

*Додаток 1*

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 ІП-13 Шиманська Ганна Артурівна  
(шифр, прізвище, ім'я, по батькові)

Перевірила 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 набуті практичних навичок їх використання під час складання програмних специфікацій.

### Варіант 34

№	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

- **Постановка задачі**

Необхідно 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 індивідуальним варіантом.
3. Ініціювання третьої змінної рівними значеннями з попередніх масивів.
4. Обрахунку кількості елементів, коди яких діляться на 3.

### Побудова математичної моделі

Складемо таблицю змінних

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
Перший масив	char	array1	Проміжні дані
Другий масив	char	array2	Проміжні дані
Третій масив	char	array3	Проміжні дані

Кількість знайдених елементів	int	Result	Вихідні дані
Лічильник i	int	I	Проміжні дані
Лічильник j	int	J	Проміжні дані
Масив для заповнення в підпрограмі	char	insertedArray	Проміжні дані
Масив спільних елементів попередніх масивів в підпрограмі	char	commonArray	Проміжні дані
Лічильник дільників	int	Counter	Вихідні дані

Використаємо також властивість `length` об'єкту `Array`, яка містить кількість елементів у масиві.

Отже, ми будемо заповнювати перший та другий масиви використовуючи підпрограму **InsertElements**, яка за допомогою арифметичного циклу заповнить масив **insertedArray** підпрограми 10 елементами. Присвоїмо масиву **array1** значення **insertedArray**, який був заповнений підпрограмою через відповідні параметри та масиву **array2** аналогічно, змінивши параметри. Також виведемо кожен масив.

Далі за допомогою підпрограми **CommonElements** порівняємо значення першого і другого масивів, записавши рівні значення у третій масив, використовуючи два арифметичних `for`-цикли та один `if`-цикл. Таким чином, підпрограма проходитиметься по першому масиву, фіксуючи один елемент і порівнюючи його з кожним елементом другого

масиву. Знайшовши спільне значення, підпрограма занесе його до третього масиву **commonArray**. Також присвоїмо масиву **array3** значення **commonArray**.

За допомогою підпрограми **GetDividedAmount** ми отримаємо кількість елементів третього масиву, коди яких діляться націло на 3. Реалізація підпрограми здійснюється використовуючи **for**-цикл для проходження по елементам масиву **array3** та **if**-цикл для перевірки кратності певного елемента трьома. Якщо такий елемент знайдено, **counter** збільшується на 1. Повертаємо значення змінної **counter**.

Присвоїмо змінній **result** результат виклику підпрограми **GetDividedAmount** та виведемо його.

### Розв'язання

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

*Крок 1.* Визначимо основні дії.

*Крок 2.* Деталізуємо дію оголошення змінних.

*Крок 3.* Деталізуємо підпрограму заповнення першого масиву елементами та її виклик.

*Крок 4.* Деталізуємо підпрограму заповнення другого масиву елементами та її виклик.

*Крок 5.* Деталізуємо підпрограму знаходження спільних елементів масиву та її виклик.

*Крок 6.* Деталізуємо підпрограму пошуку кількості елементів третього масиву, кратних трьома та її виклик.

- **Псевдокод алгоритму**

*Крок 1.*

**початок**

Оголошення змінних

Виклик підпрограми InsertElements(44)

Виклик підпрограми InsertElements(55, -1)

Виклик підпрограми CommonElements(array1, array2)

Виклик підпрограми GetDividedAmount(array3, 3) і присвоєння результату змінній result

Виведення result

**кінець**

**підпрограма** InsertElements(int a, int b = 1)

Заповнення масиву

**все підпрограма**

**підпрограма** CommonElements(char[] array1, char[] array2)

Знаходження спільних елементів масиву

**все підпрограма**

**підпрограма** GetDividedAmount(char[] array, int divider)

пошук кількості елементів третього масиву, кратних трьом

**все підпрограма**

*Крок 2.*

**початок**

char[] array1 = new char(10)

```
char[] array2 = new char(10)
```

```
char[] array3 = new char(10)
```

Виклик підпрограми InsertElements(44)

Виклик підпрограми InsertElements(55, -1)

Виклик підпрограми CommonElements(array1, array2)

Виклик підпрограми GetDividedAmount(array3, 3) і присвоєння результату змінній result

Виведення result

**кінець**

*Крок 3.*

**початок**

```
char[] array1 = new char(10)
```

```
char[] array2 = new char(10)
```

```
char[] array3 = new char(10)
```

```
array1 = InsertElements(44)
```

Виклик підпрограми InsertElements(55, -1)

Виклик підпрограми CommonElements(array1, array2)

Виклик підпрограми GetDividedAmount(array3, 3) і присвоєння результату змінній result

Виведення result

**кінець**

*Крок 4.*

**початок**

```
char[] array1 = new char(10)
```

```
char[] array2 = new char(10)
```

```
char[] array3 = new char(10)
```

```
array1 = InsertElements(44)
```

```
array2 = InsertElements(55, -1)
```

```
Виклик підпрограми CommonElements(array1, array2)
```

```
Виклик підпрограми GetDividedAmount(array3, 3) і присвоєння  
результату змінній result
```

```
Виведення result
```

**кінець**

*Крок 5.*

**початок**

```
char[] array1 = new char(10)
```

```
char[] array2 = new char(10)
```

```
char[] array3 = new char(10)
```

```
array1 = InsertElements(44)
```

```
array2 = InsertElements(55, -1)
```

```
array3 = CommonElements(array1, array2)
```

```
Виклик підпрограми GetDividedAmount(array3, 3) і присвоєння  
результату змінній result
```

```
Виведення result
```

**кінець**

*Крок 6.*

**початок**

```
char[] array1 = new char(10)
```

```
char[] array2 = new char(10)
```

```
char[] array3 = new char(10)
```

```
array1 = InsertElements(44)
array2 = InsertElements(55, -1)
array3 = CommonElements(array1, array2)
result = GetDividedAmount(array3, 3)
Виведення результату
```

**кінець**

**підпрограма** InsertElements(int a, int b = 1)

```
char[] insertedArray1 = new char(10)
для і від 0 до insertedArray1.length повторити
    insertedArray1[i] = char(a + b * i * 2)
все повторити
return insertedArray1
```

**все підпрограма**

**підпрограма** CommonElements(char[] array1, char[] array2)

```
char[] commonArray = new char[10]
для і від 0 до array1.length повторити
    для j від 0 до array2.length повторити
        якщо array1[i] == array2[j]
            то
                commonArray[i] = array1[i]
        все якщо
    все повторити
все повторити
return commonArray
```

**все підпрограма**



**підпрограма** GetDividedAmount(char[] array, int divider)

    counter = 0

**для** i від 0 до array.length **повторити**

**якщо** array[i] != 0 && array[i] % divider == 0

**то**

                counter = counter + 1

**все якщо**

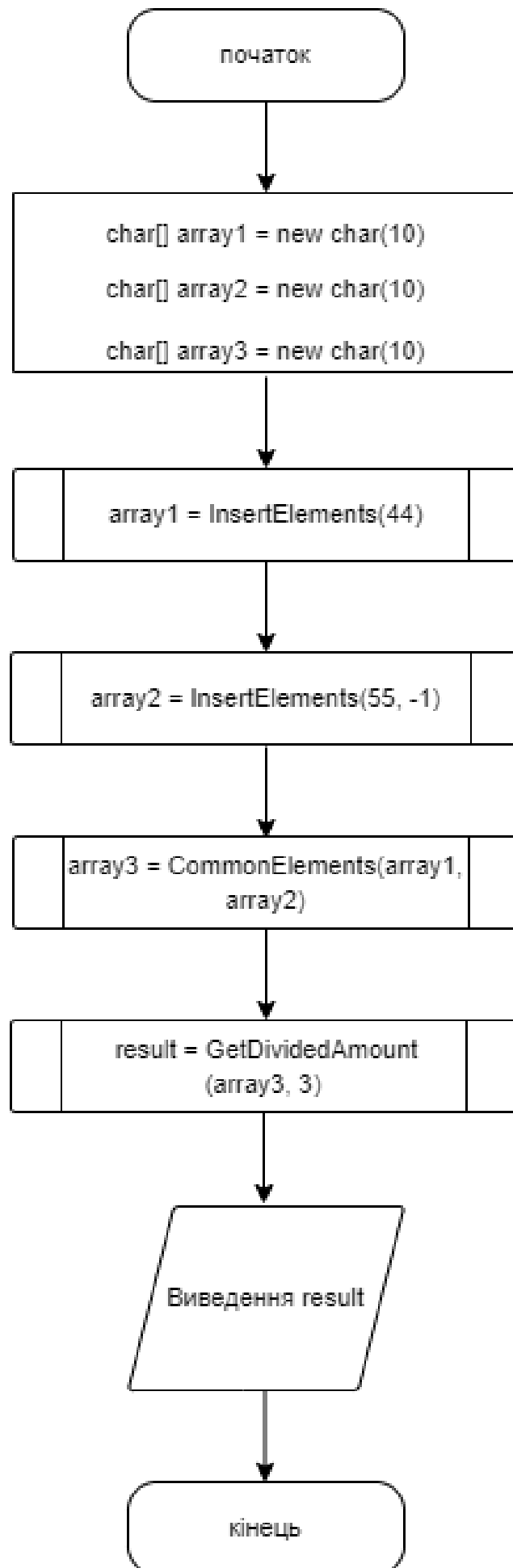
**все повторити**

**return** counter

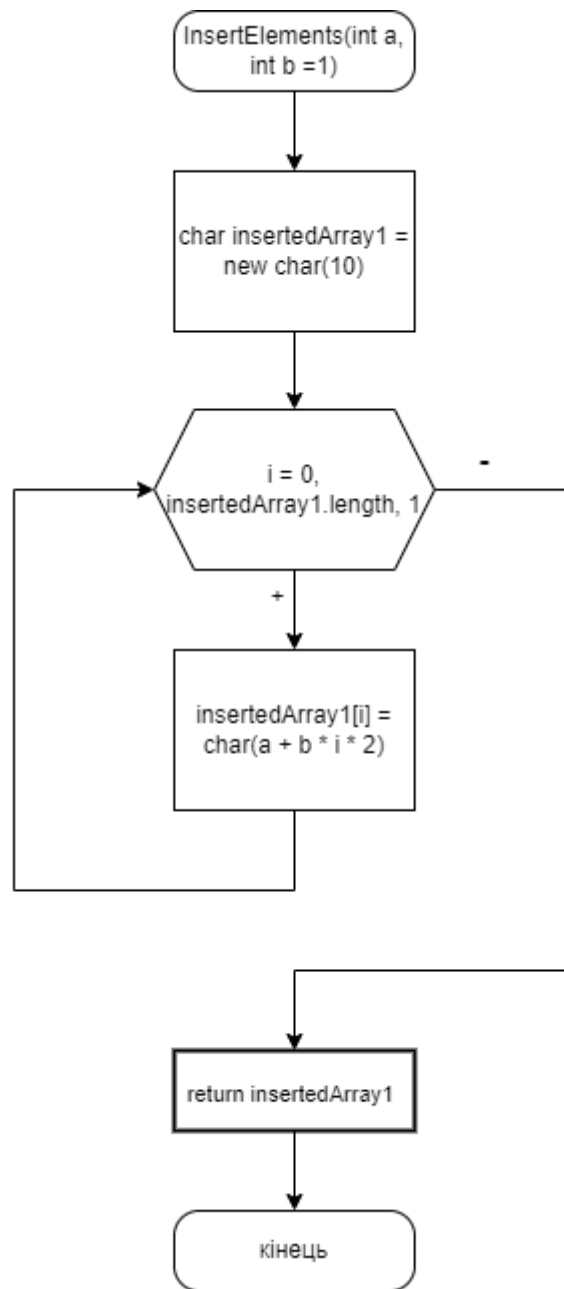
**все підпрограма**

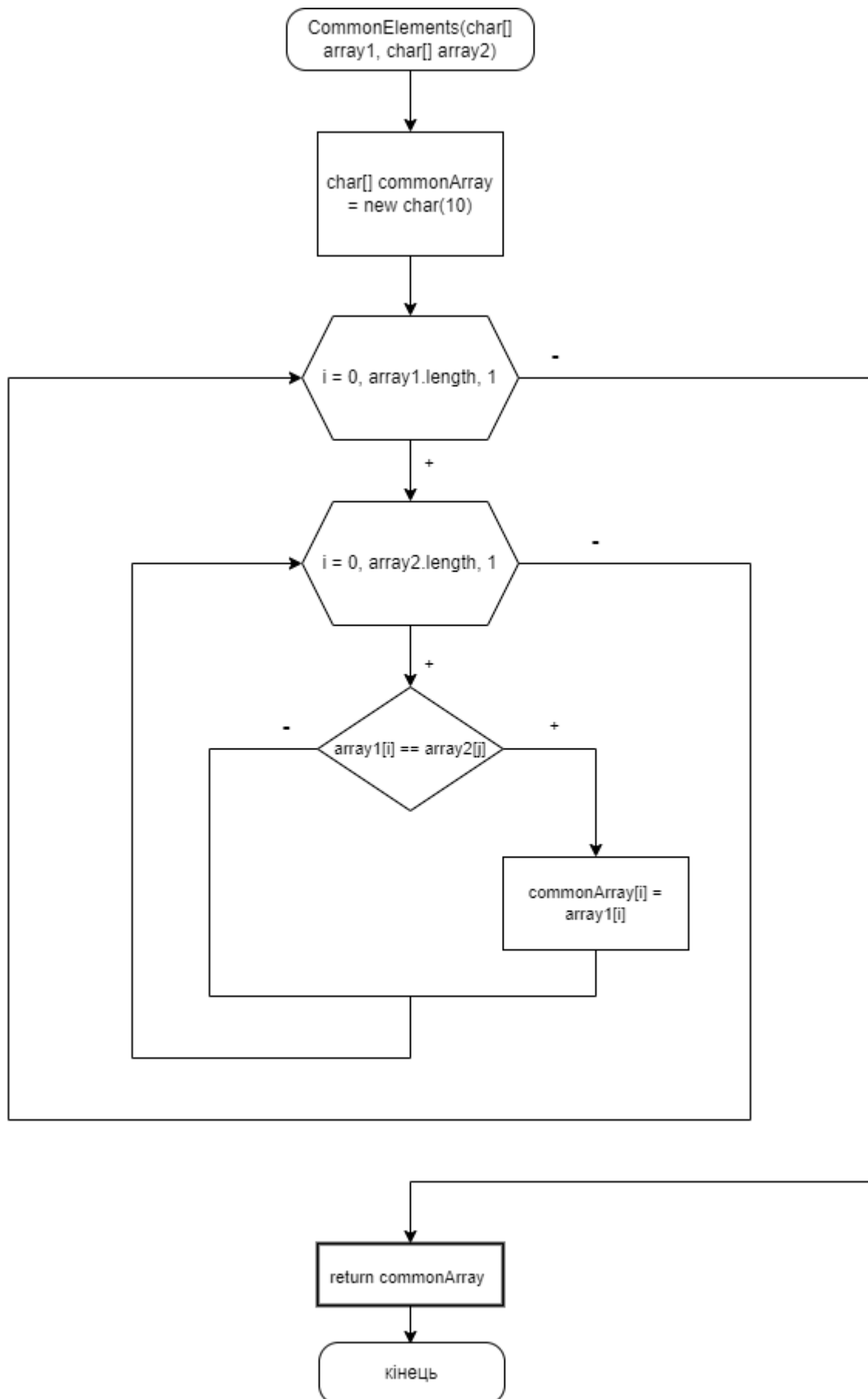
- **Блок-схема**

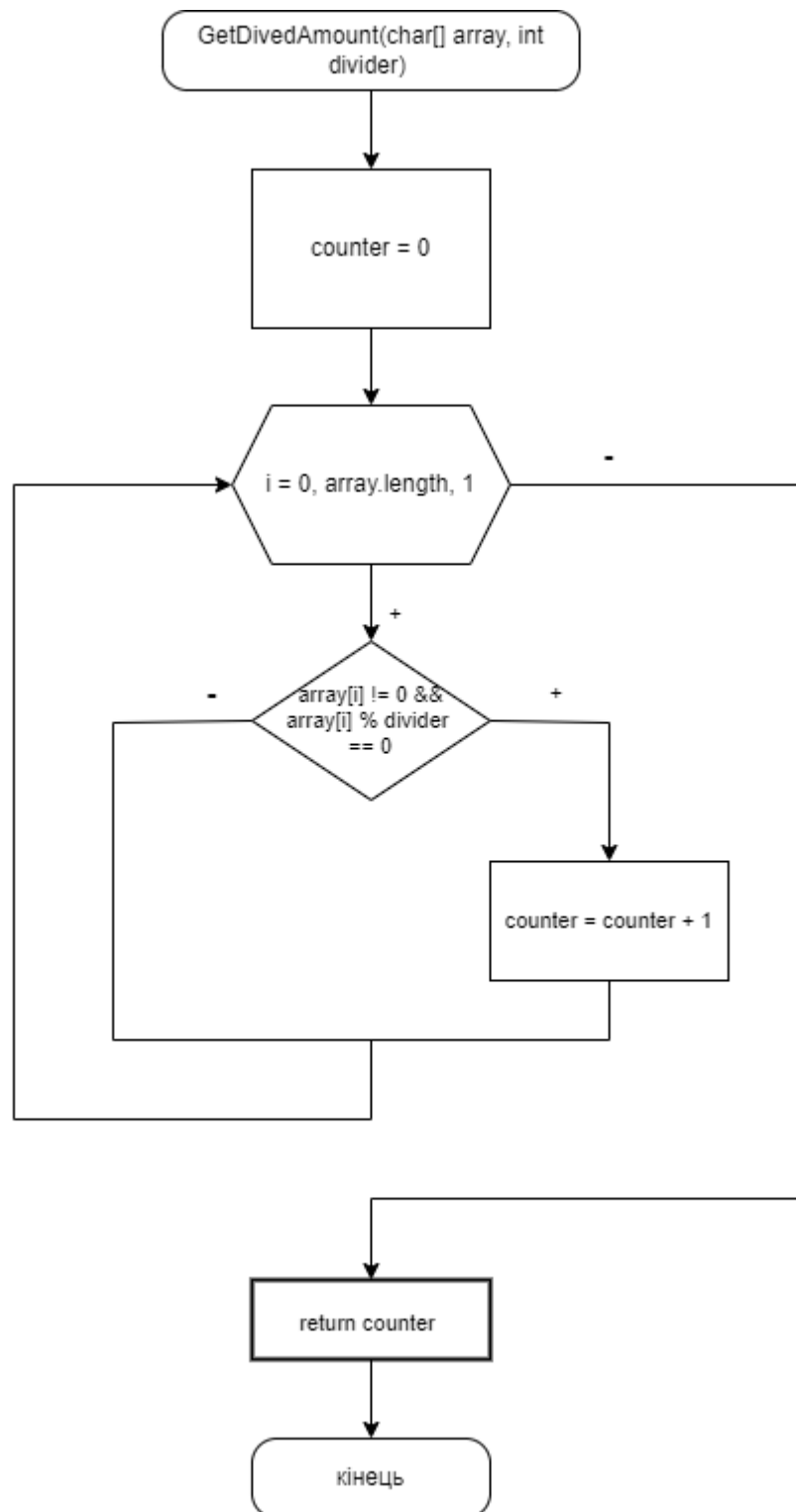
*Основна програма*



*Підпрограми*








- Код програми

### Основна програма

```
1 using System;
2
3 namespace AsdLab7
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             char[] array1 = new char[10];
10            char[] array2 = new char[10];
11            char[] array3 = new char[10];
12
13            Console.WriteLine("Перший масив: ");
14            array1 = InsertElements(44);
15            Console.WriteLine("Другий масив: ");
16            array2 = InsertElements(55, -1);
17            array3 = CommonElements(array1, array2);
18            int result = GetDividedAmount(array3, 3);
19
20
21            Console.WriteLine("Кількість елементів третього масиву, коди яких діляться на 3: " + result);
22        }
23    }
```

### Підпрограми

```
24 static char[] InsertElements(int a, int b = 1)
25 {
26     char[] insertedArray1 = new char[10];
27     for(int i = 0; i < insertedArray1.Length; i++)
28     {
29         insertedArray1[i] = (char)(a + b * 2 * i);
30         Console.WriteLine(insertedArray1[i] + " ");
31     }
32     Console.WriteLine("");
33     return insertedArray1;
34 }
35
36 static char[] CommonElements(char[] array1, char[] array2)
37 {
38     char[] commonArray = new char[10];
39     for (int i = 0; i < array1.Length; i++)
40     {
41         for (int j = 0; j < array2.Length; j++)
42         {
43             if(array1[i] == array2[j])
44             {
45                 commonArray[i] = array1[i];
46             }
47         }
48     }
49     return commonArray;
50 }
51
52 static int GetDividedAmount(char[] array, int divisor)
53 {
54     int counter = 0;
55     for (int i = 0; i < array.Length; i++)
56     {
57         if (array[i] != 0 && array[i] % divisor == 0)
58         {
59             counter++;
60         }
61     }
62     return counter;
63 }
64
65 }
```

 Консоль отладки Microsoft Visual Studio

Перший масив: , . 0 2 4 6 8 : < >

Другий масив: 7 5 3 1 / - + ) ' %

Кількість елементів третього масиву, коди яких діляться на 3: 0

- **Висновки:**

Виконуючи лабораторну роботу, я дослідила особливості методів послідовного пошуку у впорядкованих та неупорядкованих послідовностях та набула практичних навичок їх використання під час складання програмних специфікацій.