

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 № 7 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійного пошуку в послідовностях»

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 №7

### Дослідження лінійного пошуку в послідовностях

#### Варіант 26

**Мета** – дослідити методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій.

#### Постановка задачі

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

1. Опису трьох змінних індексованого типу з 10 символних значень.
2. Ініціювання двох змінних виразами згідно з варіантом (табл. 1).
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної згідно з варіантом.

26	$2 * i + 42$	$54 - 2 * i$	Елементи, які менші за максимальний код
----	--------------	--------------	-----------------------------------------

#### Математична модель

Змінна	Тип	Ім'я	Призначення
Перший масив	Символьний	a	Проміжні дані
Другий масив	Символьний	b	Проміжні дані
Третій масив	Символьний	c	Проміжні дані
Значення i	Цілий	i	Проміжні дані
Функція, що створює масиви a та b	Функція	Create_a_b	Проміжні дані
Функція, що створює масив c	Функція	Create_c	Проміжні дані

Функція, що виводить результат	Функція	Resultat	Вихідні дані
Значення j	Цілий	j	Проміжні дані
Значення max	Цілий	max	Проміжні дані
Значення len	Цілий	len	Проміжні дані

За допомогою функції Create\_a\_b заповнюємо створені до цього масиви a та b, за допомогою функції Create\_c заповнюємо створений до цього масив c, а далі за допомогою функції Resultat виводимо результат.

### Розв'язання

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо процес створення масивів a та b

Крок 3. Деталізуємо процес створення масиву c

Крок 3. Деталізуємо процес виведення результату

### Псевдокод

#### Основна програма

*Крок 1*

**початок**

```
Create_a_b(a, b);
Create_c(a, b, c);
```

Resultat(c);

**кінець**

### Підпрограма

Create\_a\_b(a, b)

**початок**

**повторити**

**для i від 0 до 9**

$a[i] = 2 * i + 42$

$b[i] = 54 - 2 * i$

**все повторити**

**кінець**

Create\_c(a, b, c)

**початок**

$len = 0$

**повторити**

**для i від 0 до 9**

**повторити**

**для j від 0 до 9**

**якщо**  $a[i] == b[j]$

**то**

$c[len] = a[i]$

$len++$

**все якщо**

**все повторити**

**все повторити**

**кінець**

Resultat(c)

**початок**

max = -1

**повторити**

**для i від 0 до 9**

**якщо**  $c[i] > \text{max}$

**то**

max =  $c[i]$

**все якщо**

**все повторити**

**повторити**

**для i від 0 до 9**

**якщо**  $c[i] < \text{max}$

**то**

Вивести:  $c[i]$

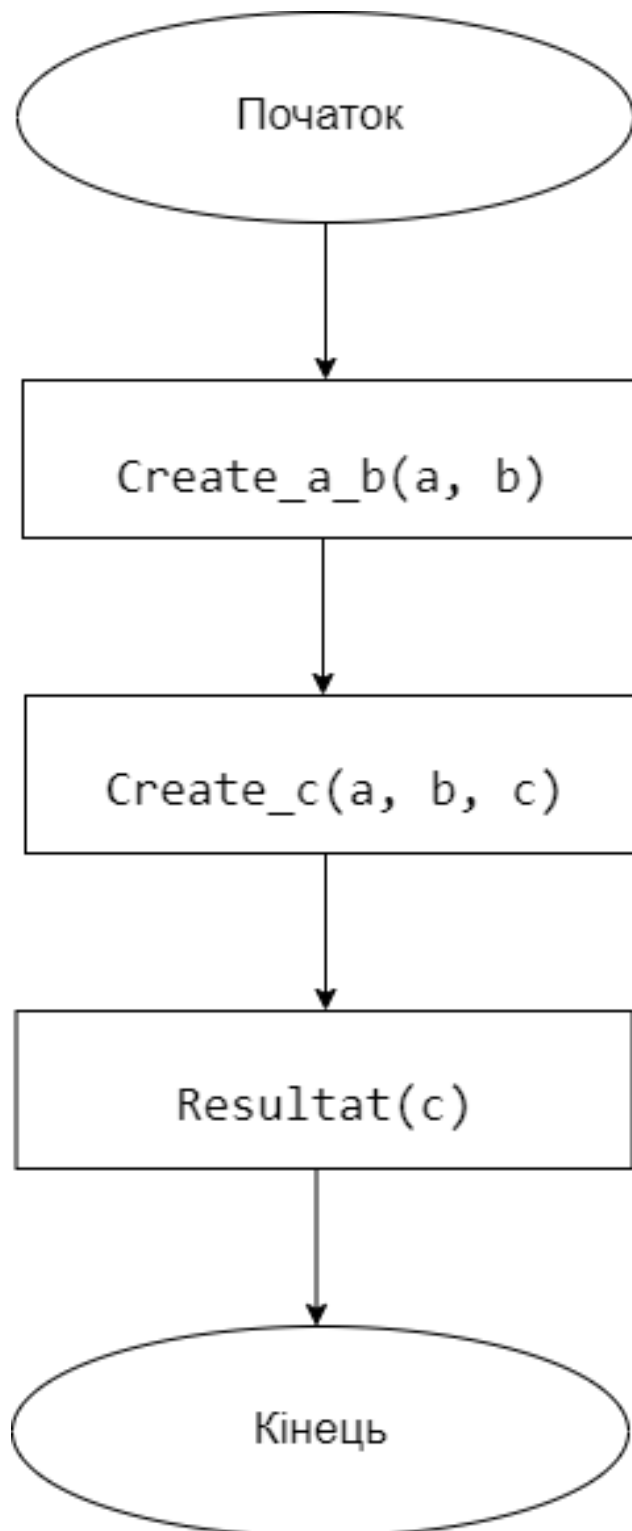
**все якщо**

**все повторити**

**кінець**

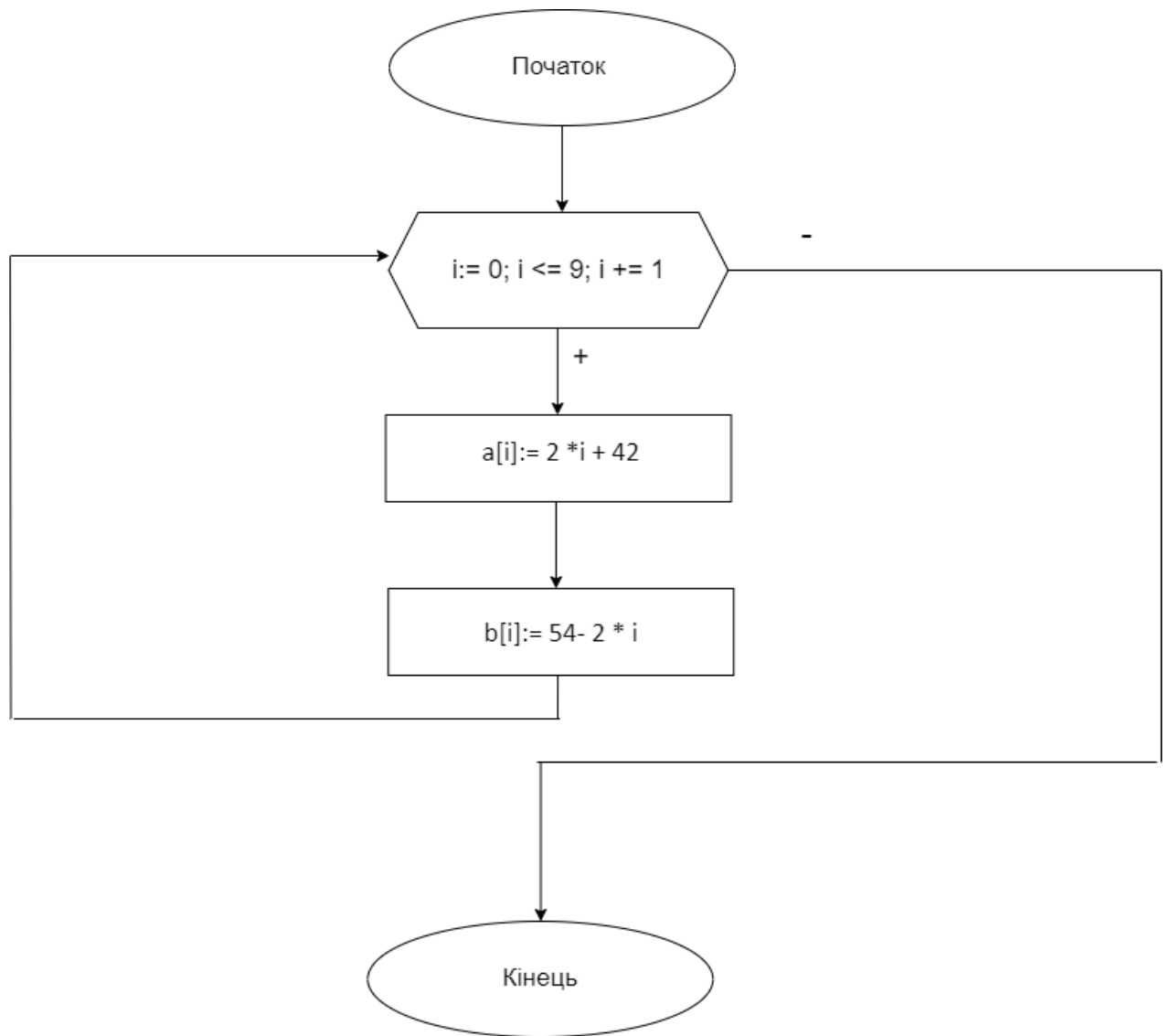
**Блок-схема**  
**Основна програма**

*Крок 1*

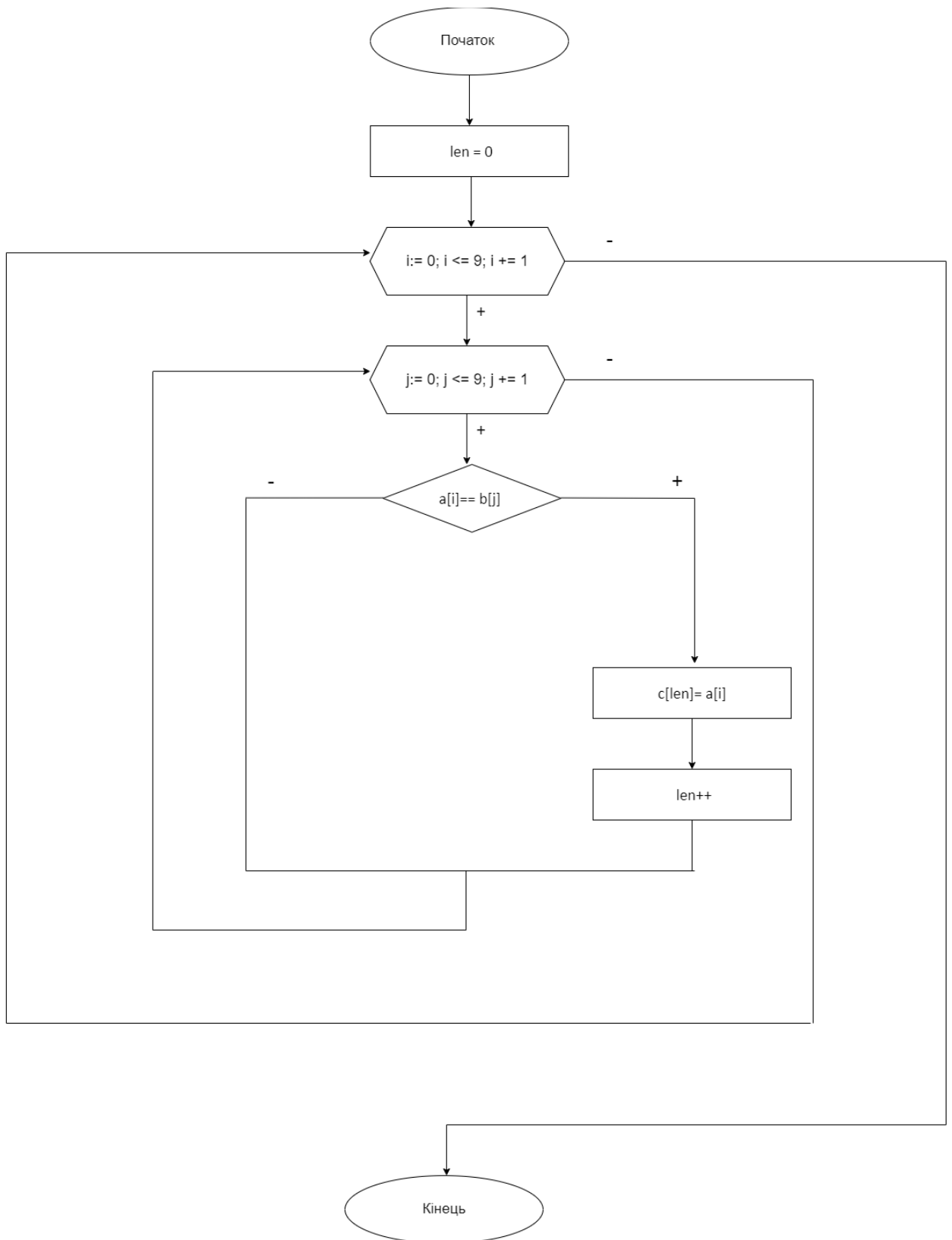


Підпрограми:

Create\_a\_b(a, b)

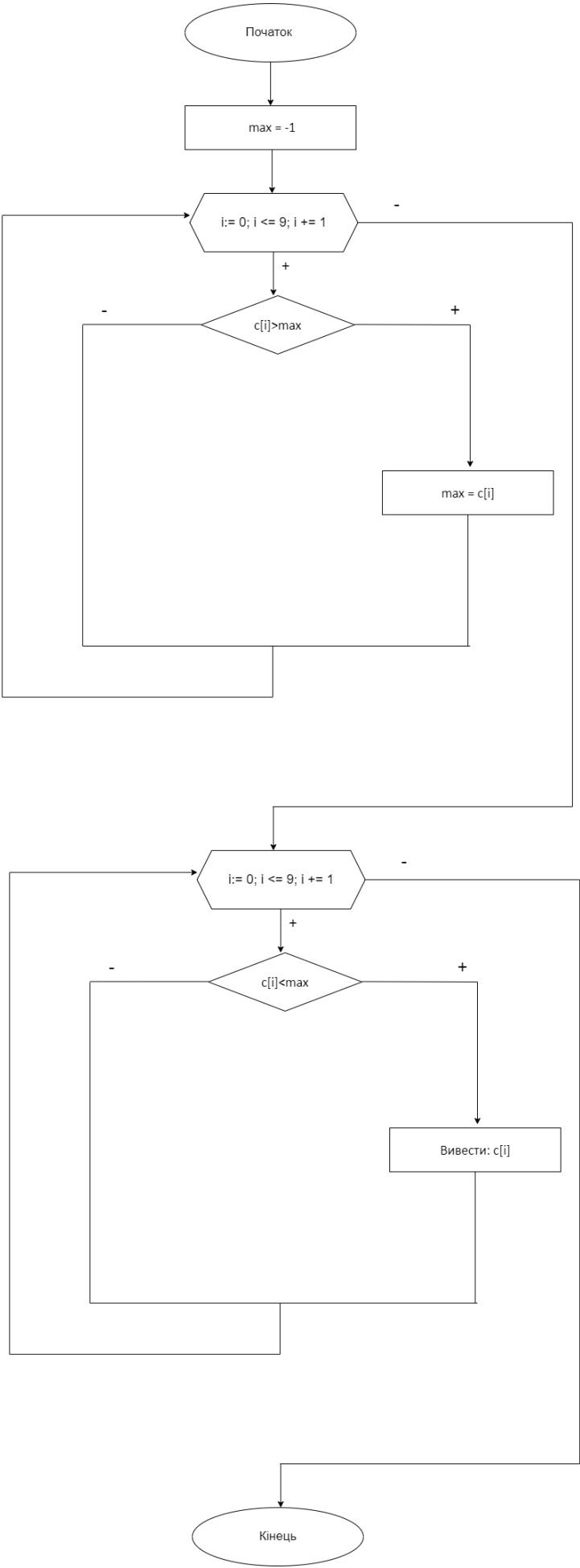


Create\_c(a, b, c)





Resultat(c)



## Код програми

```
static void Main(string[] args)
{
    char[] a = new char [10];
    char[] b = new char[10];
    char[] c = new char[10];
    Create_a_b(a, b);
    Create_c(a, b, c);
    Resultat(c);
}
```

```
static void Create_a_b (char[] a, char[] b)
{
    for (int i=0; i<=9; i++)
    {
        a[i] = Convert.ToChar(2 * i + 42);
        b[i] = Convert.ToChar(54 - 2 * i);
    }
}
```

ссылка: 1

```
static void Create_c (char[] a, char[] b, char[] c)
{
    int len = 0;
    for(int i = 0; i <= 9; i++)
    {
        for (int j = 0; j <= 9; j++)
        {
            if (a[i] == b[j])
            {
                c[len] = a[i];
                len++;
            }
        }
    }
}
```

ссылка: 1

```
static void Resultat (char[] c)
{
    int max = -1;
    for (int i = 0; i <= 9; i++)
    {
        if (Convert.ToInt32(c[i])>max)
        {
            max = Convert.ToInt32(c[i]);
        }
    }
    for (int i = 0; i <= 9; i++)
    {
        if (Convert.ToInt32(c[i])< max)
        {
            Console.WriteLine(Convert.ToString(c[i]));
        }
    }
}
```

## Випробування алгоритму

RESULT:

\*  
,  
.  
0  
2  
4

A:

*	42
,	44
.	46
0	48
2	50
4	52
6	54
8	56
:	58
<	60

B:

6	54
4	52
2	50
0	48
.	46
,	44
*	42
(	40
&	38
\$	36

C:

*	42
,	44
.	46
0	48
2	50
4	52
6	54
	0
	0
	0

## **Висновки**

Я дослідив методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набув практичних навичок їх використання під час складання програмних специфікацій.