

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

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота № 7

з дисципліни «Алгоритмізація та програмування»
на тему "Реалізація алгоритмів обробки двовимірних масивів мовою C ++"

XAI.301.175.318.16ЛР

Виконав студент гр. 318

(підпис, дата) Нікуліна К.О (П.І.Б.)

Перевірів

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

2025

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення двовимірних масивів (матриць) у мові C ++ і реалізувати декларацію, введення з консолі, обробку і 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. Варіанти завдань представлено на рис.1.

Matrix36. Дана цілочисельна матриця розміру $M \times N$, елементи якої можуть набувати значень від 0 до 100. Різні рядки матриці назвемо схожими, якщо збігаються безлічі чисел, що зустрічаються в цих рядках. Знайти кількість рядків, схожих на перший рядок цієї матриці.

Рис.1 Завдання 1

Завдання 2. Перетворити матрицю відповідно до свого варіанту завдання (див. рис.2.), розмір матриці і його елементи ввести з консолі. Вивести результати у консоль.

Matrix56. Дана матриця розміру $M \times N$ (N - парне число). Поміняти місцями ліву і праву половини матриці.

Рис.2 Завдання 2

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Matrix36.

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

M - кількість рядків матриці, дійсний тип, >0 ;

N - кількість стовпців матриці, дійсний тип, >0 ;

$matrix[i][j]$ – елемент матриці в рядку i , стовпці j , дійсний тип, $0 \leq matrix[i][j] \leq 100$.

Вихідні дані (ім'я, опис, тип):

count - кількість рядків схожих на перший, дійсний тип, $0 \leq count \leq M-1$

Алгоритм вирішення:

1. Перевірити, чи M та N більше 0. Якщо ні — вивести помилку.
2. Зчитати матрицю з консолі, перевіряючи, чи кожен елемент $\in [0, 100]$.
3. Якщо матриця порожня, вивести повідомлення.
4. Створити множину firstRowSet з елементів першого рядка.
5. Для кожного рядка i від 1 до $M-1$:
 - Створити множину currentRowSet з елементів поточного рядка.
 - Порівняти currentRowSet з firstRowSet.
 - Якщо множини ідентичні, збільшити count на 1.
6. Вивести count.

Завдання 2.

Вирішення задачі Matrix56.

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

M - кількість рядків матриці, дійсний тип, >0 ;

N - кількість стовпців матриці, дійсний тип, >0 і $/2$;

$matrix[M][N]$ - матриця в рядку M, стовпці N, дійсний тип >0 .

Вихідні дані (ім'я, опис, тип):

matrix - матриця після обміну лівої та правої частин, дійсний тип, розмірність зберігається ($M \times N$)

Алгоритм вирішення:

1. Перевірити, чи N парне. Якщо ні — вивести помилку.
2. Зчитати матрицю.
3. Вивести початкову матрицю.
4. Для кожного рядка i від 0 до $M-1$:
 - Для кожного стовпця j від 0 до $N/2 - 1$:
 - Обміняти місцями $matrix[i][j]$ та $matrix[i][j + N/2]$.
5. Вивести змінену матрицю.

Лістинг коду вирішення задачі Matrix36 і Matrix56 наведено в [дод.А](#) (стор.6). Екран роботи програми показаний на [дод.Б](#) на рис Б1.

Посилання на GitHub: https://github.com/KatyaNikulina/Lb_7.git

ВИСНОВКИ

У ході лабораторної роботи було реалізовано два алгоритми обробки матриць: підрахунок рядків з однаковими множинами елементів та обмін лівої і правої половин матриці. Отримані результати підтвердили ефективність використання множин для порівняння рядків та оптимальність in-place алгоритму для обміну половин. Робота дозволила закріпити навички роботи з двовимірними масивами та STL-контейнерами для вирішення практичних задач обробки даних.

ДОДАТОК А

Лістинг коду програми Matrix36 і Matrix56

```
#include <iostream>
#include <vector>
#include <unordered_set>
#include <limits>

using namespace std;

// 1. Функция для подсчета похожих строк
void countSimilarRowsTask()
{
    cout << "\n=== Задание 1: Подсчет строк, похожих на первую ===\n";

    int M_1, N_1;
    cout << "Введите количество строк (M) и столбцов (N) матрицы: ";
    cin >> M_1 >> N_1;

    if (M_1 <= 0 || N_1 <= 0)
    {
        cerr << "Ошибка: M и N должны быть положительными числами!\n";
        return;
    }

    vector<vector<int>> matrix(M_1, vector<int>(N_1));

    cout << "Введите элементы матрицы (числа от 0 до 100):\n";
    for (int i = 0; i < M_1; ++i)
    {
        for (int j = 0; j < N_1; ++j)
        {
            while (!(cin >> matrix[i][j]))
            {
                cerr << "Ошибка ввода! Введите число от 0 до 100: ";
                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');
            }
            if (matrix[i][j] < 0 || matrix[i][j] > 100)
            {
                cerr << "Число должно быть от 0 до 100! Повторите ввод: ";
                j--;
            }
        }
    }
}
```

```

    }
}

if (matrix.empty())
{
    cout << "Матрица пуста!\n";
    return;
}

// Создаем множество элементов первой строки
unordered_set<int> firstRowSet(matrix[0].begin(), matrix[0].end());
int count = 0;

// Сравниваем начиная со второй строки (индекс 1)
for (size_t i = 1; i < matrix.size(); ++i)
{
    unordered_set<int> currentRowSet(matrix[i].begin(), matrix[i].end());
    if (currentRowSet == firstRowSet)
    {
        count++;
    }
}

cout << "Количество строк, похожих на первую: " << count << "\n";
}

// 2. Функция для обмена половин
void swapMatrixHalvesTask()
{
    cout << "\n=== Задание 2: Обмен левой и правой половин матрицы ===\n";

    int M_2, N_2;
    cout << "Введите количество строк (M) и столбцов (N, должно быть четным): ";
    cin >> M_2 >> N_2;

    if (M_2 <= 0 || N_2 <= 0)
    {
        cerr << "Ошибка: M и N должны быть положительными числами!\n";
        return;
    }

    if (N_2 % 2 != 0)
    {
        cerr << "Ошибка: N должно быть четным числом!\n";
        return;
    }
}

```

```

    }

    vector<vector<int>> matrix(M_2, vector<int>(N_2));

    cout << "Введите элементы матрицы:\n";
    for (int i = 0; i < M_2; ++i)
    {
        for (int j = 0; j < N_2; ++j)
        {
            cin >> matrix[i][j];
        }
    }

    cout << "\nИсходная матрица:\n";
    for (const auto& row : matrix)
    {
        for (int num : row) cout << num << " ";
        cout << "\n";
    }

    int half = N_2 / 2;
    for (int i = 0; i < M_2; ++i)
    {
        for (int j = 0; j < half; ++j)
        {
            swap(matrix[i][j], matrix[i][j + half]);
        }
    }

    cout << "\nМатрица после обмена половин:\n";
    for (const auto& row : matrix)
    {
        for (int num : row) cout << num << " ";
        cout << "\n";
    }
}

int main()
{
    setlocale(LC_ALL, "rus");
    int choice;

    do
    {
        cout << "\nВыберите задание:\n"
              << "1. Подсчет строк, похожих на первую\n"

```



```

        << "2. Обмен левой и правой половин матрицы\n"
        << "0. Выход\n"
        << "Ваш выбор: ";
    cin >> choice;

    switch (choice)
    {
    case 1:
        countSimilarRowsTask();
        break;
    case 2:
        swapMatrixHalvesTask();
        break;
    case 0:
        cout << "Выход из программы.\n";
        break;
    default:
        cout << "Неверный выбор! Попробуйте еще раз.\n";
    }
} while (choice != 0);

return 0;
}

```

ДОДАТОК Б

Скріншоти вікна виконання програми

```
Выберите задание:
1. Подсчет строк, похожих на первую
2. Обмен левой и правой половин матрицы
0. Выход
Ваш выбор: 1

=== Задание 1: Подсчет строк, похожих на первую ===
Введите количество строк (М) и столбцов (N) матрицы: 5 4
Введите элементы матрицы (числа от 0 до 100):
1 2 3 4
5 6 7 8
1 2 3 4
9 10 11 12
1 2 3 4
Количество строк, похожих на первую: 2

Выберите задание:
1. Подсчет строк, похожих на первую
2. Обмен левой и правой половин матрицы
0. Выход
Ваш выбор: 1

=== Задание 1: Подсчет строк, похожих на первую ===
Введите количество строк (М) и столбцов (N) матрицы: 5 4
Введите элементы матрицы (числа от 0 до 100):
1 2 3 4
5 6 7 8
4 3 2 1
9 10 11 12
1 2 3 4
Количество строк, похожих на первую: 2

Выберите задание:
1. Подсчет строк, похожих на первую
2. Обмен левой и правой половин матрицы
0. Выход
Ваш выбор: 2

=== Задание 2: Обмен левой и правой половин матрицы ===
Введите количество строк (М) и столбцов (N, должно быть четным): 5 5
Ошибка: N должно быть четным числом!

Выберите задание:
1. Подсчет строк, похожих на первую
2. Обмен левой и правой половин матрицы
0. Выход
Ваш выбор: 2

=== Задание 2: Обмен левой и правой половин матрицы ===
Введите количество строк (М) и столбцов (N, должно быть четным): 5 4
Введите элементы матрицы:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20

Исходная матрица:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20

Матрица после обмена половин:
3 4 1 2
7 8 5 6
11 12 9 10
15 16 13 14
19 20 17 18

Выберите задание:
1. Подсчет строк, похожих на первую
2. Обмен левой и правой половин матрицы
0. Выход
Ваш выбор: 0
Выход из программы.
```

Рисунок Б.1 – Экран виконання програми для вирішення завдань
Matrix36 та Matrix56