

Лабораторна робота 8

Тема. Двовимірні масиви. Робота з файлами.

Завдання:

Складіть програму, яка здійснює обробку вхідних даних за допомогою двовимірних масивів.

При виконанні лабораторної роботи слід врахувати наступне:

1. Дати користувачеві можливість вибору: вводити вхідні дані з консолі, чи вводити їх з файлу; у випадку вводу з файлу продемонструвати дві можливості:

а) можливість користувача задати ім'я файлу в діалозі;

б) можливість перенапрявлення потоку вводу/виводу на файл командному рядку.

2. При програмуванні однотипних підзадач (сортування, ввід/вивід масивів, знаходження суми стовпця матриці тощо) слід оформлювати їх у вигляді функцій.

Варіант 1

Задайте матрицю A розмірності $(n \times m)$, де n і $m \geq 6$. Виконайте такі завдання:

1. знайдіть максимальний і мінімальний елементи матриці і поміняйте їх місцями;

2. знайдіть суму і добуток кожного рядка матриці A ;

3. упорядкуйте за зростанням значення елементів рядка і стовпця, де знаходиться максимальний елемент.

Код

```
#include <iostream>
#include <Windows.h>
#include <fstream>
#include <vector>
#include <algorithm>

using namespace std;

vector<vector<int>> readMatrixFromConsole(int rows, int cols) {
    vector<vector<int>> matrix(rows, vector<int>(cols));

    cout << "Введіть елементи матриці:" << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
```

```

        cout << "Елемент [" << i << "][" << j << "]: ";
        cin >> matrix[i][j];
    }
}

return matrix;
}

vector<vector<int>> readMatrixFromFile(const string& filename) {
    vector<vector<int>> matrix;

    ifstream file(filename);
    if (file.is_open()) {
        int rows, cols;
        file >> rows >> cols;

        matrix.resize(rows, vector<int>(cols));

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                file >> matrix[i][j];
            }
        }

        file.close();
    }
    else {
        cout << "Не вдалося відкрити файл!" << endl;
    }

    return matrix;
}

void writeMatrixToFile(const vector<vector<int>>& matrix, const string& filename) {
    ofstream file(filename);

    if (file.is_open()) {
        int rows = matrix.size();
        int cols = matrix[0].size();

        file << rows << " " << cols << endl;

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                file << matrix[i][j] << " ";
            }
            file << endl;
        }

        file.close();
    }
    else {
        cout << "Не вдалося відкрити файл!" << endl;
    }
}

void swapMinMaxElements(vector<vector<int>>& matrix) {
    int rows = matrix.size();
    int cols = matrix[0].size();

    int maxElement = matrix[0][0];
    int minElement = matrix[0][0];
    int maxRow = 0;
    int maxCol = 0;
    int minRow = 0;
    int minCol = 0;

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

```

```

        for (int j = 0; j < cols; j++) {
            if (matrix[i][j] > maxElement) {
                maxElement = matrix[i][j];
                maxRow = i;
                maxCol = j;
            }
            if (matrix[i][j] < minElement) {
                minElement = matrix[i][j];
                minRow = i;
                minCol = j;
            }
        }
    }

    swap(matrix[maxRow][maxCol], matrix[minRow][minCol]);
}

void calculateRowSumAndProduct(const vector<vector<int>>& matrix, vector<int>&
rowSums, vector<int>& rowProducts) {
    int rows = matrix.size();
    int cols = matrix[0].size();

    rowSums.resize(rows);
    rowProducts.resize(rows);

    for (int i = 0; i < rows; i++) {
        int sum = 0;
        int product = 1;

        for (int j = 0; j < cols; j++) {
            sum += matrix[i][j];
            product *= matrix[i][j];
        }

        rowSums[i] = sum;
        rowProducts[i] = product;
    }
}

void sortRowAndColumn(vector<vector<int>>& matrix) {
    int rows = matrix.size();
    int cols = matrix[0].size();

    int maxElement = matrix[0][0];
    int maxRow = 0;
    int maxCol = 0;

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (matrix[i][j] > maxElement) {
                maxElement = matrix[i][j];
                maxRow = i;
                maxCol = j;
            }
        }
    }

    sort(matrix[maxRow].begin(), matrix[maxRow].end());
    for (int i = 0; i < rows; i++) {
        swap(matrix[i][maxCol], matrix[i][maxRow]);
    }
}

void printMatrix(const vector<vector<int>>& matrix) {
    int rows = matrix.size();
    int cols = matrix[0].size();

    cout << "Матрица:" << endl;

```

```

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                cout << matrix[i][j] << " ";
            }
            cout << endl;
        }
    }

int main() {

    setlocale(LC_ALL, "Russian");
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);

    int n, m;
    cout << "Введіть розмірність матриці (n x m): ";
    cin >> n >> m;

    vector<vector<int>> matrix;

    char choice;
    cout << "Виберіть спосіб введення даних (к - консоль, ф - файл): ";
    cin >> choice;

    if (choice == 'к') {
        matrix = readMatrixFromConsole(n, m);
    }
    else if (choice == 'ф') {
        string filename;
        cout << "Введіть назву файлу: ";
        cin >> filename;

        matrix = readMatrixFromFile(filename);
    }

    if (!matrix.empty()) {
        printMatrix(matrix);

        swapMinMaxElements(matrix);
        cout << "Після обміну максимального і мінімального елементів:" << endl;
        printMatrix(matrix);

        vector<int> rowSums, rowProducts;
        calculateRowSumAndProduct(matrix, rowSums, rowProducts);
        cout << "Сума і добуток кожного рядка:" << endl;
        for (int i = 0; i < rowSums.size(); i++) {
            cout << "Рядок " << i << ": Сума = " << rowSums[i] << ", Добуток = " <<
rowProducts[i] << endl;
        }

        sortRowAndColumn(matrix);
        cout << "Після упорядкування рядка і стовпця з максимальним елементом:" <<
endl;
        printMatrix(matrix);

        string outputFilename;
        cout << "Введіть назву файлу для запису результату: ";
        cin >> outputFilename;
        writeMatrixToFile(matrix, outputFilename);
        cout << "Результат був записаний у файл " << outputFilename << endl;
    }

    return 0;
}

```