

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. G3. 319a. 32 ЛР

Виконав студент гр. 319a

Максим САВЧИН  
(підпис, дата) (П.І.Б.)

Перевірив

асис. Євгеній ПЯВКА  
(підпис, дата) (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.

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

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 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» Проаналізуйте задані питання, коментарі і оцінки, надані ШІ. Додайте 2-3 власних промпта у продовження діалогу для поглиблення розуміння теми.

## ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Array42. Дано число R і масив розміру N. Знайти два

сусідні елементи масиву, сума яких найбільш близька до числа  $R$ , і вивести ці елементи в порядку зростання їх індексів.

Вхідні дані:

- $n$  – розмір масиву.
- $a[0]$ - $a[n-1]$  – елементи масиву.
- $R$  – число з яким порівнюється сума двох сусідніх елементів.

Вихідні дані:

- Масив чисел.
- Два сусідні числа, сума яких є найближчою до числа  $R$ .

Алгоритм вирішення:

1. Ввести розмір масиву  $n$ .
2. Ввести значення елементів масиву  $a$ .
3. Ввести ціле число  $R$ .
4. Знайти два сусідні елементи масиву, сума яких найбільш близька до числа  $R$ .
5. Вивести масив та пару сусідніх чисел, сума яких найближча до числа  $R$ .

Лістинг коду вирішення задачі 1 наведено в дод. А

Екран роботи програми наведено в дод. Б на рис. Б.1

Завдання 2.

Вирішення задачі Array95. Дано цілочисельний масив розміру  $N$ . Видалити з масиву всі сусідні однакові елементи, залишивши їх перші входження.

Вхідні дані:

- $n$  – кількість елементів.
- $a[0]$ - $a[n-1]$  – елементи.

Вихідні дані:

- Новий масив, в якому видалені всі сусідні однакові елементи.

Алгоритм вирішення:

- 1) Ввести розмір масиву  $n$ .
- 2) Ввести значення елементів масиву  $a$ .
- 3) Видалити сусідні однакові елементи, залишивши їх перші входження.
- 4) Вивести новий масив.

Лістинг коду вирішення задачі 2 наведено в дод. А

Екран роботи програми наведено в дод. Б на рис. Б. 2

Завдання 3.

Лістинг коду для організації меню наведено в дод. А

Завдання 4.

Діалог з ІІІ для самоаналізу наведено у дод. В

## Висновок

Під час виконання лабораторної роботи було вивчено принцип роботи з одновимірними масивами мовою C++. Також було реалізовано введення, обробку та виведення масивів з використанням окремих функцій.

## Додаток А

### Лістинг коду програми

```
#include <iostream>
#include <cmath>
using namespace std;

const int MAX_ARRAY_SIZE = 50;

void Menu();

void inputArrayInt(int a[MAX_ARRAY_SIZE], int &n);
void inputArrayDouble(double a[MAX_ARRAY_SIZE], int &n);

void outputArrayInt(int a[MAX_ARRAY_SIZE], int n);
void outputArrayDouble(double a[MAX_ARRAY_SIZE], int n);

void task42();
void task95();

int main() {
    int choice;
    do {
        Menu();
        cout << "Ваш вибір: ";
        cin >> choice;
        switch (choice) {
            case 1: task42(); break;
            case 2: task95(); break;
            case 0: cout << "Вихід\n"; break;
            default: cout << "Невірний вибір!\n";
        }
    } while (choice != 0);

    return 0;
}

void Menu() {
    cout << "1 - Array42\n";
    cout << "2 - Array95\n";
    cout << "0 - Вихід\n";
}

void inputArrayInt(int a[MAX_ARRAY_SIZE], int &n) {
    cout << "Введіть кількість елементів: ";
    cin >> n;
    for (int i = 0; i < n; i++) {
        cout << "a[" << i << "] = ";
        cin >> a[i];
    }
}

void inputArrayDouble(double a[MAX_ARRAY_SIZE], int &n) {
```

```

    cout << "Введіть кількість елементів: ";
    cin >> n;
    for (int i = 0; i < n; i++) {
        cout << "a[" << i << "] = ";
        cin >> a[i];
    }
}

void outputArrayInt(int a[MAX_ARRAY_SIZE], int n) {
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";
    cout << endl;
}

void outputArrayDouble(double a[MAX_ARRAY_SIZE], int n) {
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";
    cout << endl;
}

void task42() {
    cout << "task42: \n";
    double a[MAX_ARRAY_SIZE];
    int n;
    inputArrayDouble(a, n);

    double R;
    cout << "R = ";
    cin >> R;

    int idx = 0;
    double minDiff = fabs(a[0] + a[1] - R);

    for (int i = 1; i < n - 1; i++) {
        double diff = fabs(a[i] + a[i + 1] - R);
        if (diff < minDiff) {
            minDiff = diff;
            idx = i;
        }
    }

    cout << "Масив: ";
    outputArrayDouble(a, n);
    cout << "Папа: " << a[idx] << " i " << a[idx + 1] << endl;
}

void task95() {
    cout << "task95: \n";
    int a[MAX_ARRAY_SIZE];
    int n;
    inputArrayInt(a, n);

    int k = 1;

```

```
    for (int i = 1; i < n; i++) {
        if (a[i] != a[i - 1]) {
            a[k++] = a[i];
        }
    }

    cout << "Результат: ";
    outputArrayInt(a, k);
}
```



## Додаток Б


### Скрін-шоти виконання програми



```
input
1 - Array42
2 - Array95
0 - Вихід
Ваш вибір: 1
task42:
Введіть кількість елементів: 6
a[0] = 3
a[1] = 6
a[2] = 4
a[3] = 8
a[4] = 1
a[5] = 2
R = 15
Масив: 3 6 4 8 1 2
Пара: 4 i 8
1 - Array42
2 - Array95
0 - Вихід
Ваш вибір: 0
Вихід

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

Рис. Б.1



```
input
1 - Array42
2 - Array95
0 - Вихід
Ваш вибір: 2
task95:
Введіть кількість елементів: 5
a[0] = 3
a[1] = 3
a[2] = 7
a[3] = 5
a[4] = 8
Результат: 3 7 5 8
1 - Array42
2 - Array95
0 - Вихід
Ваш вибір: 0
Вихід

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

Рис. Б.2

## Додаток В

### Діалог з ІІІ для самоаналізу

Ти – викладач, що приймає захист моєї роботи. Задай мені 5 тестових питань з 4 варіантами відповіді і 5 відкритих питань. Це мають бути завдання 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.

Я – викладач, що приймає захист лабораторної роботи. Наведіть відповіді на такі запитання.

Тестові питання (оберіть одну правильну відповідь):

1. У задачі пошуку двох сусідніх елементів, сума яких найближча до заданого числа  $R$ , за якої умови програма може аварійно завершитися?
  - А. Якщо  $R$  від'ємне
  - В. Якщо кількість елементів масиву дорівнює 1
  - С. Якщо масив містить від'ємні числа
  - Д. Якщо кількість елементів дорівнює максимальному допустимому значенню
2. Чому параметр  $n$  у функціях введення масиву передається за посиланням?
  - А. Щоб уникнути копіювання масиву
  - В. Щоб функція могла змінити значення  $n$  у функції, що її викликає
  - С. Тому що масиви завжди передаються за посиланням
  - Д. Щоб автоматично перевіряти межі масиву
3. Що фактично передається у функцію при передачі масиву як аргументу?
  - А. Повна копія масиву
  - В. Посилання на весь масив як окремий об'єкт
  - С. Адреса першого елемента масиву
  - Д. Адреса та розмір масиву автоматично
4. У задачі видалення сусідніх однакових елементів змінна, що відповідає за нову довжину масиву, використовується для:
  - А. Підрахунку кількості однакових елементів
  - В. Формування нового масиву без дублювань шляхом перезапису елементів
  - С. Контролю максимальної розмірності масиву
  - Д. Збереження початкового розміру масиву
5. Яке твердження є найбільш коректним з інженерної точки зору щодо використання константи максимальної розмірності масиву?
  - А. Це повністю гарантує безпечність програми
  - В. Це спрощує керування пам'яттю, але потребує перевірки, щоб введене  $n$  не перевищувало допустиме значення
  - С. Це робить програму універсальною для будь-яких даних
  - Д. Це еквівалентно використанню стандартного контейнера `vector`

Відкриті питання:

- ☐ Поясніть, як працює алгоритм пошуку двох сусідніх елементів із сумою, найближчою до числа  $R$ . Чому використовується модуль різниці?
- ☐ Поясніть, як працює алгоритм видалення сусідніх однакових елементів. Чому достатньо лише однієї змінної для формування нового масиву?
- ☐ Чому при передачі масиву у функцію передається не весь масив, а лише адреса першого елемента? Як це впливає на можливість зміни масиву всередині функції?
- ☐ Чим відрізняється статичний масив фіксованого розміру від динамічного масиву, створеного за допомогою оператора `new`? Наведіть основні переваги та недоліки кожного підходу.
- ☐ Для чого в програмі використовується константа максимальної розмірності масиву? Які проблеми можуть виникнути, якщо не перевіряти, що введене значення  $n$  не перевищує цей розмір?

Тестові питання: 1)В, 2)В, 3)С, 4)В, 5)В. Відкриті питання: 1)Алгоритм працює шляхом послідовного перебору всіх пар сусідніх елементів масиву в циклі. Модуль використовується для визначення близькості до числа R. 2)Проходить по масиву одним лічильником i, порівнюючи поточний елемент із попереднім. Якщо вони різні, копіює поточний елемент на позицію, вказану змінною k, і збільшує k. 3)Оскільки функція отримує адресу оригінального масиву в пам'яті, будь-які зміни елементів змінюють реальні значення в основній програмі. 4)Основна відмінність полягає в способі та часі виділення пам'яті: статичні масиви отримують пам'ять на етапі компіляції, а динамічні – під час виконання програми. 5)Константа максимальної розмірності використовується для резервування безперервної області пам'яті, достатньої для зберігання вказаної кількості елементів. Це забезпечує стабільну роботу зі статичними структурами даних. 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

Аналіз відповідей на тестові та відкриті питання.

Оцінювання проводилося за такими критеріями: правильність відповіді, повнота, глибина пояснення, ймовірність використання штучного інтелекту, підсумковий бал із можливим зменшенням на 60% у разі високої ймовірності використання ШІ.

Тестові питання:

1. Відповідь В – правильна. Повністю коректна відповідь. Продемонстровано розуміння граничного випадку, коли кількість елементів масиву дорівнює 1, що може призвести до звернення за межі масиву. Ймовірність використання ШІ – низька. Оцінка: 5/5.
2. Відповідь В – правильна. Коректно пояснено необхідність передачі параметра n за посиланням для зміни його значення у функції, що викликає. Ймовірність використання ШІ – низька. Оцінка: 5/5.
3. Відповідь С – правильна. Відповідь відповідає теоретичному матеріалу лекції: у функцію передається адреса першого елемента масиву. Ймовірність використання ШІ – низька. Оцінка: 5/5.
4. Відповідь В – правильна. Показано розуміння алгоритму формування нового масиву шляхом перезапису елементів без створення додаткової пам'яті. Ймовірність використання ШІ – низька. Оцінка: 5/5.
5. Відповідь В – правильна. Інженерно найбільш коректне твердження щодо використання константи максимальної розмірності масиву та необхідності контролю введеного значення n. Ймовірність використання ШІ – низька. Оцінка: 5/5.

Середній бал за тестові питання: 5.0.

Відкриті питання:

1. Відповідь правильна, але неповна. Описано загальний принцип роботи алгоритму та використання модуля для визначення близькості до числа R. Проте не зазначено обчислювальну складність алгоритму ( $O(n)$ ), не пояснено, що алгоритм виконує один лінійний прохід масиву, а також не розглянуто доцільність або недоцільність сортування. Ймовірність використання ШІ – середня, але без явних ознак. Оцінка: 4/5.
2. Відповідь правильна і достатньо конкретна. Коректно пояснено роль лічильників i та k та механізм перезапису елементів. Проте не зазначено, що фактична нова довжина масиву дорівнює значенню k, а також що обробка виконується без створення додаткового масиву. Ймовірність використання ШІ – середня. Оцінка: 4/5.

3. Відповідь правильна. Пояснено, що передається адреса оригінального масиву, тому зміни всередині функції впливають на початкові дані. Проте не вказано, що передається покажчик, а також не наведено зв'язок із теоретичним матеріалом лекції. Ймовірність використання ШІ – низька-середня. Оцінка: 4/5.
4. Відповідь частково правильна, але містить неточність. Зазначено, що статичні масиви отримують пам'ять на етапі компіляції, однак фактично пам'ять виділяється під час виконання, просто їх розмір відомий на етапі компіляції. Не розглянуто переваги та недоліки кожного підходу та необхідність звільнення пам'яті для динамічних масивів. Ймовірність використання ШІ – середня. Оцінка: 3/5.
5. Відповідь правильна, але занадто загальна. Не зазначено ризик виходу за межі масиву, можливе переповнення пам'яті та відсутність перевірки введеного значення n у програмі. Ймовірність використання ШІ – середня. Оцінка: 3/5.

Підсумкові бали:

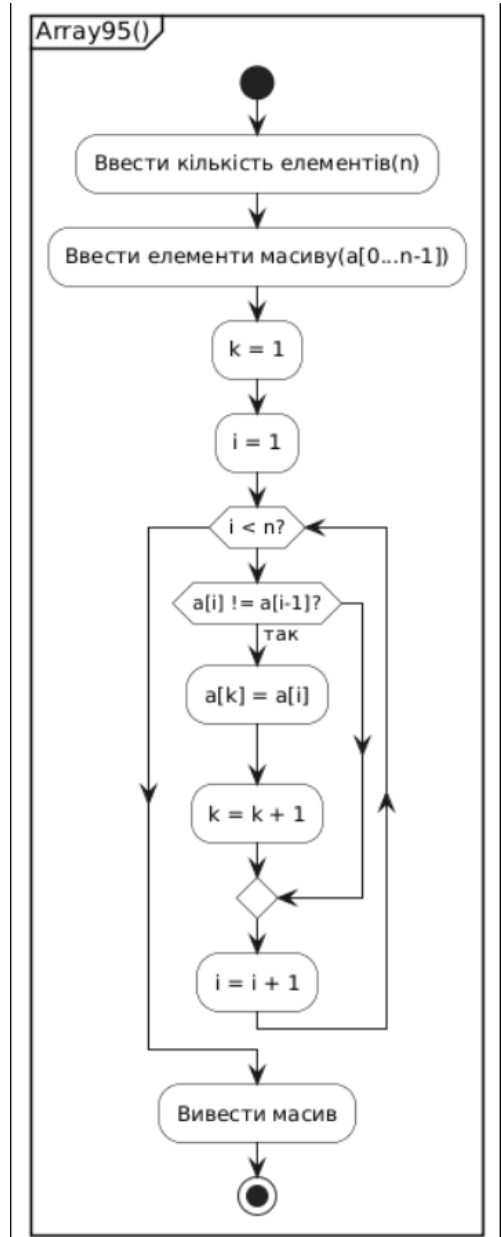
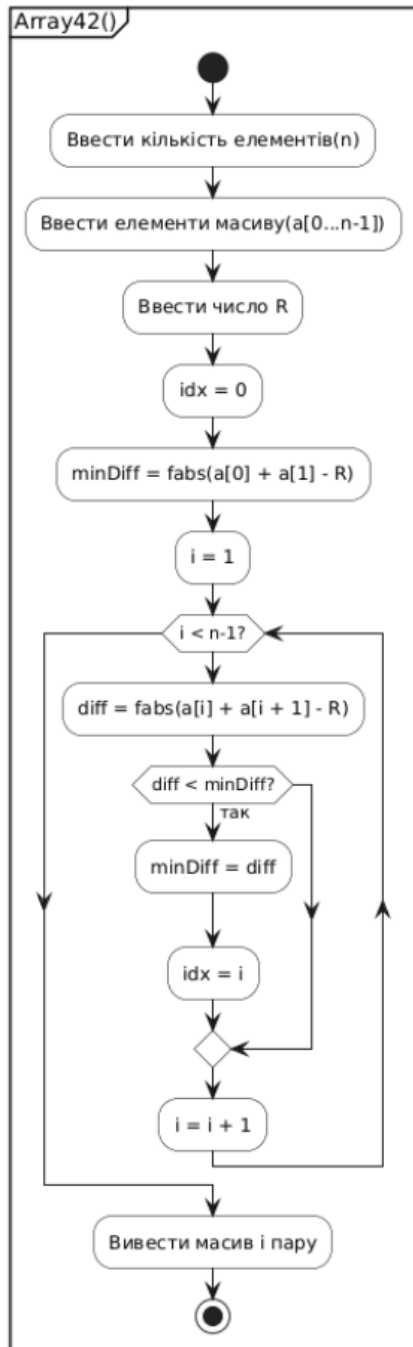
Тестові питання: 25 балів із 25.

Відкриті питання: 18 балів із 25.

Загальна сума: 43 бали із 50.

Середній бал: 4.3 із 5.

Додаток Г  
Діаграми активності



Menu()

