PILATHARA CO-OPERATIVE ARTS & SCIENCE COLLEGE

PILATHARA KANNUR- 670504

(AFFILIATED TO KANNUR UNIVERSITY)

PRACTICAL RECORD

PYTHON PROGRAMMING

NAME : ...Niranjana.c.v...

REG NO : ...PL21BCAR13...

SEMESTER : ...5th SEMESTER...

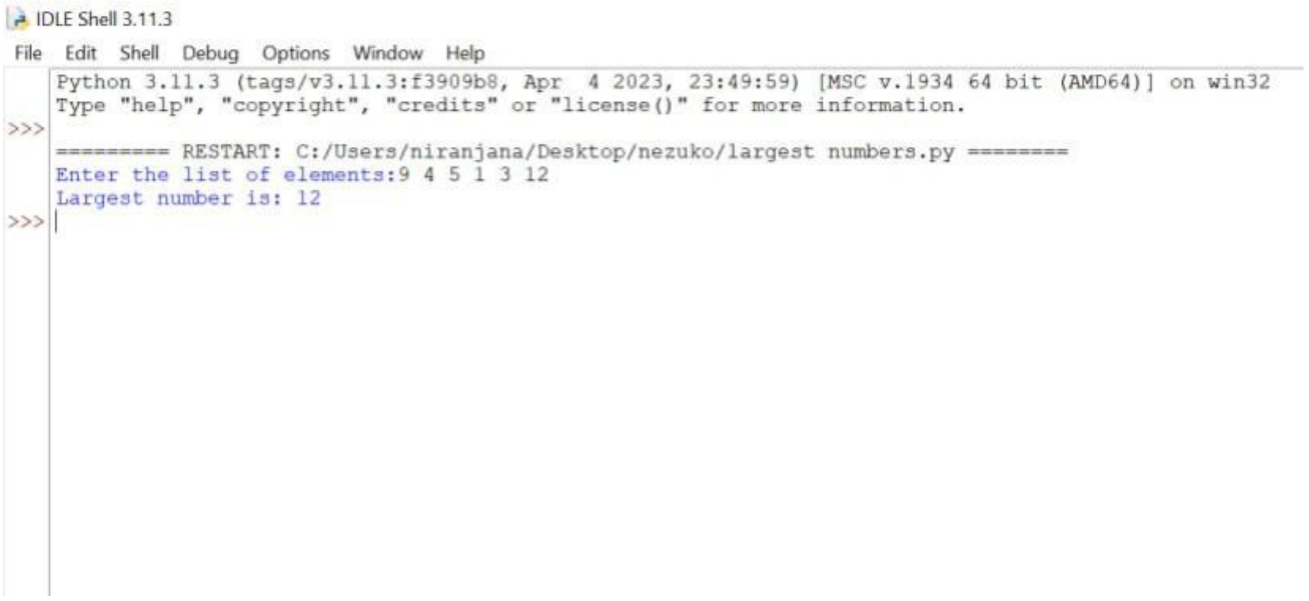
INDEX

1. Write a program to find the largest from a list of numbers.

PROGRAM

```
def find_large(numbers):
    return max(numbers)
a=input("Enter the list of elements:")
li=list(map(int,a.split( )))
print("Largest number is:",find_large(li))
```

OUTPUT



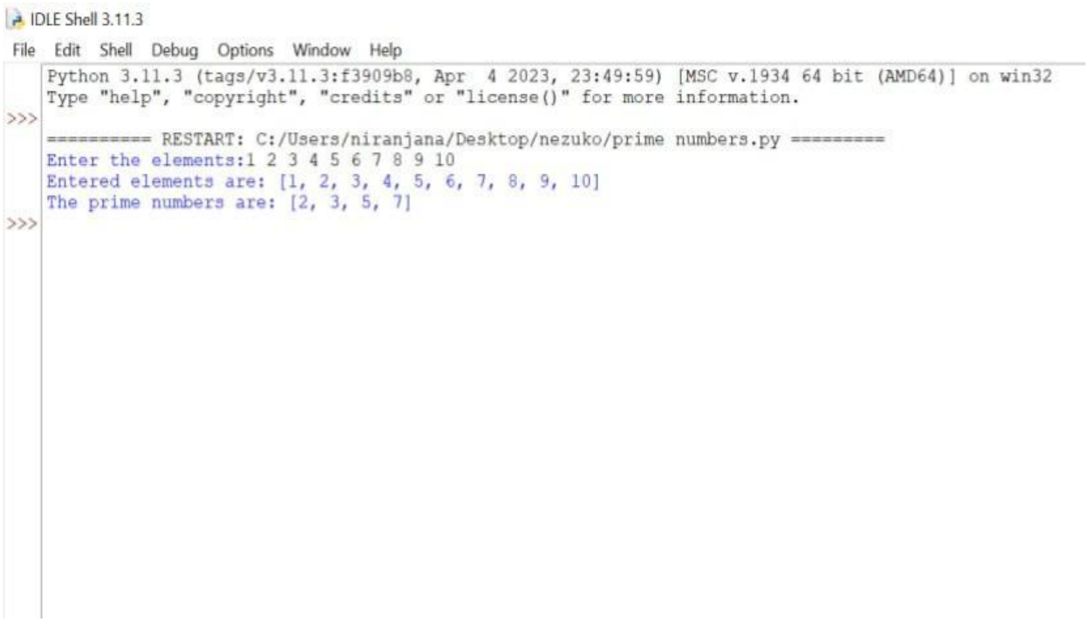
```
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:/Users/niranjana/Desktop/nezuko/largest numbers.py ========
Enter the list of elements:9 4 5 1 3 12
Largest number is: 12
>>>
```

2. Write a program to find the prime number in a list of numbers.

PROGRAM

```
str=(input("Enter the elements:"))
m=list(map(int,str.split( )))
print("Entered elements are:",m)
prime=[]
for i in m:
    c=0
    for j in range(1,i):
        if i%j==0:
            c+=1
    if c==1:
          prime.append(i)
print("The prime numbers are:",prime)
```

OUTPUT



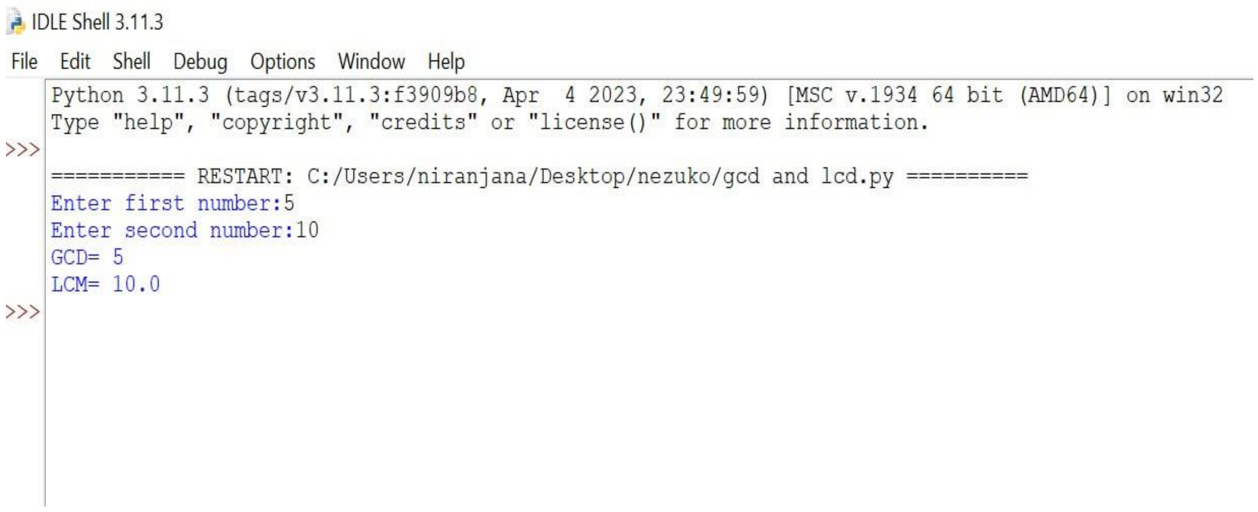
```
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/niranjana/Desktop/nezuko/prime numbers.py =========
Enter the elements:1 2 3 4 5 6 7 8 9 10
Entered elements are: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
The prime numbers are: [2, 3, 5, 7]
>>>
```

3.Write a program to find LCM and GCD of two numbers.

PROGRAM

```
a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
for i in range(1,max(a,b)+1):
    if a%i==0 and b%i==0:
        g=i
print("GCD=",g)
lcm=a*b/g
print("LCM=",lcm)
```

Output



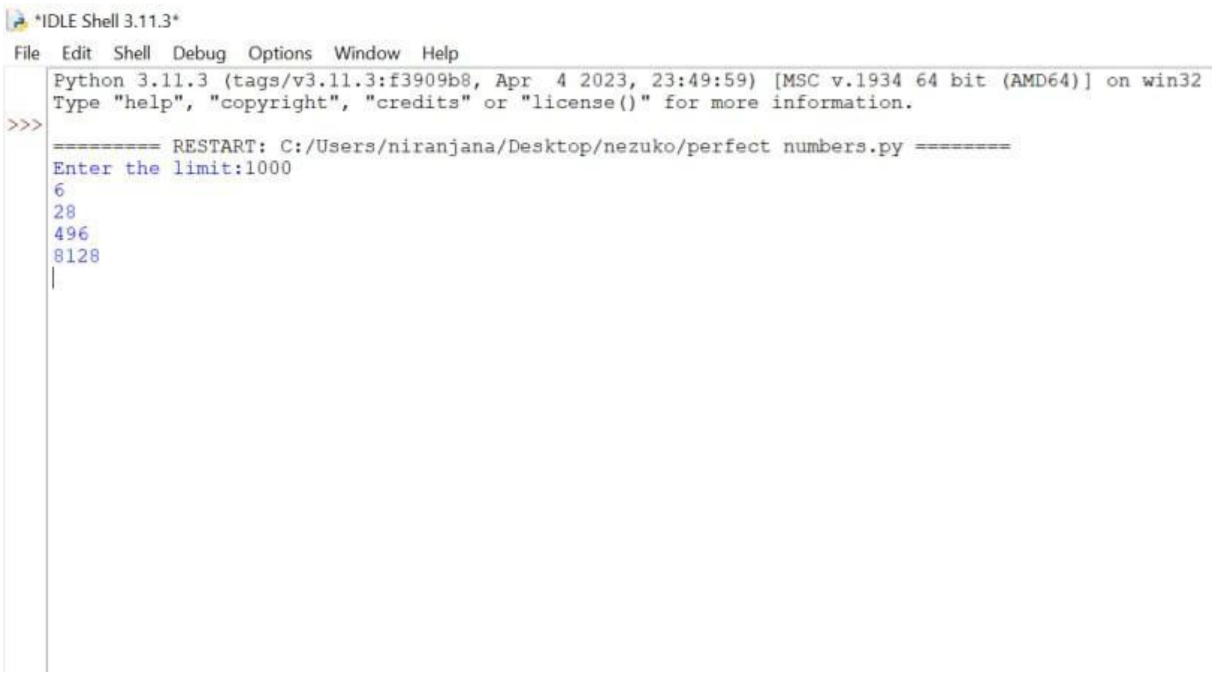
```
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=========== RESTART: C:/Users/niranjana/Desktop/nezuko/gcd and lcd.py ==========
Enter first number:5
Enter second number:10
GCD= 5
LCM= 10.0
>>>
```

4.Write a program to generate first n perfect numbers.

PROGRAM

```
Num=int(input("Enter the limit:"))
Count=i=0
While count<num:
   I=i+1
   S=0
   For item in range(1,i):
      If i%item==0:
         S=s+item
   If i==s:
      Print(i)
      Count=count+1
```

OUTPUT



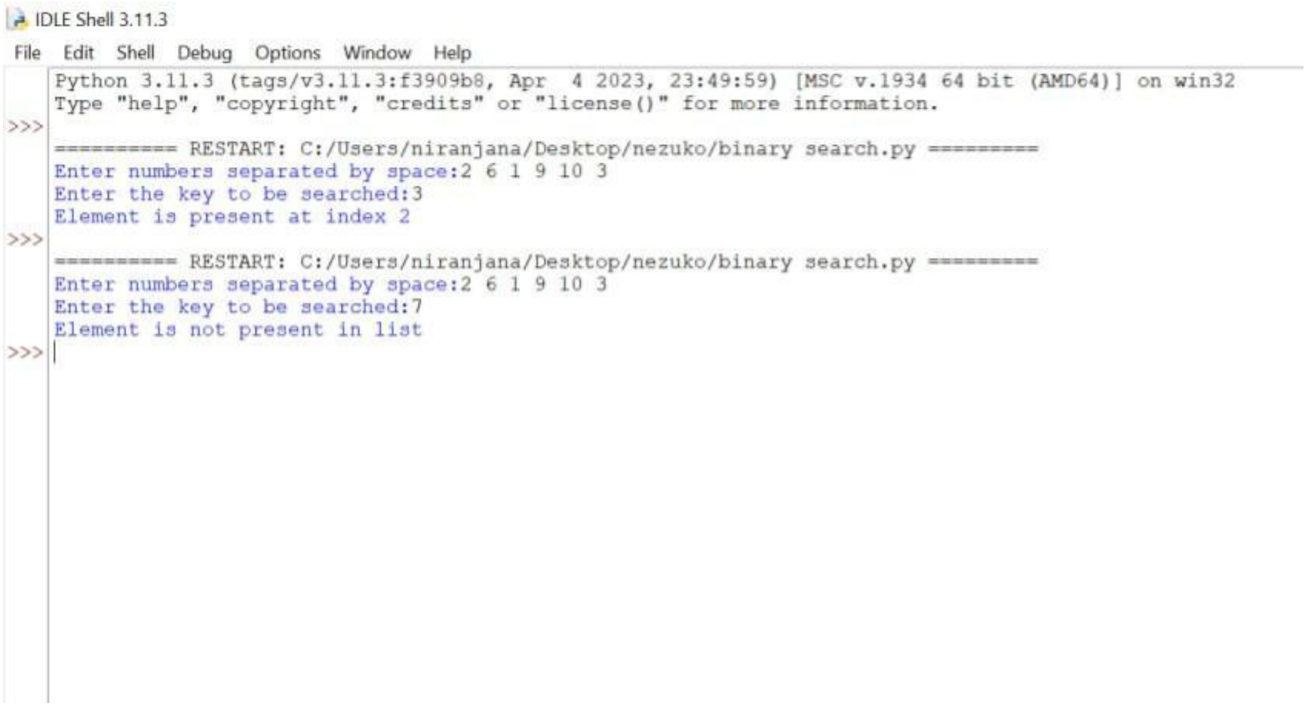
```
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:/Users/niranjana/Desktop/nezuko/perfect numbers.py ========
Enter the limit:1000
6
28
496
8128
```

5.Write a program to perform binary search.

PROGRAM

```
def binary_search(arr,x):
   low=0
   high=len(arr)-1
   while low<=high:
      mid=(high+low)//2
      if arr[mid]<x:
         low=mid+1
      elif arr[mid]>x:
         high=mid-1
      else:
         return mid
   return -1
li=input("Enter numbers separated by space:")
li=list(map(int,li.split()))
key=int(input("Enter the key to be searched:"))
li.sort()
result=binary_search(li,key)
if result!=-1:
      print("Element is present at index",str(result))
else:
   print("Element is not present in list")
```

OUTPUT



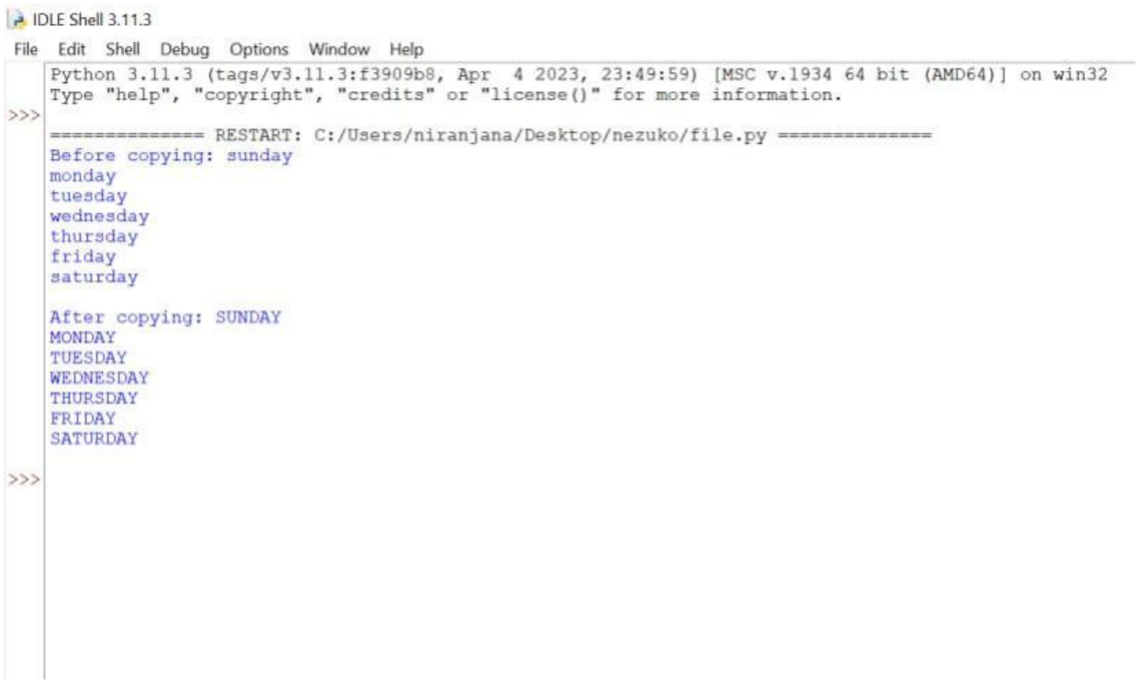
```
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/niranjana/Desktop/nezuko/binary search.py =========
Enter numbers separated by space:2 6 1 9 10 3
Enter the key to be searched:3
Element is present at index 2
>>>
========== RESTART: C:/Users/niranjana/Desktop/nezuko/binary search.py =========
Enter numbers separated by space:2 6 1 9 10 3
Enter the key to be searched:7
Element is not present in list
>>>
```

6.Write a program which reads the content of a file and copy the contents to another file after changing the entire letter to uppercase. Exceptions should be handled.

PROGRAM

```
try:
   f1=open("xyz.txt","r")
   f2=open("abc.txt","w")
   lines=f1.read()
   print("Before copying:",lines)
   lines=lines.upper()
   f2.write(lines)
   print("After copying:",lines)
   f1.close()
   f2.close()
except FileNotFoundError:
   print("FILE NOT FOUND")
except IOEroor:
   print("UNABLE TO COPY")
```

OUTPUT



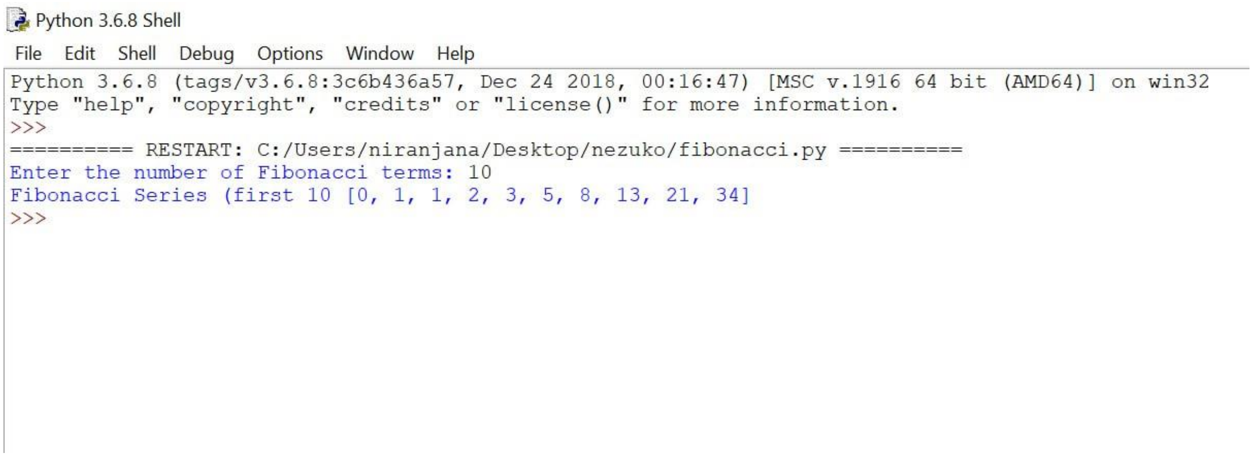
```
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:/Users/niranjana/Desktop/nezuko/file.py ==============
Before copying: sunday
monday
tuesday
wednesday
thursday
friday
saturday

After copying: SUNDAY
MONDAY
TUESDAY
WEDNESDAY
THURSDAY
FRIDAY
SATURDAY

>>>
```

7.Write a program to generate Fibonacci series using recursion

PROGRAM

```
def fibonacci(n):
    if n <= 0:
        return
    elif n == 1:
        return 0
    elif n == 2:
        return [0, 1]
    else:
        fib_list = fibonacci(n - 1)
        fib_list.append(fib_list[-1] + fib_list[-2])
        return fib_list
n = int(input("Enter the number of Fibonacci terms: "))
fib_series = fibonacci(n)
print("Fibonacci Series (first",n, fib_series)
```

OUTPUT



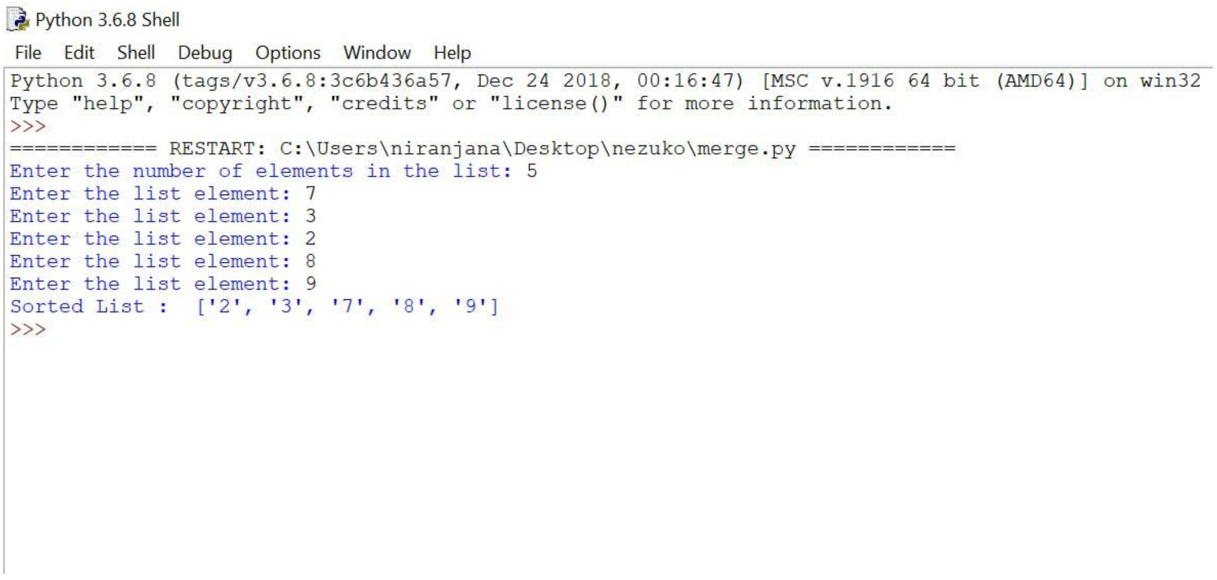
```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/niranjana/Desktop/nezuko/fibonacci.py ==========
Enter the number of Fibonacci terms: 10
Fibonacci Series (first 10 [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
>>>
```

8.Write a program to perform merge sort .

PROGRAM

```
Num=int(input("Enter the limit:"))
Count=i=0
While count<num:
   I=i+1
   S=0
   For item in range(1,i):
      If i%item==0:
         S=s+item
   If i==s:
      Print(i)
      Count=count+1
```

OUTPUT



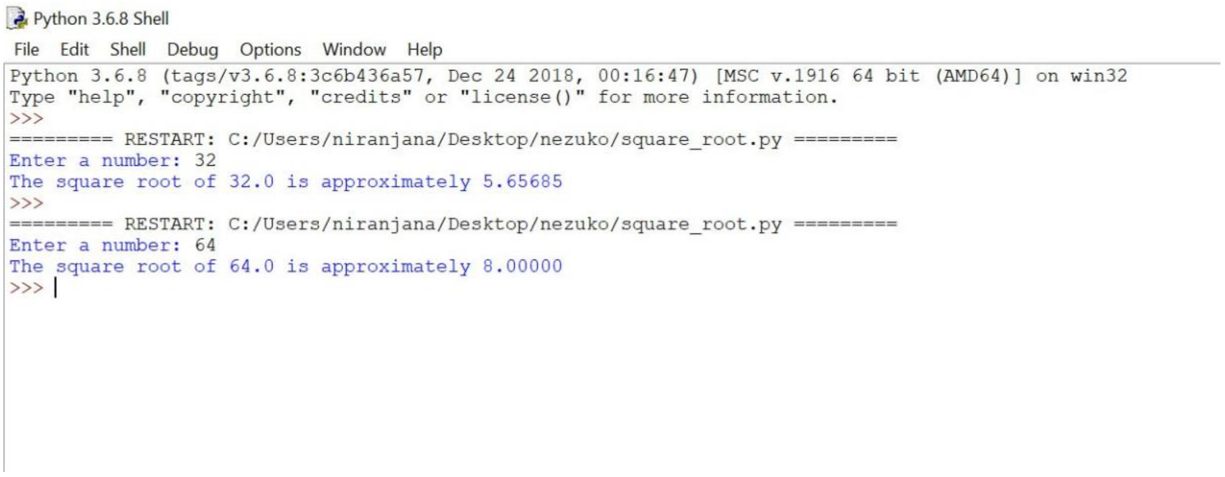
```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============ RESTART: C:\Users\niranjana\Desktop\nezuko\merge.py ============
Enter the number of elements in the list: 5
Enter the list element: 7
Enter the list element: 3
Enter the list element: 2
Enter the list element: 8
Enter the list element: 9
Sorted List :  ['2', '3', '7', '8', '9']
>>>
```

9.Write a program to find square root of a number using by-section method.

PROGRAM

```
def find_square_root(number, precision=0.00001):
    if number<0:
        raise ValueError("Square root is not defined for negative numbers")
    if number==0:
        return 0
    low = 0
    high = max(1, number)
    guess = (low + high) / 2
    while abs(guess**2 - number) > precision:
        if guess**2 < number:
            low = guess
        else:
            high = guess
        guess = (low + high) / 2
    return guess
number = float(input("Enter a number: "))
precision = 0.00001
result = find_square_root(number, precision)
print(f"The square root of {number} is approximately {result:.5f}")
```

OUTPUT



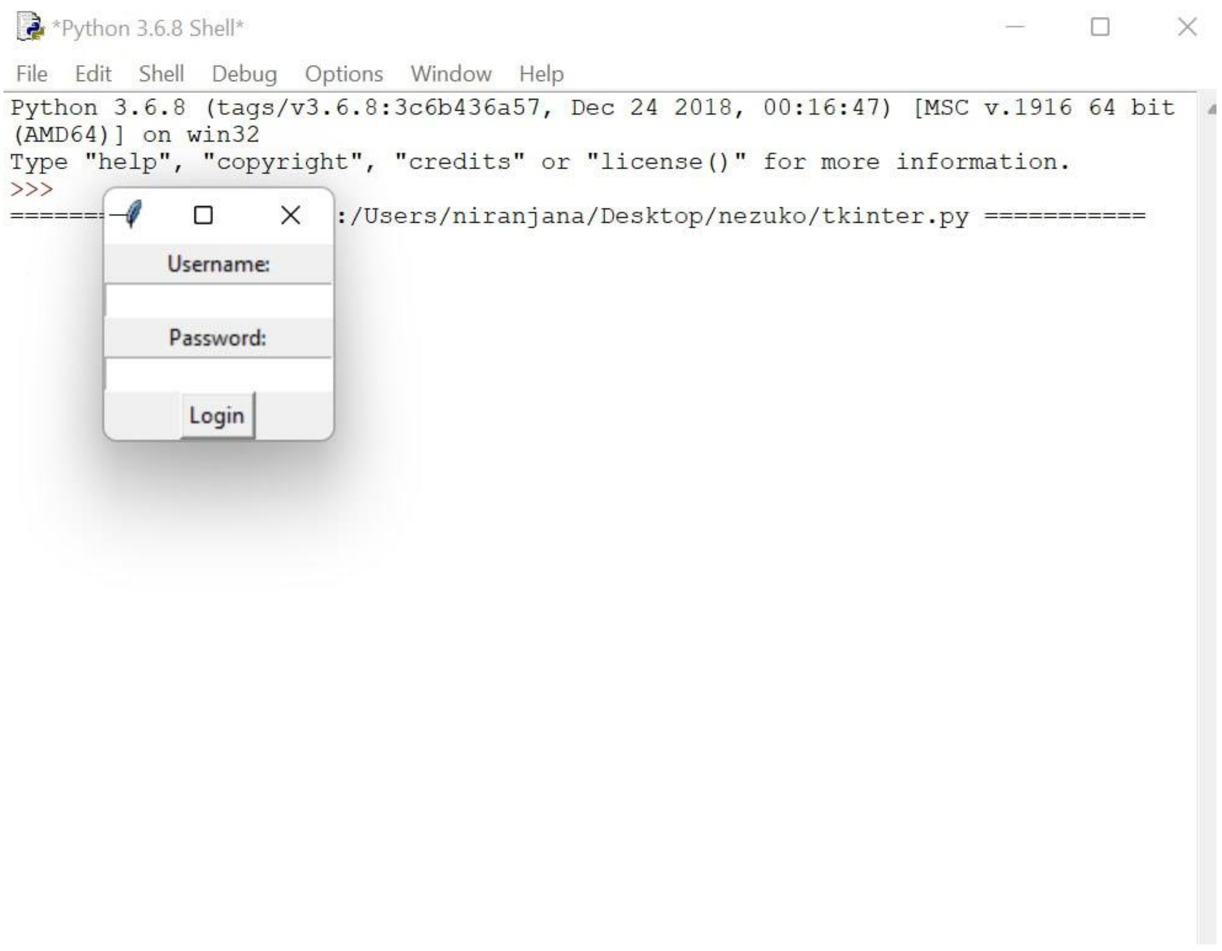
```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:/Users/niranjana/Desktop/nezuko/square_root.py =========
Enter a number: 32
The square root of 32.0 is approximately 5.65685
>>>
========= RESTART: C:/Users/niranjana/Desktop/nezuko/square_root.py =========
Enter a number: 64
The square root of 64.0 is approximately 8.00000
>>>
```

10.**Write a simple log in window using tkinter.**

PROGRAM

```
import tkinter as tk
from tkinter import messagebox
def check_login():
    username = username_entry.get()
    password = password_entry.get()
    if username == "your_username" and password == "your_password":
        messagebox.showinfo("Login", "Login successful!")
    else:
        messagebox.showerror("Login Error", "Invalid username or 
password")
root = tk.Tk()
root.title("Login Window")
username_label = tk.Label(root, text="Username:")
username_label.pack()
username_entry = tk.Entry(root)
username_entry.pack()
password_label = tk.Label(root, text="Password:")
password_label.pack()
password_entry = tk.Entry(root, show="*")
password_entry.pack()
login_button = tk.Button(root, text="Login", command=check_login)
login_button.pack()
root.mainloop()
```

OUTPUT



```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======                :/Users/niranjana/Desktop/nezuko/tkinter.py ===========
```

11.**Write a program to perform the following.**

a) Create table student with fields name,sex,rollno,mark

b) Insert some rows into table

c) Update the marks of all students by adding 2 marks

d) Delete a student with a given roll no

e) Display the details of a student with a given roll no

PROGRAM

A)<u>Creating Table</u>

```
import pymysql as ps;
try:
   db=ps.connect(host=”localhost”,
           user=”root”,
           password=”chinchuna”,
           database=”sample”)
except:
   print(“db connection error”)
try:
   cursor=db.cursor()
   sql=”create table if not exists student(name varchar(20), sex
varchar(10),regno int,mark int)”
   cursor.execute(sql)
   db.commit()
   print(“db created”)
except:
   print(“db creation error”)
   db.rollback()
db.close()
```

B)<u>Insert Operation</u>

```
Import pymysql as ps
Try:
   Db=ps.connect(host=”localhost”,
           User=”root”,
           Password=”chinchuna”,
           Database=”sample”)
Except:
   Print(“db connection error”)
Try:
   Cursor=db.cursor()
   While(True):
      Name=input(“Enter the name of student : “)
      Gender=input(“Enter the gender of student”)
      Regno=int(input(“Enter the register number of the student: “))
      Mark=int(input(“Enter the mark of student : “))
      Ins_sql=”insert into stud values(‘{}’,’{}’,{},{})”.format(name,gender,regno,mark)
      Cursor.execute(ins_sql)
      Db.commit()
     Print(“one row inserted”)
```

```
        Resp=input("Do you want to continue one more
record(Y/N):")
        Resp=resp.lower()
        If(resp!='y'):
           Break
Except:
   Print("can not insert a row")
```

C) <u>UpdateOperation</u>

```
Import pymysql as ps
Db=ps.connect(host="localhost",
            User="root",
            Password="",
            Database="sample")
Cursor = db.cursor()
Update_query = "update student set Mark = Mark + 2"
Cursor.execute(update_query)
Db.commit()
Updated_count = cursor.rowcount
Print(f"{updated_count} student' marks updated successfully.")
Db.close()
```

D) <u>Delete Operation</u>

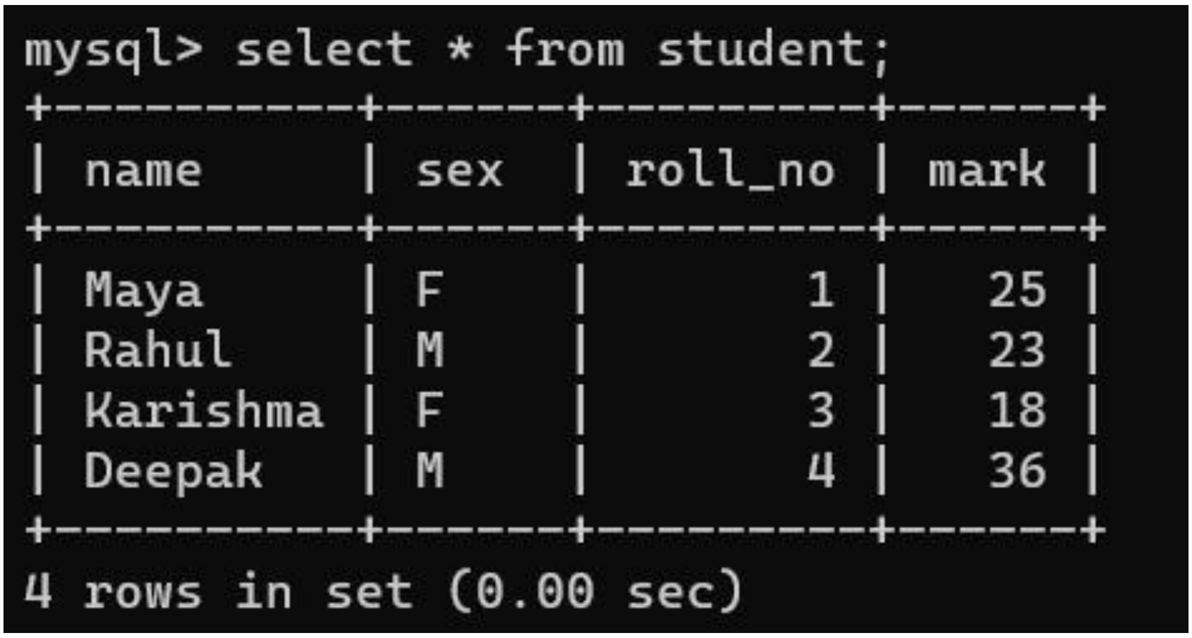
```
Import pymysql as ps
Db=ps.connect(host="localhost",
            User=”root”,
            Password=””,
            Db=”sample”)
Cursor = db.cursor()
Roll_no = input(“Enter the roll number of the student to delete: 
“)
Delete_query = “delete from student where roll_no=%s”
Cursor.execute(delete_query,(roll_no))
Db.commit()
If cursor.rowcount > 0:
   Print(f”Student with roll number {roll_no} deleted 
successfully.”)
Else:
   Print(f”No student found with roll number {roll_no}.”)
Db.close()
Print(“MySQL connection closed.”)
```

E) Display Operation

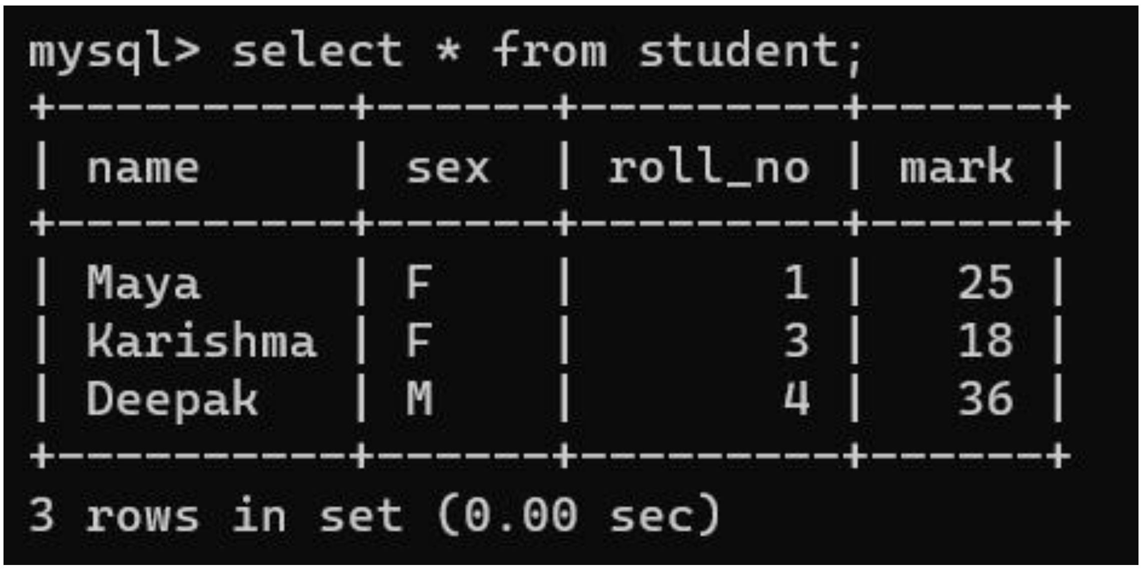
```
Import pymysql as ps
Try:
   Db=ps.connect(host=”localhost”,
              User=”root”,
             Password=””,
             Database=”sample”
              )
   Cursor = db.cursor()
```

```
    Roll_no= input("Enter the roll number of the student to
display: ")
    Select_query = "select * from student where roll_no = %s"
    Cursor.execute(select_query, (roll_no))
    Student = cursor.fetchone()
    If student:
        Print("Student Details:")
        Print(f"Name: {student[0]}")
        Print(f"sex:{student[1]}")
        Print(f"Roll_no: {student[2]}")
        Print(f"Marks: {student[3]}")
    Else:
        Print("No student found with roll number {roll_no}.")

Except ps.Error as error:
    Print(f"Error: {error}")
Finally:
    If db:
        Db.close()
        Print("MySQL connection closed.")
```

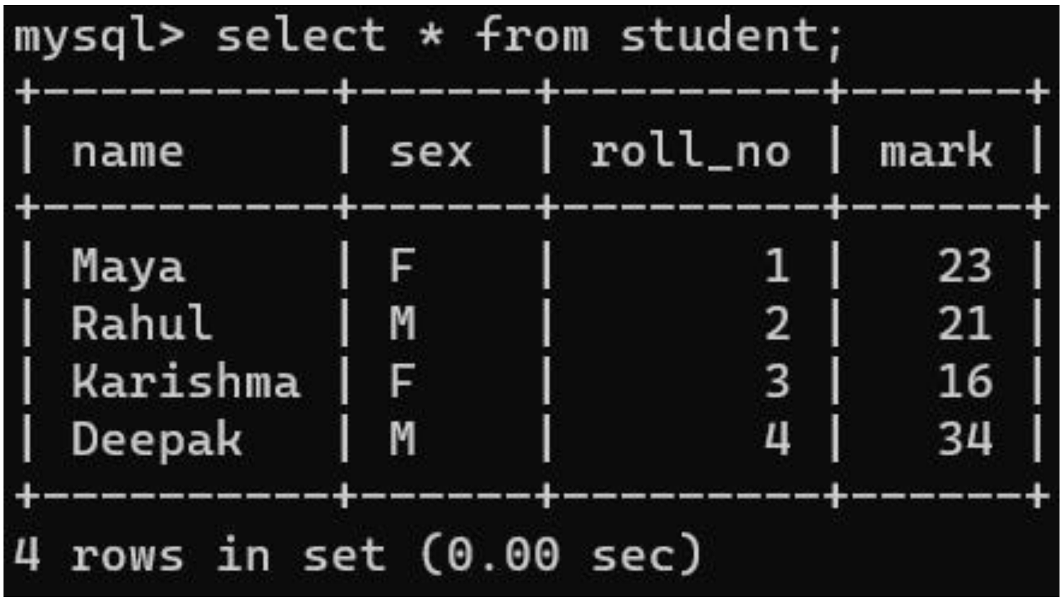
OUTPUT



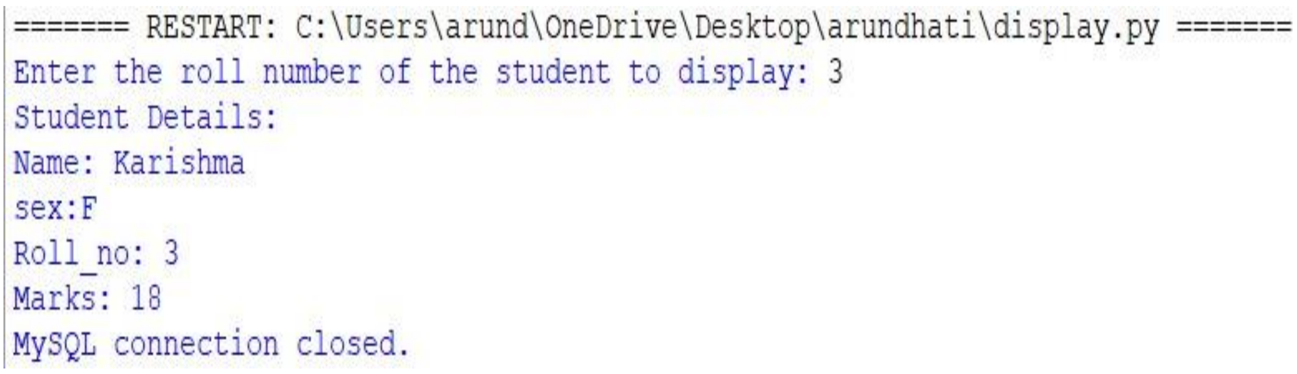
```
mysql> select * from student;
+----------+------+---------+------+
| name     | sex  | roll_no | mark |
+----------+------+---------+------+
| Maya     | F    |       1 |   25 |
| Rahul    | M    |       2 |   23 |
| Karishma | F    |       3 |   18 |
| Deepak   | M    |       4 |   36 |
+----------+------+---------+------+
4 rows in set (0.00 sec)
```



```
mysql> select * from student;
+----------+------+---------+------+
| name     | sex  | roll_no | mark |
+----------+------+---------+------+
| Maya     | F    |       1 |   25 |
| Karishma | F    |       3 |   18 |
| Deepak   | M    |       4 |   36 |
+----------+------+---------+------+
3 rows in set (0.00 sec)
```



```
mysql> select * from student;
+----------+------+---------+------+
| name     | sex  | roll_no | mark |
+----------+------+---------+------+
| Maya     | F    |       1 |   23 |
| Rahul    | M    |       2 |   21 |
| Karishma | F    |       3 |   16 |
| Deepak   | M    |       4 |   34 |
+----------+------+---------+------+
4 rows in set (0.00 sec)
```



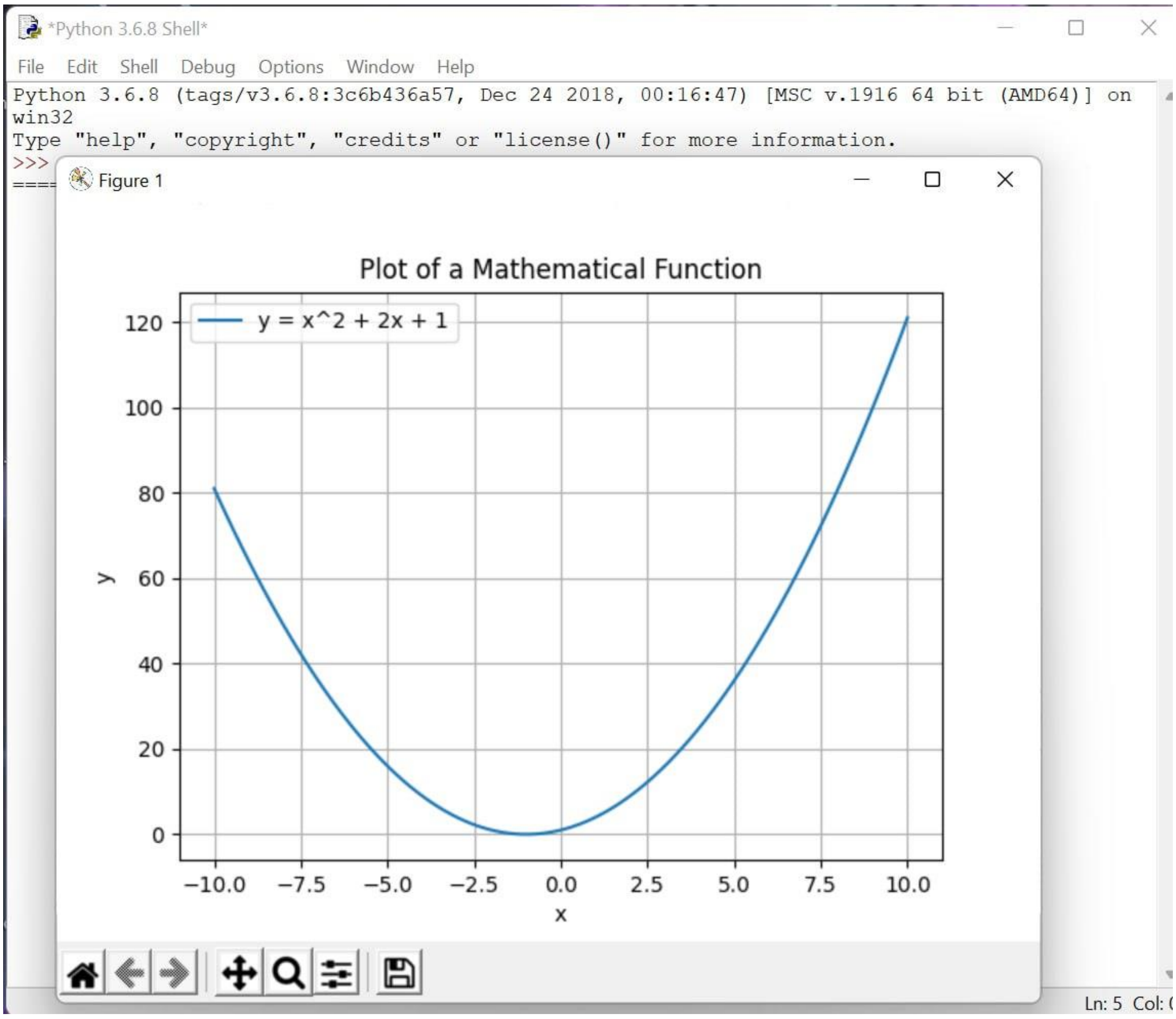
```
======= RESTART: C:\Users\arund\OneDrive\Desktop\arundhati\display.py =======
Enter the roll number of the student to display: 3
Student Details:
Name: Karishma
sex:F
Roll_no: 3
Marks: 18
MySQL connection closed.
```

12. **Create a plot for the mathematical function.**

PROGRAM

```
import matplotlib.pyplot as plt
import numpy as np
def my_function(x):
    return x**2 + 2*x + 1
x = np.linspace(-10, 10, 100)
y = my_function(x)
plt.plot(x, y, label='y = x^2 + 2x + 1')
plt.title('Plot of a Mathematical Function')
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.legend()
plt.show()
```

OUTPUT



```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on
win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
====
```