CENTRAL BOARD OF SECONDARY EDUCATION

PM SHRI Kendriya Vidyalaya-3 9 B.R.D.

A.F.S. Pune- 411014

A PROJECT RECORD FILE IS SUBMITTED FOR THE COMPUTER SCIENCE, CLASS 12, SESSION 2024-25

SUBMITTED BY:- AMAN MONDAL

CLASS:- 12th A

ROLL NUMBER:- 12116

SUBJECT TEACHER(CS):- SUSHMA SINGH CHOUHAN

CERTIFICATE

This is to certify that **Aman Mondal** student of class XI, **Kendriya Vidyalaya-3 9 B.R.D. A.F.S. Pune- 411014** has completed the **Project File** during the academic year 2024-25 towards partial fulfilment of credit for the **Computer Science** project evaluation of 2024-25 and submitted satisfactory report, as compiled in the following pages, under my supervision.

Internal Examiner
Signature

External Examiner
Signature

Date: 18.10.2024

School Seal

Principal
Signature

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude and indebtedness to our learned teacher **Sushma Singh Chouhan, C.S. Teacher, Kendriya Vidyalaya-3 9 B.R.D. A.F.S. Pune- 411014** for her invaluable help, advice and guidance in the preparation of this project.

I am also greatly indebted to our Principal **Mr. Mohd. Nassimuddin** and school authorities for providing me with the facilities and requisite laboratory conditions for making this Project file.

I also extend my thanks to a number of teachers, my classmates and my friends who helped me to complete this Project file successfully.

Aman Mondal

Index

Q1. Write a Python program to generate prime numbers for a given range.

Source Code:-

1.py - F:\Aman Mondal\Practicals\1.py (3.11.4)

File Edit Format Run Options Window Help

```
def generate_primes():
    start = int(input("Enter Start of Range: "))
    end = int(input("Enter End of Range: "))
    primes = []
    for num in range(start, end + 1):
        if num > 1:
            for i in range(2, num):
                if num % i == 0:
                    break
            else:
                primes.append(num)
    print("List with Prime Numbers is-", primes)
generate_primes()
```

Output:-

IDLE Shell 3.11.4

File Edit Shell Debug Options Window Help

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: F:\Aman Mondal\Practicals\1.py
Enter Start of Range: 1
Enter End of Range: 10
List with Prime Numbers is- [2, 3, 5, 7]
>>>
```

Q2. Develop a program that takes a list of words and returns a new list containing only the words that start with a vowel.

Source Code:-

2.py - F:\Aman Mondal\Practicals\2.py (3.11.4)

File Edit Format Run Options Window Help

```
def only_vowels():
    sent = input("Enter a sentence: ")
    words = sent.split()
    newlist=[]
    for i in words:
        if i[0] in 'AEIOUaeiou':
            newlist.append(i)
    print('New list containing only the words that start with a vowel- ', newlist)

only_vowels()
```

Output:-

IDLE Shell 3.11.4

File Edit Shell Debug Options Window Help

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: F:\Aman Mondal\Practicals\2.py
Enter a sentence: Hello, My name is Aman Mondal
New list containing only the words that start with a vowel-  ['is', 'Aman']
>>>
```

Q3. Write a Python program to find and print the sum of the digits of a user-input number.

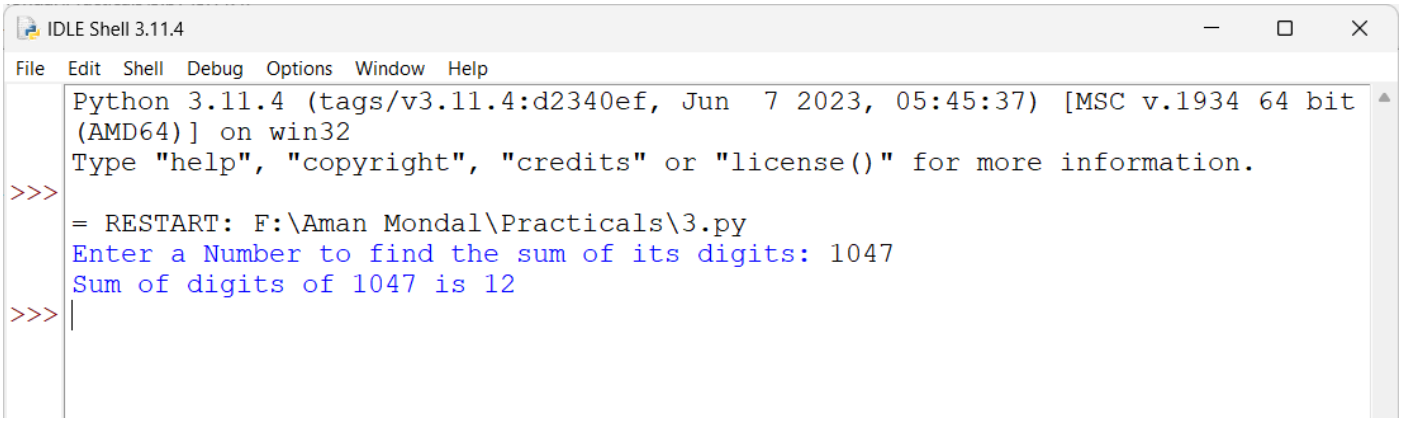
Source Code:-

3.py - F:\Aman Mondal\Practicals\3.py (3.11.4)

File Edit Format Run Options Window Help

```
num=input('Enter a Number to find the sum of its digits: ')
Sum=0
for i in range(len(num)):
    Sum+=int(num[i])
print('Sum of digits of', num, 'is', Sum)
```

Output:-



Q4. Create a program that generates a random list of numbers and finds the largest and smallest numbers without using built-in functions like max() and min().

Source Code:-

4.py - F:\Aman Mondal\Practicals\4.py (3.11.4)

File Edit Format Run Options Window Help

```
import random
n=int(input('Enter how many no you want in list of random no: '))
list1=[]
for i in range(n):
    list1.append(random.randint(0,100))
print('List with random no are: ',list1)
maximum=list1[0]
minimum=list1[0]
for j in range(len(list1)):
    if list1[j]>maximum:
        maximum=list1[j]
    if list1[j]<minimum:
        minimum=list1[j]
print('Max no in list is: ',maximum)
print('Min no in list is: ',minimum)
```

Output:-

IDLE Shell 3.11.4

File Edit Shell Debug Options Window Help

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: F:\Aman Mondal\Practicals\4.py
Enter how many no you want in list of random no: 7
List with random no are:  [90, 88, 0, 20, 11, 71, 54]
Max no in list is:  90
Min no in list is:  0
>>>
```

Q5. Create a binary file with roll number, name and class and insert 3 records in it in the format [roll-no, name, class]. Search for a given roll number and display the name, if not found display appropriate message.

Source Code:-

5.py - F:\Aman Mondal\Practicals\5.py (3.11.4)

File Edit Format Run Options Window Help

```
import pickle
def insert():
    file = open("student.dat", "wb")
    n=int(input("Enter no of records you want to enter: "))
    nested=[]
    for i in range(n):
        roll_no = int(input("Enter roll number: "))
        name = input("Enter name: ")
        Class = int(input("Enter class: "))
        d = [roll_no, name, Class]
        nested.append(d)
    pickle.dump(nested, file)
    file.close()
    print("\nRecords inserted successfully")
def search():
    file = open("student.dat", "rb")
    roll_no = int(input("Enter roll number to search: "))
    file.seek(0)
    while True:
        try:
            Data = pickle.load(file)
        except EOFError:
            break
    for i in Data:
        if i[0]==roll_no:
            print('Rollno- ',i[0], '\nName-',i[1], '\nClass- ',i[2])
            return
    else:
        print("Sorry, Student not found")

insert()
search()
```

Output:-

IDLE Shell 3.11.4

File Edit Shell Debug Options Window Help

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>

= RESTART: F:\Aman Mondal\Practicals\5.py
Enter no of records you want to enter: 3
Enter roll number: 1
Enter name: Aman
Enter class: 12
Enter roll number: 2
Enter name: Aayesha
Enter class: 12
Enter roll number: 3
Enter name: Karan
Enter class: 11

Records inserted successfully
Enter roll number to search: 1
Rollno-  1
Name- Aman
Class-  12
>>>
```

Q6. Create a binary file with book,shelf_no and author and insert 3 records in it in the format {book: [shelf_no, author]}.Search for a given author and display the name of book and shelf_no, if not found display appropriate message.

Source Code:-

6.py - F:\Aman Mondal\Practicals\6.py (3.11.4)

File Edit Format Run Options Window Help

```
import pickle
def insert():
    file = open("library.dat", "wb")
    n=int(input("Enter Number of Records you want to enter: "))
    d={}
    for i in range(n):
        book = input("Enter Book Name: ")
        shelf_no = int(input("Enter Shelf Number: "))
        author = input("Enter Author: ")
        d[book] = [shelf_no, author]
        pickle.dump(d, file)
    file.close()
    print("Records inserted successfully")

def search():
    file = open("library.dat", "rb")
    author = input("Enter Author Name to search: ")
    file.seek(0)
    while True:
        try:
            Data = pickle.load(file)
        except EOFError:
            break
    for i in Data:
        if Data[i][1]==author:
            print('Book Name-',i, '\nShelf Number',Data[i][0], '\nAuthor-',Data[i][1])
            return
    else:
        print("Sorry, Book not found")
insert()
search()
```

Output:-

IDLE Shell 3.11.4

File Edit Shell Debug Options Window Help

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: F:\Aman Mondal\Practicals\6.py
Enter Number of Records you want to enter: 2
Enter Book Name: Harry Potter
Enter Shelf Number: 1
Enter Author: J. K. Rowling
Enter Book Name: Sherlock Holmes
Enter Shelf Number: 2
Enter Author: Arthur Conan Doyle
Records inserted successfully
Enter Author Name to search: J. K. Rowling
Book Name- Harry Potter
Shelf Number 1
Author- J. K. Rowling
>>>
```

Q7. Create a CSV file by entering user-id and password, read and search the password for given user-id.

Source Code:-

7.py - F:\Aman Mondal\Practicals\7.py (3.11.4)

File Edit Format Run Options Window Help

```
import csv
def create_csv():
    file=open('users.csv', 'w', newline='')
    writer = csv.writer(file)
    n=int(input("Enter Number of values you want to enter: "))
    for i in range(n):
        user_id = input("Enter User-ID: ")
        password = input("Enter Password: ")
        writer.writerow([user_id, password])

def search_password():
    user_id=input("Enter User-ID to search: ")
    file=open('users.csv', 'r')
    reader = csv.reader(file)
    for row in reader:
        if row[0] == user_id:
            print("Password:", row[1])
            return
create_csv()
search_password()
```

Output:-

IDLE Shell 3.11.4

File Edit Shell Debug Options Window Help

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: F:\Aman Mondal\Practicals\7.py
Enter Number of values you want to enter: 3
Enter User-ID: 001
Enter Password: qwerty
Enter User-ID: 002
Enter Password: india
Enter User-ID: 003
Enter Password: password
Enter User-ID to search: 002
Password: india
>>>
```

Q8. Read a text file and display the number of vowels, consonants, uppercase and lowercase characters in the file.

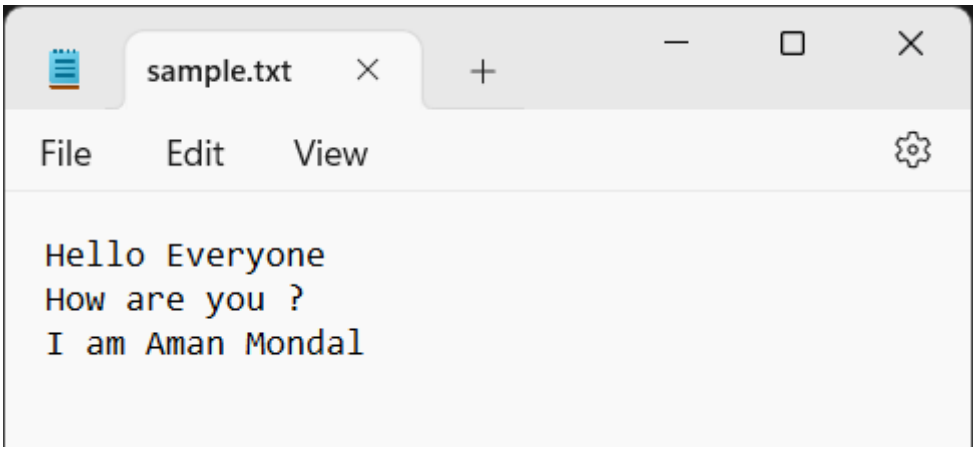
Source Code:-

8.py - F:\Aman Mondal\Practicals\8.py (3.11.4)

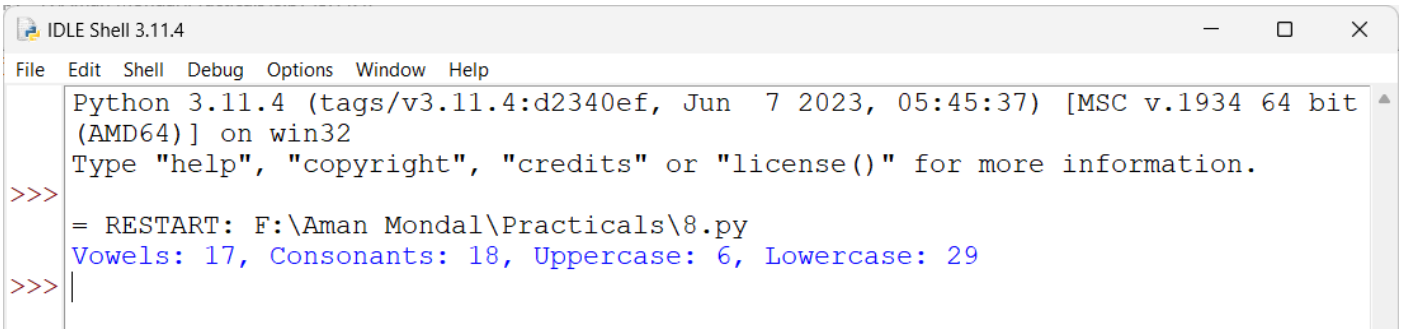
File Edit Format Run Options Window Help

```
def text_file():
    vowels = consonants = uppercase = lowercase = 0
    file= open("sample.txt", "r")
    text = file.read()
    for char in text:
        if char.isalpha():
            if char in 'AEIOUaeiou':
                vowels += 1
            else:
                consonants += 1
            if char.isupper():
                uppercase += 1
            if char.islower():
                lowercase += 1
    print(f"Vowels: {vowels}, Consonants: {consonants}, Uppercase: {uppercase}, Lowercase: {lowercase}")

text_file()
```

Sample.txt:-



Output:-



Q9. Create a program that reads a file and counts the frequency of the letter ‘H’ and ‘h’.

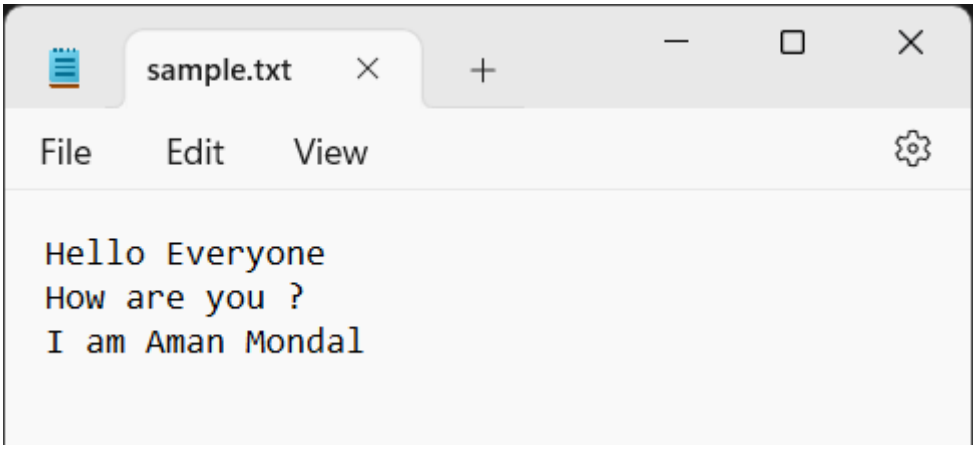
Source Code:-

9.py - F:\Aman Mondal\Practicals\9.py (3.11.4)

File Edit Format Run Options Window Help

```
def count_h_in_file():
    count_h = count_H = 0
    file = open('sample.txt', 'r')
    text = file.read()
    count_h = text.count('h')
    count_H = text.count('H')
    print(f"'h': {count_h}, 'H': {count_H}")
count_h_in_file()
```

Sample.txt:-



Output:-

IDLE Shell 3.11.4

File Edit Shell Debug Options Window Help

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: F:\Aman Mondal\Practicals\9.py
'h': 0, 'H': 2
>>>
```

Q10. Write a menu based program to perform PUSH, POP and DISPLAY operation on stack in python using list.

Source Code:-

10.py - F:\Aman Mondal\Practicals\10.py (3.11.4)

File Edit Format Run Options Window Help

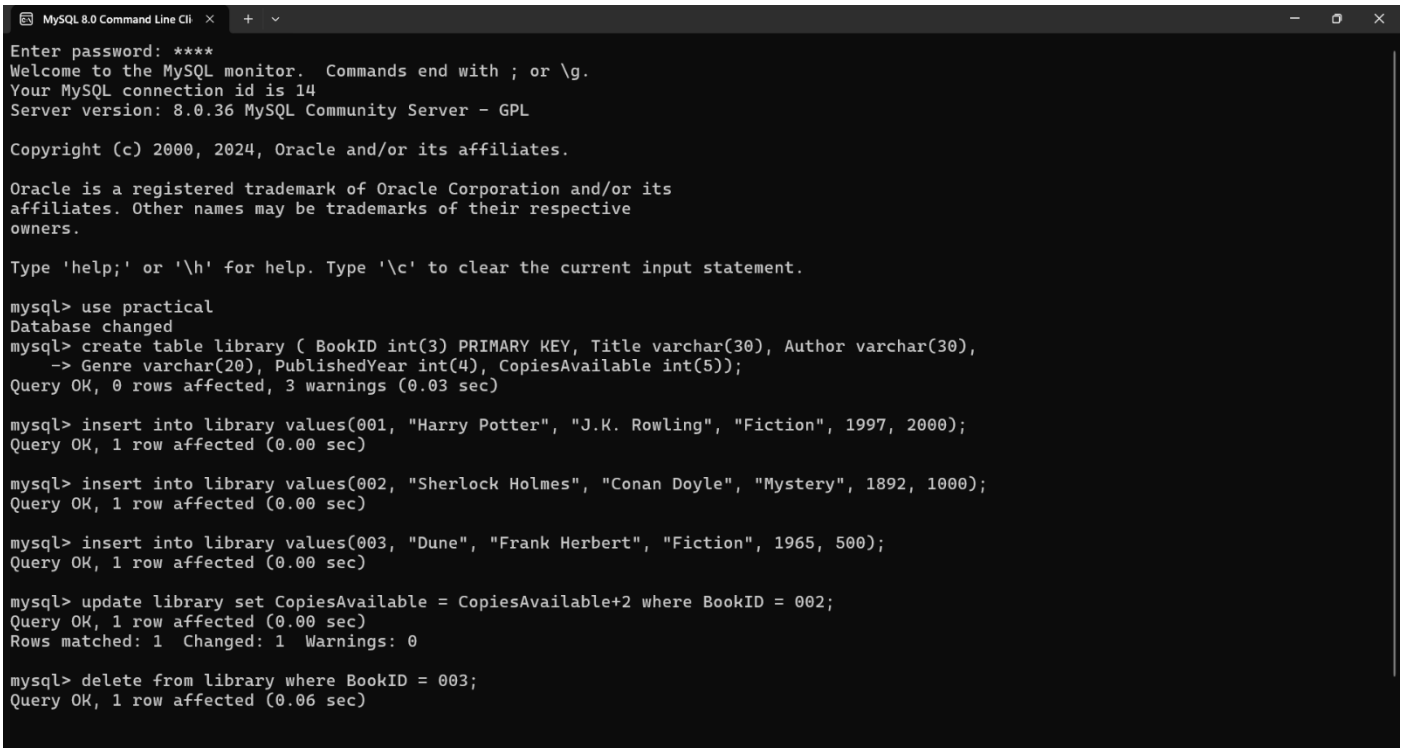
```
stack = []
def push():
    element= input("Enter element to push: ")
    stack.append(element)

def pop():
    if stack==[]:
        print("Stack is empty.")
    else:
        return stack.pop()

def display():
    if stack == []:
        print("Stack is empty.")
    else:
        print("Stack:", stack)

while True:
    print("\nMenu:")
    print("1. PUSH\n2. POP\n3. DISPLAY\n4. Exit")
    choice = int(input("Enter choice: "))

    if choice == 1:
        push()
    elif choice == 2:
        print("Popped element:", pop())
    elif choice == 3:
        display()
    elif choice == 4:
        break
    else:
        print("Invalid choice.")
```

Output:-

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: F:\Aman Mondal\Practicals\10.py

Menu:
1. PUSH
2. POP
3. DISPLAY
4. Exit
Enter choice: 1
Enter element to push: 10

Menu:
1. PUSH
2. POP
3. DISPLAY
4. Exit
Enter choice: 3
Stack: ['10']

Menu:
1. PUSH
2. POP
3. DISPLAY
4. Exit
Enter choice: 2
Popped element: 10

Menu:
1. PUSH
2. POP
3. DISPLAY
4. Exit
Enter choice: 4
```

Q11.Perform the following SQL queries
i. Write a SQL query to create a table named Library with columns BookID, Title, Author, Genre, PublishedYear, and CopiesAvailable and. And perform the following questions
ii. Write SQL queries to insert at least five records into the Library table.
iii. Write a SQL query to update the number of available copies of a specific book (e.g., increase by 2).

iv. Write a SQL query to delete a book record from the Library table where the BookID is a specific value.
Source Code And Output :-



Q12. Perform the following SQL queries

i. Create two tables: Students (with columns StudentID, Name, Age, Class) and Marks (with columns StudentID, Subject, Score). Ensure that StudentID in Marks references StudentID in Students.

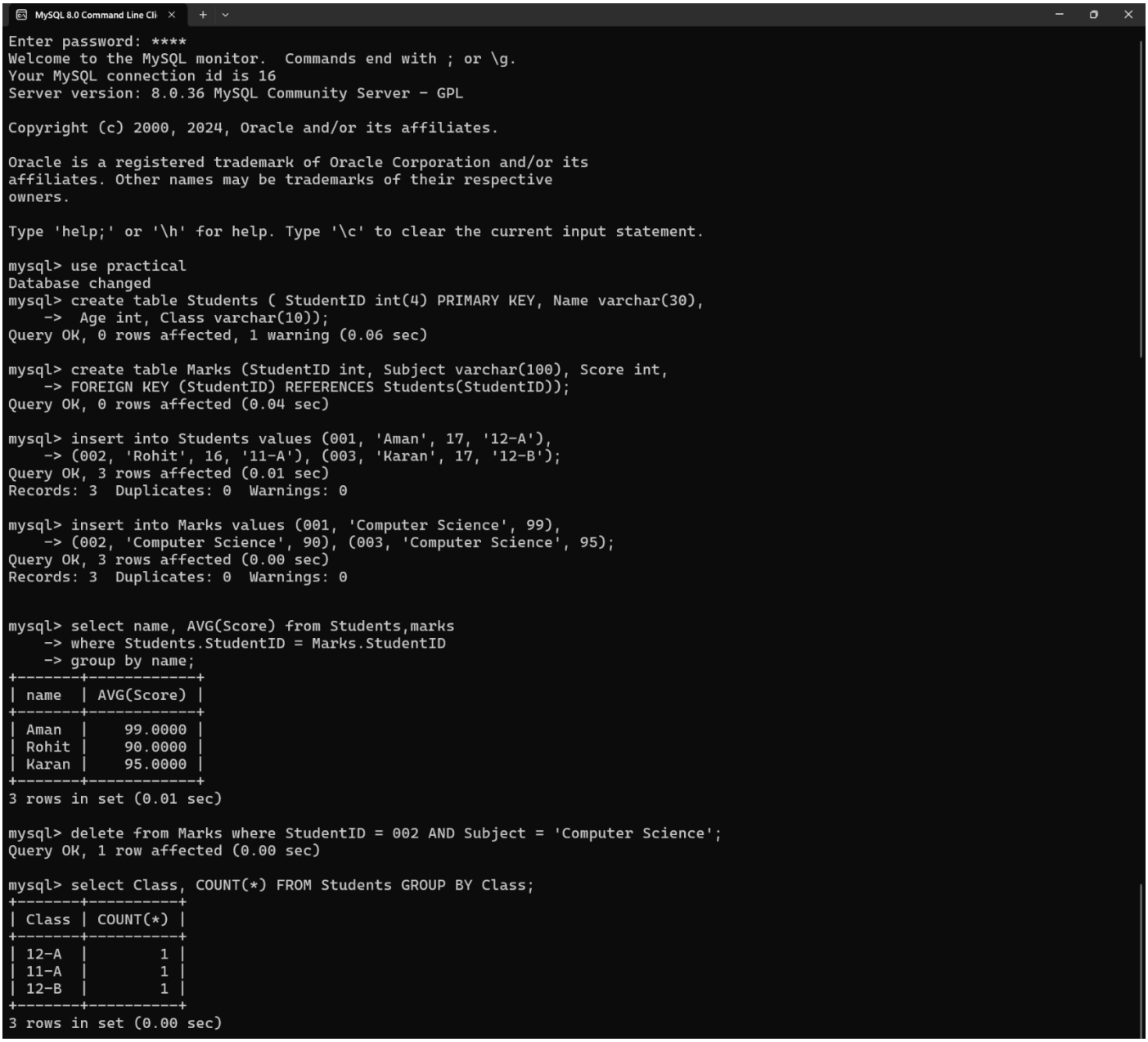
ii. Write SQL queries to insert at least three records into both the Students and Marks tables.

iii. Write a SQL query to find the average score of each student across all subjects and display the results with student names.

iv. Write a SQL query to delete a record from the Marks table for a student who has withdrawn from a subject.

v. Write a SQL query to count the total number of students in each class.

Source Code and Output:-



Q13. .Write a Python program to connect to a MySQL database named SchoolDB.

```
import mysql.connector
connection = mysql.connector.connect(
    host='localhost',user='root',
    password='root', database='SchoolDB')
cursor = connection.cursor()
```

Q14. Write a SQL query to create a table named Students with the following columns: StudentID, Name, Age, and Class. Execute this query using a Python script.

```
def create_students_table():
    cursor.execute('''create table Students (StudentID int(4) PRIMARY KEY,
Name VARCHAR(30), Age INT(2), Class VARCHAR(10))''')
```

Q15. Create a Python program that retrieves all records from the Students table and displays them in a formatted manner.

```
def retrieve_students():
    cursor.execute("select * from Students")
    rows = cursor.fetchall()
    for row in rows:
        print(f'''StudentID: {row[0]}, Name: {row[1]},
Age: {row[2]}, Class: {row[3]}''')
```

Q16.Write a Python program that updates the age of a specific student in the Students table based on StudentID.

```
def update_student_age():
    student_id= int(input("Enter Student ID"))
    new_age = int(input("Enter new age"))
    cursor.execute("update Students set Age = %s WHERE StudentID = %s",
                   (new_age, student_id))
    connection.commit()
```

Q17. Write a Python program that deletes a record from the Students table where the StudentID is a specific value.

```
def delete_student():
    student_id= int(input("Enter Student ID"))
    cursor.execute("delete from Students WHERE StudentID = %s",
                   (student_id,))
    connection.commit()
```