Name: C. Harrish
Roll No: 1801066

Ex No: 1	**Basic Python Programs**
Date:	

Aim:

To write basic programs using Python.

Question:

1. Write a Python program to calculate and display the interest on a loan amount (Rupees) using the formula:

interest = (principal * rate of interest * time) / 100
Test your code by using the given sample inputs.
Verify your code by using the **2nd** sample input(highlighted) given below:

Sample Input			**Expected Output**
Principal	Rate of Interest	Time	
20000	5	10	10000.0
7800	7.7	26	15615.6

Procedure:

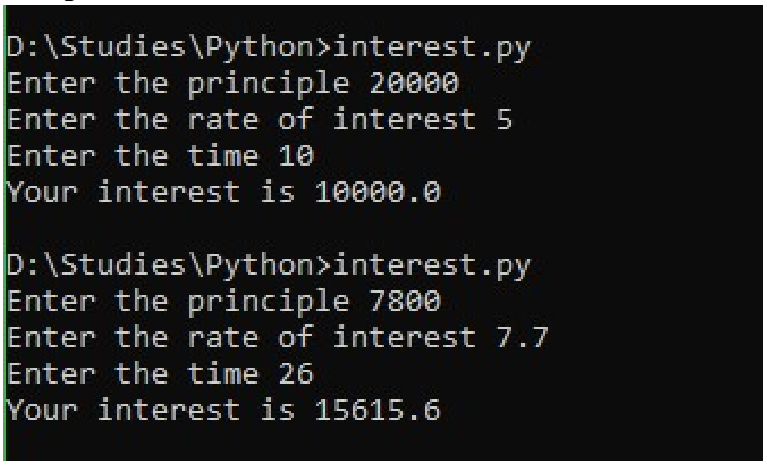
STEP 1: Get the input from the user for Principle, Rate of Interest and Time and
STEP 2: Calculate the interest by using formula ((principle * rate of interest * time)/100)
STEP 3: Print the output

Program:

```
principle = input("Enter the principle ")
rate_of_interest = input("Enter the rate of interest ")
time = input("Enter the time ")
principle = int(principle)
rate_of_interest = float(rate_of_interest)
time = int(time)
print("Your interest is", (principle*rate_of_interest*time)/100)
```

Name: C. Harrish
Roll No: 1801066

Output:



```
D:\Studies\Python>interest.py
Enter the principle 20000
Enter the rate of interest 5
Enter the time 10
Your interest is 10000.0

D:\Studies\Python>interest.py
Enter the principle 7800
Enter the rate of interest 7.7
Enter the time 26
Your interest is 15615.6
```

Name: C. Harrish
Roll No: 1801066

2. Write a python program to find out if a given year is a leap year or not. Any year which is divisible by 4 and not by 100 are leap years. Otherwise, any year which is divisible by 400 is also a leap year.

Procedure:

STEP 1: Get the year from the user

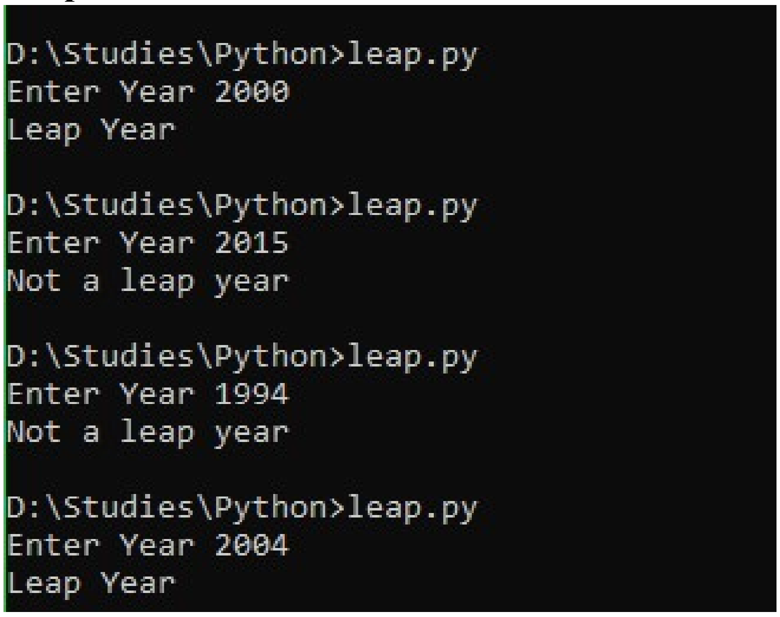
STEP 2: Check the year is leap year or not using if statement year is divisible by 4 and 400 and not divisible by 100 then it is Leap Year

STEP 3: Print the result

Program:

```
year = input("Enter Year ")
year = int(year)
if year % 4 == 0 and year % 100 != 0:
   print("Leap Year")
elif year % 400 == 0:
   print("Leap Year")
else:
   print("Not a leap year")
```

Output:



```
D:\Studies\Python>leap.py
Enter Year 2000
Leap Year

D:\Studies\Python>leap.py
Enter Year 2015
Not a leap year

D:\Studies\Python>leap.py
Enter Year 1994
Not a leap year

D:\Studies\Python>leap.py
Enter Year 2004
Leap Year
```

Name: C. Harrish
Roll No: 1801066

3. Write a python program to print the given pattern

```
*****
****
***
**
*
```

Procedure:

STEP 1: Set the range for 1 to 6

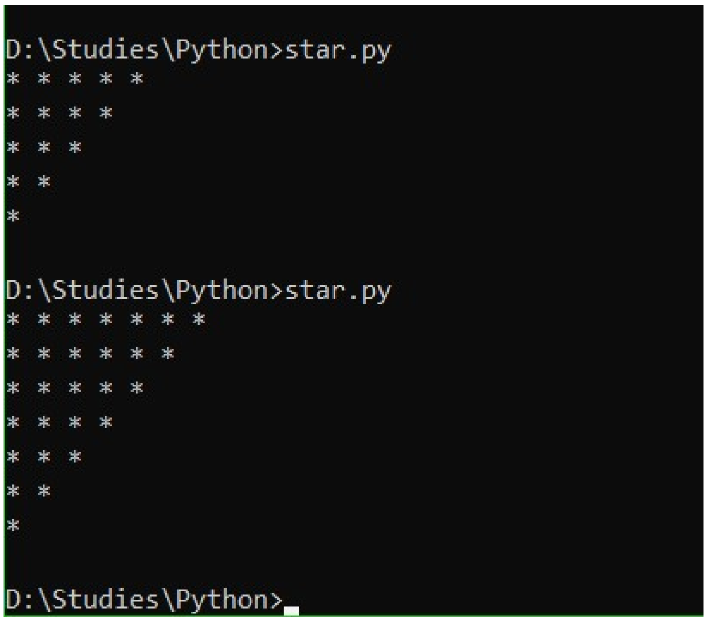
STEP 2: Using nested loop decrease the row and start

STEP 3: Finally print the pattern

Program:

```
for x in range(1,6):
    for y in range(x-1,5):
        print("*",end=' ')
    print()
```

Output:



```
D:\Studies\Python>star.py
* * * * *
* * * *
* * *
* *
*

D:\Studies\Python>star.py
* * * * * * *
* * * * * *
* * * * *
* * * *
* * *
* *
*

D:\Studies\Python>
```

Name: C. Harrish
Roll No: 1801066

4. The road transport corporation (RTC) of a city wants to know whether a particular bus-route is running on profit or loss.

Assume that the following information are given: Price
per litre of fuel = 70
Mileage of the bus in km/litre of fuel = 10
Price(Rs) per ticket = 80
The bus runs on multiple routes having different distance in kms and number of passengers.
Write a function to calculate and return the profit earned (Rs) in each route. Return -1 in case of loss.

Procedure:

STEP 1: Get the distance and number of passengers from the user

STEP 2: Calculate the total number of distance by dividing 10 and price per ticket multiplying with number of passengers

STEP 3: We know the fuel is 70 and calculate the profit using (price – total * fuel)

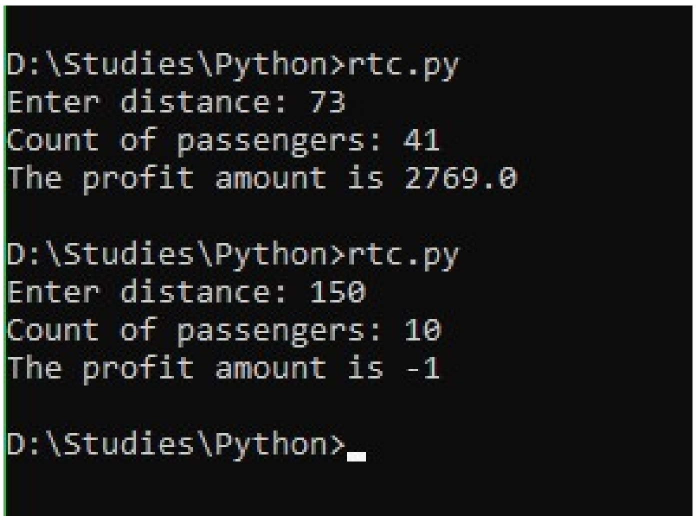
STEP 4: Print the profit amount if it is greater than 0 else print -1

Program:

```
def calculate(distance,passengers):
    total = float(distance/10)
    price = float(passengers*80)
    fuel = 70
    profit = float(price - total * fuel)
    if profit > 0:
        return profit
    return -1


distance = int(input("Enter distance: "))
passengers = int(input("Count of passengers: "))
print("The profit amount is", calculate(distance,passengers))
```

Name: C. Harrish
Roll No: 1801066

Output:



```
D:\Studies\Python>rtc.py
Enter distance: 73
Count of passengers: 41
The profit amount is 2769.0

D:\Studies\Python>rtc.py
Enter distance: 150
Count of passengers: 10
The profit amount is -1

D:\Studies\Python>_
```

Name: C. Harrish
Roll No: 1801066

5. Write a python program to display all the common characters between two strings.

Return -1 if there are no matching characters.

Note: Ignore blank spaces if there are any. Perform case sensitive string comparison wherever necessary.

Sample Input	Expected output
"I like Python" "Java is a very popular language"	lieyon

Procedure:

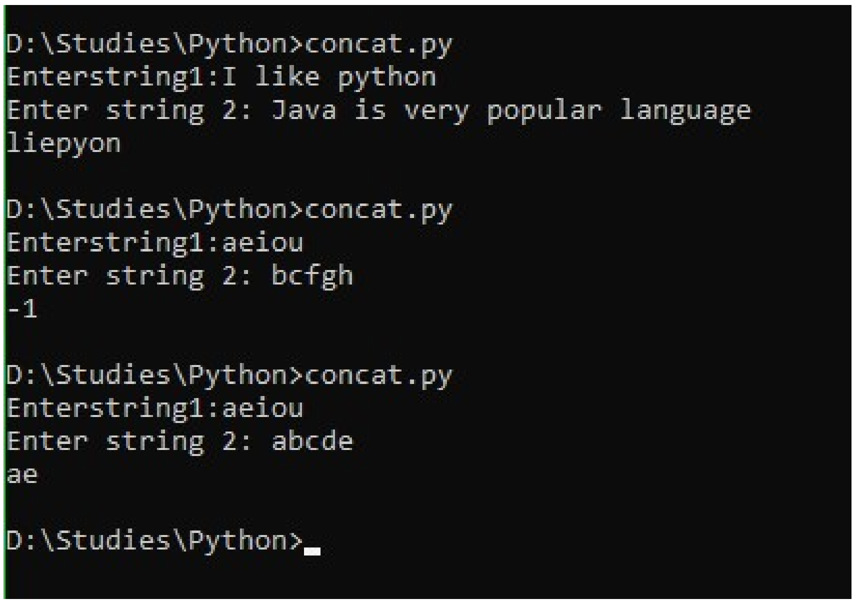
STEP 1: Get the string from the user and set the second string

STEP 2: Create a empty list called co

STEP 3: Compare the first string with second and add the common character to the list by using append()

STEP 4: Print the common character using join() or else print -1 if the list is empty

Output:



```
D:\Studies\Python>concat.py
Enterstring1:I like python
Enter string 2: Java is very popular language
liepyon

D:\Studies\Python>concat.py
Enterstring1:aeiou
Enter string 2: bcfgh
-1

D:\Studies\Python>concat.py
Enterstring1:aeiou
Enter string 2: abcde
ae

D:\Studies\Python>
```

Name: C. Harrish
Roll No: 1801066

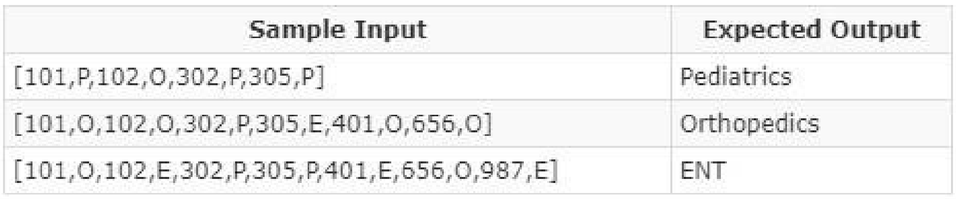
6. Care hospital wants to know the medical speciality visited by the maximum number of patients. Assume that the patient id of the patient along with the medical speciality visited by the patient is stored in a list. The details of the medical specialities are stored in a dictionary as follows:

```
{
        "P" : "Pediatrics",
        "O" : "Orthopedics",
        "E" : "ENT"
}
```

Write a function to find the medical speciality visited by the maximum number of patients and return the name of the speciality.

Note:

Assume that there is always only one medical speciality which is visited by maximum number of patients. Perform case sensitive string comparison wherever necessary.



Sample Input	Expected Output
[101,P,102,O,302,P,305,P]	Pediatrics
[101,O,102,O,302,P,305,E,401,O,656,O]	Orthopedics
[101,O,102,E,302,P,305,P,401,E,656,O,987,E]	ENT

Procedure:

STEP 1: Get the list from the user

STEP 2: Initially set the counts to 0 and create a dictionary

STEP 3: Get the list by running a for loop to count the alphabet enter by the user and increment

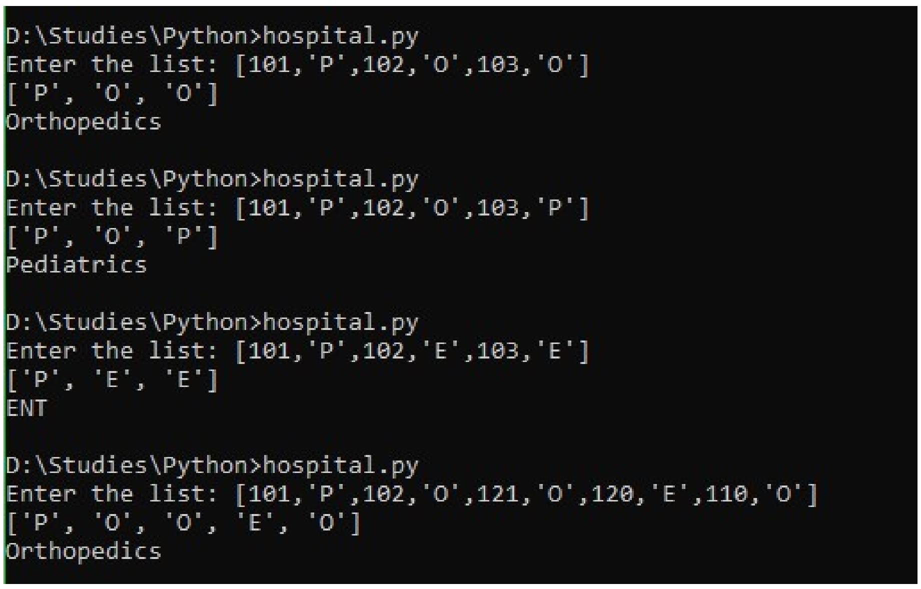
STEP 4: Use max() to get the maximum count value and store it in variable

STEP 5: Compare the max_value with count and print the corresponding Dictionary String

Program:

```
def max_finder(lis):
    list_with_the_alphabets=lis[1::2]
    print(list_with_the_alphabets)
    P_count,O_count,E_count=0,0,0
    dictionary={"P" : "Pediatrics","O" : "Orthopedics","E" : "ENT"}
    for i in list_with_the_alphabets:
        if i == "P":
            P_count += 1
        elif i == "O":
            O_count += 1
            elif i == "E":
                E_count += 1
        max_value = max(P_count,O_count,E_count)
        if max_value == P_count:
            return dictionary["P"]
        elif max_value == O_count:
            return dictionary["O"]
        elif max_value == E_count:
            return dictionary["E"]
    print(max_finder(eval(input("Enter the list: "))))
```

Output:



```
D:\Studies\Python>hospital.py
Enter the list: [101,'P',102,'O',103,'O']
['P', 'O', 'O']
Orthopedics

D:\Studies\Python>hospital.py
Enter the list: [101,'P',102,'O',103,'P']
['P', 'O', 'P']
Pediatrics

D:\Studies\Python>hospital.py
Enter the list: [101,'P',102,'E',103,'E']
['P', 'E', 'E']
ENT

D:\Studies\Python>hospital.py
Enter the list: [101,'P',102,'O',121,'O',120,'E',110,'O']
['P', 'O', 'O', 'E', 'O']
Orthopedics
```

Name: C. Harrish
Roll No: 1801066

7. A university wants to automate their admission process. Students are admitted based on marks scored in a qualifying exam.

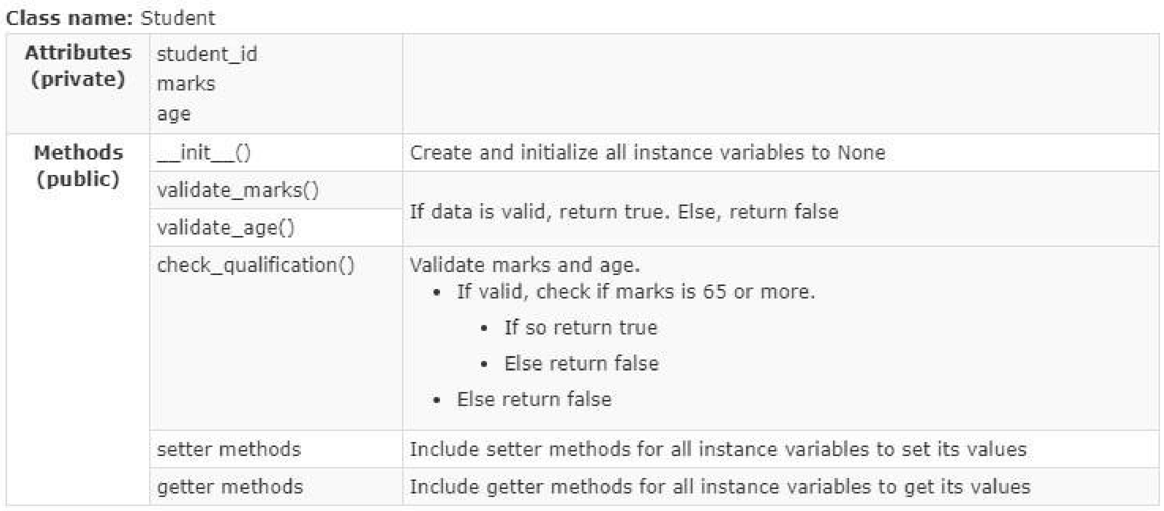
A student is identified by student id, age and marks in qualifying exam. Data are valid, if:

- Age is greater than 20
- Marks is between 0 and 100 (both inclusive) A student qualifies for admission, if
- Age and marks are valid and Marks is 65 or more

Write a python program to represent the students seeking admission in the university.

The details of student class are given below.

Class name: Student



Class name: Student

Attributes (private)	student_id marks age	
Methods (public)	__init__()	Create and initialize all instance variables to None
	validate_marks()	If data is valid, return true. Else, return false
	validate_age()	
	check_qualification()	Validate marks and age. • If valid, check if marks is 65 or more. • If so return true • Else return false • Else return false
	setter methods	Include setter methods for all instance variables to set its values
	getter methods	Include getter methods for all instance variables to get its values

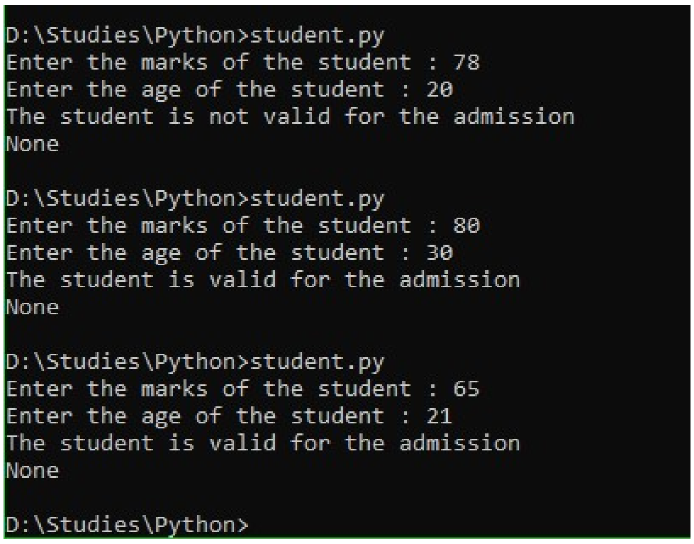
Procedure:

STEP 1: Create a class student and declare id
STEP 2: Create a function __init__ and initialize id,marks and age
STEP 3: Create a function validate_age() to check the entered age is valid or not
STEP 4: Repeat Step 3 for validate_marks()
STEP 5: In check_qualification() compare the marks and age if the condition is satisfied or not if satisfied print Valid, Else Not Valid
STEP 6: Use set and get method to the variables
STEP 7: Print the output

Program:

```
class stu():
  id=6
  def __init__(self,m,a):
    self.stu_id=stu.id
    stu.id+=1
    self.m=m
    self.a=a
    def valid_age(self):
        if self.a > 20:
            return self.a
        else:
            return -1
    def valid_marks(self):
        if self.m>=0 and self.m<=100:
            return self.m
        return -1
    def check_qualification(self):
        if self.valid_age() != -1 and self.valid_marks() !=-1 :
            if self.m>=65:
                print("The student is valid for the admission")
            else:
                print("The student is not valid for the admission")
        else:
            print("The student is not valid for the admission")
    def set_stu_m(self,m):
        self.m=m
    def set_stu_a(self,a):
        self.a=a
    def get_stu_id(self):
        return self.stu_id
    def get_stu_m(self):
        return self.m
    def get_stu_a(self):
        return self.a
m=int(input("Enter the marks of the student : "))
a=int(input("Enter the age of the student : "))
student=stu(m,a)
print(student.check_qualification())
```

Output:



```
D:\Studies\Python>student.py
Enter the marks of the student : 78
Enter the age of the student : 20
The student is not valid for the admission
None

D:\Studies\Python>student.py
Enter the marks of the student : 80
Enter the age of the student : 30
The student is valid for the admission
None

D:\Studies\Python>student.py
Enter the marks of the student : 65
Enter the age of the student : 21
The student is valid for the admission
None

D:\Studies\Python>
```

NOMENCLATURE	MAX MARKS	MARKS OBTAINED
Pre-Lab Viva	05	
Pre-Lab Preparation	06	
In Lab Performance	07	
Post-Lab Viva	02	
Total	**20**	
Signature of Faculty		

Result:

Thus, the basic python programs were executed and verified successfully

Name: C. Harrish
Roll No: 1801066

Ex No: 2 Date:	Regular Expression

Aim:

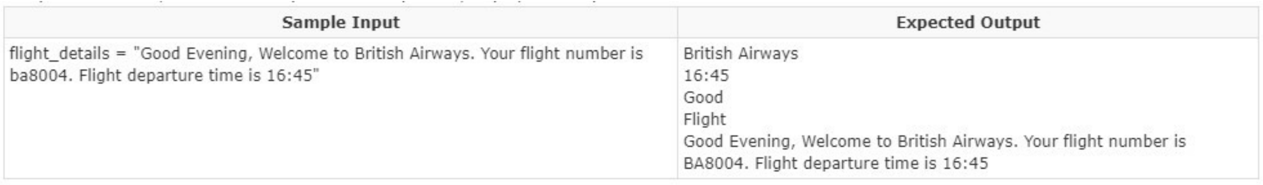
To write programs using Regular Expression with Python.

Question:

1. Write a python program to perform the following operations on the given string using the various regular expression functions. flight_details="Good Evening, Welcome to British Airways. Your flight number is ba8004. Flight departure time is 16:45"

1. Find whether the flight service name, British Airways is present in the given string and if so, display it.
2. Find whether the flight details ends with the departure time, 16:45 and if so, display it.
3. Find whether the flight details starts with "Good" and if so, display it.
4. Find a word which starts with 'F' and having 6 characters in it, if so display it.
5. Search for a word which starts with "ba" followed by exactly 4 digits. If found, replace the two alphabets with the corresponding uppercase alphabets.

For questions 1 to 4, if the searched pattern is not present, display "No Output":



Sample Input	Expected Output
flight_details = "Good Evening, Welcome to British Airways. Your flight number is ba8004. Flight departure time is 16:45"	British Airways 16:45 Good Flight Good Evening, Welcome to British Airways. Your flight number is BA8004. Flight departure time is 16:45

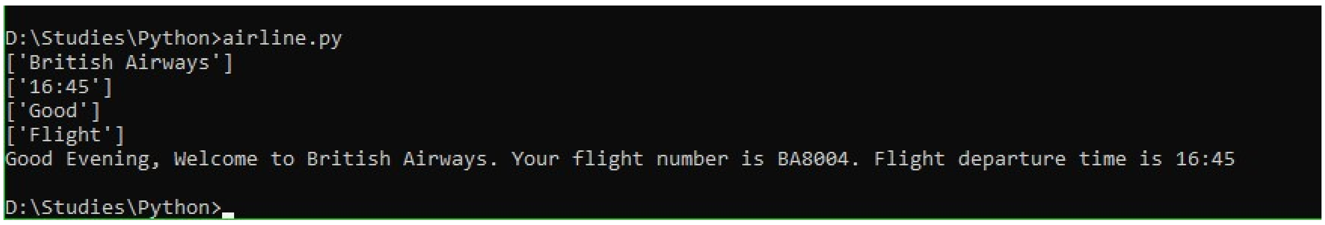
Procedure:

STEP 1: Import the re library to use Regular expression functions
STEP 2: Enter the string
STEP 3: Use findall() function to find the string present in the string
STEP 4: Use sub() to change the string
STEP 5: Print the output

Program:

```
import re
string ="Good Evening, Welcome to British Airways. Your flight number is ba8004.
Flight departure time is 16:45"
print(re.findall("British Airways", string))
print(re.findall("16.45$", string))
print(re.findall("^Good", string))
print(re.findall("F.....", string))
print(re.sub("ba","BA",string))
```

Output:



```
D:\Studies\Python>airline.py
['British Airways']
['16:45']
['Good']
['Flight']
Good Evening, Welcome to British Airways. Your flight number is BA8004. Flight departure time is 16:45

D:\Studies\Python>
```

Name: C. Harrish
Roll No: 1801066

2. Write a python code to do the following operations.

i) Find the string "X-DSPAM-Result:" in the mbox-short.txt file provided to you and replace it with "X-DSPAM-Outcome:". You should display the string after the replace operation as well as the number of replacements done.

ii) Find the average of all X-DSPAM-Confidence: in the file and print the result in float.

Procedure:

STEP 1: Open the file “mbox-short.txt” using open()

STEP 2: Read the file in for and and search for X-DSPAM-Result and increase the count

STEP 3: Use sub() to change the string to X-DSPAM-Outcome

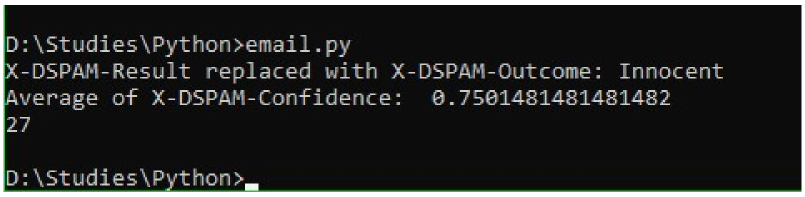
STEP 4: Calculate the Average of X-DSPAM-Confidence value

STEP 5: Print the output

Program:

```
import re
file = open("mbox-short.txt")
c = 0
a = 0
for l in file:
   l = l.rstrip()
   if(re.search("X-DSPAM-Result",l)):
      c = c + 1
      r = re.sub("X-DSPAM-Result","X-DSPAM-Outcome",l)
   if not l.startswith("X-DSPAM-Confidence:"): continue
   a += float(l[20:-1].strip())
print("X-DSPAM-Result replaced with",r)
print("Average of X-DSPAM-Confidence: ",a/c)
print(c)
```

Output:



```
D:\Studies\Python>email.py
X-DSPAM-Result replaced with X-DSPAM-Outcome: Innocent
Average of X-DSPAM-Confidence:  0.7501481481481482
27

D:\Studies\Python>_
```

Name: C. Harrish
Roll No: 1801066

3. Write a python program to validate the password and display whether the password is valid or not. The requirements for a valid password are

i. It should have both upper case and lower-case characters.
ii. It should have one special symbol [$#@!&].
iii. It should have at least one number.
iv. Minimum length for password should be 6 and maximum length should be 10

Procedure:

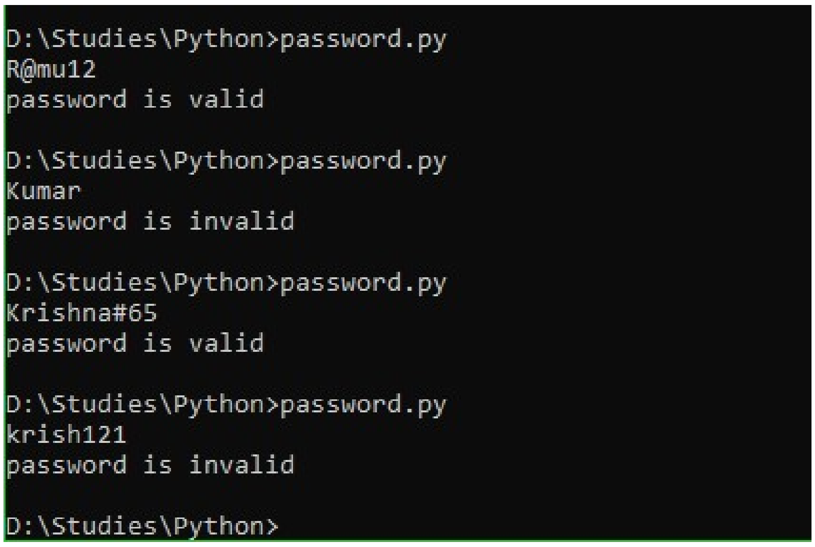
STEP 1: Import the regular expression module
STEP 2: Get the input from user
STEP 3: If the password not have both Uppercase and lower case return false
STEP 4: If the password does not have special symbol return false
STEP 5: If the password does not have atleast one number return false
STEP 6: If the password does not have 6 to 10 character return false

Program:

```
import re

p = input()
pattern = "^.*(?=.{6,10})(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#$%^&+=]).*$"
r = re.match(pattern,p)
if(r):
   print("password is valid");
else:
   print("password is invalid");
```

Name: C. Harrish
Roll No: 1801066

Output:



```
D:\Studies\Python>password.py
R@mu12
password is valid

D:\Studies\Python>password.py
Kumar
password is invalid

D:\Studies\Python>password.py
Krishna#65
password is valid

D:\Studies\Python>password.py
krish121
password is invalid

D:\Studies\Python>
```

NOMENCLATURE	**MAX MARKS**	**MARKS OBTAINED**
Pre-Lab Viva	05	
Pre-Lab Preparation	06	
In Lab Performance	07	
Post-Lab Viva	02	
Total	**20**	
Signature of Faculty		

Result:

Thus, the python programs using regular expressions were executed and verified successfully

Name: C. Harrish
Roll No: 1801066

Ex No: 3 **Date:**	**File Transfer using TCP / IP**

Aim:

To write program using Python to send file from server to client using TCP Protocol and display the contents of the file in client side.

Question:

1. Write a client/server-based python code to perform File transfer using TCP/IP. The contents of the file transferred should be displayed at the client side.

Procedure:

STEP 1: Import socket module and create a socket with gethostname() which is localhost and port 3000
STEP 2: Connect the host and port in the socket
STEP 3: listen() is for queued connection which default value is 5
STEP 4: Once the connection is established the server will send the file to client with buffer size 1024 bytes
STEP 5: In client side the file content is printed and connection is closed

Program:

Server.py

```
import socket
port = 3000
s = socket.socket()
host = socket.gethostname()
s.bind((host, port))
s.listen(5)
print('Server listening....')
while True:
    conn, addr = s.accept()
    print('Got connection from', addr)
    data = conn.recv(1024)
    print('Server received', repr(data))
    filename='sample.txt'
    f = open(filename,'rb')
    l = f.read(1024)
    while (l):
       conn.send(l)
       print('Sent ',repr(l))
       l = f.read(1024)
    f.close()
```

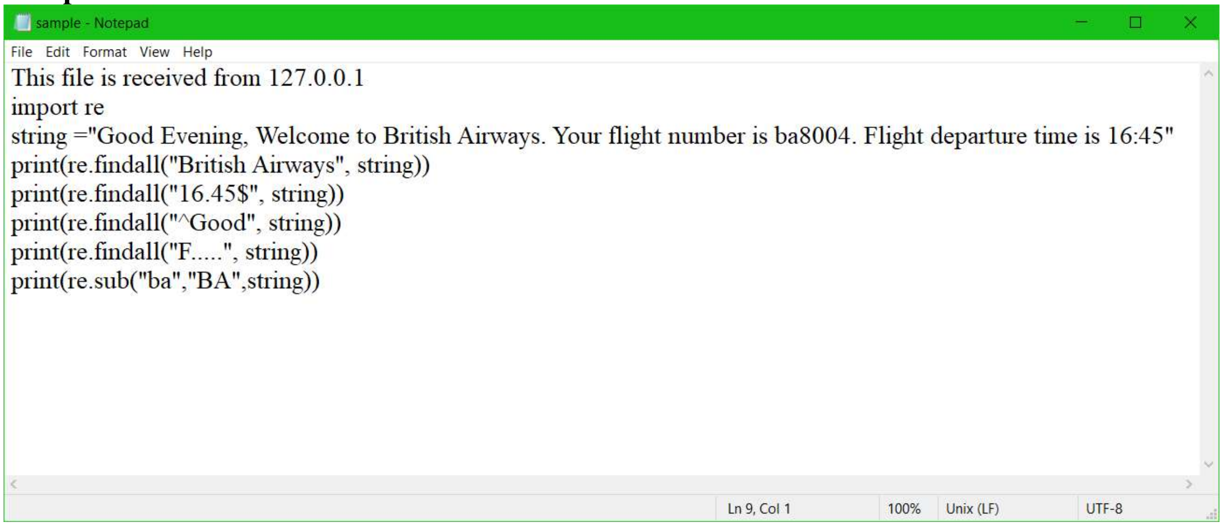
```
        print('Done sending')
        conn.send(bytes('Thank you for connecting','utf-8'))
        conn.close()
```

client.py

```
import socket
s = socket.socket()
host = socket.gethostname()
port = 3000
s.connect((host, port))
s.send(bytes("Hello server!",'utf-8'))
with open('received_file', 'wb') as f:
    print('file opened')
    while True:
        print('receiving data...')
        data = s.recv(1024)
        print('data=%s', (data))
        if not data:
            break
        f.write(data)
f.close()
print('Successfully get the file')
s.close()
print('connection closed')
```

Output:

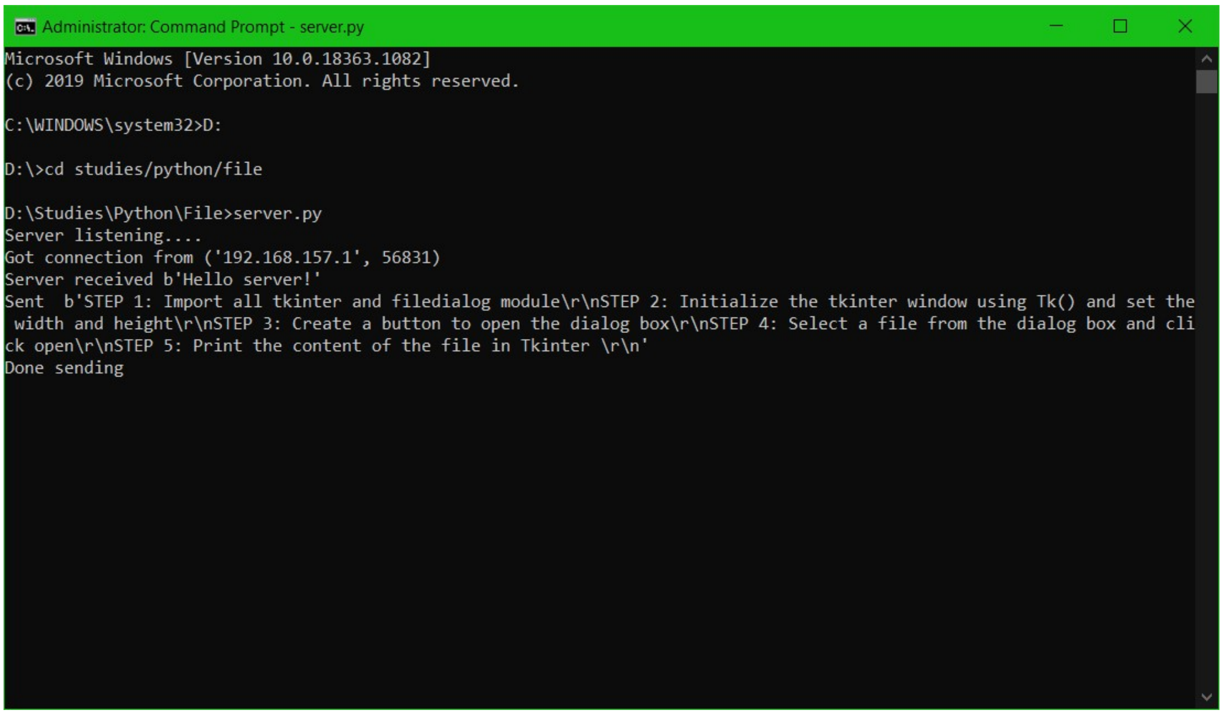
Sample.txt



```
This file is received from 127.0.0.1
import re
string ="Good Evening, Welcome to British Airways. Your flight number is ba8004. Flight departure time is 16:45"
print(re.findall("British Airways", string))
print(re.findall("16.45$", string))
print(re.findall("^Good", string))
print(re.findall("F.....", string))
print(re.sub("ba","BA",string))
```

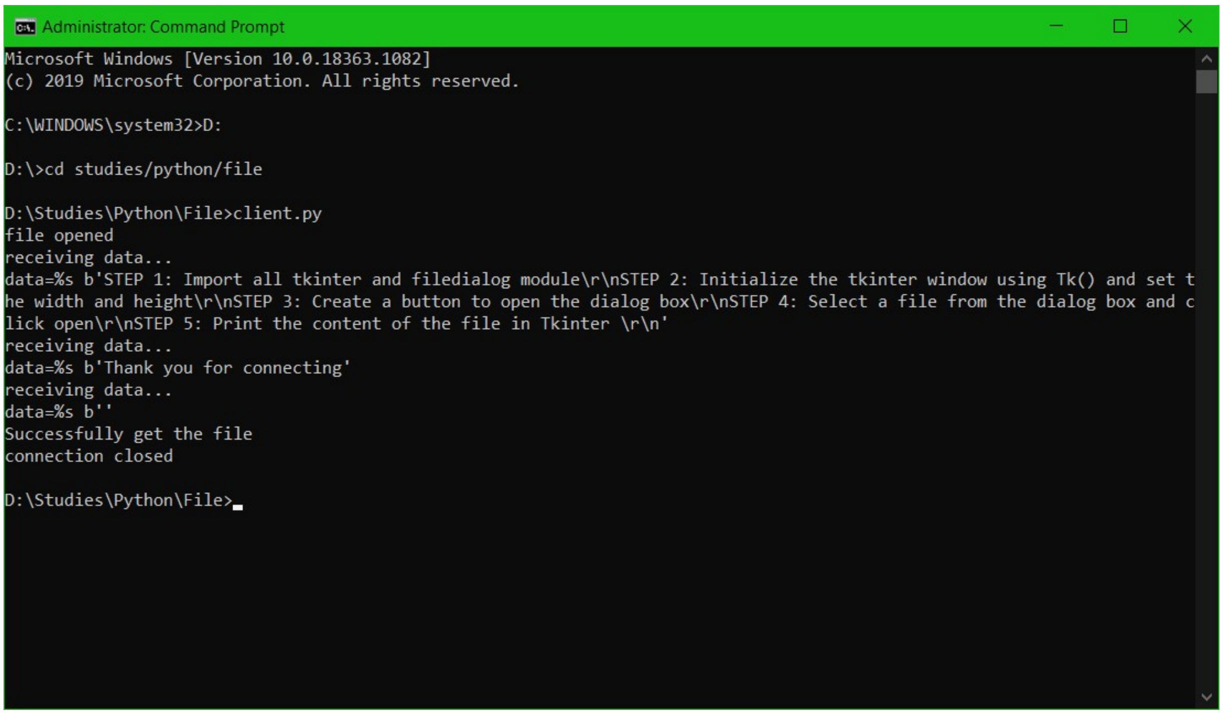
Name: C. Harrish
Roll No: 1801066

Server.py


Administrator: Command Prompt - server.py

```
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>D:

D:\>cd studies/python/file

D:\Studies\Python\File>server.py
Server listening....
Got connection from ('192.168.157.1', 56831)
Server received b'Hello server!'
Sent  b'STEP 1: Import all tkinter and filedialog module\r\nSTEP 2: Initialize the tkinter window using Tk() and set the
 width and height\r\nSTEP 3: Create a button to open the dialog box\r\nSTEP 4: Select a file from the dialog box and cli
ck open\r\nSTEP 5: Print the content of the file in Tkinter \r\n'
Done sending
```

Client.py


Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>D:

D:\>cd studies/python/file

D:\Studies\Python\File>client.py
file opened
receiving data...
data=%s b'STEP 1: Import all tkinter and filedialog module\r\nSTEP 2: Initialize the tkinter window using Tk() and set t
he width and height\r\nSTEP 3: Create a button to open the dialog box\r\nSTEP 4: Select a file from the dialog box and c
lick open\r\nSTEP 5: Print the content of the file in Tkinter \r\n'
receiving data...
data=%s b'Thank you for connecting'
receiving data...
data=%s b''
Successfully get the file
connection closed

D:\Studies\Python\File>
```

Name: C. Harrish
Roll No: 1801066

NOMENCLATURE	MAX MARKS	MARKS OBTAINED
Pre-Lab Viva	05	
Pre-Lab Preparation	06	
In Lab Performance	07	
Post-Lab Viva	02	
Total	**20**	
Signature of Faculty		

Result:

Thus, The Python program to send file from server to client using TCP protocol was executed and verified successfully

Name: C. Harrish
Roll No: 1801066

Ex No: 4 **Date:**	**Chat Application using UDP**

Aim:

To write program using Python to create a chat application between client and server using UDP Protocol.

Question:

Write a client/server-based python code to perform Chatting using UDP.

Procedure:

STEP 1: Import socket module and create Host,port and buffer size
STEP 2: Declare AF_INET and SOCK_DGRAM is socket()
STEP 3: Bind the host and port
STEP 4: Open the connection and wait for the client to connect
STEP 5: The data received from Client will be stored in buffer and decoded to UTF-8
STEP 6: Repeat Step 5 for the Client side
STEP 7: Print the output

Program:

Server.py

```
from socket import *

HOST = '127.0.0.1'
PORT = 3000
BUFFER_SIZE = 1024

sock = socket(AF_INET,SOCK_DGRAM)
sock.bind((HOST, PORT))

while True:
    print('\nWaiting for Message...\n')
    data, address = sock.recvfrom(BUFFER_SIZE)
    data = data.decode("utf-8")

    print(f'Message from {address} :')
    print(f'{data}\n')

    print('Enter Message :')
    data = input('\t> ')
    sock.sendto(data.encode(), address)
```

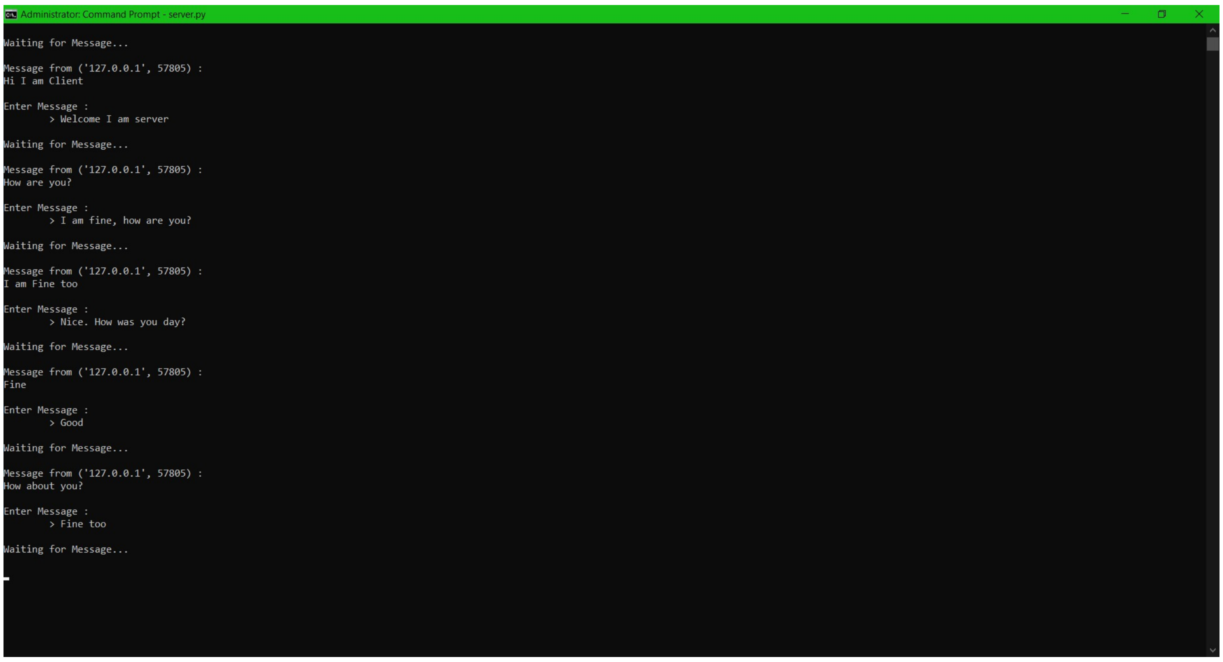
Name: C. Harrish
Roll No: 1801066

client.py

```
from socket import *

HOST = '127.0.0.1'
PORT = 3000
BUFFER_SIZE = 1024

sock = socket(AF_INET, SOCK_DGRAM)

while True:
    print('\n Enter Message :')
    data = input('\t> ')
    sock.sendto(data.encode(), (HOST, PORT))

    print('\nWaiting for Message...\n')
    data, address = sock.recvfrom(BUFFER_SIZE)
    data = data.decode("utf-8")

    print(f'Message from {address} :')
    print(f'{data}\n')
```

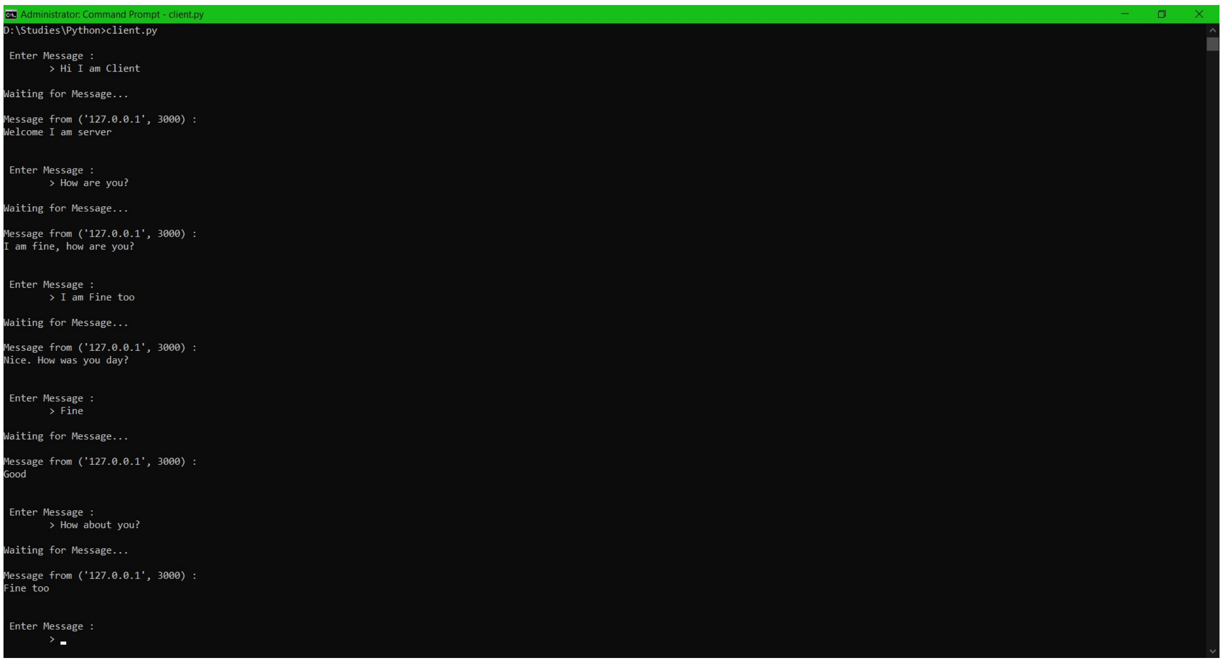
Output:

Server



```
Waiting for Message...

Message from ('127.0.0.1', 57805) :
Hi I am Client

Enter Message :
        > Welcome I am server

Waiting for Message...

Message from ('127.0.0.1', 57805) :
How are you?

Enter Message :
        > I am fine, how are you?

Waiting for Message...

Message from ('127.0.0.1', 57805) :
I am Fine too

Enter Message :
        > Nice. How was you day?

Waiting for Message...

Message from ('127.0.0.1', 57805) :
Fine

Enter Message :
        > Good

Waiting for Message...

Message from ('127.0.0.1', 57805) :
How about you?

Enter Message :
        > Fine too

Waiting for Message...
```

Client



```
D:\Studies\Python>client.py

 Enter Message :
        > Hi I am Client

Waiting for Message...

Message from ('127.0.0.1', 3000) :
Welcome I am server


 Enter Message :
        > How are you?

Waiting for Message...

Message from ('127.0.0.1', 3000) :
I am fine, how are you?


 Enter Message :
        > I am Fine too

Waiting for Message...

Message from ('127.0.0.1', 3000) :
Nice. How was you day?


 Enter Message :
        > Fine

Waiting for Message...

Message from ('127.0.0.1', 3000) :
Good


 Enter Message :
        > How about you?

Waiting for Message...

Message from ('127.0.0.1', 3000) :
Fine too


 Enter Message :
        >
```

NOMENCLATURE	MAX MARKS	MARKS OBTAINED
Pre-Lab Viva	05	
Pre-Lab Preparation	06	
In Lab Performance	07	
Post-Lab Viva	02	
Total	**20**	
Signature of Faculty		

Result:

Thus, the python to create a chat application between client and server using UDP Protocol was executed and verified successfully.

Name: C. Harrish
Roll No: 1801066

Ex No: 5 **Date:**	**GUI using Tkinter**

Aim:

To write program using Python to create GUI applications using Tkinter.

Question:

1. Develop a simple calculator application using tkinter and show output for all basic operations. Your application should contain three text fields:
1. To display the first number that is clicked 2.
To display the second number that is clicked
3. To display the result.

Procedure:

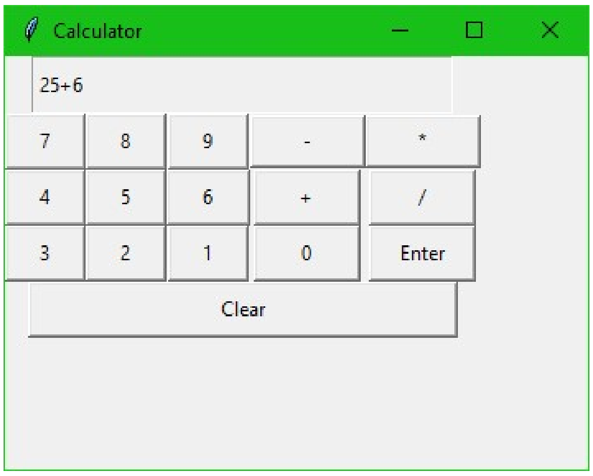
STEP 1: Import the all tkinter module
STEP 2: Create a tkinter window using Tk() and size of the window
STEP 3: Get the input from the user
STEP 4: Based on the input from the user perform the operation
STEP 5: Print the result in the label

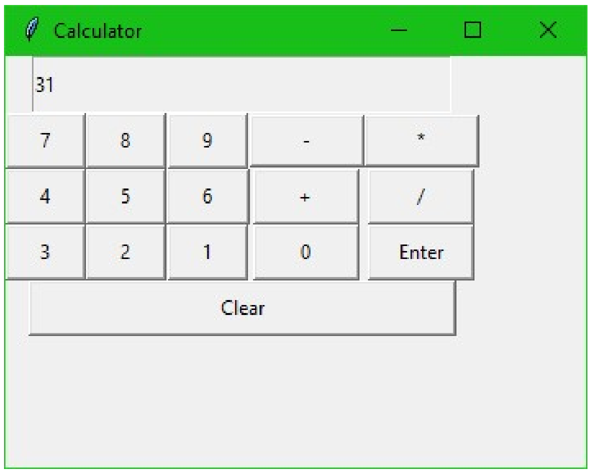
Program:

```
from tkinter import *

exp = " "
def press(num):
    global exp
    exp+=str(num)
    equation.set(exp)

def equalpress():
    try:
        global exp
        total = str(eval(exp))
        equation.set(total)
        exp = " "
    except:
        equation.set('error')
        exp = " "


def clear ():
    global exp
    exp = " "
    equation.set(" ")



if __name__ == "__main__":


    dk = Tk()
    dk.title('Calculator')
    dk.geometry('258x170')

equation = StringVar()
dis_entry = Entry(dk,width = 40, state = 'readonly', background ='red', textvariable =
equation)
dis_entry.grid(row = 0 , columnspan = 9, ipadx = 6 , ipady = 8,column = 0)
dis_entry.focus()

btn7 = Button(dk, text = '7' , width = 5 ,  command = lambda : press(7) )
btn7.grid(row = 1 , column = 0 ,ipady = 4 , ipadx = 2)

btn8 = Button(dk, text = '8' , width = 5 ,  command = lambda : press(8) )
btn8.grid(row = 1 , column = 1 ,ipady = 4, ipadx = 2)

btn9 = Button(dk, text = '9' , width = 5 ,  command = lambda : press(9) )
btn9.grid(row = 1 , column = 2,ipady = 4, ipadx = 2)

btnmines = Button(dk, text = '-' , width = 8 ,  command = lambda : press('-') )
btnmines.grid(row = 1 , column = 3 , ipady = 3, ipadx = 2)

btnmul = Button(dk, text = '*' , width = 8 ,  command = lambda : press("*") )
btnmul.grid(row = 1 , column = 4 , ipady = 3, ipadx = 2)

btn4 = Button(dk, text = '4' , width = 5 ,  command = lambda : press(4) )
btn4.grid(row = 2 , column = 0 ,ipady = 4 , ipadx = 2)

btn5 = Button(dk, text = '5' , width = 5 ,  command = lambda : press(5) )
btn5.grid(row = 2 , column = 1 ,ipady = 4, ipadx = 2)

btn6 = Button(dk, text = '6' , width = 5 ,  command = lambda : press(6) )
btn6.grid(row = 2 , column = 2,ipady = 4, ipadx = 3)

btnplus = Button(dk, text = '+' , width = 5 ,  command = lambda : press("+") )
btnplus.grid(row = 2 , column = 3,ipady = 4, ipadx = 10)
```

```
btndiv = Button(dk, text = '/' , width = 5 ,  command = lambda : press("/")  )
btndiv.grid(row = 2 , column = 4,ipady = 4, ipadx = 10)

btnequal = Button(dk, text = 'Enter' , width = 5,  command = equalpress )
btnequal.grid(row = 3 , column = 4,ipady = 4, ipadx = 10)

btn0= Button(dk, text = '0' , width = 5,  command = lambda : press(0)  )
btn0.grid(row = 3 , column = 3,ipady = 4, ipadx = 10)

btn3 = Button(dk, text = '3' , width = 5,  command = lambda : press(3)  )
btn3.grid(row = 3 , column = 0 ,ipady = 4 , ipadx = 2)

btn2 = Button(dk, text = '2' , width = 5,  command = lambda : press(2)  )
btn2.grid(row = 3 , column = 1 ,ipady = 4, ipadx = 2)

btn1 = Button(dk, text = '1' , width = 5 ,  command = lambda : press(1) )
btn1.grid(row = 3 , column = 2,ipady = 4, ipadx = 2)

btnclr = Button(dk, text = 'Clear' , width = 5 ,  command = clear )
btnclr.grid(row = 4 , columnspan = 6,ipady = 4, ipadx = 108)

dk.mainloop()
```

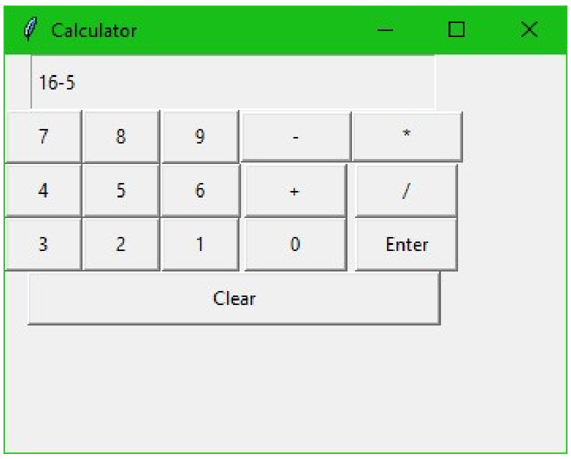
Output:
Addition

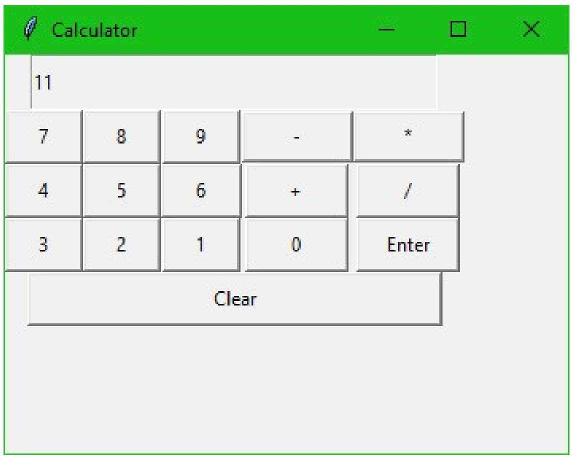




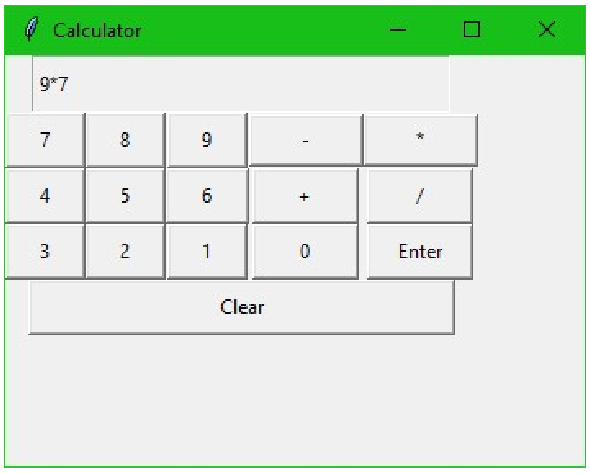
Name: C. Harrish
Roll No: 1801066

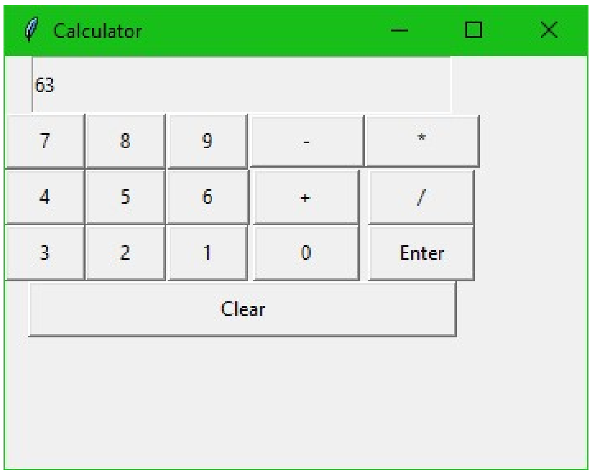
Subtraction:



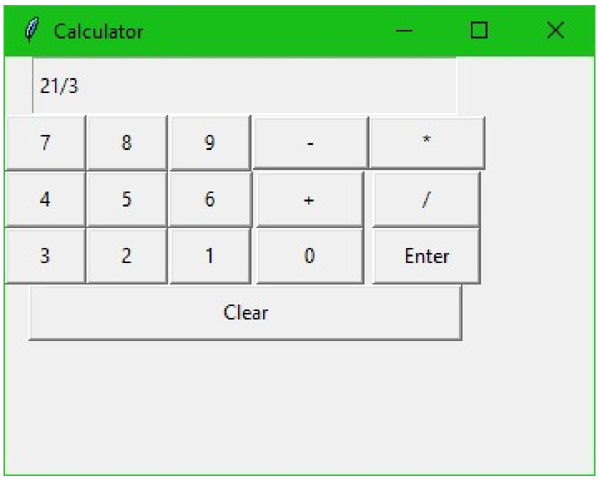


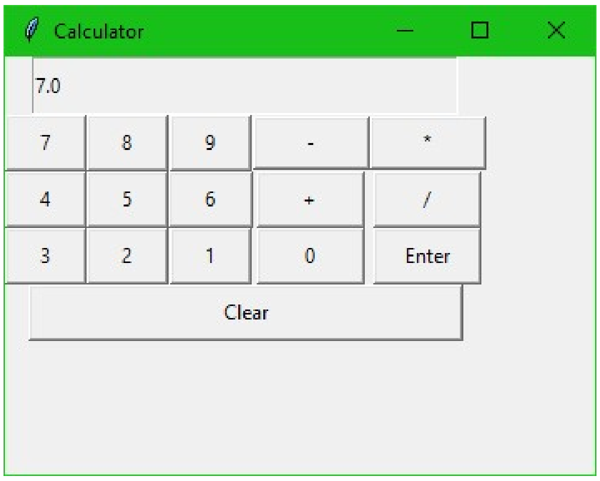
Multiplication:





Division





Name: C. Harrish
Roll No: 1801066

2. Create a tkinter application to open a text file and print the contents of the file in a tkinter window

Procedure:

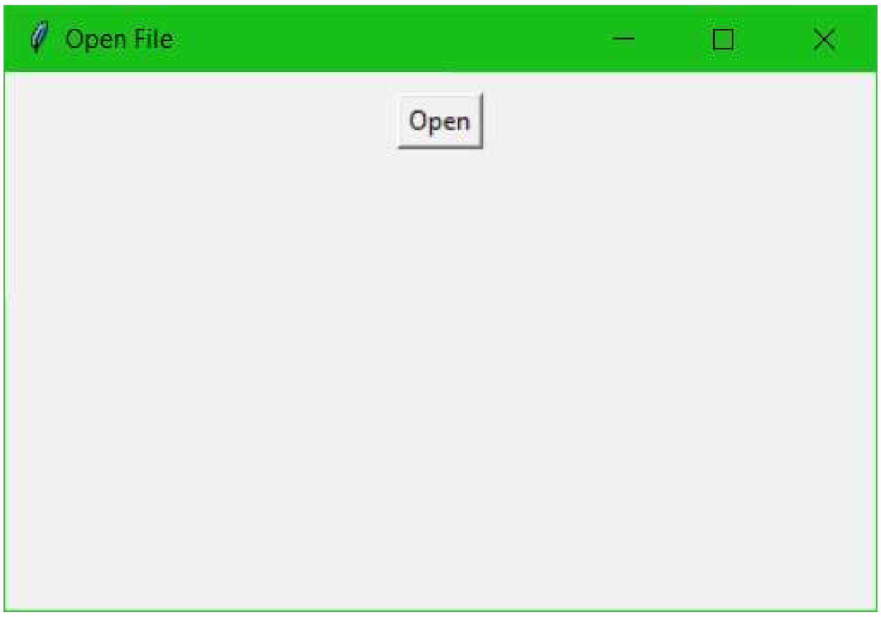
STEP 1: Import all tkinter and filedialog module
STEP 2: Initialize the tkinter window using Tk() and set the width and height
STEP 3: Create a button to open the dialog box
STEP 4: Select a file from the dialog box and click open
STEP 5: Print the content of the file in Tkinter

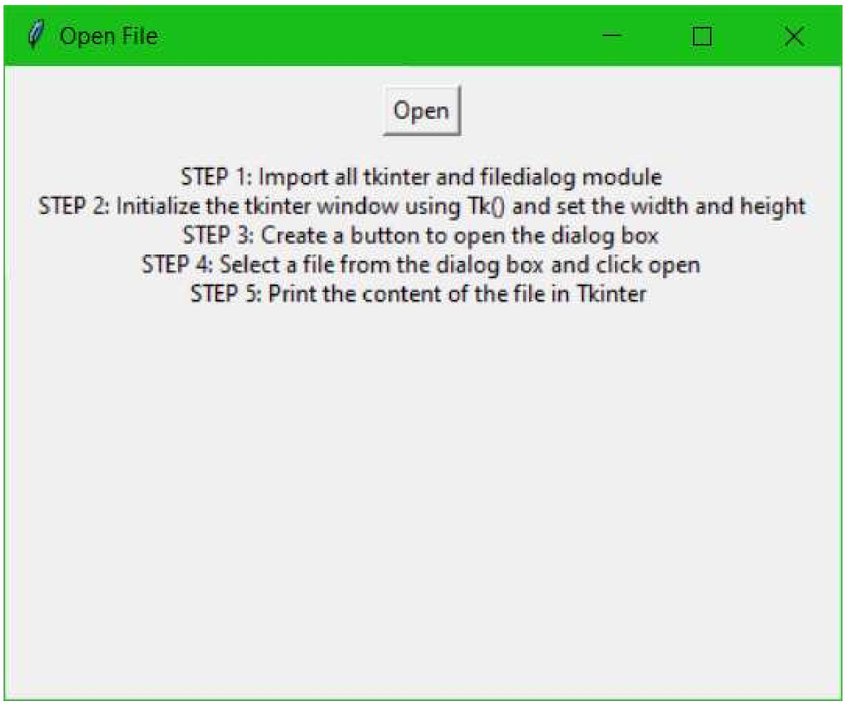
Program:

```
from tkinter import *
from tkinter import filedialog
root = Tk()
root.title("Open File")
root.geometry('200x100')
def open_file():
file = filedialog.askopenfile(initialdir="/",filetypes=[('All files','*.*')])
label1 = Label(text=file.read())
label1.pack()
btn = Button(root, text ='Open', command = lambda:open_file())
btn.pack(side = TOP, pady = 10)
root.mainloop()
```

Name: C. Harrish
Roll No: 1801066

Output:





```
STEP 1: Import all tkinter and filedialog module
STEP 2: Initialize the tkinter window using Tk() and set the width and height
STEP 3: Create a button to open the dialog box
STEP 4: Select a file from the dialog box and click open
STEP 5: Print the content of the file in Tkinter
```

Name: C. Harrish
Roll No: 1801066

NOMENCLATURE	**MAX MARKS**	**MARKS OBTAINED**
Pre-Lab Viva	05	
Pre-Lab Preparation	06	
In Lab Performance	07	
Post-Lab Viva	02	
Total	**20**	
Signature of Faculty		

Result:

Thus, the python program to create a GUI application using Tkinter was executed and verified successfully.

Name: C. Harrish
Roll No: 1801066

Ex No: 6 **Date:**	**Course Registration Form using various GUI**

Aim:

To write program using Python to create a registration form using different GUI modules and link it to the database.

Question:

1. Create a course registration form containing all possible widgets using various GUI's like Tkinter, Tk Interface eXtensions (Tix), Python MegaWidgets (PMW) and Tile/Ttk. Also create a DB and a table for maintaining the course registration details. If u apply for a course using GUI, the details should be saved to the Data Base.
   The registration form should contain the following details
   - Name
   - DOB (Use calendar)
   - Gender (Use drop down)
   - Mobile Number
   - E-mail ID
   - Address (Use text area)
   - Course Choice (Use combo box)
   - Reason for choosing the course (Text area input)

Procedure:

STEP 1: Import all tkinter module and mysql module for database connection

STEP 2: Connect the database with connect()

STEP 3: Create a submitbut() to insert the data into the database and validate the entry field for empty entries

STEP 4: Get the input from the user and when clicking the button the submitbut() will work

STEP 5: Check the database if the inserted value is present

Program:

```
import mysql.connector
from tkinter import *
from tkcalendar import *
from tkinter import ttk
from tkinter import messagebox
```

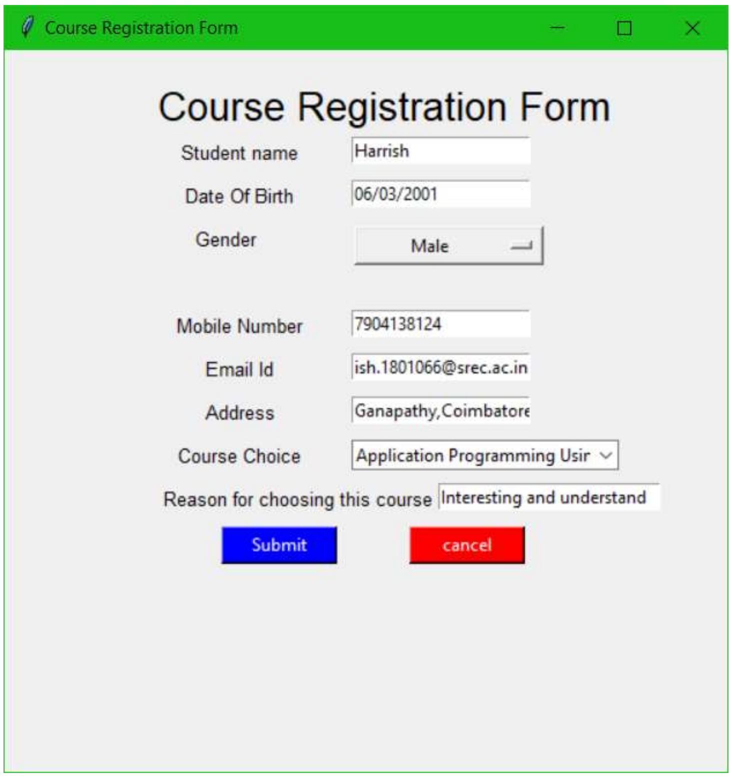
```
mydb =
mysql.connector.connect(host="localhost",user="root",password="harrish07",database="
python")
cursor = mydb.cursor()

def submitbut():
   name = name_t.get()
   dob = dob_a.get()
   gender = ce.get()
   phone = phno_t.get()
   email = mail_t.get()
   address = add_t.get()
   course = n.get()
   reason = rea_t.get()

   if not all((name,dob,gender,phone,email,course)) or len(address)==1 or
len(reason)==1:
      messagebox.showerror(title='Error', message='All Fields are Required')
      return
   sql="INSERT INTO register VALUES(%s,%s,%s,%s,%s,%s,%s,%s)"
   cursor.execute(sql, (name,dob,gender,phone,email,address,course,reason))
   mydb.commit()
   messagebox.showinfo("Registration","Registration successful")



root = Tk()
root.geometry('500x500')
root.title("Course Registration Form")

ce=StringVar()

reg_l = Label(root, text="Course Registration Form",width=20,font=("bold",20))
reg_l.place(x=100,y=20)

name_l = Label(root, text="Student name",width=20,font=("bold", 10))
name_l.place(x=80,y=60)
name = StringVar()
name_t = Entry(root,textvariable=name)
name_t.place(x=240,y=60)
dob_l = Label(root, text="Date Of Birth",width=20,font=("bold",10))
```

```
dob_l.place(x=80,y=90)
dob = StringVar()
dob_a = Entry(root,textvariable=dob)
dob_a.place(x=240,y=90)
gender_l = Label(root, text="Gender",width=20,font=("bold", 10))
gender_l.place(x=70,y=120)

list1 = ['Male','Female','Others'];

droplist=OptionMenu(root, ce, *list1)
droplist.config(width=15)
ce.set('Select your gender')
droplist.place(x=240,y=120)

phno_l = Label(root, text="Mobile Number",width=20,font=("bold", 10))
phno_l.place(x=80,y=180)
phone = StringVar()
phno_t = Entry(root, textvariable=phone)
phno_t.place(x=240,y=180)

mail_l = Label(root, text="Email Id",width=20,font=("bold", 10))
mail_l.place(x=80,y=210)
email = StringVar()
mail_t = Entry(root, textvariable=email)
mail_t.place(x=240,y=210)

add_l = Label(root, text="Address",width=20,font=("bold", 10))
add_l.place(x=80,y=240)
address = StringVar()
add_t = Entry(root, textvariable=address)
add_t.place(x=240,y=240)

course_l = Label(root, text="Course Choice", width=20,font=("bold", 10))
course_l.place(x=80,y=270)
n = StringVar()
course = ttk.Combobox(root, width = 27, textvariable = n)
course['values'] = ('Application Programming Using Python',
                    'Web Development',
                    ' Automata Compiler and Design',
                    'Software Engineering')
```

```
course.current(0)
course.place(x=240,y=270)

rea_l = Label(root, text="Reason for choosing this course",width=40,font=("bold", 10))
rea_l.place(x=40,y=300)
reason = StringVar()
rea_t = Entry(root, textvariable=reason,width=25)
rea_t.place(x=300,y=300)


Button(root,
text='Submit',width=10,bg='blue',fg='white',command=submitbut).place(x=150,y=330)
Button(root,
text='cancel',width=10,bg='red',fg='white',command=root.destroy).place(x=280,y=330)
root.mainloop()
```

Output:

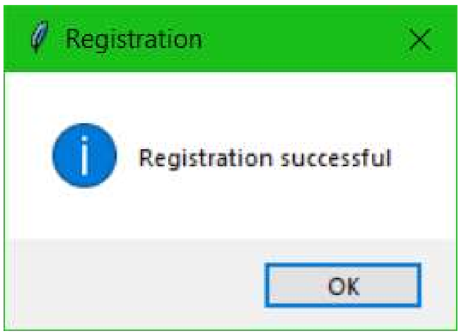
Registration Form



Course Registration Form

Course Registration Form

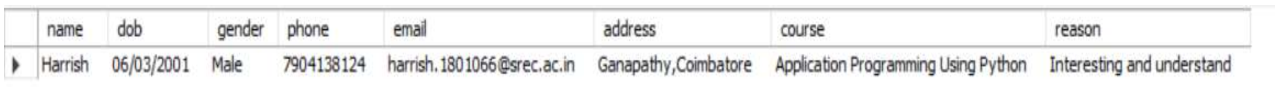
Student name: Harrish
Date Of Birth: 06/03/2001
Gender: Male
Mobile Number: 7904138124
Email Id: ish.1801066@srec.ac.in
Address: Ganapathy,Coimbatore
Course Choice: Application Programming Usir
Reason for choosing this course: Interesting and understand



Registration

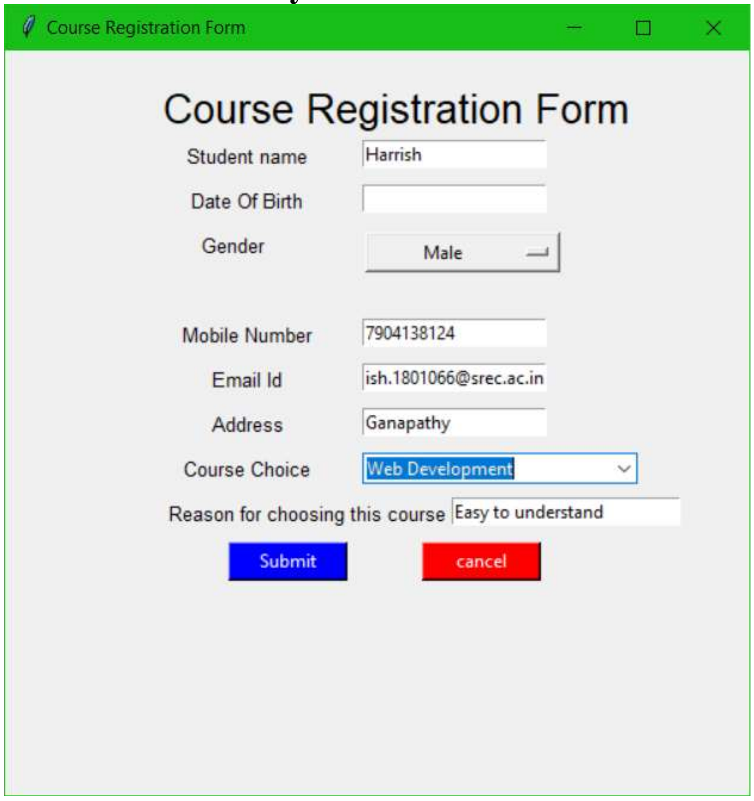
Registration successful

Database after insert



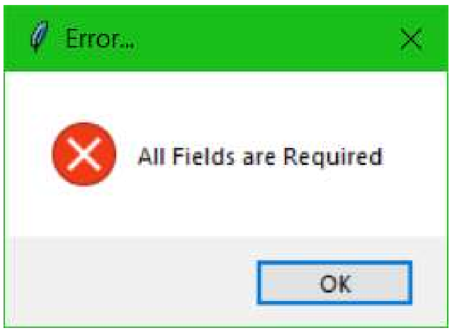
	name	dob	gender	phone	email	address	course	reason
▸	Harrish	06/03/2001	Male	7904138124	harrish.1801066@srec.ac.in	Ganapathy,Coimbatore	Application Programming Using Python	Interesting and understand

Validation of Entry fields



Course Registration Form

Student name	Harrish
Date Of Birth	
Gender	Male
Mobile Number	7904138124
Email Id	ish.1801066@srec.ac.in
Address	Ganapathy
Course Choice	Web Development
Reason for choosing this course	Easy to understand



All Fields are Required

Name: C. Harrish
Roll No: 1801066

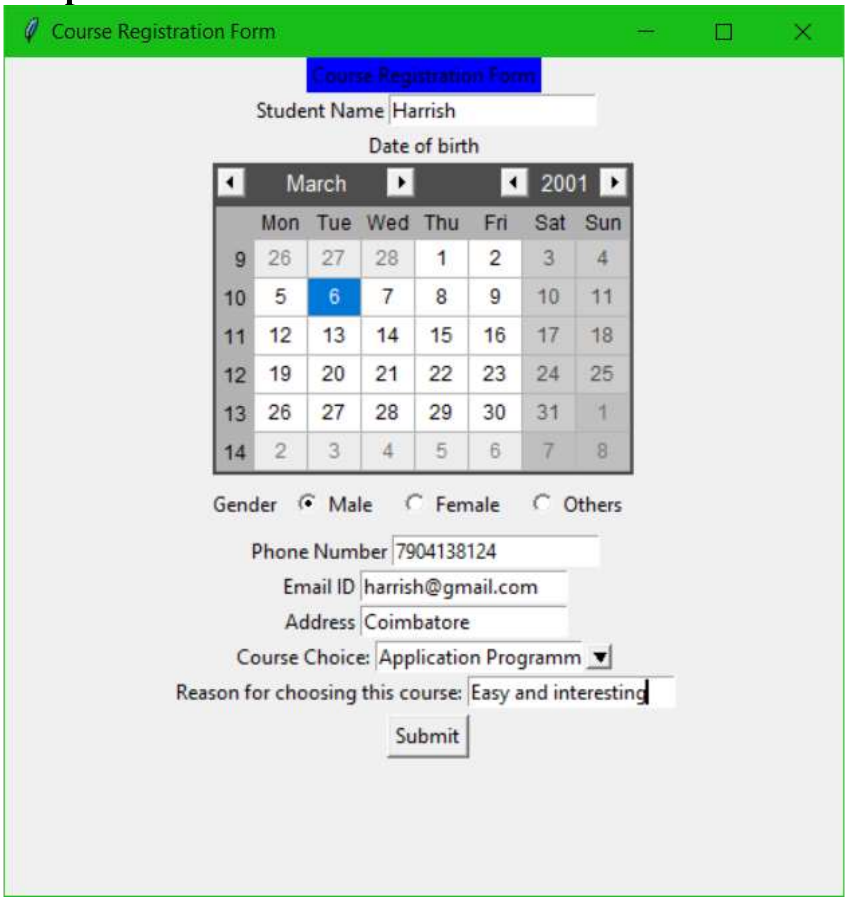
PMW(Python MegaWidgets)

Program:

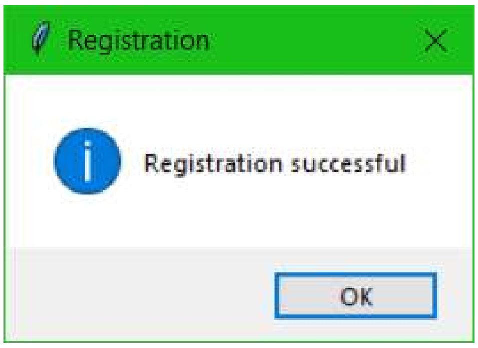
```
from tkinter import *
import Pmw
from tkcalendar import *
from tkinter import messagebox


root = Pmw.initialise()
root.title('Course Registration Form')
root.geometry('500x500')


l = Label(root, text="Course Registration Form",background="blue")
l.pack()
name = Pmw.EntryField(root, labelpos=W,label_text="Student Name")
name.pack()
dob_l = Label(root, text="Date of birth")
dob_l.pack()
dob = Calendar(root, selectmode="day",year=2020,month=8,day=26)
dob.pack()
gender = Pmw.RadioSelect(root, buttontype = 'radiobutton',orient = 'horizontal', labelpos
= 'w',label_text = 'Gender')
gender.pack()
for text in ('Male', 'Female','Others'):
    gender.add(text)
gender.invoke('Male')

phone = Pmw.EntryField(root, labelpos=W,label_text="Phone Number")
phone.pack()

email = Pmw.EntryField(root, labelpos=W,label_text="Email ID")
email.pack()

address = Pmw.EntryField(root,labelpos=W,label_text="Address")
address.pack()

choice = ('Application Programming Using Python', 'Web Development', ' Automata
Compiler and Design', 'Software Engineering')
combobox = Pmw.ComboBox(root, label_text='Course Choice:', labelpos='w',
                    listbox_width=24, dropdown=1,
                    scrolledlist_items=choice)
combobox.pack()
combobox.selectitem(choice[0])
```

```
reason = Pmw.EntryField(root, label_text="Reason for choosing this
course:",labelpos=W)
reason.pack()

def buttonPress(btn):
    messagebox.showinfo("Registration","Registration successful")
buttonBox = Pmw.ButtonBox(root, labelpos='w')
buttonBox.add('Submit', command = lambda b='submit':     buttonPress(b))
buttonBox.pack()
root.mainloop()
```

Output:



Student Name Harrish
Date of birth
Gender (●) Male ( ) Female ( ) Others
Phone Number 7904138124
Email ID harrish@gmail.com
Address Coimbatore
Course Choice: Application Programm
Reason for choosing this course: Easy and interesting



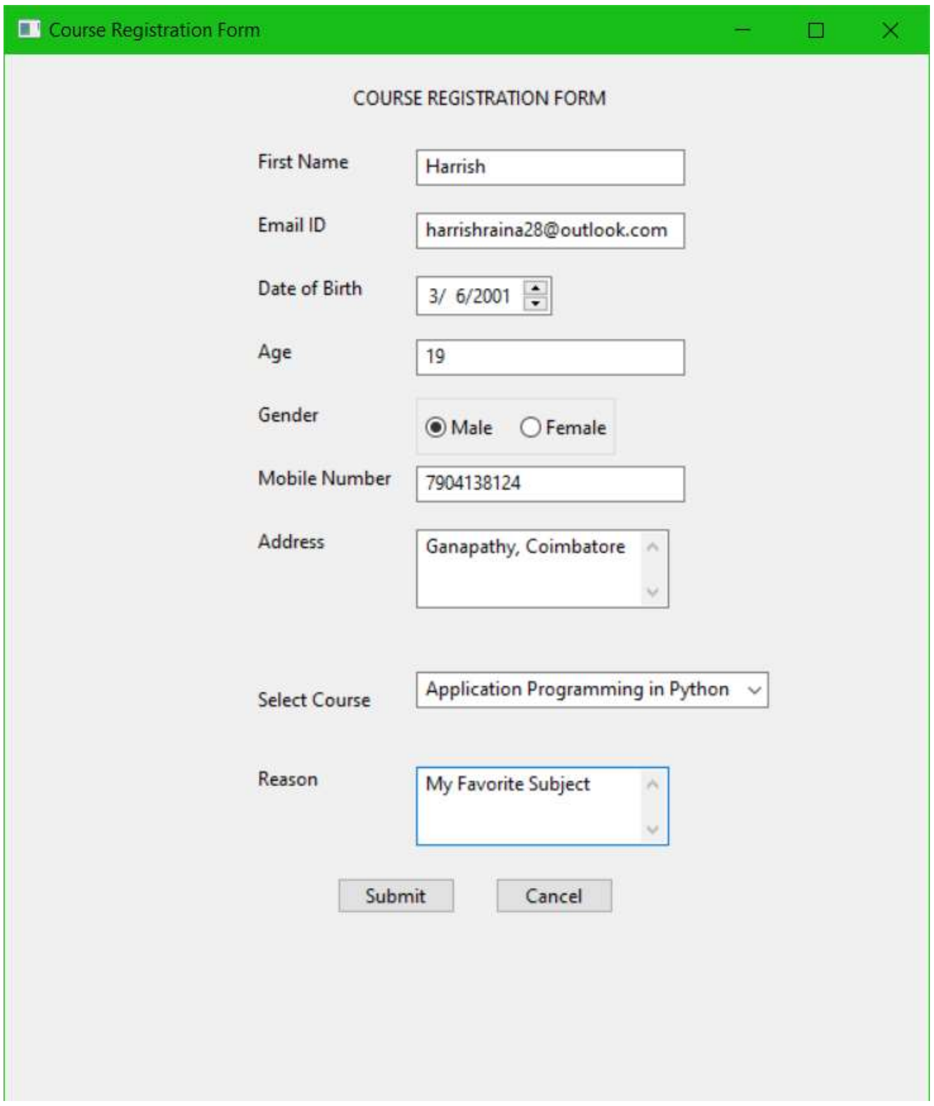
Registration successful

Name: C. Harrish
Roll No: 1801066

WX

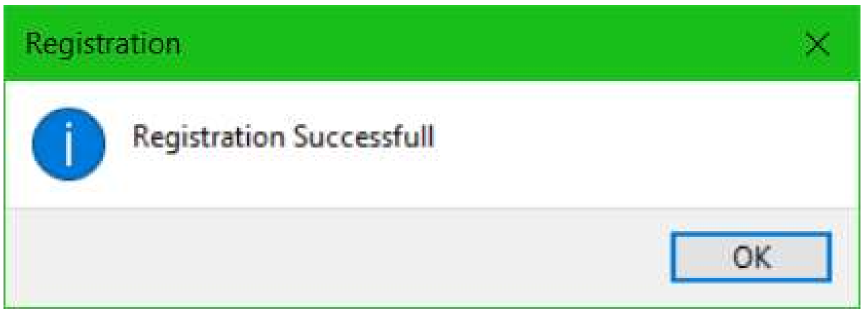
```
import wx
from wx.adv import DatePickerCtrl
class TextFrame(wx.Frame):
  def __init__(self):
    wx.Frame.__init__(self, None, -1, 'Course Registration Form', size=(600, 700))
    panel = wx.Panel(self, -1)
    Lbl = wx.StaticText(panel, -1, "COURSE REGISTRATION FORM", pos=(220,
20))
    fnameLbl = wx.StaticText(panel, -1, "First Name ", pos=(160, 60))
    fname = wx.TextCtrl(panel, -1, size=(170, -1), pos=(260, 60))
    fname.SetInsertionPoint(0)
    mailLbl = wx.StaticText(panel, -1, "Email ID ", pos=(160, 100))
    mail = wx.TextCtrl(panel, -1, size=(170, -1), pos=(260, 100))
    dobLbl = wx.StaticText(panel, -1, "Date of Birth ", pos=(160, 140))
    dob = DatePickerCtrl(panel, pos=(260, 140))
    ageLbl = wx.StaticText(panel, -1, "Age ", pos=(160, 180))
    age = wx.TextCtrl(panel, -1, size=(170, -1), pos=(260, 180))
    age.SetInsertionPoint(0)
    genLbl = wx.StaticText(panel, -1, "Gender ", pos=(160, 220))
    genderlst = ['Male', 'Female']
    rbox = wx.RadioBox(panel, choices=genderlst, majorDimension=1,
style=wx.RA_SPECIFY_ROWS, pos=(260, 210))
    mobLbl = wx.StaticText(panel, -1, "Mobile Number ", pos=(160, 260))
    mob = wx.TextCtrl(panel, -1, size=(170, -1), pos=(260, 260))
    addLbl = wx.StaticText(panel, -1, "Address ", pos=(160, 300))
    add = wx.TextCtrl(panel, -1, size=(160, 50), style=wx.TE_MULTILINE,
pos=(260,300))
    courseLbl = wx.StaticText(panel, -1, "Select Course ", pos=(160, 400))
    course = wx.ComboBox(panel, -1, pos=(260, 390), choices=['Application
Programming in Python', 'Web Technology', 'Software Engineering Practices','Automata
and Complier Design'])
    reasonLbl = wx.StaticText(panel, -1, "Reason", pos=(160, 450))
    reason = wx.TextCtrl(panel, -1, size=(160, 50), style=wx.TE_MULTILINE,
pos=(260,450))
    def sub(event):
      wx.MessageBox('Registration Successfull', "Registration", wx.OK
|wx.ICON_INFORMATION)
    def cancel(event):
      wx.Exit()
    btn = wx.Button(panel, -1, "Submit", pos=(210, 520))
    btn1 = wx.Button(panel, -1, "Cancel", pos=(310, 520))
app = wx.App()
frame = TextFrame()
frame.Show()
app.MainLoop()
```

OUTPUT


Course Registration Form

COURSE REGISTRATION FORM

First Name	Harrish
Email ID	harrishraina28@outlook.com
Date of Birth	3/ 6/2001
Age	19
Gender	[✓ Male] Female
Mobile Number	7904138124
Address	Ganapathy, Coimbatore
Select Course	Application Programming in Python
Reason	My Favorite Subject


Registration

Registration Successfull

Name: C. Harrish
Roll No: 1801066

NOMENCLATURE	**MAX MARKS**	**MARKS OBTAINED**
Pre-Lab Viva	05	
Pre-Lab Preparation	06	
In Lab Performance	07	
Post-Lab Viva	02	
Total	**20**	
Signature of Faculty		

Result:

Thus, the python program to create a various GUI application using Tkinter and PMW was executed and verified successfully

Name: C. Harrish
Roll No: 1801066

Ex No: 7 **Date:**	**Implementation of Single Threading vs Multi-Threading**

Aim:

To write program using Python implement single threading and multi-threading for the same application and compare the performance.

Question:

Write a python program to show the implementation of Single Threading vs MultiThreading. (You can consider any application with three threads and you have to show how these three threads can be executed following the single thread and multi thread methods).

Procedure:

STEP 1: Import threading module and time
STEP 2: Create three functions example fib(),print_add,print_mul
STEP 3: Create a list and store the functions
STEP 4: For single threading run the functions one by one and print the starting, finishing time and result
STEP 5: For multithreading use Thread() to create a thread and start() to start the thread and join() to finish
STEP 6: Repeat the Step 4 to print

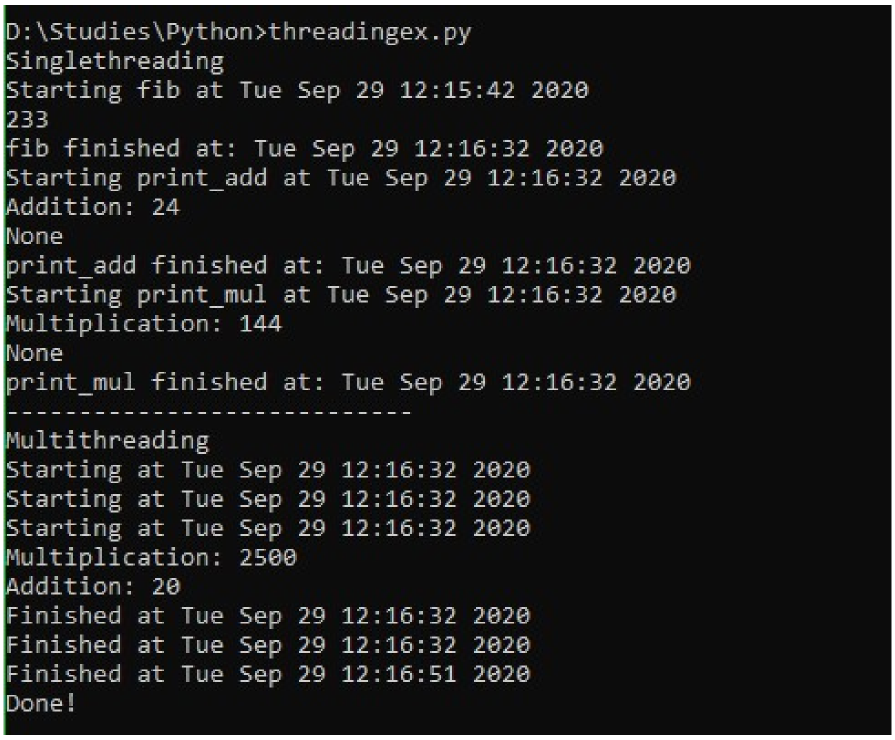
Program:

```
import threading
import time
import random

def fib(num1):
  time.sleep(0.1)
  if num1 < 2: return 1
  return (fib(num1 - 2) + fib(num1 - 1))

def print_add(num1):
  time.sleep(0.1)
  sub = num1 + num1
  print("Addition:",sub)
def print_mul(num1):
  time.sleep(0.1)
  mul = num1 * num1
  print("Multiplication:",mul)

funcs=[fib,print_add,print_mul]
```

```
n = 12

if __name__ == "__main__":

  nfuncs = range(len(funcs))

  print("Singlethreading")
  for i in nfuncs:
    print("Starting",funcs[i].__name__, "at", time.ctime())
    print(funcs[i](n))
    print(funcs[i].__name__,"finished at:", time.ctime())
  print("----------------------------")


if __name__ == "__main__":
  print("Multithreading")
  t1 = threading.Thread(target=print_add, args=(10,))
  print("Starting at",time.ctime())
  t2 = threading.Thread(target=print_mul, args=(50,))
  print("Starting at",time.ctime())
  t3 = threading.Thread(target=fib, args=(10,))
  print("Starting at",time.ctime())

  t1.start()
  t2.start()
  t3.start()

  t1.join()
  print("Finished at",time.ctime())
  t2.join()
  print("Finished at",time.ctime())
  t3.join()
  print("Finished at",time.ctime())

  print("Done!")
```

Name: C. Harrish
Roll No: 1801066

Output:



```
D:\Studies\Python>threadingex.py
Singlethreading
Starting fib at Tue Sep 29 12:15:42 2020
233
fib finished at: Tue Sep 29 12:16:32 2020
Starting print_add at Tue Sep 29 12:16:32 2020
Addition: 24
None
print_add finished at: Tue Sep 29 12:16:32 2020
Starting print_mul at Tue Sep 29 12:16:32 2020
Multiplication: 144
None
print_mul finished at: Tue Sep 29 12:16:32 2020
----------------------------
Multithreading
Starting at Tue Sep 29 12:16:32 2020
Starting at Tue Sep 29 12:16:32 2020
Starting at Tue Sep 29 12:16:32 2020
Multiplication: 2500
Addition: 20
Finished at Tue Sep 29 12:16:32 2020
Finished at Tue Sep 29 12:16:32 2020
Finished at Tue Sep 29 12:16:51 2020
Done!
```

NOMENCLATURE	MAX MARKS	MARKS OBTAINED
Pre-Lab Viva	05	
Pre-Lab Preparation	06	
In Lab Performance	07	
Post-Lab Viva	02	
Total	**20**	
Signature of Faculty		

Result:

Thus, the python program to implement single threading and multi-threading was executed and verified successfully.

Name: C. Harrish
Roll No: 1801066

Ex No: 8 **Date:**	**Implementation of Producer / Consumer Problem**

Aim:

Implementation of Producer/Consumer Problem**.** To write program using Python implement producer / consumer problem using single threading and multi-threading.

Question:

Model Producer-Consumer Strategy using Queue with and without threading and analyse its execution methodology

Procedure:

STEP 1: Import threading, multiprocessing for queue and time modules

STEP 2: Create a class called MyThread() and initial variables

STEP 3: Create a function called run() to print the starting and finishing time

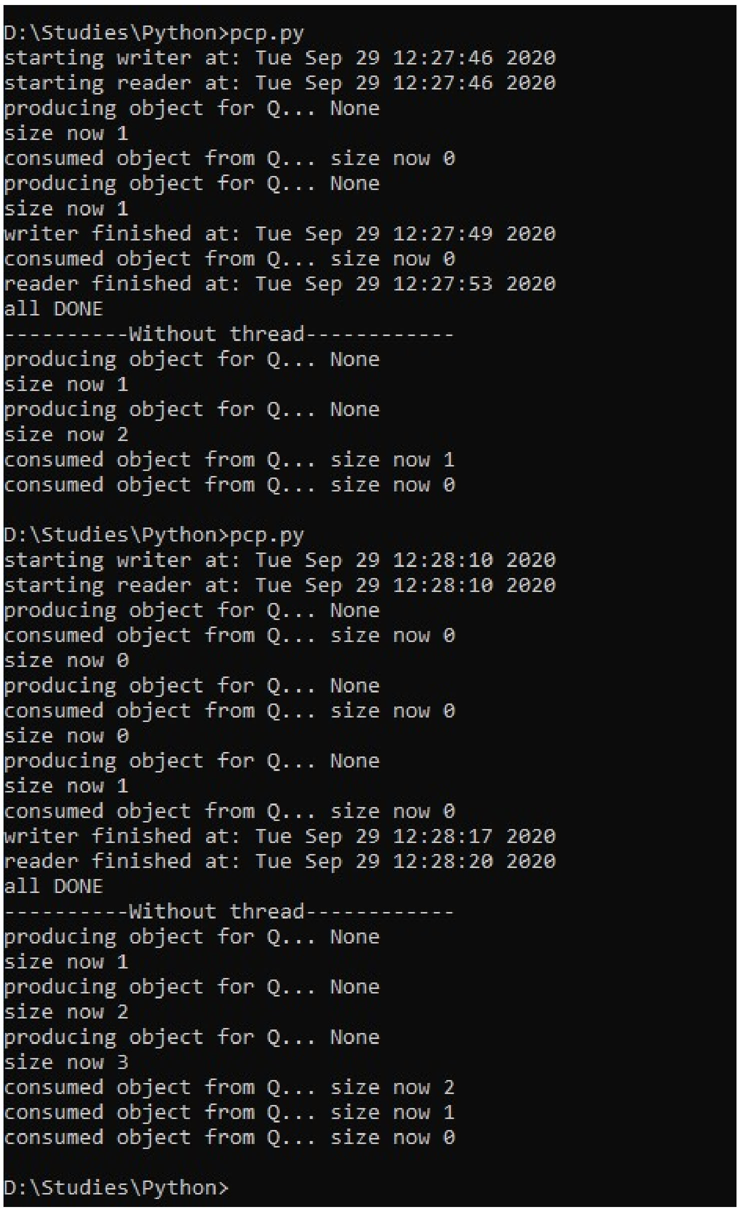
STEP 4: The writeQ and readQ is used for storing and accessing by producer and consumer

STEP 5: For Single threading run it in the loops and print the result

Program:

```
from random import randint
from time import sleep,ctime
from multiprocessing import Queue
import threading
class MyThread(threading.Thread):
    def __init__(self, func, args, name=''):
        threading.Thread.__init__(self)
        self.name = name
        self.func = func
        self.args = args
    def getResult(self):
        return self.res
    def run(self):
        print ('starting', self.name, 'at:', \
                ctime())
            self.res = self.func(*self.args)
            print (self.name, 'finished at:', \
                ctime())
def writeQ(queue):
    print ('producing object for Q...',
    queue.put('xxx', 1))
    print ("size now", queue.qsize())
def readQ(queue):
    val = queue.get(1)
    print ('consumed object from Q... size now',
    queue.qsize())
def writer(queue, loops):
    for i in range(loops):
        writeQ(queue)
        sleep(randint(1, 3))
def reader(queue, loops):
    for i in range(loops):
        readQ(queue)
        sleep(randint(2, 5))
funcs = [writer, reader]
nfuncs = range(len(funcs))
def main():
    nloops = randint(2, 5)
    q = Queue(32)
    threads = []
    for i in nfuncs:
        t = MyThread(funcs[i], (q, nloops), funcs[i].__name__)
        threads.append(t)
    for i in nfuncs:
        threads[i].start()
    for i in nfuncs:
        threads[i].join()
    print ('all DONE')
    print('----------Without thread------------')
    for i in nfuncs:
        funcs[i](q,nloops)
if __name__ == '__main__':
    main()
```

Output:



```
D:\Studies\Python>pcp.py
starting writer at: Tue Sep 29 12:27:46 2020
starting reader at: Tue Sep 29 12:27:46 2020
producing object for Q... None
size now 1
consumed object from Q... size now 0
producing object for Q... None
size now 1
writer finished at: Tue Sep 29 12:27:49 2020
consumed object from Q... size now 0
reader finished at: Tue Sep 29 12:27:53 2020
all DONE
----------Without thread------------
producing object for Q... None
size now 1
producing object for Q... None
size now 2
consumed object from Q... size now 1
consumed object from Q... size now 0

D:\Studies\Python>pcp.py
starting writer at: Tue Sep 29 12:28:10 2020
starting reader at: Tue Sep 29 12:28:10 2020
producing object for Q... None
consumed object from Q... size now 0
size now 0
producing object for Q... None
consumed object from Q... size now 0
size now 0
producing object for Q... None
size now 1
consumed object from Q... size now 0
writer finished at: Tue Sep 29 12:28:17 2020
reader finished at: Tue Sep 29 12:28:20 2020
all DONE
----------Without thread------------
producing object for Q... None
size now 1
producing object for Q... None
size now 2
producing object for Q... None
size now 3
consumed object from Q... size now 2
consumed object from Q... size now 1
consumed object from Q... size now 0

D:\Studies\Python>
```

Name: C. Harrish
Roll No: 1801066

NOMENCLATURE	MAX MARKS	MARKS OBTAINED
Pre-Lab Viva	05	
Pre-Lab Preparation	06	
In Lab Performance	07	
Post-Lab Viva	02	
Total	**20**	
Signature of Faculty		

Result:

Thus, the python program implement producer / consumer problem using single and multithreading was executed and verified successfully.