Алгоритми та структури данних. Основи алгоритмів розгалуження.

*Додаток 1*

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 № 8 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійного пошуку в послідовностях»

Варіант 15

Виконав студент ІП-12, Кириченко Владислав Сергійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 202 1

**Лабораторна робота №** 8

**Назва роботи**: Дослідження алгоритмів пошуку та сортування

**Мета**:дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

**Варіант 15**

**Умова задачі***:*

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом (*Розмірність -* ***8х5****. Тип данних елементів -* ***Дійсний***).
2. Ініціювання змінної, що описана в п.1 даного завдання.

3. Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями, що обчислюються згідно з варіантом (*Із суми додатних значень елементів рядків двовимірного масиву. Відсортувати обміном* ***за зростанням***).

**Постановка задачі**:

*Початкові дані* **-** із початкових даних маємо лише розміз двовимірного масиву(***8х5***).

*Згенерувати двовимірний масив випадкових дійсних значень,виокремити з нього додатні елементи у окремий масив, та відсортувати цей масив за зростанням.*

*Результат* - одновимірний масив, відсортований **обміном за зростанням**.

**Побудова математичної моделі:**

Складемо таблицю змінних:

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Кількість рядків матриці | цілочисельний | ***i*** | Початкове значення |
| Кількість стовпців матриці | індексований | ***j*** | Початкове значення |
| Двовимірний масив | індексований | ***arr2D*** | Проміжкове значення |
| одновимірний масив | індексований | ***arr1D*** | Проміжкове значення |
| Лічильник | цілочисельний | ***i*** | Проміжкове значення |
| Лічильник | цілочисельний | ***j*** | Проміжкове значення |
| формальний параметр(перший масив) | індексований | ***arr1*** | Проміжкове значення |
| формальний параметр(другий масив) | індексований | ***arr2*** | Проміжкове значення |
| формальний параметр(третій масив) | індексований | ***newArr*** | Проміжкове значення |
| результат роботи підпрогрмамми | цілочисельний | ***prod*** | Результат(підпрограмма) |
| результат роботи прогрмамми | цілочисельний | ***product*** | Результат |
|  |  |  |  |

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

**Крок 1.** Визначимо основні дії.

**Крок 2.**Деталізація ініціалізації промжкових змінних ***size***, ***arrA,*** ***arrB,arrC***

**Крок 3.**Деталізація заповненя масивів ***arrA, arrB*** елементами

**Крок 4.**Деталізація заповнення масива ***arrC*** рівними елементами масивів ***arrA, arrB***

**Крок 5.**Деталізація знаходження значення змінної ***product***

*Псевдокод(основна прогрма):*

*Крок 1.*

**початок**

введення

ініціалізація промжкових змінних ***size***, ***arrA,*** ***arrB,arrC***

заповненя масивів ***arrA, arrB*** елементами

заповнення масива ***arrC*** рівними елементами масивів ***arrA, arrB***

знаходження значення змінної ***product***

виведення ***product***

**кінець**

*Крок 2.*

**початок**

введення

***size=10***

***arrA[size]***

***arrB[size]***

***arrC[size]***

заповненя масивів ***arrA, arrB*** елементами

заповнення масива ***arrC*** рівними елементами масивів ***arrA, arrB***

знаходження значення змінної ***product***

виведення ***product***

**кінець**

*Крок 3.*

**початок**

введення

***size=10***

***arrA[size]***

***arrB[size]***

***arrC[size]***

***fillArrays(arrA, arrB)***

заповнення масива ***arrC*** рівними елементами масивів ***arrA, arrB***

знаходження значення змінної ***product***

виведення ***product***

**кінець**

*Крок 4.*

**початок**

введення

***size=10***

***arrA[size]***

***arrB[size]***

***arrC[size]***

***fillArrays(arrA, arrB)***

***generateArr(arrA, arrB, arrC)***

знаходження значення змінної ***product***

виведення ***product***

**кінець**

*Крок 5.*

**початок**

введення

***size=10***

***arrA[size]***

***arrB[size]***

***arrC[size]***

***fillArrays(arrA, arrB)***

***generateArr(arrA, arrB, arrC)***

***product = calclulateProduct(arrC)***

виведення ***product***

**кінець**

*Псевдокод(підпрогрма - fillArrays):*

*Крок 1.*

**функція fillArrays(arr1[], arr2[])**

заповнення порожніх масивів елементами

**кінець**

*Крок 2.*

**функція fillArrays(arr1[], arr2[])**

**повторити**

**для *і* від** 0 **до *size* із кроком** 1

***arr1[i]*** = 43 - ***i***

***arr2[i]*** = 37 + ***і***

**все повторити**

**кінець**

*Псевдокод(підпрогрма - generateArr ):*

*Крок 1.*

**функція *generateArr(arr1, arr2,newArr,size)***

заповнення масива ***arrC*** рівними елементами масивів ***arrA, arrB***

**кінець**

Крок 2.

**функція *generateArr(arr1, arr2,newArr,size)***

**повторити**

**для *і* від** 0 **до *size* із кроком** 1

***newArr[i] = 0***

***j = 0***

**повторити**

**поки j < size та newArr[i] == 0**

**якщо arr1[i] == arr2[j]**

**то**

**newArr[i] = arr1[i]**

**все якщо**

***j ++***

**все повторити**

**все повторити**

**кінець**

*Псевдокод(підпрогрма - calclulateProduct ):*

*Крок 1.*

**функція** *calclulateProduct* ***(arr3[],size)***

обрахування значення змінної **prod**

перевірка чи були знайдені елементи, що задовольняють умову задачі

повернути **prod**

**кінець**

*Крок 2.*

**функція** *calclulateProduct* ***(newArr[],size)***

prod = 1

**повторити**

**для *і* від** 0 **до *size* із кроком** 1

**якщо *newArr[i]*** > 40

**то**

**prod \*= newArr[i]**

**все якщо**

**все повторити**

перевірка чи були знайдені елементи, що задовольняють умову задачі

повернути **prod**

**кінець**

*Крок 3.*

**функція** *calclulateProduct* ***(newArr[],size)***

prod = 1

**повторити**

**для *і* від** 0 **до *size* із кроком** 1

**якщо *newArr[i]*** > 40

**то**

**prod \*= newArr[i]**

**все якщо**

**все повторити**

**якщо**  **prod == 1**

**то**

**prod =0**

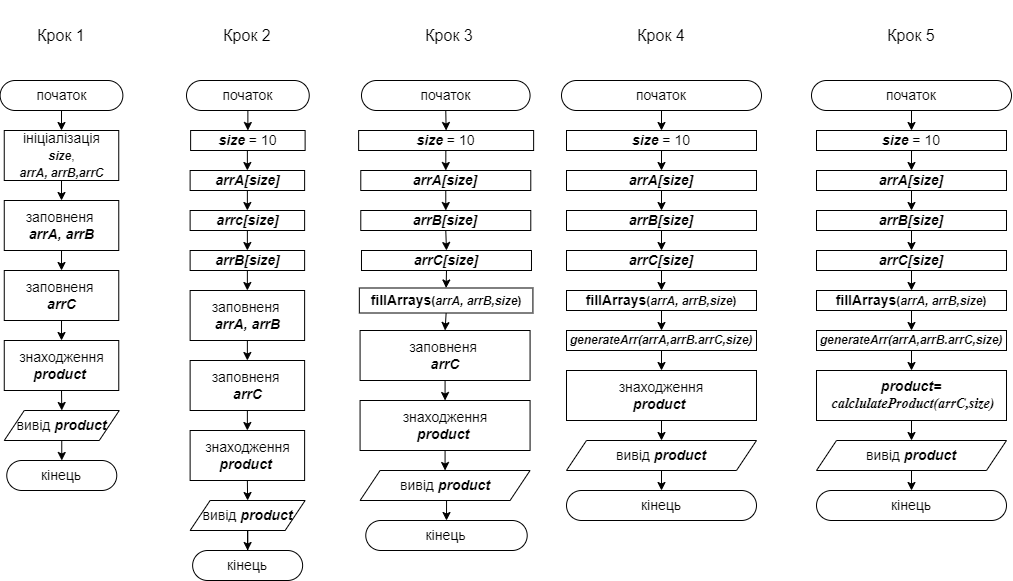
**все якщо**

поверненути **prod**

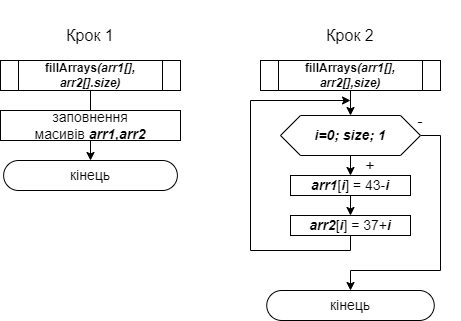
**кінець**

Блок схема:

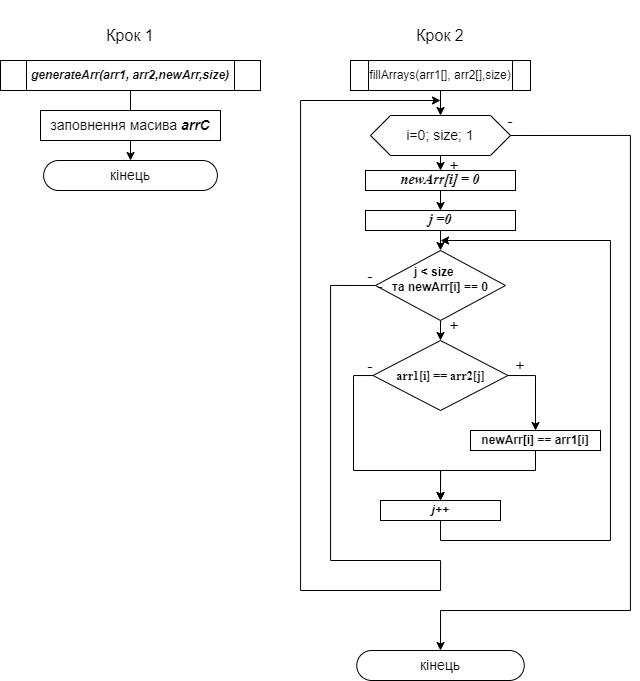
*Основна програма*



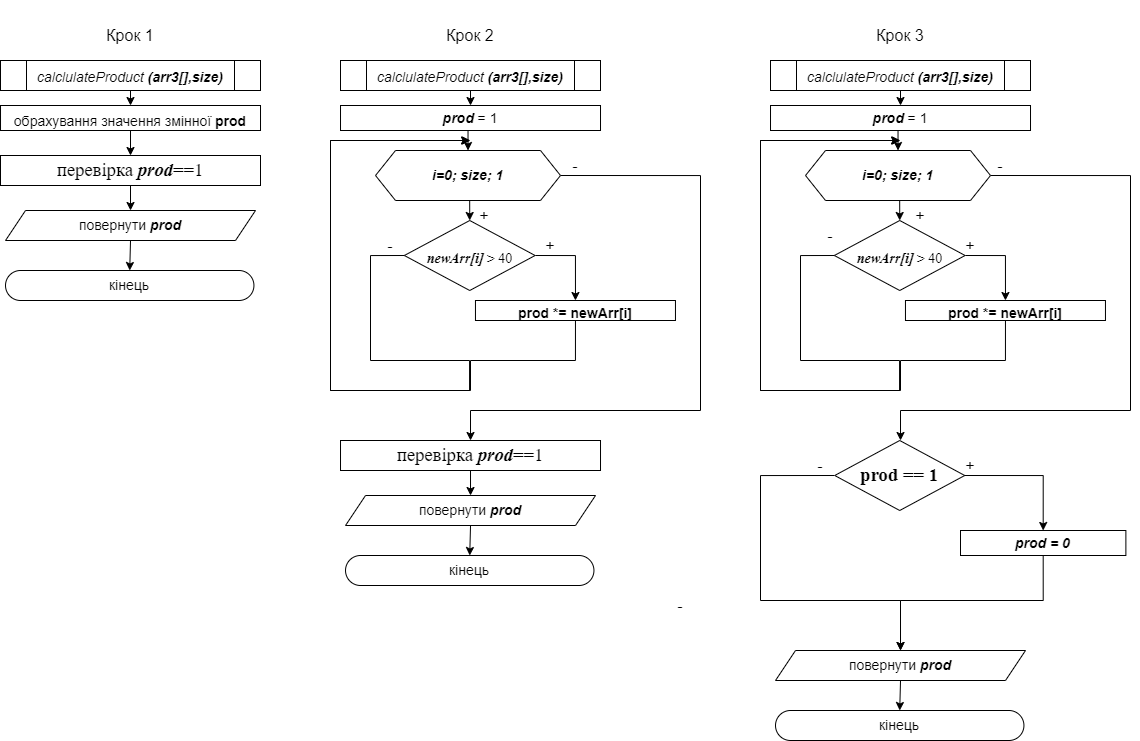
*Підпрограма(*fillArrays*)*



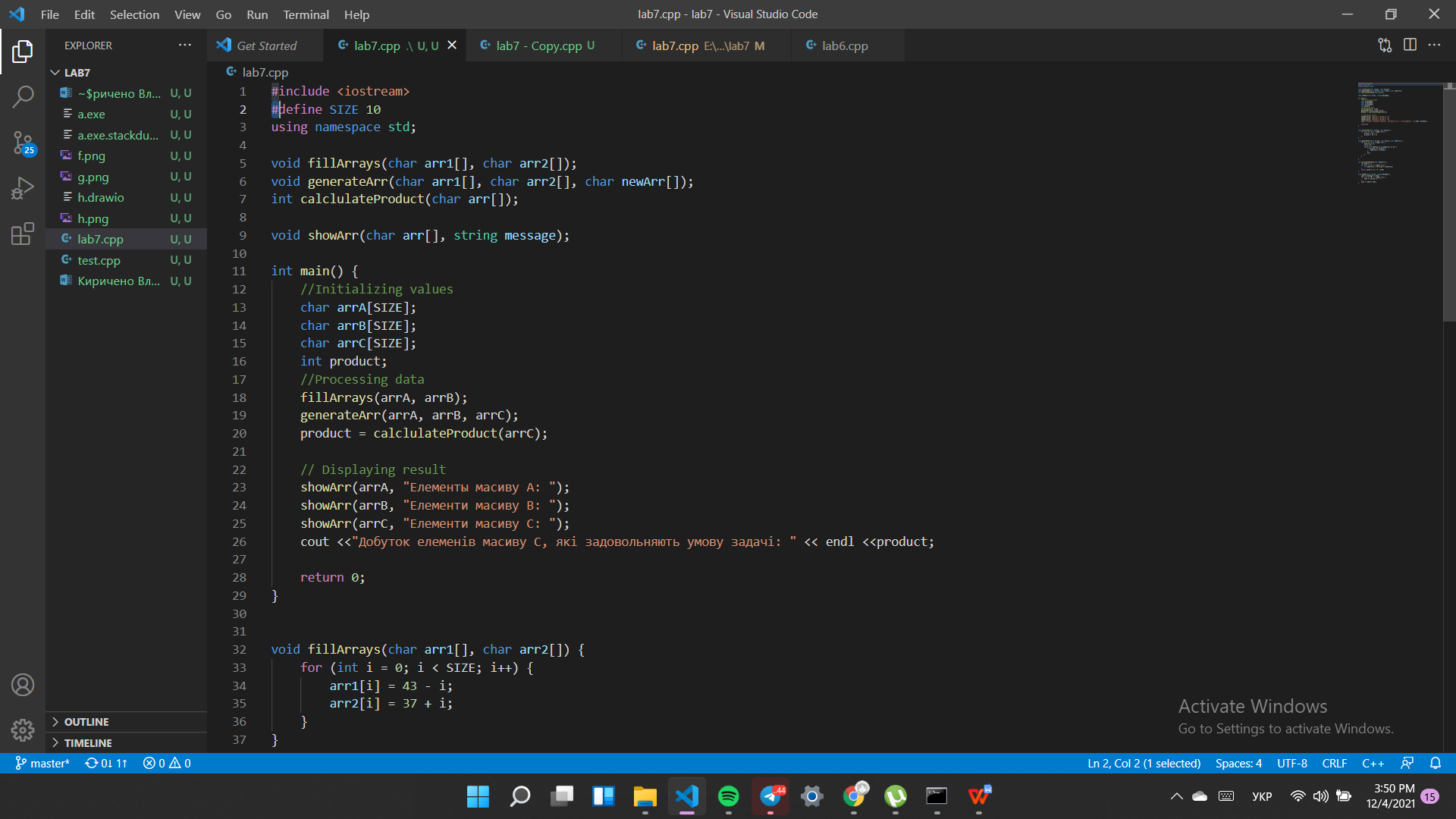
*Підпрограма(*generateArr*)*

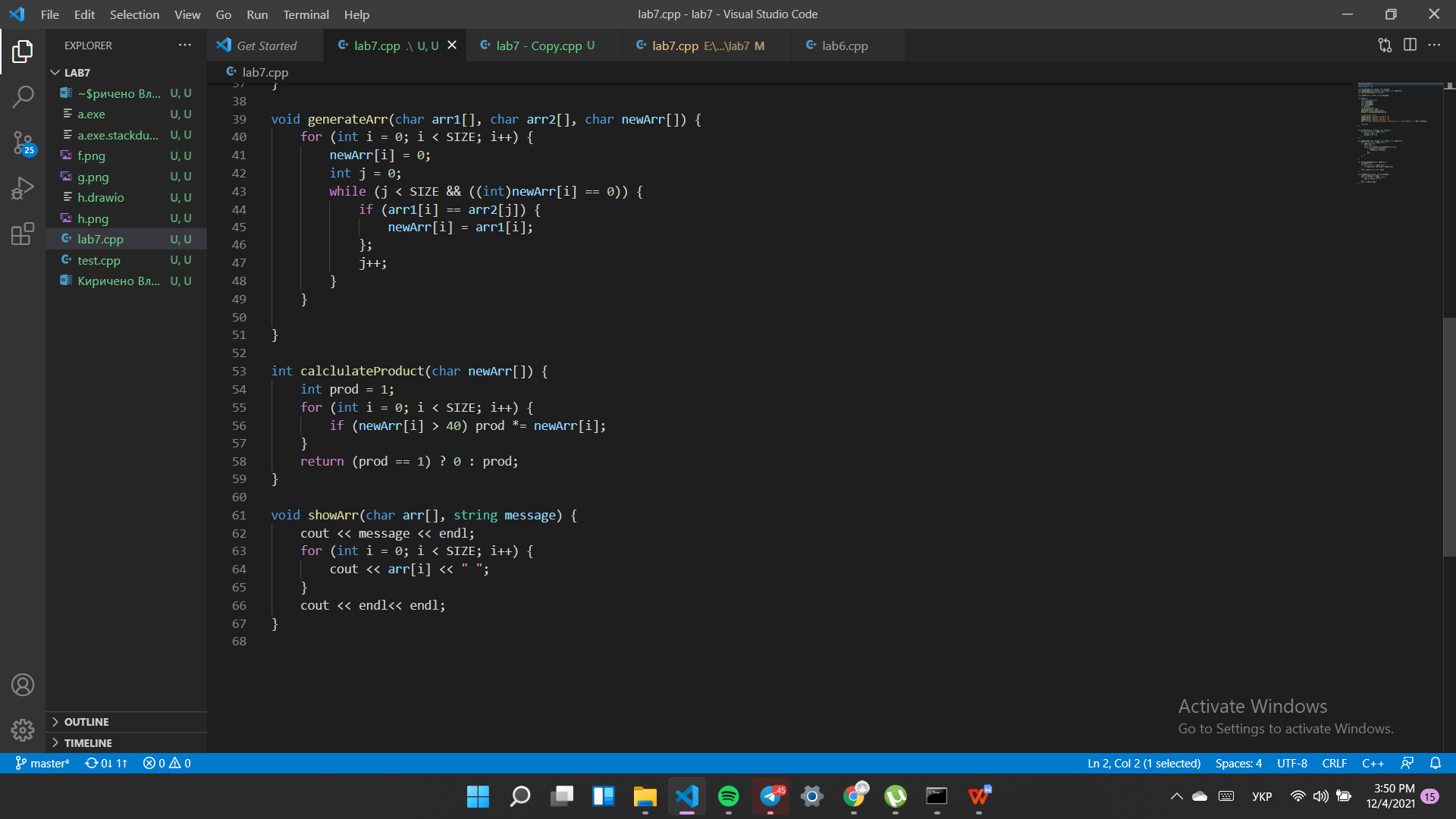


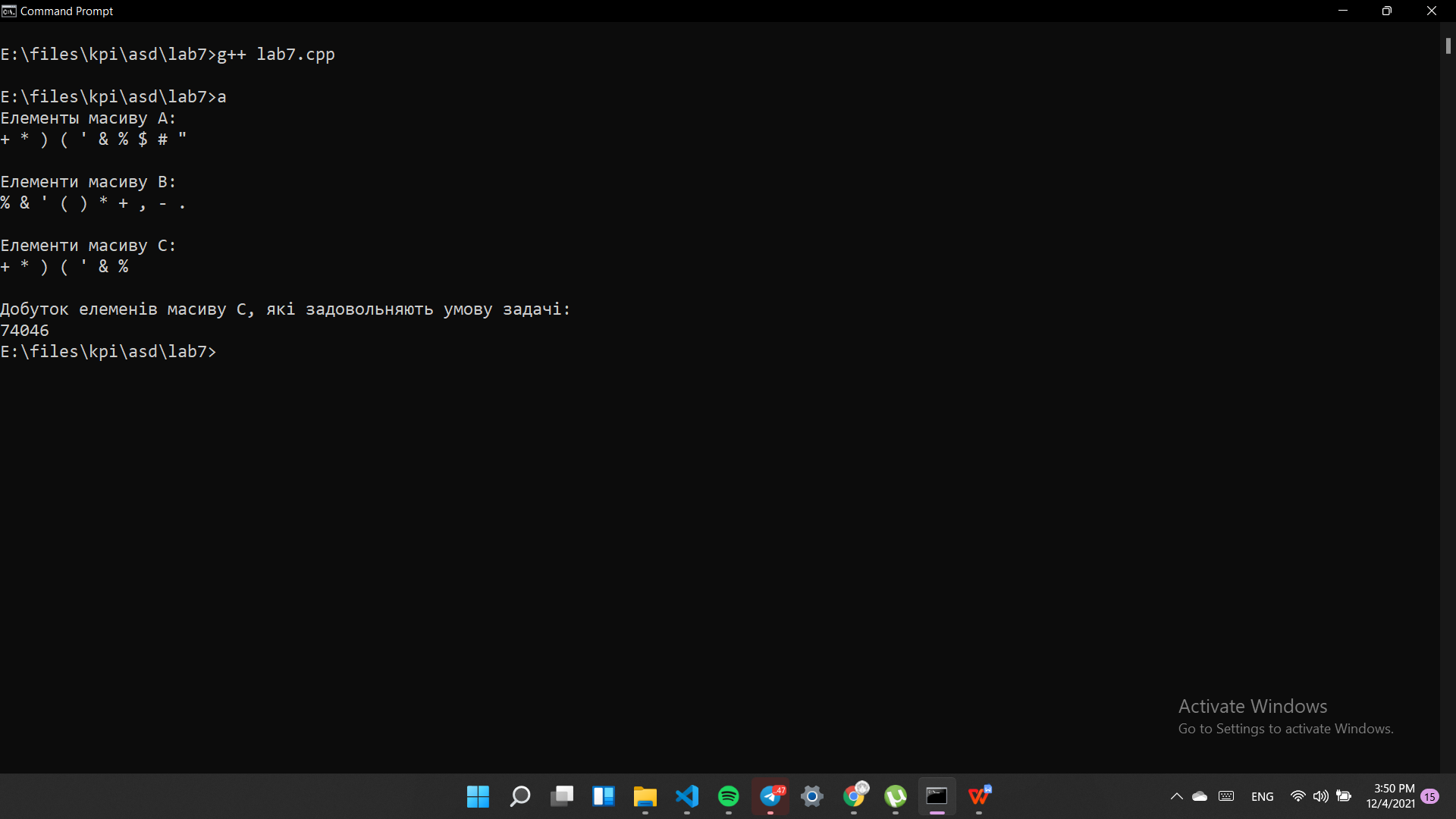
*Підпрограма(calclulateProduct )*



1. Код програми(С++)







**Висновок -** Було досліджено методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набуто практичних навичок їх використання під час складання програмних специфікацій.

Декомпозовано задачу на 3 етапи:

1. Генерація перших двох масивів.

2. Генерація третього масиву з рівних елементами перших двух

3.Знаходження добутку усіх кодів елементів третього масива, що більше за 40.