

CONCORD ARTS AND SCIENCE COLLEGE

CONCORD EDUCITY, MUTTANNUR, (PO) PATTANNUR



**FIFTH SEMESTER BACHELOR OF
COMPUTER APPLICATION**

PYTHON PROGRAMMING

PRACTICAL RECORD 2022-2023

CONCORD ARTS AND SCIENCE COLLEGE

CONCORD EDUCITY, MUTTANNUR, (PO) PATTANNUR



CERTIFICATE

It is certified that this is a bonafide record of the original work done by

Mr./Mrs..... Reg.no.....

**of Vth semester BCA in the PYTHON PROGRAMMING lab during the year
2022-2023.**

HOD:

Lecturer in charge:

Submitted for practical examination held on

External Examiner

1.

2.

INDEX

NO	PROGRAM	PAGE NO
1	Program to find the largest from a list of numbers	1
2	Program to generate first n perfect numbers	2
3	Program to perform the binary search	3
4	Program to find the square root of a number using bisection search method	5
5	Program to generate Fibonacci series using recursion	7
6	Write a program to find the LCM and GCD of 2 numbers	9
7	Write a program to perform merge sort	10
8	Write a program which reads the contents of a file and copy the contents to another file after changing all the letter to upper case. Exceptions should be handled.	12
9	Write a program to find the prime numbers in a list of numbers.	14
10	Write a python program to Create table students with fields name,sex,rollno,marks	16
11	Create a simple Login window using Tkinter	20
12	Create a plot for the mathematical function x^2 . The title of the plot and the axes should be labelled.	22

1.

AIM

Program to find largest from a list of numbers.

PROGRAM

```
n=int(input("Enter the no of elements:"))
```

```
list=[]
```

```
print("Enter the numbers:")
```

```
for x in range(n):
```

```
    a=int(input())
```

```
    list.append(a)
```

```
max_val=list[0]
```

```
for a in list:
```

```
    if a>max_val:
```

```
        max_val=a
```

```
print("The given list of numbers:",list)
```

```
print("Largest from the list:",max_val)
```

OUTPUT

Enter the no of elements:4

Enter the numbers:

9

6

8

3

The given list of numbers: [9, 6, 8, 3]

Largest from the list: 9

2.

AIM

Program to generate first n perfect numbers.

PROGRAM

```
n=int(input("Enter the limit:"))
for n in range(1,n+1):
    sum=0
    for i in range(1,n):
        if n%i==0:
            sum=sum+i
    if n==sum:
        print(n)
```

OUTPUT

Enter the limit:100

6

28

3.

AIM

Program to perform the Binary search.

PROGRAM

```
n=int(input("Enter the limit:"))
num=[]
flag=0
print("Enter the elements:")
for i in range(1,n+1):
    a=int(input())
    num.append(a)
num.sort()
print("Elements after sorting")
print(num)
item=int(input("Enter the item to be searched:"))
beg=0
end=n-1
while beg<=end:
    mid=(beg+end)//2
    if item==num[mid]:
        print("Search is successfull")
        loc=mid
        print(item,"is present at location",loc+1)
        flag=1
        break
```

```
elif item<num[mid]:  
    end=mid-1  
else:  
    beg=mid+1  
if flag==0:  
    print("search is unsuccessful")
```

OUTPUT

Enter the limit:4

Enter the elements:

8

9

0

6

Elements after sorting

[0, 6, 8, 9]

Enter the item to be searched:9

Search is successfull

9 is present at location 4

4.

AIM

Program to find the the square root of a number using bisection search method.

PROGRAM

```
def f(x,num):  
    return x*x-num  
def square_root(num):  
    a=0  
    b=num  
    while (True):  
        f_a=f(a,num)  
        f_b=f(b,num)  
        if ((f_a * f_b) <0):  
            mid = (a+b)/2.0  
            f_mid=f(mid,num)  
            f_mid_rnd=round(f_mid, 6)  
            if (f_mid_rnd<0):  
                a=mid  
            elif (f_mid_rnd>0):  
                b=mid  
            else:  
                break  
    else:  
        break
```



```
return round(mid,2)
```

```
Num=int(input("Enter a number"))
```

```
res = square_root(Num)
```

```
print("Sqaure Root of ",Num," is ",res)
```

OUTPUT

Enter a number144

Sqaure Root of 144 is 12.0

5.

AIM

Program to generate fibonacci series using recursion.

PROGRAM

```
def fibonacci(n):  
    if n==1:  
        return 0  
  
    if n==2:  
        return 1  
    else:  
        return(fibonacci(n-1) + fibonacci(n-2))  
  
#main program  
n = int(input("Enter number of terms:"))  
if n<= 0:  
    print("Plese enter a positive integer")  
else:  
    print("Fibonacci sequence:")  
    for i in range(1,n+1):  
        print(fibonacci(i))
```

OUTPUT

Enter number of terms:4

Fibonacci sequence:

0

1

1

2

6.

AIM

Program to find the LCM and GCD of two numbers.

PROGRAM

```
def find_gcd(a,b):  
    gcd = 1  
    for i in range(1,a+1):  
        if a%i==0 and b%i==0:  
            gcd = i  
    return gcd  
  
# Reading numbers from user  
first = int(input('Enter first number: '))  
second = int(input('Enter second number: '))  
  
# Function call & displaying output HCF (GCD)  
print('HCF or GCD of %d and %d is %d' %(first, second, find_gcd(first, second)))  
  
# Calculating LCM  
lcm = first * second / find_gcd(first, second)  
print('LCM of %d and %d is %d' %(first, second, lcm))
```

OUTPUT

Enter first number: 8

Enter second number: 2

HCF or GCD of 8 and 2 is 2

LCM of 8 and 2 is

7.

AIM

Program to perform Merge Sort

PROGRAM

```
def merge(list):  
    list_length = len(list)  
    if list_length == 1:  
        return list  
    mid = list_length // 2  
    left = merge(list[:mid])  
    right = merge(list[mid:])  
    return mergesort(left, right)  
  
def mergesort(left, right):  
    output = []  
    i = j = 0  
    while i < len(left) and j < len(right):  
        if left[i] < right[j]:  
            output.append(left[i])  
            i += 1  
        else:  
            output.append(right[j])  
            j += 1  
  
    output.extend(left[i:])
```

```
    output.extend(right[j:])  
    return output  
given_list = [3,15,6,2,19,7,3,5,10,12,4,0,8]  
print("GIVEN LIST : ", given_list)  
sorted_list = merge(given_list)  
print("SORTED LIST : ",sorted_list)
```

OUTPUT

GIVEN LIST : [3, 15, 6, 2, 19, 7, 3, 5, 10, 12, 4, 0, 8]

SORTED LIST : [0, 2, 3, 3, 4, 5, 6, 7, 8, 10, 12, 15, 19]

8.

AIM

Program which reads the contents of a file and copy the contents to another file after changing all letter to uppercase. Exceptions should be handled.

PROGRAM

```
try:
    with open("first_file.txt","r")as file1:
        try:
            with open("second_file.txt","w")as file2:
                for line in file1:
                    file2.write(line.upper())
                print("file copied successful")
                file2.close()
                print("destination file is closed")
        except:
            print("ERROR in file writing")
    finally:
        file1.close()
        print("source file is closed")
except:
    print("ERROR in file opening in readmode")
```

OUTPUT

file copied successful

destination file is closed

source file is closed

9.

AIM

Program to find prime numbers in a list of numbers.

PROGRAM

```
list1=[]
n=int(input("Enter number of elements in the list:"))
print("Enter the elements:")
for i in range(1,n+1):
    item=int(input())
    list1.append(item)
result=[]
for num in list1:
    if num==0 or num==1:
        continue
    for i in range(2,num//2+1):
        if num%i==0:
            break
    else:
        result.append(num)
if len(result):
    print("Prime numbers:")
    for res in result:
        print(res,end=",")
else:
    print("No prime numbers in the list:")
```

OUTPUT

Enter number of elements in the list:6

Enter the elements:

8

9

4

3

2

1

Prime numbers:

3,2,

10.

AIM

Program to perform the following:

- Create table students with fields name, sex, rollno, marks.
- Insert some rows into the table.
- Update the marks of all students by adding 2 marks.
- Delete a student with a given rollno.
- Display the details of a student with a given rollno.

PROGRAM

```
import mysql.connector
```

```
try:
```

```
mydb=mysql.connector.connect(host="localhost",user="root",password="password123",database="mydatabase")
```

```
    print(mydb)
```

```
except:
```

```
    print("Data base connection error")
```

```
try:
```

```
    mycursor=mydb.cursor()
```

```
    sql1="create table if not exists student(rollno int primary key,name  
varchar(20),sex varchar(10),marks int)"
```

```
    mycursor.execute(sql1)
```

```
    mydb.commit()
```

```
    while(True):
```

```
        rollno=int(input("Enter the rollnumber:"))
```

```

        name=input("Enter the name:")
        sex=input("Enter gender:")
        marks=int(input("Enter Marks:"))
        sql2="insert into student
values({},{},{},{})".format(rollno,name,sex,marks)
        mycursor.execute(sql2)
        mydb.commit()
        ch=input("Do you want to insert one more record , press Y/N:")
        ch=ch.lower()
        if ch!='y':
            break
except:
    print("Error while inserting")
    mydb.rollback()
try:
    mycursor=mydb.cursor()
    sql3="update student set marks=marks+2"
    mycursor.execute(sql3)
    mydb.commit()
except:
    print("Error while updating")
    mydb.rollback()
try:
    mycursor=mydb.cursor()
    rollno=int(input("Enter the rollno whose details to be displayed"))
    sql4="select * from student where rollno={ }".format(rollno)

```

```

mycursor.execute(sql4)
result=mycursor.fetchone()
if(result):
    print("Rollno:",result[0])
    print("Name:",result[1])
    print("Sex:",result[2])
    print("Marks",result[3])
else:
    print("Record not found")
mycursor.reset()
except:
    print("Error while displaying")
try:
    mycursor=mydb.cursor()
    rollno=int(input("enter the rollno whose details to be deleted:"))
    sql5="delete from student where rollno={ }".format(rollno)
    mycursor.execute(sql5)
    mydb.commit()
except:
    print("Error while displaying")
    mydb.rollback()

```

OUTPUT

Enter the rollnumber: 3

Enter the name: ASWANI

Enter gender:FEMALE

Enter Marks:86

Do you want to insert one more record , press Y/N:Y

Enter the rollnumber:4

Enter the name:HARSHA

Enter gender:FEMALE

Enter Marks:87

Do you want to insert one more record , press Y/N:Y

Enter the rollnumber:5

Enter the name:JILNA

Enter gender:FEMALE

Enter Marks:87

Do you want to insert one more record , press Y/N:N

Enter the rollno whose details to be displayed 3

Rollno: 3

Name: ASWANI

Sex: FEMALE

Marks 88

enter the rollno whose details to be deleted:5

11.

AIM

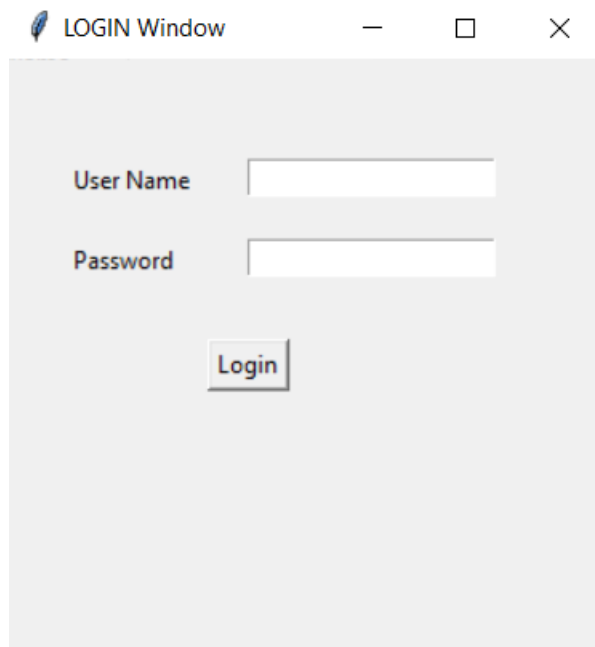
Program to create a simple Login window using Tkinter.

PROGRAM

```
from tkinter import *
def button_clk():
    if pwd.get()=='password':
        window=Tk()
        window.geometry('400x300')
        window.title('Homepage Window')
        message="Hello "+nm.get()+" , Login Successful"
        Label(window,text=message, fg='blue').place(x=100, y=100)
    else:
        Label(top,text='Invalid Username and/or Password', fg='red').place(x=40, y=200)
#Main program
top = Tk()
top.geometry("300x300")
top.title('LOGIN Window')
nm=StringVar()
pwd=StringVar()
name = Label(top, text = "User Name").place(x = 30,y = 50)
password = Label(top, text = "Password").place(x = 30, y = 90)
sbmitbtn = Button(top, text = "Login",activebackground =
"orange",activeforeground = "blue", command=button_clk).place(x = 100, y = 140)
e1 = Entry(top,textvariable=nm).place(x = 120, y = 50)
```

```
e2 = Entry(top,textvariable=pwd,show="*").place(x = 120, y = 90)
top.mainloop()
```

OUTPUT



12.

AIM

Program to create a plot for the mathematical function x^2 . The title of the plot and the axes should be labelled.

PROGRAM

```
import matplotlib.pyplot as plt
x_cords = range(-50,50)
y_cords = [x*x for x in x_cords]
plt.plot(x_cords, y_cords)
plt.title("y=x^2 function")
plt.xlabel("values of x")
plt.ylabel("values of y")
plt.show()
```

OUTPUT

