# 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 № з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант 7

Виконав студент ІП-12 Васильєв Єгор Костянтинович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота №7**

**Дослідження лінійних алгоритмів**

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

**Індивідуальне завдання:**



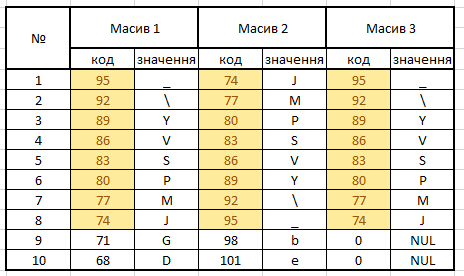
**Постановка задачі**

Результатом розв’язку є три масиви, перші два сформовані за відповідними формулами, а третій містить спільні елементи попередніх двох, та середнє арифметичне елементів третього масиву, коди яких менше 82.

**Побудова математичної моделі**

Перші два масиви будемо створювати за допомогою арифметичних циклів, а третій шляхом присвоювання кожному елементу значення з першого масиву, яке також є трапляється у другому масиві. Оскільки третій масив розрахований на 10 елементів, але фактично буде містити менше, він заповниться пропусками, що відповідно до стандарту ASCII має код 0. Оскільки 0<82, то ці елементи будуть впливати на значення лічильника, який є дільником суми елементів для пошуку середнього арифметичного. Отже необхідно окрім заданої умови для кодів елементів третього масиву також використати умову що не враховує пропуски (тобто елементи з кодом 0).

*Складемо таблицю кодів та значень елементів*



*Складемо таблицю змінних*

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Константа | Цілий | length | Довжина масиву |
| Символьний масив | Індексований | first | Перший масив |
| Символьний масив | Індексований | second | Другий масив |
| Символьний масив | Індексований | third | Третій масив |
| Значення суми | Цілий | Sum | Сума кодів третього масиву |
| Значення к-сті елементів | Цілий | counter | Лічильник для середнього арифметичного |
| Значення лічильника | Цілий | i | Лічильник для роботи з масивами |
| Значення лічильника | Цілий | k | Лічильник для порівняння двох масивів |

**Розв’язання**

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

*Крок 1.* Деталізація основних дій

*Крок 2.* Деталізація знаходження двох масивів

*Крок 3.* Деталізація знаходження третього масиву

*Крок 4.* Деталізація пошуку середнього арифметичного

**Псевдокод**

Крок 1

**початок**

length=10

first[length]

second[length]

third[length]

Sum=0

counter=0

-деталізація знаходження двох масивів

-деталізація знаходження третього масиву

**-**деталізація пошуку середнього арифметичного

**кінець**

Крок 2

**початок**

length=10

first[length]

second[length]

third[length]

Sum=0

counter=0

**повторити** (i=0, length) **раз**

first[i]=95-3\*i

second[i]=74+3\*i

**кінець повторити**

-деталізація знаходження третього масиву

**-**деталізація пошуку середнього арифметичного

**кінець**

Крок 3

**початок**

length=10

first[length]

second[length]

third[length]

Sum=0

counter=0

**повторити** (i=0, length) **раз**

first[i]=95-3\*i

second[i]=74+3\*i

**кінець повторити**

**повторити** (i=0, length) **раз**

**повторити** (k=0, length) **раз**

**якщо** first[i] == second[k]

**то**

third[i] = first[i]

**все якщо**

**кінець повторити**

**кінець повторити**

**-**деталізація пошуку середнього арифметичного

**кінець**

Крок 4

**початок**

length=10

first[length]

second[length]

third[length]

Sum=0

counter=0

**повторити** (i=0, length) **раз**

first[i]=95-3\*i

second[i]=74+3\*i

**кінець повторити**

**повторити** (i=0, length) **раз**

**повторити** (k=0, length) **раз**

**якщо** first[i] == second[k]

**то**

third[i] = first[i]

**все якщо**

**кінець повторити**

**кінець повторити**

**повторити** (i=0, length) **раз**

**якщо** third[i] < 82 та third[i]!=0

**то**

Sum =Sum+third[i]

counter=counter+1

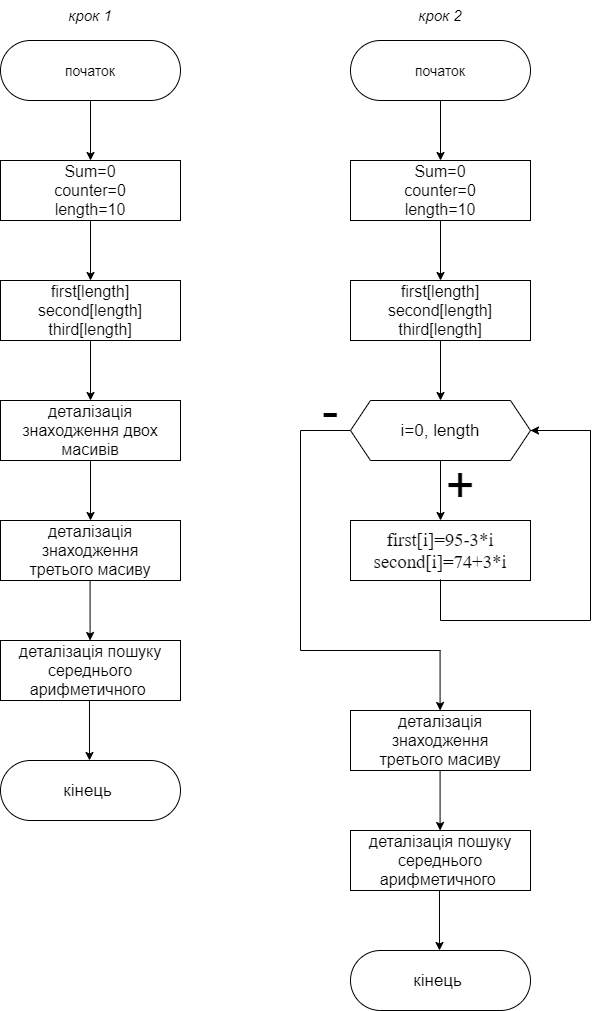
**все якщо**

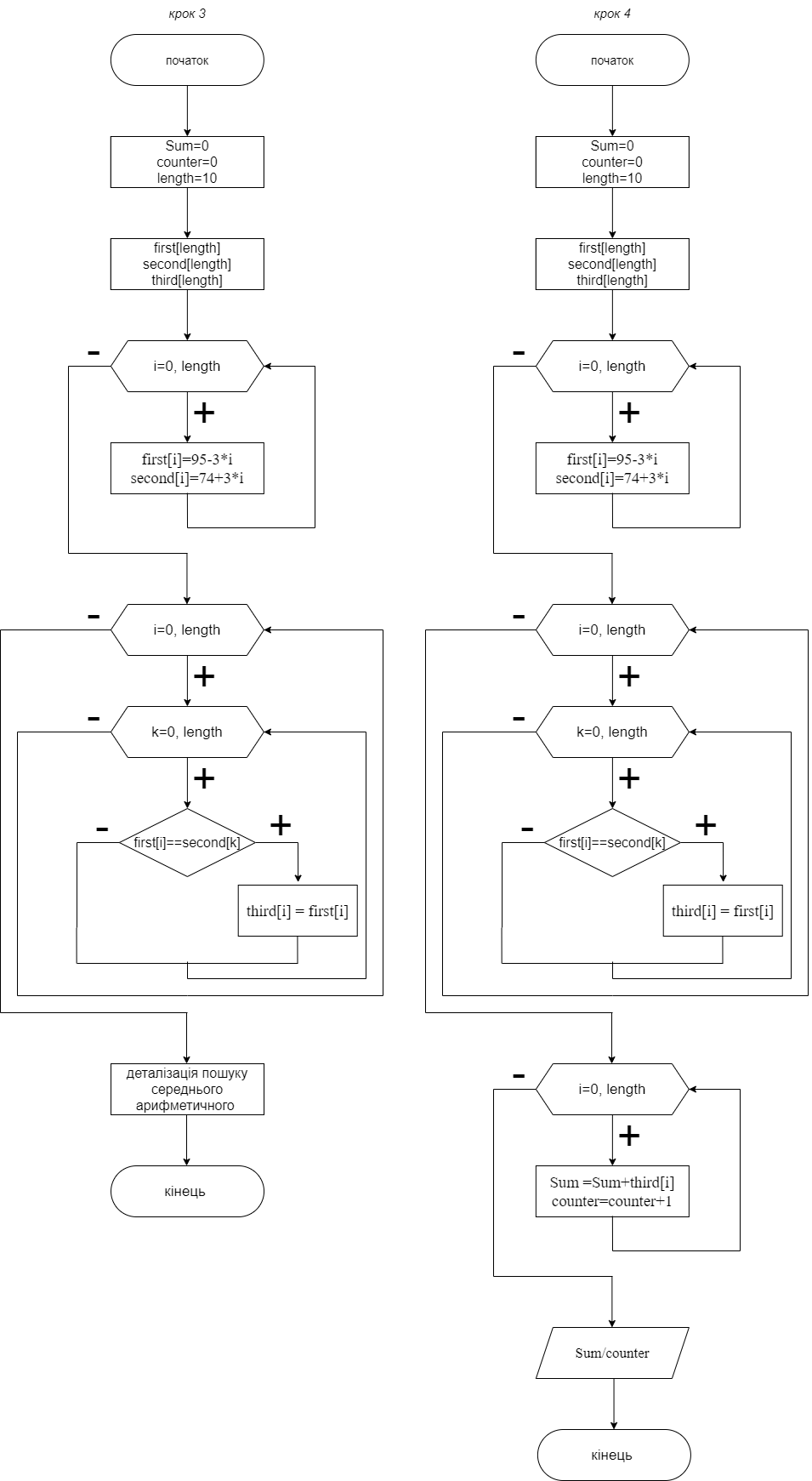
**кінець повторити**

вивести Sum/counter

**кінець**

**Блок-схема**



****

**Перевірка**

*крок 1*

**початок**

**-**деталізація знаходження двох масивів

**-**деталізація знаходження третього масиву

**-**деталізація пошуку середнього арифметичного

**кінець**

*крок 2*

**початок**

**-**first[0]=95 -second[0]=74

**-**first[1]=92 -second[1]=77

**-**first[2]=89 -second[2]=80

**-**first[3]=86 -second[3]=83

**-**first[4]=83 -second[4]=86

**-**first[5]=80 -second[5]=89

**-**first[6]=77 -second[6]=92

**-**first[7]=74 -second[7]=95

**-**first[8]=71 -second[8]=98

**-**first[9]=67 -second[9]=101

**-**деталізація знаходження третього масиву

**-**деталізація пошуку середнього арифметичного

**кінець**

*крок 3*

**початок**

**-**first[0]=95 -second[0]=74

**-**first[1]=92 -second[1]=77

**-**first[2]=89 -second[2]=80

**-**first[3]=86 -second[3]=83

**-**first[4]=83 -second[4]=86

**-**first[5]=80 -second[5]=89

**-**first[6]=77 -second[6]=92

**-**first[7]=74 -second[7]=95

**-**first[8]=71 -second[8]=98

**-**first[9]=67 -second[9]=101

**-**first[0]!= second[0]

. . .

**-**first[0]==second[7]

-third[0] = first[0]

. . .

**-**first[1]!= second[0]

. . .

**-**first[1]==second[6]

-third[1] = first[1]

. . .

**-**first[2]==second[5]

-third[2] = first[2]

. . .

**-**first[3]==second[4]

-third[3] = first[3]

. . .

**-**first[4]==second[3]

-third[4] = first[4]

. . .

**-**first[5]==second[2]

-third[5] = first[5]

. . .

**-**first[6]==second[1]

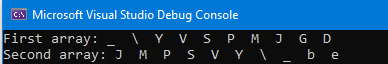
-third[6] = first[6]

. . .

**-**first[7]==second[0]

-third[7] = first[7]

**-**деталізація пошуку середнього арифметичного

**кінець**

*крок 4*

**початок**

**-**first[0]=95 -second[0]=74

**-**first[1]=92 -second[1]=77

**-**first[2]=89 -second[2]=80

**-**first[3]=86 -second[3]=83

**-**first[4]=83 -second[4]=86

**-**first[5]=80 -second[5]=89

**-**first[6]=77 -second[6]=92

**-**first[7]=74 -second[7]=95

**-**first[8]=71 -second[8]=98

**-**first[9]=67 -second[9]=101

**-**first[0]!= second[0]

. . .

**-**first[0]==second[7]

-third[0] = first[0]

. . .

**-**first[1]!= second[0]

. . .

**-**first[1]==second[6]

-third[1] = first[1]

. . .

**-**first[2]==second[5]

-third[2] = first[2]

. . .

**-**first[3]==second[4]

-third[3] = first[3]

. . .

**-**first[4]==second[3]

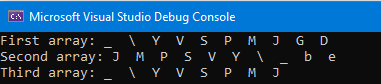
-third[4] = first[4]

. . .

**-**first[5]==second[2]

-third[5] = first[5]

. . .

**-**first[6]==second[1]

-third[6] = first[6]

. . .

**-**first[7]==second[0]

-third[7] = first[7]

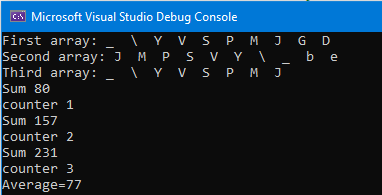
**-**third[0]>82

**-**third[1]>82

. . .

-third[5]<82

-Sum=80

-counter=1

-third[6]<82

-Sum=157

-counter=2

-third[7]<82

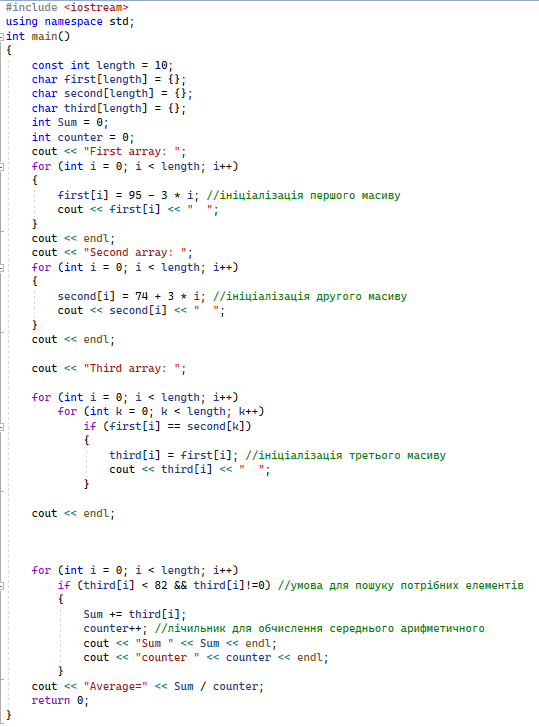
-Sum=231

-counter=3

. . .

“Average”=77(231/3)

**кінець**

**Код програми**

**Висновок**

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