PILATHARA CO-OPERATIVE ARTS & SCIENCE COLLEGE

PILATHARA KANNUR- 670504

(AFFILIATED TO KANNUR UNIVERSITY)

PRACTICAL RECORD

PYTHON PROGRAMMING

NAME : HRITHIK.S.RAVEENDRAN

REG NO : PL21BCAR17

SEMESTER : 5th SEMESTER

PILATHARA CO-OPERATIVE ARTS AND SCIENCE COLLEGE

PILATHARA KANNUR- 670504

(AFFILIATED TO KANNUR UNIVERSITY)

PRACTICAL RECORD

CERTIFICATE

CERTIFIED THAT THIS IS A BONAFIDE RECORD OF THE ORIGINAL WORK DONE BY ....

REG. NO...................................... OF 3rd BCA IN PYTHON PROGRAMMING

DURING THE YEAR 2023-2024.

EXAMINERS: **LECTURER IN CHARGE:**

1.

2.

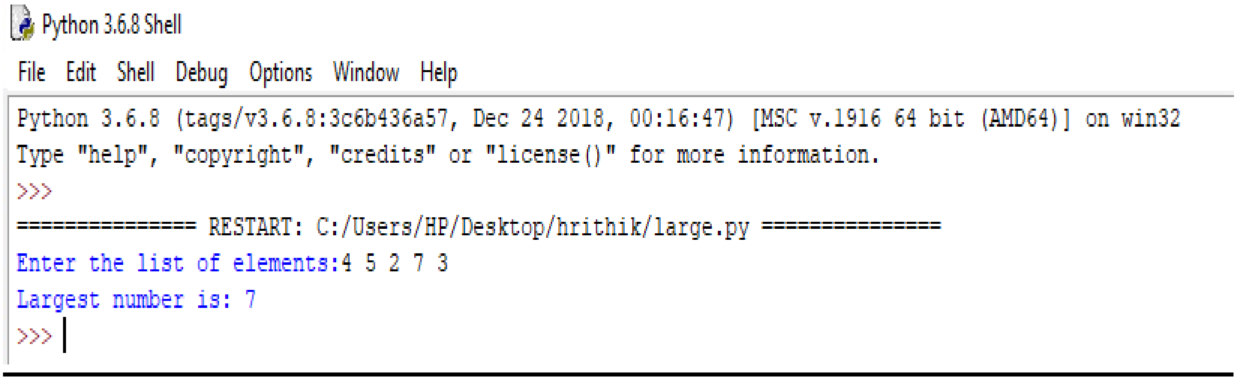
INDEX

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



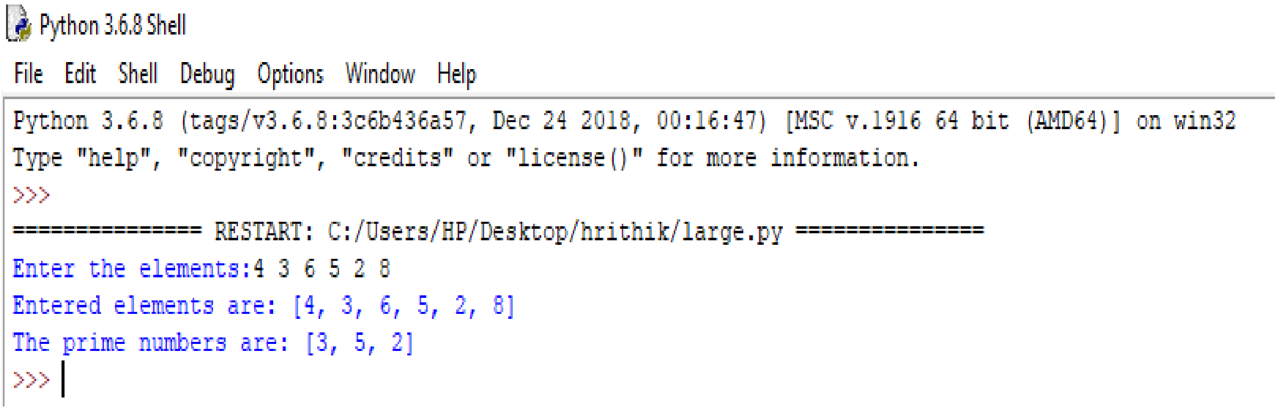
```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:/Users/HP/Desktop/hrithik/large.py ===============
Enter the list of elements:4 5 2 7 3
Largest number is: 7
>>>
```

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



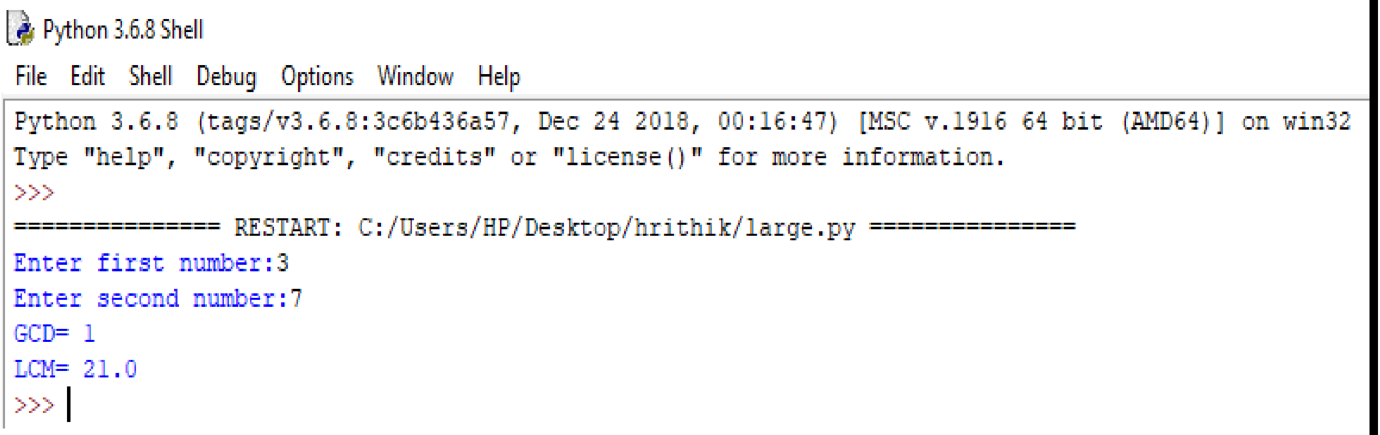
```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:/Users/HP/Desktop/hrithik/large.py ===============
Enter the elements:4 3 6 5 2 8
Entered elements are: [4, 3, 6, 5, 2, 8]
The prime numbers are: [3, 5, 2]
>>> |
```

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



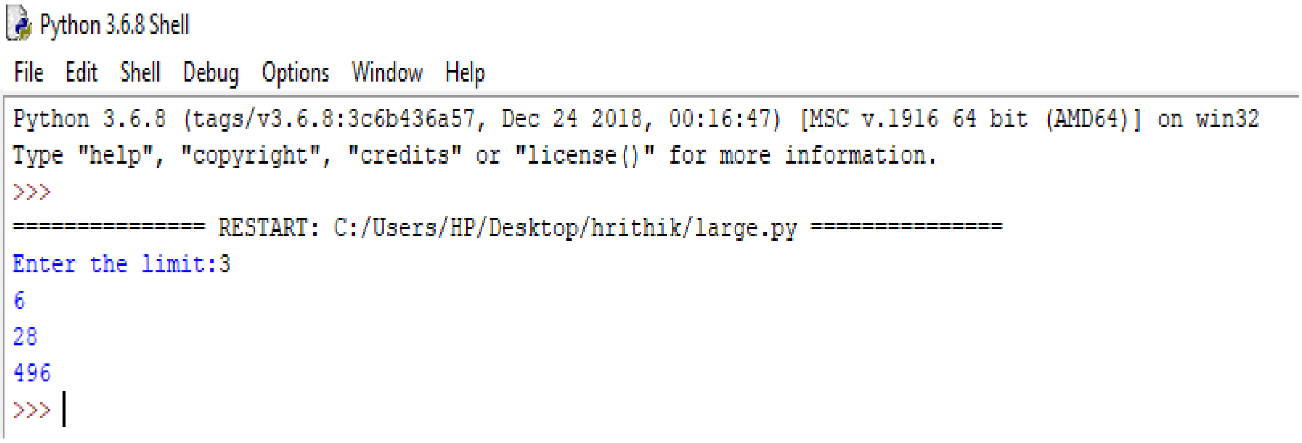
```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
=============== RESTART: C:/Users/HP/Desktop/hrithik/large.py ===============
Enter first number:3
Enter second number:7
GCD= 1
LCM= 21.0
>>> 
```

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



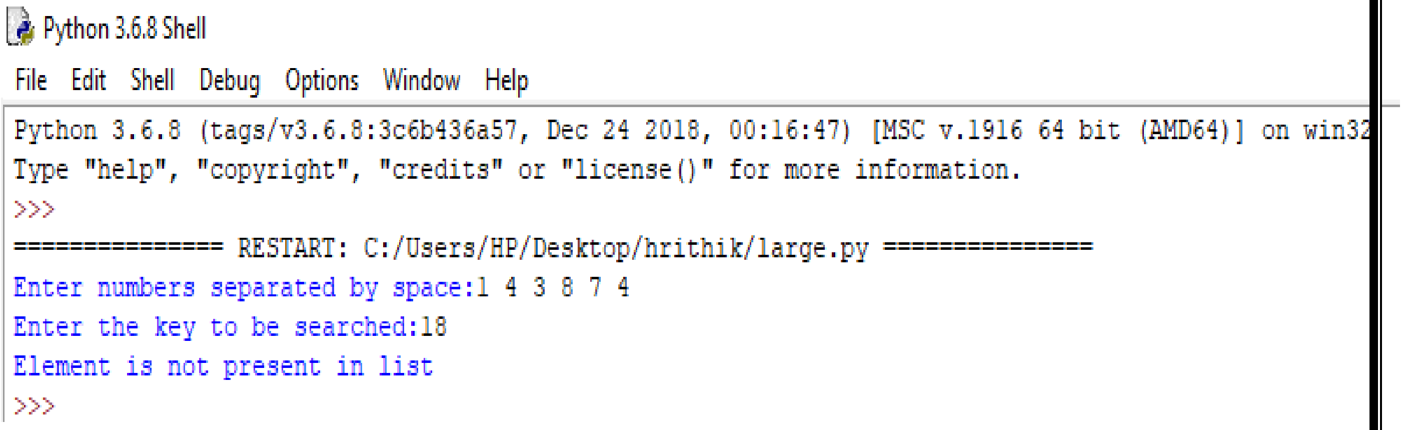
```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:/Users/HP/Desktop/hrithik/large.py ===============
Enter the limit:3
6
28
496
>>> 
```

5. Write a program to perform binary search.

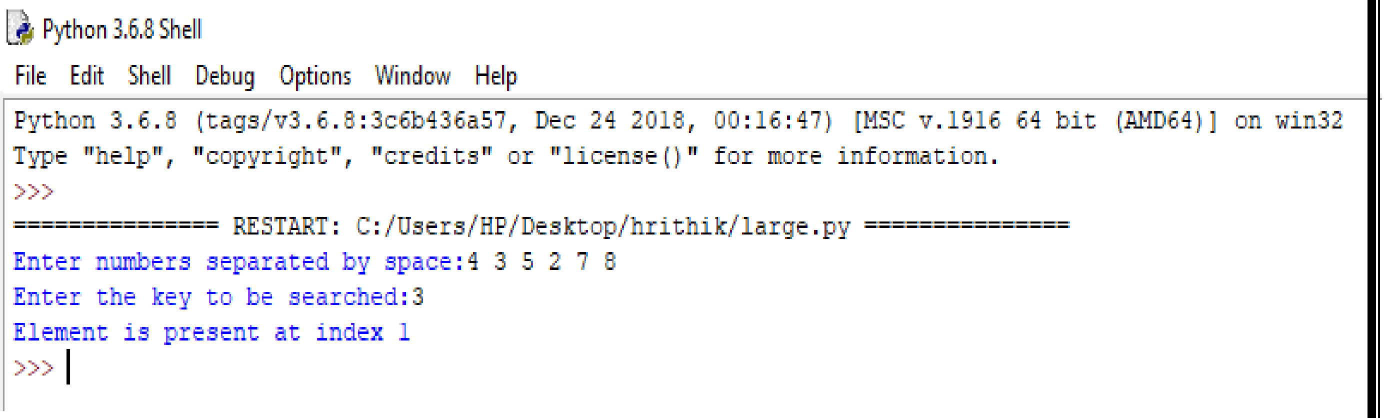
PROGRAM

```
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 3.6.8 (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/HP/Desktop/hrithik/large.py ===============
Enter numbers separated by space:1 4 3 8 7 4
Enter the key to be searched:18
Element is not present in list
>>> 
```



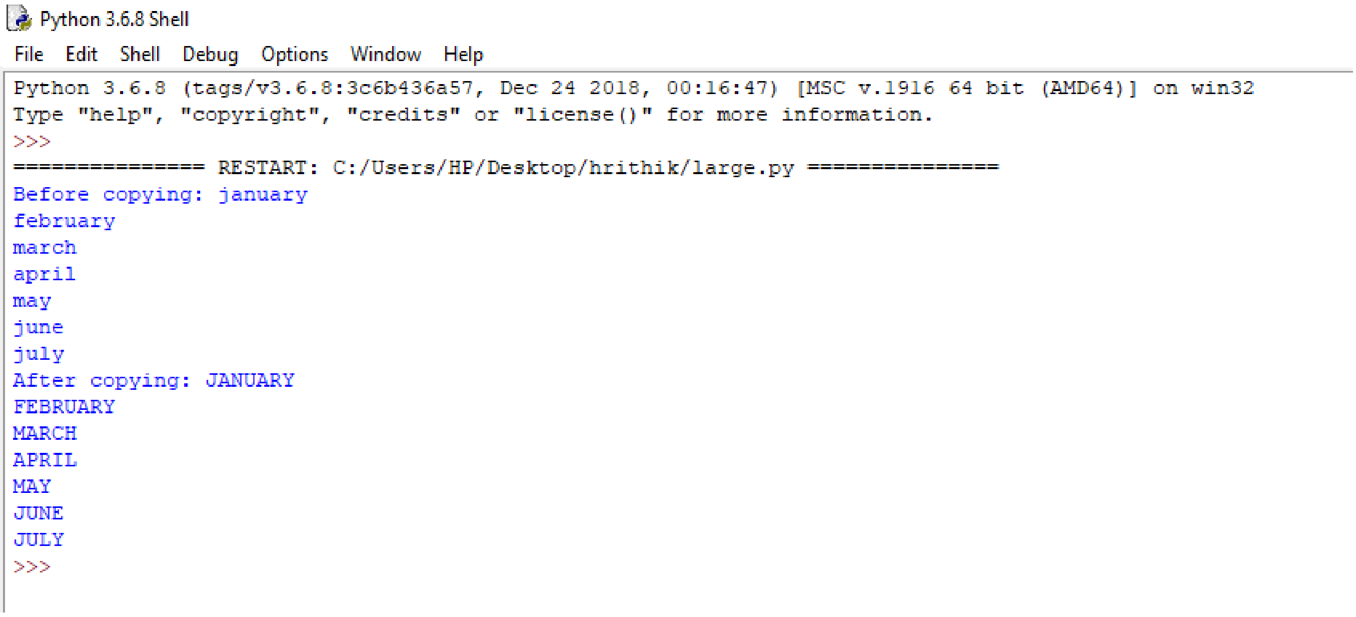
```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
=============== RESTART: C:/Users/HP/Desktop/hrithik/large.py ===============
Enter numbers separated by space:4 3 5 2 7 8
Enter the key to be searched:3
Element is present at index 1
>>> 
```

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



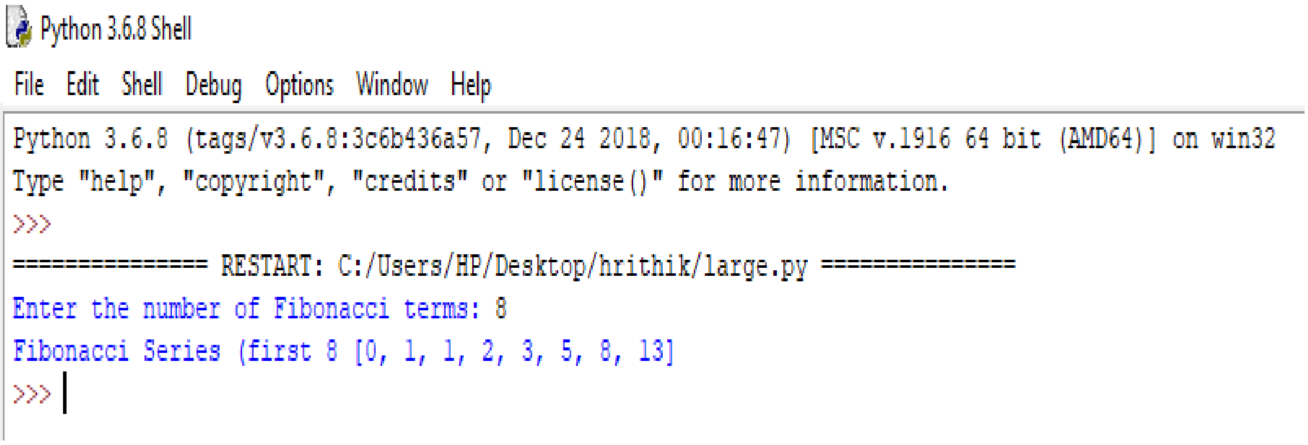
```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:/Users/HP/Desktop/hrithik/large.py ===============
Before copying: january
february
march
april
may
june
july
After copying: JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
JULY
>>>
```

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



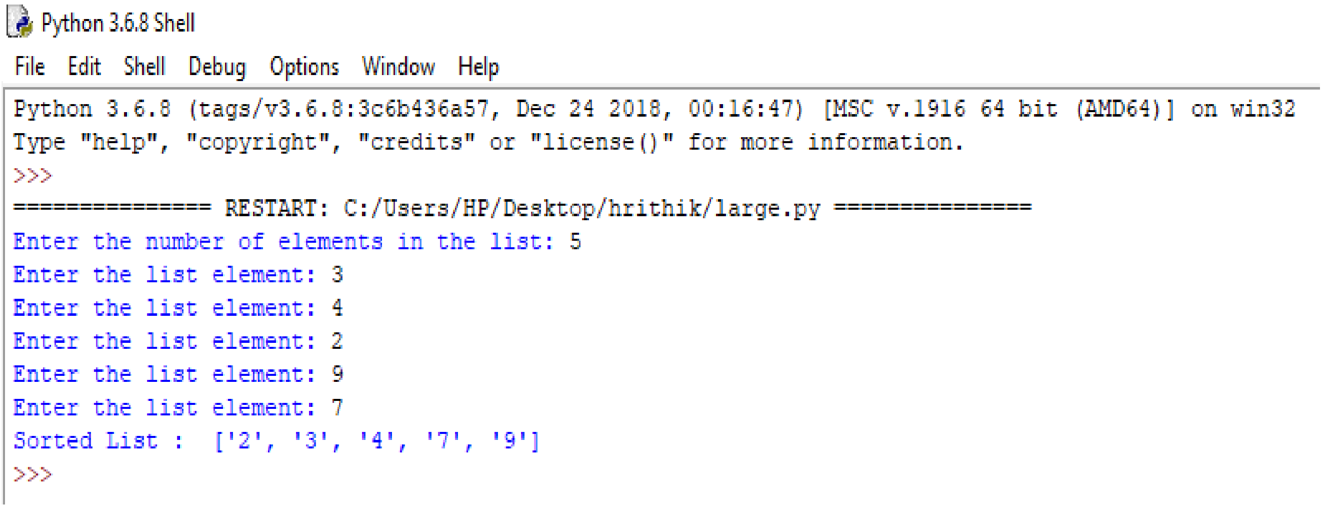
```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:/Users/HP/Desktop/hrithik/large.py ===============
Enter the number of Fibonacci terms: 8
Fibonacci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

8. **Write a program to perform merge sort .**

```
def merge(left,right):
      output=[]
      i=j=0
      while i<len(left) and j<len(right):
          if left[i]<right[j]:
              output.append(left[i])
              i=i+1
          else:
              output.append(right[j])
              j=j+1
          output.extend(left[i:])
          output.extend(right[j:])
          return output
def mergesort(lis):
    list_length=len(lis)
    if list_length==1:
        return lis
    mid_point=list_length//2
    left_partition=mergesort(lis[:mid_point])
    right_partition=mergesort(lis[mid_point:])
    return merge(left_partition,right_partition)
li=[]
n=int(input("Enter the number of elements in the list: "))
```

```
i=0
while i<n:
    a=input("Enter the list element: ")
    li.append(a)
    i=i+1
s=mergesort(li)
print("Sorted List : ",s)
```

OUTPUT



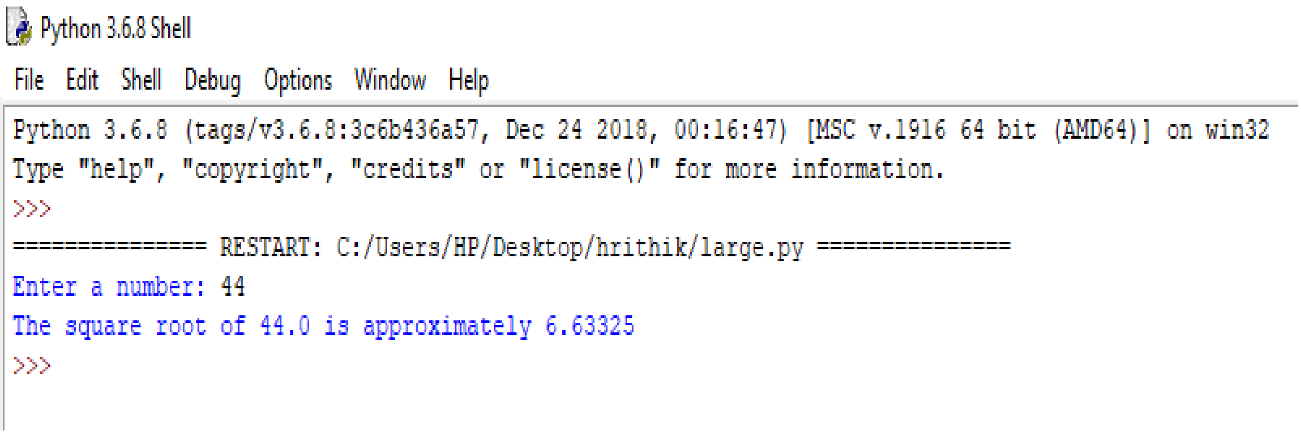
```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
=============== RESTART: C:/Users/HP/Desktop/hrithik/large.py ===============
Enter the number of elements in the list: 5
Enter the list element: 3
Enter the list element: 4
Enter the list element: 2
Enter the list element: 9
Enter the list element: 7
Sorted List :  ['2', '3', '4', '7', '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



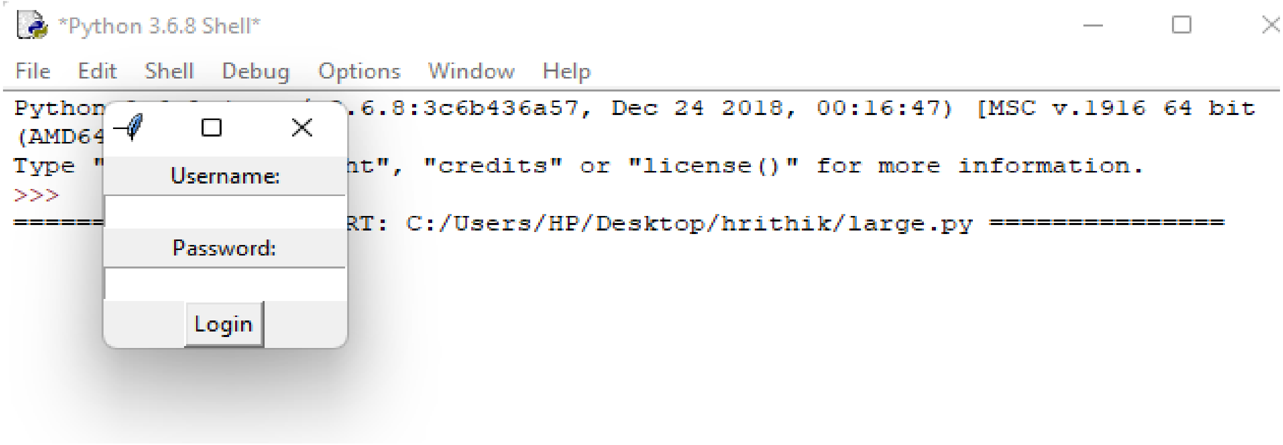
```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
=============== RESTART: C:/Users/HP/Desktop/hrithik/large.py ===============
Enter a number: 44
The square root of 44.0 is approximately 6.63325
>>> 
```

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            .6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit
(AMD64
Type "           ht", "credits" or "license()" for more information.
>>>
======           RT: C:/Users/HP/Desktop/hrithik/large.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

a) CREATE TABLE

```
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)INSERT OPERATION

```
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) UPDATE OPEARTION

```
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) DELETE OPERATION

```
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()
```

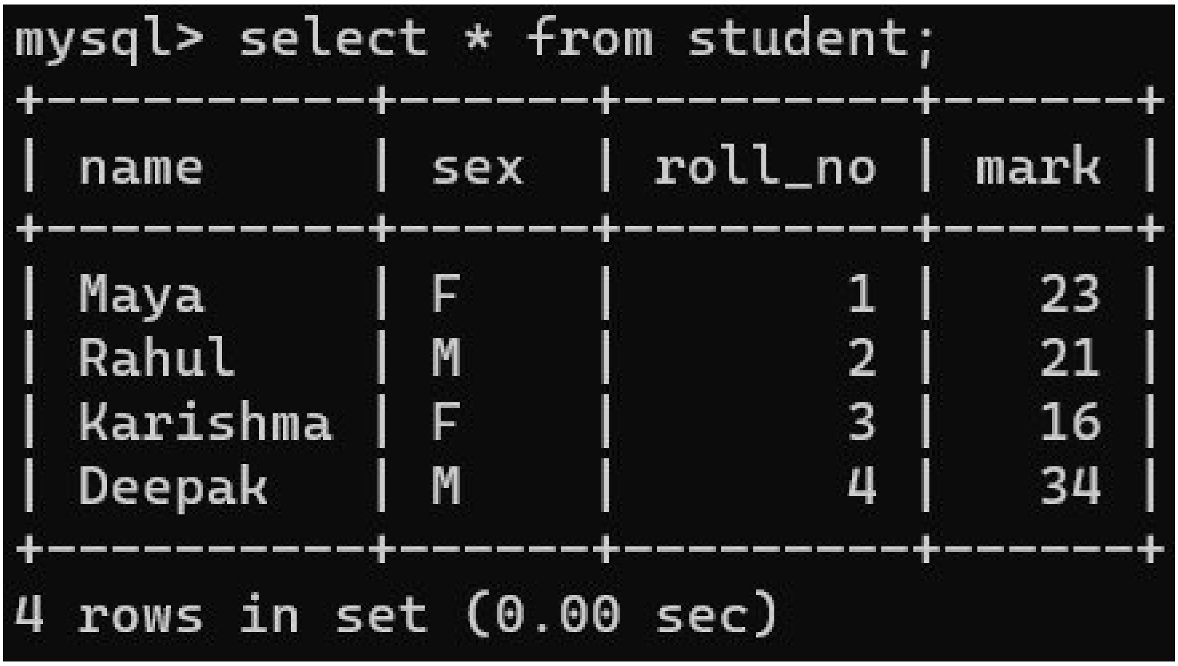
```
if cursor.rowcount > 0:
    print(f"Student with roll number {roll_no} deleted successfully.")
else:
    print(f"No student found with roll number {roll_no}.")
db.close()
print("MySQL connection closed.")
```

E) DISPLAY OPERATION

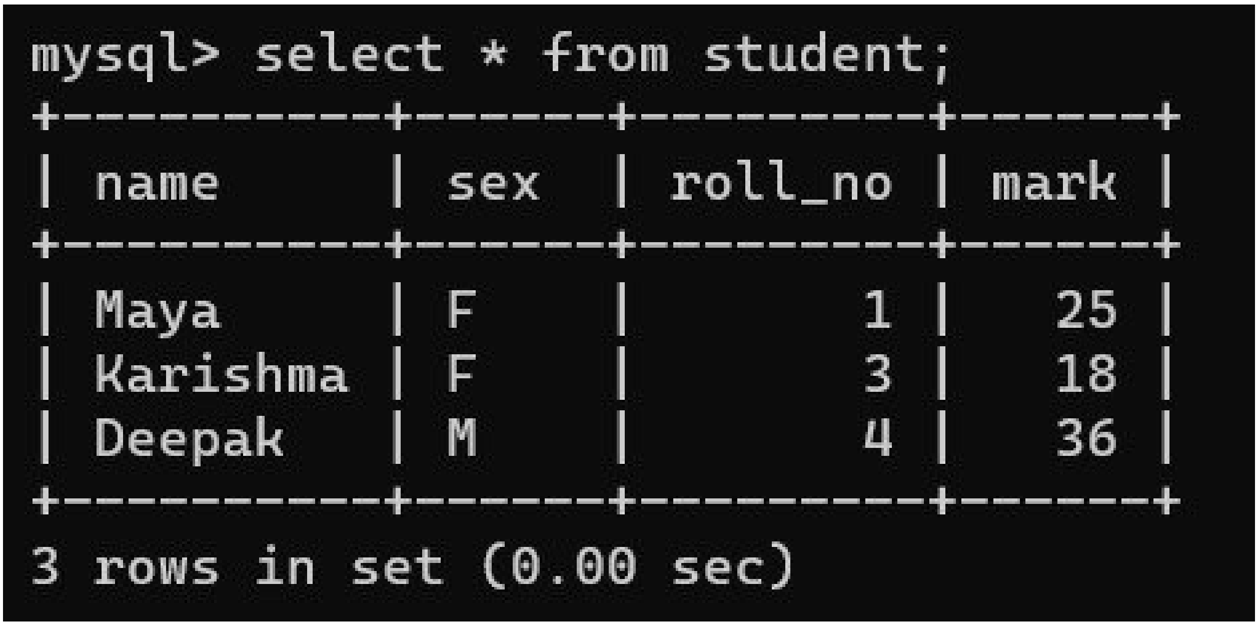
```
import pymysql as ps
try:
    db=ps.connect(host="localhost",
                  user="root",
                 password="",
                 database="sample"
                  )
    cursor = db.cursor()
    roll_no= input("Enter the roll number of the student to display: ")
    select_query = "select * 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.")
```

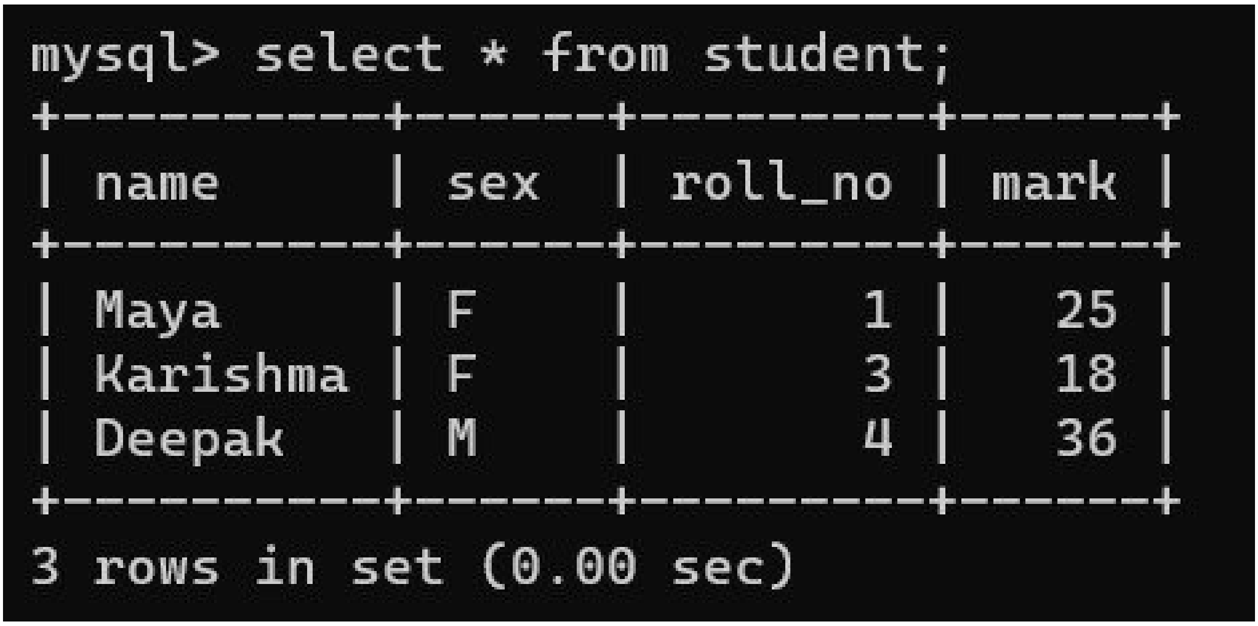
OUTPUT



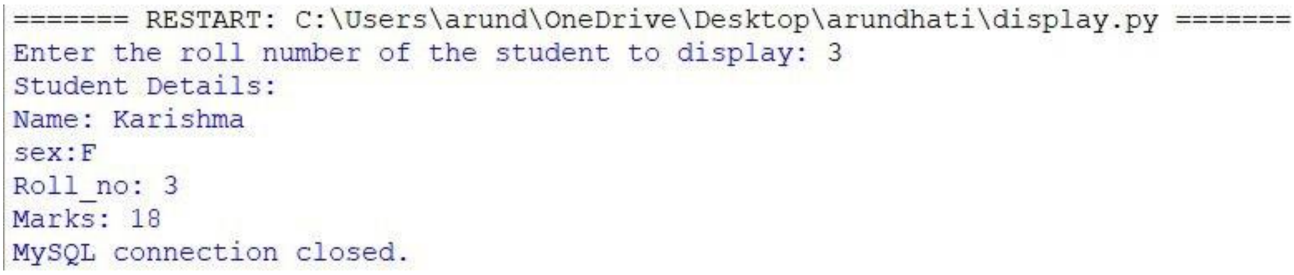
```
mysql> select * from student;
+----------+------+---------+------+
| name     | sex  | roll_no | mark |
+----------+------+---------+------+
| Maya     | F    |       1 |   23 |
| Rahul    | M    |       2 |   21 |
| Karishma | F    |       3 |   16 |
| Deepak   | M    |       4 |   34 |
+----------+------+---------+------+
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 |   25 |
| Karishma | F   |       3 |   18 |
| Deepak   | M   |       4 |   36 |
+----------+-----+---------+------+
3 rows in set (0.00 sec)
```



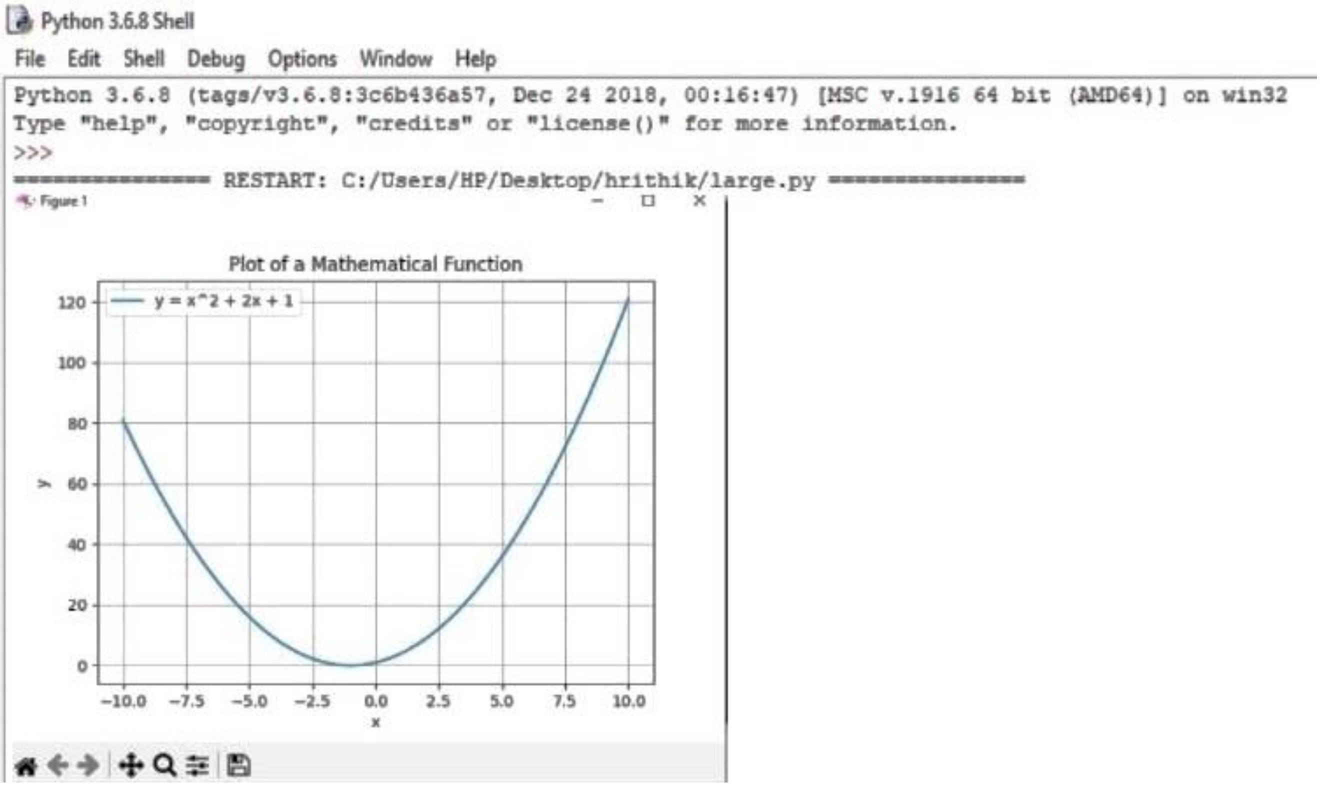
```
======= RESTART: C:\Users\arund\OneDrive\Desktop\arundhati\display.py =======
Enter the roll number of the student to display: 3
Student Details:
Name: Karishma
sex:F
Roll_no: 3
Marks: 18
MySQL connection closed.
```

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.
>>>
=============== RESTART: C:/Users/HP/Desktop/hrithik/large.py ===============
```