

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 № 8 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження алгоритмів пошуку та сортування»

Варіант 26

Виконав студент: ІП-15 Поліщук Валерій Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірила: Вечерковська Анастасія Сергіївна

( прізвище, ім'я, по батькові)

Київ 2021

## Лабораторна робота №8

### Дослідження алгоритмів пошуку та сортування

#### Варіант 26

**Мета** – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

#### Постановка задачі

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом.
2. Ініціювання змінної, що описана в п.1 даного завдання.
3. Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями, що обчислюються згідно з варіантом .

26	6 x 5	Дійсний	Із середнього арифметичного від'ємних значень елементів рядків двовимірного масиву. Відсортувати методом бульбашки за спаданням.
----	-------	---------	--

#### Математична модель

Змінна	Тип	Ім'я	Призначення
Двовимірний масив	Дійсний	a	Проміжні дані
Одновимірний масив	Дійсний	b	Проміжні дані, Вихідні дані
Значення j	Цілий	j	Проміжні дані
Значення i	Цілий	i	Проміжні дані
Значення tmp	Дійсний	tmp	Проміжні дані

Значення sum	Дійсний	sum	Проміжні дані
Значення count	Цілий	count	Проміжні дані
Функція, що сортує одновимірний масив методом бульбашки	Функція	Sort_Array	Вихідні дані
Функція, що заповнює одновимірний масив	Функція	Create_Array	Проміжні дані
Функція, що заповнює матрицю	Функція	Create_Matrix	Проміжні дані

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

Ми створюємо матрицю потрібної розмірності, заповнюємо її за допомогою функції Create\_Matrix з використанням функції Random(), потім створюємо одновимірний масив та заповнюємо його за допомогою функції Create\_Array. Далі сортуємо методом бульбашки та виводимо його за допомогою функції Sort\_Array.

### 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 заповнення одновимірного масиву.

Крок 3. Деталізуємо процес сортування одновимірного масиву методом бульбашки та його виводу.

## Псевдокод

### Основна програма

*Крок 1*

**початок**

```
double a[6][5], b[6]
Create_Matrix(a)
Create_Array(a, b)
Sort_Array(b)
```

**кінець**

#### Підпрограма

Create\_Matrix(a)

**Початок**

**повторити**

**для і від 0 до 5**

**повторити**

**для j від 0 до 4**

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

**все повторити**

**все повторити**

**Кінець**

Create\_Array(a, b)

**початок**

sum = 0

count = 0

**повторити**

**для і від 0 до 5**

**повторити**

**для j від 0 до 4**

якщо a[i,j]<0

то

sum += a[i,j]

count++

все якщо

**все повторити**

b[i] = sum / count

sum = 0

count = 0

**все повторити**

**кінець**

Sort\_Array(b)

**початок**

tmp = 0

**повторити**

**для i від 0 до 5**

**повторити**

**для j від 0 до 4 - i**

якщо  $b[j] < b[j + 1]$

то

tmp =  $b[j]$

$b[j] = b[j + 1]$

$b[j + 1] = \text{tmp}$

все якщо

**все повторити**

**все повторити**

**повторити**

**для i від 0 до 5**

Вивести:  $b[i]$  + " "

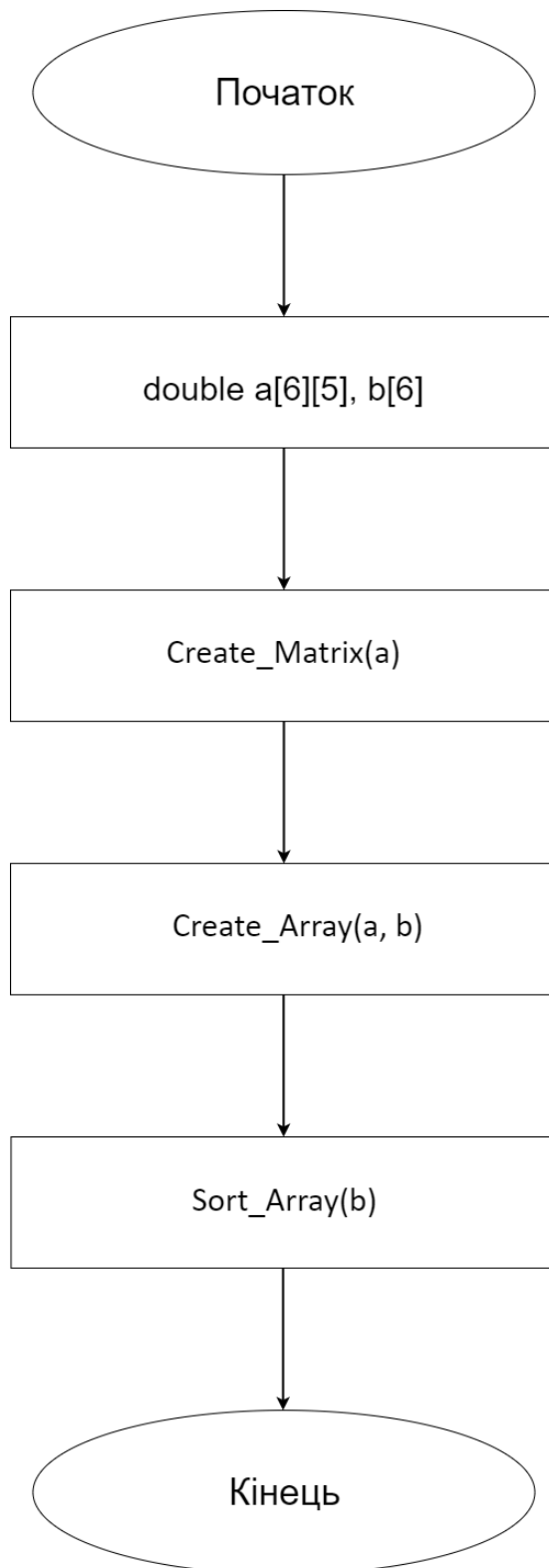
**все повторити**

**Кінець**

# Блок-схема

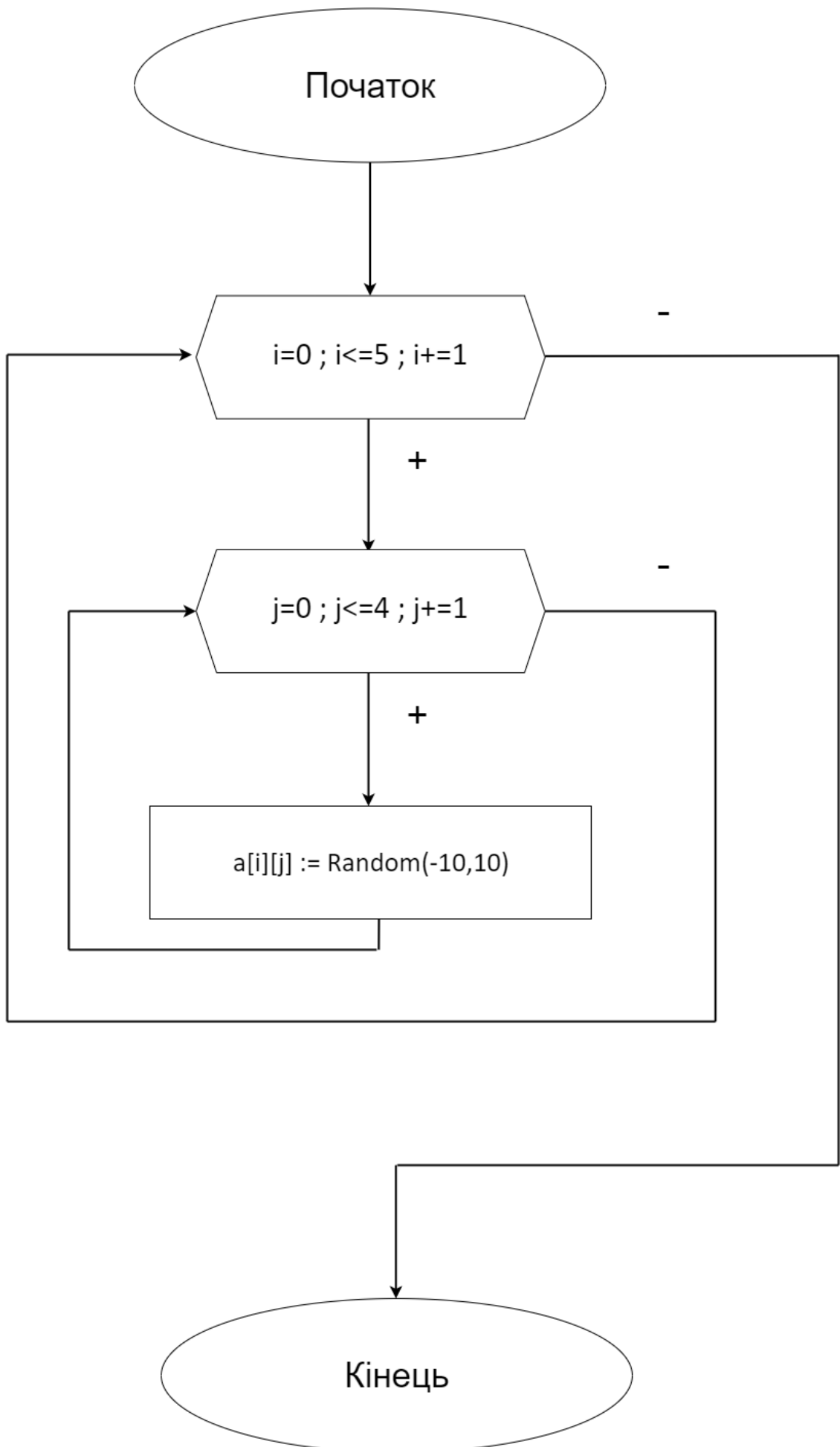
## Основна програма

*Крок 1*

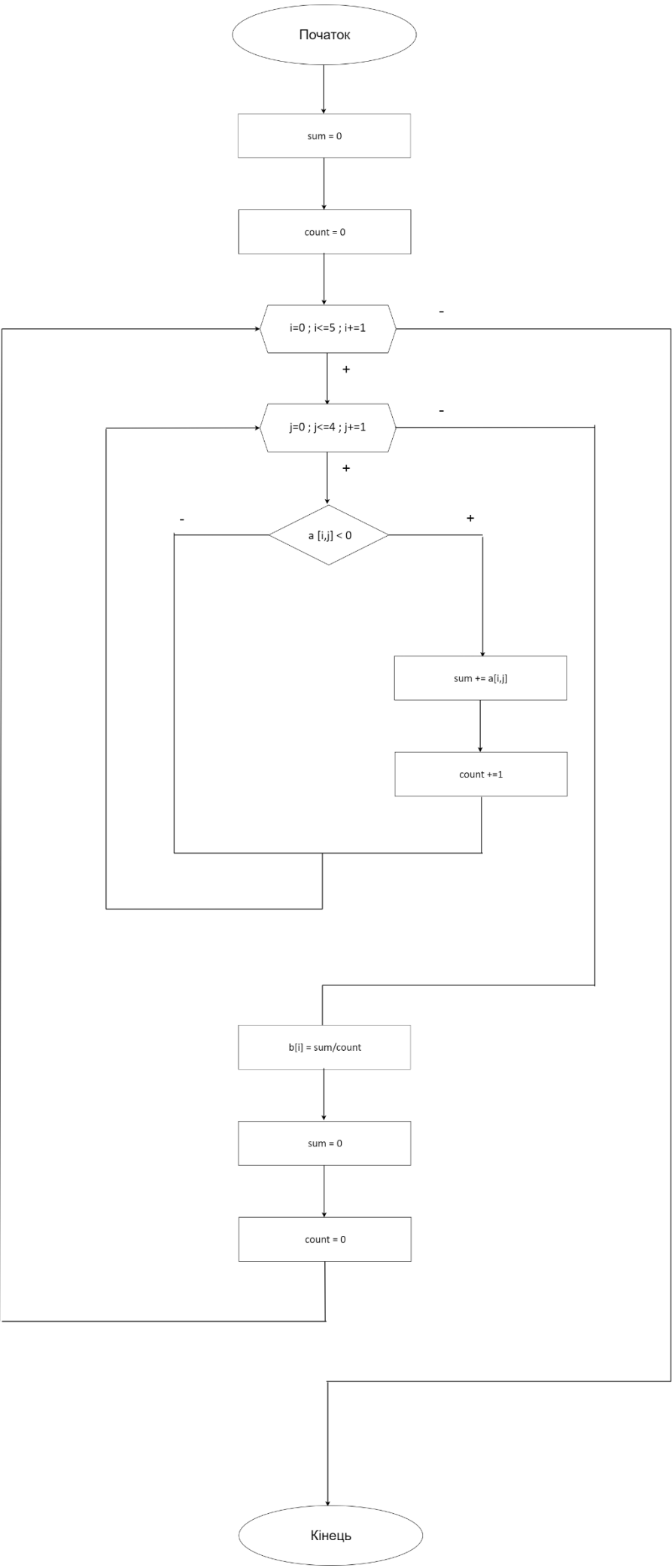


*Підпрограми:*

Create\_Matrix(a)

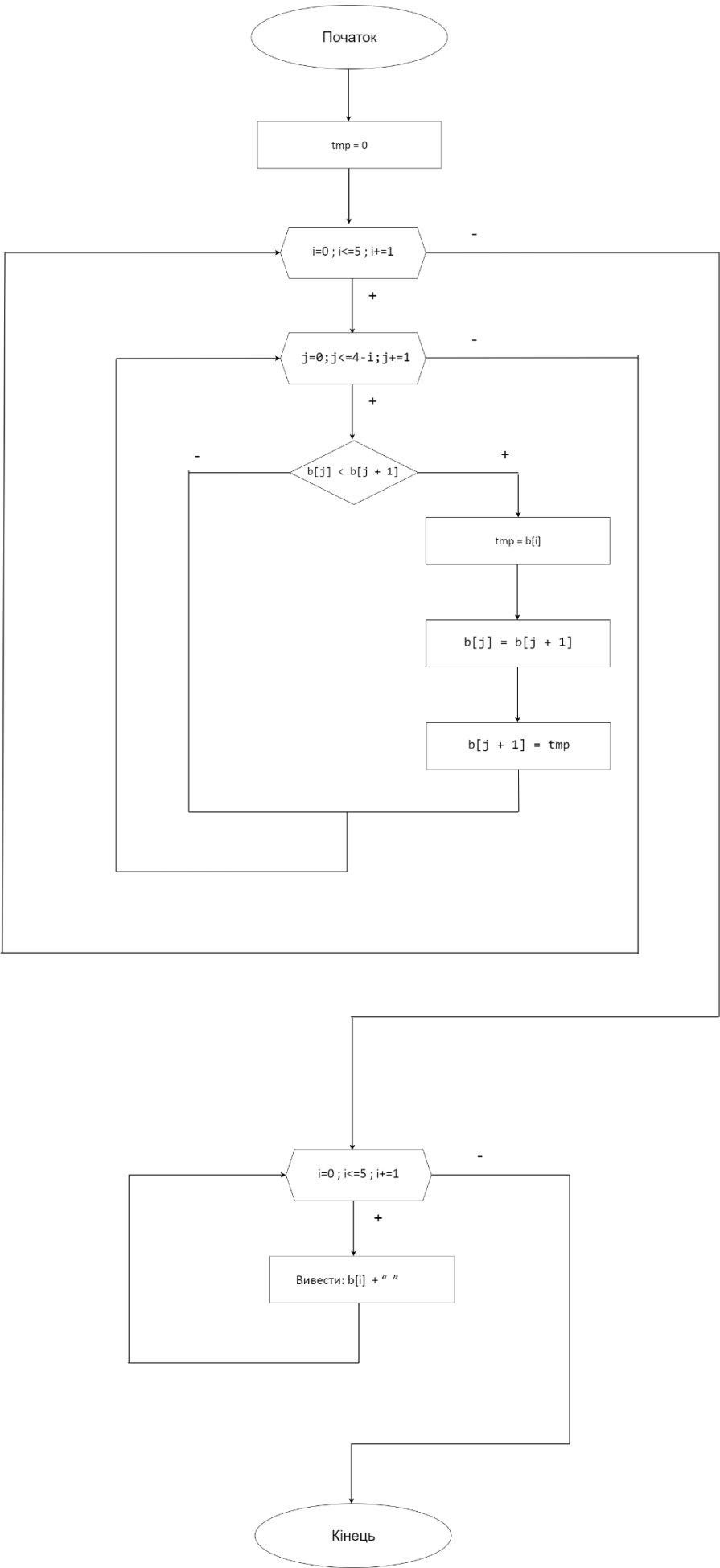


Create\_Array(a, b)





Sort\_Array(b)



## Код програми

ссылка: 1

```
private static void Create_Matrix(double[,] a, Random rnd)
{
    for (int i = 0; i <= 5; i++)
    {
        for (int j = 0; j <= 4; j++)
        {
            a[i, j] = Math.Round(rnd.Next(-10, 10) + rnd.NextDouble(), 4);
        }
    }
}
```

ссылка: 1

```
private static void Create_Array(double[,] a, double[] b)
{
    double sum = 0;
    int count = 0;
    for (int i = 0; i <= 5; i++)
    {
        for (int j = 0; j <= 4; j++)
        {
            if (a[i, j] < 0)
            {
                sum += a[i, j];
                count++;
            }
        }
        b[i] = Math.Round(sum / count, 4);
        sum = 0;
        count = 0;
    }
}
```

ссылка: 1

```
private static void Sort_Array(double[] b)
{
    Console.WriteLine("Sort Array: ");
    Console.WriteLine();
    double tmp = 0;
    for (int i = 0; i < 6; i++)
    {
        for (int j = 0; j < 5 - i; j++)
        {
            if (b[j] < b[j + 1])
            {
                tmp = b[j];
                b[j] = b[j + 1];
                b[j + 1] = tmp;
            }
        }
    }
    for (int i = 0; i <= 5; i++)
    {
        Console.Write(b[i] + "    ");
    }
}
```

ссылка: 0

```
static void Main(string[] args)
{
    double[,] a = new double[6, 5];
    double[] b = new double[6];
    Random rnd = new Random();
    Create_Matrix(a, rnd);
    Create_Array(a, b);
    Sort_Array(b);
}
```

## Випробування алгоритму

Matrix:

3,2181	-2,7853	6,1893	-9,5262	0,321
8,8365	5,1595	-5,0723	2,6697	2,8748
-0,7975	-5,0693	-8,1823	5,5662	-0,767
-1,4105	9,8828	-9,0651	-5,7178	0,0542
3,7348	-8,5005	4,8629	-4,3684	8,8201
1,2118	5,2975	-0,5097	9,4672	-7,0028

Array:

-6,1557	-5,0723	-3,704	-5,3978	-6,4344	-3,7562
---------	---------	--------	---------	---------	---------

Sort Array:

-3,704	-3,7562	-5,0723	-5,3978	-6,1557	-6,4344
--------	---------	---------	---------	---------	---------

## **Висновки**

Я дослідив алгоритми пошуку та сортування, набув практичних навичок використання цих алгоритмів під час складання програмних специфікацій.