

DATA SCIENCE USING PYTHON

B.A / B.Com.(Computer Applications)

THIRD YEAR, SEMESTER-V

Lesson Writers:

Dr. K. Lavanya
Asst. Professor,
Dept. of Comp. Science & Engg.,
Acharya Nagarjuna University,

Mr. G.V. Suresh,
Assoc. Professor
Dept. of Comp. Science & Engg.,
Lakireddy Balireddy College of
Engineering, Mylavaram

Mrs. A. Sravani
Assoc. Professor,
Dept. of Information Technology,
Lakireddy Balireddy College of
Engineering, Mylavaram

Editor & Lesson Writer

Dr. U. Surya Kameswari
Asst. Professor,
Dept. of Comp. Science & Engg.,
Acharya Nagarjuna University

Director

Prof. V.VENKATESWARLU

M.A., (Soc), M.S.W.,M.Phil., Ph.D.

Centre for Distance Education
Acharya Nagarjuna University
Nagarjuna Nagar 522 510

Ph: 0863-2346222, 2346208
0863- 2346259 (Study Material)

Website www.anucde.info

E-mail: anucdedirector@gmail.com

B.A/ B.Com. (C.A.)

First Edition : 2021

No. of Copies :

© Acharya Nagarjuna University

**This book is exclusively prepared for the use of students of B.A/ B.Com C.A),
Centre for Distance Education, Acharya Nagarjuna University and this book is meant
for limited circulation only.**

Published by:

Prof. V.VENKATESWARLU,
Director
**Centre for Distance Education,
Acharya Nagarjuna University**

Printed at:

FOREWORD

Since its establishment in 1976, Acharya Nagarjuna University has been forging ahead in the path of progress and dynamism, offering a variety of courses and research contributions. I am extremely happy that by gaining 'A' grade from the NAAC in the year 2016, Acharya Nagarjuna University is offering educational opportunities at the UG, PG levels apart from research degrees to students from over 443 affiliated colleges spread over the two districts of Guntur and Prakasam.

The University has also started the Centre for Distance Education in 2003-04 with the aim of taking higher education to the door step of all the sectors of the society. The centre will be a great help to those who cannot join in colleges, those who cannot afford the exorbitant fees as regular students, and even to housewives desirous of pursuing higher studies. Acharya Nagarjuna University has started offering B.A., and B.Com courses at the Degree level and M.A., M.Com., M.Sc., M.B.A., and L.L.M., courses at the PG level from the academic year 2003-2004 onwards.

To facilitate easier understanding by students studying through the distance mode, these self-instruction materials have been prepared by eminent and experienced teachers. The lessons have been drafted with great care and expertise in the stipulated time by these teachers. Constructive ideas and scholarly suggestions are welcome from students and teachers involved respectively. Such ideas will be incorporated for the greater efficacy of this distance mode of education. For clarification of doubts and feedback, weekly classes and contact classes will be arranged at the UG and PG levels respectively.

It is my aim that students getting higher education through the Centre for Distance Education should improve their qualification, have better employment opportunities and in turn be part of country's progress. It is my fond desire that in the years to come, the Centre for Distance Education will go from strength to strength in the form of new courses and by catering to larger number of people. My congratulations to all the Directors, Academic Coordinators, Editors and Lesson- writers of the Centre who have helped in these endeavours.

*Prof. K. Gangadhara Rao
Vice-Chancellor, I/c.
Acharya Nagarjuna University*

PRACTICAL MANUAL

510BCE21-DATA SCIENCE USING PYTHON LAB

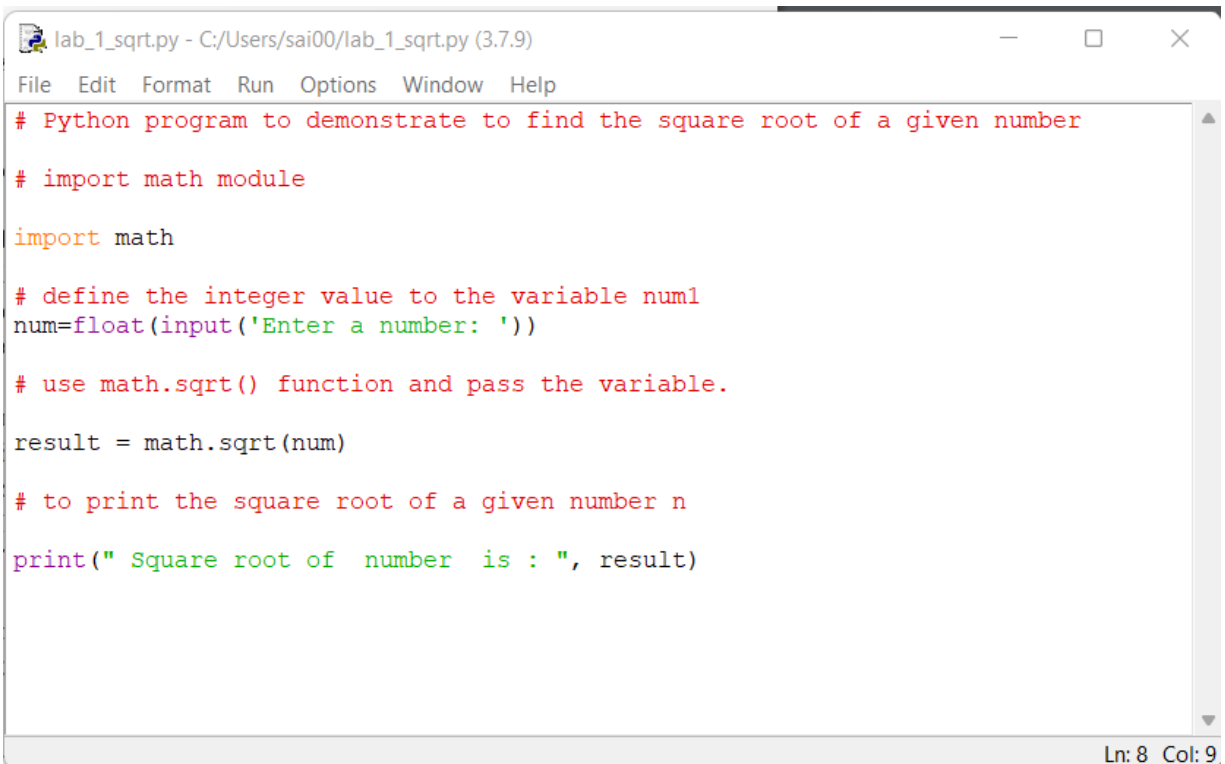
DSUP-P

OBJECTIVES:

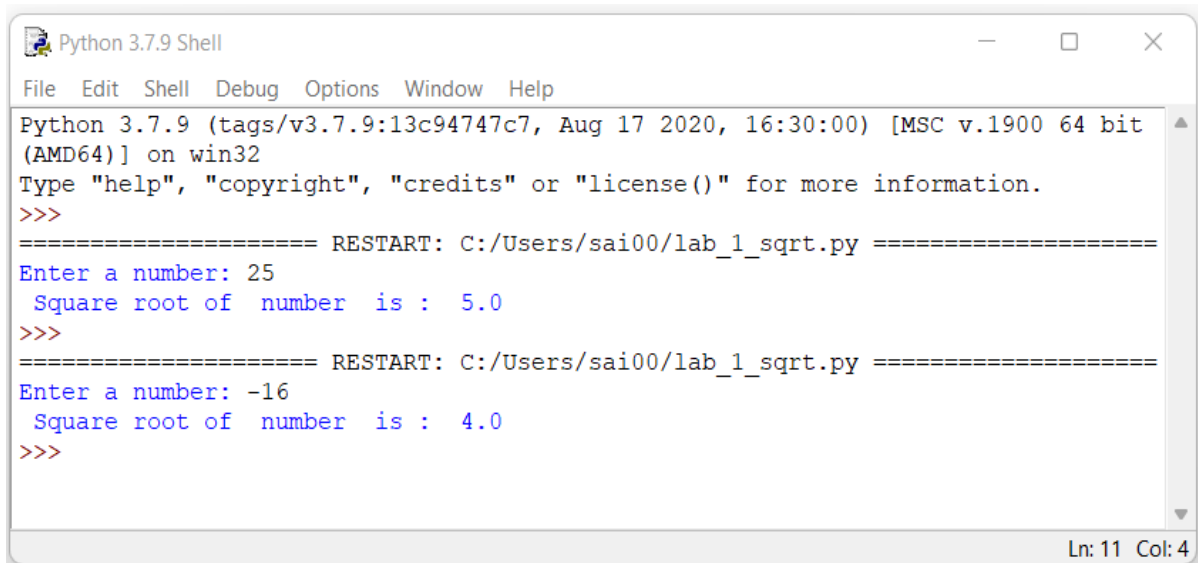
The objective of this lab is to master the python programming in data science field and to learn how to work with real time applications using python programming. These programs are widely used in most real-time scenario. After the end of lab, students will be able know the complete practical exposure on data science using python.

STRUCTURE

- 1) Python program to demonstrate to find the square root of a given number.
- 2) Python program to demonstrate swap two variables.
- 3) Python program to generate random number.
- 4) Python Program to Check if a Number is Odd or Even
- 5) Python Program to Find the largest among Three numbers.
- 6) Python program to Check Prime Number
- 7) Python Program to Display Multiplication Table
- 8) Python Program to Print Fibonacci Series
- 9) Python Program to Find the Sum of Natural Numbers
- 10) Python Program to Find Factorial of Number Using Recursion
- 11) Python program to work with string methods.
- 12) Python program to crate dictionary and print its content.
- 13) Python program to create class and objects.

1) Python program to demonstrate to find the square root of a given number.**CODE:**

```
lab_1_sqrt.py - C:/Users/sai00/lab_1_sqrt.py (3.7.9)
File Edit Format Run Options Window Help
# Python program to demonstrate to find the square root of a given number
# import math module
import math
# define the integer value to the variable num1
num=float(input('Enter a number: '))
# use math.sqrt() function and pass the variable.
result = math.sqrt(num)
# to print the square root of a given number n
print(" Square root of number is : ", result)
Ln: 8 Col: 9
```

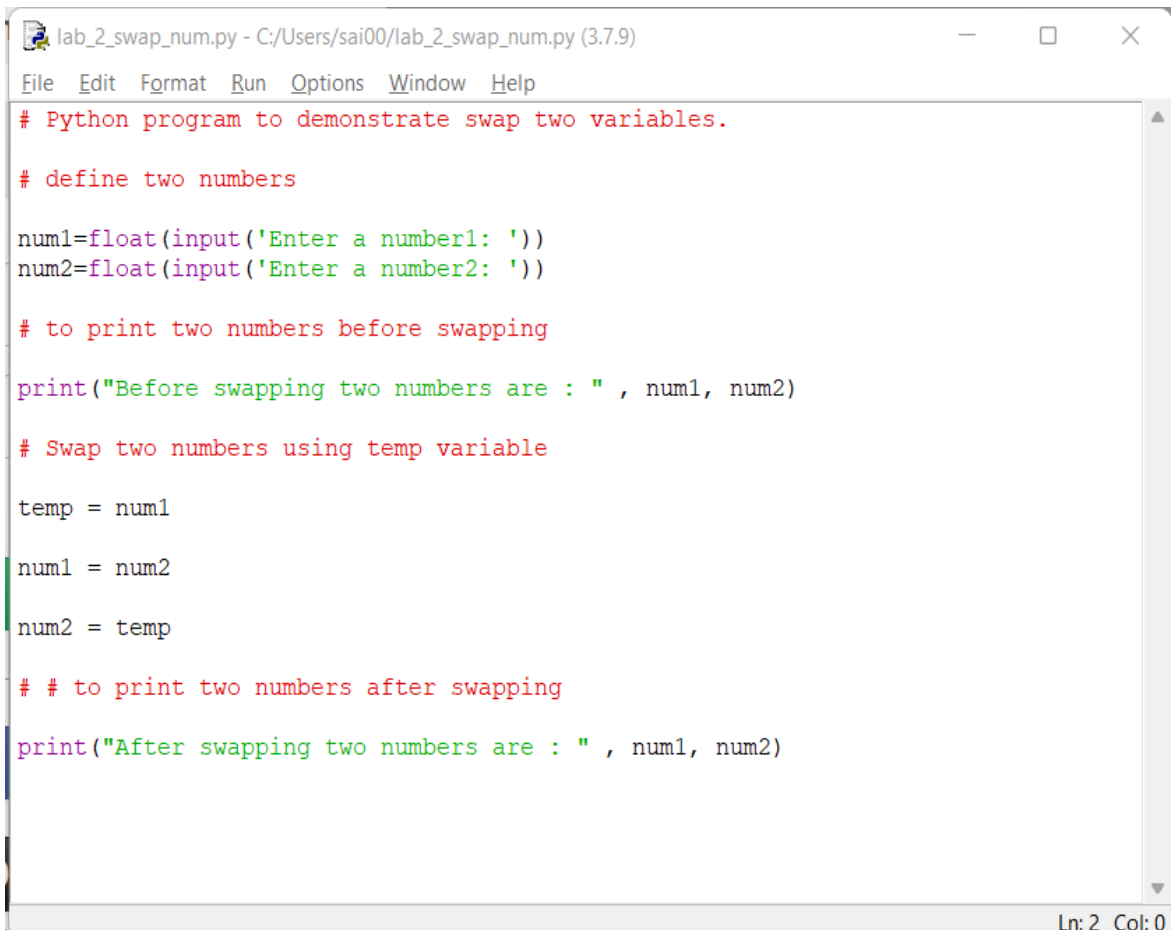
OUTPUT:

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/sai00/lab_1_sqrt.py =====
Enter a number: 25
Square root of number is : 5.0
>>>
===== RESTART: C:/Users/sai00/lab_1_sqrt.py =====
Enter a number: -16
Square root of number is : 4.0
>>>
Ln: 11 Col: 4
```

In the given program, the integer is stored in 'num' and the square root is calculated using the `math.sqrt()` function from the `math` module. For example, given `num = 25`, the program completed properly and returned "Square root of number is: 5.0". Similarly, for `num = -16`, the output is "Square root of number is: 4.0". This software works with all positive real numbers. However, it does not work for negative or complex values.

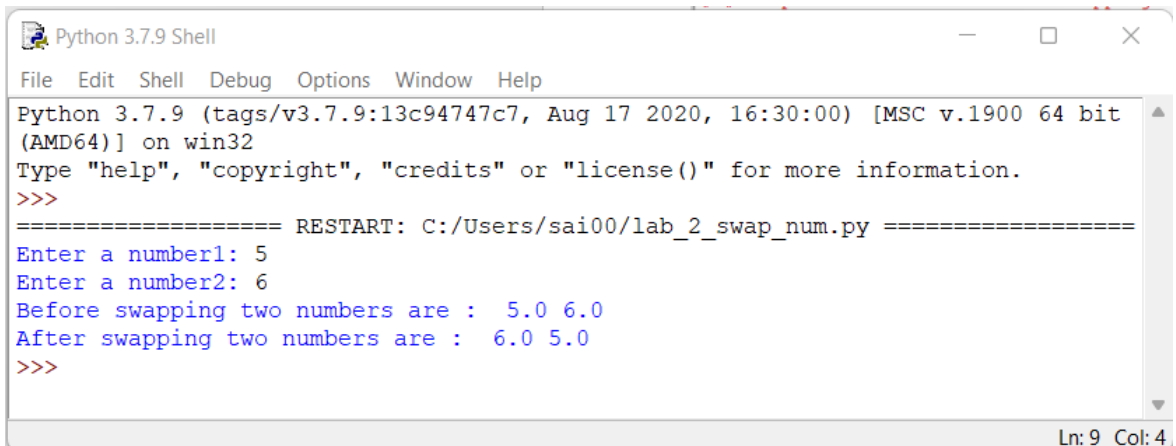
2) Python program to demonstrate swap two variables.

CODE:



```
lab_2_swap_num.py - C:/Users/sai00/lab_2_swap_num.py (3.7.9)
File Edit Format Run Options Window Help
# Python program to demonstrate swap two variables.
# define two numbers
num1=float(input('Enter a number1: '))
num2=float(input('Enter a number2: '))
# to print two numbers before swapping
print("Before swapping two numbers are : " , num1, num2)
# Swap two numbers using temp variable
temp = num1
num1 = num2
num2 = temp
# # to print two numbers after swapping
print("After swapping two numbers are : " , num1, num2)
Ln: 2 Col: 0
```

OUTPUT:



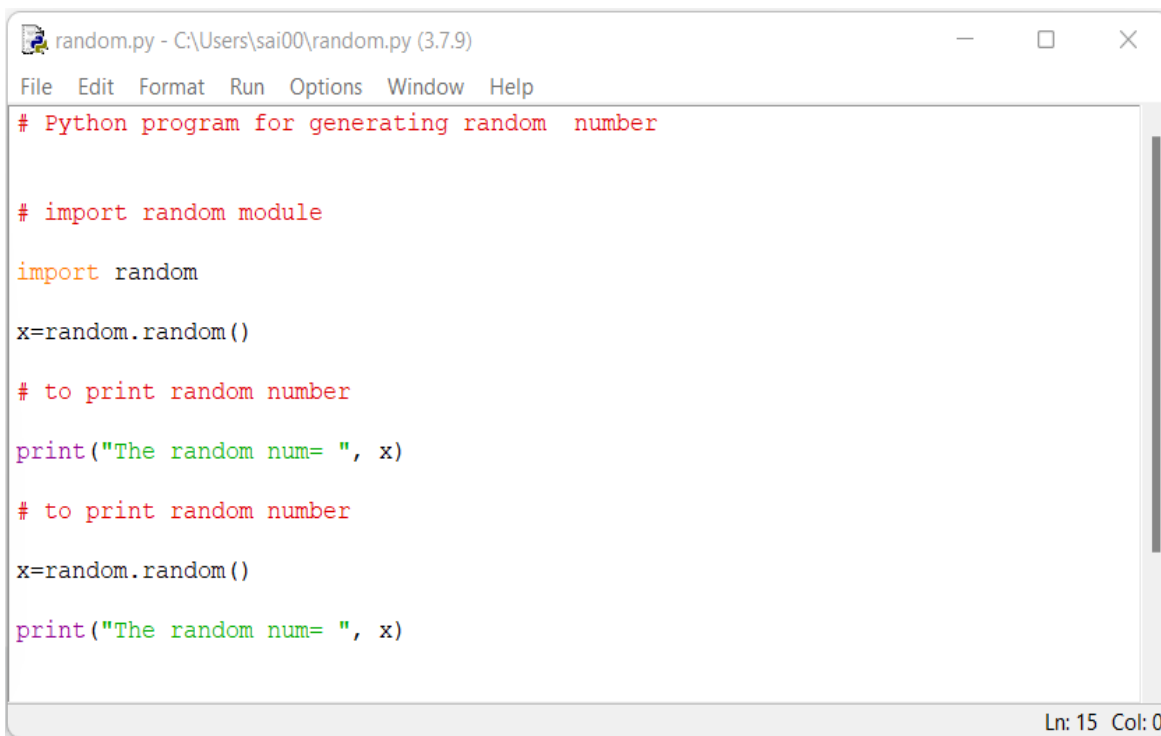
```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/sai00/lab_2_swap_num.py =====
Enter a number1: 5
Enter a number2: 6
Before swapping two numbers are : 5.0 6.0
After swapping two numbers are : 6.0 5.0
>>>
Ln: 9 Col: 4
```

above program accepts two numbers as input, 'num1' and 'num2'. Later, two numbers are displayed before attempting the switching process. The swapping operation between numbers 1

and 2 was then conducted using the third variable, 'temp'. Finally, the two numerals were presented following the swapping operation. The whole result is displayed in the output.

3) Python program to generate random number.

CODE:



```
random.py - C:\Users\sai00\random.py (3.7.9)
File Edit Format Run Options Window Help
# Python program for generating random number

# import random module
import random

x=random.random()

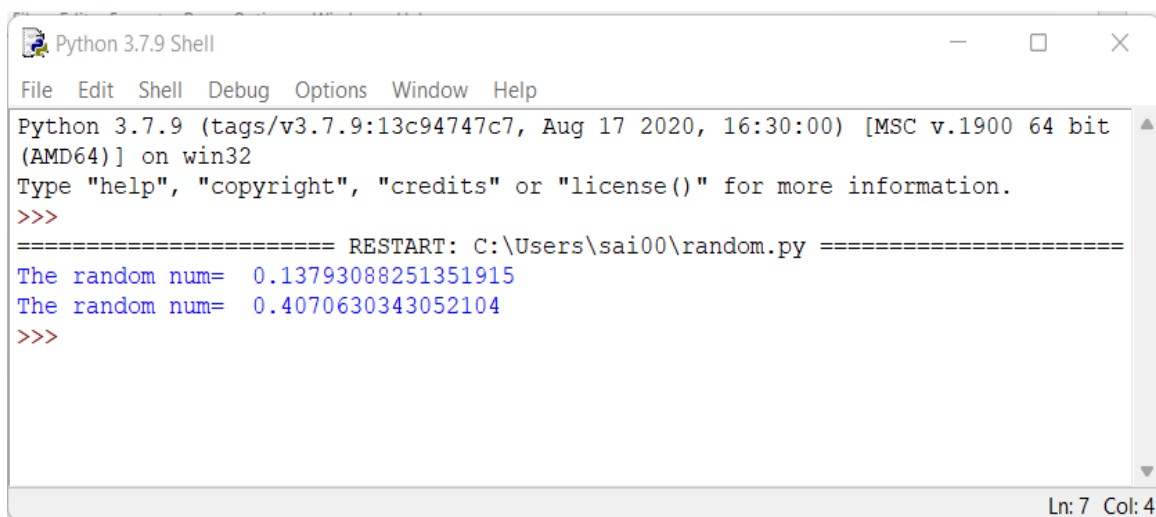
# to print random number
print("The random num= ", x)

# to print random number
x=random.random()

print("The random num= ", x)

Ln: 15 Col: 0
```

OUTPUT:



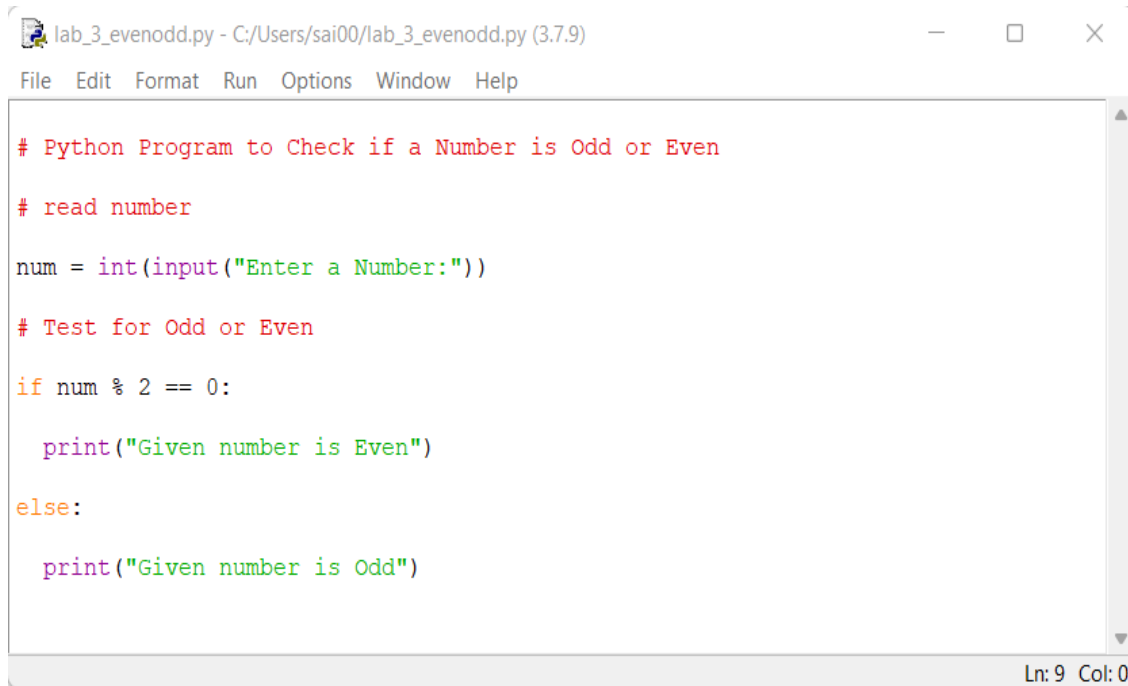
```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\sai00\random.py =====
The random num= 0.13793088251351915
The random num= 0.4070630343052104
>>>

Ln: 7 Col: 4
```

The above program generates a random integer using the random module's random() function. First, the program imports the random module and then calls the random() function, which generates a random float integer between 0.0 and 1.0. In the given code, the 'x' variable is assigned a distinct random number each time. The final produced random is printed as 'The

random num= 0.13793088251351915' and 'The random num= 0.4070630343052104' on the screen and presented in the output.

4) Python Program to Check if a Number is Odd or Even



```
lab_3_evenodd.py - C:/Users/sai00/lab_3_evenodd.py (3.7.9)
File Edit Format Run Options Window Help

# Python Program to Check if a Number is Odd or Even

# read number

num = int(input("Enter a Number:"))

# Test for Odd or Even

if num % 2 == 0:

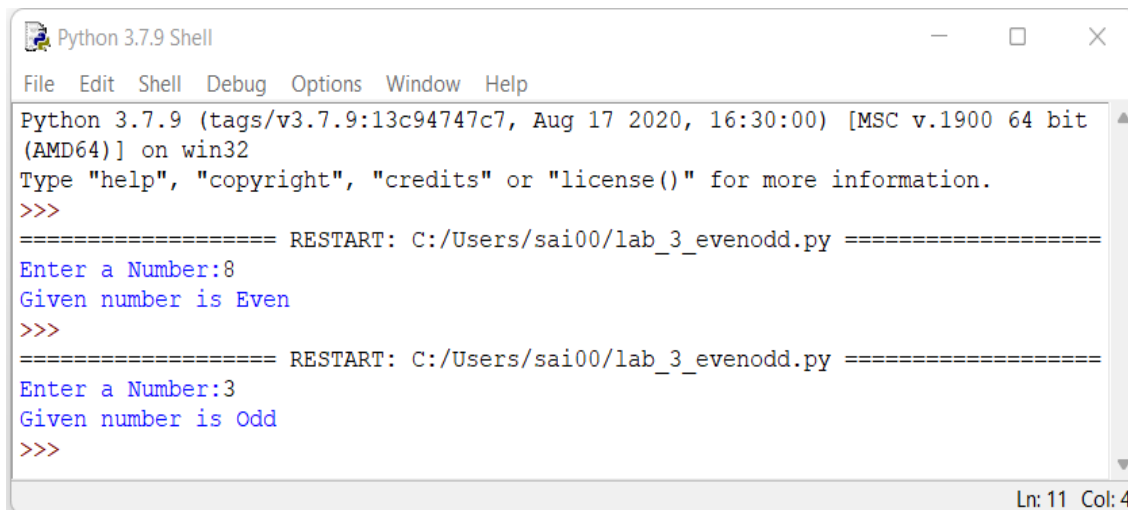
    print("Given number is Even")

else:

    print("Given number is Odd")

Ln: 9 Col: 0
```

OUTPUT:



```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help

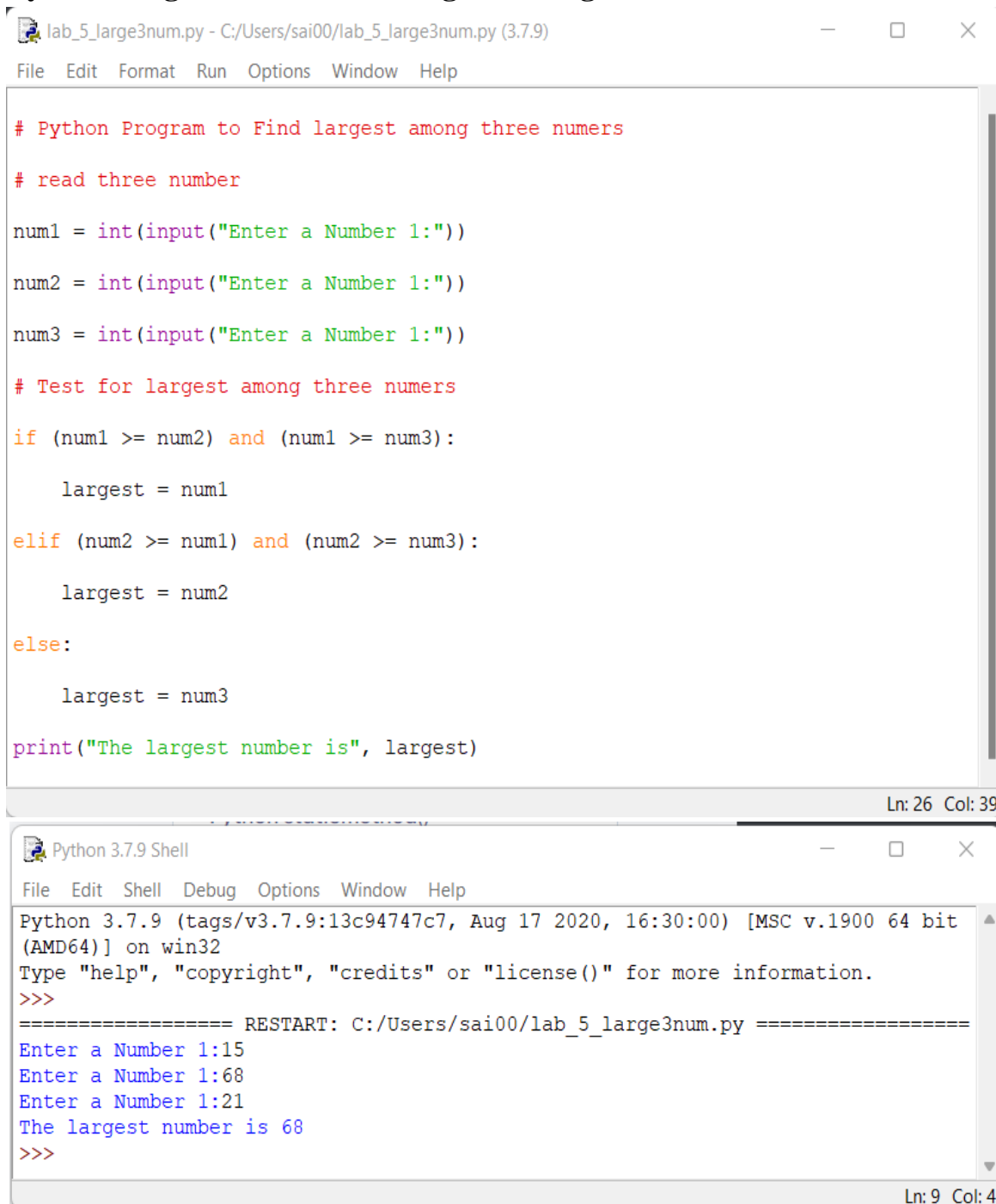
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/sai00/lab_3_evenodd.py =====
Enter a Number:8
Given number is Even
>>>
===== RESTART: C:/Users/sai00/lab_3_evenodd.py =====
Enter a Number:3
Given number is Odd
>>>

Ln: 11 Col: 4
```

In Python, given an integer, the goal is to determine if it is even or odd. To do so, we determine if it is divisible by two or not. If true, then it is even; otherwise, it is odd. The first number entered into the variable num is 8, which is then examined to see if it is divisible by two. If it is, the true condition is executed, and the printed message "Given number is Even" appears.

Similarly, when num=3, the condition failed, and the message "Given number is even" was printed.

5) Python Program to Find the largest among Three numbers.



```
lab_5_large3num.py - C:/Users/sai00/lab_5_large3num.py (3.7.9)
File Edit Format Run Options Window Help

# Python Program to Find largest among three numbers
# read three number
num1 = int(input("Enter a Number 1:"))
num2 = int(input("Enter a Number 1:"))
num3 = int(input("Enter a Number 1:"))
# Test for largest among three numbers
if (num1 >= num2) and (num1 >= num3):
    largest = num1
elif (num2 >= num1) and (num2 >= num3):
    largest = num2
else:
    largest = num3
print("The largest number is", largest)

Ln: 26 Col: 39

Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/sai00/lab_5_large3num.py =====
Enter a Number 1:15
Enter a Number 1:68
Enter a Number 1:21
The largest number is 68
>>>

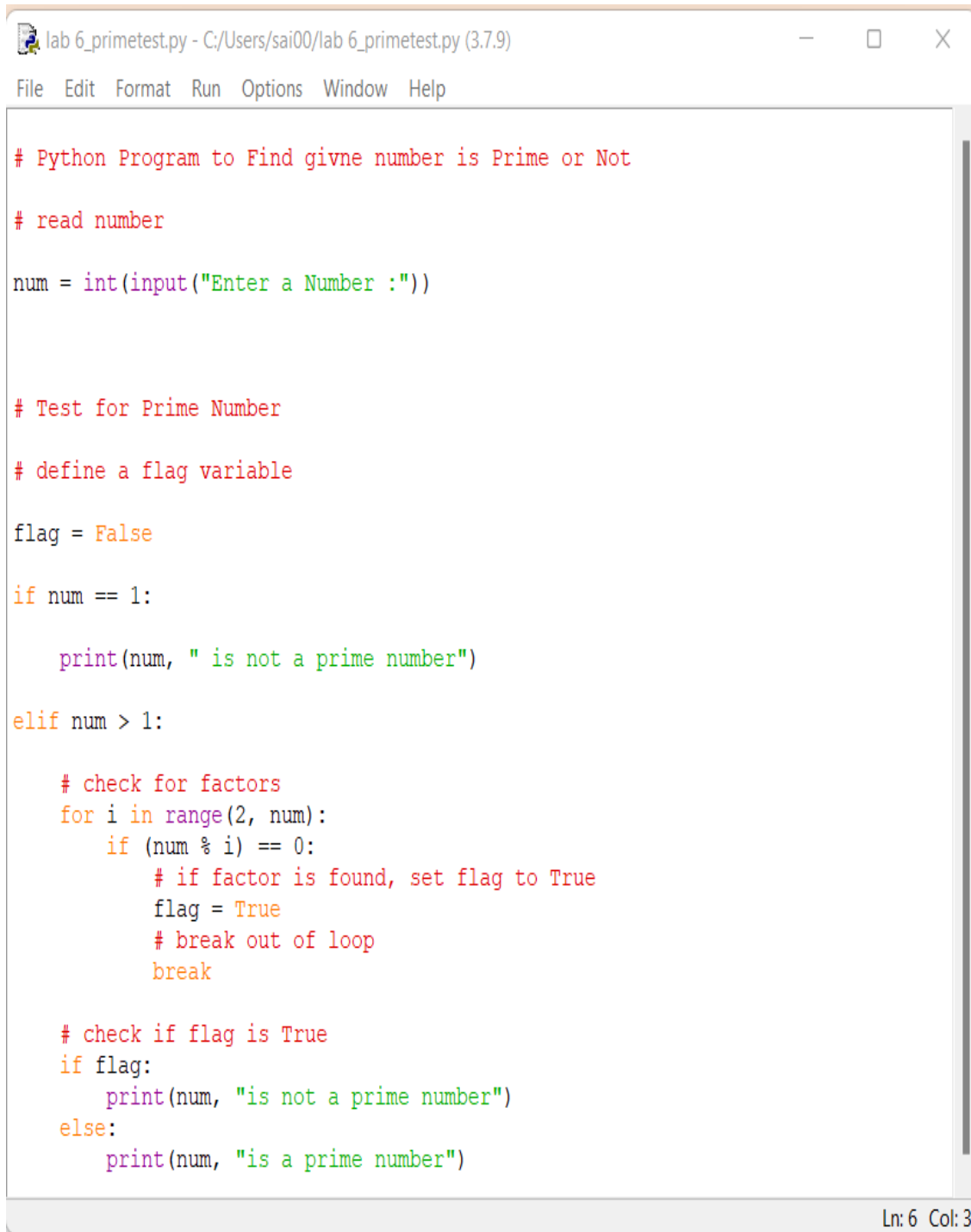
Ln: 9 Col: 4
```

In the program below, the three numbers are stored in num1, num2, and num3, accordingly. We utilized the if...elif...else ladder to determine the largest of the three and show it. First, enter num1=15, num2=68, and num3=21, then use conditional statements to ensure that 15 is not more

than num2 or num3. As a result, the elif statement condition is true, and num2 is set to the greatest value. The last largest number is displayed as "The largest number is" 68.

6) Python program to Check Prime Number

CODE:



```
lab 6_primetest.py - C:/Users/sai00/lab 6_primetest.py (3.7.9)
File Edit Format Run Options Window Help

# Python Program to Find givne number is Prime or Not

# read number

num = int(input("Enter a Number :"))

# Test for Prime Number

# define a flag variable

flag = False

if num == 1:

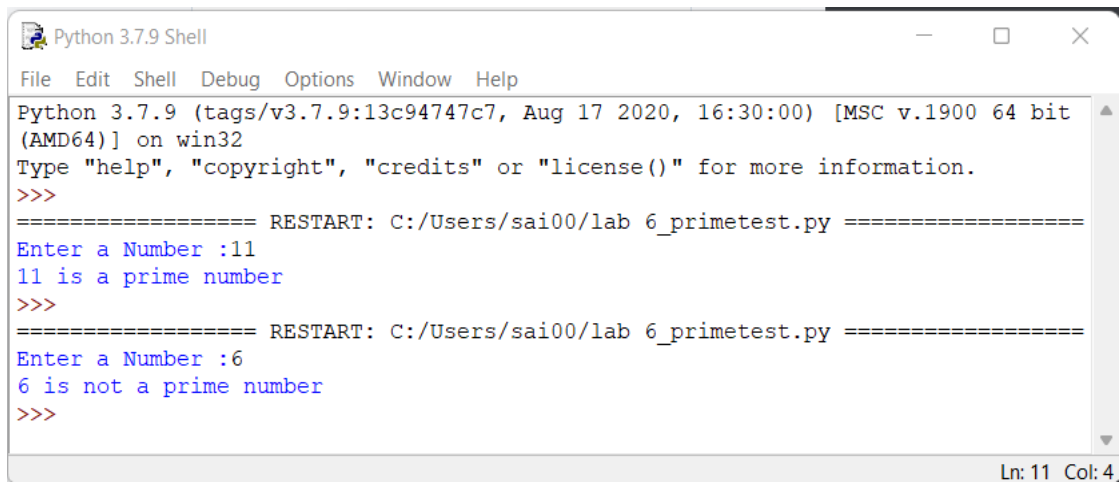
    print(num, " is not a prime number")

elif num > 1:

    # check for factors
    for i in range(2, num):
        if (num % i) == 0:
            # if factor is found, set flag to True
            flag = True
            # break out of loop
            break

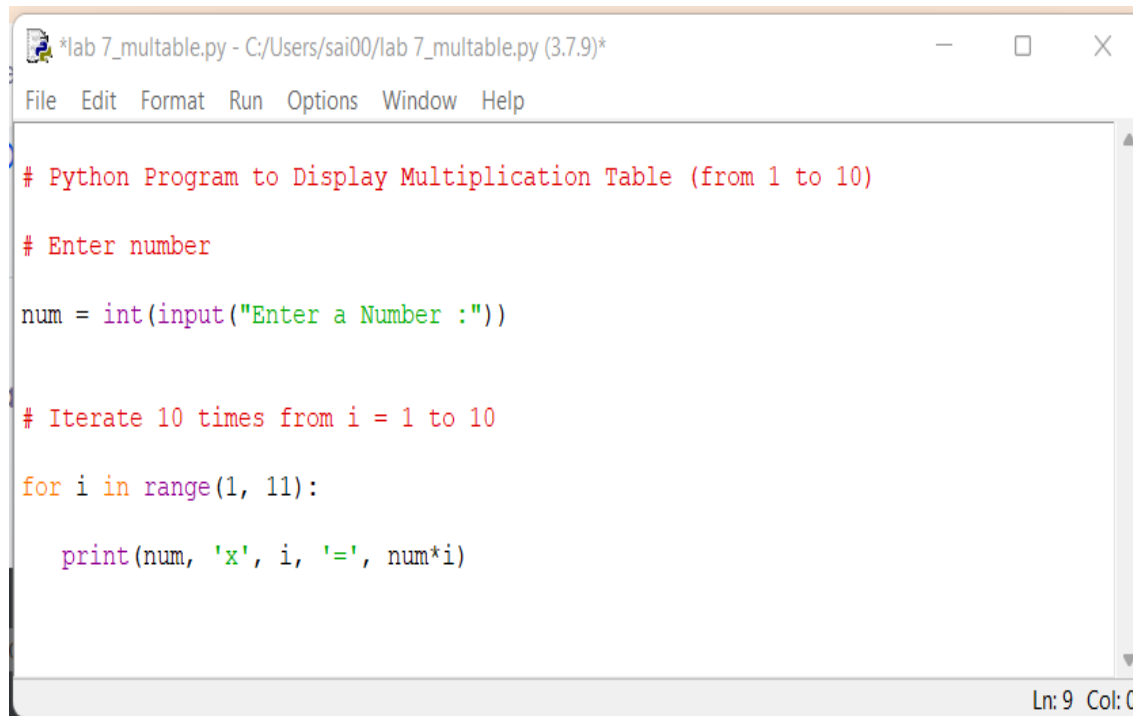
    # check if flag is True
    if flag:
        print(num, "is not a prime number")
    else:
        print(num, "is a prime number")

Ln: 6 Col: 3
```

OUTPUT:

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/sai00/lab 6_primetest.py =====
Enter a Number :11
11 is a prime number
>>>
===== RESTART: C:/Users/sai00/lab 6_primetest.py =====
Enter a Number :6
6 is not a prime number
>>>
```

In this program, we checked whether num is prime or not. Numbers less than or equal to one are not primes. As a result, we will only proceed if the number is more than one. We determine whether num is exactly divisible by any number from 2 to num - 1. If we find a factor inside that range, the integer is not prime, thus we set the flag to True and exit the loop. Outside the loop, we check whether the flag is True or False. If it is true, then num is not a prime number. If it is false, then num is a prime number.

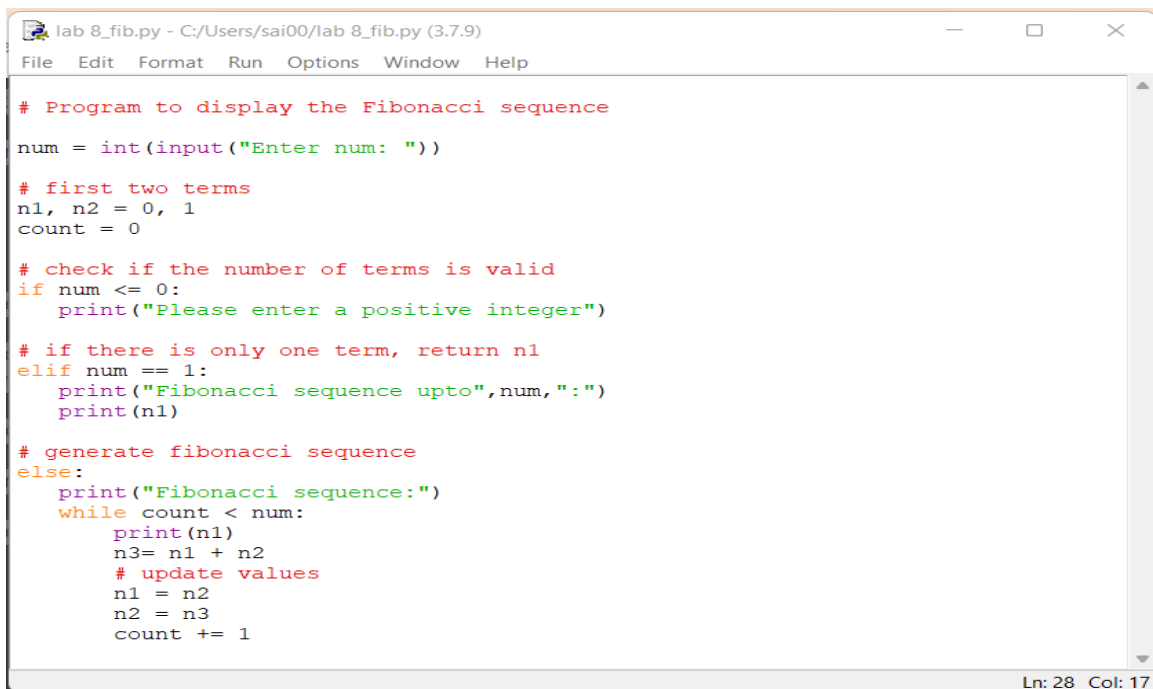
Python Program to Display Multiplication Table**CODE:**

```
*lab 7_multable.py - C:/Users/sai00/lab 7_multable.py (3.7.9)*
File Edit Format Run Options Window Help
# Python Program to Display Multiplication Table (from 1 to 10)
# Enter number
num = int(input("Enter a Number :"))
# Iterate 10 times from i = 1 to 10
for i in range(1, 11):
    print(num, 'x', i, '=', num*i)
```

OUTPUT:

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/sai00/lab 7_multable.py =====
Enter a Number :2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
>>>
```

We used the for loop and the range() function to iterate ten times. The arguments for the range() function are (1, 11). Meaning, greater than or equal to 1 but less than 11. We have shown the multiplication table for variable num (which is 02 in our example).

Python Program to Print Fibonacci Series**CODE:**

```
lab 8_fib.py - C:/Users/sai00/lab 8_fib.py (3.7.9)
File Edit Format Run Options Window Help

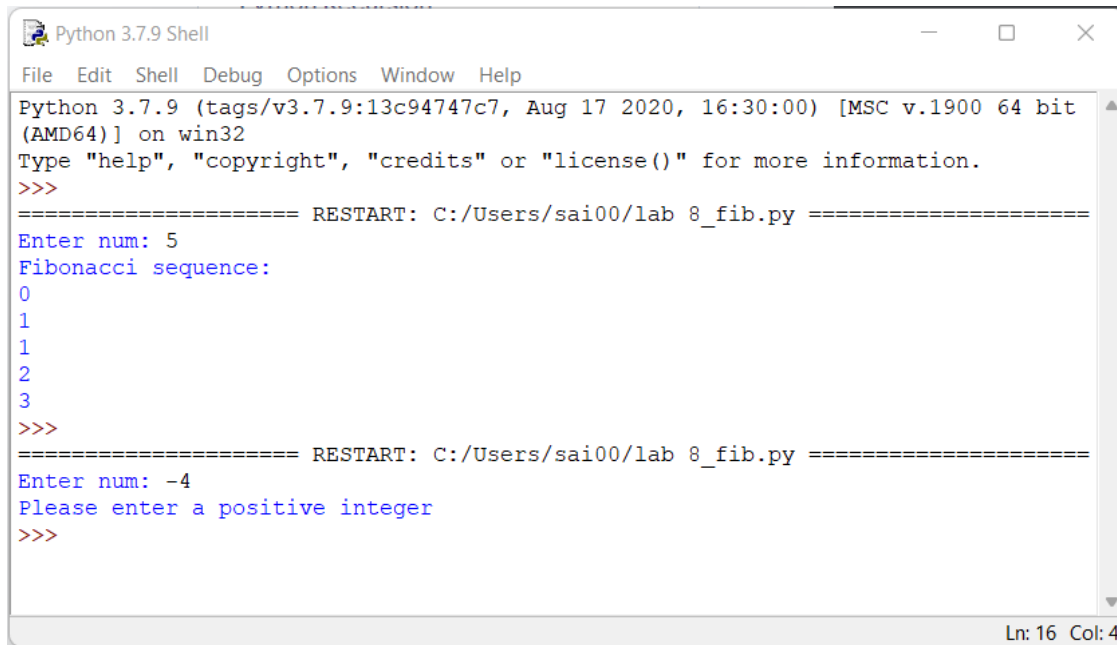
# Program to display the Fibonacci sequence
num = int(input("Enter num: "))

# first two terms
n1, n2 = 0, 1
count = 0

# check if the number of terms is valid
if num <= 0:
    print("Please enter a positive integer")

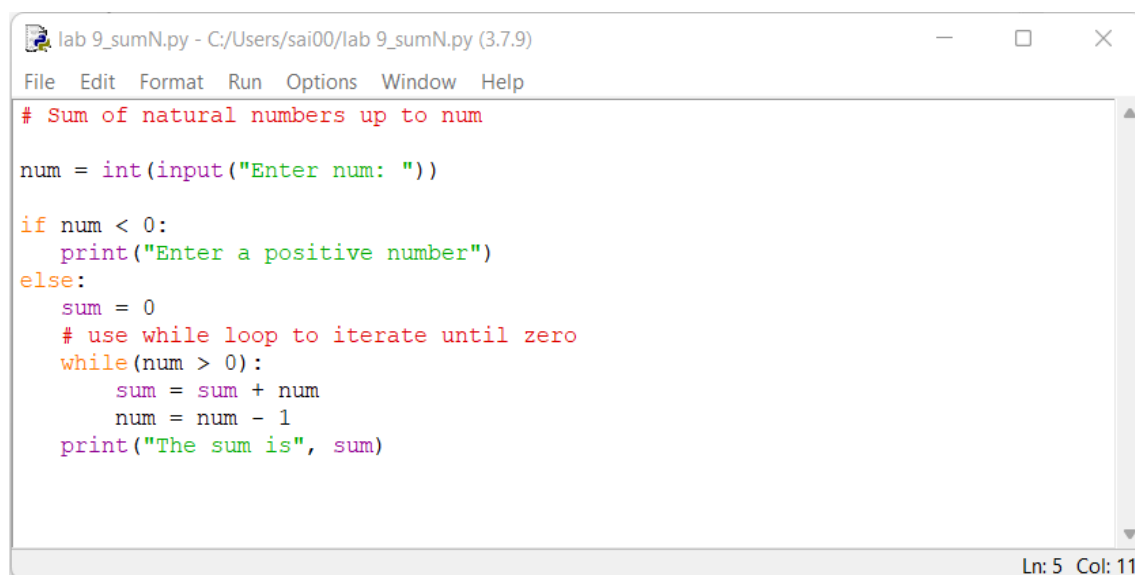
# if there is only one term, return n1
elif num == 1:
    print("Fibonacci sequence upto", num, ":")
    print(n1)

# generate fibonacci sequence
else:
    print("Fibonacci sequence:")
    while count < num:
        print(n1)
        n3= n1 + n2
        # update values
        n1 = n2
        n2 = n3
        count += 1
```

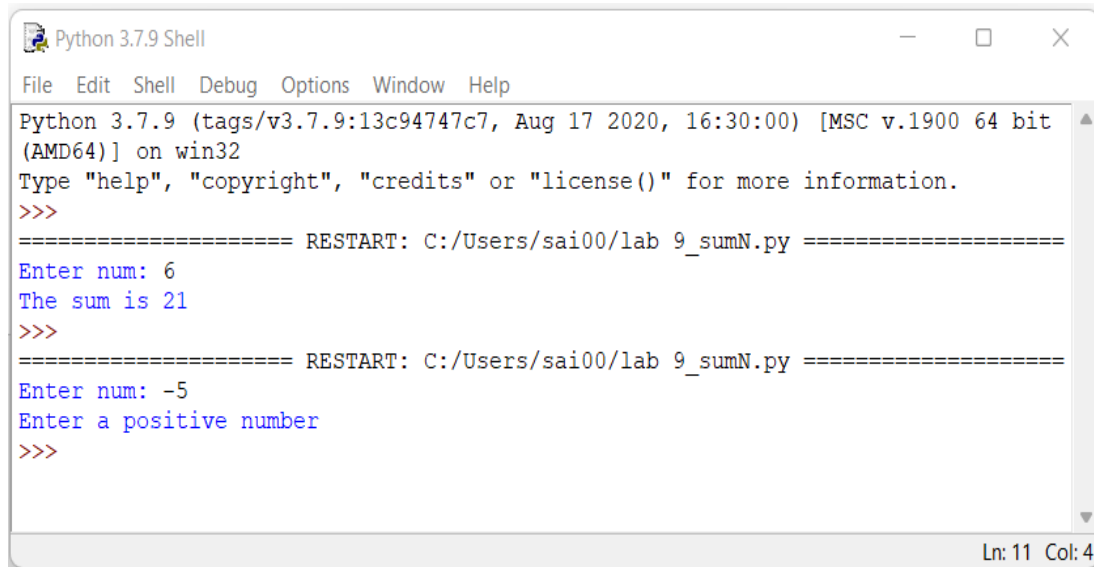
OUTPUT:

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/sai00/lab 8_fib.py =====
Enter num: 5
Fibonacci sequence:
0
1
1
2
3
>>>
===== RESTART: C:/Users/sai00/lab 8_fib.py =====
Enter num: -4
Please enter a positive integer
>>>
```

We store the number of terms in num. We set the first term to zero and the second term to one. If there are more than two terms in the sequence, we utilize a while loop to find the next term by adding the two terms that came before it. We then change the variables (update it) and repeat the process.

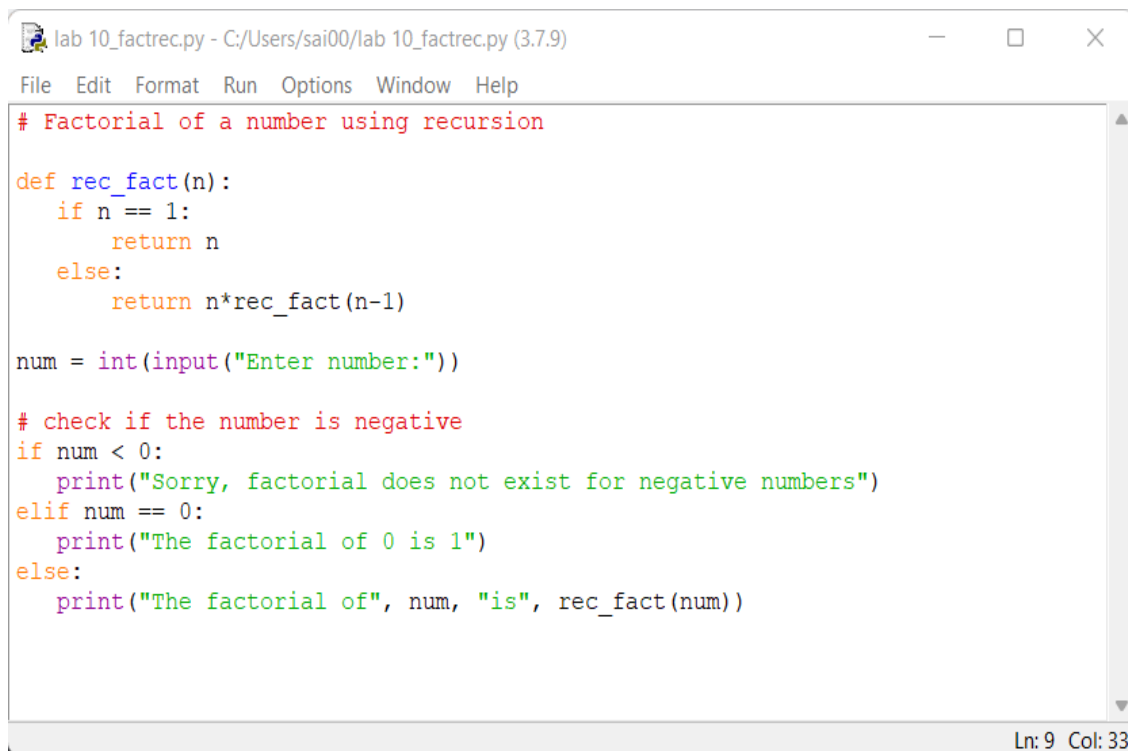
7) Python Program to Find the Sum of Natural Numbers**CODE:**

```
lab 9_sumN.py - C:/Users/sai00/lab 9_sumN.py (3.7.9)
File Edit Format Run Options Window Help
# Sum of natural numbers up to num
num = int(input("Enter num: "))
if num < 0:
    print("Enter a positive number")
else:
    sum = 0
    # use while loop to iterate until zero
    while(num > 0):
        sum = sum + num
        num = num - 1
    print("The sum is", sum)
```

OUTPUT:

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/sai00/lab 9_sumN.py =====
Enter num: 6
The sum is 21
>>>
===== RESTART: C:/Users/sai00/lab 9_sumN.py =====
Enter num: -5
Enter a positive number
>>>
```

Initially, the sum is set to zero. And the number is saved in the variable num. Then, we used the while loop to iterate until the number reached zero. In each iteration of the loop, we add the num to the sum and lower the value of num by one.

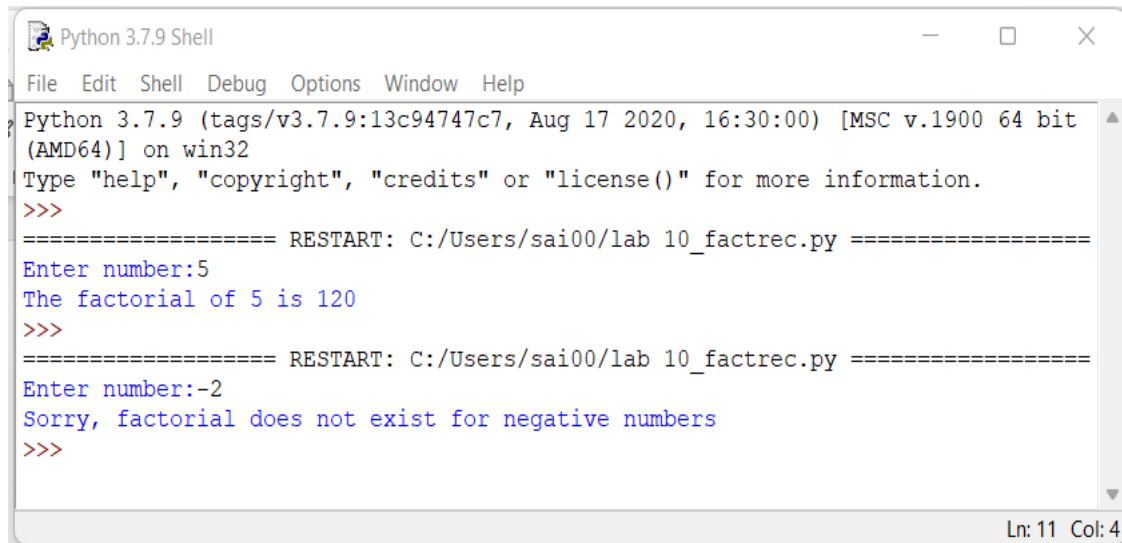
Python Program to Find Factorial of Number Using Recursion**CODE:**

```
lab 10_factrec.py - C:/Users/sai00/lab 10_factrec.py (3.7.9)
File Edit Format Run Options Window Help
# Factorial of a number using recursion

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

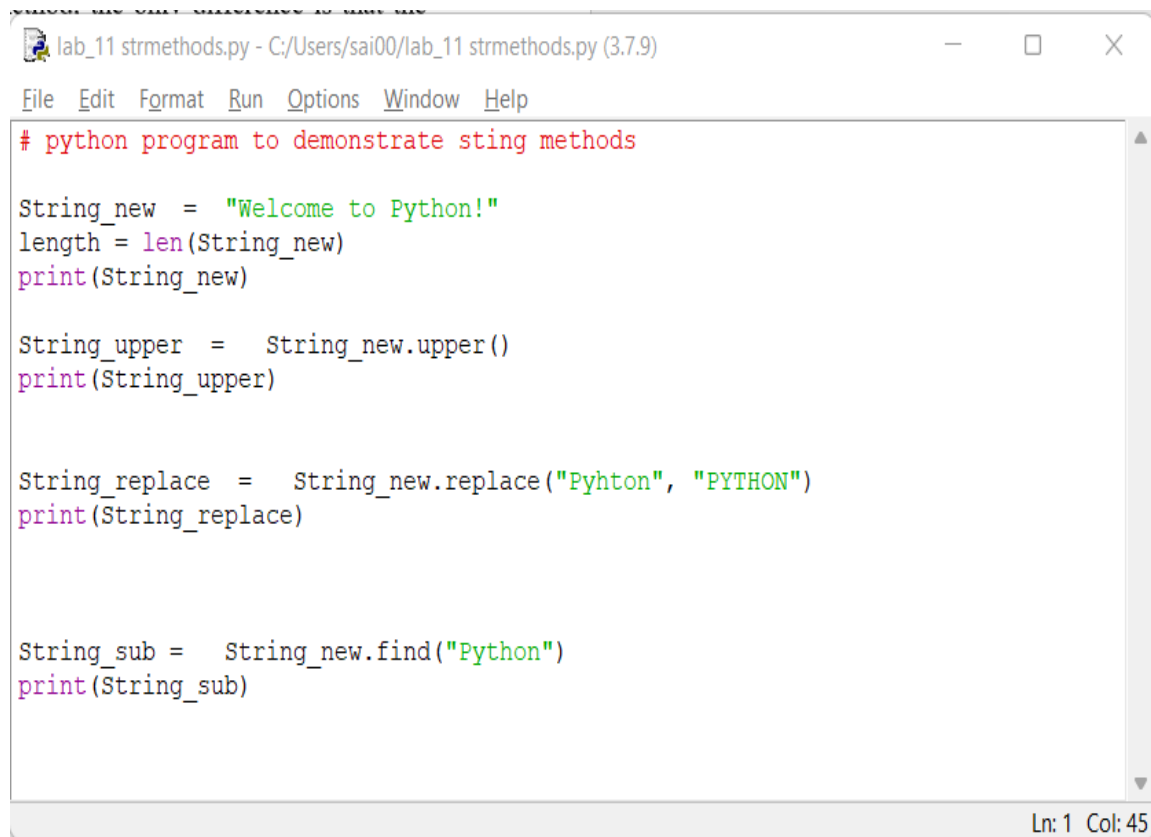
num = int(input("Enter number:"))

# 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", rec_fact(num))
```

OUTPUT:

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/sai00/lab 10_factrec.py =====
Enter number:5
The factorial of 5 is 120
>>>
===== RESTART: C:/Users/sai00/lab 10_factrec.py =====
Enter number:-2
Sorry, factorial does not exist for negative numbers
>>>
Ln: 11 Col: 4
```

The number will be stored in the num variable here. The factorial of the number is computed by passing the number to the recur_factorial() function, which then returns the result.

8) Python program to work with string methods.**CODE:**

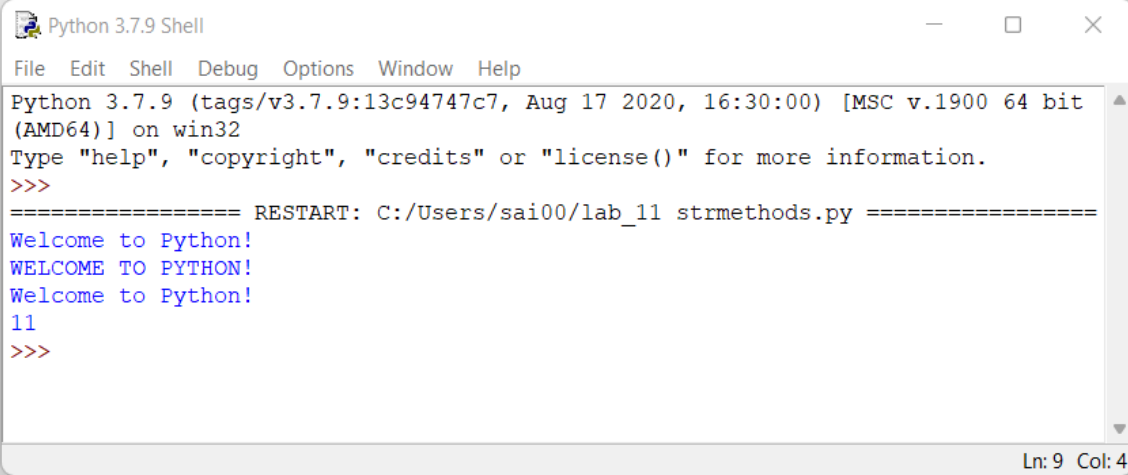
```
lab_11 strmethods.py - C:/Users/sai00/lab_11 strmethods.py (3.7.9)
File Edit Format Run Options Window Help
# python program to demonstrate sting methods

String_new = "Welcome to Python!"
length = len(String_new)
print(String_new)

String_upper = String_new.upper()
print(String_upper)

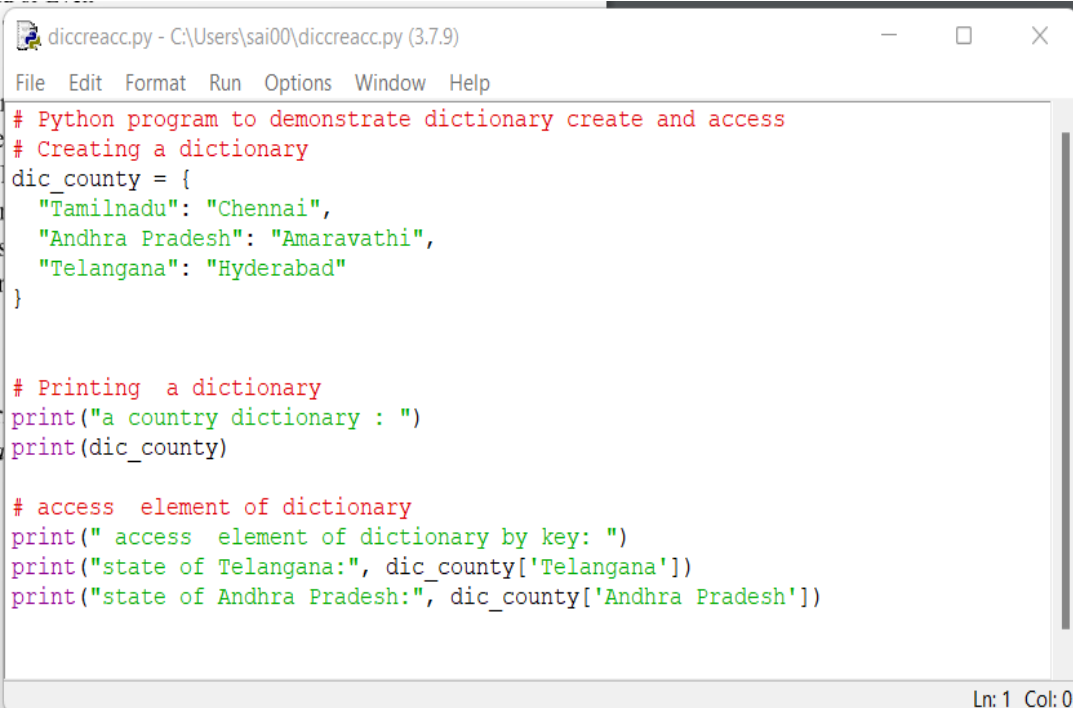
String_replace = String_new.replace("Pyhton", "PYTHON")
print(String_replace)

String_sub = String_new.find("Python")
print(String_sub)
Ln: 1 Col: 45
```

OUTPUT:

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c9474c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/sai00/lab_11 strmethods.py =====
Welcome to Python!
WELCOME TO PYTHON!
Welcome to Python!
11
>>>
Ln: 9 Col: 4
```

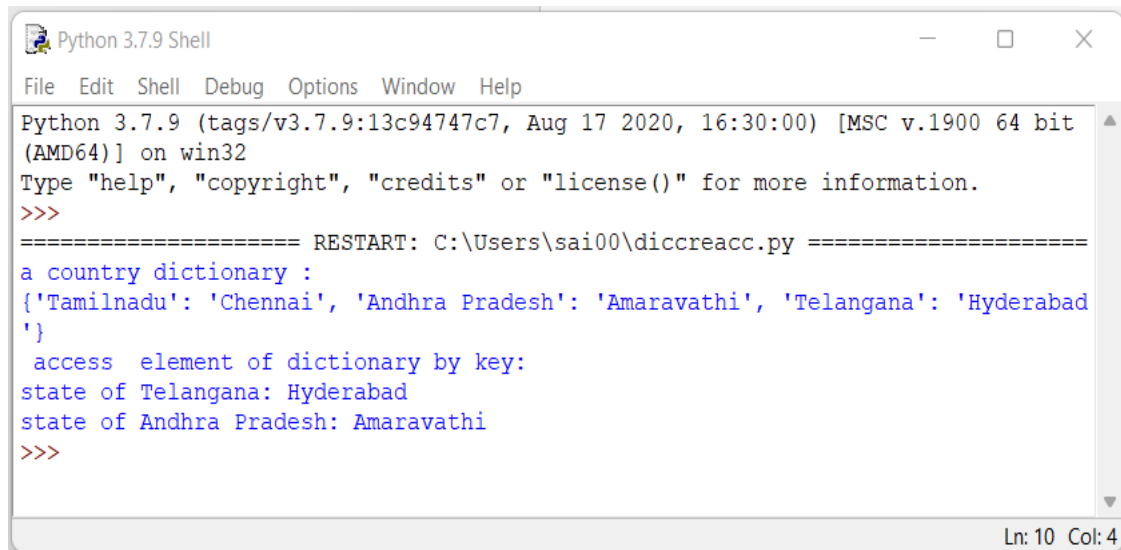
The number will be stored string in variable `String_new`. Four methods applied over the string `len()`, `upper()`, `replace()` and `find()` respectively. The result of each method is displayed in the output. The `len()` will return number of characters is the given sting, once we applied `len()` function to the `String_new` it return value as 17. Similarly, `upper()` method return sting into upper case and returns result as “ WELCOME TO PYTHON!” . The `replace()` method replaces old sting to new sting . The `find()` method returns the index where sub string is find. Here, we passed sub string as “Python” and it is index 11.

9) Python program to crate dictionary and print its content.**CODE:**

```
diccreacc.py - C:\Users\sai00\diccreacc.py (3.7.9)
File Edit Format Run Options Window Help
# Python program to demonstrate dictionary create and access
# Creating a dictionary
dic_county = {
    "Tamilnadu": "Chennai",
    "Andhra Pradesh": "Amaravathi",
    "Telangana": "Hyderabad"
}

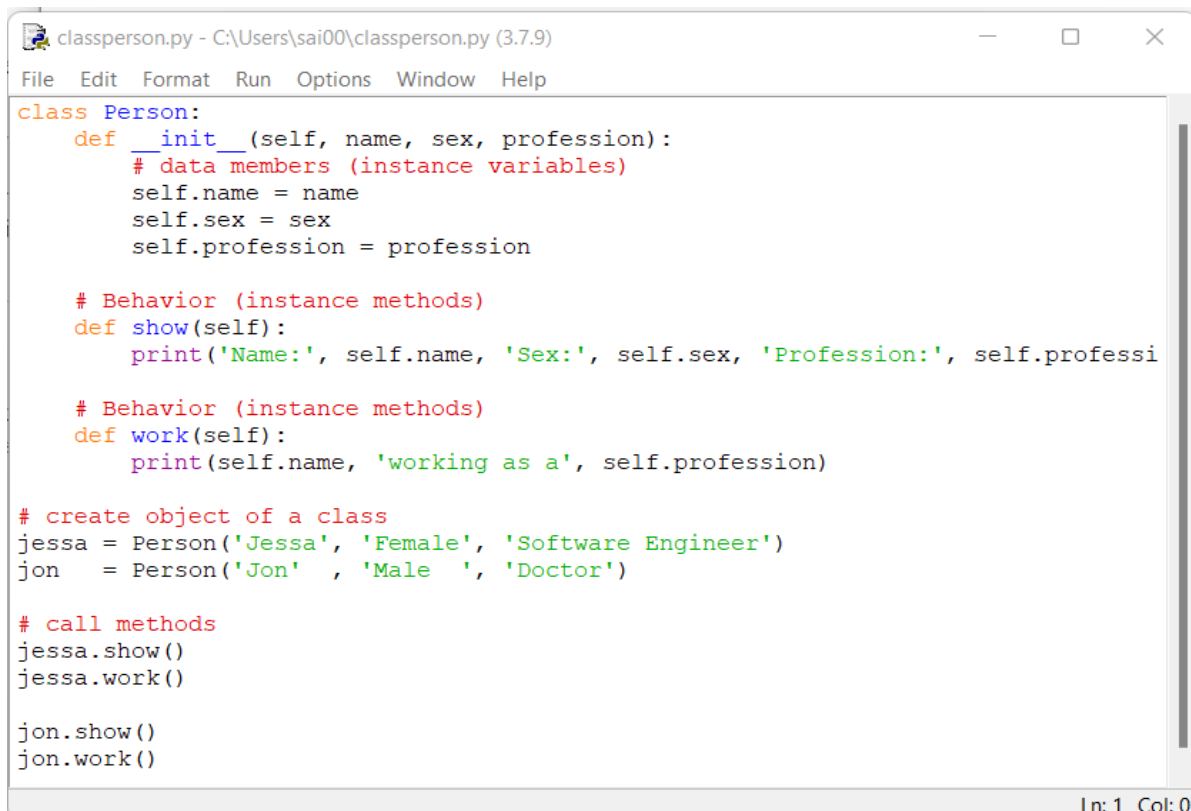
# Printing a dictionary
print("a country dictionary : ")
print(dic_county)

# access element of dictionary
print(" access element of dictionary by key: ")
print("state of Telangana:", dic_county['Telangana'])
print("state of Andhra Pradesh:", dic_county['Andhra Pradesh'])
Ln: 1 Col: 0
```


OUTPUT:

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\sai00\diccreacc.py =====
a country dictionary :
{'Tamilnadu': 'Chennai', 'Andhra Pradesh': 'Amaravathi', 'Telangana': 'Hyderabad
'}
access element of dictionary by key:
state of Telangana: Hyderabad
state of Andhra Pradesh: Amaravathi
>>>
Ln: 10 Col: 4
```

The above program created the dictionary 'dic_county' with three items. Which holds st
Python program to create class and objects.

CODE:

```
classperson.py - C:\Users\sai00\classperson.py (3.7.9)
File Edit Format Run Options Window Help
class Person:
    def __init__(self, name, sex, profession):
        # data members (instance variables)
        self.name = name
        self.sex = sex
        self.profession = profession

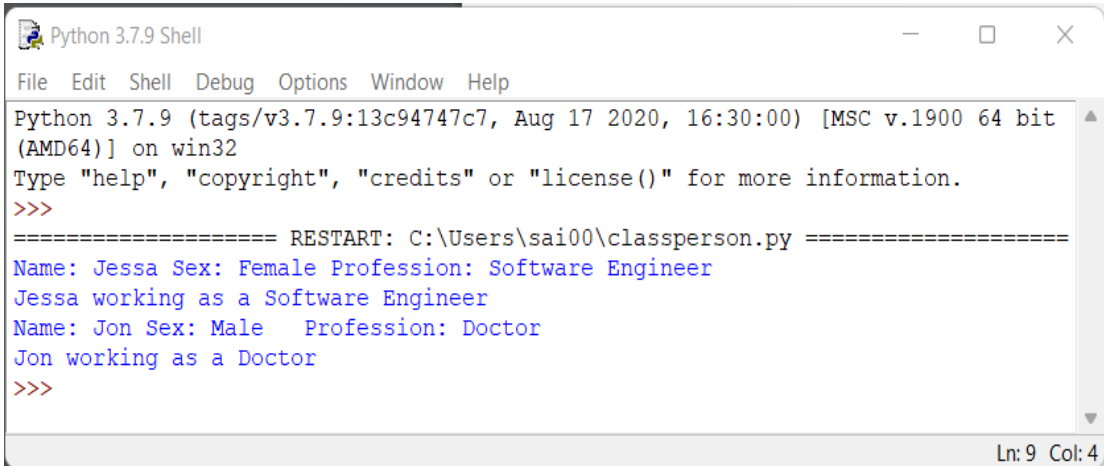
    # Behavior (instance methods)
    def show(self):
        print('Name:', self.name, 'Sex:', self.sex, 'Profession:', self.professi

    # Behavior (instance methods)
    def work(self):
        print(self.name, 'working as a', self.profession)

# create object of a class
jessa = Person('Jessa', 'Female', 'Software Engineer')
jon = Person('Jon' , 'Male ', 'Doctor')

# call methods
jessa.show()
jessa.work()

jon.show()
jon.work()
Ln: 1 Col: 0
```

OUTPUT:

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\sai00\classperson.py =====
Name: Jessa Sex: Female Profession: Software Engineer
Jessa working as a Software Engineer
Name: Jon Sex: Male Profession: Doctor
Jon working as a Doctor
>>>
Ln: 9 Col: 4
```