

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»

Тема: "Реалізація алгоритмів обробки двовимірних масивів мовою C ++"

XAI.301.141.319a. 27 ЛР

Виконав студент гр. 319a

Владислав РУДНЄВ

(Підпис, дата)

(П.І.Б.)

Перевірів к.т.н., доц.

(вчена ступінь, вчене звання)

Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

2024

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.

Matrix20. Дана матриця розміру $M \times N$. Для кожного стовпця матриці знайти добуток його елементів.

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

Matrix71. Дана матриця розміру $M \times N$. Продублювати стовпець матриці, що містить її мінімальний елемент.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Matrix20.

Вхідні дані:

- Масив чисел (arr): Одновимірний масив цілих чисел.
- Розмір масиву (n): Кількість елементів у масиві ($n > 0$).

Вихідні дані:

- Початковий масив (arr): Масив, який користувач ввів.
- Масив у зворотному порядку (reversedArr): Масив після виведення у зворотному порядку.

Алгоритм:

1. Введення даних: Користувач вводить розмір масиву n (якщо $n \leq 0$ – генерується помилка) та n елементів масиву.
2. Обробка даних: Виводиться початковий масив, а потім за допомогою циклу `for` створюється масив, який містить ті самі елементи, але у зворотному порядку.
3. Виведення результату: Виводиться спочатку початковий масив, а потім його варіант із зворотним порядком.

Завдання 2.

Вирішення задачі Matrix72.

Вхідні дані:

- Масив чисел (`arr`): Одновимірний масив цілих чисел.
- Розмір масиву (`n`): Кількість елементів у масиві ($n > 0$).
- Індеси для перестановки підмасиву (`k`, `l`): Початковий (`k`) і кінцевий (`l`) індеси підмасиву, за умовою $1 \leq k < l \leq n$.

Вихідні дані:

- Початковий масив (`arr`): Масив, який користувач ввів.
- Модифікований масив (`modifiedArr`): Масив після перестановки підмасиву (елементи між `k` та `l` стають у зворотному порядку).
- Повідомлення про помилки (`errorMsg`): Якщо користувач вводить некоректні значення для `k` і `l` (наприклад, $k < 1$, $l > n$ або $k \geq l$), виводиться текстове повідомлення про помилку.

Алгоритм:

1. Введення даних: Користувач задає розмір масиву n (якщо $n \leq 0$ – генерується помилка), вводить n елементів, а також значення `k` і `l`. Якщо значення `k` та `l` не задовольняють умову ($1 \leq k < l \leq n$), виводиться повідомлення про помилку.
2. Перестановка підмасиву: Викликається функція `reverseSubarray()`, яка змінює порядок елементів між індексами `k` та `l` (зауважте, що

користувацькі індекси можуть бути введені як 1-базовані, тому може знадобитися перетворення для роботи з вектором).

3. Виведення результату: Виводиться початковий масив, після чого – модифікований масив з переставленим підмасивом.

Лістинг коду дод. А (стр.5 — 7)

Скрін-шоти вікна виконання програми дод. Б (сторінка 8 — 10)

ВИСНОВОК

У ході роботи було вивчено основи представлення двовимірних масивів у мові C++ та реалізовано їх декларацію, введення, обробку й виведення в консоль. Отримані знання були закріплені шляхом розв’язання практичних задач у середовищі Visual Studio.

ДОДАТОК А

Лістинінг коду

```

#include <iostream>
#include <vector>

using namespace std;

// Константи для мінімального та максимального розміру матриці
const int MIN_SIZE = 2, MAX_SIZE = 20;

// Функція для введення розміру матриці (кількість рядків та стовпців)
void inputMatrixSize(int& M, int& N) {
    do {
        cout << "Enter the number of rows (from 1 to 20): ";
        cin >> M;

        cout << "Enter the number of columns (from 1 to 20): ";
        cin >> N;

        // Перевірка, чи введений розмір знаходиться в допустимих межах
        if (M < MIN_SIZE || M > MAX_SIZE || N < MIN_SIZE || N > MAX_SIZE) {
            cout << "Error! Matrix size must be between 1 and 20. Try
again.\n";
        }
    } while (M < MIN_SIZE || M > MAX_SIZE || N < MIN_SIZE || N > MAX_SIZE);
}

// Функція для введення елементів матриці
void inputMatrix(vector<vector<int>>& matrix, int M, int N) {
    cout << "Enter the elements of the matrix by rows:" << endl;
    for (int i = 0; i < M; i++) {
        for (int j = 0; j < N; j++) {
            cin >> matrix[i][j];
        }
    }
}

// Функція для виведення елементів матриці
void printMatrix(const vector<vector<int>>& matrix) {
    for (const auto& row : matrix) {
        for (int elem : row) {
            cout << elem << " ";
        }
        cout << endl;
    }
}

```

```

    }
}

// Завдання 1: Обчислення добутку елементів кожного стовпця
void task1() {
    int M, N;
    inputMatrixSize(M, N); // Введення розмірів матриці

    // Створення матриці заданого розміру
    vector<vector<int>> matrix(M, vector<int>(N));
    inputMatrix(matrix, M, N); // Введення елементів

    // Ініціалізація вектора для збереження добутку по кожному стовпцю
    vector<int> column_products(N, 1);

    // Обчислення добутку по кожному стовпцю
    for (int j = 0; j < N; j++) {
        for (int i = 0; i < M; i++) {
            column_products[j] *= matrix[i][j];
        }
    }

    // Виведення результатів
    cout << "Product of the elements of each column:" << endl;
    for (int prod : column_products) {
        cout << prod << " ";
    }
    cout << endl;
}

// Завдання 2: Дублювання стовпця, що містить мінімальний елемент (в кінець)
void task2() {
    int M, N;
    inputMatrixSize(M, N); // Введення розмірів

    // Створення матриці заданого розміру
    vector<vector<int>> matrix(M, vector<int>(N));
    inputMatrix(matrix, M, N); // Введення елементів

    // Початкове значення мінімального елемента – припустимо перший елемент
    int minElement = matrix[0][0];
    int minColumn = 0;

    // Пошук мінімального елемента та індексу його стовпця
    for (int j = 0; j < N; j++) {
        for (int i = 0; i < M; i++) {
            if (matrix[i][j] < minElement) {
                minElement = matrix[i][j];
                minColumn = j;
            }
        }
    }
}

```

```

        }
    }
}

// Додавання дубльованого стовпця у кінець
for (int i = 0; i < M; i++) {
    matrix[i].push_back(matrix[i][minColumn]); // Додаємо значення з
мінімального стовпця
}
// Виведення оновленої матриці
cout << "Matrix after duplicating the column with the minimum element to
the end:" << endl;
printMatrix(matrix);
}
// Меню для вибору завдання
void menu() {
    while (true) {
        int choice;
        cout << "Menu:\n";
        cout << "1. Task 1: Calculate the product of the elements of each
column\n";
        cout << "2. Task 2: Duplicate the column with the minimum element\n";
        cout << "3. Exit\n";
        cout << "Choose an option: ";
        cin >> choice;

        // Обробка вибору
        if (choice == 3) break;

        switch (choice) {
            case 1:
                task1(); // Завдання 1
                break;
            case 2:
                task2(); // Завдання 2
                break;
            default:
                cout << "Invalid choice, please try again." << endl;
        }
    }
}

// Головна функція
int main() {
    menu(); // Запуск меню
    return 0;
}

```

ДОДАТОК Б

Скрін-шоти вікна виконання програми та діаграми

На рис.1 показано код виконання програми Лабораторна робота №7

```
1. Task 1: Calculate the product of the elements of each column
2. Task 2: Duplicate the column with the minimum element
3. Exit
Choose an option: 1
Enter the number of rows (from 2 to 20): 1
Enter the number of columns (from 2 to 20): 1
Error! Matrix size must be between 2 and 20. Try again.
Enter the number of rows (from 2 to 20): 2
Enter the number of columns (from 2 to 20): 2
Enter the elements of the matrix by rows:
2 3
4 1
Product of the elements of each column:
8 3
Menu:
1. Task 1: Calculate the product of the elements of each column
2. Task 2: Duplicate the column with the minimum element
3. Exit
Choose an option: 2
Enter the number of rows (from 2 to 20): 2
Enter the number of columns (from 2 to 20): 2
Enter the elements of the matrix by rows:
4 2
5 2
Matrix after duplicating the column with the minimum element to the end:
4 2 2
5 2 2
Menu:
1. Task 1: Calculate the product of the elements of each column
2. Task 2: Duplicate the column with the minimum element
3. Exit
Choose an option: 3
```

Рисунок 1 — Лабораторна робота №7

На рис.2 показано діаграму коду програми для завдання 1

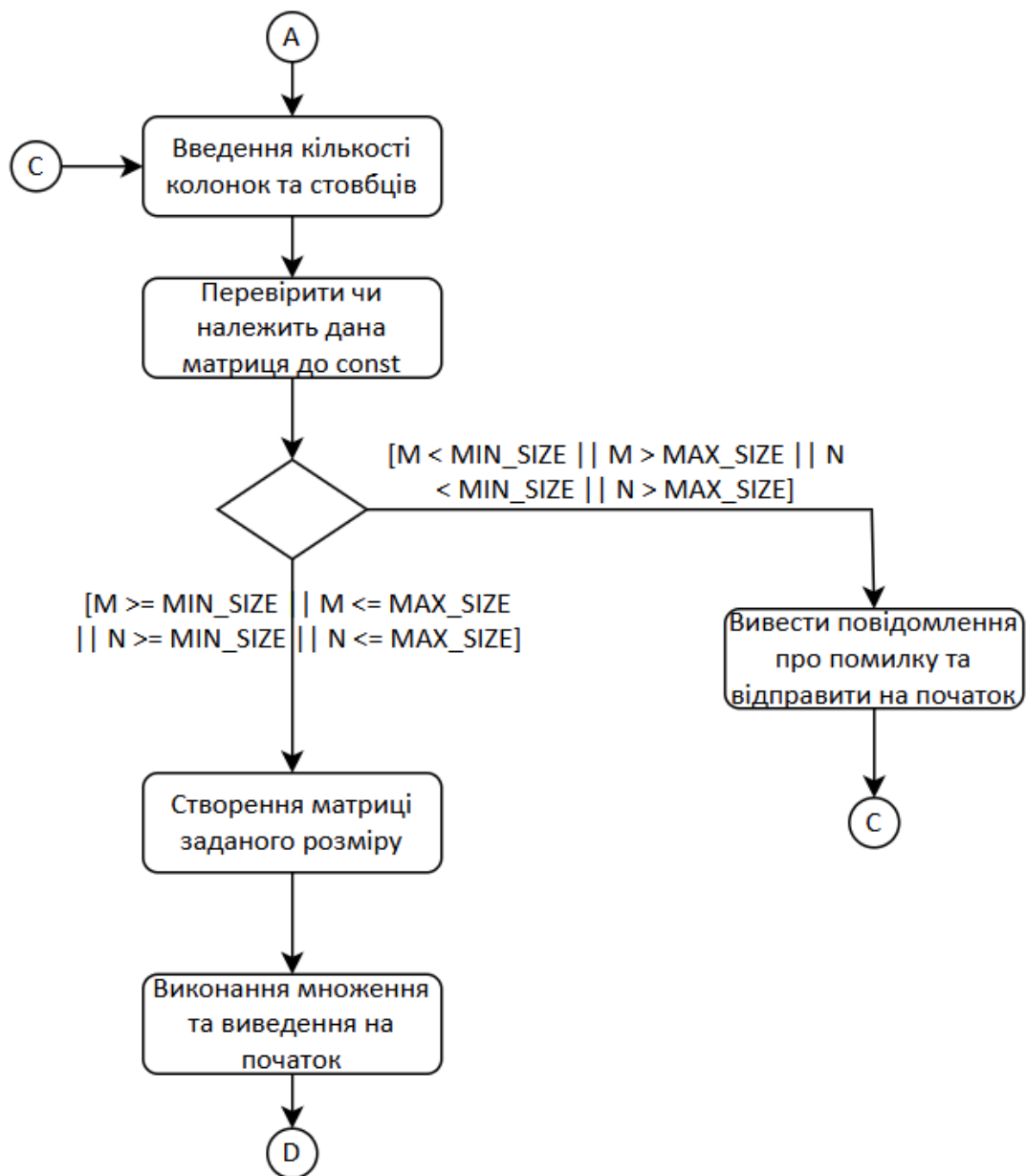


Рисунок 2 — Діаграма активності коду Matrix20.

На рис.3 показано діаграму коду програми для завдання 3

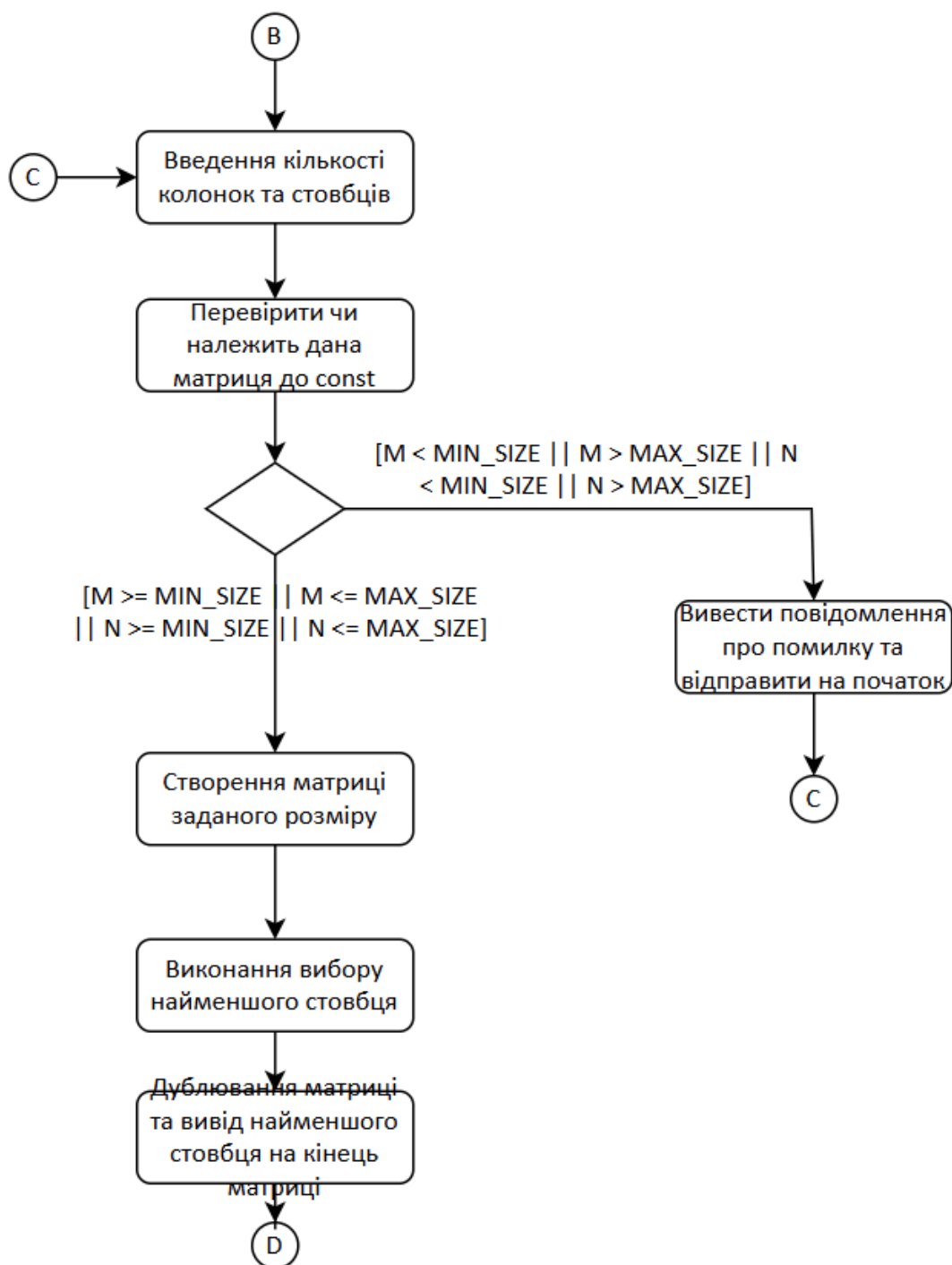


Рисунок 3 — Діаграма активності коду для Matrix71.