

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 ++»

ХАІ.301.173.310.1 ЛР

Виконав студент гр. \_\_\_\_\_ 310 \_\_\_\_\_

\_\_\_\_\_ *Микола АНДРЮШКІН* \_\_\_\_\_  
(підпис, дата) (П.І.Б.)

Перевірів

\_\_\_\_\_ 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 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. Дана цілочисельна матриця розміру  $M \times N$ . Знайти елемент, який є максимальним у своєму рядку і мінімальним в своєму стовпці. Якщо таки елемент відсутній, то вивести 0. Matrix46.

Завдання 2. Перетворити матрицю відповідно до свого варіанту завдання, розмір матриці і його елементи ввести з консолі. Вивести результати у консоль. Дана квадратна матриця порядку  $M$ . Обнулити елементи матриці, що лежать одночасно нижче головної діагоналі (включаючи цю діагональ) і нижче побічної діагоналі (також включаючи цю діагональ). Умовний оператор не використовувати. Matrix83.

## ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Matrix 46.

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

$M$  — ширина матриці, int, від 1 до 20.

$N$  — ширина матриці, int, від 1 до 20.

matrix — матриця, int, від -2147483647 до 2147483647.

Вихідні дані (ім'я, опис, тип):

matrix — матриця, int, від -2147483647 до 2147483647.

Алгоритм вирішення показано на рис. 1, inputMatrix() на рис. 2, printMatrix() на рис. 3, findAndPrintResults() на рис. 4.

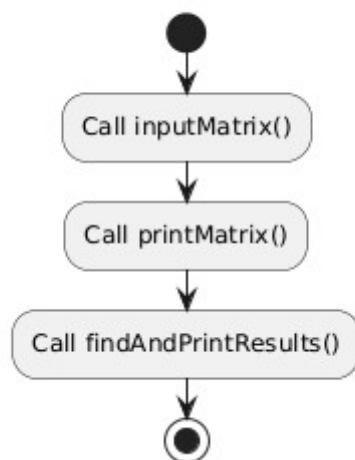
**matrix46() Execution Path**

Рисунок 1 – Matrix46

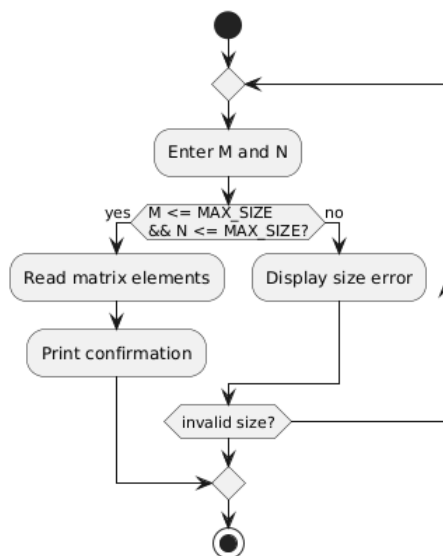
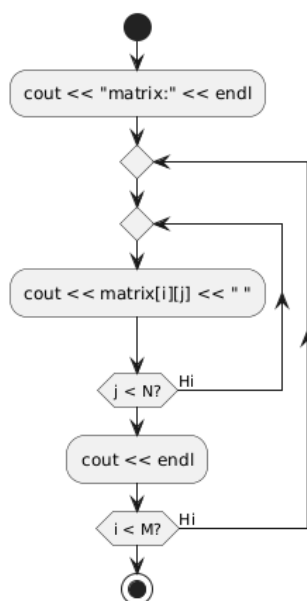
**inputMatrix() Validation Loop**

Рисунок 2 – inputMatrix()



#### findAndPrintResults() Analysis

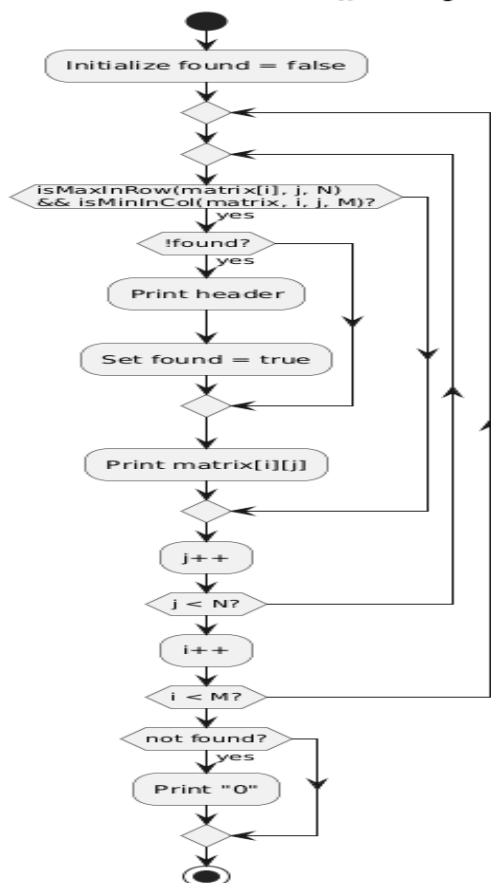


Рисунок 4 – findAndPrintResults()

Лістинг коду вирішення задачі matrix 46 наведено в дод. А (стор. 8).  
Екран роботи програми показаний на рис. Б.1.

Завдання 1.

Вирішення задачі mmatrix 83.

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

M — ширина матриці, int, від 1 до 20.

N — ширина матриці, int, від 1 до 20.

matrix — матриця, int, від -2147483647 до 2147483647.

Вихідні дані (ім'я, опис, тип):

matrix — матриця, int, від -2147483647 до 2147483647.

Алгоритм вирішення показано на рис. 5, zeroingMatrix() на рис. 6.

**matrix83() Execution Path**

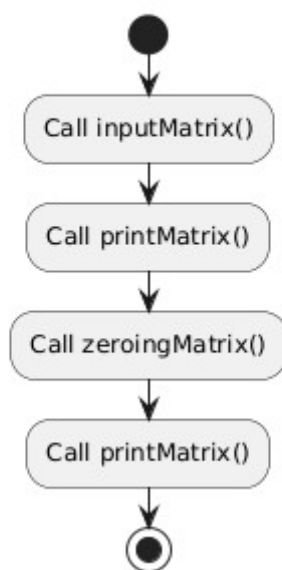
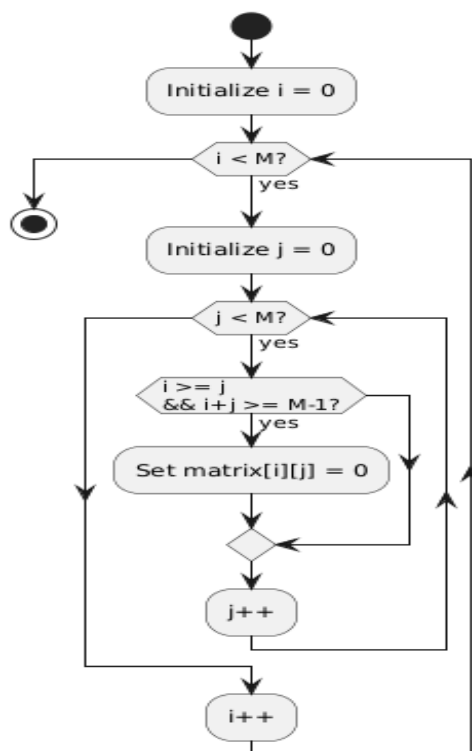


Рисунок 5 – matrix83()

**zeroingMatrix() Element Processing**Рисунок 6 – `zeroingMatrix()`

Лістинг коду вирішення задачі розділ і номер задач(i) наведено в дод. А (стор. 8).

Екран роботи програми показаний на рис. Б.2.

## ВИСНОВКИ

Були вивчені алгоритми обробки двовимірних масивів. Закріплено на практиці методи обробки масивів. Отримано навички обробки двовимірних масивів. Виникли труднощі з виконанням Matrix83 однак труднощі були поборені.

## ДОДАТОК А

### Лістинг коду програми

```

#include <iostream>
using namespace std;

const int MAX_SIZE = 20;

int matrix46();
int matrix83();

int main() {
    int lp = 1;
    while(lp == 1){
        cout << "Selection menu: 1 - Matrix_46; 2 - Matrix_83; 3 - Exit;" << endl <<
        "You select - "; //Message for user
        int crs;
        cin >> crs;
        switch (crs) { //Menu
            case 1:
                matrix46();
                break;
            case 2:
                matrix83();
                break;
            case 3:
                lp = 0;
                break;
            default:
                cout << "Wrong input" << endl;
                break;
        }
    }
}

// Функція для введення матриці
void inputMatrix(int matrix[][MAX_SIZE], int &M, int &N) {
    bool chk = true;

    while(chk){

        cout << "Enter matrix size(row col): ";
        cin >> M >> N;

        if (M <= MAX_SIZE || N <= MAX_SIZE) {
            cout << "Enter matrix elements: ";

```



```

        for (int i = 0; i < M; ++i) {
            for (int j = 0; j < N; ++j) {
                cin >> matrix[i][j];
            }
        }
        cout << "You entered this";
        chk = false;
    }
    else {
        cout << "Error: matrix size is too big" << endl;
    }
}

// Перевірка, чи є елемент максимальним у своєму рядку
bool isMaxInRow(int row[], int currentCol, int N) {
    for (int j = 0; j < N; ++j) {
        if (row[j] > row[currentCol]) {
            return false;
        }
    }
    return true;
}

// Перевірка, чи є елемент мінімальним у своєму стовпці
bool isMinInCol(int matrix[][MAX_SIZE], int currentRow, int currentCol, int M) {
    for (int i = 0; i < M; ++i) {
        if (matrix[i][currentCol] < matrix[currentRow][currentCol]) {
            return false;
        }
    }
    return true;
}

void printMatrix(int matrix[][MAX_SIZE], int M, int N) {
    cout << " matrix: " << endl;
    for (int i = 0; i < M; ++i) {
        for (int j = 0; j < N; ++j) {
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }
}

// Пошук та виведення результатів
void findAndPrintResults(int matrix[][MAX_SIZE], int M, int N) {
    bool found = false;
    for (int i = 0; i < M; ++i) {
        for (int j = 0; j < N; ++j) {
            if (isMaxInRow(matrix[i], j, N) && isMinInCol(matrix, i, j, M)) {

```

```

        if (!found) {
            cout << "Found elements: ";
            found = true;
        }
        cout << matrix[i][j] << " ";
    }
}

if (!found) {
    cout << "0";
}
cout << endl;
}

void zeroingMatrix(int matrix[][MAX_SIZE], int M) {
    for (int i = 0; i < M; ++i) {
        for (int j = 0; j < M; ++j) {
            int condition = (i >= j) && (i + j >= M - 1);
            matrix[i][j] *= !condition;
        }
    }
    cout << "Zeroed ";
}

```

```

int matrix46() {
    int M, N;
    int matrix[MAX_SIZE][MAX_SIZE];

    inputMatrix(matrix, M, N);
    printMatrix(matrix, M, N);
    findAndPrintResults(matrix, M, N);

    return 0;
}

```

```

int matrix83() {
    int M, N;
    int matrix[MAX_SIZE][MAX_SIZE];

    inputMatrix(matrix, M, N);
    printMatrix(matrix, M, N);
    zeroingMatrix(matrix, M);
    printMatrix(matrix, M, N);

    return 0;
}

```

ДОДАТОК Б  
Скрін-шоти вікна виконання програми

```
Selection menu: 1 - Matrix_46; 2 - Matrix_83; 3 - Exit;
You select - 1
Enter matrix size(row col): 150 150
Error: matrix size is too big
Enter matrix size(row col): 3 3
Enter matrix elements: 1 2 3 4 5 6 7 8 9
You entered this matrix:
1 2 3
4 5 6
7 8 9
Found elements: 3
Selection menu: 1 - Matrix_46; 2 - Matrix_83; 3 - Exit;
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання  
Matrix46

```
Selection menu: 1 - Matrix_46; 2 - Matrix_83; 3 - Exit;
You select - 2
Enter matrix size(row col): 150 150
Error: matrix size is too big
Enter matrix size(row col): 3 3
Enter matrix elements: 1 2 3 4 5 6 7 8 9
You entered this matrix:
1 2 3
4 5 6
7 8 9
Zeroed matrix:
1 2 3
4 0 6
0 0 0
Selection menu: 1 - Matrix_46; 2 - Matrix_83; 3 - Exit;
You select - █
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання  
Matrix83