

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 7

Виконав студент ІІІ-15, Гуменюк Олександр Володимирович

(шифр, прізвище, ім'я, по батькові)

Перевірила 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 неупорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій.

Варіант 7

Завдання

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

1. Опису трьох змінних індексованого типу з 10 символьних 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.

7	$95 - 3 * i$	$74 + 3 * i$	Середнє арифметичне елементів, коди яких менше 82
---	--------------	--------------	---

Постановка задачі

Використовуємо 3 підпрограми для ініціалізації трьох потрібних масивів. Перші два заповнюємо елементами типу char за вказаними формулами, а третій – це масив з елементів, які присутні і в першому, і в другому масиві та нулів. Після ініціалізації всіх трьох масивів, використовуємо підпрограми для знаходження середнього арифметичного елементів, коди яких менше 82, та виводимо це число.

Результатом розв'язку є знаходження та виведення середнього арифметичного елементів, коди яких менше 82, та виводимо це число.

Побудова математичної моделі

Таблиця імен змінних

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
Перший масив	char	firstArray	Проміжні дані
Другий масив	char	secondArray	Проміжні дані
Третій масив	char	thirdArray	Проміжні дані
Кількість елементів в масивах	int	arrayLength	Початкові дані
Середнє арифметичне елементів, коди яких менше 82	int	result	Результат
Лічильник i	int	i	Проміжні дані
Параметр: масив	char	array	Початкові дані
Процедура для ініціалізація першого масиву	Процедура	initializeFirstArray	Початкові дані
Процедура для ініціалізація другого масиву	Процедура	initializeSecondArray	Початкові дані

Процедура для ініціалізація третього масиву	Процедура	initializeThirdArray	Початкові дані
Параметр: перший масив	char	arr1	Початкові дані
Параметр: другий масив	char	arr2	Початкові дані
Лічильник j	int	j	Проміжні дані
Лічильник k	int	k	Проміжні дані
Кількість ненульових елементів в третьому масиві	int	num	Проміжні дані
Функція для знаходження середнього арифметичного елементів, коди яких менше 82	Функція (int)	processThirdArray	Початкові дані
Сума ненульових елементів в третьому масиві	int	sum	Проміжні дані

Спочатку задаємо 10 як значення константи `arrayLength`, яка позначає розмір масивів. Після цього по черзі ініціалізуємо перший (`firstArray`), другий (`secondArray`) та третій масив (`thirdArray`), використовуючи відповідні підпрограми.

- Для першого масиву використовуємо процедуру `initializeFirstArray`, яка приймає масив як параметр. Використовуючи арифметичний цикл з лічильником `i`, який набуває значень від 1 до `arrayLength`, процедура заповнює кожен відповідний елемент масиву за формулою $array[i] = 95 - 3 * i$. Після закінчення арифметичного циклу, процедура закінчується.
- Для другого масиву використовуємо процедуру `initializeSecondArray`, яка приймає масив як параметр. Використовуючи арифметичний цикл з лічильником `i`, який набуває значень від 1 до `arrayLength`, процедура заповнює кожен відповідний елемент масиву за формулою $array[i] = 74 + 3 * i$. Після закінчення арифметичного циклу, процедура закінчується.
- Для третього масиву використовуємо процедуру `initializeThirdArray`, яка приймає масив як параметр, а також два додаткових масиви (`arr1` і `arr2`), які вона буде порівнювати. Спочатку задаємо 0 у значення змінної `num` (кількість ненульових елементів в третьому масиві). Потім, використовуючи зовнішній арифметичний цикл з лічильником `i` та внутрішній арифметичний цикл з лічильником `j`, які обидва набувають значень від 1 до 10, функція порівнює кожне значення першого масиву з кожним значенням другого (`arr1[i] == arr2[j]`). Якщо знайдеться елемент, який буде присутній і в першому і в другому масиві, він записується в третій масив $array[num] = arr1[i]$, і значення `num`

збільшується на один. Після закінчення складного арифметичного циклу, процедура заповнює решту елементів третього масива нулями $array[k] = 0$; для цього використовуємо арифметичний цикл з лічильником k , який набуває значень від num до $arrayLength$. Після цього циклу процедура закінчується.

- Далі використовуємо функцію `processThirdArray` для знаходження середнього арифметичного елементів, коди яких менше 82. Ця функція приймає масив як параметр. Спочатку ми задаємо 0 у значення змінних sum і num . Далі, використовуємо арифметичний цикл з лічильником i , який набуває значень від 1 до $arrayLength$, перевіряємо необхідні умови для кожного елемента третього масиву ($array[i] < 82 \ \&\& \ array[i] \neq 0$), операція $\&\&$ - «логічне і»; якщо умова виконується додаємо це число до суми $sum += array[i]$, та збільшуємо значення sum на один. Після циклу, функція повертає значення виразу: sum/num .

Значення, яке повернуло функція `processThirdArray`, записуємо в змінну `result` та виводимо її.

Розв'язання

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

Крок 1. Визначимо основні дії.

Крок 2. Ініціалізація перших двох масивів та константи `arrayLength`

Крок 3. Ініціалізація третього масиву

Крок 4. Обчислення та виведення `result` (Середнє арифметичне елементів, коди яких менше 82)

Псевдокод

Основна програма:

Крок 1

початок

Ініціалізація перших двох масивів та константи arrayLength

Ініціалізація третього масиву

Обчислення та виведення result

кінець

Крок 2

початок

arrayLength := 10

initializeFirstArray (firstArray)

initializeSecondArray(secondArray)

Ініціалізація третього масиву

Обчислення та виведення result

кінець

Крок 3

початок

arrayLength := 10

initializeFirstArray (firstArray)

initializeSecondArray(secondArray)

initializeThirdArray(thirdArray, firstArray, secondArray)

Обчислення та виведення result

кінець

Крок 4

початок

arrayLength := 10

initializeFirstArray (firstArray)

initializeSecondArray(secondArray)

initializeThirdArray(thirdArray, firstArray, secondArray)

result := processThirdArray(thirdArray)

виведення result

кінець

Підпрограми:

initializeFirstArray (array)

повторити для i від 1 до arrayLength

$array[i] = 95 - 3*i$

все повторити

кінець

initializeSecondArray (array)

повторити для i від 1 до arrayLength

$array[i] = 74 + 3*i$

все повторити

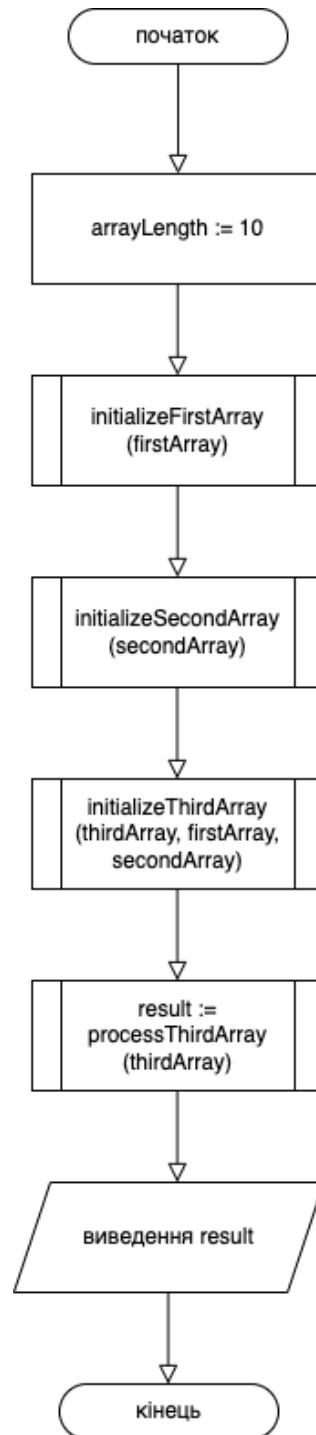
кінець


```
initializeThirdArray (array, arr1, arr2)
    num := 0
    повторити для i від 1 до arrayLength
        повторити для j від 1 до arrayLength
            якщо arr1[i] == arr2[j]
                то
                    array[num] := arr1[i]
                    num := num + 1
            все якщо
        все повторити
    все повторити
    повторити для k від num+1 до arrayLength
        array[k] := 0
    все повторити
кінець
```

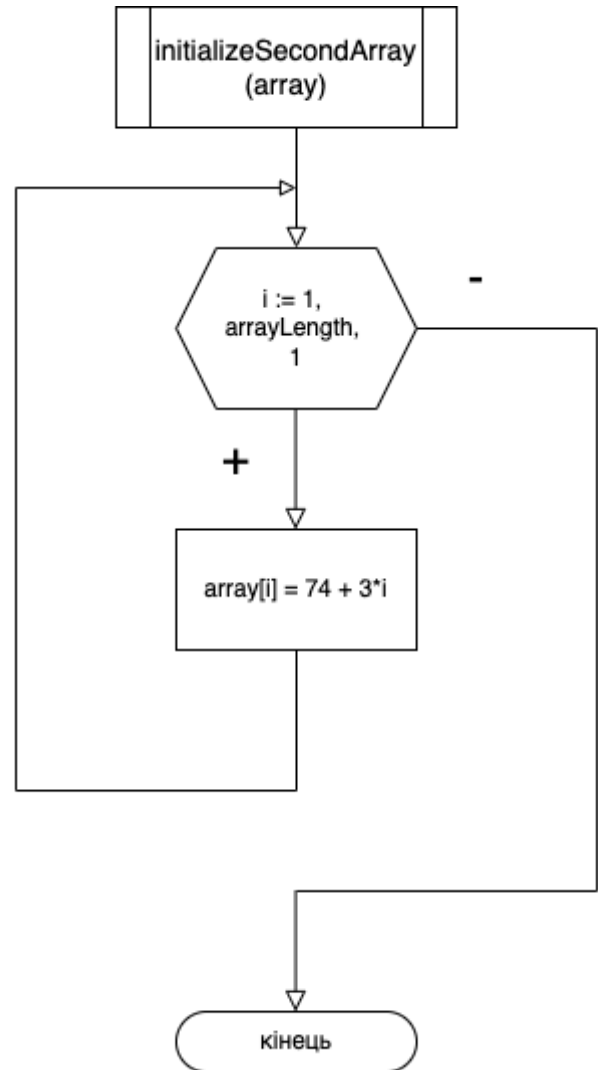
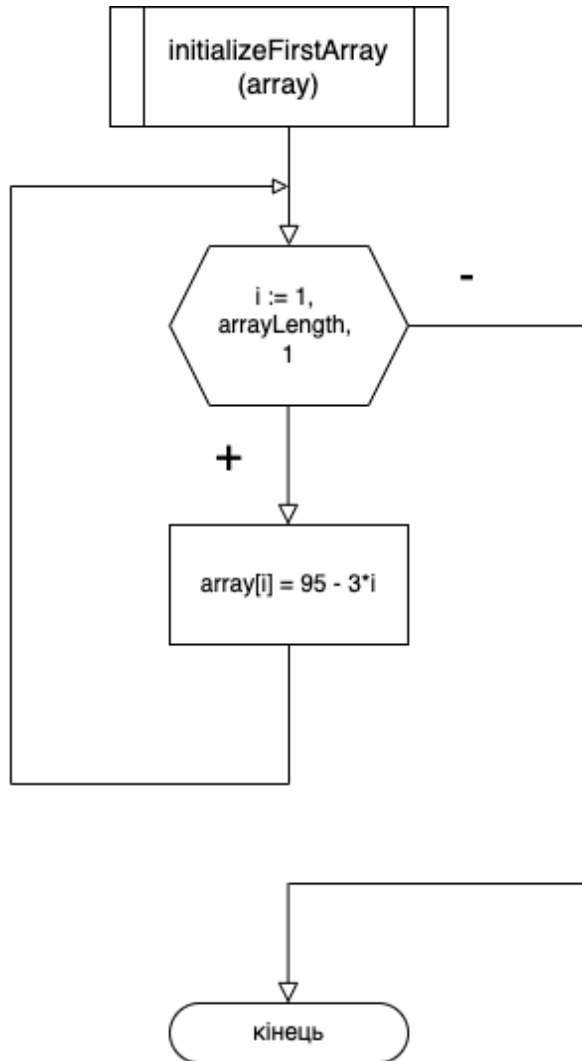
```
processThirdArray(array)
    sum := 0
    num := 0
    повторити для i від 1 до arrayLength
        якщо (array[i] < 82 && array[i] != 0)
            то
                sum := sum + array[i]
                num := num + 1
        все якщо
    все повторити
    повернути (sum/num)
кінець
```

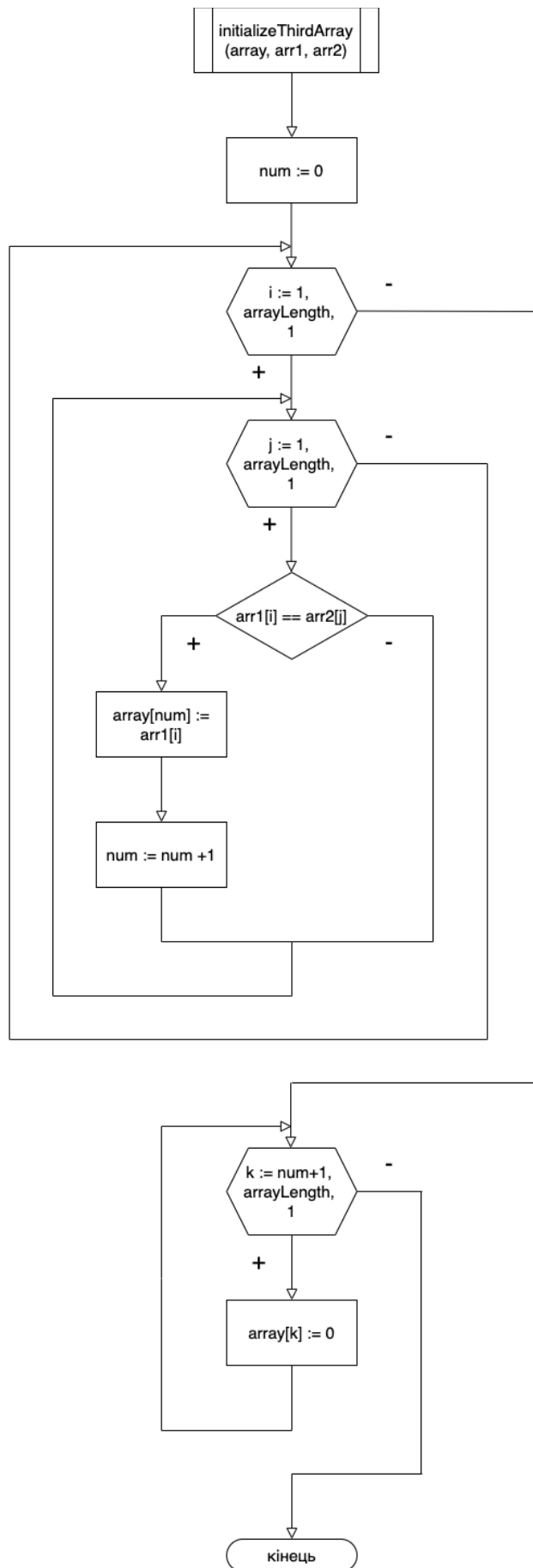
Блок-схема:

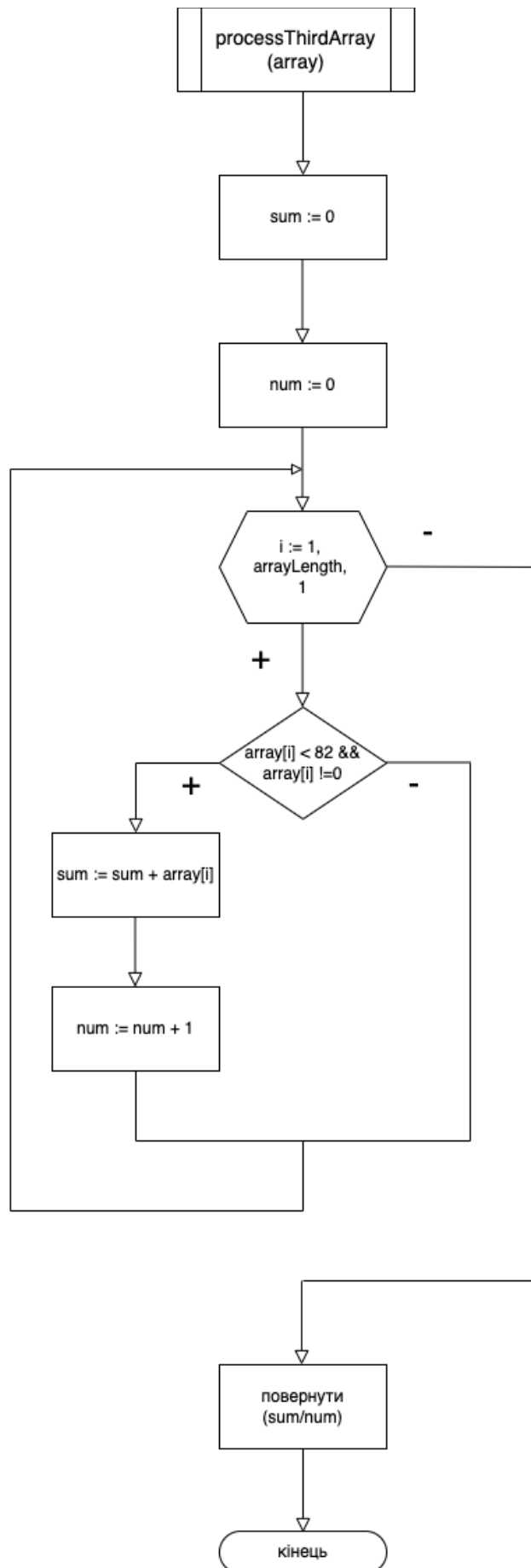
Основна програма:



Підпрограми:







Код:

Основна програма:

```
const int arrayLength = 10;

int main() {

    char firstArray[arrayLength], secondArray[arrayLength], thirdArray[arrayLength];

    initializeFirstArray(firstArray);
    initializeSecondArray(secondArray);
    initializeThirdArray(thirdArray, firstArray, secondArray);

    int result = processThirdArray(thirdArray);

    cout << "First Array:" << "\t";
    for (int i = 0; i<=arrayLength-1; i++){
        cout << firstArray[i] << "\t";
    }

    cout << "\nSecond Array:"<< "\t";
    for (int i = 0; i<=arrayLength-1; i++){
        cout << secondArray[i] << "\t";
    }

    cout << "\nThird Array:" << "\t";
    for (int i = 0; i<=arrayLength-1; i++){
        cout << thirdArray[i] << "\t";
    }

    cout << "\n" <<"The result is: \t"<< result << endl;

    return 0;
}
```

Підпрограми:

```
void initializeFirstArray(char array[]){  
    for (int i = 1; i<=arrayLength; i++){  
        array[i-1] = 95 - 3*i;  
    }  
}
```

```
void initializeSecondArray(char array[]){  
    for (int i = 1; i<=arrayLength; i++){  
        array[i-1] = 74 + 3*i;  
    }  
}
```

```
void initializeThirdArray(char array[],char arr1[], char arr2[]){  
  
    int num = 0;  
    for (int i = 0; i<=arrayLength-1; i++){  
  
        for (int j = 0; j<=arrayLength-1;j++){  
  
            if (arr1[i] == arr2[j]){  
                array[num] = arr1[i];  
                num++;  
            }  
        }  
    }  
  
    for (int k = num; k<=arrayLength-1;k++){  
        array[k] = 0;  
    }  
}
```

```

int processThirdArray(char array[]){
    int sum = 0;
    int num = 0;

    for (int i = 0; i <= arrayLength-1; i++){
        if (array[i] < 82 && array[i] !=0){
            sum += array[i];
            num++;
        }
    }

    return sum/num;
}

```

```

First Array:  \  Y  V  S  P  M  J  G  D  A
Second Array: M  P  S  V  Y  \  _  b  e  h
Third Array:  \  Y  V  S  P  M
The result is: 78
Program ended with exit code: 0

```

Тестування

Блок	Дія (основна програма)	Дія (підпрограми)
	Початок	
1	arrayLength := 10	
2	initializeFirstArray(firstArray)	
3		для і від 1 до 10: firstArray[i] = 95 - 3*i

4		firstArray = {92, 89, 86, 83, 80, 77, 74, 71, 68, 65} = {«\», «Y», «V», «S», «P», «M», «J», «G», «D», «A»}
5	initializeSecondArray(secondArray)	
6		для i від 1 до 10: firstArray[i] = 74 + 3*i
7		secondArray = {77,80,83,86,89,92,95,98,101,104} = {«M», «P», «S», «V», «Y», «\», «_», «b», «e», «h»}
8	initializeThirdArray(thirdArray, firstArray, secondArray);	
9		Однакові елементи: 92, 89, 86, 83, 80, 77
10		thirdArray = {92, 89, 86, 83, 80, 77, 0, 0, 0, 0} ={«\», «Y», «V», «S», «P», «M»}
11	result := processThirdArray(thirdArray);	
12		sum, num = 0
13		2 ненульових елемента менших за 86 → num = 2
14		sum = 80 + 77 = 157
15		157/2 = 78 повернути 78
16	result := 78	
17	виведення 78	
	Кінець	

Висновки

Протягом сьомої лабораторної роботи я дослідив методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набув практичних навичок їх використання під час складання програмних специфікацій. В результат я отримав алгоритм, що генерує масиви за даними формулами, а також будує новий масив, порівнюючі два інших. Також я отримав підпрограму, яка обчислює середнє арифметичне елементів масиву, які підходять під умови задачі.