SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA

Think Excellence. Live Excellence.

SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY

Department of Information Technology

Python Lab (BTCS407)
LAB File

Submitted To : Prof. Dinesh Patel

Submitted By: Rohit Pawar

1810DMTIT04475

IT 'C'

CONTENTS

Pratical-1

Introduction Of python

- **<u>Introduction</u>**

Python is a simple, general purpose, high level, and object-oriented programming language. Python is an interpreted scripting language also. *Guido Van Rossum* is known as the founder of Python programming.

Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry.

Python is a programming language that lets you work quickly and integrate systems more efficiently. There are two major Python versions: **Python 2 and Python 3**. Both are quite different.

- **<u>History of Python</u>**

Python was invented by **Guido van Rossum** in 1991 at CWI in Netherland. The idea of Python programming language has taken from the ABC programming language or we can say that ABC is a predecessor of Python language.

There is also a fact behind the choosing name Python. Guido van Rossum was a fan of the popular BBC comedy show of that time, **"Monty Python's Flying Circus"**. So he decided to pick the name **Python** for his newly created programming language.

Python has the vast community across the world and releases its version within the short period.

➢ Features

1) Easy to Learn andUse

Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

2) ExpressiveLanguage

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type **print("Hello World")**. It will take only one line to execute, while Java or C takes multiple lines.

3) InterpretedLanguage

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

4) Cross-platformLanguage

Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

5) Free and OpenSource

Python is freely available for everyone. It is freely available on its official website www.python.org. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

6) Object-OrientedLanguage

Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

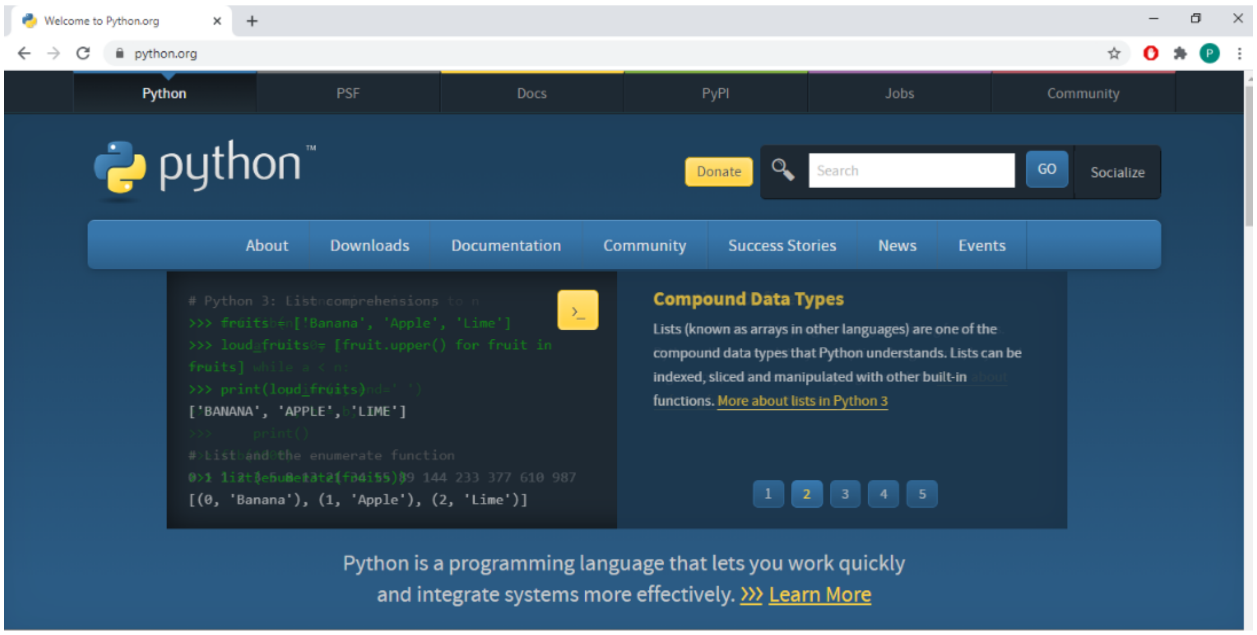
7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

- **Installation ofPython**

  Visit the link *https://www.python.org/downloads/* to download the latest release of Python. In this process, we will install Python 3.8.6 on our Windows operating system. When we click on the above link, it will bring us the following page.

- **Step - 1: Select the Python's version todownload.**
- click on the download button.

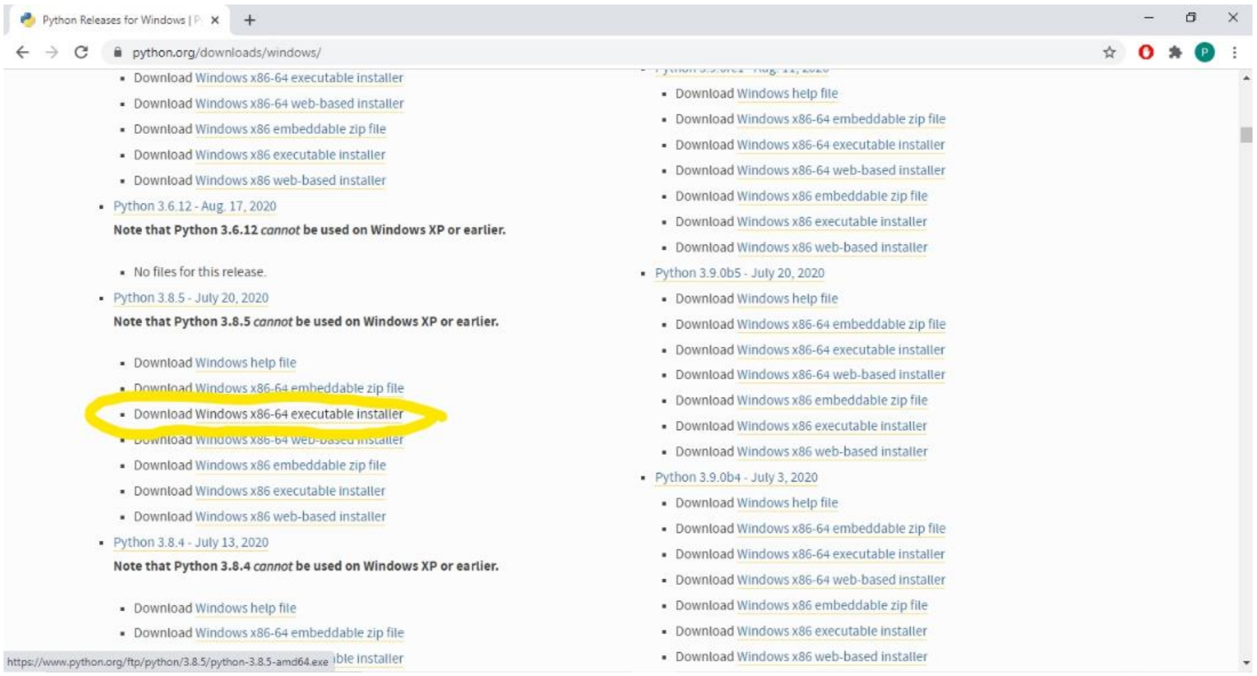


```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in
fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. More about lists in Python 3

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More



Python Releases for Windows | P

python.org/downloads/windows/

- Download Windows x86-64 executable installer
- Download Windows x86-64 web-based installer
- Download Windows x86 embeddable zip file
- Download Windows x86 executable installer
- Download Windows x86 web-based installer

- Python 3.6.12 - Aug. 17, 2020

  Note that Python 3.6.12 *cannot* be used on Windows XP or earlier.

  - No files for this release.
- Python 3.8.5 - July 20, 2020

  Note that Python 3.8.5 *cannot* be used on Windows XP or earlier.

  - Download Windows help file
  - Download Windows x86-64 embeddable zip file
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 web-based installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86 executable installer
  - Download Windows x86 web-based installer
- Python 3.8.4 - July 13, 2020

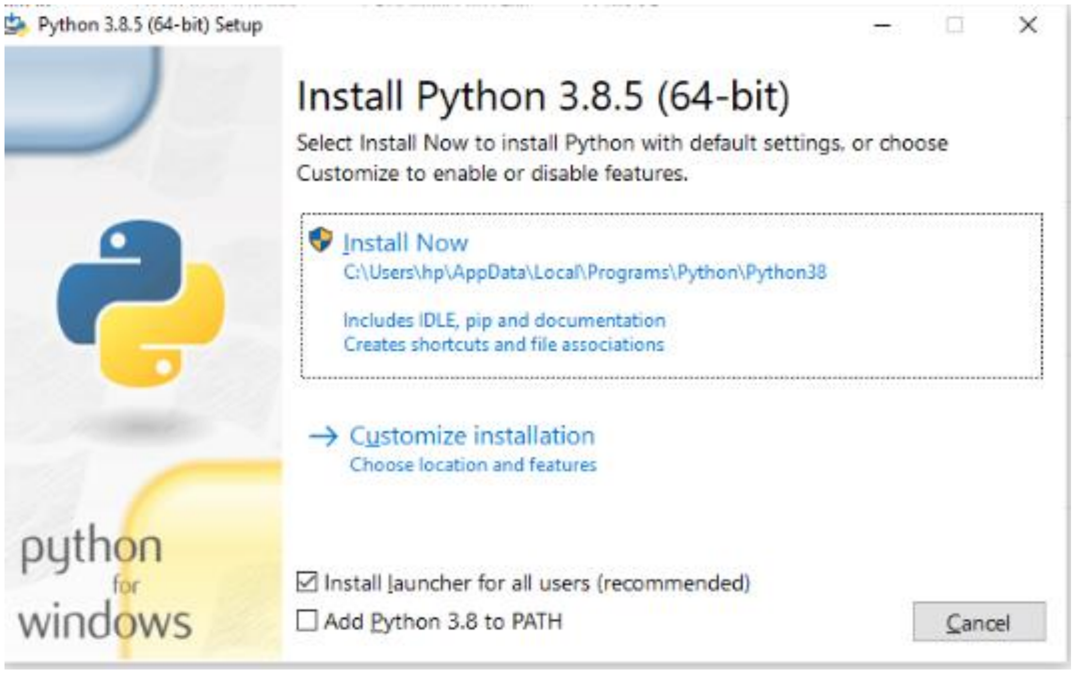
  Note that Python 3.8.4 *cannot* be used on Windows XP or earlier.

  - Download Windows help file
  - Download Windows x86-64 embeddable zip file
  - ble installer

https://www.python.org/ftp/python/3.8.5/python-3.8.5-amd64.exe

- Download Windows help file
- Download Windows x86-64 embeddable zip file
- Download Windows x86-64 executable installer
- Download Windows x86-64 web-based installer
- Download Windows x86 embeddable zip file
- Download Windows x86 executable installer
- Download Windows x86 web-based installer
- Python 3.9.0b5 - July 20, 2020
  - Download Windows help file
  - Download Windows x86-64 embeddable zip file
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 web-based installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86 executable installer
  - Download Windows x86 web-based installer
- Python 3.9.0b4 - July 3, 2020
  - Download Windows help file
  - Download Windows x86-64 embeddable zip file
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 web-based installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86 executable installer
  - Download Windows x86 web-based installer

- **Step - 2: Click on the InstallNow**
- Double-click the executable file, which is downloaded; the following window will open. Select Customize installation and proceed. Click on the Add Path check box, it will set the Python path automatically.



Python 3.8.5 (64-bit) Setup

Install Python 3.8.5 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\hp\AppData\Local\Programs\Python\Python38

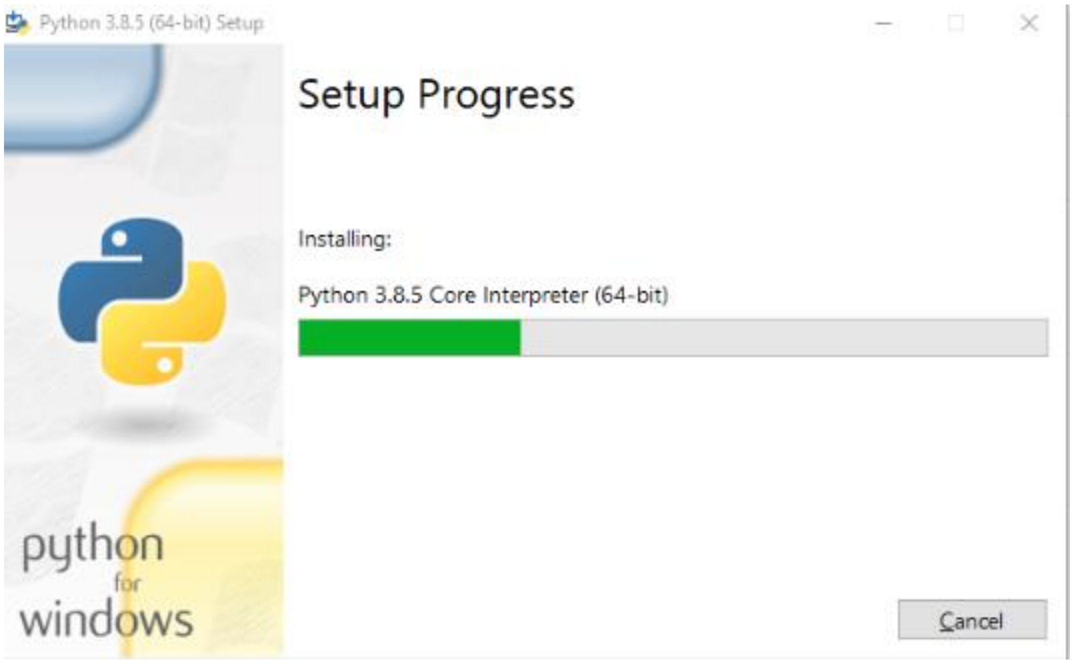
Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☐ Add Python 3.8 to PATH

- We can also click on the customize installation to choose desired location and features.Other important thing is install launcher for the all user must bechecked.

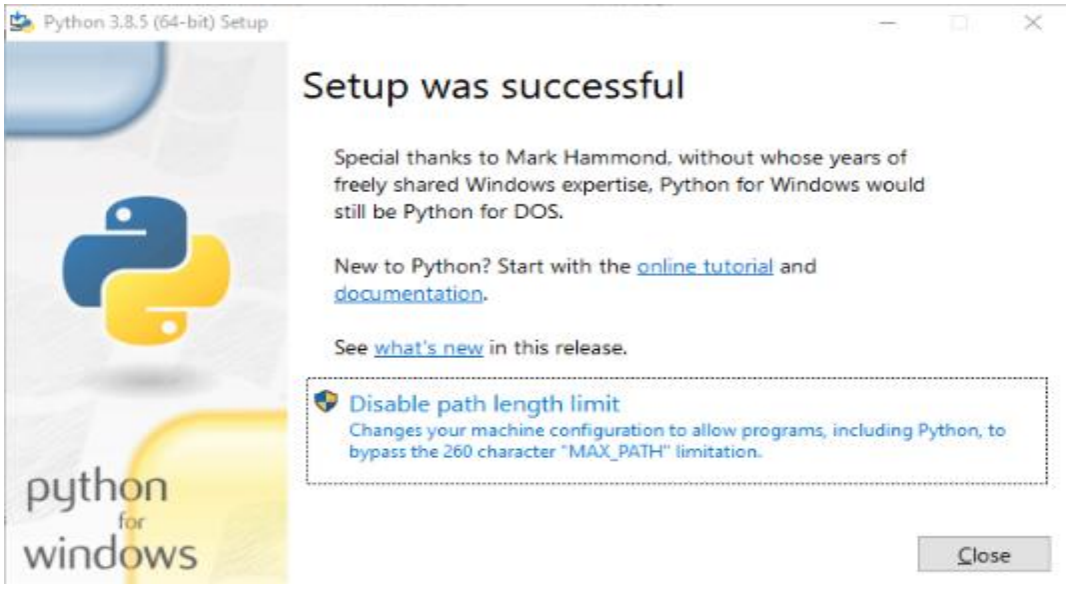
- **Step - 3 Installation inProcess**


Python 3.8.5 (64-bit) Setup

Setup Progress

Installing:

Python 3.8.5 Core Interpreter (64-bit)


Python 3.8.5 (64-bit) Setup

Setup was successful

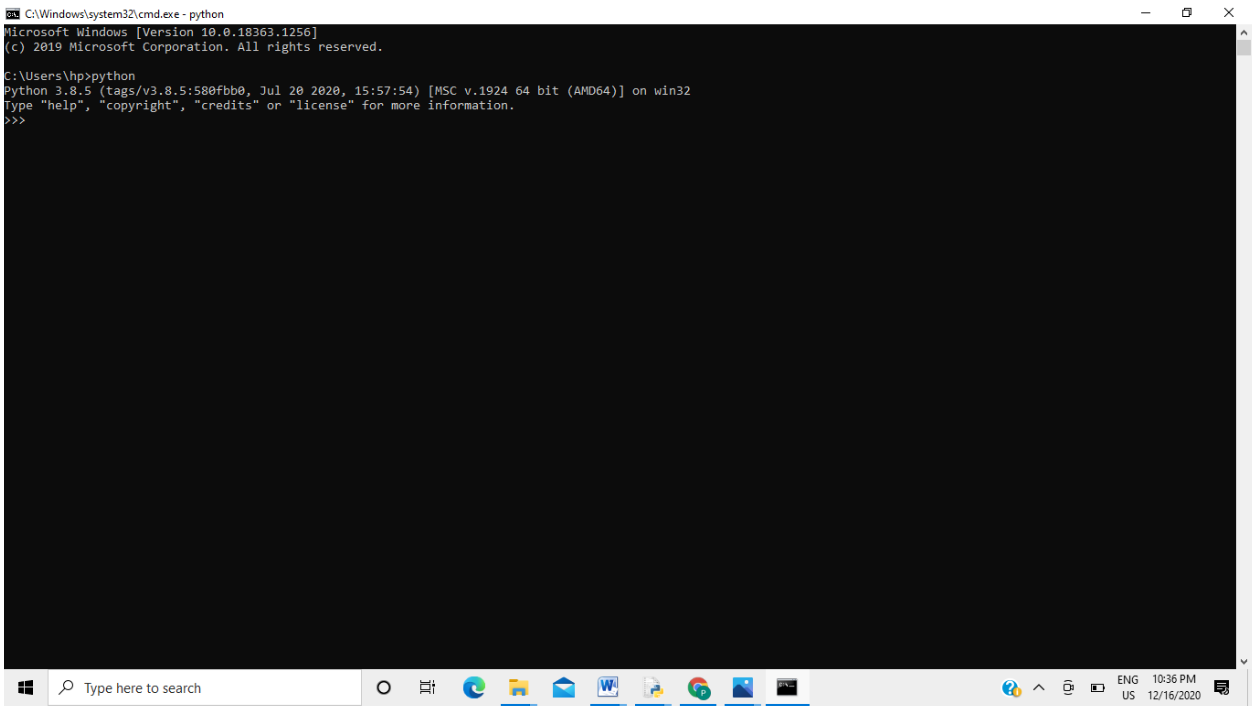
Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Now, try to run python on the command prompt. Type the command python -version in case of python3.



```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hp>python
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

We are ready to work with thePython.

Practical 2

Write a program to demonstrate basic data type in python.

Aim: Write a program to demonstrate basic data type in python.

Parameter-: a,n

Program

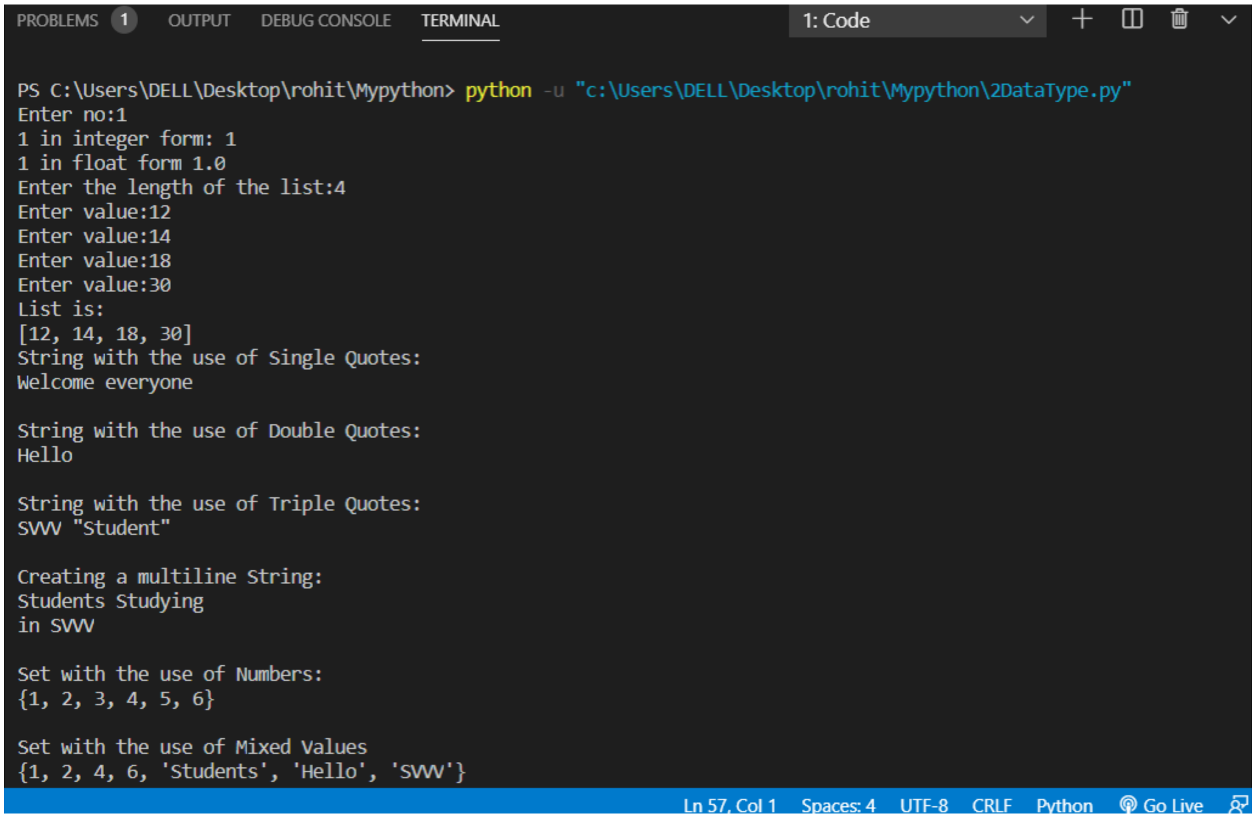
```
#int ,float data type

a=int(input("Enter no:"))

b=int(a)

c=float(a)


print(a,"in integer form:",b)

print(a,"in float form",c)


#list data type


lst=[]

n=int(input("Enter the length of the list:"))

for i in range(0,n):

    x=int(input("Enter value:"))

    lst.append(x)


print("List is:")

print(lst)


#String data type
```

```
String1 = 'Welcome everyone'

print("String with the use of Single Quotes: ")

print(String1)

String1 = "Hello "

print("\nString with the use of Double Quotes: ")

print(String1)

String1 = '''SVVV "Student"'''

print("\nString with the use of Triple Quotes: ")

print(String1)

String1 = '''Students Studying

in SVVV'''

print("\nCreating a multiline String: ")

print(String1)


#set data type

set1 = set([1, 2, 4, 4, 3, 3, 3, 6, 5])

print("\nSet with the use of Numbers: ")

print(set1)

set1 = set([1, 2, 'Hello', 4, 'SVVV', 6, 'Students'])

print("\nSet with the use of Mixed Values")

print(set1)


# dictionary data type

Dict = {}

print("Empty Dictionary: ")
print(Dict)
Dict = dict({1: 'AAA', 2: 'BBB', 3:'CCC'})
print("\nDictionary with the use of dict(): ")
print(Dict)
Dict = dict([(1, 'Hello'), (2, 'World')])
print("\nDictionary with each item as a pair: ")
print(Dict)
#tupule data type
tuple1 = (0, 1, 2, 3)
tuple2 = ('Hello', 'world')
print("Concatenation of two tuples:")
print(tuple1 + tuple2)
```

FILE -:

2DataType.py

Output



```
PS C:\Users\DELL\Desktop\rohit\Mypython> python -u "c:\Users\DELL\Desktop\rohit\Mypython\2DataType.py"
Enter no:1
1 in integer form: 1
1 in float form 1.0
Enter the length of the list:4
Enter value:12
Enter value:14
Enter value:18
Enter value:30
List is:
[12, 14, 18, 30]
String with the use of Single Quotes:
Welcome everyone

String with the use of Double Quotes:
Hello

String with the use of Triple Quotes:
SVVV "Student"

Creating a multiline String:
Students Studying
in SVVV

Set with the use of Numbers:
{1, 2, 3, 4, 5, 6}

Set with the use of Mixed Values
{1, 2, 4, 6, 'Students', 'Hello', 'SVVV'}
```

PRACTICAL 3
Write a program to demonstrate basic data type in python.

Aim: Write a program to demonstrate basic data type in python

Parameter:

Program:

```
#id() function return the Memory Address of variable

a="rohit"

b="rohit"

print("Address of a",id(a))

print("Address of b",id(b))

if(id(a)==id(b)):

    print("Same memory block is Shared by a and b var")


#type() it is used to identify which type of data is contained by the variable

c=[]

d={}

e=()

f=10

g=10.0

print(type(a))

print(type(b))

print(type(c))

print(type(d))

print(type(e))
```

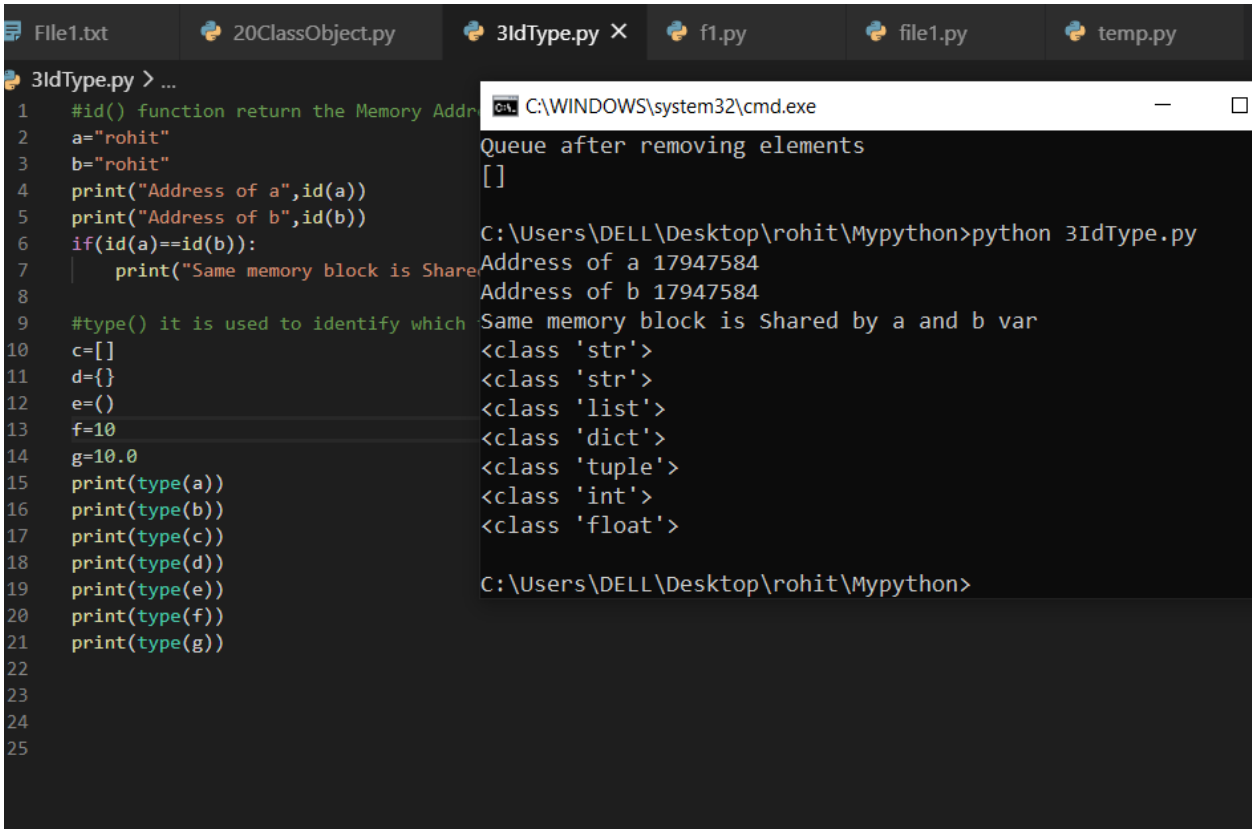
print(type(f))

print(type(g))

File-:

3IDtype.py

Output



FIle1.txt | 20ClassObject.py | 3IdType.py | f1.py | file1.py | temp.py

3IdType.py > ...

```python
#id() function return the Memory Addr
a="rohit"
b="rohit"
print("Address of a",id(a))
print("Address of b",id(b))
if(id(a)==id(b)):
    print("Same memory block is Share

#type() it is used to identify which
c=[]
d={}
e=()
f=10
g=10.0
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
print(type(f))
print(type(g))
```

C:\WINDOWS\system32\cmd.exe

```
Queue after removing elements
[]

C:\Users\DELL\Desktop\rohit\Mypython>python 3IdType.py
Address of a 17947584
Address of b 17947584
Same memory block is Shared by a and b var
<class 'str'>
<class 'str'>
<class 'list'>
<class 'dict'>
<class 'tuple'>
<class 'int'>
<class 'float'>

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 4
Write a program to find GCD of two numbers.

Aim: Write a program to find GCD of two numbers.

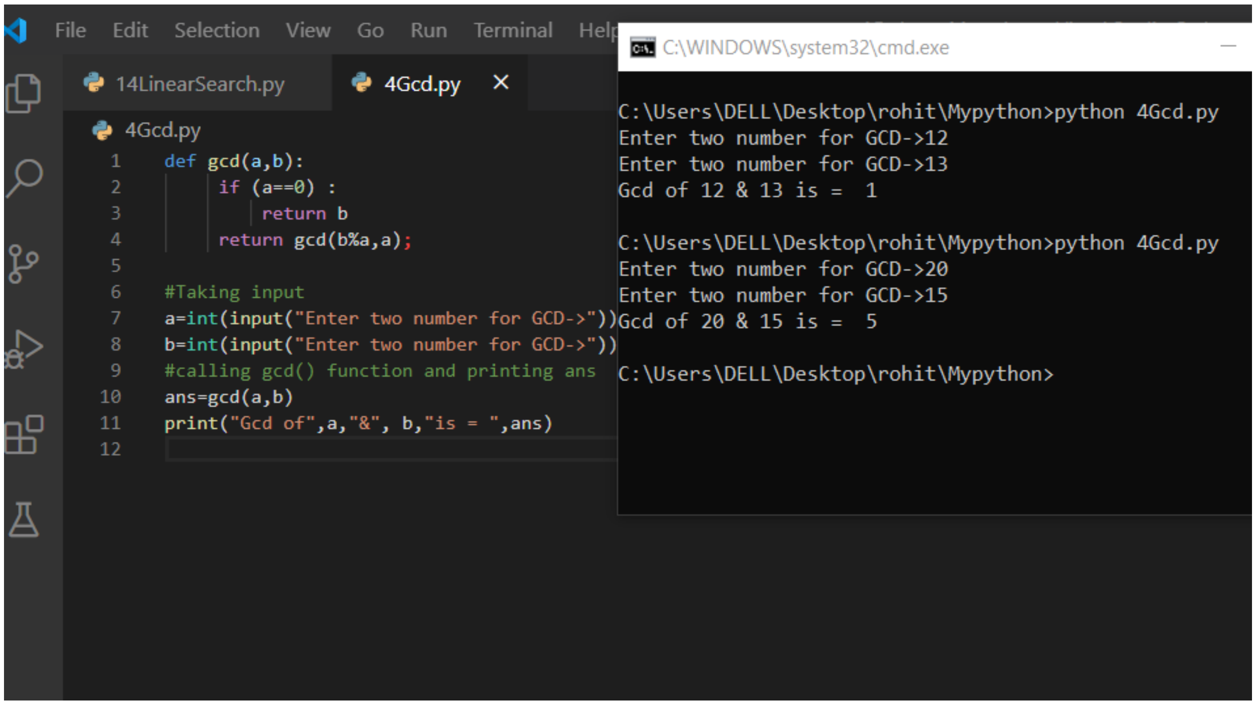
Parameter: a,b

Program:

```
def gcd(a,b):

    if (a==0) :

        return b

    return gcd(b%a,a);


#Taking input

a=int(input("Enter two number for GCD->"))

b=int(input("Enter two number for GCD->"))

#calling gcd() function and printing ans

ans=gcd(a,b)

print("Gcd of",a,"&", b,"is = ",ans)
```

FILE:- 4Gcd.py

OUTPUT-:



4Gcd.py

```python
def gcd(a,b):
    if (a==0) :
        return b
    return gcd(b%a,a);

#Taking input
a=int(input("Enter two number for GCD->"))
b=int(input("Enter two number for GCD->"))
#calling gcd() function and printing ans
ans=gcd(a,b)
print("Gcd of",a,"&", b,"is = ",ans)
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 4Gcd.py
Enter two number for GCD->12
Enter two number for GCD->13
Gcd of 12 & 13 is =  1

C:\Users\DELL\Desktop\rohit\Mypython>python 4Gcd.py
Enter two number for GCD->20
Enter two number for GCD->15
Gcd of 20 & 15 is =  5

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 5

Write a program to compute distance between two points taking input from the user (Pythagorean Theorem).

Aim: Write a program to compute distance between two points taking input from the user (Pythagorean Theorem).

Parameter x1,x2,y1,y2

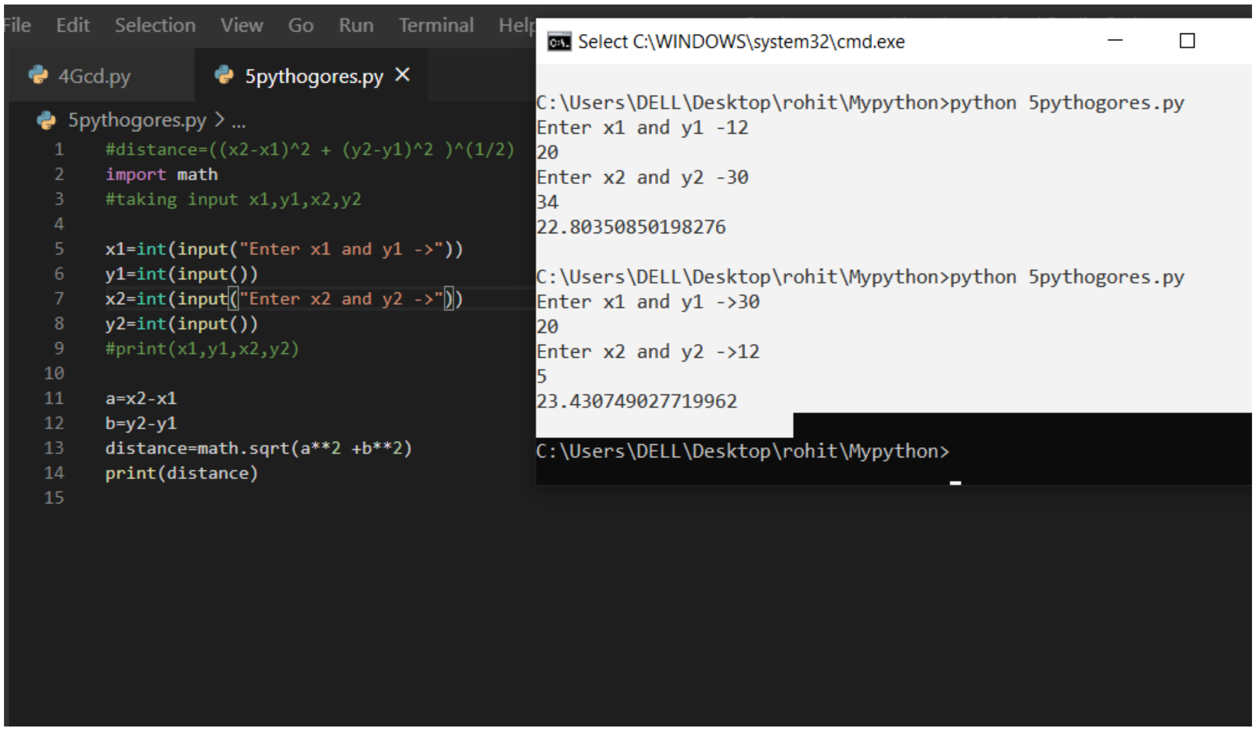
Program

```
#distance=((x2-x1)^2 + (y2-y1)^2 )^(1/2)

import math

#taking input x1,y1,x2,y2


x1=int(input("Enter x1 and y1 ->"))

y1=int(input())

x2=int(input("Enter x2 and y2 ->"))

y2=int(input())

#print(x1,y1,x2,y2)


a=x2-x1

b=y2-y1

distance=math.sqrt(a**2 +b**2)

print(distance)
```

File:-

5pythogores.py

Output-:


File Edit Selection View Go Run Terminal Help
4Gcd.py
5pythogores.py
5pythogores.py > ...
Select C:\WINDOWS\system32\cmd.exe

```python
#distance=((x2-x1)^2 + (y2-y1)^2 )^(1/2)
import math
#taking input x1,y1,x2,y2

x1=int(input("Enter x1 and y1 ->"))
y1=int(input())
x2=int(input("Enter x2 and y2 ->"))
y2=int(input())
#print(x1,y1,x2,y2)

a=x2-x1
b=y2-y1
distance=math.sqrt(a**2 +b**2)
print(distance)

```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 5pythogores.py
Enter x1 and y1 -12
20
Enter x2 and y2 -30
34
22.80350850198276

C:\Users\DELL\Desktop\rohit\Mypython>python 5pythogores.py
Enter x1 and y1 ->30
20
Enter x2 and y2 ->12
5
23.430749027719962

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 6

Write a python Program to find the square root of a number taking input from user (Newton's Method).

Aim: Write a python Program to find the square root of a number taking input from user (Newton's Method).

Parameter a

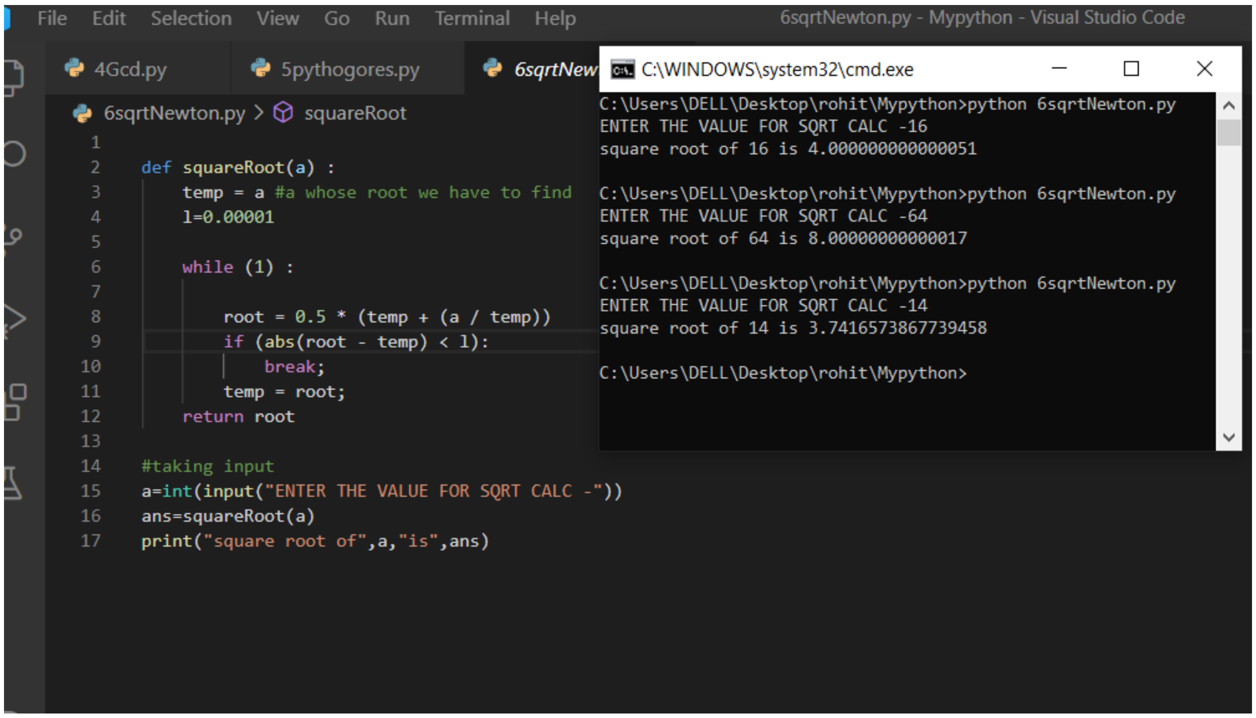
Program

```
def squareRoot(a) :
    temp = a #a whose root we have to find
    l=0.00001

    while (1) :

        root = 0.5 * (temp + (a / temp))
        if (abs(root - temp) < l):
            break;
        temp = root;
    return root

#taking input
a=int(input("ENTER THE VALUE FOR SQRT CALC -"))
ans=squareRoot(a)
print("square root of",a,"is",ans)
```

FILE-:

6sqrtNewton.py

OUTPUT-:



File Edit Selection View Go Run Terminal Help
6sqrtNewton.py - Mypython - Visual Studio Code
4Gcd.py
5pythogores.py
6sqrtNew
6sqrtNewton.py > squareRoot
C:\WINDOWS\system32\cmd.exe

```python

def squareRoot(a) :
    temp = a #a whose root we have to find
    l=0.00001

    while (1) :

        root = 0.5 * (temp + (a / temp))
        if (abs(root - temp) < l):
            break;
        temp = root;
    return root

#taking input
a=int(input("ENTER THE VALUE FOR SQRT CALC -"))
ans=squareRoot(a)
print("square root of",a,"is",ans)
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 6sqrtNewton.py
ENTER THE VALUE FOR SQRT CALC -16
square root of 16 is 4.000000000000051

C:\Users\DELL\Desktop\rohit\Mypython>python 6sqrtNewton.py
ENTER THE VALUE FOR SQRT CALC -64
square root of 64 is 8.00000000000017

C:\Users\DELL\Desktop\rohit\Mypython>python 6sqrtNewton.py
ENTER THE VALUE FOR SQRT CALC -14
square root of 14 is 3.7416573867739458

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 7

Write a program add.py that takes 2 numbers as command line arguments and perform their sum.

Aim: Write a program add.py that takes 2 numbers as command line arguments and perform their sum.

Parameter: command line parameter

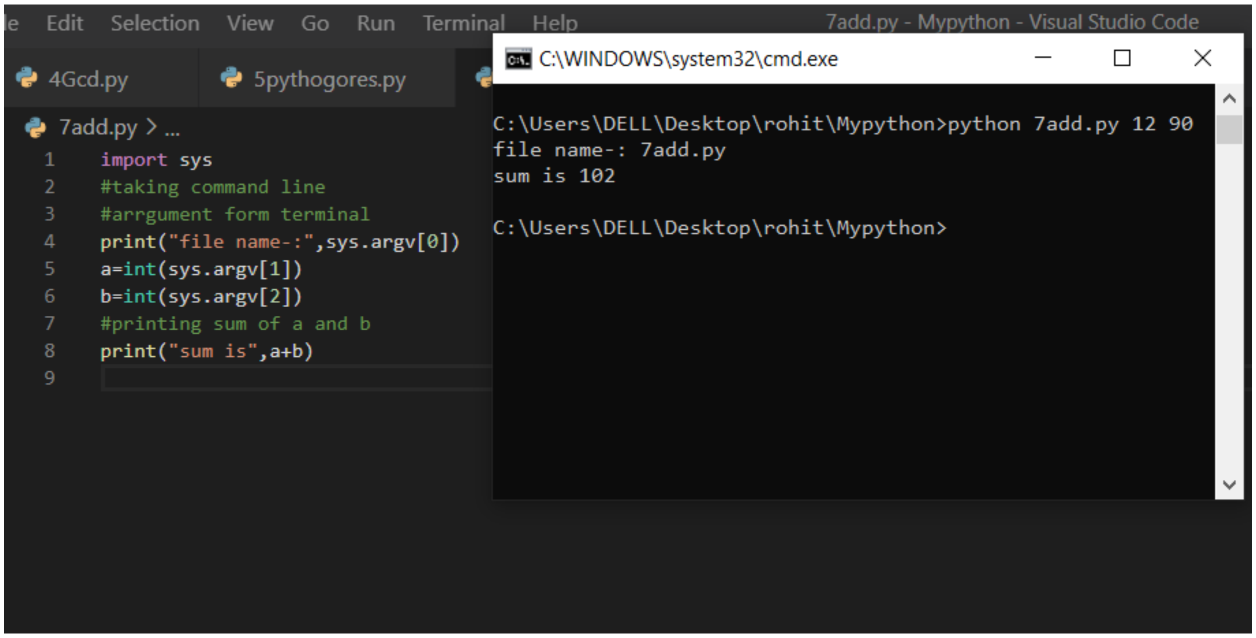
Program:

```
 import sys
#taking command line
#arrgument form terminal
print("file name-:",sys.argv[0])
a=int(sys.argv[1])
b=int(sys.argv[2])
#printing sum of a and b
print("sum is",a+b)
```

File

7add.py

Output


le Edit Selection View Go Run Terminal Help
7add.py - Mypython - Visual Studio Code
4Gcd.py
5pythogores.py
7add.py > ...
C:\WINDOWS\system32\cmd.exe

```
import sys
#taking command line
#arrgument form terminal
print("file name-:",sys.argv[0])
a=int(sys.argv[1])
b=int(sys.argv[2])
#printing sum of a and b
print("sum is",a+b)
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 7add.py 12 90
file name-: 7add.py
sum is 102

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 8

Write a program to purposefully raise Indentation Error and Correct it.

Aim: Write a program to purposefully raise Indentation Error and Correct it.

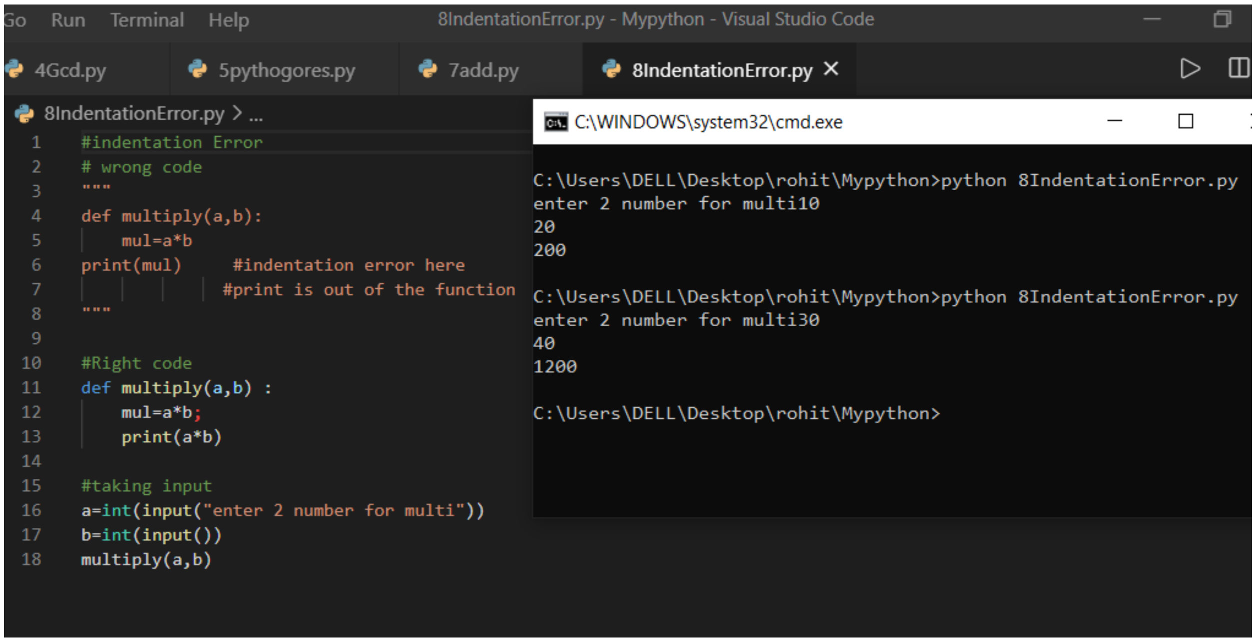
Parameter: a,b

Program

```
 #indentation Error
# wrong code
"""
def multiply(a,b):
    mul=a*b
print(mul)     #indentation error here
              #print is out of the function
"""


#Right code
def multiply(a,b) :
    mul=a*b;
    print(a*b)


#taking input
a=int(input("enter 2 number for multi"))
b=int(input())
multiply(a,b)
```

File: 8IndentationError.py

Output


Go Run Terminal Help
8IndentationError.py - Mypython - Visual Studio Code
4Gcd.py
5pythogores.py
7add.py
8IndentationError.py
8IndentationError.py > ...
C:\WINDOWS\system32\cmd.exe

```python
#indentation Error
# wrong code
"""
def multiply(a,b):
    mul=a*b
print(mul)     #indentation error here
               #print is out of the function
"""

#Right code
def multiply(a,b) :
    mul=a*b;
    print(a*b)

#taking input
a=int(input("enter 2 number for multi"))
b=int(input())
multiply(a,b)
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 8IndentationError.py
enter 2 number for multi10
20
200

C:\Users\DELL\Desktop\rohit\Mypython>python 8IndentationError.py
enter 2 number for multi30
40
1200

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 9
Write a program to find the maximum from a list of numbers.

Aim: Write a program to find the maximum from a list of numbers.

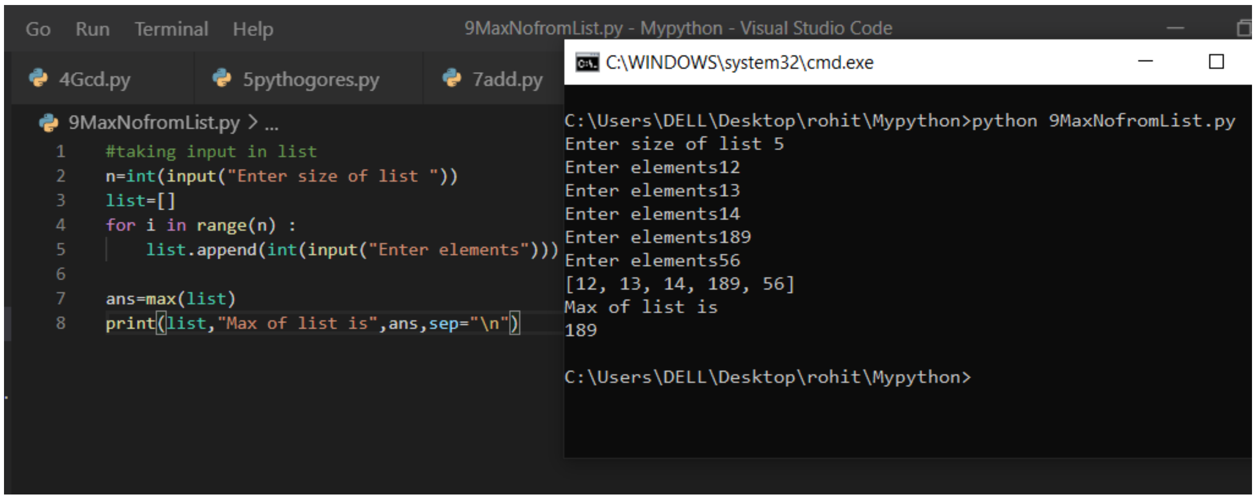
Parameter: n,list

Program

```
 #taking input in list

n=int(input("Enter size of list "))

list=[]

for i in range(n) :

    list.append(int(input("Enter elements")))



ans=max(list)

print(list,"Max of list is",ans,sep="\n")
```

File: 9MaxNoFromList.py

Output


Go Run Terminal Help
9MaxNofromList.py - Mypython - Visual Studio Code
4Gcd.py
5pythogores.py
7add.py
9MaxNofromList.py > ...
C:\WINDOWS\system32\cmd.exe

```
#taking input in list
n=int(input("Enter size of list "))
list=[]
for i in range(n) :
    list.append(int(input("Enter elements")))

ans=max(list)
print(list,"Max of list is",ans,sep="\n")
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 9MaxNofromList.py
Enter size of list 5
Enter elements12
Enter elements13
Enter elements14
Enter elements189
Enter elements56
[12, 13, 14, 189, 56]
Max of list is
189

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 10

Write a program to find all prime numbers within a given range.

Aim: Write a program to find all prime numbers within a given range.

Parameter: n

Program

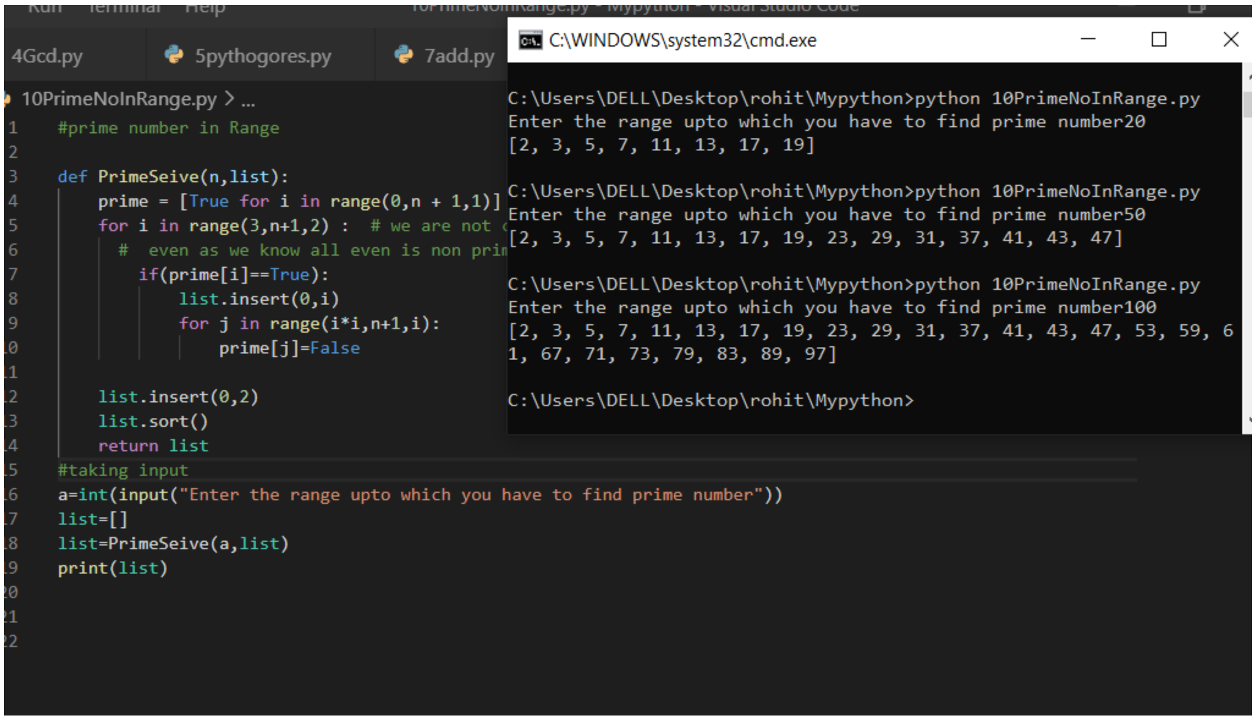
```
 #prime number in Range


def PrimeSeive(n,list):
    prime = [True for i in range(0,n + 1,1)]
    for i in range(3,n+1,2) :  # we are not checking for
      #  even as we know all even is non prime except 2
        if(prime[i]==True):
            list.insert(0,i)
            for j in range(i*i,n+1,i):
                prime[j]=False


    list.insert(0,2)
    list.sort()
    return list
#taking input
a=int(input("Enter the range upto which you have to find prime number"))
list=[]
list=PrimeSeive(a,list)
```

print(list

File:

10PrimeNoInRange.py

Output



```
#prime number in Range

def PrimeSeive(n,list):
    prime = [True for i in range(0,n + 1,1)]
    for i in range(3,n+1,2) :  # we are not c
      #  even as we know all even is non prim
        if(prime[i]==True):
            list.insert(0,i)
            for j in range(i*i,n+1,i):
                prime[j]=False

    list.insert(0,2)
    list.sort()
    return list
#taking input
a=int(input("Enter the range upto which you have to find prime number"))
list=[]
list=PrimeSeive(a,list)
print(list)
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 10PrimeNoInRange.py
Enter the range upto which you have to find prime number20
[2, 3, 5, 7, 11, 13, 17, 19]

C:\Users\DELL\Desktop\rohit\Mypython>python 10PrimeNoInRange.py
Enter the range upto which you have to find prime number50
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

C:\Users\DELL\Desktop\rohit\Mypython>python 10PrimeNoInRange.py
Enter the range upto which you have to find prime number100
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 6
1, 67, 71, 73, 79, 83, 89, 97]

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 11

Write a program to count the numbers of characters in the string and store them in a dictionary data structure.

Aim: Write a program to count the numbers of characters in the string and store them in a dictionary data structure.

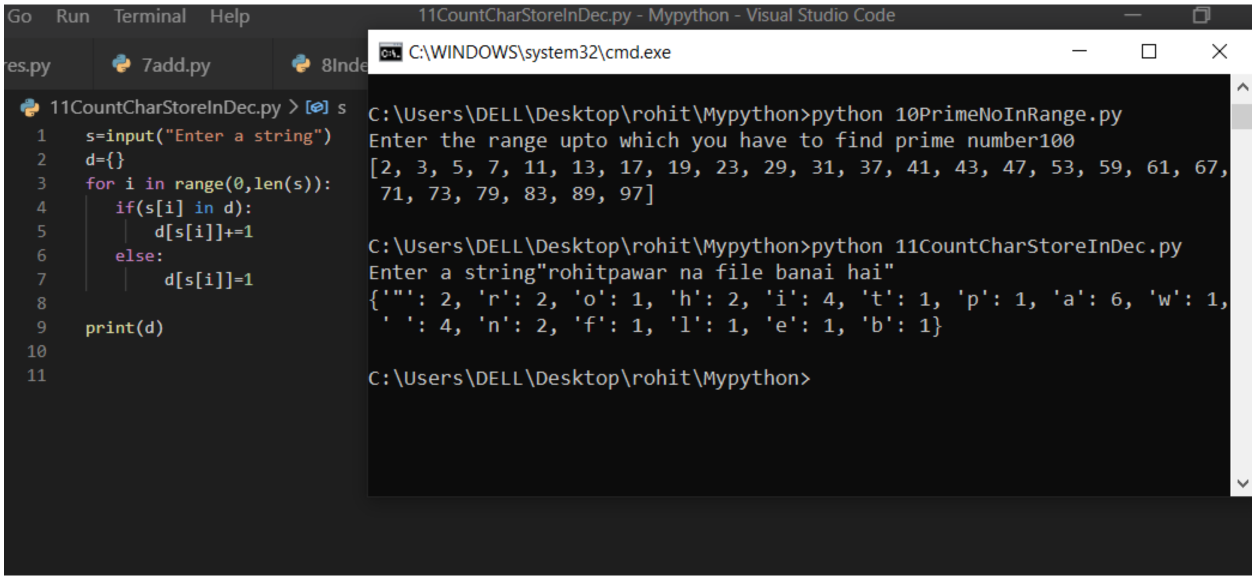
Parameter: s(string)

Program

```
 s=input("Enter a string")

d={}

for i in range(0,len(s)):

   if(s[i] in d):

       d[s[i]]+=1

   else:

        d[s[i]]=1


print(d)
```

File: 11CountCharStoreInDec.py

Output


Go Run Terminal Help
11CountCharStoreInDec.py - Mypython - Visual Studio Code
es.py
7add.py
8Inde
11CountCharStoreInDec.py > s
C:\WINDOWS\system32\cmd.exe

```python
s=input("Enter a string")
d={}
for i in range(0,len(s)):
    if(s[i] in d):
        d[s[i]]+=1
    else:
        d[s[i]]=1

print(d)
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 10PrimeNoInRange.py
Enter the range upto which you have to find prime number100
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67,
 71, 73, 79, 83, 89, 97]

C:\Users\DELL\Desktop\rohit\Mypython>python 11CountCharStoreInDec.py
Enter a string"rohitpawar na file banai hai"
{'"': 2, 'r': 2, 'o': 1, 'h': 2, 'i': 4, 't': 1, 'p': 1, 'a': 6, 'w': 1,
 ' ': 4, 'n': 2, 'f': 1, 'l': 1, 'e': 1, 'b': 1}

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 12

Write a program to multiply matrices using function.

Aim: Write a program to multiply matrices using function.

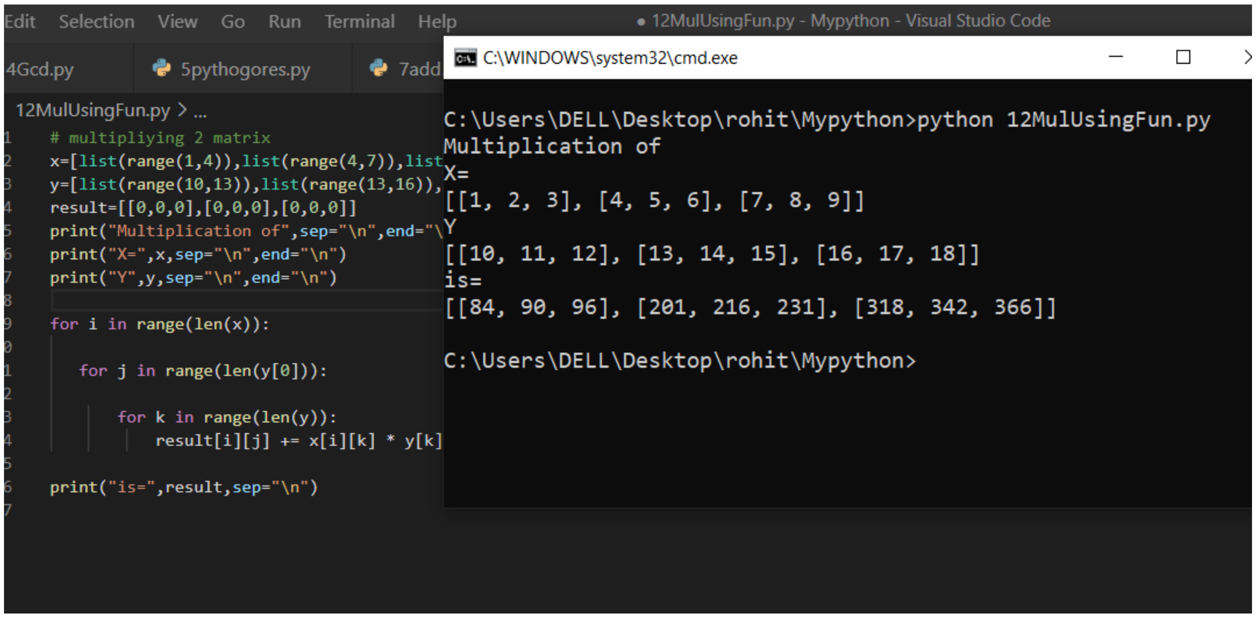
Parameter: x,y,result

Program

```
x=[list(range(1,4)),list(range(4,7)),list(range(7,10))]

y=[list(range(10,13)),list(range(13,16)),list(range(16,19))]

result=[[0,0,0],[0,0,0],[0,0,0]]

print("Multiplication of",sep="\n",end="\n")

print("X=",x,sep="\n",end="\n")

print("Y",y,sep="\n",end="\n")


for i in range(len(x)):

   for j in range(len(y[0]))

       for k in range(len(y)):

           result[i][j] += x[i][k] * y[k][j]


print("is=",result,sep="\n")
```

File: 12MulUsingFun.py

Output


Edit Selection View Go Run Terminal Help
● 12MulUsingFun.py - Mypython - Visual Studio Code
4Gcd.py
5pythogores.py
7add
12MulUsingFun.py > ...
C:\WINDOWS\system32\cmd.exe

```
# multipliying 2 matrix
x=[list(range(1,4)),list(range(4,7)),list
y=[list(range(10,13)),list(range(13,16)),
result=[[0,0,0],[0,0,0],[0,0,0]]
print("Multiplication of",sep="\n",end="\
print("X=",x,sep="\n",end="\n")
print("Y",y,sep="\n",end="\n")

for i in range(len(x)):

    for j in range(len(y[0])):

        for k in range(len(y)):
            result[i][j] += x[i][k] * y[k]

print("is=",result,sep="\n")
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 12MulUsingFun.py
Multiplication of
X=
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Y
[[10, 11, 12], [13, 14, 15], [16, 17, 18]]
is=
[[84, 90, 96], [201, 216, 231], [318, 342, 366]]

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 13

Write a program to print n terms of Fibonacci series using iteration

Aim: Write a program to print n terms of Fibonacci series using iteration

Parameter: a , b

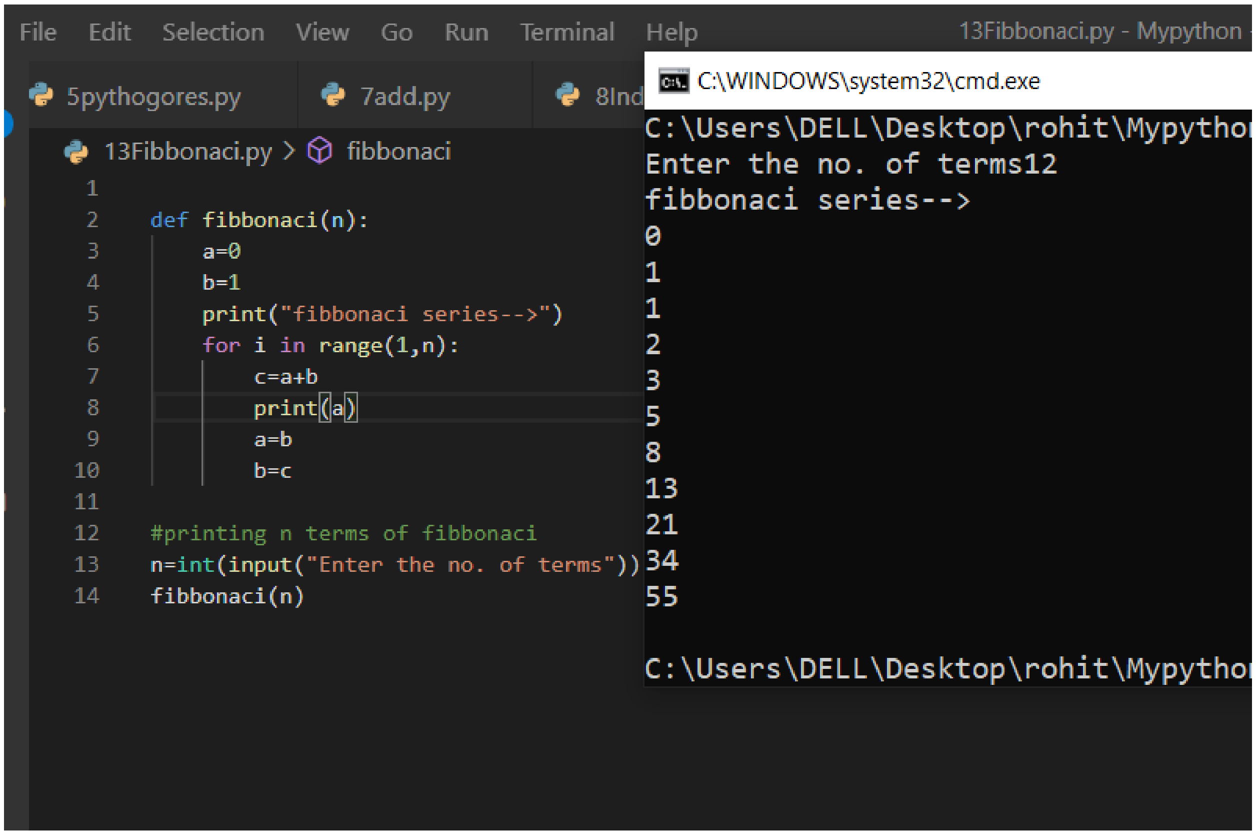
Programme

```
def fibbonaci(n):
    a=0
    b=1
    print("fibbonaci series-->")
    for i in range(1,n):
        c=a+b
        print(a)
        a=b
        b=c


#printing n terms of fibbonaci
n=int(input("Enter the no. of terms"))
fibbonaci(n)
```

File:

13Fibbonaci.py

Output


File Edit Selection View Go Run Terminal Help
13Fibbonaci.py - Mypython
5pythogores.py
7add.py
8Ind
13Fibbonaci.py > fibbonaci
C:\WINDOWS\system32\cmd.exe

```python

def fibbonaci(n):
    a=0
    b=1
    print("fibbonaci series-->")
    for i in range(1,n):
        c=a+b
        print(a)
        a=b
        b=c

#printing n terms of fibbonaci
n=int(input("Enter the no. of terms"))
fibbonaci(n)
```

```
C:\Users\DELL\Desktop\rohit\Mypytho
Enter the no. of terms12
fibbonaci series-->
0
1
1
2
3
5
8
13
21
34
55

C:\Users\DELL\Desktop\rohit\Mypytho
```

PRACTICAL 14

Write a Python Program to perform Linear Search

Aim: Write a Python Program to perform Linear Search

Parameter: s(string) ,find(char)

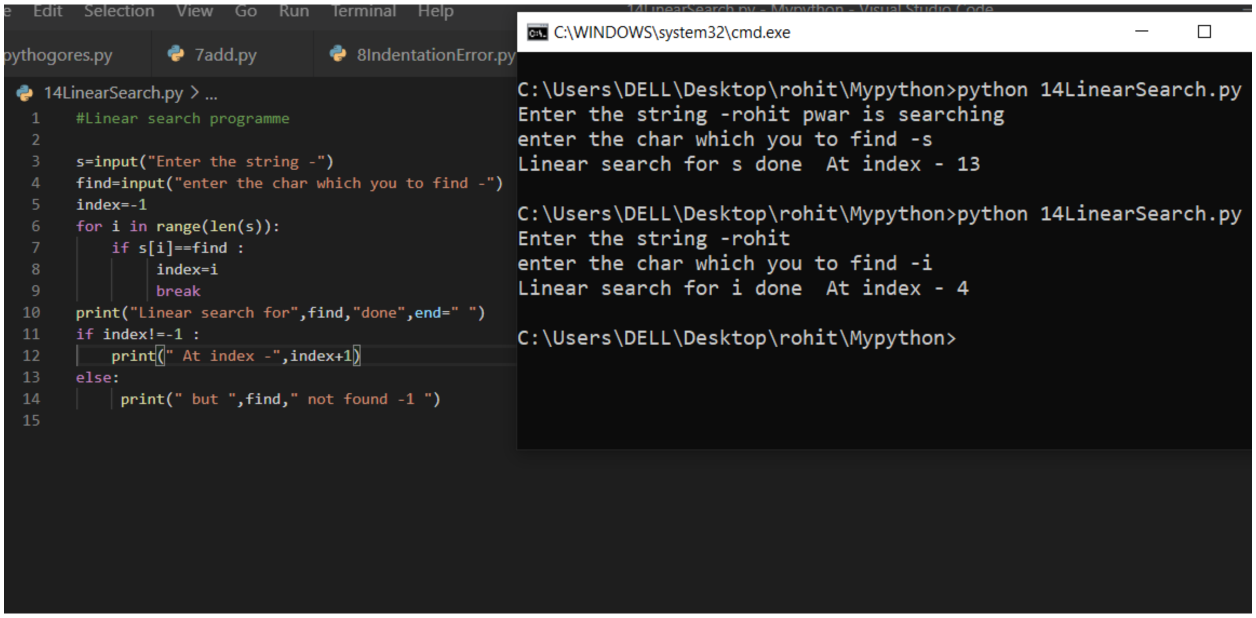
Program

```
 #Linear search programme


s=input("Enter the string -")
find=input("enter the char which you to find -")
index=-1
for i in range(len(s)):
    if s[i]==find :
         index=i
         break
print("Linear search for",find,"done",end=" ")
if index!=-1 :
    print(" At index -",index+1)
else:
     print(" but ",find," not found -1 ")
```

File:

14LinearSearch.py

Output


e Edit Selection View Go Run Terminal Help
pythogores.py
7add.py
8IndentationError.py
14LinearSearch.py > ...
C:\WINDOWS\system32\cmd.exe

```
#Linear search programme

s=input("Enter the string -")
find=input("enter the char which you to find -")
index=-1
for i in range(len(s)):
    if s[i]==find :
        index=i
        break
print("Linear search for",find,"done",end=" ")
if index!=-1 :
    print(" At index -",index+1)
else:
     print(" but ",find," not found -1 ")
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 14LinearSearch.py
Enter the string -rohit pwar is searching
enter the char which you to find -s
Linear search for s done  At index - 13

C:\Users\DELL\Desktop\rohit\Mypython>python 14LinearSearch.py
Enter the string -rohit
enter the char which you to find -i
Linear search for i done  At index - 4

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 15

Write a Python Program to perform Binary Search

Aim: Write a Python Program to perform Binary Search

Parameter: list,key(int)

Program

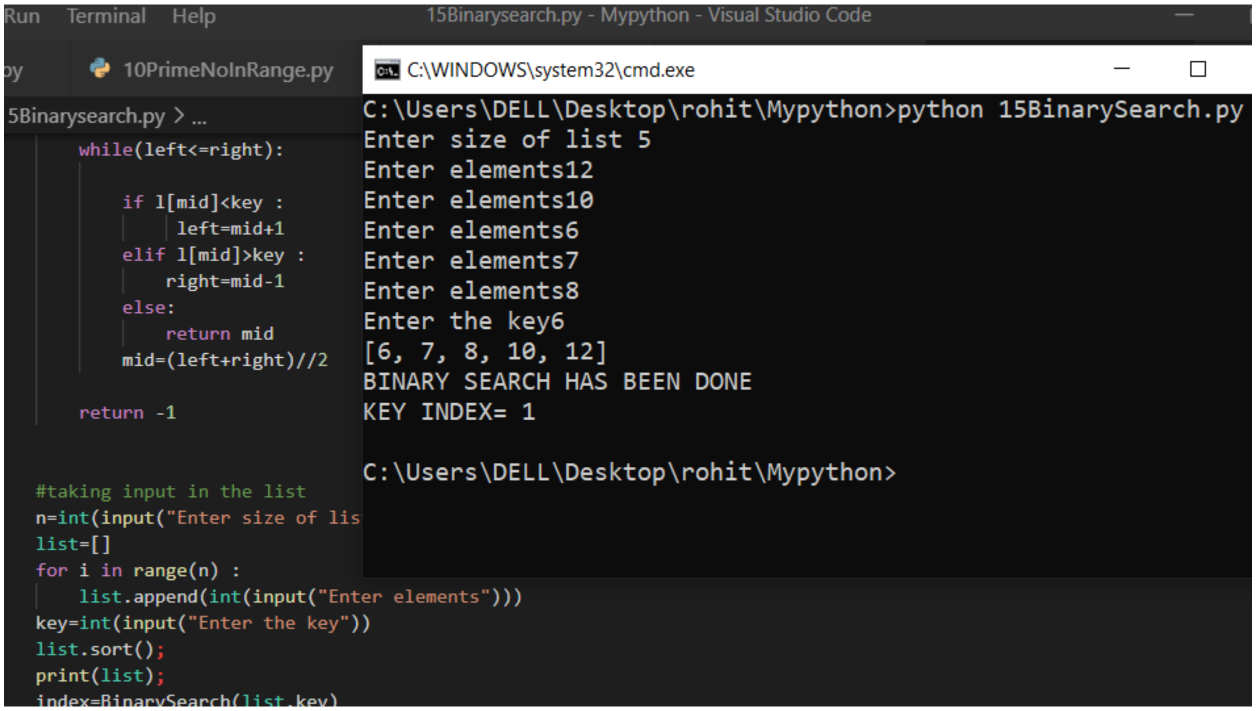
```
 #BinarySearch -:it is only applied when int sequence is sorted.

#finding key in the List of Integer sir.

def BinarySearch(l,key):

    n=len(l)

    left=0

    right=n-1

    mid=(left+right//2)

    while(left<=right):


        if l[mid]<key :

             left=mid+1

        elif l[mid]>key :

            right=mid-1

        else:

            return mid

        mid=(left+right)//2

    return -1
```

```
#taking input in the list
n=int(input("Enter size of list "))
list=[]
for i in range(n) :
    list.append(int(input("Enter elements")))
key=int(input("Enter the key"))
list.sort();
print(list);
index=BinarySearch(list,key)
print("BINARY SEARCH HAS BEEN DONE")
if index==-1:
    print("key not exist's")
else:
    print("KEY INDEX=",index+1)
```

FILE

15Binarysearch.py

Output


Run Terminal Help
15Binarysearch.py - Mypython - Visual Studio Code
py
10PrimeNoInRange.py
5Binarysearch.py > ...
C:\WINDOWS\system32\cmd.exe

```
    while(left<=right):

        if l[mid]<key :
            left=mid+1
        elif l[mid]>key :
            right=mid-1
        else:
            return mid
        mid=(left+right)//2

    return -1


#taking input in the list
n=int(input("Enter size of lis
list=[]
for i in range(n) :
    list.append(int(input("Enter elements")))
key=int(input("Enter the key"))
list.sort();
print(list);
index=BinarySearch(list,key)
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 15BinarySearch.py
Enter size of list 5
Enter elements12
Enter elements10
Enter elements6
Enter elements7
Enter elements8
Enter the key6
[6, 7, 8, 10, 12]
BINARY SEARCH HAS BEEN DONE
KEY INDEX= 1

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 16

Write a program to implement following sorting algorithms

- **Selection Sort**
- **Insertion Sort**
- **Merge Sort**.

Aim: Write a program to implement following sorting algorithms

- Selection Sort
- Insertion Sort
- Merge Sort.

Parameter: list,n

Program

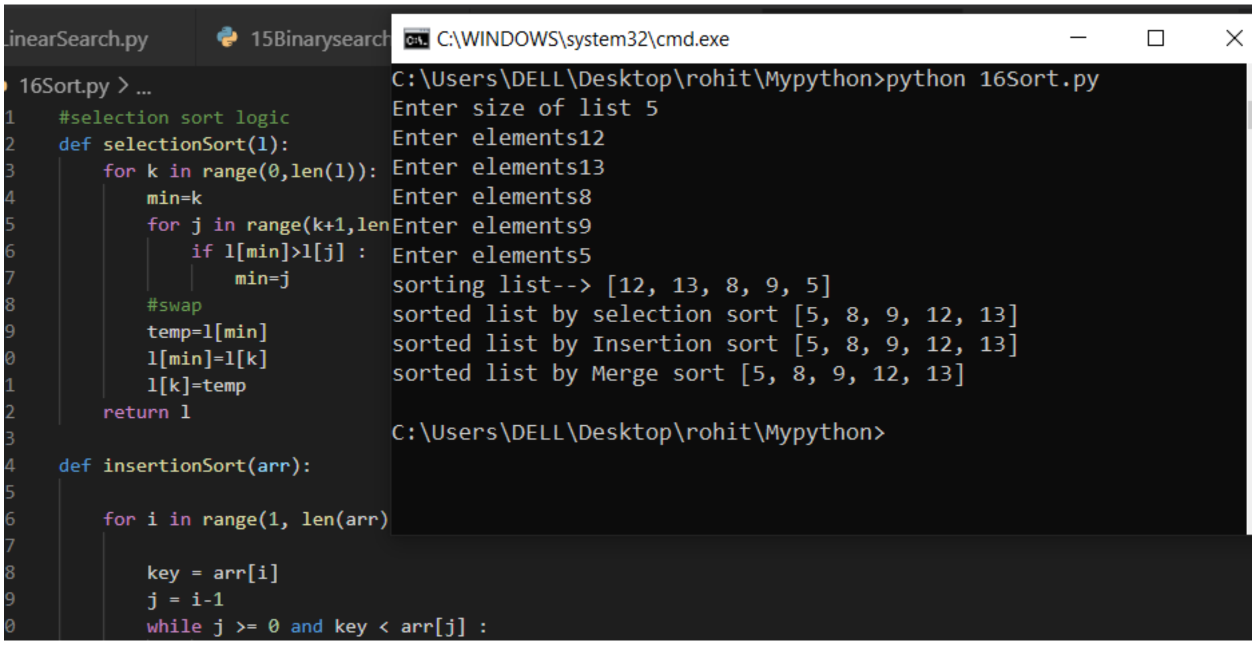
```
 #selection sort logic

def selectionSort(l):

    for k in range(0,len(l)):

        min=k

        for j in range(k+1,len(l)):

            if l[min]>l[j] :

                min=j

        #swap

        temp=l[min]

        l[min]=l[k]

        l[k]=temp

    return l


def insertionSort(arr):
    for i in range(1, len(arr)):

        key = arr[i]
        j = i-1
        while j >= 0 and key < arr[j] :
                arr[j + 1] = arr[j]
                j -= 1
        arr[j + 1] = key
    return arr

def mergeSort(arr,l,r):
    if l < r:
        m = (l+(r-1))//2
        mergeSort(arr, l, m)
        mergeSort(arr, m+1, r)
        merge(arr, l, m, r)
    return arr

def merge(arr,l,m,r):
     n1 = m - l + 1
     n2 = r- m
     L = [0] * (n1)
     R = [0] * (n2)
     for i in range(0 , n1):
        L[i] = arr[l + i]
     for j in range(0 , n2):
        R[j] = arr[m + 1 + j]
     i = 0
     j = 0
     k = l
     while i < n1 and j < n2 :
        if L[i] <= R[j]:
            arr[k] = L[i]
            i += 1
        else:
            arr[k] = R[j]
            j += 1
        k += 1
     while i < n1:
        arr[k] = L[i]
        i += 1
        k += 1
     while j < n2:
        arr[k] = R[j]
        j += 1
        k += 1

#taking input in the list
n=int(input("Enter size of list "))
```

```
list=[]

for i in range(n) :

    list.append(int(input("Enter elements")))

print("sorting list-->",list)

l=selectionSort(list)

print("sorted list by selection sort",l)

l=insertionSort(list)

print("sorted list by Insertion sort",l)

l=mergeSort(list,0,len(list)-1)

print("sorted list by Merge sort",l)
```

File

16Sort.py

Output



```
#selection sort logic
def selectionSort(l):
    for k in range(0,len(l)):
        min=k
        for j in range(k+1,len
            if l[min]>l[j] :
                min=j
        #swap
        temp=l[min]
        l[min]=l[k]
        l[k]=temp
    return l

def insertionSort(arr):

    for i in range(1, len(arr)

        key = arr[i]
        j = i-1
        while j >= 0 and key < arr[j] :
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 16Sort.py
Enter size of list 5
Enter elements12
Enter elements13
Enter elements8
Enter elements9
Enter elements5
sorting list--> [12, 13, 8, 9, 5]
sorted list by selection sort [5, 8, 9, 12, 13]
sorted list by Insertion sort [5, 8, 9, 12, 13]
sorted list by Merge sort [5, 8, 9, 12, 13]

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 17

Write a program to compute the number of characters, words and lines in a file.

Aim: Write a program to compute the number of characters, words and lines in a file.

Parameter: file.txt

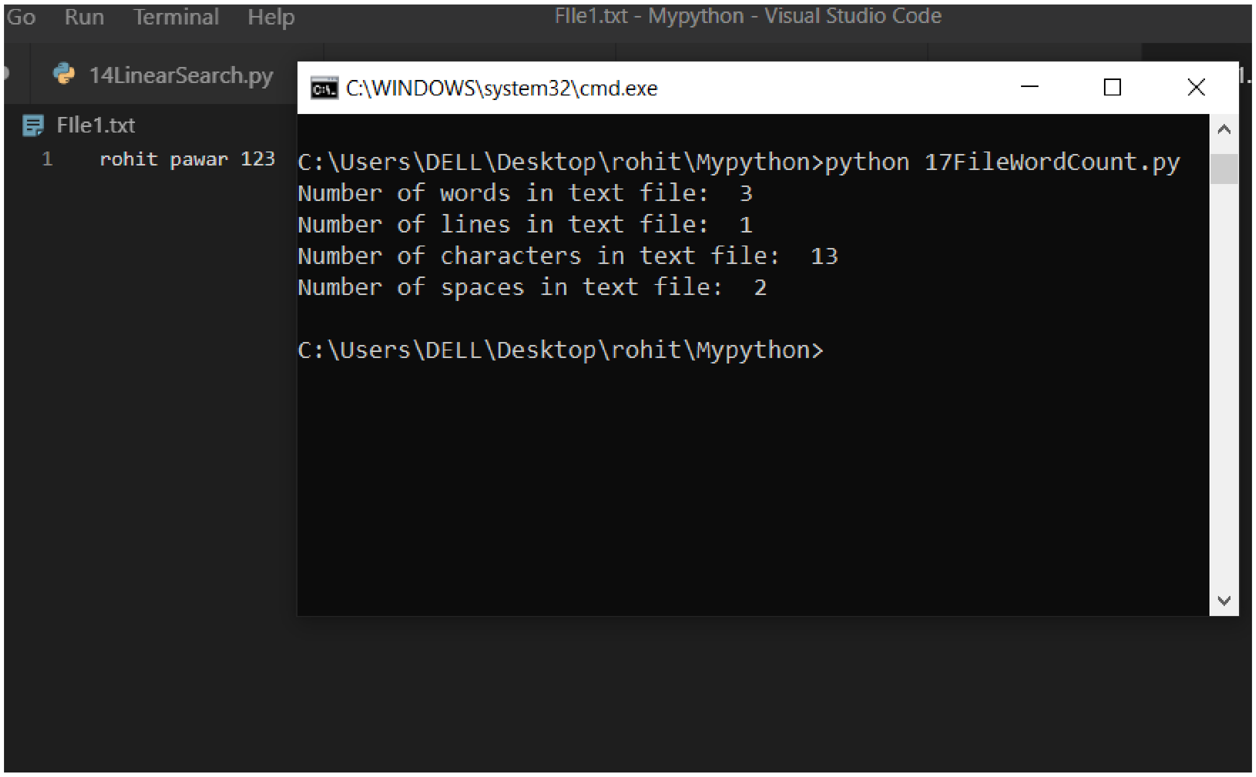
Program

```
def counter(fname):
    num_words = 0
    num_lines = 0
    num_charc = 0
     num_spaces = 0
    with open(fname, 'r') as f:
        for line in f:
             num_lines += 1
            word = 'Y'
            for letter in line:
                if (letter != ' ' and word == 'Y'):
                    num_words += 1
                    word = 'N'
                elif (letter == ' '):
                     num_spaces += 1
                    word = 'Y'
            for i in letter:
                if(i !=" " and i !="\n")
                    num_charc += 1
    print("Number of words in text file: ", num_words)


    print("Number of lines in text file: ", num_lines)


    print('Number of characters in text file: ', num_charc)


    print('Number of spaces in text file: ', num_spaces)



if __name__ == '__main__':
    fname = 'File1.txt'
    try:
        counter(fname)
    except:
        print('File not found')
```

File

17FileWordCount.py

Output


Go Run Terminal Help
FIle1.txt - Mypython - Visual Studio Code
14LinearSearch.py
FIle1.txt
C:\WINDOWS\system32\cmd.exe

```
rohit pawar 123
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 17FileWordCount.py
Number of words in text file:  3
Number of lines in text file:  1
Number of characters in text file:  13
Number of spaces in text file:  2

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 18

Write a program to implement stack using list.

Aim: Write a program to implement stack using list.

Parameter: list

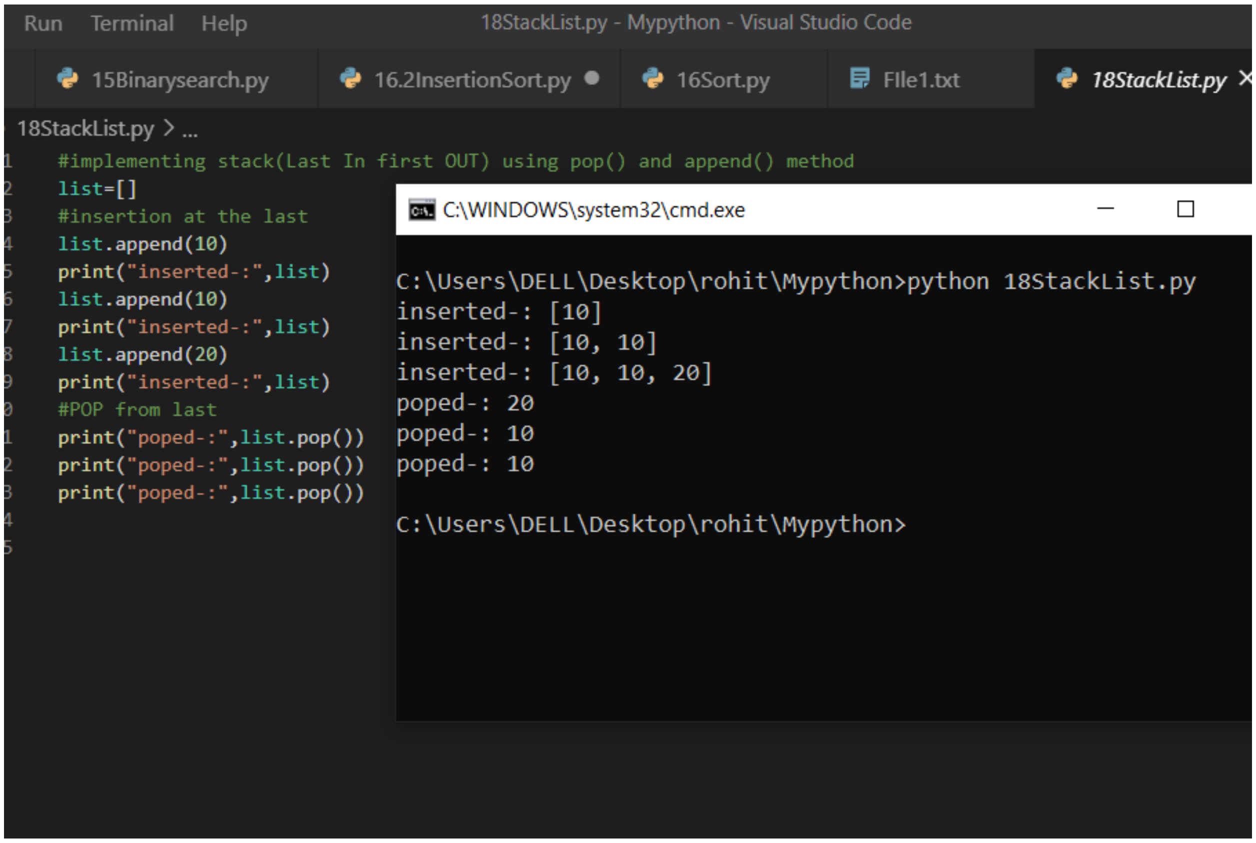
Program

```
 #implementing stack(Last In first OUT) using pop() and append() method
list=[]
#insertion at the last
list.append(10)
print("inserted-:",list)
list.append(10)
print("inserted-:",list)
list.append(20)
print("inserted-:",list)
#POP from last
print("poped-:",list.pop())
print("poped-:",list.pop())
print("poped-:",list.pop())
```

File

18StackList.py

Output



```
#implementing stack(Last In first OUT) using pop() and append() method
list=[]
#insertion at the last
list.append(10)
print("inserted-:",list)
list.append(10)
print("inserted-:",list)
list.append(20)
print("inserted-:",list)
#POP from last
print("poped-:",list.pop())
print("poped-:",list.pop())
print("poped-:",list.pop())
```

```
C:\Users\DELL\Desktop\rohit\Mypython>python 18StackList.py
inserted-: [10]
inserted-: [10, 10]
inserted-: [10, 10, 20]
poped-: 20
poped-: 10
poped-: 10

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 19

Write a program to implement queue using list.

Aim: Write a program to implement queue using list.

Parameter: queue

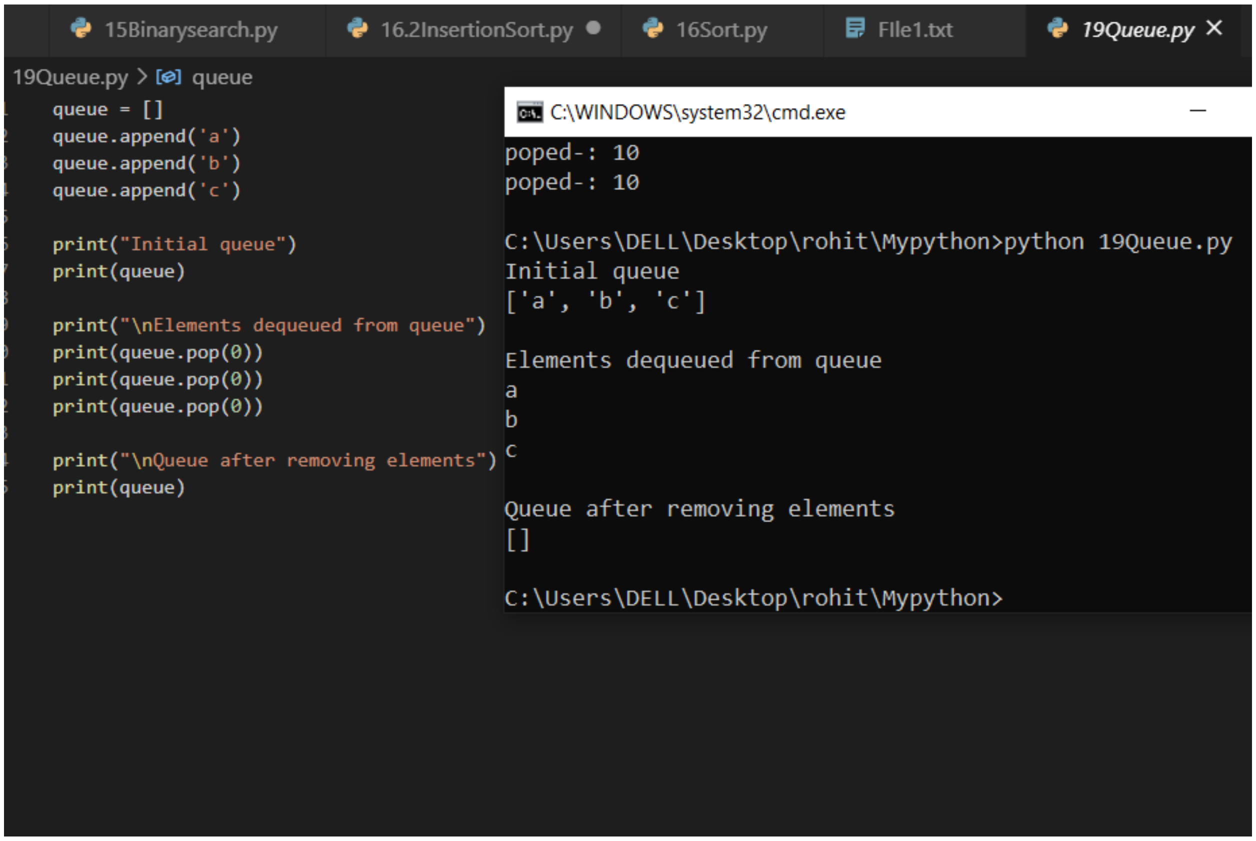
Program

```
 queue = []
queue.append('a')
queue.append('b')
queue.append('c')


print("Initial queue")
print(queue)


print("\nElements dequeued from queue")
print(queue.pop(0))
print(queue.pop(0))
print(queue.pop(0))


print("\nQueue after removing elements")
print(queue)
```

File

19Queue.py

Output



```python
queue = []
queue.append('a')
queue.append('b')
queue.append('c')

print("Initial queue")
print(queue)

print("\nElements dequeued from queue")
print(queue.pop(0))
print(queue.pop(0))
print(queue.pop(0))

print("\nQueue after removing elements")
print(queue)
```

```
poped-: 10
poped-: 10

C:\Users\DELL\Desktop\rohit\Mypython>python 19Queue.py
Initial queue
['a', 'b', 'c']

Elements dequeued from queue
a
b
c

Queue after removing elements
[]

C:\Users\DELL\Desktop\rohit\Mypython>
```

PRACTICAL 20

Write a program to demonstrate working of classes and objects.

Aim: Write a program to demonstrate working of classes and objects.

Parameter:- nothing

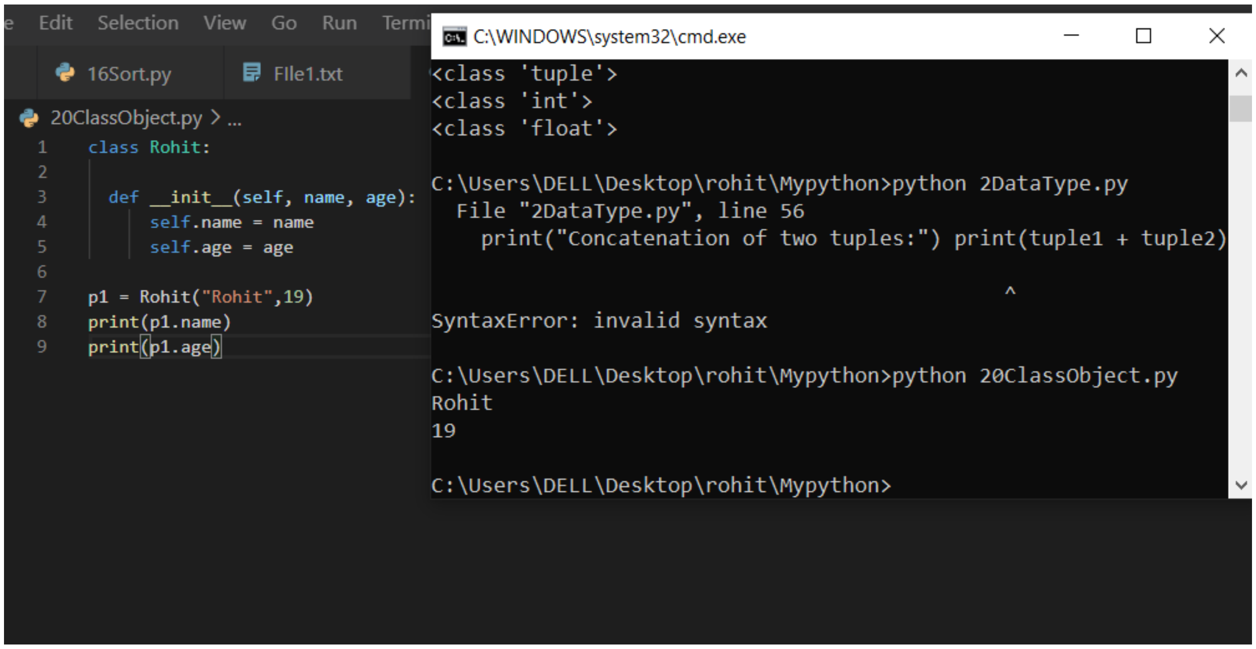
Program

```
 class Rohit:


 def __init__(self, name, age):
     self.name = name
     self.age = age


p1 = Rohit("Rohit",19)
print(p1.name)
print(p1.age)
```

File

20ClassObject.py

Output



```python
class Rohit:

    def __init__(self, name, age):
        self.name = name
        self.age = age

p1 = Rohit("Rohit",19)
print(p1.name)
print(p1.age)
```

```
<class 'tuple'>
<class 'int'>
<class 'float'>

C:\Users\DELL\Desktop\rohit\Mypython>python 2DataType.py
  File "2DataType.py", line 56
    print("Concatenation of two tuples:") print(tuple1 + tuple2)
                                              ^
SyntaxError: invalid syntax

C:\Users\DELL\Desktop\rohit\Mypython>python 20ClassObject.py
Rohit
19

C:\Users\DELL\Desktop\rohit\Mypython>
```