**PROBLEM STATEMENT:**

Develop a program in C++ to create a database of student’s information system containing the following information: Name, Roll number, Class, Division, Date of Birth, Blood group, Contact address, Telephone number, Driving license no. and other. Construct the database with suitable member functions. Make use of constructor, default constructor, copy constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete as well as exception handling.

**PROGRAM/SOURCE CODE:**

***#include* <iostream>**

**using namespace std;**

**class Student**

**{**

**public:**

**static int counterStd;**

**int localCounter = 0;**

**int roll\_number;**

**long long telePhoneNumber, DrivingLicenseNumber;**

**string name, std, division, DateOfBirth, BloodGroup, CtAddress;**

**friend class Faculty;**

**Student()**

**{**

***this*->roll\_number = 0;**

***this*->DrivingLicenseNumber = 0;**

***this*->telePhoneNumber = 0;**

***this*->name = "Student";**

***this*->std = "not filled";**

***this*->division = "not filled";**

***this*->DateOfBirth = "not filled";**

***this*->BloodGroup = "not filled";**

***this*->CtAddress = "not filled";**

**}**

**static int getCounter()**

**{**

***return* counterStd;**

**}**

**Student(int roll\_number, long long number, long long DrivingLicenseNumber, string name, string std, string division, string DateOfBirth, string BloodGroup, string CtAddress)**

**{**

***this*->roll\_number = roll\_number;**

***this*->telePhoneNumber = number;**

***this*->name = name;**

***this*->std = std;**

***this*->division = division;**

***this*->DateOfBirth = DateOfBirth;**

***this*->BloodGroup = BloodGroup;**

***this*->CtAddress = CtAddress;**

***this*->DrivingLicenseNumber = DrivingLicenseNumber;**

**}**

**Student(const Student& other) {**

***this*->roll\_number = other.roll\_number;**

***this*->telePhoneNumber = other.telePhoneNumber;**

***this*->DrivingLicenseNumber = other.DrivingLicenseNumber;**

***this*->name = other.name;**

***this*->std = other.std;**

***this*->division = other.division;**

***this*->DateOfBirth = other.DateOfBirth;**

***this*->BloodGroup = other.BloodGroup;**

***this*->CtAddress = other.CtAddress;**

***this*->localCounter = other.localCounter;**

**}**

**~Student() { cout << "\nDestructor executed of class student"; }**

**void addStudent()**

**{**

**cout << "Enter the roll number   of the student  to add  to the  list of students" << endl;**

**cin >> *this*->roll\_number;**

**cout << "Enter the name of the student to add to the list of students " << endl;**

**cin.ignore();**

**getline(cin, *this*->name);**

**cout << "Enter the year of studying the student " << endl;**

**cin >> *this*->std;**

**cout << "enter the division of the student " << endl;**

**cin >> *this*->division;**

**cout << "enter the date of the birth if the student " << endl;**

**cin >> *this*->DateOfBirth;**

**cout << "Enter the blood group of the student " << endl;**

**cin >> *this*->BloodGroup;**

**cout << "enter the telephone number of the student " << endl;**

**cin >> *this*->telePhoneNumber;**

**cout << "enter the address of the student " << endl;**

**cin.ignore();**

**getline(cin, *this*->CtAddress);**

**cout << "enter the driving license number of student" << endl;**

**cin >> *this*->DrivingLicenseNumber;**

**}**

**inline void displayData()**

**{**

**cout << "----------------------------"**

**<< "this is record no " << *this*->localCounter << "-------------------------" << endl;**

**cout << "----------------------------STUDENT INFORMATION-------------------------" << endl;**

**cout << "------------------------------------------------------------------------" << endl;**

**cout << "| ROLL NUMBER        : " << *this*->roll\_number << endl;**

**cout << "| NAME               : " << *this*->name << endl;**

**cout << "| YEAR OF STUDY      : " << *this*->std << endl;**

**cout << "| DIVISION           : " << *this*->division << endl;**

**cout << "| DATE OF BIRTH      : " << *this*->DateOfBirth << endl;**

**cout << "| BLOOD GROUP        : " << *this*->BloodGroup << endl;**

**cout << "| TELEPHONE NUMBER   : " << *this*->telePhoneNumber << endl;**

**cout << "| ADDRESS            : " << *this*->CtAddress << endl;**

**cout << "| DRIVING LICENSE NO.: " << *this*->DrivingLicenseNumber << endl;**

**cout << "------------------------------------------------------------------------" << endl;**

**cout << "----------------------------STUDENT INFORMATION-------------------------" << endl;**

**cout << "------------------------------------------------------------------------" << endl;**

**cout << endl;**

**}**

**};**

**int Student::counterStd = 0;**

**class Faculty**

**{**

**public:**

**void showDataFacWise(string divChoise, Student &student)**

**{**

***if* (student.division == divChoise)**

**{**

**student.displayData();**

**}**

**}**

**};**

**int main()**

**{**

**int counter = 0;**

**Student \*student = new Student[10];**

**int choise;**

***while* (1)**

**{**

**cout << "----------------------------------------------------------------" << endl;**

**cout << "--------              -ENTER YOUR CHOICE               --------" << endl;**

**cout << "----------------------------------------------------------------" << endl;**

**cout << "1) ADD STUDENT\n2) DISPLAY STUDENT LIST\n 3)DISPLAY FACULTY WISE LIST\n4) EXIT";**

**cout << "----------------------------------------------------------------" << endl;**

**cout << "--------             WELCOME TO STUDENT DATABASE       --------" << endl;**

**cout << "----------------------------------------------------------------" << endl;**

***try***

**{**

**cin >> choise;**

***if* (choise > 4)**

**{**

***throw* std::runtime\_error("Choise must be selected from options given above not greater than 4...");**

**}**

**}**

***catch* (const std::exception &e)**

**{**

***// Catch and handle the exception***

**std::cerr << "Exception caught: " << e.what() << std::endl;**

**}**

**string divChoise;**

***switch* (choise)**

**{**

***case* 1:**

**cout << "you have selected option 1" << endl;**

**student[counter].addStudent();**

**Student::counterStd++;**

**student[counter].localCounter = Student::counterStd;**

**counter++;**

***break*;**

***case* 2:**

**cout << "you have selected option 2" << endl;**

***for* (int i = 0; i < counter; i++)**

**{**

**student[i].displayData();**

**}**

***break*;**

***case* 3:**

***// string divChoise;***

**cout << "you have selected option 3" << endl;**

**Faculty f;**

**cout << "As you are accessing data as an faculty which division data you want to access" << endl;**

**cin >> divChoise;**

***for* (int i = 0; i < counter; i++)**

**{**

**f.showDataFacWise(divChoise, student[i]);**

**}**

***break*;**

***case* 4:**

**cout << "you have selected to exit Student database system" << endl;**

**cout << "exiting........ " << endl;**

**delete[] student;**

***break*;**

**}**

***if* (choise == 4)**

**{**

***break*;**

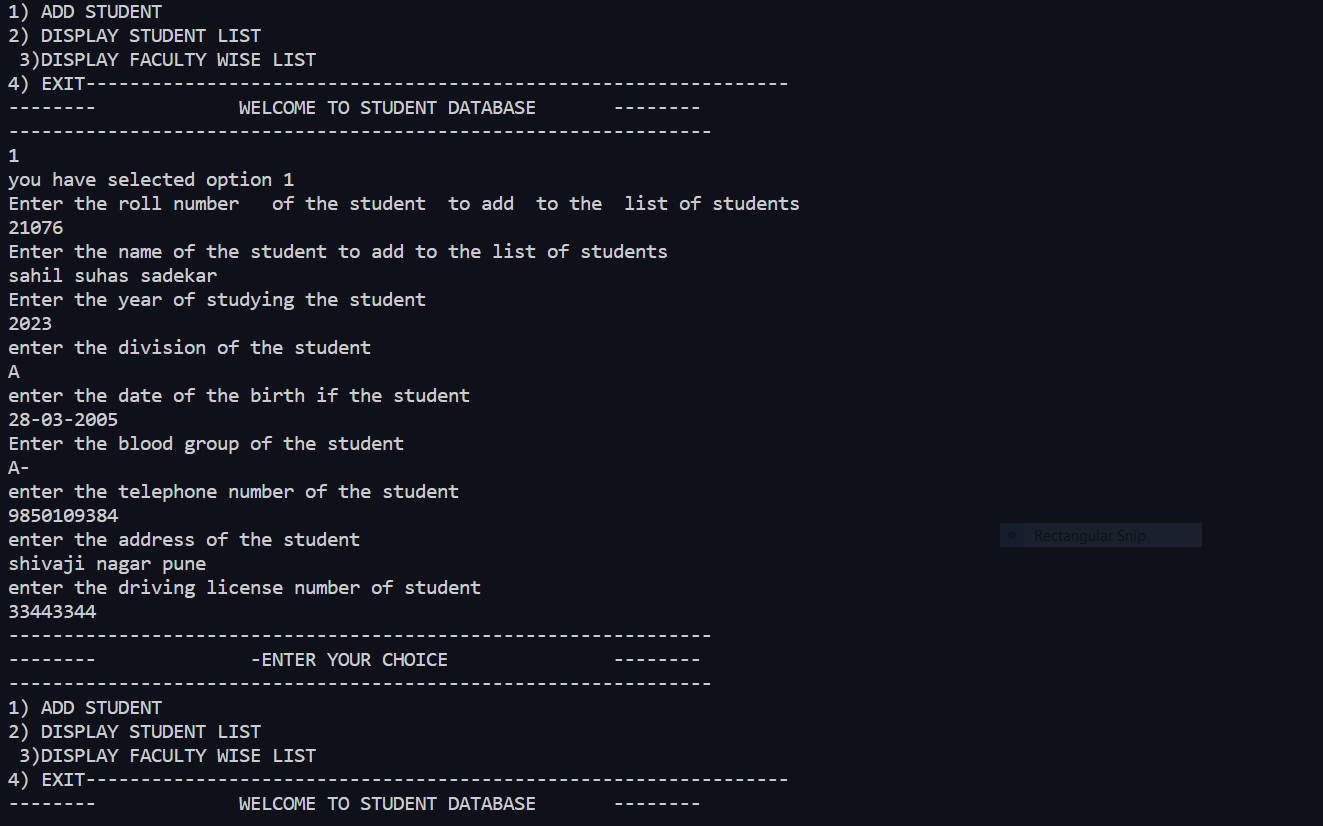
**}**

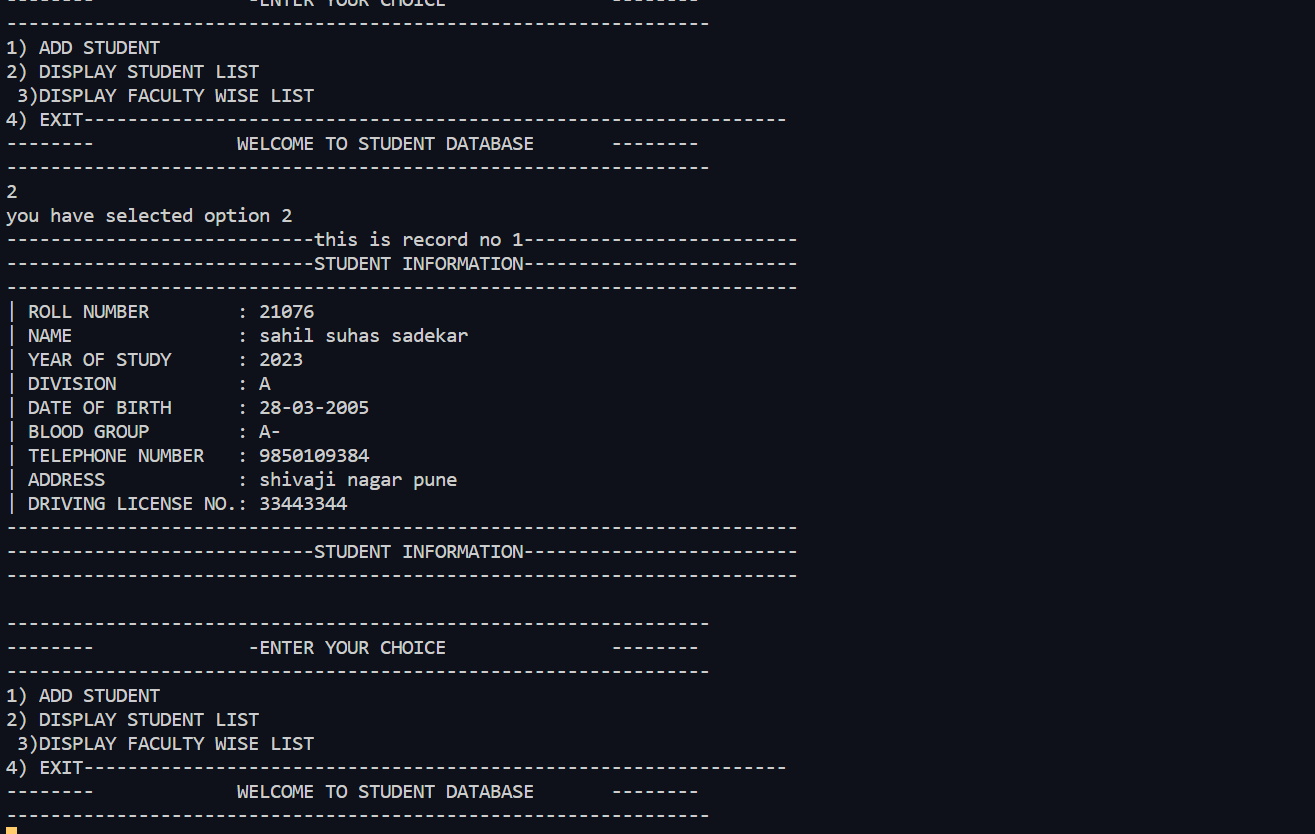
**}**

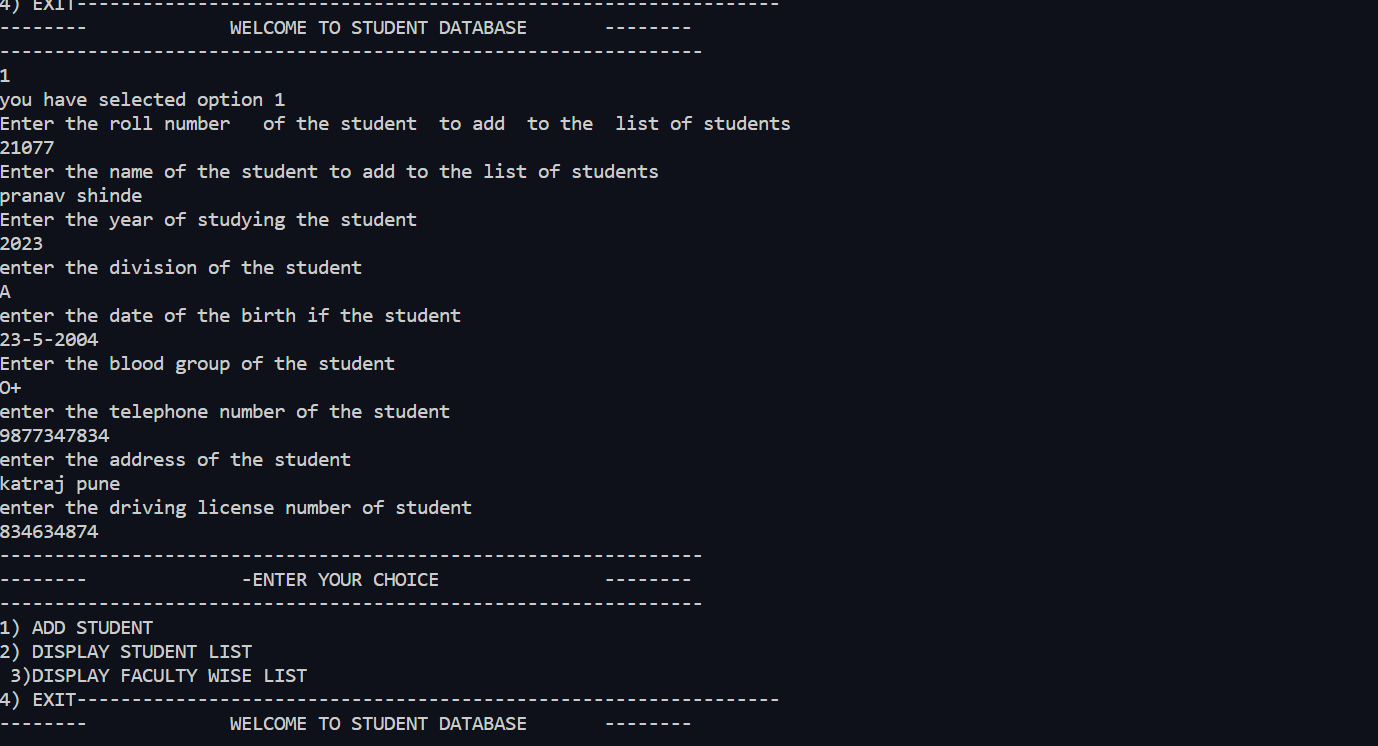
***return* 0;**

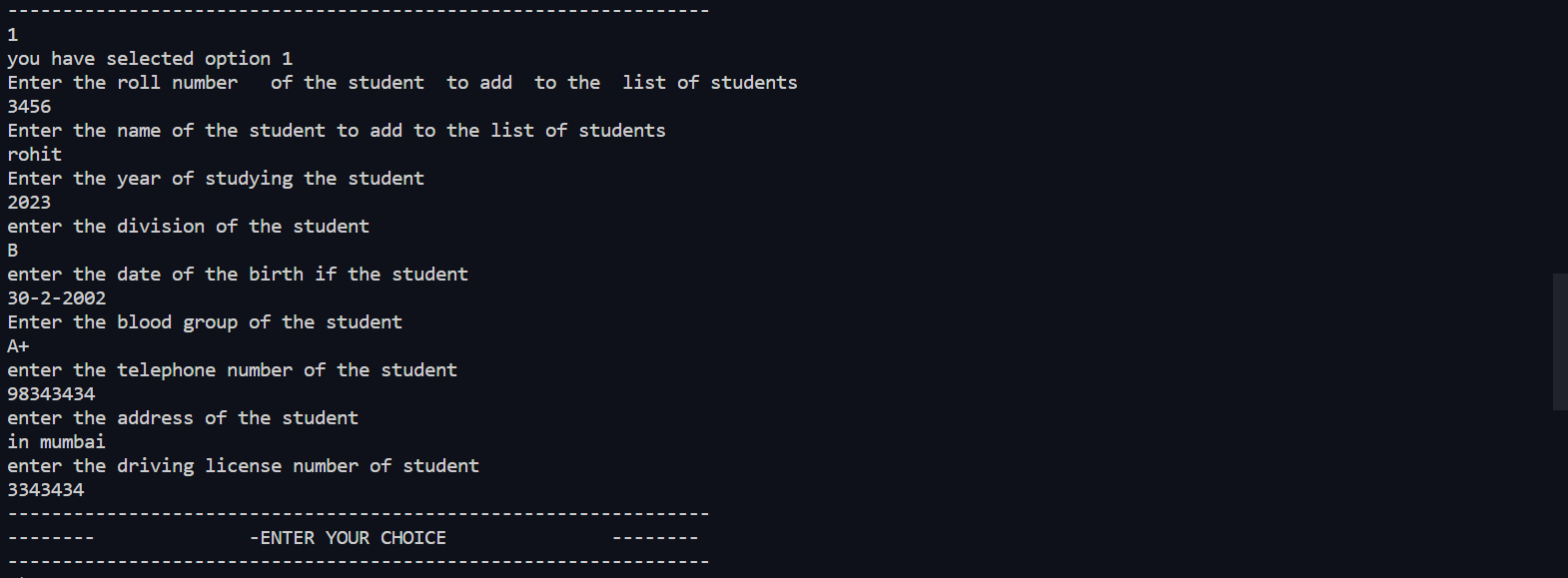
**}**

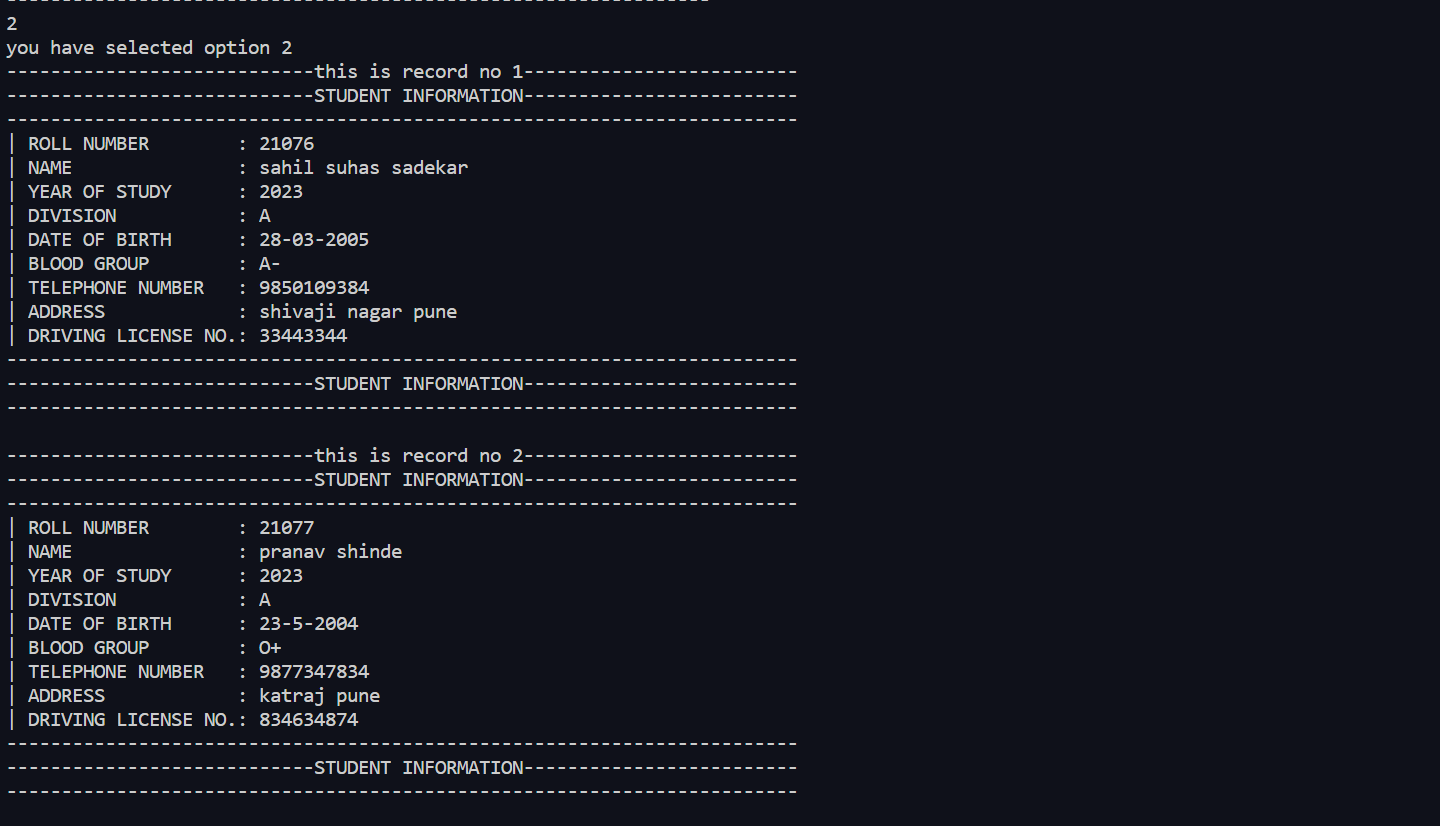
**OUTPUT:**

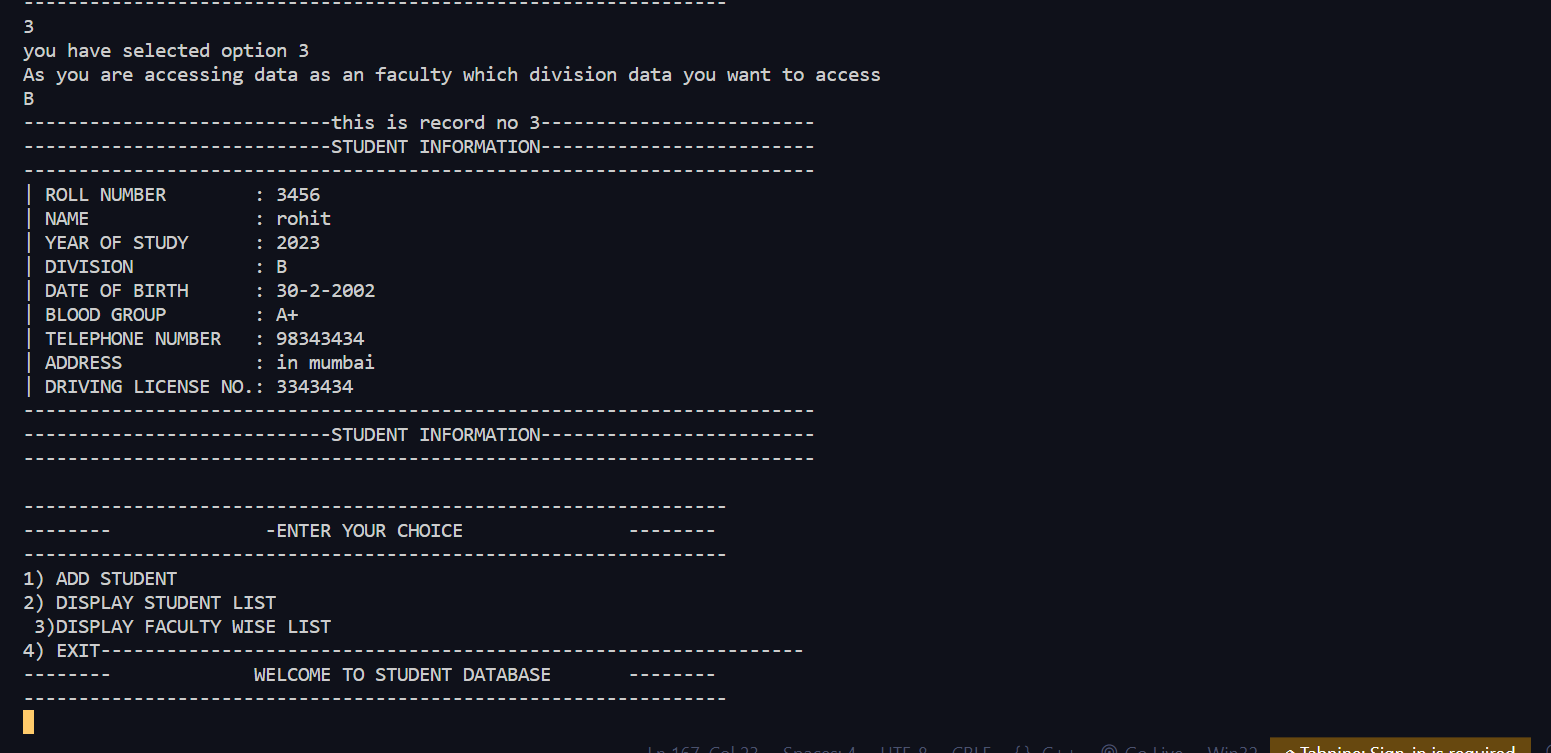
****

****

****

****

****

****