CMR TECHNICAL CAMPUS
UGC AUTONOMOUS

Kandlakoya(V), Medchal Road, Hyderabad – 501 401

Accredited by NBA and NAAC with A Grade

Approved by AICTE, New Delhi and Affiliated to JNTU, Hyderabad

DEPARTMENT OF CSE (AI & ML)

R PROGRAMMING LAB MANUAL

COURSE CODE: 20DS506PC (R20)

Prepared by:

S. SRIKANTH

Assistant Professor

20CS307PC: Python Lab

B.Tech. III SEM	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Prerequisites: Students should install Python on Linux platform.

Course Objectives:

- To be able to introduce core programming basics and program design with functions using Python programming language.
- To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.
- To understand the high-performance programs designed to strengthen the practical expertise.

Course Outcomes:

- Student should be able to understand the basic concepts scripting and the contributions of scripting language
- Ability to explore python especially the object oriented concepts, and the built in objects of Python.
- Ability to create practical and contemporary applications such as TCP/IP network programming, Web applications, discrete event simulations

PROGRAMMING WITH PYTHON LAB

Year: II Year I Sem

S. No	Experiment Name
1	Write a program to demonstrate different number datatypes in Python.
2	Write a program to perform different Arithmetic Operations on numbers in Python.
3	Write a program to create, concatenate and print a string and accessing sub-string from a given string.
4	Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST2017”
5	Write a program to create, append, and remove lists in python.
6	Write a program to demonstrate working with tuples in python.
7	Write a program to demonstrate working with dictionaries in python.
8	Write a python program to find largest of three numbers.
9	Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula: c/5 = f-32/9]
10	Write a Python program to construct the following pattern, using a nested for loop *
11	Write a Python script that prints prime numbers less than20.
12	Write a python program to find factorial of a number using Recursion.
13	Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is aright triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).

14	Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
15	Write a python program to define a module and import a specific function in that module to another program.
16	Write a script named **copyfile.py**. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.
17	Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
18	Write a Python class to convert an integer to a roman numeral.
19	Write a Python class to implement pow (x, n)
20	Write a Python class to reverse a string word by word.

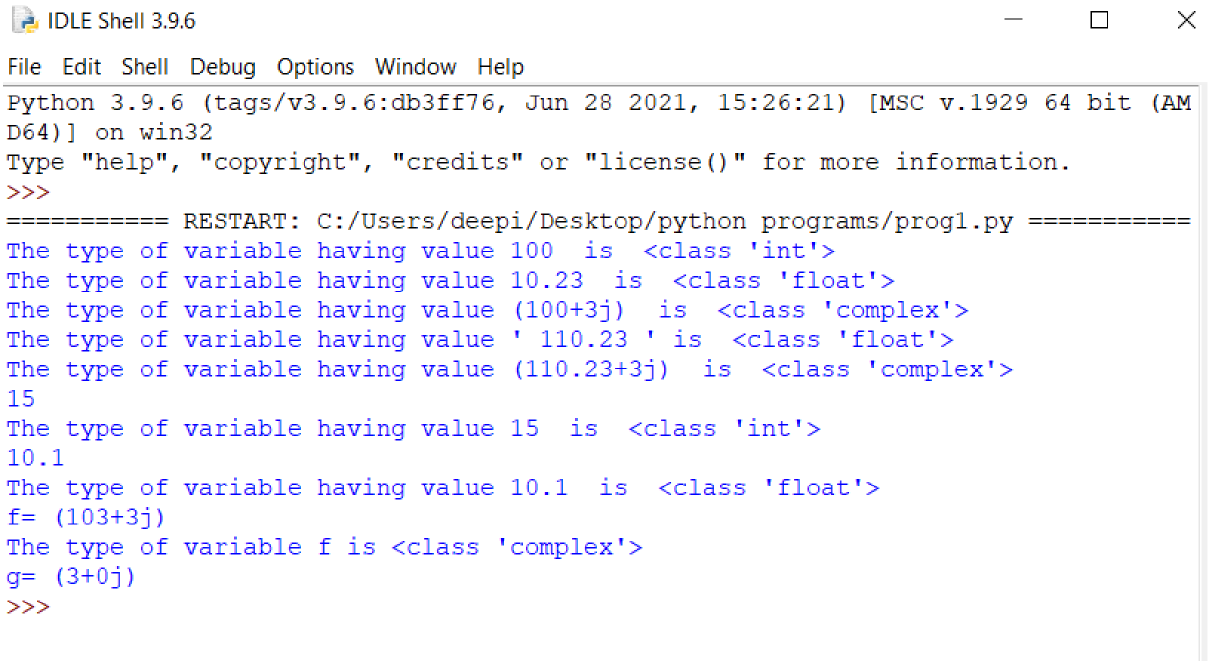
Experiment 1:

Aim: To write a program to demonstrate different number data types in Python.

Program:

```
a = 100

b= 10.23

c= 100+3j

print ("The type of variable having value", a, " is ", type(a))

print ("The type of variable having value", b, " is ", type(b))

print ("The type of variable having value", c, " is ", type(c))

print ("The type of variable having value '", a + b, "' is ", type (a + b))

print ("The type of variable having value", b + c, " is ", type (b + c))

d = int (15)

print(d)

print ("The type of variable having value", d, " is ", type(d))

e = float (10.1)

print ( e)

print ("The type of variable having value", e, " is ", type(e))

f = c + 3

print("f=",f)

print ("The type of variable f is", type(f))

g = complex (3)

print ("g=", g)
```

Output



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=========== RESTART: C:/Users/deepi/Desktop/python programs/prog1.py ===========
The type of variable having value 100  is  <class 'int'>
The type of variable having value 10.23  is  <class 'float'>
The type of variable having value (100+3j)  is  <class 'complex'>
The type of variable having value ' 110.23 ' is  <class 'float'>
The type of variable having value (110.23+3j)  is  <class 'complex'>
15
The type of variable having value 15  is  <class 'int'>
10.1
The type of variable having value 10.1  is  <class 'float'>
f= (103+3j)
The type of variable f is <class 'complex'>
g= (3+0j)
>>>
```

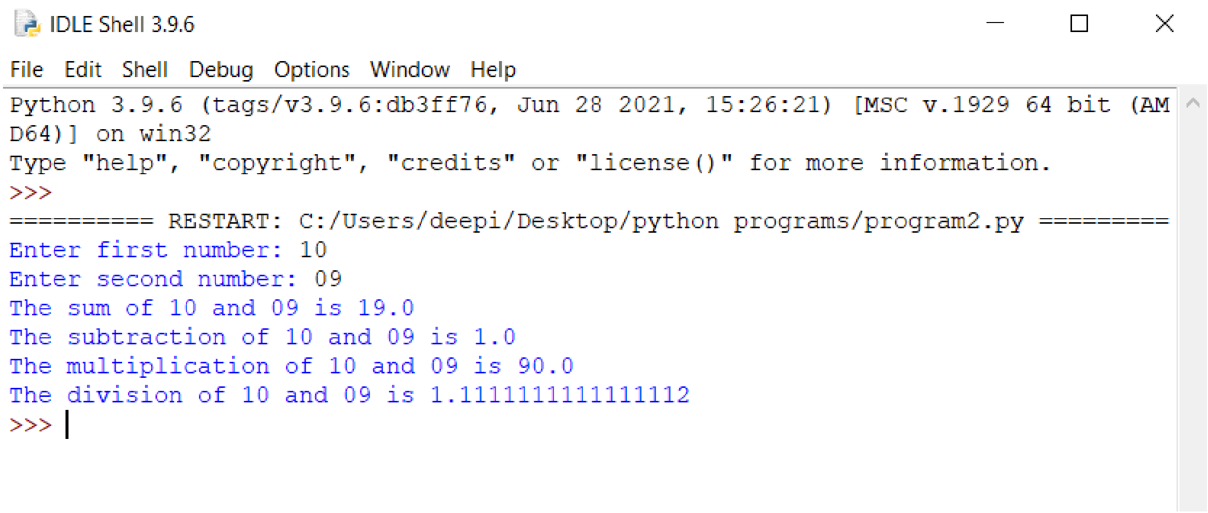
Experiment 2:

Aim: To write a program to perform different Arithmetic Operations on numbers in Python.

Program:

```
# Store input numbers:
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')
# Add two numbers
sum = float(num1) + float(num2)
# Subtract two numbers
min = float(num1) - float(num2)
# Multiply two numbers
mul = float(num1) * float(num2)
#Divide two numbers
div = float(num1) / float(num2)
div1 = float(num1) // float(num2)
# Display the sum
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
# Display the subtraction
print('The subtraction of {0} and {1} is {2}'.format(num1, num2, min))
# Display the multiplication
print('The multiplication of {0} and {1} is {2}'.format(num1, num2, mul))
# Display the division
print('The division of {0} and {1} is {2}'.format(num1, num2, div))
print('The floor division of {0} and {1} is {2}'.format(num1, num2, div1))
```

Output:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/deepi/Desktop/python programs/program2.py =========
Enter first number: 10
Enter second number: 09
The sum of 10 and 09 is 19.0
The subtraction of 10 and 09 is 1.0
The multiplication of 10 and 09 is 90.0
The division of 10 and 09 is 1.1111111111111112
>>>
```

Experiment 3:

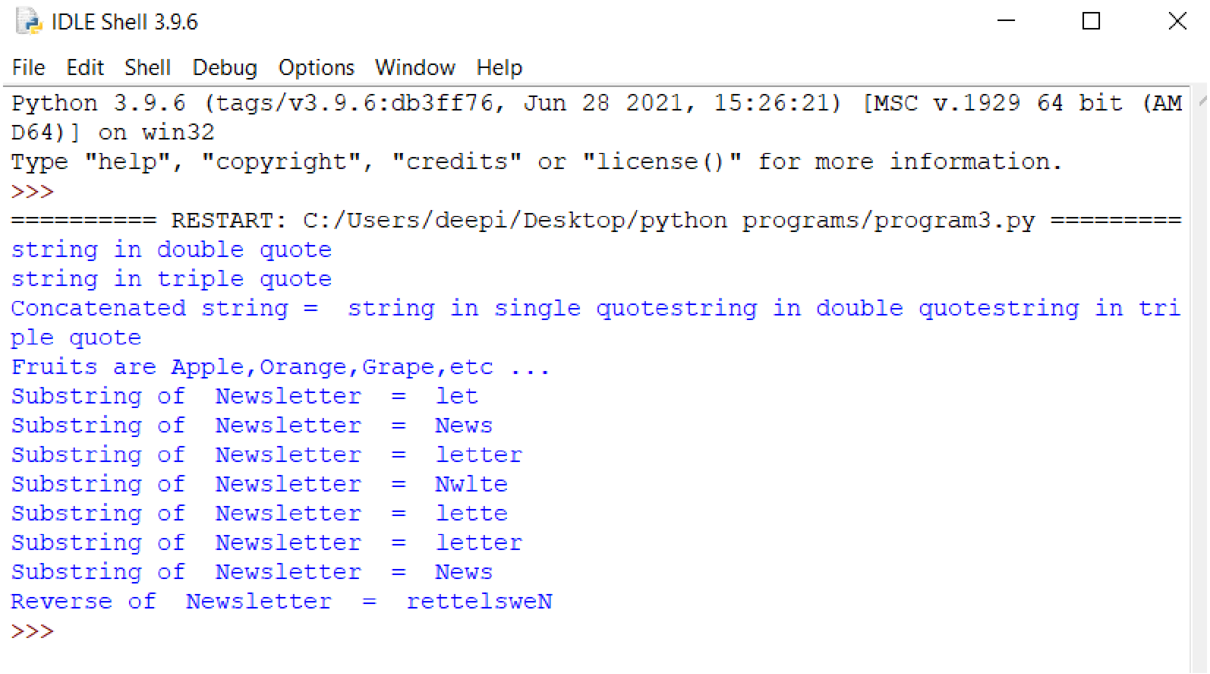
Aim: To write a program to create, concatenate and print a string and accessing sub-string from a given string.

Program:

```
# String creation

s1='string in single quote'

s2="string in double quote"

s3="""string in triple quote"""

s4='.'

s5='Newsletter'


# Printing string print(s1)

print(s2)

print(s3)


#string concatenation

s6=s1+s2+s3

print("Concatenated string = ",s6)

print("Fruits are Apple,Orange,Grape,etc",s4*3)

#substring or slicing

print("Substring of ",s5," = ",s5[4:7])

print("Substring of ",s5," = ",s5[:4])

print("Substring of ",s5," = ",s5[4:])

print("Substring of ",s5," = ",s5[::2])

print("Substring of ",s5," = ",s5[-6:-1])

print("Substring of ",s5," = ",s5[-6:])

print("Substring of ",s5," = ",s5[-10:-6])
```

print("Reverse of ",s5," = ",s5[::-1])

Output:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/deepi/Desktop/python programs/program3.py =========
string in double quote
string in triple quote
Concatenated string =  string in single quotestring in double quotestring in tri
ple quote
Fruits are Apple,Orange,Grape,etc ...
Substring of  Newsletter  =  let
Substring of  Newsletter  =  News
Substring of  Newsletter  =  letter
Substring of  Newsletter  =  Nwlte
Substring of  Newsletter  =  lette
Substring of  Newsletter  =  letter
Substring of  Newsletter  =  News
Reverse of  Newsletter  =  rettelsweN
>>>
```

Experiment 4

Aim: To write a program to print the current date in the following format “Sun May 29 02:26:23 IST 2017”

Program:

import datetime

from datetime import datetime

now = datetime.now()

current_time = now.strftime("%H:%M:%S")

current_year = now.strftime("%Y")

current_month=now.strftime("%B")

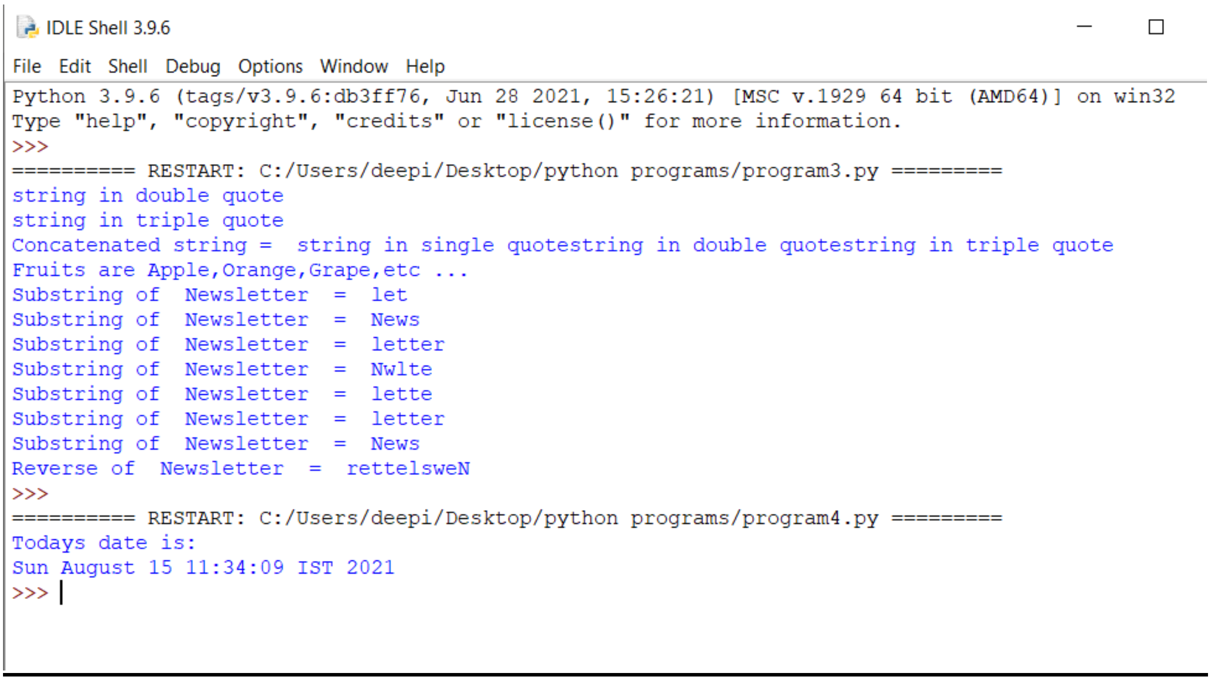
current_day = now.strftime("%a")

current_date = now.strftime("%d")

print("Todays date is:")

print("{} {} {} {} IST {}".format(current_day,current_month,current_date,current_time,current_year))

Output:



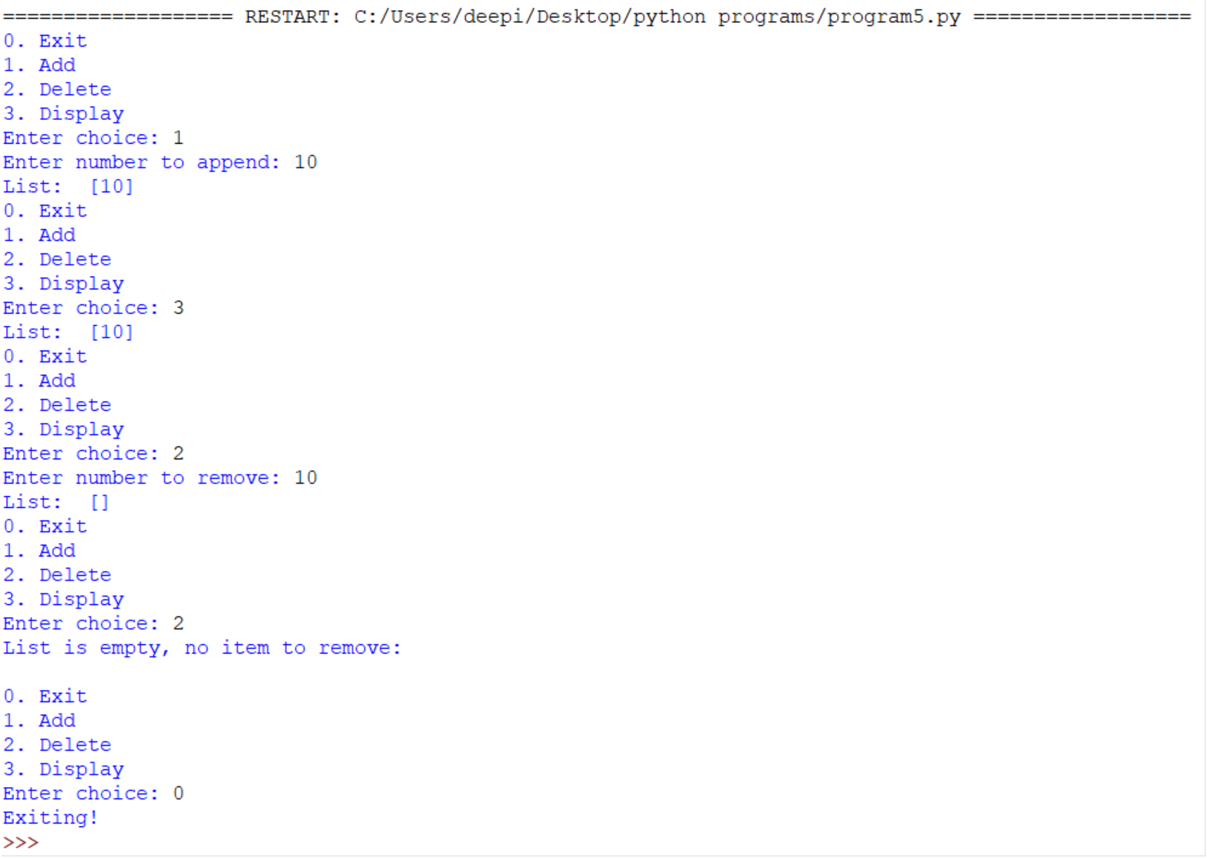
```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/deepi/Desktop/python programs/program3.py =========
string in double quote
string in triple quote
Concatenated string =  string in single quotestring in double quotestring in triple quote
Fruits are Apple,Orange,Grape,etc ...
Substring of  Newsletter  =  let
Substring of  Newsletter  =  News
Substring of  Newsletter  =  letter
Substring of  Newsletter  =  Nwlte
Substring of  Newsletter  =  lette
Substring of  Newsletter  =  letter
Substring of  Newsletter  =  News
Reverse of  Newsletter  =  rettelsweN
>>>
========== RESTART: C:/Users/deepi/Desktop/python programs/program4.py =========
Todays date is:
Sun August 15 11:34:09 IST 2021
>>> |
```

Experiment 5:

Aim: To write a program to create, append, and remove lists in python.

```
lst = []

choice=1

while choice!=0:

   print("0. Exit")

   print("1. Add")

   print("2. Delete")

   print("3. Display")

   choice=int(input("Enter choice: "))

   if choice==1:

     n=int(input("Enter number to append: "))

     lst.append(n)

     print("List: ",lst)

   elif choice==2:

     if len(lst)==0:

         print("List is empty, no item to remove:")

         print()

         continue

     n=int(input("Enter number to remove: "))

     if n not in lst:

        print("The item to be removed not in list:")

        print()

        continue

     lst.remove(n)

     print("List: ",lst)

   elif choice==3:
     print("List: ",lst)
   elif choice==0:
     print("Exiting!")
   else:
     print("Invalid choice!!")
     print()
```

Output:



```
=================== RESTART: C:/Users/deepi/Desktop/python programs/program5.py ==================
0. Exit
1. Add
2. Delete
3. Display
Enter choice: 1
Enter number to append: 10
List:  [10]
0. Exit
1. Add
2. Delete
3. Display
Enter choice: 3
List:  [10]
0. Exit
1. Add
2. Delete
3. Display
Enter choice: 2
Enter number to remove: 10
List:  []
0. Exit
1. Add
2. Delete
3. Display
Enter choice: 2
List is empty, no item to remove:

0. Exit
1. Add
2. Delete
3. Display
Enter choice: 0
Exiting!
>>>
```

Experiment 6:

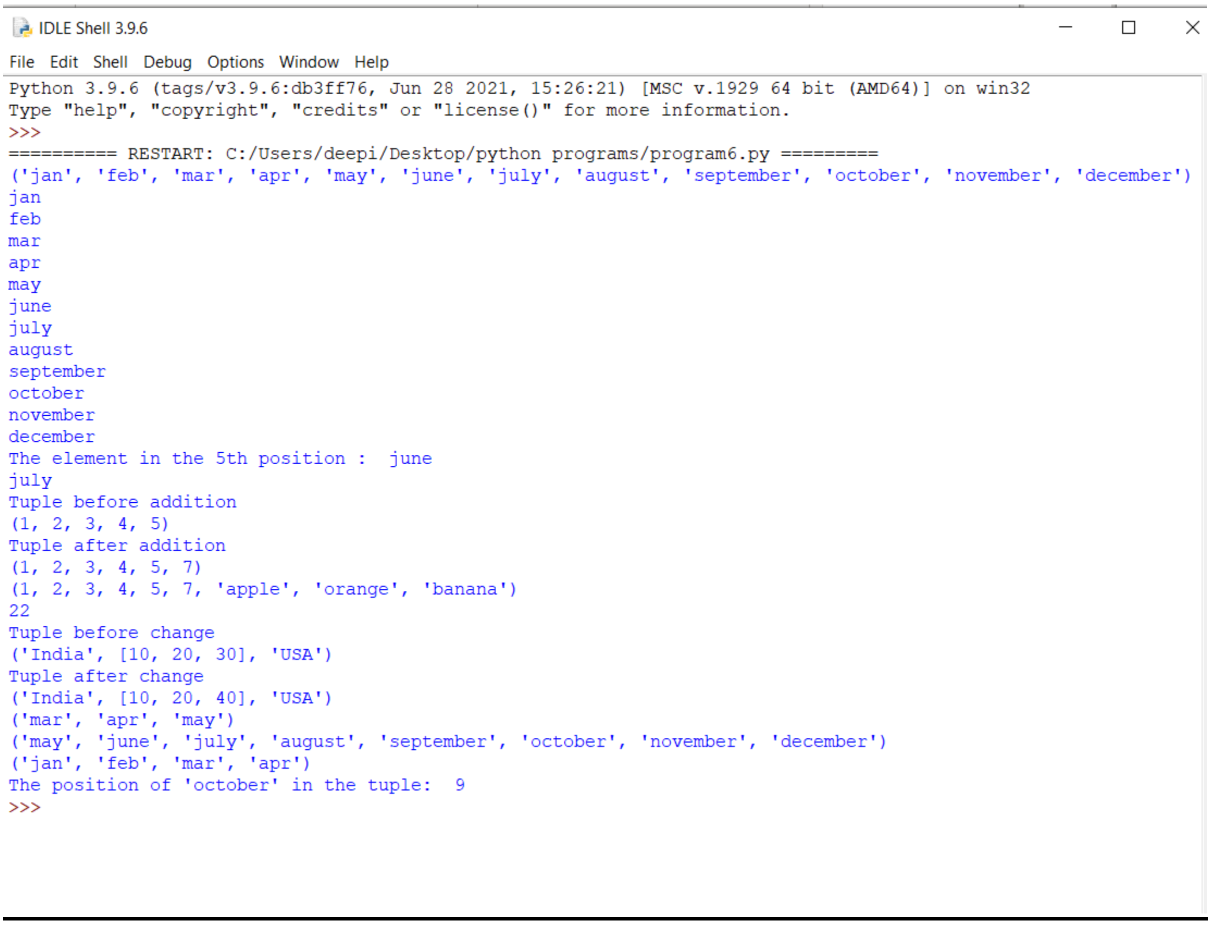
Aim: To write a program to demonstrate working with tuples in python.

Program:

```
tup1 =
('jan','feb','mar','apr','may','june','july','august','september','october','november','december')
tup2 = (1, 2, 3, 4, 5)
tup3=(1, "Hello", (11, 22, 33))
tup4=('India',[10,20,30],'USA')
#print the tuple
print(tup1)

#iterating in tuple
for mon in tup1:
  print(mon,end=' ')
  print()

#accessing the tuple element
print("The element in the 5th position : ",tup1[5])
print(tup1[-6])
print("Tuple before addition")
print(tup2)
tup2 = tup2 + (7,)
print("Tuple after addition")
print(tup2)
tup2=tup2+('apple','orange','banana')
print(tup2)

#accessing nested tuple element
print(tup3[2][1])

#tuple element is mutable, element change is possible
print("Tuple before change")
print(tup4)
tup4[1][2]=40
print("Tuple after change")
print(tup4)
```

```
#Slicing operation in tuples
print(tup1[2:5])
print(tup1[4:])
print(tup1[:4])

#finding position of the element
print("The position of 'october' in the tuple: ",tup1.index('october'))
```

Output:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/deepi/Desktop/python programs/program6.py =========
('jan', 'feb', 'mar', 'apr', 'may', 'june', 'july', 'august', 'september', 'october', 'november', 'december')
jan
feb
mar
apr
may
june
july
august
september
october
november
december
The element in the 5th position :  june
july
Tuple before addition
(1, 2, 3, 4, 5)
Tuple after addition
(1, 2, 3, 4, 5, 7)
(1, 2, 3, 4, 5, 7, 'apple', 'orange', 'banana')
22
Tuple before change
('India', [10, 20, 30], 'USA')
Tuple after change
('India', [10, 20, 40], 'USA')
('mar', 'apr', 'may')
('may', 'june', 'july', 'august', 'september', 'october', 'november', 'december')
('jan', 'feb', 'mar', 'apr')
The position of 'october' in the tuple:  9
>>>
```

Experiment 7:

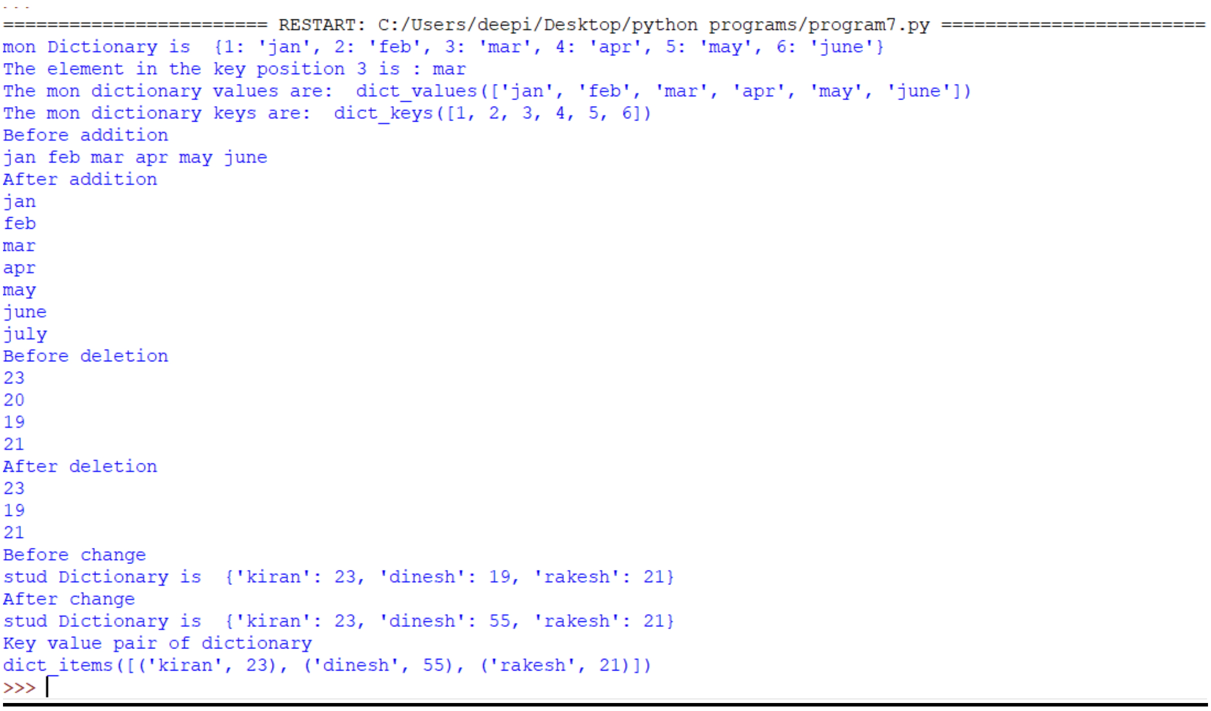
Aim: To Write a program to demonstrate working with dictionaries in python.

Program:

```
mon={1:'jan',2:'feb',3:'mar',4:'apr',5:'may',6:'june'}
stud={'kiran':23,'kumar':20,'dinesh':19,'rakesh':21}

print("mon Dictionary is ",str(mon))
print("The element in the key position 3 is :",mon[3])
print("The mon dictionary values are: ",mon.values())
print("The mon dictionary keys are: ",mon.keys())

#adding an item in the dictionary
print("Before addition")
for item in mon.values():
   print(item,end=' ')

mon[7]='july'
print()
print("After addition")
for item in mon.values():
   print(item,end=' ')
   print()
print("Before deletion")
for item in stud.values():
   print(item,end=' ')
   print()

del stud['kumar']

print("After deletion")
for item in stud.values():
    print(item,end=' ')
    print()

print("Before change")
print("stud Dictionary is ",str(stud))
stud['dinesh']=55
print("After change")
```

```
          print("stud Dictionary is ",str(stud))

          print("Key value pair of dictionary")
print(stud.items())
```

Output:



```
...
======================== RESTART: C:/Users/deepi/Desktop/python programs/program7.py ========================
mon Dictionary is  {1: 'jan', 2: 'feb', 3: 'mar', 4: 'apr', 5: 'may', 6: 'june'}
The element in the key position 3 is : mar
The mon dictionary values are:  dict_values(['jan', 'feb', 'mar', 'apr', 'may', 'june'])
The mon dictionary keys are:  dict_keys([1, 2, 3, 4, 5, 6])
Before addition
jan feb mar apr may june
After addition
jan
feb
mar
apr
may
june
july
Before deletion
23
20
19
21
After deletion
23
19
21
Before change
stud Dictionary is  {'kiran': 23, 'dinesh': 19, 'rakesh': 21}
After change
stud Dictionary is  {'kiran': 23, 'dinesh': 55, 'rakesh': 21}
Key value pair of dictionary
dict_items([('kiran', 23), ('dinesh', 55), ('rakesh', 21)])
>>>
```

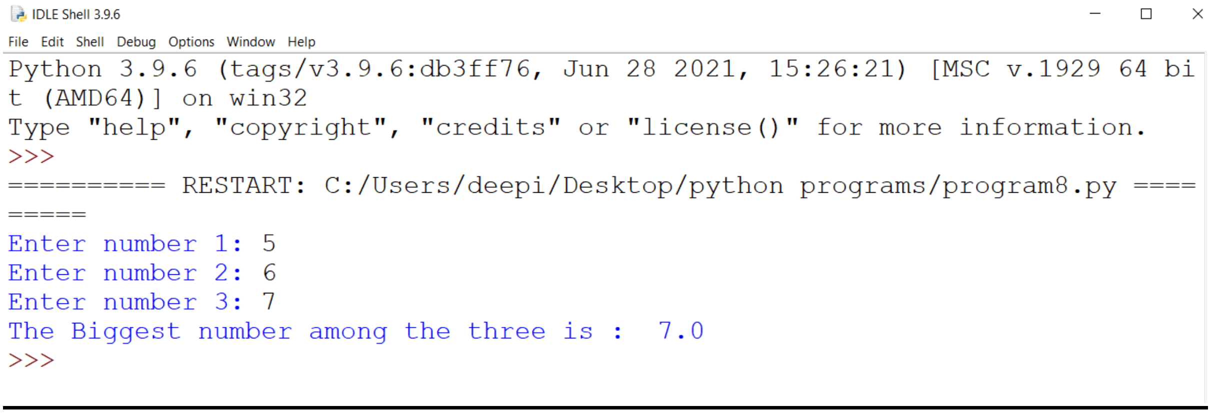
Experiment 8:

Aim: To write a python program to find largest of three numbers

Program:

```
num1 = float(input("Enter number 1: "))
num2 = float(input("Enter number 2: "))
num3 = float(input("Enter number 3: "))

big=num1

if (big<num2): big=num2
if (big<num3): big=num3

print("The Biggest number among the three is : ",big)
```

Output:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bi
t (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/deepi/Desktop/python programs/program8.py ====
=====
Enter number 1: 5
Enter number 2: 6
Enter number 3: 7
The Biggest number among the three is :  7.0
>>>
```

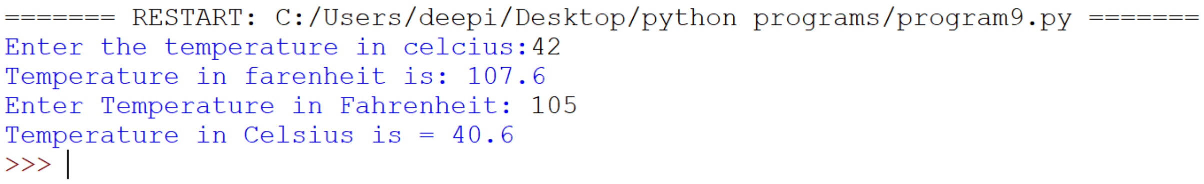
Experiment 9:

Aim: To write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula : c/5 = f-32/9]

Program:

```
        # Convertion from celsius to Farenheit

        celsius=float(input("Enter the temperature in celcius:"))
        f=(celsius*1.8)+32
        print("Temperature in farenheit is:", round(f,1))

        # Convertion from Fahrenheit to Celsius

        fahrenheit = float(input("Enter Temperature in Fahrenheit: "))
        c = (fahrenheit-32)/1.8
print("Temperature in Celsius is =", round(c,1))
```

Output:



```
======= RESTART: C:/Users/deepi/Desktop/python programs/program9.py =======
Enter the temperature in celcius:42
Temperature in farenheit is: 107.6
Enter Temperature in Fahrenheit: 105
Temperature in Celsius is = 40.6
>>> |
```

Experiment 10:

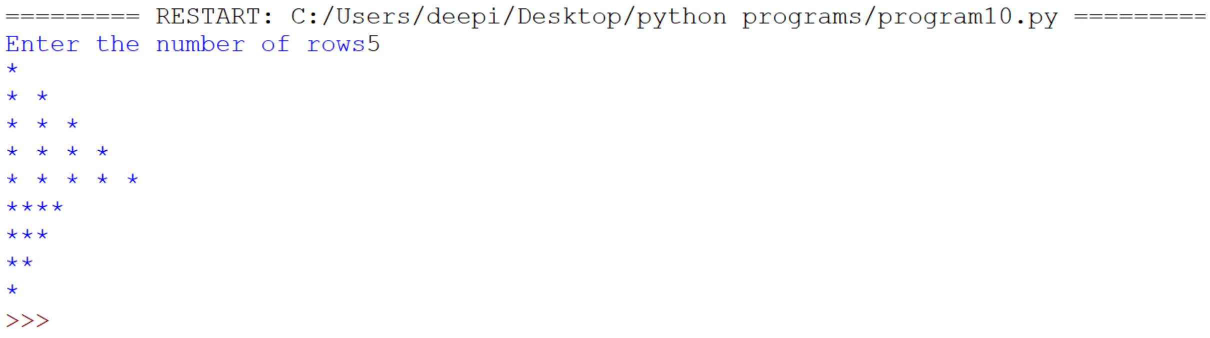
Aim: To Write a Python program to construct the following pattern, using a nested for loop

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

Program:

```
# This is the example of print simple pyramid pattern
n = int(input("Enter the number of rows"))
# outer loop to handle number of rows
for i in range(0, n):
    # inner loop to handle number of columns
    # values is changing according to outer loop
        for j in range(0, i + 1):
            # printing stars
            print("* ", end="")

        # ending line after each row
        print()
for i in range(n, 1, -1):
      for j in range(1,i):
        print("*",end="")
       print()
```

Output:



```
========= RESTART: C:/Users/deepi/Desktop/python programs/program10.py =========
Enter the number of rows5
*
* *
* * *
* * * *
* * * * *
****
***
**
*
>>>
```

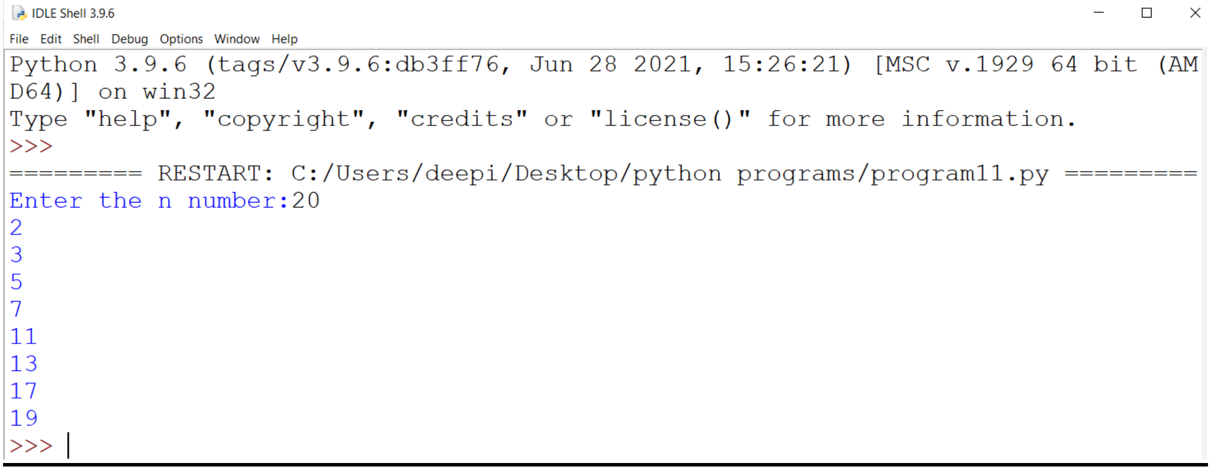
Experiment 11:

Aim: To write a Python script that prints prime numbers less than 20.

Program:

```
start_val = 1
n = int(input("Enter the n number:"))
for num in range(start_val, n+1):
    if(num>1):
        for i in range(2,num):
            if(num%i)==0:
                break
        else:
    print(num)
```

Output:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:/Users/deepi/Desktop/python programs/program11.py =========
Enter the n number:20
2
3
5
7
11
13
17
19
>>> 
```

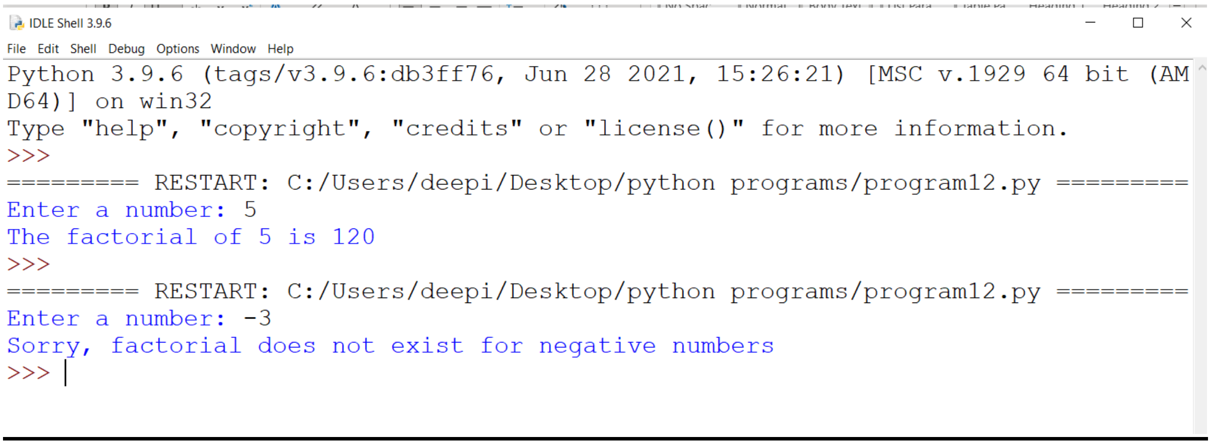
Experiment 12

Aim: To Write a python program to find factorial of a number using Recursion.

Program:

```
def factorial(n):
   if n == 1:
      return n
   else:
      return n* factorial(n-1)

# take input from the user
num = int(input("Enter a number: "))

# check if the number is negative
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:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:/Users/deepi/Desktop/python programs/program12.py =========
Enter a number: 5
The factorial of 5 is 120
>>>
========= RESTART: C:/Users/deepi/Desktop/python programs/program12.py =========
Enter a number: -3
Sorry, factorial does not exist for negative numbers
>>>
```

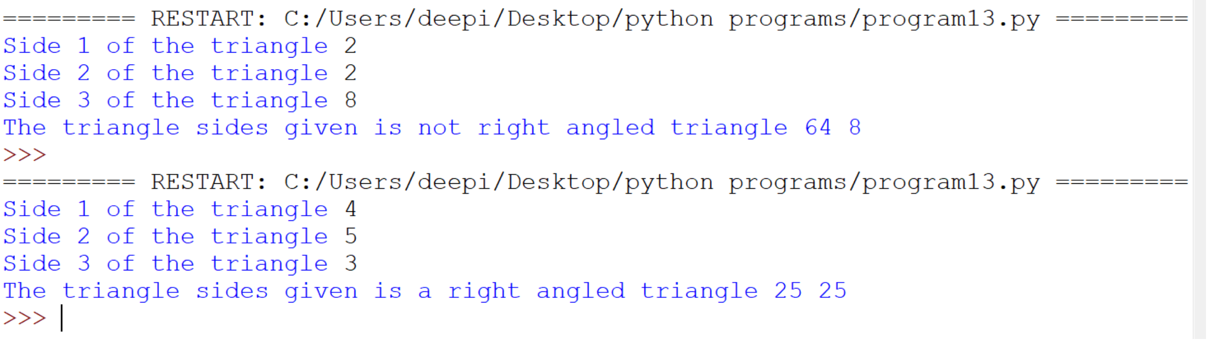
Experiment 13:

Aim: Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle

Program:

```
s1=int(input("Side 1 of the triangle "))
s2=int(input("Side 2 of the triangle "))
s3=int(input("Side 3 of the triangle "))
big = max(s1,s2,s3)
if(big==s1):
  if(s1**2 == (s2**2 + s3**2)):
    print("The triangle sides given is a right angled triangle", s1**2, (s2**2+s3**2))
  else:
      print("The triangle sides given is not right angled triangle",s1**2,
(s2**2+s3**2))
elif(big==s2):
    if(s2**2==(s1**2+s3**2)):
      print("The triangle sides given is a right angled triangle",s2**2,(s1**2+s3**2))
    else:
      print("The triangle sides given is not right angled triangle",
s2**2,(s1**2+s3**2))
elif(big==s3):
    if(s3**2==(s1**2+s2**2)):
      print("The triangle sides given is a right angled triangle",s3**2,(s1**2+s2**2))
    else:
          print("The triangle sides given is not right angled triangle", s3**2,
(s1**2+s2**2))
```

Output:



```
========= RESTART: C:/Users/deepi/Desktop/python programs/program13.py =========
Side 1 of the triangle 2
Side 2 of the triangle 2
Side 3 of the triangle 8
The triangle sides given is not right angled triangle 64 8
>>>
========= RESTART: C:/Users/deepi/Desktop/python programs/program13.py =========
Side 1 of the triangle 4
Side 2 of the triangle 5
Side 3 of the triangle 3
The triangle sides given is a right angled triangle 25 25
>>> |
```

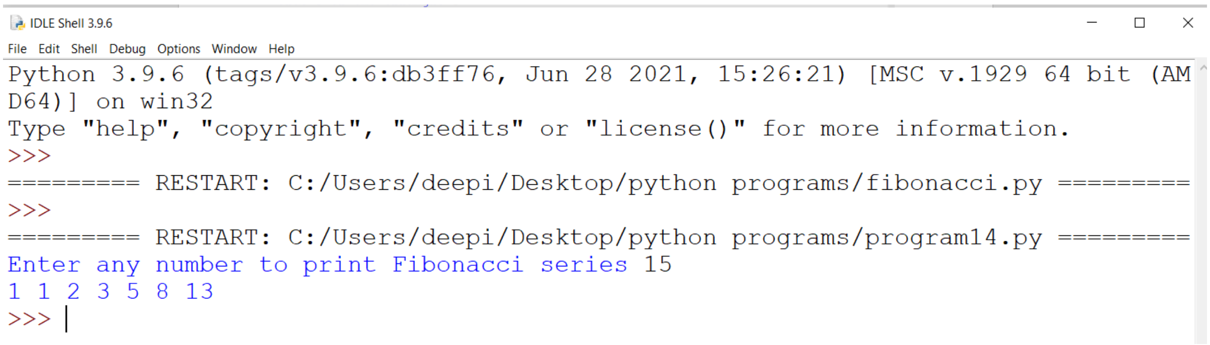
Experiment 14:

Aim: To Write a python program to define a module to find Fibonacci Numbers and import the module to another program.

Program:

```
    # Fibonacci numbers module


    def fib(n):    # write Fibonacci series up to n
       a, b = 0, 1
       while b < n:
          print(b, end =" ")
 a, b = b, a+b
    '''Write a python program to define a module to find Fibonacci Numbers and import the
    module to another program'''
    #import fibonacci module
    import fibonacci
    num=int(input("Enter any number to print Fibonacci series "))
    fibonacci.fib(num)
```

Output:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:/Users/deepi/Desktop/python programs/fibonacci.py =========
>>>
========= RESTART: C:/Users/deepi/Desktop/python programs/program14.py =========
Enter any number to print Fibonacci series 15
1 1 2 3 5 8 13
>>> 
```

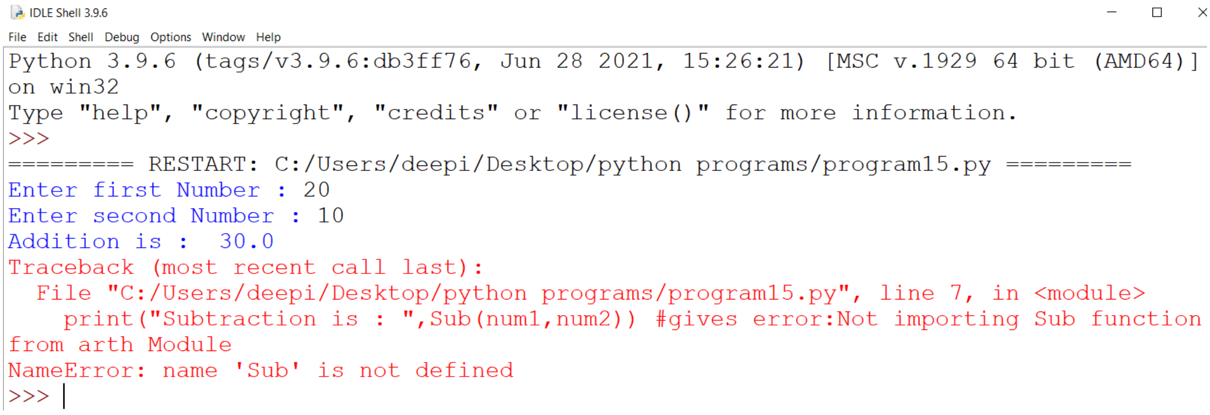
Experiment 15:

Aim: To write a python program to define a module and import a specific function in that module to another program.

Program:

```
''' Arithmetic Operations Module with Multiple functions'''

def Add(a,b):

    c=a+b

    return c

def Sub(a,b):

    c=a-b

    return c

def Mul(a,b):

    c=a*b

    return c

program to define a module and import a specific function in that

module to another program.'''

from arth import Add

num1=float(input("Enter first Number : "))

num2=float(input("Enter second Number : "))

print("Addition is : ",Add(num1,num2))

print("Subtraction is : ",Sub(num1,num2)) #gives error:Not importing Sub function from arth Module
```

Output:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:/Users/deepi/Desktop/python programs/program15.py =========
Enter first Number : 20
Enter second Number : 10
Addition is :  30.0
Traceback (most recent call last):
  File "C:/Users/deepi/Desktop/python programs/program15.py", line 7, in <module>
    print("Subtraction is : ",Sub(num1,num2)) #gives error:Not importing Sub function
from arth Module
NameError: name 'Sub' is not defined
>>> |
```

Experiment 16:

Aim: To write a script named **copyfile.py**. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.

Program:

File1.txt

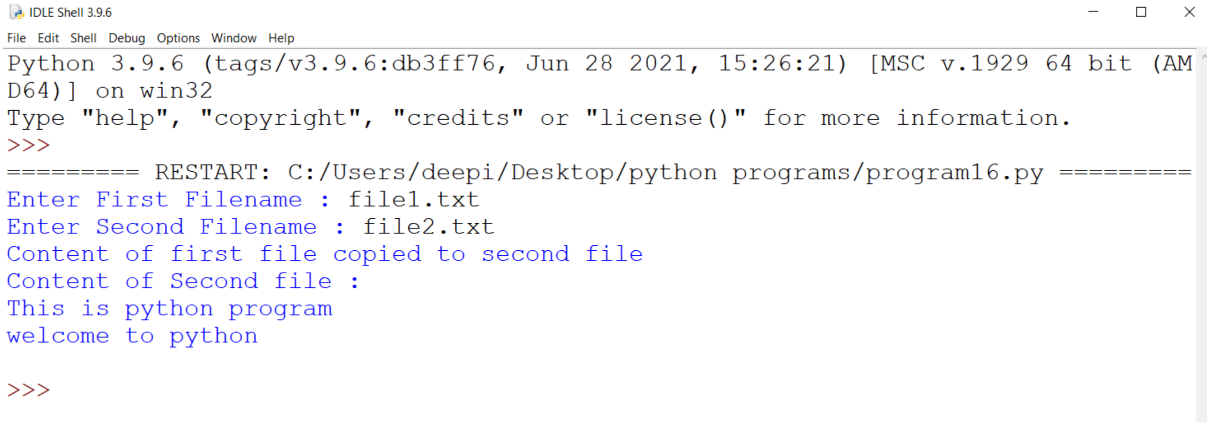
This is python program

welcome to python

```
'''Write a script named copyfile.py. This script should prompt the user for the names
of two text files. The contents of the first file should be input and written to the second
file'''
file1=input("Enter First Filename : ")
file2=input("Enter Second Filename : ")
# open file in read mode
fn1 = open(file1, 'r')

# open other file in write mode
fn2 = open(file2, 'w')

# read the content of the file line by line
cont = fn1.readlines()
#type(cont)
for i in range(0, len(cont)):
    fn2.write(cont[i])

# close the file
fn2.close()
print("Content of first file copied to second file ")

# open file in read mode
fn2 = open(file2, 'r')

# read the content of the file
cont1 = fn2.read()

# print the content of the file
print("Content of Second file :")
print(cont1)
```

```
# close all files
fn1.close()
fn2.close()
```

Output:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:/Users/deepi/Desktop/python programs/program16.py =========
Enter First Filename : file1.txt
Enter Second Filename : file2.txt
Content of first file copied to second file
Content of Second file :
This is python program
welcome to python

>>>
```

Experiment 17:

Aim: To write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.

Program:

File1.txt

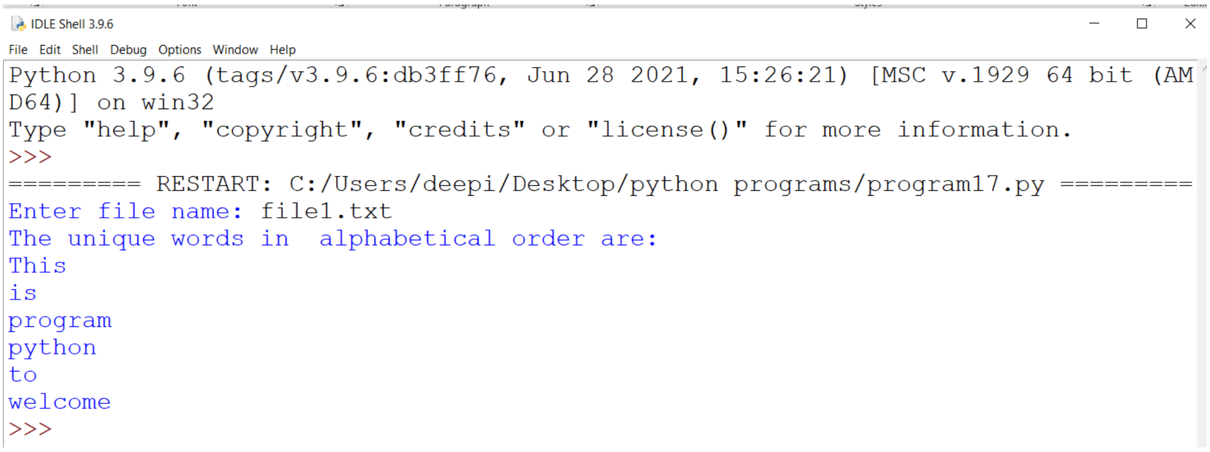
This is python program

welcome to python

```
'''Write a program that inputs a text file. The program should print all of the unique
words in the file in alphabetical order'''
fname = input("Enter file name: ")
fh = open(fname)
lst = []                    # list for the desired output
words=[];
for line in fh:                  # to read every line of file romeo.txt
    words += line.split()
words.sort()

# display the sorted words

print("The unique words in  alphabetical order are:")
for word in words:
    if word in lst:         # if element is repeated
        continue            # do nothing
    else:                   # else if element is not in the list
        lst.append(word)
        print(word)
```

```
#print(lst)
```

Output:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:/Users/deepi/Desktop/python programs/program17.py =========
Enter file name: file1.txt
The unique words in  alphabetical order are:
This
is
program
python
to
welcome
>>>
```

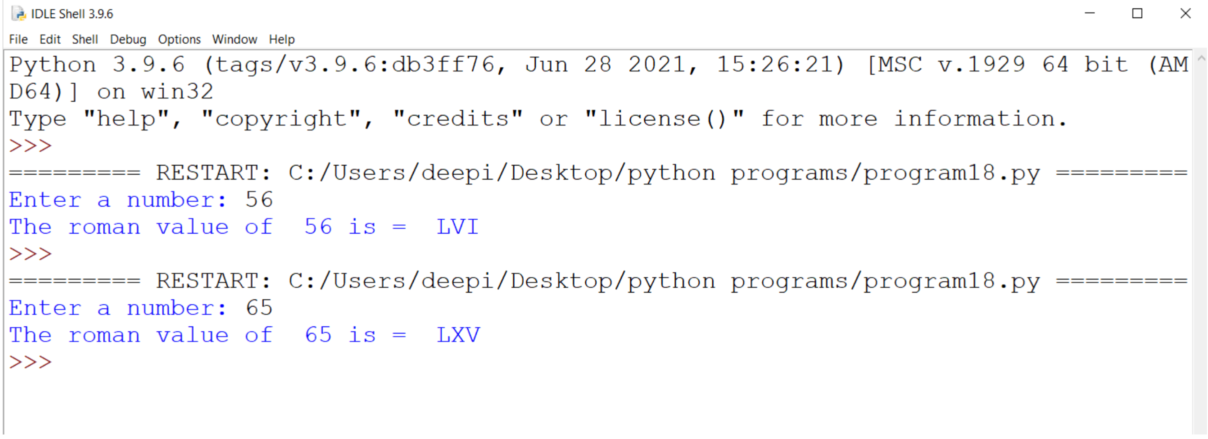
Experiment 18:

Aim: To write a Python class to convert an integer to a roman numeral

Program:

```
import sys

ROMAN_NUMERAL_TABLE = [ ("M", 1000), ("CM", 900), ("D", 500),
("CD", 400), ("C", 100), ("XC", 90),
("L", 50),      ("XL", 40), ("X", 10),
("IX", 9),      ("V", 5),        ("IV", 4),
("I", 1)
]
class Roman(object):

  def convert_to_roman(self, number):
      roman_numerals = []
      for numeral, value in ROMAN_NUMERAL_TABLE:
        while value <= number:
          number -= value
          roman_numerals.append(numeral)
      return ''.join(roman_numerals)



n=int(input("Enter a number: "))
r = Roman()
print("The roman value of ",n,"is = ", r.convert_to_roman(n))
```

Output:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
========= RESTART: C:/Users/deepi/Desktop/python programs/program18.py =========
Enter a number: 56
The roman value of  56 is =  LVI
>>> 
========= RESTART: C:/Users/deepi/Desktop/python programs/program18.py =========
Enter a number: 65
The roman value of  65 is =  LXV
>>> 
```

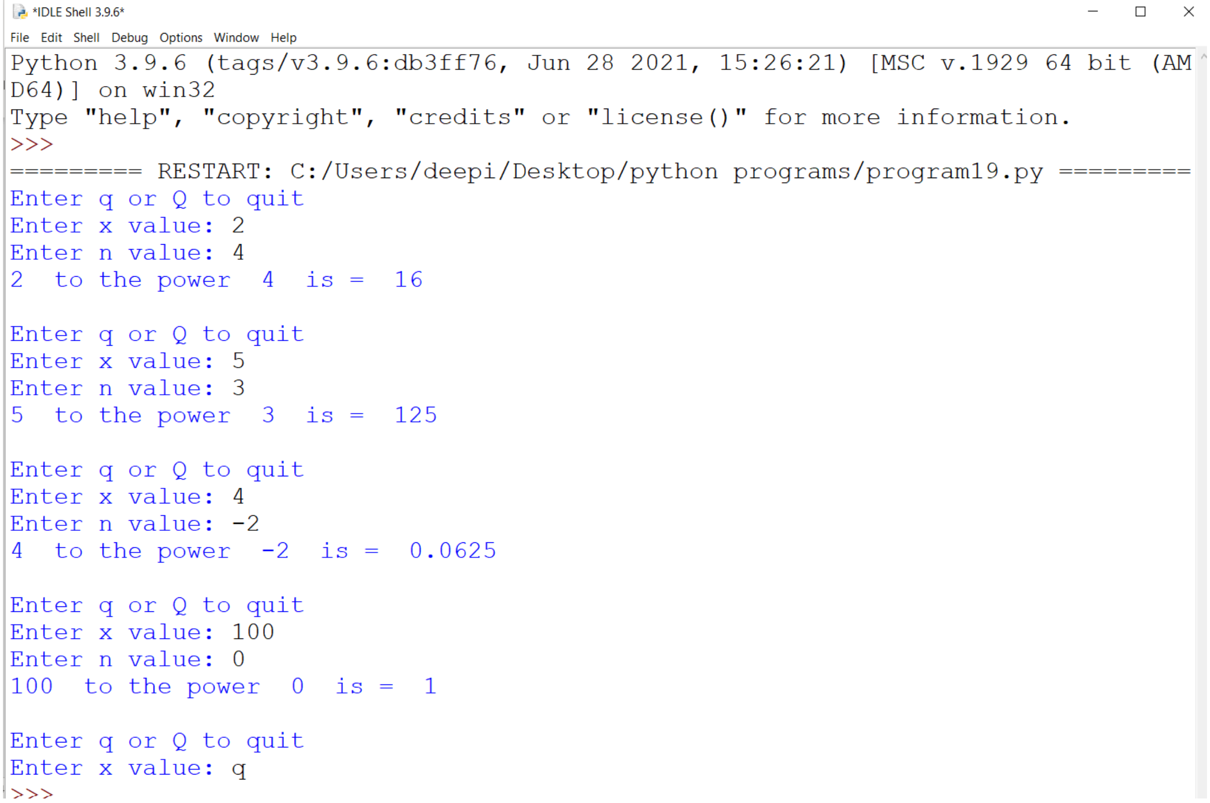
Experiment 19:

Aim: To write a Python class to implement pow(x, n)

Program:

```
    class py_solution:
      def pow(self, x, n):
        if x==0 or x==1 or n==1:
          return x
        if x==-1:
          if n%2 ==0:
            return 1
          else:
            return -1
        if n==0:
          return 1
        if n<0:
          return 1/self.pow(x,-n)
        val = self.pow(x,n//2)
        if n%2 ==0:
         return val*val
        return val*val*x

    while True:
      print("Enter q or Q to quit ")
      x=input("Enter x value: ")
      if (x=='q' or x=='Q'):
        break
      else:
        n=input("Enter n value: ")
        n=int(n)
        x=int(x)
        p=py_solution().pow(x,n)
        print(x," to the power ",n," is = ",p)
        print()
continue
```

<u>Output:</u>



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
========= RESTART: C:/Users/deepi/Desktop/python programs/program19.py =========
Enter q or Q to quit
Enter x value: 2
Enter n value: 4
2  to the power  4  is =  16

Enter q or Q to quit
Enter x value: 5
Enter n value: 3
5  to the power  3  is =  125

Enter q or Q to quit
Enter x value: 4
Enter n value: -2
4  to the power  -2  is =  0.0625

Enter q or Q to quit
Enter x value: 100
Enter n value: 0
100  to the power  0  is =  1

Enter q or Q to quit
Enter x value: q
>>> 
```

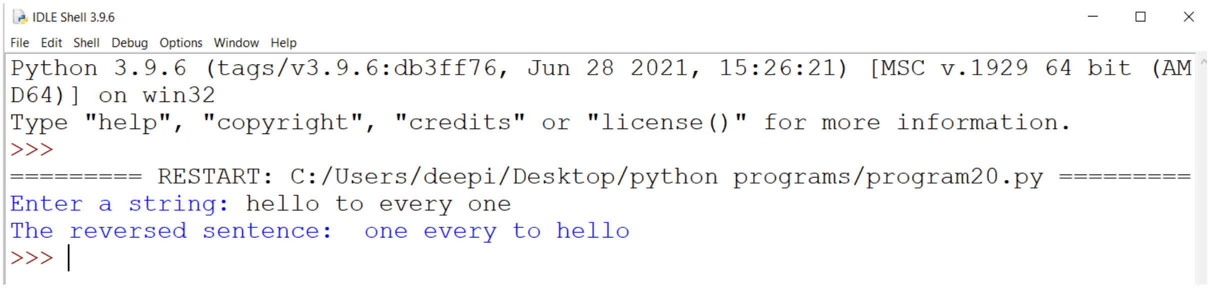
Experiment 20:

Aim: To write a Python class to reverse a string word by word.

Program:

```
          class py_solution:

             def reverse_words(self, s):
                lst = s.strip().split(" ")
                lst = lst[::-1]
                st=""
                for w in lst:
                   st = st + " " + w
                return st.strip()

          s=input("Enter a string: ")
          rs=py_solution()
          r = rs.reverse_words(s)
print("The reversed sentence: ",r)
```

Output:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:/Users/deepi/Desktop/python programs/program20.py =========
Enter a string: hello to every one
The reversed sentence:  one every to hello
>>> |
```