

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 30

Виконав студент ІІ-13 Симолюк Денис Андрійович

(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна

( прізвище, ім'я, по батькові)

## Лабораторна робота 6

### Дослідження рекурсивних алгоритмів

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

### Варіант 30

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

30	43 - i	37 + i	Добуток елементів, коди яких більше 40
----	--------	--------	----------------------------------------

### Постановка задачі

Для вирішення задачі створюємо 2 масиви з елементами, що обчислюються за даними формулами. Елементи третього масиву обраховуються за вкладеним циклом пошуку однакових елементів у перших двох. Елементи, що залишилися, заповнюються нулями. Для знаходження добутку створюємо цикл, де за умови, що код елемента  $>40$ , змінній присвоюється значення самої себе, помноженої на поточний елемент.

## Побудова математичної моделі

Побудуємо таблицю змінних:

Змінна	Тип	Призначення
Перший масив arr1	Символьний	Проміжні дані
Другий масив arr2	Символьний	Проміжні дані
Третій масив arr3	Символьний	Проміжні дані
Лічильник циклу i	Цілий	Проміжні дані, лічильник
Лічильник циклу j	Цілий	Проміжні дані, лічильник
Процедура InnitArr1	-	Проміжні дані
Процедура InnitArr2	-	Проміжні дані
Процедура InnitArr3	-	Проміжні дані
Функція CountRes	Цілий	Проміжні дані
Добуток елементів arr3, що >40 res	Цілий	Вихідні дані

Крок 1. Визначимо основні дії.

Крок 2. Ініціалізуємо масив arr1.

Крок 3. Деталізуємо процедуру InnitArr1.

Крок 4. Ініціалізуємо масив arr2.

Крок 5. Деталізуємо процедуру InnitArr2.

Крок 6. Ініціалізуємо масив arr3.

Крок 7. Деталізуємо процедуру InnitArr3.

Крок 8. Ініціалізуємо res.

Крок 9. Деталізуємо функцію CountRes.

## Псевдокод алгоритму

### Початок

```
char arr1[10], arr2[10], arr3[10]{ }
```

```
InnitArr1(arr1)
```

InnitArr2(arr2)

InnitArr3(arr1, arr2, arr3)

res = CountRes(arr3)

Виведення res

**Кінець**

**Процедура InnitArr1(arr1)**

повторити для  $i$  від 0 до 10 не включно з кроком 1:

$arr1[i] = 43 - i$

**Все процедура**

**Процедура InnitArr2(arr2)**

повторити для  $i$  від 0 до 10 не включно з кроком 1:

$arr2[i] = 37 + i$

**Все процедура**

**Процедура InnitArr3(arr1, arr2, arr3)**

повторити для  $i$  від 0 до 10 не включно з кроком 1:

повторити для  $j$  від 0 до 10 не включно з кроком 1:

**Якщо** ( $arr1[i] == arr2[j]$ )

**то**

$arr3[i] = arr1[i]$

**Все якщо**

**Все повторити**

**Все повторити**

**Все процедура**

**Процедура InnitArr1(arr1)**

повторити для  $i$  від 0 до 10 не включно з кроком 1:

$arr1[i] = 43 - i$

**Все процедура**

**Функція CountRes(arr3)**

res = 1

**повторити для i від 0 до 10 не включно з кроком 1:**

**якщо (arr3[i] > 40)**

**то**

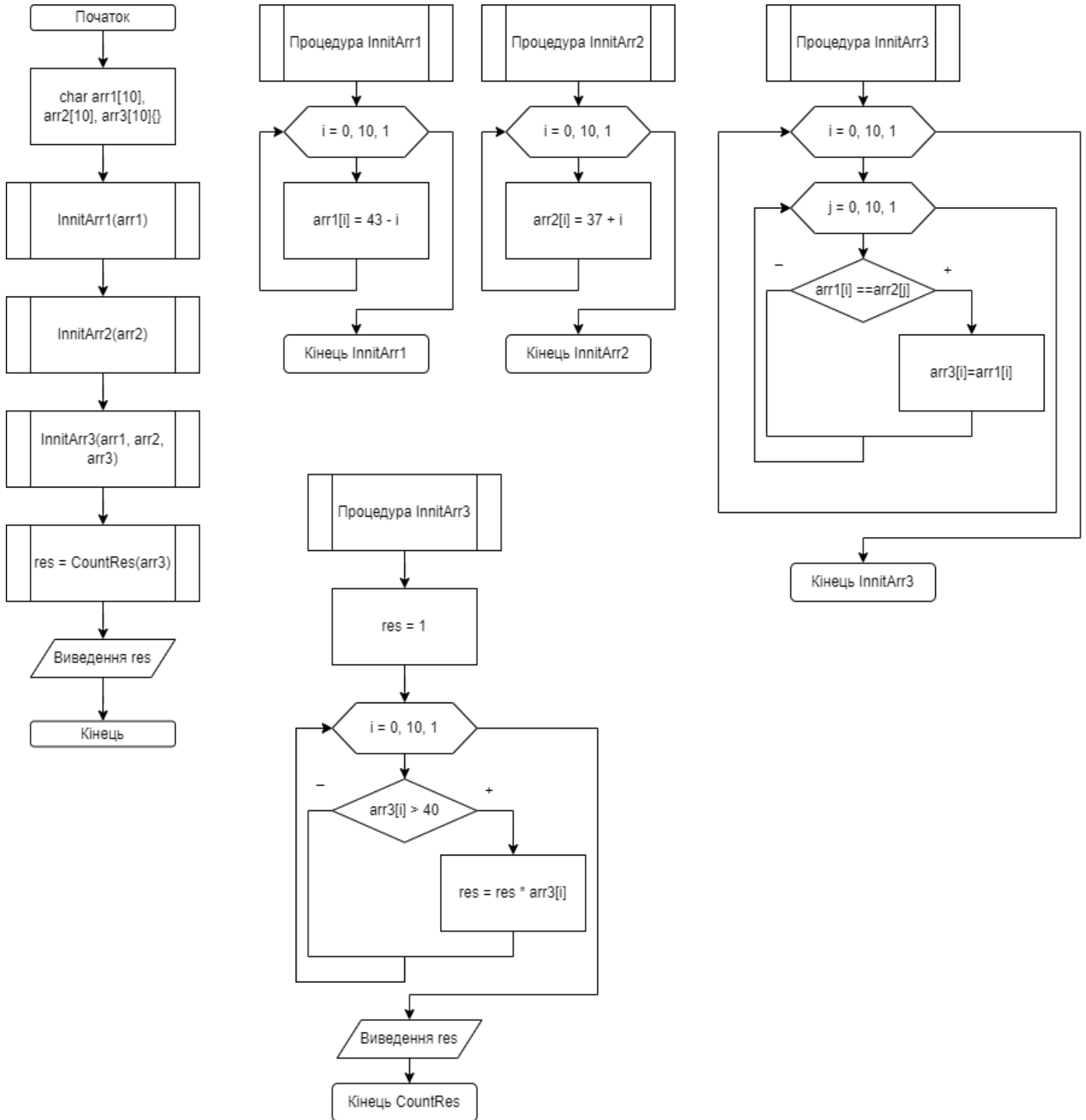
res = res \* arr3[i]

**все якщо**

**все повторити**

**Все функція**

## Блок-схема алгоритму



## Код програми (C++)

```
1  #include <iostream>
2  using namespace std;
3
4  void InnitArr1(char arr[]), InnitArr2(char arr[]), InnitArr3(char arr1[], char arr2[], char arr3[]);
5  int CountRes(char arr3[]);
6
7  int main()
8  {
9      int res;
10     char arr1[10], arr2[10], arr3[10] = {};
11     InnitArr1(arr1);
12     InnitArr2(arr2);
13     InnitArr3(arr1, arr2, arr3);
14     res = CountRes(arr3);
15     cout << "Res = " << res << "\n";
16 }
17
18 void InnitArr1(char arr1[])
19 {
20     for (int i = 0; i < 10; i++)
21     {
22         arr1[i] = 43 - i;
23     }
24 }
25
26 void InnitArr2(char arr2[])
27 {
28     for (int i = 0; i < 10; i++)
29     {
30         arr2[i] = 37 + i;
31     }
32 }
33
34 void InnitArr3(char arr1[], char arr2[], char arr3[])
35 {
36     for (int i = 0; i < 10; i++)
37     {
38         for (int j = 0; j < 10; j++)
39         {
40             if (arr1[i] == arr2[j])
41                 arr3[i] = arr1[i];
42         }
43     }
44 }
45
46 int CountRes(char arr3[])
47 {
48     int res = 1;
49     for (int i = 0; i < 10; i++)
50     {
51         if (arr3[i] > 40)
52             res *= arr3[i];
53     }
54     return res;
55 }
```

## Тестування програми

Консоль отладки Microsoft Visual Studio

Res = 74046

D:\Projects\lab7 asd\Debug\lab7 asd.exe (процесс 2936) завершил работу с кодом 0.  
Нажмите любую клавишу, чтобы закрыть это окно...

## **Висновки**

В даній роботі я дослідив методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набув практичних навичок їх використання під час складання алгоритму та написання програми на мові програмування C++.