

13685246

PRACTICAL FILE

COMPUTER SCIENCE

**SUBMITTED BY:
SANSKRITI CHATURVEDI**

**SUBMITTED TO:
MR. ANISH BANSAL**

python



INDEX

S. No.	Content	Pg no.
1	Add, sub, even odd with no parameters.	3-4
2	Programme 1 with parameters.	5-6
3	Programme 2 with return.	7-8
4	Fibonacci sequence.	9
5	Armstrong number.	10
6	Dice simulator.	11-12
7	Count Me and My.	13
8	Count A,a and M,m.	14-15
9	Stacks.	16-17
10	Count vowels.	18

INDEX

S. No.	Content	Pg no.
11	Read, write and append a text file.	19-22
12	Read, write and append a csv file.	23-26
13	Pickle module.	27
14	Remove lines containing 'a' from a text file and writing them to another file.	28-29
15	Binary file- write, read and update.	30-32
16	Binary file- write, read and search.	33-35
17	MySQL.connector- Connection.	36
18	MySQL.connector- Database.	37
19	MySQL.connector- Insert.	38-39
20	MySQL.connector- Select.	40-41

#Program 1:

```
● ● ●  
...  
Program defining functions for addition, subtraction and verifying whether the user input is  
odd or even with no parameters.  
...  
  
def add():  
    x=int(input("Enter x:"))  
    y=int(input("Enter y:"))  
    z=x+y  
    print(x,"+",y,"=",z)  
  
def sub():  
    x=int(input("Enter x:"))  
    y=int(input("Enter y:"))  
    z=x-y  
    print(x,"-",y,"=",z)  
  
def evenodd():  
    x=int(input("Enter x:"))  
    if x%2==0:  
        print(x,"is an even number.")  
    else:  
        print(x,"is an odd number.")  
  
c="y"  
while c=="y" or c=="yes" :  
    print("enter 1 for addition")  
    print("enter 2 for subtraction")  
    print("enter 3 for evenodd")  
    f=int(input("Enter your function choice: "))  
    print ("Your choice is: ",f)  
    if f==1:  
        add()  
    elif f==2:  
        sub()  
    elif f==3:  
        evenodd()  
    else:  
        print("invalid input")  
    c=input("Would you like to continue? ")
```

Output 1:

```
● ● ●

enter 1 for addition
enter 2 for subtraction
enter 3 for evenodd
Enter your function choice: 1
Your choice is: 1
Enter x:4
Enter y:5
4 + 5 = 9
Would you like to continue? y
enter 1 for addition
enter 2 for subtraction
enter 3 for evenodd
Enter your function choice: 2
Your choice is: 2
Enter x:7
Enter y:5
7 - 5 = 2
Would you like to continue? y
enter 1 for addition
enter 2 for subtraction
enter 3 for evenodd
Enter your function choice: 3
Your choice is: 3
Enter x:9
9 is an odd number.
Would you like to continue? y
enter 1 for addition
enter 2 for subtraction
enter 3 for evenodd
Enter your function choice: 3
Your choice is: 3
Enter x:14
14 is an even number.
Would you like to continue? n
```

#Program 2:

```
● ● ●  
...  
Program defining functions for addition, subtraction and verifying whether the user input is  
odd or even with parameters (and arguments).  
...  
  
def add(x:int,y:int):  
    z=x+y  
    print(x,"+",y,"=",z)  
  
def sub(x:int,y:int):  
    z=x-y  
    print(x,"-",y,"=",z)  
  
def evenodd(x:int):  
    if x%2==0:  
        print(x,"is an even number.")  
    else:  
        print(x,"is an odd number.")  
  
c="y"  
while c=="y" or c=="yes" :  
    print("enter 1 for addition")  
    print("enter 2 for subtraction")  
    print("enter 3 for evenodd")  
    f=int(input("Enter your function choice: "))  
    print ("Your choice is: ",f)  
    n1=int(input("Enter n1:"))  
    n2=int(input("Enter n2:"))  
    if f==1:  
        add(n1,n2)  
    elif f==2:  
        sub(n1,n2)  
    elif f==3:  
        evenodd(n1)  
        evenodd(n2)  
    else:  
        print("invalid input")  
    c=input("Would you like to continue? ")
```

Output 2:

```
● ● ●

enter 1 for addition
enter 2 for subtraction
enter 3 for evenodd
Enter your function choice: 1
Your choice is: 1
Enter n1:5
Enter n2:10
5 + 10 = 15
Would you like to continue? y
enter 1 for addition
enter 2 for subtraction
enter 3 for evenodd
Enter your function choice: 2
Your choice is: 2
Enter n1:15
Enter n2:10
15 - 10 = 5
Would you like to continue? y
enter 1 for addition
enter 2 for subtraction
enter 3 for evenodd
Enter your function choice: 3
Your choice is: 3
Enter n1:15
Enter n2:20
15 is an odd number.
20 is an even number.
Would you like to continue? n
```

#Program 3:

```
● ● ●  
...  
Program defining functions for addition, subtraction and verifying whether the user input is  
odd or even with parameters (and arguments) and return.  
...  
  
def add(x:int,y:int):  
    z=x+y  
    return z  
  
def sub(x:int,y:int):  
    z=x-y  
    return z  
  
def evenodd(x:int):  
    if x%2==0:  
        return 1  
    else:  
        return 3  
  
c="y"  
while c=="y" or c=="yes" :  
    print("enter 1 for addition")  
    print("enter 2 for subtraction")  
    print("enter 3 for evenodd")  
    f=int(input("Enter your function choice: "))  
    print ("Your choice is: ",f)  
    n1=int(input("Enter n1:"))  
    n2=int(input("Enter n2:"))  
    if f==1:  
        a=add(n1,n2)  
        print(n1,"+",n2,"=",a)  
    elif f==2:  
        a=sub(n1,n2)  
        print(n1,"-",n2,"=",a)  
    elif f==3:  
        a=evenodd(n1)  
        b=evenodd(n2)  
        if a==1:  
            print(n1,"is an even number.")  
        else:  
            print(n1,"is an odd number.")  
        if b==1:  
            print(n2,"is an even number.")  
        else:  
            print(n2,"is an odd number.")  
    else:  
        print("invalid input")  
    c=input("Would you like to continue? ")  
    print("Goodbye")
```

Output 3:

```
• • •  
enter 1 for addition  
enter 2 for subtraction  
enter 3 for evenodd  
Enter your function choice: 1  
Your choice is: 1  
Enter n1:5  
Enter n2:7  
5 + 7 = 12  
Would you like to continue? y  
enter 1 for addition  
enter 2 for subtraction  
enter 3 for evenodd  
Enter your function choice: 2  
Your choice is: 2  
Enter n1:8  
Enter n2:10  
8 - 10 = -2  
Would you like to continue? y  
enter 1 for addition  
enter 2 for subtraction  
enter 3 for evenodd  
Enter your function choice: 3  
Your choice is: 3  
Enter n1:10  
Enter n2:15  
10 is an even number.  
15 is an odd number.  
Would you like to continue? n
```

#Program 4:

```
...
Program to define a function to generate a Fibonacci series of
the user input length.
...

def fib():
    x=0
    y=1
    a=int(input("Enter the length of the series: "))
    print(x)
    print(y)
    for n in range(3,a+1):
        z=x+y
        print(z)
        x=y
        y=z
fib()
```

Output 4:

```
...
Enter the length of the series: 10
0
1
1
2
3
5
8
13
21
34
```

#Program 5:

```
...  
...  
Program to verify whether a 3 digit number is an Armstrong number or not.  
...  
  
def armno(x:int,y:int,z:int):  
    x3=x**3  
    y3=y**3  
    z3=z**3  
    s=x3+y3+z3  
    if s==int(n) :  
        print("Yes,",int(n),"is an Armstrong number.")  
    else:  
        print("No,",int(n),"is not an Armstrong number.")  
  
n=str(input("Enter the 3 digit number to be verified: "))  
  
a=int(n[0])  
b=int(n[1])  
c=int(n[2])  
  
armno (a,b,c)
```

Output 5:

```
...  
...  
Enter the 3 digit number to be verified: 153  
Yes, 153 is an Armstrong number.  
-----  
Enter the 3 digit number to be verified: 234  
No, 234 is not an Armstrong number.
```

#Program 6:

```
...
Program to generate a random number between 1 to 6 (including 1 and 6) using the random module
(dice simulator).
...

import random

x = "y"

while x=="y" or x=="yes" :
    n = random.randint(1,6)

    if n==1:
        print("----")
        print("    ")
        print("  0 ")
        print("    ")
        print("----")

    elif n==2:
        print("----")
        print("0   ")
        print("   ")
        print("   0")
        print("----")

    elif n==3:
        print("----")
        print("0   ")
        print("  0 ")
        print("  0")
        print("----")

    elif n==4:
        print("----")
        print("0  0")
        print("   ")
        print("0  0")
        print("----")

    elif n==5:
        print("----")
        print("0  0")
        print("  0 ")
        print("0  0")
        print("----")

    elif n==6:
        print("----")
        print("0  0")
        print("0  0")
        print("0  0")
        print("----")

    x = input("Would you like to continue?: ")
    print("\n")
```

Output 6:

```
• • •  
-----  
0 0  
0 0  
0 0  
-----  
Would you like to continue?: y  
  
-----  
0  
0  
0  
-----  
Would you like to continue?: n
```

#Program 7:

```
...
Program to count the no. of "Me" or "My" words present in a
file story.txt. If the story.txt contents are as follows:
My first book
was Me and
My Family. It
gave Me
a chance to be
Known to the
world.
-----
The output should read :
Total no. of Me and My= 4
...

def countMeMy():
    num=0
    f=open("story.txt","r")
    n=f.read()
    m=n.split()
    for x in m:
        if x == "Me" or x == "My" :
            num=num+1
    f.close()
    print("Total no. of Me and My= ", num)

countMeMy()
```

Output 7:

```
Total no. of Me and My= 5
```

#Program 8:

```
...
Program which reads each character in a text file story.txt and displays and counts the
occurrence of each A or a and M or m. If the story.txt contents are as follows:
My first book
Was me and
My family. It
Gave Me
A chance to be
Known to the
World.
...

def amcount():
    l=0
    m=0
    f=open("story.txt",'r')
    n=f.read()
    for x in n:
        if x=="a" or x=="A" :
            print(x)
            l = l+1
        elif x=="m" or x=="M" :
            print(x)
            m = m+1
    f.close()
    print("Total no. of a or A= ",l)
    print("Total no. of m or M= ",m)

amcount()
```

Output 8:

```
M
a
m
a
M
a
m
a
M
A
a
Total no. of a or A= 6
Total no. of m or M= 5
```

#Program 9:

```
...
Program to implement a stack.
...
def stacks():
    l=[]
    c="y"
    while c=="y":
        print("1.Push")
        print("2.Pop")
        print("3.Display")
        f=int(input("Enter your function choice: "))
        if f==1:
            s=input("Enter element to be pushed:")
            l.append(s)
        elif f==2:
            if l==[]:
                print("Stack is empty")
            else:
                print("Removed item is: ",l.pop())
        elif f==3:
            for x in l:
                print(x)
        else:
            print("Invalid input")
    c=input("Would you like to continue? ")
stacks()
```

Output 9:

```
● ● ●

1.Push
2.Pop
3.Display
Enter your function choice: 1
Enter element to be pushed:1
Would you like to continue? y
1.Push
2.Pop
3.Display
Enter your function choice: 1
Enter element to be pushed:2
Would you like to continue? y
1.Push
2.Pop
3.Display
Enter your function choice: 1
Enter element to be pushed:2
Would you like to continue? y
1.Push
2.Pop
3.Display
Enter your function choice: 3
1
2
2
Would you like to continue? y
1.Push
2.Pop
3.Display
Enter your function choice: 2
Removed item is: 2
Would you like to continue? y
1.Push
2.Pop
3.Display
Enter your function choice: 3
1
2
Would you like to continue? y
1.Push
2.Pop
3.Display
Enter your function choice: 4
Invalid input
Would you like to continue? n
```

#Program 10:

```
...
Program to count the no. of vowels present in a file
story.txt. If the story.txt contents are as follows:
My first book
was Me and
My Family. It
gave Me
a chance to be
Known to the
world.
-----
The output should read :
Total no. of Vowels =  21
...

def countv():
    num=0
    f=open("story.txt","r")
    n=f.read()
    for x in n.lower() :
        if x in ['a','e','i','o','u']:
            num=num+1
    f.close()
    print("Total no. of Vowels = ", num)

countv()
```

Output 10:

```
Total no. of Vowels =  21
```

#Program 11:

```
...
Programme to read, write and append a text file.

def fread():
    print("1.Only read")
    print("2.Read and write")
    c2=int(input("Enter your choice: "))
    if c2==1:
        f=open('story.txt','r')
        r=f.read()
        print(r)
        f.close()
    elif c2==2:
        f=open('story.txt','r+')
        r=f.read()
        print(r)
        f.seek(0)
        i=input("Enter what you would like to write:")
        f.write(i)
        f.close()
    else:
        print("Invalid input")

def fwrite():
    print("1.Only write")
    print("2.Write and read")
    c2=int(input("Enter your choice: "))
    if c2==1:
        f=open('story.txt','w')
        i=input("Enter what you would like to write: ")
        w=f.write(i)
        f.close()
    elif c2==2:
        f=open('story.txt','w+')
        i=input("Enter what you would like to write: ")
        f.write(i)
        f.seek(0)
        r=f.read()
        print(r)
        f.close()
    else:
        print("Invalid input")
```

```
def fappend():
    print("1.Only append")
    print("2.Append and read")
    c2=int(input("Enter your choice: "))
    if c2==1:
        f=open('story.txt','a')
        i=input("Enter what you would like to append: ")
        f.write(i)
        f.close()
    elif c2==2:
        f=open('story.txt','a+')
        i=input("Enter what you would like to append: ")
        a=f.write(i)
        f.seek(0)
        r=f.read()
        print(r)
        f.close()
    else:
        print("Invalid input")

def menu():
    c="y"
    while c=="y":
        print("1.Read")
        print("2.Write")
        print("3.Append")
        c1=int(input("Enter your choice: "))
        if c1==1:
            fread()
        elif c1==2:
            fwrite()
        elif c1==3:
            fappend()
        else:
            print("Invalid input")
    c=input("Would you like to continue? ")

menu()
```

Output 11:

```
● ● ●

1.Read
2.Write
3.Append
Enter your choice: 1
1.Only read
2.Read and write
Enter your choice: 1
My first book
Was me and
My family. It
Gave Me
A chance to be
Known to the
World.
Would you like to continue? y
1.Read
2.Write
3.Append
Enter your choice: 1
1.Only read
2.Read and write
Enter your choice: 2
My first book
Was me and
My family. It
Gave Me
A chance to be
Known to the
World.
Enter what you would like to write:testing_r+
Would you like to continue? y
1.Read
2.Write
3.Append
Enter your choice: 1
1.Only read
2.Read and write
Enter your choice: 1
testing_r+ook
Was me and
My family. It
Gave Me
A chance to be
Known to the
World.
```

```
Would you like to continue? y
1.Read
2.Write
3.Append
Enter your choice: 2
1.Only write
2.Write and read
Enter your choice: 1
Enter what you would like to write: testing_w
Would you like to continue? y
1.Read
2.Write
3.Append
Enter your choice: 1
1.Only read
2.Read and write
Enter your choice: 1
testing_w
Would you like to continue? y
1.Read
2.Write
3.Append
Enter your choice: 2
1.Only write
2.Write and read
Enter your choice: 2
Enter what you would like to write: testing_w+
testing_w+
Would you like to continue? y
1.Read
2.Write
3.Append
Enter your choice: 3
1.Only append
2.Append and read
Enter your choice: 1
Enter what you would like to append: testing_a
Would you like to continue? y
1.Read
2.Write
3.Append
Enter your choice: 1
1.Only read
2.Read and write
Enter your choice: 1
testing_w+testing_a
Would you like to continue? y
1.Read
2.Write
3.Append
Enter your choice: 3
1.Only append
2.Append and read
Enter your choice: 2
Enter what you would like to append: testing_a+
testing_w+testing_atesting_a+
Would you like to continue? n
```

#Program 12:

```

...
Programme to read, write and append a csv file.
...

import csv

def fread():
    with open('demo_csv.csv', mode="r") as csv_file:
        reader = csv.reader(csv_file)
        for item in reader:
            print(item)

def fwrite():
    column_name = ["Name", "Sex", "Age", "Height (in)", "Weight (lbs)"]
    inm=input("Enter name: ")
    isx=input("Enter sex: ")
    iage=int(input("Enter age: "))
    iht=int(input("Enter height: "))
    iwt=int(input("Enter weight: "))
    data = [ inm, isx, iage, iht, iwt]
    with open('demo_csv.csv', 'w') as f:
        writer = csv.writer(f)
        writer.writerow(column_name)
        writer.writerow(data)

def fappend():
    field_names = ['Name', 'Sex', 'Age', 'Height (in)', 'Weight (lbs)']
    inm=input("Enter name: ")
    isx=input("Enter sex: ")
    iage=int(input("Enter age: "))
    iht=int(input("Enter height: "))
    iwt=int(input("Enter weight: "))
    dict = {"Name": inm, "Sex":isx, "Age":iage, "Height (in)":iht, "Weight (lbs)": iwt}
    with open('demo_csv.csv', 'a') as csv_file:
        dict_object = csv.DictWriter(csv_file, fieldnames=field_names)
        dict_object.writerow(dict)

def menu():
    c="y"
    while c=="y":
        print("1.Read")
        print("2.Write")
        print("3.Append")
        c1=int(input("Enter your choice: "))
        if c1==1:
            fread()
        elif c1==2:
            fwrite()
        elif c1==3:
            fappend()
        else:
            print("Invalid input")
        c=input("Would you like to continue? ")

menu()

```

The following csv file demo_csv.csv is used in this programme:

```
● ● ●  
"Name", "Sex", "Age", "Height (in)", "Weight (lbs)"  
"Alex", "M", 41, 74, 170  
"Elly", "F", 30, 66, 124  
"Hank", "M", 30, 71, 158  
"Ivan", "M", 53, 72, 175  
"Page", "F", 31, 67, 135
```

Output 12:

```
● ● ●  
1. Read  
2. Write  
3. Append  
Enter your choice: 1  
['Name', 'Sex', 'Age', 'Height (in)', 'Weight (lbs)']  
[{'Name': 'Alex', 'Sex': 'M', 'Age': 41, 'Height (in)': 74, 'Weight (lbs)': 170},  
 {'Name': 'Elly', 'Sex': 'F', 'Age': 30, 'Height (in)': 66, 'Weight (lbs)': 124},  
 {'Name': 'Hank', 'Sex': 'M', 'Age': 30, 'Height (in)': 71, 'Weight (lbs)': 158},  
 {'Name': 'Ivan', 'Sex': 'M', 'Age': 53, 'Height (in)': 72, 'Weight (lbs)': 175},  
 {'Name': 'Page', 'Sex': 'F', 'Age': 31, 'Height (in)': 67, 'Weight (lbs)': 135}]
```

```
Would you like to continue? y
1.Read
2.Write
3.Append
Enter your choice: 2
Enter name: Gwen
Enter sex: F
Enter age: 26
Enter height: 64
Enter weight: 121
```

The csv file now reads:

```
Name,Sex,Age,Height (in),Weight (lbs)
Gwen,F,26,64,121
```

The terminal shows:

```
Would you like to continue? y
1.Read
2.Write
3.Append
Enter your choice: 3
Enter name: Ivan
Enter sex: M
Enter age: 29
Enter height: 72
Enter weight: 175
Would you like to continue? n
```

The csv file reads:

```
...  
Name,Sex,Age,Height (in),Weight (lbs)  
Gwen,F,26,64,121  
Ivan,M,29,72,175
```

#Program 13:

```
...
Programme using pickle module- pickle.load and pickle.dump .
...

import pickle

fd=open('studentdetails.dat','ab')
name=input("Enter the student's name: ")
roll=input("Enter the student's roll no.: ")
marks=input("Enter the student's marks: ")
l=[name, roll, marks]
pickle.dump(l,fd)
fd.close()

fl=open('studentdetails.dat','rb')
pickle.load(fl)
fl.close()
```

Output 13:

```
...
Enter the student's name: sc
Enter the student's roll no.: 38
Enter the student's marks: 70
```

#Program 14:

```
...
Programme to remove all the lines that contain the character 'a' in a file and
write it to another file.
...

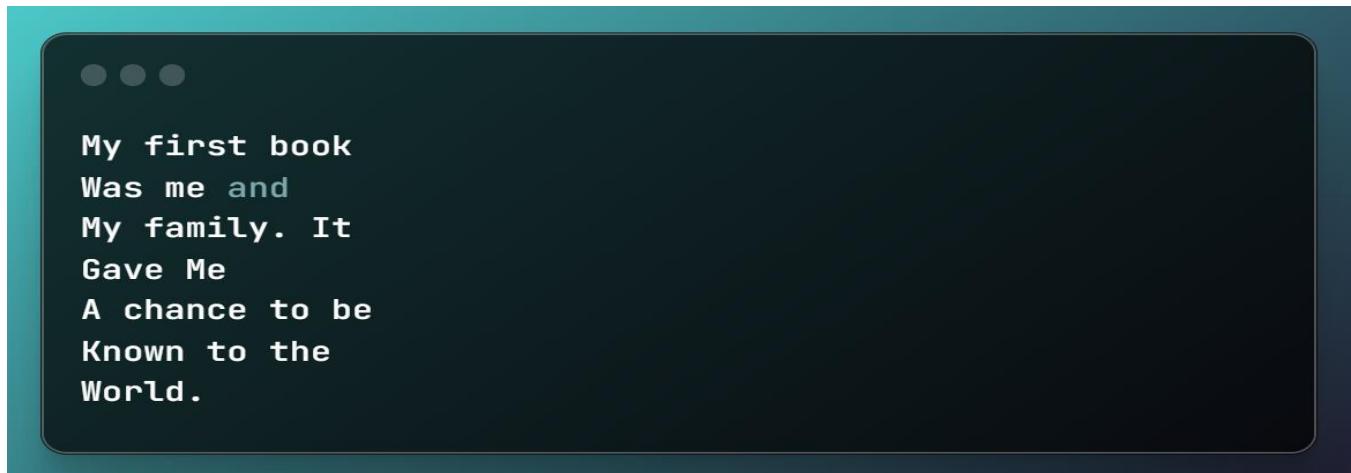
def read_a():
    global l
    l=[]
    f=open("story.txt","r")
    fl=f.readlines()
    for line in fl:
        for ch in line:
            if ch=='a':
                line=line.replace("\n","");
                l.append(line)
                break
    f.close()

def write_a():
    for i in l:
        f=open("story1.txt","a")
        f.write(i)
        f.close()
        f=open("story1.txt","a")
        f.write("\n")
        f.close()

def remove_a():
    global l1
    l1=[]
    f=open("story.txt","r+")
    global fl
    fl=f.readlines()
    a=0
    for line in fl:
        a=line.find("a")
        if a == -1:
            line=line.replace("\n","");
            l1.append(line)
    for i in l1:
        if i==l1[0]:
            f=open("story.txt","w")
            f.write(i)
            f.close()
            f=open("story.txt","a")
            f.write("\n")
            f.close()
        else:
            f=open("story.txt","a")
            f.write(i)
            f.close()
            f=open("story.txt","a")
            f.write("\n")
            f.close()

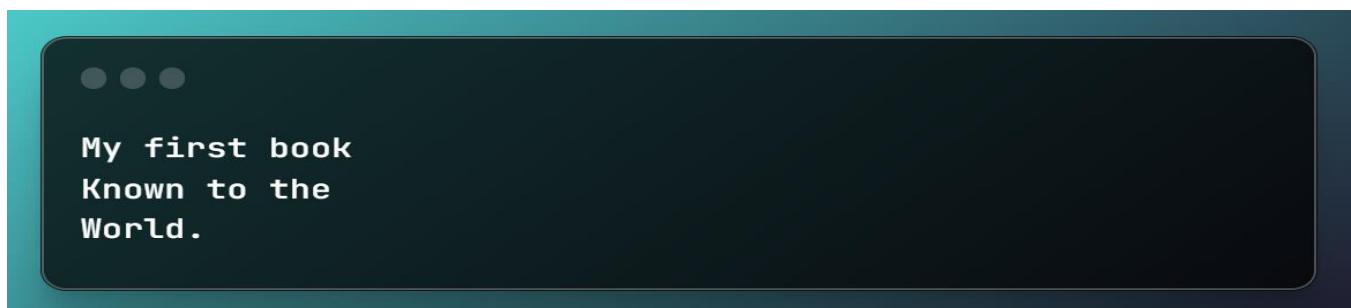
    read_a()
    write_a()
    remove_a()
```

The story.txt file used in this programme is as follows:

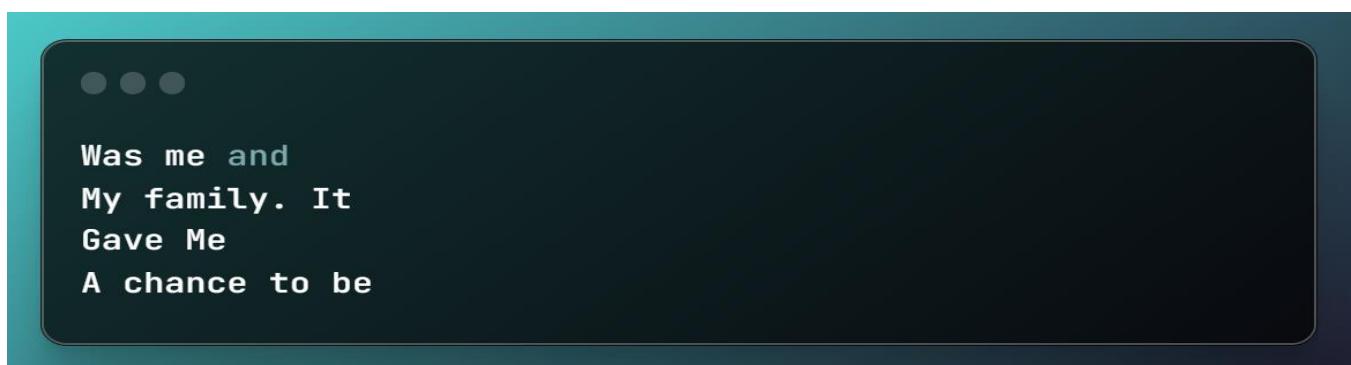


Output 14:

After running the programme, story.txt read:



While the newly formed story1.txt file read :



#Program 15:

```
...
Program to create a binary file with roll number, name and marks; Input a roll
number and update the marks.
...

import pickle

def menu():
    available_functions = {
        1: "View Records",
        2: "Add Records",
        3: "Update Records",
        4: "Exit"
    }

    print("Available functions are: ")
    for i, j in available_functions.items():
        print(f"{i}. {j}")

    user_choice = None
    while user_choice not in available_functions:
        user_choice = int(input("Enter your choice: "))

    match user_choice:
        case 1:
            read()
        case 2:
            write()
        case 3:
            update()
        case 4:
            exit()

def proceed():
    ask = input("Do you want to continue (y/n): ")
    match ask:
        case "y":
            menu()
        case "n", _:
            exit()

def write():
    name = input("Enter task name: ")
    roll_no = int(input("Enter roll no: "))
    marks = int(input("Enter marks: "))
    with open("students.dat", "ab") as file:
        pickle.dump([name, roll_no, marks], file)

proceed()
```

```
● ● ●

def read():
    with open("students.dat", "rb") as file:
        try:
            while True:
                loaded_data = pickle.load(file)
                print(loaded_data)
        except EOFError:
            pass

    proceed()

def update():
    roll_no = int(input("Enter the roll no. for which you would like to update marks: "))
    updated_marks = int(input("Enter updated marks: "))
    with open("students.dat", "rb+") as file:
        try:
            while True:
                record = pickle.load(file)
                if record[1] == roll_no:
                    record[2] = updated_marks
                    print(record)

                    file.seek(file.tell() - pickle.dumps(record).__len__())
                    pickle.dump(record, file)

                    file.seek(0, 2)
                    break
                else:
                    print("Record not found.")
        except EOFError:
            pass

    proceed()
```

Output 15:

```
● ● ●

Available functions are:
1. View Records
2. Add Records
3. Update Records
4. Exit
Enter your choice: 2
Enter name: sc
Enter roll no: 38
Enter marks: 100
Do you want to continue (y/n): y
Available functions are:
1. View Records
2. Add Records
3. Update Records
4. Exit
Enter your choice: 2
Enter name: sd
Enter roll no: 39
Enter marks: 90
Do you want to continue (y/n): y
Available functions are:
1. View Records
2. Add Records
3. Update Records
4. Exit
Enter your choice: 1
['sc', 38, 100]
['sd', 39, 90]
Do you want to continue (y/n): y
Available functions are:
1. View Records
2. Add Records
3. Update Records
4. Exit
Enter your choice: 3
Enter the roll no. for which you would like to update marks: 38
Enter updated marks: 120
['sc', 38, 120]
Do you want to continue (y/n): y
Available functions are:
1. View Records
2. Add Records
3. Update Records
4. Exit
Enter your choice: 1
['sc', 38, 120]
['sd', 39, 90]
Do you want to continue (y/n): n
```

#Program 16:

```
...
Create a binary file with name and roll number. Search for a given roll number and
display the name, if not found display appropriate message.
...

import pickle

def menu():
    available_functions = {
        1: "Add Records",
        2: "View Records",
        3: "Search Records",
        4: "Exit"
    }

    print("Available functions are: ")
    for i, j in available_functions.items():
        print(f"{i}. {j}")

    user_choice = None
    while user_choice not in available_functions:
        user_choice = int(input("Enter your choice: "))

    match user_choice:
        case 1:
            write()
        case 2:
            read()
        case 3:
            search_roll_no()
        case 4:
            exit()

def proceed():
    ask = input("Do you want to continue (y/n): ")
    match ask:
        case "y":
            menu()
        case "n", _:
            exit()

def write():
    name = input("Enter name: ")
    roll_no = int(input("Enter roll no: "))
    with open("students.dat", "ab") as file:
        pickle.dump([name, roll_no], file)

    proceed()
```

```
...  
  
def read():  
    with open("students.dat", "rb") as file:  
        try:  
            while True:  
                loaded_data = pickle.load(file)  
                print(loaded_data)  
        except EOFError:  
            pass  
  
    proceed()  
  
def search_roll_no():  
    record = False  
    searched_roll_no = int(input("Enter roll no: "))  
    with open("students.dat", "rb") as file:  
        try:  
            while True:  
                searched_data = pickle.load(file)  
                if searched_roll_no in searched_data:  
                    record = True  
                    break  
  
        except EOFError:  
            pass  
  
    if record:  
        print("Name: ", searched_data[0])  
    else:  
        print("Record not found. Please try again.")  
  
    proceed()  
  
menu()
```

Output 16:

```
● ● ●

Available functions are:
1. Add Records
2. View Records
3. Search Records
4. Exit
Enter your choice: 1
Enter name: sc
Enter roll no: 38
Do you want to continue (y/n): y
Available functions are:
1. Add Records
2. View Records
3. Search Records
4. Exit
Enter your choice: 1
Enter name: sd
Enter roll no: 39
Do you want to continue (y/n): y
Available functions are:
1. Add Records
2. View Records
3. Search Records
4. Exit
Enter your choice: 2
['sc', 38]
['sd', 39]
Do you want to continue (y/n): y
Available functions are:
1. Add Records
2. View Records
3. Search Records
4. Exit
Enter your choice: 3
Enter roll no: 38
Name: sc
Do you want to continue (y/n): n
```

#Program 17:

```
...
MySQL Connector- Program to create a connection to the database.
...

import mysql.connector

mydb= mysql.connector.connect(
    host="host",#Enter host
    user="username",#Enter your username#
    pass="password"#Enter your password
)

print(mydb)
```

Output 17:

If executed without error, the connection is established.

```
<mysql.connector.connection.MySQLConnection object at 0x016645F0>
```

#Program 18:

```
...
MySQL Connector- Program to create a database and check whether it exists or not
...

import mysql.connector

def createdb():
    mydb= mysql.connector.connect(
        host="host",#Enter host
        user="username",#Enter your username#
        passwd="password"#Enter your password
    )
    mycursor=mydb.cursor()
    mycursor.execute("CREATE DATABASE mydatabase")

def verification():
    mydb= mysql.connector.connect(
        host="host",
        user="username",
        passwd="password"
    )
    mycursor=mydb.cursor()
    mycursor.execute("SHOW DATABASES")
    for x in mycursor:
        print(x)

createdb()
verification()
```

Output 18:

The `createdb()` creates a database while the `verification()` shows:

```
('mydatabase',)
```

#Program 19:

```
...
MySQL Connector- Program to create a table with user inputs.
...

import mysql.connector

mydb = mysql.connector.connect(
    host="host",
    user="username",
    passwd="password",
    database="mydatabase"
)

mycursor = mydb.cursor()

mycursor.execute("""
CREATE TABLE student(
    ROLL_NO integer,
    NAME varchar(30),
    CLASS varchar(2),
    SECTION varchar(2),
    MARKS numeric(5,2)
)
""")

roll_no = int(input("Enter roll no.: "))
name = input("Enter name: ")
class_ = input("Enter class: ")
section = input("Enter section: ")
marks = input("Enter marks: ")

mycursor.execute(f"""
    INSERT INTO student VALUES(
        {roll_no}, {name}, {class_}, {section}, {marks}
    )
""")
mydb.commit()
```

Output 19:

```
● ● ●  
Enter roll no.: 38  
Enter name: sc  
Enter class: 12  
Enter section: B  
Enter marks: 75  
Would you like to continue inserting records?(y/n): y  
Enter roll no.: 39  
Enter name: sd  
Enter class: 12  
Enter section: C  
Enter marks: 70  
Would you like to continue inserting records?(y/n): n
```

User inputs will be entered as a new record in the table ,here, student.

The table afterwards will appear as follows:

```
● ● ●  
(38,'sc','12','B',075.00)  
(39,'sd','12','C',070.00)
```

#Program 20:

```
"""
MySQL Connector- Program to select records of students
whose percentage > 75, from table student.
"""

import mysql.connector

mydb= mysql.connector.connect(
    host="host",
    user="username",
    passwd="password",
    database="mydatabase"
)

cursor=mydb.cursor()
cursor.execute("""
    SELECT * FROM student WHERE marks > 75;
""")

rec=cursor.fetchall()
for x in rec:
    print(x)

mydb.commit()
```

Output 20:

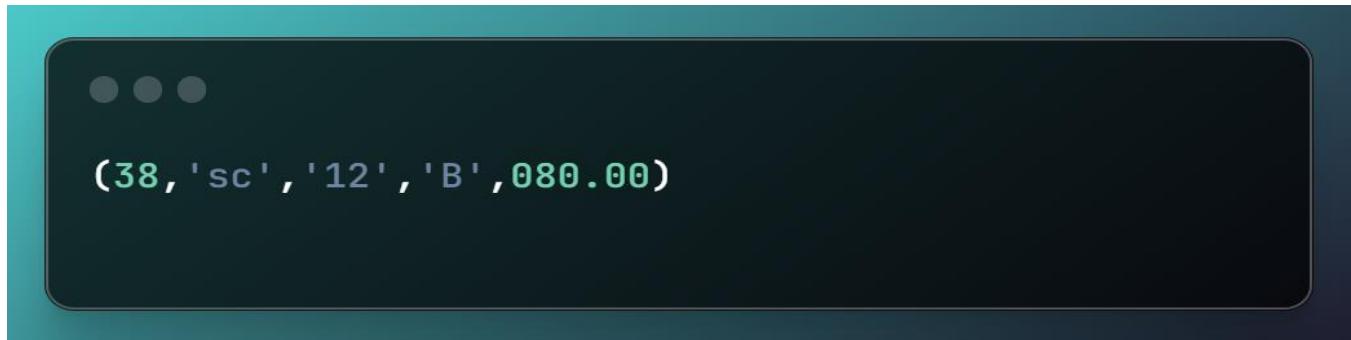
If the table consists of the following records:

(38,'sc','12','B',080.00)

(39,'sd','12','C',070.00)

Then,

For the given programme the output shall read:



=====

MySQL



INDEX

<i>S. No.</i>	<i>Content</i>	<i>Pg no.</i>
1	SET- 1	45
2	SET- 2	46-47
3	SET- 3	48-49
4	SET- 4	50-51
5	SET- 5	52-53

SET- 1

TABLE - 1

GAMES

GCODE	GAME NAME	NUMBER	PRIZE MONEY	SCHEDULED AT
101	CARROM BOARD	2	5000	12-11-2023
102	BADMINTON	2	12000	14-11-2023
103	TABLE TENNIS	4	8000	16-11-2023
105	CHESS	2	9000	05-12-2023
108	LAWN TENNIS	4	25000	07-12-2023

1. Select count(distinct number) from games;

Output: 2

2. Select max("scheduled at"), min("scheduled at") from games;

Output:

MAX(Scheduled at)	MIN(Scheduled at)
07-12-2023	12-11-2023

3. Select avg("prize money") from games;

Output: 11800

4. Select * from games where number = 2 orderby gcode desc;

Output:

GCODE	GAME NAME	NUMBER	PRIZE MONEY	SCHEDULTED AT
105	CHESS	2	9000	05-12-2023
102	BADMINTON	2	12000	14-11-2023
101	CARROM BOARD	2	5000	12-11-2023

5. Select count(gcode) from games where "prize money">> 10000 ;

Output: 2

SET- 2

TABLE – 2

1. SQL Query to create the table

```
Create table EMP(
    EMPNO integer,
    NAME varchar(30),
    DEPT varchar(10),
    DESIGNATION varchar(5),
    GENDER varchar(2),
    SALARY integer,
    CITY varchar(10)
);
```

Output: A table as defined above is formed, if no error is met during the given command's execution.

2. Insert into EMPNO values (1, 'Deepika', 'Sales', 'Manager', 'F', 55000, 'Delhi');

Insert into EMPNO values (2, 'Gagan', 'Accounts', 'Operator', 'M', 20000, 'Mumbai');

Insert into EMPNO values (3, 'Neelam', 'Computers', 'Operator', 'F', 22000, 'Kolkata');

Insert into EMPNO values (4, 'Vijay', 'Sales', 'Manager', 'M', 45000, 'Cochin');

Insert into EMPNO values (5, 'Sunitha', 'Accounts', 'Operator', 'F', 18000, 'Calicut');

Insert into EMPNO values (5, 'Rajesh', 'Sales', 'Clerk', 'M', 48000, 'Chennai');

Insert into EMPNO values (7, 'Kunal', 'Computers', 'Manager', 'M', 51000, 'Delhi');

Select * from EMP;

Output:**EMP**

EMPNO	NAME	DEPT	DESIGNATION	GENDER	SALARY	CITY
1	Deepika	Sales	Manager	F	55000	Delhi
2	Gagan	Accounts	Operator	M	20000	Mumbai
3	Neelam	Computers	Operator	F	22000	Kolkata
4	Vijay	Sales	Manager	M	45000	Cochin
5	Sunitha	Accounts	Operator	F	18000	Calicut
5	Rajesh	Sales	Clerk	M	48000	Chennai
7	Kunal	Computers	Manager	M	51000	Delhi

3. Update EMP set empno=6 where designation = 'Sales';
 Select * from EMP;

Output:

EMPNO	NAME	DEPT	DESIGNATION	GENDER	SALARY	CITY
1	Deepika	Sales	Manager	F	55000	Delhi
2	Gagan	Accounts	Operator	M	20000	Mumbai
3	Neelam	Computers	Operator	F	22000	Kolkata
4	Vijay	Sales	Manager	M	45000	Cochin
5	Sunitha	Accounts	Operator	F	18000	Calicut
6	Rajesh	Sales	Clerk	M	48000	Chennai
7	Kunal	Computers	Manager	M	51000	Delhi

4. Select gender from emp where dept is not null;

Output:

GENDER
F
M
F
M
F
M
M

5. Select sum(salary) from emp where salary >= 45000;

Output: 199000

SET- 3

TABLE – 3

COLLEGE

No	Name	Age	Department	Date of join	Basic	Sex
1	Shalaz	45	Biology	13/02/88	10500	M
2	Sameera	54	Biology	10/01/90	9500	F
3	Yagyen	43	Physics	27/02/98	8500	M
4	Pratyush	34	Chemistry	11/01/93	7500	M
5	Aren	51	Mathematics	22/01/91	8500	M
6	Reeta	27	Chemistry	14/02/94	9000	F

1. Update COLLEGE set basic = 10500 where date of join > '01/02/89' and age > 50;

Select * from college where date of join > '01/02/89' and age > 50 ;

Output:

No	Name	Age	Department	Date of join	Basic	Sex
2	Sameera	54	Biology	10/01/90	10500	F
5	Aren	51	Mathematics	22/01/91	10500	M

2. Select Name, Age and Basic from COLLEGE where department in ('Physics', 'Chemistry');

Output:

Name	Age	Basic
Yagyen	43	8500
Pratyush	34	7500
Reeta	27	9000

3. Delete from College where name= 'Shalaz';

Output:

No	Name	Age	Department	Date of Join	Basic	Sex
2	Sameera	34	Biology	10/01/90	9500	F
3	Yagyen	43	Physics	27/02/98	8500	M
4	Pratyush	34	Chemistry	11/01/93	7500	M
5	Aren	51	Mathematics	22/01/91	8500	M
6	Reeta	27	Chemistry	14/02/94	9000	F

4. Insert into college values (11, 'Saurav', 50, 'Biology', '18/05/93', 12000, 'M');

Output:

No	Name	Age	Department	Date of Join	Basic	Sex
2	Sameera	34	Biology	10/01/90	9500	F
3	Yagyen	43	Physics	27/02/98	8500	M
4	Pratyush	34	Chemistry	11/01/93	7500	M
5	Aren	51	Mathematics	22/01/91	8500	M
6	Reeta	27	Chemistry	14/02/94	9000	F
11	Saurav	50	Biology	18/05/93	12000	M

5. select distinct(department) from college;

Output:

Department
Biology
Physics
Chemistry
Mathematics

SET- 4

TABLE - 4

EMPLOYEE

No	Name	Salary	Zone	Age	Grade	Department
1	Chandan Kataruka	60000	West	28	A	10
2	Bikram Mahato	65000	Central	30	A	10
3	Rik Singh	62000	West	40	Null	20
4	Ayush Poddar	68000	North	38	C	30
5	Souvik Senapati	62000	East	26	Null	20
6	Sattyam Agarwal	67000	South	28	B	10
7	Snehashish Naik	66000	North	26	A	30

1. select sum(Salary) from Employee where Zone='West';
Output: 122000
2. select count(no) from employee where Grade is Null;
Output: 2
3. select count(grade) from Employee;
Output: 5
4. select zone, salary from Employee order by salary desc group by zone;

Output:

Zone	Salary
North	68000
North	66000
South	67000
Central	65000
West	62000
West	60000
East	62000

5. drop table employee;

Output: With this command, the table itself is dropped i.e. deleted from the database.

SET- 5

TABLE - 5

CABHUB

VehicleCode	VehicleName	Make	Colour	Capacity	Charges
100	Innova	Toyota	White	7	15
102	SX4	Suzuki	Blue	4	14
104	C Class	Mercedes	Red	4	35
105	A-Star	Suzuki	White	3	14
108	Indigo	Tata	Silver	3	12

CUSTOMER

CustomerCode	CustomerName	VehicleCode
1	Hemant Sahu	101
2	Raj Lal	108
3	Feroza Shah	105
4	Ketan Dhal	104

1. select VehicleName, Capacity from Cabhub c, Customer s where c.VehicleCode = s.VehicleCode and c.capacity < 4;

Output:

VehicleName	Capacity
A-Star	3
Indigo	3

2. select VehicleCode, VehicleName, Make from Cabhub, Customer where Cabhub.VehicleCode = Customer.VehicleCode and Cabhub.Charges > 13;

Output:

VehicleCode	VehicleName	Make
104	C Class	Mercedes
105	A-Star	Suzuki

3. desc Customer;

Output:

Field	Type	Null	Key	Default	Extra
CustomerCode	int	NO	PRI	NULL	
CustomerName	Varchar(30)	YES		NULL	
VehicleCode	int	NO		NULL	

4. select Cabhub.VehicleCode, CustomerCode, VehicleName from Cabhub, Customer orderby Cabhub.make where Cabhub.VehicleCode = Customer.VehicleCode;

Output:

VehicleCode	CustomerCode	VehicleName
104	4	C Class
105	3	A-Star
108	2	Indigo

5. Alter table Customer modify CustomerName varchar(30) NOT NULL;
desc Customer;

Output:

Field	Type	Null	Key	Default	Extra
CustomerCode	int	NO	PRI	NULL	
CustomerName	Varchar(30)	NO		NULL	
VehicleCode	int	NO		NULL	



Thank
You