

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 23

Виконав студент ПІ-13 Недельчев Євген Олександрович  
(шифр, прізвище, ім'я, по батькові)

Перевірила Вечерковська Анастасія Сергіївна  
(прізвище, ім'я, по батькові)

Київ 20211

## Лабораторна робота 7

### Дослідження лінійного пошуку в послідовностях

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

#### Індивідуальне завдання

##### Варіант 23

№	Вираз для обчислення елемента		Знайти
	1-го масиву	2-го масиву	
23	$35 + 3 * i$	$56 - 2 * i$	Середнє арифметичне елементів, коди яких більше 38

#### Постановка задачі

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 згідно з варіантом.
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної згідно з варіантом.

#### Побудова математичної моделі

Змінна	Тип	Ім'я	Призначення
Перший змінна індексованого типу з 10 символьних значень	Символьний	array1[10]	Початкові дані
Друга змінна індексованого типу з 10 символьних значень	Символьний	array2[10]	Початкові дані
Третя змінна індексованого типу з 10 символьних значень	Символьний	array3[10]	Результат
Лічильник у циклах	Цілий	i	Лічильник
Лічильник у вкладеному циклі	Цілий	j	Лічильник
Кількість повторювальних елементів у першому та другому масивах	Цілий	count	Проміжний результат
Сума елементів третього масиву, коди яких більші за 38	Цілий	sum	Проміжний результат
Кількість елементів третього масиву, коди яких більші за 38	Цілий	n	Проміжний результат

Таким чином математичне формулювання задачі зводиться до обчислення двох масивів розмірністю 10, де елементи знаходяться за формулами:

1.  $35 + 3 \cdot i$

2.  $56 - 2 \cdot i$ ,

де  $i$  – це порядковий номер елемента в масиві ( $i \in [0, 9]$ ).

Елементи третього масиву - це повторювальні елементи першого і другого масиву, тобто їх перетин.

Таким чином, перший масив повинен мати вигляд:

[ # & ) , / 2 5 8 ; > ],

другий: [ 8 6 4 2 0 . , \* ( & ],

третій: [ & , 2 8 ].

Для пошуку середнього арифметичного елементів третього масиву, коди яких більші за 38, створимо функцію `arithmetical__mean(char* arr, int count)`, параметрами якої є покажчик на масив, середнє арифметичне елементів якого ми шукаємо, та кількість елементів цього масиву. Функція повертає значення виразу  $\text{sum} / n$ , де  $\text{sum}$  – сума елементів третього масиву, коди яких більші за 38, а  $n$  – кількість елементів третього масиву, коди яких більші за 38.

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

### Розв'язання

Крок 1. Визначимо основні дії.

Крок 2. Опис трьох змінних індексованого типу з 10 символічних значень.

Крок 3. Ініціалізація першого та другого масивів за заданими формулами.

Крок 4. Ініціалізація третього масиву перетином першого та другого масивів.

Крок 5. Пошук та виведення середнього арифметичного елементів третього масиву, коди яких більші за 38.

## Псевдокод

Крок 1.

**функція** `arithmetical__mean(char* arr, int count)`

пошук середнього арифметичного елементів масиву, коди яких >38

**все функція**

**початок**

опис трьох змінних індексованого типу з 10 символьних значень

ініціалізація першого та другого масивів за заданими формулами

ініціалізація третього масиву перетином першого та другого масивів

виклик функції `arithmetical__mean(array3)`

**кінець**

Крок 2.

**функція** `arithmetical__mean(char* array3, int count)`

пошук середнього арифметичного елементів масиву, коди яких >38

**все функція**

**початок**

`array1[10]`

`array2[10]`

`array3[10]`

ініціалізація першого та другого масивів за заданими формулами

ініціалізація третього масиву перетином першого та другого масивів

виклик функції `arithmetical__mean(array3, count)`

**кінець**

Крок 3.

**функція** `arithmetical__mean(char* array3, int count)`

пошук середнього арифметичного елементів масиву, коди яких >38

**все функція**

**початок**

`array1[10]`

`array2[10]`

`array3[10]`

**повторити** для  $i$  від 0 до 10

`array1[i] := 35 + 3*i`

`array2[i] := 56 - 2 * i`

**все повторити**

ініціалізація третього масиву перетином першого та другого масивів  
виклик функції `arithmetical__mean(array3, count)`

**кінець**

Крок 4.

**функція** `arithmetical__mean(char* array3, int count)`

пошук середнього арифметичного елементів масиву, коди яких  $>38$

**все функція**

**початок**

`array1[10]`

`array2[10]`

`array3[10]`

**повторити** для  $i$  від 0 до 10

`array1[i] := 35 + 3*i`

`array2[i] := 56 - 2 * i`

**все повторити**

`count := 0`

**повторити** для  $i$  від 0 до 10

**повторити** для  $j$  від 0 до 10

**якщо** `array1[i] = array2[j]`

`array3[count] := array1[i]`

`count++`

**все якщо**

**все повторити**

**все повторити**

виклик функції `arithmetical__mean(array3, count)`

**кінець**

Крок 5.

**функція** arithmetical\_\_mean(char\* array3, int count)

sum := 0

n := 0

**повторити** для i від 0 до count

**якщо** array3[i] > 38

sum += array3[i]

n++

**все якщо**

**все повторити**

повернути (sum / n)

**все функція**

**початок**

array1[10]

array2[10]

array3[10]

**повторити** для i від 0 до 10

array1[i] := 35 + 3\*i

array2[i] := (56 - 2 \* i

**все повторити**

count := 0

**повторити** для i від 0 до 10

**повторити** для j від 0 до 10

**якщо** array1[i] = array2[j]

array3[count] := array1[i]

count++

**все якщо**

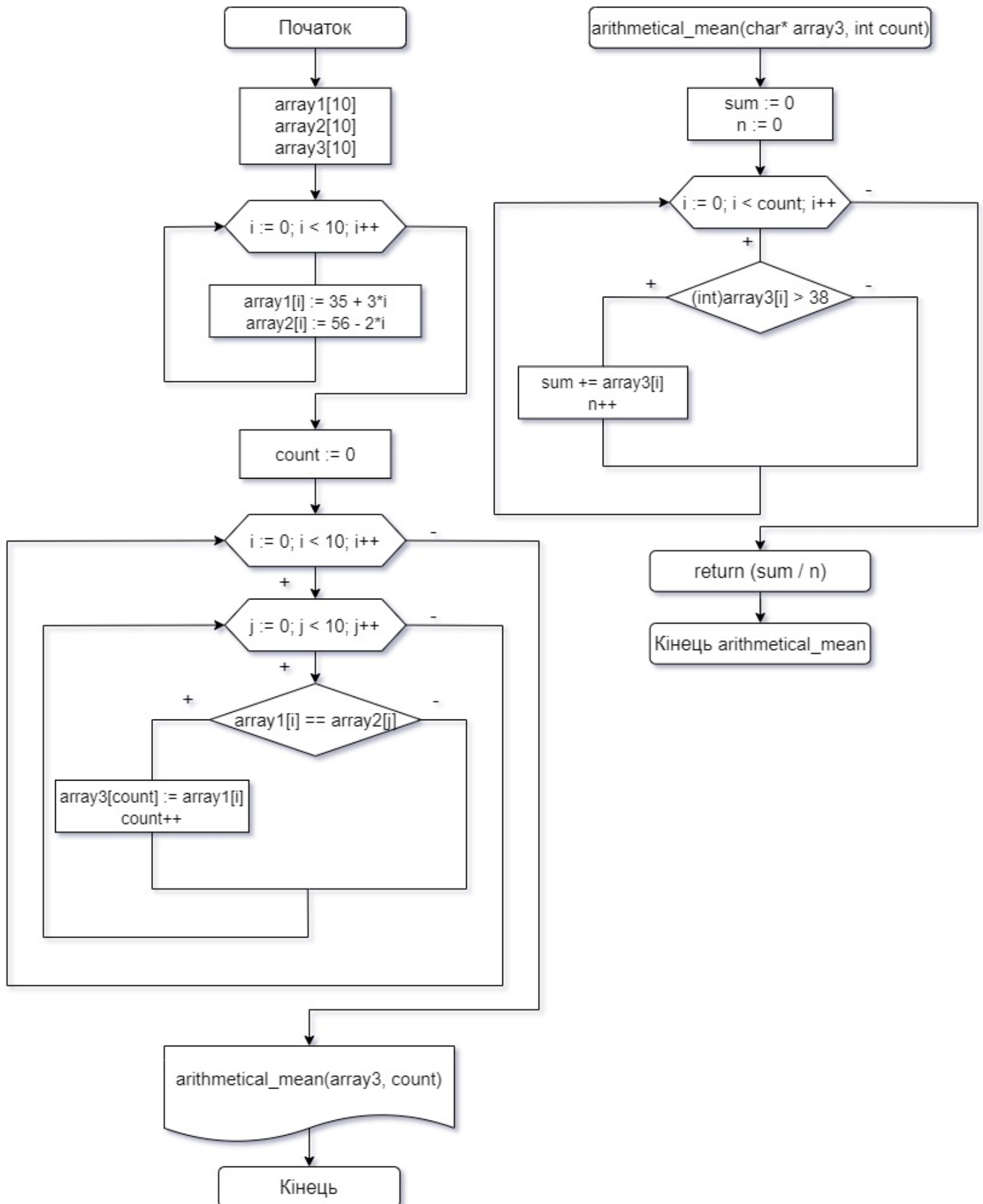
**все повторити**

**все повторити**

виведення arithmetical \_\_mean(array3, count)

**кінець**

## Блок-схема



## Код програми

```
main.cpp
ASD_lab7 (Глобальная область)

1  #include <iostream>
2
3  using std::cout;
4
5  double arithmetical_mean(char* array3, int count) {
6      int sum = 0, n = 0;
7      for (int i = 0; i < count; i++) {
8          if (array3[i] > 38) {
9              sum += array3[i];
10             n++;
11         }
12     }
13     return sum / n;
14 }
15
16 int main() {
17     char array1[10], array2[10], array3[10];
18     for (int i = 0; i < 10; i++) {
19         array1[i] = 35 + 3 * i;
20         array2[i] = 56 - 2 * i;
21     }
22     int count = 0;
23     for (int i = 0; i < 10; i++) {
24         for (int j = 0; j < 10; j++) {
25             if (array1[i] == array2[j]) {
26                 array3[count] = array1[i];
27                 count++;
28             }
29         }
30     }
31     cout << "array3: [ ";
32     for (int i = 0; i < count; i++) {
33         cout << array3[i] << ' ';
34     }
35     cout << ']' << std::endl;
36     cout << "The arithmetical mean is: " << arithmetical_mean(array3, count);
37 }
```

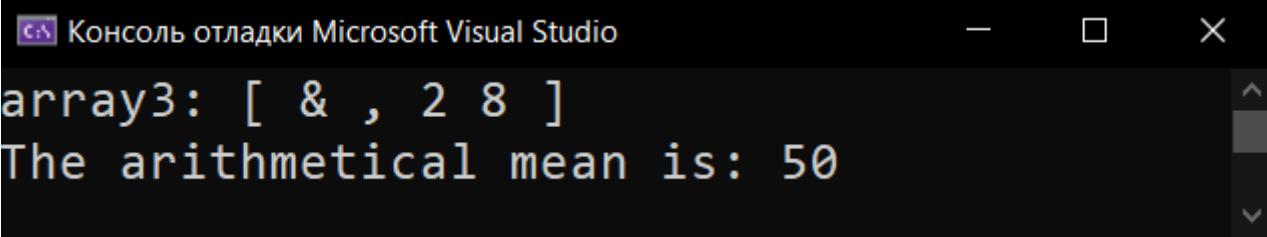
87 %



Проблемы не найдены.



## Тестування алгоритму



```
Консоль отладки Microsoft Visual Studio  
array3: [ & , 2 8 ]  
The arithmetical mean is: 50
```

## Висновки

Під час виконання цієї лабораторної роботи я дослідив методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набув практичних навичок їх використання під час складання програмних специфікацій.