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 25

Виконав ІП-15, Плугатирьов Дмитро Валерійович

студент (шифр, прізвище, ім’я, по батькові)

Перевірив Вєчерковська Анастасія Сергіївна

(прізвище, ім’я, по батькові)

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.

**Варіант 25**

**Завдання**

25.) 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 (табл. 1).

3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.

4. Обробки третьої змінної згідно з варіантом.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Вираз для обчислення**  **елемента** | | **Знайти** |
| **1-го масиву** | **2-го масиву** |
| **25** | **100 + і** | **100 – і \* і** | **Добуток кодів елементів більших за 100** |

**1. П о с т а н о в к а з а д а ч і**

Описати три змінні індексованого типу з 10 символьних значень з таблиці ASCII. Ініціювати їх виразами з таблиці. Ініціювати третю змінну рівними значеннями першої та другої. Знайти добуток кодів елементів більших за 100 за допомогою вкладеного циклу та умови на величину значення символьних змінних.

**2. П о б у д о в а м а т е м а т и ч н о ї м о д е л і**

|  |  |  |  |
| --- | --- | --- | --- |
| *Змінна* | *Тип* | *Ім’я* | *Призначення* |
| **Розмір масивів** | **Цілочисельний** | **SIZE** | **Початкові дані** |
| **Перший масив** | **Символьний** | **arr1** | **Початкові дані** |
| **Другий масив** | **Символьний** | **arr2** | **Початкові дані** |
| **Третій масив** | **Символьний** | **arr3** | **Проміжні дані** |
| **Добуток** | **Цілочисельний** | **multiplication** | **Результат** |
| **Індекс третього масиву** | **Цілочисельний** | **temp** | **Проміжні дані** |
| **Добуток елементів третього масиву** | **Підпрограма** | **mult\_elements** | **Проміжні дані** |
| **Ініціалізація масивів** | **Підпрограма** | **init\_arrs** | **Початкові дані** |
| **Лічильник циклу** | **Цілочисельний** | **i** | **Проміжні дані** |
| **Лічильник вкладеного циклу** | **Цілочисельний** | **z** | **Проміжні дані** |

* Реалізація пошуку рівних значень у першому та другому масивах відбувається шляхом проходження по елементам другого масиву з використанням вкладеного циклу і порівняння кожного з них із першим першого з подальшим збільшенням його значення на одиницю. В разі рівності двох елементів їх значення заноситься до третього масиву.

Дія x++ означає x := x + 1.

Дія х \*= і означає x := x \* і.

Дія (int)x означає приведення змінної х до цілочисельного типу.

**3. Р о з в ' я з а н н я**

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

*Крок 1*. Визначити основні дії.

*Крок 2*. Опис й ініціювання першого та другого масивів.

*Крок 3*. Ініціювання третього масиву, пошук добутку елементів третього масиву з індексом більше 100.

**4. П с е в д о к о д**

**Основна програма:**

*Крок 1*

**початок**

опис й ініціювання першого та другого масивів

ініціювання третього масиву, пошук добутку елементів третього масиву з індексом більше 100

**кінець**

*Крок 2*

**початок**

SIZE := 10

**init\_arrs(arr1, arr2, arr3, SIZE)**

ініціювання третього масиву, пошук добутку елементів третього масиву з індексом більше 100

**кінець**

*Крок 3*

**початок**

SIZE := 10

**init\_arrs(arr1, arr2, arr3, SIZE)**

**mult\_elements(arr3, SIZE)**

**кінець**

**Підпрограми:**

**init\_arrs(arr1, arr2, arr3, SIZE)**

temp := 0

**повторити для** i **від** 0 **до** SIZE

arr1[i] := 100 + i

arr2[i] := 110 – i \* i

**все повторити**

**повторити для** i **від** 0 **до** SIZE

**повторити для** z **від** 0 **до** SIZE

**якщо** arr1[i] == arr2[z]

**то**

arr3[temp] := arr1[i]

temp++

**все якщо**

**все повторити**

**все повторити**

**кінець**

**mult\_elements(arr3, SIZE)**

multiplication := 1

**повторити для** i **від** 0 **до** SIZE

**якщо** arr3[i] > 100

**то**

multiplication \*= arr3[i]

**все якщо**

**все повторити**

**якщо** multiplication == 1

**то**

multiplication := 0

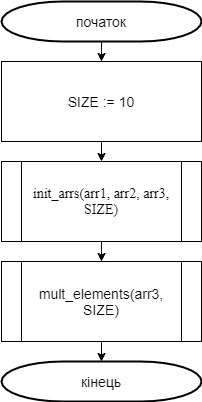
**все якщо**

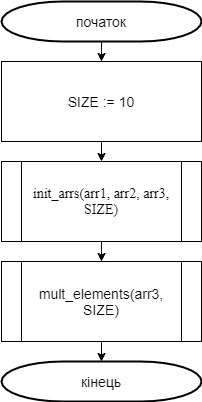
**вивести** multiplication

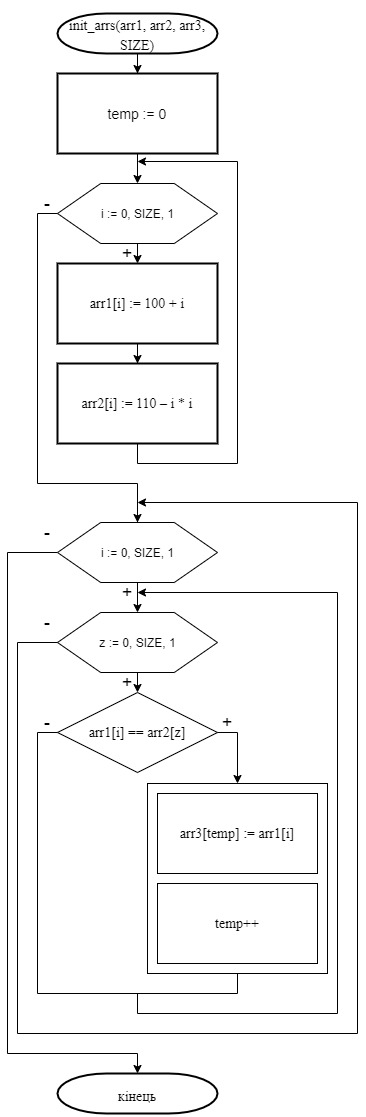
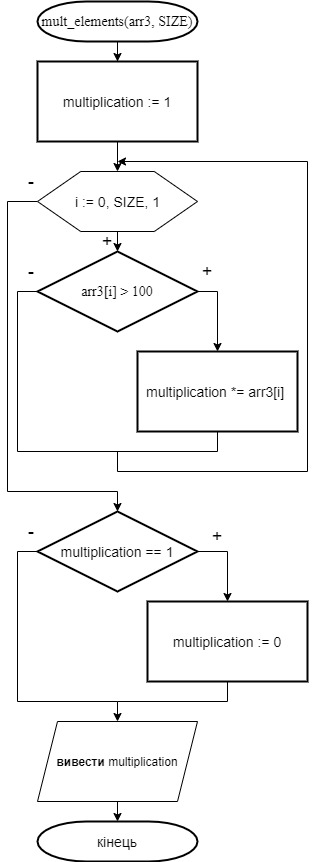
**кінець**

**Основна програма:**

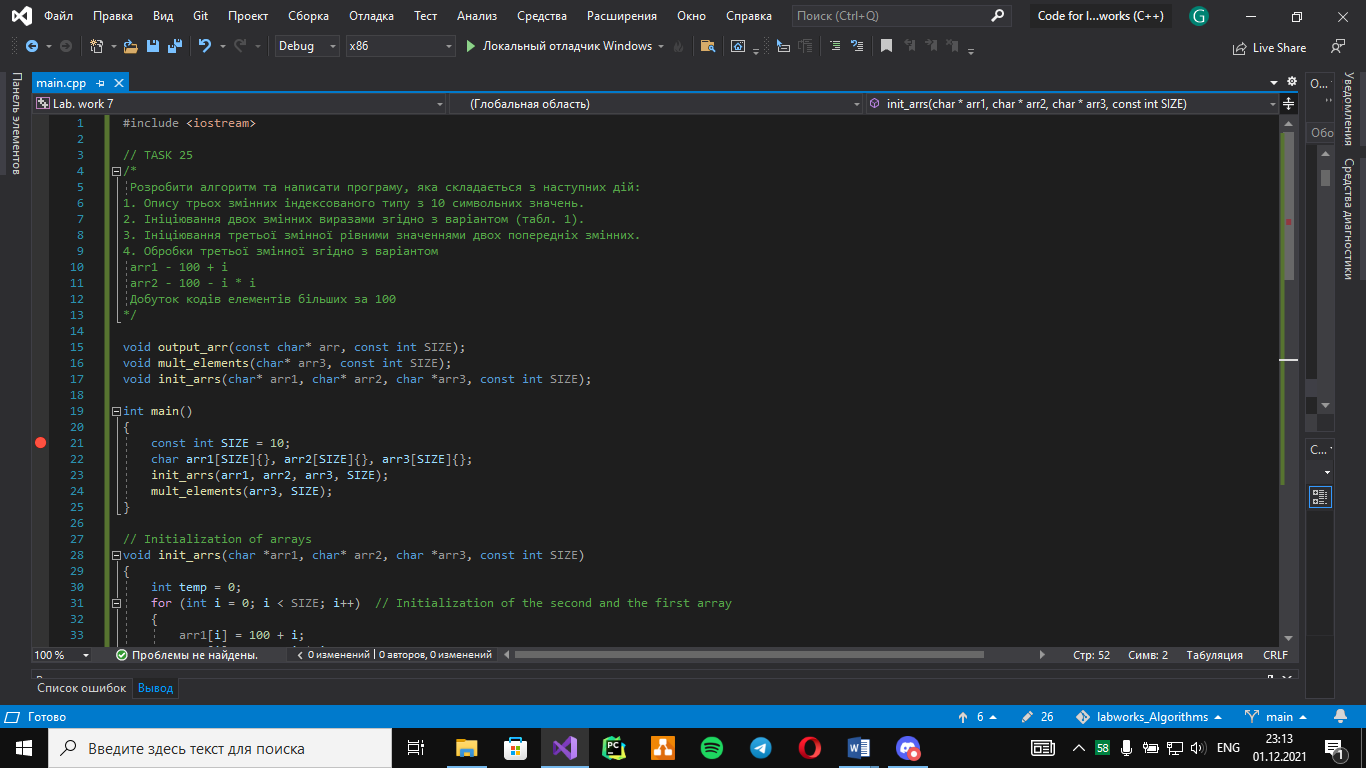
*Блок-схема*

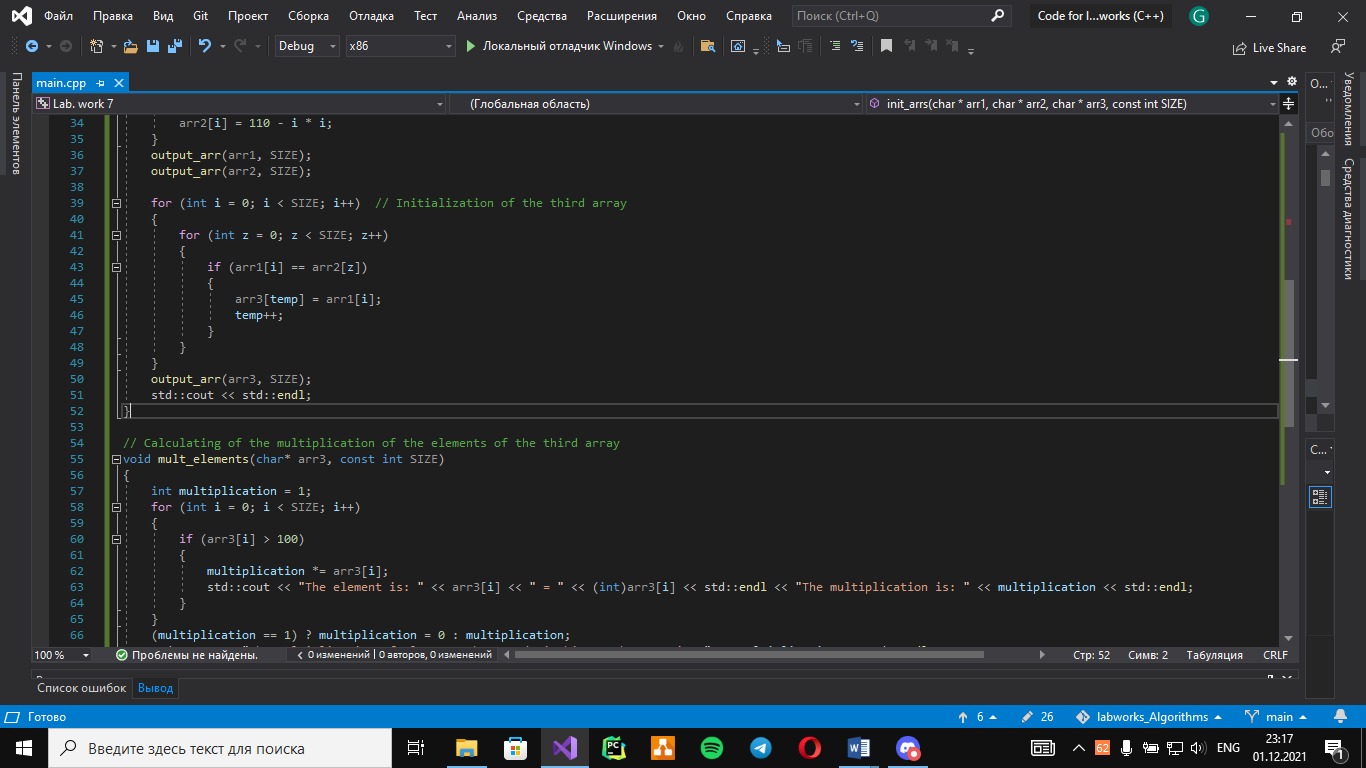


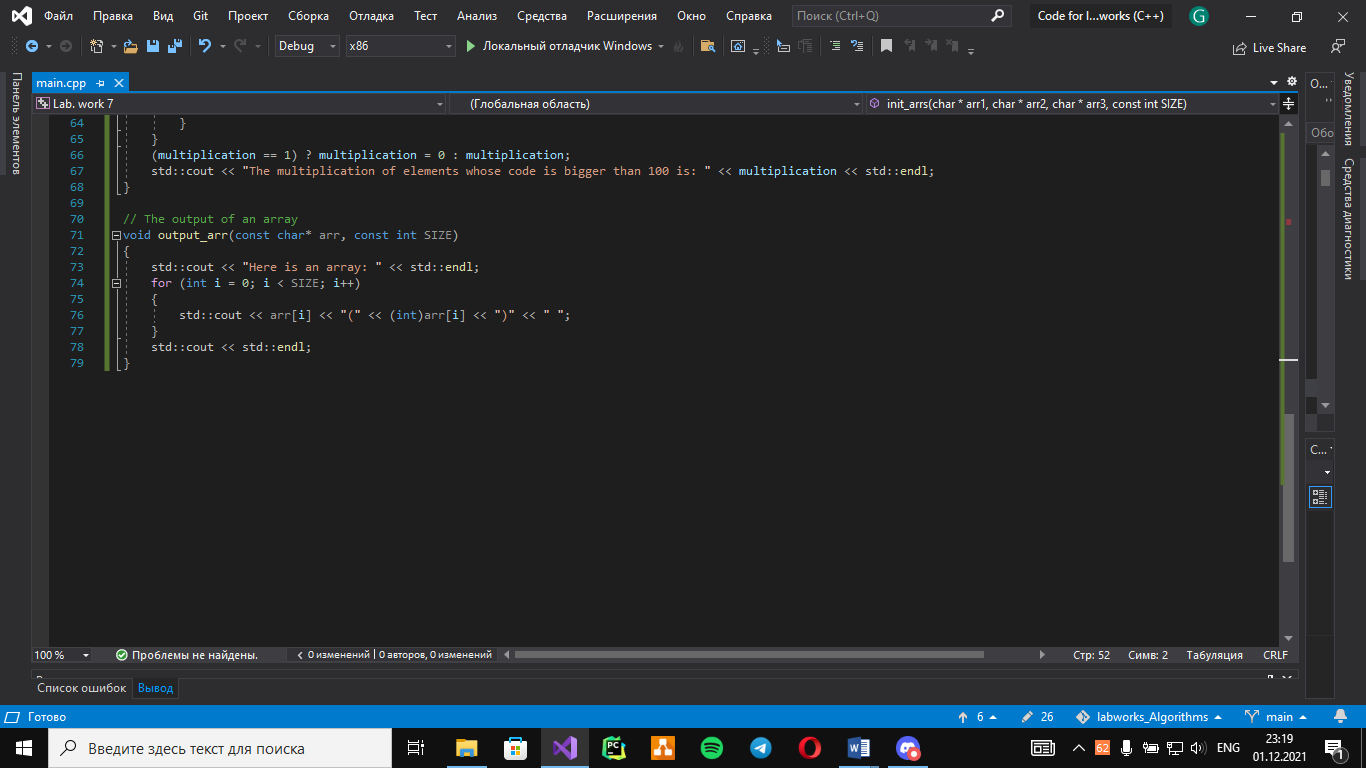
**Підпрограми:**

****

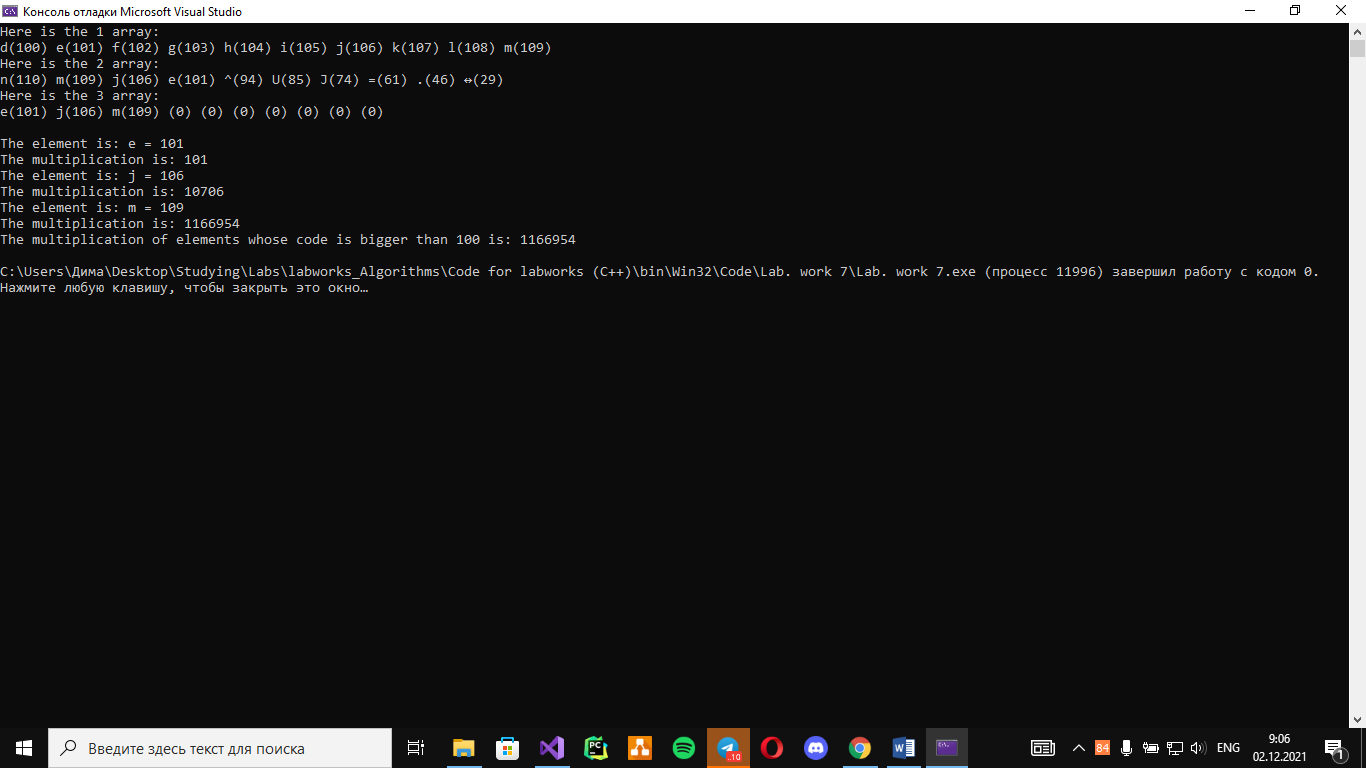
**5. К о д п р о г р а м и**







**5. Т е с т у в а н н я**



|  |  |
| --- | --- |
| **Блок** | **Дія 1** |
|  | **Початок** |
| **1** | **SIZE := 10** |
| **2** | **arr1[SIZE], arr2[SIZE], arr3[SIZE]** |
| **3** | **init\_arrs(arr1, arr2, arr3, SIZE)** |
| **4** | **temp := 0** |
| **5** | **i := 0** |
| **6** | **arr1[i] := 100 + i** |
| **7** | **arr2[i] := 110 – i \* i** |
| **…** | **…** |
| **166** | **arr3[temp] := arr1[i]** |
| **167** | **temp++** |
| **…** | **…** |
| **509** | **multiplication \*= arr3[i]** |
| **…** | **…** |
| **538** | **Вивести multiplication** |
|  | **Кінець** |

**6. В и с н о в о к**

В цій лабораторній роботі я дослідив методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набув практичних навичок їх використання під час складання програмних специфікацій. Мені довелося ініціювати три символьні масиви та працювати з їх елементами, вважаючи їх за числа або символи в залежності від конкретної ситуації та порівнювати їх, використовуючи певні методи пошуку.