

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 29

Виконав студент ІП-13 Романюк Діана Олексіївна
(шифр, прізвище, ім'я, по батькові)

Перевірив Всчерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

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 29

Постановка задачі:

29	$i + 58$	$63 - i$	Елементи, які менші за середньоарифметичне
----	----------	----------	--

Побудова математичної моделі:

Таблиця імен змінних

Змінна	Тип	Ім'я	Призначення
Масив	Символьний	FirstArray	Проміжні дані
Масив	Символьний	SecondArray	Проміжні дані
Масив	Символьний	ThirdArray	Проміжні дані
Середнє значення	Дійсний	Avg	Проміжні дані
Сума кодів елементів	Цілий	Sum	Проміжні дані
Лічильник к-сті елементів	Цілий	n	Проміжні дані
Лічильник	Цілий	i	Проміжні дані
Лічильник	Цілий	j	Проміжні дані
Середнє арифметичне кодів елементів третього масиву	Дійсний	aSum	Проміжні дані

inFirstArray() – функція(підпрограма) для ініціалізації першого масиву.

inSecondArray() – функція(підпрограма) для ініціалізації другого масиву.

inThirdArray() – функція(підпрограма) для ініціалізації третього масиву.

Out() – функція(підпрограма) для виведення масивів.

Average() – функція(підпрограма) для обчислення середнього арифметичного кодів елементів третього масиву.

Res() – функція(підпрограма) для порівняння кожного елементу масиву з середнім арифметичним і виведення результату.

(float) – приведення числа до дійсного типу, щоб при обрахунку і подальшому порівнянні коректно шукалось перше значення.

FirstArray[i] = $(i + 58)$ – шукаємо елементи першого масиву за формулою.

SecondArray[i] = $(63 - i)$ – шукаємо елементи другого масиву за формулою.

Sum += ThirdArray[i] – обрахунок суми кодів елементів третього масиву.

aSum = (float)Sum / n (де n – лічильник, що буде збільшуватись на один з кроком додання елементу третього масиву).

Задача:

Ініціалізувати два масиви задані формулами, записати значення однакових елементів у третьому і вивести їх. Знайти середнє арифметичне кодів цих елементів та вивести.

Псевдокод алгоритму:

Крок 1

початок

FirstArray[10]

SecondArray[10]

ThirdArray[10]

inFirstArray(FirstArray)

inFirstArray(FirstArray)

inThirdArray(FirstArray, SecondArray, ThirdArray)

Out(FirstArray)

Out(SecondArray)

Out(ThirdArray)

Avg = Average(ThirdArray)

Res(ThirdArray, Avg)

кінець

Псевдокод підпрограм:

підпрограма

inFirstArray(FirstArray):

повторити для i від 0 до 10

 FirstArray[i] = (i + 58)

все повторити

все підпрограма

підпрограма

inSecondArray(SecondArray):

повторити для i від 0 до 10

 SecondArray[i] = (63 - i)

все повторити

все підпрограма

підпрограма

inThirdArray(FirstArray, SecondArray, ThirdArray):

повторити для i від 0 до 10

повторити для j від 0 до 10

якщо FirstArray[i] = SecondArray[j]

 ThirdArray[i] = FirstArray[j]

все повторити

все повторити

все підпрограма

підпрограма

Out(Array):

повторити для i від 0 до 10

 вивід Array[i]

все повторити

все підпрограма

підпрограма

Average(ThirdArray):

Sum = 0

n = 0

повторити для i від 0 до 10

якщо ThirdArray[i] != 0

n++

Sum += ThirdArray[i]

все повторити

якщо n == 0

то return 0

інакше aSum = (float)Sum / n

return aSum

все якщо

все підпрограма

підпрограма

Res(ThirdArray, aSum):

повторити для i від 0 до 10

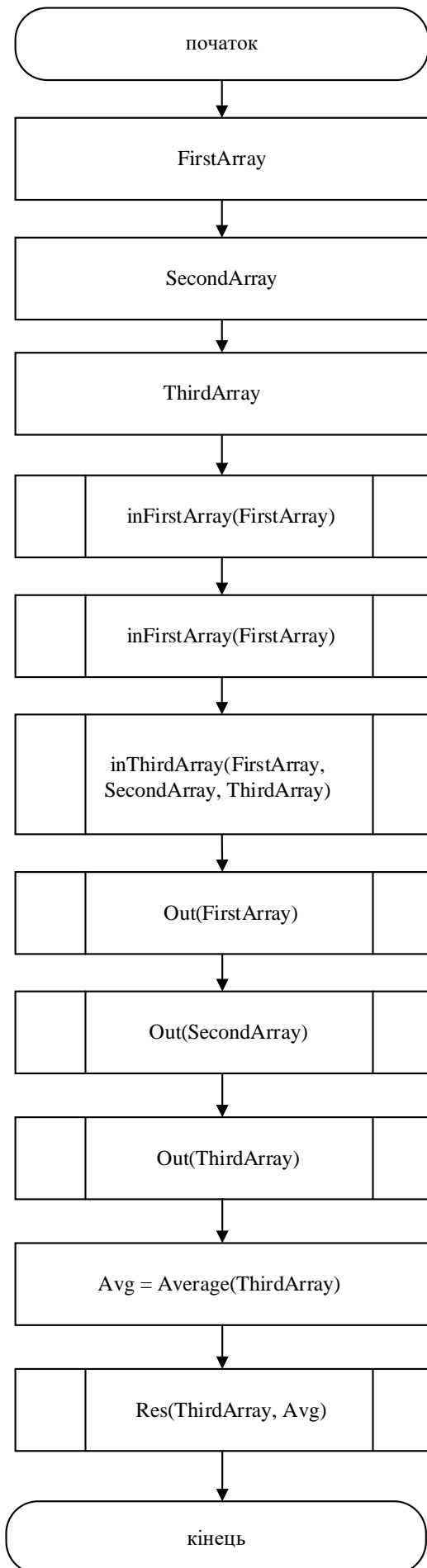
якщо ThirdArray[i] < aSum

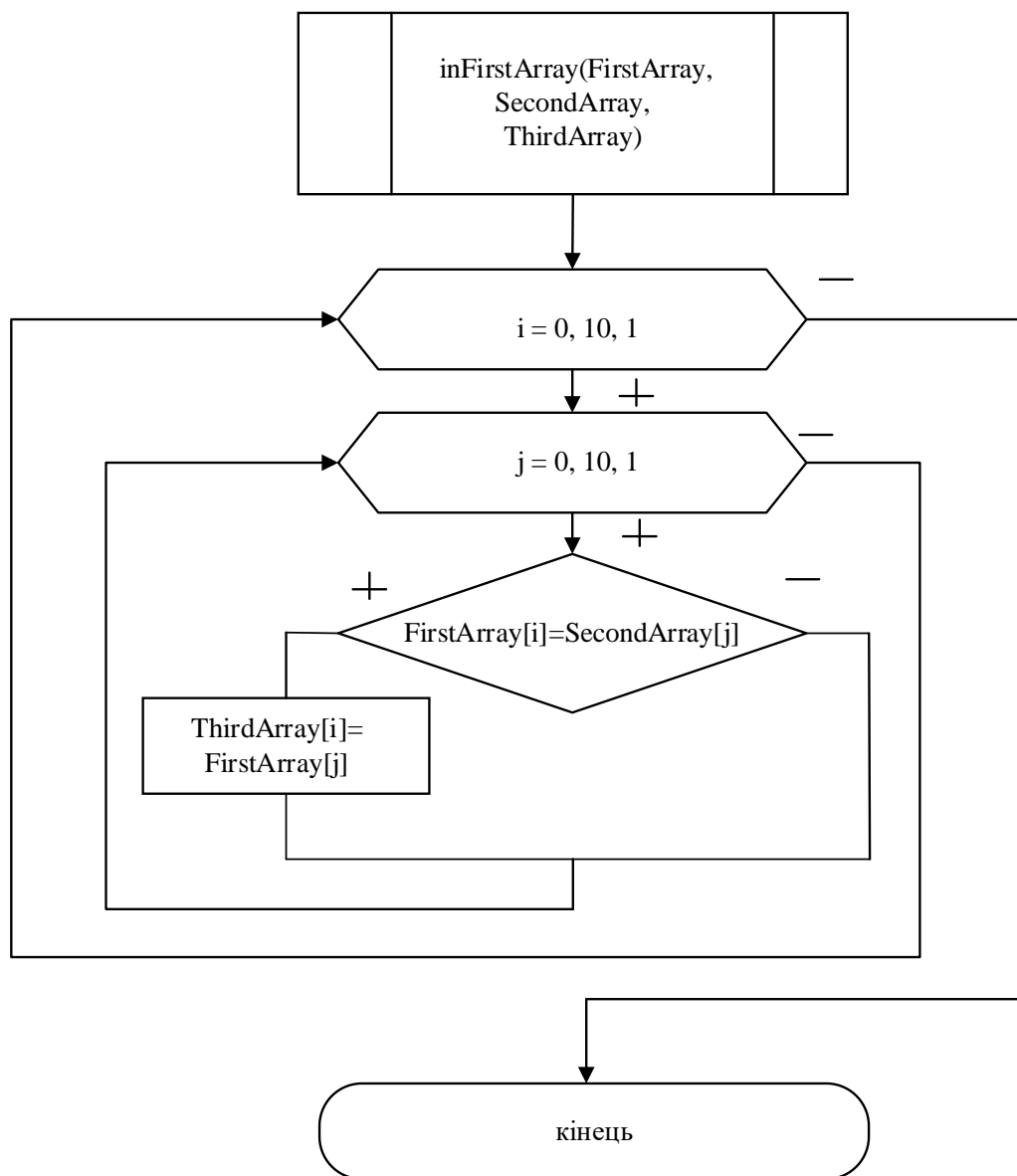
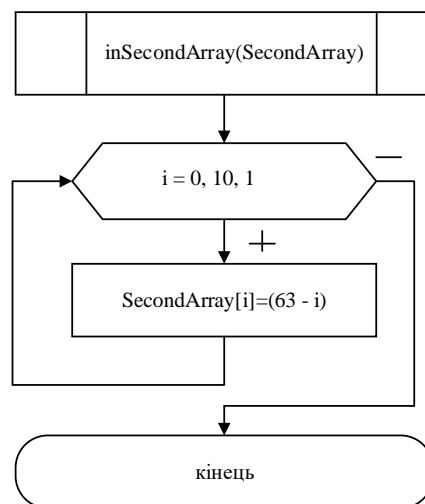
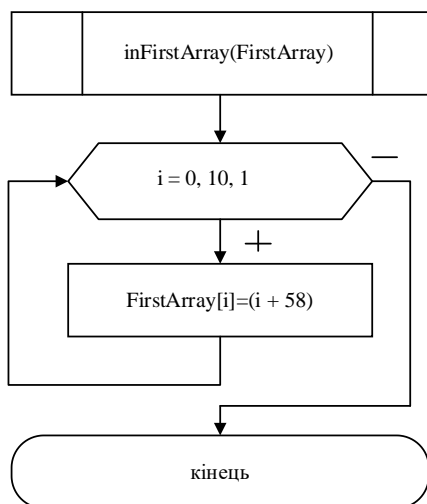
вивід ThirdArray[i]

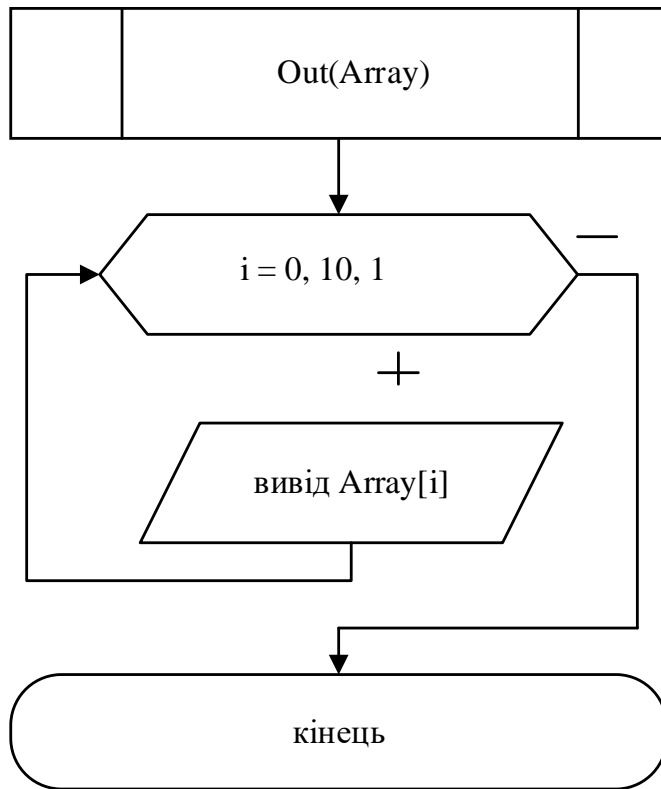
все повторити

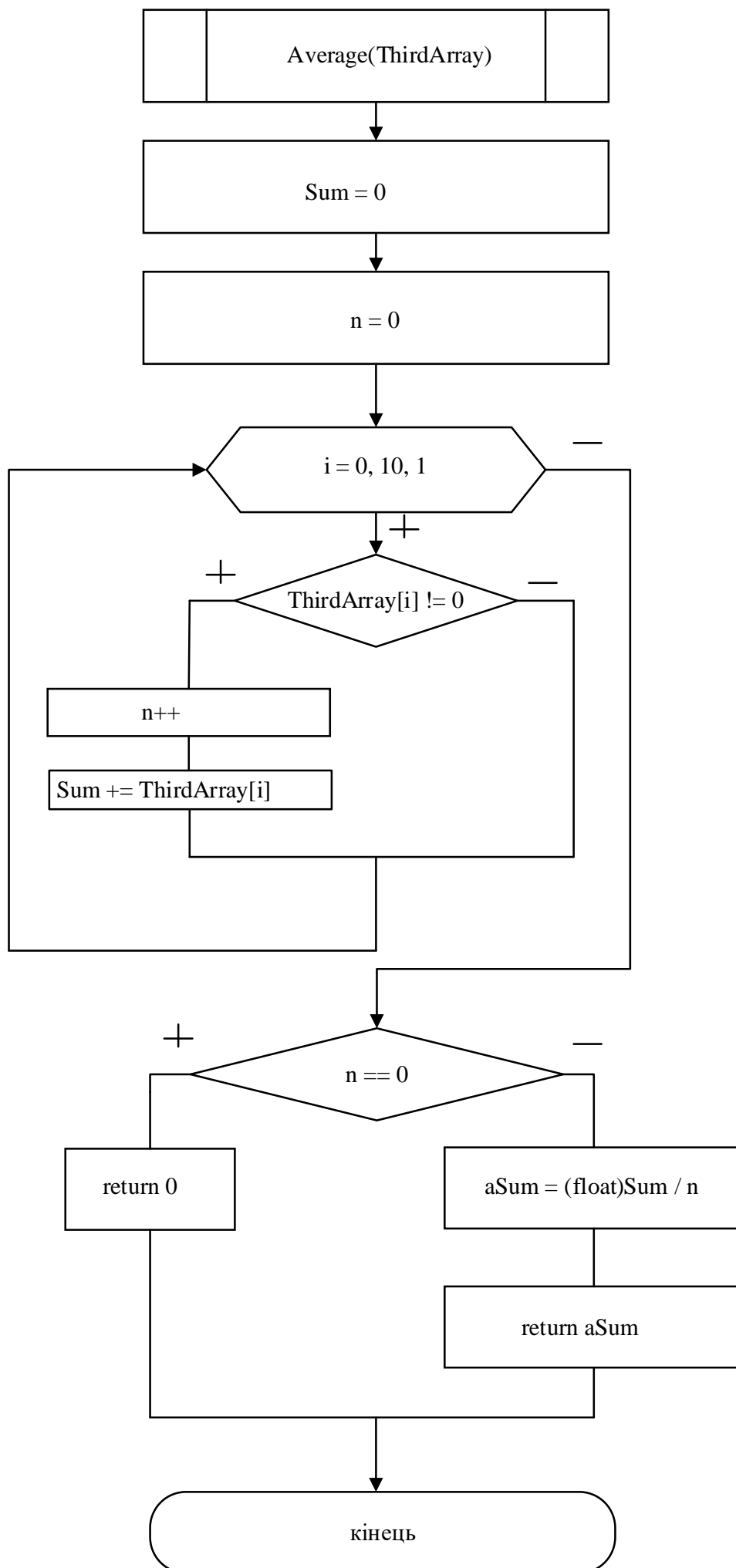
все підпрограма

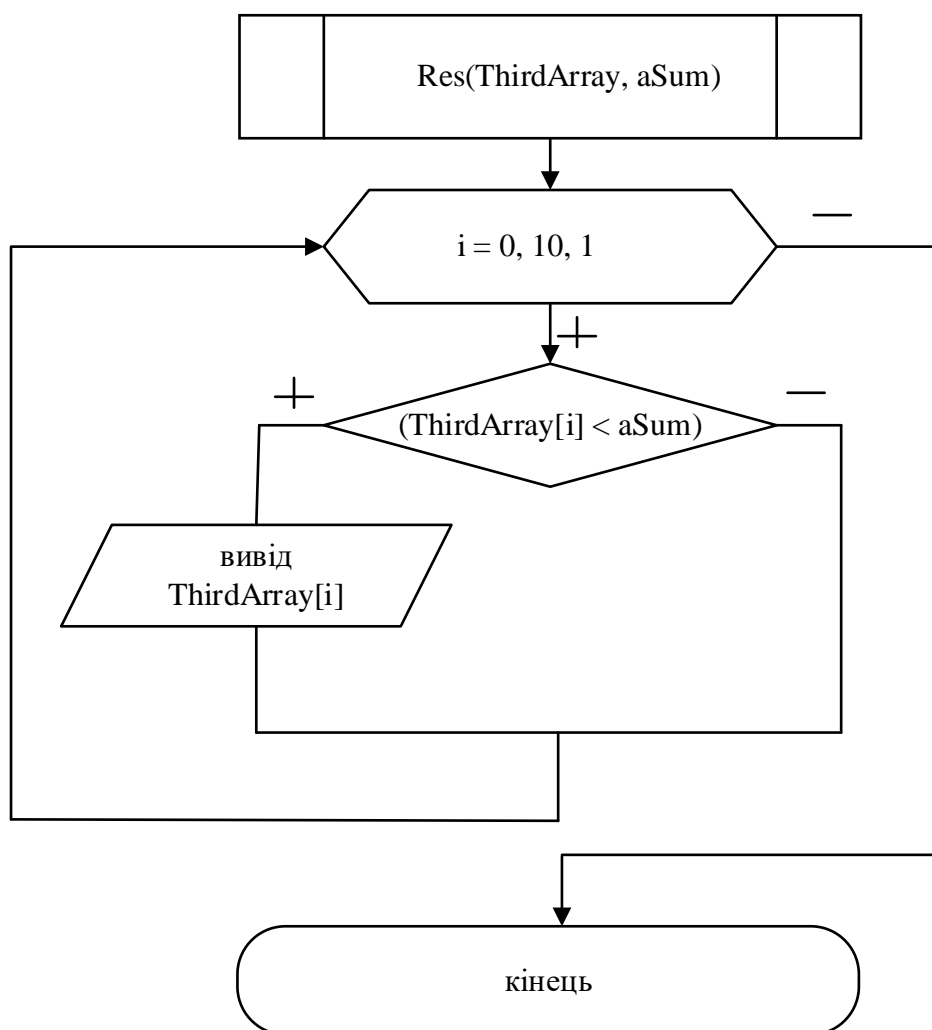
Блок-схеми:











Код програми:

```
7.cpp*  [X]
Lab7 (Глобальная область)
4  #include <stdio.h>
5
6
7  using namespace std;
8
9  void inFirstArray(char*);
10 void inSecondArray(char*);
11 void inThirdArray(char*, char*, char*);
12 void Out(char*);
13 float Average(char*);
14 void Res(char*, float);
15
16
17
18 int main()
19 {
20     char FirstArray[10] = {};
21     char SecondArray[10] = {};
22     char ThirdArray[10] = {};
23
24     inFirstArray(FirstArray);
25     inSecondArray(SecondArray);
26     inThirdArray(FirstArray, SecondArray, ThirdArray);
27
28     Out(FirstArray);
29     Out(SecondArray);
30     Out(ThirdArray);
31
32     float Avg = Average(ThirdArray);
33     Res(ThirdArray, Avg);
34 }
35 void inFirstArray(char* FirstArray) {
36     for (int i = 0; i < 10; i++) {
37         *(FirstArray + i) = char(i + 58);
38     }
39 }
40 void inSecondArray(char* SecondArray) {
41     for (int i = 0; i < 10; i++) {
42         *(SecondArray + i) = char(63 - i);
43     }
44 }
45
46 void inThirdArray(char* FirstArray, char* SecondArray, char* ThirdArray) {
47     for (int i = 0; i < 10; i++) {
48         for (int j = 0; j < 10; j++) {
49             if (*(FirstArray + i) == *(SecondArray + j))
50                 *(ThirdArray + i) = *(FirstArray + i);
51         }
52     }
53 }
54
```

```
55 void Out(char* Array){
56     for (int i = 0; i < 10; i++) {
57         cout << *(Array + i) << " ";
58     }
59     cout << endl;
60 }
61
62 float Average(char* ThirdArray) {
63     int Sum = 0;
64     int n = 0;
65     for (int i = 0; i < 10; i++) {
66         if (ThirdArray[i] != 0) {
67             n++;
68             Sum += ThirdArray[i];
69         }
70     }
71     if (n == 0) return 0;
72     float aSum = (float)Sum / n;
73     return aSum;
74 }
75
76 void Res(char* ThirdArray, float aSum) {
77     for (int i = 0; i < 10; i++) {
78         if (ThirdArray[i] < aSum)
79             cout << ThirdArray[i] << " ";
80     }
81 }
```



Консоль отладки Microsoft Visual Studio

```
: ; < = > ? @ A B C
? > = < ; : 9 8 7 6
: ; < = > ?
: ; <
```

Висновки:

Під час виконання лабораторної роботи досліджено методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях. Отримано практичні навички їх використання під час складання лінійних програмних специфікацій. Побудовано математичну модель задачі та таблицю імен змінних. Розроблено псевдокод вирішення даної математичної задачі. Умовно розбито виконання коду на кроки, а також описано його виконання за допомогою створення відповідної блок-схеми. Перевірено умовне виконання коду за допомогою випробування алгоритму.