# 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.

Основи алгоритмізації»

Варіант 13

Виконав студент ІП-11 Калашніков Андрій Євгенович

(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова О.П.

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота №7**

**Дослідження лінійного пошуку в послідовностях**

**Мета –** ослідити методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набути практичних навичок їх

використання під час складання програмних специфіка

**Індивідуальне завдання:**

13. 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 третьоїзмінноїрівними значеннями двох попередніх змінних.

4. Обробки третьої змінної згідно з варіантом.



**Постановка задачі**

Потрібно описати 3 змінні індексованого типу з 10 символьних значень, тобто

три масиви символьного типу , які містять 10 елементів. Потім, використовуючи

арифметичні цикли, заповнити 2 з них відповідно до умови. Третій масив

треба заповнити спільними елементами для 1 та 2 масивів, тобто ті, що мають

однаковий код. У третьому масиві треба знайти елементи з максимальним та

мінімальним кодом та знайти їх суму.

**Побудова математичної моделі**

Складемо таблицю змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Перший масив | Символьний | arr1 | Початкове дане |
| Другий масив | Символьний | arr2 | Початкове дане |
| Третій масив | Символьний | arr3 | Проміжне дане |
| Мінімальний код | Цілий | min\_code | Проміжне дане |
| Максимальний код | Цілий | max\_code | Проміжне дане |
| Сума мінімального та максимального кодів | Цілий | sum | Результат |
| Змінна для пошуку максимального / мінімального коду | Цілий | temp | Проміжне дане |
| Індекс в третьому масиві | Цілий | k | Проміжне дане |

**Розв’язання**

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді

блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо заповнення перших двох масивів.

Крок 3. Деталізуємо виведення масиву.

Крок 4. Деталізуємо заповнення третього масиву.

Крок 5. Деталізуємо знаходження мінімального коду третього масиву.

Крок 6. Деталізуємо знаходження максимального коду третього масиву.

**Псевдокод**

Основна програма

**Початок**

n:= 10

arr1[n], arr2[n], arr3[n]

fill\_arrays(arr1, arr2, n)

**виведення**“Array 1:\n”

display\_array(arr1, n)

**виведення**“Array 2:\n”

display\_array(arr2, n)

fill\_third\_array(arr1, arr2, arr3, n)

**виведення**“Array 3:\n”

display\_array(arr3, n)

min\_code := find\_min\_code(arr3, n)

**виведення**“Min code is “,min\_code,“\n”

max\_code := find\_max\_code(arr3, n)

**виведення** “Max code is “, max\_code,“\n”

sum := min\_code + max\_code

**виведення** “Sum of code of minimal and maximum elements is ”, sum

**Кінець**

Підпрограми

**Початок**

**fill\_arrays(arr1, arr2, length)**

**повторити для i від 0** до length

arr1[i] := 66 + 3 \* i

arr2[i] := 78 –i

**все повторити**

**Кінець fill\_arrays**

**display\_array(arr, length)**

**повторити для i від 0** до length

**виведення** arr[i], “ “

**все повторити**

**виведення** “\n”

**Кінець display\_array**

**fill\_third\_array(arr1,arr2,arr3, length)**

k := 0

**повторити для i від 0 до** length

**повторити для j від 0** **до** length

**якщо** arr1[i] == arr2[j]

**то**

arr3[k] := arr1[i]

k+= 1

**все якщо**

**все повторити**

**все повторити**

**Кінець fill\_third\_array**

**min\_code(arr, length)**

i:= 1

temp := arr[0]

**поки** arr[i] != 0іi < length **повторити**

**якщо** arr[i] <temp

**то**

temp:= arr[i]

**все якщо**

i+= 1

**все повторити**

**повернути temp**

**Кінець min\_code**

**max\_code(arr, length)**

i:= 1

temp := arr[0]

**поки** arr[i] != 0 і i < length **повторити**

**якщо** arr[i] > temp

**то**

temp:= arr[i]

**все якщо**

i+= 1

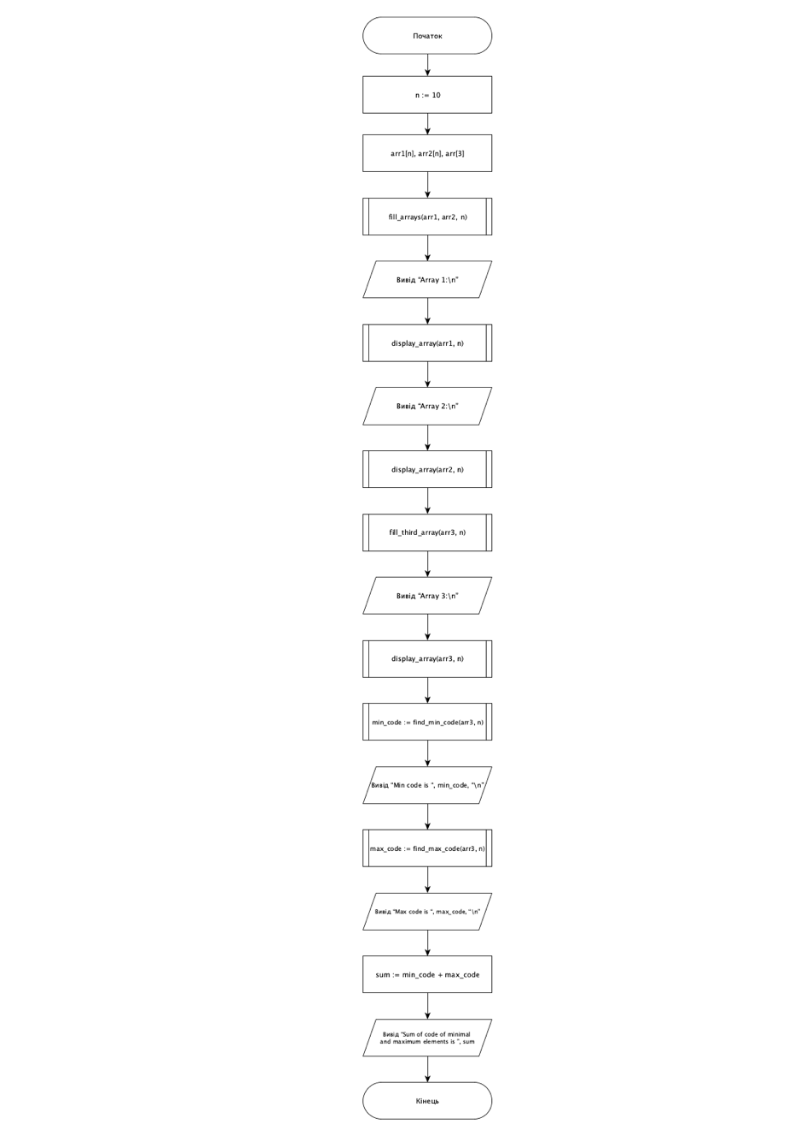
**все повторити**

**повернути** temp

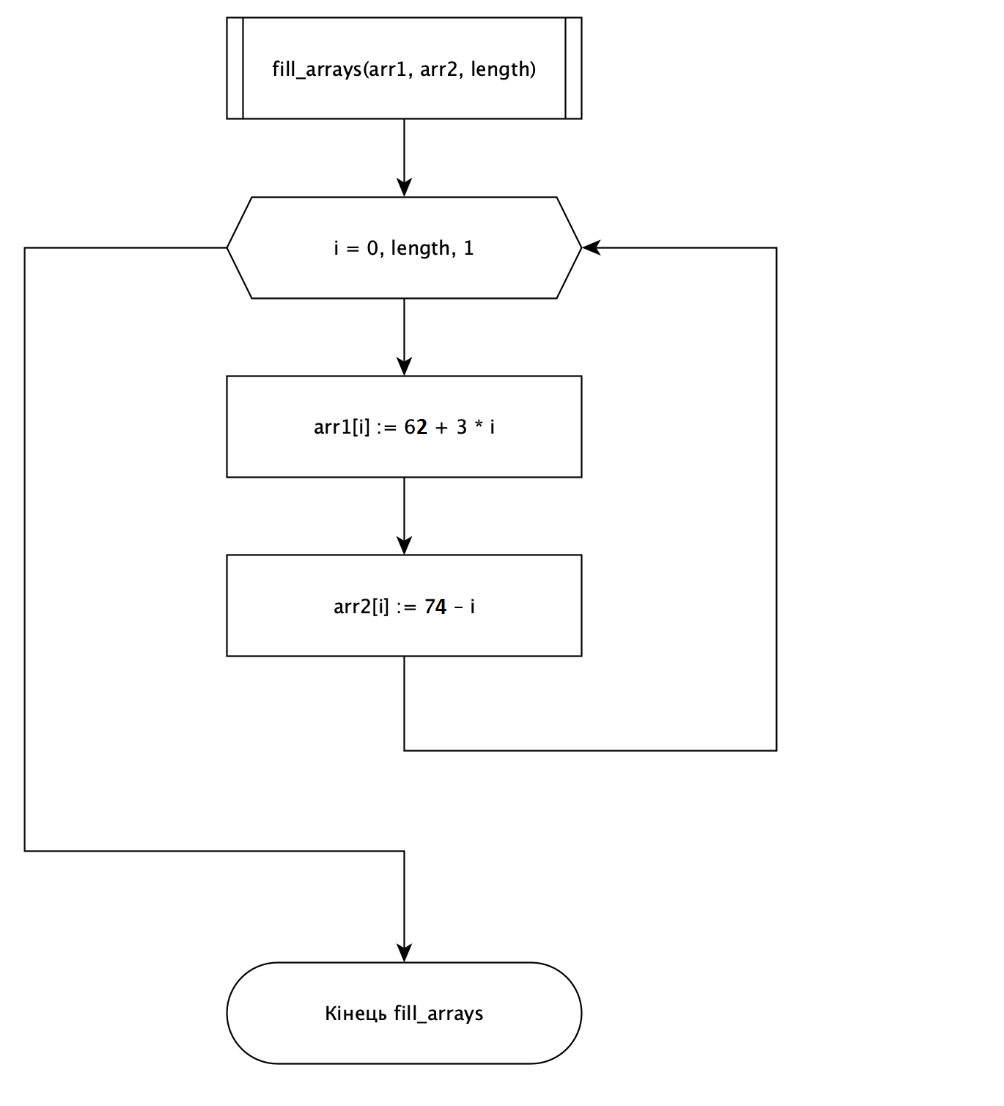
**Кінець max\_code**

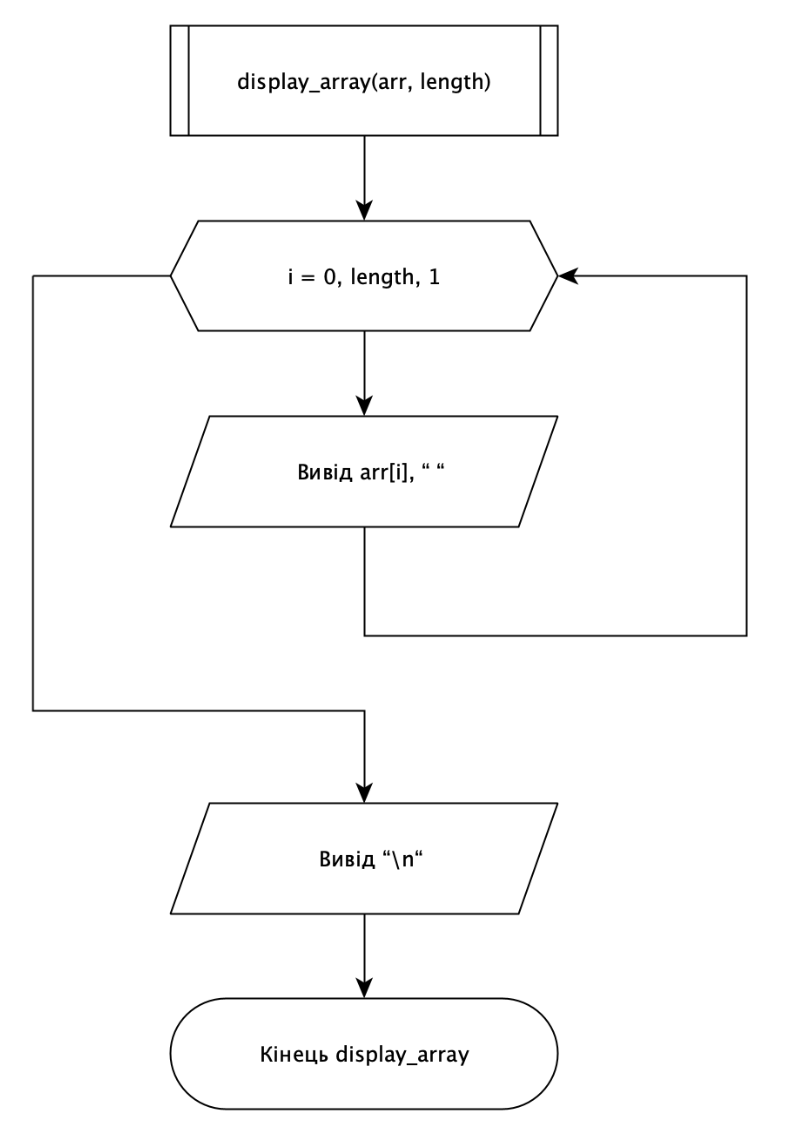
**Блок-схема**

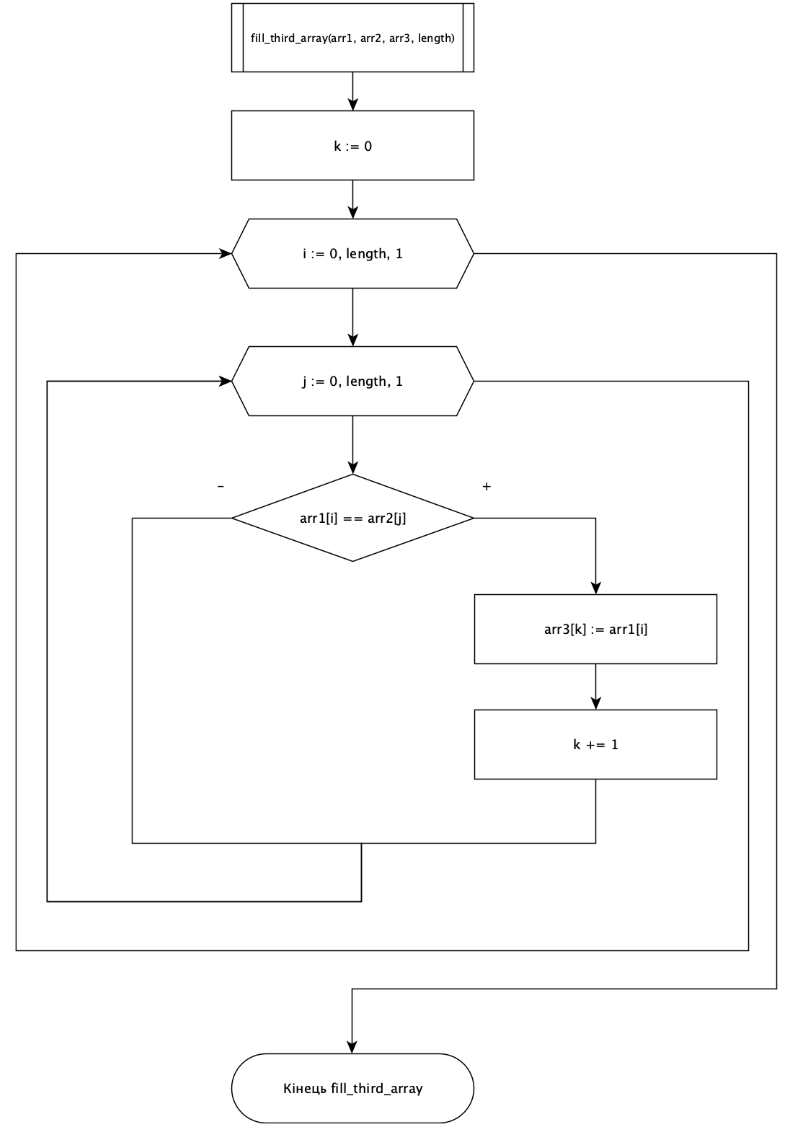
**1)**

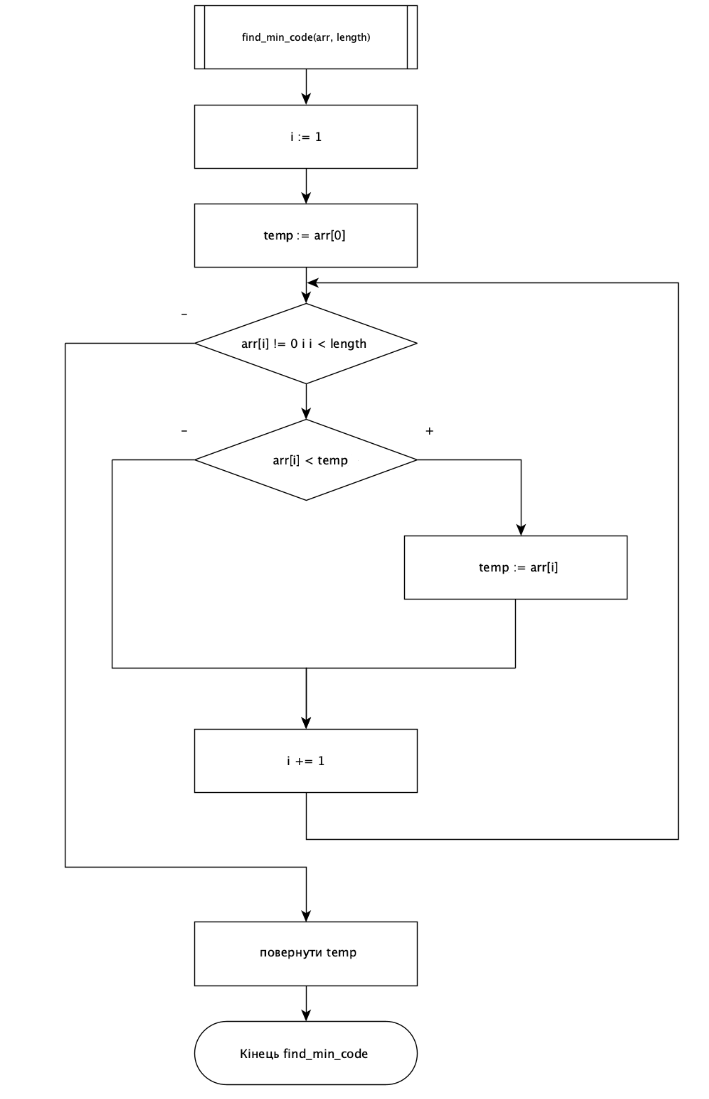
****

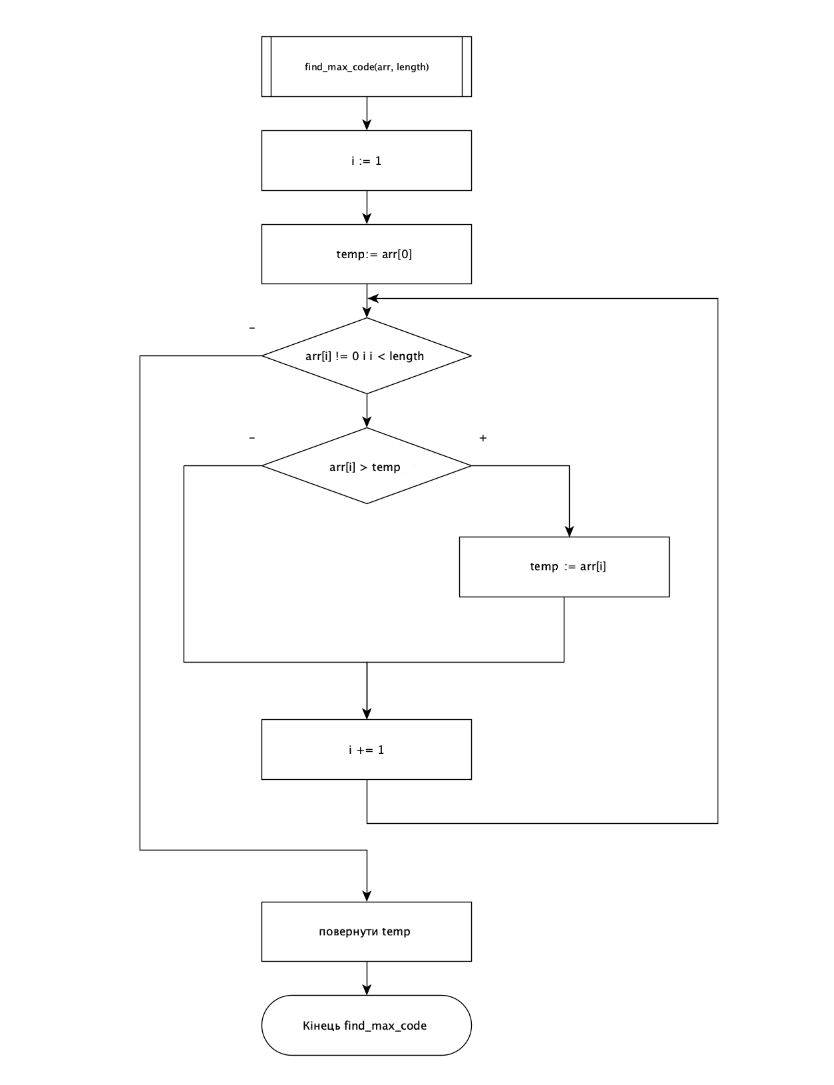
**2)**

****

****

****

****

****

**Код на С++ :**

#include <iostream>

using namespace std;

void fill\_arrays(char[], char[], int);

void display\_array(char[], int);

int fill\_third\_array(char[], char[], char[], int);

int find\_max\_code(char[], int);

int find\_min\_code(char[], int);

int main() {

const int n = 10;

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

int sum, k, min\_code, max\_code;

fill\_arrays(arr1, arr2, n);

cout << "Array 1:\n";

display\_array(arr1, n);

cout << "Array 2:\n";

display\_array(arr2, n);

k=fill\_third\_array(arr1, arr2, arr3, n);

cout << "Array 3:\n";

display\_array(arr3, n);

min\_code = find\_min\_code(arr3, k);

cout << "Min code is " << min\_code << '\n';

max\_code = find\_max\_code(arr3, k);

cout << "Max code is " << max\_code << '\n';

sum = min\_code + max\_code;

cout << "Sum of codes of minimal and maximum elements is " << sum;

return 0;

}

void fill\_arrays(char arr1[], char arr2[], int length) {

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

arr1[i] = char(62 + 3 \* i);

arr2[i] = char(74 - i);

}

}

void display\_array(char arr[], int length) {

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

cout << arr[i] << " ";

}

cout << '\n';

}

int fill\_third\_array(char arr1[], char arr2[], char arr3[], int length) {

int k = 0;

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

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

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

arr3[k] = arr1[i];

k++;

}

}

}

return k;

}

int find\_min\_code(char arr[], int k) {

int i = 1;

int temp = int(arr[0]);

while (int(arr[i]) != 0 && i < k) {

if (int(arr[i]) < temp) {

temp = int(arr[i]);

}

i++;

}

return temp;

}

int find\_max\_code(char arr[], int k) {

int i = 1;

int temp = int(arr[0]);

while (int(arr[i]) != 0 && i < k) {

if (int(arr[i]) > temp) {

temp = int(arr[i]);

}

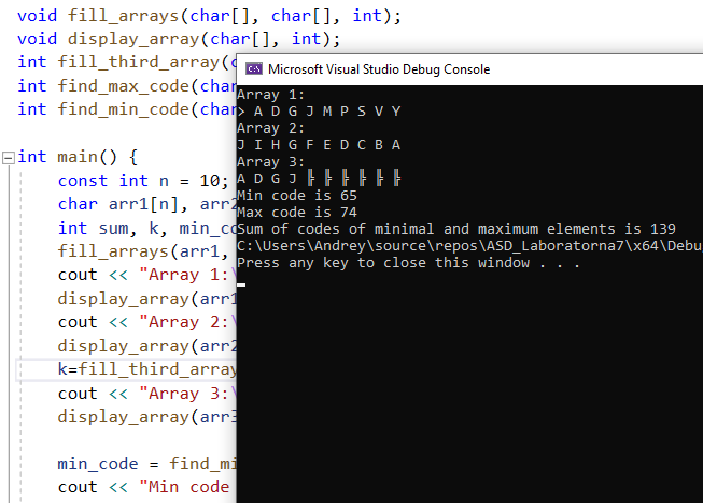
i++;

}

return temp;

}

**Скріншот результатів C++ :**

****

**Випробування**

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | n = 10 |
| 2 | Виведення: Array 1:  >A D G J M P S V Y |
| 3 | Виведення: Array 2:  J I H G F E D C B A |
| 4 | Виведення: Array 3:  A D G J |
| 5 | Виведення: Min\_code is 65 |
| 6 | Виведення: Max\_code is 74 |
| 7 | Виведення: Sum of code of minimal and maximum elements is 139 |
|  | Кінець |

**Висновок**

Під час лабораторної роботи я дослідив особливості роботи методів послідовного пошуку в послідовностях та набув практичних навичок їх використання, створивши алгоритм для заповнення двох масивів за заданою умовою, пошуку спільних елементів для цих масивів та пошуку найменшого та найбільшого елемента в третьому масиві, заповненого цими спільними елементами. Порахувавши їх суму, я отримав коректний результат, отже алгоритм працює правильно.