Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної инженериії

Звіт

з лабораторної работи №7 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійного пошуку в послідовностях»

Варіант 1

Виконав студент ІП-14 Аджигельдієва Мадіна Алімівна

Перевірів Мартинова Оксана Петрівна

Київ 2021

**Лабораторна робота 7**

**Дослідження лінійного пошуку в послідовностях**

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

Задача 1. Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису трьох змінних індексованого типу з 10 символьних значень.
2. Ініціювання двох змінних виразами.
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної.



**Розв’язання**

**Постановка задачі.** Результатом розв’язку є виведення суми двох мінімальних елементів.

**Побудова математичної моделі.**

|  |  |  |
| --- | --- | --- |
| **Змінна** | **Тип** | **Призначення** |
| LENGTH | Цілий | Постійна змінна для визначення розміру масива |
| findElement | Логічний | Функція, що повертає «брехня» або «істина» (ФХІ) |
| ai | Символьний | Ім'я параметра ФХІ |
| b[] |
| length |
| j | Цілий | Локальні змінні ФХІ |
| a[LENGTH], b[LENGTH], c[LENGTH] | Символьний | Одновимірний масив |
| k, i, first, second | Символьний | Локальні змінні в основній функції |
| i | Цілий |
| first + second | Символьний | Результат сума мінімальних елементів |

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

Визначимо основні дії:

*Крок 1.* Введення const LENGTH, i, k, first, second, func.

*Крок 2.* Ініціювання масивів a, b, c символьного типу.

*Крок 3.* Знаходження однакових елементів в масивах a та b.

*Крок 4.* Знаходження мінімальних елементів.

*Крок 5.* Виведення мінимальних елементів та їх суму.

**Псевдокод алгоритму.**

*Крок 1.*

**початок**

Введення const LENGTH, i, k, first, second, func

Ініціювання масивів a, b символьного типу

Знаходження однакових елементів в масивах a та b

Знаходження мінімальних елементів

Виведення мінимальних елементів та їх суму

**кінець**

*Крок 2.*

**початок**

Введення const LENGTH = 10, i, k = 0, first, second, func

**для** (i = 0; i < LENGTH; i++)

a[i] = 117 \* i;

b[i] = 127 – 2 \* i;

Знаходження однакових елементів в масивах a та b

Знаходження мінімальних елементів

Виведення мінимальних елементів та їх суму

**кінець**

*Крок 3.*

**початок**

Введення const LENGTH = 10, i, k = 0, first, second, func

**для** (i = 0; i < LENGTH; i++)

a[i] = 117 \* i;

b[i] = 127 – 2 \* i;

**для** (i = 0; i < LENGTH; i++)

func = findElement a[i];

**якщо** (func, b, LENGTH)

c[k] = a[i];

k++;

Знаходження мінімальних елементів

Виведення мінимальних елементів та їх суму

**кінець**

*Крок 4.*

**початок**

Введення const LENGTH = 10, i, k = 0, first, second, func

**для** (i = 0; i < LENGTH; i++)

a[i] = 117 \* i;

b[i] = 127 – 2 \* i;

**для** (i = 0; i < LENGTH; i++)

func = findElement a[i];

**якщо** (func, b, LENGTH)

c[k] = a[i];

k++;

**якщо** (k < 2)

**виведення** повідомлення «Масив *c* містить менше 2 елементів»

**для** (i = 0; i < k; i++)

**якщо** (c[i] < first)

second = first;

first = c[i];

**інакше якщо** (с[i] < second та c[i] != first)

second = c[i]

Виведення мінимальних елементів та їх суму

**кінець**

**Псевдокод функції.**

*Крок 1.*

**початок**

Введення j

Ініціювання масива

Повернення значення

**кінець**

*Крок 2.*

**початок**

Введення j

**для** (j = 0; j < length; j++)

**якщо** (ai == b[j])

**повернення** «істина»

Повернення значення

**кінець**

*Крок 3.*

**початок**

Введення j

**для** (j = 0; j < length; j++)

**якщо** (ai == b[j])

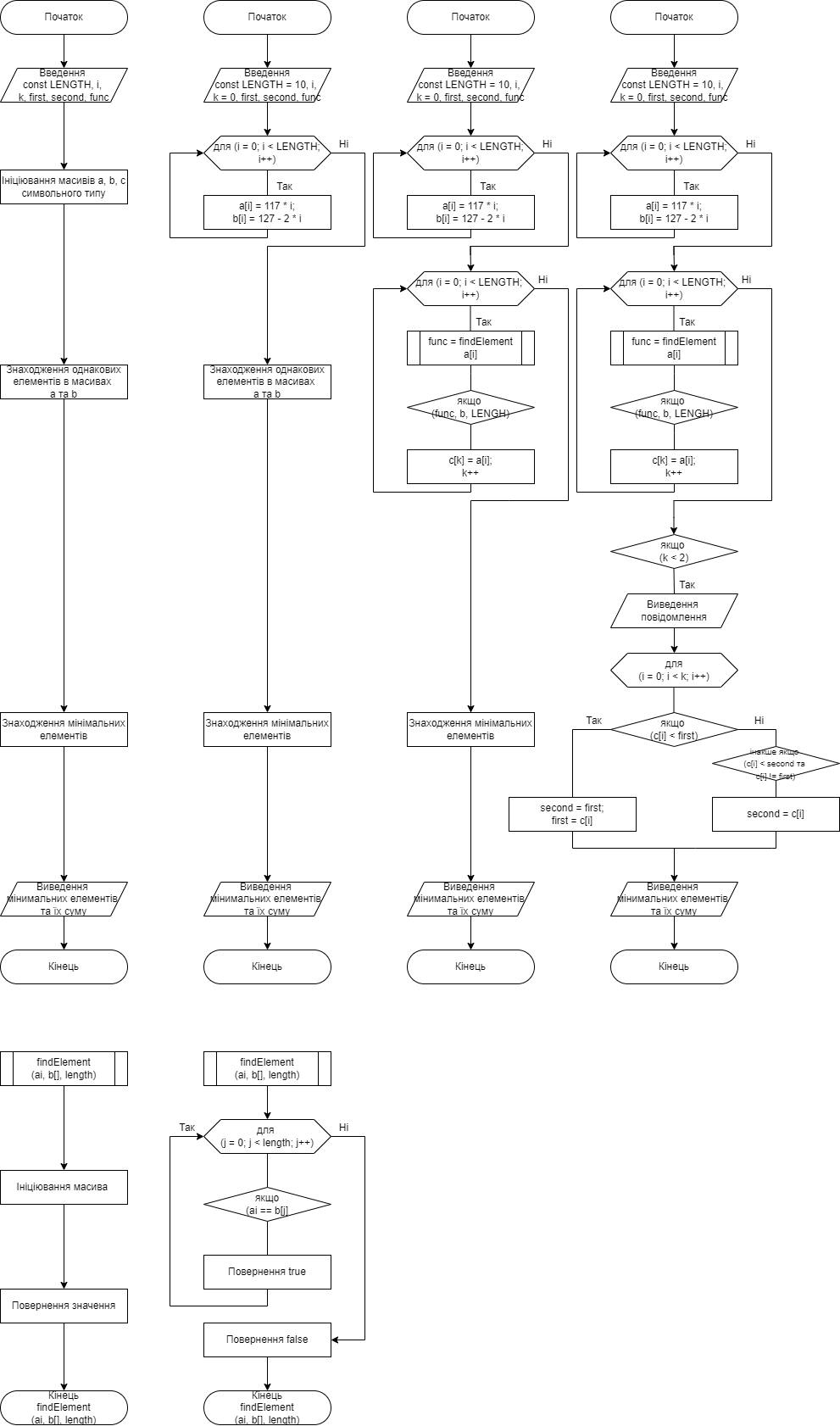
**повернення** «істина»

**повернення** «брехня»

**кінець**

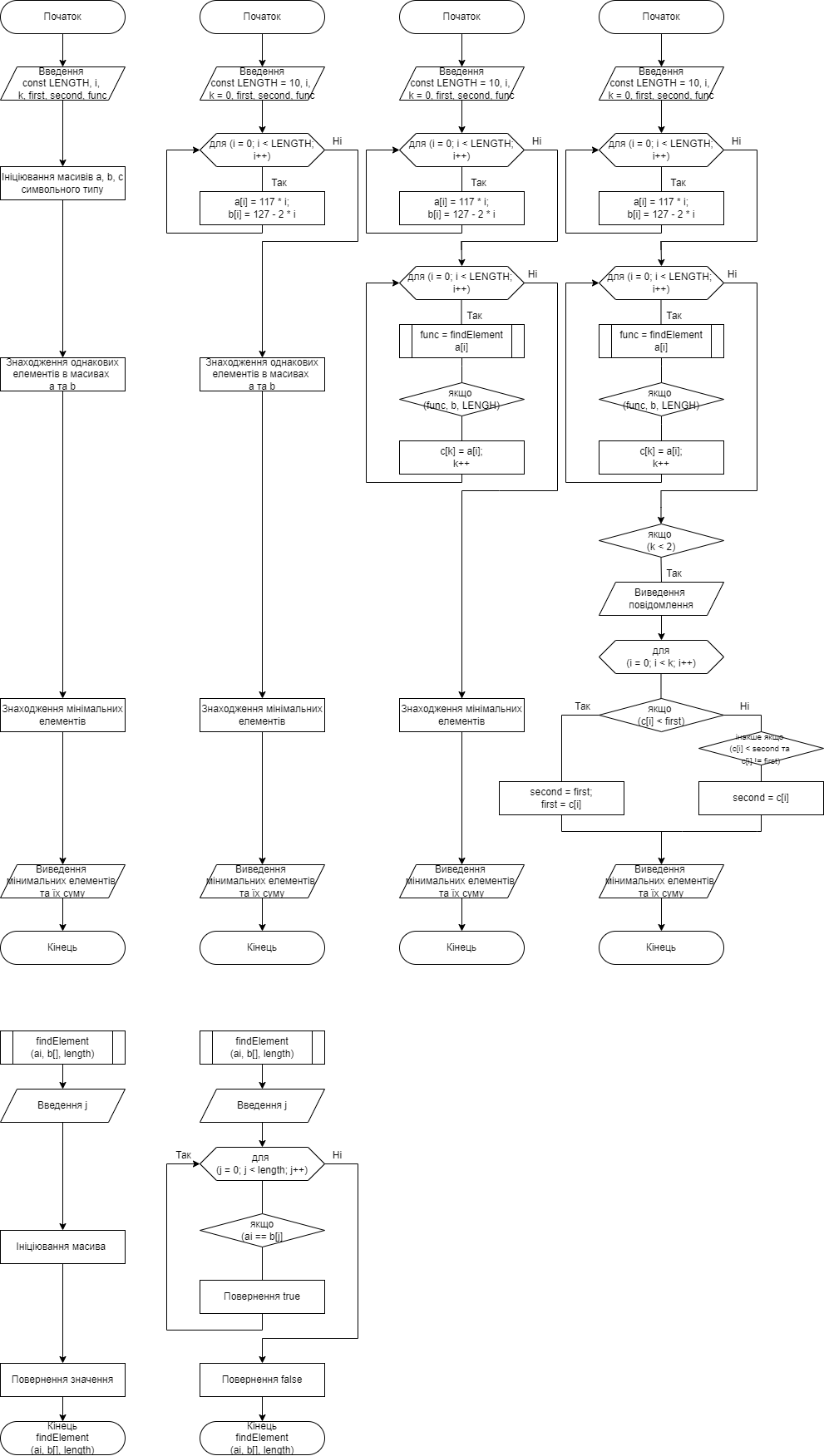
**Блок-схема алгоритму.**

*Крок 1 Крок 2 Крок 3 Крок 4*



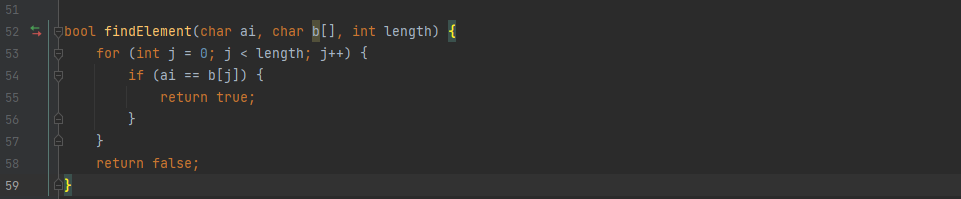
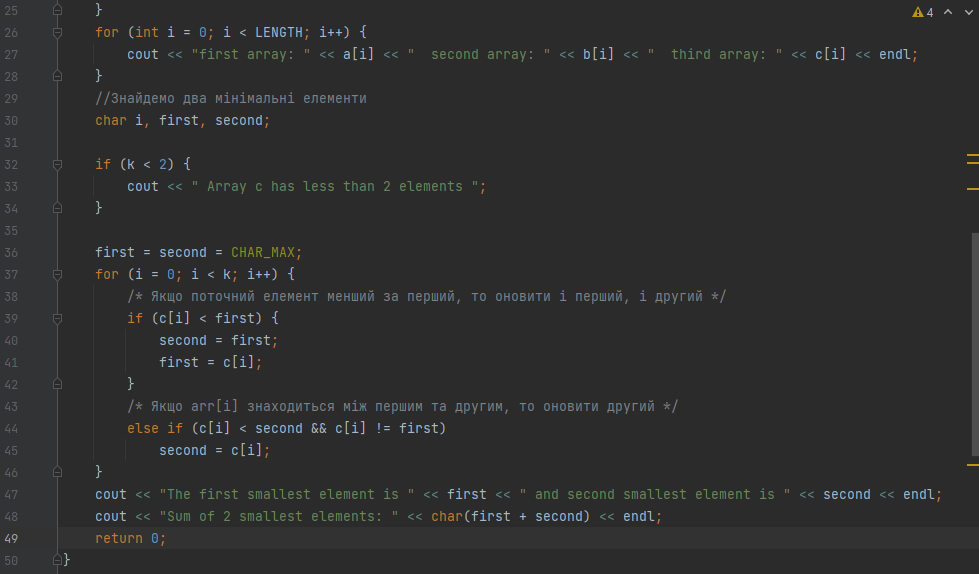
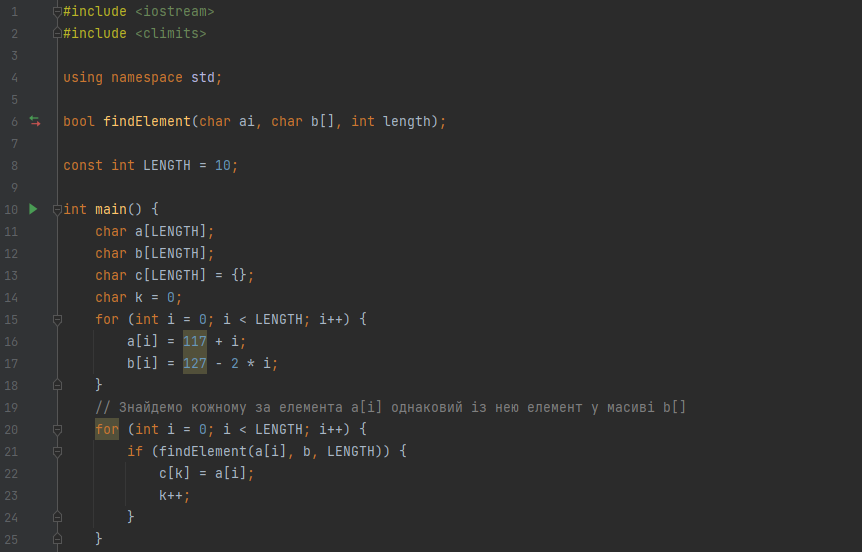
**Блок-схема функцій.**

*Крок1 Крок 2*

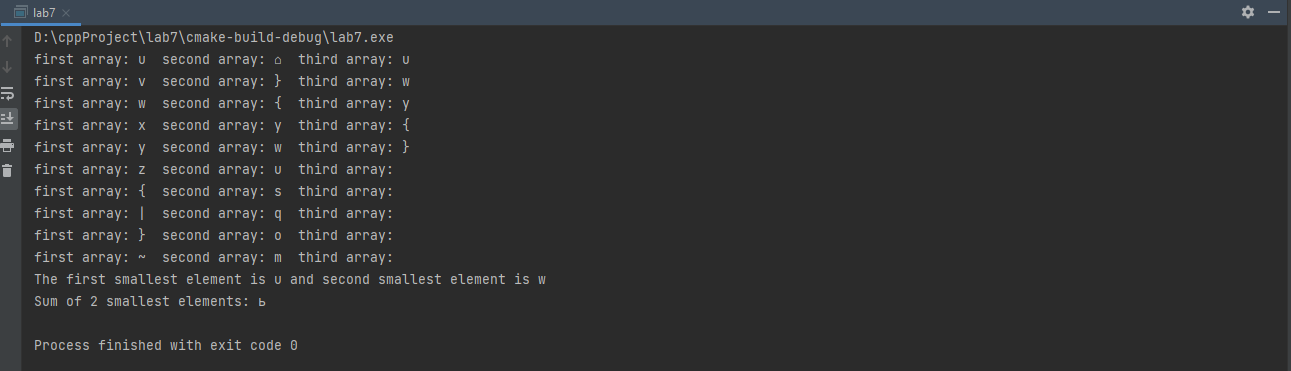


**Код програми.**

Представимо у вигляді коду програми та виведемо результат роботи.



**Результат коду програми.**



**Випробування алгоритму.**

Перевіримо правільність алгоритму на довільних конкретних значеннях початкових даних.

|  |  |
| --- | --- |
| **Блок** | **Дія** |
|  | **Початок** |
| 1 | **Введення** const LENGTH = 10, i, k = 0, first, second, func |
| 2 | **для** (i = 0; i < LENGTH; i++)  a[i] = 117 \* i;  b[i] = 127 – 2 \* i; |
| 3 | **для** (i = 0; i < LENGTH; i++)  func = findElement a[i];  **якщо** (func, b, LENGTH)  c[k] = a[i]  k++; |
| 4 | **якщо** (k < 2)  **виведення** повідомлення «Масив c містить менше 2 елементів»  **для** (i = 0; i < k; i++)  **якщо** (c[i] < first)  second = first;  first = c[i];  **інакше якщо** (с[i] < second та c[i] != first)  second = c[i] |
| 5 | **виведення «**Першим найменшим елементом є *u*, а другим найменшим елементом є *w.*  Сума 2 найменших елементів: *ь*» |
|  | **Кінець** |

**Випробування функції.**

|  |  |
| --- | --- |
| **Блок** | **Дія** |
|  | **Початок** |
| 1 | **Введення** j |
| 2 | **для** (j = 0; j < length; j++)  **якщо** (ai == b[j]  **повернення** «істина»;  **повернення** «брехня»; |
|  | **Кінець** |

**Висновок.**

В результаті виконання лабораторної роботи:

- набула теоритичні знання о лінійних пошуках в послідовностях: визначила як виконується в алгоритмах, де застосовується, які оператори використовуються для обробки значень.

- визначила основні кроки дій, записала програмні специфікації у псевдокоді, зобразила алгоритм у графічній формі у вигляді блок-схеми.

- представила програмні спеціфікації у вигляді коду програми на мові програмування С++, отримала правильний результат, виконала перевірку алгоритму підставляючи конкретні значення у певні дані.