

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 лінійних алгоритмів ”

Варіант:12

Виконав студент: ІІ-12 Єльчанінов Артем Юрійович
(шифр, прізвище, ім'я, по батькові)

Перевірив: _____
(прізвище, ім'я, по батькові)

Лабораторна робота 7

Дослідження лінійного пошуку в послідовностях

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

Варіант 12

Задача: Знайти суму елементів, коди яких більше 101

Вираз для обчислення 1-го масиву: $95 + i$

Вираз для обчислення 2-го масиву: $105 - i$

Постановка задачі

Результатом розв'язку задачі є знаходження суми елементів третього масиву, коди яких більше 101.

Спершу заповнюємо за даними умовою задачі виразами для знаходження елементів перший та другий масиви, але враховуючи специфіку задачі ми ще в дії заповнення масивів заповнюємо третій масив пустими елементами. Потім виводимо на екран за допомогою допоміжної функції елементи 1-го та 2-го масивів. Далі відбувається заповнення 3-го масиву рівними елементами 1-го та 2-го масивів, після цієї дії виведення на екран елементів масиву. Останньою дією є знаходження суми елементів, коди яких більше 101, яке відбувається перебором елементів 3-го масиву, і якщо код елемента більше 101, то він додається до змінної, яка зберігатиме значення суми. І після виконання цієї дії задача буде виконана.

Математична модель

Змінна	Тип	Ім'я	Призначення
Розмір масивів	Цілий	size	Вхідне дане
Перший масив	Символьний	array_1	Промідне дане
Другий масив	Символьний	array_2	Проміжне дане
Третій масив	Символьний	array_3	Проміжне дане
Лічильник для арифметичних цилів	Цілий	i	Проміжне дане
Додатковий лічильник для арифметичного циклу	Цілий	j	Проміжне дане

Сума елементів, коди яких більше 101	Цілий	sum	Вихідне дане
Функція для виведення значень масиву	Відсутній (void)	output_array	Допоміжний алгоритм

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію заповнення трьох масивів.

Крок 3. Деталізуємо дію заповнення 3-го масиву рівними елементами 1-го та 2-го масивів.

Крок 4. Деталізуємо дію обчислення суми.

Псевдокод алгоритму

Крок 1:

Початок

Ініціалізація змінних

Заповнення трьох масивів

Виведення елементів першого масиву

Виведення елементів другого масиву

Перебір елементів 1-го масиву

Перебір елементів 2-го масиву

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

Виведення елементів третього масиву

Обчислення суми

Виведення sum

Кінець

Крок 2:

Початок

size:= 10; sum:= 0;

Заповнення трьох масивів

Виведення елементів першого масиву

Виведення елементів другого масиву

Перебір елементів 1-го масиву

Перебір елементів 2-го масиву

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

Виведення елементів третього масиву

Обчислення суми

Виведення sum

Кінець

Крок 3:

Початок

size:= 10; sum:= 0;

для i від 0 до size з кроком 1

array_1[i] := 95 + i

array_2[i] := 105 – i

array_3[i] := 0

все повторити

Виведення елементів першого масиву

Виведення елементів другого масиву

Перебір елементів 1-го масиву

Перебір елементів 2-го масиву

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

Виведення елементів третього масиву

Обчислення суми

Виведення sum

Кінець

Крок 4:

Початок

size:= 10; sum:= 0;

для i від 0 до size з кроком 1

array_1[i] := 95 + i

array_2[i] := 105 – i

array_3[i] := 0

все повторити

output_array(array_1, size)

Виведення елементів другого масиву

Перебір елементів 1-го масиву

Перебір елементів 2-го масиву

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

Виведення елементів третього масиву

Обчислення суми

Виведення sum

Кінець

Крок 5:

Початок

size:= 10; sum:= 0;

для i від 0 до size з кроком 1

array_1[i] := 95 + i

array_2[i] := 105 – i

array_3[i] := 0

все повторити

output_array(array_1, size)

output_array(array_2, size)

Перебір елементів 1-го масиву

Перебір елементів 2-го масиву

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

Виведення елементів третього масиву

Обчислення суми

Виведення sum

Кінець

Крок 6:

Початок

size:= 10; sum:= 0;

для і від 0 до size з кроком 1

array_1[i] := 95 + i

array_2[i] := 105 – i

array_3[i] := 0

все повторити

output_array(array_1, size)

output_array(array_2, size)

для і від 0 до size з кроком 1

Перебір елементів 2-го масиву

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

все повторити

Виведення елементів третього масиву

Обчислення суми

Виведення sum

Кінець

Крок 7:

Початок

size:= 10; sum:= 0;

для і від 0 до size з кроком 1

array_1[i] := 95 + i

array_2[i] := 105 – i

array_3[i] := 0

все повторити

output_array(array_1, size)

output_array(array_2, size)

для і від 0 до size з кроком 1

для j від 0 до size з кроком 1

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

все повторити

все повторити

Виведення елементів третього масиву

Обчислення суми

Виведення sum

Кінець

Крок 8:

Початок

size:= 10; sum:= 0;

для і від 0 до size з кроком 1

array_1[i] := 95 + i

array_2[i] := 105 – i

array_3[i] := 0

все повторити

output_array(array_1, size)

output_array(array_2, size)

для і від 0 до size з кроком 1

для j від 0 до size з кроком 1

якщо array_1[i] == array_2[j]

то

array_3[i] := array_1[i]

все якщо

все повторити

все повторити

Виведення елементів третього масиву

Обчислення суми

Виведення sum

Кінець

Крок 9:

Початок

size:= 10; sum:= 0;

для і від 0 до size з кроком 1

array_1[i] := 95 + i

array_2[i] := 105 – i

array_3[i] := 0

все повторити

output_array(array_1, size)

output_array(array_2, size)

для і від 0 до size з кроком 1

для j від 0 до size з кроком 1

якщо array_1[i] == array_2[j]

то

array_3[i] := array_1[i]

все якщо

все повторити

все повторити

output_array(array_3, size)

Обчислення суми

Виведення sum

Кінець

Крок 10:

Початок

size:= 10; sum:= 0;

для і від 0 до size з кроком 1

array_1[i] := 95 + i

array_2[i] := 105 – i

array_3[i] := 0

все повторити

output_array(array_1, size)

output_array(array_2, size)

для і від 0 до size з кроком 1

для j від 0 до size з кроком 1

якщо array_1[i] == array_2[j]

то

array_3[i] := array_1[i]

все якщо

все повторити

все повторити

output_array(array_3, size)

для і від 0 до size з кроком 1

якщо array_3[i] > 101

то

sum += array_3[i]

все якщо

все повторити

Виведення sum

Кінець

output_array(array[], size):

Початок

для і від 0 до size з кроком 1

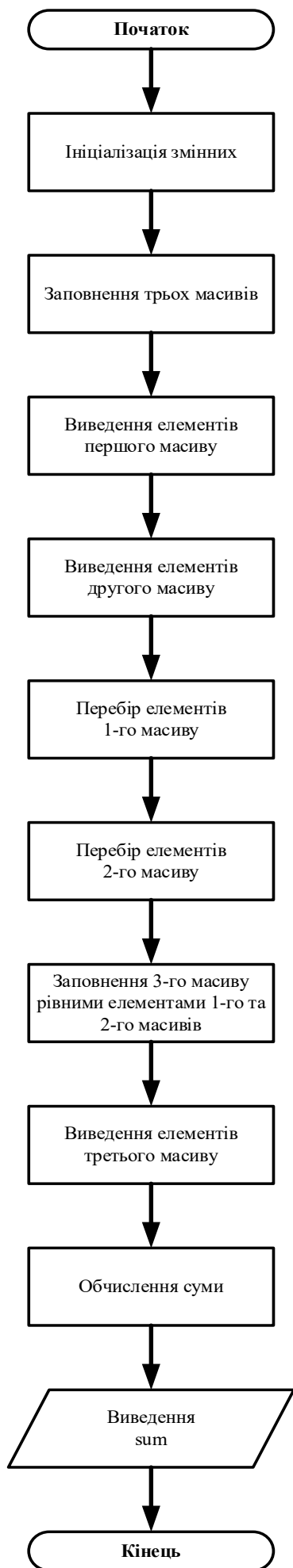
Виведення array[i]

все повторити

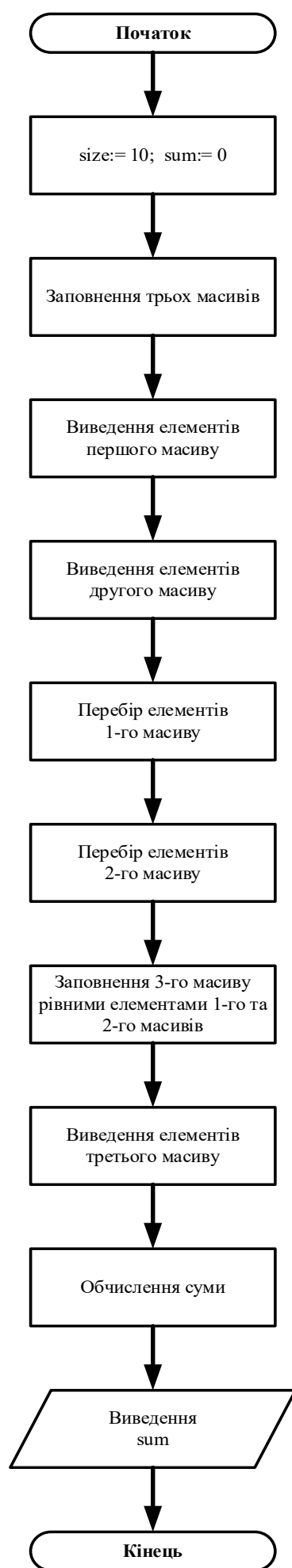
Кінець

Блок-схема

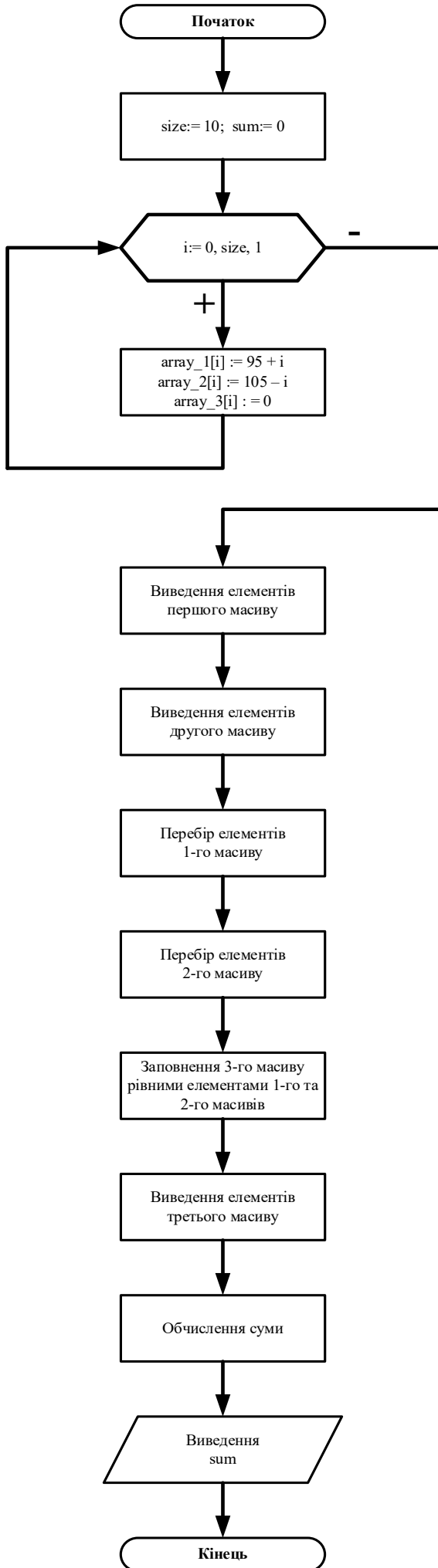
Крок 1:



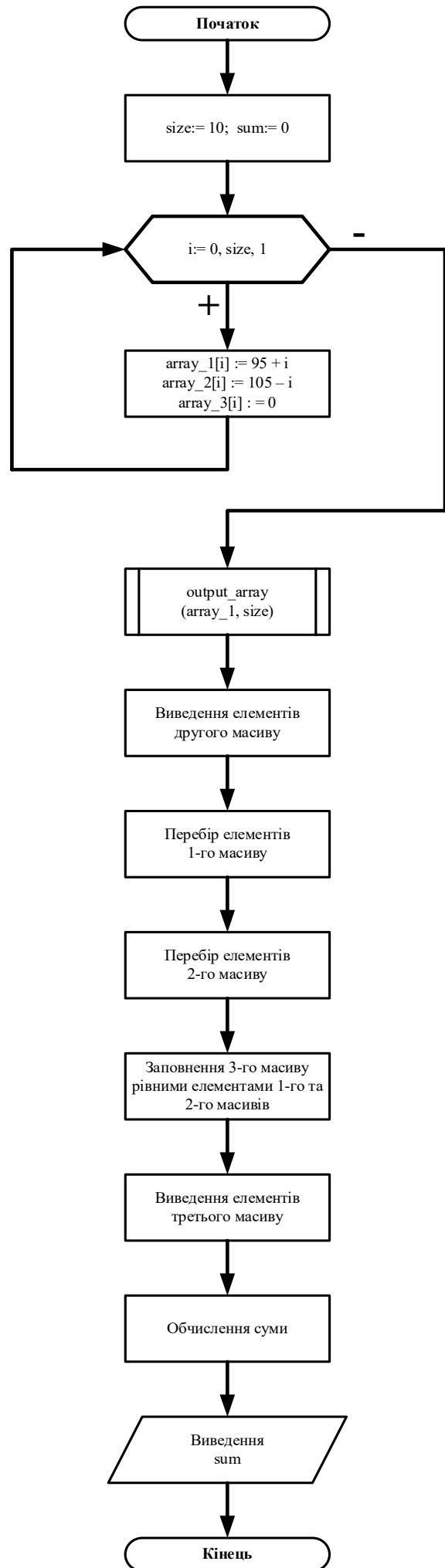
Крок 2:



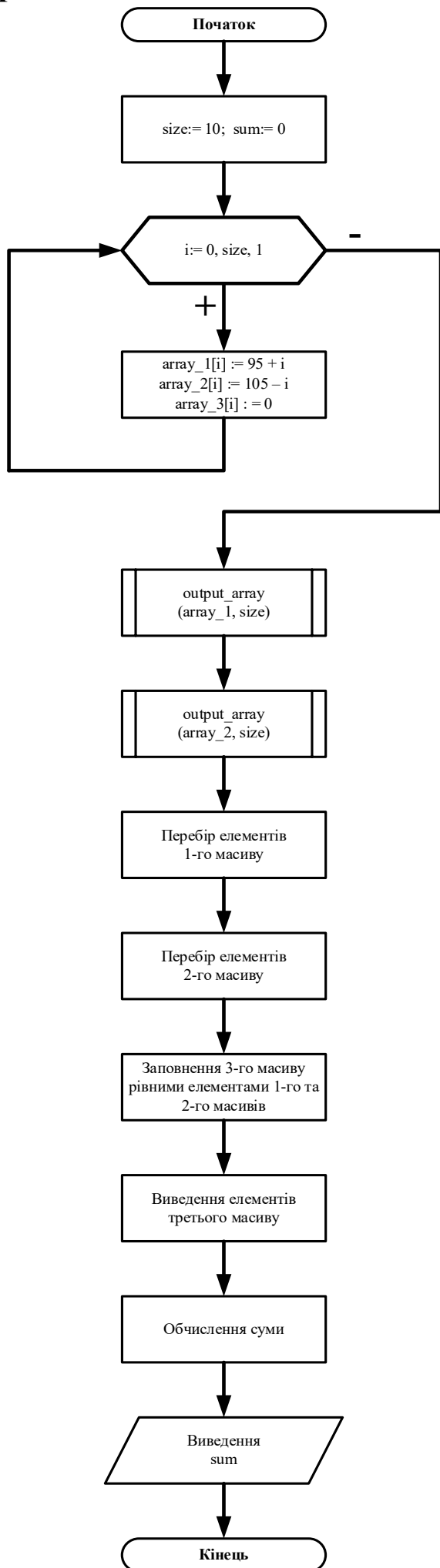
Крок 3:



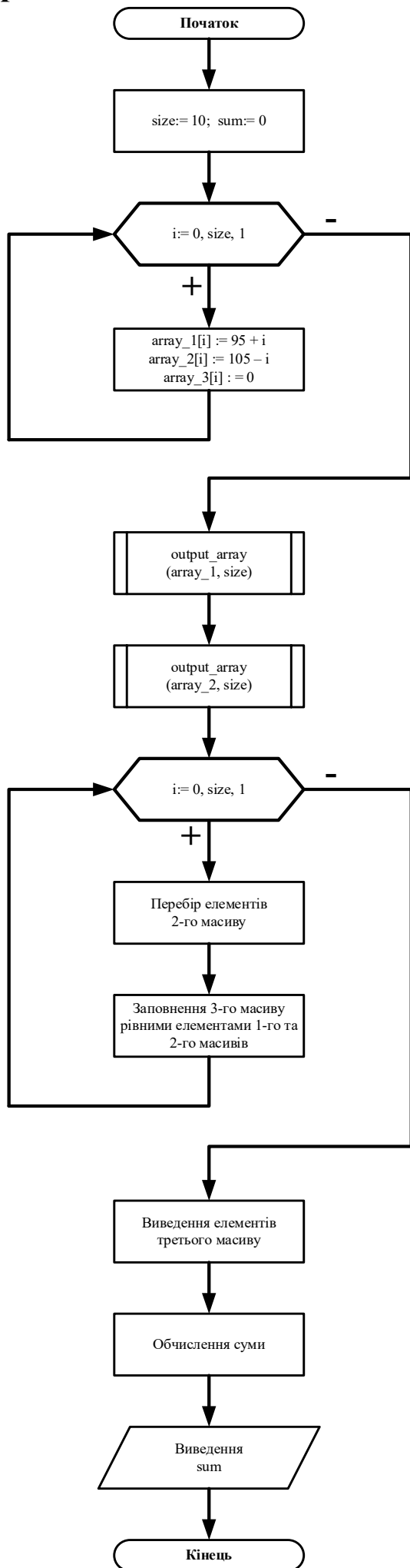
Крок 4:



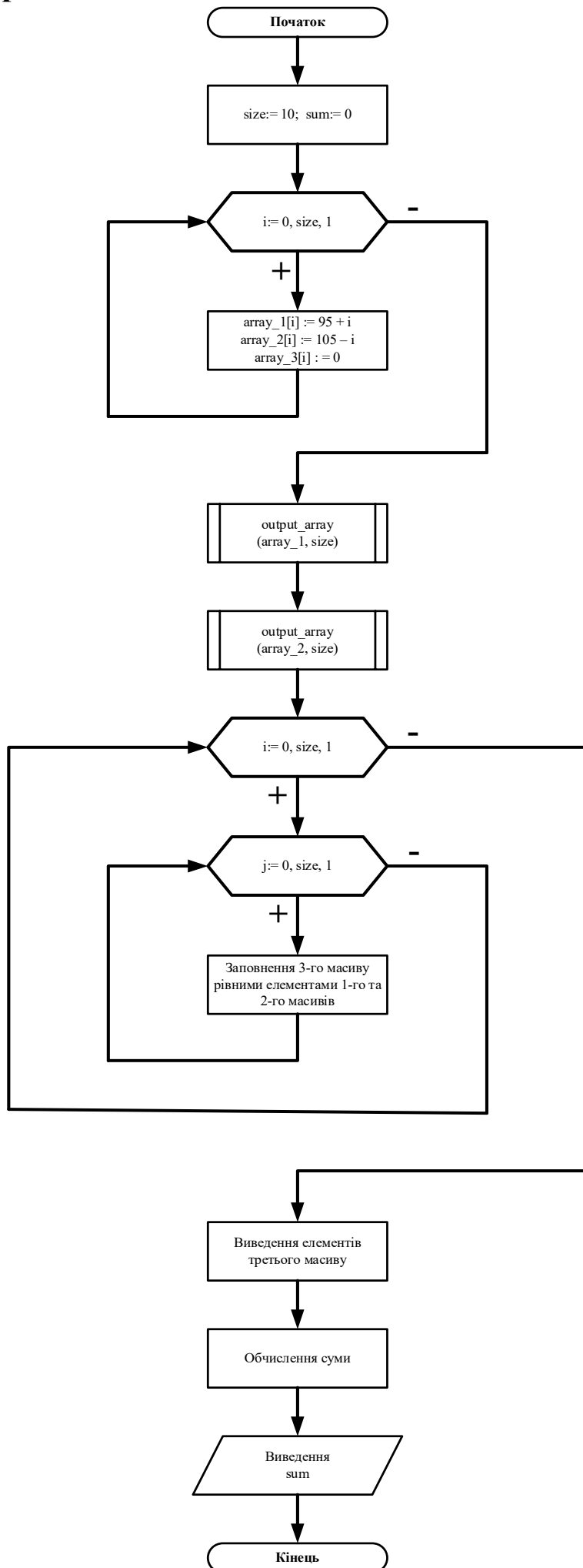
Крок 5:



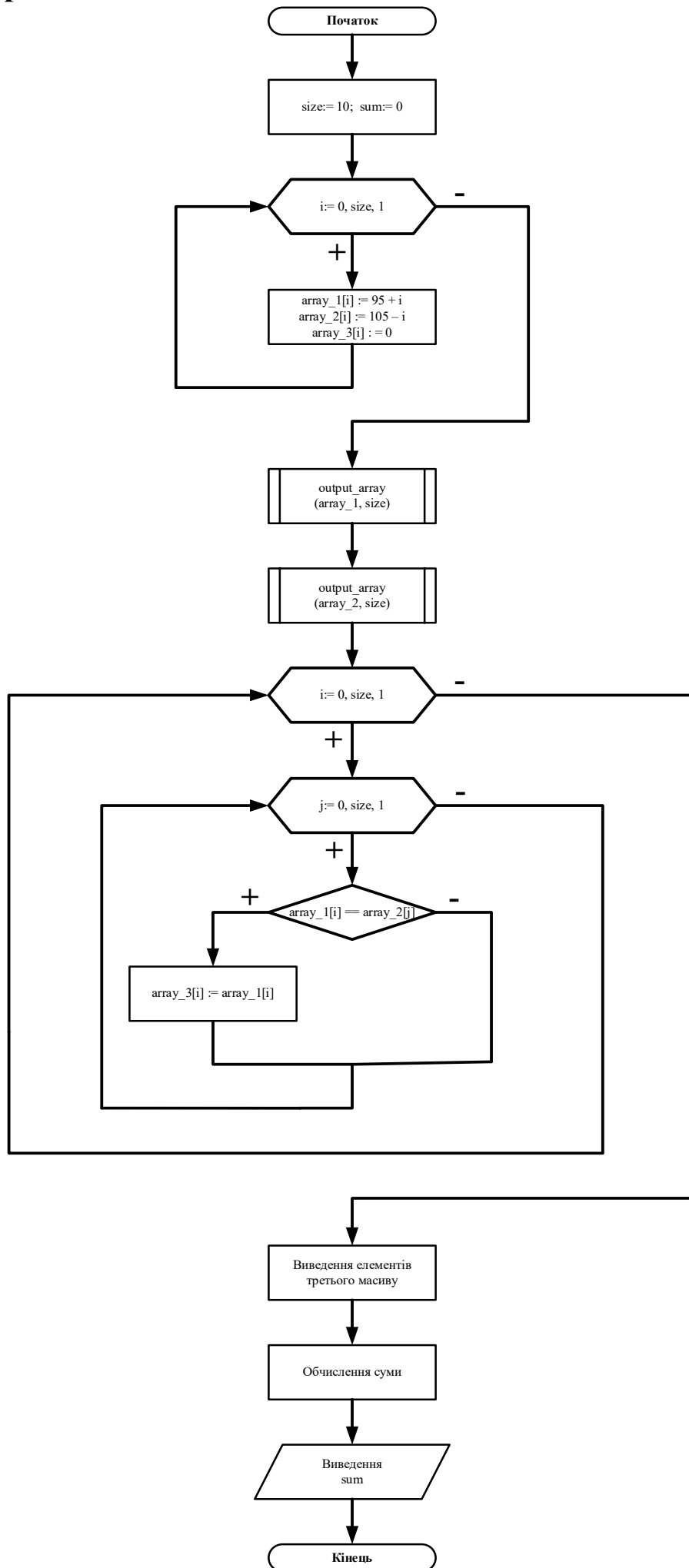
Крок 6:



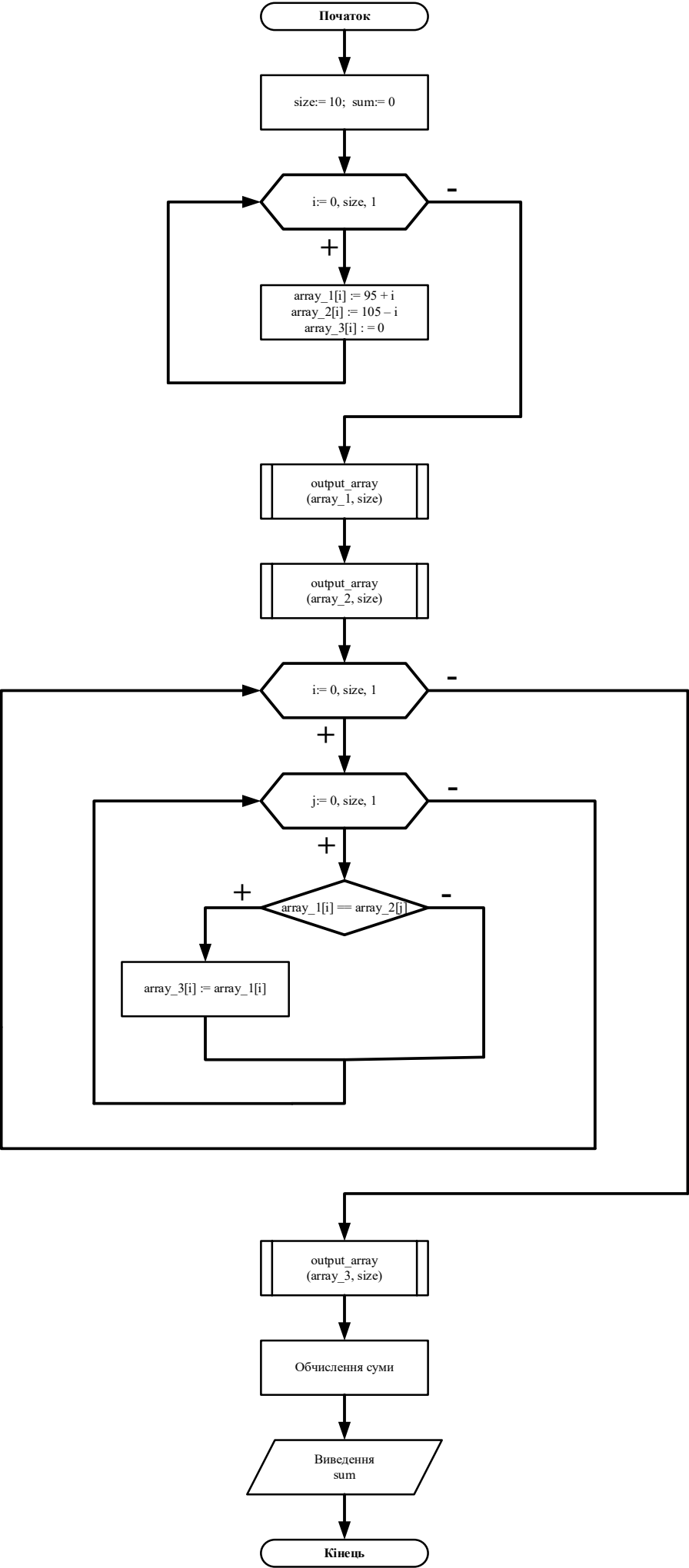
Крок 7:



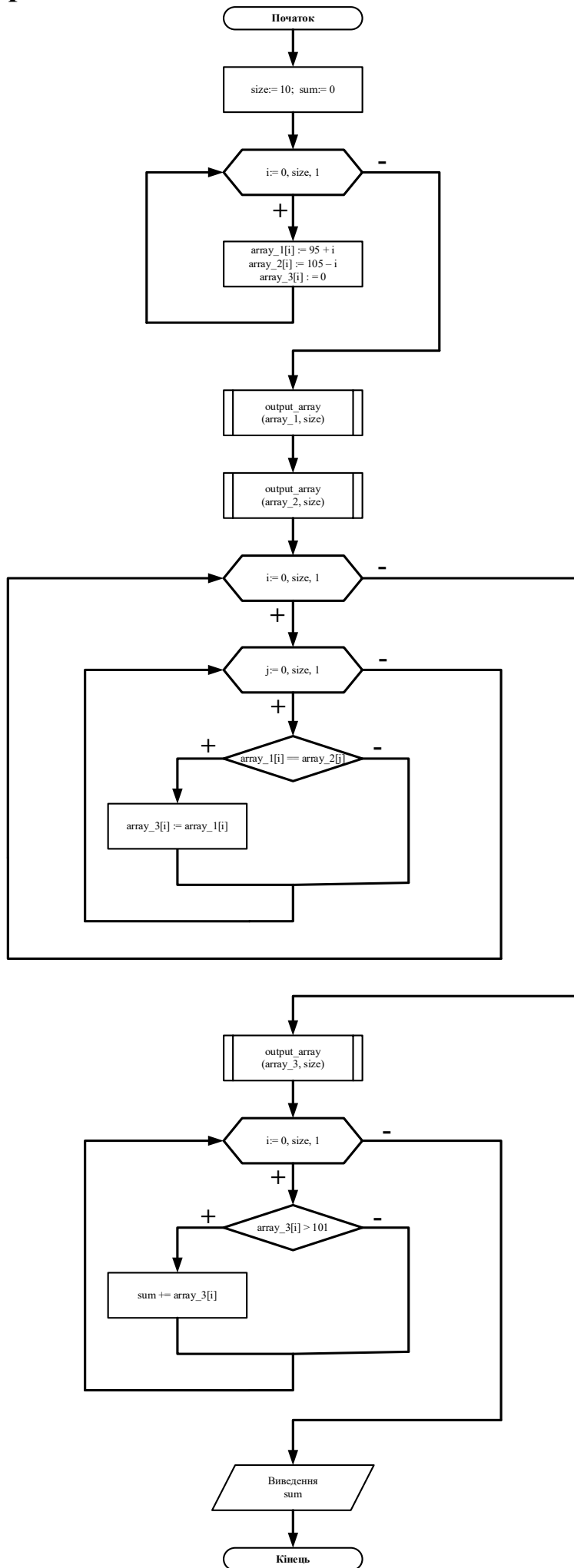
Крок 8:



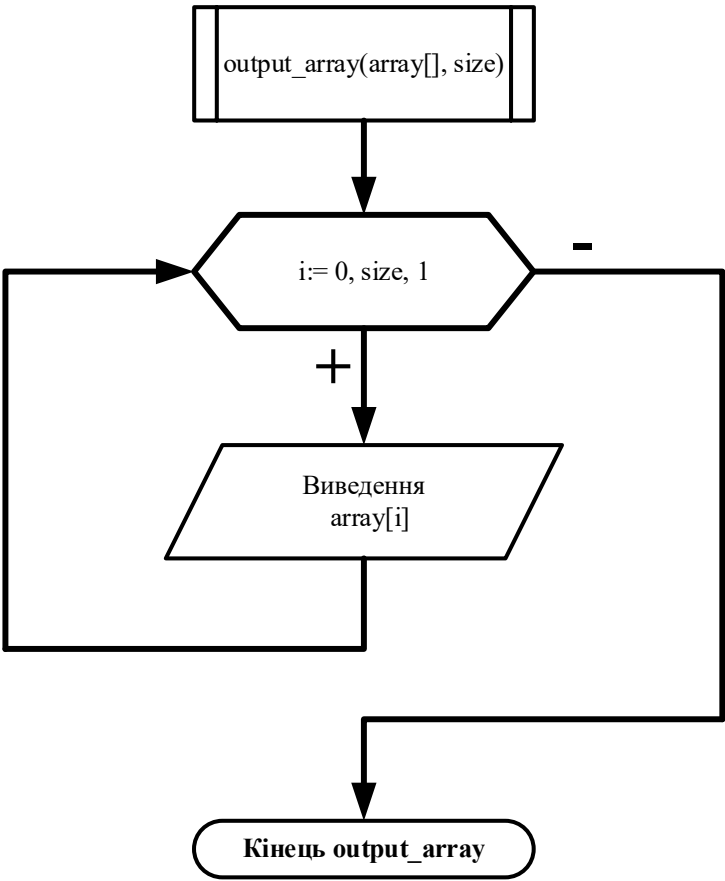
Крок 9:



Крок 10:



output_array(array[], size):



Код програми на мові C++ :

```
#include <iostream>
#include <iomanip>
using namespace std;

void output_array(char array[], int size);

int main() {
    int sum = 0;
    const int size = 10;
    char array_1[size], array_2[size], array_3[size];
    for (int i = 0; i < size; i++) {
        array_1[i] = 95 + i;
        array_2[i] = 105 - i;
        array_3[i] = 0;
    }
    cout << "array_1:" << endl;
    output_array(array_1, size);
    cout << "array_2:" << endl;
    output_array(array_2, size);
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (array_1[i] == array_2[j]) {
                array_3[i] = array_1[i];
            }
        }
    }
    cout << "array_3:" << endl;
    output_array(array_3, size);
    for (int i = 0; i < size; i++) {
        if (array_3[i] > 101) {
            sum += array_3[i];
        }
    }
    cout << "sum: " << sum << endl;
    system("pause");
    return 0;
}

void output_array(char array[], int size) {
    for (int i = 0; i < size; i++) {
        cout << setw(2) << array[i];
    }
    cout << endl << endl;
}
```

Тестування програми:

```
array_1:
_ ` a b c d e f g h

array_2:
i h g f e d c b a `

array_3:
` a b c d e f g h

sum: 309
```

Висновок.

У результаті лабораторної роботи було розроблено математичну модель, що відповідає постановці задачі; псевдокод та блок-схеми, які пояснюють логіку алгоритму. Було набуто практичного новичок у використанні методів лінійного пошуку в послідовностях та їх інтерпретації у блок-схеми і псевдокод.

Алгоритм був випробуваний з використанням значень виразів для знаходження елементів 1-го ($95 + i$) та 2-го ($105 - i$) масивів, заданих умовою задачі. У підсумку було отримано, що $sum = 309$. Таким чином, було доведено вірність складеного алгоритму. Отже, його можна застосовувати для лінійного пошуку в послідовностях.