

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.301.312.2

Виконав студент гр. _____312_____

_____Ксенія ВЕЛІКОДАНОВА_____

(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 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 в табл.1. Завдання 2. Перетворити матрицю відповідно до свого варіанту завдання (див. табл.2.), розмір матриці і його елементи ввести з консолі. Вивести результати у консоль.

Matrix 35, табл.1. Дана цілочисельна матриця розміру $M \times N$. Знайти номер першого з її стовпців, що містять тільки непарні числа. Якщо таких стовпців немає, то вивести 0.

Matrix 60, табл. 2. Дана матриця розміру $M \times N$. Дзеркально відобразити її елементи відносно вертикальної осі симетрії матриці (при цьому поміняються місцями стовпчики з номерами 1 і N , 2 і $N - 1$ і т. Д.).

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Matrix 35

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

- 1) MAX, Максимально допустима кількість рядків і стовпців матриці,
const int, 20
- 2) m, Кількість рядків у матриці, int, 2 ... 20
- 3) n, Кількість стовпців у матриці, int, 2 ... 20
- 4) A, Вхідна двовимірна ціла матриця розміром $m \times n$, int
A[MAX][MAX]кожен елемент — будь-яке ціле число

Вихідні дані (ім'я, опис, тип):

- 1) A (відображення) Повний вміст введеної матриці, надрукований таблично
- 2) colNum Номер першого стовпця, усі елементи якого непарні (нумерація з 1). Якщо таких стовпців немає, значення 0

3) «Немає стовпців, що складаються лише з непарних чисел.» Текстове сповіщення, що виводиться замість colNum, коли результат = 0

Алгоритм вирішення показано на рис. 1

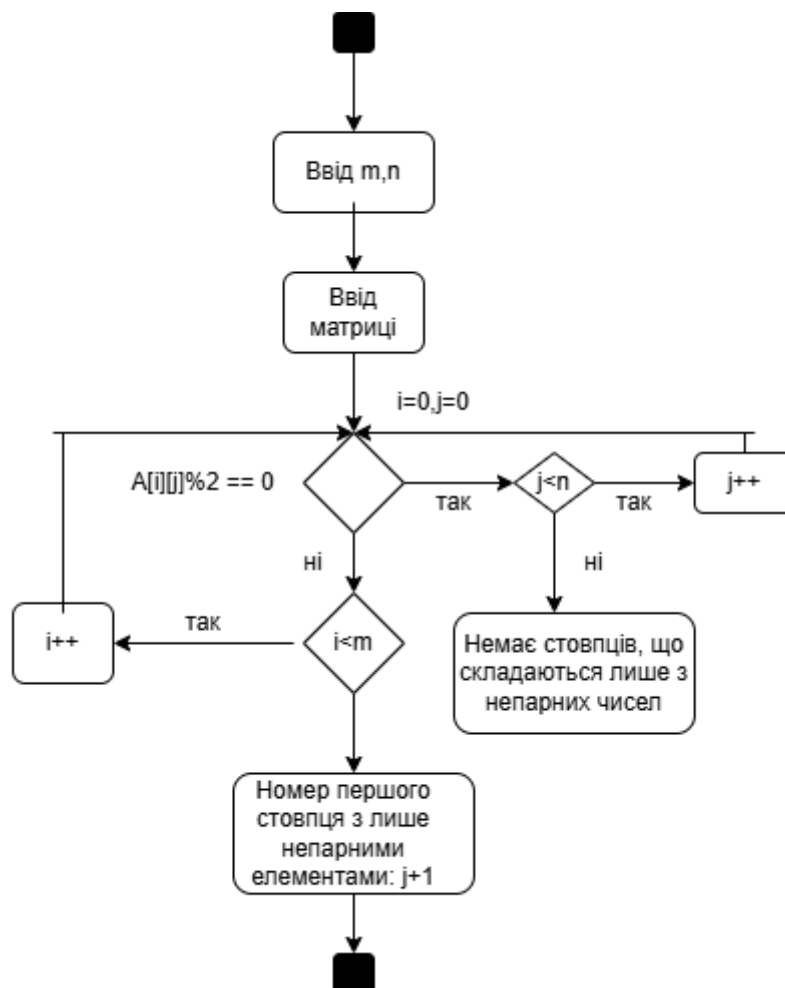


Рисунок 1 – Алгоритм пошуку стовпця з непарними елементами

Лістинг коду вирішення задачі Matrix 35 наведено в дод. А (стор. 6).
Екран роботи програми показаний на рис. Б.1.

Завдання 2.

Вирішення задачі Matrix 60

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

- 1) MAX, Максимальна кількість рядків і стовпців матриці, const int 20
- 2) m, Кількість рядків у матриці, int, 2 ... 20

- 3) n , Кількість стовпців у матриці, `int`, $2 \dots 20$
 4) A , Вхідна двовимірна ціла матриця розміром $m \times n$, `int`
 $A[\text{MAX}][\text{MAX}]$, кожен елемент — будь-яке ціле число

Вихідні дані (ім'я, опис, тип):

- 1) A (початкова), Матриця, надрукована у вихідному вигляді
 2) A (дзеркальна), Та сама матриця після відображення по вертикальній осі

Алгоритм вирішення показано на рис. 2

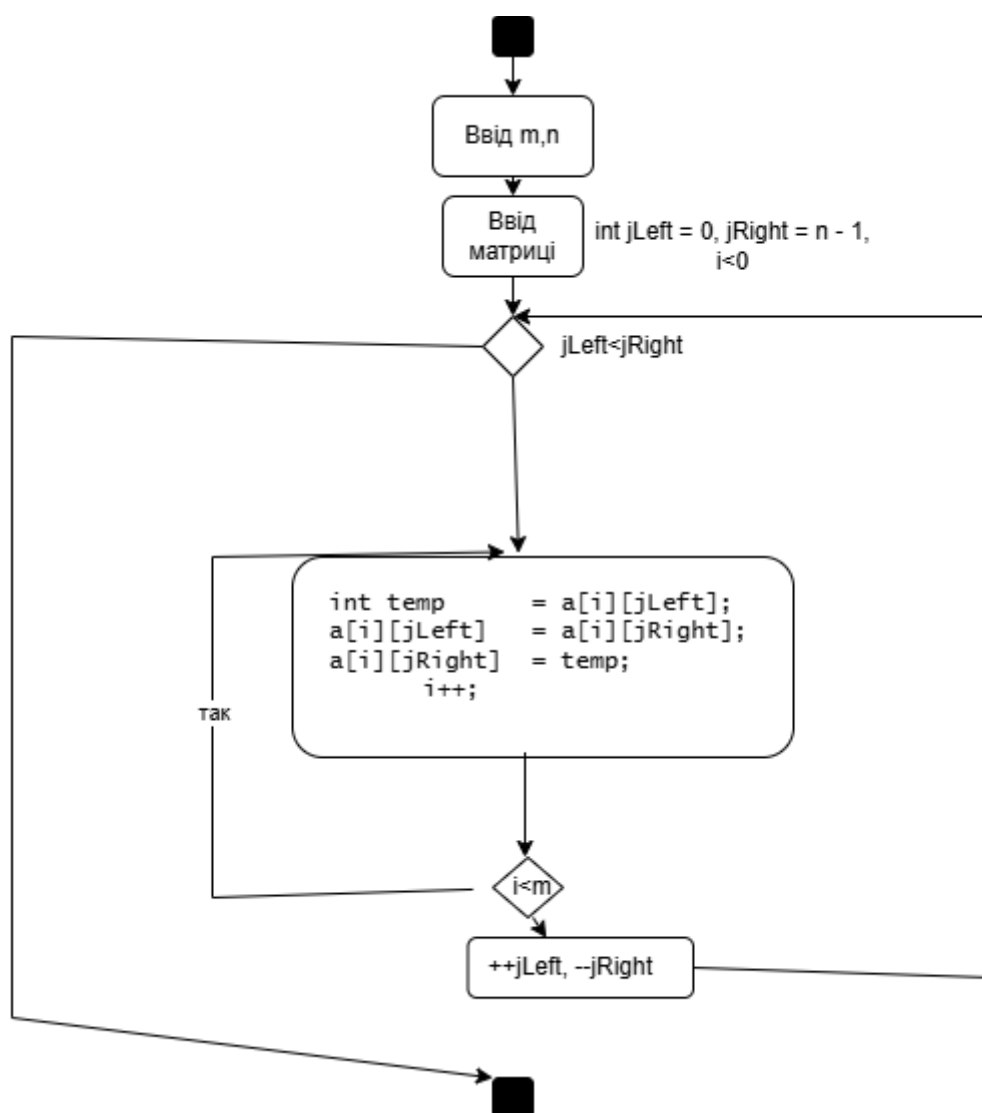


Рисунок 2 – Алгоритм дзеркального відображення

Лістинг коду вирішення задачі Matrix 60 наведено в дод. А (стор. 6).
 Екран роботи програми показаний на рис. Б.2.

ВИСНОВКИ

Було вивчено прийоми обробки двовимірних цілих масивів: пошук «чисто непарного» стовпця та дзеркальне відображення матриці. Закріплено навички роботи зі статичними масивами, перевірки введення й модульної організації коду. Найбільші труднощі спричинила обробка некоректного вводу, проте вони усунуті за допомогою контролю стану потоку та циклів повторного введення

ДОДАТОК А

Лістинг коду програми

```
#include <iostream>
#include <limits>

using namespace std;

/* -----
   Глобальні константи
   ----- */
const int MAX = 20;    // максимальна кількість рядків / стовпців

/* -----
   Прототипи допоміжних функцій з параметрами
   ----- */
void    inputSize(int& m, int& n);
void    inputMatrix(int a[][MAX], int m, int n);
void    printMatrix(const int a[][MAX], int m, int n);
bool    isColumnAllOdd(const int a[][MAX], int m, int col);
int     firstOddOnlyColumn(const int a[][MAX], int m, int n);
void    mirrorMatrix(int a[][MAX], int m, int n);

/* -----
   Завдання 1: пошук першого стовпця, що містить
   тільки непарні числа.
   Функція без параметрів, як вимагає умова.
   ----- */
void task1()
{
    int m, n;
    int a[MAX][MAX];

    cout << "\n=== Завдання 1 ===\n";
    inputSize(m, n);
    inputMatrix(a, m, n);

    int col = firstOddOnlyColumn(a, m, n);

    if (col == 0)
        cout << "Немає стовпців, що складаються лише з непарних чисел\n";
    else
        cout << "Номер першого стовпця з лише непарними елементами: " << col <<
'\n';
}

/* -----
```

Завдання 2: дзеркальне відображення за вертикальною віссю.

Функція без параметрів, як вимагає умова.

```

----- */
void task2()
{
    int m, n;
    int a[MAX][MAX];

    cout << "\n=== Завдання 2 ===\n";
    inputSize(m, n);
    inputMatrix(a, m, n);

    mirrorMatrix(a, m, n);

    cout << "\nМатриця після дзеркального відображення:\n";
    printMatrix(a, m, n);
}

/* -----
Головне меню з можливістю багаторазового вибору
----- */
int main()
{
    setlocale(0, ""); // Коректне виведення кирилиці
    while (true)
    {
        cout << "\n*****\n";
        cout << " 1 - Виконати завдання 1\n";
        cout << " 2 - Виконати завдання 2\n";
        cout << " 0 - Вийти з програми\n";
        cout << "Ваш вибір: ";

        int choice;
        cin >> choice;

        if (cin.fail()) { // якщо введено не число
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Помилка введення. Спробуйте ще раз.\n";
            continue;
        }

        switch (choice)
        {
            case 1: task1(); break;
            case 2: task2(); break;
            case 0: cout << "Завершення роботи.\n"; return 0;
            default: cout << "Неприпустимий вибір. Спробуйте ще раз.\n";
        }
    }
}

```

```

}

/* =====
Реалізація допоміжних функцій
===== */

/* ---- Введення розмірів матриці з перевіркою ---- */
void inputSize(int& m, int& n)
{
    while (true)
    {
        cout << "Введіть кількість рядків (2-20): ";
        cin >> m;
        cout << "Введіть кількість стовпців (2-20): ";
        cin >> n;

        if (cin.fail() || m < 2 || m > MAX || n < 2 || n > MAX)
        {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Розміри матриці мають бути цілими числами у діапазоні 2-20. Спробуйте ще раз.\n";
        }
        else
            return;          // коректні дані
    }
}

/* ---- Введення матриці ---- */
void inputMatrix(int a[][MAX], int m, int n)
{
    cout << "Введіть елементи матриці (" << m << "x" << n << "):\n";
    for (int i = 0; i < m; ++i)
        for (int j = 0; j < n; ++j)
        {
            while (true)
            {
                cout << " a[" << i + 1 << "][" << j + 1 << "] = ";
                cin >> a[i][j];

                if (cin.fail())
                {
                    cin.clear();
                    cin.ignore(numeric_limits<streamsize>::max(), '\n');
                    cout << "Помилка введення. Спробуйте ще раз.\n";
                }
                else
                    break;
            }
        }
}

```



```

}

/* ---- Виведення матриці ---- */
void printMatrix(const int a[][MAX], int m, int n)
{
    for (int i = 0; i < m; ++i)
    {
        for (int j = 0; j < n; ++j)
            cout << a[i][j] << '\t';
        cout << '\n';
    }
}

/* ---- Перевірка, чи всі елементи стовпця непарні ---- */
bool isColumnAllOdd(const int a[][MAX], int m, int col)
{
    for (int i = 0; i < m; ++i)
        if (a[i][col] % 2 == 0)    // хоча б один парний - не підходить
            return false;
    return true;
}

/* ---- Пошук першого «непарного» стовпця; нумерація з 1 ---- */
int firstOddOnlyColumn(const int a[][MAX], int m, int n)
{
    for (int j = 0; j < n; ++j)
        if (isColumnAllOdd(a, m, j))
            return j + 1;    // +1, бо користувачу зручніше бачити з 1
    return 0;                // не знайдено
}

/* ---- Дзеркальне відображення (обмін стовпців) ---- */
void mirrorMatrix(int a[][MAX], int m, int n)
{
    for (int jLeft = 0, jRight = n - 1; jLeft < jRight; ++jLeft, --jRight)
        for (int i = 0; i < m; ++i)
        {
            int temp      = a[i][jLeft];
            a[i][jLeft]    = a[i][jRight];
            a[i][jRight]   = temp;
        }
}

```

ДОДАТОК Б
Скрін-шоти вікна виконання програми

```
*****
1 - Виконати завдання 1
2 - Виконати завдання 2
0 - Вийти з програми
Ваш вибір: 1

=== Завдання 1 ===
Введіть кількість рядків (2-20): 2
Введіть кількість стовпців (2-20): 2
Введіть елементи матриці (2×2):
a[1][1] = 1
a[1][2] = 2
a[2][1] = 3
a[2][2] = 4
Номер першого стовпця з лише непарними елементами: 1
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання
Matrix 35

```
1 - Виконати завдання 1
2 - Виконати завдання 2
0 - Вийти з програми
Ваш вибір: 2

=== Завдання 2 ===
Введіть кількість рядків (2-20): 2
Введіть кількість? стовпців (2-20): 2
Введіть елементи матриці (2×2):
a[1][1] = 1
a[1][2] = 2
a[2][1] = 3
a[2][2] = 4

Матриця після дзеркального відображення:
2      1
4      3
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання
Matrix 60