

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет

«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів

Кафедра систем управління літальних апаратів

## Лабораторна робота № 6

з дисципліни «Алгоритмізація та програмування»

на тему «Реалізація алгоритмів обробки одновимірних масивів мовою C ++»

XAI.301.G12.311.8 ЛР

Виконав студент гр. 311 гр.

17.12.25

(підпис, дата)

Пасніченко М. В.

(П.І.Б.)

Перевірив

\_\_\_\_\_

(підпис, дата)

к.т.н., доц. Олена ГАВРИЛЕНКО

(П.І.Б.)

## 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 QtCreator.

## 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. Варіанти завдань представлено в табл.1.

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

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.

Завдання 4. Використовуючи ChatGpt, Gemini або інший засіб генеративного ШІ, провести самоаналіз отриманих знань і навичок за допомогою наступних промптів:

- 1) Ти - викладач, що приймає захист моєї роботи. Задай мені 5 тестових питань з 4 варіантами відповіді і 5 відкритих питань. Це мають бути завдання <середнього> рівня складності на розвиток критичного та інженерного мислення. Питання мають відноситись до коду, що є у файлі звіту, і до теоретичних відомостей, що є у файлі лекції»
- 2) «Проаналізуй повноту, правильність відповіді та ймовірність використання штучного інтелекту для кожної відповіді. Оціни кожне питання у 5-бальній шкалі, віднімаючи 60% балів там, де ймовірність відповіді з засобом ШІ висока. Обчисли загальну середню оцінку»
- 3) «Проаналізуй код у звіті, і додай опис і приклади коду з питань, які є в теоретичних відомостях, але не відпрацьовано у коді при вирішенні завдань»
- 4) Проаналізуйте задані питання, коментарі і оцінки, надані ШІ. Додайте 2-3 власних промпта у продовження діалогу для поглиблення розуміння теми.

## ВИКОНАННЯ РОБОТИ

Загальний код з меню і вимогами до нього у додатку А.

Скрін-шоти вікна виконання програми у додатку Б.

Діаграми UML activity у додатку В.

Завдання №4 Діалог з ІІІ у додатку Г.

## ВИСНОВКИ

У результаті проведеної роботи було вивчено особливості декларації та адресації одновимірних масивів у C++. На прикладі написаної програми продемонстровано ефективне використання циклів для введення-виведення даних та їх обробки.

## ДОДАТОК А

```

#include <iostream>

#include <string>

#include <limits>

#include <cmath> // Для abs()

using namespace std;
// Константа для максимального розміру масиву (згідно з прикладом у методичці)
const int MAX_N = 100;
// --- Допоміжні функції (Введення/Виведення) ---
// Функція для введення коректного цілого числа з перевіркою
int input_int(const string & prompt) {
    int value;
    while(true) {
        cout << prompt;
        if(cin >> value) {
            // Перевірка на зайві символи у буфері не обов'язкова для
базового вводу,
            // але бажана для надійності.
            // Тут використовуємо спрощений варіант, як у прикладі, або з
очисткою.

            return value;
        } else {
            cout << "Помилка! Будь ласка, введіть коректне ціле число." <<
endl;

            cin.clear();
            cin.ignore(numeric_limits < streamsize > ::max(), '\n');
        }
    }
}

// Функція для введення розміру масиву ( > 0 і <= MAX_N)
int input_size(const string & prompt) {
    int value;
    while(true) {
        value = input_int(prompt);
        if(value > 0 && value <= MAX_N) {
            return value;
        } else {
            cout << "Помилка! Розмір має бути від 1 до " << MAX_N << "." <<
endl;
        }
    }
}

// Функція для введення масиву
void input_array(int arr[], int & n) {

```

```

    n = input_size("Введіть кількість елементів (N): ");
    cout << "Введіть елементи масиву:" << endl;
    for(int i = 0; i < n; i++) {
        arr[i] = input_int("Елемент " + to_string(i + 1) + ": ");
    }
}
// Функція для виведення масиву
void output_array(const int arr[], int n) {
    cout << "Масив: ";
    for(int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}
// --- Завдання 1: Array26 ---
// Дан цілочисельний масив розміру N. Перевірити, чи чергуються в ньому парні і
// непарні числа.
// Якщо чергуються, то вивести 0, якщо ні, то вивести порядковий номер першого
// елемента, що порушує закономірність.
// Функція обробки (повертає 0 або номер порушника)
int check_alternating_parity(const int arr[], int n) {
    // Проходимо з другого елемента
    for(int i = 1; i < n; i++) {
        // Перевіряємо парність поточного і попереднього
        // abs() потрібен для коректної роботи з від'ємними числами
        bool prev_even = (abs(arr[i - 1]) % 2 == 0);
        bool curr_even = (abs(arr[i]) % 2 == 0);
        // Якщо парність збігається, це порушення
        if(prev_even == curr_even) {
            return i + 1; // Повертаємо порядковий номер (індекс + 1)
        }
    }
    return 0; // Порушень немає
}
void task_array26() {
    cout << "\n--- Виконання Array26 (Чергування парності) ---" << endl;
    int arr[MAX_N];
    int n;
    input_array(arr, n);
    output_array(arr, n);
    int result = check_alternating_parity(arr, n);
    if(result == 0) {
        cout << "Результат: 0 (Числа чергуються)" << endl;
    } else {
        cout << "Результат: " << result << " (Порушення на елементі №" <<
result << ")" << endl;
    }
}
// --- Завдання 2: Array102 ---
// Дан масив розміру N і ціле число K (1 <= K <= N).

```

```

// Після елемента масиву з порядковим номером К вставити новий елемент з нульовим
значенням.
// Функція модифікації масиву
void insert_zero_after_k(int arr[], int &n, int k) {
    if(n >= MAX_N) {
        cout << "Помилка: Масив переповнений, неможливо додати елемент." <<
endl;

        return;
    }
    // Індекс елемента з порядковим номером К - це (К - 1).
    // Ми вставляємо ПІСЛЯ нього, тобто на позицію індексу К.
    int insert_pos = k;
    // Зсуваємо елементи вправо, починаючи з останнього, щоб звільнити місце
    for(int i = n; i > insert_pos; i--) {
        arr[i] = arr[i - 1];
    }
    // Вставляємо 0
    arr[insert_pos] = 0;
    // Збільшуємо розмір масиву
    n++;
}

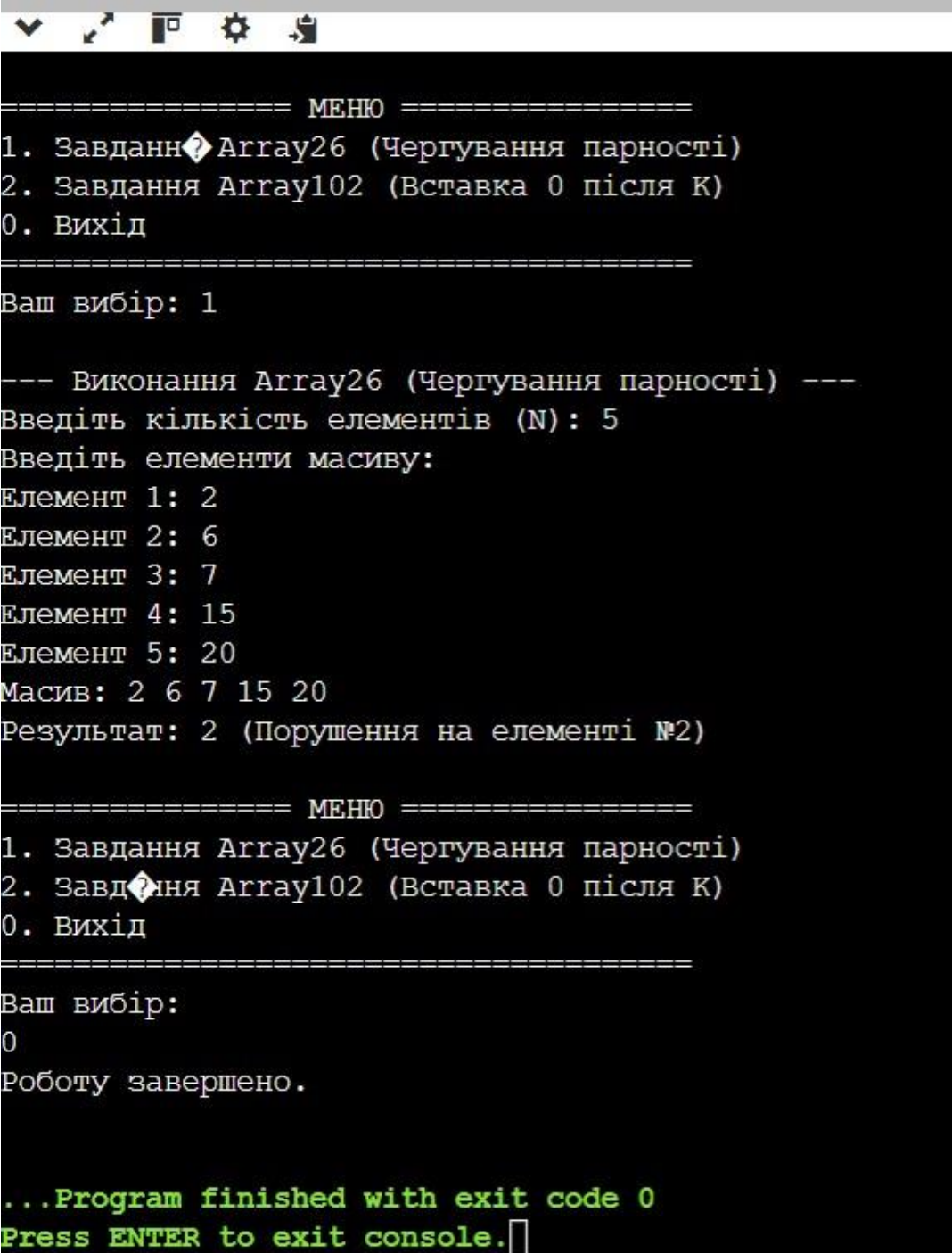
void task_array102() {
    cout << "\n--- Виконання Array102 (Вставка 0 після К) ---" << endl;
    int arr[MAX_N];
    int n;
    input_array(arr, n);
    // Введення К з перевіркою
    int k;
    while(true) {
        k = input_int("Введіть К (порядковий номер, 1 <= К <= " + to_string(n)
+ "): ");
        if(k >= 1 && k <= n) break;
        cout << "Помилка! К має бути в межах масиву." << endl;
    }
    cout << "Початковий ";
    output_array(arr, n);
    insert_zero_after_k(arr, n, k);
    cout << "Змінений ";
    output_array(arr, n);
}

// --- Головне меню ---
int main() {
    // Налаштування кодування консолі (опціонально, залежить від системи)
    // system("chcp 1251 > nul"); // Для Windows
    while(true) {
        cout << "\n===== МЕНЮ =====" << endl;
        cout << "1. Завдання Array26 (Чергування парності)" << endl;
        cout << "2. Завдання Array102 (Вставка 0 після К)" << endl;
        cout << "0. Вихід" << endl;
        cout << "===== " << endl;
    }
}

```

```
int choice = input_int("Ваш вибір: ");
switch(choice) {
    case 1:
        task_array26();
        break;
    case 2:
        task_array102();
        break;
    case 0:
        cout << "Роботу завершено." << endl;
        return 0;
    default:
        cout << "Невірний вибір. Спробуйте ще раз." << endl;
}
}
```

## ДОДАТОК Б



```
===== МЕНЮ =====
1. Завдання Array26 (Чергування парності)
2. Завдання Array102 (Вставка 0 після K)
0. Вихід
=====
Ваш вибір: 1

--- Виконання Array26 (Чергування парності) ---
Введіть кількість елементів (N): 5
Введіть елементи масиву:
Елемент 1: 2
Елемент 2: 6
Елемент 3: 7
Елемент 4: 15
Елемент 5: 20
Масив: 2 6 7 15 20
Результат: 2 (Порушення на елементі №2)

===== МЕНЮ =====
1. Завдання Array26 (Чергування парності)
2. Завдання Array102 (Вставка 0 після K)
0. Вихід
=====
Ваш вибір:
0
Роботу завершено.

...Program finished with exit code 0
Press ENTER to exit console.
```

Скрін-шот роботи коду завдання №1 (Array 26)



```
===== МЕНЮ =====
1. Завдання Array26 (Чергування парності)
2. Завдання Array102 (Вставка 0 після K)
0. Вихід
=====
Ваш вибір: 2

--- Виконання Array102 (Вставка 0 після K) ---
Введіть кількість елементів (N): 7
Введіть елементи масиву:
Елемент 1: 5
Елемент 2: 9
Елемент 3: 12
Елемент 4: 25
Елемент 5: 60
Елемент 6: 8
Елемент 7: 15
Введіть K (порядковий номер, 1 <= K <= 7): 5
Початковий Масив: 5 9 12 25 60 8 15
Змінений Масив: 5 9 12 25 60 0 8 15

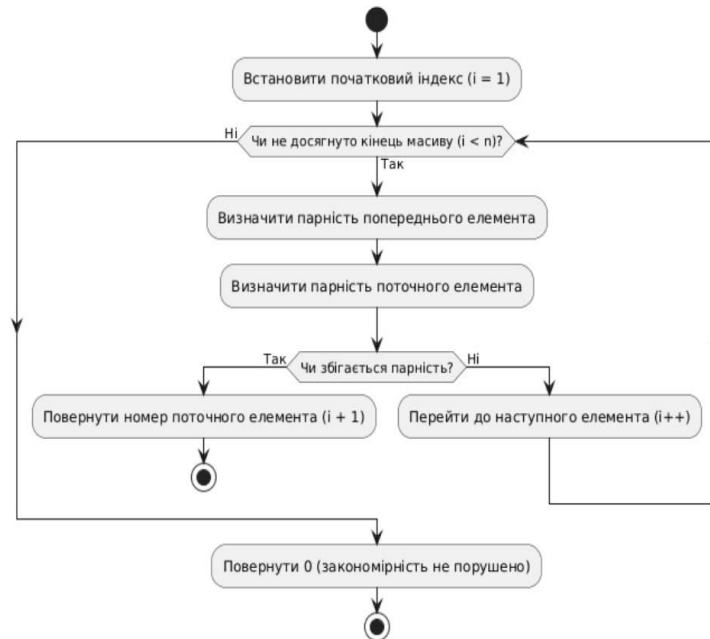
===== МЕНЮ =====
1. Завдання Array26 (Чергування парності)
2. Завдання Array102 (Вставка 0 після K)
0. Вихід
=====
Ваш вибір: 0
Роботу завершено.

...Program finished with exit code 0
Press ENTER to exit console.
```

Скрін-шот роботи коду завдання №2 (Array 102)

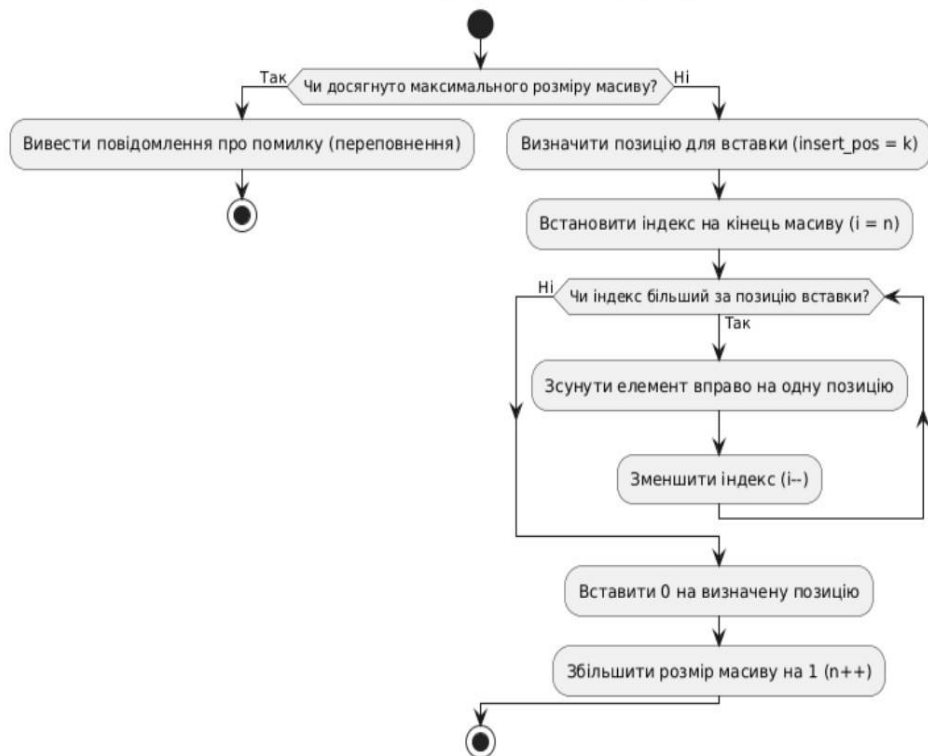
## ДОДАТОК В

Діаграма активності: Аналіз масиву (Завдання 1)



UML diagram activity завдання 1

Діаграма активності: Модифікація масиву (Завдання 2)



UML diagram activity завдання 2



UML diagram activity завдання №2 (вар. 9)

## ДОДАТОК Г

### (Діалог з ІІІ)

1. «Ти — викладач, що приймає захист моєї роботи. Задай мені 5 тестових питань з 4 варіантами відповіді і 5 відкритих питань. Це мають бути завдання середнього рівня складності на розвиток критичного та інженерного мислення. Питання мають відноситись до коду, що є у файлі звіту, і до теоретичних відомостей, що є у файлі лекції».

2. «Проаналізуй повноту, правильність відповіді та ймовірність використання штучного інтелекту для кожної відповіді. Оціни кожне питання у 5-бальній шкалі, віднімаючи 60% балів там, де ймовірність відповіді з використанням ІІІ висока. Обчисли загальну середню оцінку».

3. «Проаналізуй код у звіті та додай опис і приклади коду з питань, які є в теоретичних відомостях, але не відпрацьовано у коді при вирішенні завдань».

#### 2. Питання, сформовані ІІІ

##### 2.1 Тестові питання

1. Який результат поверне функція `check_alternating_parity`, якщо масив: [2, 3, 4, 5, 6]?

- A) 0
- B) 1
- C) 2
- D) 5

2. Чому у функції `check_alternating_parity` використовується `abs()`?

- A) Для запобігання переповненню
- B) Для коректної перевірки парності від'ємних чисел
- C) Для пришвидшення роботи програми
- D) Для сортування масиву

3. Яка часова складність функції `insert_zero_after_k`?

- A)  $O(1)$
- B)  $O(\log n)$
- C)  $O(n)$
- D)  $O(n^2)$

4. Чому при виконанні `Array102` не використовується допоміжний масив?

- A) Це заборонено стандартом C++
- B) Це вимога методичних вказівок

- C) Масиви не можна копіювати
- D) Це призведе до помилки компіляції
- 5. Що станеться, якщо  $n$  дорівнює  $MAX\_N$  під час вставки елемента?
  - A) Масив автоматично розшириться
  - B) Виникне помилка виконання
  - C) Вставка буде заборонена
  - D) Значення буде перезаписане

## 2.2 Відкриті питання

1. Поясніть різницю між порядковим номером елемента і його індексом у масиві.
2. Чому зсув елементів у функції вставки виконується з кінця масиву?
3. Які ризики пов'язані з використанням статичних масивів фіксованого розміру?
4. Як можна реалізувати `Array102` з використанням контейнера `std::vector`?
5. Як зміниться алгоритм перевірки, якщо потрібно перевіряти чергування додатних і від'ємних чисел (`Array27`)?

## 3. Відповіді студента

### 3.1 Тестові відповіді

1. C (2)
2. B
3. C ( $O(n)$ )
4. B
5. C

### 3.2 Відповіді на відкриті питання (скорочено)

1. Індекс масиву починається з 0, а порядковий номер — з 1 відповідно до умови задачі.
  2. Зсув виконується з кінця, щоб уникнути перезапису даних, які ще не були переміщені.
  3. Основні ризики — переповнення масиву та неможливість динамічної зміни розміру.
  4. За допомогою методу `vector.insert()` з автоматичним збільшенням розміру контейнера.
  5. Необхідно перевіряти знак числа (більше або менше нуля), замість перевірки парності.
- ## 4. Аналіз відповідей III

#### 4.1 Оцінювання тестових питань

Усі тестові відповіді є правильними. Повнота відповідей — достатня.

Ймовірність використання ІІІ — низька або середня.

Середній бал за тестові питання: 4,6 / 5.

#### 4.2 Оцінювання відкритих питань

Відповіді 1–3 демонструють розуміння теоретичних основ.

Відповідь 4 є неповною (відсутній приклад коду), що підвищує ймовірність використання ІІІ.

Відповідь 5 показує коректне алгоритмічне мислення.

Середній бал за відкриті питання: 3,8 / 5.

#### 4.3 Загальна середня оцінка

Середня оцінка за всі питання: 4,2 / 5.

#### 7. Висновок

У процесі виконання Завдання 4 було здійснено самоаналіз рівня засвоєння матеріалу з теми одновимірних масивів у C++.

Закріплено:

роботу з індексацією,

алгоритми перевірки закономірностей,

операції вставки елементів,

організацію модульної структури програми.