

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 рекурсивних алгоритмів»

Варіант 27

Виконав студент

ІІ-13 Пархомчук Ілля Вікторович
(шифр, прізвище, ім'я, по батькові)

Перевірив

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 набути практичних навичок їх використання під час складання програмних специфікацій.

Варіант 27

№	Вираз для обчислення елемента		Знайти
	1-го масиву	2-го масив	
27	$95 + i$	$105 - i$	Суму елементів, коди яких більше 101

Постановка задачі

Потрібно спочатку згенерувати 2 масива, потім на їх основі третій і виконати відповідні дії над ним.

Побудова математичної моделі

Таблиця змінних

Змінна	Тип	Ім'я	Призначення
Перший масив	Індексований символний	first_Arr	Проміжні дані
Другий масив	Індексований символний	second_Arr	Проміжні дані
Третій масив	Індексований символний	third_Arr	Проміжні дані
Лічильник	Цілочисельний	i	Проміжні дані
Лічильник	Цілочисельний	j	Проміжні дані
Лічильник елементів 3-го масива	Цілочисельний	arr3_size	Проміжні дані
Результат обробки 3-ї змінної	Цілочисельний	result	Вихідні дані

Таблиця аргументів підпрограм

Масив до друку	Індексований символьний	arr	Проміжні дані
Розмір масиву до друку	Цілочисельний	size	Проміжні дані

Таблиця підпрограм

Підпрограма	Оголошення	Опис
print_arr	print_arr(arr[], size)	Друкує переданий масив

Таблиця констант

Змінна	Тип	Ім'я	Значення
Розмір першого і другого масиву	Цілочисельний	SIZE	10

Перший масив потрібно ініціалізувати за виразом $95 + i$, другий ініціалізувати за виразом $105 - i$, де i – це номер елемента масиву.

Третій масив потрібно ініціалізувати спільними елементами, що є i в 1-му і в 2-му масивах.

Далі, обчислити суму всіх елементів в 3-му масиві, що більші за 101.

Розв'язання

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо ініціалізацію першого і другого масиву з використанням арифметичного оператора повторення.

Крок 3. Деталізуємо ініціалізацію третього масиву з використанням арифметичного оператора повторення та умовної форми вибору.

Крок 4. Деталізуємо обчислення результату з використанням арифметичного оператора повторення та умовної форми вибору.

Крок 5. Деталізуємо вивід трьох масивів і результату обробки третього.

Псевдокод алгоритму

крок 1

Основна програма

початок

ініціювання першого та другого масиву

ініціювання третього масиву

обчислення результату

вивід масивів та результату

кінець

Підпрограми

підпрограма print_arr(arr[], size)

вивід переданого масиву

все підпрограма

крок 2

Основна програма

початок

для i від 0 до $SIZE$, з кроком 1 повторити

$first_Arr[i] := 95 + i$

$second_Arr[i] := 105 - i$

все повторити

ініціювання третього масиву

обчислення результату

вивід масивів та результату

кінець

Підпрограми

підпрограма print_arr(arr[], size)

вивід переданого масиву

все підпрограма

крок 3

Основна програма

початок

для i від 0 до $SIZE$, з кроком 1 повторити

$first_Arr[i] := 95 + i$

$second_Arr[i] := 105 - i$

все повторити

для i від 0 до $SIZE$, з кроком 1 повторити

 для j від 0 до $SIZE$, з кроком 1 повторити

 якщо $first_Arr[i] == second_Arr[j]$

то

$third_Arr[arr3_size] := first_Arr[i]$

$arr3_size := arr3_size + 1$

все якщо

все повторити

все повторити

обчислення результату

вивід масивів та результату

кінець

Підпрограми

підпрограма $print_arr(arr[], size)$

 вивід переданого масиву

все підпрограма

крок 4

Основна програма

початок

для i **від** 0 **до** $SIZE$, **з кроком** 1 **повторити**

$first_Arr[i] := 95 + i$

$second_Arr[i] := 105 - i$

все повторити

для i **від** 0 **до** $SIZE$, **з кроком** 1 **повторити**

для j **від** 0 **до** $SIZE$, **з кроком** 1 **повторити**

якщо $first_Arr[i] == second_Arr[j]$

то

$third_Arr[arr3_size] := first_Arr[i]$

$arr3_size := arr3_size + 1$

все якщо

все повторити

все повторити

для i **від** 0 **до** $arr3_size$, **з кроком** 1 **повторити**

якщо $third_Arr[i] > 101$

то

$result := result + third_Arr[i]$

все якщо

все повторити

вивід масивів та результату

кінець

Підпрограми

підпрограма $print_arr(arr[], size)$

вивід переданого масиву

все підпрограма

крок 5

Основна програма

початок

для i **від** 0 **до** $SIZE$, **з кроком** 1 **повторити**

$first_Arr[i] := 95 + i$

$second_Arr[i] := 105 - i$

все повторити

для i **від** 0 **до** $SIZE$, **з кроком** 1 **повторити**

для j **від** 0 **до** $SIZE$, **з кроком** 1 **повторити**

якщо $first_Arr[i] == second_Arr[j]$

то

$third_Arr[arr3_size] := first_Arr[i]$

$arr3_size := arr3_size + 1$

все якщо

все повторити

все повторити

для i **від** 0 **до** $arr3_size$, **з кроком** 1 **повторити**

якщо $third_Arr[i] > 101$

то

$result := result + third_Arr[i]$

все якщо

все повторити

вивід «First array: »

$print_arr(first_Arr, SIZE)$

вивід «Second array: »

$print_arr(second_Arr, SIZE)$

вивід «Third array: »

$print_arr(third_Arr, arr3_size)$

вивід $result$

кінець

Підпрограми

підпрограма $print_arr(arr[], size)$

для i **від** 0 **до** $size$, **з кроком** 1 **повторити**

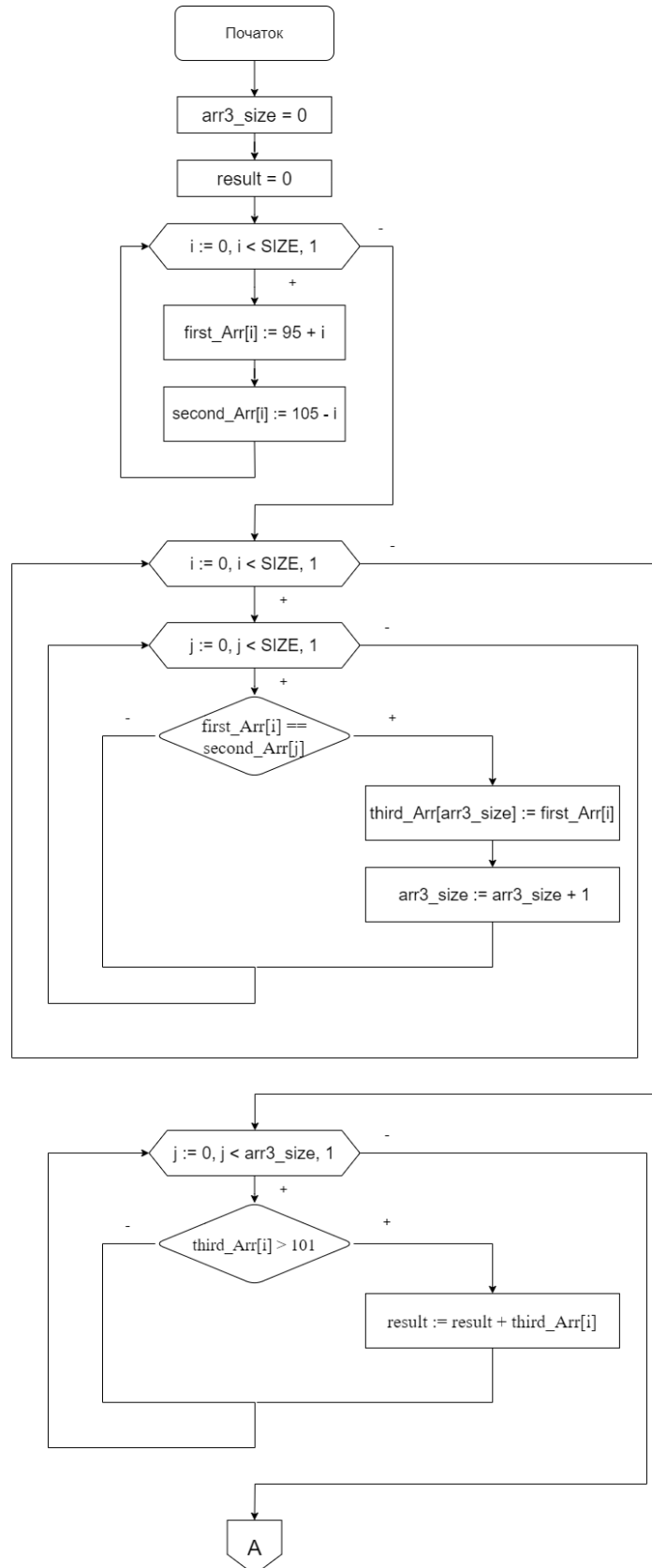
вивід $arr[i]$

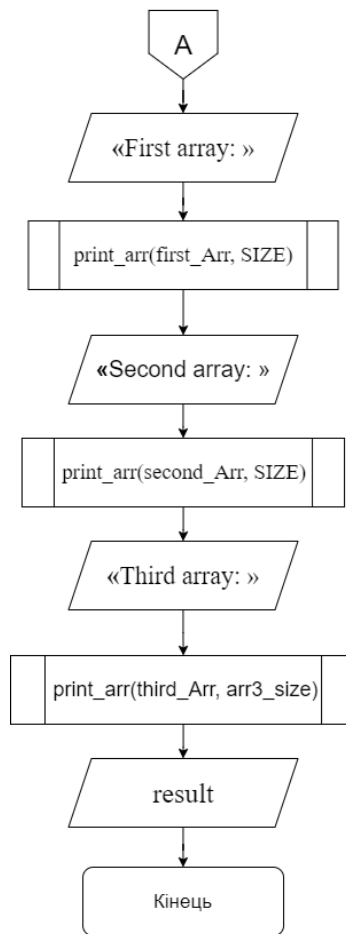
все повторити

все підпрограма

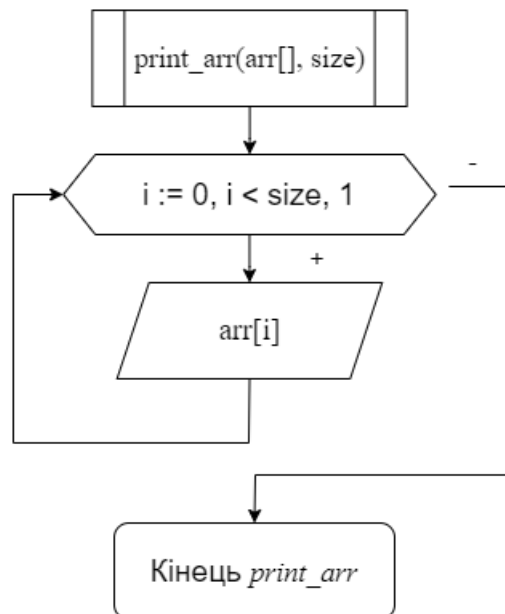
Блок-схема

Основна програма:





Підпрограма:



Код програми

```
#include <iostream>

using std::cout;
using std::cin;

void print_arr(char arr[], int size);

int main()
{
    const int SIZE = 10;
    char first_Arr[SIZE];
    char second_Arr[SIZE];
    char third_Arr[SIZE] = { '\0' };
    int arr3_size = 0;
    int result = 0;

    for (int i = 0; i < SIZE; ++i)
    {
        first_Arr[i] = 95 + i;
        second_Arr[i] = 105 - i;
    }

    for (int i = 0; i < SIZE; i++)
    {
        for(int j = 0; j < SIZE;j++)

            if (first_Arr[i] == second_Arr[j])
            {
                third_Arr[arr3_size] = first_Arr[i] ;
                ++arr3_size;
            }
    }

    for (int i = 0; i < arr3_size; i++)
    {
        if (third_Arr[i] > 101)
        {
            result += third_Arr[i];
        }
    }

    cout << "First array:\n";
    print_arr(first_Arr, SIZE);
    cout << "Second array:\n";
    print_arr(second_Arr, SIZE);
    cout << "Third array:\n";
    print_arr(third_Arr, arr3_size);

    cout << "\nres = " << result << '\n';

    system("pause");
    return 0;
}

void print_arr(char arr[], int size)
{
    for (int i = 0; i < size; i++)
    {
        cout << arr[i] << " ";
    }
    cout << '\n';
}
```

Випробування алгоритму

```
First array:  
_ ` a b c d e f g h  
Second array:  
i h g f e d c b a `  
Third array:  
` a b c d e f g h  
5)  
res = 309
```

Висновки

На лабораторній роботі я дослідив методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набув практичних навичок їх використання під час складання програмних специфікацій. Спочаку я ініціалізував перший і другий масиви за вказаними виразами. Потім ініціалізував третій масив елементами спільними для другого і першого і обчислив суму елементів цього масиву, що більші за 101. Також створив підпрограму для виводу масиву. Мною був написаний псевдокод, створено блок-схеми та проведено тестування алгоритму.