

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота № 6

з дисципліни «Алгоритмізація та програмування»

Тема: "Реалізація алгоритмів обробки одновимірних масивів мовою C ++"

XAI.301.141.319a. 27 ЛР

Виконав студент гр. 319a

Владислав РУДНЄВ

(Підпис, дата)

(П.І.Б.)

Перевірів к.т.н., доц.

(вчена ступінь, вчене звання)

Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

2024

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення одновимірних і масивів на мові C ++ і реалізувати декларацію, введення з консолі, обробку і виведення в консоль одновимірних масивів на мові C ++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на аналіз і виведення елементів одновимірного масиву. Розмір масиву і його елементи ввести з консолі. Спершу вивести весь масив у рядок в порядку зростання індексів, потім – елементи чи підраховані результати відповідно до завдання.

Array7. Дан масив розміру N. Вивести його елементи в зворотному порядку.

Завдання 2. Вирішити завдання на перетворення одновимірного масиву. Розмір масиву і його елементи ввести з консолі. Спершу вивести у консоль заданий масив, потім – змінений.

Array72. Дано масив A розміру N і цілі числа K і L ($1 \leq K < L \leq N$). Переставити в зворотному порядку елементи масиву, розташовані між елементами AK і AL, включаючи ці елементи.

Завдання 3. У функції main() організувати багаторазовий вибір одного з двох завдань. Кожне завдання описати окремою функцією без параметрів. Введення, виведення, обробку масивів реалізувати окремими функціями з параметрами.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Array7.

Вхідні дані:

1. Масив чисел:

- Ім'я змінної: arr
- Опис: Одновимірний масив цілих чисел
- Тип: `vector<int>`
- Обмеження: $n > 0$

2. Розмір масиву:

- Ім'я змінної: n
- Опис: Кількість елементів у масиві
- Тип: `int`
- Обмеження: $n > 0$

Вихідні дані:

1. Початковий масив:

- Ім'я змінної: arr
- Опис: Масив, який користувач ввів
- Тип: `vector<int>`

2. Масив у зворотному порядку:

- Ім'я змінної: `reversedArr`
- Опис: Масив після виведення у зворотному порядку
- Тип: `vector<int>`

Алгоритм:

1. Введення даних: Користувач вводить розмір масиву, якщо число менше нуля видати помилку, користувач вводить n масиву

2. Обробка даних: Виводиться початковий масив у введеному порядку. Масив виводиться у зворотному порядку (за допомогою циклу `for`).

3. Виведення результату: Відображається оригінальний масив а потім у зворотному напрямку.

Завдання 2.

Вирішення задачі Array72.

Вхідні дані:

1. Масив чисел:

- Ім'я змінної: arr
- Опис: Одновимірний масив цілих чисел
- Тип: `vector<int>`
- Обмеження: $n > 0$

2. Розмір масиву:

- Ім'я змінної: n
- Опис: Кількість елементів у масиві
- Тип: `int`
- Обмеження: $n > 0$

3. Індеси для перестановки підмасиву:

- Ім'я змінних: k, l
- Опис: Початковий (k) і кінцевий (l) індеси підмасиву
- Тип: `int`
- Обмеження: $1 \leq k < l \leq n$

Вихідні дані:

1. Початковий масив:

- Ім'я змінної: arr
- Опис: Масив, який користувач ввів
- Тип: `vector<int>`

2. Модифікований масив:

- Ім'я змінної: modifiedArr
- Опис: Масив після перестановки підмасиву
- Тип: `vector<int>`

3. Повідомлення про помилки:

- Ім'я змінної: errorMsg

- Опис: Текстове повідомлення, якщо користувач ввів некоректні значення k і l
- Тип: string (виводиться через cout).

Алгоритм:

1.Введення даних: Користувач вводить масиву n , якщо масив менше нуля видати помилку, користувач вводить кількість елементів масиву, та вводить початковий та кінцевий індекс масиву, якщо k та l , $k < l$, $l > n$, або $k \geq l$ видати помилку.

2.Перестановка підмасиву: Викликається функція reverseSubarray() яка змінює порядок елементів у підмасиви.

3.Виведення результату: Відображається початковий масив а потім масив після переставки.

Лістинг коду дод. А (стр.6 — 8)

Скрін-шоти вікна виконання програми дод. Б (сторінка 9 — 11)

ВИСНОВОК

Вивчив теоретичний матеріал з основ представлення одновимірних і масивів на мові C ++ і реалізувати декларацію, введення з консолі, обробку і виведення в консоль одновимірних масивів на мові C ++ в середовищі Visual Studio.Отримані знання можуть бути корисними для подальшого вивчення структур даних, а також при розробці більш складних алгоритмів роботи з масивами.

ДОДАТОК А

Лістинінг коду

```

#include <iostream>
#include <vector>
using namespace std;

// Прототипи функцій
void task1(); // Завдання 1: Виведення масиву у зворотному порядку
void task2(); // Завдання 2: Перестановка підмасиву у зворотному порядку
void inputArray(vector<int>& arr, int& n); // Функція для введення масиву
void outputArray(const vector<int>& arr); // Функція для виведення масиву
void reverseSubarray(vector<int>& arr, int k, int l); // Функція для
перестановки підмасиву

int main() {
    while (true) {
        // Меню вибору завдання
        cout << "Choose a task (1 or 2). Enter 0 to exit: ";
        int choice;
        cin >> choice;

        // Перевірка вибору користувача
        if (choice == 0) {
            cout << "Exiting the program." << endl;
            break; // Завершення програми
        }

        // Виконання вибраного завдання
        switch (choice) {
            case 1:
                task1(); // Виконати завдання 1
                break;
            case 2:
                task2(); // Виконати завдання 2
                break;
            default:
                cout << "Invalid choice! Please choose 1, 2, or 0 to exit." <<
endl;
        }
    }
    return 0; // Успішне завершення програми
}

// Завдання 1: Виведення масиву у зворотному порядку
void task1() {
    int n;
    vector<int> arr;

```

```

// Виклик функції для введення масиву
inputArray(arr, n);

// Виведення початкового масиву
cout << "Original array:" << endl;
outputArray(arr);

// Виведення масиву у зворотному порядку
cout << "Array in reverse order:" << endl;
for (int i = n - 1; i >= 0; --i) {
    cout << arr[i] << " "; // Виведення елемента
}
cout << endl;
}

// Завдання 2: Перестановка підмасиву у зворотному порядку
void task2() {
    int n, k, l;
    vector<int> arr;

    // Виклик функції для введення масиву
    inputArray(arr, n);

    // Виведення початкового масиву
    cout << "Original array:" << endl;
    outputArray(arr);

    // Введення значень K і L
    cout << "Enter K and L ( $1 \leq K < L \leq N$ ): ";
    cin >> k >> l;

    // Перевірка коректності введених значень
    if (k < 1 || l < 1 || k >= l || l > n) {
        cout << "Invalid values for K and L. Ensure that  $1 \leq K < L \leq N$ ." <<
endl;
        return; // Завершення функції у разі помилки
    }

    // Виклик функції для перестановки підмасиву
    reverseSubarray(arr, k, l);

    // Виведення зміненого масиву
    cout << "Modified array:" << endl;
    outputArray(arr);
}

// Функція для введення масиву
void inputArray(vector<int>& arr, int& n) {

```

```

cout << "Enter the size of the array (N): ";
cin >> n; // Введення розміру масиву

// Перевірка розміру масиву
if (n <= 0) {
    cout << "The size of the array must be greater than 0!" << endl;
    exit(1); // Завершення програми у разі помилки
}

// Зміна розміру вектора
arr.resize(n);
cout << "Enter " << n << " elements of the array:" << endl;
for (int i = 0; i < n; ++i) {
    cin >> arr[i]; // Введення елементів масиву
}
}

// Функція для виведення масиву
void outputArray(const vector<int>& arr) {
    for (int elem : arr) { // Ітерація по всіх елементах масиву
        cout << elem << " "; // Виведення елемента
    }
    cout << endl;
}

// Функція для перестановки підмасиву у зворотному порядку
void reverseSubarray(vector<int>& arr, int k, int l) {
    // Приведення K і L до індексів масиву
    int start = k - 1; // Початковий індекс
    int end = l - 1;   // Кінцевий індекс

    // Перестановка елементів
    while (start < end) {
        swap(arr[start], arr[end]); // Обмін місцями елементів
        ++start; // Зміщення початкового індексу
        --end;   // Зміщення кінцевого індексу
    }
}

```


ДОДАТОК Б

Скрін-шоти вікна виконання програми та діаграми

На рис.1 показано код виконання програми Лабораторна робота №6

```
Choose a task (1 or 2). Enter 0 to exit: 1
Enter the size of the array (N): 5
Enter 5 elements of the array:
1 2 3 4 5
Original array:
1 2 3 4 5
Array in reverse order:
5 4 3 2 1
Choose a task (1 or 2). Enter 0 to exit: 2
Enter the size of the array (N): 3
Enter 3 elements of the array:
1 2 3
Original array:
1 2 3
Enter K and L (1 ? K < L ? N): 2 3
Modified array:
1 3 2
Choose a task (1 or 2). Enter 0 to exit: 0
Exiting the program.
```

Рисунок 1 — Лабораторна робота №6

На рис.2 показано діаграму коду програми для завдання 1 та 2

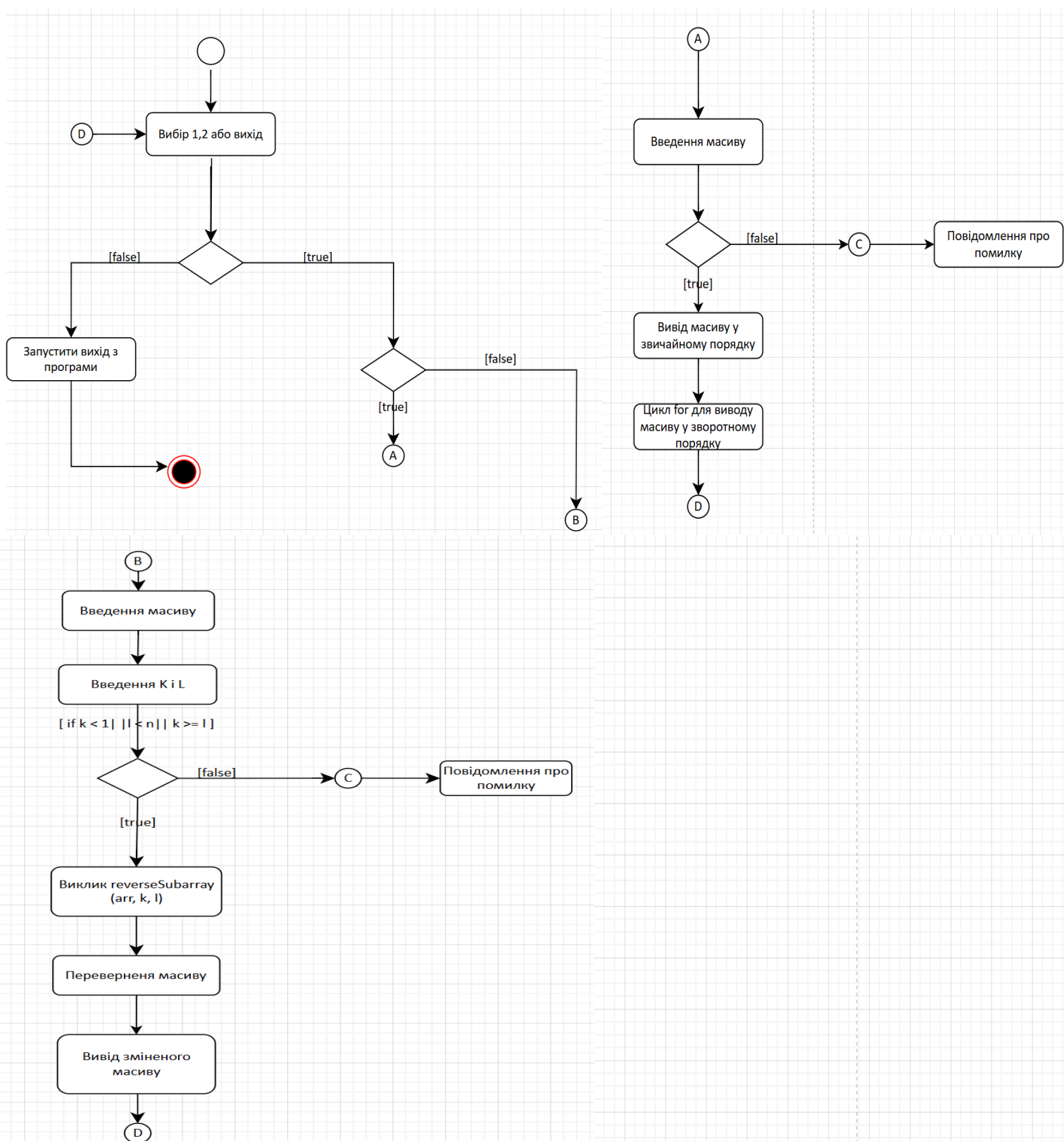


Рисунок 2 — Діаграма активності коду Array7 та Array72

На рис.3 показано діаграму коду програми для завдання 3

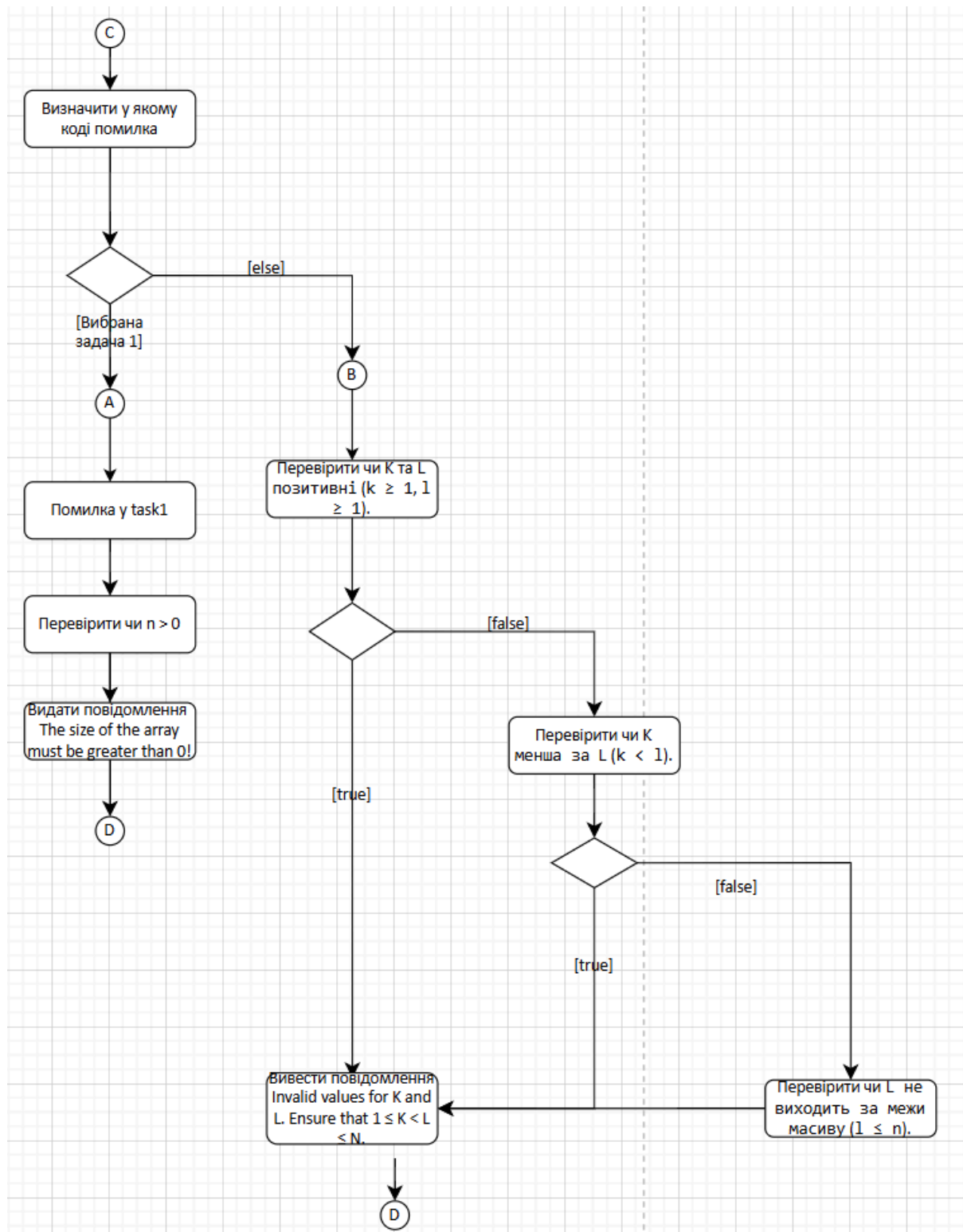


Рисунок 3 — Діаграма активності коду для перевірки помилок в кодї