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 16

Виконав студент ІП-11 Кузьменков Дмитро Олегович

(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова. О. П

( 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.

# 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 (табл. 1).
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної згідно з варіантом.



**Постановка задачі:**

Почнемо з побудови арифметичного циклу, що буде знаходити перші 10 елементів 1-го та 2-го масиву ініційованих згідно з заданими виразами. Далі будуємо складний цикл, що знаходитиме елементи цих масивів що співпадають та рахуватиме їх кількість (**y**). Задаємо розмірність **y** 3-му масиву (динамічний масив). Далі за допомогою того ж циклу знаходимо усі елементи 3-го масиву. Наступний цикл знаходитиме найбільші елементи масиву. Знаходимо їх та сумуємо. Виводимо 1-й, 2-й, 3-й масиви та суму двох максимальних елементів третього.

# Побудова математичної моделі. Складемо таблицю імен змінних:

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Перший масив | Масив з натуральними елементами | arr1[] | Проміжне дане, результат |
| Другий масив | Масив з натуральними елементами | arr2[] | Проміжне дане, результат |
| Третій масив | Масив з натуральними елементами | arr3[] | Проміжне дане, результат |
| Лічильник, показник індексу | Натуральне | i | Початкове дане, проміжне дане |
| Лічильник | Натуральне | j | Початкове дане, проміжне дане |
| Розмірність третього  масиву | Натуральне | y | Початкове дане, проміжне дане |
| Показник індексу третього масиву | Натуральне | yi | Початкове дане, проміжне дане |
| Максимальний елемент третього масиву | Натуральне | Max | Проміжне дане |
| Другий максимальний елемент третього масиву | Натуральний | Smax | Проміжне дане |
| Сума двох максимальних елементів третього масиву | Натуральний | arr3MaxSum | Результат |

Крок 1. Знаходимо arr1 та arr2.

Крок 2. Знаходимо кількість елементів arr3 та задаємо його розмірність.

Крок 3. Знаходимо усі елементи arr3.

Крок 4. Знаходимо два максимальні елементи arr3 та їх суму.

Виводимо arr1, arr2, arr3, arr3MaxSum.

1. **Псевдокод:**

# Початок

1) Знаходимо arr1 та arr2

2) Знаходимо кількість елементів arr3 та задаємо його розмірність

3) Знаходимо усі елементи arr3

4) Знаходимо два максимальні елементи arr3 та їх суму

Виводимо arr1, arr2, arr3, arr3MaxSum

**Кінець**

*Крок1*

# Початок

1) y:= 0; yi:=0;

**Поки** i:=0; i<=9; i++;

**Повторити**

arr1[i]:= 115 + i;

arr2[i]:= 125 – 2\*i;

**Все повторити**

2) Знаходимо кількість елементів arr3 та задаємо його розмірність

3) Знаходимо усі елементи arr3

4) Знаходимо два максимальні елементи arr3 та їх суму

Виводимо arr1, arr2, arr3, arr3MaxSum

**Кінець**

*Крок2*

# Початок

1) y:= 0; yi:=0;

**Поки** i:=0; i<=9; i++

**Повторити**

arr1[i]:= 115 + i;

arr2[i]:= 125 – 2\*i;

**Все повторити**

2) **Поки** i:=0; i<=9; i++

**Повторити**

**Поки** j:=0; j<=9; j++

**Повторити**

**Якщо** arr1[i] == arr2[j]

**то** y++;

**Все повторити**

**Все повторити**

int \*arr3 = new int[y];

3) Знаходимо усі елементи arr3

4) Знаходимо два максимальні елементи arr3 та їх суму

Виводимо arr1, arr2, arr3, arr3MaxSum

**Кінець**

*Крок3*

# Початок

1) y:= 0; yi:=0;

**Поки** i:=0; i<=9; i++

**Повторити**

arr1[i]:= 115 + i;

arr2[i]:= 125 – 2\*i;

**Все повторити**

2) **Поки** i:=0; i<=9; i++

**Повторити**

**Поки** j:=0; j<=9; j++

**Повторити**

**Якщо** arr1[i] == arr2[j]

**то** y++;

**Все повторити**

**Все повторити**

int \*arr3 = new int[y];

3) **Поки** i:=0; i<=9; i++

**Повторити**

**Поки** j:=0; j<=9; j++

**Повторити**

**Якщо** arr1[i] == arr2[j]

**то** arr3[yi] = arr1[i];

yi++;

**Все повторити**

**Все повторити**

4) Знаходимо два максимальні елементи arr3 та їх суму

Виводимо arr1, arr2, arr3, arr3MaxSum

**Кінець**

*Крок4*

# Початок

1) y:= 0; yi:=0;

**Поки** i:=0; i<=9; i++

**Повторити**

arr1[i]:= 115 + i;

arr2[i]:= 125 – 2\*i;

**Все повторити**

2) **Поки** i:=0; i<=9; i++

**Повторити**

**Поки** j:=0; j<=9; j++

**Повторити**

**Якщо** arr1[i] == arr2[j]

**то** y++;

**Все повторити**

**Все повторити**

int \*arr3 = new int[y];

3) **Поки** i:=0; i<=9; i++

**Повторити**

**Поки** j:=0; j<=9; j++

**Повторити**

**Якщо** arr1[i] == arr2[j]

**то** arr3[yi] = arr1[i];

yi++;

**Все повторити**

**Все повторити**

4) max:= arr3[1];

Smax:= arr3[0];

**Поки** i = 2; i<y; y++

**Повторити**

**Якщо** arr3[i] > max

**то** Smax:= max;

max:= arr3[i]

**Все повторити**

arr3MaxSum:= max + Smax;

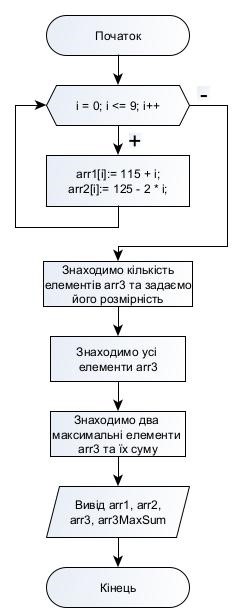
Виводимо arr1, arr2, arr3, arr3MaxSum

**Кінець**

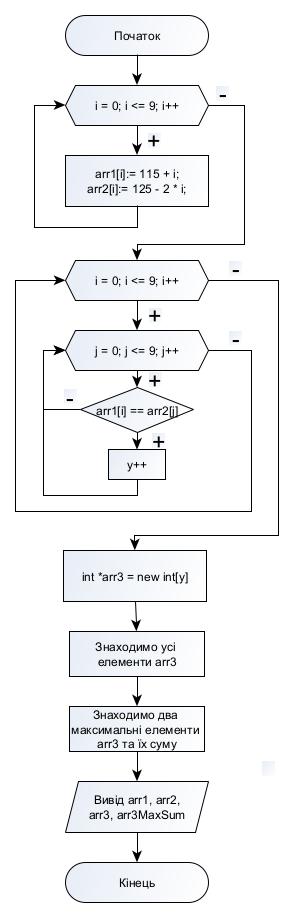
1. **Блок-схема**



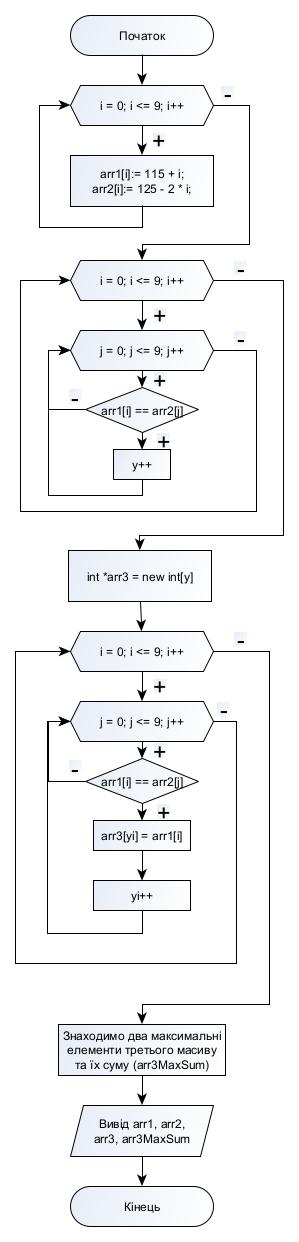
*Крок1*



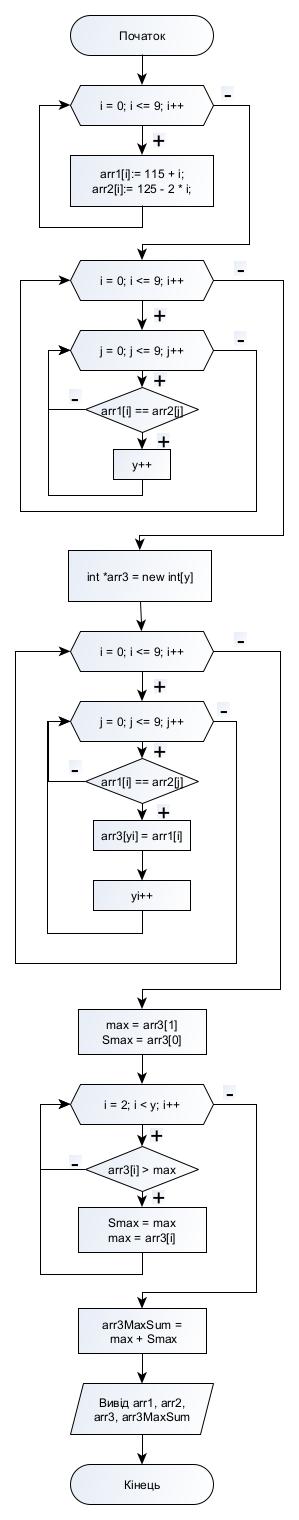
*Крок2*



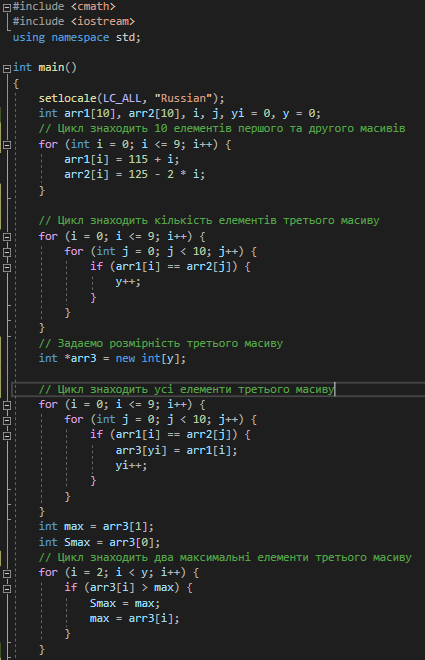
*Крок3*

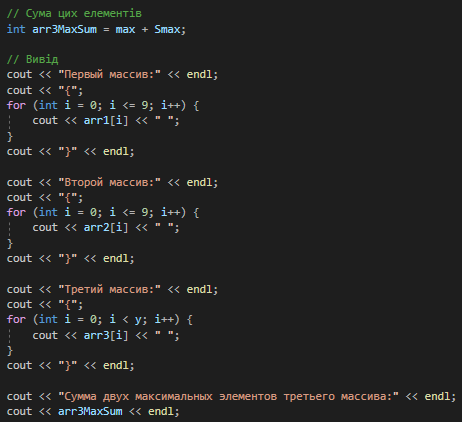


*Крок4*

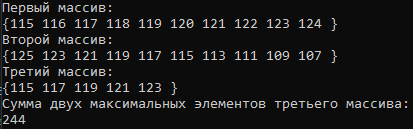


**4. Код (C++):**

****

****

**Результат:**

****

**Висновок:** Виконуючи лабораторну роботу, я дослідив методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набув практичних навичок їх використання під час складання програмних специфікацій. Побудував алгоритм та написав програму знаходження двох масивів ініційованих згідно з виразами:



та третього масиву, що складається з однакових елементів перших двох, а також суми двох його максимальних елеменів.