

Додаток 1

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 лінійних алгоритмів»

Варіант 26

Виконав студент ПІ-13 Паламарчук Олександр Олександрович
(шифр, прізвище, ім'я, по батькові)

Перевірила Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 202 1

Лабораторна робота 7

Дослідження лінійного пошуку в послідовностях

Мета - дослідити методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набутти практичних навичок їх використання під час складання програмних специфікацій.

Варіант 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-й масив за правилом $2 * i + 42$, 2-й масив за правилом $54 - 2 * i$.
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної знаходженням всіх елементів, які менші за максимальний код.

◆ Постановка задачі

Ініціювати дві змінні індексованого типу з 10 символьних значень за правилами: 1-й масив за правилом $2 * i + 42$, 2-й масив за правилом $54 - 2 * i$. Розробити функцію, за допомогою якої ми зможемо знайти рівні значення з першого та другого масиву і заповнити ними третій масив. Розробити функцію, що допоможе обробити третій масив, а саме заповнити результуючий масив всіма значеннями, що менші за максимальний код. Вихідним даним є результуючий масив, що складається з символьних значень.

◆ Побудова математичної моделі

Складемо таблицю змінних

Змінна	Тип	Призначення
<i>firstArray</i> []	Індексований	Проміжне значення
<i>secondArray</i> []	Індексований	Проміжне значення
<i>thirdArray</i> []	Індексований	Проміжне значення
<i>result</i> []	Індексований	Кінцеве дане
<i>isTrue</i>	Логічний	Проміжне значення

<i>isEquals</i>	Логічний	Проміжне значення
<i>max</i>	Символьний	Проміжне значення

firstArray[], secondArray[], thirdArray[], result[] - змінні індексованого типу, що складаються з 10 символьних значень. Для вирішення задачі використаємо допоміжні алгоритми (підпрограми), виклики яких мають вигляд *arrayOfEqualsElements(first[], second[])*, *findElementsLessThanMaxCode(array[])*, де *first[], second[], array[]* - формальні параметри функцій.

Термінальна гілка першої функції буде мати вигляд “**повернути result[]**”, другої функції “**повернути result[]**”, де *result[]* - змінна індексованого типу, що складається з 10 символьних значень.

◆ Розв’язання

Програмні специфікації запишемо у псевдокодi, графічні схемi у вигляді блок-схеми, та у вигляді коду.

1. Основна програма

Крок 1. Визначимо основні дії

Крок 2. Ініціалізація першого та другого масиву.

Крок 3. Ініціалізація третього масиву однаковими елементами з першого та другого масиву

Крок 4. Обробка третього масиву.

2. Підпрограма *arrayOfEqualsElements(first[], second[])*

Крок 1. Визначимо основні дії

Крок 2. Знаходження однакових елементів з масивів *first[]* і *second[]*

3. Підпрограма *findElementsLessThanMaxCode(array[])*

Крок 1. Визначимо основні дії

Крок 2. Присвоєння першого елемента *array[]* змінній *max*

Крок 3. Знаходження елемента з максимальним кодом.

Крок 4. Знаходження елементів менших за максимальний код.

◆ Псевдокод алгоритму основної програми

Крок 1

Початок

Ввід

Ініціалізація першого та другого масиву

Ініціалізація третього масиву однаковими елементами з першого та другого масиву

Обробка третього масиву

Вивід *result*[]

Кінець

Крок 2

Початок

Ввід

Повторити

Для i від 0 до 10, крок 1

$firstArray[i] = 2 * i + 42$

$secondArray[i] = 54 - 2 * i$

Все повторити

Ініціалізація третього масиву однаковими елементами з першого та другого масивів

Обробка третього масиву

Вивід *result*[]

Кінець

Крок 3

Початок

Ввід

Повторити

Для i від 0 до 10, крок 1

$firstArray[i] = 2 * i + 42$

$secondArray[i] = 54 - 2 * i$

Все повторити

$thirdArray[] = arrayOfEqualsElements(firstArray[], secondArray[])$

Обробка третього масиву

Вивід *result*[]

Кінець

Крок 4

Початок

Ввід

Повторити

Для i від 0 до 10, крок 1

$firstArray[i] = 2 * i + 42$

$secondArray[i] = 54 - 2 * i$

Все повторити

$thirdArray[] = arrayOfEqualsElements(firstArray[], secondArray[])$

$result[] = findElementsLessThanMaxCode(thirdArray[])$

Вивід $result[]$

Кінець

◆ **Псевдокод алгоритму** $arrayOfEqualsElements(first[], second[])$

Крок 1

$arrayOfEqualsElements(first[], second[])$

Знаходження однакових елементів з масивів $first[]$ і $second[]$

Повернути $result[]$

Кінець $arrayOfEqualsElements$

Крок 2

$arrayOfEqualsElements(first[], second[])$

Повторити

Для i від 0 до 10, крок 1

Повторити

Для j від 0 до 10, крок 1

$isEquals = first[i] == second[j]$

Якщо $isEquals$

То

$result[i] = first[i]$

Інакше

Все якщо

Все повторити

Все повторити

Повернути $result[]$

Кінець arrayOfEqualsElements

◆ **Псевдокод алгоритму** findElementsLessThanMaxCode(*array*[])

Крок 1

findElementsLessThanMaxCode(*array*[])

Присвоєння першого елементу *array*[] змінній *max*

Знаходження елемента з максимальним кодом.

Знаходження елементів менших за максимальний код.

Повернути *result*[]

Кінець findElementsLessThanMaxCode

Крок 2

findElementsLessThanMaxCode(*array*[])

max = *array*[0]

Знаходження елемента з максимальним кодом

Знаходження елементів менших за максимальний код

Повернути *result*[]

Кінець findElementsLessThanMaxCode

Крок 3

findElementsLessThanMaxCode(*array*[])

max = *array*[0]

Повторити

Для *i* від 0 до 10, **крок 1**

isTrue = *array*[*i*] > *max*

Якщо *isTrue*

То

max = *array*[*i*]

Інакше

Все якщо

Все повторити

Знаходження елементів менших за максимальний код

Повернути *result*[]

Кінець findElementsLessThanMaxCode

Крок 4

findElementsLessThanMaxCode(array[])

***max* = array[0]**

Повторити

Для *i* від 0 до 10, крок 1

isTrue* = array[*i*] > *max

Якщо *isTrue*

То

***max* = array[*i*]**

Інакше

Все якщо

Все повторити

Повторити

Для *i* від 0 до 10, крок 1

isTrue* = array[*i*] < *max

Якщо *isTrue*

То

***result*[*i*] = array[*i*]**

Інакше

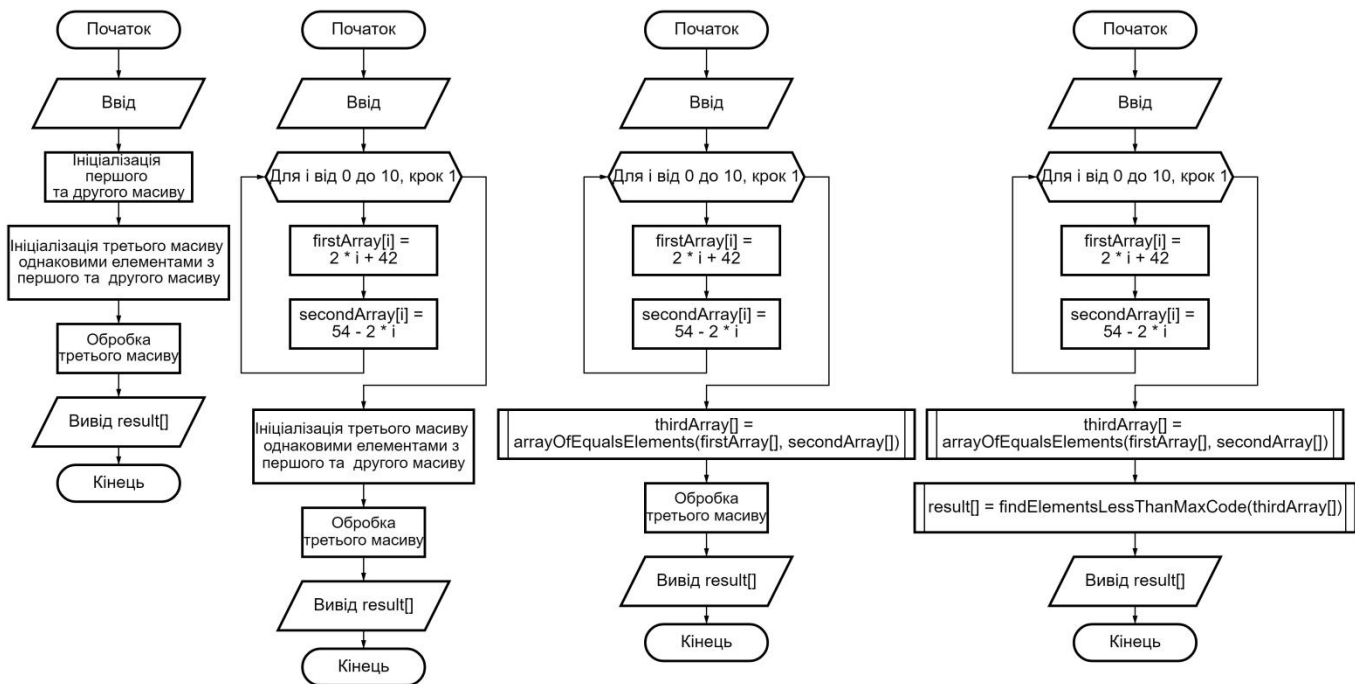
Все якщо

Все повторити

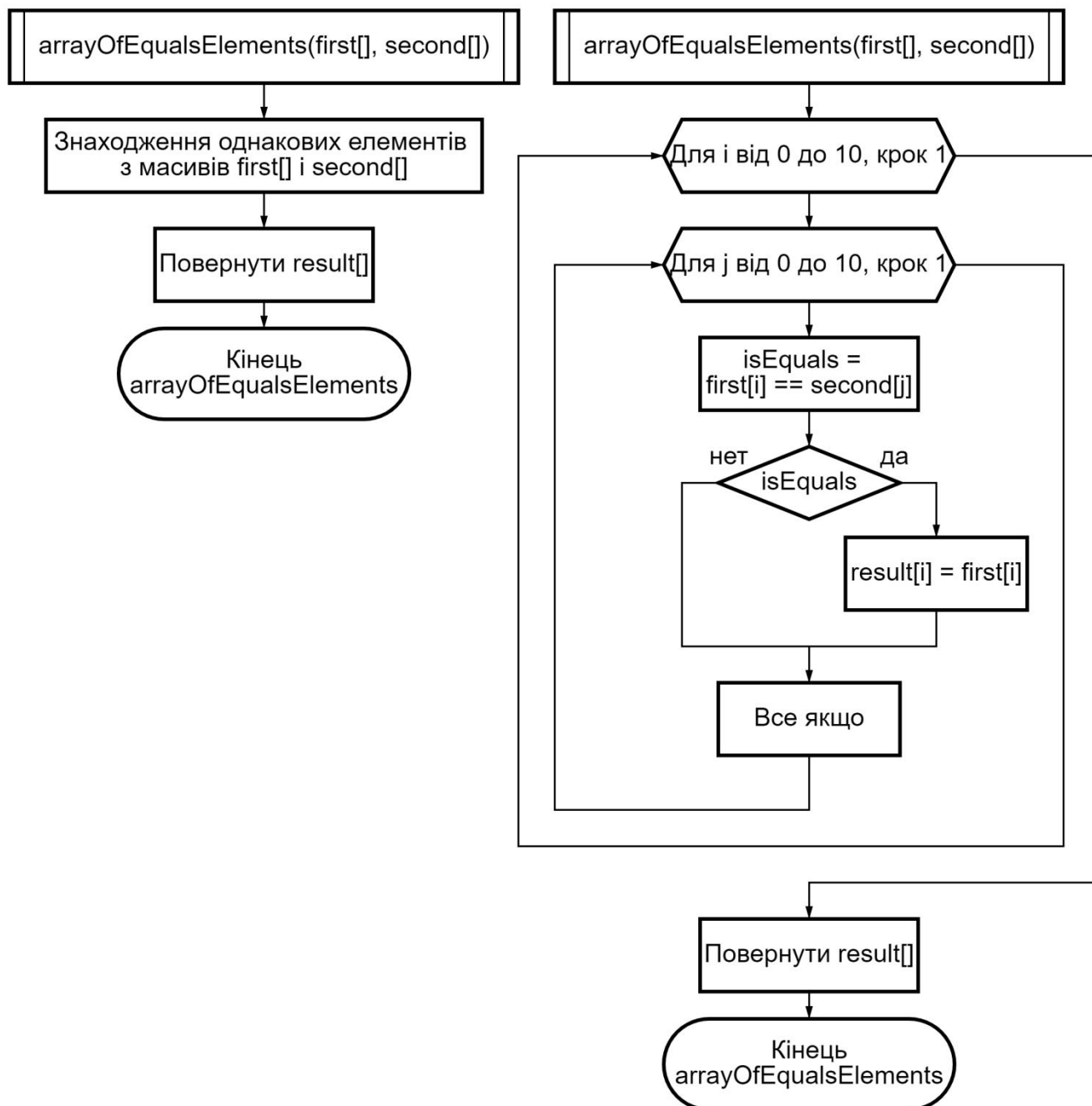
Повернути *result*[]

Кінець findElementsLessThanMaxCode

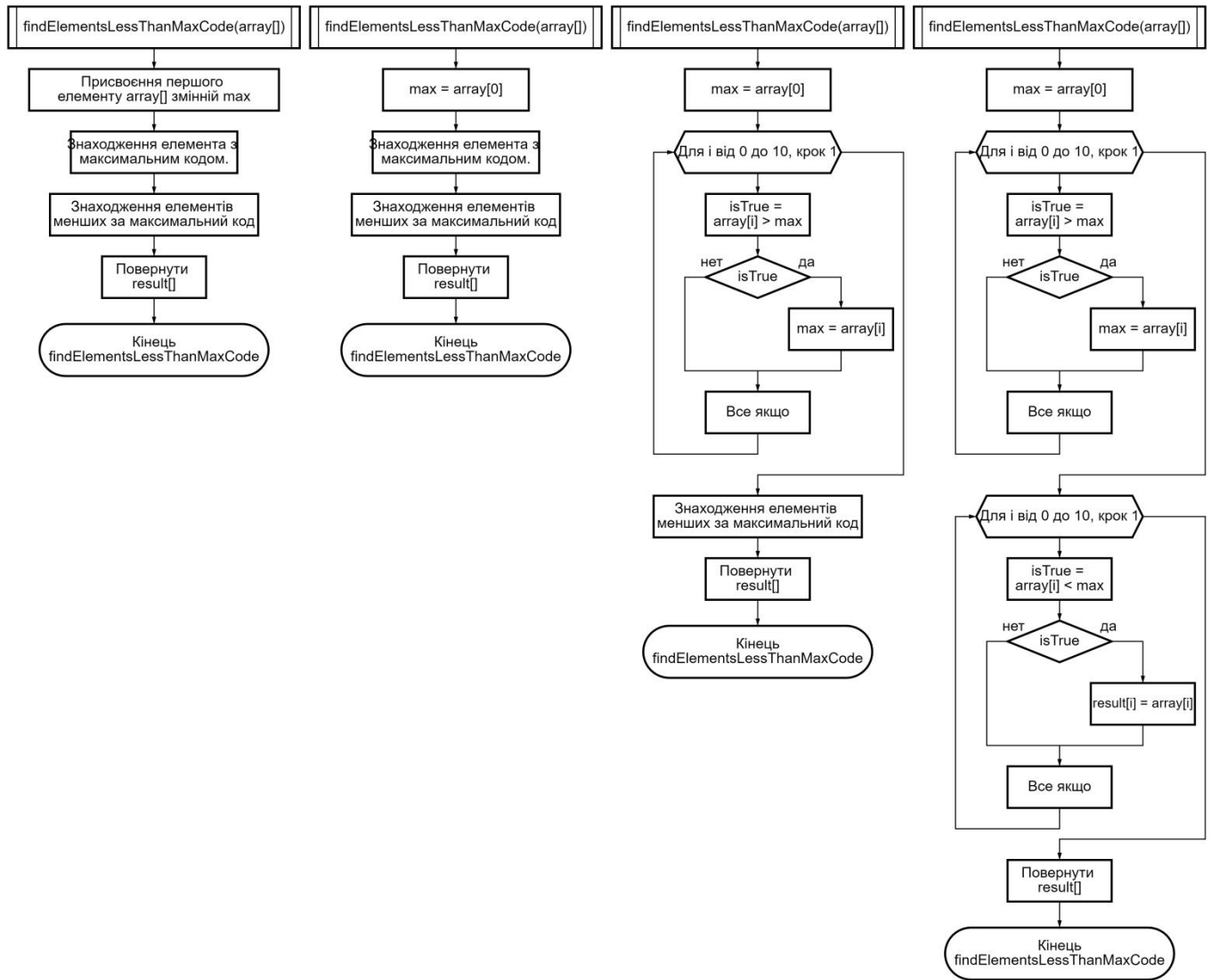
◆ Блок-схема алгоритму основної програми



◆ Блок-схема алгоритму *arrayOfEqualsElements(first[], second[])*



◆ Блок-схема алгоритму *findElementsLessThanMaxCode(array[])*



◆ Код програми

```

2
3 public class Main {
4     public static void main(String[] args) {
5         char[] firstArray = new char[10];
6         char[] secondArray = new char[10];
7         char[] thirdArray;
8         char[] result;
9         for(int i = 0; i < 10; i++){
10             firstArray[i] = (char) (2 * i + 42);
11             secondArray[i] = (char) (54 - 2 * i);
12         }
13         thirdArray = arrayOfEqualsElements(firstArray, secondArray);
14         result = findElementsLessThanMaxCode(thirdArray);
15         display(firstArray, secondArray, thirdArray, result);
16     }
17 @ private static char[] arrayOfEqualsElements(char[] first, char[] second){
18     char[] result = new char[10];
19     for(int i = 0; i < 10; i++){
20         for(int j = 0; j < 10; j++){
21             boolean isEqual = first[i] == second[j];
22             if(isEqual)
23                 result[i] = first[i];
24         }
25     }
26     return result;
27 }
28 @ private static char[] findElementsLessThanMaxCode(char[] array){
29     char[] result = new char[10];
30     char max = array[0];
31     for(int i = 0; i < 10; i++){

```

```
Main.java x Main2.java x
28 @ private static char[] findElementsLessThanMaxCode(char[] array){
29     char[] result = new char[10];
30     char max = array[0];
31     for(int i = 0; i < 10; i++){
32         boolean isTrue = array[i] > max;
33         if(isTrue){
34             max = array[i];
35         }
36     }
37     for(int i = 0; i < 10; i++){
38         boolean isTrue = array[i] < max;
39         if(isTrue)
40             result[i] = array[i];
41     }
42     return result;
43 }
44
45 private static void display(char[] first, char[] second, char[] third, char[] result){
46     System.out.println("Елементи першого масиву\n"+ Arrays.toString(first));
47     System.out.println("Елементи другого масиву\n"+Arrays.toString(second));
48     System.out.println("Елементи третього масиву\n"+Arrays.toString(third));
49     System.out.println("Елементи результуючого масиву\n"+Arrays.toString(result));
50 }
51 }
52
```

◆ Висновок

На лабораторній роботі було декомпозовано задачу на такі етапи: основна програма: визначення основних дій, ініціалізація першого та другого масиву, ініціалізація третього масиву однаковими елементами з першого та другого масиву, обробка третього масиву, підпрограма *arrayOfEqualsElements(first[], second[])*: визначення основних дій, знаходження однакових елементів з масивів *first[]* і *second[]*, підпрограма *findElementsLessThanMaxCode(array[])*: визначення основних дій, присвоєння першого елемента *array[]* змінній *max*, знаходження елемента з максимальним кодом, знаходження елементів менших за максимальний код. Було

досліджено методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набуто практичних навичок їх використання під час складання програмних специфікацій.