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 № 6 з дисципліни «Алгоритми та структури даних-1. Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант 34

Виконав студент ІП-12 Шоман Данило Володимирович

(шифр, прізвище, ім'я, по батькові)

Перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 6**

**Дослідження рекурсивних алгоритмів**

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

**Варіант 34**

**Умова:**

34. 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 **(44 + 2 \* i; 55 – 2 \* i).**

3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.

4. Обробки третьої змінної згідно з варіантом. **(Кількість елементів, коди яких діляться на 3)**

**Постановка задачі:** Введемо змінні індексованого типу: arrA, arrB, arrC, відповідно 1, 2 і 3-ій масиви, змінну size = 10 – розмір. Результатом розв’язку є значення змінної counter – кількості елементів 3-го масиву, що діляться на 3. Для визначення результату повинні бути задані масиви arrA і arrB. Функції: % - знаходження остачі від ділення.

**Математична модель:**

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Ім’я | Тип | Призначення |
| Розмір масивів | size | Цілий | Початкове дане |
| Масив з формулою елемента 44 + 2 \* і | arrA | Індексований | Проміжне значення |
| Масив з формулою елемента 55 – 2 \* i | arrB | Індексований | Проміжне значення |
| Масив рівних елементів перший двух масивів | arrC | Індексований | Проміжне значення |
| Лічильник | цілочисельний | ***i*** | Проміжне значення |
| Лічильник | цілочисельний | ***j*** | Проміжне значення |
| формальний параметр(перший масив) | індексований | ***arr1*** | Проміжне значення |
| формальний параметр(другий масив) | індексований | ***arr2*** | Проміжне значення |
| формальний параметр(третій масив) | індексований | ***arr3*** | Проміжне значення |
| результат роботи прогрмамми | цілочисельний | ***count*** | Результат |

**План:**

**Крок 1.** Визначимо основні дії.

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

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

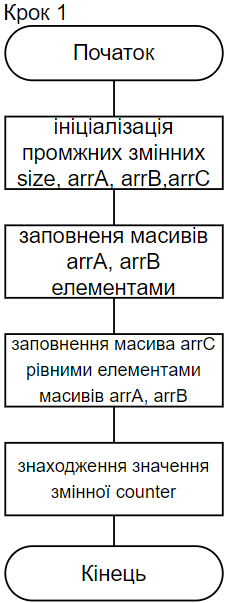
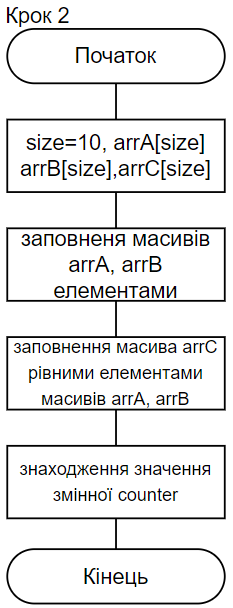
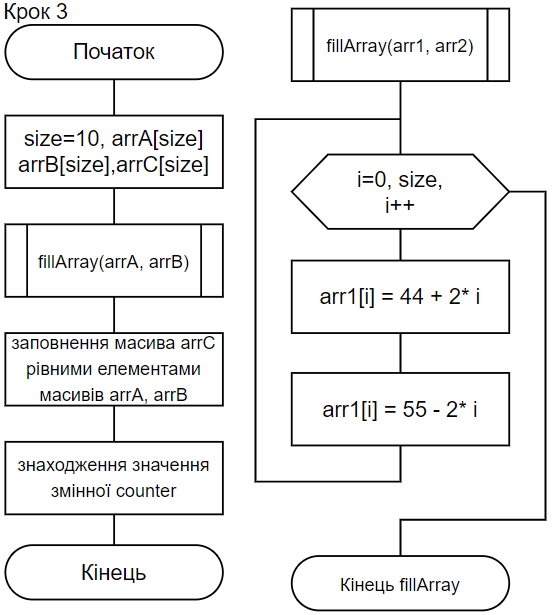
**Крок 4.** Деталізація заповнення масива ***arrC*** рівними елементами масивів ***arrA, arrB***

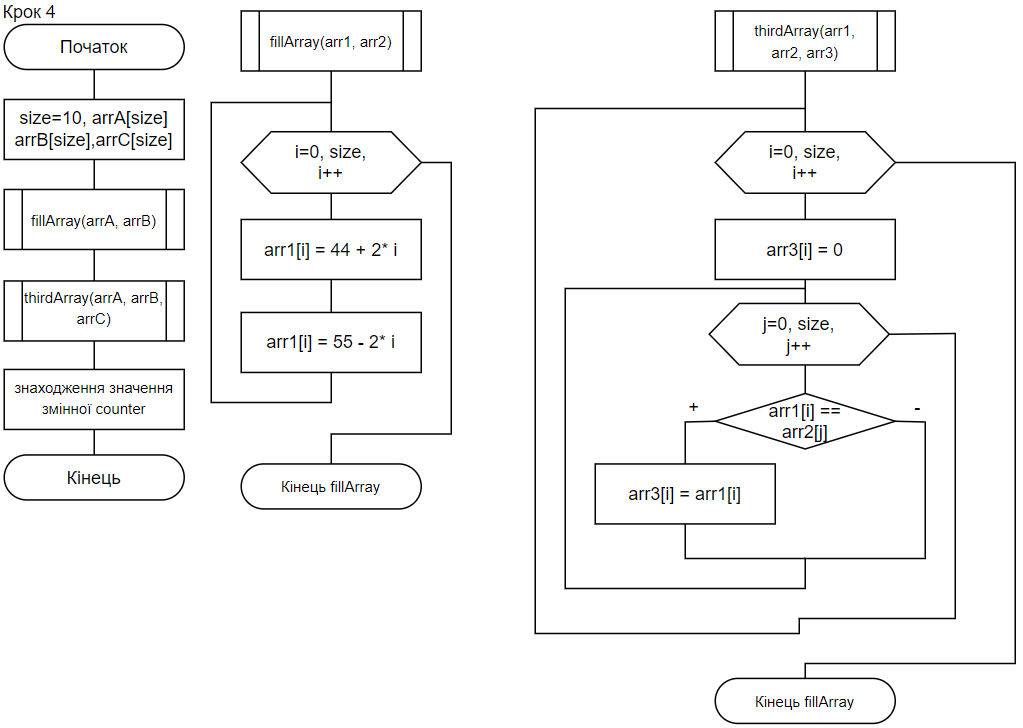
**Крок 5.** Деталізація знаходження значення змінної ***counter***

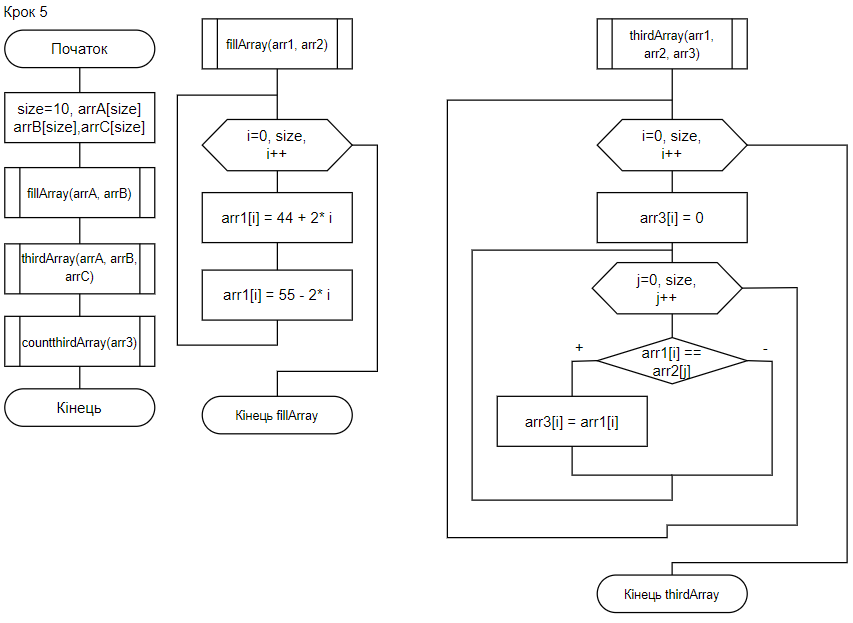
**Псевдокод:**

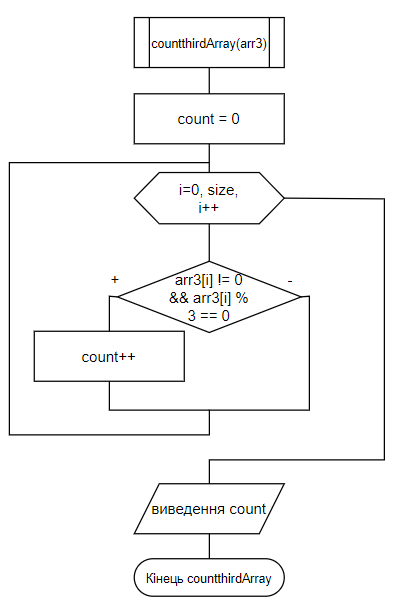
|  |  |
| --- | --- |
| *крок 1*  **початок**  ініціалізація промжних змінних size, arrA, arrB,arrC  заповненя масивів arrA, arrB елементами  заповнення масива arrC рівними елементами масивів arrA, arrB  знаходження значення змінної counter  **кінець** | *крок 2*  **початок**  size=10  arrA[size]  arrB[size]  arrC[size]  заповненя масивів arrA, arrB елементами  заповнення масива arrC рівними елементами масивів arrA, arrB  знаходження значення змінної counter  **кінець** |
| *крок 3*  **початок**  size=10  arrA[size]  arrB[size]  arrC[size]  **виклик підпрограми** fillArray(arrA, arrB)  заповнення масива arrC рівними елементами масивів arrA, arrB  знаходження значення змінної counter  **Підпрограма** fillArray(arrA, arrB)  **повторити**  **для *і* від** 0 **до *size, і++***  *arr1[i]* = 44 + 2\* і  *arr2[i]* = 55 – 2 \* і  **все повторити**  **кінець підпрограма**  **кінець** |  |
| *крок 4*  **початок**  size=10  arrA[size]  arrB[size]  arrC[size]  **виклик підпрограми** fillArray(arrA, arrB)  **виклик підпрограми** thirdArray(arrA, arrB, arrC)  знаходження значення змінної counter  **Підпрограма** fillArray(*arr1, arr2*)  **повторити**  **для *і* від** 0 **до *size, і++***  *arr1[i]* = 44 + 2\* і  *arr2[i]* = 55 – 2 \* і  **все повторити**  **кінець підпрограма**  **Підпрограма** thirdArray(*arr1, arr2, arr3*)  **повторити**  **для *і* від** 0 **до *size* i++**  *arr3 [i] = 0*  **повторити**  **для *j* від** 0 **до *size j*++**  **якщо** arr1[i] == arr2[j]  **то:**  *arr3* [i] = arr1[i]  **все якщо**  **все повторити**  **все повторити**  **кінець підпрограма**  **кінець** |  |
| *крок 5*  **початок**  size=10  arrA[size]  arrB[size]  arrC[size]  **виклик підпрограми** fillArray(arrA, arrB)  **виклик підпрограми** thirdArray(arrA, arrB, arrC)  **виклик підпрограми** countthirdArray(arrC)  **Підпрограма** fillArray(*arr1, arr2*)  **повторити**  **для *і* від** 0 **до *size, і++***  *arr1[i]* = 44 + 2\* і  *arr2[i]* = 55 – 2 \* і  **все повторити**  **кінець підпрограма**  **Підпрограма** thirdArray(*arr1, arr2, arr3*)  **повторити**  **для *і* від** 0 **до *size* i++**  *arr3 [i] = 0*  **повторити**  **для *j* від** 0 **до *size j*++**  **якщо** arr1[i] == arr2[j]  **то:**  *arr3* [i] = arr1[i]  **все якщо**  **все повторити**  **все повторити**  **кінець підпрограма**  **Підпрограма** countthirdArray(*arr3*)  count = 0  **повторити**  **для *і* від** 0 **до *size* із кроком** 1  **якщо** *arr3[i] !=0 && arr3[i] % 3 = 0*  **то:**  count++  **все якщо**  **все повторити**  **виведення** count  **кінець підпрограма**  **кінець** |  |

**Блок-схема:**

**** **** 

****

****

****

**Код:**

#include <iostream>

#define size 10

using namespace std;

void fillArray(unsigned char array\_1[], unsigned char array\_2[]);

void thirdArray(unsigned char array\_1[], unsigned char array\_2[], unsigned char array\_3[]);

void showArray(unsigned char array[], string massage);

void countthirdArray(unsigned char array\_3[]);

int main() {

unsigned char array\_1[size], array\_2[size], array\_3[size];

fillArray(array\_1, array\_2);

thirdArray(array\_1, array\_2, array\_3);

showArray(array\_1, "array 1");

showArray(array\_2, "array 2");

showArray(array\_3, "array 3");

countthirdArray(array\_3);

return 0;

}

void fillArray(unsigned char array\_1[], unsigned char array\_2[]) {

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

array\_1[i] = 44 + 2 \* i;

array\_2[i] = 55 - 2 \* i;

}

}

void thirdArray(unsigned char array\_1[], unsigned char array\_2[], unsigned char array\_3[]) {

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

array\_3[i] = 0;

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

if (array\_1[i] == array\_2[j]) {

array\_3[i] = array\_1[i];

}

}

}

}

void countthirdArray(unsigned char array\_3[]) {

int count = 0;

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

if (array\_3[i] != 0 && array\_3[i] % 3 == 0)

count++;

}

cout << "nuber in third array :" << count << endl;

}

void showArray(unsigned char array[], string massage) {

cout << massage << endl;

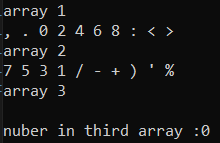
for (int i = 0; i < size; i++) {

cout << array[i] << " ";

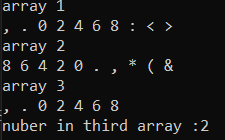
}

cout << endl;

}

****

Якщо поміняти 55-2\*і на 56-2\*і (Щоб елементи двох масивів могли співпасти):

****

**Випробування:**

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| arrA | Елементи з кодами: 44, 46, 48, 50,52, 54, 56, 58, 60, 62 |
| arrB | Елементи з кодами: 55, 53, 51, 49, 47, 45, 43, 41, 39, 37 |
| arrC | Співпадаючих елементів немає |
| count | 0 |
|  | Кінець |

Якщо поміняти 55-2\*і на 56-2\*і

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| arrA | Елементи з кодами: 44, 46, 48, 50,52, 54, 56, 58, 60, 62 |
| arrB | Елементи з кодами: 56, 54, 52, 50, 48, 46, 44, 42, 40, 38 |
| arrC | Елементи з кодами: 56, 54, 52, 50, 48, 46, 44 |
| Кратні 3 | Елементи з кодами: 54, 48 |
| count | 2 |
|  | Кінець |

**Висновки:** На цій лабараторній я досліджував методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях. Мій варіант лабараторної включав роботу з операторами % - пошук остаці від ділення, роботу з масивами: їх формування, використання алгоритмів пошуку тощо, побудову логічних дій розгалудження, написання підпрограм, деталізацію різних частин псевдокоду і блоксхем. Також я практикувався в умінні оформлювати лабараторну роботу, а саме: титульний аркуш, математичну модель, псевдокод алгоритму, блок схему алгоритму, випробування алгоритму, висновки.