

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. 310 група, 7 номер в списку ЛР

Виконала студентка гр. 310

Стеценко

СофіяОлександрівна

(підпис, дата)

(П.І.Б.)

Перевірів к.т.н., доц. Олена

ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

Харків 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.

Розробити програму для аналізу та виведення елементів цілочисельної матриці розміру  $M \times N$ . Забезпечити введення та виведення даних через командний рядок. Визначити номер останнього рядка, що містить максимальну кількість однакових елементів.

### Завдання 2.

Розробити програму для перетворення матриці розміру  $M \times N$  відповідно до заданих умов. Ввести розмір матриці та її елементи з консолі, а результати вивести у консоль. Перед першим стовпцем, що містить лише додатні елементи, вставити стовпець із одиниць. Якщо такого стовпця немає, вивести матрицю без змін.

## ВИКОНАННЯ РОБОТИ

### Завдання 1.

Вирішення задачі Matrix40.

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

Кількість рядків та стовпців матриці (в межах від 2 до 20), елементи матриці (цілі числа).

Вихідні дані (ім'я, опис, тип):

Номер останнього рядка з максимальною кількістю однакових елементів.

Алгоритм вирішення представлено в дод. Г (стор. 12).

Лістинг коду вирішення задачі наведено в дод. Б (стор. 9).

Екран роботи програми показаний на Рис. 1.

```
97     int choice;
98
99     do {
100         cout << "Виберіть завдання (1 - Matrix40, 2 - Matrix72, 0 - Вихід)";
101     } while (choice != 0);
```

input

Виберіть завдання (1 - Matrix40, 2 - Matrix72, 0 - Вихід): 1  
Введіть кількість рядків (2-20): 3  
Введіть кількість стовпців (2-20): 5  
Введіть елементи:  
12 -6 78 90 5  
4 71 33 4 903  
-781 34 5 21 7

Матриця:

12	-6	78	90	5
4	71	33	4	903
-781	34	5	21	7

Останній рядок з максимальною кількістю однакових елементів: 2  
Виберіть завдання (1 - Matrix40, 2 - Matrix72, 0 - Вихід):

Рис. 1 – Завдання 1

## Завдання 2.

## Вирішення задачі Matrix72.

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

Кількість рядків та стовпців матриці (в межах від 2 до 20), елементи матриці (цілі числа).

Вихідні дані (ім'я, опис, тип):

Змінена матриця після вставки стовпця з одиниць.

Алгоритм вирішення представлено в дод. Д (стор. 13).

Лістинг коду вирішення задачі наведено в дод. В (стор. 11).

Екран роботи програми показаний на Рис. 2.

```

101
input
Виберіть завдання (1 - Matrix40, 2 - Matrix72, 0 - Вихід): 2
Введіть кількість рядків (2-20): 5
Введіть кількість стовпців (2-20): 4
Введіть елементи:
31 567 980 2
762 -8 112 45
49 7 203 5
801 23 56 -7
1 32 76 803

Матриця:
31      567      980      2
762     -8      112     45
49       7      203      5
801      23      56     -7
1        32      76     803
Матриця після вставки:
Матриця:
1      31      567      980      2
1      762     -8      112     45
1      49       7      203      5
1      801      23      56     -7
1       1       32      76     803
Виберіть завдання (1 - Matrix40, 2 - Matrix72, 0 - Вихід):

```

Рис. 2 - Завдання 2

## ВИСНОВКИ

У ході роботи було вивчено основи представлення двовимірних масивів у С++ та реалізовано їх декларацію, введення, обробку і виведення в консоль. Програму розроблено та протестовано в середовищі Visual Studio та онлайн компіляторі. Для допомоги в пошуку потрібних бібліотек та перевірки функцій було використано штучний інтелект "Blackbox AI".

Отримані навички можуть виявитися корисними для роботи з табличними даними.

## ДОДАТОК А

## Лістинг коду програми

```

#include using namespace std;

const int M = 20, N = 20; // Максимальні розміри матриці

// Функція введення матриці void get_matr(int in_matr[M][N], int &in_m, int
&in_n) { do { cout << "Введіть кількість рядків (2-20): "; cin >> in_m; cout
<< "Введіть кількість стовпців (2-20): "; cin >> in_n; } while (in_n < 2 || in_n
> N || in_m < 2 || in_m > M);

cout << "Введіть елементи: " << endl;
for (int i = 0; i < in_m; i++)
    for (int j = 0; j < in_n; j++)
        cin >> in_matr[i][j];

}

// Завдання 1: Matrix40 // Функція для знаходження номера останнього
рядка з максимальною кількістю однакових елементів int
last_row_with_max_equal(int matr[M][N], int m, int n) { int max_count = 0; int
last_row = -1; for (int i = 0; i < m; i++) {
    int count[101] = {0}; // Припустимо, що елементи матриці в межах 0-100
    for (int j = 0; j < n; j++) {
        count[matr[i][j]]++;
    }

    int current_max = 0;
    for (int k = 0; k < 101; k++) {
        if (count[k] > current_max) {
            current_max = count[k];
        }
    }

    if (current_max > 1) {
        if (current_max >= max_count) {
            max_count = current_max;
            last_row = i + 1; // Номери рядків з 1
        }
    }
}

```

```

    }
}
return last_row;

}

// Функція виведення матриці void show_matr(const int out_matr[M][N],
const int m, const int n) { cout << endl << "Матриця: " << endl; for (int i = 0; i
< m; i++) { for (int j = 0; j < n; j++) cout << out_matr[i][j] << "\t"; cout <<
endl; } }

// Завдання 2: Matrix72 // Функція для вставки стовпця з одиниць перед
першим стовпцем з додатніми елементами void
insert_column_before_positive(int matr[M][N], int &m, int &n) { int insert_col
= -1;

// Знаходимо перший стовпець з додатніми елементами
for (int j = 0; j < n; j++) {
    bool has_positive = true;
    for (int i = 0; i < m; i++) {
        if (matr[i][j] <= 0) {
            has_positive = false;
            break;
        }
    }
    if (has_positive) {
        insert_col = j;
        break;
    }
}

// Якщо знайдено стовпець, вставляємо новий стовпець
if (insert_col != -1) {
    for (int i = 0; i < m; i++) {
        for (int j = n; j > insert_col; j--) {
            matr[i][j] = matr[i][j - 1];
        }
        matr[i][insert_col] = 1; // Вставляємо одиницю
    }
    n++; // Збільшуємо кількість стовпців
}

```

```
}
```

```
}
```

```
// Головна функція int main() { int matr[M][N]; // Оголошення матриці int
row, col; // Реальні розміри матриці int choice;
```

```
do {
```

```
    cout << "Виберіть завдання (1 - Matrix40, 2 - Matrix72, 0 - Вихід): ";
    cin >> choice;
```

```
    switch (choice) {
```

```
        case 1: {
```

```
            get_matr(matr, row, col); // Введення матриці
```

```
            show_matr(matr, row, col); // Виведення матриці
```

```
            int lastRow = last_row_with_max_equal(matr, row, col); // Зберігаємо
результат
```

```
            if (lastRow != -1) {
```

```
                cout << "Останній рядок з максимальною кількістю однакових
елементів: " << lastRow << endl;
```

```
            } else {
```

```
                cout << "Рядки не знайдено." << endl;
```

```
            }
```

```
            break;
```

```
        }
```

```
        case 2:
```

```
            get_matr(matr, row, col); // Введення матриці
```

```
            show_matr(matr, row, col); // Виведення матриці
```

```
            insert_column_before_positive(matr, row, col); // Вставка стовпця
```

```
            cout << "Матриця після вставки: " << endl;
```

```
            show_matr(matr, row, col); // Виведення зміненої матриці
```

```
            break;
```

```
        case 0:
```

```
            cout << "Вихід..." << endl;
```

```
            break;
```

```
        default:
```

```
            cout << "Невірний вибір. Спробуйте ще раз." << endl;
```

```
    }
```

```
} while (choice != 0);
```

```
return 0;
```

```
}
```

## ДОДАТОК Б

### Лістинг коду вирішення задачі 1

```
// Завдання 1: Matrix40 // Функція для знаходження номера останнього
// рядка з максимальною кількістю однакових елементів int
last_row_with_max_equal(int matr[M][N], int m, int n) { int max_count = 0; int
last_row = -1;

for (int i = 0; i < m; i++) {
    int count[101] = {0}; // Припустимо, що елементи матриці в межах 0-100
    for (int j = 0; j < n; j++) {
        count[matr[i][j]]++;
    }

    int current_max = 0;
    for (int k = 0; k < 101; k++) {
        if (count[k] > current_max) {
            current_max = count[k];
```



```

    }
}

// Якщо поточна максимальна кількість однакових елементів більша за
1
if (current_max > 1) {
    if (current_max >= max_count) {
        max_count = current_max;
        last_row = i + 1; // Номери рядків з 1
    }
}
}
return last_row;

}

// Функція виведення матриці void show_matr(const int out_matr[M][N],
const int m, const int n) { cout << endl << "Матриця: " << endl; for (int i = 0; i
< m; i++) { for (int j = 0; j < n; j++) cout << out_matr[i][j] << "\t"; cout <<
endl; } }

```

## ДОДАТОК В

### Лістинг коду вирішення задачі 2

```
// Завдання 2: Matrix72 // Функція для вставки стовпця з одиниць перед
// першим стовпцем з додатніми елементами void
insert_column_before_positive(int matr[M][N], int &m, int &n) { int insert_col
= -1;

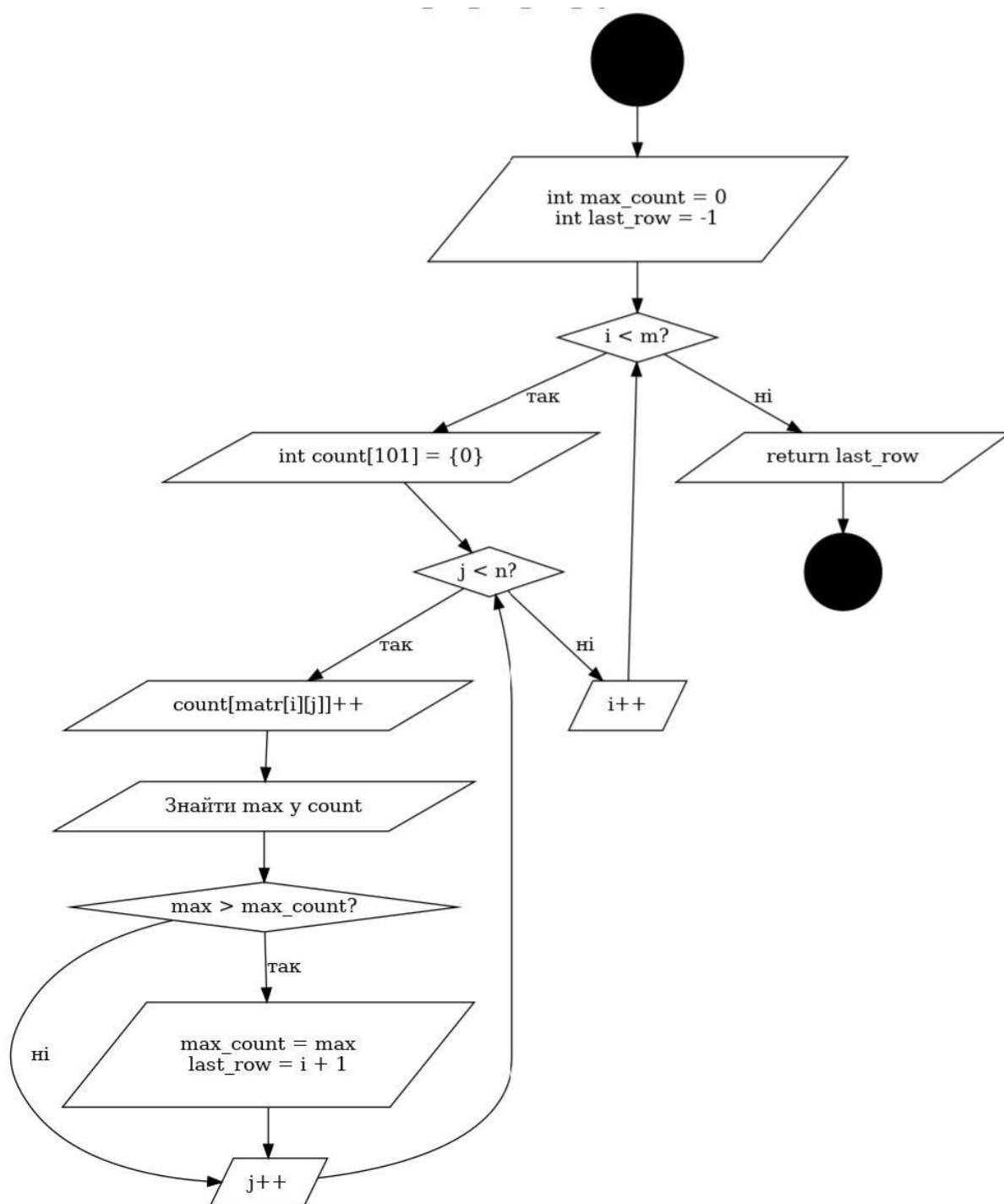
// Знаходимо перший стовпець з додатніми елементами
for (int j = 0; j < n; j++) {
    bool has_positive = true;
    for (int i = 0; i < m; i++) {
        if (matr[i][j] <= 0) {
            has_positive = false;
            break;
        }
    }
    if (has_positive) {
        insert_col = j;
        break;
    }
}

// Якщо знайдено стовпець, вставляємо новий стовпець
if (insert_col != -1) {
    for (int i = 0; i < m; i++) {
        for (int j = n; j > insert_col; j--) {
            matr[i][j] = matr[i][j - 1];
        }
        matr[i][insert_col] = 1; // Вставляємо одиницю
    }
    n++; // Збільшуємо кількість стовпців
}
```

}

## ДОДАТОК Г

## Алгоритм вирішення завдання 1



## ДОДАТОК Д

## Алгоритм вирішення завдання 2

