1. **Write a python program using a function to print Fibonacci series up to n numbers 0 1 1 2 3 5 8 13**

```
def fib(c):
    f,s=0,1
    print (f, s, end=" ")
    i=2
    while i<c:
        t=f+s;
        print(t,end=" ")
        f,s=s,t
        i=i+1
count=int(input("Enter the limit: "))
fib(count)
```

2. Write a Menu driven program in python to find factorial, and sum of natural Numbers using a function.

```
def fact(n):
    return 1 if (n == 1 or n == 0) else n * fact(n - 1)
def sum1(n):
    return 0 if (n == 0) else n + sum1(n - 1)
num=int(input("enter any number"))
print("1-To find the Factorial, 2-To find the Sum, 3-Exit ")
opt = int(input("Enter your option (1-3): "))
if opt == 1:
    print("Factorial of ", num, " is ", fact(num))
elif opt == 2:
    print("Sum of numbers up to ", num, " is ", sum1(num))
elif opt == 3:
    print("Exiting the program")
else:
    print("Invalid option! Please enter 1, 2, or 3")
```

3. Write a Python program using user defined function to calculate interest amount using simple interest method and compound interest method and find the difference of interest amount between the two methods.

```
def simpInt(principle,time,rate):
	si=float(principle*time*rate/100)
	return si
def compInt(principle,time,rate):
	ci=float(principle*((1+rate/100)**time))
	return ci
principle=float(input("Enter Amount: "))
time=float(input("Enter Time: "))
rate=float(input("Enter Rate: "))
si=simpInt(principle,time,rate)
ci=compInt(principle,rate,time)
print("Simple Interest is Rs.%8.2f" %si)
print("Compound Interest is Rs %8.2f" %ci)
diffInt=ci-si;
print("Difference is Rs %8.2f" %diffInt)
```

4. Write a python program to read a text file and display the number of vowels, consonants, and upper case and lower case characters in the file.

```
def count_characters(file_name):
    vowels="aeiouAEIOU"
    vowel_count=0;
    consonant_count=0
    up_count=0
    low_count=0

    with open(file_name,"r")as file:
        text=file.read()

    for ch in text:
        if ch.isalpha():
            if ch in vowels:
                vowel_count+=1
            else:
                consonant_count+=1

        if ch.isupper():
                up_count+=1

        elif ch.islower():
                low_count+=1
    print("Vowels Count=",vowel_count)
    print("Consonant Count=",consonant_count)
    print("Upper Case Characters=",up_count)
    print("Lower Case Characters =",low_count)

def create_text_file(file_name, content):
        with open(file_name,"w")as file:
                file.write(content)

file_name="sample.txt"
content=input("Enter few sentences to create a text file with content: ")
create_text_file(file_name,content)
count_characters(file_name)
```

5. Write a code to count the number of lines, number of words and number of characters in a text file

```
def count_text_file(file_name):
    line_count=0
    word_count=0
    char_count=0
    with open(file_name,"r")as file:
        for line in file:
            line_count+=1
            word_count+=len(line.split())
            char_count+=len(line)
    print("Line Count=",line_count)
    print("Word Count=",word_count)
    print("Character Count=",char_count)

def create_text_file(file_name,content):
    with open(file_name,"w")as file:
        file.write(content)
file_name="count.txt"
content="Hello students this is a sample text file It contains multiple lines***"
create_text_file(file_name,content)
count_text_file(file_name)
```

6. Write a python program to create to read records in binary file with student name and marks of six subjects.

```
import pickle
while True:
    print("1.Create binary file.2.Display the file.3.Exit")
    a=int(input("Choose a command: "))
    if a==1:
        f=open("student.dat","wb")
        x=int(input("How many students?  "))
        for i in range(x):
                  name=input("Name: ")
                  english=int(input("English Marks: "))
                  lan=int(input("Language Marks: "))
                  phy=int(input("Physics Marks: "))
                  chem=int(input("Chemistry Marks: "))
                  maths=int(input("Maths Marks: "))
                  cs=int(input("CSc Marks: "))
                  t=[name,english,lan,phy,chem,maths,cs]
                  pickle.dump(t,f)
        f.close()
    elif a==2:
        f=open("student.dat","rb")
        try:
            while True:
                p=pickle.load(f)
                print(p)
        except:
            f.close()
    if a>2:
        break
```

7. Write a python program to copy the records of the students having percentage 90 and above from the binary file into another file.

```
import pickle
while True:
    print("1.Create Binary file.2.Display the main File.3.Create new file with>90.4.Exit")
    a=int(input("Choose an option: "))
    if a==1:
        with open("student.dat","wb") as f, open("top.dat","wb") as o:
            x=int(input("How many Students? "))
            for i in range(x):
                name=input("Name: ")
                english=int(input("English Marks: "))
                lan=int(input("Language Marks: "))
                phy=int(input("Physics Marks: "))
                chem=int(input("Chemistry mark: "))
                maths=int(input("Math Marks:"))
                cs=int(input("CSc Marks: "))
                total=phy+chem+cs+maths+english+lan
                per=(total/600)*100
                t=[name,english,lan,phy,chem,maths,cs,total,per]
                g=[name,english,lan,phy,chem,maths,cs,total,per]
                pickle.dump(t,f)
                if per>=90:
                    pickle.dump(g,o)
    elif a==2:
        with open("student.dat","rb")as f:
            try:
                while True:
                    p=pickle.load(f)
                    print(p)
            except:
                f.close()
    elif a==3:
            print("Students with >=90 Marks")
            try:
               with open("top.dat","rb") as f:
                    while True:
                        o=pickle.load(f)
                        print(o)
            except:
                f.close()
    else:
        break
```

8. Write a python program using function to sort the elements of a list using bubble sort method.

```
def bubble_sort(numlist):
    n=len(numlist)
    for i in range(n):
        for j in range(0,n-i-1):
            if numlist[j]>numlist[j+1]:
                numlist[j],numlist[j+1]=numlist[j+1],numlist[j]
numlist=[]
n=int(input("Enter the number of Elements: "))
for i in range(n):
        element=int(input("Enter the Element: "))
        numlist.append(element)
print("original list",numlist)
bubble_sort(numlist)
print("Sorted list",numlist)
```

9. Write a python using function to sort the elements of a list using selection sort method.

```
def selection_sort(numlist):
    n=len(numlist)
    for i in range(n):
        min_index=i
        flag=0
        for j in range(i+1,n):
            if numlist[j]<numlist[min_index]:
                min_index=j
                flag=1
        if flag==1:
                numlist[i],numlist[min_index]=numlist[min_index],numlist[i]
numlist=[]
n=int(input("Enter the number of Elements: "))
for i in range(n):
        element=int(input("Enter the Element: "))
        numlist.append(element)
print("Original list=",numlist)
selection_sort(numlist)
print("Sorted list=",numlist)
```

10. Write a python using function to sort the elements of a list using insertion sort method.

```
def insertionSort(arr):
    n = len(numlist)
    for i in range(1, n):
        temp = numlist[i]
        j = i-1
        while j >= 0 and  numlist [j] > temp:
            numlist [j+1] = numlist [j]
            j -= 1
        numlist [j+1] = temp
numlist = [12, 11, 13, 5, 6]
insertionSort(numlist)
print(numlist)
```

11. Write a python program using function to search an element in a list using linear search method

```
def linear_search(numlist,key):
    for i in range(len(numlist)):
        if numlist[i]==key:
            return i+1
    return -1
numlist=[]
n=int(input("Enter the number of elements in the list: "))
for i in range(n):
        element=int(input("Enter the Element: " ))
        numlist.append(element)
key=int(input("Enter the element to be Searched: "))
pos=linear_search(numlist,key)
if pos!=-1:
    print("Element ”, key, “found at position=",pos)
else:
    print("Element ”, key,” not found in the list")
```

12. **Write a python program using function to search an element in a list using binary search method.**

```
def binary_search(numlist,key):
    low,high=0,len(numlist)-1
    while low<=high:
        mid=(low+high)//2
        if numlist[mid]==key:
            return mid+1
        elif numlist[mid]<key:
            low=mid+1
        else:
            high=mid-1
    return -1
numlist=[]
n=int(input("Enter the number of elements: "))
for i in range(n):
        ele=int(input("Enter the elements in ascending order: "))
        numlist.append(ele)
key=int(input("Enter the element to be Searched: "))
pos=binary_search(numlist,key)
if pos!=-1:
    print("Element",key,"is found at position",pos)
else:
    print("Element", key, "is not found in the list")
```

13. Write a program in Python to add and display elements from a stack using list.

```
stack=[]
print("Initially stack is empty",stack)
stack.append('x')
stack.append('y')
stack.append('z')
print("After pushing stack is: ",stack)
print("After poped from stack: ")
print(stack.pop())
print(stack.pop())
print(stack.pop())
print("\n My stack after elements are poped=",stack)
```

14. Write a program in Python to add and display elements from a queue using list.

```
import queue
def display_queue(q):
    print("Queue elements are=",end=" ")
    while not q.empty():
        ele=q.get()
        print(ele,end=" ")
    print("\n Queue size after remove is: ",q.qsize())
q=queue.Queue()
q.put(10)
q.put(20)
q.put(30)
print("Queue size after insert is: ",q.qsize())
display_queue(q)
```

MYSQL

B1) Create a table with the following fields and enter 10 records into the table
Entity Name: marks

Attribute Name	Type	Size	Constraints
Rollno	Int	5	
Sname	Varchar	15	Not null
Lang_mks	Int	3	Between 0 and 100
Eng_mks	Int	3	Between 0 and 100
Sub1_mks	Int	3	Between 0 and 100
Sub2_mks	Int	3	Between 0 and 100
Sub3_mks	Int	3	Between 0 and 100
Sub4_mks	Int	3	Between 0 and 100

```
Create table marks(rollno INT(5),
Sname VARCHAR(15)NOTNULL,
Lang_mks INT(3)CHECK(Lang_mks BETWEEN 0 AND 100),
Eng_mks INT(3)CHECK(Eng_mks BETWEEN 0 AND 100),
Sub1_mks INT(3)CHECK(Sub1_mks BETWEEN 0 AND 100),
Sub2_mks INT(3)CHECK(Sub2_mks BETWEEN 0 AND 100),
Sub3_mks INT(3)CHECK(Sub3_mks BETWEEN 0 AND 100),
Sub4_mks INT(3)CHECK(Sub4_mks BETWEEN 0 AND 100),
);
```

Data to be entered (Values are Indicative)

Rollno	Sname	Lang_mks	Eng_mks	Sub1_mks	Sub2_mks	Sub3_mks	Sub4_mks
1010	RAJ	89	97	98	99	86	95
1026	KIRAN	67	62	72	86	72	62
1042	ANAND	78	87	92	82	72	76
1250	RAM	72	86	72	62	87	68
5212	VIJAYA	46	58	86	92	72	62
3622	MANOJ	86	56	62	86	52	64
1948	REEHAN	63	68	52	56	96	76
1482	KAJOL	49	54	48	76	62	55
1947	KUMAR	98	98	99	100	97	99
1951	REEMA	82	72	62	98	73	64

```
INSERT INTO mark VALUES
(1010,’RAJ’,89,97,98,99,86,95),
(1026,’KIRAN’,67,62,72,86,72,62),
(1042,’ANAND’,78,87,92,82,72,76),
(1250,’RAM’,72,86,72,62,87,68),
(5212,’VIJAYA’,46,58,86,92,72,62),
(3622,’MANOJ’,86,56,62,86,52,64)
(1948,’REEHAN’,63,68,52,56,96,76),
(1482,’KAJOL’,49,54,48,76,62,55),
(1947,’KUMAR’,98,98,99,100,97,99)
(1951,’REEMA’,82,72,62,98,73,64);
```

(i) List all the records

```
Select * from marks;
```

(ii)Display the description of the table

```
Describe marks;
```

(iii)add the new attributes total and percent

```
Alter table marks add(total int(3),percent float(5,3));
```

(iv)Calculate total and percentage of marks for all the students

```
Update marks set total=lang_mks+eng_mks+sub1_mks+ sub2_mks+
sub3_mks+ sub4_mks;

Update marks set percent=total/600*100
```

(v)List the students whose percentage of marks is more than 60%

```
Select sname,percent from marks where percent>=60;
```

(vi)List the students whose percentage is between 60% and 85%

```
Select sname,percent from marks where percent between 60 and 85;
```

(vii)Arrange the students based on percentage of marks from highest to lowest.

```
Select sname,percent from marks order by percent desc;
```

B2) Create table for house hold Electricity bill with the following fields and enter 10 records Entity Name:BESCOM

Attributename	Type	Size	Constraint
RRNO	Varchar	10	Primary key
CUSTNAME	Varchar	25	Not null
BILLDATE	DATE		
UNITS	INT	4	

```
CREATE TABLE BESCOM(
RRNO VARCHAR(10)PRIMARY KEY,
CUSTNAME VARCHAR(25)NOT NULL,
BILLDATE DATE,UNITS INT(4)
);
```

Data to be entered (Values are indicative)

RRNO	CUSTNAME	BILLDATE	UNITS
E1120	RAJ	2024-05-5	250
E2210	KIRAN	2024-03-26	178
E1450	ANAND	2024-04-15	56
E2126	RAM	2024-05-8	782
E1562	MANJULA	2024-05-2	562
E6221	MANOJ	2024-05-18	72
E5822	REEHAN	2024-02-19	92
E1692	KAJOL	2024-03-25	73
E6721	KUMAR	2024-07-14	589
E2682	REEMA	2024-05-11	100

```
INSERT INTO BESCOM VALUES

(‘E1120’,’RAJ’,’2024-05-05’,250),

(‘E2210’,’KIRAN’,’2024-03-26’,178),

(‘E1450’,’ANAND’,’2024-04-15’,56),

(‘E2126’,’RAM’,’2024-05-08’,782)

(‘E1562’,’MANJULA’,’2024-05-02’,562),

(‘E6221’,’MANOJ’,’2024-05-18’,72),

(‘E5822’,’REEHAN’,’2024-02-19’,92),

(‘E1692’,’KAJOL’,’2024-03-25’,73),

(‘E6721’,’KUMAR’,’2024-07-14’,589),

(‘E2682’,’REEMA’,’2024-05-11’,100);
```

7. View the structure of table.

```
Describe bescom;
```

8. List all the record

```
Select * from bescom;
```

9. Add a new field for bill amount in the name of billamt.

```
Alter table BESCOM add billamt float(7,2);
```

10. Compute the bill amount for each consumer as per the following rules.

a.minimum Amount	**Rs.100**
b.For first 100 units	**Rs.7.50/unit**
c.For the above 100 Units	**Rs.8.50/Unit**

```
(i)UPDATE BESCOM SET BILLAMT=100+UNITS*7.50 WHERE UNITS<=100;

(ii)UPDATE BESCOM SET BILLAMT=100+(100*7.50)+(UNITS-100)*8.50
WHERE UNITS>100;
```

11. Display the maximum ,minimum and total bill amount.

```
SELECT MAX(BILLAMT),min(billamt),avg(billamt),sum(billamt) FROM
BESCOM;
```

12. List all the bills generated in a sorted order based on RRNO.

```
Select rrno,billamt  from bescom order by rrno;
```

B3) Create a table with the following details and enter 10 records into the table

Entity Name: student

Attribute name	Type	Size	Constraint
Rollno	Int	5	Primary key
Sname	Varchar	15	Not null
DOB	Date		
Gender	Char	1	
Combn	Varchar	5	
Class	Char	6	

```
CREATE TABLE student(

Rollno int(5)PRIMARY KEY,

Sname VARCHAR(15)NOT NULL,

Dob date,

Gender CHAR(1),

Combn VARCHAR(5),

Class VARCHAR(6)

);
```

Data to be enetered(Values are Indicative)

RollNo	Sname	DOB	Gender	Combn	Class

```
insert into student values(3739,’Uday’,’2004-9-12’,’M’,’PCMC’,’2B’)

insert into student values(1001,’Raj Kumar’,’2005-5-21’,’M’,’BASC’,’2A’)

insert into student values(1005,’Kiran,’2004-11-15’,’M’,’PCMC’,’2B’)

insert into student values(1042,’Anand’ ,’2005-12-22’,’M’,’CEBA’,’2C’)

insert into student values(1250,’Ram’,’2004-6-18’,’M’,’PCMC’,’2A’)

insert into student values(5212,’Vijaya’,’2007-7-28’,’F’,’PCMC’,’2A’)

insert into student values(1029,’Bharat’,’2005-1-12’,’M’,’BASC’,’2B’)

insert into student values(2152,’Rekha’,’2006-6-8’,’F’,’CEBA’,’2C’)

insert into student values(1948,’Reehan’,’2005-4-17’,’M’,’CEBA’,’2C’)

insert into student values(2443,’Manjula’,’2005-8-15’,’F’,’PCMC’,’2B’)
```

(i)List all the students

```
Select * from student;
```

(ii)List only those students who are in BASC and CEBA combination

```
Select * from student where combn=’BASC’ or combn=’CEBA’;
```

(iii)List only the combination by removing duplicate values.

```
Select distinct(combn)from student;
```

(iv)List the students alphabetically

```
Select sname from student order by sname;
```

(v)List the students alphabetically class-wise

```
Select sname, class from student class order by class, sname;
```

(vi)List the students who born in the month of june of any year

```
Select * from student WHERE MONTH (dob)=6;
```

(vii)Count the number of students Gender –wise.

```
Select gender ,COUNT(*) FROM student GROUP BY gender;
```

B4) Create a table with following fields and enter 10 records into the table.

Entity name: Library

Attributename	Type	Size	Constraint
Title	Varchar	75	Not null
Author	Varchar	60	
Year	Int	4	
Category	Varchar	25	
Price	Float	7.2	
Qty	Int	4	

```
Create table Library(
Title Varchar(75)NOTNULL,
Author Varchar(60),
Year int(4),
Category Varchar(25),
Price float(7.2),
Qty int(4)
);
```

Data to be entered :(Values are Indicative)

Title	Author	year	Category	Price	Qty
The Datascience Handbook	Darshan	2019	Data Science	750	12
Introduction to computer programs	Harshavardhan	Null	Computer Science	700	6
Computer science Text book class-12	Reeta Sahoo’	2019	Textbook	450	4
A Book on AI	Sagar	2016	AI	200	7
Robotics and Automation	Dushyanth	2018	Digital Technology	750	9
AI 2041	Chen& Lee	2021	AI	1000	2
Computer Hardware and software	Chethan	2000	Computer science	500	7
The future of work	Dev Kumar	2018	Digital technology	750	4
Healthcare and AI	Eshwar	2019	AI	950	9
Introduction to Datascience	Dravid	2014	Data Science	400	5

```
INSERT INTO LIBRARY VALUES

(‘The Datascience Handbook’,’Darshan’,2019’,’Data Science’,750.00,12),

(‘Introduction to computer programs’,’Harshavardhan’,Null,’Computer
Science’,700.00,6),

(‘Computer science Text book class-12,’Reeta Sahoo’,2019,’Textbook’,450.00,4),

(‘A Book on AI’,’Sagar’,2016,’AI’,200.00,7),

(‘Robotics and Automation’,’ Dushyanth’, 2018, ‘Digital Technology’,750.00,9),

(‘AI 2041’,’ Chen& Lee’, 2021, ‘AI’,1000.00,2),

(‘Computer Hardware and software’,’Chethan’, 2000, ‘Computer
science’,500.00,7),

(‘Computer Hardware and software’,’Chethan’, 2000, ‘Computer
science’,500.00,7),

(‘The Future of Work’,’Dev Kumar’, 2018, ‘Digital Technology’,750.00,4),

(‘Healthcare and AI’,’Eshwar’, 2019, ‘AI’,950.00,9),

(‘Introduction to DataScience’‘Dravid’,’2014’,’Data Science’,400.00,5));
```

(i)List all the books

```
Select * from Library;
```

(ii)Calculate Amount by altering table by adding a new column ‘Amount’

```
(a)Alter table library ass(amount float(7,2));

(b)update library set amount=price*qty;
```

(iii)List the records of all those books price is between 400 and 900.

```
Select price from library where price between 400 and 900
```

(iv)List those records with no value in the attribute year.

```
Select title,year from library where year is NULL;
```

(v)List the names of the authors whose name starts with letter ‘C’ or ‘D’.

```
Select author form library where author like ‘C%’ or author like ‘D%’;
```

(vi)List title ,year and category from the table library with category field has word ‘science’.

Select Title,Year ,Category from library WHERE category LIKE '%science%';

(vii)List all those records whose year of publication is 2010 onwards with book price is less than Rs.750.

SELECT year ,price FROM Library WHERE Year>=2010 AND Price <750;