

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 № 6 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження лінійного пошуку в послідовностях»

Варіант 35

Виконав студент ІП-15, Шабанов Метін Шаміль огли
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Лабораторна робота 7

Дослідження складних циклічних алгоритмів

Мета – дослідити методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій.

Індивідуальне завдання

Варіант 35

Умова задачі

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.
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної згідно з варіантом

$i * i + 74$	$78 - i$	Знайти суму двох мінімальних елементів
--------------	----------	--

Постановка задачі

Обчислити суму двох мінімальних елементів масивів, у яких елементи генеруються за правилами $i * i + 74$ та $78 - i$ відповідно. Пошук мінімального елемента буде відбуватися за допомогою алгоритму порівняння елементів.

Побудова математичної моделі

Змінна	Тип	Ім'я	Призначення
Генератор першого масиву	Підпрограма	firstArrayGenerator()	Функція
Генератор другого масиву	Підпрограма	secondArrayGenerator()	Функція
Пошук мінімального елемента	Підпрограма	findMinimum(array[], length)	Функція
Перший масив	Індексований	array1[]	Проміжні дані
Другий масив	Індексований	array2[]	Проміжні дані
Довжина масива	Ціле додатне	size	Проміжні дані
Поточний елемент першого масиву, лічильник	Ціле додатне	i	Проміжні дані

першого циклу			
Поточний елемент другого масиву, лічильник другого циклу	Ціле додатне	j	Проміжні дані
Лічильник циклу для пошуку мін. елемента	Ціле додатне	f	Проміжні дані
Перший параметр функції findMinimum	Індексований	array[]	Проміжні дані
Другий параметр функції findMinimum	Ціле додатне	length	Проміжні дані
Мінімальний елемент першого масиву	Ціле додатне	min1	Проміжні дані
Мінімальний елемент другого масива	Ціле додатне	min2	Проміжні дані
Сума двох мінімальних елементів	Ціле	sum	Результат

Для створення масивів скористаємося функціями firstArrayGenerator та secondArrayGenerator та відповідними циклами, після чого в першому та другому масиві за допомогою функції findMinimum знайдемо мінімальні елементи, а потім визначимо їх суму.

Розв'язання

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

Крок 1. Визначимо основні дії;

Крок 2. Деталізуємо присвоєння значення змінній size

Крок 3. Деталізуємо роботу циклів для побудови масиву;

Крок 4. Деталізуємо роботу та виклик функцій firstArrayGenerator та secondArrayGenerator.

Крок 5. Деталізуємо виклик функції findMinimum в основній програмі.

Крок 6. Деталізуємо роботу функції findMinimum.

Крок 7. Деталізуємо знаходження суми мінімальних елементів.

Псевдокод

Крок 1

Початок

Присвоєння значення змінній `size`

Робота циклів для побудови масиву

Робота та виклик функцій `firstArrayGenerator` та `secondArrayGenerator`

Виклик функції `findMinimum` в основній програмі

Робота функції `findMinimum`

Знаходження суми мінімальних елементів

Кінець

Крок 2

Початок

`size = 10`

Робота циклів для побудови масиву

Робота та виклик функцій `firstArrayGenerator` та `secondArrayGenerator`

Виклик функції `findMinimum` в основній програмі

Робота функції `findMinimum`

Знаходження суми мінімальних елементів

Кінець

Крок 3

Початок

Основна програма

`size = 10`

повторити

для `i` від 1 до `size`

Робота та виклик функцій `firstArrayGenerator`

все повторити

повторити

для j від 1 до size

Робота та виклик функцій secondArrayGenerator

все повторити

Виклик функції findMinimum в основній програмі

Робота функції findMinimum

Знаходження суми мінімальних елементів

Кінець

Крок 4

Початок

Основна програма

size = 10

повторити

для i від 1 до size

array1[i] = firstArrayGenerator(i)

все повторити

повторити

для j від 1 до size

array2[j] = secondArrayGenerator(j)

все повторити

Виклик функції findMinimum в основній програмі

Робота функції findMinimum

Знаходження суми мінімальних елементів

Підпрограма firstArrayGenerator (i)

повернути $i * i + 74$

Підпрограма secondArrayGenerator (j)

повернути $78 - j$

Кінець

Крок 5

Початок

Основна програма

size = 10

повторити

для i від 0 до size

array1[i] = firstArrayGenerator(i)

все повторити

повторити

для j від 0 до size

array2[j] = secondArrayGenerator(j)

все повторити

min1 = findMinimum(array1, size)

min2 = findMinimum(array2, size)

Робота функції findMinimum

Знаходження суми мінімальних елементів

Підпрограма firstArrayGenerator (i)

повернути $i * i + 74$

Підпрограма secondArrayGenerator (j)

повернути $78 - j$

Кінець

Крок 6

Основна програма

size = 10

повторити

для i від 0 до size

array1[i] = firstArrayGenerator(i)

все повторити

повторити

для j від 0 до size

array2[j] = secondArrayGenerator(j)

все повторити

min1 = findMinimum(array1, size)

min2 = findMinimum(array2, size)

Знаходження суми мінімальних елементів

Підпрограма firstArrayGenerator (i)

повернути $i * i + 74$

Підпрограма secondArrayGenerator (j)

повернути $78 - j$

Підпрограма findMinimum (array[], length)

min = array[0]

повторити

для f від 0 до length

якщо array[f] < min

min = array[f]

все якщо

повернути min

Кінець

Крок 7

Основна програма

size = 10

повторити

для i від 0 до size

array1[i] = firstArrayGenerator(i)

все повторити

повторити

для j від 0 до size

array2[j] = secondArrayGenerator(j)

все повторити

min1 = findMinimum(array1, size)

min2 = findMinimum(array2, size)

sum = min1 + min2

Підпрограма firstArrayGenerator (i)

повернути i * i + 74

Підпрограма secondArrayGenerator (j)

повернути 78 – j

Підпрограма findMinimum (array[], length)

min = array[0]

повторити

для f від 0 до length

якщо array[f] < min

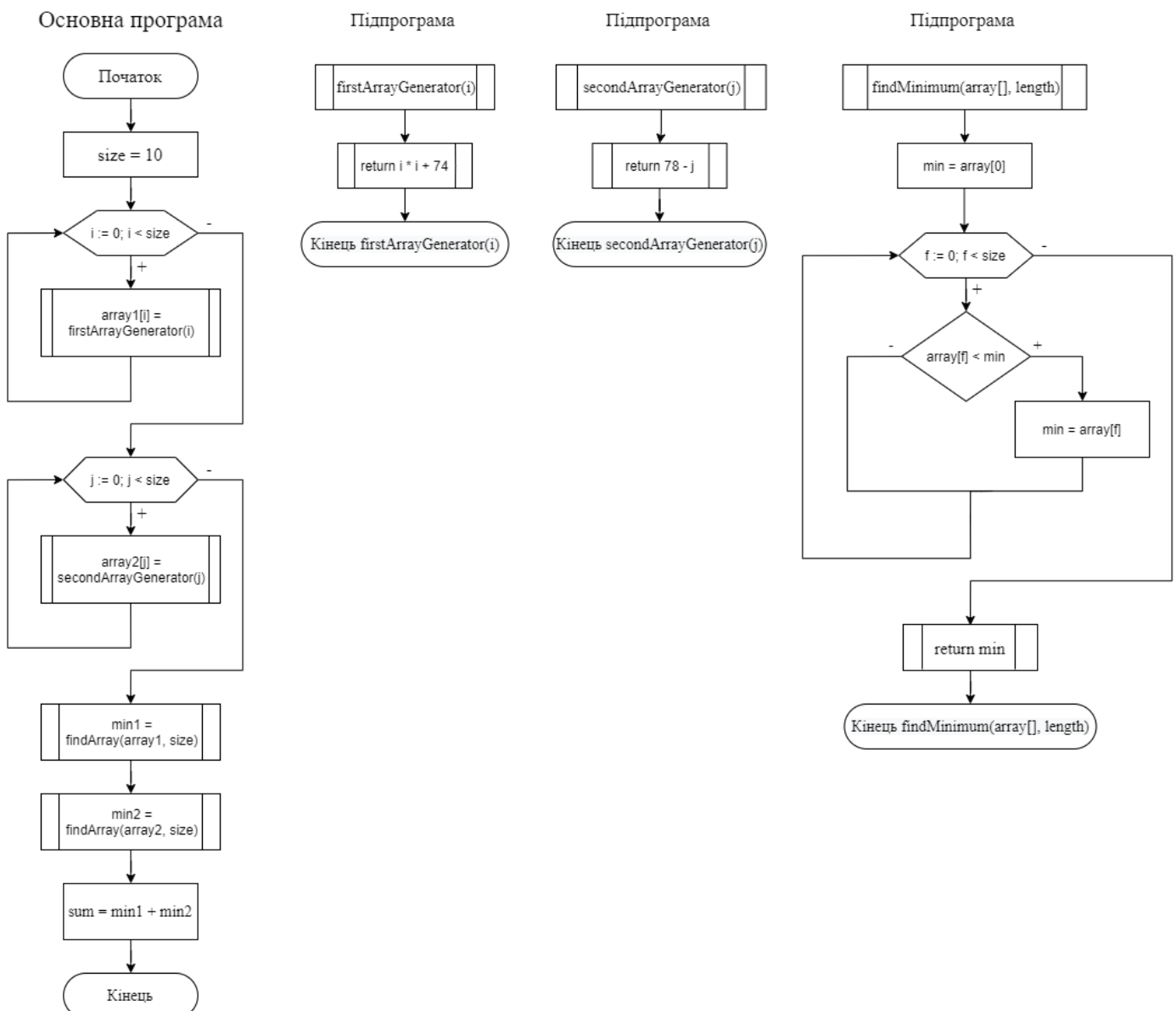
min = array[f]

все якщо

повернути min

Кінець

Блок схема



Код програми

```
#include <iostream>
#include <iomanip>
using namespace std;

unsigned char firstArrayGenerator(int i);

unsigned char secondArrayGenerator(int j);

unsigned char findMinimum(unsigned char array[], int length);

int main()
{
    const int size = 10;
    unsigned char array1[size] = {0};
    cout << "First array: " << endl;
    for (int i = 0; i < size; i++)
    {
        array1[i] = firstArrayGenerator(i);
        cout << array1[i] << setw(5) << "id: " << (int)array1[i] << endl;
    }

    cout << endl;

    cout << "Second array: " << endl;
    unsigned char array2[size] = {0};
    for (int j = 0; j < size; j++)
    {
        array2[j] = secondArrayGenerator(j);
        cout << array2[j] << setw(5) << "id: " << (int)array2[j] << endl;
    }

    cout << endl;

    unsigned char min1 = findMinimum(array1, size);
    unsigned char min2 = findMinimum(array2, size);
    cout << "The first minimum is " << (int)min1 << ", and the second is " << (int)min2 << endl;

    cout << endl;
    unsigned char sum = (int)min1 + (int)min2;
    cout << "The sum is " << (int)sum;
}

unsigned char firstArrayGenerator(int i)
{
    return i * i + 74;
}

unsigned char secondArrayGenerator(int j)
{
    return 78 - j;
}

unsigned char findMinimum(unsigned char array[], int length)
{
    unsigned char min = array[0];
    for (int f = 0; f < length; f++)
    {
        if (array[f] < min)
        {
            min = array[f];
        }
    }
    return min;
}
```

Випробовування



```
Консоль отладки Microsoft Visual Studio
First array:
J id: 74
K id: 75
N id: 78
S id: 83
Z id: 90
c id: 99
n id: 110
{ id: 123
K id: 138
W id: 155

Second array:
N id: 78
M id: 77
L id: 76
K id: 75
J id: 74
I id: 73
H id: 72
G id: 71
F id: 70
E id: 69

The first minimum is 74, and the second is 69

The sum is 143
C:\Users\mitya\source\repos\C++Lessons\Debug\C++Lessons.exe (процесс 9972) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Висновки

Ми дослідили методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набули практичних навичок їх використання під час складання програмних специфікацій. Як результат, ми отримали алгоритм обчислення суми двох мінімальних елементів масивів, у яких елементи генеруються за правилами $i * i + 74$ та $78 - i$ відповідно, розділивши задачу на сім кроків: визначення основних дій, присвоєння значення змінній `size`, робота циклів для побудови масиву, робота та виклик функцій `firstArrayGenerator` та `secondArrayGenerator`, виклик функції `findMinimum` в основній програмі, робота функції `findMinimum`, знаходження суми мінімальних елементів. В процесі випробовування ми отримали результат 143.