

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 7
з дисципліни «Алгоритмізація та програмування»

Тема: *«Реалізація алгоритмів обробки двовимірних масивів мовою C ++»*

XAI.301. 174. 319. 8 ЛР

Виконав студент гр. 319

Шаньгін Андрій Андрійович
(підпис, дата) (П.І.Б.)

Перевірив
_____ к.т.н., доц. Олена ГАВРИЛЕНКО
(підпис, дата) (П.І.Б.)

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення двовимірних масивів (матриць) у мові C++ і реалізувати декларацію, введення з консолі, обробку виведення в консоль матриць мовою C++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на аналіз і виведення елементів матриці.

Введення і виведення даних здійснити в командному вікні. Знайти номер першого з її стовпців, що складається тільки з непарних чисел. Якщо таких стовпців немає, то виведіть 0.

Завдання 2. Перетворити матрицю відповідно до свого варіанту завданн

Дана матриця розміру $M \times N$ і ціле число K ($1 \leq K \leq N$). Видалити стовпець матриці з номером K .

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі *розділ і номер 35*

Вхідні дані (ім'я, опис, тип, обмеження):

максимальний розмір $M = N = 20$, ціле, константа;

2) кількість рядків *row*, ціле, 2..20

2) кількість стовпців *col*, ціле, 2..20

2) цілочисельний двовимірний масив *matr1*.

Вихідні дані (ім'я, опис, тип):

елементи матриці;

Алгоритм вирішення показано на рис. 1

```
Введіть кількість рядків (2-20): 3
Введіть кількість стовпців (2-20): 4
Введіть елементи матриці:
1135
79 11 4
13 15 11 6
2 4 6 8
Початкова матриця:
1135      79      11      4
13      15      11      6
2       4       6      8
Номер першого стовпця з непарними числами: 0
```

Рисунок 1 – <вирішення першого завдання>

Лістинг коду вирішення задачі *розділ і номер задачі(i)* наведено в дод. А (стор. x). Екран роботи програми показаний на рис. 1

Завдання 2.

Вирішення задачі *розділ і номер 62*

Вхідні дані (ім'я, опис, тип, обмеження):

максимальний розмір $M = N = 20$, ціле, константа;

2) кількість рядків *row*, ціле, 2..20

2) кількість стовпців *col*, ціле, 2..20

2) цілочисельний двовимірний масив *matr1*.

Вихідні дані (ім'я, опис, тип):

елементи матриці;

Алгоритм вирішення показано на рис. 2

```
Введіть номер стовпця для видалення (1-4): 1
Матриця після видалення стовпця 1:
79      11      4
15      11      6
4        6      8
```

Рисунок 2 – <вирішення першого завдання>

Лістинг коду вирішення задачі *розділ і номер задачі(i)* наведено в дод. А (стор. x). Екран роботи програми показаний на рис. 2

ВИСНОВКИ

У ході вивчення теми було розглянуто основи представлення двовимірних масивів у мові C++, включаючи їхню декларацію, ініціалізацію, введення та виведення. Було реалізовано алгоритми для обробки матриць, що дозволяють виконувати операції пошуку, модифікації та видалення елементів.

Практична реалізація в середовищі Visual Studio дала змогу закріпити теоретичні знання, зокрема в роботі з динамічною та статичною пам'яттю, а також у використанні вкладених циклів для обробки елементів матриці. Крім того, було приділено увагу коректному введенню та виведенню даних, що є важливим для створення зручних і зрозумілих користувацьких програм.

Отримані знання та практичний досвід можуть бути застосовані для розв'язання широкого спектра задач, зокрема у сфері чисельних розрахунків, комп'ютерної графіки, штучного інтелекту та обробки великих обсягів даних.

```
#include <iostream>
#include <vector>

using namespace std;

void printMatrix(const vector<vector<int>>& matrix) {
    for (const auto& row : matrix) {
        for (int elem : row) {
            cout << elem << "t";
        }
        cout << endl;
    }
}

int findFirstOddColumn(const vector<vector<int>>& matrix) {
    int row = matrix.size();
    int col = matrix[0].size();

    for (int j = 0; j < col; j++) {
        bool allOdd = true;
        for (int i = 0; i < row; i++) {
            if (matrix[i][j] % 2 == 0) {
                allOdd = false;
                break;
            }
        }
        if (allOdd) return j + 1; // возвращаем номер столбца (с 1)
    }
    return 0;
}

void deleteColumn(vector<vector<int>>& matrix, int K) {
    if (K < 1 || K > matrix[0].size()) {
        cout << "Неправильный номер столбца!" << endl;
        return;
    }

    for (auto& row : matrix) {
        row.erase(row.begin() + (K - 1)); // удаляем K-й столбец
    }
}
```

```

int main() {
    int row, col, K;
    cout << "Введите количество строк (2-20): ";
    cin >> row;
    cout << "Введите количество столбцов (2-20): ";
    cin >> col;

    if (row < 2 || row > 20 || col < 2 || col > 20) {
        cout << "Ошибка: неверные размеры матрицы!" << endl;
        return 1;
    }

    vector<vector<int>> matrix(row, vector<int>(col));

    cout << "Введите элементы матрицы:" << endl;
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            cin >> matrix[i][j];
        }
    }

    cout << "Начальная матрица:" << endl;
    printMatrix(matrix);

    int firstOddColumn = findFirstOddColumn(matrix);
    cout << "Номер первого столбца с нечетными числами: " << firstOddColumn
    << endl;

    cout << "Введите номер столбца для удаления (1-" << col << "): ";
    cin >> K;

    deleteColumn(matrix, K);

    cout << "Матрица после удаления столбца " << K << ":" << endl;
    printMatrix(matrix);

    return 0;
}

```