

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 22

Виконав студент \_\_\_\_\_ Мєшков\_Андрій\_Ігорович\_\_\_\_\_

(шифр, прізвище, ім'я, по батькові)

Перевірив \_\_\_\_\_ Вєчерковська 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 невпорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій.

### Варіант 22

#### Завдання

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису трьох змінних індексованого типу з 10 символічних значень.
2. Ініціювання двох змінних виразами згідно з варіантом (табл. 1).
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної згідно з варіантом.

22	92 - 3 * i	71 + 3 * i	Середнє арифметичне елементів, коди яких менше 82
----	------------	------------	---

**Постанова задачі.** Згенерувати два масиви за заданими формулами:  $\text{first}=92-3*i$ ,  $\text{second}=71+3*i$ . Створити третій масив як перетин значень перших двох. Використовуючи метод послідовного пошуку, знайти елементи, коди яких менше 82.

Результатом розв'язку є виведення початкових масивів, а також середнє арифметичне знайдених елементів.

**Побудова математичної моделі:** для більшої наочності складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
Перший масив	Символьний	first	Проміжні дані, результат
Другий масив	Символьний	second	Проміжні дані, результат
Третій масив	Символьний	third	Проміжні дані, результат
Перший параметр функцій	Символьний	a	Проміжні дані
Другий параметр функцій	Символьний	b	Проміжні дані
Третій параметр функцій	Символьний	c	Проміжні дані

Кількість елементів масиву	Цілочисельний	<b>n</b>	Проміжні дані
Параметр циклу	Цілочисельний	<b>i</b>	Проміжні дані
Параметр додаткового циклу	Цілочисельний	<b>j</b>	Проміжні дані
Сума знайдених елементів	Цілочисельний	<b>sum</b>	Проміжні дані
Кількість знайдених елементів	Цілочисельний	<b>num</b>	Проміжні дані
Середнє арифметичне	Дійсний	<b>av</b>	Результат
Створення масивів <b>first</b> та <b>second</b>	Процедура	<b>create</b>	Початкові дані
Виведення масивів <b>first</b> та <b>second</b>	Процедура	<b>output</b>	Початкові дані
Створення масиву <b>third</b>	Процедура	<b>third_arr</b>	Початкові дані
Виведення масиву <b>third</b>	Процедура	<b>out_th</b>	Початкові дані
Пошук середнього арифметичного	Процедура	<b>average</b>	Початкові дані

Масиви **first** та **second** згенеруємо за допомогою арифметичного циклу, за заданою формулою всередині функції **create**(параметри **a** та **b**). Вхідними даними ф-ції є порожні масиви з 10 «зарезервованими» місцями **first** та **second**. Одразу виведемо значення масивів у функції **output**(параметри **a** та **b**) по-елементно, використовуючи арифметичний цикл. За допомогою внутрішнього і зовнішнього арифметичних циклів знайдемо значення третього масиву **third** у ф-ції **third\_arr**, порівнюючи значення **first** та **second**. Виведемо масив **third** за допомогою ф-ції **out\_th**. У ф-ції **average** знайдемо суму **sum** та кількість **num** елементів, код яких менше за 82 та порахуємо середнє арифметичне **av**.

Виклик всіх функцій відбувається в основній програмі.

*Розв'язання.* Програмні специфікації запишемо у псевдокоді та у графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію генерації двох масивів за допомогою підпрограми.

Крок 3. Деталізуємо дію виведення двох масивів за допомогою підпрограми.

Крок 4. Деталізуємо дію генерації третього масиву за допомогою підпрограми.

Крок 5. Деталізуємо дію виведення третього масиву за допомогою підпрограми.

Крок 6. Деталізуємо дію знаходження середнього арифметичного за допомогою підпрограми.

## **Псевдокод**

### **Програма:**

#### **Початок**

n:=10

create(first, second, n)

output(first, second, n)

third\_arr(first, second, third, n)

out\_th(third, n)

average(third, n)

#### **Кінець**

## Підпрограми:

**create(a, b, n)**

**повторити**

**для i від 0 до n**

**a[i]:= 92 – 3 \* i**

**b[i]:= 71 + 3\*i**

**все повторити**

**кінець**

**output(a, b, n)**

Вивести "First array: "

**повторити**

**для i від 0 до n**

**Вивести a[i]**

**все повторити**

Вивести "Second array: "

**повторити**

**для i від 0 до n**

**Вивести b[i]**

**все повторити**

**кінець**

**third\_arr(a, b, c, n):**

**повторити**

**для i від 0 до n**

**c[i]:=0**

**все повторити**

**повторити**

**для i від 0 до n**

**повторити**

**для j від 0 до n**

**якщо a[i]==b[j]**

**то**

**c[i]:=a[i]**

**все повторити**

**все повторити**

**кінець**

**out\_th(c, n)**

Вивести "Third array: "

**повторити**

**для i від 0 до 10**

    Вивести c[i]

**все повторити**

**кінець**

**average(c, n)**

sum:=0

num:=0

**повторити**

**для i від 0 до n**

**якщо** c[i]<82

**то**

            sum:=sum+c[i]

**якщо** c[i]!=0

**то**

                num:=num+1

**все повторити**

av:=sum/num

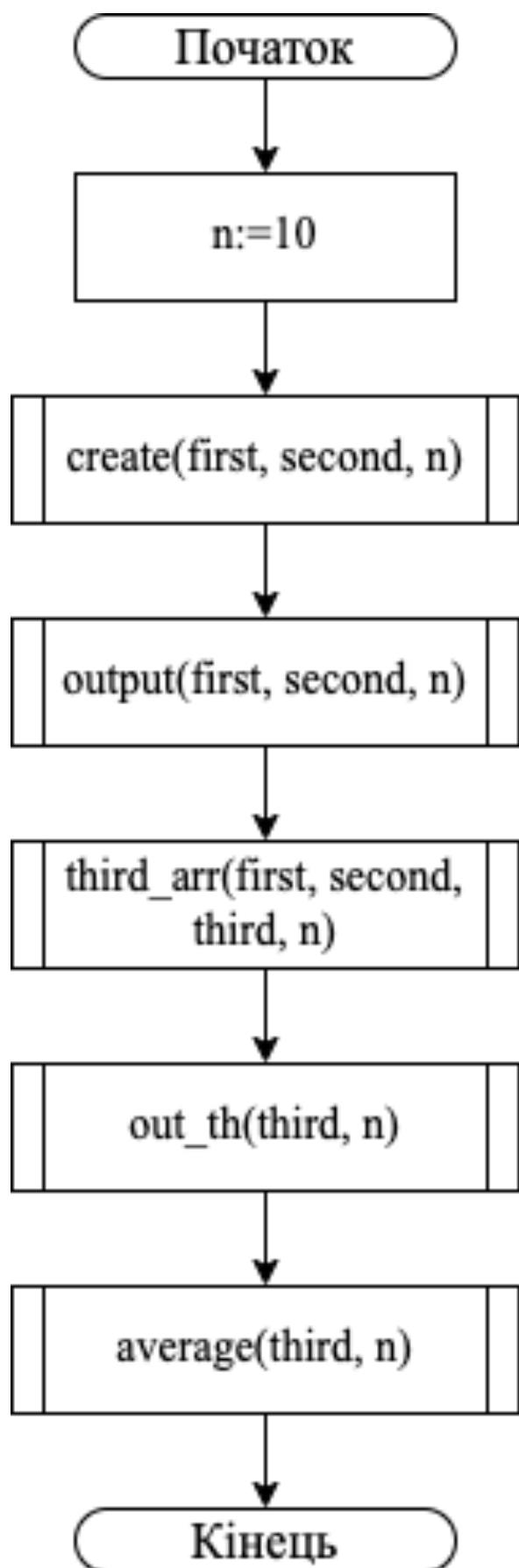
Вивести "Average: "

Вивести av

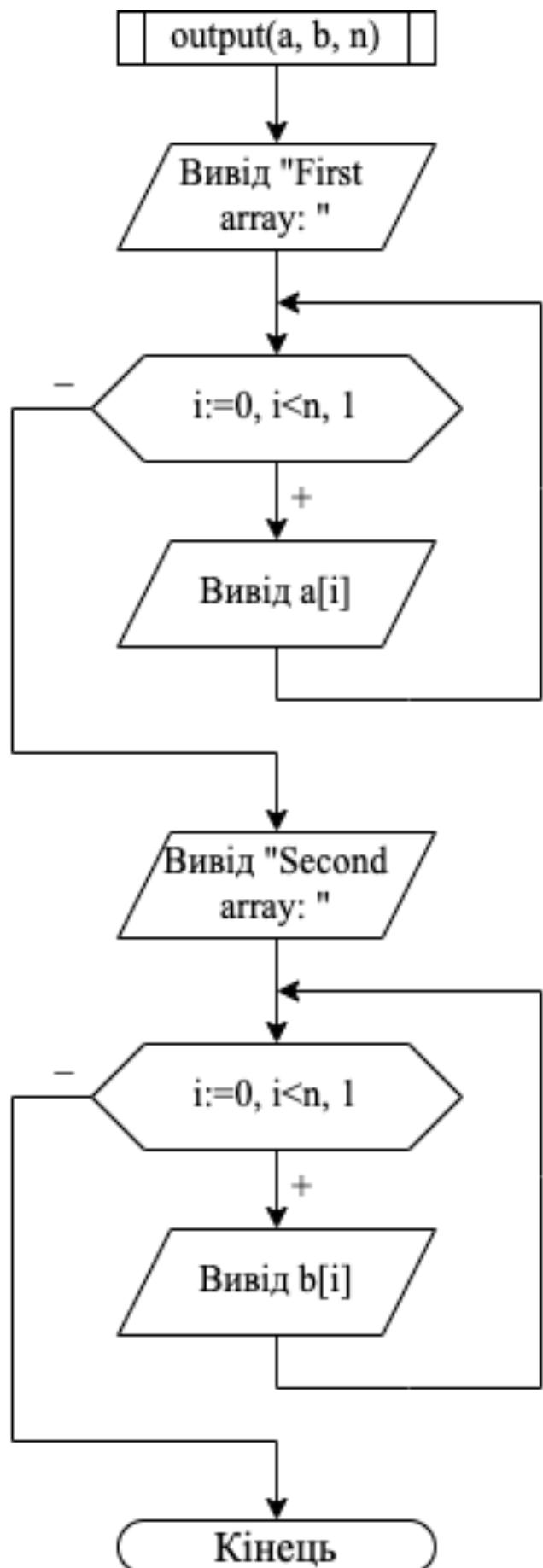
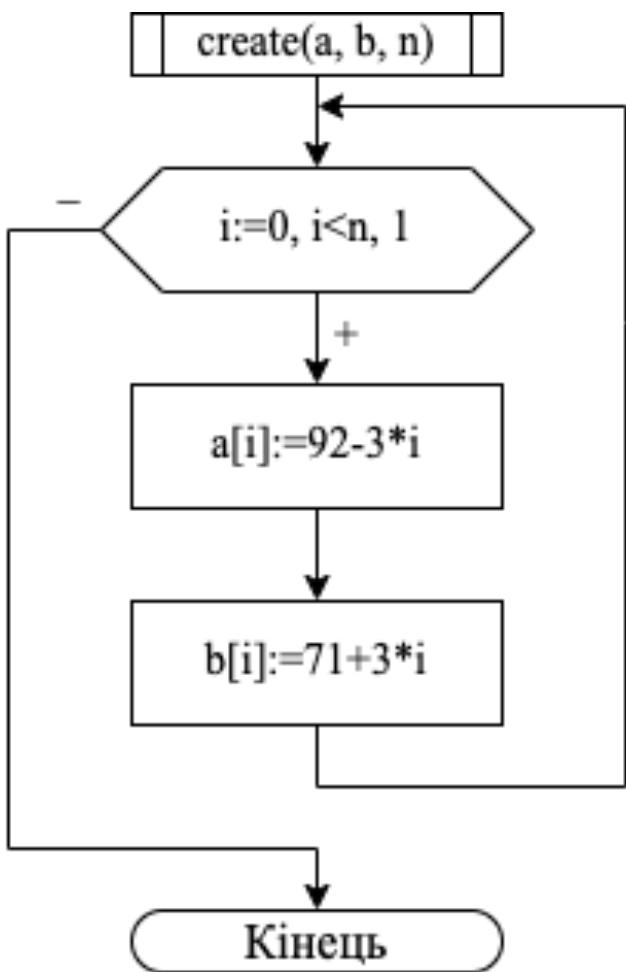
**кінець**

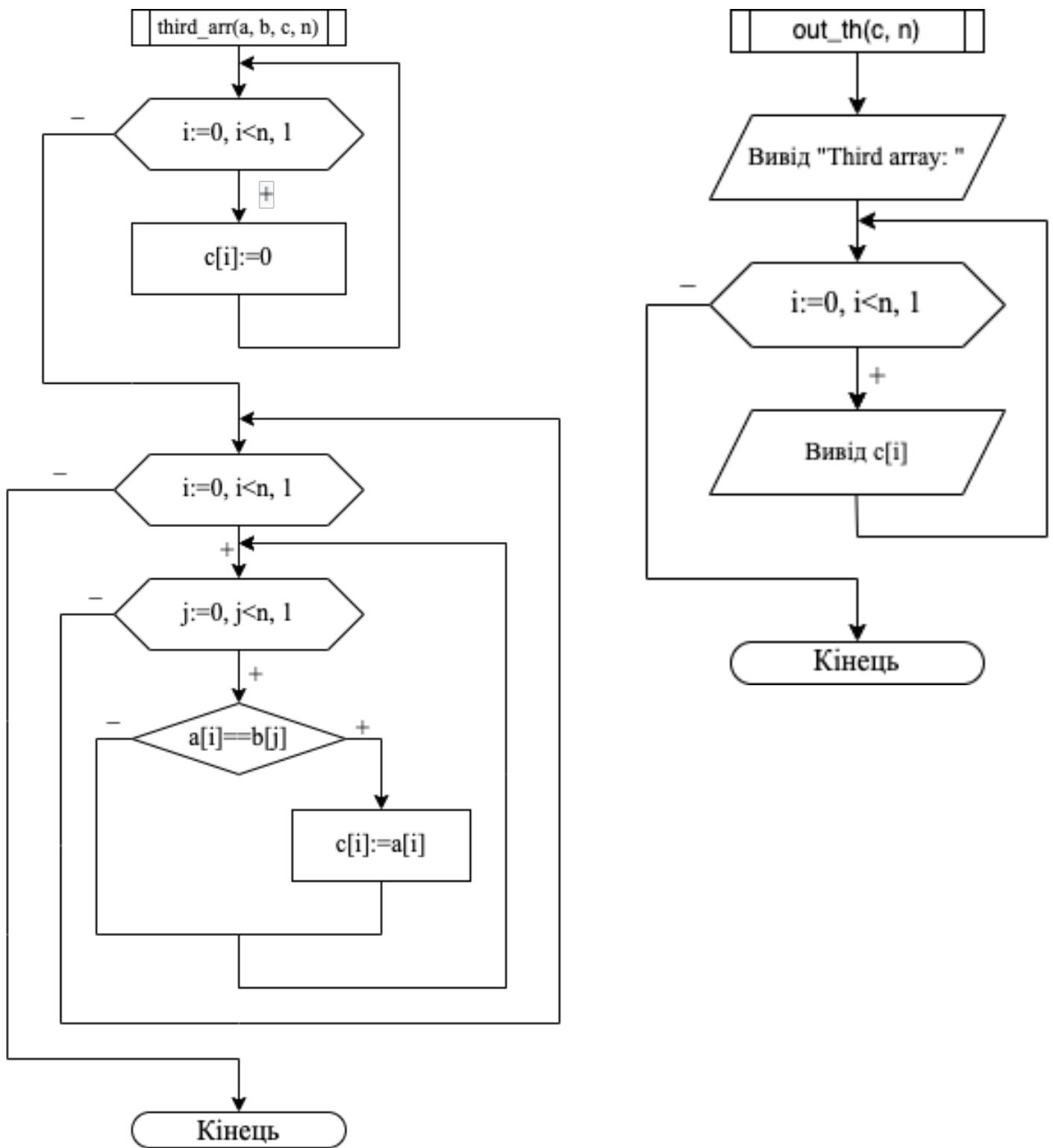
## Блок-схема

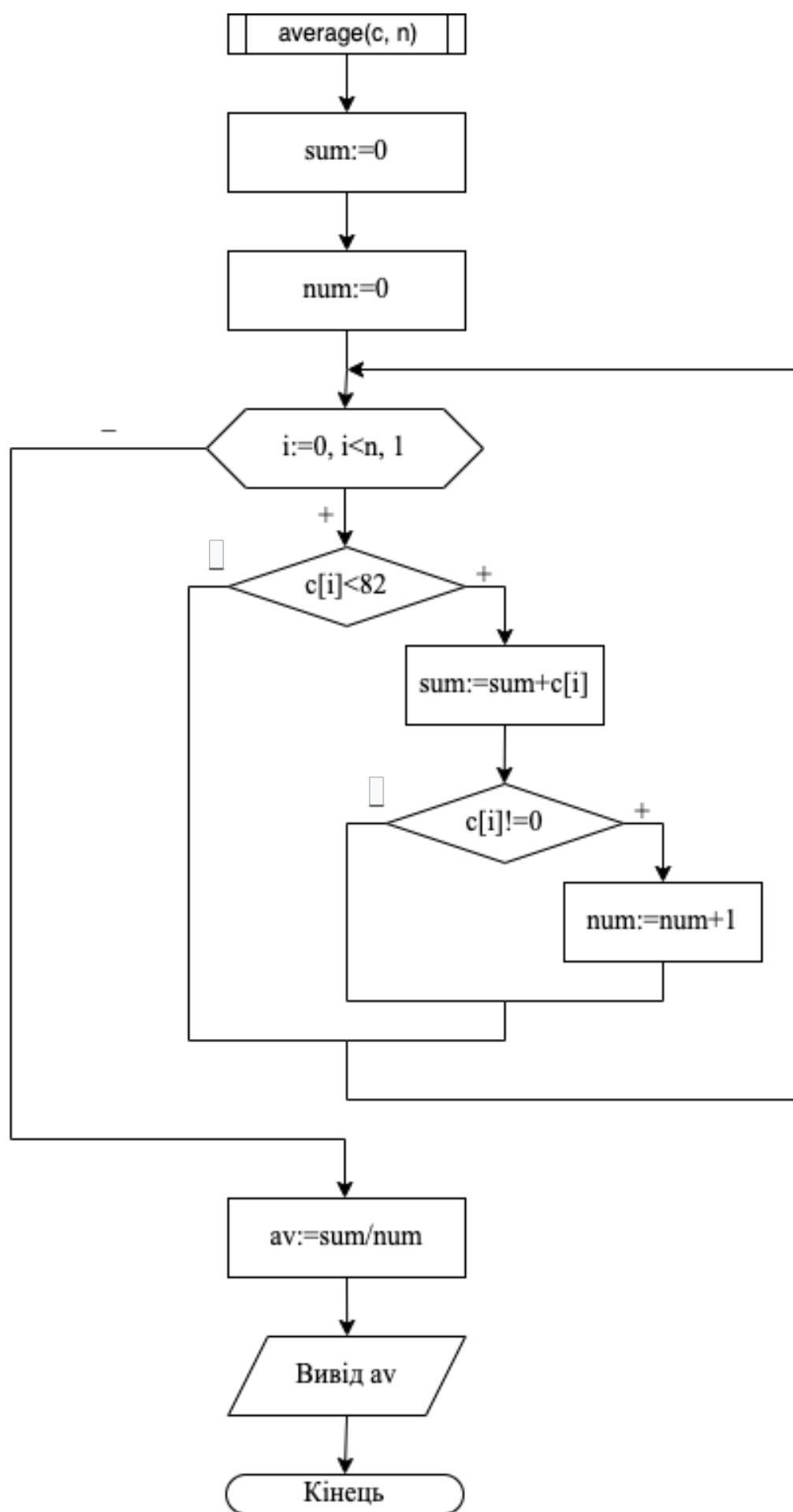
Програма:



## Підпрограми:







## Код программы:

```
1 #include <iostream>
2 #include <cmath>
3 #include <cstdio>
4 #include <cstdlib>
5 using namespace std;
6 void create(char a[], char b[], int n);
7 void output(char a[], char b[], int n);
8 void third_arr(char a[], char b[], char c[], int n);
9 void out_th(char c[], int n);
10 void average(char c[], int n);
11
12 int main() {
13     const int n=10;
14     char first[n], second[n], third[n];
15     create(first, second, n);
16     output(first, second, n);
17     third_arr(first, second, third, n);
18     out_th(third, n);
19     average(third, n);
20 }
21
22 void create(char a[], char b[], int n){
23     for(int i=0; i<n; i++){
24         a[i]=92 - 3 * i;
25         b[i]=71 + 3 * i;
26     }
27 }
28
29 void output(char a[], char b[], int n){
30     cout << "First array: "<<endl;
31     for(int i = 0; i < n; i++){
32         cout << a[i] << " ";
33     }
34     cout<< endl<<endl;
35     cout << "Second array: "<<endl;
36     for(int i = 0; i < n; i++){
37         cout << b[i] << " ";
38     }
39     cout<< endl<<endl;
40 }
41
```

```

42 void third_arr(char a[], char b[], char c[], int n){
43     for(int i = 0; i < n; i++){
44         c[i]=0;
45     }
46     for(int i = 0; i < n; i++){
47         for(int j = 0; j < n; j++){
48             if(a[i]==b[j]) c[i]=a[i];
49         }
50     }
51 }
52
53 void out_th(char c[], int n){
54     cout << "Third array: "<< endl;
55     for(int i = 0; i < n; i++){
56         cout << c[i] << " ";
57     }
58     cout<< endl<< endl;
59 }
60
61 void average(char c[], int n){
62     float av;
63     int sum = 0, num=0;
64     for(int i = 0; i < n; i++){
65         if(c[i]<82){sum+=c[i];
66             if(c[i]!=0){
67                 num++;
68             }
69         }
70     }
71     av=sum/float(num);
72     cout << "Average: " << av << endl;
73 }
74

```

```

First array:
\ Y V S P M J G D A

Second array:
G J M P S V Y \ _ b

Third array:
\ Y V S P M J G

Average: 75.5

```

**Висновок:** отже, в результаті виконання лабораторної роботи ми отримали алгоритм для знаходження середнього арифметичного. Було досліджено алгоритми лінійного пошуку, проаналізовано подане завдання, декомпозовано та виконано. Також були розроблені псевдокод, код програми та блок-схема поставленого алгоритму.