Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 9 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження алгоритмів обходу масивів»

Варіант 21

Виконав студент ІП-11, Ляля Іван Олександрович

Перевірила Мартинова Оксана Петрівна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ 2021

**Лабораторна робота 9**

**Дослідження алгоритмів обходу масивів**

Мета – дослідити алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Задача:

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом .

2. Ініціювання змінної, що описана в п.1 даного завдання.

3. Обчислення змінної, що описана в п.1, згідно з варіантом.

|  |  |
| --- | --- |
| **№** | **Опис варіанту** |
| 21 | Задано матрицю дійсних чисел A[m,n]. При обході матриці по рядках знайти в ній останній максимальний елемент Х і його місцезнаходження. Обміняти знайдене значення Х з елементом першого стовбця. |

Постановка задачі:

Спочатку ми зчитуємо кількість рядків та стовпців матриці n та m відповідно і оголошуємо масив дійних чисел такого розміру. Далі перебором арифметичним циклом всіх елементів матриці присвоюємо кожному з них випадкове дійсне з проміжку [-10;10]. Ініціюємо змінну dir=1, що визначатиме напрямок перебору елементів в поточному рядку. Ініціюємо змінні n\_max та m\_max зі значенням 1, які позначатимуть рядок і стовпець останнього максимального елемента Х. Далі для всіх рядків треба виконати таке: якщо dir>0, тобто напрямок обходу зліва направо, то для елемента кожного стовпчика відповідного рядка перевірити, чи не більший рівний він поточному найбільшому – якщо буде два однакових, то не пропуститься саме ОСТАННІЙ максимальний; якщо ця умова виконується, то записати розташування елемента, що перевіряється, в змінні n\_max, m\_max. Якщо ж dir<0, то напрямок справа наліво: для всіх елементів поточного рядка перебрати стовпці від m до 1 з перевіркою умови строгої більшості елемента, що перевіряється, за поточний максимальний. В кінці обходу рядка змінити знак напрямку dir. Маємо індекси останнього максимального елемента, за індексами витягуємо сам елемент Х, виводимо його і його індекси. Міняємо місцями елемент A[n\_max][m\_max] з A[n\_max][1] і виводимо нову матрицю.

Математична модель:

|  |  |  |
| --- | --- | --- |
| **Змінна** | **Значення** | **Тип** |
| **A[][]** | **Двовимірний вхідний масив** | **Масив дійсних чисел** |
| **n, m** | **Кількість рядків, стовпчиків матриці** | **Натуральні числа** |
| **dir** | **Напрямок при обході поточного рядка матриці** | **Ціле число** |
| **n\_max, m\_max** | **Індекси останнього максимального елемента матриці** | **Натуральне число** |
| **X** | **Значення останнього максимального елемента в матриці** | **Дійсне число** |

**Крок 1**. Визначаємо основні дії

**Крок 2**. Деталізуємо опис та ініціювання матриці

**Крок 3**. Деталізуємо обхід матриці "змійкою" по рядках з одночасним пошуком останнього максимального елемента X

**Крок 4**. Деталізуємо обмін знайденого значення Х з елементом першого стовбця

**Крок 5**. Деталізуємо виведення потрібних даних

***Псевдокоди з блок-схемами:***

*Крок 1*

**початок**

опис матриці дійсних чисел A[n,m]

ініціювання A[n][m]

обхід матриці "змійкою" по рядках з одночасним пошуком останнього максимального елемента X

вивести останній максимальний елемент Х і його місцезнаходження

обміняти знайдене значення Х з елементом першого стовбця

вивести нову матрицю

**кінець**

*Крок 2*

**Початок**

**Ввести** n,m;

**float** A[n][m];

**для всіх** і **від** 1 **до** n **з кроком** 1:

**для всіх** j **від** 1 **до** m **з кроком** 1:

A[i][j]=random[-10;10];

обхід матриці "змійкою" по рядках з одночасним пошуком останнього максимального елемента X

вивести останній максимальний елемент Х і його місцезнаходження

обміняти знайдене значення Х з елементом першого стовбця

вивести нову матрицю

**кінець**

*Крок 3*

**Початок**

**Ввести** n,m;

**float** A[n][m];

**для всіх** і **від** 1 **до** n **з кроком** 1:

**для всіх** j **від** 1 **до** m **з кроком** 1:

A[i][j]=random[-10;10];

**int** dir=1, n\_max=1, m\_max=1;

**для всіх** і **від** 1 **до** n **з кроком** 1:

**якщо** (dir>0):

**для всіх** j **від** 1 **до** m **з кроком** 1:

**якщо** (A[i][j] >= A[n\_max][m\_max]), **то** n\_max=i, m\_max=j;

**інакше:**

**для всіх** j **від** m **до** 1 **з кроком** -1:

**якщо** (A[i][j] > A[n\_max][m\_max]), **то** n\_max=i, m\_max=j;

dir=-dir;

**float** X=A[n\_max][m\_max];

вивести останній максимальний елемент Х і його місцезнаходження

обміняти знайдене значення Х з елементом першого стовбця

вивести нову матрицю

**кінець**

*Крок 4*

**Початок**

**Ввести** n,m;

**float** A[n][m];

**для всіх** і **від** 1 **до** n **з кроком** 1:

**для всіх** j **від** 1 **до** m **з кроком** 1:

A[i][j]=random[-10;10];

**int** dir=1, n\_max=1, m\_max=1;

**для всіх** і **від** 1 **до** n **з кроком** 1:

**якщо** (dir>0):

**для всіх** j **від** 1 **до** m **з кроком** 1:

**якщо** (A[i][j] >= A[n\_max][m\_max]), **то** n\_max=i, m\_max=j;

**інакше:**

**для всіх** j **від** m **до** 1 **з кроком** -1:

**якщо** (A[i][j] > A[n\_max][m\_max]), **то** n\_max=i, m\_max=j;

dir=-dir;

**float** X=A[n\_max][m\_max];

вивести останній максимальний елемент Х і його місцезнаходження

A[n\_max][m\_max] = A[n\_max][1];

A[n\_max][1] = X;

вивести нову матрицю

**кінець**

*Крок 5*

**Початок**

**Ввести** n,m;

**float** A[n][m];

**для всіх** і **від** 1 **до** n **з кроком** 1:

**для всіх** j **від** 1 **до** m **з кроком** 1:

A[i][j]=random[-10;10];

**int** dir=1, n\_max=1, m\_max=1;

**для всіх** і **від** 1 **до** n **з кроком** 1:

**якщо** (dir>0):

**для всіх** j **від** 1 **до** m **з кроком** 1:

**якщо** (A[i][j] >= A[n\_max][m\_max]), **то** n\_max=i, m\_max=j;

**інакше:**

**для всіх** j **від** m **до** 1 **з кроком** -1:

**якщо** (A[i][j] > A[n\_max][m\_max]), **то** n\_max=i, m\_max=j;

dir=-dir;

**float** X=A[n\_max][m\_max];

**вивести** X, n\_max, m\_max;

A[n\_max][m\_max] = A[n\_max][1];

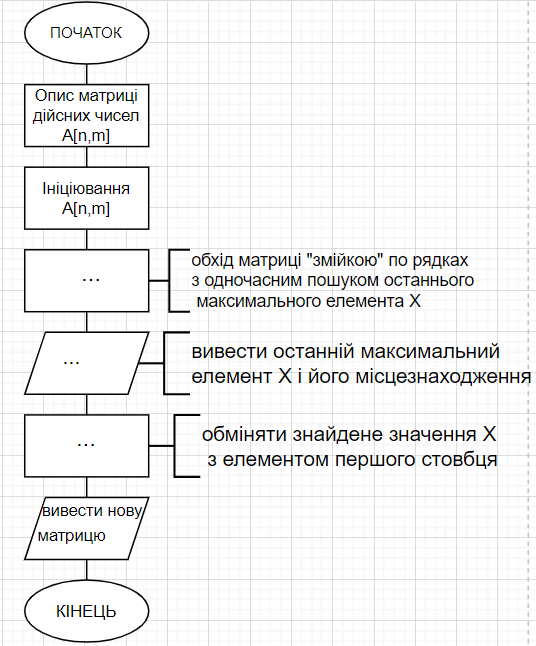
A[n\_max][1] = X;

**для всіх** і **від** 1 **до** n **з кроком** 1:

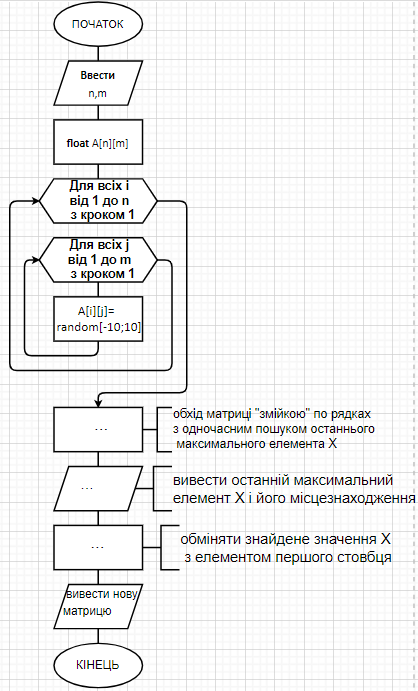
**для всіх** j **від** 1 **до** m **з кроком** 1:

**вивести** A[i][j];

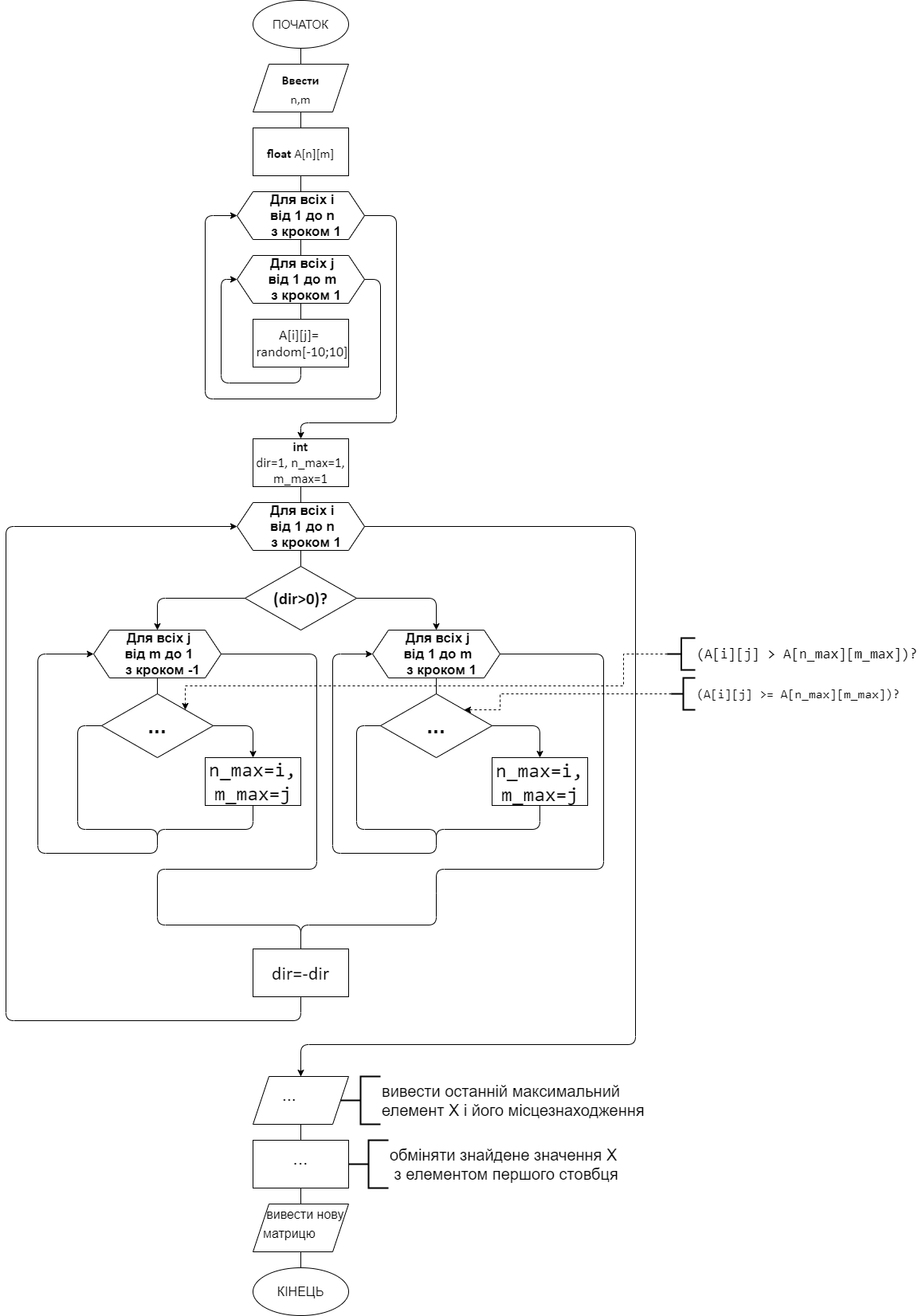
**кінець**

****

**КРОК 1**

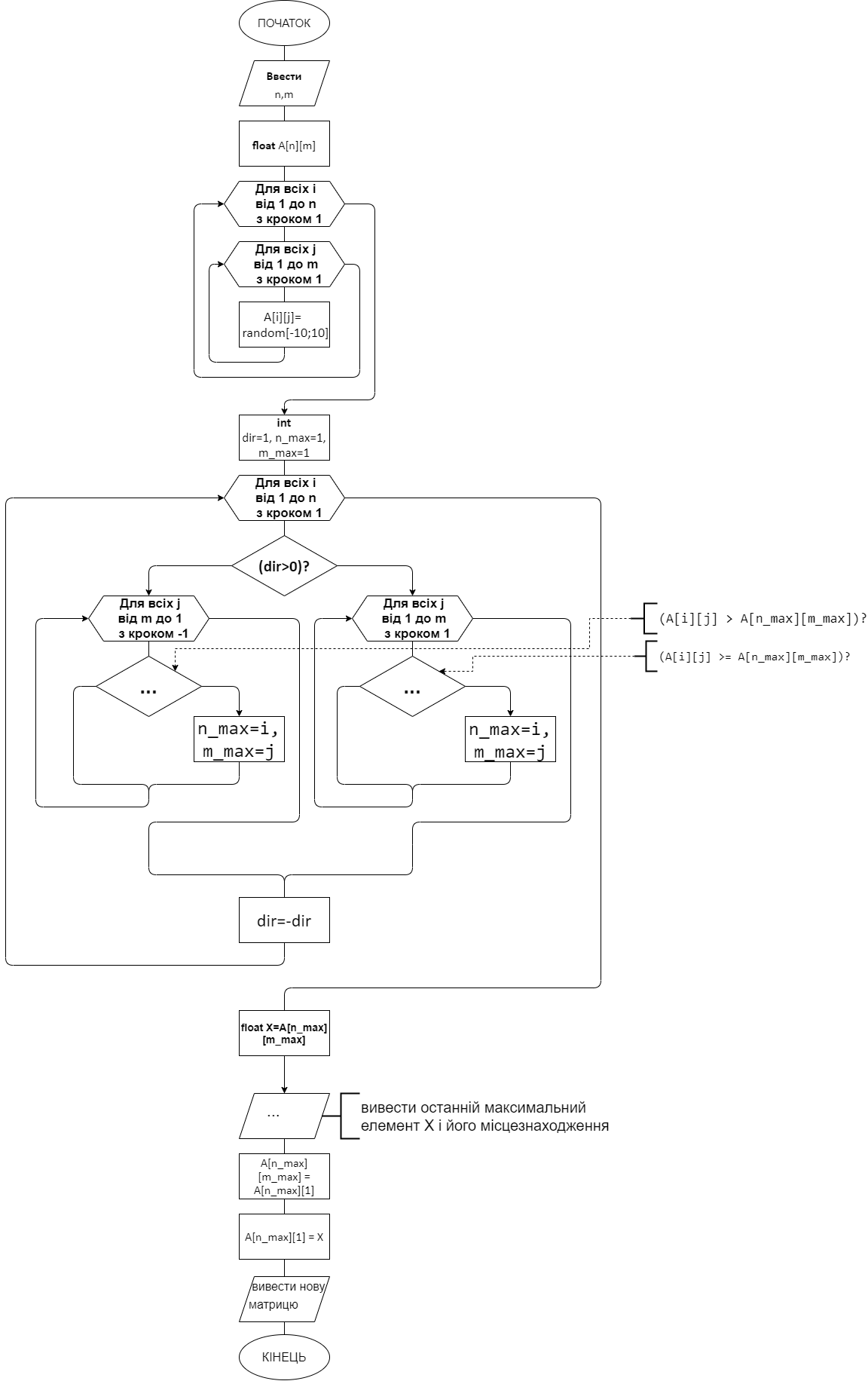
****

**КРОК 2**

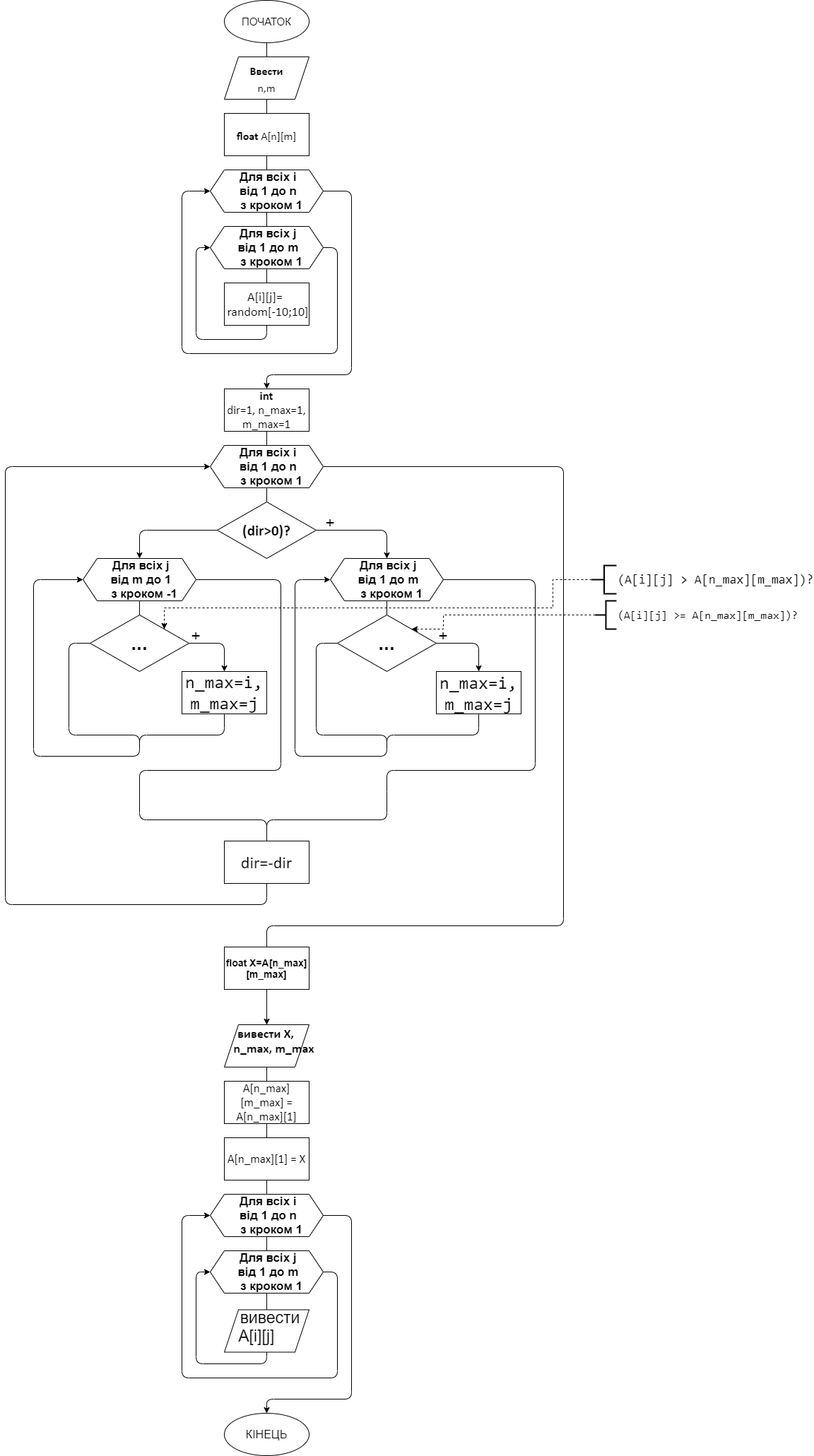
****

**КРОК 3**

**КРОК 4**



**КРОК 5**



**Код і тестування програми на C++:**

#include <iostream>

#include <iomanip>

#include <stdlib.h>

using namespace std;

int main()

{

unsigned int n, m;

cout << "n="; cin >> n;

cout << "m="; cin >> m;

float\*\* A = new float\*[n];

for (int i = 0; i < n; i++)

A[i] = new float[m]; //опис матриці дійсних чисел A[m,n]

srand(time(NULL));

for (int i = 0; i < n; i++) //ініціювання матриці дійсними числами в проміжку від -10 до 10 і одночасне виведення їх

{

for (int j = 0; j < m; j++)

{

A[i][j] = -10 + rand() % 20 + (float)(rand() % 101) / 100;

cout << setw(6) << A[i][j];

}

cout << "\n";

}

int dir = 1, n\_max = 0, m\_max = 0;

for (int i = 0; i < n; i++) //обхід матриці "змійкою" по рядках з одночасним пошуком останнього максимального елемента X

{

if (dir > 0)

for (int j = 0; j < m; j++)

if (A[i][j] >= A[n\_max][m\_max])

{

n\_max = i;

m\_max = j;

}

else

for (int j = m - 1; j >= 0; j--)

if (A[i][j] > A[n\_max][m\_max])

{

n\_max = i;

m\_max = j;

}

dir = -dir;

}

float X = A[n\_max][m\_max];

cout << "Last max element X = " << X << ". It's location: row " << n\_max+1 << " column " << m\_max+1 << endl;

A[n\_max][m\_max] = A[n\_max][0];

A[n\_max][0] = X; //обмінюємо знайдене значення Х з елементом першого стовбця в рядку, де знаходиться Х

cout << "New matrix:" << endl;

for (int i = 0; i < n; i++) //виводимо нову матрицю

{

for (int j = 0; j < m; j++)

cout << setw(6) << A[i][j];

cout << "\n";

}

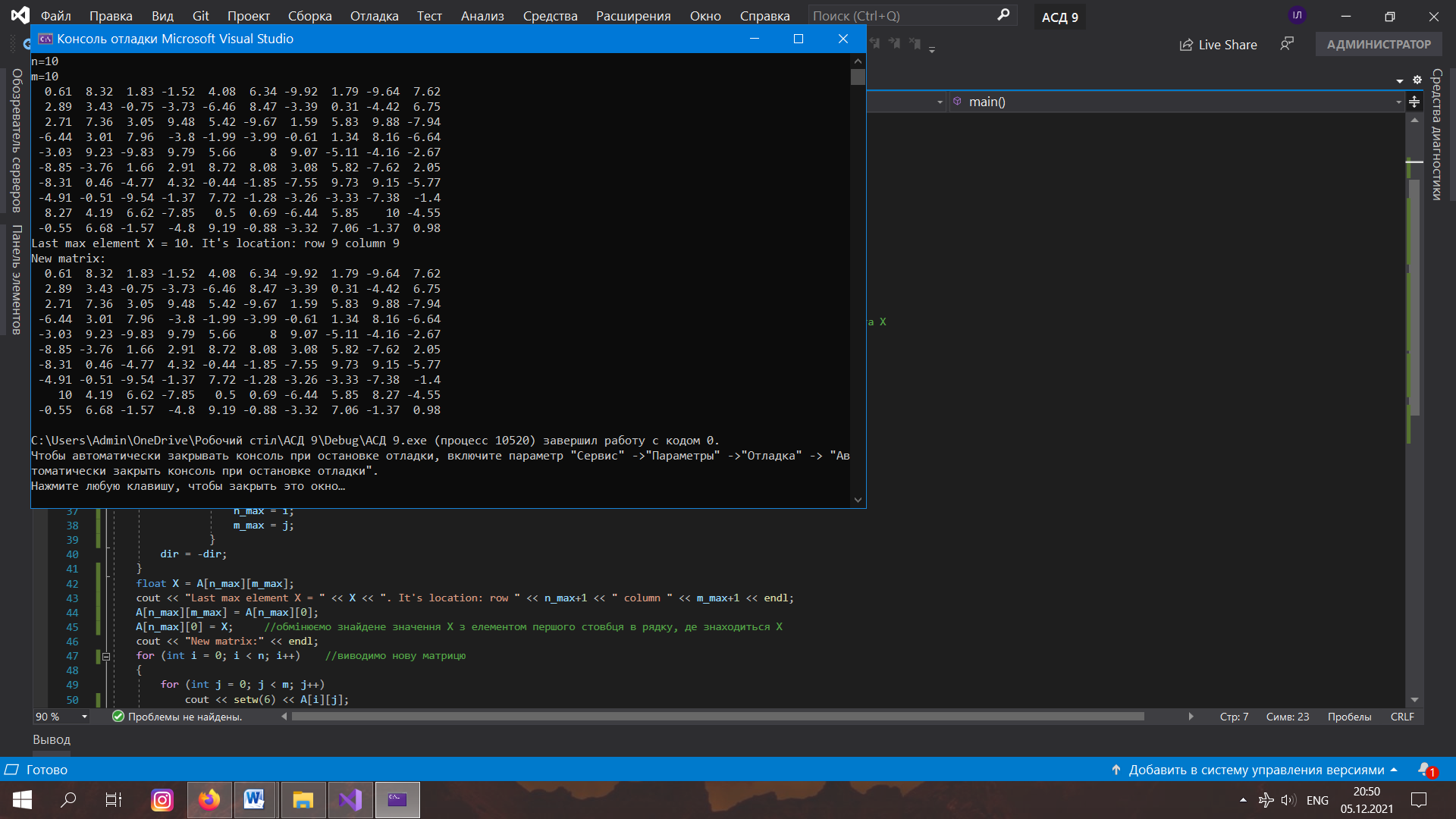
for (int i = 0; i < n; i++)

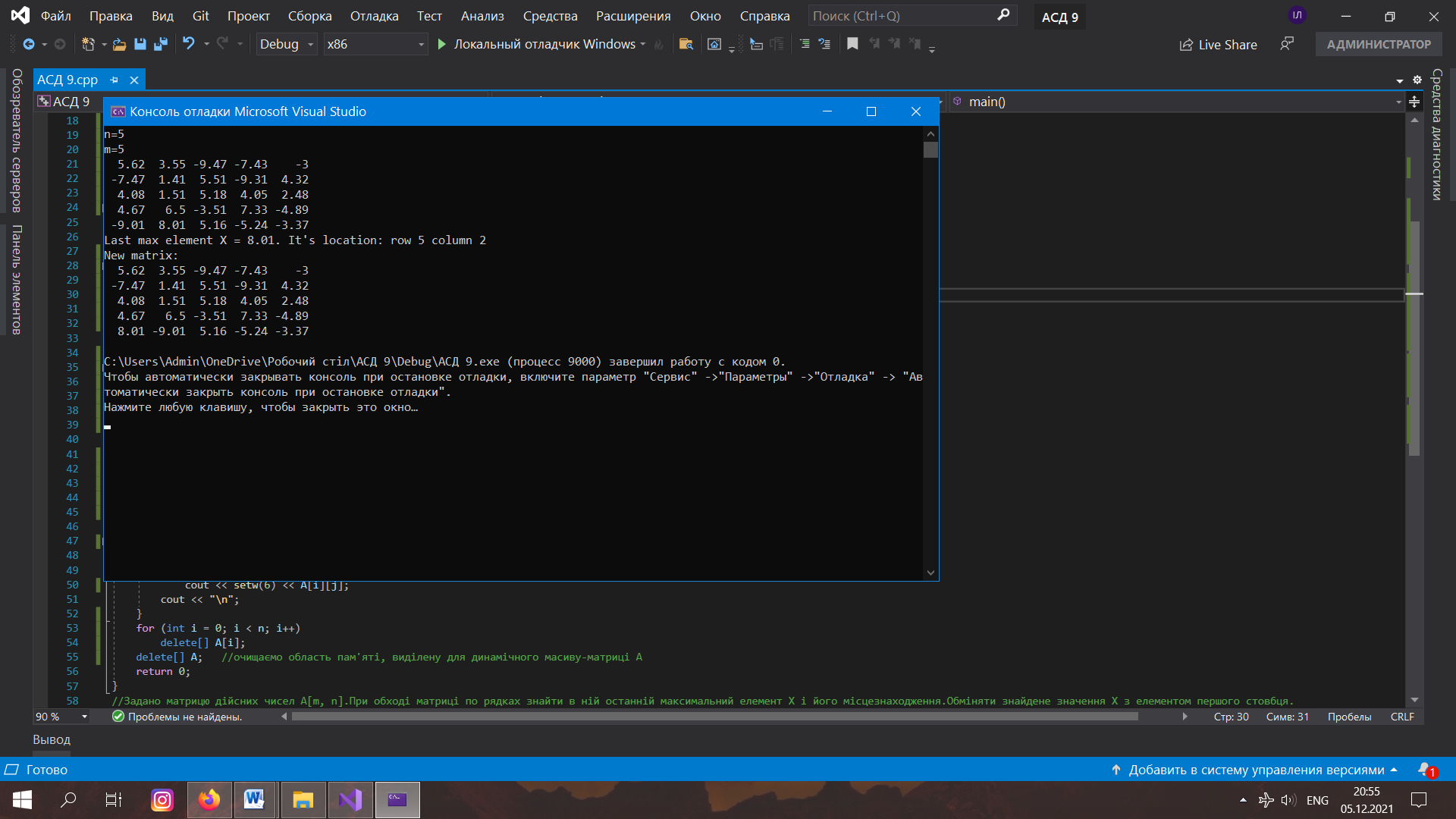
delete[] A[i];

delete[] A; //очищаємо область пам'яті, виділену для динамічного масиву-матриці А

return 0;

}





*Програма виводить спочатку випадково згенерований двовимірний масив розміру, вказаного користувачем, потім шуканий останній максимальний елемент Х, номер його рядка та стовпця. В кінці програма виводить нову матриця з поміняними місцями елементом Х та елементом першого стовпця з рядка, в якому знаходиться Х.*

**Висновок:**

На лабораторній роботі я дослідив алгоритм обходу матриці змійкою по рядках. В моїй задачі довелось також застосувати алгоритм пошуку індексів елемента в матриці за ключем «останній максимальний» і провести обмін значеннями певних елементів матриці. Особливістю є те, що при обході рядків зліва-направо і справа-наліво ключ порівняння дещо відрізняється. Весь алгоритм поділяється на 6 кроків, фактично його можна розділити на 4 основні етапи.