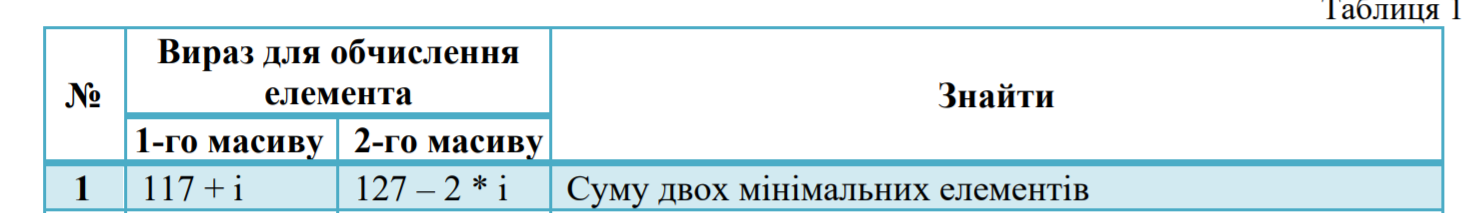
**Лабораторна робота 7**

**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.

Постановка задачі



Знайти мінімальні елементи масивів за допомогою порівняння послідовних елементів

**Побудова мат моделі**

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім'я | Призначення |
| Вхідна змінна | Символьний | mas1 | Вхідні дані |
| Вхідна змінна | Символьний | mas2 | Вхідні дані |
| Додаткова змінна | Цілий | n | Розмірність масиву |
| Додаткова змінна | Символьний | Mas3 | Об’єднання двох масивів |
| Додаткова змінна | Цілий | i | Лічільник |
| Функція | Символьний | input | Ініціалізація даних в масивах |
| Функція | Символьний | min | Знаходження мінімального елементу масива |
| Вихідна змінна | Символьний | Min\_1 | Перший мінімальний елемент |
| Вихідна змінна | Символьний | Min\_2 | Другий мінімальний елемент |

Крок 1 Ініціалізація масивів

Крок 2 Ініціалізація мінімальних елементів

Крок 3 Ініціалізація функції знаходження мінімального значення та суми

**Псевдокод**

**Початок**

Ініціалізація масивів

Ініціалізація функції знаходження мінімального значення

Ініціалізація та вивід суми

**Кінець**

Крок 1

input(char mas1[], char mas2[],n)

Початок

Для i Від 0 до n з кроком 1

Mas1[i]=117+i

Mas2[i]=127-2\*i

Все повторити

Кінець

**Початок**

input(char mas1[], char mas2[])

Ініціалізація функції знаходження мінімального значення

Ініціалізація та вивід суми

**Кінець**

Крок 2

**Початок**

input(char mas1[], char mas2[])

min\_1 = mas1[0]

min\_2= mas1[0]

Ініціалізація та вивід суми

**Кінець**

**Крок 3**

**min**(mas,mas1,mas3,min\_1,min\_2,n)

**Початок**

**Для i Від 0 до n з кроком 1**

**Для j Від 0 до n з кроком 1**

**Якщо** mas[i]=mas1[j]

**То** mas3[i]=mas[i]

**Все якщо**

**Все повторити**

**Все повторити**

**Index = 0**

**Для i Від 0 до n з кроком 1**

**Якщо** min\_1>mas3[i] та mas3[i]!=0

**То** min\_1=mas3[i]

**Index = i**

**Все якщо**

**Все повторити**

**Mas[index]=127**

**Для i Від 0 до n з кроком 1**

**Якщо** min\_2>mas3[i] та mas3[i]!=0

**То** min\_2=mas3[i]

**Все якщо**

**Все повторити**

**Вивести min\_1 , min\_2, min\_1+min\_2**

**Кінець**

**Початок**

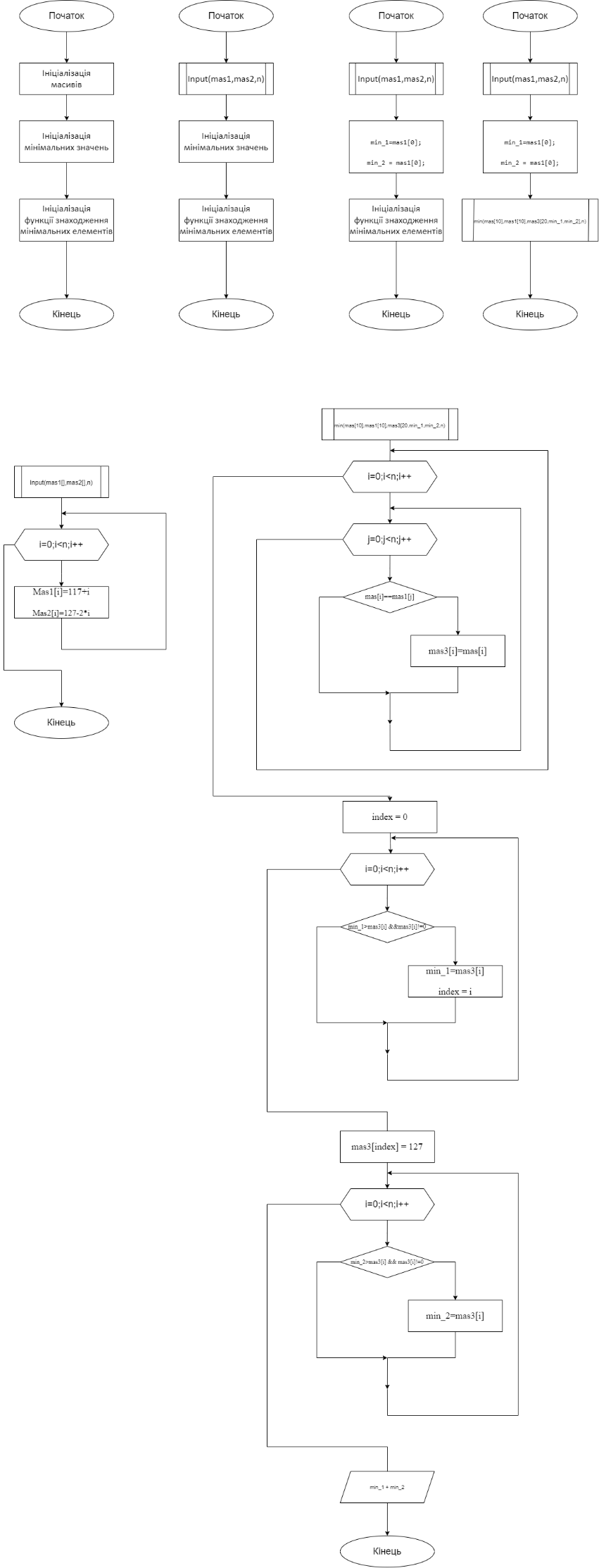
input(mas1[], mas2[],n)

min\_1 = mas1[0]

min\_2= mas1[0]

**min**(mas,mas1,mas3,min\_1,min\_2,n)

**Кінець**

**Блок-схема**

**Код програми**

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

using namespace std;

void min(char mas[], char mas1[], char mas3[], char min\_1, char min\_2, int n);

void input(char mas1[], char mas2[], int n);

void output(char mas[], int n);

int main()

{

int n = 10;

char\* mas1 = new char[n];

char\* mas2 = new char[n];

char\* mas3 = new char[n];

input(mas1, mas2,n);

char min\_1 = mas2[0];

char min\_2 = mas2[0];

output(mas1, n);

output(mas2, n);

min(mas1, mas2, mas3, min\_1, min\_2, n);

output(mas3, n);

}

void min(char mas[], char mas1[], char mas3[], char min\_1, char min\_2, int n) {

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

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

if (mas[i] == mas1[j])

mas3[i] = mas[i];

}

}

int index = 0;

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

if (int(min\_1) > int(mas3[i]) && int(mas3[i]) != 0) {

min\_1 = mas3[i];

index = i;

}

}

mas3[index] = 127;

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

if (int(min\_2) > int(mas3[i]) && int(mas3[i]) != 0) {

min\_2 = mas3[i];

}

}

cout << "Min elements - " << min\_1 << " , " << min\_2 << endl;

cout << int(min\_1) << " + " << int(min\_2) << " = " << int(min\_1) + int(min\_2) << endl;

}

void output(char mas[], int n) {

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

cout << mas[i] << " ";

}

cout << endl;

}

void input(char mas1[], char mas2[], int n) {

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

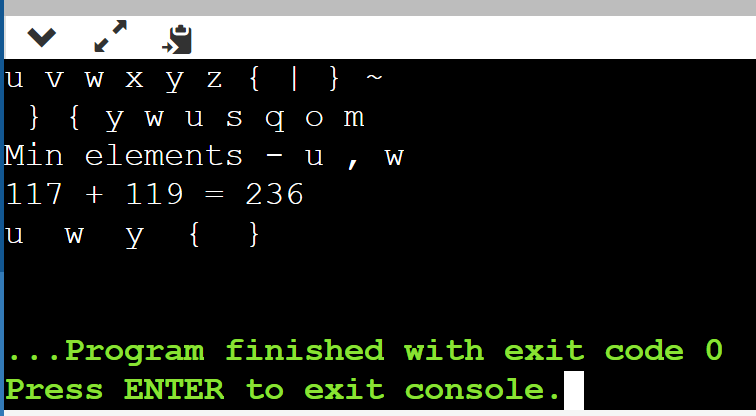
mas1[i] = 117 + i;

mas2[i] = 127 - 2 \* i;

}

}

**Випробування**



**Висновок**

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

Специфікою даного завдання був послідовний пошук мінімального значення символьного масиву.