

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 25

Виконав студент

Павленко Микита Андрійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

Вечерковська Анастасія Сергіївна

(прізвище, ім'я, по батькові)

Лабораторна робота 7

Дослідження складних рекурсивних алгоритмів

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

Варіант 25

Завдання

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 ($100 + i$, $110 - i * i$)
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної згідно з варіантом.

1) Постановка задачі

Необхідно заповнити 2 масиви символами з кодами заданими згідно з наданими виразами ($100 + i$, $110 - i * i$). Сформувати 3-ій масив із рівних елементів перших двох та вивести добуток кодів його елементів, які більші за 100.

2) Побудова математичної моделі

Таблиця імен змінних:

Змінна	Тип	Ім'я	Призначення
Перший масив	Одновимірний масив	array1	Вхідні дані
Другий масив	Одновимірний масив	array2	Вхідні дані
Третій масив	Одновимірний масив	array3	Проміжні дані
Результат	Цілий	result	Вихідні дані
Ітератори	Цілий	i, j	Проміжні дані

Отже, математичне формулювання задачі зводиться до складання циклів із заповнення перших двох масивів, формування третього масиву із однакових елементів 1-го і 2-го та виведення добутку кодів його елементів, що задовольняють умові.

3) Псевдокод алгоритму

Крок 1:

Функція Filling1 (array1)

Цикл заповнення масиву

Все функція

Функція Filling2 (array2)

Цикл заповнення масиву

Все функція

початок

Ініціалізація змінних **array1, array2, array3, result**

Заповнення масивів

Формування **array3**

Пошук добутку кодів елементів 3-го масиву

Виведення результату

кінець

Крок 2:

Функція Filling1 (char[] array)

Для i від 0 до 10 повторити

array[i] = (char)(100 + i)

все повторити

Все функція

Функція Filling2 (char[] array)

Для i від 0 до 10 повторити

array[i] = (char)(100 - i*i)

все повторити

Все функція

початок

Ініціалізація змінних **array1, array2, array3, result**

Виклик функції **Filling1(array1)**

Виклик функції **Filling2(array2)**

Формування **array3**

Пошук добутку кодів елементів 3-го масиву

Виведення результату

кінець

Крок 3:

Функція Filling1 (char[] array)

Для **i** від 0 до 10 повторити

array[i] = (char)(100 + i)

все повторити

Все функція

Функція Filling2 (char[] array)

Для **i** від 0 до 10 повторити

array[i] = (char)(100 - i*i)

все повторити

Все функція

початок

Ініціалізація змінних **array1, array2, array3, result**

Виклик функції **Filling1(array1)**

Виклик функції **Filling2(array2)**

Для **i** від 0 до 10 повторити

Для **j** від 0 до 10 повторити

Якщо **array1[i] == array2[j]** то

array3[i] = array1[i]

все якщо

все повторити

все повторити

Пошук добутку кодів елементів 3-го масиву

Виведення результату

кінець

Крок 4:

Функція Filling1 (char[] array)

Для **i** від 0 до 10 повторити

array[i] = (char)(100 + i)

все повторити

Все функція

Функція Filling2 (char[] array)

Для **i** від 0 до 10 повторити

array[i] = (char)(100 - i*i)

все повторити

Все функція

початок

Ініціалізація змінних **array1, array2, array3, result**

Виклик функції **Filling1(array1)**

Виклик функції **Filling2(array2)**

Для **i** від 0 до 10 повторити

Для **j** від 0 до 10 повторити

Якщо **array1[i] == array2[j]** то

array3[i] = array1[i]

все якщо

все повторити

все повторити

Для **i** від 0 до 10 повторити

Якщо $(\text{int})(\text{array3}[\text{i}]) > 100$ то

result = **result** * $(\text{int})(\text{array3}[\text{i}])$

все якщо

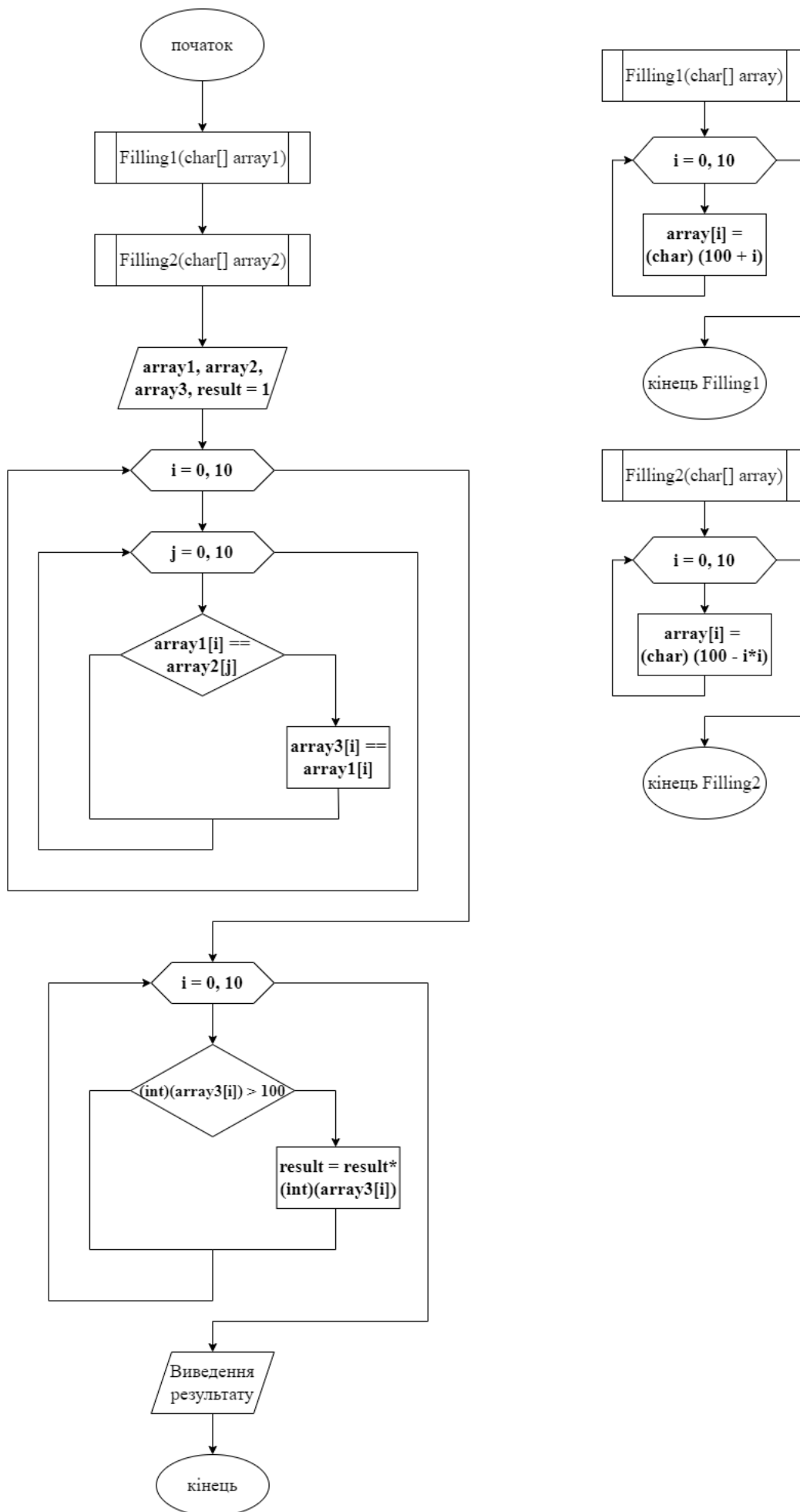
все повторити

Виведення результату

кінець

4) Блок-схема алгоритму


Алгоритми та структури даних. Основи алгоритмізації



5) Код програми

```
Ссылка: 0
5 class Program
6 {
7     ссылка: 1
8     static void Filling1(char[] array)
9     {
10         for(int i = 0; i < 10; i++)
11         {
12             array[i] = (char)(100 + i);
13             Console.Write(array[i] + " ");
14         }
15     }
16     ссылка: 1
17     static void Filling2(char[] array)
18     {
19         for(int i = 0; i < 10; i++)
20         {
21             array[i] = (char)(110 - i * i);
22             Console.Write(array[i] + " ");
23         }
24     }
25     static void Main(string[] args)
26     {
27         int result = 1;
28         char[] array1 = new char[10];
29         char[] array2 = new char[10];
30         char[] array3 = new char[10];
31
32         Filling1(array1);
33         Console.WriteLine();
34         Filling2(array2);
35
36         for(int i = 0; i < 10; i++)
37         {
38             for(int j = 0; j < 10; j++)
39             {
40                 if(array1[i] == array2[j])
41                 {
42                     array3[i] = array1[i];
43                 }
44             }
45         }
46
47         for(int i = 0; i < 10; i++)
48         {
49             if((int)(array3[i]) > 100)
50             {
51                 result = result * (int)(array3[i]);
52             }
53         }
54
55         Console.WriteLine();
56         Console.WriteLine(result);
57         Console.ReadKey();
58     }
59 }
60 }
61 }
```


Результат роботи програми

 Консоль отладки Microsoft Visual Studio

```
d e f g h i j k l m
n m j e ^ U J = . ⇐
1166954
```

5) Висновки

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