

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 34

Виконав студент ІІІ-15, Чінь Хоанг Вьет
Перевірив Вечерковська Анастасія Сергіївна

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.

Варіант 34

Постановка задачі

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.

№	Вираз для обчислення елемента		Знайти
	1-го масиву	2-го масиву	
34	$44 + 2 * i$	$55 - 2 * i$	Кількість елементів, коди яких діляться на 3

Розв’язання: За допомогою виразів для масивів *a* та *b* знайдемо значення їхніх елементів. Потім створимо третій масив з їх рівних значень, якщо вони є. Потім шукаємо кількість символів, коди яких діляться на 3, якщо вони є відповідно. Це буде результат розв’язку завдання.

Згенеруємо масиви *a* та *b* за допомогою арифметичного циклу у функції *input*. За допомогою цих масивів згенеруємо масив *c* усередині функції *length*. Потім виведемо ці масиви функцією *output*. І нарешті виводимо кількість елементів, які нам потрібно знайти за допомогою функції *amount*.

Математична модель:

Змінна	Тип	Ім'я	Призначення
Масив А	Символьний	a	Проміжні дані, результат
Масив В	Символьний	b	Проміжні дані, результат
Масив С	Символьний	c	Проміжні дані, результат дані
Штучний масив	Символьний	m	Проміжні дані
Лічильник 1	Цілочисельний та	i	Параметр

	натуральний		циклу
Лічильник 2	Цілочисельний та натуральний	j	Параметр циклу
Кількість елементів	Цілочисельний	Amount	Проміжні дані
Розмір масиву	Цілочисельний	len	Проміжні дані
Створення масивів a та b	Процедура	input	Початкові дані
Створення масиву c	Процедура	length	Початкові дані
Вивід масивів	Процедура	output	Початкові дані
Пошук кількості чисел з масиву c, що ділиться на 3	Процедура	amount	Початкові дані

1. Визначаємо основні дії.
2. Ініціалізація початкових даних.
3. Створення підпрограми, яка обчислює значення елементів масивів A і B.
4. Створення підпрограми, яка знаходить значення елементів масиву C, якщо вони є.
5. Створення підпрограми, яка виводить значення масивів A, B і C.
6. Створення підпрограми, яка підраховує кількість елементів у масиві C, які діляться на 3, якщо вони є.

Псевдокод:

Головна програма:

Початок

input(a, b)

length(a, b, c)

output(a)

output(b)

output(c)

amount(c)

Кінець

Підпрограми:

Крок 3

Початок **input**

Для **i** від **0** до **10** повторити

$a[i] = 44 + 2*i;$

$b[i] = 55 - 2*i;$

Все повторити

Кінець **input**

Крок 4

Початок **length**

Ввід $len = 0;$

Для **j** від **0** до **10** повторити

Для **i** від **0** до **10** повторити

Якщо $a[j] == b[i];$

$c[len] = a[j];$

Все якщо

Все повторити

$len = len + 1;$

Все повторити

Кінець **length**

Крок 5

Початок **output**

Для **i** від **0** до **10** повторити

Вивід $m[i]$

Все повторити

Кінець **output**

Крок 6

Початок **amount**

$Amount = 0;$

Для **i** від **0** до **10** повторити

Якщо $c[i] \% 3 == 0$

$Amount = Amount + 1;$

Все якщо

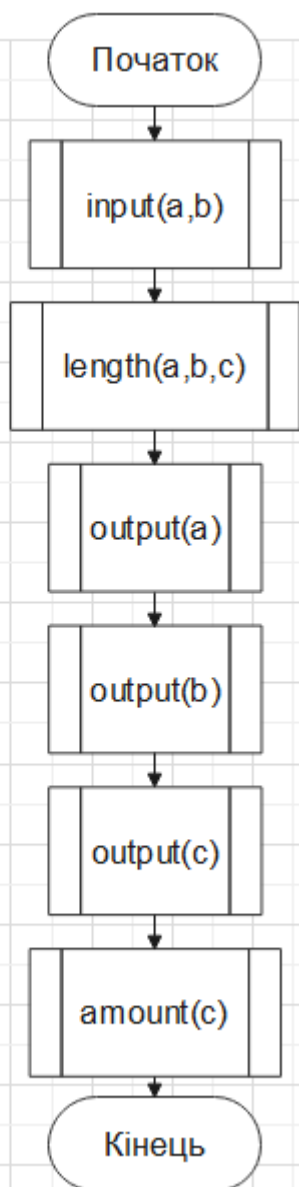
Все повторити

Вивід **Amount**

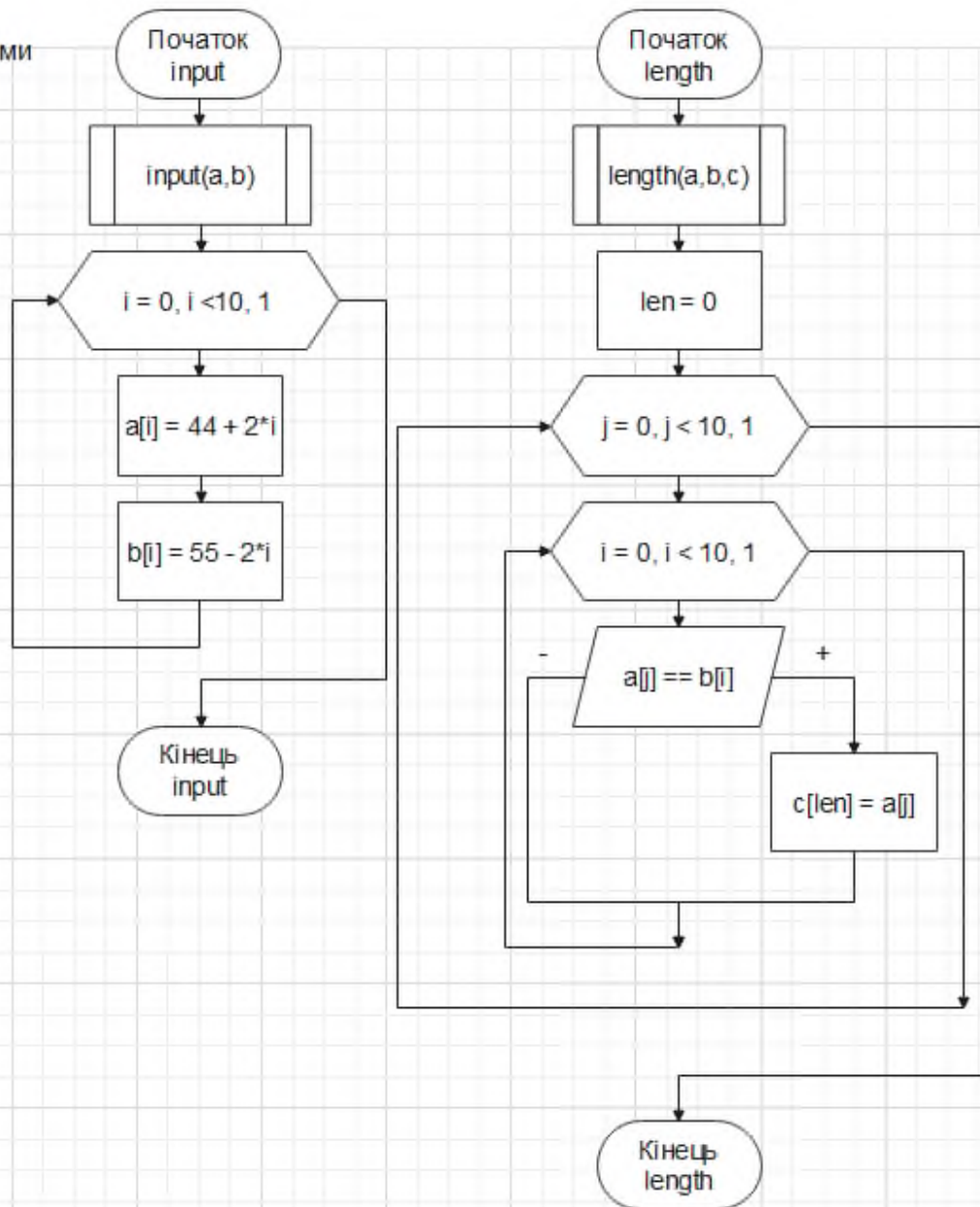
Кінець **amount**

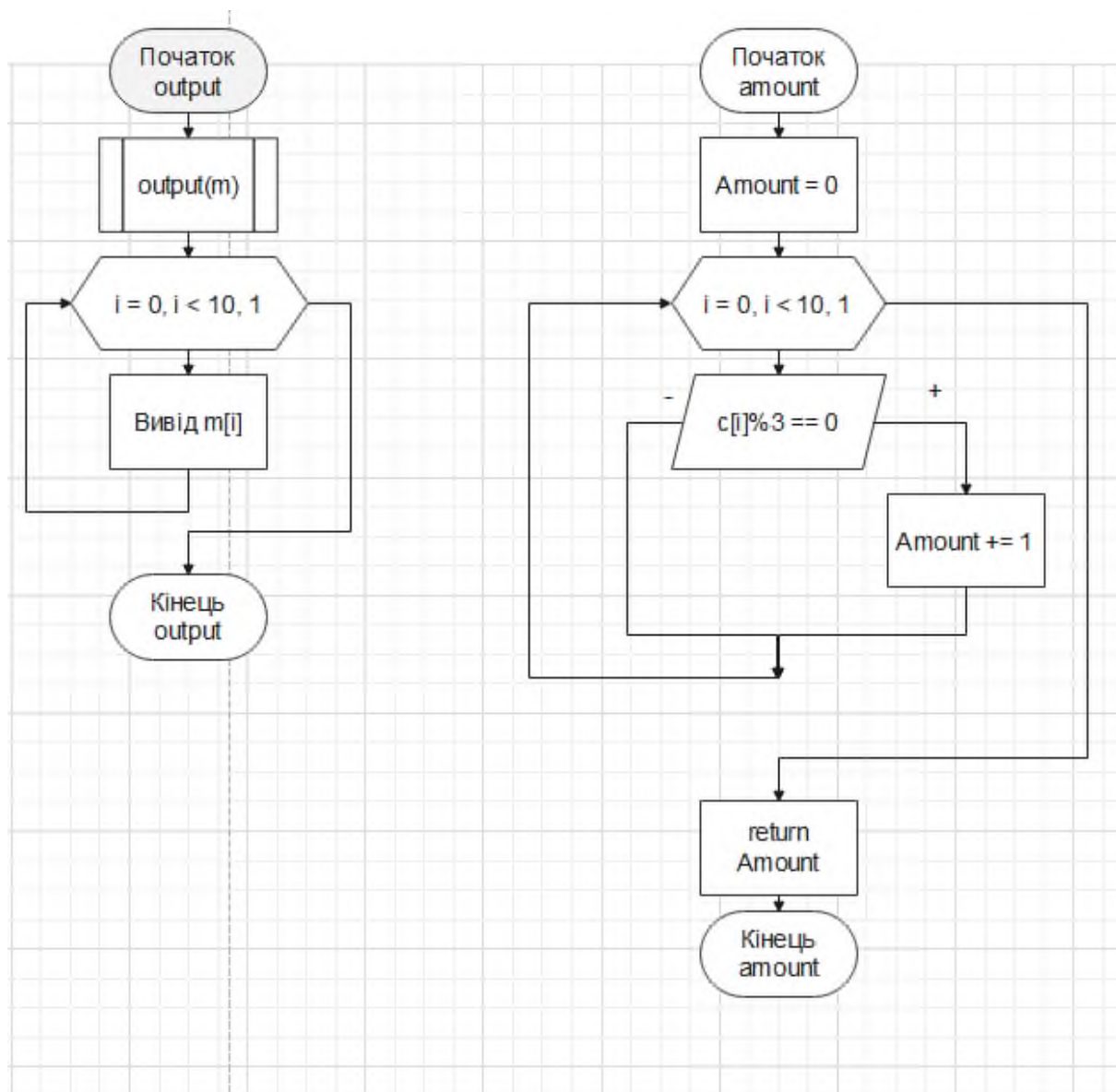
Блок-схема:

Головна
програма



Підпрограми






Код програми:


```

1  #include <iostream>
2  #include <iomanip>
3
4  using namespace std;
5  void input(char[10], char[10]); // генерація масивів а та б
6  void output(char[10]);         // вивід масиву
7  void length(char[10],char[10],char[10]); // генерація масиву с
8  int amount(char[10]);          // Кількість елементів, яку потрібно знайти
9
10 int main() {
11     char a[10], b[10], c[10]; // масиви з 10 символічних значень
12     input(a, b);
13     length(a, b, c);
14     cout << "A:"; output(a);
15     cout << "B:"; output(b);
16     cout << "C:"; output(c);
17     amount(c);
18 }
19
20 void input(char A[10],char B[10]) { // Функція генерації масивів а та б
21     for (int i = 0; i < 10; i++) {
22         A[i] = 44 + 2 * i;
23         B[i] = 55 - 2 * i;
24     }
25 }
26 void output(char mas[10]) { // Функція виводу масиву
27     for (int i = 0; i < 10; i++)
28         cout << setw(3) << mas[i];
29
30     cout << "\n";
31 }
32 void length(char FIRSTmas[10], char SECONDmas[10], char THIRDMas[10]) { // Функція генерації масиву с
33     int len = 0;
34     for (int j = 0; j < 10; j++) {
35         for (int i = 0; i < 10; i++) {
36             if (FIRSTmas[j] == SECONDmas[i])
37                 THIRDMas[len] = FIRSTmas[j];
38             len += 1;
39         }
40     }
41 }
42 int amount(char last[10]) { // Функція визначення Кількості елементів, яку потрібно знайти
43     int amount = 0;
44     for (int i = 0; i < 10; i++) {
45         if (last[i] % 3 == 0)
46             amount += 1;
47     }
48     cout << "The amount is " << amount;
49     return amount;
50 }

```

Тестування програми:

 Microsoft Visual Studio Debug Console

```

A:  ,  .  0  2  4  6  8  :  <  >
B:  7  5  3  1  /  -  +  )  '  %
C:  || || || || || || || || ||
The amount is 0

```

Висновок: На цій лабораторній роботі, ми дослідили методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набули практичних навичок їх використання під час складання програмних специфікацій. Ми написали алгоритм, у якому ми описували логічні дії та створили програму, яка виводить нам символічні значення масивів a , b і c та кількість елементів, які потрібно знайти за умовою.