24k-0918 **DSA** LAB 02 TASKS

TASK-1

#include <iostream>

using namespace std;

class Matrix{

    int rows;

    int cols;

    int \*\*arr;

    public:

        Matrix(int *r*, int *c*) : rows(*r*), cols(*c*){

            arr = new int\*[rows];

            for(int i=0; i<rows; i++){

                arr[i] = new int[cols];

            }

        }

        void define(){

            cout<<"define Matrix: "<<endl;

            for(int i=0; i<rows; i++){

                cout<<"For row "<<i+1<<": "<<endl;

                for(int j=0; j<cols; j++){

                    cout<<"matrix["<<i+1<<"]["<<j+1<<"]: ";

                    cin>>arr[i][j];

                }

            }

        }

        void resize(){

            int newR;

            int newC;

            cout<<"Enter new size of matrix: ";

            cin>>newR>>newC;

            int \*\*newArr = new int\*[newR];

            for(int i=0; i<newR; i++){

                newArr[i] = new int[newC];

            }

            for(int i=0; i<min(rows,newR); i++){

                for(int j=0; j<min(cols, newC); j++){

                    newArr[i][j] = arr[i][j];

                }

            }

*// for if new values are there*

            cout<<"New values for resize Matrix: "<<endl;

            for(int i=0; i<newR; i++){

                for(int j=0; j<newC; j++){

                    if(i >= rows || j >= cols){

                        cout<<"matrix["<<i+1<<"]["<<j+1<<"]: ";

                        cin>>newArr[i][j];

                    }

                }

            }

            for (int i = 0; i < rows; i++) {

                delete[] arr[i];

            }

            delete[] arr;

            arr = newArr;

            rows = newR;

            cols = newC;

        }

        Matrix transpose(){

            Matrix result(cols, rows);

            for (int i = 0; i < rows; i++) {

                for (int j = 0; j < cols; j++) {

                    result.arr[j][i] = arr[i][j];

                }

            }

            return result;

        }

        void display(){

            for(int i=0; i<rows; i++){

                for(int j=0; j<cols; j++){

                    cout<<arr[i][j]<<" ";

                }

                cout<<endl;

            }

        }

        void displayAfterAddingTwo(){

            for(int i=0; i<rows; i++){

                for(int j=0; j<cols; j++){

                    if(j%2 != 0){

                        cout<<arr[i][j]+2<<" ";

                    }

                    else{

                        cout<<arr[i][j]<<" ";

                    }

                }

                cout<<endl;

            }

        }

        ~Matrix(){

            for(int i=0; i<rows; i++){

                delete[] arr[i];

            }

            delete[] arr;

        }

};

#include <iostream>

using namespace std;

int main() {

*//Create matrix*

    int r, c;

    cout << "Enter rows and cols of matrix: ";

    cin >> r >> c;

    Matrix m(r, c);

*//Define (input values)*

    m.define();

*//Display original matrix*

    cout << "\nOriginal Matrix:\n";

    m.display();

*//Resize matrix*

    cout << "\n--- Resize Matrix ---\n";

    m.resize();

    cout << "\nMatrix after resize:\n";

    m.display();

*//Transpose*

    cout << "\n--- Transpose Matrix ---\n";

    Matrix t = m.transpose(); *// new transposed matrix*

    cout << "\nMatrix after transpose:\n";

    t.display();

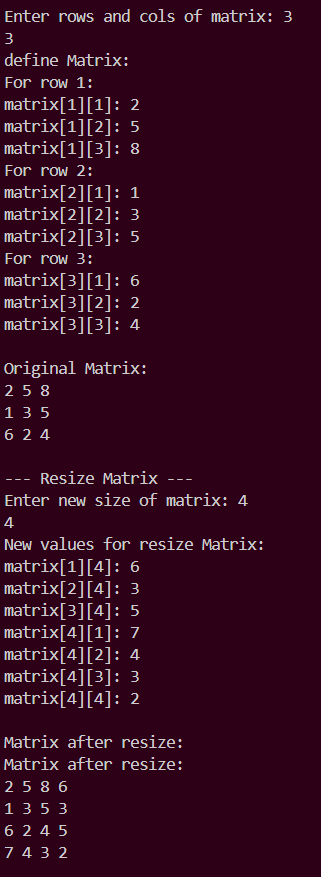
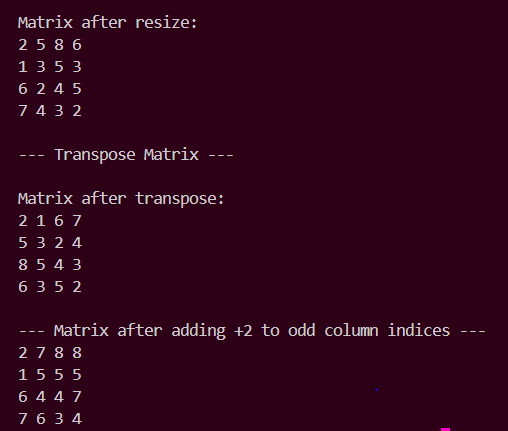
*//Display after adding +2 to odd indices*

    cout << "\n--- Matrix after adding +2 to odd column indices ---\n";

    m.displayAfterAddingTwo();

    return 0;

}

TASK-2

#include <iostream>

using namespace std;

class Exam{

    int students;

    int subjects;

    int \*\*marks;

    public:

        Exam(int *s*, int *sub*) : students(*s*), subjects(*sub*) {

            marks = new int\*[students];

            for (int i = 0; i < students; i++) {

                marks[i] = new int[subjects];

            }

        }

        void inputMarks(){

            for(int i=0; i<students; i++){

                cout<<"Marks for Student"<<i+1<<endl;

                for(int j=0; j<subjects; j++){

                    cout<<"subject "<<j+1<<": ";

                    cin>>marks[i][j];

                }

            }

        }

        void displayTotal(){

            cout<<"Total marks for all students:"<<endl;

            for(int i=0; i<students; i++){

                int total = 0;

                for(int j=0; j<subjects; j++){

                    total += marks[i][j];

                }

                cout<<"Marks for Student"<<i+1<<": "<<total<<endl;

            }

        }

        void displayAvg(){

            cout << "Average Marks of Each Subject:"<<endl;

            for (int j = 0; j < subjects; j++) {

                double sum = 0;

                for (int i = 0; i < students; i++) {

                    sum += marks[i][j];

                }

            cout<<"Subject "<<j + 1<<": "<<sum / students<<endl;

            }

        }

        void topStd(){

            int topperIdx = 0;

            int topperMarks = 0;

            for(int i=0; i<students; i++){

                int total = 0;

                for(int j=0; j<subjects; j++){

                    total += marks[i][j];

                }

                if (total > topperMarks) {

                    topperMarks = total;

                    topperIdx = i;

                }

            }

            cout<<"Topper is Student "<<topperIdx+1<<" with "<<topperMarks<<" marks"<<endl;

        }

        ~Exam() {

            for (int i = 0; i < students; i++) {

                delete[] marks[i];

            }

            delete[] marks;

        }

};

int main(){

    Exam exam(5, 4); *// 5 students, 4 subjects*

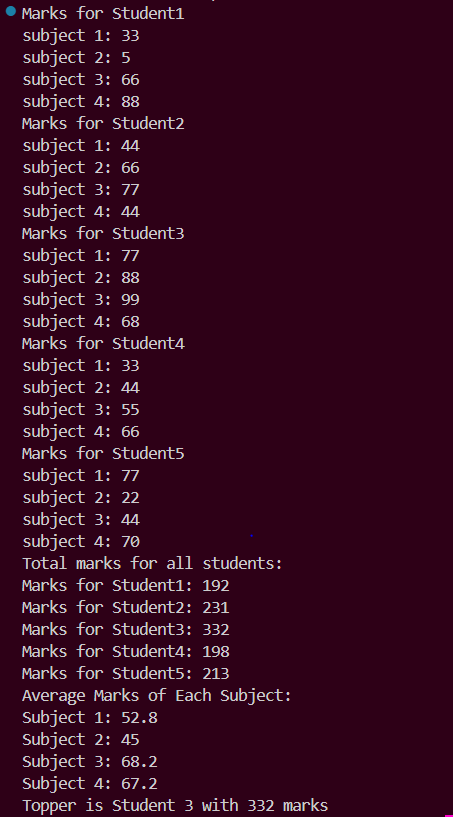
    exam.inputMarks();

    exam.displayTotal();

    exam.displayAvg();

    exam.topStd();

}



TASK-3

#include <iostream>

using namespace std;

class Student{

    int subject[5];

    int totalMarks;

    public:

        Student(){

            for(int i=0; i<5; i++) subject[i] = 0;

            totalMarks = 0;

        }

        void inputMarks(){

            totalMarks = 0;

            for(int i=0; i<5; i++){

                cout<<"for subject "<<i+1<<": ";

                cin>>subject[i];

                totalMarks += subject[i];

            }

        }

        void display(){

            for(int i=0; i<5; i++){

                cout<<subject[i]<<" ";

            }

            cout<<"Total: "<<totalMarks<<endl;

        }

        int getMarks() const{

            return totalMarks;

        }

};

void calcAndDisplay(Student \**depStd*, int *stdCount*, int *deptNo*){

    int highest = INT\_MIN;

    int lowest = INT\_MAX;

    int total = 0;

    cout<<endl<<"Department "<<*deptNo*+1<<endl;

    for(int i=0; i<*stdCount*; i++){

        cout<<"Student "<<i+1<<": ";

*depStd*[i].display();

        int marks = *depStd*[i].getMarks();

        total += marks;

        highest = max(highest, marks);

        lowest = min(lowest, marks);

    }

    cout<<"Highest: "<<highest<<endl;

    cout<<"Lowest: "<<lowest<<endl;

    cout<<"Average: "<<(double)total / *stdCount*<<endl;

}

int main(){

    int dep;

    cout<<"Enter no of dep: ";

    cin>>dep;

    int \*stdCount  = new int[dep];

    Student \*\*StudentMarks = new Student\*[dep];

    for (int i = 0; i < dep; i++) {

        cout << "Enter number of students in department " << i + 1 << ": ";

        cin >> stdCount[i];

        StudentMarks[i] = new Student[stdCount[i]];

        for (int j = 0; j < stdCount[i]; j++) {

            cout << "Enter marks of student " << j + 1 << " in dept " << i + 1 << endl;

            StudentMarks[i][j].inputMarks();

        }

    }

*// Display results for each department using function*

    for (int i = 0; i < dep; i++) {

        calcAndDisplay(StudentMarks[i], stdCount[i], i);

    }

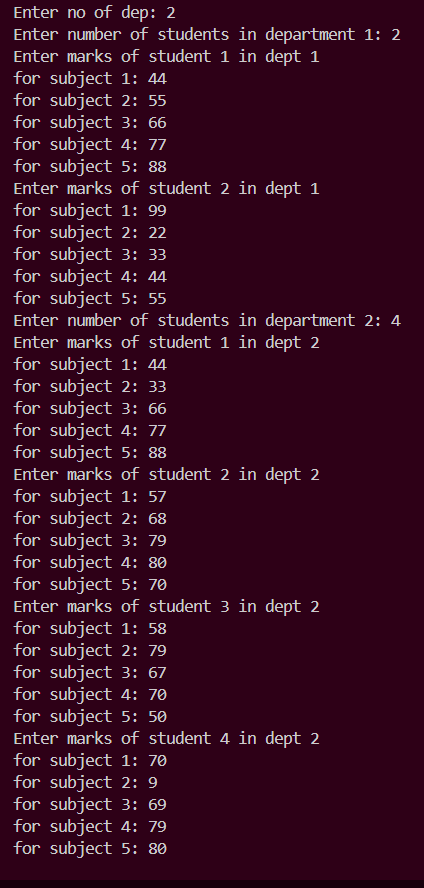
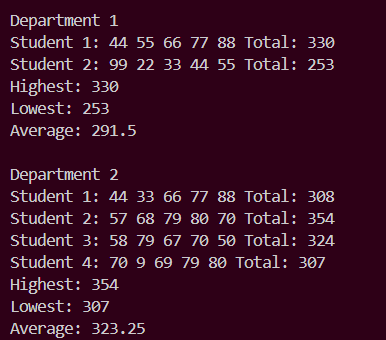
*// cleanup*

    for (int i = 0; i < dep; i++) delete[] StudentMarks[i];

    delete[] StudentMarks;

    delete[] stdCount;

}

TASK-4

#include <iostream>

using namespace std;

class Friends{

    bool matrix[5][5];

    public:

        Friends(){

            bool temp[5][5] = {

            {0, 1, 0, 1, 1}, *// 0 is friends with 1,3,4*

            {1, 0, 1, 0, 1}, *// 1 is friends with 0,2,4*

            {0, 1, 0, 0, 0}, *// 2 is friends with 1*

            {1, 0, 0, 0, 1}, *// 3 is friends with 0,4*

            {1, 1, 0, 1, 0} *// 4 is friends with 0,1,3*

            };

            for(int i=0; i<5; i++){

                for(int j=0; j<5; j++){

                    matrix[i][j] = temp[i][j];

                }

            }

        }

        bool hasCommonFrnd(int *f1*, int *f2*){

            for(int i=0; i<5; i++){

                if(i != *f1* && i != *f2*){

                    if(matrix[*f1*][i] && matrix[*f2*][i]) {

                        return true;

                    }

                }

            }

            return false;

        }

        void displayMatrix() {

            cout << "Friendship Matrix (1 = friends, 0 = not friends):\n";

            for(int i=0;i<5;i++) {

                for(int j=0;j<5;j++) {

                    cout << matrix[i][j] << " ";

                }

                cout << endl;

            }

        }

};

int main(){

    Friends f;

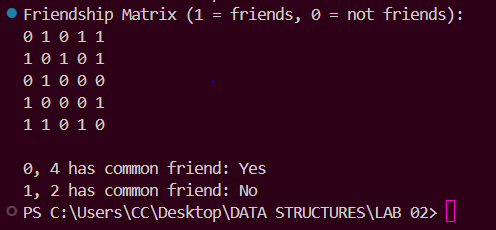
    f.displayMatrix();

    cout<<endl;

    cout<<"0, 4 has common friend: "<<(f.hasCommonFrnd(0,4) ? "Yes" : "No")<<endl;

    cout<<"1, 2 has common friend: "<<(f.hasCommonFrnd(1,2) ? "Yes" : "No")<<endl;

}



TASK-5

#include <iostream>

#include <iomanip>

using namespace std;

class ManageGPA{

    int departments;

    int \*course;

    double \*\*gpa;

    public:

        ManageGPA() : departments(4){

            course = new int[departments]{3, 4, 2, 1};

            gpa = new double\*[departments];

            for(int i=0; i<departments; i++){

                gpa[i] = new double[course[i]];

            }

        }

        void inputDetails(){

            string deptNames[4] = {"Software Engineering (SE)",

                                  "Artificial Intelligence (AI)",

                                  "Computer Science (CS)",

                                  "Data Science (DS)"};

            for(int i=0; i< departments; i++){

                cout<<"Enter GPA for department "<<deptNames[i]<<": "<<endl;

                for(int j=0; j<course[i]; j++){

                    cout<<"Core course "<<j+1<<": ";

                    cin>>gpa[i][j];

                }

            }

        }

        void displayDetails(){

            cout<<endl;

            string deptNames[4] = {"Software Engineering (SE)",

                                  "Artificial Intelligence (AI)",

                                  "Computer Science (CS)",

                                  "Data Science (DS)"};

            cout<<fixed<<setprecision(2);

            for(int i=0; i< departments; i++){

                cout<<"GPA for department "<<deptNames[i]<<": "<<endl;

                for(int j=0; j<course[i]; j++){

                    cout<<"Core course "<<j+1<<": "<<gpa[i][j]<<endl;;

                }

            }

        }

        ~ManageGPA(){

            for(int i=0; i<departments; i++){

                delete[] gpa[i];

            }

            delete[] gpa;

            delete[] course;

        }

};

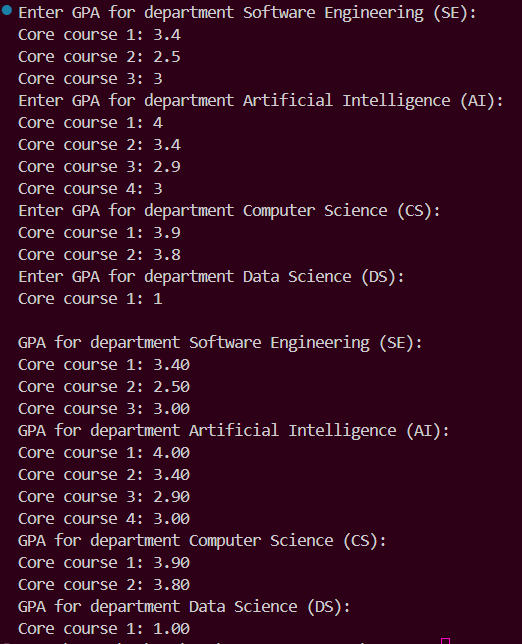
int main(){

    ManageGPA gpa;

    gpa.inputDetails();

    gpa.displayDetails();

}



TASK-6

#include <iostream>

#include <string>

using namespace std;

class SeatingChart{

    int rows;

    int \*seats;

    string \*\*chart;

    public:

        SeatingChart(int *r*) : rows(*r*){

            seats = new int[rows];

            for(int i=0; i<rows; i++){

                cout<<"Enter no of seats for row "<<i+1<<": ";

                cin>>seats[i];

            }

            chart = new string\*[rows];

            for(int i=0; i<rows; i++){

                chart[i] = new string[seats[i]];

            }

        }

        void inputNames(){

            cin.ignore();

            for(int i=0; i<rows; i++){

                for(int j=0; j<seats[i]; j++){

                    cout<<"Enter name of row "<<i+1<<" seat "<<j+1<<": ";

                    getline(cin, chart[i][j]);

                }

            }

        }

        void display(){

            cout<<endl<<"Seating Chart"<<endl;

            for(int i=0; i<rows; i++){

                cout<<"Row "<<i+1<<": ";

                for(int j=0; j<seats[i]; j++){

                    cout<<chart[i][j]<<", ";

                }

                cout<<endl;

            }

        }

        ~SeatingChart(){

            for(int i=0; i<rows; i++){

                delete[] chart[i];

            }

            delete[] chart;

            delete[] seats;

        }

};

int main(){

    int rows;

    cout<<"Enter total rows for chart: ";

    cin>>rows;

    SeatingChart s(rows);

    s.inputNames();

    s.display();}

