CLASS-XII

SUBJECT – COMPUTER SCIENCE (083)

PRACTICAL FILE SOLUTION

PRACTICAL NO.	OBJECTIVE & SOLUTION
1.	Write a program in python to check a number whether it is prime or not.

SOURCE CODE:	num=int(input("Enter the number: "))
	for i in range(2,num):
	  if num%i==0:
	  print(num, "is not prime number")
	  break;
	else:
	  print(num,"is prime number")

OUTPUT:	Enter the number: 13 13 is prime number
2.	Write a program to check a number whether it is palindrome or not.
SOURCE CODE:	num=int(input("Enter a number : ")) n=num res=0
	while num>0: rem=num%10 res=rem+res*10 num=num//10 if res==n: print("Number is Palindrome") else: print("Number is not Palindrome")
OUTPUT:	Enter a number : 6556 Number is Palindrome
3.	Write a program to calculate compound interest.
SOURCE CODE:	p=float(input("Enter the principal amount : ")) r=float(input("Enter the rate of interest : ")) t=float(input("Enter the time in years : ")) x=(1+r/100)**t CI= p*x-p print("Compound interest is : ", round(CI,2))

OUTPUT:	Enter the principal amount : 5000 Enter the rate of interest : 4 Enter the time in years : 2 Compound interest is : 408.0
4.	Write a program to display ASCII code of a character and vice versa.

SOURCE CODE:	var=True
	while var:
	choice=int(input("Press-1 to find the ordinal value of a character \nPress-2 to find a character of a value\n"))
	if choice==1:
	ch=input("Enter a character : ")
	print(ord(ch))
	elif choice==2:
	val=int(input("Enter an integer value: "))
	print(chr(val))
	else:
	print("You entered wrong choice")
	
	print("Do you want to continue? Y/N")
	option=input()
	if option=='y' or option=='Y':
	var=True
	else:

	var=False
OUTPUT:	Press-1 to find the ordinal value of a character Press-2 to find a character of a value 1 Enter a character : a 97 Do you want to continue? Y/N Y Press-1 to find the ordinal value of a character Press-2 to find a character of a value 2 Enter an integer value: 65 A Do you want to continue? Y/N
5.	Write a program to input a character and to print whether a given character is an alphabet, digit or any other character.

SOURCE CODE:	ch=input("Enter a character: ")
	if ch.isalpha():
	print(ch, "is an alphabet")
	elif ch.isdigit():
	print(ch, "is a digit")
	elif ch.isalnum():
	print(ch, "is alphabet and numeric")

	else: print(ch, "is a special symbol")
OUTPUT:	Enter a character: 7 7 is a digit Enter a character: P P is an alphabet
6.	Write a program to calculate the factorial of an integer using recursion.

SOURCE CODE:	def factorial(n):
	  if n == 1:
	  return n
	  else:
	  return n*factorial(n-1)
	
	num=int(input("enter the number: "))
	if num < 0:
	  print("Sorry, factorial does not exist for negative numbers")
	elif num == 0:
	  print("The factorial of 0 is 1")
	else:
	  print("The factorial of ",num," is ", factorial(num))

OUTPUT:	enter the number: 5 The factorial of 5 is 120
7.	Write a program to print fibonacci series using recursion.

SOURCE CODE:	def fibonacci(n):
	  if n<=1:
	  return n
	  else:
	  return(fibonacci(n-1)+fibonacci(n-2))
	
	num=int(input("How many terms you want to display: "))
	for i in range(num):
	  print(fibonacci(i)," ", end=" ")

OUTPUT:	How many terms you want to display: 8 0 1 1 2 3 5 8 13
8.	Write a program for binary search.

SOURCE CODE:	def Binary_Search(sequence, item, LB, UB):
	  if LB>UB:
	  return -5 # return any negative value
	  mid=int((LB+UB)/2)
	  if item==sequence[mid]:
	  return mid
	  elif item<sequence[mid]:
	  UB=mid-1
	  return Binary_Search(sequence, item, LB, UB)
	  else:

```
        LB=mid+1
        return Binary_Search(sequence, item, LB, UB)

L=eval(input("Enter the elements in sorted order: "))
n=len(L)
element=int(input("Enter the element that you want to search :"))
found=Binary_Search(L,element,0,n-1)
if found>=0:
    print(element, "Found at the index : ",found)
else:
    print("Element not present in the list")
```

OUTPUT:

```
Enter the elements in sorted order: 12,23,35,46,58,69,75,88,99
Enter the element that you want to search :69
69 Found at the index : 5
```

9. Write a recursive python program to test if a string is palindrome or not.

SOURCE CODE:

```
def isStringPalindrome(str):
    if len(str)<=1:
        return True
    else:
        if str[0]==str[-1]:
            return isStringPalindrome(str[1:-1])
        else:
            return False

#__main___

s=input("Enter the string : ")
y=isStringPalindrome(s)

if y==True:
    print("String is Palindrome")
else:
    print("String is Not Palindrome")
```

OUTPUT:

```
Enter the string : madam
String is Palindrome
```

10. Write a program to count the number of vowels present in a text file.

SOURCE CODE:

```
fin=open("D:\\python programs\\Book.txt",'r')
str=fin.read( )
count=0
for i in str:
    if i=='a' or i=='e' or i=='i' or i=='o' or i=='u':
        count=count+1

print(count)
```

OUTPUT:

```
9
```

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

SOURCE CODE:	fin=open("E:\\book.txt","r")
	fout=open("E:\\story.txt","a")
	s=fin.readlines()
	for j in s:
	  if 'p' in j:
	  fout.write(j)
	
	fin.close()
	fout.close()

OUTPUT:	**Write contents of book.txt and story.txt
12.	Write a program to count number of words in a file.
SOURCE CODE:	fin=open("D:\\python programs\\Book.txt",'r') str=fin.read( ) L=str.split() count_words=0 for i in L: count_words=count_words+1 print(count_words)
OUTPUT:	16
13.	Write a python function sin(x,n) to calculate the value of sin(x) using its taylor series expansion up to n terms.

SOURCE CODE:	import math
	def fact(k):
	  if k<=1:
	  return 1
	  else:
	  return k*fact(k-1)
	
	step=int(input("How many terms : "))
	x=int(input("Enter the value of x :"))
	sum=0
	for i in range(step+1):
	  sum+=(math.pow(-1,i)*math.pow(x,2*i+1))/fact(2*i+1)
	
	print("The result of sin",'(', x, ')', "is :", sum)

OUTPUT:	How many terms : 5 Enter the value of x :2 The result of sin (2) is : 0.9092961359628027

14.	Write a program to generate random numbers between 1 to 6 and check whether a user won a lottery or not.

SOURCE CODE:	import random
	n=random.randint(1,6)
	guess=int(input("Enter a number between 1 to 6 :"))
	if n==guess:
	  print("Congratulations, You won the lottery ")
	else:
	  print("Sorry, Try again, The lucky number was : ", n)

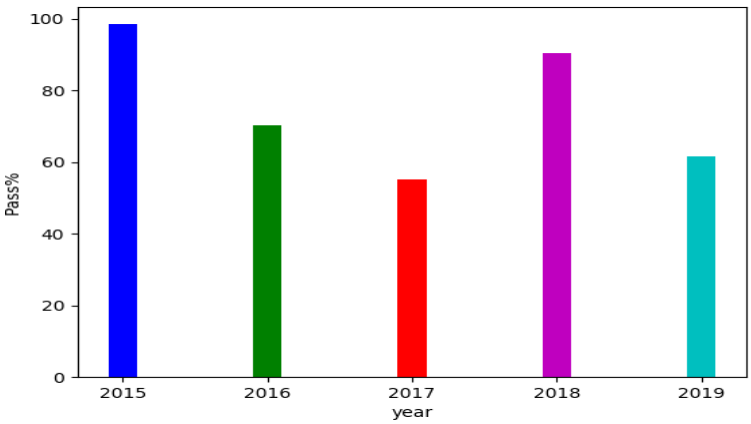
OUTPUT:	Enter a number between 1 to 6 : 4 Sorry, Try again, The lucky number was : 1
15.	Write a program to create a library in python and import it in a program.

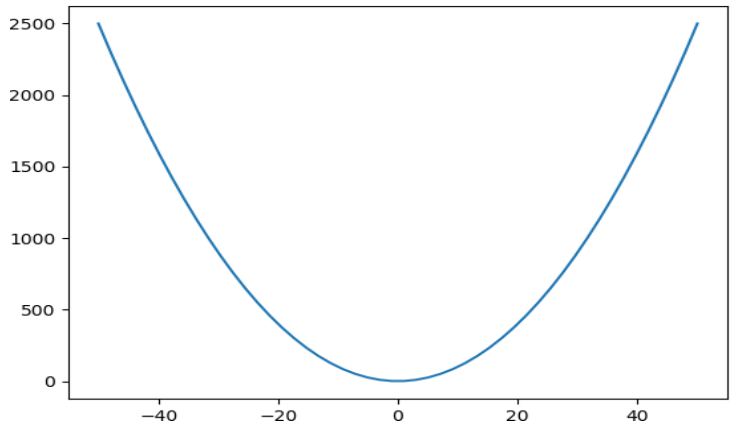
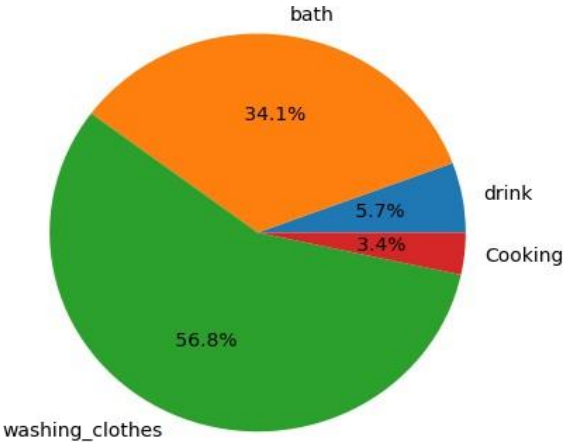
SOURCE CODE:	#Rect.py
	class Rectangle:
	  def__init__(self):
	  print("Rectangle")
	
	  def Area(self, length, width):
	  self.l=length
	  self.w=width
	  print("Area of Rectangle is : ", self.l*self.w)

	#Sq.py
	class Square:
	  def__init__(self):
	  print("Square")
	  def Area(self, side):
	  self.a=side
	  print("Area of square is : ", self.a*self.a)
	
	#Tri.py
	class Triangle:
	  def__init__(self):
	  print("Trinagle")
	
	  def Area(self, base, height):
	  self.b=base
	  self.h=height
	  ar= (1/2)*self.b*self.h
	  print("Area of Triangle is : ", ar)
	
	#main.py

	from Shape import Rect from Shape import Sq from Shape import Tri r=Rect.Rectangle( ) #Create an object r for Rectangle class r.Area(10,20) # Call the module Area( ) of Rectangle class by passing argument s=Sq.Square( ) #Create an object s for Square class s.Area(10) # Call the module Area( ) of Square class by passing argument t=Tri.Triangle( ) #Create an object t for Triangle class t.Area(6,8) # Call the module Area( ) of Triangle class by passing argument
OUTPUT:	Rectangle Area of Rectangle is : 200 Square Area of square is : 100 Trinagle Area of Triangle is : 24.0
16.	Write a program to plot a bar chart in python to display the result of a school for five consecutive years.

SOURCE CODE:	import matplotlib.pyplot as pl
	year=['2015','2016','2017','2018','2019'] # list of years
	p=[98.50,70.25,55.20,90.5,61.50] #list of pass percentage
	j=['b','g','r','m','c'] # color code of bar charts
	pl.bar(year, p, width=0.2, color=j) # bar( ) function to create the bar chart
	pl.xlabel("year") # label for x-axis
	pl.ylabel("Pass%") # label for y-axis
	pl.show( ) # function to display bar chart

OUTPUT:	
17.	Write a program in python to plot a graph for the function $y = x^2$

SOURCE CODE:	import matplotlib.pyplot as pl import numpy as np x= np.linspace(-50,50); y= x**2 pl.plot(x,y,linestyle='-') pl.show( )
OUTPUT:	
18.	Write a program in python to plot a pie chart on consumption of water in daily life.
SOURCE CODE:	import matplotlib.pyplot as pl consumption=[5,30,50,3]
	pl.pie(consumption, labels=['drink','bath','washing_clothes','Cooking'], autopct= ' %1.1f%% ') pl.show( )
OUTPUT:	
19.	Write a program for linear search.
SOURCE CODE:	L=eval(input("Enter the elements: ")) n=len(L) item=eval(input("Enter the element that you want to search : ")) for i in range(n):

	if L[i]==item:
	print("Element found at the position :", i+1)
	break
	else:
	print("Element not Found")

OUTPUT:	Enter the elements: 23,67,44,99,65,33,78,12 Enter the element that you want to search : 33 Element found at the position : 6
20.	Write a program for bubble sort.

SOURCE CODE:	L=eval(input("Enter the elements:"))
	n=len(L)
	for p in range(0,n-1):
	for i in range(0,n-1):
	if L[i]>L[i+1]:
	t=L[i]
	L[i]=L[i+1]
	L[i+1]=t
	print("The sorted list is : ", L)

OUTPUT:	Enter the elements:[67,13,89,34,65,8,74,19] The sorted list is : [8, 13, 19, 34, 65, 67, 74, 89]
21.	Write a menu based program to perform the operation on stack in python.

SOURCE CODE:	class Stack:
	def__init__(self):
	self.items = [ ]
	
	def isEmpty(self): # Checks whether the stack is empty or not
	return self.items == [ ]
	
	def push(self, item): #Insert an element
	self.items.append(item)
	
	def pop(self): # Delete an element
	return self.items.pop( )
	
	def peek(self): #Check the value of top
	return self.items[len(self.items)-1]
	
	def size(self): # Size of the stack i.e. total no. of elements in stack
	return len(self.items)
	
	s = Stack( )
	print("MENU BASED STACK")
	cd=True

```
while cd:
    print(" 1. Push ")
    print(" 2. Pop ")
    print(" 3. Display ")
    print(" 4. Size of Stack ")
    print(" 5. Value at Top ")

    choice=int(input("Enter your choice (1-5) : "))

    if choice==1:
        val=input("Enter the element: ")
        s.push(val)
    elif choice==2:
        if s.items==[ ]:
            print("Stack is empty")
        else:
            print("Deleted element is :", s.pop( ))
    elif choice==3:
        print(s.items)
    elif choice==4:
        print("Size of the stack is :", s.size( ))
    elif choice==5:
        print("Value of top element is :", s.peek( ))
    else:
        print("You enetered wrong choice ")

    print("Do you want to continue? Y/N")
    option=input( )
    if option=='y' or option=='Y':
        var=True
    else:
        var=False
```

OUTPUT:

```
MENU BASED STACK
 1. Push
 2. Pop
 3. Display
 4. Size of Stack
 5. Value at Top
Enter your choice (1-5) : 1
Enter the element: 45
Do you want to continue? Y/N
y
 1. Push
 2. Pop
 3. Display
```

	4. Size of Stack 5. Value at Top Enter your choice (1-5) : 3 ['45'] Do you want to continue? Y/N y 1. Push 2. Pop 3. Display 4. Size of Stack 5. Value at Top
22.	Write a menu based program to perform the operation on queue in python.

SOURCE CODE:	class Queue:
	def__init__(Q):
	Q.items = [ ]
	
	def isEmpty(Q): # Checks whether the queue is empty or not
	return Q.items == [ ]
	
	def Enqueue(Q, item): #Insert an element
	Q.items.append(item)
	if len(Q.items)==1:
	front=rear=0

	else:
	rear=len(Q.items)
	
	def Dequeue(Q): # Delete an element
	return Q.items.pop(0)
	
	def peek(Q): #Check the value of rear
	return Q.items[len(Q.items)-1]
	
	def size(Q): # Size of the queue i.e. total no. of elements in queue
	return len(Q.items)
	
	q = Queue( )
	print("MENU BASED QUEUE")
	cd=True
	while cd:
	print(" 1. ENQUEUE ")
	print(" 2. DEQUEUE ")
	print(" 3. Display ")
	print(" 4. Size of Queue ")
	print(" 5. Value at rear ")

```
    choice=int(input("Enter your choice (1-5) :  "))

    if choice==1:
        val=input("Enter the element: ")
        q.Enqueue(val)
    elif choice==2:
        if q.items==[ ]:
            print("Queue is empty")
        else:
            print("Deleted element is :", q.Dequeue( ))
    elif choice==3:
        print(q.items)
    elif choice==4:
        print("Size of the queue is :", q.size( ))
    elif choice==5:
        print("Value of rear element is :", q.peek( ))
    else:
        print("You enetered wrong choice ")

    print("Do you want to continue? Y/N")
    option=input( )
    if option=='y' or option=='Y':
        cd=True
    else:
        cd=False
```

OUTPUT:

```
MENU BASED QUEUE
 1.  ENQUEUE
 2.  DEQUEUE
 3.  Display
 4.  Size of Queue
 5.  Value at rear
Enter your choice (1-5) : 1
Enter the element: 10
Do you want to continue? Y/N
y
 1.  ENQUEUE
 2.  DEQUEUE
 3.  Display
 4.  Size of Queue
 5.  Value at rear
Enter your choice (1-5) : 1
Enter the element: 45
Do you want to continue? Y/N
y
 1.  ENQUEUE
```

	2. DEQUEUE 3. Display 4. Size of Queue 5. Value at rear Enter your choice (1-5) : 3 ['10', '45'] Do you want to continue? Y/N y 1. ENQUEUE 2. DEQUEUE 3. Display 4. Size of Queue 5. Value at rear Enter your choice (1-5) : 2 Deleted element is : 10 Do you want to continue? Y/N
23.	Write a menu based program for circular queue.

SOURCE CODE:	class CircularQueue:
	
	def__init__(CQ): #Constructor
	CQ.queue = [None]*7 # Create a list with None values with the size 7
	CQ.front = 0
	CQ.rear = 0

	CQ.maxSize = 7 def C_enqueue(CQ,data): #Adding elements to the queue CQ.queue[CQ.rear]=data CQ.rear = (CQ.rear + 1) % CQ.maxSize def C_dequeue(CQ): #Removing elements from the queue CQ.queue.pop(CQ.front) CQ.front = (CQ.front + 1) % CQ.maxSize q = CircularQueue() print("MENU BASED CIRCULAR QUEUE") cd=True while cd: print("1. ENQUEUE")

```
    print("2. DEQUEUE")
    print("3. DISPLAY ")
    print("4. Front Position ")
    print("5. Rear Position ")

    choice=int(input("Enter your choice (1-5) : "))

    if choice==1:
        val=input("Enter the element: ")
        q.C_enqueue(val)

    elif choice==2:
        q.C_dequeue()

    elif choice==3:
        print(q.queue)

    elif choice==4:
        print("Front element position :", q.front)

    elif choice==5:
        print("Rear element position : ", q.rear)
```

```
    else:
        print("You entered invalid choice: ")


    print("Do you want to continue? Y/N")
    option=input( )
    if option=='y' or option=='Y':
        cd=True
    else:
        cd=False
```

OUTPUT:

```
MENU BASED CIRCULAR QUEUE
1. ENQUEUE
2. DEQUEUE
3. DISPLAY
4. Front Position
5. Rear Position
Enter your choice (1-5) : 1
Enter the element: 56
Do you want to continue? Y/N
y
1. ENQUEUE
2. DEQUEUE
```

	3. DISPLAY 4. Front Position 5. Rear Position Enter your choice (1-5) : 1 Enter the element: 87 Do you want to continue? Y/N y 1. ENQUEUE 2. DEQUEUE 3. DISPLAY 4. Front Position 5. Rear Position Enter your choice (1-5) : 3 ['56', '87', None, None, None, None, None] Do you want to continue? Y/N
24.	Create a graphical calculator using tkinter library.

SOURCE CODE:	from tkinter import *
	def btnClick(number):
	global operator
	operator=operator+str(number)
	strvar.set(operator)
	def btnClear():
	global operator

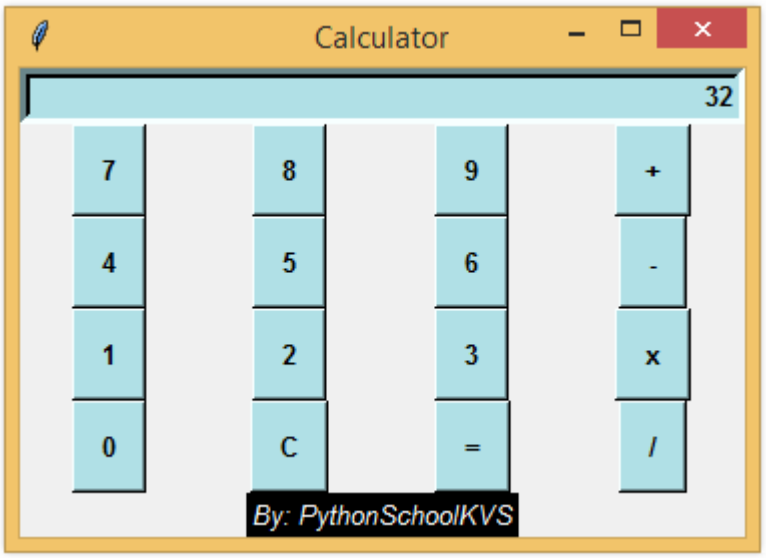
	operator='' strvar.set(operator) def result(): global operator res=str(eval(operator)) strvar.set(res) root=Tk() root.title("Calculator") operator='' strvar=StringVar() ent=Entry(root,width=50,bd=5,font=('arial',10,"bold"),bg="powder blue",textvariable=strvar,justify="right").grid(columnspan=4) btn7=Button(root,text="7",padx=10,pady=10,font=('arial',10,"bold"),bg="powder blue",command=lambda:btnClick(7)).grid(row=1,column=0)

```
btn8=Button(root,text="8",padx=10,pady=10,font=('arial',10,"bold"),bg="powd
er blue",command=lambda:btnClick(8)).grid(row=1,column=1)

btn9=Button(root,text="9",padx=10,pady=10,font=('arial',10,"bold"),bg="powd
er blue",command=lambda:btnClick(9)).grid(row=1,column=2)

btnPlus=Button(root,text="+",padx=10,pady=10,font=('arial',10,"bold"),bg="po
wder blue",command=lambda:btnClick('+')).grid(row=1,column=3)

btn4=Button(root,text="4",padx=10,pady=10,font=('arial',10,"bold"),bg="powd
er blue",command=lambda:btnClick(4)).grid(row=2,column=0)
btn5=Button(root,text="5",padx=10,pady=10,font=('arial',10,"bold"),bg="powd
er blue",command=lambda:btnClick(5)).grid(row=2,column=1)
btn6=Button(root,text="6",padx=10,pady=10,font=('arial',10,"bold"),bg="powd
er blue",command=lambda:btnClick(6)).grid(row=2,column=2)
btnMinus=Button(root,text="-
",padx=10,pady=10,font=('arial',10,"bold"),bg="powder
blue",command=lambda:btnClick('-')).grid(row=2,column=3)
btn1=Button(root,text="1",padx=10,pady=10,font=('arial',10,"bold"),bg="powd
er blue",command=lambda:btnClick(1)).grid(row=3,column=0)
btn2=Button(root,text="2",padx=10,pady=10,font=('arial',10,"bold"),bg="powd
er blue",command=lambda:btnClick(2)).grid(row=3,column=1)
btn3=Button(root,text="3",padx=10,pady=10,font=('arial',10,"bold"),bg="powd
er blue",command=lambda:btnClick(3)).grid(row=3,column=2)
btnMulti=Button(root,text="x",padx=10,pady=10,font=('arial',10,"bold"),bg="p
owder blue",command=lambda:btnClick('*')).grid(row=3,column=3)
btn0=Button(root,text="0",padx=10,pady=10,font=('arial',10,"bold"),bg="powd
er blue",command=lambda:btnClick(0)).grid(row=4,column=0)
btnClear=Button(root,text="C",padx=10,pady=10,font=('arial',10,"bold"),bg="p
owder blue",command=btnClear).grid(row=4,column=1)
btnEqual=Button(root,text="=",command=result,padx=10,pady=10,font=('arial',
10,"bold"),bg="powder blue").grid(row=4,column=2)
btnDivide=Button(root,text="/",padx=10,pady=10,font=('arial',10,"bold"),bg="p
owder blue",command=lambda:btnClick('/')).grid(row=4,column=3)
Label(root,text="By:
PythonSchoolKVS",font=('arial',10,'italic'),fg='white',bg='black').grid(row=5,co
lumnspan=4)
root.mainloop( )
```

OUTPUT:	
25.	Write a program to open a webpage using urllib library.
SOURCE CODE:	import urllib.request data = urllib.request.urlopen('https://pythonschoolkvs.wordpress.com/') print(data.read())
OUTPUT:	squeezed text (364 lines).
26.	Write a program to calculate EMI for a loan using numpy.
SOURCE CODE:	import numpy as np
	interest_rate= float(input("Enter the interest rate : ")) monthly_rate = (interest_rate)/ (12*100) years= float(input("Enter the total years : ")) number_month = years * 12 loan_amount= float(input("Enter the loan amount : ")) emi = abs(np.pmt(monthly_rate, number_month, loan_amount)) print("Your EMI will be Rs. ", round(emi, 2))
OUTPUT:	Enter the interest rate : 7.5 Enter the total years : 15 Enter the loan amount : 200000 Your EMI will be Rs. 1854.02
27.	Write a program to find the most common words in a file.
SOURCE CODE:	import collections fin = open('E:\\email.txt','r') a= fin.read()

```
d={ }
L=a.lower().split()

for word in L:
    word = word.replace(".","")
    word = word.replace(",","")
    word = word.replace(":","")
    word = word.replace("\"","")
    word = word.replace("!","")
    word = word.replace("&","")
    word = word.replace("*","")



for k in L:
    key=k
    if key not in d:
        count=L.count(key)
        d[key]=count

n_print = int(input("How many most common words to print: "))

print("\nOK. The {} most common words are as follows\n".format(n_print))
```

```
word_counter = collections.Counter(d)

for word, count in word_counter.most_common(n_print):
    print(word, ": ", count)

fin.close()
```

OUTPUT:

```
How many most common words to print: 5

OK. The 5 most common words are as follows

the :  505
a :  297
is :  247
in :  231
to :  214
```

28. Write a program to perform read and write operation with .csv file.

SOURCE CODE:

```
import csv
def readcsv():
    with open('C:\\Users\\ViNi\\Downloads\\data.csv','rt')as f:
        data = csv.reader(f)       #reader function to generate a reader object
```

```
        for row in data:
            print(row)


def writecsv( ):
    with open('C:\\Users\\ViNi\\Downloads\\data.csv', mode='a', newline='') as
file:
        writer = csv.writer(file, delimiter=',', quotechar='"')

        #write new record in file
        writer.writerow(['4', 'Devansh', 'Arts', '404'])


print("Press-1 to Read Data and Press-2 to Write data: ")
a=int(input())
if a==1:
    readcsv()
elif a==2:
    writecsv()
else:
    print("Invalid value")
```

OUTPUT:

```
Press-1 to Read Data and Press-2 to Write data:
1
['Roll No.', 'Name of student', 'stream', 'Marks']
['1', 'Anil', 'Arts', '426']
['2', 'Sujata', 'Science', '412']
['3', 'Shivani', 'Commerce', '448']
['4', 'Devansh', 'Arts', '404']
```

29. Write a Django based web application and write the data to a csv file.

SOURCE CODE:

```
# settings.py
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'NATIONALS'
]


#models.py
from django.db import models
```

```
class PLAYER(models.Model):
    pid = models.CharField(max_length=10)
    pname = models.CharField(max_length=50)
    dob = models.CharField(max_length=20)
    gender = models.CharField(max_length=10)
    game = models.CharField(max_length=30)
    region = models.CharField(max_length=35)
    class Meta:
        db_table = 'player'


#forms.py
from django import forms
from NATIONALS.models import PLAYER
class PLAYERFORM (forms.ModelForm):
    class Meta:
        model = PLAYER
        fields = "__all__"


#style.css
table, th, td {
  border: 1px solid black;
}

div {
    font-weight:bold;
}


from django.shortcuts import render, redirect
from NATIONALS.forms import PLAYERFORM
from NATIONALS.models import PLAYER

# Create your views here.
def ply(request):
    if request.method == "POST":
        form = PLAYERFORM(request.POST)
        if form.is_valid( ):
            try:
                form.save( )
                return redirect('/show')
            except:
                pass
    else:
        form = PLAYERFORM( )
```

```
    return render(request,"index.html",{'form':form})

def show(request):
    players = PLAYER.objects.all( )
    return render(request,"show.html",{'players':players})

def edit(request, id):
    player = PLAYER.objects.get(id=id)
    return render(request, "edit.html", {'player': player})

def update(request, id):
    player = PLAYER.objects.get(id=id)
    form = PLAYERFORM(request.POST, instance= player)
    if form.is_valid():
        form.save()
        return redirect('/show')
    return render(request, "edit.html", {'player': player})

def delete (request, id):
    player = PLAYER.objects.get(id=id)
    player.delete()
    return redirect('/show')
```

```
#index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Index Page</title>
    {% load staticfiles %}
    <link rel="stylesheet" href="{% static '/style.css' %}" />
</head>
<body>
<center><strong><u><h1>PLAYER DATA</h1></u></strong></center>
<center>

<form method = "POST" class = "post-form" action="/ply">
        {% csrf_token %}
    <div class="container">
        <br>
        <div class = "form-group row">
            <label class="col-sm-1 col-form-label"></label>
            <div class="col-sm-4">
                <h3>Enter Details</h3>
            </div>
```

```
        </div>
        <div class = "form-group row" >
            <label class="col-sm-2 col-form-label">Player ID: </label> 
   
            {{ form.pid }}
        </div>
        <br>

        <div class = "form-group row" >
            <label class="col-sm-2 col-form-label">Player Name: </label>
            {{ form.pname }}
        </div>
        <br>

        <div class = "form-group row" >
            <label class="col-sm-2 col-form-label">Date of Birth: </label>
            {{ form.dob }}
        </div>
        <br>

        <div class = "form-group row">
            <label class="col-sm-2 col-form-label">Gender:
</label>        
            {{ form.gender }}
        </div>
        <br>

        <div class = "form-group row">
            <label class="col-sm-2 col-form-label">Game:
</label>         
            {{ form.game }}
        </div>
        <br>

        <div class = "form-group row">
            <label class="col-sm-2 col-form-label">Region:
</label>        
            {{ form.region }}
        </div>
        <br>

        <div class = "form-group row">
            <label class="col-sm-1 col-form-label"> </label>
            <div class="col-sm-4">
```

```
            <button type="submit" class="btn btn-primary">Submit</button>
          </div>
        </div>
    </div>

</form></center>
</body>
</html>



#show.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Show Player Details</title>
    {% load staticfiles %}
    <link rel = "stylesheet " href="{% static '/style.css' %}" />
</head>
<body>
<center><strong><u><h1>PLAYER DETAILS</h1></u></strong></center>
<center>
<table class="table table-striped table-bordered table=80%">
    <thead class= "thead-dark">
    <tr>
        <th>Player ID</th>
        <th>Player Name</th>
        <th>Date of Birth</th>
        <th>Gender</th>
        <th>Game</th>
        <th>Region</th>
        <th colspan="2">Action</th>
    </tr>
    </thead>
    <tbody>
    {% for player in players %}
    <tr>
        <td>{{ player.pid }}</td>
        <td>{{ player.pname }}</td>
        <td>{{ player.dob }}</td>
        <td>{{ player.gender }}</td>
        <td>{{ player.game }}</td>
        <td>{{ player.region }}</td>
        <td>
            <a href = "/edit/{{player.id}}"><span class="glyphicon glyphicon-
pencil">Edit</span> </a>
```

```
        </td>
        <td>
            <a href="/delete/{{player.id}}">Delete</a>
        </td>
    </tr>
    {% endfor %}
    </tbody>
</table>
    </center>
<br>
<br>
<center><a href="/ply" class="btn btn-primary">Add New
Record</a></center>

</body>
</html>



#edit.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title> Edit Record </title>
    {% load staticfiles %}
    <link rel = "stylesheet" href = "{% static '/style.css' %}" />
</head>
<body>
<center>
<form method="POST" class="post-form" action="/update/{{player.id}}" >
    {% csrf_token %}
    <div class ="container">
        <br>
        <div class="form-group row">
            <label class="col-sm-1 col-form-label"></label>
            <div class = "col-sm-4">
                <h3>Update Details</h3>
            </div>
        </div>
        <br>

        <div class="form-group row">
            <label class="col-sm-2 col-form-label">Player ID :
</label>    
            <input type="text" name="pid" id="id_pid" required maxlength="10"
value="{{player.pid}}"/>
```

```
        </div>
        <br>

        <div class="form-group row">
           <label class="col-sm-2 col-form-label">Player Name : </label>
              <input type="text" name="pname" id="id_pname" required
maxlength="50" value="{{player.pname}}"/>
        </div>
        <br>

        <div class="form-group row">
           <label class="col-sm-2 col-form-label">DOB :
</label>         
              <input type="text" name="dob" id="id_dob" required
maxlength="20" value="{{player.dob}}"/>
        </div>
        <br>

        <div class="form-group row">
           <label class="col-sm-2 col-form-label">Gender :
</label>      
              <input type="text" name="gender" id="id_gender" required
maxlength="10" value="{{player.gender}}"/>
```

```
        </div>
        <br>

        <div class="form-group row">
           <label class="col-sm-2 col-form-label">Game :
</label>      
              <input type="text" name="game" id="id_game" required
maxlength="30" value="{{player.game}}"/>
        </div>
        <br>

        <div class="form-group row">
           <label class="col-sm-2 col-form-label">Region :
</label>      
              <input type="text" name="region" id="id_region" required
maxlength="35" value="{{player.region}}"/>
        </div>
        <br>

        <div class="form-group row">
           <label class="col-sm-1 col-form-label"></label>
           <div class="col-sm-4">
              <button type="submit" class="btn btn-success">Update</button>
```

```
            </div>
        </div>

    </div>
</form>
    </center>
</body>
</html>


#urls.py
from django.contrib import admin
from django.urls import path
from NATIONALS import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('ply', views.ply),
    path('show', views.show),
    path('edit/<int:id>', views.edit),
    path('update/<int:id>', views.update),
    path('delete/<int:id>', views.delete),
```

```
]
```

OUTPUT:

PLAYER DETAILS

Player ID	Player Name	Date of Birth	Gender	Game	Region	Action

Add New Record

PLAYER DATA

Enter Details

Player ID:

Player Name:

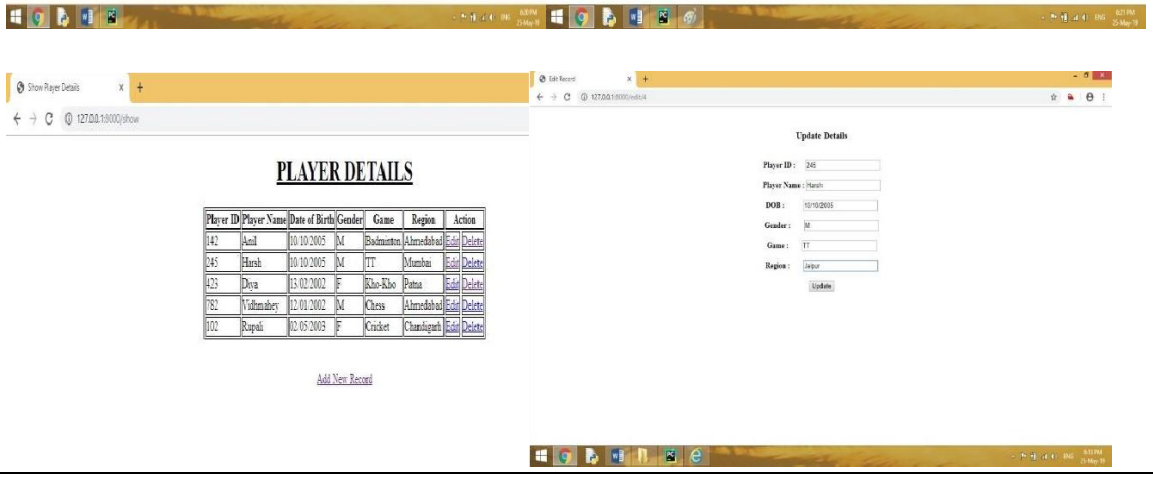
Date of Birth:

Gender:

Game:

Region:

Submit



PLAYER DETAILS

Player ID	Player Name	Date of Birth	Gender	Game	Region	Action
142	Amit	10/10/2005	M	Badminton	Ahmedabad	Edit Delete
245	Harsh	10/10/2005	M	TT	Mumbai	Edit Delete
423	Diya	13/02/2002	F	Kho-Kho	Patna	Edit Delete
782	Vidhnabey	12/01/2002	M	Chess	Ahmedabad	Edit Delete
102	Rupali	02/05/2003	F	Cricket	Chandigarh	Edit Delete

Add New Record

Update Details

Player ID : 245

Player Name : Harsh

DOB : 10/10/2005

Gender : M

Game : TT

Region : Jaipur

Update

30.	Queries using DISTINCT, BETWEEN, IN, LIKE, IS NULL, ORDER BY, GROUP BY, HAVING
A.	Display the name of departments. Each department should be displayed once.
SOLUTION	SELECT DISTINCT(Dept) FROM EMPLOYEE;
B.	Find the name and salary of those employees whose salary is between 35000 and 40000.
SOLUTION	SELECT Ename, salary FROM EMPLOYEE WHERE salary BETWEEN 35000 and 40000;
C.	Find the name of those employees who live in guwahati, surat or jaipur city.
SOLUTION	SELECT Ename, city FROM EMPLOYEE WHERE city IN('Guwahati','Surat','Jaipur');
D.	Display the name of those employees whose name starts with 'M'.
SOLUTION	SELECT Ename FROM EMPLOYEE WHERE Ename LIKE 'M%';
E.	List the name of employees not assigned to any department.
SOLUTION	SELECT Ename FROM EMPLOYEE WHERE Dept IS NULL;
F.	Display the list of employees in descending order of employee code.
SOLUTION	SELECT * FROM EMPLOYEE ORDER BY ecode DESC;
G.	Find the average salary at each department.
SOLUTION	SELECT Dept, avg(salary) FROM EMPLOYEE group by Dept;
H.	Find maximum salary of each department and display the name of that department which has maximum salary more than 39000.
31.	Queries for Aggregate functions- SUM( ), AVG( ), MIN( ), MAX( ), COUNT( )
	a. Find the average salary of the employees in employee table. Solution:- SELECT avg(salary)

	FROM EMPLOYEE; **b. Find the minimum salary of a female employee in EMPLOYEE table.** Solution:- SELECT Ename, min(salary) FROM EMPLOYEE WHERE sex='F'; **c. Find the maximum salary of a male employee in EMPLOYEE table.** Solution:- SELECT Ename, max(salary) FROM EMPLOYEE WHERE sex='M'; **d. Find the total salary of those employees who work in Guwahati city.** Solution:- SELECT sum(salary) FROM EMPLOYEE WHERE city='Guwahati'; **e. Find the number of tuples in the EMPLOYEE relation.** Solution:- SELECT count(*) FROM EMPLOYEE;
32.	Write a program to connect Python with MySQL using database connectivity and perform the following operations on data in database: Fetch, Update and delete the data.
A.	**CREATE A TABLE**
SOLUTION	import mysql.connector demodb = mysql.connector.connect(host="localhost", user="root", passwd="computer", database="EDUCATION") democursor=demodb.cursor( ) democursor.execute("CREATE TABLE STUDENT (admn_no int primary key, sname varchar(30), gender char(1), DOB date, stream varchar(15), marks float(4,2))")
B.	**INSERT THE DATA**
SOLUTION	import mysql.connector demodb = mysql.connector.connect(host="localhost", user="root", passwd="computer", database="EDUCATION") democursor=demodb.cursor( ) democursor.execute("insert into student values (%s, %s, %s, %s, %s, %s)", (1245, 'Arush', 'M', '2003-10-04', 'science', 67.34)) demodb.commit( )

C.	**FETCH THE DATA**
SOLUTION	import mysql.connector demodb = mysql.connector.connect(host="localhost", user="root", passwd="computer", database="EDUCATION") democursor=demodb.cursor( ) democursor.execute("select * from student") for i in democursor: print(i)
D.	**UPDATE THE RECORD**
SOLUTION	import mysql.connector demodb = mysql.connector.connect(host="localhost", user="root", passwd="computer", database="EDUCATION") democursor=demodb.cursor( ) democursor.execute("update student set marks=55.68 where admn_no=1356") demodb.commit( )
E.	**DELETE THE DATA**
SOLUTION	import mysql.connector demodb = mysql.connector.connect(host="localhost", user="root", passwd="computer", database="EDUCATION") democursor=demodb.cursor( ) democursor.execute("delete from student where admn_no=1356") demodb.commit( )