

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. 175. 318. 07 ЛР

Виконав студент гр. _____ 318 _____

_____ Микита Глебов _____
(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.

Matrix45. Дана матриця розміру $M \times N$. Знайти максимальний серед елементів тих стовпців, які впорядковані або по зростанню, або по спадаючій. Якщо впорядковані стовпці в матриці відсутні, то вивести 0.

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

Matrix48. Дана матриця розміру $M \times N$ і цілі числа $K1$ і $K2$ ($1 \leq K1 < K2 \leq N$). Поміняти місцями стовпці матриці з номерами $K1$ і $K2$.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Matrix45.

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

rows - кількість рядків у матриці, ціле число, 2-20.

cols - кількість стовпців у матриці, ціле число, 2-20.

matr[i][j] - елемент матриці, ціле число .

Вихідні дані (ім'я, опис, тип):

max_element - виведення масиву, ціле число, якщо немає таких стовпців - 0

Лістинг коду вирішення задачі Matrix45 наведено в дод. А (стор. 5-8)

Екран роботи програми показаний на рис. Б.1. (дод. Б, стор. 9)

Приклад діаграми для завдання Matrix45 наведено на рис. Б.3. (дод. Б, стор. 11)

Завдання 2.

Вирішення задачі Matrix48.

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

rows - кількість рядків у матриці, ціле число, 2-20.

cols - кількість стовпців у матриці, ціле число, 2-20.

matr[i][j] - елемент матриці, ціле число.

Вихідні дані (ім'я, опис, тип):

matr[i][j] - матриця після обміну стовпців K1 і K2, ціле число, стовпці K1 і K2 поміняні місцями.

Лістинг коду вирішення задачі Matrix48 наведено в дод. А (стор. 5-8)

Екран роботи програми показаний на рис. Б.2. (дод. Б, стор. 10)

Приклад діаграми для завдання Matrix48 наведено на рис. Б.4. (дод. Б, стор. 12)

Завдання 3.

Організація меню.

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

<<Оберіть завдання>>

Вихідні дані (ім'я, опис, тип):

<< Завдання 1 (максимум серед упорядкованих стовпців)>> - якщо введено число 1, виводяться розрахунки задачі Matrix45.

<<Завдання 2 (обмін стовпців K1 і K2)>> - якщо введено число 2, виводяться розрахунки задачі Matrix48.

Лістинг коду вирішення завдання 3 наведено в дод. А (стор. 5-8)

Приклад діаграми для завдання 3 наведено на рис. Б.5. (дод. Б, стор. 13)

ВИСНОВКИ

Було вивчено теоретичний матеріал з основ представлення одновимірних і масивів на мові C++. Були опрацьовані взаємодії з двовимірними масивами та їх використанням для виконання завдань в середовищі Visual Studio.

ДОДАТОК А

Лістинг коду програми

```
#include <iostream>    // Стандартна бібліотека вводу/виводу
#include <climits>      // Для використання INT_MIN
#include <string>        // Для зчитування рядків
#include <cstdio>        // Для sscanf

using namespace std;

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

// Оголошення функцій
void task1(); // Завдання 1
void task2(); // Завдання 2
void get_matr(int matr[M][N], int &rows, int &cols); // Введення матриці
void show_matr(const int matr[M][N], int rows, int cols); // Виведення матриці
bool is_sorted_up(const int matr[M][N], int rows, int col); // Перевірка на зростання
bool is_sorted_down(const int matr[M][N], int rows, int col); // Перевірка на спадання
int max_in_column(const int matr[M][N], int rows, int col); // Максимум у стовпці

// Головна функція
int main() {
    int choice;
    do {
        cout << "\n===== " << endl;
        cout << "Меню:\n";
        cout << "1 - Завдання 1 (максимум серед упорядкованих стовпців)\n";
        cout << "2 - Завдання 2 (обмін стовпців K1 i K2)\n";
        cout << "0 - Вихід\n";
        cout << "Ваш вибір: ";
        cin >> choice;

        switch (choice) {
            case 0:
                cout << "Завершення програми..." << endl;
                break;
            case 1:
                task1();
                break;
            case 2:
                task2();
                break;
            default:
                break;
        }
    } while (choice != 0);
}
```

```

        cout << "Неправильний номер завдання!" << endl;
    }
} while (choice != 0);

return 0;
}

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

    cout << "Введіть елементи матриці:" << endl;
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++) {
            cout << "matr[" << i + 1 << "][" << j + 1 << "] = ";
            cin >> matr[i][j];
        }
}

// Функція виведення матриці
void show_matr(const int matr[M][N], int rows, int cols) {
    cout << "\nМатриця:" << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++)
            cout << matr[i][j] << "\t";
        cout << endl;
    }
}

// Чи впорядкований стовпець по зростанню
bool is_sorted_up(const int matr[M][N], int rows, int col) {
    for (int i = 0; i < rows - 1; i++)
        if (matr[i][col] > matr[i + 1][col])
            return false;
    return true;
}

// Чи впорядкований стовпець по спаданню
bool is_sorted_down(const int matr[M][N], int rows, int col) {
    for (int i = 0; i < rows - 1; i++)
        if (matr[i][col] < matr[i + 1][col])
            return false;
    return true;
}

```

```

// Знаходження максимуму в стовпці
int max_in_column(const int matr[M][N], int rows, int col) {
    int max_val = matr[0][col];
    for (int i = 1; i < rows; i++)
        if (matr[i][col] > max_val)
            max_val = matr[i][col];
    return max_val;
}

// Matrix45. Дана матриця розміру M × N. Знайти максимальний серед елементів тих
// стовпців, які впорядковані або по зростанню, або по спадаючій. Якщо впорядковані
// стовпці в матриці відсутні, то вивести 0.
void task1() {
    int matr[M][N], rows, cols;
    get_matr(matr, rows, cols);
    show_matr(matr, rows, cols);

    int max_element = INT_MIN;
    bool found = false;

    // Перевірка кожного стовпця
    for (int j = 0; j < cols; j++) {
        if (is_sorted_up(matr, rows, j) || is_sorted_down(matr, rows, j)) {
            int cur_max = max_in_column(matr, rows, j);
            if (!found || cur_max > max_element) {
                max_element = cur_max;
                found = true;
            }
        }
    }

    if (found)
        cout << "\nМаксимум серед елементів впорядкованих стовпців: " <<
max_element << endl;
    else
        cout << "\nВ матриці немає впорядкованих стовпців. Результат: 0" << endl;
}

// Matrix48. Дана матриця розміру M × N і цілі числа K1 і K2 (1 ≤ K1 < K2 ≤ N).
// Поміняти місцями стовпці матриці з номерами K1 і K2.
void task2() {
    int matr[M][N], rows, cols;
    get_matr(matr, rows, cols);
    show_matr(matr, rows, cols);

    int k1, k2;

```

```

// Повторне введення, поки не буде коректно
do {
    cout << "Введіть номери стовпців K1 і K2 для обміну (від 1 до " << cols
<< "): ";

    cin.clear();
    cin.ignore(INT_MAX, '\n');

    string line;
    getline(cin, line);

    int parsed = sscanf(line.c_str(), "%d %d", &k1, &k2);

    if (parsed < 2) {
        if (parsed == 1)
            cout << "Введено лише одне число. Потрібно два!\n";
        else
            cout << "Некоректне введення. Введіть два цілі числа через
пробіл.\n";
        continue;
    }

    if (k1 < 1 || k2 < 1 || k1 > cols || k2 > cols) {
        cout << "Номери мають бути в межах від 1 до " << cols << "!\n";
        continue;
    }

    if (k1 == k2) {
        cout << "Номери стовпців повинні бути різними!\n";
        continue;
    }

    break;
} while (true);

// Обмін значень у стовпцях
for (int i = 0; i < rows; i++) {
    int temp = matr[i][k1 - 1];
    matr[i][k1 - 1] = matr[i][k2 - 1];
    matr[i][k2 - 1] = temp;
}

cout << "\nМатриця після обміну стовпців " << k1 << " і " << k2 << ":\n";
show_matr(matr, rows, cols);
}

```


ДОДАТОК Б
Скрін-шоти вікна виконання програми

```
Меню:
1 - Завдання 1 (максимум серед упорядкованих стовпців)
2 - Завдання 2 (обмін стовпців K1 і K2)
0 - Вихід
Ваш вибір: 1
Введіть кількість рядків (2-20): 3
Введіть кількість стовпців (2-20): 4
Введіть елементи матриці:
matr[1][1] = 2
matr[1][2] = 3
matr[1][3] = 8
matr[1][4] = 5
matr[2][1] = 1
matr[2][2] = 4
matr[2][3] = 5
matr[2][4] = 7
matr[3][1] = 3
matr[3][2] = 9
matr[3][3] = 6
matr[3][4] = 1

Матриця:
2      3      8      5
1      4      5      7
3      9      6      1

Максимум серед елементів впорядкованих стовпців: 9
```

Рисунок Б.1 - Екран виконання програми для вирішення завдання Matrix45

```
Ваш вибір: 2
Введіть кількість рядків (2-20): 4
Введіть кількість стовпців (2-20): 3
Введіть елементи матриці:
matr[1][1] = 2
matr[1][2] = 1
matr[1][3] = 7
matr[2][1] = 3
matr[2][2] = 6
matr[2][3] = 45
matr[3][1] = 8
matr[3][2] = 32
matr[3][3] = 7
matr[4][1] = 2
matr[4][2] = 6
matr[4][3] = 2

Матриця:
2      1      7
3      6     45
8     32      7
2      6      2
Введіть номери стовпців K1 і K2 для обміну (від 1 до 3): 1 3

Матриця після обміну стовпців 1 і 3:

Матриця:
7      1      2
45     6      3
7     32      8
2      6      2
```

Рисунок Б.2 - Екран виконання програми для вирішення завдання Matrix48

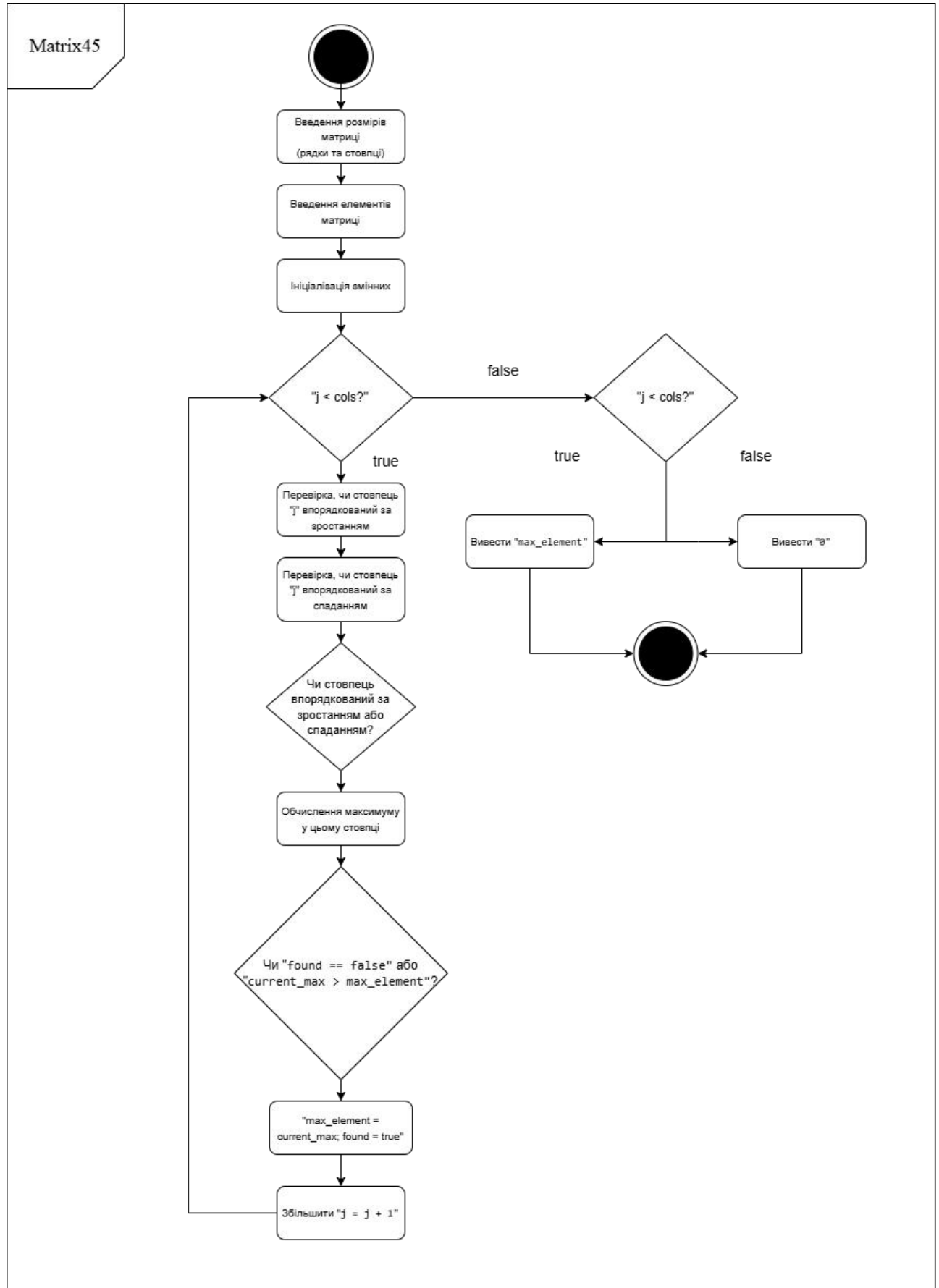


Рисунок Б.3 - Діаграма для завдання Matrix45

Matrix48

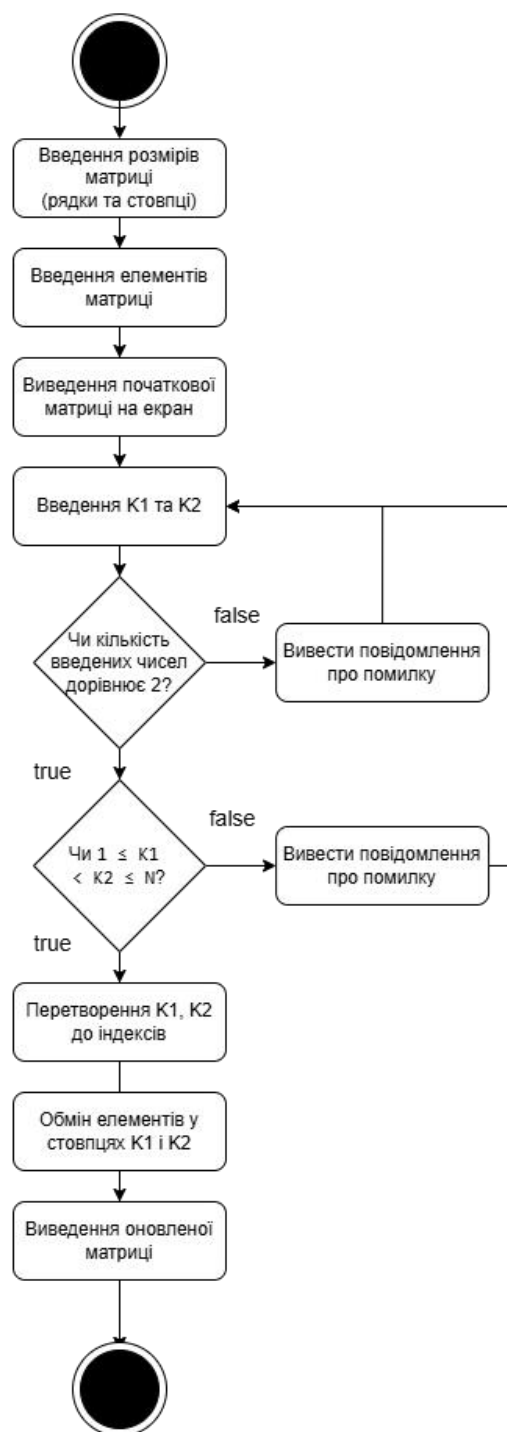


Рисунок Б.4 - Діаграма для завдання Matrix48

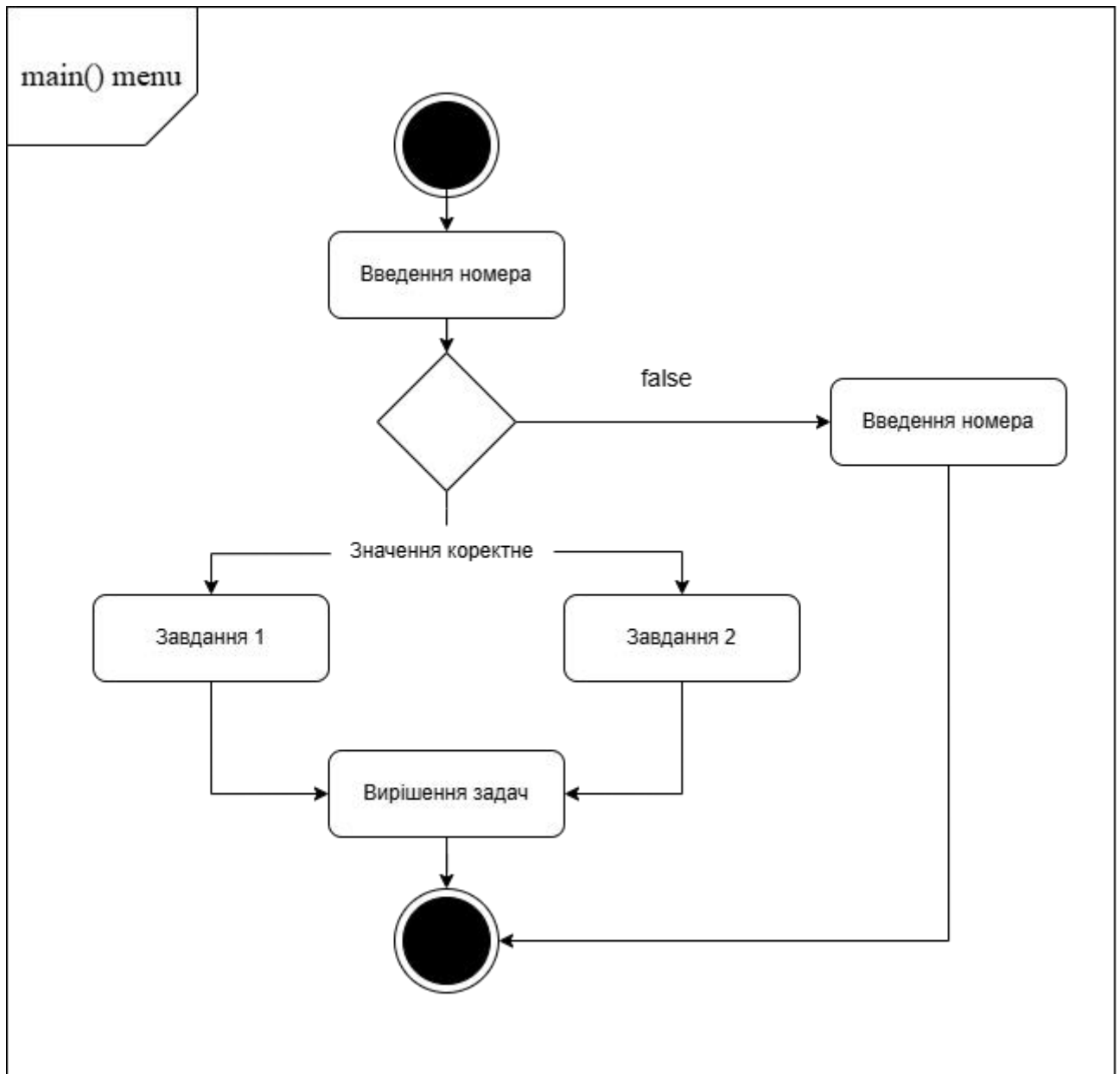


Рисунок Б.5 - Діаграма для завдання 3