

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 5

Виконав студент

ІП-15, Буяло Дмитро Олександрович
(шифр, прізвище, ім'я, по батькові)

Перевірів

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

Київ 2021

Лабораторна робота 7

Дослідження лінійного пошуку в послідовностях

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

Індивідуальне завдання

Варіант 5

Завдання

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.
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної згідно з варіантом.

№	Вираз для обчислення елемента		Знайти
	1-го масиву	2-го масиву	
5	120-i	110+i	Кількість елементів, коди яких менше 115

1. Постановка задачі

1) Термінологія в формуванні задачі повністю зрозуміла та не потребує пояснень. 2) Маємо початкові формули для генерації масивів. 3) Необхідно згенерувати третій масив, що містить однакові елементи з перших двох, використовуючи вхідні дані, користуючись методом послідовного пошуку, маємо знайти кількість елементів, коди яких менше 115. 4) Як загальну властивість можна виділити те, перші два масиви генеруються за заданими формулами. 5) Існує багато розв'язків даної задачі, будемо використовувати, на мою думку, найпростіший, найефективніший та найшвидший. 6) Даних цілком достатньо, всі потрібні та припущень робити не потрібно.

2. Побудова математичної моделі

Складемо таблицю імен змінних

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
Перший масив	Цілочисельний	firstArr	Проміжні дані
Другий масив	Цілочисельний	secondArr	Проміжні дані
Третій масив	Цілочисельний	thirdArr	Проміжні дані
Генерація масивів	Процедура	createArr	Проміжні дані
Лічильник 1	Цілочисельний	i	Проміжні дані
Лічильник 2	Цілочисельний	j	Проміжні дані
Розмір масиву	Процедура	lengthArr	Проміжні дані
Розмір третього масива	Цілочисельний	lengthThirdArr	Проміжні дані
Вивід масивів	Процедура	outputArr	Проміжні дані
Результат	Процедура	result	Результат
Кількість елементів	Цілочисельний	counter	Результат

Щоб знайти кількість елементів з кодом менше 115 у третьому масиві, нам потрібно застосувати підпрограми, в яких ми будемо генерувати масив з початкових значень, знаходити довжину масиву, виводити в консоль отримані три масиви та підпрограму власне для виконання умови. Вхідними параметрами функції createArr є пусті масиви, підпрограма буде створювати масиви за заданою формулою генерації. Вхідними параметрами функції lengthArr є три масиви, підпрограма буде додавати елементи в третій масив за заданою умовою, вираховувати та повертати довжину масиву. Вхідними параметрами функції outputArr є три масиви та довжина масиву, підпрограма виводить кожне значення кожного масиву. Вхідними параметрами функції result є масив та довжина масиву, підпрограма рахує, скільки елементів третього масиву має код менше 115.

Всі виклики функцій відбуваються в основній програмі.

Розв’язання

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

Крок 1. Визначимо основні дії

Крок 2. Деталізуємо дію знаходження кількості елементів, код яких менше 115.

3. Псевдокод алгоритму

Основна програма

Крок 1

Початок

Створення firstArr,
secondArr, thirdArr

Створення масивів

Визначення довжини масиву

Вивід масивів

Обчислення результати

Кінець

Крок 2

Початок

firstArr[10], secondArr[10], thirdArr[10]

createArr(firstArr, secondArr)

lengthThirdArr=lengthArr(firstArr,

secondArr, thirdArr)

output(firstArr, secondArr, thirdArr,

lengthThirdArr)

result(thirdArr, lengthThirdArr)

Кінець

Підпрограми:

createArr(firstArr, secondArr)
Початок
Повторити для i від 0 до 10
firstArr[i]=120-i
secondtArr[i]=120-i
Все повторити
Кінець

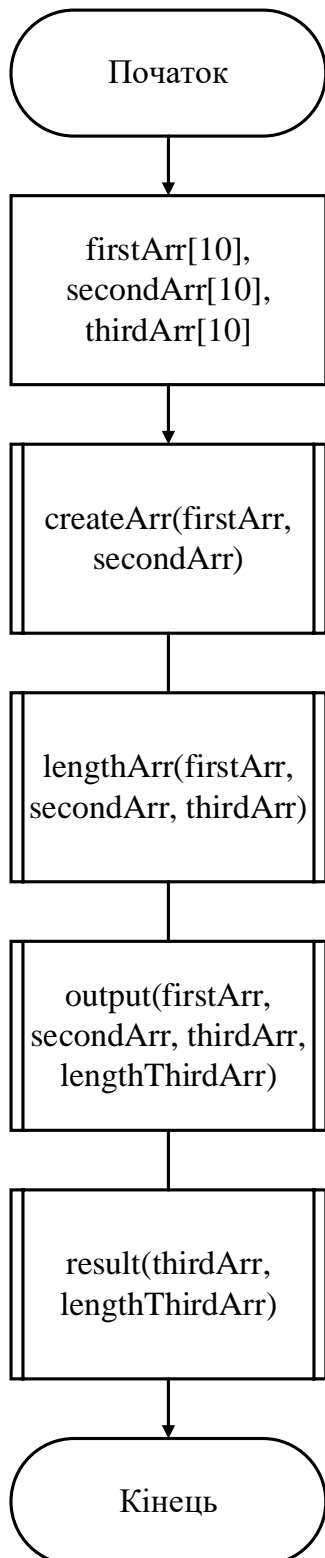
lengthArr(firstArr, secondArr, thirdArr)
Початок
lengthThirdArr = 0;
Повторити для i від 0 до 10
Повторити для j від 0 до 10
Якщо firstArr[i] == secondArr[j]
то
thirdArr[lengthThirdArr] = firstArr[i]
lengthThirdArr ++
Все Якщо
Все повторити
Все повторити
Кінець

output (firstArr, secondArr, thirdArr, lengthThirdArr)
Початок
Повторити для i від 0 до lengthThirdArr
Виведення firstArr[i], secondArr[i], thirdArr[i]
Все повторити
Кінець

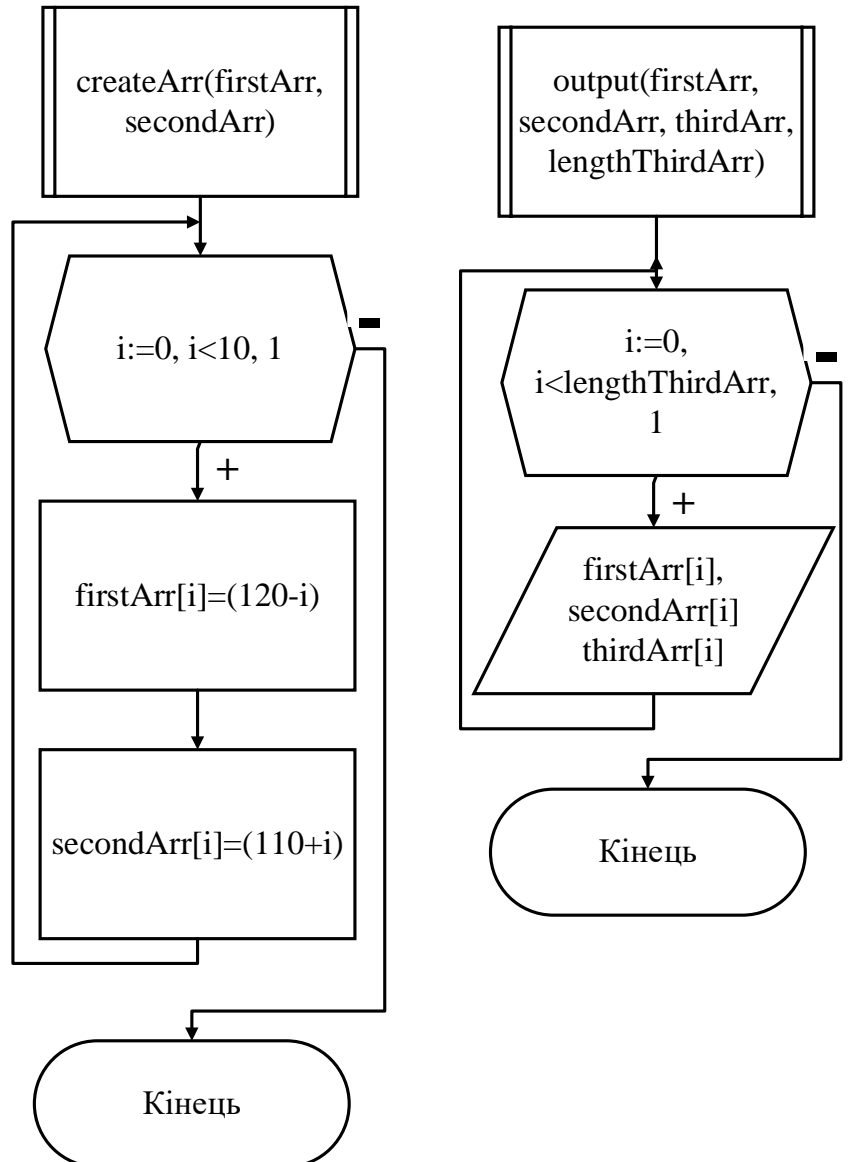
result (thirdArr, lengthThirdArr)
Початок
counter =0
Повторити для i від 0 до lengthThirdArr
Якщо thirdArr[i]<115
то
counter++
Все Якщо
Все повторити
Кінець

4. Блок-схема алгоритму

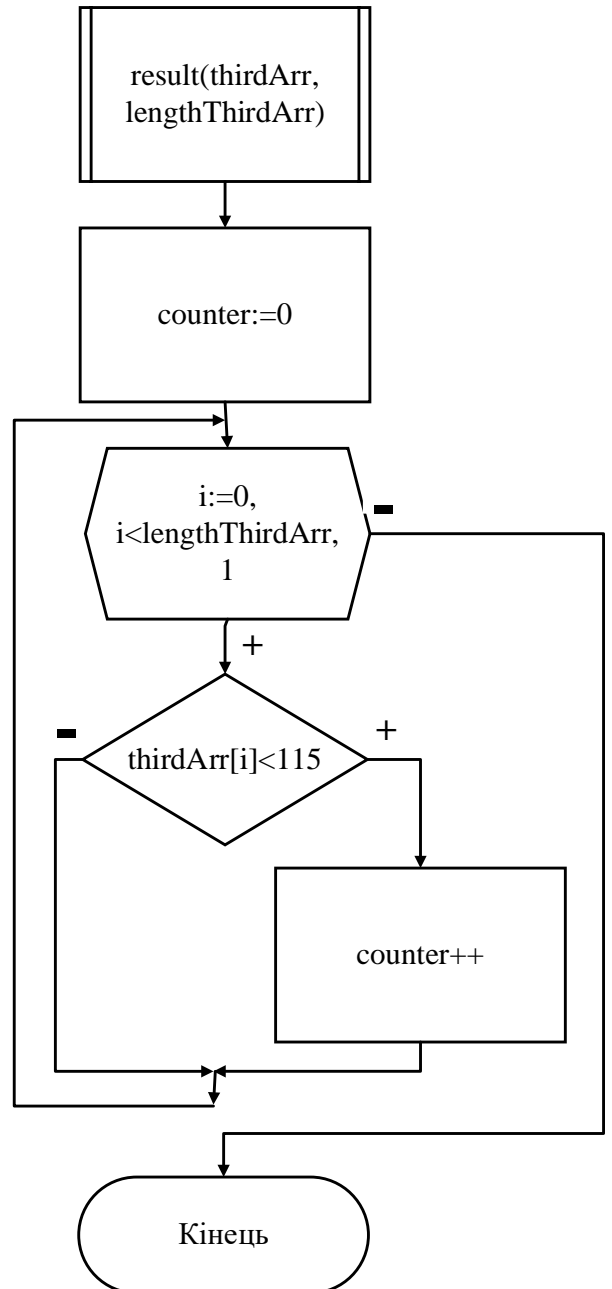
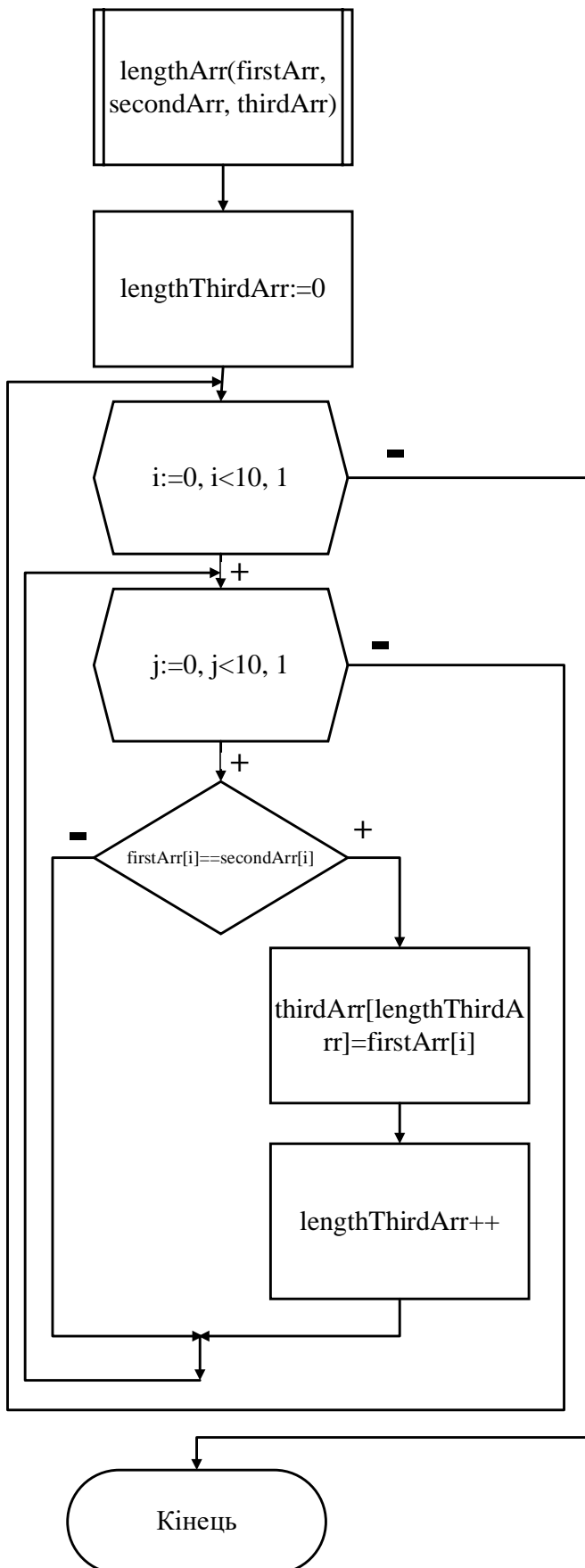
Основна програма



Підпрограми



Підпрограми



5. Код програми

Код написаний на мові програмування Java

```

Main.java x
5  public static void main(String[] args) {
6      char[] firstArr = new char[10];
7      char[] secondArr = new char[10];
8      char[] thirdArr = new char[10];
9      createArr(firstArr, secondArr);
10     int lengthThirdArr=lengthArr(firstArr,secondArr,thirdArr);
11     outputArr(firstArr,secondArr,thirdArr,lengthThirdArr);
12     System.out.println("result = "+result(thirdArr,lengthThirdArr));
13 }
14 public static void createArr(char[] a, char[] b) {
15     for(int i=0;i<10;i++) {
16         a[i]= (char) (120-i);
17         b[i]= (char) (110+i);
18     }
19 }
20 public static int lengthArr(char[] a, char[] b, char[] c) {
21     int lengthThirdArr = 0;
22     for(int i=0;i<10;i++) {
23         for(int j = 0; j<10; j++) {
24             if(a[i]==b[j]) {
25                 c[lengthThirdArr] =a[i];
26                 lengthThirdArr ++;
27             }
28         }
29     }
30     return lengthThirdArr;
31 }
    
```

```
32 public static void outputArr(char[] a, char[] b, char[] c, int lengthArr) {
33     System.out.println("a - firstArr, b - secondArr, c - thirdArr");
34     for(int i=0; i<=lengthArr;i++) {
35         System.out.println("a["+i+"] = "+a[i]+"    b["+i+"] = "+b[i]+"    c["+i+"] = "+c[i])
36     }
37 }
38 public static int result(char[] arr, int lengthArr) {
39     int counter=0;
40     for(int i =0; i<lengthArr; i++) {
41         if(arr[i]<115) {
42             counter++;
43         }
44     }
45     return counter;
46 }
47 }
```

Вивід оформлений не викликом функції, а по рядкам через особливості виводу результату у консоль, а саме виведення у стовпчик.

Для економії місця була зроблена заміна назв масивів(перший рядок консолі).

```
a - firstArr, b - secondArr, c - thirdArr
a[0] = x    b[0] = n    c[0] = w
a[1] = w    b[1] = o    c[1] = v
a[2] = v    b[2] = p    c[2] = u
a[3] = u    b[3] = q    c[3] = t
a[4] = t    b[4] = r    c[4] = s
a[5] = s    b[5] = s    c[5] = r
a[6] = r    b[6] = t    c[6] = q
a[7] = q    b[7] = u    c[7] = p
a[8] = p    b[8] = v    c[8] = o
a[9] = o    b[9] = w    c[9] = 
result = 4
```

6. Висновки

Ми дослідили методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набули практичних навичок їх використання під час складання програмних специфікацій. В результаті виконання лабораторної роботи ми отримали алгоритм для створення третього масиву, що містить однакові елементи з перших двох, та знаходження в ньому кількості елементів масиву, які менше 115. Дискретували задачу на 2 кроки: визначили основні дії, деталізували знаходження count/result, тобто кількість елементів фінального масиву, що менше 115, за допомогою підпрограм. Алгоритм є ефективним та результативним, бо забезпечує розв'язок за мінімальний час із мінімальними витратами ресурсів та отримує чіткий кінцевий результат.