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 на мові С ++"»

ХАІ.301. 174. 319. 21 ЛР

Виконав студент гр. _____319_____

_____ Сисоєв Володимир_____
(підпис, дата) (П.І.Б.)

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

2023

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення двовимірних масивів (матриць) на мові С ++ і реалізувати оголошення, введення з консолі, обробку і виведення в консоль матриць на мові С ++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на аналіз і виведення елементів матриці.
Введення і виведення даних здійснити в командному вікні.
Matrix43. Дана матриця розміру M × N. Знайти кількість її стовпців, елементи яких впорядковані за спаданням.
Завдання 2. Перетворити матрицю відповідно до свого варіанту завданнярозмір матриці і його елементи ввести з консолі. Вивести результати у консоль.
Matrix60. Дана матриця розміру M × 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 43
Вхідні дані (ім’я, опис, тип, обмеження):
1)Максимальний розмір ROW = COL = 10 , ціле , константа
2)кількість рядків row, ціле, 2-10
2)кількість стовпців col , ціле 2-10
2)цілочисельний двовимірний масив matr1.
Вихідні дані (ім’я, опис, тип):
1)елементи матриці
2)кількість стовпців впорядкованих за спаданням count

Алгоритм вирішення показано на рис. 1

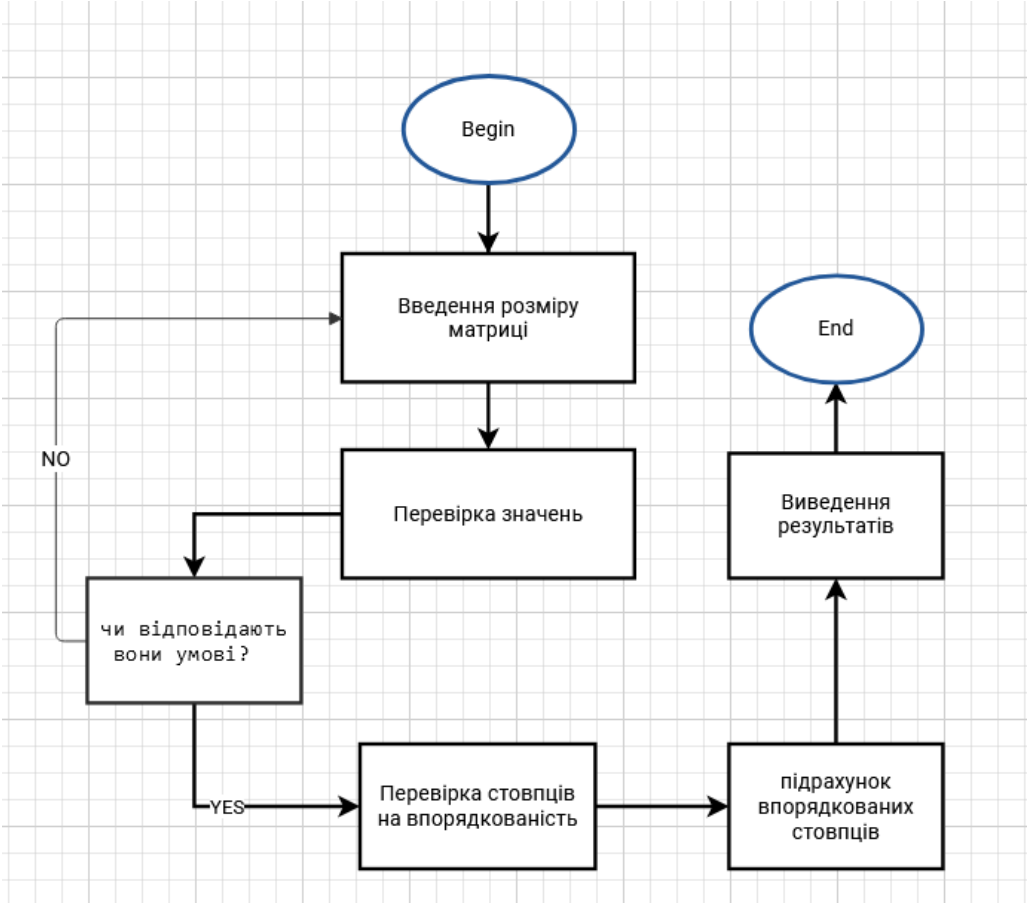

Рисунок 1 – Matrix 43

Лістинг коду вирішення задачі розділ і номер задачі 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)Максимальний розмір ROW = COL = 10 , ціле , константа
2)кількість рядків row, ціле, 2-10
2)кількість стовпців col , ціле 2-10
3) цілочисельний двовимірний масив matr1.

Вихідні дані (ім’я, опис, тип):
1)елементи матриці
2)дзеркально відображена матриця matrix

Алгоритм вирішення показано на рис. 2

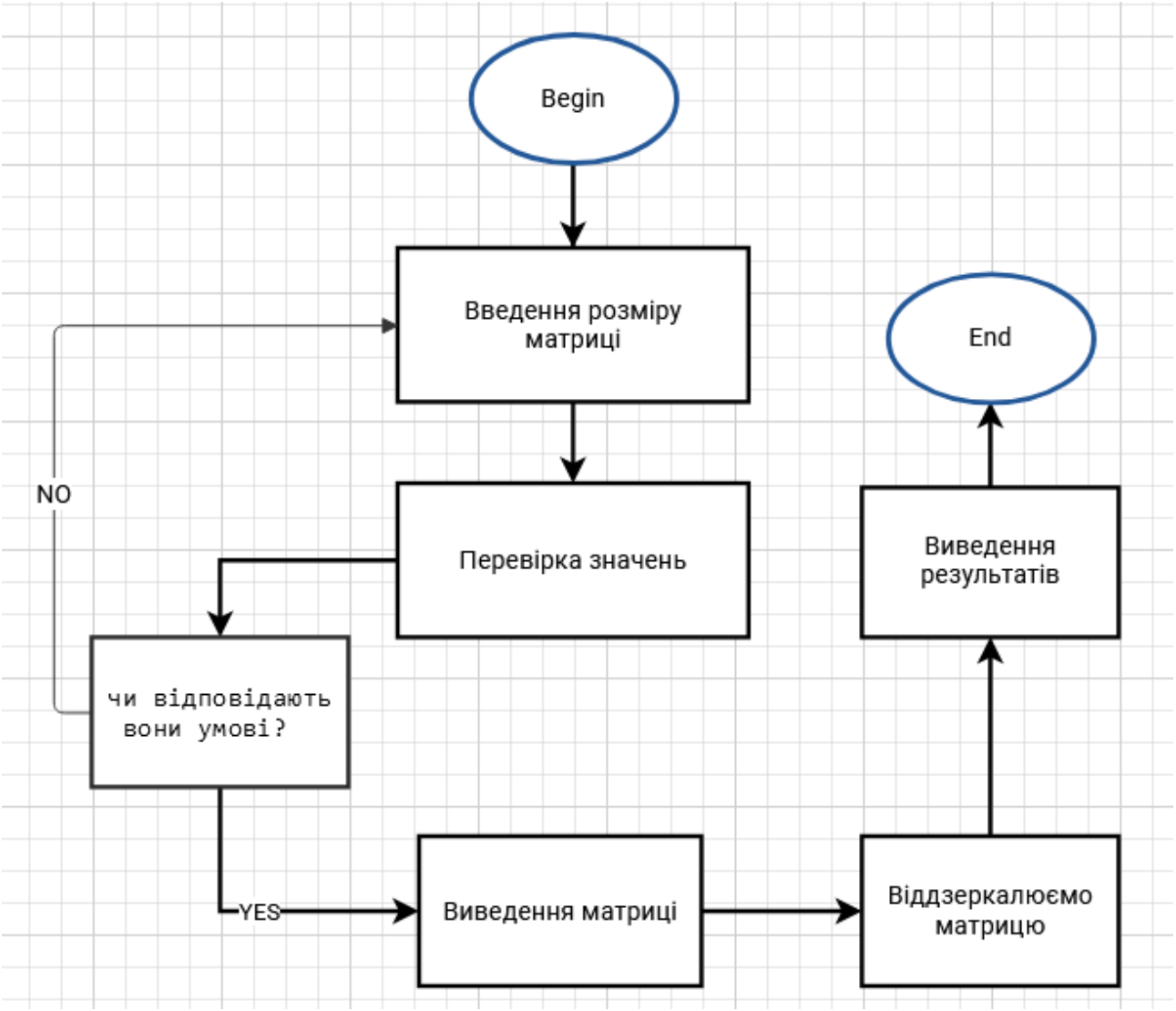

Рисунок 2 – Matrix 60

Лістинг коду вирішення задачі розділ і номер задачі наведено в дод. А (стор. 7).

Екран роботи програми показаний на рис. Б.2.

ВИСНОВКИ

Було вивчено теоретичний матеріал з основ представлення двовимірних масивів(матриць) на мові С ++ і реалізовано оголошення, введення з консолі, оброблення і виведення в консоль матриць на мові С ++ в середовищі Visual Studio.

ДОДАТОК А

Лістинг коду програми

MATRIX 43

```
const int ROW = 10, COL = 10; //matrix size limit

void get_matr43(int in_matr[ROW][COL], int& in_row, int& in_col) {
    do
    {
        cout << "Enter rows count (2-10): "; //request matrix size
        cin >> in_row;
        cout << "Enter columns count (= Rows !!!) (2-10): ";
//request matrix size
        cin >> in_col;
    }
    while (in_col < 2 || in_col>COL || in_row <2 || in_row>ROW || in_row !=
in_col); // validation
    cout << "Enter elements: " << endl; // request elements of matrix
    for (int i = 0; i < in_row; i++)
        for (int j = 0; j < in_col; j++)
            cin >> in_matr[i][j];}
//request matrix

bool check_col(int matrix[ROW][COL], int rows, int column) {
    for (int i = 1; i < rows; ++i) {
        if (matrix[i][column] > matrix[i - 1][column]) {
            return false;
        }
    }
    return true;
}
//column descending check

int count_col(int matrix[ROW][COL], int rows, int columns) {
    int count = 0;
    for (int j = 0; j < columns; ++j) {
        if (check_col(matrix, rows, j)) {
            ++count;
        }
    }
    return count;}
// count columns

void show_matr43(const int out_matr[ROW][COL], const int row, const int col) {
    cout << endl << "Matrix: " << endl;
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++)
            cout << out_matr[i][j] << "\t";
        cout << endl;}
}
//show matrix

void matrix43() {
    int matr1[ROW][COL];
    int row, col;
    get_matr43(matr1, row, col);
    show_matr43(matr1, row, col);
    int count = count_col(matr1, row, col);
    cout << "Number of columns with descending elements: " << count << endl;
    cout << endl;}
//task function
```

MATRIX 60

```
void get_matr60(int in_matr[ROW][COL], int& in_row, int& in_col) {
    do
    {
        cout << "Enter rows count (2-10): ";//request matrix size
        cin >> in_row;
        cout << "Enter columns count (= Rows !!!) (2-10): ";//request matrix
size
        cin >> in_col;
    }
    while (in_col < 2 || in_col>COL || in_row <2 || in_row>ROW || in_row !=
in_col); // validation
    cout << "Enter elements: " << endl;// request elements of matrix
    for (int i = 0; i < in_row; i++)
        for (int j = 0; j < in_col; j++)
            cin >> in_matr[i][j];}
//request matrix

void show_matr60(const int out_matr[ROW][COL], const int row, const int col) {
    cout << endl << "Matrix: " << endl;
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++)
            cout << out_matr[i][j] << "\t";
        cout << endl;
    }
}
//show matrix

void mirror_matr60(int matrix[ROW][COL], int row, int col) {
    for (int i = 0; i < row; ++i) {
        for (int j = 0; j < col / 2; ++j) {
            int temp = matrix[i][j];
            matrix[i][j] = matrix[i][col - 1 - j];
            matrix[i][col - 1 - j] = temp;
        }
    }
}
// mirror matrix

void show_mirror_matr60(int matrix[ROW][COL], int row, int col) {
    for (int i = 0; i < row; ++i) {
        for (int j = 0; j < col; ++j) {
            cout << matrix[i][j] << "\t";
        }
        cout << endl;
    }
}
//show mirrored matrix

void matrix60() {
    int matr1[ROW][COL];
    int row, col;
    get_matr60(matr1, row, col);
    show_matr60(matr1, row, col);
    mirror_matr60(matr1, row, col);
    cout << "Mirrored matrix:" << endl;
    show_mirror_matr60(matr1, row, col);}
//task function
```

ДОДАТОК Б
Скрін-шоти вікна виконання програми

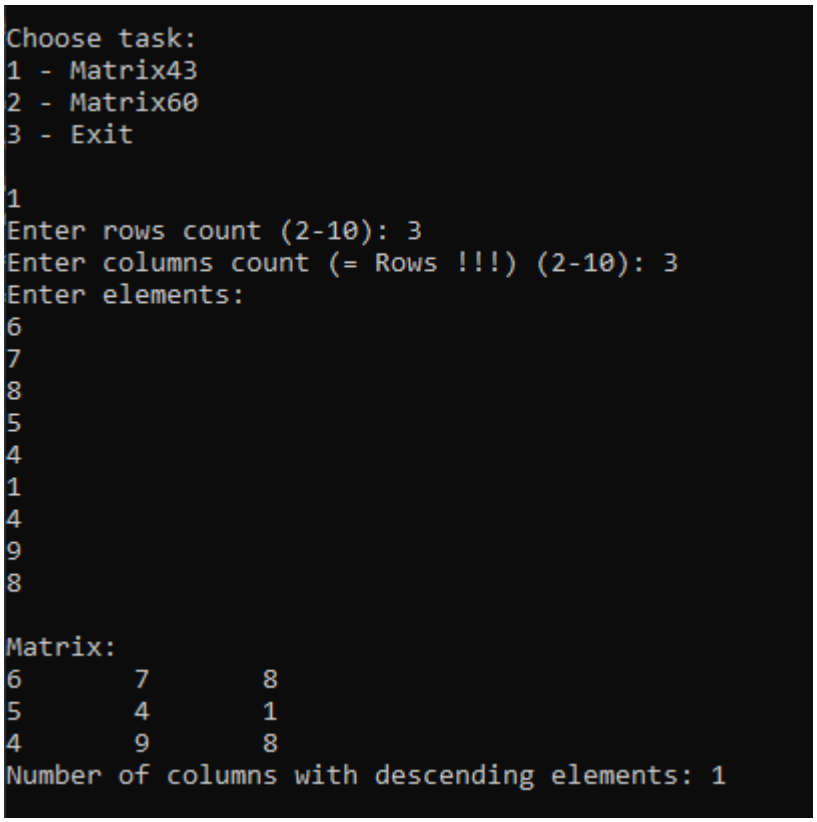

Рисунок Б.1 – Екран виконання програми для вирішення завдання Matrix43

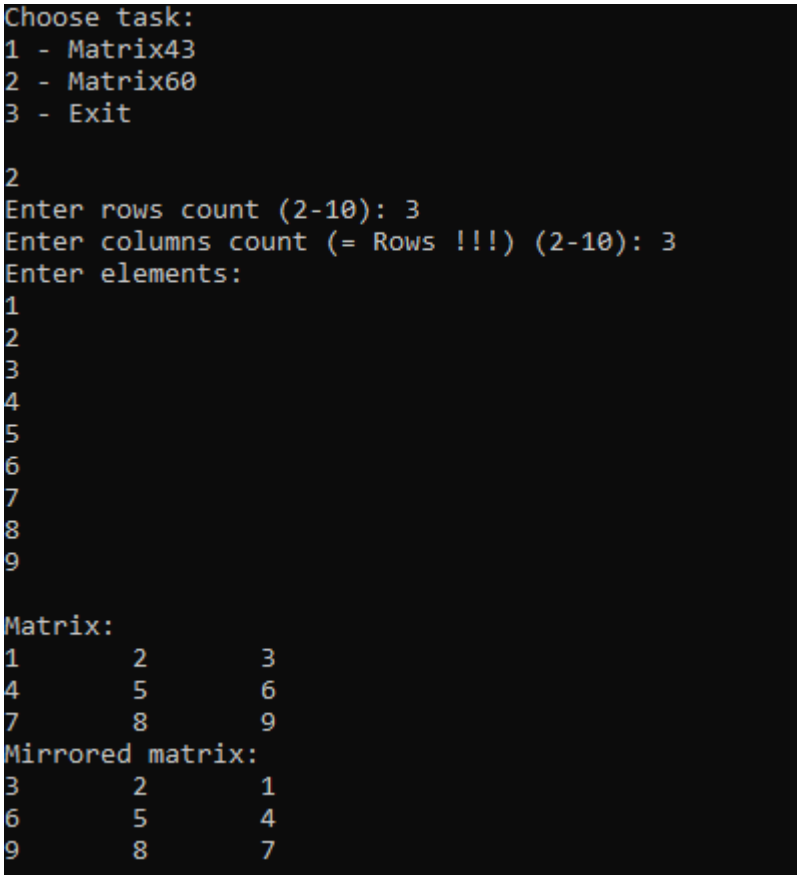

Рисунок Б.2 – Екран виконання програми для вирішення завдання Matrix60