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.

Основи алгоритмізації»

«Дослідження лінійного пошуку в послідовностях»

Варіант 10

Виконав студент ІП-12 Горобець Олексій Сергійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 202 1

**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 10**

***Задача 1.10***.

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 для 1 масиву 100 + i, для другого 110 – i\*i.
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Знайти добуток кодів елементів.

***Постановка задачі*.** Результатом розв’язку є число рівне добутку кодів елементів у третьому масиві. Для визначення цього числа нам необхідно заповнити третій масив. Для заповнення третього масиву нам необхідно порівняти задана формула для визначення елементів першого масиву, а саме 100 + i, та другого 110 – i\*i. Інших початкових даних для розв’язку не потрібно.

***Математична побудова.*** Складемо таблицю змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| **Масив визначений за формулою 100+і** | Індексований | arr1 | Проміжне значення |
| **Масив визначений за формулою 100-і\*і** | Індексований | arr2 | Проміжне значення |
| **Третій масив** | Індексований | arr3 | Проміжне значення |
| **Число k** | Цілий | k | Лічильник |
| **Число SIZE** | Цілий | SIZE | Початкове дане |
| **Число res** | Цілий | res | Результат |

Таким чином, математичне формулювання задачі зводиться до розрахування елементів першого масиву arr1 та arr2 за заданими формулами. Та запису в третій масив спільні елементи для першого та другого масиву.

***Розв’язання:***

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

**Крок 1**. Визначимо основні дії.

**Крок 2.** Деталізуємо ініціалізації початкових змінних ***size***, ***arr1,*** ***arr2, arr3***

**Крок 3**. Деталізуємо заповнення масивів ***arr1, arr2***

**Крок 4.** Деталізуємо заповнення масива ***arr3*** рівними елементами масивів ***arr1, arr2***

***Крок 5.*** Деталізуємо знаходження значення змінної ***res***

***Псевдокод***

*крок 1*

**початок**

Ініціалізації початкових змінних ***size***, ***arr1[size],*** ***arr2[size], arr3[size]***

Деталізуємо дію заповненя масивів ***arr1, arr2***

Деталізуємо дію заповнення масива ***arr3*** рівними елементами масивів ***arr1, arr2***

Деталізуємо дію знаходження значення змінної ***res***

**Кінець**

*крок 2*

**початок**

***size = 10***

***arr1[size]***

***arr2[size]***

***arr3[size]***

***fillArrays(arr1, arr2)***

Деталізуємо дію заповнення масива ***arr3*** рівними елементами масивів ***arr1, arr2***

Деталізуємо дію знаходження значення змінної ***res***

**Кінець**

**початок функції fillArrays** (arr1, arr2)

Деталізуємо дію заповненя масивів ***arr1, arr2***

**кінець функції fillArrays**

*крок 3*

**початок**

***size = 10***

***arr1[size]***

***arr2[size]***

***arr3[size]***

***fillArrays(arr1, arr2)***

***thirdArrFill(arr1, arr2, arr3)***

Деталізуємо дію знаходження значення змінної ***res***

**кінець**

**початок функції fillArrays** (arr1, arr2)

**повторити**

**для і від 0 до SIZE**

arr1[i] := 100 + i;

arr2[i] := 100 - i \* i;

**все повторити**

**кінець функції fillArrays**

**початок функції thirdArrFill** (arr1, arr2)

Деталізуємо дію заповнення масива ***arr3*** рівними елементами масивів ***arr1, arr2***

**кінець функції thirdArrFill**

Крок 4

**початок**

***size = 10***

***arr1[size]***

***arr2[size]***

***arr3[size]***

***fillArrays(arr1, arr2)***

***thirdArrFill(arr1, arr2, arr3)***

Деталізуємо дію знаходження значення змінної ***res***

**кінець**

**початок функції fillArrays** (arr1, arr2)

**повторити**

**для і від 0 до SIZE**

arr1[i] := 100 + i;

arr2[i] := 100 - i \* i;

**все повторити**

**кінець функції fillArrays**

**початок функції thirdArrFill** (arr1, arr2)

k:=0

**повторити**

**для і від 0 до SIZE**

**повторити**

**для j від 0 до SIZE**

**якщо** arr1[i] == arr2[j]

**то**

arr3[k]:= arr1[i]

k:= k +1

**все якщо**

**все повторити**

**все повторити**

**кінець функції thirdArrFill**

Крок 5

**початок**

***size = 10***

***arr1[size]***

***arr2[size]***

***arr3[size]***

***fillArrays(arr1, arr2)***

***thirdArrFill(arr1, arr2, arr3)***

**countRes(arr3)**

**кінець**

**початок функції fillArrays** (arr1, arr2)

**повторити**

**для і від 0 до SIZE**

arr1[i] := 100 + i;

arr2[i] := 100 - i \* i;

**все повторити**

**кінець функції fillArrays**

**початок функції thirdArrFill** (arr1, arr2)

k:=0

**повторити**

**для і від 0 до SIZE**

**повторити**

**для j від 0 до SIZE**

**якщо** arr1[i] == arr2[j]

**то**

arr3[k]:= arr1[i]

k:= k +1

**все якщо**

**все повторити**

**все повторити**

**кінець функції thirdArrFill**

**початок функції countRes** (arr3)

res:=1

**повторити**

**для і від 0 до SIZE**

**якщо** arr3[i] != 0

**то**

res:=res \* arr3[i]

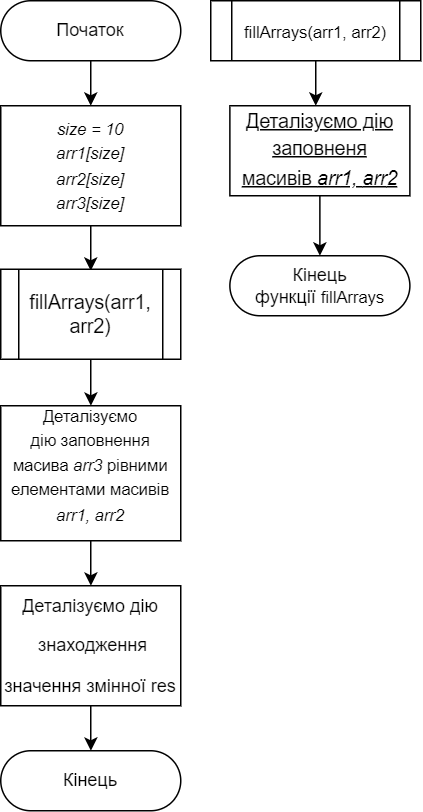
**все якщо**

**все повторити**

**кінець функції thirdArrFill**

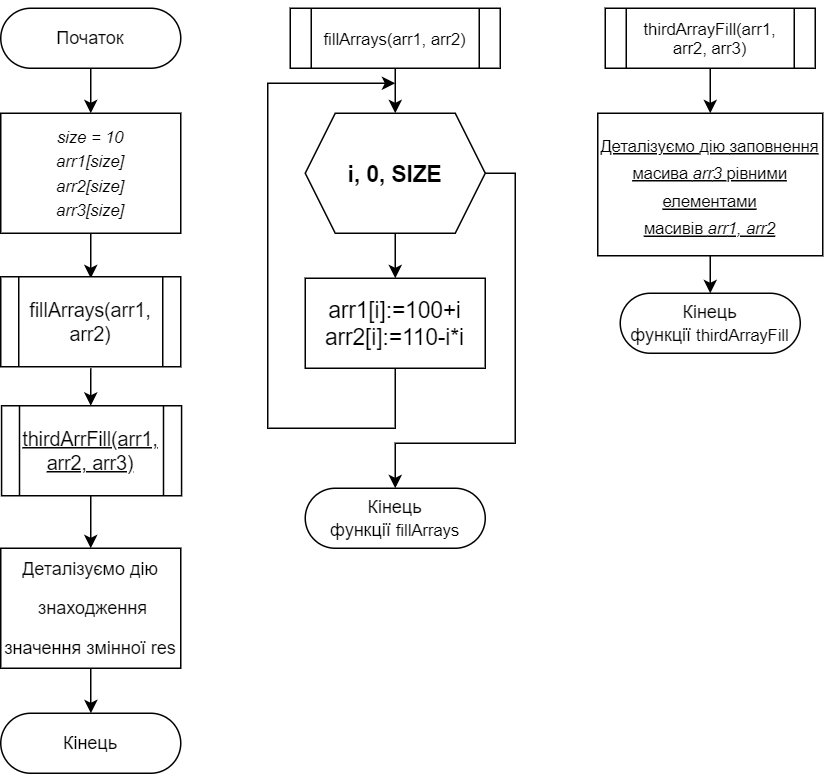
***Блок-схема***

***Крок 1 Крок 2***

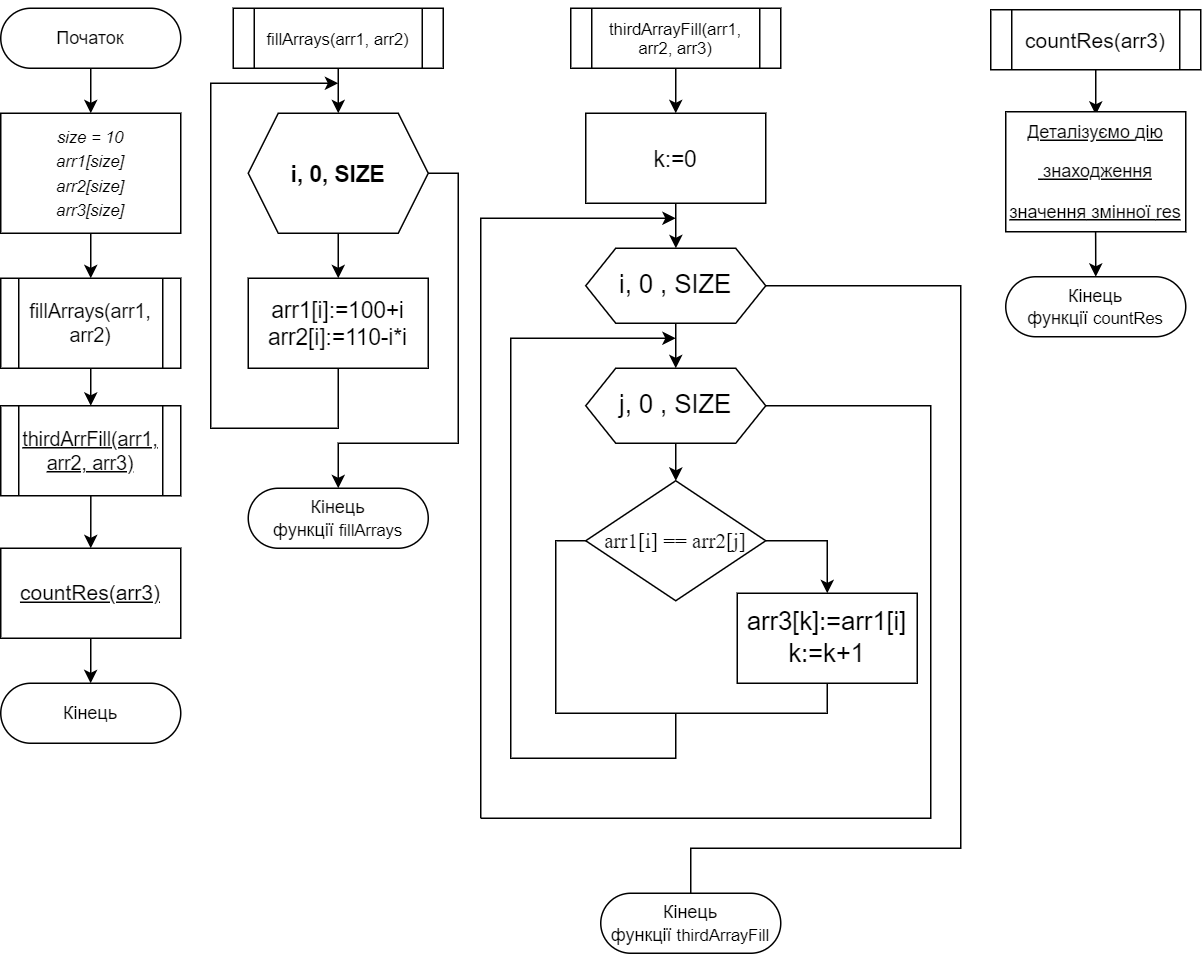
**

**

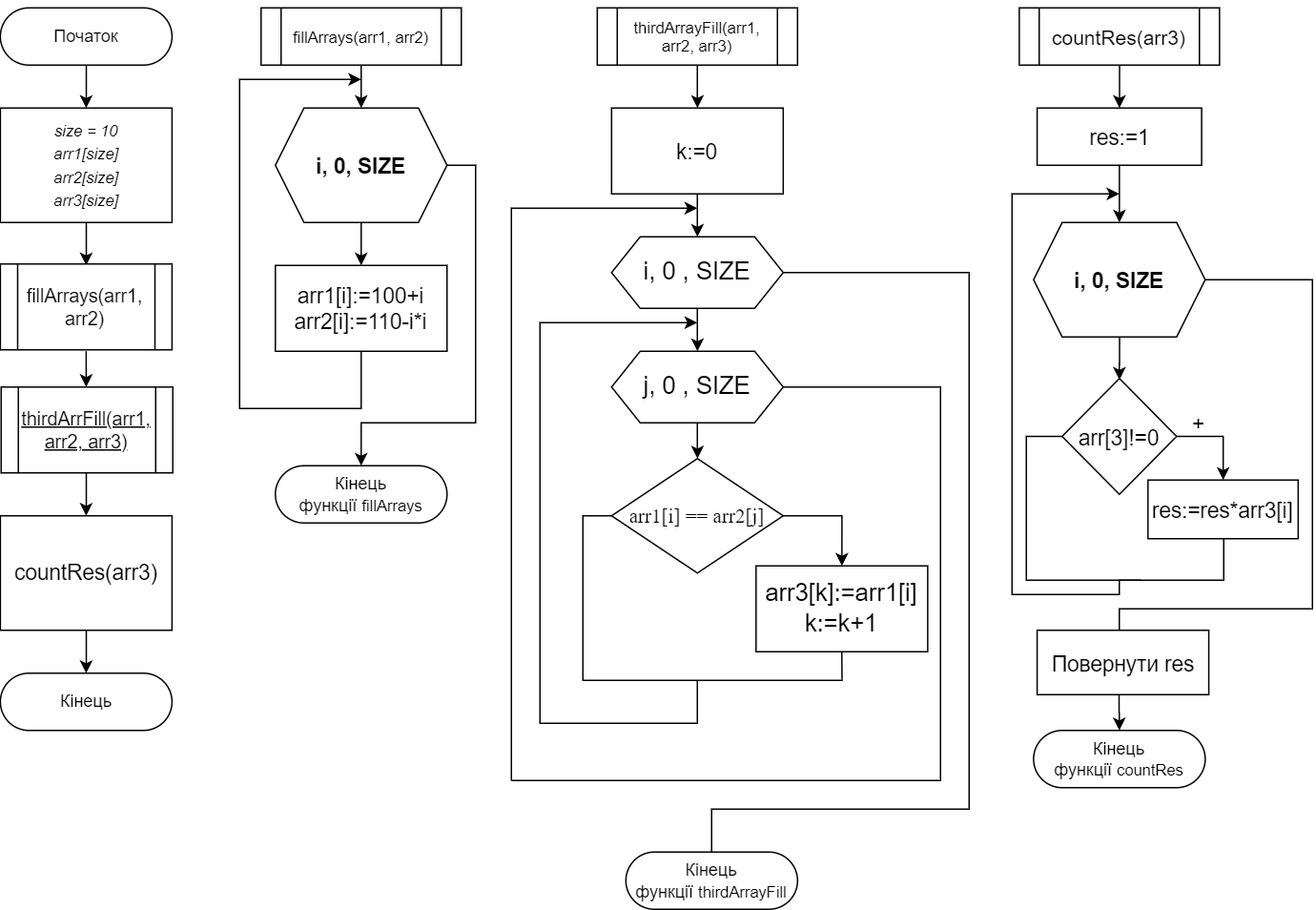
***Крок 3***

******

***Крок 4***

******

***Крок 5***

******

***Код програми c++***

#include <ctime>

#include <iostream>

using namespace std;

const int SIZE = 10;

void fillArrays(char\* , char\*);

void thirdArrFill(char\*, char\*, char\*);

int countRes(char\*);

void charArrOutput(char\*, string);

int main()

{

setlocale(LC\_ALL, "rus");

char arr1[SIZE], arr2[SIZE], arr3[SIZE] = {0};

fillArrays(arr1, arr2);

thirdArrFill(arr1, arr2, arr3);

charArrOutput(arr1, "First arr elements:");

charArrOutput(arr2, "Second arr elements:");

charArrOutput(arr3, "Third arr elements:");

cout << "Result: " << countRes(arr3)<<endl;

}

void charArrOutput(char\* arr, string msg) {

cout << msg << endl;

for (int i = 0; i < SIZE; i++) {

cout << arr[i] << " ";

}

cout << endl << endl;

}

void fillArrays(char\* arr1, char\* arr2) {

for (int i = 0; i < SIZE; i++) {

arr1[i] = 100 + i;

arr2[i] = 110 - i \* i;

}

}

void thirdArrFill(char\* arr1, char\* arr2, char\* arr3) {

int k = 0;

for (int i = 0; i < SIZE; i++) {

for (int j = 0; j < SIZE; j++) {

if (arr1[i] == arr2[j]) {

arr3[k] = arr1[i];

k++;

}

}

}

}

int countRes(char\* arr3) {

int res = 1;

for (int i = 0; i<SIZE; i++) {

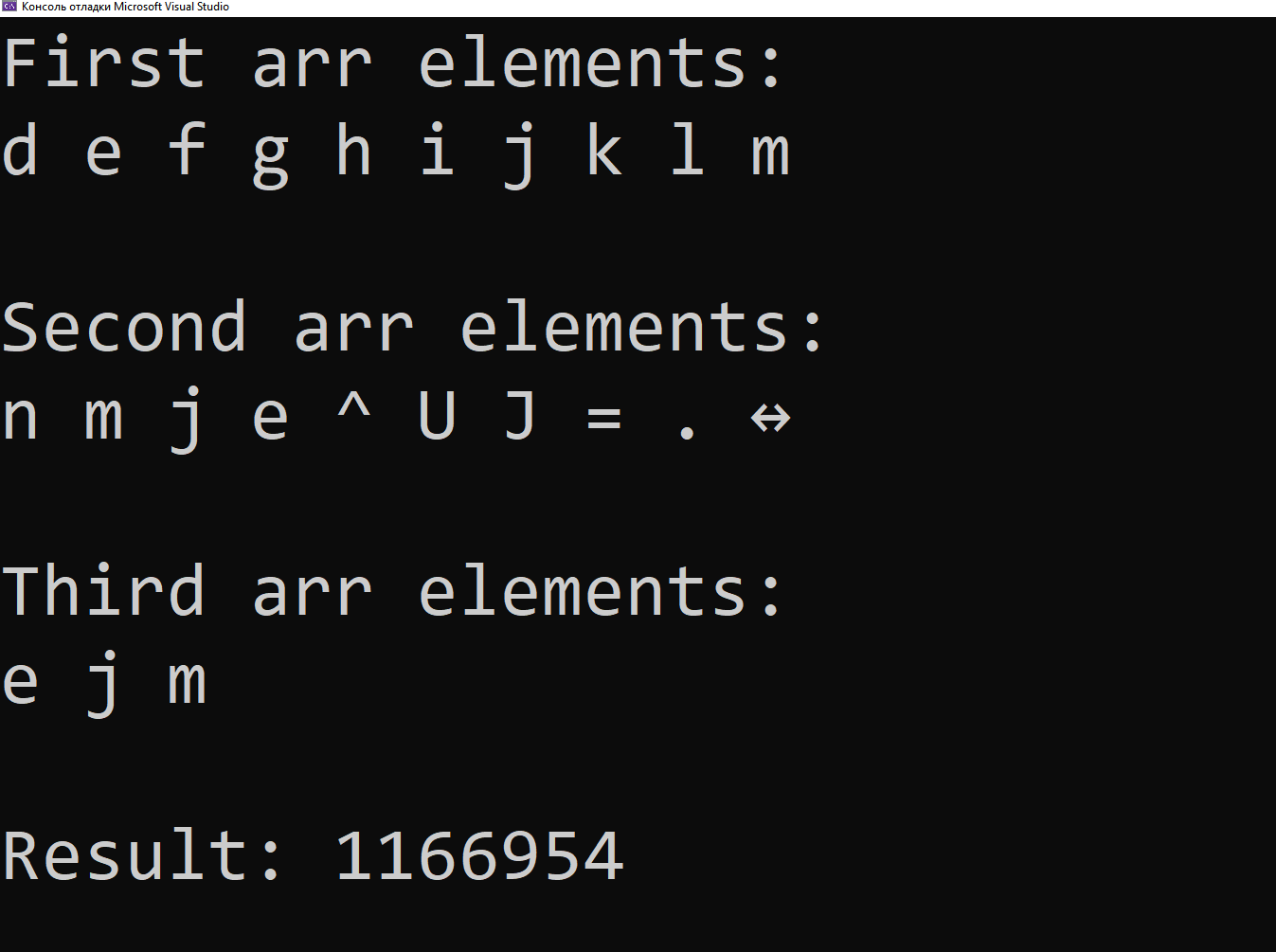
if(arr3[i]!=0)

res \*= int(arr3[i]);

}

return res;

}



***Висновок***

Протягом лабораторної роботи було досліджено методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набуто практичних навичок їх використання під час складання програмних специфікацій. Особливістю роботи було:

1. Розробка функція запису значень у масив за заданою формулою в залежності від індекса елемента.
2. Розробка підпрограми для пошуку спільних значень в двох масивах-рядках за допомогою двох вкладених арифметичних циклів та умовного оператора, та подальшого запису цих значень у третій масив.
3. Розробка підпрограми для знаходження добутку табличних кодів ASCII елементів масиву.