Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Катедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 7 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійного пошуку в послідовностях»

Варіант 18

Виконав студент ІП-11 Лесів Владислав Ігорович

Перевірив Мартинова О.П.

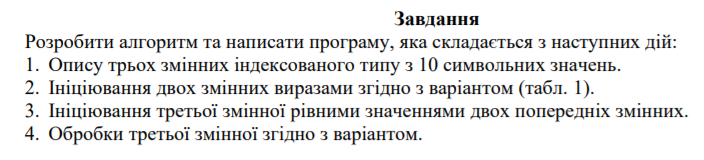
Київ 2021

**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 №18.**

**Постановка задачі.** Результатом розв’язку індекс першого входження елементу з кодом 52 до третього масиву. Для визначення результату повинні бути задані дві послідовності символьних значень, елементи яких необхідно буде обрахувати за заданими виразами. Інших початкових даних для розв’язку не потрібно.

**Побудова математичної моделі.** Складемо таблицю імен змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім'я | Призначення |
| Перший масив | Символьний | arr1 | Початкове дане |
| Другий масив | Символьний | arr2 | Початкове дане |
| Третій масив | Символьний | arr3 | Проміжний результат |
| Кількість елементів третьої послідовности | Цілий | sizeArr3 | Проміжний результат |
| Розмір масиву (у функції initArray3) | Цілий | size | Проміжний результат |
| Існування першого входження (у функції) | Логічна | k | Проміжний результат |

Математичне формулювання задачі зводиться до знаходження елементів першого та другого масивів за формулами, заданими в умові задачі, знаходження між ними рівних, виокремлення їх до третього масиву. Між виокремленими елементами шукаємо індекс першого входження елементу з кодом 52. Коли індекс знайдено, виводимо його.

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

*Крок 1.* Визначимо основні дії.

*Крок 2.* Деталізуємо дію ініціалізації першого і другого масивів за формулами, заданими в умові задачі, за допомогою підпрограми.

*Крок 3.* Деталізуємо дію ініціалізації третього масиву і знаходження розміру послідовності за допомогою підпрограми.

*Крок 4.* Деталізуємо дію виведення масивів за допомогою підпрограми.

*Крок 5.* Деталізуємо дію обробки третього масиву згідно з умовою задачі.

*Псевдокод*

*крок 1*

**початок**

ініціалізація першого і другого масивів

ініціалізація третього масиву і знаходження розміру

виведення масивів

обробка третього масиву

**кінець**

*крок 2*

**початок**

initArrays1And2(arr1, arr2)

ініціалізація третього масиву і знаходження розміру

виведення масивів

обробка третього масиву

**кінець**

**підпрограма** initArrays1And2(arr1, arr2)

**повторити**

**для** і **від** 0 **до** 9

arr1[i]:=char(60 - 2 \* i)

arr2[i]:=char(40 + 3 \* i)

**все повторити**

**кінець підпрограми**

*крок 3*

**початок**

initArrays1And2(arr1, arr2)

sizeArr3:= initArray3(arr1, arr2, arr3)

виведення масивів

обробка третього масиву

**кінець**

**підпрограма** initArrays1And2(arr1, arr2)

**повторити**

**для** і **від** 0 **до** 9

arr1[i]:=char(60 - 2 \* i)

arr2[i]:=char(40 + 3 \* i)

**все повторити**

**кінець підпрограми**

**підпрограма** initArray3(arr1, arr2, arr3)

size:=0

**повторити**

**для** і **від** 0 **до** 9

**повторити**

**для** j **від** 0 **до** 9

**якщо** arr1[i] == arr2[j]

**то**

arr3[size]:= arr1[i]

size:=size+1

**все якщо**

**все повторити**

**все повторити**

**повернути** size

**кінець підпрограми**

*крок 4*

**початок**

initArrays1And2(arr1, arr2)

sizeArr3:= initArray3(arr1, arr2, arr3)

outputArray(arr1, 10)

outputArray(arr2, 10)

outputArray(arr3, sizeArr3)

обробка третього масиву

**кінець**

**підпрограма** initArrays1And2(arr1, arr2)

**повторити**

**для** і **від** 0 **до** 9

arr1[i]:=char(60 - 2 \* i)

arr2[i]:=char(40 + 3 \* i)

**все повторити**

**кінець підпрограми**

**підпрограма** initArray3(arr1, arr2, arr3)

size:=0

**повторити**

**для** і **від** 0 **до** 9

**повторити**

**для** j **від** 0 **до** 9

**якщо** arr1[i] == arr2[j]

**то**

arr3[size]:= arr1[i]

size:=size+1

**все якщо**

**все повторити**

**все повторити**

**повернути** size

**кінець підпрограми**

**підпрограма** outputArray(arr, size)

**повторити**

**для** і **від** 0 **до** size-1

виведення arr[i]

**все повторити**

**кінець підпрограми**

*крок 5*

**початок**

initArrays1And2(arr1, arr2)

sizeArr3:= initArray3(arr1, arr2, arr3)

outputArray(arr1, 10)

outputArray(arr2, 10)

outputArray(arr3, sizeArr3)

processArray3(arr3, sizeArr3)

**кінець**

**підпрограма** initArrays1And2(arr1, arr2)

**повторити**

**для** і **від** 0 **до** 9

arr1[i]:=char(60 - 2 \* i)

arr2[i]:=char(40 + 3 \* i)

**все повторити**

**кінець підпрограми**

**підпрограма** initArray3(arr1, arr2, arr3)

size:=0

**повторити**

**для** і **від** 0 **до** 9

**повторити**

**для** j **від** 0 **до** 9

**якщо** arr1[i] == arr2[j]

**то**

arr3[size]:= arr1[i]

size:=size+1

**все якщо**

**все повторити**

**все повторити**

**повернути** size

**кінець підпрограми**

**підпрограма** outputArray(arr, size)

**повторити**

**для** і **від** 0 **до** size-1

виведення arr[i]

**все повторити**

**кінець підпрограми**

**підпрограма** processArray3(arr3, size)

k:=0

**повторити**

**для** і **від** 0 **до** size-1

**якщо (**int(arr3[i]) == 52 && !k)

**то**

виведення i

k:=1

**все якщо**

**все повторити**

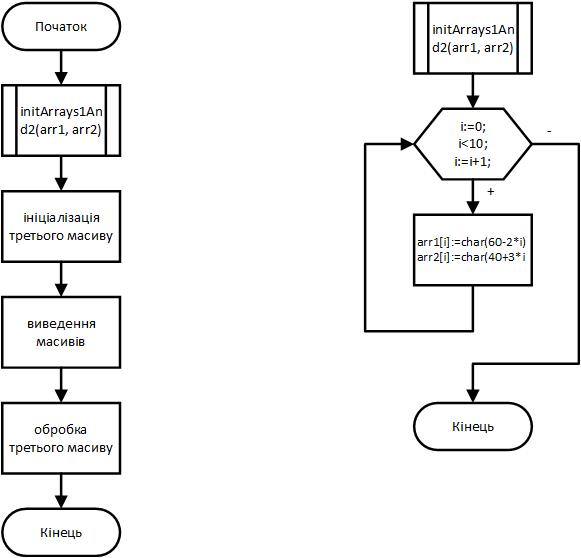
**кінець підпрограми**

*Блок-схема*

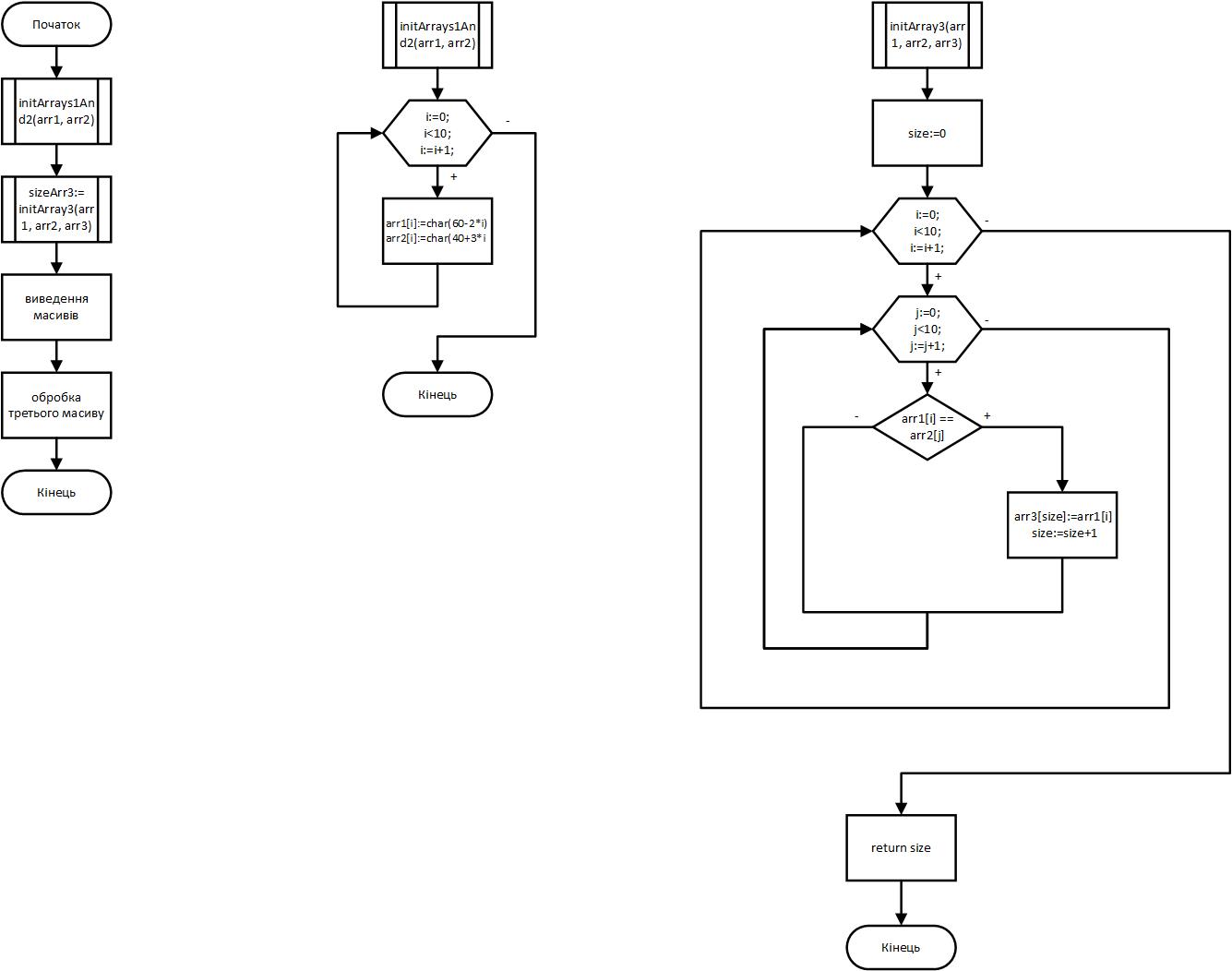
*Крок 1*

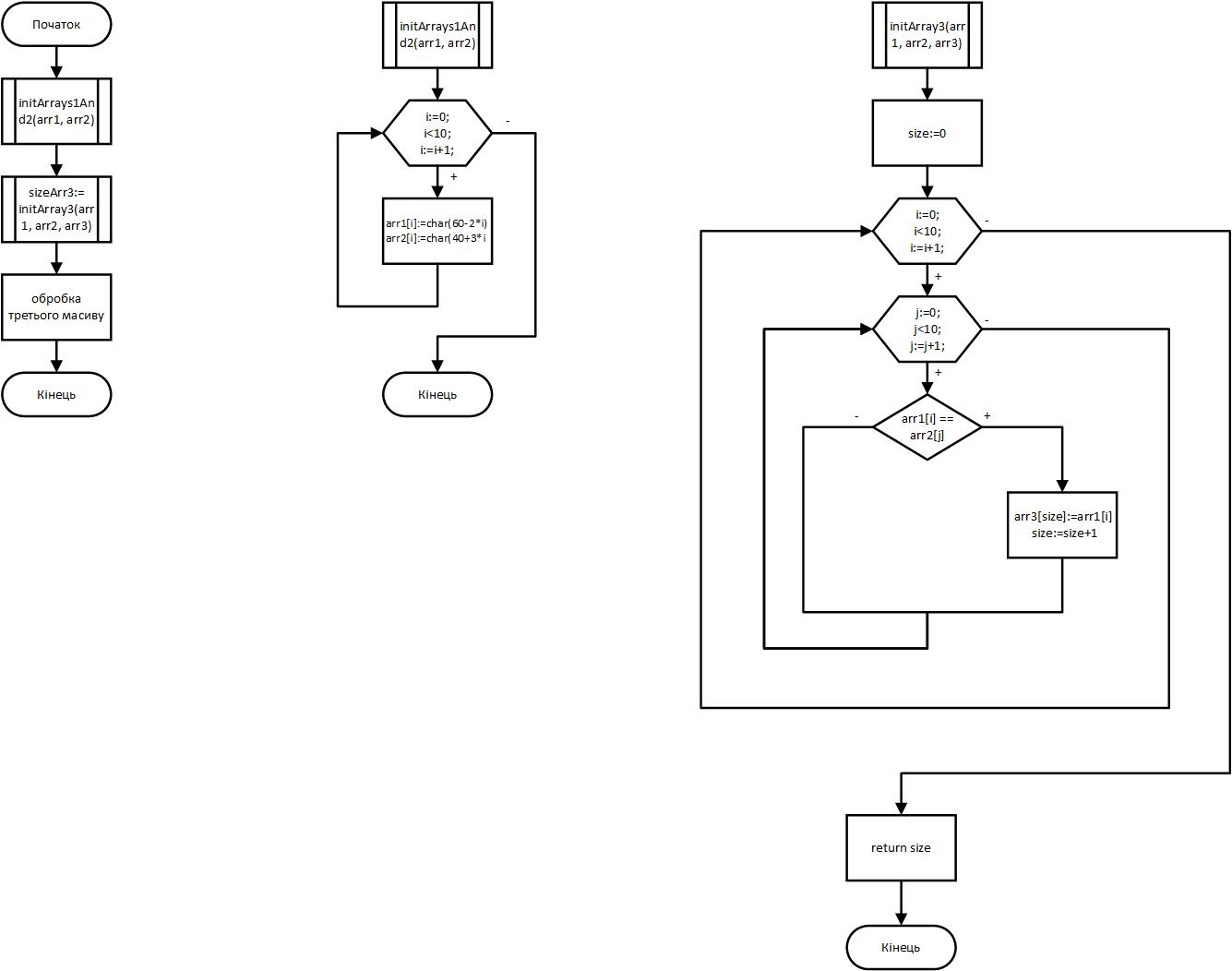


*Крок 2*

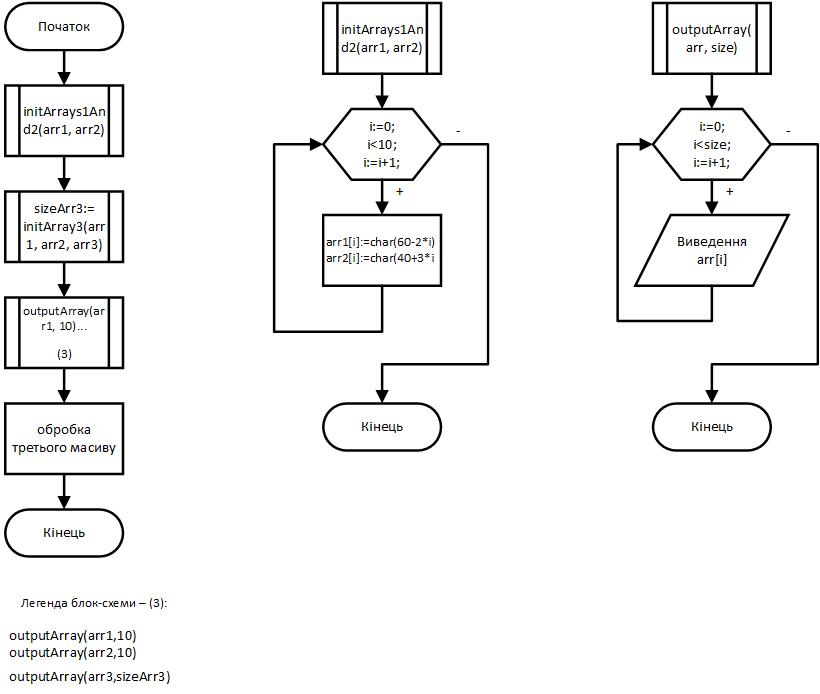


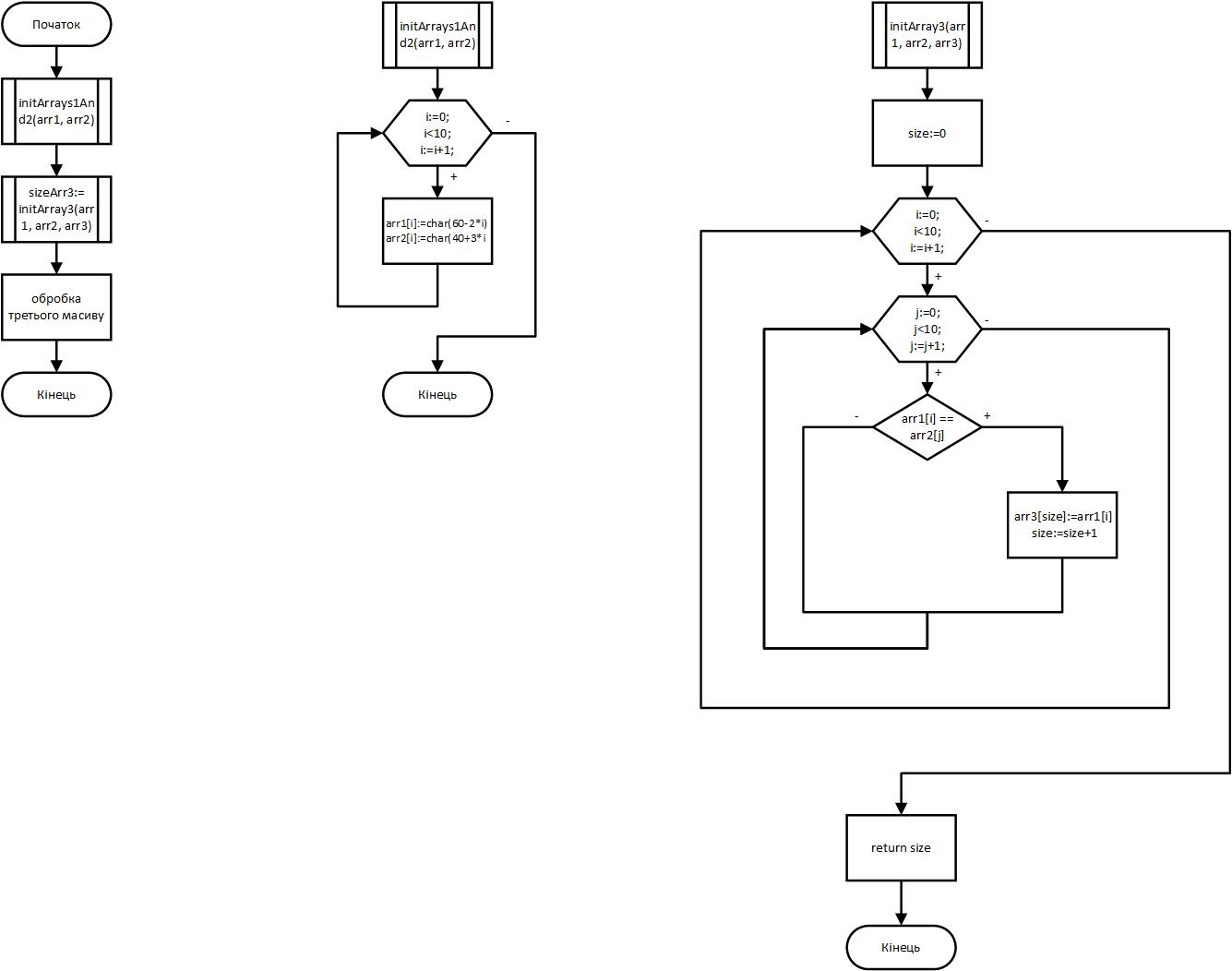
*Крок 3*

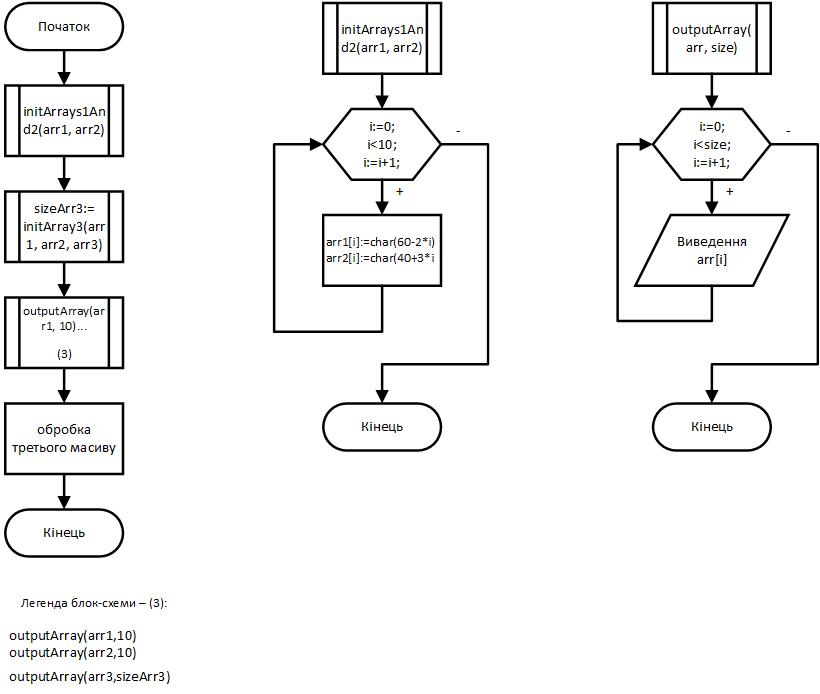




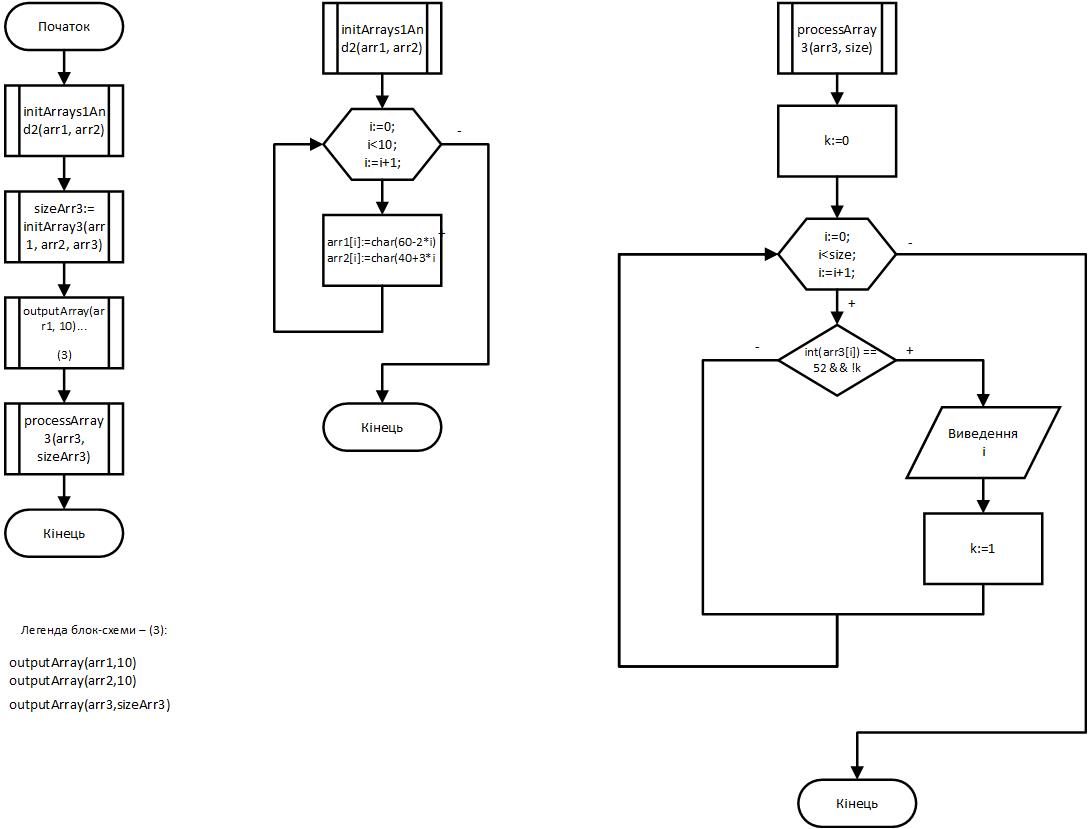
*Крок 4*

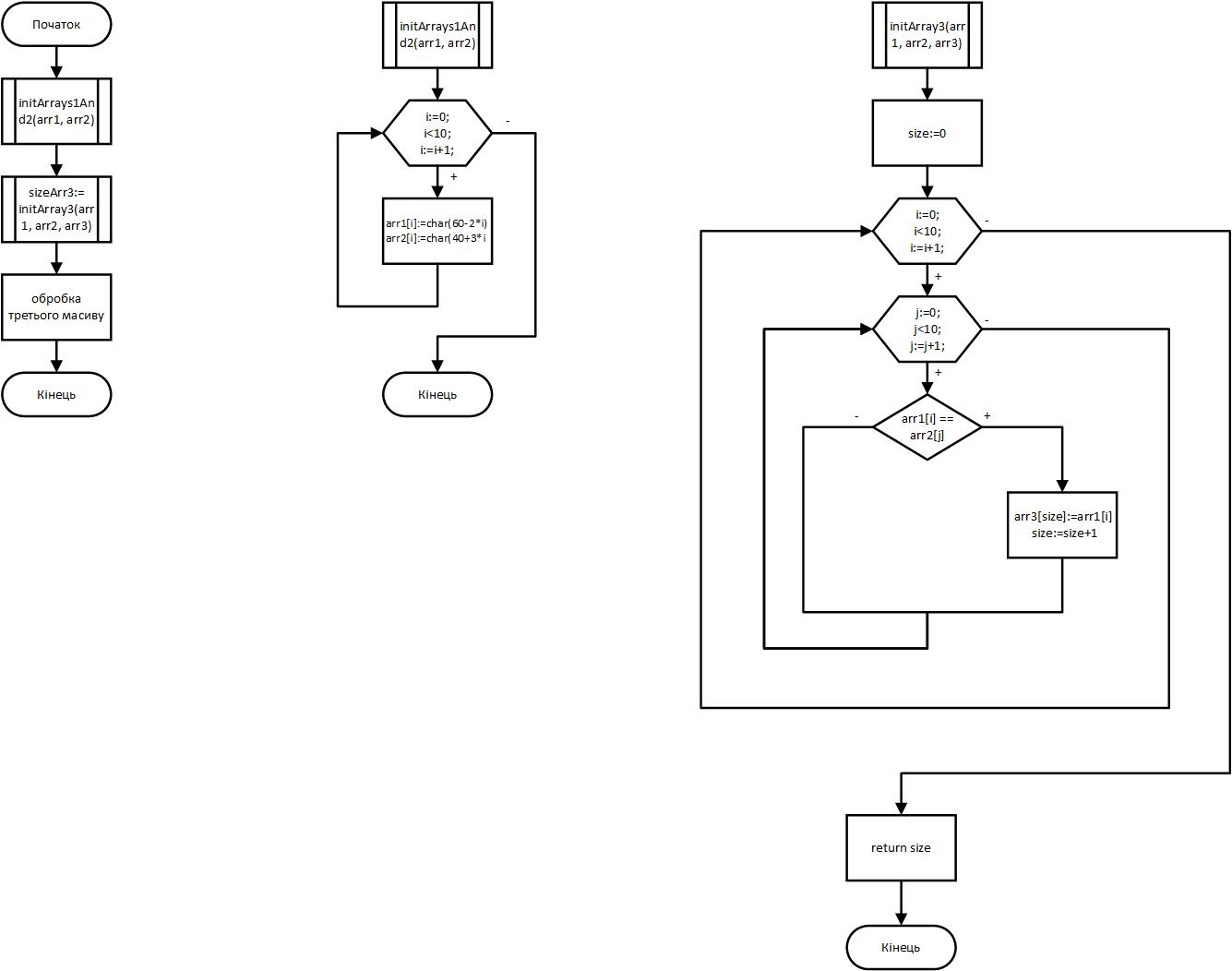


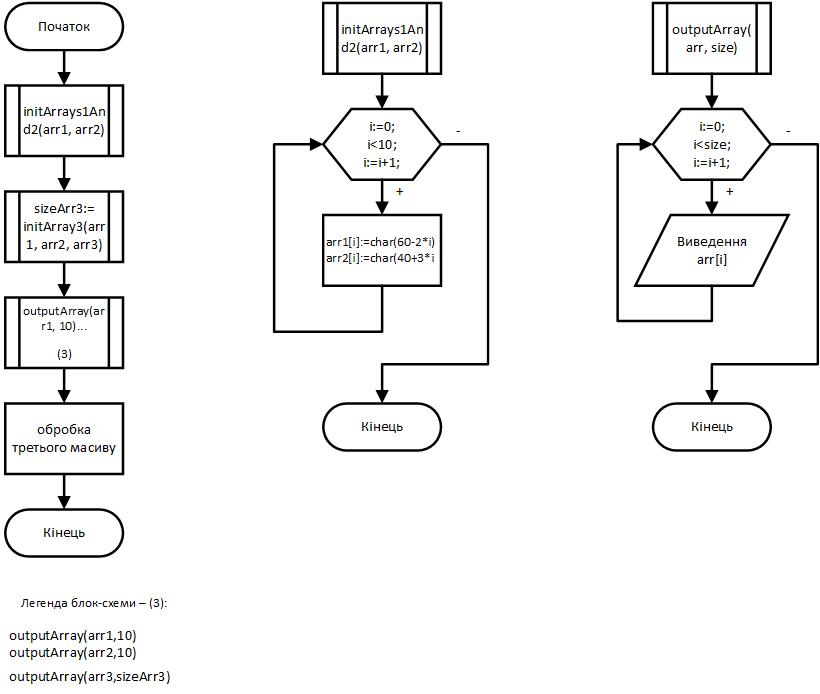


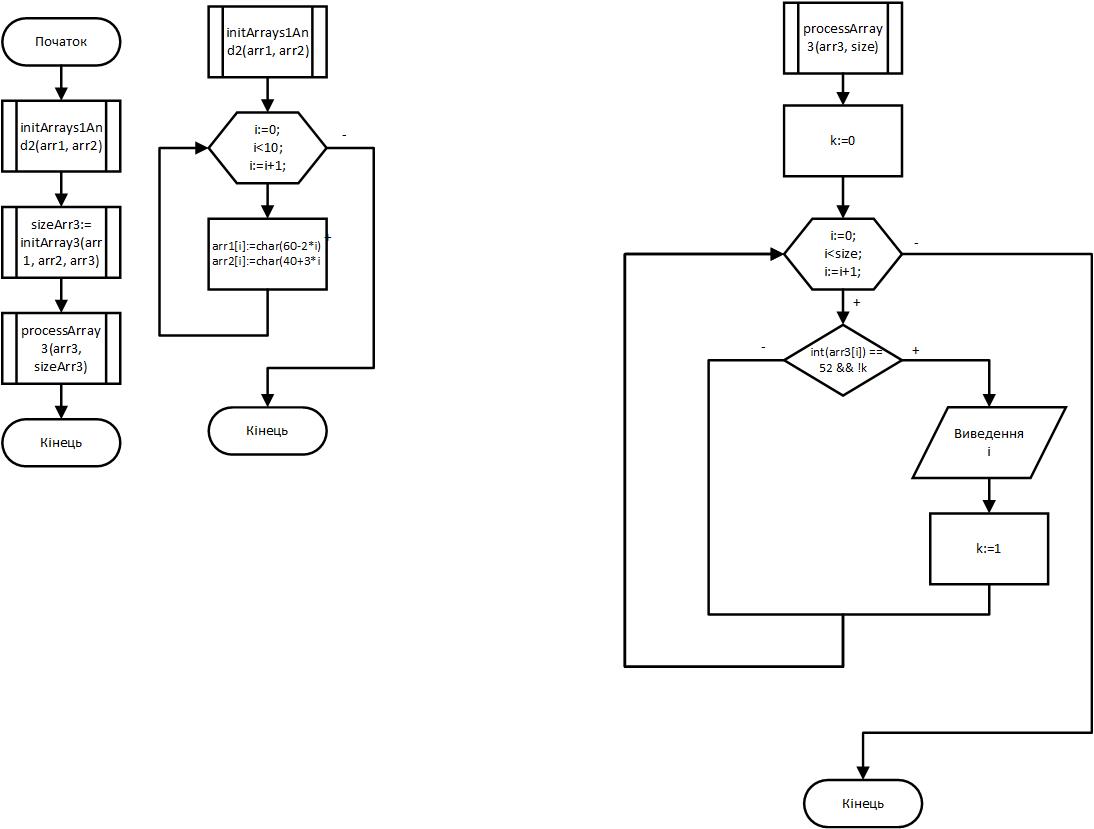


*Крок 5*









**Виконання мовою C++.**

**Код програми:**

#include <iostream>

using namespace std;

void initArrays1And2(char[], char[]); //Прототипи функцій: ініціалізація масивів

int initArray3(char[], char[], char[]); //Заповнення третього масиву

void processArray3(char[],int); //Обробка третього масиву

void outputArray(char[], int);

int main()

{

setlocale(LC\_ALL, "ukr");

char arr1[10], arr2[10], arr3[10];

initArrays1And2(arr1, arr2);

int sizeArr3 = initArray3(arr1, arr2, arr3);

cout << "Перший масив: ";

outputArray(arr1, 10);

cout << "\nДругий масив: ";

outputArray(arr2, 10);

cout << "\nТретiй масив: ";

outputArray(arr3, sizeArr3);

processArray3(arr3, sizeArr3);

}

void initArrays1And2(char arr1[], char arr2[]) {

for (int i = 0;i < 10;i++) {

arr1[i] = char(60 - 2 \* i); //Заповненнями виразами, заданими в умові

arr2[i] = char(40 + 3 \* i);

}

}

int initArray3(char arr1[], char arr2[], char arr3[]) {

int size = 0;

for (int i = 0;i < 10;i++) {

for (int j = 0;j < 10;j++) {

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

arr3[size] = arr1[i];

size++;

}

}

}

return size;

}

void outputArray(char arr[], int size) {

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

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

}

}

void processArray3(char arr3[], int size){

bool k = 0; //Змінна для перевірки входження

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

if (int(arr3[i]) == 52 && !k) {

cout << "\n\nПерше входження елементу з кодом 52 ('4') в третьому масивi пiд iндексом i = " << i;

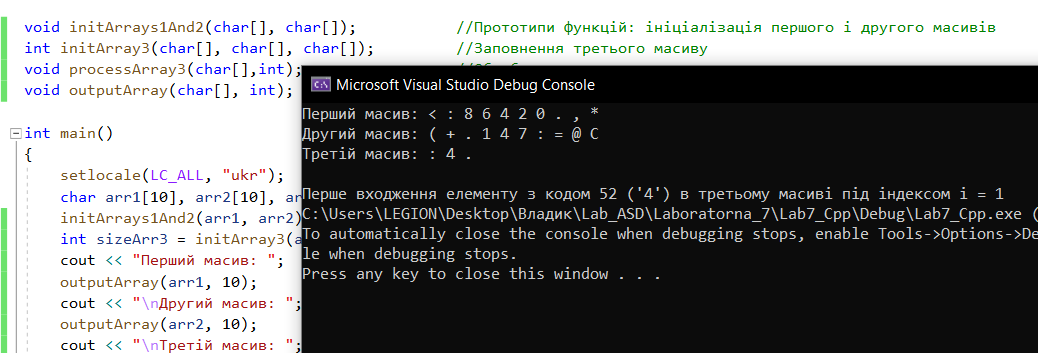
k = 1;

}

}

}

**Випробування алгоритму.**



**Перевірка алгоритму.**

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | Виведення: перший масив: < : 8 6 4 2 0 . , \* |
| 2 | Виведення: другий масив: ( + . 1 4 7 : = @ C |
| 3 | Виведення: третій масив: : 4 . (коди 58, 52, 46) (спільні елементи двох послідовностей) |
| 4 | Виведення: перше входження елементу з кодом 52 під i=1 |
|  | Кінець |

**Висновок.** Отже, у цій роботі я дослідив методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій. У результаті лабораторної роботи було розроблено математичну модель, що відповідає постановці задачі; псевдокод та блок-схеми, які пояснюють логіку алгоритму. Використовуючи чотири підпрограми для роботи з впорядкованими послідовностями, отримуємо коректний результат.