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 26

Виконав студент ІП-12, Саркісян Валерія Георгіївна

Перевірив _____
(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.

Задача 26. 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 (1-ий масив 2*i+42 , 2-ий масив 54-2*i)
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної згідно з варіантом (елементи, які менші за максимальний код).

1. ***Постановка задачі.***

   За умовою задачі необхідно ініціалізувати 3 масиви, що складаються з 10 символьних значень. 1-ий і 2-ий масиви згенерувати за даними в умові правилами, а 3-ій заповнити рівними значеннями з перших двох масивів. Далі потрібно створити новий масив, заповнюючи його всіма елементами 3-го масива, що менші за максимальний код.

2. ***Побудова математичної моделі***.

Змінна	Тип	Ім’я	Призначення
Розмір масиву	цілий, сталий	size	Початкове дане, константа
1-ий масив (2*i+42)	індексований	First[]	Проміжне дане
2-ий масив (54-2*i)	індексований	Second[]	Проміжне дане
3-ий масив	індексований	Third[]	Проміжне дане
Результуючий масив	індексований	NewArr[]	результат
Генерація 1-го і 2-го масивів	Функція, порожній	generateArr(arr1[], arr2[], size)	Генерація і виведення масивів
Генерація результуючого масиву	Функція, порожній	generateNewArr(arr3[], size)	Генерація елементів, менших за макс. код
Лічильник	цілий	i	Проміжне дане
Лічильник	цілий	j	Проміжне дане
Формальний параметр функції	індексований	arr1[]	Проміжне дане
Формальний параметр функції	індексований	arr2[]	Проміжне дане
Формальний параметр функції	індексований	arr3[]	Проміжне дане

Максимальний код у 3-му масиві	символьний	max	Проміжне дане

3. Розв'язання.

Основна програма:

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію ініціалізації змінних.

Крок 3. Деталізуємо дію генерації 1-го та 2-го масивів.

Крок 4. Деталізуємо дію генерації 3-го масиву.

Крок 5. Деталізуємо дію генерації результуючого масиву.

Підпрограми:

Крок 3.1. Деталізуємо дію генерації та виведення 1-го масиву.

Крок 3.2. Деталізуємо дію генерації та виведення 2-го масиву.

Крок 5.1. Деталізуємо дію знаходження елементу з максимальним кодом.

Крок 5.2. Деталізуємо дію генерації та виведення результуючого масиву.

Псевдокод.

Крок 1

Початок

Ініціалізація змінних
генерації 1-го та 2-го масивів
генерація 3-го масиву
генерація результуючого масиву

Кінець

Крок 2

Початок

size =10, First[size], Second[size], Third[size]
генерації 1-го та 2-го масивів
генерація 3-го масиву
генерація результуючого масиву

Кінець

Крок 3

Початок

size =10, First[size], Second[size], Third[size]
generateArr(First, Second, size)
генерація 3-го масиву

генерація результуючого масиву
Кінець

Крок 4
Початок
size =10, First[size], Second[size], Third[size]
generateArr(First, Second, size)
Для i від 0 до size із кроком 1 **повторити**
Third[i] = 0;
Для j від 0 до size із кроком 1 **повторити**
якщо First[i] == Second[i]
то
Third[i]= First[i]
Вивести Third[i]
Все якщо
Все повторити
Все повторити
генерація результуючого масиву
Кінець

Крок 5
Початок
size =10, First[size], Second[size], Third[size]
generateArr(First, Second, size)
Для i від 0 до size із кроком 1 **повторити**
Third[i] = 0;
Для j від 0 до size із кроком 1 **повторити**
якщо First[i] == Second[i]
то
Third[i]= First[i]
Вивести Third[i]
Все якщо
Все повторити
Все повторити
generateNewArr(Third, size)
Кінець

Псевдокод підпрограм.

Крок 3.1. Функція **generateArr(arr1[], arr2[], size)**

Крок 3.2. Функція **generateArr(arr1[], arr2[], size)**

Початок generateArr(arr1[],arr2[], size)
Для і від 0 до size із кроком 1 **повторити**
arr1[i] = 2 * i + 42
вивести arr1[i]
Все повторити
Генерація 2-го масиву
Кінець generateArr

Початок generateArr(arr1[],arr2[], size)
Для і від 0 до size із кроком 1 **повторити**
arr1[i] = 2 * i + 42
вивести arr1[i]
Все повторити
Для j від 0 до size із кроком 1 **повторити**
arr2[j] =52 - 2 * j
вивести arr2[j]
Все повторити
Кінець generateArr

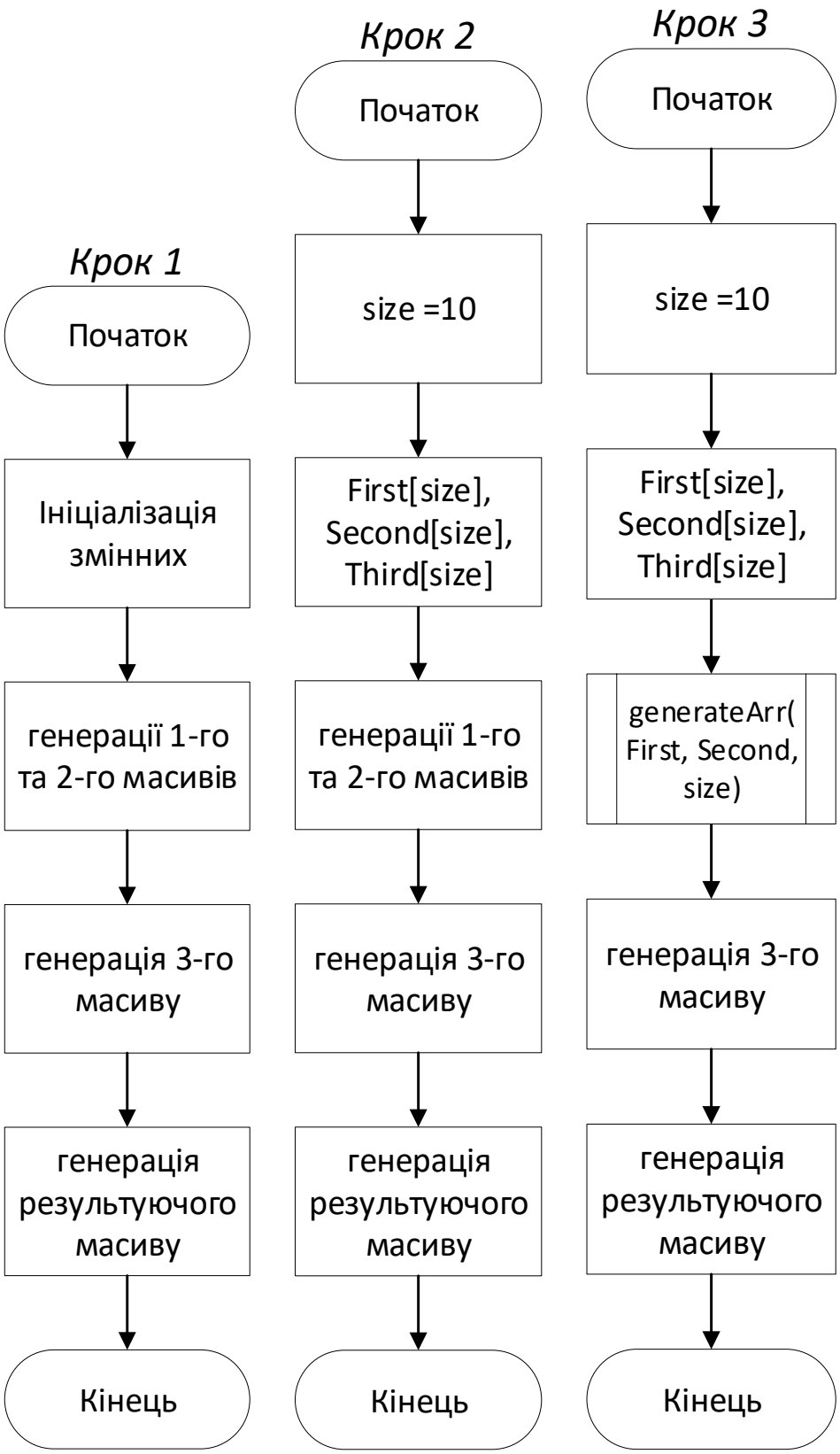
Крок 5.1. Функція **generateNewArr (arr3[], size)**

Початок generateNewArr (arr3[], size)
NewArr[10], max = arr3[0]
Для і від 0 до size із кроком 1 **повторити**
Якщо arr3[i] > max **то**
max = arr3[i]
Все якщо
Все повторити
генерації та виведення результуючого масиву
Кінець generateArr

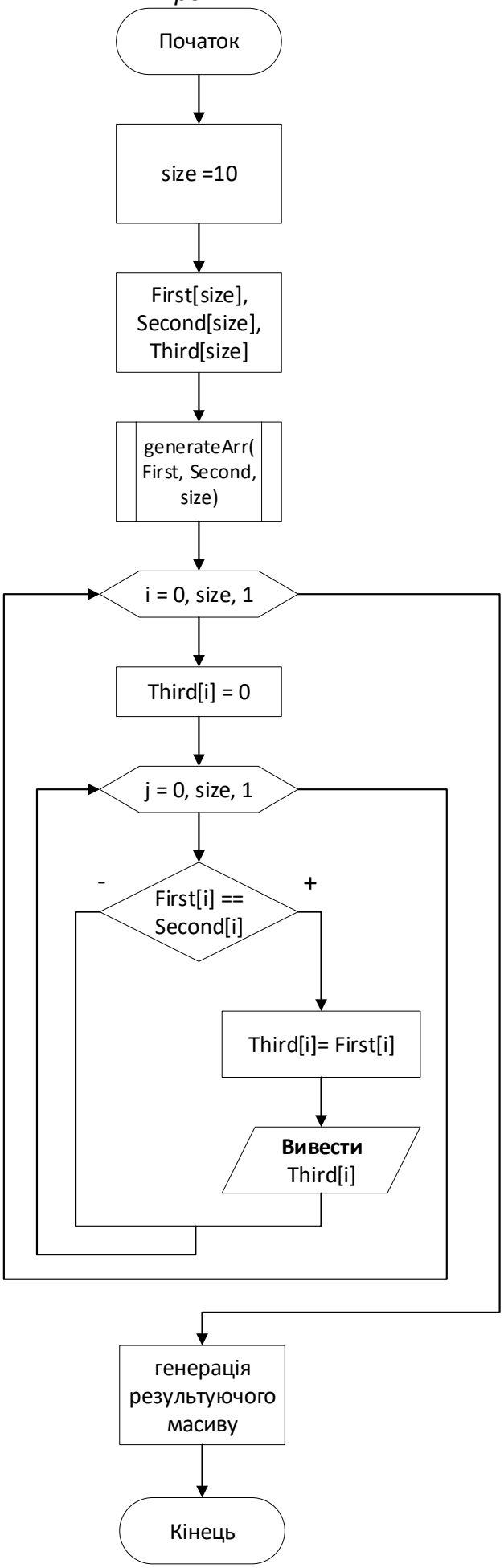
Крок 5.2. Функція generateNewArr (arr3[], size)

Початок generateNewArr (arr3[], size)
NewArr[10], max = arr3[0]
Для і від 0 до size із кроком 1 **повторити**
Якщо arr3[i] > max **то**
max = arr3[i]
Все якщо
Все повторити
Для і від 0 до size із кроком 1 **повторити**
Якщо arr3[i] < max **то**
NewArr[i] = arr3[i]
вивести NewArr[i]
Все якщо
Все повторити
Кінець generateArr

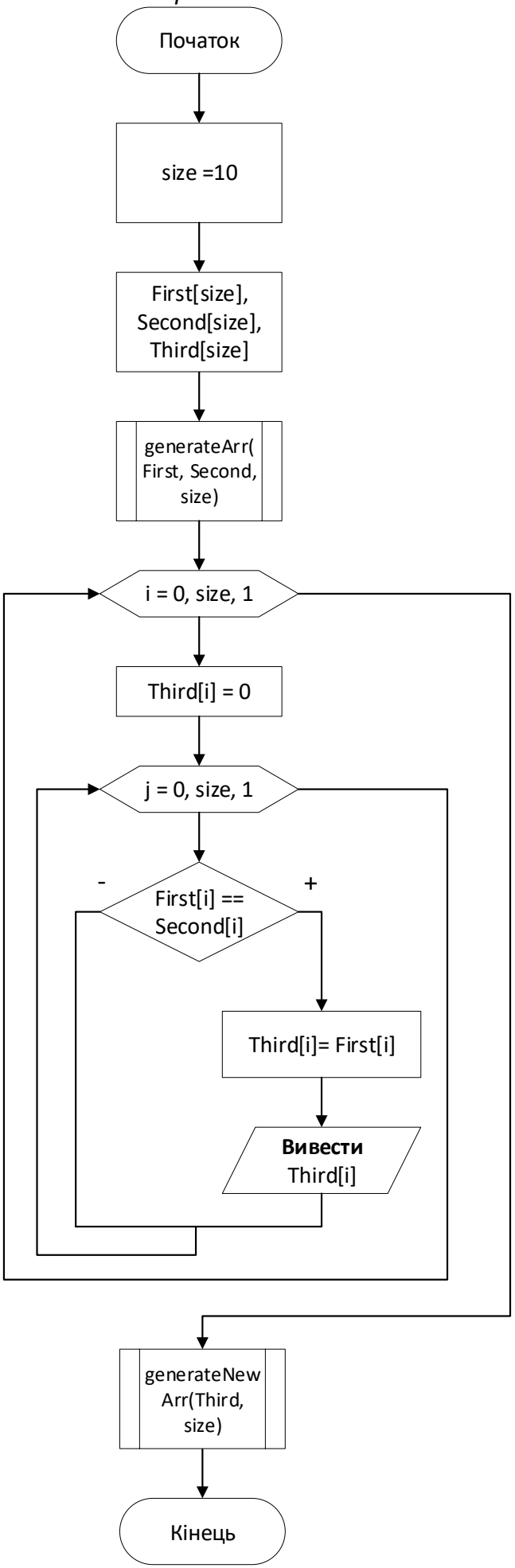
4. Блок-схема.


Крок 1
Початок
Ініціалізація змінних
генерації 1-го та 2-го масивів
генерація 3-го масиву
генерація результуючого масиву
Кінець
Крок 2
Початок
size =10
First[size], Second[size], Third[size]
генерації 1-го та 2-го масивів
генерація 3-го масиву
генерація результуючого масиву
Кінець
Крок 3
Початок
size =10
First[size], Second[size], Third[size]
generateArr(First, Second, size)
генерація 3-го масиву
генерація результуючого масиву
Кінець

Крок 4

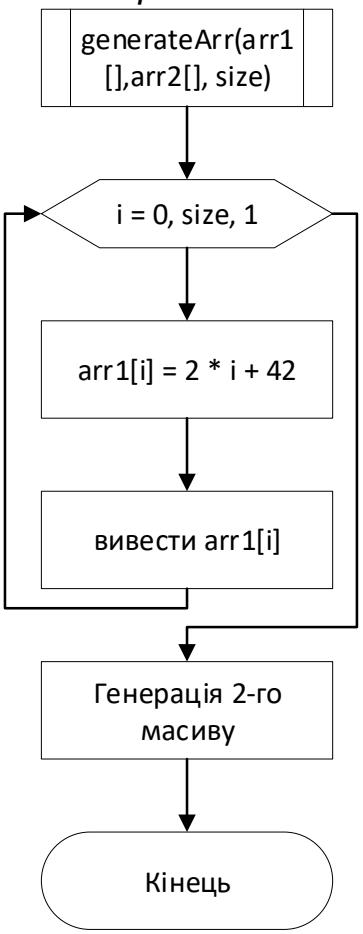

Початок
size =10
First[size], Second[size], Third[size]
generateArr(First, Second, size)
i = 0, size, 1
Third[i] = 0
j = 0, size, 1
-
First[i] == Second[i]
+
Third[i]= First[i]
Вивести Third[i]
генерація результуючого масиву
Кінець

Крок 5

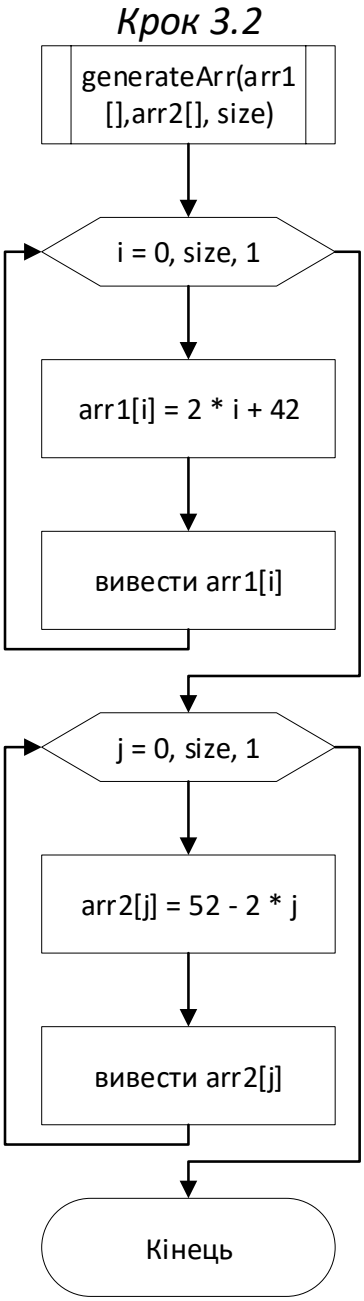

Початок
size =10
First[size], Second[size], Third[size]
generateArr(First, Second, size)
i = 0, size, 1
Third[i] = 0
j = 0, size, 1
-
First[i] == Second[i]
+
Third[i]= First[i]
Вивести Third[i]
generateNew Arr(Third, size)
Кінець

Підпрограми:

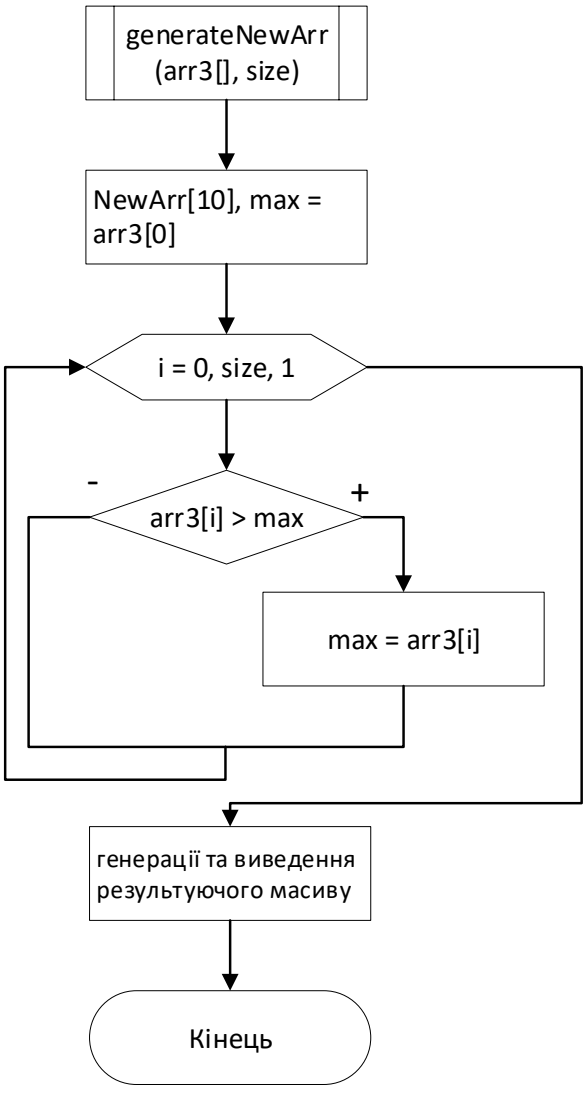
Крок 3.1



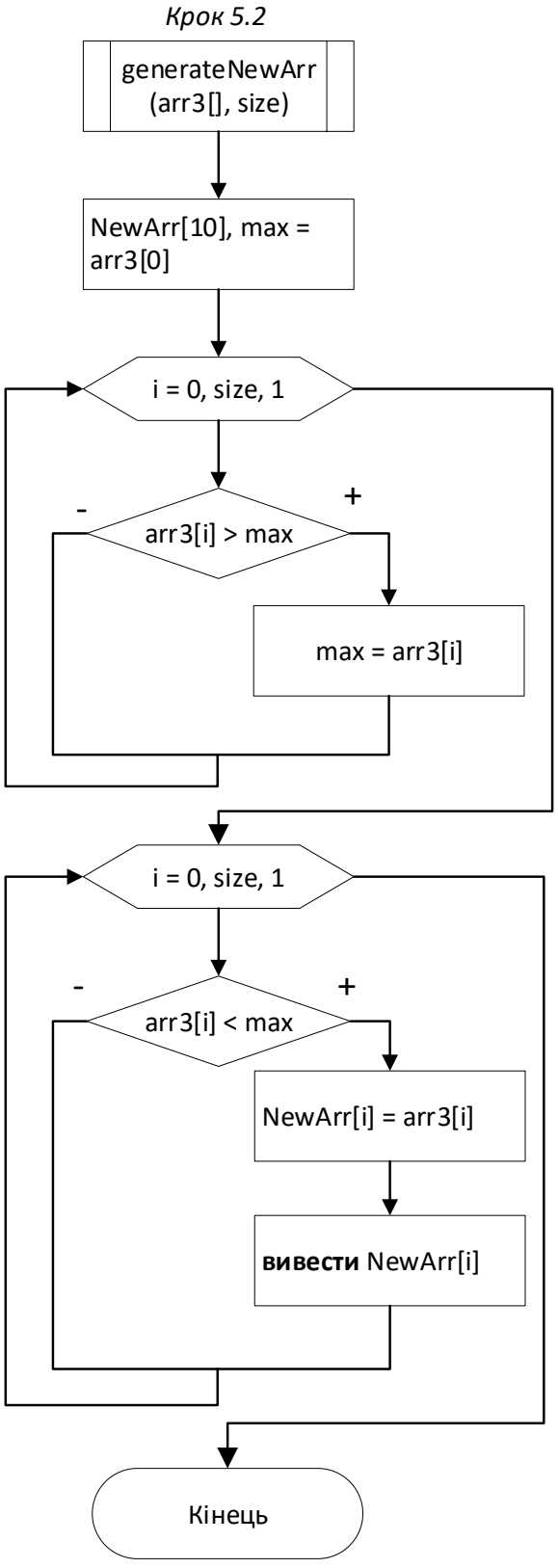
Крок 3.2



Крок 5.1



Крок 5.2



5. *Код програми (C++):*

```
#include <iostream>
using namespace std;

void generateArr(char arr1[], char arr2[], int size);
void generateNewArr(char arr3[], int size);
```

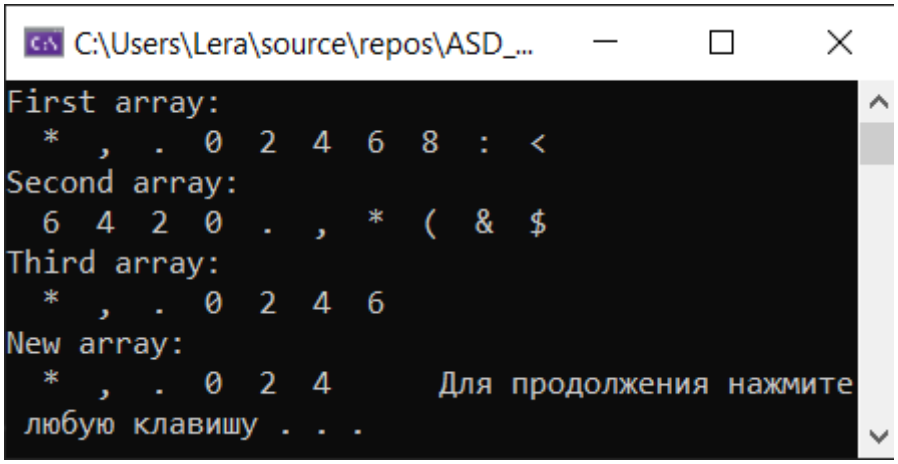
```
int main() {
	const int size = 10;
	char First[size], Second[size], Third[size];
	generateArr(First, Second, size);
	cout << endl << "Third array:" << endl;
	for (int i = 0; i < size; i++) {
		Third[i] = 0;
		for (int j = 0; j < size; j++) {
			if (First[i] == Second[j]) {
				Third[i] = First[i];
				printf("%3c", Third[i]);
			}
		}
	}
	cout << endl << "New array:" << endl;
	generateNewArr(Third, size);
	system("pause");
}
void generateArr(char arr1[], char arr2[], int size) {
	printf("%-20s\n", "First array: ");
	for (int i = 0; i < size; i++) {
		arr1[i] = 2 * i + 42;

		printf("%3c", arr1[i]);

	}
	printf("%-20s\n", "\nSecond array: ");
	for (int j = 0; j < size; j++) {
		arr2[j] = 54 - 2 * j;
		printf("%3c", arr2[j]);
	}
}

void generateNewArr(char arr3[], int size) {
	char NewArr[10];
	char max = arr3[0];
	for (int i = 0; i < size; i++) {
		if (arr3[i] > max)
			max = arr3[i];
	}
	for (int i = 0; i < size; i++) {
		if (arr3[i] < max) {
			NewArr[i] = arr3[i];
			printf("%3c", NewArr[i]);
		}
	}
 }
```

Тестування:



6. ***Висновки.***

На цій лабораторній роботі було досліджено методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набуто практичних навичок їх використання під час складання програмних специфікацій. Для розв’язання поставленої задачі було використано метод послідовного лінійного пошуку для знаходження елемента з максимальним кодом у символьній послідовності.