1.

#include <iostream>

using namespace std;

class Matrix {

    int rows, cols;

    int \*\*data;

public:

    Matrix(int r = 0, int c = 0) : rows(r), cols(c) {

        data = new int\*[rows];

        for (int i = 0; i < rows; i++) data[i] = new int[cols]{0};

    }

    void Fill() {

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

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

                cout << "Enter value at (" << i + 1 << "," << j + 1 << "): ";

                cin >> data[i][j];

            }

    }

    void ResizeMatrix(int newRows, int newCols) {

        int \*\*newData = new int\*[newRows];

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

            newData[i] = new int[newCols];

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

                if (i < rows && j < cols) newData[i][j] = data[i][j];

                else newData[i][j] = -1;

            }

        }

        for (int i = 0; i < rows; i++) delete[] data[i];

        delete[] data;

        data = newData;

        rows = newRows;

        cols = newCols;

    }

    void Transpose() {

        int \*\*newData = new int\*[cols];

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

            newData[i] = new int[rows];

            for (int j = 0; j < rows; j++) newData[i][j] = data[j][i];

        }

        for (int i = 0; i < rows; i++) delete[] data[i];

        delete[] data;

        data = newData;

        swap(rows, cols);

    }

    void print() {

        cout << "BEFORE:\n";

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

            cout << "[ ";

            for (int j = 0; j < cols; j++) cout << data[i][j] << " ";

            cout << "]\n";

        }

        cout << "AFTER:\n";

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

            cout << "[ ";

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

                int val = data[i][j];

                if ((i + j) % 2 == 1) val += 2;

                cout << val << " ";

            }

            cout << "]\n";

        }

    }

    ~Matrix() {

        for (int i = 0; i < rows; i++) delete[] data[i];

        delete[] data;

    }

};

int main() {

    Matrix m(2, 3);

    m.Fill();

    cout << "\nOriginal Matrix:\n";

    m.print();

    cout << "\nAfter Resizing to 3x4:\n";

    m.ResizeMatrix(3, 4);

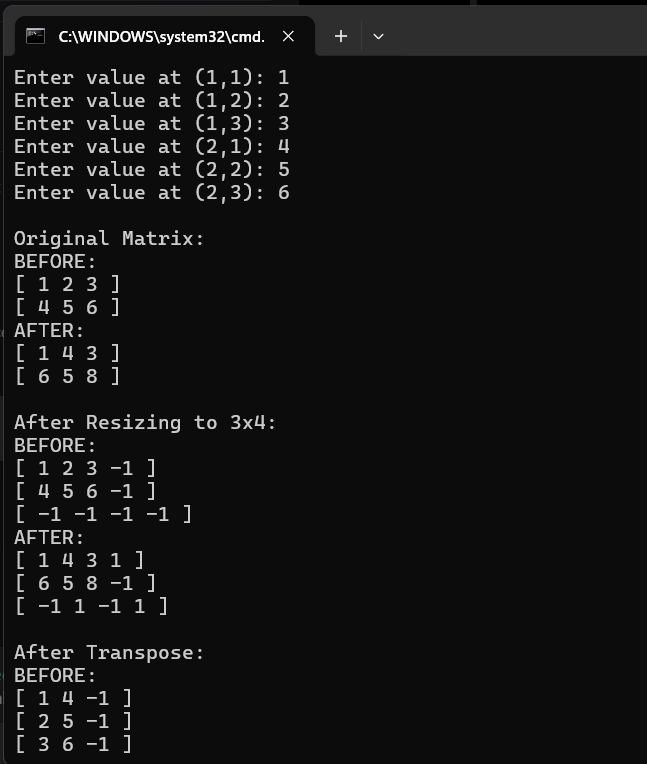
    m.print();

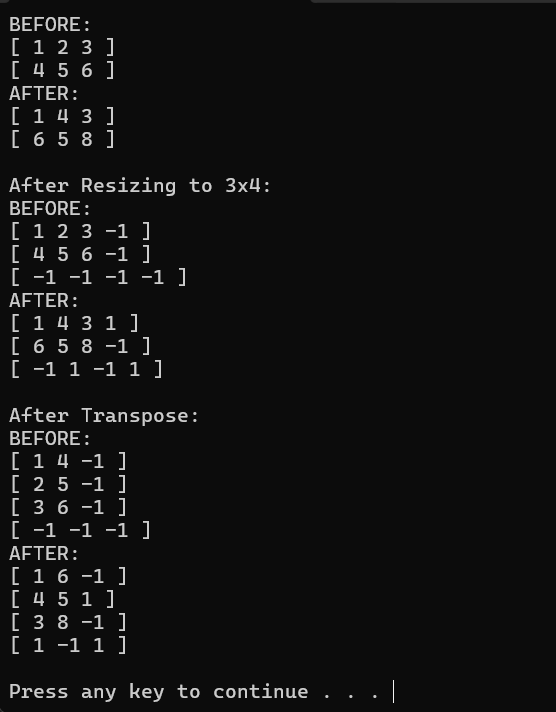
    cout << "\nAfter Transpose:\n";

    m.Transpose();

    m.print();

}





2.

#include <iostream>

using namespace std;

int main() {

    int students = 5, subjects = 4;

    int \*\*marks = new int\*[students];

    for (int i = 0; i < students; i++) marks[i] = new int[subjects];

    cout << "Enter marks of " << students << " students in " << subjects << " subjects:\n";

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

        cout << "Student " << i + 1 << ":\n";

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

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

            cin >> marks[i][j];

        }

    }

    int \*totalMarks = new int[students];

    int topperIndex = 0;

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

        totalMarks[i] = 0;

        for (int j = 0; j < subjects; j++) totalMarks[i] += marks[i][j];

        cout << "Total marks of Student " << i + 1 << " = " << totalMarks[i] << endl;

        if (totalMarks[i] > totalMarks[topperIndex]) topperIndex = i;

    }

    cout << "\nAverage marks of each subject:\n";

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

        float sum = 0;

        for (int i = 0; i < students; i++) sum += marks[i][j];

        cout << "Subject " << j + 1 << " Average = " << (sum / students) << endl;

    }

    cout << "\nTopper Student is Student " << (topperIndex + 1)

         << " with total marks = " << totalMarks[topperIndex] << endl;

    for (int i = 0; i < students; i++) delete[] marks[i];

    delete[] marks;

    delete[] totalMarks;

}

