# 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 № 1 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійного пошуку в послідовностях»

Варіант 19

Виконав студент ІП-11 Лисенко Андрій Юрійович

(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова Оксана Петрівна

( 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.

**Індивідуальне завдання**

**Варіант 19**

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.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Вираз для обчислення елемента | | Знайти |
| 19 | *2 \* i + 22* | *48 - 2 \* i* | Суму кодів між максимальним та мінімальним  елементами |

**Постановка задачі**

Ініціалізуємо масиви 12 та 2 за допомогою арифметичного циклу у відповідності до заданих формул. Перевіримо усі можливі комбінації значень з масивів 1 та 2, знайдемо рівні з них і запишемо у масив 3. Послідовно перевіряючи відношення між наступним та попереднім значеннями знайдемо мінімальне та максимальне значення масиву 3.

**Побудова математичної моделі**

Складемо таблицю змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Масив 1, Масив 2, Масив 3 | *char* | *arr1, arr2, arr3* | Зберігати дані про значення змінних індексованого типу |
| Кількість символьних значень масиву 3 | *int* | *len\_arr3* | Зберігати дані про кількість символьних значень масиву 3 |
| Лічильник *i, j* | *int* | *i, j* | Зберігати дані про ітерацію циклу |
| Тимчасовий лічильник | *int* | *i\_temp* | Зберігати дані про ітерацію циклу |
| Чи належить масиву 2 | *int* | *is\_in\_arr2* | Зберігати дані про наявність значень у масиві 2 |
| Максимальний та мінімальний елемент масиву 3 | *char* | *min, max* | Зберігати дані про максимальний та мінімальний елемент масиву 3 |

Складемо таблицю підпрограм:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Підпрограма | Тип | Ім’я | Параметри | Призначення |
| Ініціювання масиву 1 та 2 | *void* | *init\_arr12()* | *arr1[], arr2[]* | Ініціювати масив 1 та 2 значення заданими послідовностями |
| Ініціювання масиву 3 | *int* | *init\_arr3()* | *arr1[], arr2[], arr3[]* | Ініціювати масив 3 рівними значення масивів 1 та 2 |
| Перегляд | *void* | *browse()* | *arr[], n, len* | Переглянути значення масиву |
| Розв'язок | *void* | *solution()* | *arr[], len* | Знайти суму мінімального та максимального значення |

**Розв’язання**

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

***Крок 1:*** Визначимо основні дії

***Крок 2:*** Деталізуємо дію ініціалізування масивів 1 та 2

***Крок 3:*** Деталізуємо дію ініціалізування масиву 3

***Крок 4:*** Деталізування дію знаходження суми мінімального та максимального

елемента

**Псевдокод**

**Основна програма:**

**Початок**

*init\_arr12(arr1, arr2);*

*browse (arr1, 1, 10);*

*browse(arr2, 2, 10);*

*len\_arr3 = init\_arr3(arr1, arr2, arr3);*

*browse(arr3, 3, len\_arr3);*

*solution(arr3, len\_arr3);*

**Кінець**

**Підпрограма *init\_arr12:***

**Початок *init\_arr12***

**Повторити для *i*****від 0 до 10:**

*arr1[i] = 2 \* i + 22;*

*arr2[i] = 48 - 2 \* i;*

**Все повторити**

**Кінець *init\_arr12***

**Підпрограма *init\_arr3:***

**Початок *init\_arr3***

*i\_temp = 0;*

**Повторити для *i*****від 0 до 10:**

*j = 0, is\_in\_arr2 = 0;*

**Повторити поки** *j < 10 && !is\_in\_arr2:*

**Якщо** *arr1[i] == arr2[j]* **то:**

*arr3[i\_temp] = arr1[i];*

*is\_in\_arr2 = 1;*

*i\_temp++;*

**Все якщо**

*j++;*

**Все повторити**

**Все повторити**

**Повернути** *i\_temp;*

**Кінець *init\_arr3***

**Підпрограма *browse:***

**Початок *browse***

**Повторити для *i* від 0 до 10:**

**Виведення** *arr[i];*

**Все повторити**

**Кінець *browse***

**Підпрограма *solution:***

**Початок *solution***

*min = arr[0], max = arr[0];*

**Повторити для *i* від 0 до 10:**

**Якщо** *arr[i] > arr[i - 1]* **то:**

*max = arr[i];*

**Якщо** *arr[i] < arr[i - 1]* **то:**

*min = arr[i];*

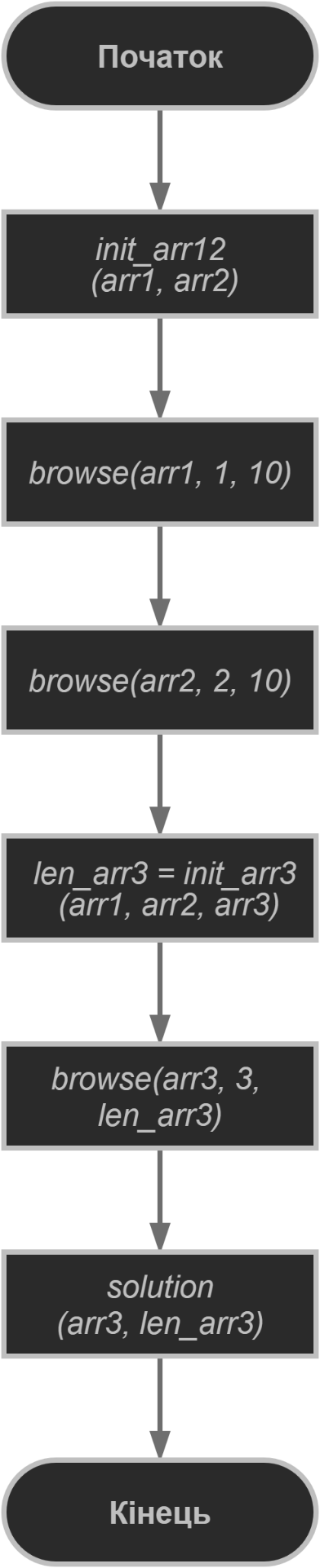
**Все повторити**

**Виведення** *min, max, min + max;*

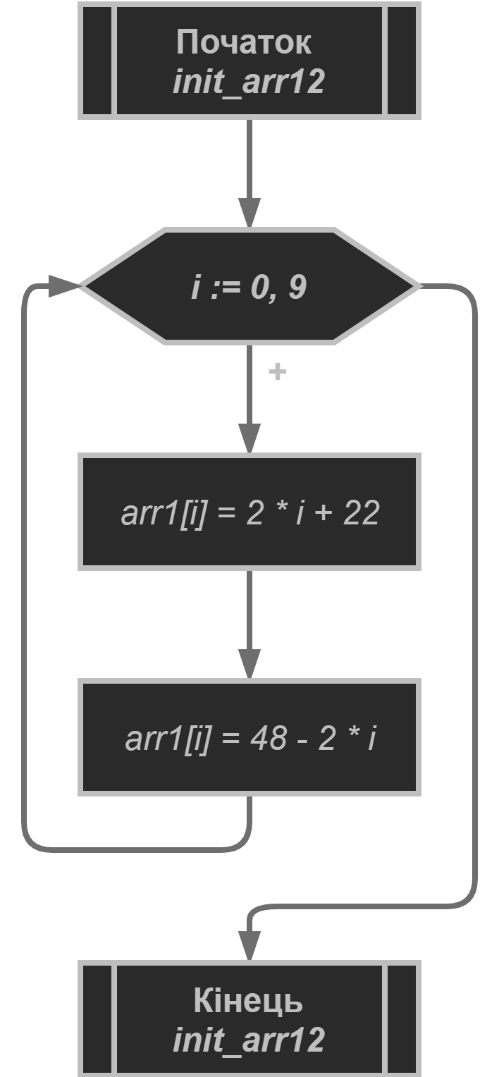
**Кінець *solution***

**Блок-схема**

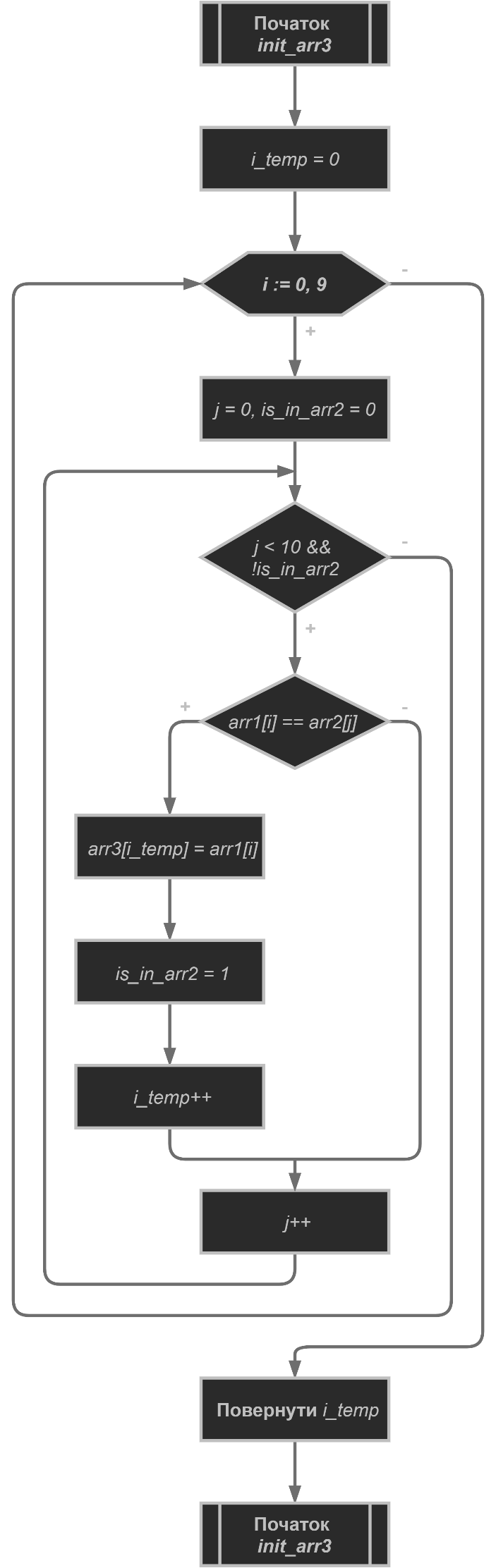
**Основна програма:**

****

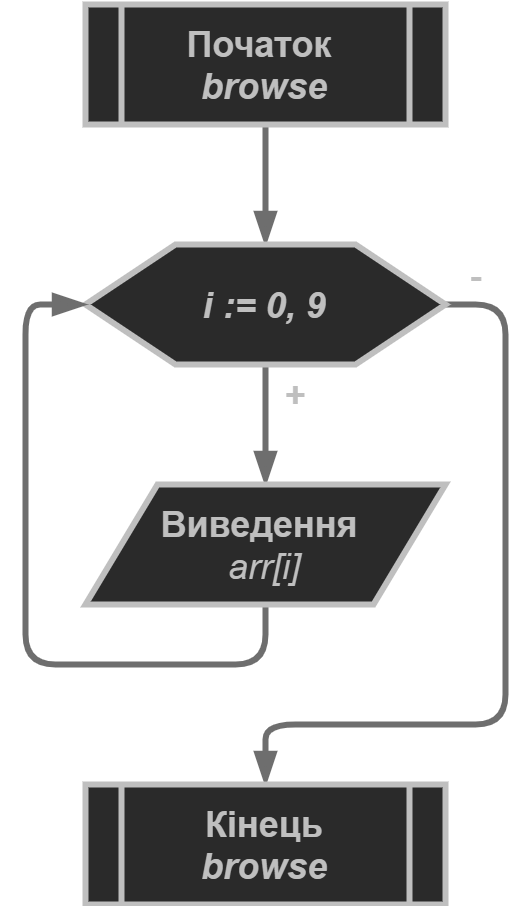
**Підпрограма *init\_arr12:***

****

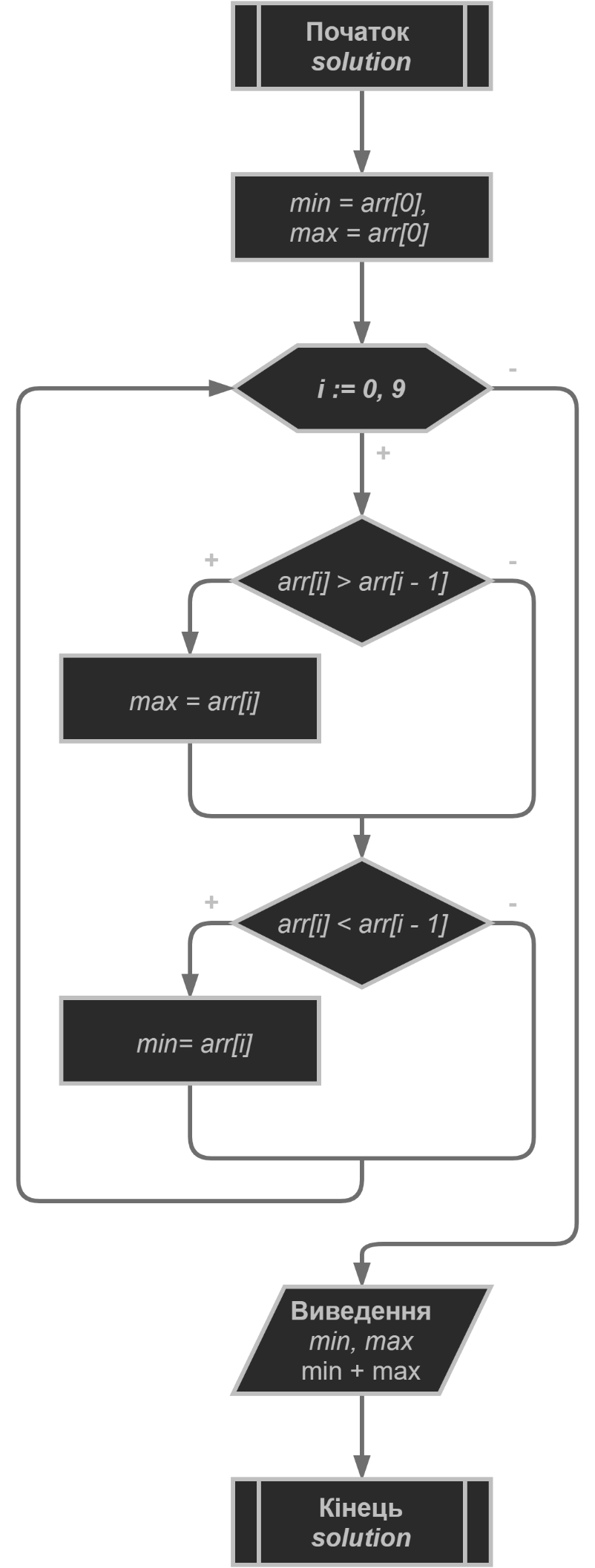
**Підпрограма *init\_arr3:***

****

**Підпрограма *browse:***

****

**Підпрограма *solution:***

****

**Код програми на C++**

#include <stdlib.h>

#include <stdio.h>

#include <iostream>

using namespace std;

void init\_arr12(char[], char[]);

int init\_arr3(char[], char[], char[]);

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

void solution(char[], int);

int main() {

int len\_arr3;

char arr1[10], arr2[10], arr3[10];

init\_arr12(arr1, arr2);

browse(arr1, 1, 10); browse(arr2, 2, 10);

len\_arr3 = init\_arr3(arr1, arr2, arr3);

browse(arr3, 3, len\_arr3);

solution(arr3, len\_arr3);

system("PAUSE>0");

return 0;

}

void init\_arr12(char arr1[], char arr2[]) {

for (int i = 0; i < 10; i++) {

arr1[i] = 2 \* i + 22;

}

for (int i = 0; i < 10; i++) {

arr2[i] = 48 - 2 \* i;

}

}

int init\_arr3(char arr1[], char arr2[], char arr3[]) {

int i\_temp = 0;

for (int i = 0; i < 10; i++) {

int j = 0, is\_in\_arr2 = 0;

while (j < 10 && !is\_in\_arr2) {

if (arr1[i] == arr2[j]) {

arr3[i\_temp] = arr1[i];

is\_in\_arr2 = 1;

i\_temp++;

}

j++;

}

}

return i\_temp;

}

void browse(char arr[], int n, int len) {

cout << "Array " << n << ": [";

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

printf("%2c", arr[i]);

}

cout << " ]" << endl;

}

void solution(char arr[], int len) {

char min = arr[0], max = arr[0];

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

if (arr[i] > arr[i - 1]) {

max = arr[i];

}

if (arr[i] < arr[i - 1]) {

min = arr[i];

}

}

cout << "The min code is: " << min << " (" << int(min) << ")" << endl;

cout << "The max code is: " << max << " (" << int(max) << ")" << endl;

cout << "The sum of the min and max codes is: " << int(min) + int(max);

}

**Перевірка**

|  |  |
| --- | --- |
| Блок | Дія |
| 0 | Початок |
| 1 | Array 1: [ ▬ ↑ → ∟ ▲ " $ & ( ] |
| 2 | Array 2: [ 0 . , \* ( & $ " ▲ ] |
| 3 | Array 3: [ ▲ " $ & ( ] (▲ " $ & ( належать обом масивам) |
| 4 | min = ▲(30) |
| 5 | max = ( (40) |
| 6 | min + max = 70 |
| 7 | кінець |

**Висновок**

Ми використали лінійний пошук для знаходження рівних значень серед двох масивів та використали їх для знаходження значень масиву 3. Також ми знайшли мінімальний та максимальний елемент цього масиву. Отже, ми дослідили методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набули практичних навичок їх використання під час складання програмних специфікацій.