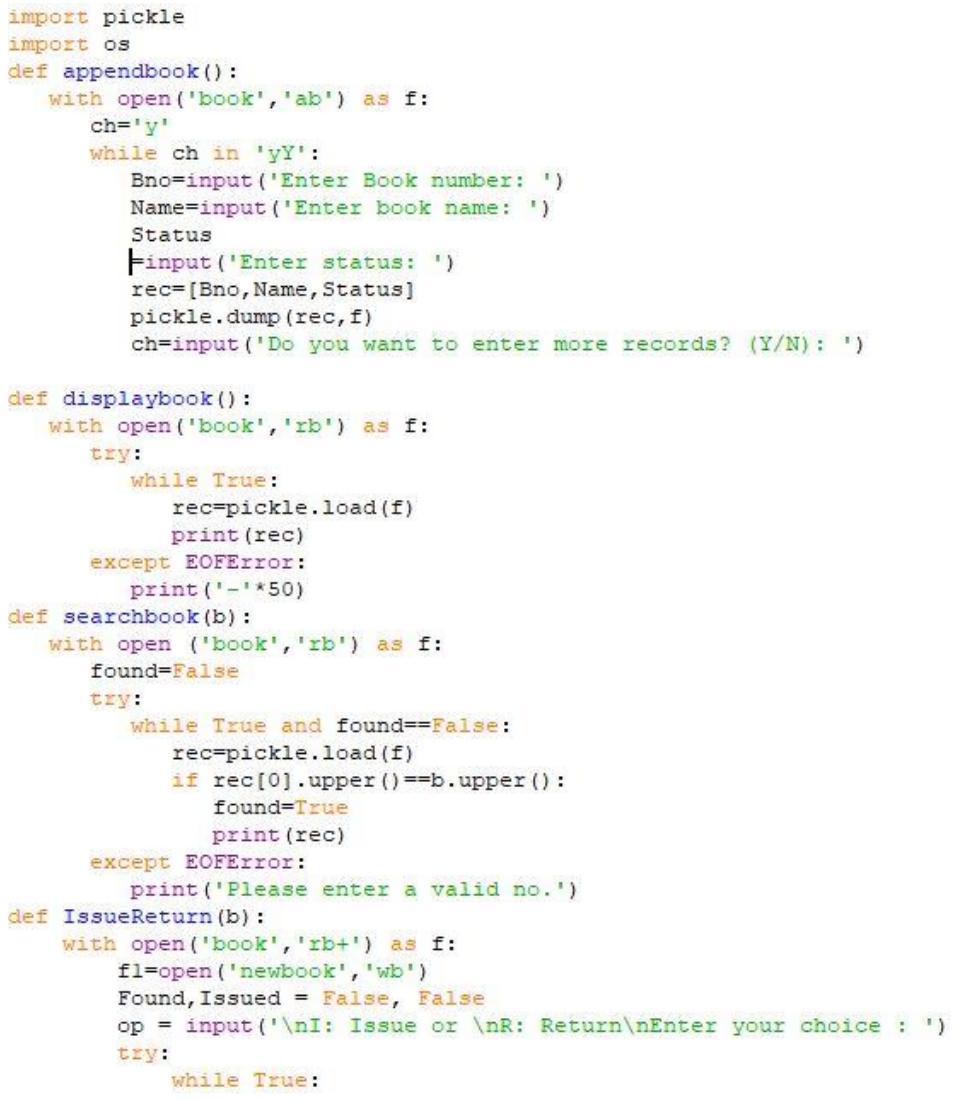
. Display all records from the file.

. Search by Book Number and display details.

. Issue / Release book.

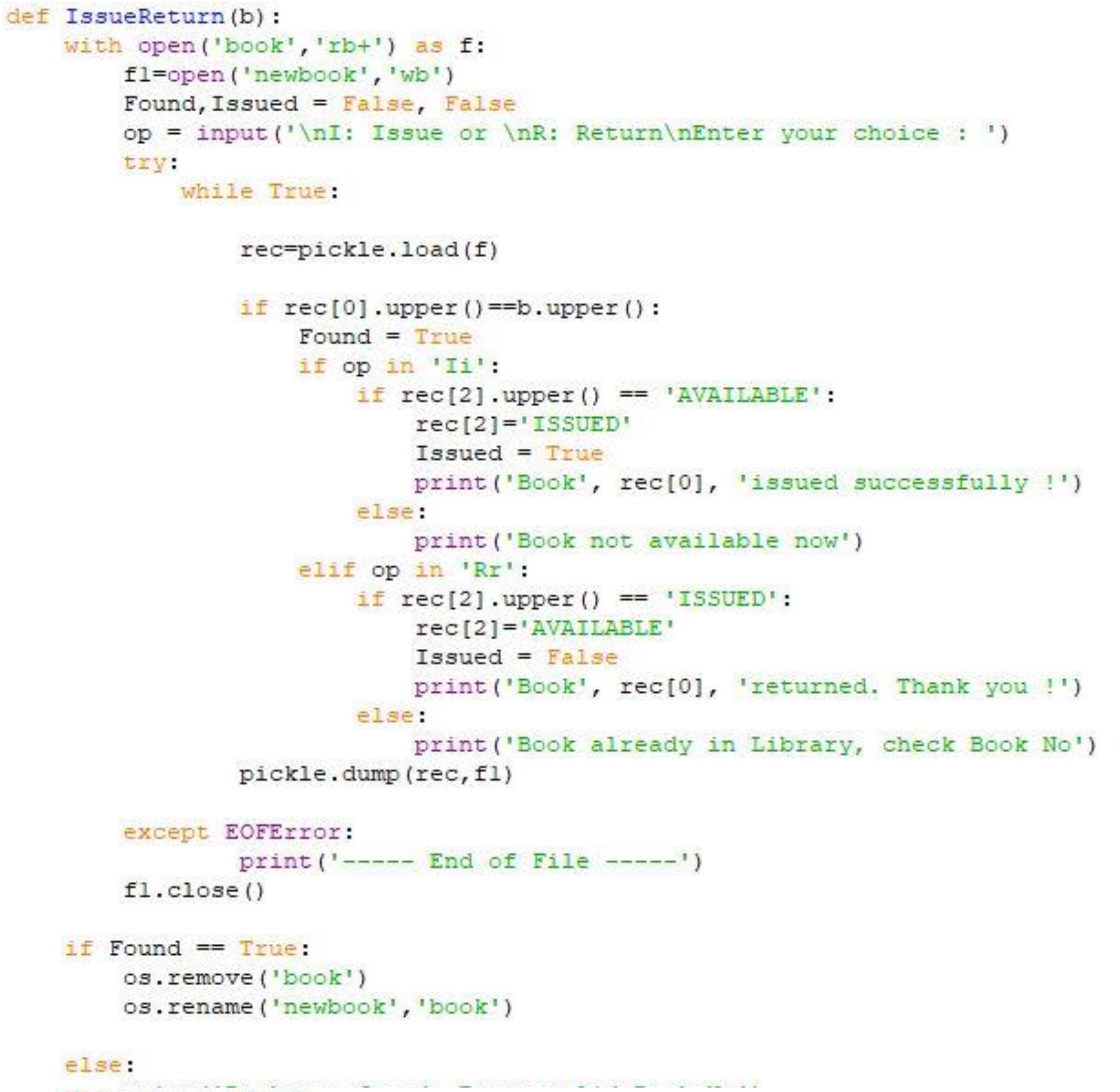
Concept Used: File Handling – Binary files.

Source code:

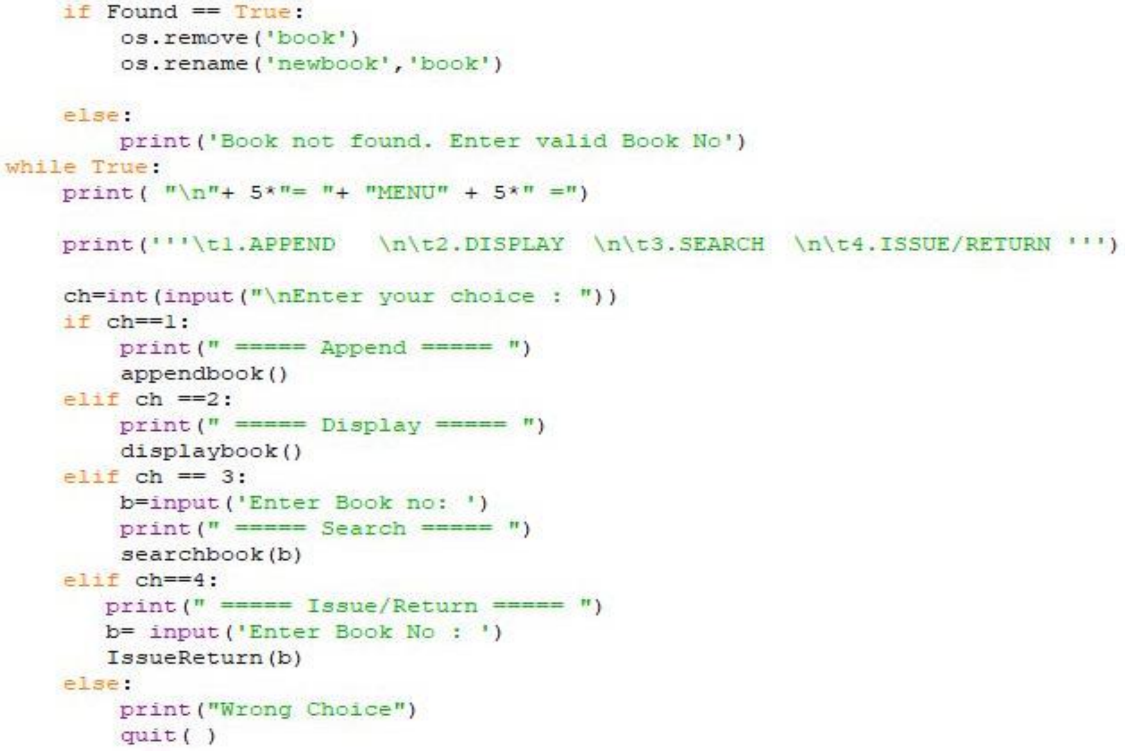


```
import pickle
import os
def appendbook():
   with open('book','ab') as f:
      ch='y'
      while ch in 'yY':
         Bno=input('Enter Book number: ')
         Name=input('Enter book name: ')
         Status
         =input('Enter status: ')
         rec=[Bno,Name,Status]
         pickle.dump(rec,f)
         ch=input('Do you want to enter more records? (Y/N): ')

def displaybook():
   with open('book','rb') as f:
      try:
         while True:
            rec=pickle.load(f)
            print(rec)
      except EOFError:
         print('-'*50)
def searchbook(b):
   with open ('book','rb') as f:
      found=False
      try:
         while True and found==False:
            rec=pickle.load(f)
            if rec[0].upper()==b.upper():
               found=True
               print(rec)
      except EOFError:
         print('Please enter a valid no.')
def IssueReturn(b):
    with open('book','rb+') as f:
        f1=open('newbook','wb')
        Found,Issued = False, False
        op = input('\nI: Issue or \nR: Return\nEnter your choice : ')
        try:
            while True:
```

-

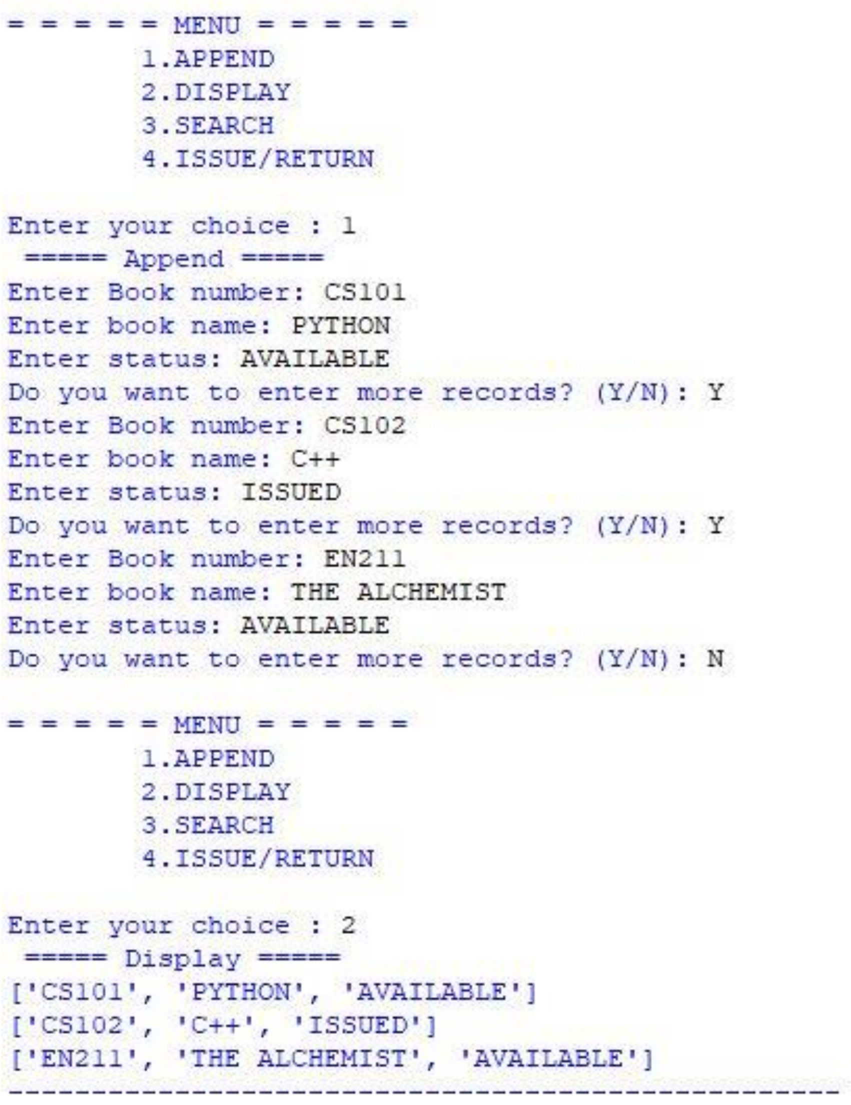


```
def IssueReturn(b):
    with open('book','rb+') as f:
        fl=open('newbook','wb')
        Found,Issued = False, False
        op = input('\nI: Issue or \nR: Return\nEnter your choice : ')
        try:
            while True:

                rec=pickle.load(f)

                if rec[0].upper()==b.upper():
                    Found = True
                    if op in 'Ii':
                        if rec[2].upper() == 'AVAILABLE':
                            rec[2]='ISSUED'
                            Issued = True
                            print('Book', rec[0], 'issued successfully !')
                        else:
                            print('Book not available now')
                    elif op in 'Rr':
                        if rec[2].upper() == 'ISSUED':
                            rec[2]='AVAILABLE'
                            Issued = False
                            print('Book', rec[0], 'returned. Thank you !')
                        else:
                            print('Book already in Library, check Book No')
                pickle.dump(rec,fl)

        except EOFError:
                print('----- End of File -----')
        fl.close()

    if Found == True:
        os.remove('book')
        os.rename('newbook','book')

    else:
```

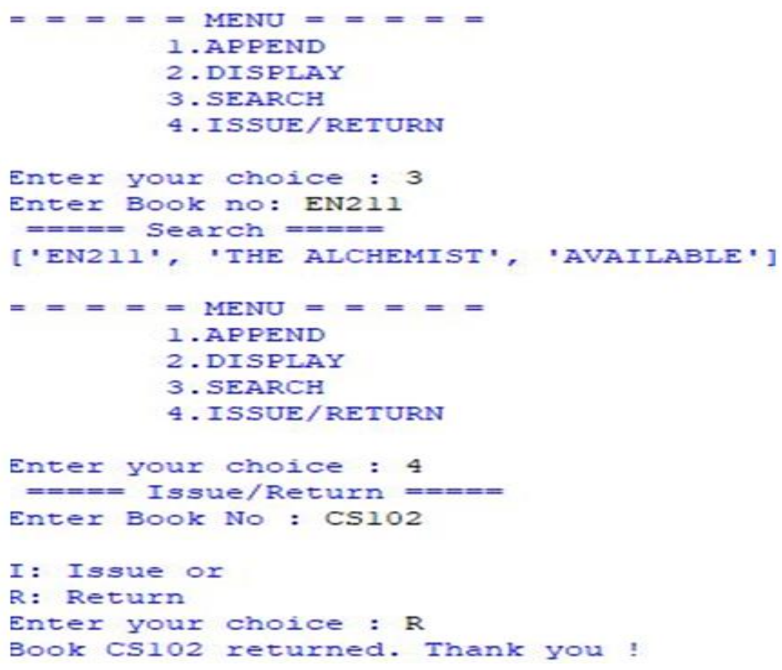


```
    if Found == True:
        os.remove('book')
        os.rename('newbook','book')

    else:
        print('Book not found. Enter valid Book No')
while True:
    print( "\n"+ 5*"= "+ "MENU" + 5*" =")

    print('''\t1.APPEND   \n\t2.DISPLAY  \n\t3.SEARCH  \n\t4.ISSUE/RETURN ''')

    ch=int(input("\nEnter your choice : "))
    if ch==1:
        print(" ===== Append ===== ")
        appendbook()
    elif ch ==2:
        print(" ===== Display ===== ")
        displaybook()
    elif ch == 3:
        b=input('Enter Book no: ')
        print(" ===== Search ===== ")
        searchbook(b)
    elif ch==4:
       print(" ===== Issue/Return ===== ")
       b= input('Enter Book No : ')
       IssueReturn(b)
    else:
        print("Wrong Choice")
        quit( )
```

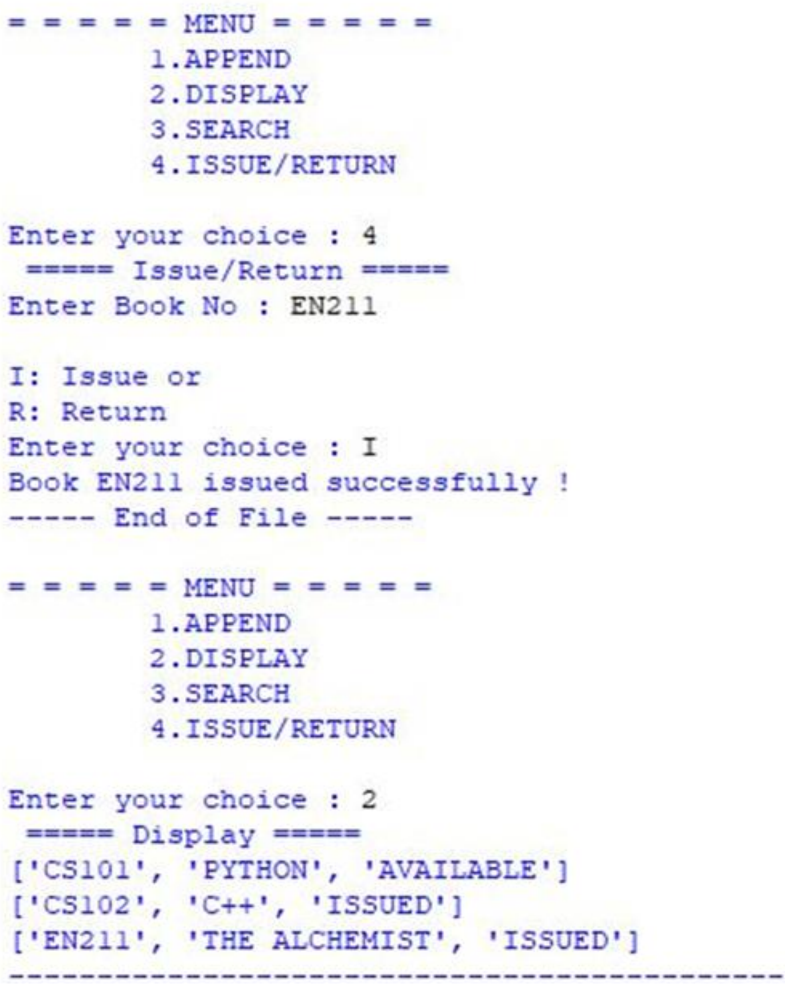
Output:



```
= = = = = MENU = = = = =
        1.APPEND
        2.DISPLAY
        3.SEARCH
        4.ISSUE/RETURN

Enter your choice : 1
 ===== Append =====
Enter Book number: CS101
Enter book name: PYTHON
Enter status: AVAILABLE
Do you want to enter more records? (Y/N): Y
Enter Book number: CS102
Enter book name: C++
Enter status: ISSUED
Do you want to enter more records? (Y/N): Y
Enter Book number: EN211
Enter book name: THE ALCHEMIST
Enter status: AVAILABLE
Do you want to enter more records? (Y/N): N

= = = = = MENU = = = = =
        1.APPEND
        2.DISPLAY
        3.SEARCH
        4.ISSUE/RETURN

Enter your choice : 2
 ===== Display =====
['CS101', 'PYTHON', 'AVAILABLE']
['CS102', 'C++', 'ISSUED']
['EN211', 'THE ALCHEMIST', 'AVAILABLE']
--------------------------------------------------
```



```
= = = = = MENU = = = = =
        1.APPEND
        2.DISPLAY
        3.SEARCH
        4.ISSUE/RETURN

Enter your choice : 3
Enter Book no: EN211
 ===== Search =====
['EN211', 'THE ALCHEMIST', 'AVAILABLE']

= = = = = MENU = = = = =
        1.APPEND
        2.DISPLAY
        3.SEARCH
        4.ISSUE/RETURN

Enter your choice : 4
 ===== Issue/Return =====
Enter Book No : CS102

I: Issue or
R: Return
Enter your choice : R
Book CS102 returned. Thank you !
```



```
= = = = = MENU = = = = =
        1.APPEND
        2.DISPLAY
        3.SEARCH
        4.ISSUE/RETURN

Enter your choice : 4
 ===== Issue/Return =====
Enter Book No : EN211

I: Issue or
R: Return
Enter your choice : I
Book EN211 issued successfully !
----- End of File -----

= = = = = MENU = = = = =
        1.APPEND
        2.DISPLAY
        3.SEARCH
        4.ISSUE/RETURN

Enter your choice : 2
 ===== Display =====
['CS101', 'PYTHON', 'AVAILABLE']
['CS102', 'C++', 'ISSUED']
['EN211', 'THE ALCHEMIST', 'ISSUED']
--------------------------------------------
```

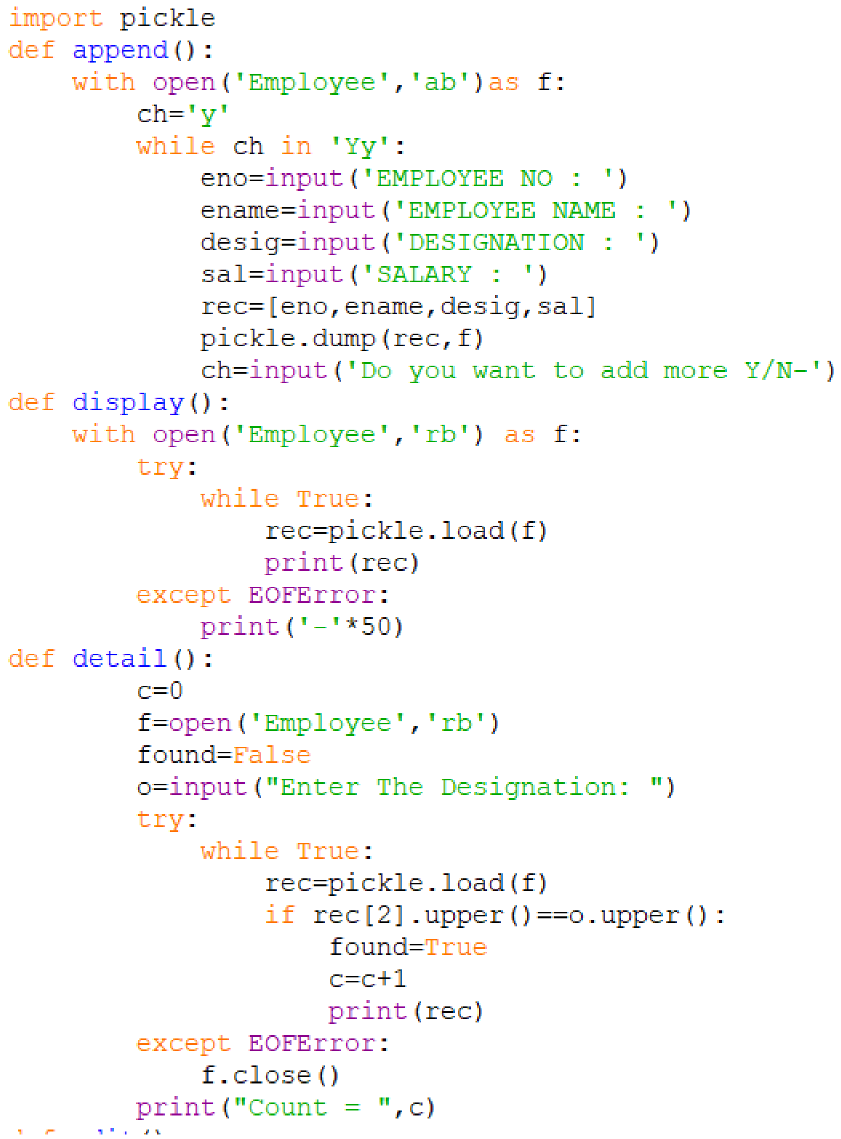
RECORD FILE – C.6

Objective: A data file ‘Employee’ contains the following information about each employee: Employee number, employee name, designation and salary.

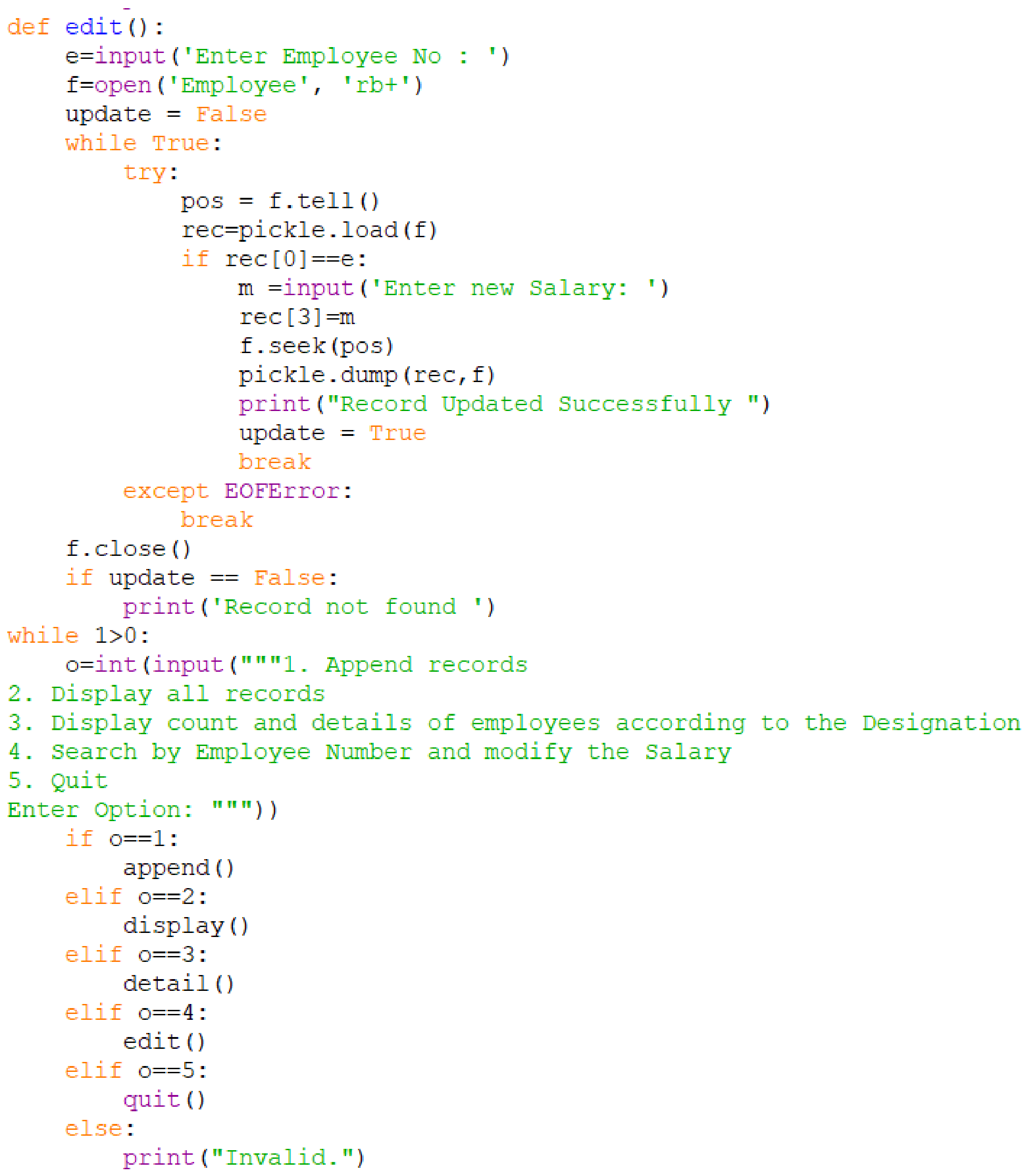
Write a menu-driven program in Python to perform the following tasks:
. Append records to the file.
. Display all records from the file.
. Display count & details of employees according to Department given by user.
. Search by Employee Number and Modify the salary with amount given by user.

Concept Used: File Handling – Binary Files , List.

Source code:

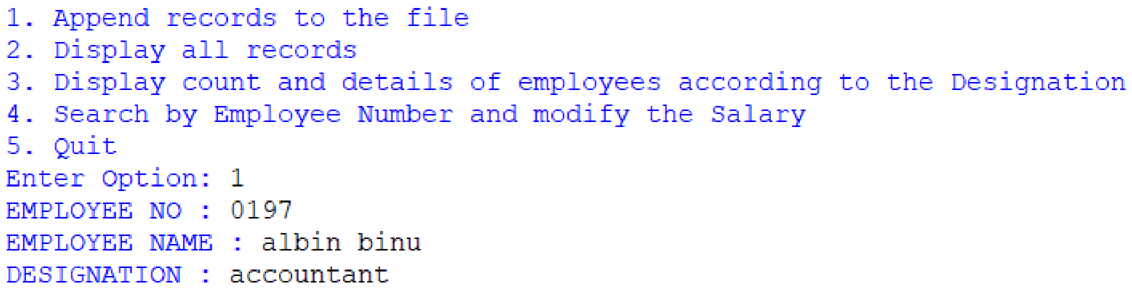


```
import pickle
def append():
    with open('Employee','ab')as f:
        ch='y'
        while ch in 'Yy':
            eno=input('EMPLOYEE NO : ')
            ename=input('EMPLOYEE NAME : ')
            desig=input('DESIGNATION : ')
            sal=input('SALARY : ')
            rec=[eno,ename,desig,sal]
            pickle.dump(rec,f)
            ch=input('Do you want to add more Y/N-')
def display():
    with open('Employee','rb') as f:
        try:
            while True:
                rec=pickle.load(f)
                print(rec)
        except EOFError:
            print('-'*50)
def detail():
        c=0
        f=open('Employee','rb')
        found=False
        o=input("Enter The Designation: ")
        try:
            while True:
                rec=pickle.load(f)
                if rec[2].upper()==o.upper():
                    found=True
                    c=c+1
                    print(rec)
        except EOFError:
            f.close()
        print("Count = ",c)
```

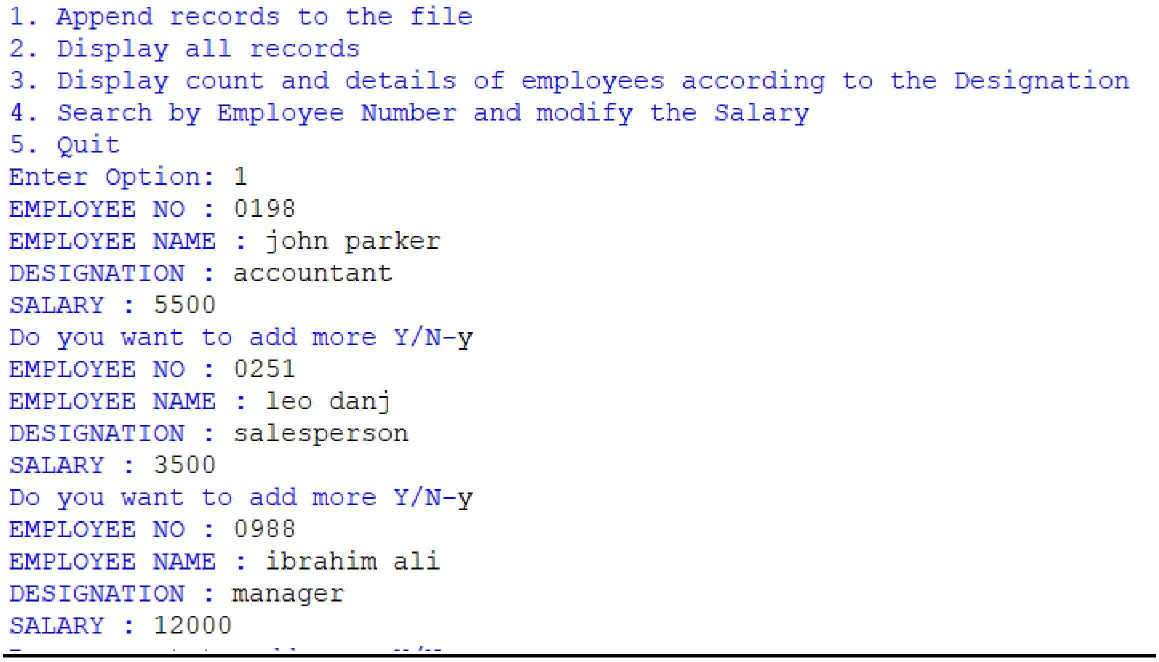


```
def edit():
    e=input('Enter Employee No : ')
    f=open('Employee', 'rb+')
    update = False
    while True:
        try:
            pos = f.tell()
            rec=pickle.load(f)
            if rec[0]==e:
                m =input('Enter new Salary: ')
                rec[3]=m
                f.seek(pos)
                pickle.dump(rec,f)
                print("Record Updated Successfully ")
                update = True
                break
        except EOFError:
            break
    f.close()
    if update == False:
        print('Record not found ')
while 1>0:
    o=int(input("""1. Append records
2. Display all records
3. Display count and details of employees according to the Designation
4. Search by Employee Number and modify the Salary
5. Quit
Enter Option: """))
    if o==1:
        append()
    elif o==2:
        display()
    elif o==3:
        detail()
    elif o==4:
        edit()
    elif o==5:
        quit()
    else:
        print("Invalid.")
```

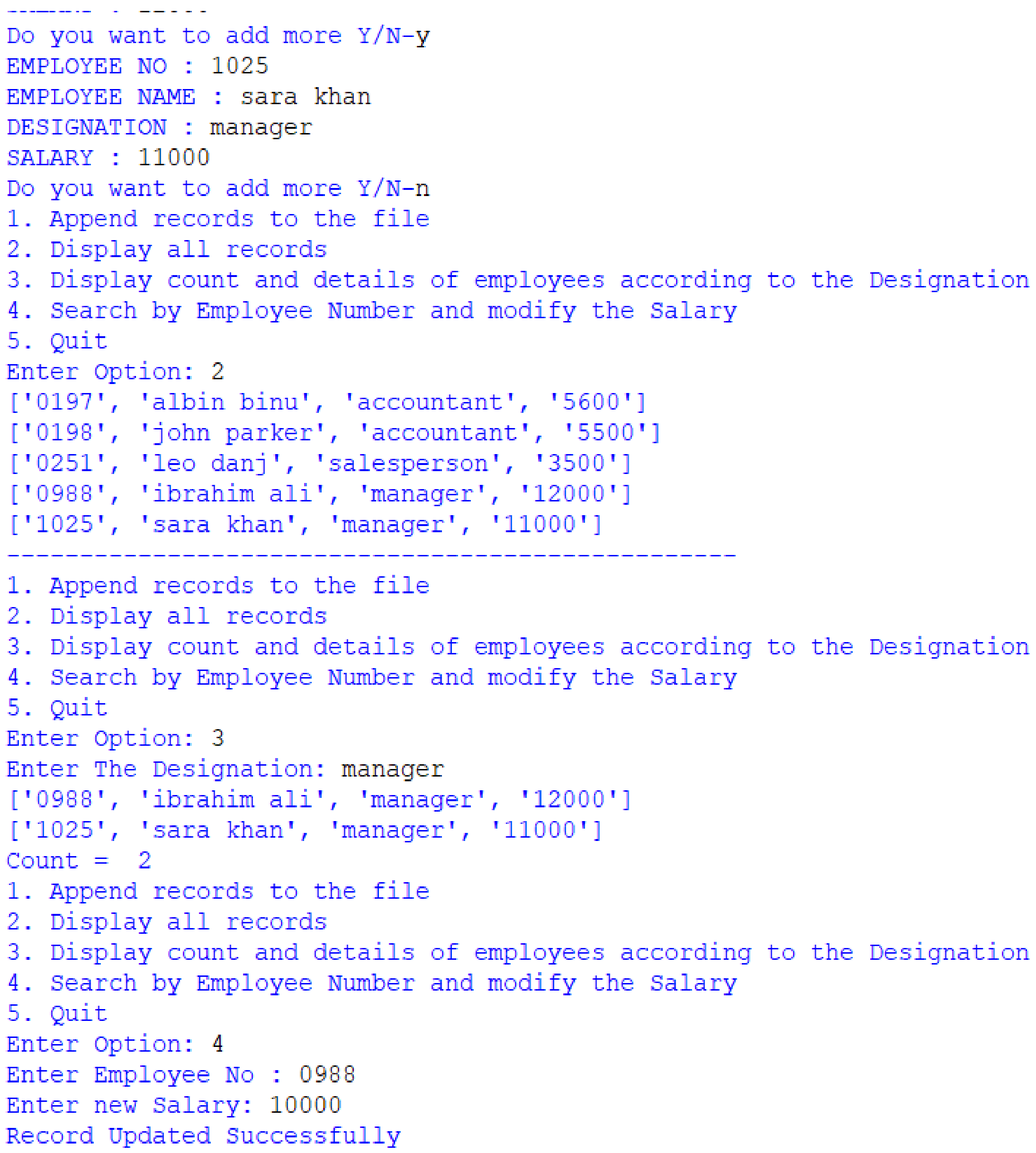
Output:



```
1. Append records to the file
2. Display all records
3. Display count and details of employees according to the Designation
4. Search by Employee Number and modify the Salary
5. Quit
Enter Option: 1
EMPLOYEE NO : 0197
EMPLOYEE NAME : albin binu
DESIGNATION : accountant
```



```
1. Append records to the file
2. Display all records
3. Display count and details of employees according to the Designation
4. Search by Employee Number and modify the Salary
5. Quit
Enter Option: 1
EMPLOYEE NO : 0198
EMPLOYEE NAME : john parker
DESIGNATION : accountant
SALARY : 5500
Do you want to add more Y/N-y
EMPLOYEE NO : 0251
EMPLOYEE NAME : leo danj
DESIGNATION : salesperson
SALARY : 3500
Do you want to add more Y/N-y
EMPLOYEE NO : 0988
EMPLOYEE NAME : ibrahim ali
DESIGNATION : manager
SALARY : 12000
```



```
Do you want to add more Y/N-y
EMPLOYEE NO : 1025
EMPLOYEE NAME : sara khan
DESIGNATION : manager
SALARY : 11000
Do you want to add more Y/N-n
1. Append records to the file
2. Display all records
3. Display count and details of employees according to the Designation
4. Search by Employee Number and modify the Salary
5. Quit
Enter Option: 2
['0197', 'albin binu', 'accountant', '5600']
['0198', 'john parker', 'accountant', '5500']
['0251', 'leo danj', 'salesperson', '3500']
['0988', 'ibrahim ali', 'manager', '12000']
['1025', 'sara khan', 'manager', '11000']
--------------------------------------------------
1. Append records to the file
2. Display all records
3. Display count and details of employees according to the Designation
4. Search by Employee Number and modify the Salary
5. Quit
Enter Option: 3
Enter The Designation: manager
['0988', 'ibrahim ali', 'manager', '12000']
['1025', 'sara khan', 'manager', '11000']
Count =  2
1. Append records to the file
2. Display all records
3. Display count and details of employees according to the Designation
4. Search by Employee Number and modify the Salary
5. Quit
Enter Option: 4
Enter Employee No : 0988
Enter new Salary: 10000
Record Updated Successfully
```

RECORD FILE – C.7

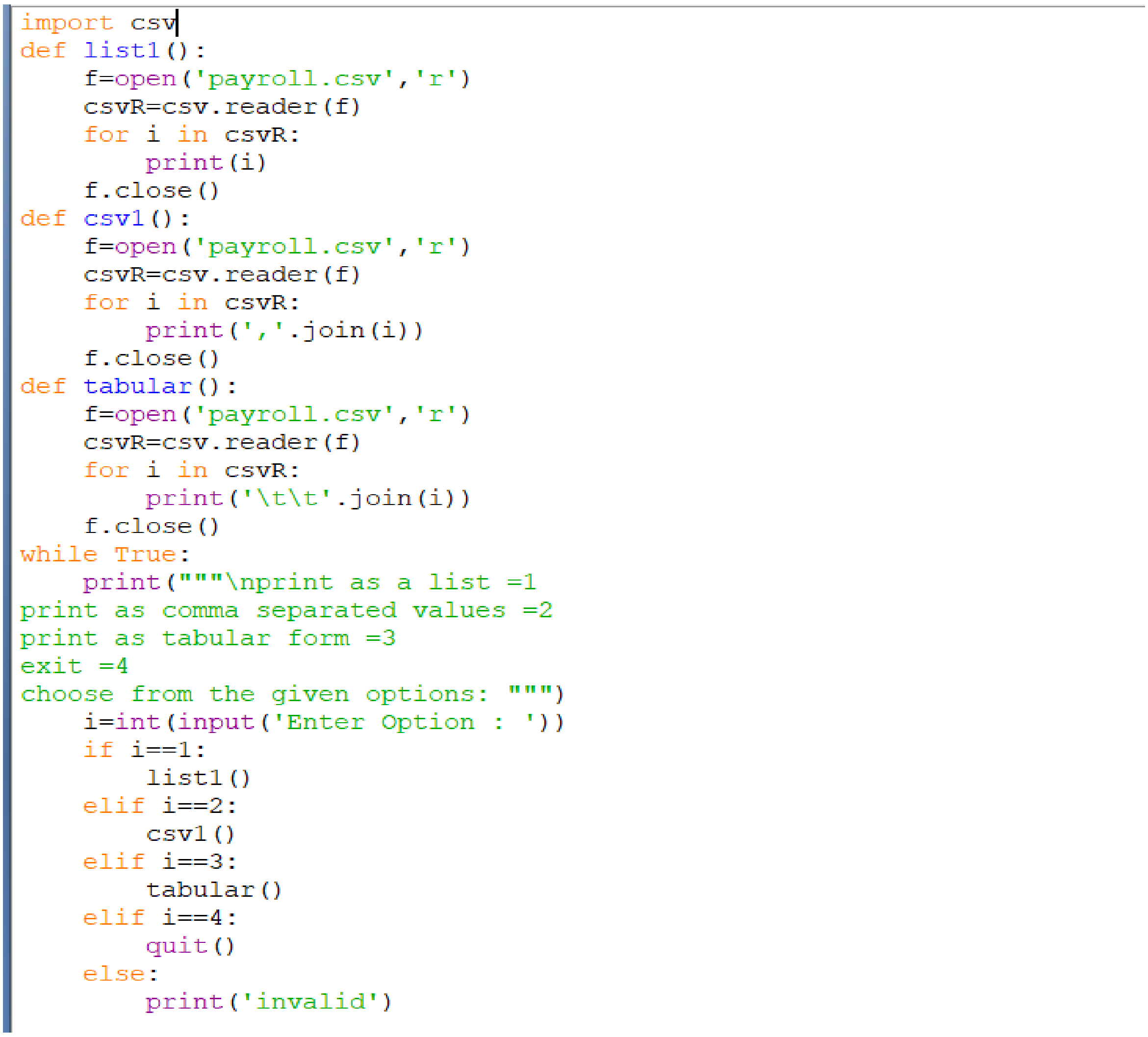
Objective: Create a Spreadsheet ‘Payroll.xlsx’ with Employee No, Employee Name, Salary and convert to CSV File.

Design Python Program to print the data in ‘Payroll.csv’ as:

. List
. Comma separated values
. Tabulated form

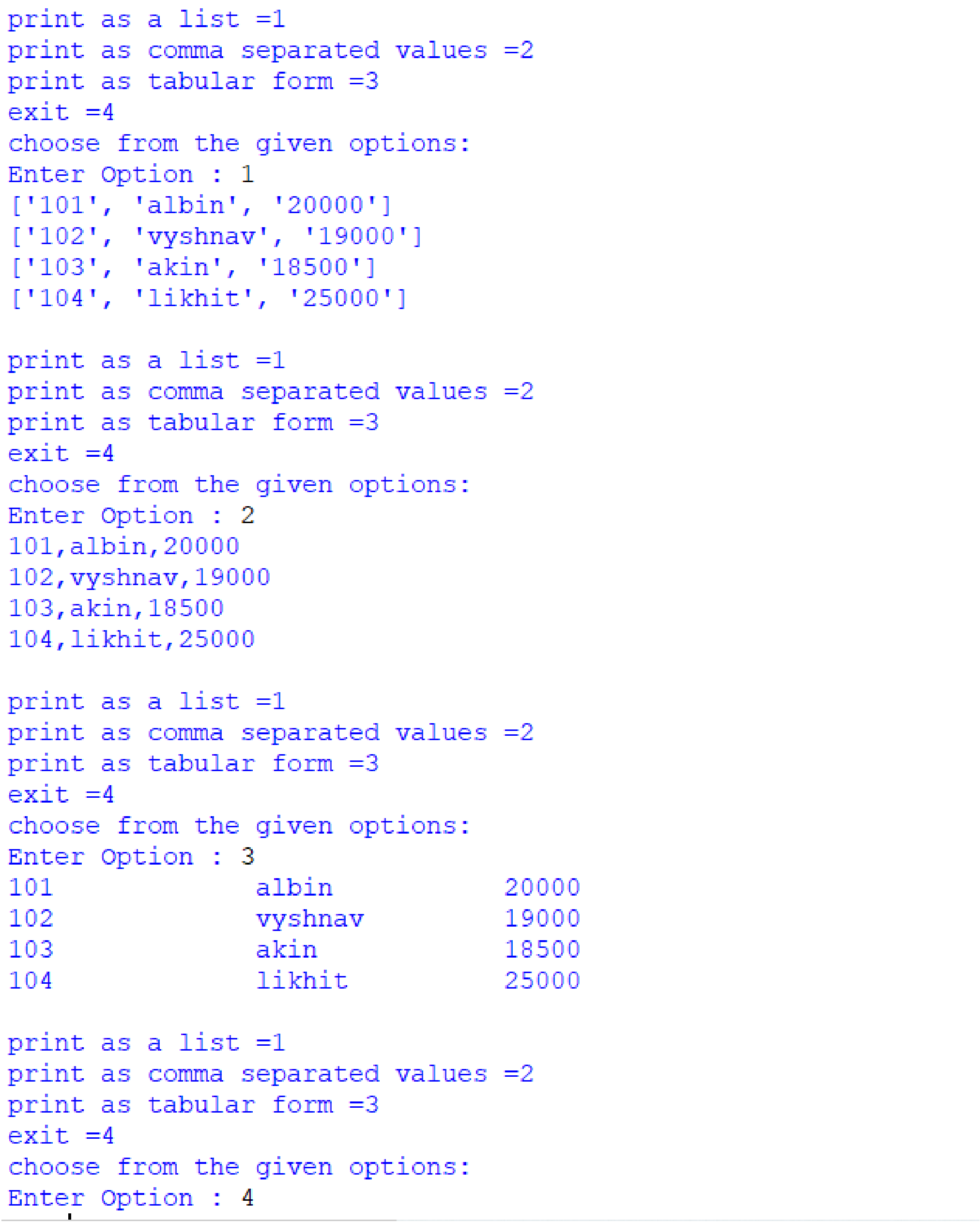
Concept Used: Excel CSV , CSV File.

Source code:



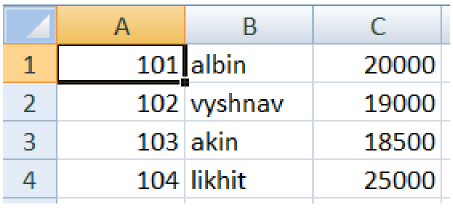
```
import csv
def list1():
    f=open('payroll.csv','r')
    csvR=csv.reader(f)
    for i in csvR:
        print(i)
    f.close()
def csv1():
    f=open('payroll.csv','r')
    csvR=csv.reader(f)
    for i in csvR:
        print(','.join(i))
    f.close()
def tabular():
    f=open('payroll.csv','r')
    csvR=csv.reader(f)
    for i in csvR:
        print('\t\t'.join(i))
    f.close()
while True:
    print("""\nprint as a list =1
print as comma separated values =2
print as tabular form =3
exit =4
choose from the given options: """)
    i=int(input('Enter Option : '))
    if i==1:
        list1()
    elif i==2:
        csv1()
    elif i==3:
        tabular()
    elif i==4:
        quit()
    else:
        print('invalid')
```

Output:



```
print as a list =1
print as comma separated values =2
print as tabular form =3
exit =4
choose from the given options:
Enter Option : 1
['101', 'albin', '20000']
['102', 'vyshnav', '19000']
['103', 'akin', '18500']
['104', 'likhit', '25000']

print as a list =1
print as comma separated values =2
print as tabular form =3
exit =4
choose from the given options:
Enter Option : 2
101,albin,20000
102,vyshnav,19000
103,akin,18500
104,likhit,25000

print as a list =1
print as comma separated values =2
print as tabular form =3
exit =4
choose from the given options:
Enter Option : 3
101             albin           20000
102             vyshnav         19000
103             akin            18500
104             likhit          25000

print as a list =1
print as comma separated values =2
print as tabular form =3
exit =4
choose from the given options:
Enter Option : 4
```

Payroll.csv:



	A	B	C
1	101	albin	20000
2	102	vyshnav	19000
3	103	akin	18500
4	104	likhit	25000

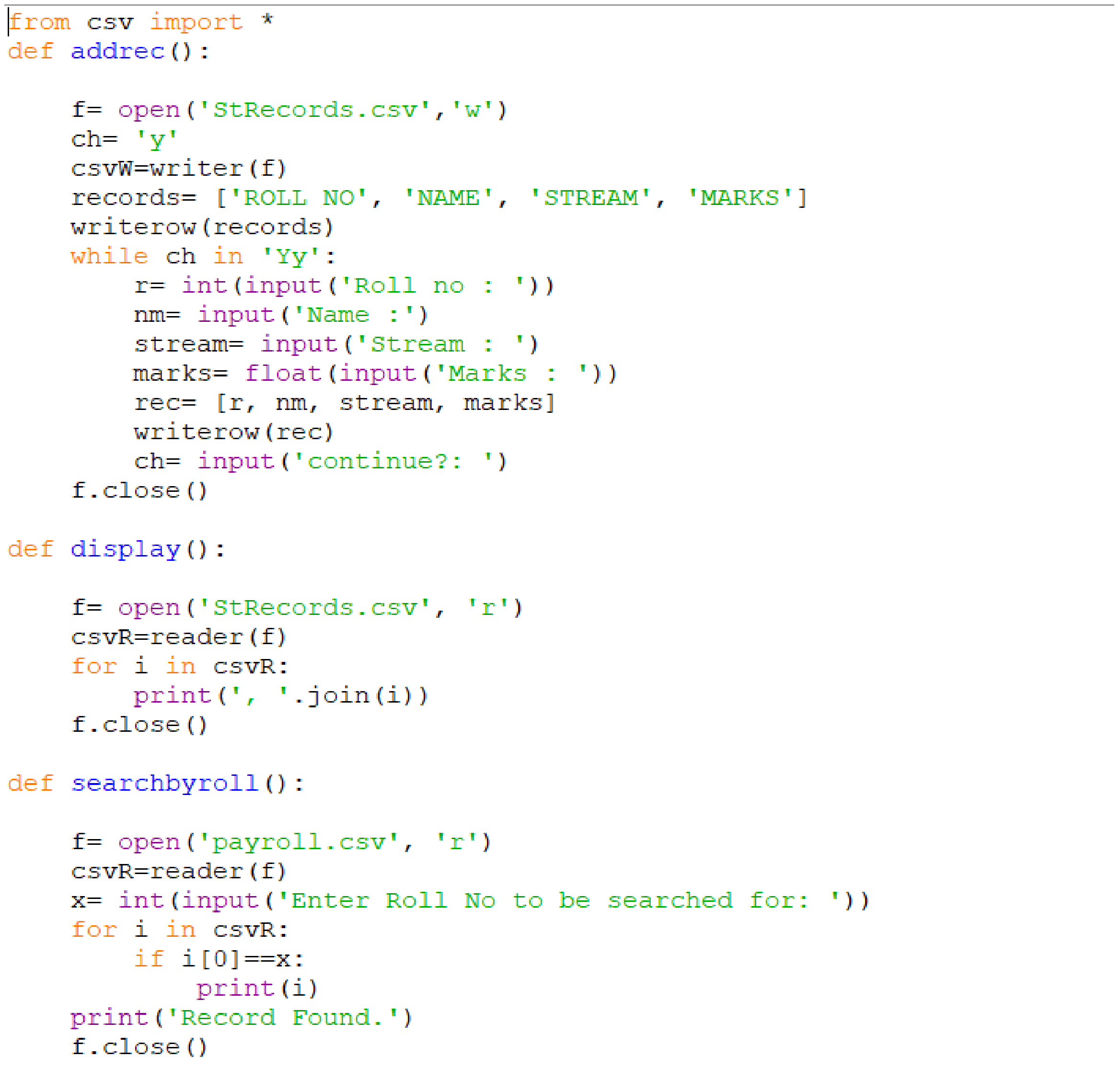
RECORD FILE – C.8

Objective: A CSV File ‘StRecords.csv’ contains the fields: Roll No, Name, Stream, Marks. Write a Menu driven program to:

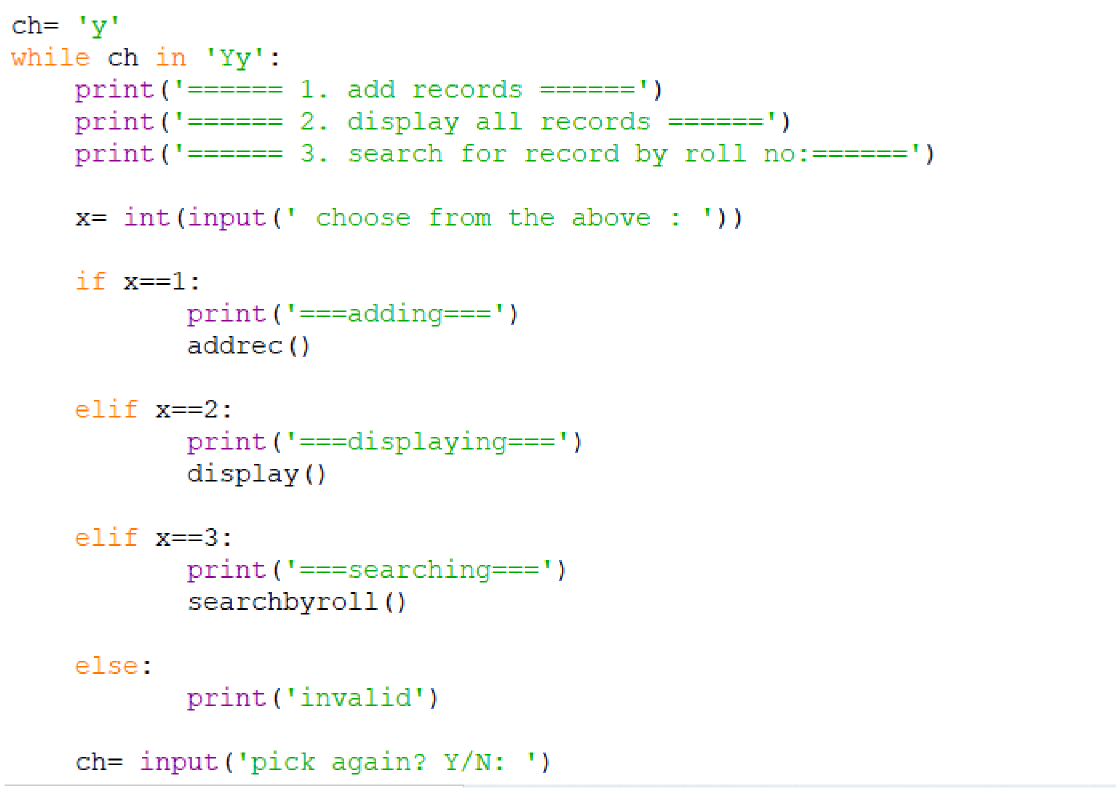
. Add Records
. Display Records
. Search by Roll No & Display details

Concept Used: CSV File.

Source code:

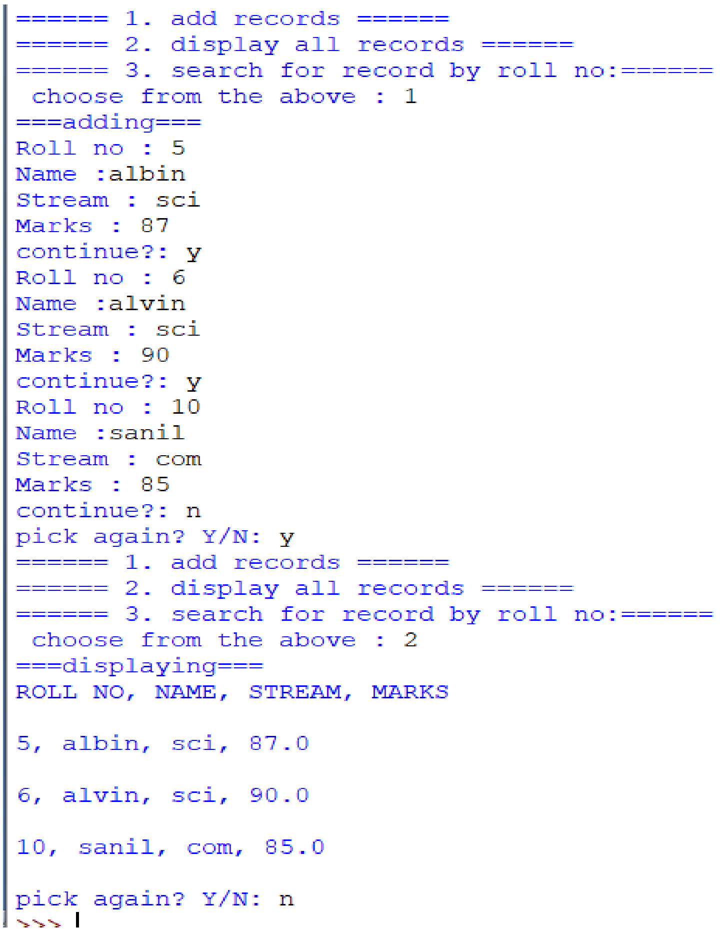


```
from csv import *
def addrec():

    f= open('StRecords.csv','w')
    ch= 'y'
    csvW=writer(f)
    records= ['ROLL NO', 'NAME', 'STREAM', 'MARKS']
    writerow(records)
    while ch in 'Yy':
        r= int(input('Roll no : '))
        nm= input('Name :')
        stream= input('Stream : ')
        marks= float(input('Marks : '))
        rec= [r, nm, stream, marks]
        writerow(rec)
        ch= input('continue?: ')
    f.close()

def display():

    f= open('StRecords.csv', 'r')
    csvR=reader(f)
    for i in csvR:
        print(', '.join(i))
    f.close()

def searchbyroll():

    f= open('payroll.csv', 'r')
    csvR=reader(f)
    x= int(input('Enter Roll No to be searched for: '))
    for i in csvR:
        if i[0]==x:
            print(i)
    print('Record Found.')
    f.close()
```



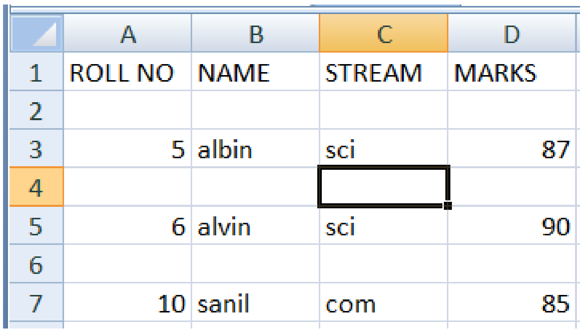
```python
ch= 'y'
while ch in 'Yy':
    print('====== 1. add records ======')
    print('====== 2. display all records ======')
    print('====== 3. search for record by roll no:======')

    x= int(input(' choose from the above : '))

    if x==1:
            print('===adding===')
            addrec()

    elif x==2:
            print('===displaying===')
            display()

    elif x==3:
            print('===searching===')
            searchbyroll()

    else:
            print('invalid')

    ch= input('pick again? Y/N: ')
```

Output:



```
====== 1. add records ======
====== 2. display all records ======
====== 3. search for record by roll no:======
 choose from the above : 1
===adding===
Roll no : 5
Name :albin
Stream : sci
Marks : 87
continue?: y
Roll no : 6
Name :alvin
Stream : sci
Marks : 90
continue?: y
Roll no : 10
Name :sanil
Stream : com
Marks : 85
continue?: n
pick again? Y/N: y
====== 1. add records ======
====== 2. display all records ======
====== 3. search for record by roll no:======
 choose from the above : 2
===displaying===
ROLL NO, NAME, STREAM, MARKS

5, albin, sci, 87.0

6, alvin, sci, 90.0

10, sanil, com, 85.0

pick again? Y/N: n
>>> 
```

StRecords.csv:



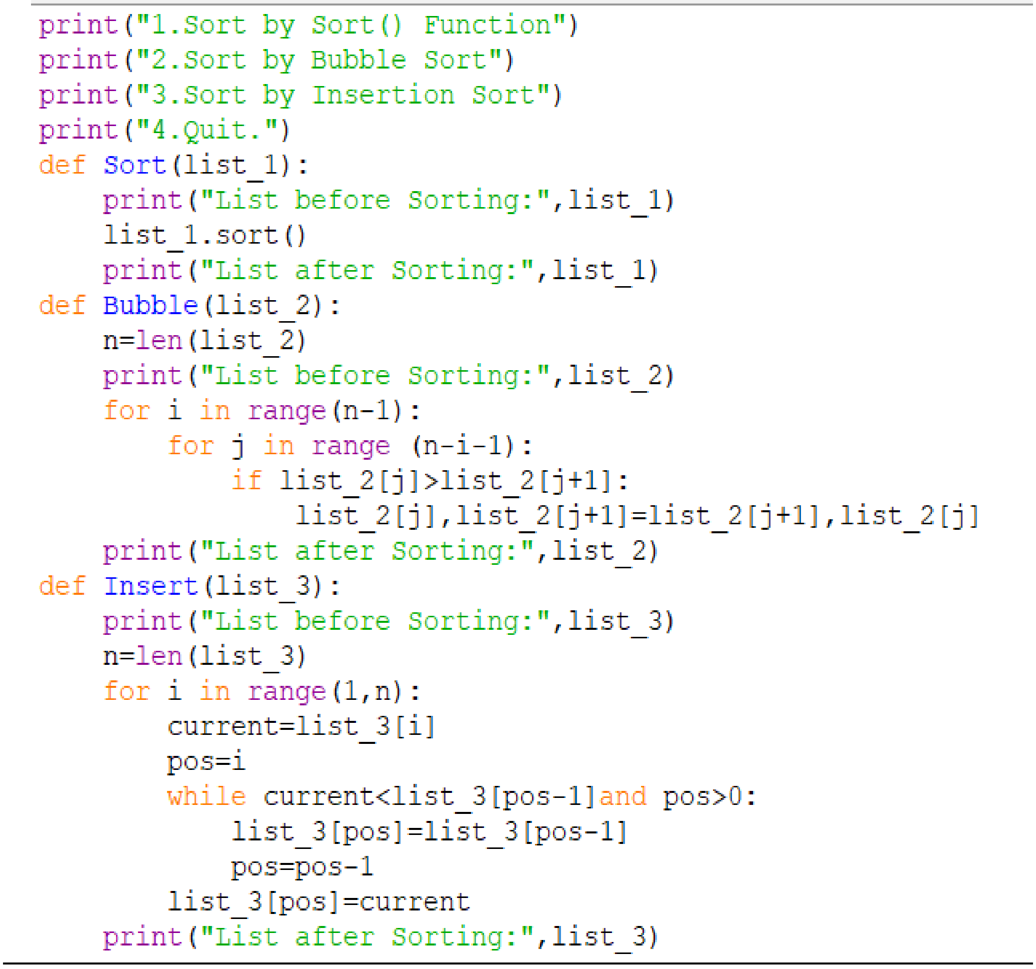
	A	B	C	D
1	ROLL NO	NAME	STREAM	MARKS
2				
3	5	albin	sci	87
4				
5	6	alvin	sci	90
6				
7	10	sanil	com	85

RECORD FILE-D.1

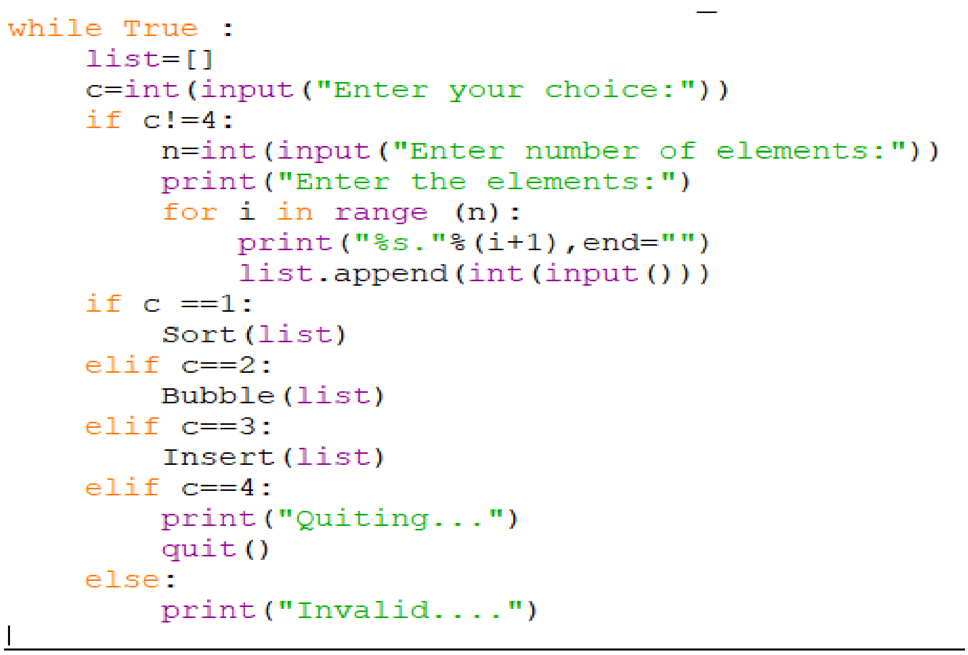
Objective: Write a menu driven program to accept a list from the user and sort by:

. Sort() Function
. Bubble Sort
. Insertion Sort.

Source code:

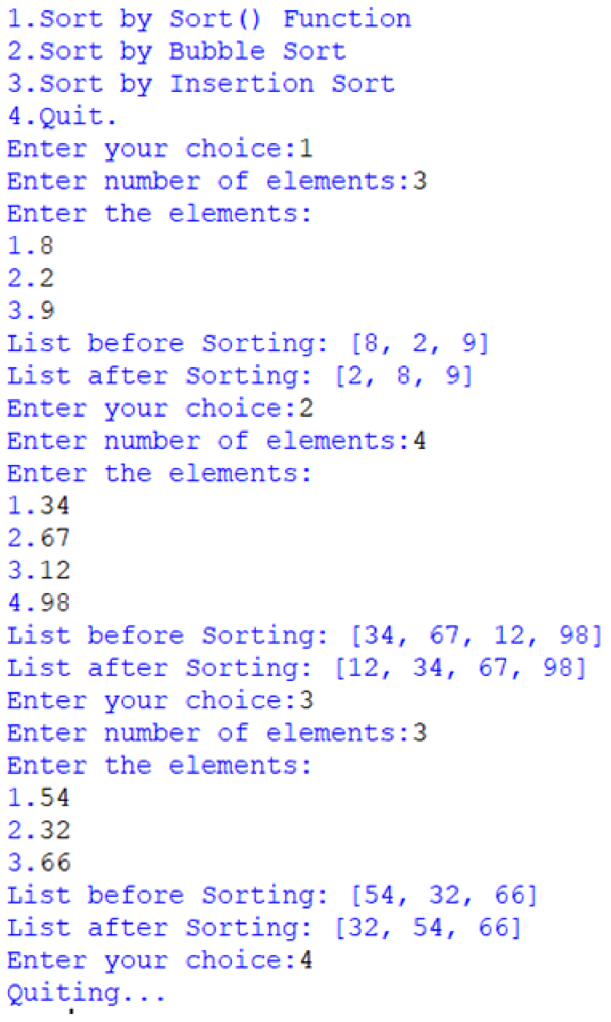


```
print("1.Sort by Sort() Function")
print("2.Sort by Bubble Sort")
print("3.Sort by Insertion Sort")
print("4.Quit.")
def Sort(list_1):
    print("List before Sorting:",list_1)
    list_1.sort()
    print("List after Sorting:",list_1)
def Bubble(list_2):
    n=len(list_2)
    print("List before Sorting:",list_2)
    for i in range(n-1):
        for j in range (n-i-1):
            if list_2[j]>list_2[j+1]:
                list_2[j],list_2[j+1]=list_2[j+1],list_2[j]
    print("List after Sorting:",list_2)
def Insert(list_3):
    print("List before Sorting:",list_3)
    n=len(list_3)
    for i in range(1,n):
        current=list_3[i]
        pos=i
        while current<list_3[pos-1]and pos>0:
            list_3[pos]=list_3[pos-1]
            pos=pos-1
        list_3[pos]=current
    print("List after Sorting:",list_3)
```



```
while True :
    list=[]
    c=int(input("Enter your choice:"))
    if c!=4:
        n=int(input("Enter number of elements:"))
        print("Enter the elements:")
        for i in range (n):
            print("%s."%(i+1),end="")
            list.append(int(input()))
    if c ==1:
        Sort(list)
    elif c==2:
        Bubble(list)
    elif c==3:
        Insert(list)
    elif c==4:
        print("Quiting...")
        quit()
    else:
        print("Invalid....")
```

Output:



```
1.Sort by Sort() Function
2.Sort by Bubble Sort
3.Sort by Insertion Sort
4.Quit.
Enter your choice:1
Enter number of elements:3
Enter the elements:
1.8
2.2
3.9
List before Sorting: [8, 2, 9]
List after Sorting: [2, 8, 9]
Enter your choice:2
Enter number of elements:4
Enter the elements:
1.34
2.67
3.12
4.98
List before Sorting: [34, 67, 12, 98]
List after Sorting: [12, 34, 67, 98]
Enter your choice:3
Enter number of elements:3
Enter the elements:
1.54
2.32
3.66
List before Sorting: [54, 32, 66]
List after Sorting: [32, 54, 66]
Enter your choice:4
Quiting...
```

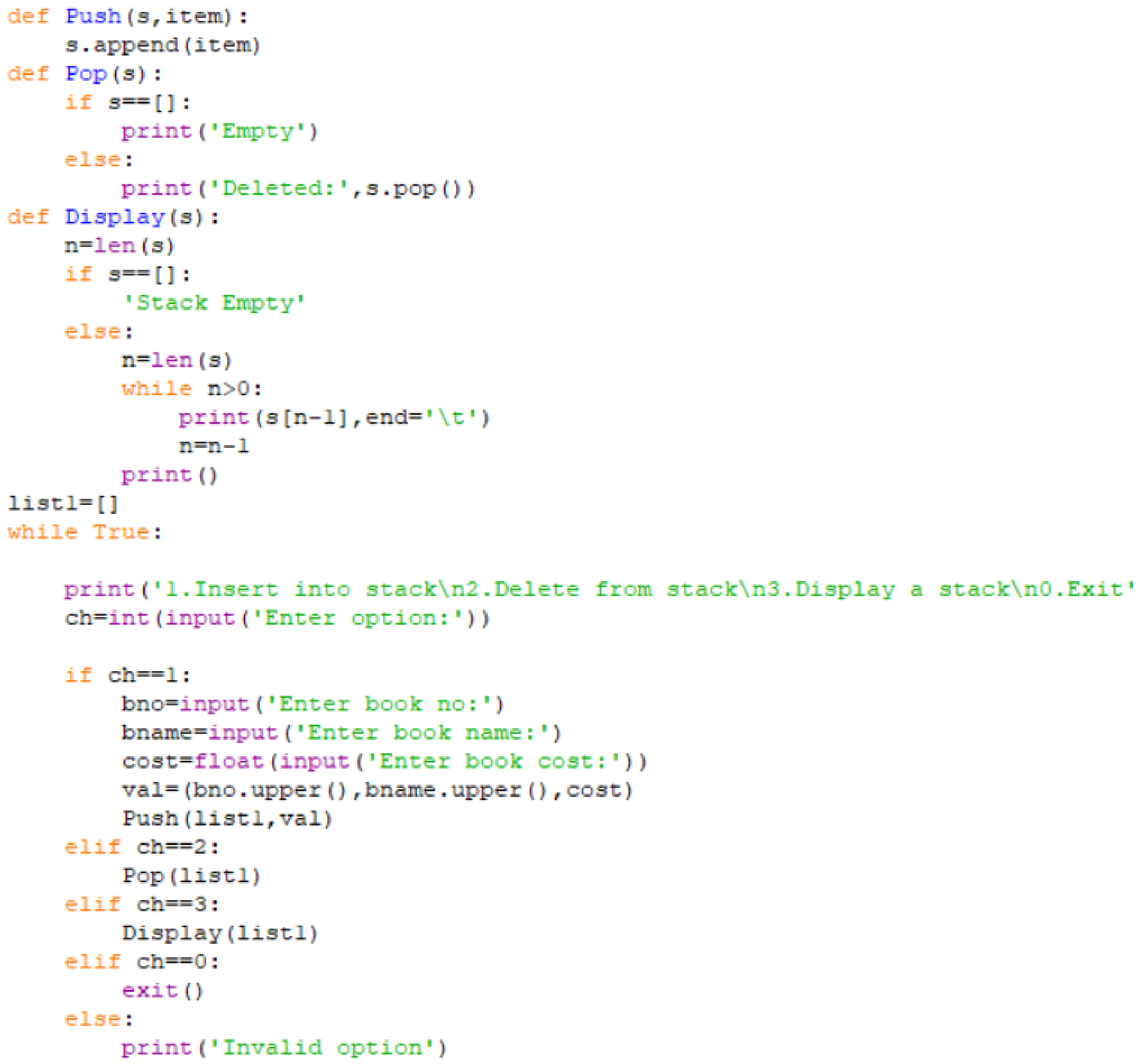
RECORD FILE – D.2

Objective: Write a menu driven program to implement a stack for the book details:

. Book No – String
. Book name – String
. Cost – Float.

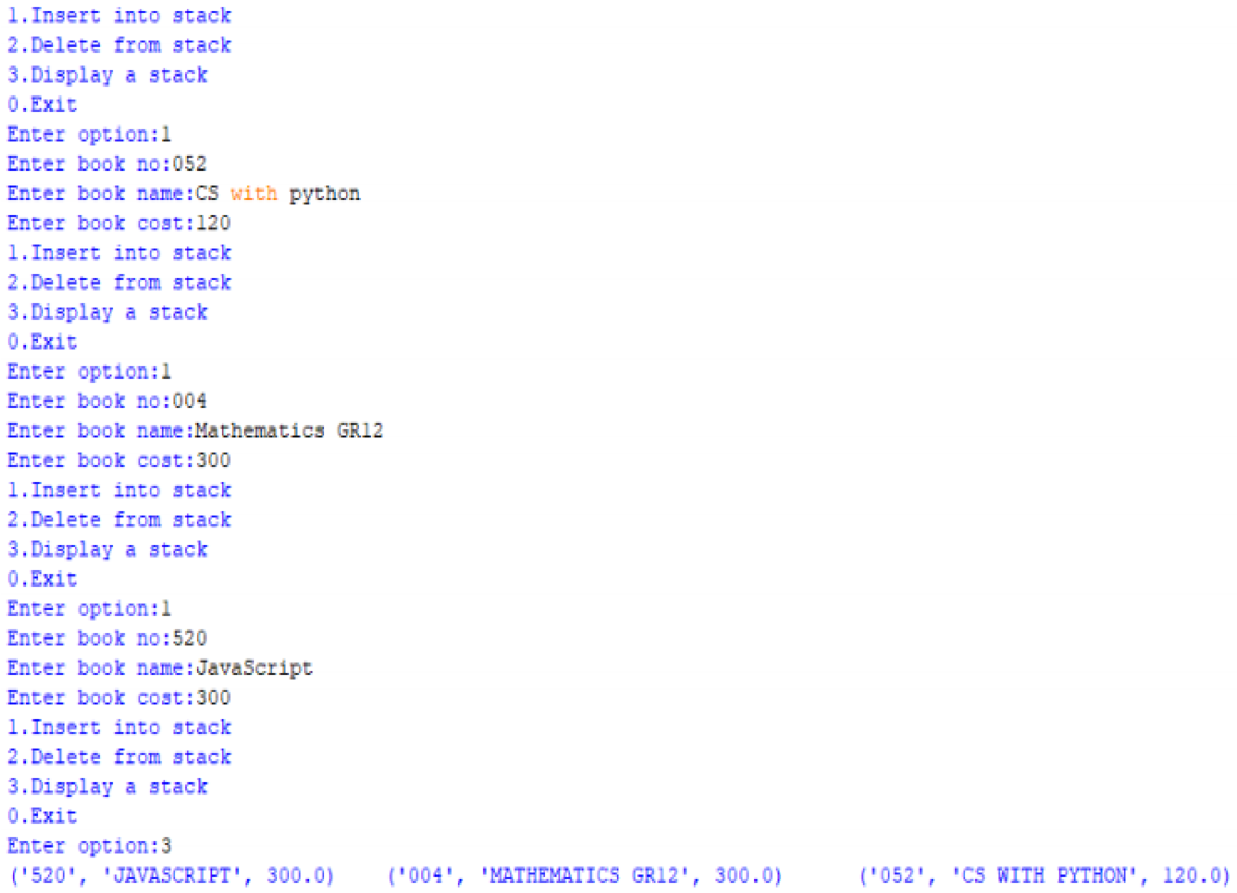
Concepts used: Stack.

Source code:

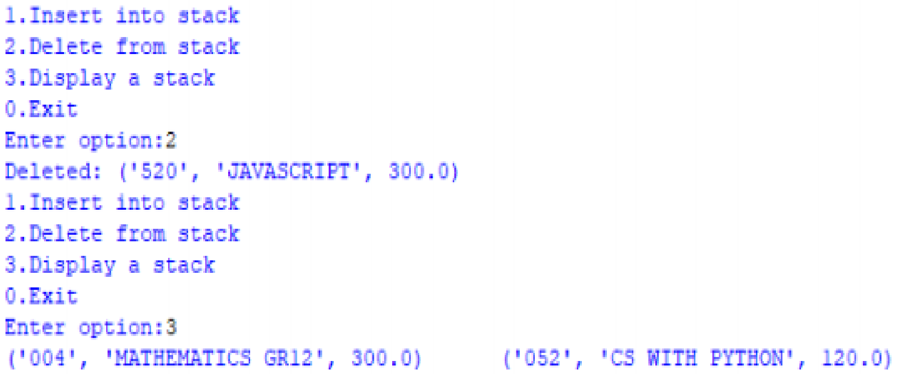


```
def Push(s,item):
    s.append(item)
def Pop(s):
    if s==[]:
        print('Empty')
    else:
        print('Deleted:',s.pop())
def Display(s):
    n=len(s)
    if s==[]:
        'Stack Empty'
    else:
        n=len(s)
        while n>0:
            print(s[n-1],end='\t')
            n=n-1
        print()
list1=[]
while True:

    print('1.Insert into stack\n2.Delete from stack\n3.Display a stack\n0.Exit'
    ch=int(input('Enter option:'))

    if ch==1:
        bno=input('Enter book no:')
        bname=input('Enter book name:')
        cost=float(input('Enter book cost:'))
        val=(bno.upper(),bname.upper(),cost)
        Push(list1,val)
    elif ch==2:
        Pop(list1)
    elif ch==3:
        Display(list1)
    elif ch==0:
        exit()
    else:
        print('Invalid option')
```

Output:



```
1.Insert into stack
2.Delete from stack
3.Display a stack
0.Exit
Enter option:1
Enter book no:052
Enter book name:CS with python
Enter book cost:120
1.Insert into stack
2.Delete from stack
3.Display a stack
0.Exit
Enter option:1
Enter book no:004
Enter book name:Mathematics GR12
Enter book cost:300
1.Insert into stack
2.Delete from stack
3.Display a stack
0.Exit
Enter option:1
Enter book no:520
Enter book name:JavaScript
Enter book cost:300
1.Insert into stack
2.Delete from stack
3.Display a stack
0.Exit
Enter option:3
('520', 'JAVASCRIPT', 300.0)    ('004', 'MATHEMATICS GR12', 300.0)      ('052', 'CS WITH PYTHON', 120.0)
```



```
1.Insert into stack
2.Delete from stack
3.Display a stack
0.Exit
Enter option:2
Deleted: ('520', 'JAVASCRIPT', 300.0)
1.Insert into stack
2.Delete from stack
3.Display a stack
0.Exit
Enter option:3
('004', 'MATHEMATICS GR12', 300.0)      ('052', 'CS WITH PYTHON', 120.0)
```

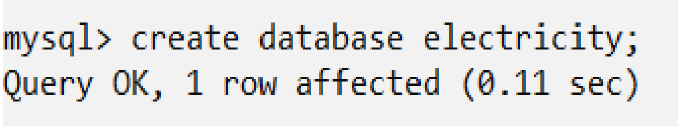
RECORD FILE – E.1

Objective: To create a database ELECTRICITY with table EBILL, and generate output of the following queries in MySQL.

Concepts Used: Relational Database Management implemented through MySQL.

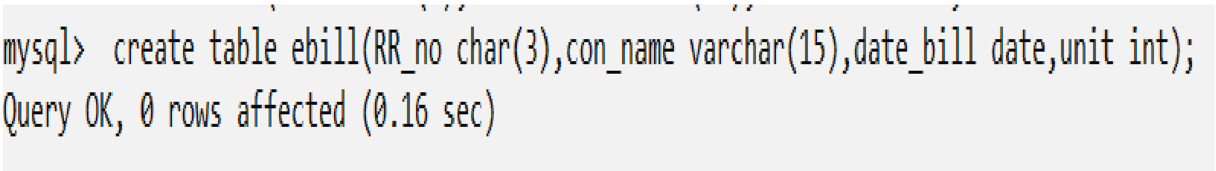
Queries:

1. **Create a database Electricity.**



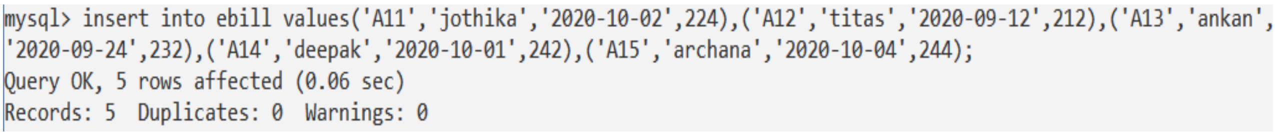
```
mysql> create database electricity;
Query OK, 1 row affected (0.11 sec)
```

2. **Design table Ebill (RR_No, Con_Name, Date_Bill, Unit).**



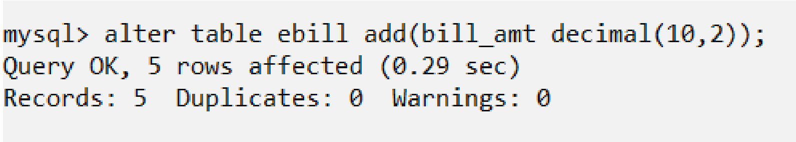
```
mysql>  create table ebill(RR_no char(3),con_name varchar(15),date_bill date,unit int);
Query OK, 0 rows affected (0.16 sec)
```

3. **Insert 5 records in to the table.**



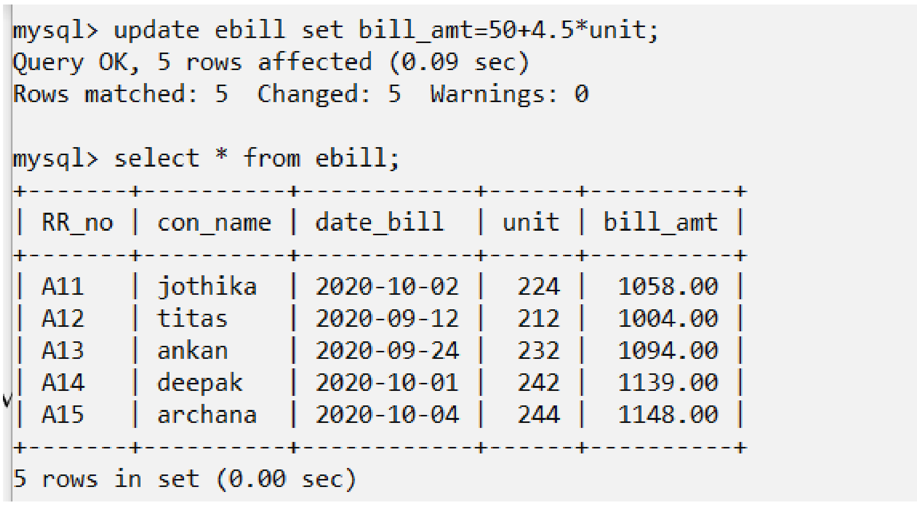
```
mysql> insert into ebill values('A11','jothika','2020-10-02',224),('A12','titas','2020-09-12',212),('A13','ankan',
'2020-09-24',232),('A14','deepak','2020-10-01',242),('A15','archana','2020-10-04',244);
Query OK, 5 rows affected (0.06 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

4. **Add new field to the table-Bill_amt (decimal – 10,2).**



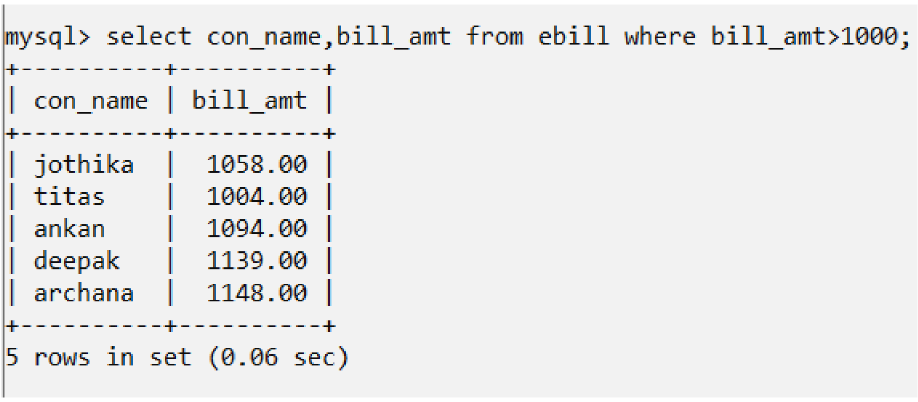
```
mysql> alter table ebill add(bill_amt decimal(10,2));
Query OK, 5 rows affected (0.29 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

5. **Compute the bill amount for each customer as MinAmt + 4.50 per unit (MinAmt = 50).**



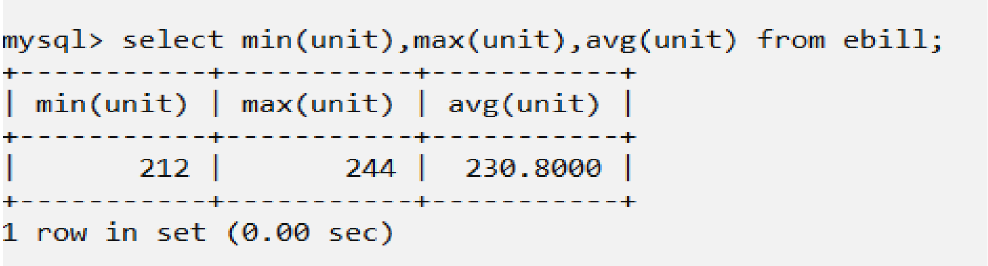
```
mysql> update ebill set bill_amt=50+4.5*unit;
Query OK, 5 rows affected (0.09 sec)
Rows matched: 5  Changed: 5  Warnings: 0

mysql> select * from ebill;
+-------+----------+------------+------+----------+
| RR_no | con_name | date_bill  | unit | bill_amt |
+-------+----------+------------+------+----------+
| A11   | jothika  | 2020-10-02 |  224 |  1058.00 |
| A12   | titas    | 2020-09-12 |  212 |  1004.00 |
| A13   | ankan    | 2020-09-24 |  232 |  1094.00 |
| A14   | deepak   | 2020-10-01 |  242 |  1139.00 |
| A15   | archana  | 2020-10-04 |  244 |  1148.00 |
+-------+----------+------------+------+----------+
5 rows in set (0.00 sec)
```

6. **Display Customer name and Bill Amount where Bill Amount > 1000.**



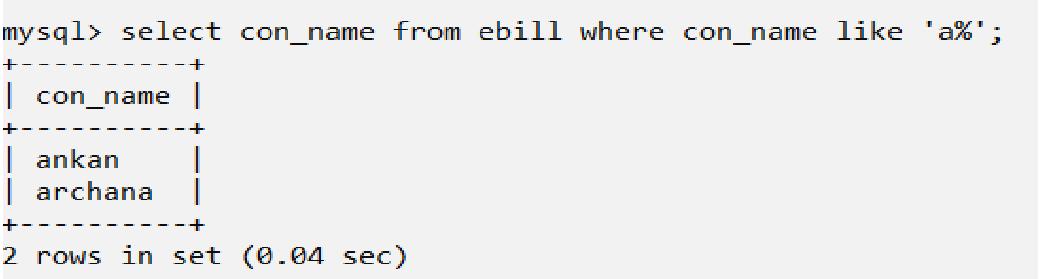
```
mysql> select con_name,bill_amt from ebill where bill_amt>1000;
+----------+----------+
| con_name | bill_amt |
+----------+----------+
| jothika  |  1058.00 |
| titas    |  1004.00 |
| ankan    |  1094.00 |
| deepak   |  1139.00 |
| archana  |  1148.00 |
+----------+----------+
5 rows in set (0.06 sec)
```

7. **Display the min, max , sum and average of Unit.**



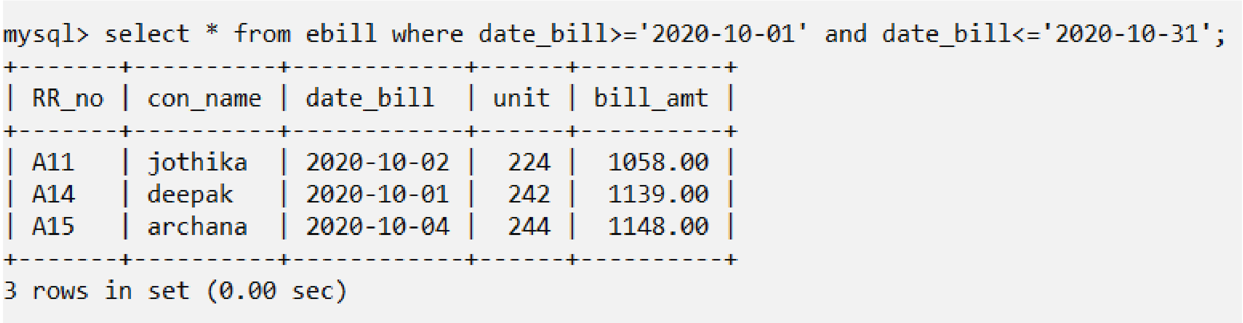
```
mysql> select min(unit),max(unit),avg(unit) from ebill;
+-----------+-----------+-----------+
| min(unit) | max(unit) | avg(unit) |
+-----------+-----------+-----------+
|       212 |       244 |  230.8000 |
+-----------+-----------+-----------+
1 row in set (0.00 sec)
```

8. **Display details of Customers whose name starts with ‘A’.**



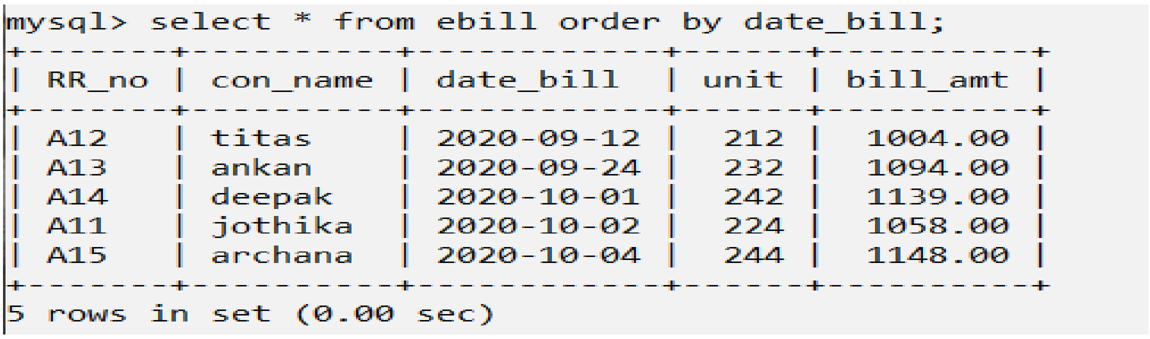
```
mysql> select con_name from ebill where con_name like 'a%';
+----------+
| con_name |
+----------+
| ankan    |
| archana  |
+----------+
2 rows in set (0.04 sec)
```

9. **Display Bills generated in the month of October.**



```
mysql> select * from ebill where date_bill>='2020-10-01' and date_bill<='2020-10-31';
+-------+----------+------------+------+----------+
| RR_no | con_name | date_bill  | unit | bill_amt |
+-------+----------+------------+------+----------+
| A11   | jothika  | 2020-10-02 |  224 |  1058.00 |
| A14   | deepak   | 2020-10-01 |  242 |  1139.00 |
| A15   | archana  | 2020-10-04 |  244 |  1148.00 |
+-------+----------+------------+------+----------+
3 rows in set (0.00 sec)
```

10. List all the bills generated in ascending order of Bill Date.



```
mysql> select * from ebill order by date_bill;
+--------+----------+------------+------+----------+
| RR_no  | con_name | date_bill  | unit | bill_amt |
+--------+----------+------------+------+----------+
| A12    | titas    | 2020-09-12 |  212 |  1004.00 |
| A13    | ankan    | 2020-09-24 |  232 |  1094.00 |
| A14    | deepak   | 2020-10-01 |  242 |  1139.00 |
| A11    | jothika  | 2020-10-02 |  224 |  1058.00 |
| A15    | archana  | 2020-10-04 |  244 |  1148.00 |
+--------+----------+------------+------+----------+
5 rows in set (0.00 sec)
```

RECORD FILE – E.2

Objective: To create a database in MySQL & generate output of the following queries.

Concepts used: Sequential Query Language (SQL).

Queries:

1. **Design database STORE with tables:**
   .PRODUCTS (PID, PName, Qty, Price, Company, Supcode)
   .SUPPLIERS (SupCode, Sname, Qty)
2. **Insert into values into given table as shown:**

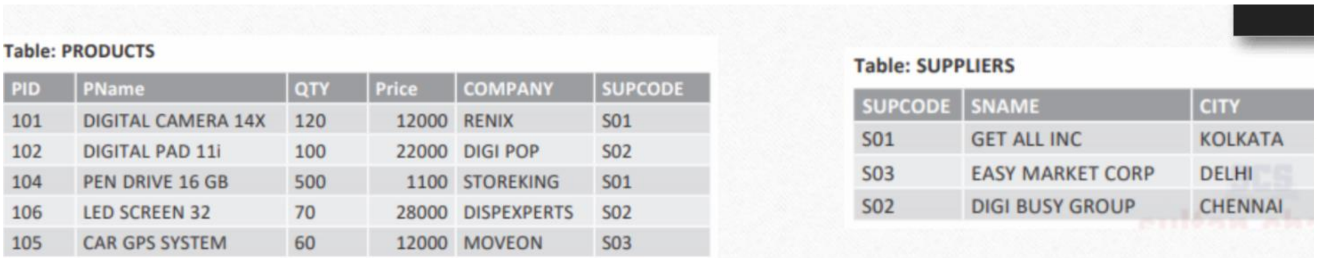
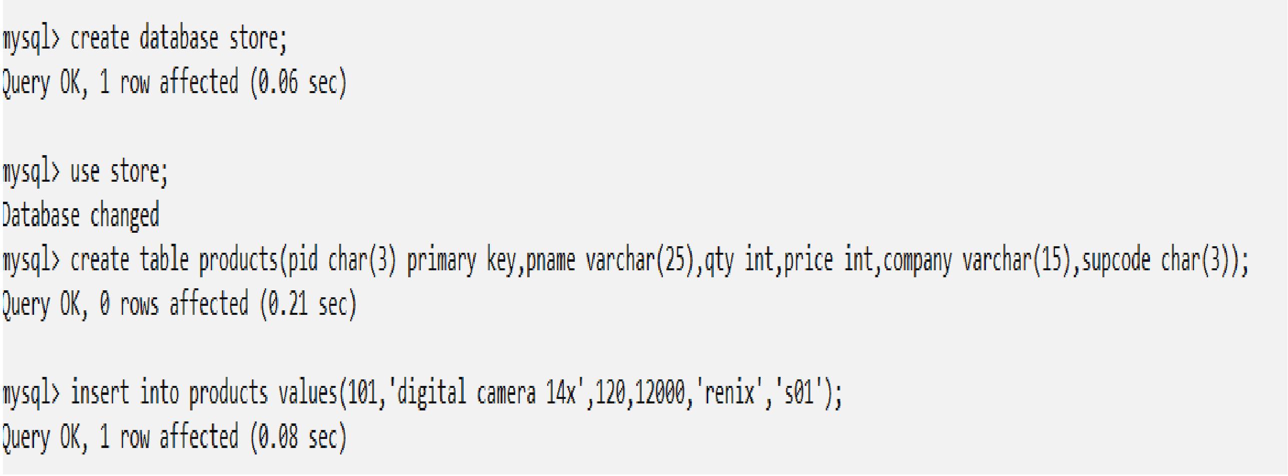


Table: PRODUCTS

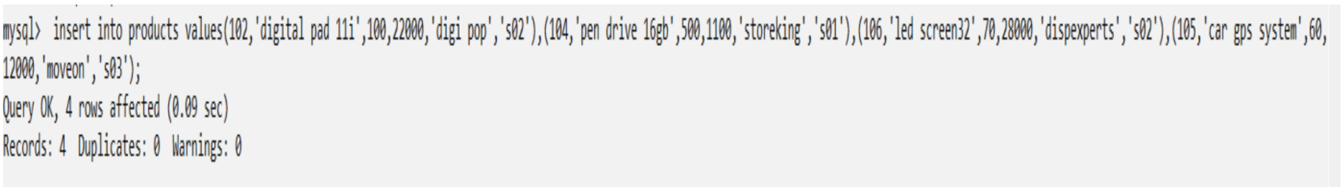
PID	PName	QTY	Price	COMPANY	SUPCODE
101	DIGITAL CAMERA 14X	120	12000	RENIX	S01
102	DIGITAL PAD 11i	100	22000	DIGI POP	S02
104	PEN DRIVE 16 GB	500	1100	STOREKING	S01
106	LED SCREEN 32	70	28000	DISPEXPERTS	S02
105	CAR GPS SYSTEM	60	12000	MOVEON	S03

Table: SUPPLIERS

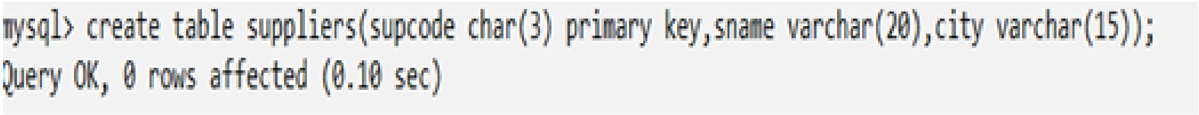
SUPCODE	SNAME	CITY
S01	GET ALL INC	KOLKATA
S03	EASY MARKET CORP	DELHI
S02	DIGI BUSY GROUP	CHENNAI



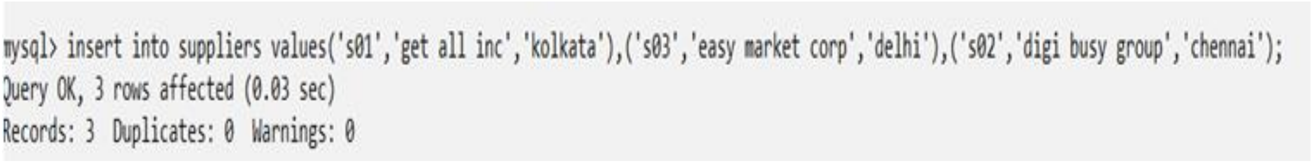
```
mysql> create database store;
Query OK, 1 row affected (0.06 sec)

mysql> use store;
Database changed
mysql> create table products(pid char(3) primary key,pname varchar(25),qty int,price int,company varchar(15),supcode char(3));
Query OK, 0 rows affected (0.21 sec)

mysql> insert into products values(101,'digital camera 14x',120,12000,'renix','s01');
Query OK, 1 row affected (0.08 sec)
```



```
mysql>  insert into products values(102,'digital pad 11i',100,22000,'digi pop','s02'),(104,'pen drive 16gb',500,1100,'storeking','s01'),(106,'led screen32',70,28000,'dispexperts','s02'),(105,'car gps system',60,
12000,'moveon','s03');
Query OK, 4 rows affected (0.09 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

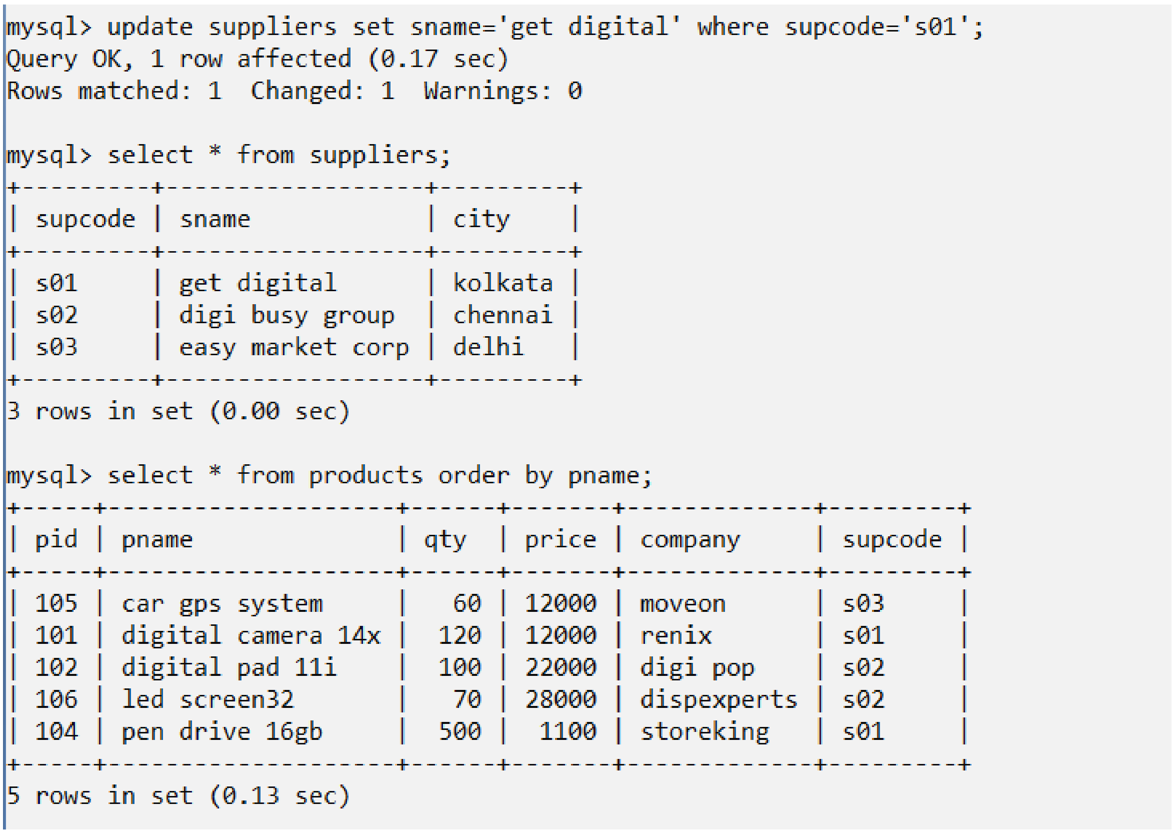


```
mysql> create table suppliers(supcode char(3) primary key,sname varchar(20),city varchar(15));
Query OK, 0 rows affected (0.10 sec)
```



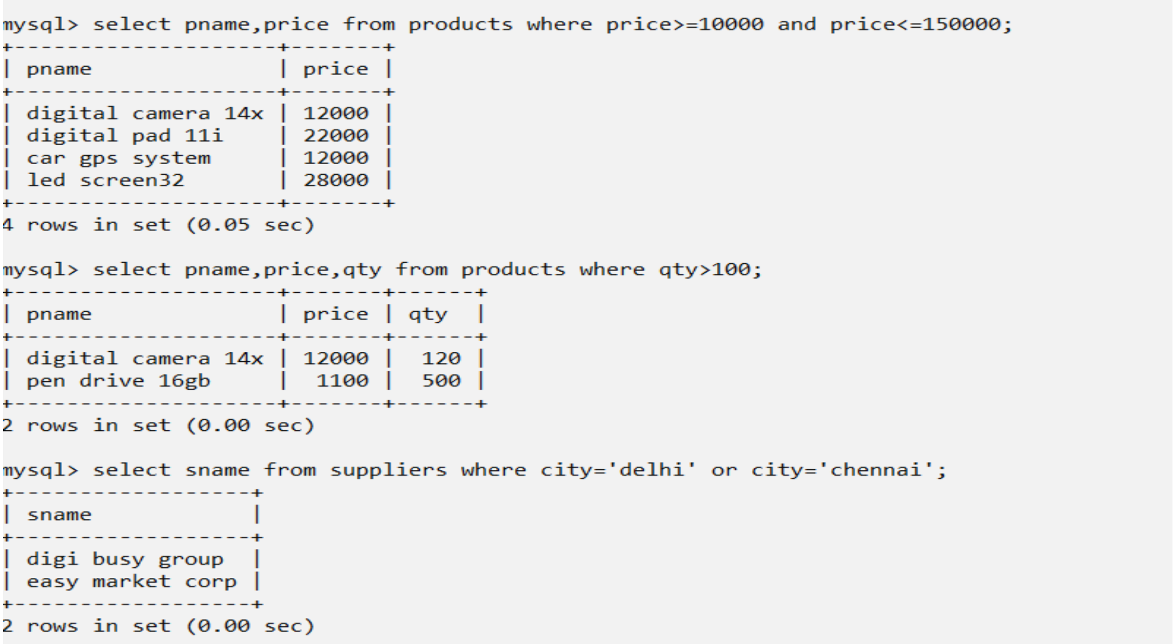
```
mysql> insert into suppliers values('s01','get all inc','kolkata'),('s03','easy market corp','delhi'),('s02','digi busy group','chennai');
Query OK, 3 rows affected (0.03 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

3. **Change name of Supplier ‘S01’ from ‘Get All Inc’ to ‘Get Digital’.**
4. **Display all records of PRODUCTS in ascending order of Product Name.**



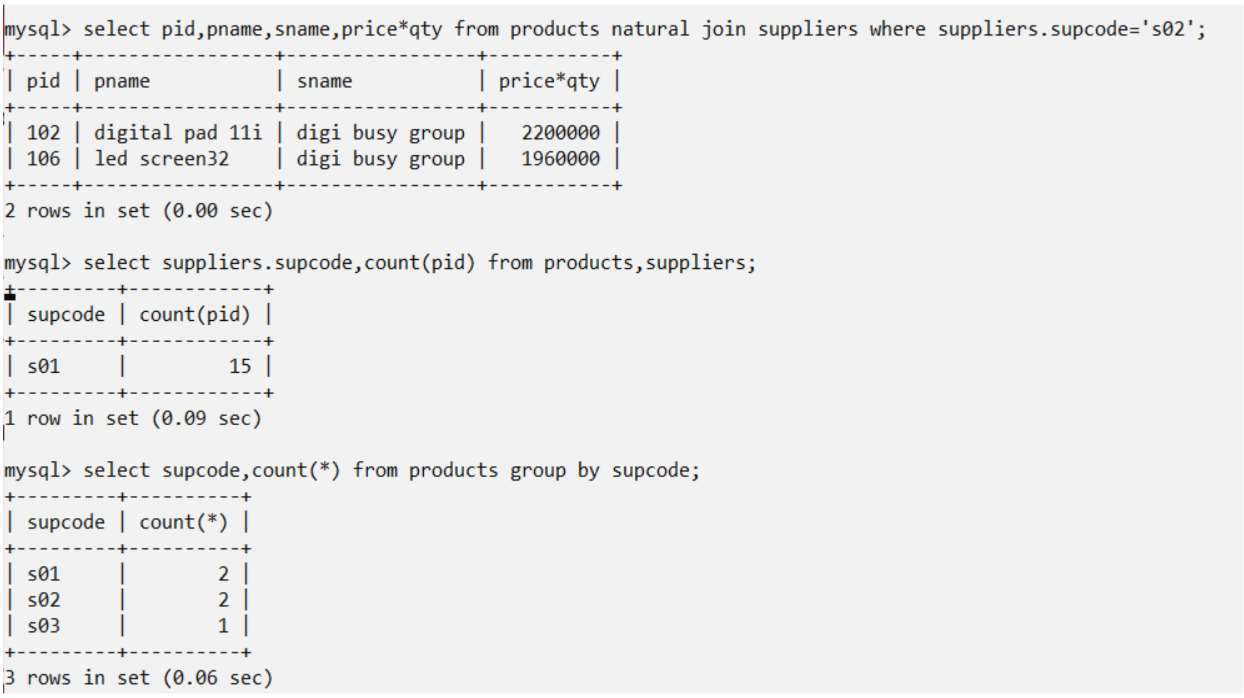
```
mysql> update suppliers set sname='get digital' where supcode='s01';
Query OK, 1 row affected (0.17 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from suppliers;
+---------+------------------+---------+
| supcode | sname            | city    |
+---------+------------------+---------+
| s01     | get digital      | kolkata |
| s02     | digi busy group  | chennai |
| s03     | easy market corp | delhi   |
+---------+------------------+---------+
3 rows in set (0.00 sec)

mysql> select * from products order by pname;
+-----+--------------------+------+-------+-------------+---------+
| pid | pname              | qty  | price | company     | supcode |
+-----+--------------------+------+-------+-------------+---------+
| 105 | car gps system     |   60 | 12000 | moveon      | s03     |
| 101 | digital camera 14x |  120 | 12000 | renix       | s01     |
| 102 | digital pad 11i    |  100 | 22000 | digi pop    | s02     |
| 106 | led screen32       |   70 | 28000 | dispexperts | s02     |
| 104 | pen drive 16gb     |  500 |  1100 | storeking   | s01     |
+-----+--------------------+------+-------+-------------+---------+
5 rows in set (0.13 sec)
```

5. **Display Product Name, Price of all products in the price range 10000 to 150000 (inclusive of both values).**
6. **Display the Product Name, Price & Quantity of all products with Quantity more than 100.**
7. **Display names of Suppliers from Delhi and Chennai.**



```
mysql> select pname,price from products where price>=10000 and price<=150000;
+--------------------+-------+
| pname              | price |
+--------------------+-------+
| digital camera 14x | 12000 |
| digital pad 11i    | 22000 |
| car gps system     | 12000 |
| led screen32       | 28000 |
+--------------------+-------+
4 rows in set (0.05 sec)

mysql> select pname,price,qty from products where qty>100;
+--------------------+-------+------+
| pname              | price | qty  |
+--------------------+-------+------+
| digital camera 14x | 12000 |  120 |
| pen drive 16gb     |  1100 |  500 |
+--------------------+-------+------+
2 rows in set (0.00 sec)

mysql> select sname from suppliers where city='delhi' or city='chennai';
+-------------------+
| sname             |
+-------------------+
| digi busy group   |
| easy market corp  |
+-------------------+
2 rows in set (0.00 sec)
```

8. **Display names of Product Name, Supplier Name, City from Kolkata.**
9. **Display Product ID, Product Name, Supplier Name and Price*Quantity of all products from Supplier 'S02'.**
10. **Display the number of Products supplied by each Supplier.**



```
mysql> select pid,pname,sname,price*qty from products natural join suppliers where suppliers.supcode='s02';
+-----+-----------------+-----------------+-----------+
| pid | pname           | sname           | price*qty |
+-----+-----------------+-----------------+-----------+
| 102 | digital pad 11i | digi busy group |   2200000 |
| 106 | led screen32    | digi busy group |   1960000 |
+-----+-----------------+-----------------+-----------+
2 rows in set (0.00 sec)

mysql> select suppliers.supcode,count(pid) from products,suppliers;
+---------+------------+
| supcode | count(pid) |
+---------+------------+
| s01     |         15 |
+---------+------------+
1 row in set (0.09 sec)

mysql> select supcode,count(*) from products group by supcode;
+---------+----------+
| supcode | count(*) |
+---------+----------+
| s01     |        2 |
| s02     |        2 |
| s03     |        1 |
+---------+----------+
3 rows in set (0.06 sec)
```

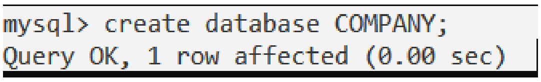
RECORD FILE – E.3

Objective: To create a database in MySQL & create an API using Python Connectivity.

Concept used: Python – MySQL connectivity.

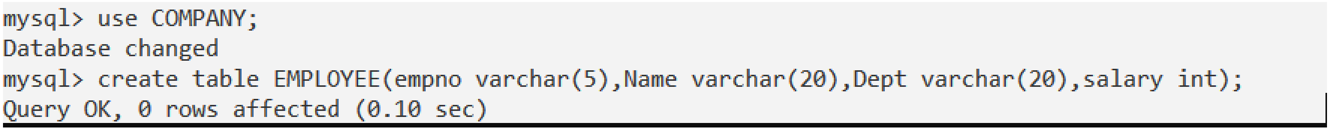
Queries:

1. Create a database COMPANY in MySQL.



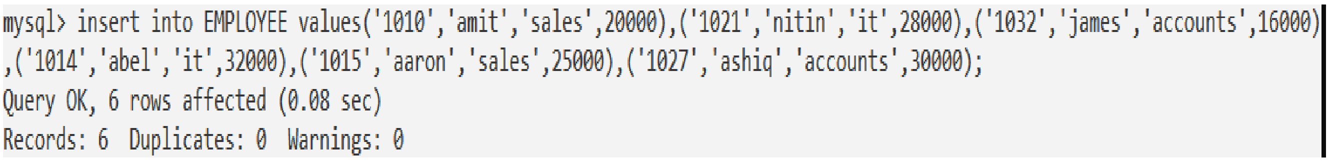
```
mysql> create database COMPANY;
Query OK, 1 row affected (0.00 sec)
```

2. Design table EMPLOYEE with:
.empno varcar(5)
.Name varchar(20)
.Dept varchar(20)
.Salary int



```
mysql> use COMPANY;
Database changed
mysql> create table EMPLOYEE(empno varchar(5),Name varchar(20),Dept varchar(20),salary int);
Query OK, 0 rows affected (0.10 sec)
```

3. Insert values in the table.



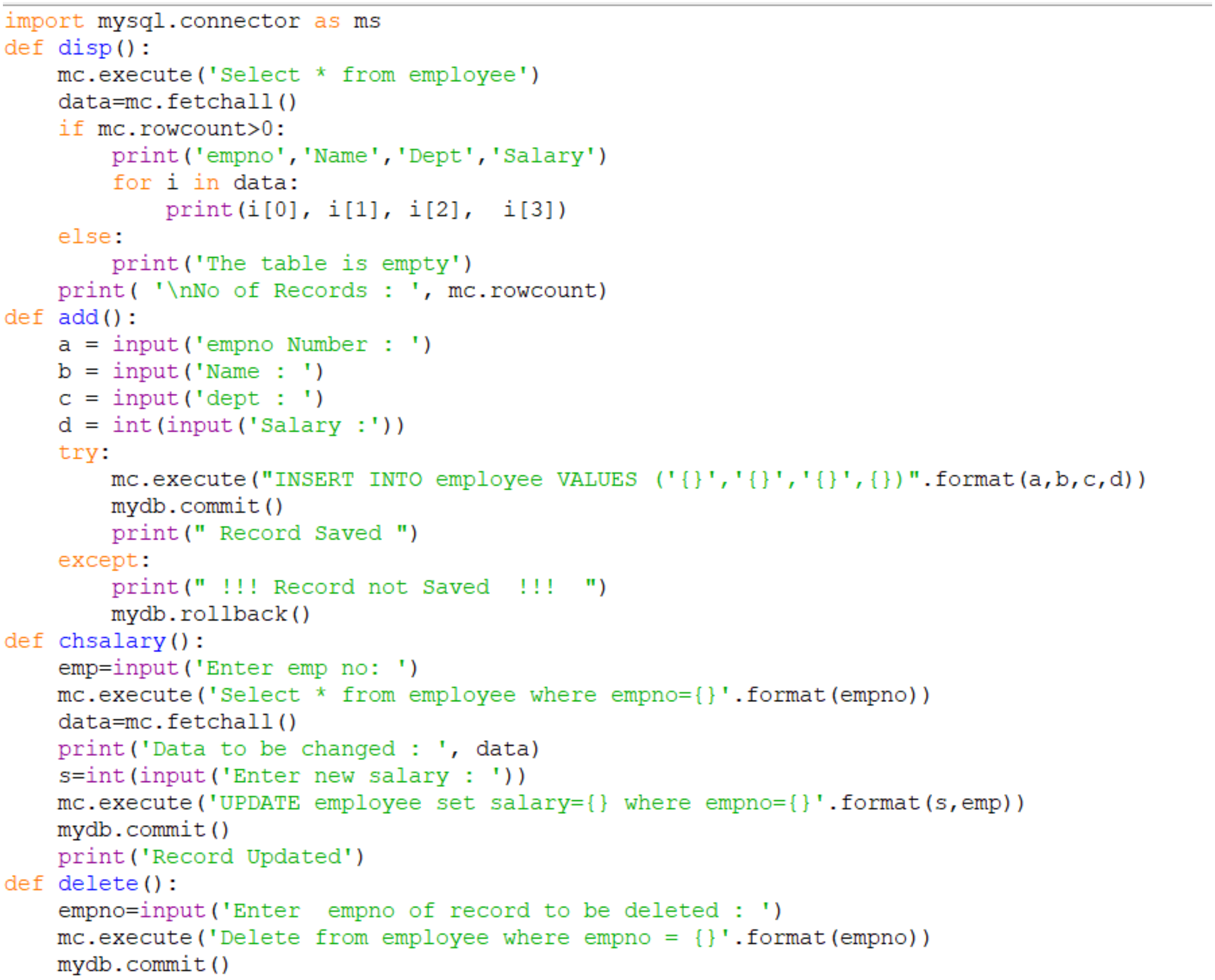
```
mysql> insert into EMPLOYEE values('1010','amit','sales',20000),('1021','nitin','it',28000),('1032','james','accounts',16000)
,('1014','abel','it',32000),('1015','aaron','sales',25000),('1027','ashiq','accounts',30000);
Query OK, 6 rows affected (0.08 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

4. Connect the db to Python

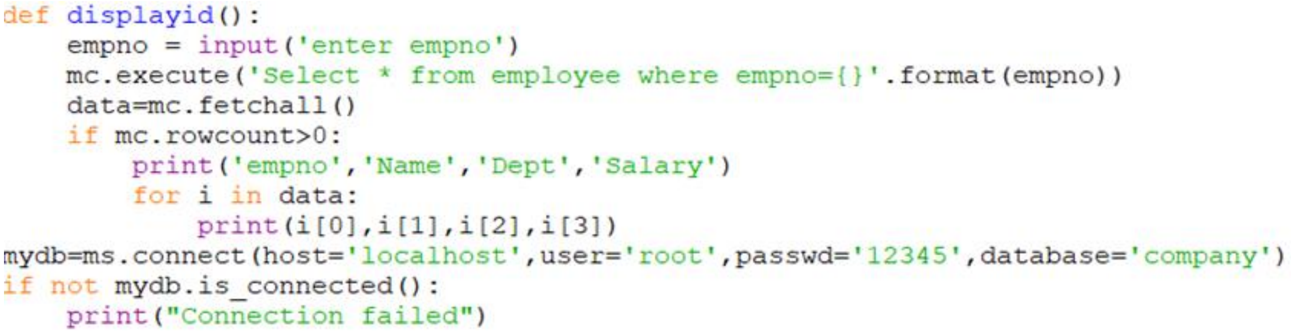
5. Write a menu driven program to:

a. ACCEPT new employee details and display all records.
b. DISPLAY employee details by employee number; display appropriate message if employee number not matched.
c. UPDATE the Salary by employee number.
d. DELETE Record by employee number.

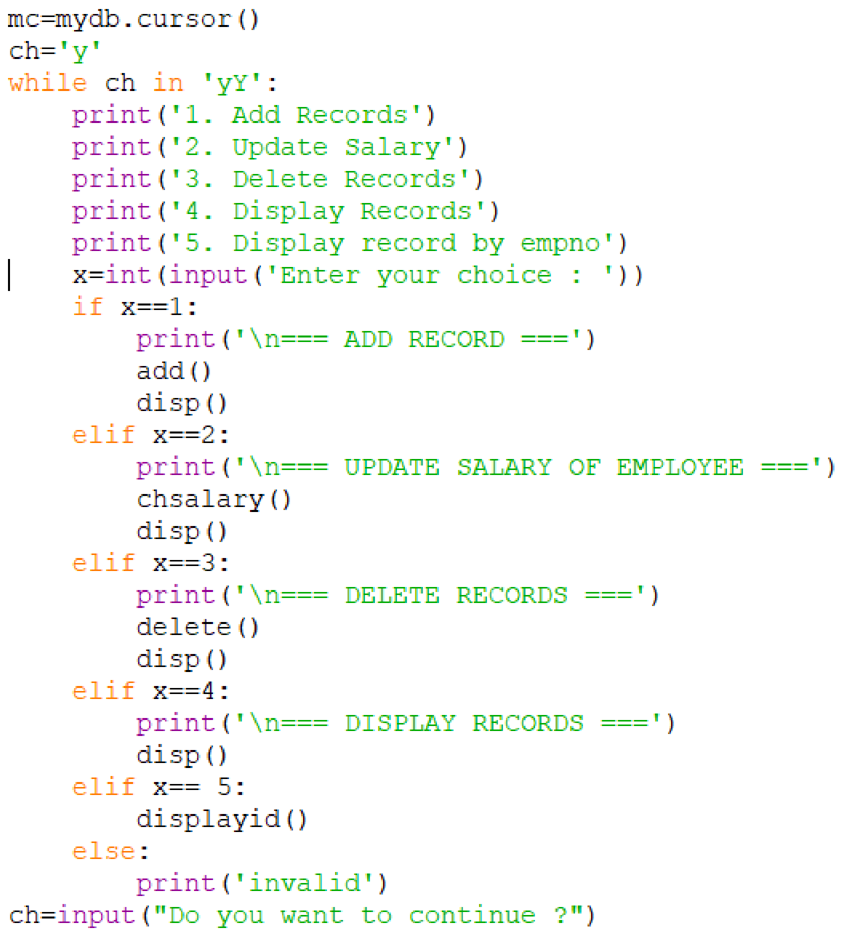
Source Code:



```
import mysql.connector as ms
def disp():
    mc.execute('Select * from employee')
    data=mc.fetchall()
    if mc.rowcount>0:
        print('empno','Name','Dept','Salary')
        for i in data:
            print(i[0], i[1], i[2],  i[3])
    else:
        print('The table is empty')
    print( '\nNo of Records : ', mc.rowcount)
def add():
    a = input('empno Number : ')
    b = input('Name : ')
    c = input('dept : ')
    d = int(input('Salary :'))
    try:
        mc.execute("INSERT INTO employee VALUES ('{}','{}','{}',{})".format(a,b,c,d))
        mydb.commit()
        print(" Record Saved ")
    except:
        print(" !!! Record not Saved  !!!  ")
        mydb.rollback()
def chsalary():
    emp=input('Enter emp no: ')
    mc.execute('Select * from employee where empno={}'.format(empno))
    data=mc.fetchall()
    print('Data to be changed : ', data)
    s=int(input('Enter new salary : '))
    mc.execute('UPDATE employee set salary={} where empno={}'.format(s,emp))
    mydb.commit()
    print('Record Updated')
def delete():
    empno=input('Enter  empno of record to be deleted : ')
    mc.execute('Delete from employee where empno = {}'.format(empno))
    mydb.commit()
```

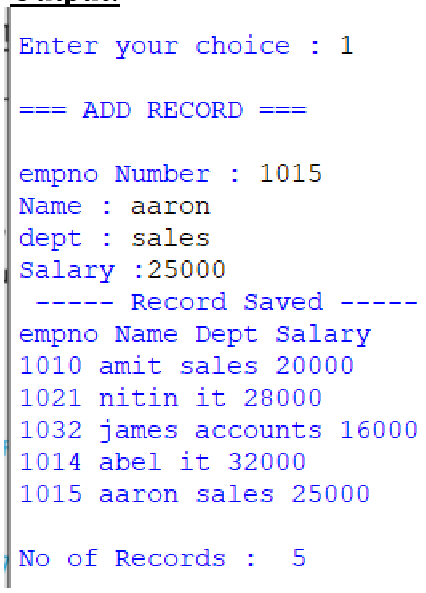


```
def displayid():
    empno = input('enter empno')
    mc.execute('Select * from employee where empno={}'.format(empno))
    data=mc.fetchall()
    if mc.rowcount>0:
        print('empno','Name','Dept','Salary')
        for i in data:
            print(i[0],i[1],i[2],i[3])
mydb=ms.connect(host='localhost',user='root',passwd='12345',database='company')
if not mydb.is_connected():
    print("Connection failed")
```

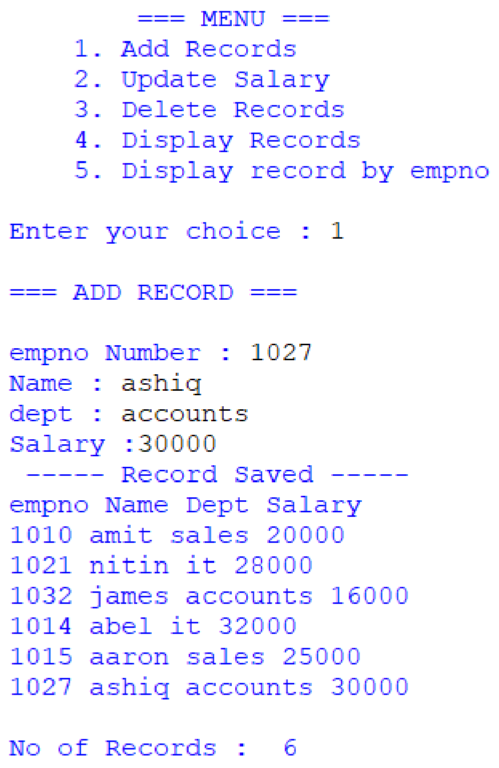


```
mc=mydb.cursor()
ch='y'
while ch in 'yY':
    print('1. Add Records')
    print('2. Update Salary')
    print('3. Delete Records')
    print('4. Display Records')
    print('5. Display record by empno')
    x=int(input('Enter your choice : '))
    if x==1:
        print('\n=== ADD RECORD ===')
        add()
        disp()
    elif x==2:
        print('\n=== UPDATE SALARY OF EMPLOYEE ===')
        chsalary()
        disp()
    elif x==3:
        print('\n=== DELETE RECORDS ===')
        delete()
        disp()
    elif x==4:
        print('\n=== DISPLAY RECORDS ===')
        disp()
    elif x== 5:
        displayid()
    else:
        print('invalid')
ch=input("Do you want to continue ?")
```

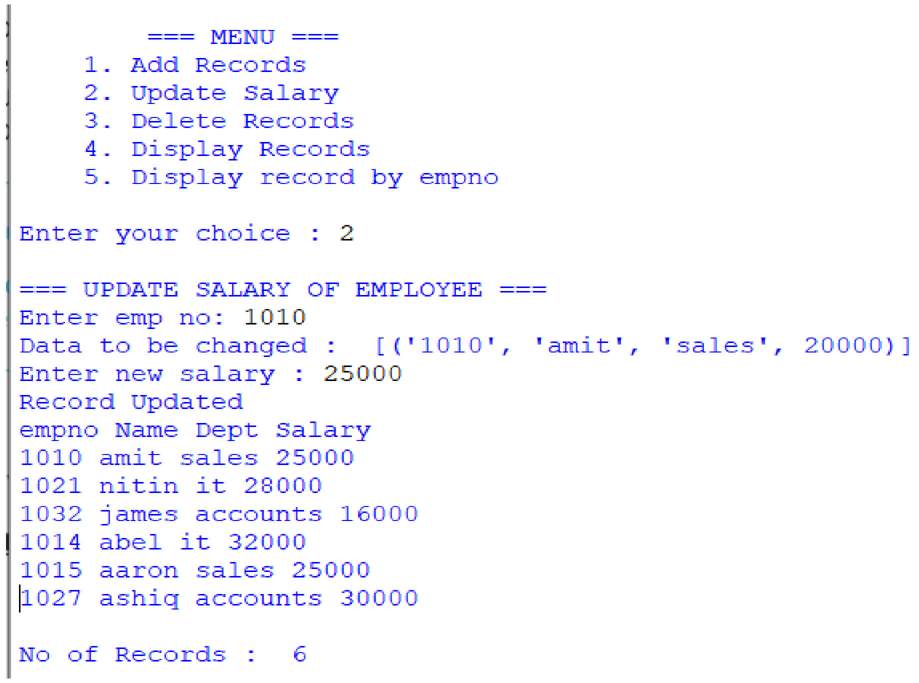
Output:



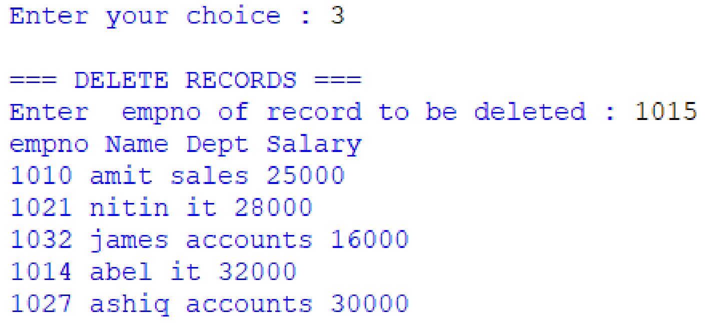
```
Enter your choice : 1

=== ADD RECORD ===

empno Number : 1015
Name : aaron
dept : sales
Salary :25000
 ----- Record Saved -----
empno Name Dept Salary
1010 amit sales 20000
1021 nitin it 28000
1032 james accounts 16000
1014 abel it 32000
1015 aaron sales 25000

No of Records :  5
```



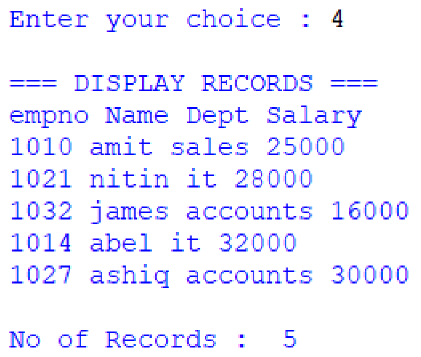
```
        === MENU ===
    1. Add Records
    2. Update Salary
    3. Delete Records
    4. Display Records
    5. Display record by empno

Enter your choice : 1

=== ADD RECORD ===

empno Number : 1027
Name : ashiq
dept : accounts
Salary :30000
 ----- Record Saved -----
empno Name Dept Salary
1010 amit sales 20000
1021 nitin it 28000
1032 james accounts 16000
1014 abel it 32000
1015 aaron sales 25000
1027 ashiq accounts 30000

No of Records :  6
```



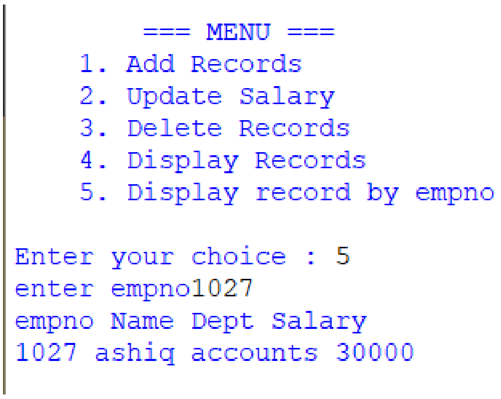
```
        === MENU ===
    1. Add Records
    2. Update Salary
    3. Delete Records
    4. Display Records
    5. Display record by empno

Enter your choice : 2

=== UPDATE SALARY OF EMPLOYEE ===
Enter emp no: 1010
Data to be changed :  [('1010', 'amit', 'sales', 20000)]
Enter new salary : 25000
Record Updated
empno Name Dept Salary
1010 amit sales 25000
1021 nitin it 28000
1032 james accounts 16000
1014 abel it 32000
1015 aaron sales 25000
1027 ashiq accounts 30000

No of Records :  6
```



```
Enter your choice : 3

=== DELETE RECORDS ===
Enter  empno of record to be deleted : 1015
empno Name Dept Salary
1010 amit sales 25000
1021 nitin it 28000
1032 james accounts 16000
1014 abel it 32000
1027 ashiq accounts 30000
```



```
Enter your choice : 4

=== DISPLAY RECORDS ===
empno Name Dept Salary
1010 amit sales 25000
1021 nitin it 28000
1032 james accounts 16000
1014 abel it 32000
1027 ashiq accounts 30000

No of Records :  5
```



```
        === MENU ===
    1. Add Records
    2. Update Salary
    3. Delete Records
    4. Display Records
    5. Display record by empno

Enter your choice : 5
enter empno1027
empno Name Dept Salary
1027 ashiq accounts 30000
```

RECORD FILE – E.4

Objective: To create a database in MySQL & create an API using Python Connectivity.

1. Create a database Record_E4 in MySQL.

2. Design table STUDENT with :

- **RollNo int**
- **Name varchar(20)**
- **Percentage decimal(4,1)**
- **Section char(1)**
- **Assignment varchar(15)**

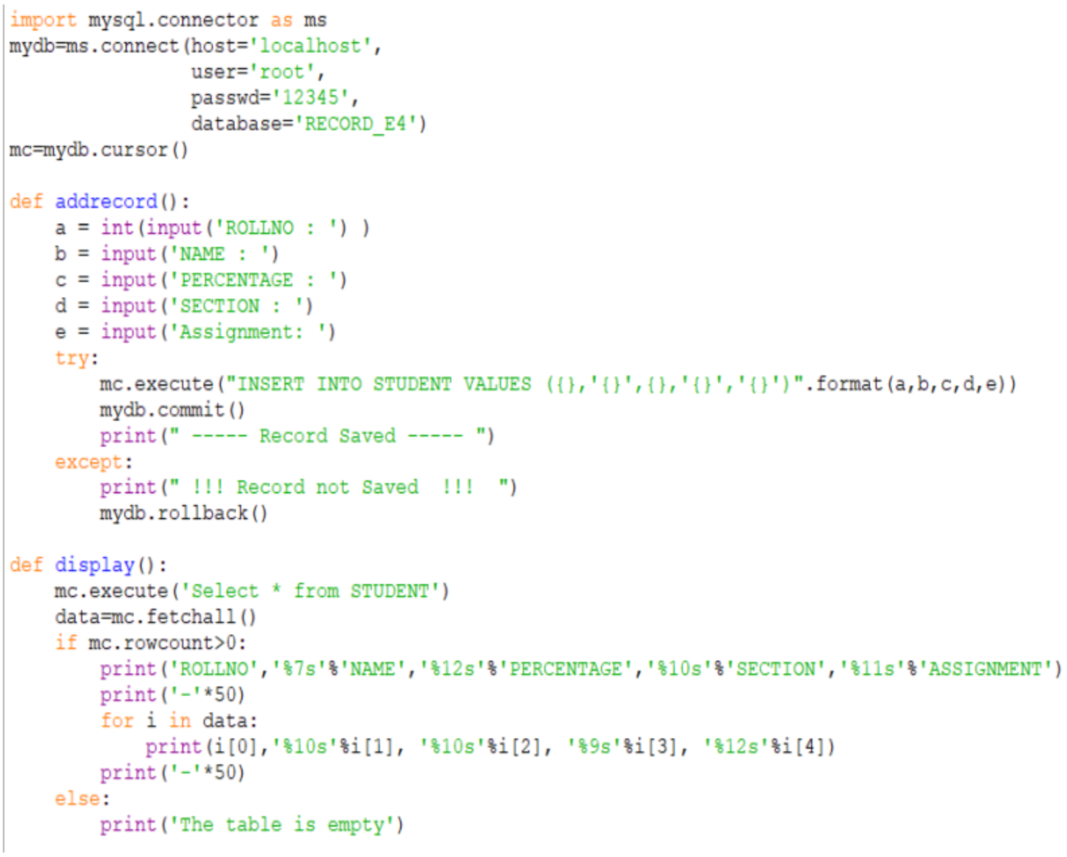
3. Insert values in the table.

Write a menu driven program to:

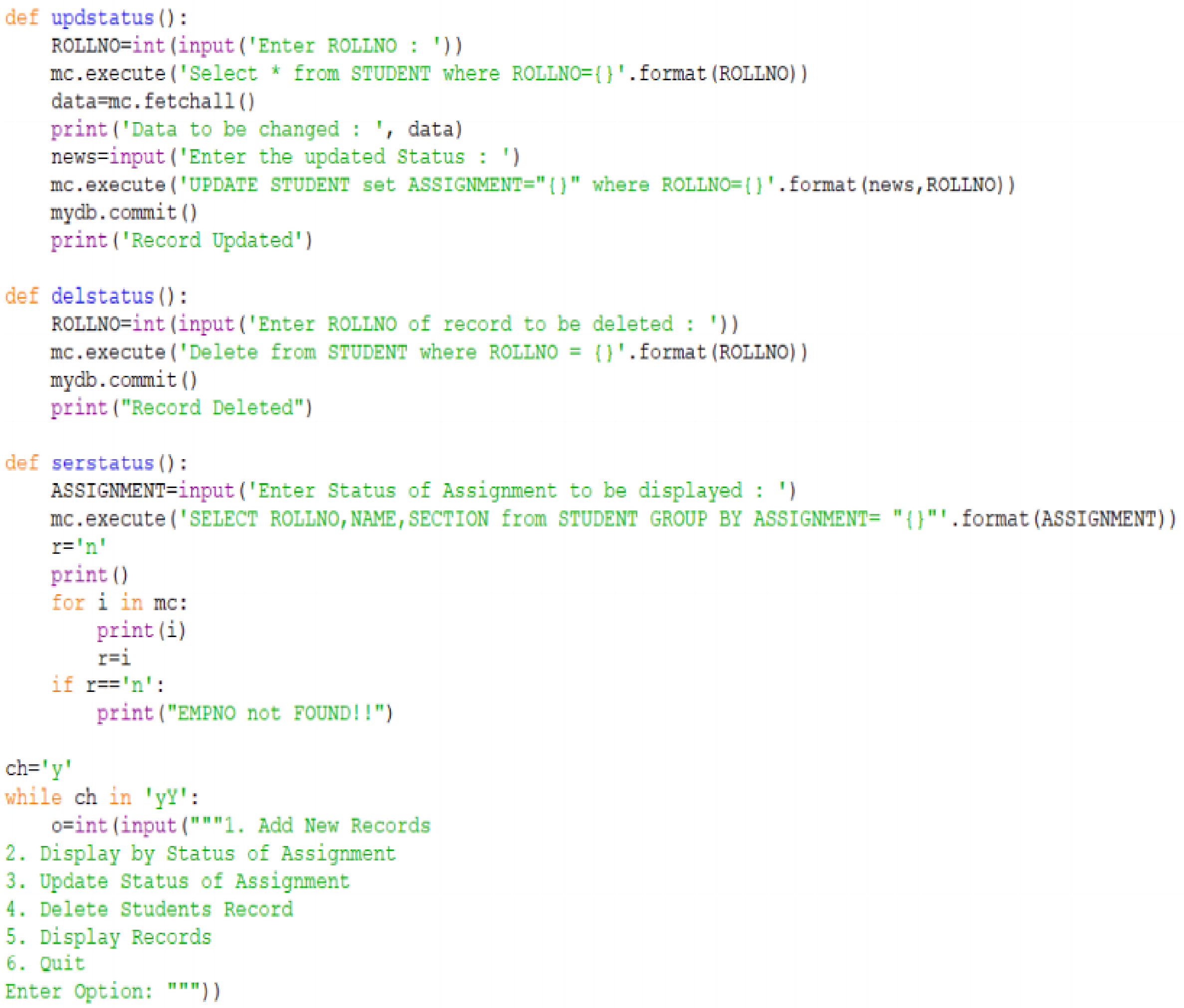
a. ACCEPT new student details
b. SEARCH & DISPLAY student details based on status of ASSIGNMENT
c. SEARCH & UPDATE field ASSIGNMENT
d. DELETE Record by student Roll No.
e. DISPLAY all Records.

Concept Used: Python - MySQL Connectivity.

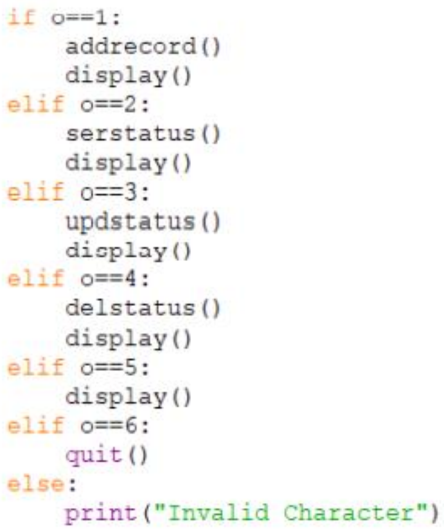
Source code:



```python
import mysql.connector as ms
mydb=ms.connect(host='localhost',
                user='root',
                passwd='12345',
                database='RECORD_E4')
mc=mydb.cursor()

def addrecord():
    a = int(input('ROLLNO : ') )
    b = input('NAME : ')
    c = input('PERCENTAGE : ')
    d = input('SECTION : ')
    e = input('Assignment: ')
    try:
        mc.execute("INSERT INTO STUDENT VALUES ({},'{}',{},'{}','{}')".format(a,b,c,d,e))
        mydb.commit()
        print(" ----- Record Saved ----- ")
    except:
        print(" !!! Record not Saved  !!!  ")
        mydb.rollback()

def display():
    mc.execute('Select * from STUDENT')
    data=mc.fetchall()
    if mc.rowcount>0:
        print('ROLLNO','%7s'%'NAME','%12s'%'PERCENTAGE','%10s'%'SECTION','%11s'%'ASSIGNMENT')
        print('-'*50)
        for i in data:
            print(i[0],'%10s'%i[1], '%10s'%i[2], '%9s'%i[3], '%12s'%i[4])
        print('-'*50)
    else:
        print('The table is empty')
```

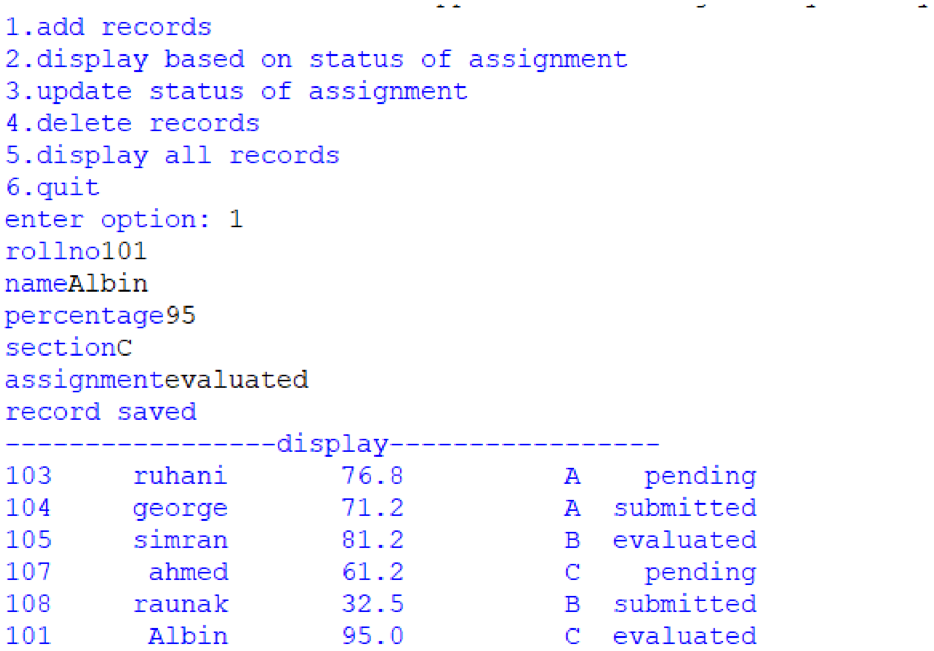


```
def updstatus():
    ROLLNO=int(input('Enter ROLLNO : '))
    mc.execute('Select * from STUDENT where ROLLNO={}'.format(ROLLNO))
    data=mc.fetchall()
    print('Data to be changed : ', data)
    news=input('Enter the updated Status : ')
    mc.execute('UPDATE STUDENT set ASSIGNMENT="{}" where ROLLNO={}'.format(news,ROLLNO))
    mydb.commit()
    print('Record Updated')

def delstatus():
    ROLLNO=int(input('Enter ROLLNO of record to be deleted : '))
    mc.execute('Delete from STUDENT where ROLLNO = {}'.format(ROLLNO))
    mydb.commit()
    print("Record Deleted")

def serstatus():
    ASSIGNMENT=input('Enter Status of Assignment to be displayed : ')
    mc.execute('SELECT ROLLNO,NAME,SECTION from STUDENT GROUP BY ASSIGNMENT= "{}"'.format(ASSIGNMENT))
    r='n'
    print()
    for i in mc:
        print(i)
        r=i
    if r=='n':
        print("EMPNO not FOUND!!")

ch='y'
while ch in 'yY':
    o=int(input("""1. Add New Records
2. Display by Status of Assignment
3. Update Status of Assignment
4. Delete Students Record
5. Display Records
6. Quit
Enter Option: """))
```

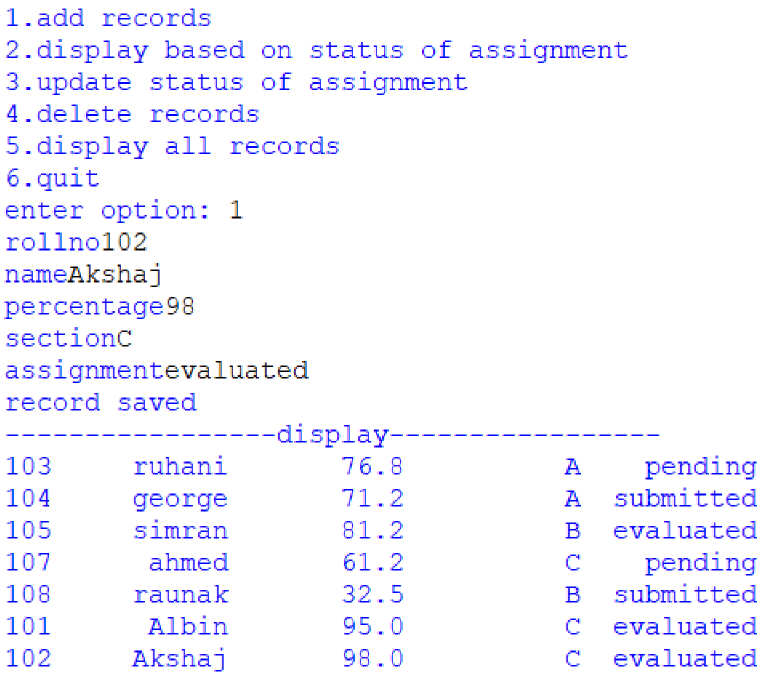


```
    if o==1:
        addrecord()
        display()
    elif o==2:
        serstatus()
        display()
    elif o==3:
        updstatus()
        display()
    elif o==4:
        delstatus()
        display()
    elif o==5:
        display()
    elif o==6:
        quit()
    else:
        print("Invalid Character")
```

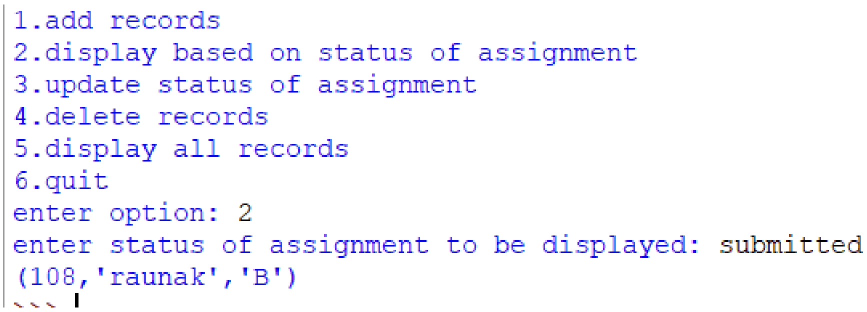
<u>Output:</u>



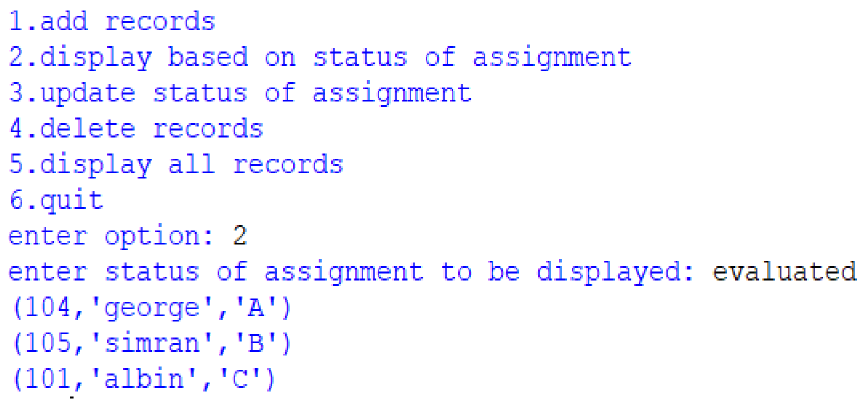
```
1.add records
2.display based on status of assignment
3.update status of assignment
4.delete records
5.display all records
6.quit
enter option: 1
rollno101
nameAlbin
percentage95
sectionC
assignmentevaluated
record saved
-----------------display-----------------
103     ruhani       76.8          A    pending
104     george       71.2          A  submitted
105     simran       81.2          B  evaluated
107      ahmed       61.2          C    pending
108     raunak       32.5          B  submitted
101      Albin       95.0          C  evaluated
```



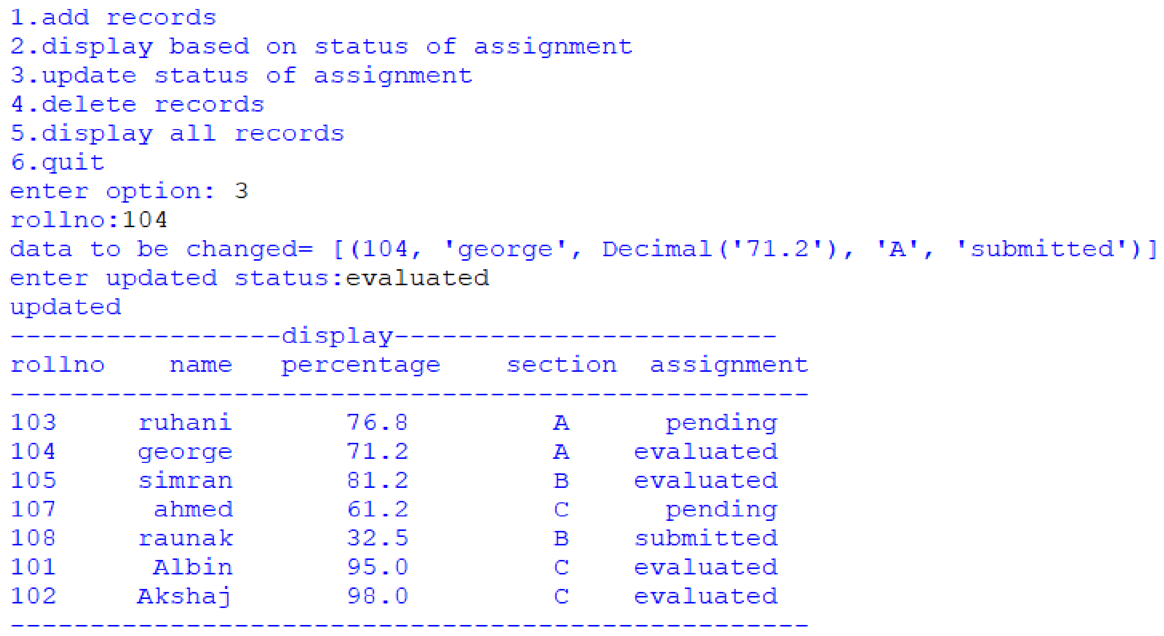
```
1.add records
2.display based on status of assignment
3.update status of assignment
4.delete records
5.display all records
6.quit
enter option: 1
rollno102
nameAkshaj
percentage98
sectionC
assignmentevaluated
record saved
-----------------display-----------------
103     ruhani       76.8          A    pending
104     george       71.2          A  submitted
105     simran       81.2          B  evaluated
107      ahmed       61.2          C    pending
108     raunak       32.5          B  submitted
101      Albin       95.0          C  evaluated
102     Akshaj       98.0          C  evaluated
```



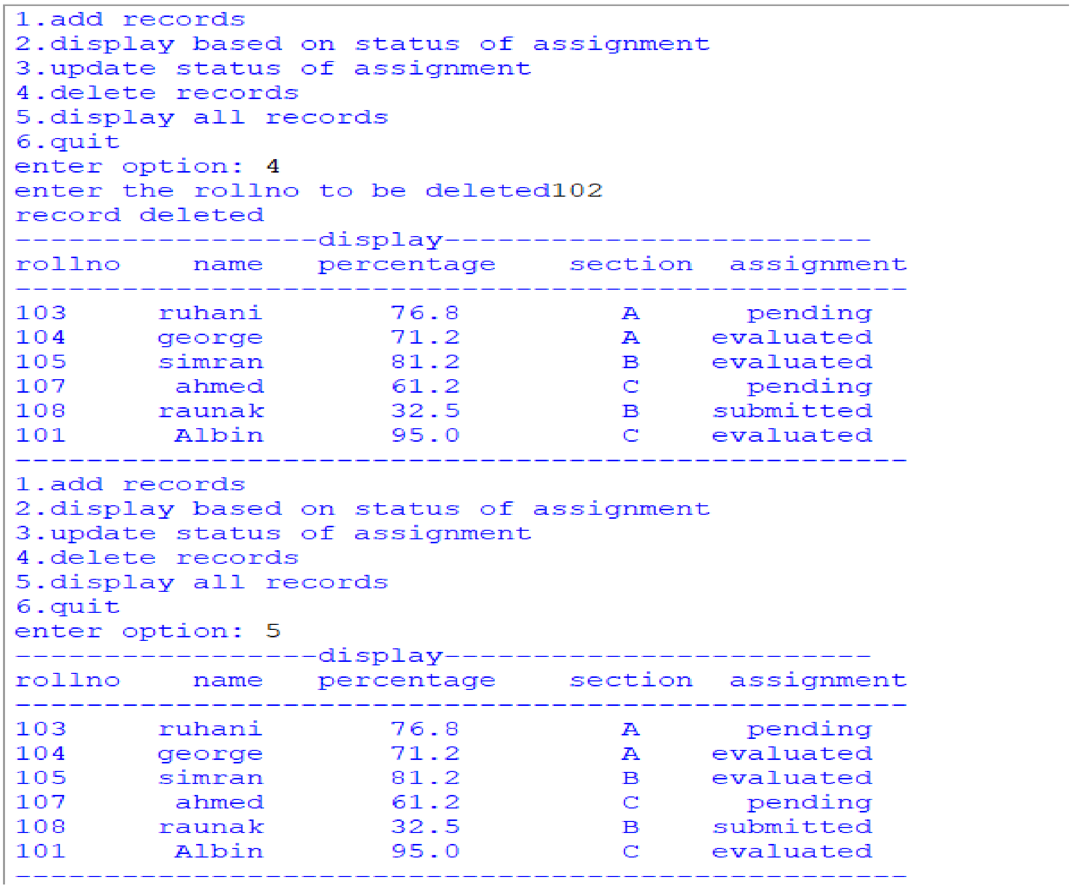
```
1.add records
2.display based on status of assignment
3.update status of assignment
4.delete records
5.display all records
6.quit
enter option: 2
enter status of assignment to be displayed: submitted
(108,'raunak','B')
```



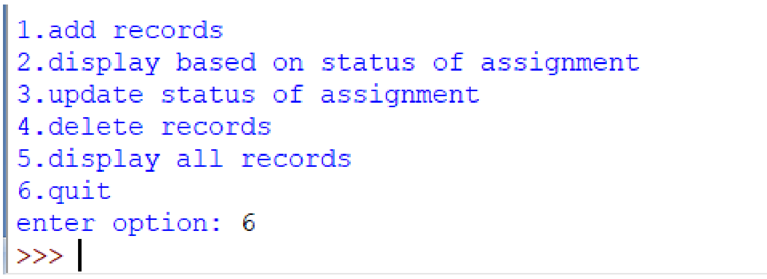
```
1.add records
2.display based on status of assignment
3.update status of assignment
4.delete records
5.display all records
6.quit
enter option: 2
enter status of assignment to be displayed: evaluated
(104,'george','A')
(105,'simran','B')
(101,'albin','C')
```



```
1.add records
2.display based on status of assignment
3.update status of assignment
4.delete records
5.display all records
6.quit
enter option: 3
rollno:104
data to be changed= [(104, 'george', Decimal('71.2'), 'A', 'submitted')]
enter updated status:evaluated
updated
-----------------display------------------------
rollno    name   percentage    section  assignment
--------------------------------------------------
103     ruhani       76.8         A      pending
104     george       71.2         A    evaluated
105     simran       81.2         B    evaluated
107      ahmed       61.2         C      pending
108     raunak       32.5         B    submitted
101      Albin       95.0         C    evaluated
102     Akshaj       98.0         C    evaluated
--------------------------------------------------
```



```
1.add records
2.display based on status of assignment
3.update status of assignment
4.delete records
5.display all records
6.quit
enter option: 4
enter the rollno to be deleted102
record deleted
-----------------display-----------------------
rollno    name   percentage    section  assignment
---------------------------------------------------
103     ruhani       76.8         A      pending
104     george       71.2         A    evaluated
105     simran       81.2         B    evaluated
107      ahmed       61.2         C      pending
108     raunak       32.5         B    submitted
101      Albin       95.0         C    evaluated
---------------------------------------------------
1.add records
2.display based on status of assignment
3.update status of assignment
4.delete records
5.display all records
6.quit
enter option: 5
-----------------display-----------------------
rollno    name   percentage    section  assignment
---------------------------------------------------
103     ruhani       76.8         A      pending
104     george       71.2         A    evaluated
105     simran       81.2         B    evaluated
107      ahmed       61.2         C      pending
108     raunak       32.5         B    submitted
101      Albin       95.0         C    evaluated
---------------------------------------------------
```



```
1.add records
2.display based on status of assignment
3.update status of assignment
4.delete records
5.display all records
6.quit
enter option: 6
>>> 
```