
Алгоритми та структури даних. Основи алгоритмізації

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 лінійних алгоритмів»

Варіант 9

Виконав студент Григоренко Родіон Ярославович
(шифр, прізвище, ім'я, по батькові)

Перевірив Вечерковська А.С.
(прізвище, ім'я, по батькові)

Київ 2021

Алгоритми та структури даних. Основи алгоритмізації

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.

Індивідуальне завдання

9	$i * i + 76$	$85 - i$	Кількість елементів, коди яких діляться на 5
---	--------------	----------	--

Варіант 9 Постановка задачі

Результатом є виведення кількості елементів третього масиву, коди яких діляться на 5.

Побудова математичної моделі

Складемо таблицю імен змінних та функцій

Змінна	Тип	Ім'я	Призначення
Перший масив	Символьний	array1	Проміжні дані
Другий масив	Символьний	array2	Проміжні дані
Третій масив	Символьний	array3	Проміжні дані
Функція для ініціювання першого масиву	void	arr1	Ініціювання першого масиву
Функція для ініціювання другого масиву	void	arr2	Ініціювання другого масиву
Функція для ініціювання третього масиву	void	arr2	Ініціювання третього масиву
Функція для знаходження результату	Цілочисельний	Res	Знаходження результату
Результат	Цілочисельний	res	Результат

Алгоритми та структури даних. Основи алгоритмізації

arr1 - функція для ініціювання першого масиву, приймає як параметр перший масив.

arr2 - функція для ініціювання другого масиву, приймає як параметр другий масив.

arr3 - функція для ініціювання третього масиву, приймає як параметр перший, другий і третій масиви.

array1, array2, array3 – 10-елементні масиви.

Res - функція, що обчислює змінну res, приймає як параметр третій масив.

res - змінна, що містить кількість елементів третього масиву, коди яких діляться на 5.

Розв'язання

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо функцію arr1.

Крок 3. Деталізуємо функцію arr2.

Крок 4. Деталізуємо функцію arr3.

Крок 5. Деталізуємо функцію Res.

Псевдокод

Крок 1

функція arr1(array1)

Ініціювання масиву array1

все функція

функція arr2(array2)

Ініціювання масиву array2

все функція

функція arr3(array1, array2, array3)

Ініціювання масиву array3

все функція

функція Res(array3)

Знаходження кількості елементів третього масиву, коди яких діляться на 5

все функція

початок

int res

char array1[10]

char array2[10]

char array3[10]

arr1(array1)

arr2(array2)

arr3(array1, array2, array3)

res = Res(array3)

Алгоритми та структури даних. Основи алгоритмізації

виведення res
кінець

Крок 2

функція arr1(array1)
 для i від 0 до 10, збільшувати на 1
 array[i] = i*i + 76
все функція

функція arr2(array2)
 Ініціювання масиву array2
все функція

функція arr3(array1,array2,array3)
 Ініціювання масиву array3
все функція

функція Res(array3)
 Знаходження кількості елементів третього масиву, коди яких діляться на 5
все функція

початок
 int res
 char array1[10]
 char array2[10]
 char array3[10]
 arr1(array1)
 arr2(array2)
 arr3(array1,array2,array3)
 res = Res(array3)
 виведення res
кінець

Крок 3

функція arr1(array1)
 для i від 0 до 10, збільшувати на 1
 array1[i] = i*i + 76
все функція

функція arr2(array2)
 для i від 0 до 10, збільшувати на 1
 array2[i] = 85 - i
все функція

Алгоритми та структури даних. Основи алгоритмізації

функція arr3(array1,array2,array3)
Ініціювання масиву array3

все функція

функція Res(array3)
Знаходження кількості елементів третього масиву, коди яких діляться на 5
все функція

початок

```
int res
char array1[10]
char array2[10]
char array3[10]
arr1(array1)
arr2(array2)
arr3(array1,array2,array3)
res = Res(array3)
виведення res
```

кінець

Крок 4

функція arr1(array1)
для і від 0 до 10, збільшувати на 1
array1[i] = i*i + 76
все функція

функція arr2(array2)
для і від 0 до 10, збільшувати на 1
array2[i] = 85 - i
все функція

функція arr3(array1,array2,array3)
для і від 0 до 10, збільшувати на 1
для j від 0 до 10, збільшувати на 1
якщо array1[i] == array2[j]
то
array3[i] = array1[i]
все якщо
все функція

функція Res(array3)
Знаходження кількості елементів третього масиву, коди яких діляться на 5
все функція

Алгоритми та структури даних. Основи алгоритмізації

початок

```
int res
char array1[10]
char array2[10]
char array3[10]
arr1(array1)
arr2(array2)
arr3(array1,array2,array3)
res = Res(array3)
виведення res
```

кінець

Крок 5

функція arr1(array1)

```
    для i від 0 до 10, збільшувати на 1
        array1[i] = i*i + 76
```

все функція

функція arr2(array2)

```
    для i від 0 до 10, збільшувати на 1
        array2[i] = 85 - i
```

все функція

функція arr3(array1,array2,array3)

```
    для i від 0 до 10, збільшувати на 1
        array3[i] = 1
        для j від 0 до 10, збільшувати на 1
            якщо array1[i] == array2[j]
                то
                    array3[i] = array1[i]
        все якщо
```

все функція

функція Res(array3)

```
int res = 0
    для i від 0 до 10, збільшувати на 1
        якщо array3[i] % 5 == 0
            то
                res += 1
    все якщо
```

все функція

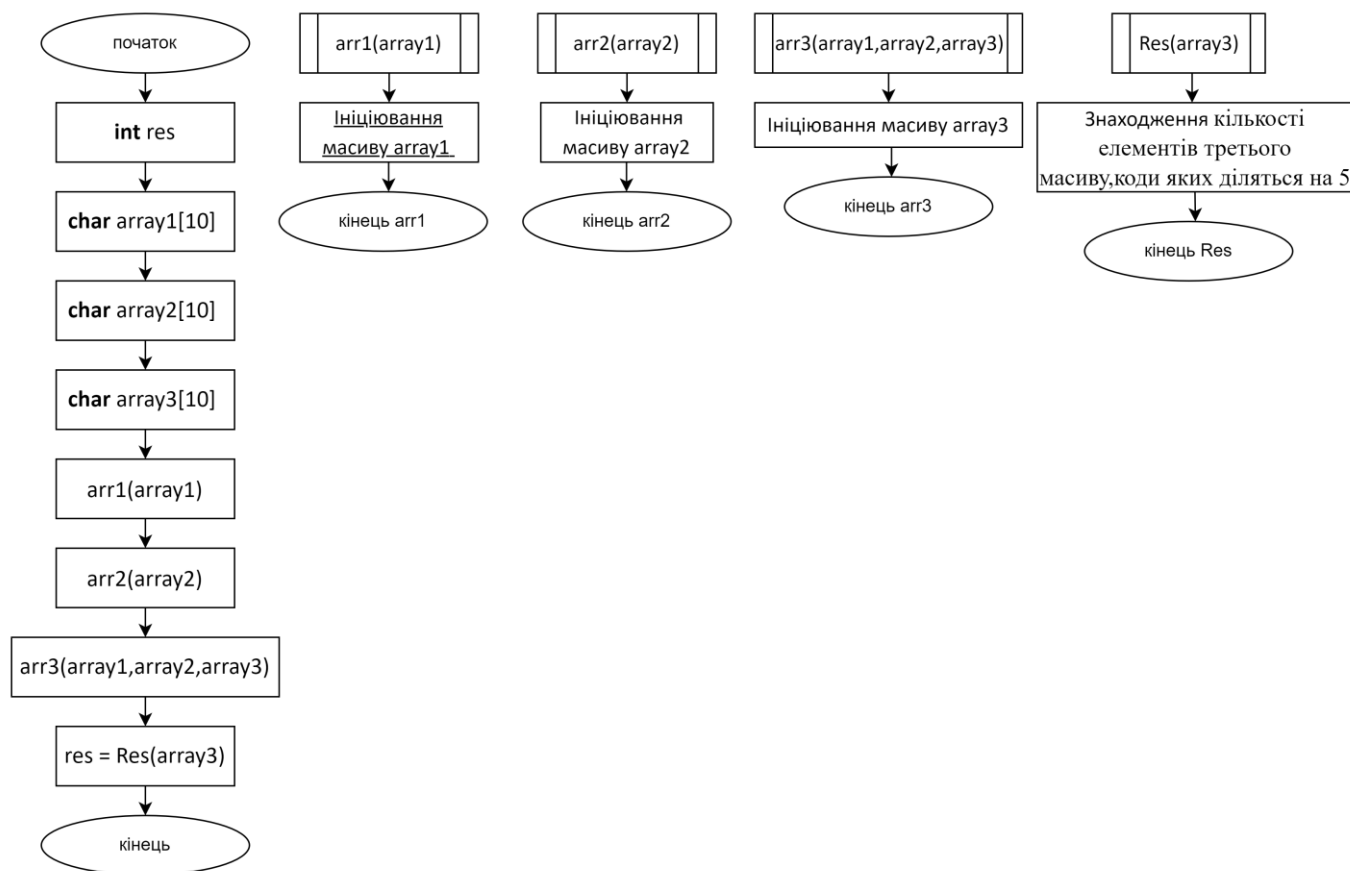
початок

Алгоритми та структури даних. Основи алгоритмізації

```
int res
char array1[10]
char array2[10]
char array3[10]
arr1(array1)
arr2(array2)
arr3(array1,array2,array3)
res = Res(array3)
виведення res
кінець
```

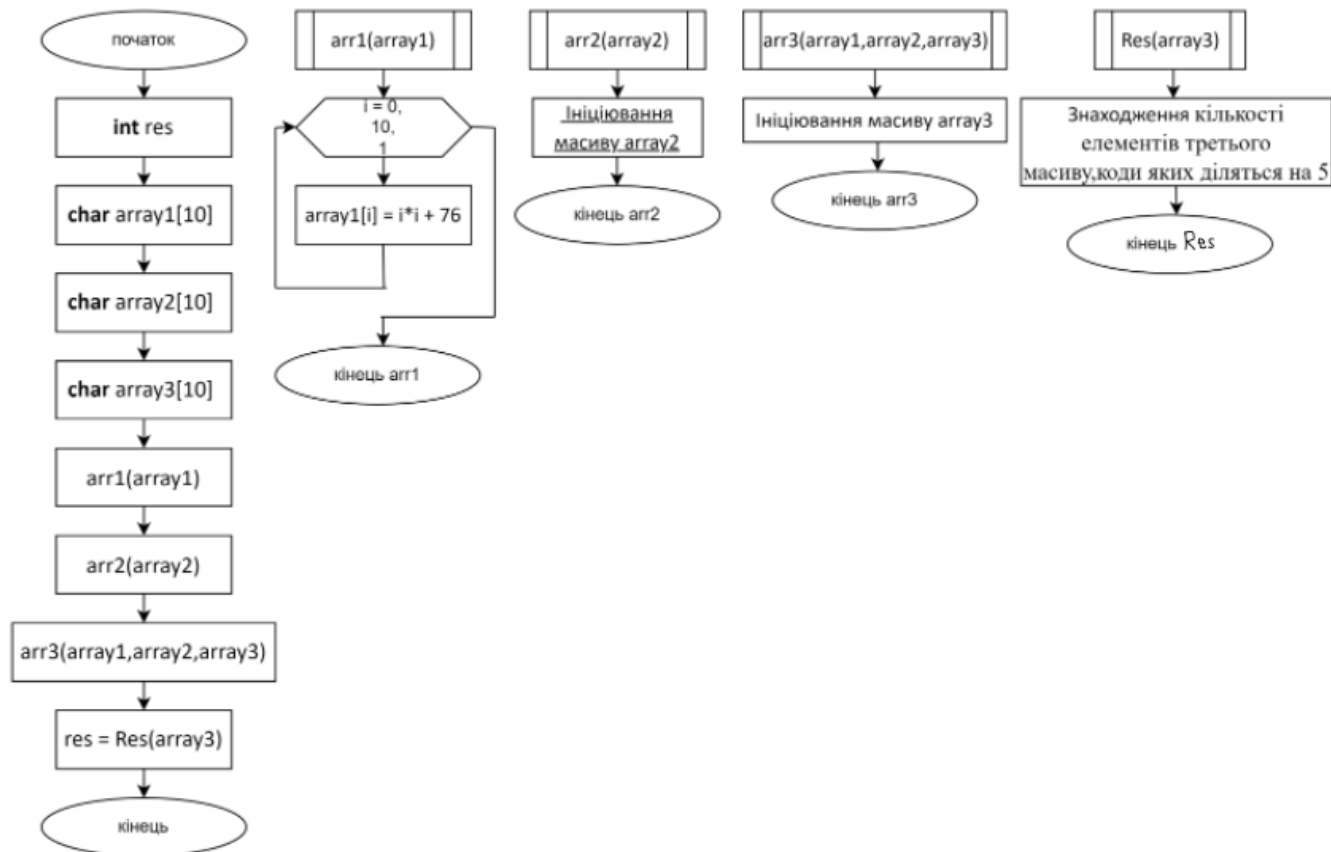
Блок-схема

Крок 1

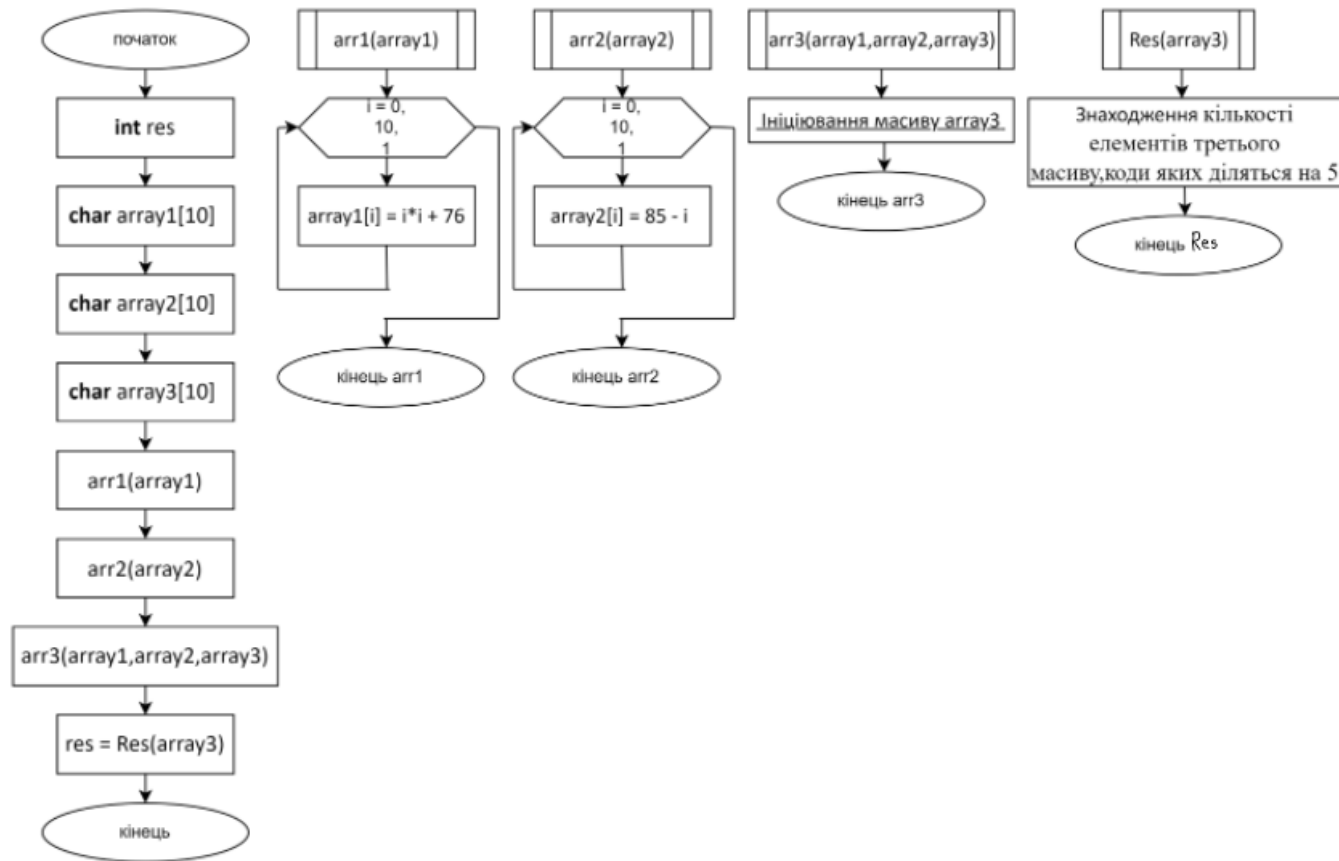


Крок 2

Алгоритми та структури даних. Основи алгоритмізації

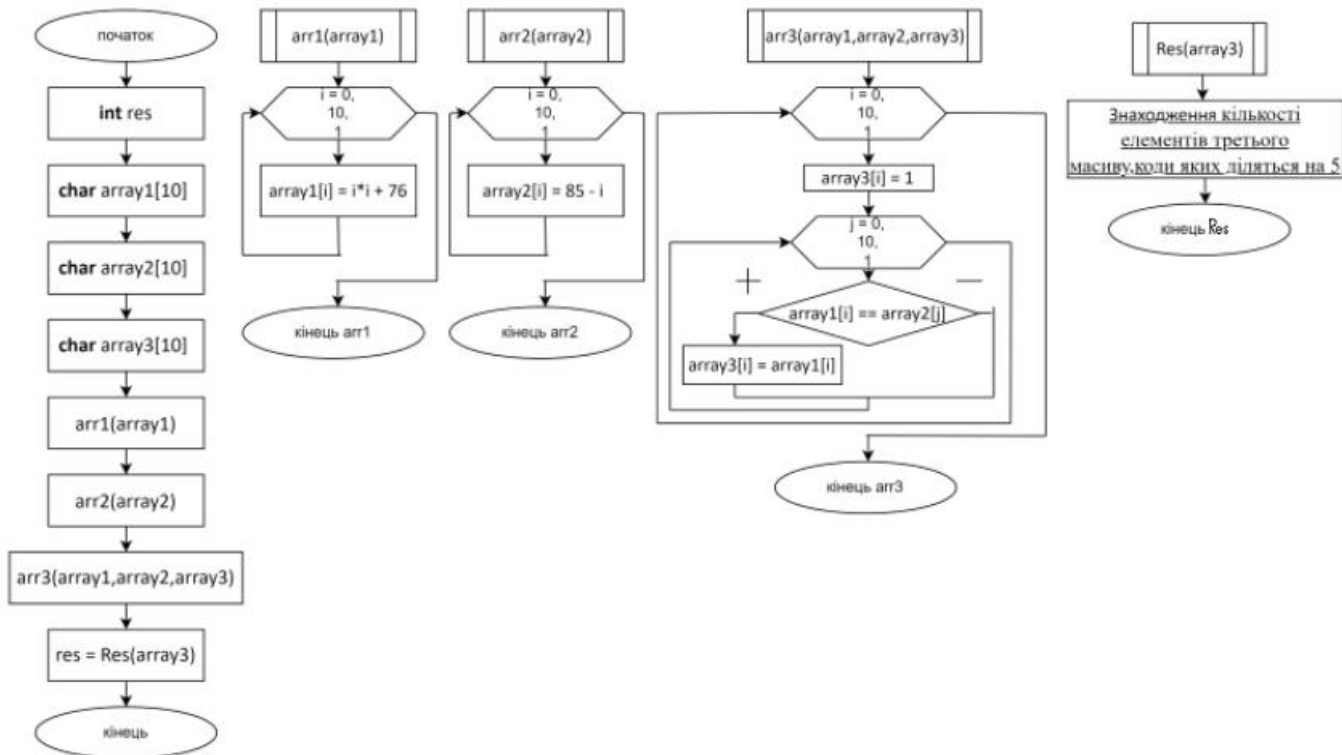


Крок 3

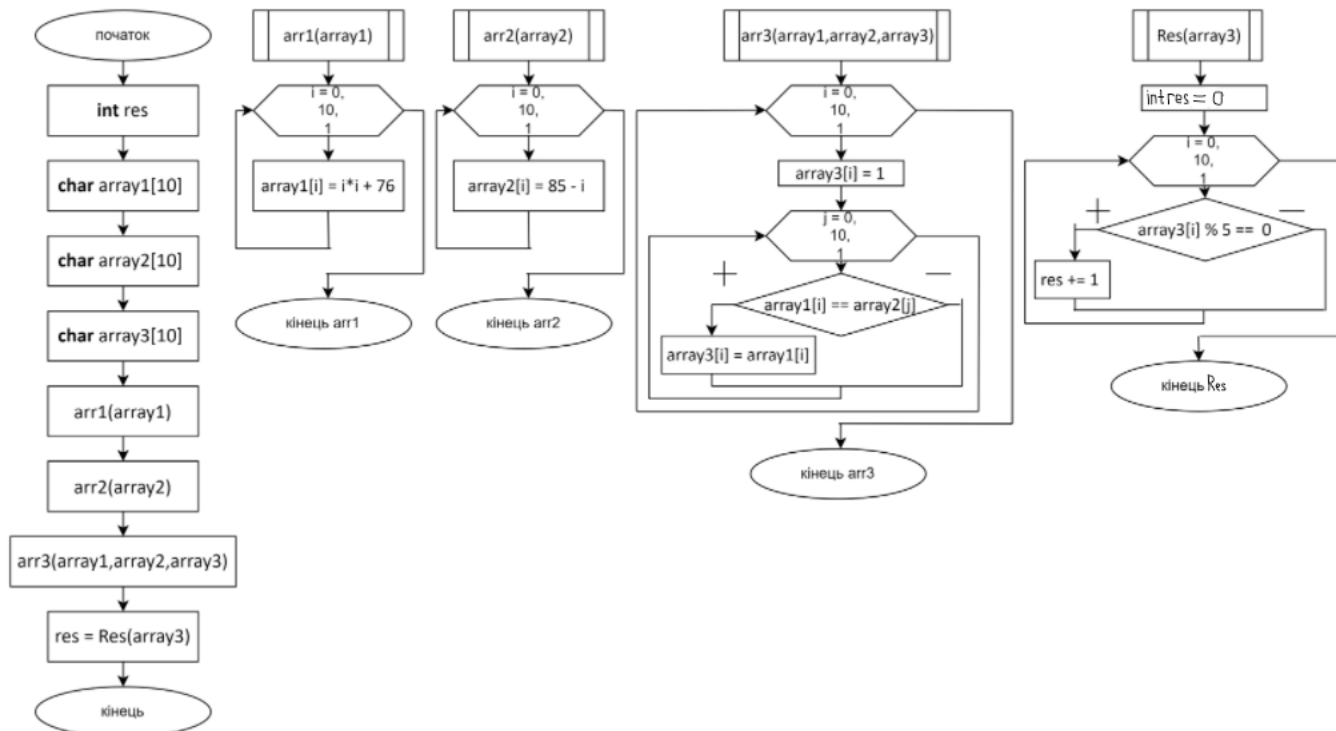


Алгоритми та структури даних. Основи алгоритмізації

Крок 4



Крок 5



Алгоритми та структури даних. Основи алгоритмізації

Тестування

Блок	Дія
	Початок
	firstNum = 1, Step = 3, Depth = 7, Dob = 1, Dep = 1
	Функція arr1,цикл з параметром i
1	i = 0 (int) array1[i] = 76
2	i = 1 (int) array1[i] = 77
3	i = 2 (int) array1[i] = 80
4	i = 3 (int) array1[i] = 85
5	i = 4 (int) array1[i] = 92
6	i = 5 (int) array1[i] = 101
7	i = 6 (int) array1[i] = 112
8	i = 7 (int) array1[i] = 125
9	i = 8 (int) array1[i] = 140
10	i = 9 (int) array1[i] = 157
	Функція arr2,цикл з параметром i
1	i = 0 (int) array1[i] = 85
2	i = 1 (int) array1[i] = 84
3	i = 2 (int) array1[i] = 83
4	i = 3 (int) array1[i] = 82

Алгоритми та структури даних. Основи алгоритмізації

5	i = 4 (int) array1[i] = 81
6	i = 5 (int) array1[i] = 80
7	i = 6 (int) array1[i] = 79
8	i = 7 (int) array1[i] = 78
9	i = 8 (int) array1[i] = 77
10	i = 9 (int) array1[i] = 76
	Функція arr3
1	i = 0 j = 0 array3[i] = 1
	...
100	i = 9 j = 9 array3[i] = 1
	Функція Res,цикл з параметром i
1	i = 0 array3[i] = 76 res = 0
	...
10	i = 9 array3[i] = 1 res = 2
	виведення res

Алгоритми та структури даних. Основи алгоритмізації

	Кінець
--	--------

Код

```
#include <iostream>
using namespace std;
void arr1(char array1[]);
void arr2(char array2[]);
void arr3(char array1[], char array2[], char array3[]);
int Res(char array3[]);
void test(char array1[], char array2[], char array3[],int res);
int main()
{
    int res;
    char array1[10];
    char array2[10];
    char array3[10];
    arr1(array1);
    arr2(array2);
    arr3(array1,array2,array3);
    res = Res(array3);
    cout << res;
    //test(array1, array2, array3, res);
}
```

```
void arr1(char array1[]) {
    for (int i = 0; i < 10; i++) {
        array1[i] = i * i + 76;
    }
}

void arr2(char array2[]) {
    for (int i = 0; i < 10; i++) {
        array2[i] = 85 - i;
    }
}

void arr3(char array1[], char array2[], char array3[]) {
    for (int i = 0; i < 10; i++) {
        array3[i] = 1;
        for (int j = 0; j < 10; j++) {
            if (array1[i] == array2[j]) {
                array3[i] = array1[i];
            }
        }
    }
}

int Res(char array3[]) {
    int res = 0;
    for (int i = 0; i < 10; i++) {
        if (array3[i] % 5 == 0) {
            res++;
        }
    }
    return res;
}
```

Алгоритми та структури даних. Основи алгоритмізації

```
void test(char array1[], char array2[], char array3[], int res) {
    for (int i = 0; i < 10; i++) {
        cout << (int)array1[i] << "\n";
    }
    cout << "\n\n";
    for (int j = 0; j < 10; j++) {
        cout << (int)array2[j] << "\n";
    }
    cout << "\n\n";
    for (int k = 0; k < 10; k++) {
        cout << (int)array3[k] << "\n";
    }
    cout << "\n\n";
    cout << res;
}
```

Результат коду

Microsoft Visual Studio Debug Console

```
2
C:\Users\STRIX\source\repos\ASD_Lab7\Debug\ASD_Lab7.exe (process 24032) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

З використанням функції test:

Microsoft Visual Studio Debug Console

```
76
77
80
85
92
101
112
125
-116
-99

85
84
83
82
81
80
79
78
77
76

76
77
80
85
1
1
1
1
1
1

2
C:\Users\STRIX\source\repos\ASD_Lab7\Debug\ASD_Lab7.exe (process 11488) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Алгоритми та структури даних. Основи алгоритмізації

Висновки

Я дослідив методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набув практичних навичок їх використання під час складання програмних специфікацій. Побудував мат. модель, псевдокод та блок схему. Протестував алгоритм.