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 26

Виконав студент: ІП-15 Поліщук Валерій Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірила: Вєчерковська Анастасія Сергіївна

( прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота №9

**Дослідження алгоритмів обходу масивів**

**Варіант 26**

**Мета** – дослідити алгоритми обходу масивів, набути практичних

навичок використання цих алгоритмів під час складання програмних

специфікацій.

**Постановка задачі**

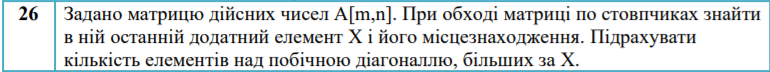
Розробити алгоритм та написати програму, яка складається з наступних

дій:

1. Опису змінної індексованого типу згідно з варіантом

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

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



**Математична модель**

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім'я** | **Призначення** |
|  |  |  |  |
| Двовимірний масив | Дійсний | a | Проміжні дані |
| Значення j | Цілий | j | Проміжні дані |
| Значення і | Цілий | і | Проміжні дані |
| Значення x | Дійсний | x | Проміжні дані |
| Значення m | Цілий | m | Проміжні дані |
| Значення n | Цілий | n | Проміжні дані |
| Значення x\_i | Цілий | x\_i | Вихідні дані |
| Значення x\_j | Цілий | x\_j | Вихідні дані |
| Значення dir | Цілий | dir | Проміжні дані |
| Значення count | Цілий | count | Вихідні дані |
| Функція, що обчислює кількість елементів, що менші за х, над побічною діагоналлю | Функція | Search\_Count | Вихідні дані |
| Функція, що знаходить останній додатній елемент при обході стовпцями | Функція | SearchMax | Вихідні дані |
| Функція, що заповнює матрицю | Функція | Create\_Matrix | Проміжні дані |

Random(a,b) – повертає випадкове дійсне число в проміжку від а до b

Вивести: вивід у консоль

Ми створюємо матрицю потрібної розмірності, заповнюємо її за допомогою функції Create\_Matrix з використанням функції Random(), потім обходом стовпцями знаходимо останній додатній елемент х та його розташування, після чого знаходимо кількість елементів масиву над побічною діагоналлю, що більші за х.

**Розв’язання**

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо процес заповнення матриці випадковими значеннями.

Крок 3. Деталізуємо процес знаходження останнього додатного елементу матриці х за допомогою обходу стовпцями.

Крок 3. Деталізуємо процес знаходження кількості елементів матриці над побічною діагоналлю, що більші за х.

**Псевдокод**

**Основна програма**

*Крок 1*

**початок**

doublea[m][n]

Create\_Matrix(a,m,n)

x = Last\_Positive(a, m, n)

SearchCount(a, m, n, x)

**кінець**

**Підпрограма**

SearchCount (a, m, n, x)

**початок**

count = 0

**повторити**

**для i від 0 до m-1**

**повторити**

**для j від 0 до n-1**

якщо ((i + j - 1)< n-2) && a[i, j] > x

то

count++

все якщо

**все повторити**

**все повторити**

**вивести :** "Кількість єлементів над побічною діагоналлю що більші за Х - " +count

**кінець**

Create\_Matrix(a,m,n)

**Початок**

**повторити**

**для i від 0 до m-1**

**повторити**

**для j від 0 до n-1**

a[i][j] := Random(-10,10)

**все повторити**

**все повторити**

**Кінець**

Last\_Positive(a,m,n)

**початок**

dir = -1

# повторити

**для** j від 0 до n-1

**якщо** dir < 0

# то

**повторити**

**для** i від 0 до m-1

якщо a[i,j]>0

то

x = a[i, j]

x\_i = i

x\_j = j

все якщо

# все повторити

dir = -dir

# інакше

**повторити**

**для** i від m-1 до 0 включно

якщо a[i,j]>0

то

x = a[i, j]

x\_i = i

x\_j = j

все якщо

# все повторити

dir = -dir

# все якщо

# все повторити

# вивести : "x = " + x

# вивести : "рядок № " + (x\_i + 1) вивести : " стовпець № " + (x\_j + 1)

# повернути x

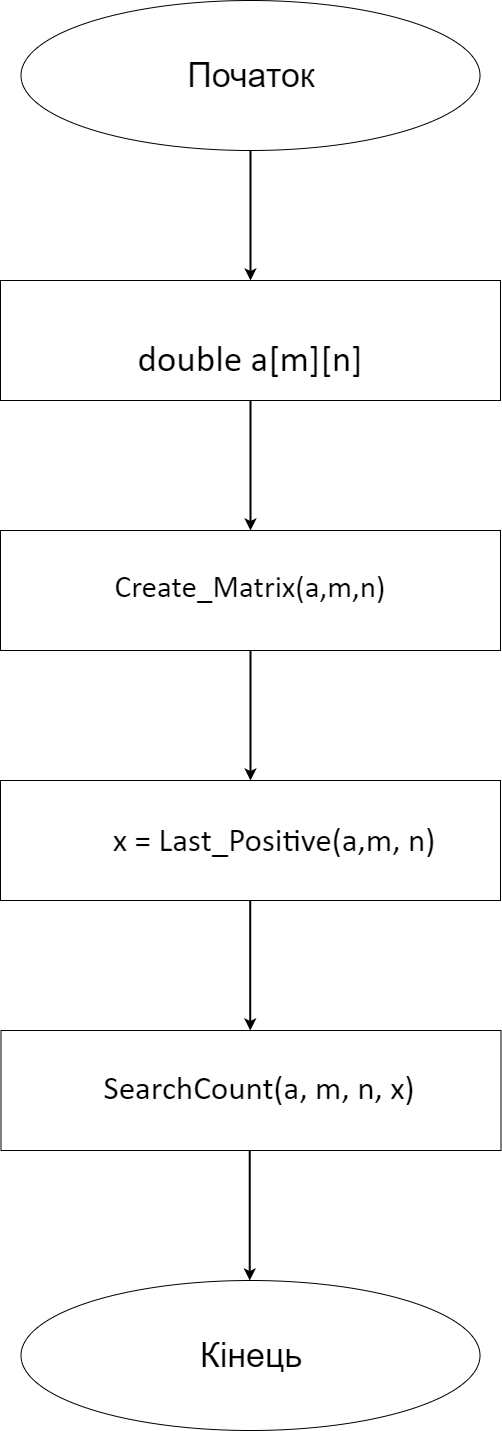
**Кінець**

**кінець**

**Блок-схема**

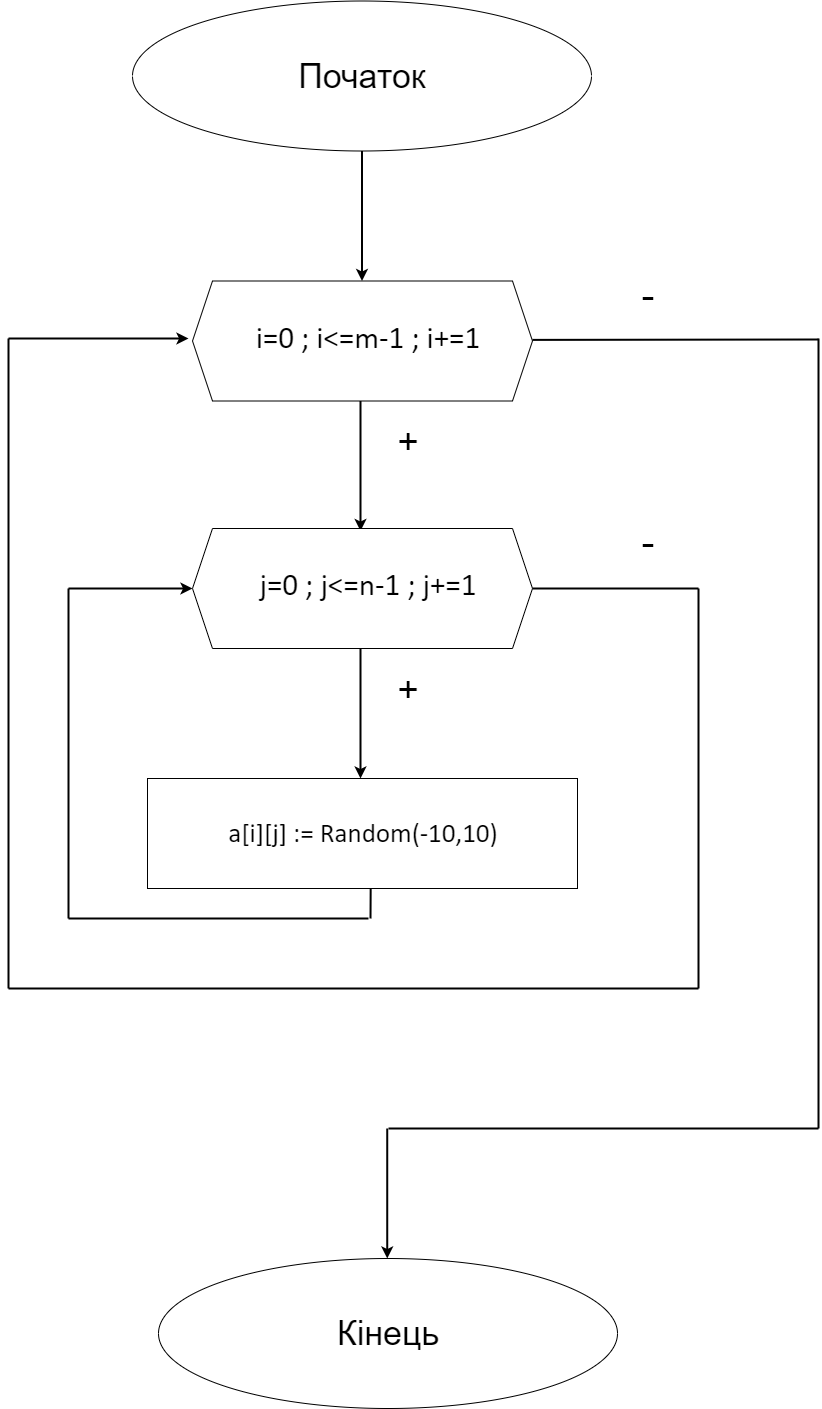
**Основна програма**

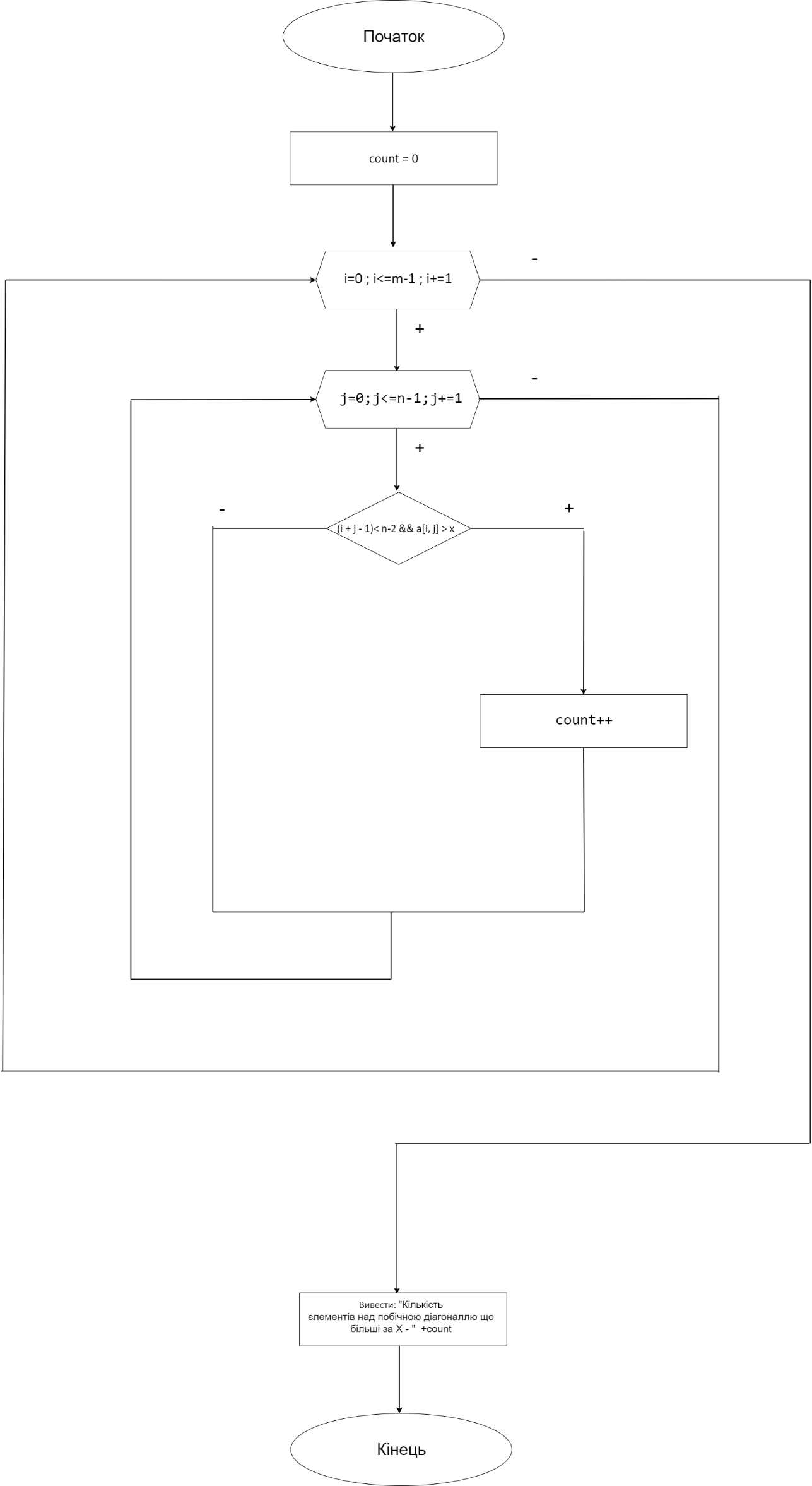
*Крок 1*



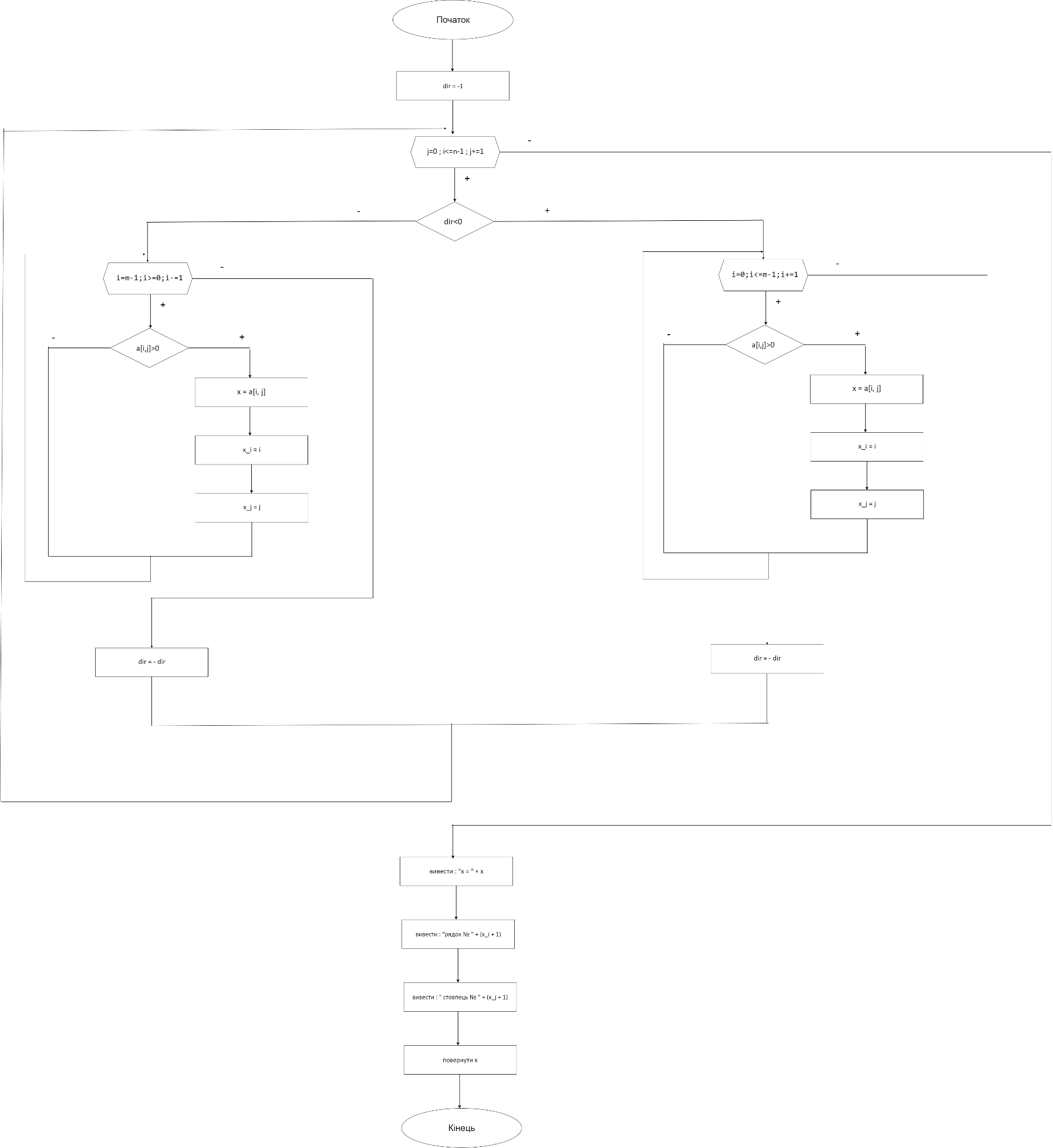
Create\_Matrix(a,m,n)

*Підпрограми:*

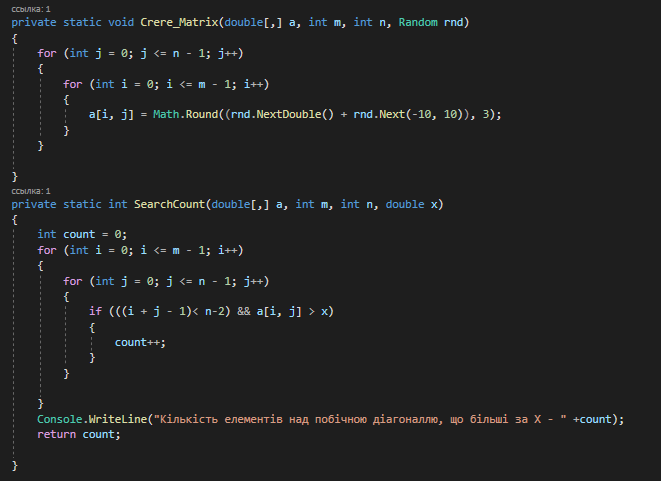
****

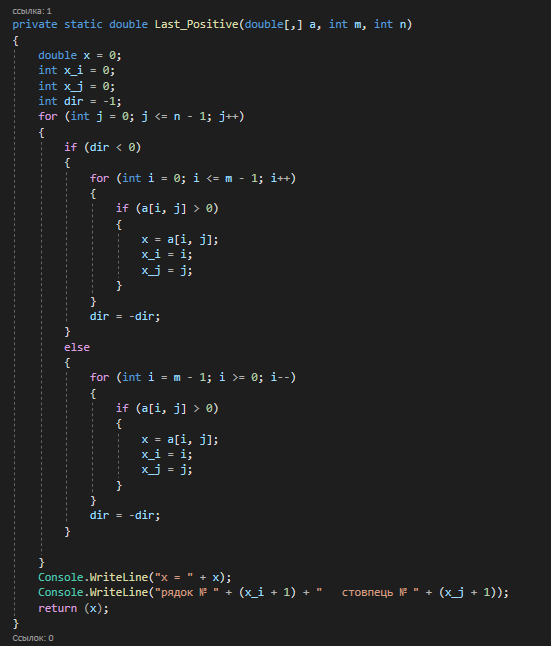
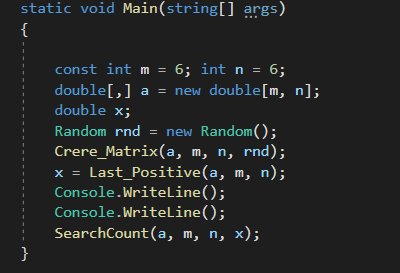
****

SearchCount (a, m, n, x)

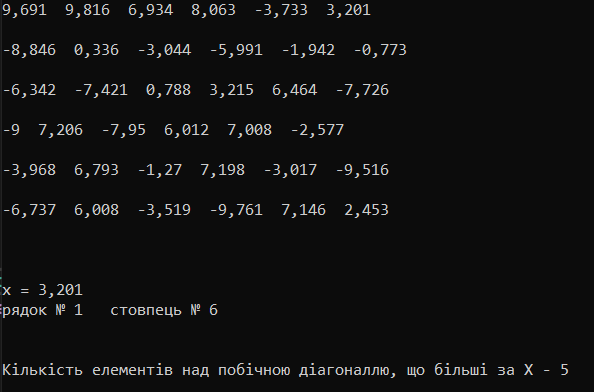
****

Last\_Positive(a,m,n)

**Код програми**



**Випробування алгоритму**



**Висновки**

Я дослідив алгоритми обходу масивів, набув практичних

навичок використання цих алгоритмів під час складання програмних

специфікацій.