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 28

Виконав студент ІП-12, Сімчук Андрій Володимирович

(шифр, прізвище, ім'я, по батькові)

Перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 7**

**Дослідження лінійного пошуку в послідовностях**

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

**I. Задача.**

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 (***66 + 3 \* і****;* ***78 – і***).

3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.

4. Обробки третьої змінної згідно з варіантом (***Сума кодів мінімального та максимального елементів***).

**II. Розв’язання.**

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

*Крок 1.* Визначимо основні дії.

*Крок 2.* Деталізація ініціалізації змінних.

*Крок 3.* Деталізуємо дію заповнення перших двох змінних індексованого типу.

*Крок 4.* Деталізуємо дію заповнення третьої змінної індексованого типу.

*Крок 5.* Деталізуємо дію знаходження суми кодів мінімального та максимального елементів третьої змінної індексованого типу.

***Побудова математичної моделі***

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Розмір масивів | Цілий | size | Початкове дане |
| Масив з формулою елемента  ***66 + 3 \* і*** | Індексований | array1 | Проміжне значення |
| Масив з формулою елемента ***78 – і*** | Індексований | array2 | Проміжне значення |
| Масив рівних елементів перший двох масивів | Індексований | Array3 | Проміжне значення |
| Кількість елементів третього масиву | Натуральний | k | Проміжне значення |
| Максимальний елемент третього масиву | Цілий | maxel | Проміжне значення |
| Мінімальний елемент третього масиву | Цілий | minel | Проміжне значення |
| Показник лічильнику | Цілий | i | Проміжне значення |
| Показник лічильнику | Цілий | j | Проміжне значення |
| Сума кодів мінімального та максимального елементів | Цілий | res | Результат |

***Функції:***

FillArrays() – функція заповнення перших двох змінних індексованого типу;

FillThirdArray() – функція заповнення третьої змінної індексованого типу;

SumOfMinMax() – функція знаходження суми кодів мінімального та максимального елементів третьої змінної індексованого типу;

***Псевдокод (основна програма):***

*Крок 1.*

**Початок**

ініціалізація змінних

заповнення перших двох змінних індексованого типу

заповнення третьої змінної індексованого типу

знаходження суми кодів мінімального та максимального елементів третьої

змінної індексованого типу

**Виведення res**

**Кінець**

*Крок 2.*

**Початок**

size = 10

k = 0

array1[size]

array2[size]

array3[size]

заповнення перших двох змінних індексованого типу

заповнення третьої змінної індексованого типу

знаходження суми кодів мінімального та максимального елементів третьої

змінної індексованого типу

**Виведення res**

**Кінець**

*Крок 3.*

**Початок**

size = 10

k = 0

array1[size]

array2[size]

array3[size]

FillArrays(array1[], array2[], size)

заповнення третьої змінної індексованого типу

знаходження суми кодів мінімального та максимального елементів третьої

змінної індексованого типу

**Виведення res**

**Кінець**

*Крок 4.*

**Початок**

size = 10

k = 0

array1[size]

array2[size]

array3[size]

FillArrays(array1[], array2[], size)

FillThirdArray(array1[], array2[], array3[], size, k)

знаходження суми кодів мінімального та максимального елементів третьої

змінної індексованого типу

**Виведення res**

**Кінець**

*Крок 5.*

**Початок**

size = 10

k = 0

array1[size]

array2[size]

array3[size]

FillArrays(array1[], array2[], size)

FillThirdArray(array1[], array2[], array3[], size, k)

res = SumOfMinMax(array3[], k)

**Виведення res**

**Кінець**

***Псевдокод (функція FillArrays):***

**FillArrays(array1[], array2[], size)**

**повторити**

**для** і **від** 0 **до** size **із кроком** 1

array1[i] = 66 + 3 \* i

array2[i] = 78 – i

**все повторити**

**кінець**

***Псевдокод (функція FillThirdArray):***

**FillThirdArray(array1[], array2[], array3[], size, k)**

**повторити**

**для** і **від** 0 **до** size **із кроком** 1

**повторити**

**для** j **від** 0 **до** size **із кроком** 1

**якщо** array1[i] = array2[j]

**то**

array3[k] = array1[i]

k = k + 1

**все якщо**

**все повторити**

**все повторити**

**кінець**

***Псевдокод (функція SumOfMinMax):***

**SumOfMinMax(array3[], k)**

res = 0

minel = array3[0]

maxel = array3[0]

**повторити**

**для** і **від** 0 **до** k **із кроком** 1

**якщо** array3[i] < minel

**то**

minel = array3[i]

**все якщо**

**якщо** array3[i] > maxel

**то**

maxel = array3[i]

**все якщо**

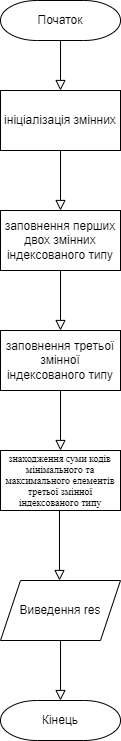
**все повторити**

res = minel + maxel

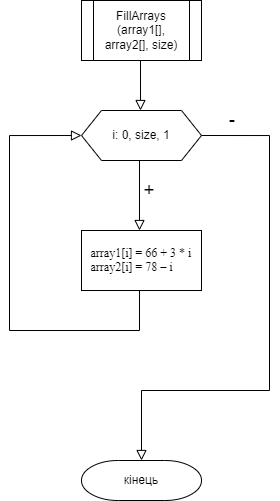
**повернути** res

***Блок-схема (Основна програма)***

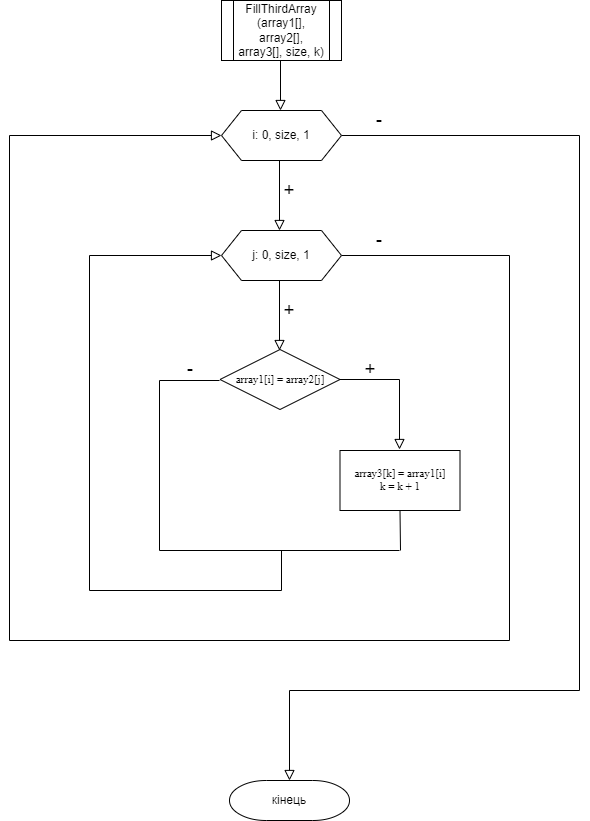
*Крок 1. Крок 2. Крок3. Крок 4. Крок 5.*



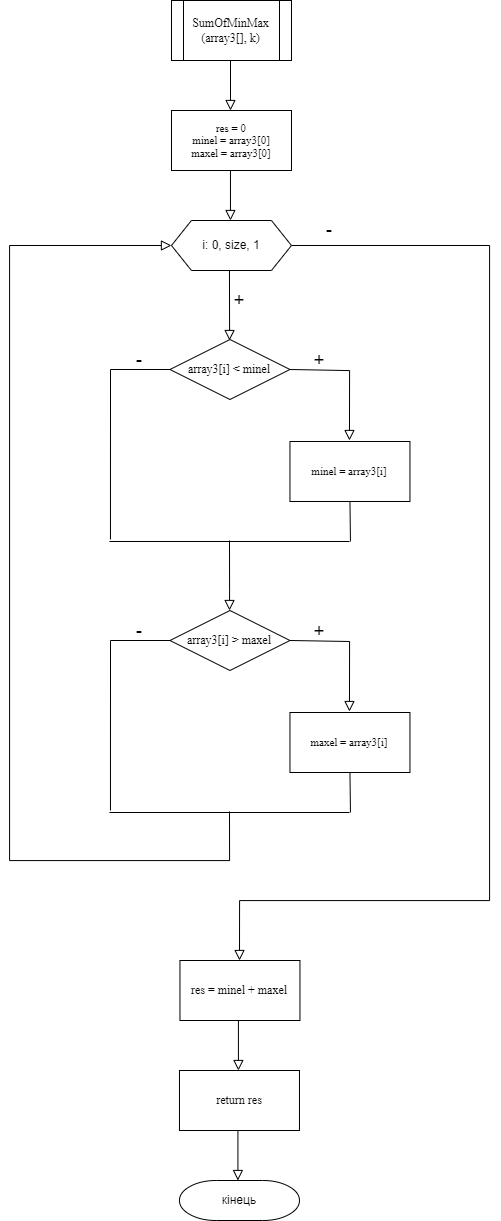
***Блок-схема (функція FillArrays)***



***Блок-схема (функція FillThirdArray)***



***Блок-схема (функція SumOfMinMax)***



***Код програми на мові С++***

#include <iostream>

#include <math.h>

#include <stdio.h>

using namespace std;

void FillArrays(char[], char[], int);

void FillThirdArray(char[], char[], char[], int, int&);

int SumOfMinMax(char[], int);

void OutputArray(char[], int);

int main() {

int res, k = 0;

const int size = 10;

char array1[size], array2[size], array3[size];

FillArrays(array1, array2, size);

cout << "Array1: ";

OutputArray(array1, size);

cout << "Array2: ";

OutputArray(array2, size);

FillThirdArray(array1, array2, array3, size, k);

res = SumOfMinMax(array3, k);

cout << "Array3: ";

OutputArray(array3, k);

cout << "The sum of the codes of the minimum and maximum elements: " << res << endl;

system("pause");

return 0;

}

void FillArrays(char array1[], char array2[], int size) {

for (int i = 0; i < size; i++) {

array1[i] = 66 + 3 \* i;

array2[i] = 78 - i;

}

}

void FillThirdArray(char array1[], char array2[], char array3[], int size, int &k) {

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

if (array1[i] == array2[j]) {

array3[k] = array1[i];

k++;

}

}

}

}

int SumOfMinMax(char array3[], int k) {

int res;

char minel = array3[0], maxel = array3[0];

for (int i = 1; i < k; i++) {

if (array3[i] < minel) {

minel = array3[i];

}

if (array3[i] > maxel) {

maxel = array3[i];

}

}

res = minel + maxel;

return res;

}

void OutputArray(char array[], int size) {

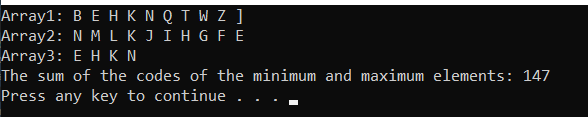
for (int i = 0; i < size; i++) {

cout << array[i] << " ";

}

cout << endl;

}



**IV. Висновки.**

Було досліджено методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та було набуто практичних навичок їх використання під час складання програмних специфікацій.

Для задачі використав 3 функції для:

1. Генерації перших двох масивів;

2. Генерації третього масиву з рівних елементів перших двох;

3.Знаходженнясуми кодів мінімального та максимального елементів.