

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота № 4

з дисципліни «Алгоритмізація та програмування»
на тему «Структурування програм з використанням функцій»

ХАІ.301. G3 Електрична Інженерія 319а 18 ЛР

Виконав студент гр. 319а

Тучак Владислав
(підпис, дата) (П.І.Б.)

Перевірив

 к.т.н., доц. Олена ГАВРИЛЕНКО
(підпис, дата) (П.І.Б.)

МЕТА РОБОТИ

Вивчити теоретичний матеріал із синтаксису оголошення, визначення і виклику функцій в C++ і реалізувати консольний додаток з використанням функцій з параметрами і поверненням результату на мові програмування C++ в середовищі Qt Creator.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити дві задачі з реалізації функцій. Для кожної задачі описати функцію (декларація, визначення і виклик) відповідно до варіанту. У тілі і при виклику функцій не використовувати цикли. Формулювання Proc14 , Proc43 - табл.1

ВИКОНАННЯ РОБОТИ

Завдання 1

Задача: Proc14. Описати процедуру ShiftRight3(A, B, C), яка виконує правий циклічний зсув трьох дійсних чисел: значення A переходить у B, значення B — у C, а значення C — у A. За допомогою цієї процедури виконати циклічний зсув для двох наборів з трьох чисел.

Вхідні дані:

У програмі вводяться два набори дійсних чисел: A1, B1, C1 та A2, B2, C2. Всі значення є дійсними числами у діапазоні від -100 до 100.

Вихідні дані:

Результатом роботи є два набори чисел після циклічного правого зсуву. Після виконання процедури значення кожного набору змінюються відповідно до правила: A отримує попереднє значення C, B отримує попереднє значення A, а C отримує попереднє значення B.

Алгоритм вирішення:

Користувач вводить перший набір чисел A1, B1, C1. Користувач вводить другий набір чисел A2, B2, C2. Для кожного набору викликається процедура ShiftRight3, яка переставляє значення змінних по циклу вправо. Після виконання процедури виводяться змінені значення обох наборів.

Задача: Proc43.

Описати функцію IsTriangle(a, b, c), яка повертає True, якщо з трьох додатних чисел a, b, c можна утворити трикутник, і False — в іншому випадку. За допомогою цієї функції перевірити три набори чисел. Вхідні дані: У програмі вводяться три набори додатних дійсних чисел: (a1, b1, c1), (a2, b2, c2), (a3, b3, c3). Усі значення повинні бути додатними. Вихідні дані: Для кожного набору виводиться логічне значення True або False, залежно від того, чи можливо зі сторін утворити трикутник. Алгоритм вирішення: Користувач вводить три набори додатних чисел. Для кожного набору викликається функція IsTriangle. Функція перевіряє виконання умови існування трикутника: кожна сторона повинна бути меншою за суму двох інших. Якщо умова виконується, повертається True, інакше — False.

ВИСНОВКИ

У ході виконання лабораторної роботи було вивчено та закріплено принципи створення підпрограм у мові C++ — процедур і функцій з параметрами. На практиці реалізовано процедуру циклічного зсуву трьох чисел та функцію для перевірки можливості побудови трикутника за заданими сторонами. Отримано навички роботи з передаванням параметрів за посиланням і використанням логічних функцій. Труднощів під час виконання не виникло, усі завдання були реалізовані та протестовані.

ДОДАТОК А

Лістинг коду програми

```

#include <iostream>
#include <limits>

// ----- ОГОЛОШЕННЯ (перед main) -----

// Proc14: правий циклічний зсув трійки чисел (вхід/вихідні параметри)
void ShiftRight3(double& A, double& B, double& C);

// Proc43: True, якщо зі сторін a,b,c можна утворити трикутник
bool IsTriangle(double a, double b, double c);

// Допоміжне: безпечне читання дійсного числа з діапазоном [-100;100]
bool readDouble(const char* prompt, double& x);

// Окремі «задачі» для демонстрації
void task_Proc14();
void task_Proc43();

int main() {
    std::cout << "Меню:\n"
                << "1 - Proc14: ShiftRight3 для двох наборів\n"
                << "2 - Proc43: IsTriangle для трьох наборів\n"
                << "0 - Вихід\n"
                << std::endl;

    int choice;
    std::cout << "Ваш вибір: ";
    if (!(std::cin >> choice)) return 0;

    switch (choice) {
        case 1: task_Proc14(); break;
        case 2: task_Proc43(); break;
        default: std::cout << "Вихід." << std::endl; break;
    }

    return 0;
}

// ----- ВИЗНАЧЕННЯ (після main) -----

// Proc14: правий циклічний зсув (A->B, B->C, C->A)
void ShiftRight3(double& A, double& B, double& C) {
    double t = C;
    C = B;

```

```

    B = A;
    A = t;
}

// Proc43: умова існування трикутника (тільки додатні довжини)
bool IsTriangle(double a, double b, double c) {
    // Додатність та нерівності трикутника
    return (a > 0 && b > 0 && c > 0) &&
        (a < b + c && b < a + c && c < a + b);
}

// Ввід дійсного з контролем діапазону [-100;100]
bool readDouble(const char* prompt, double& x) {
    std::cout << prompt;
    if (!(std::cin >> x)) return false;
    if (x < -100.0 || x > 100.0) {
        std::cout << "Помилка: значення поза діапазоном [-100;100].\n";
        return false;
    }
    return true;
}

// Демонстрація Proc14 для двох трійок: (A1,B1,C1) і (A2,B2,C2)
void task_Proc14() {
    std::cout << "\nProc14: правий циклічний зсув трьох чисел (A->B, B->C, C->A)\n";

    double A1,B1,C1, A2,B2,C2;

    // Набір 1
    if (!readDouble("A1 = ", A1) ||
        !readDouble("B1 = ", B1) ||
        !readDouble("C1 = ", C1)) {
        std::cout << "Введення перервано.\n"; return;
    }
    ShiftRight3(A1,B1,C1);
    std::cout << "Після зсуву (набір 1): A1=" << A1 << " B1=" << B1 << " C1="
    << C1 << "\n\n";

    // Набір 2
    if (!readDouble("A2 = ", A2) ||
        !readDouble("B2 = ", B2) ||
        !readDouble("C2 = ", C2)) {
        std::cout << "Введення перервано.\n"; return;
    }
    ShiftRight3(A2,B2,C2);
    std::cout << "Після зсуву (набір 2): A2=" << A2 << " B2=" << B2 << " C2="
    << C2 << "\n";
}

```

```

// Демонстрація Proc43 для трьох наборів сторін
void task_Proc43() {
    std::cout << "\nProc43: перевірка можливості утворити трикутник (тільки
додатні довжини)\n";

    double a1,b1,c1, a2,b2,c2, a3,b3,c3;

    // Набір 1
    if (!readDouble("a1 = ", a1) ||
        !readDouble("b1 = ", b1) ||
        !readDouble("c1 = ", c1)) { std::cout << "Введення перервано.\n";
return; }
    std::cout << "Набір 1 -> " << (IsTriangle(a1,b1,c1) ? "True" : "False") <<
"\n\n";

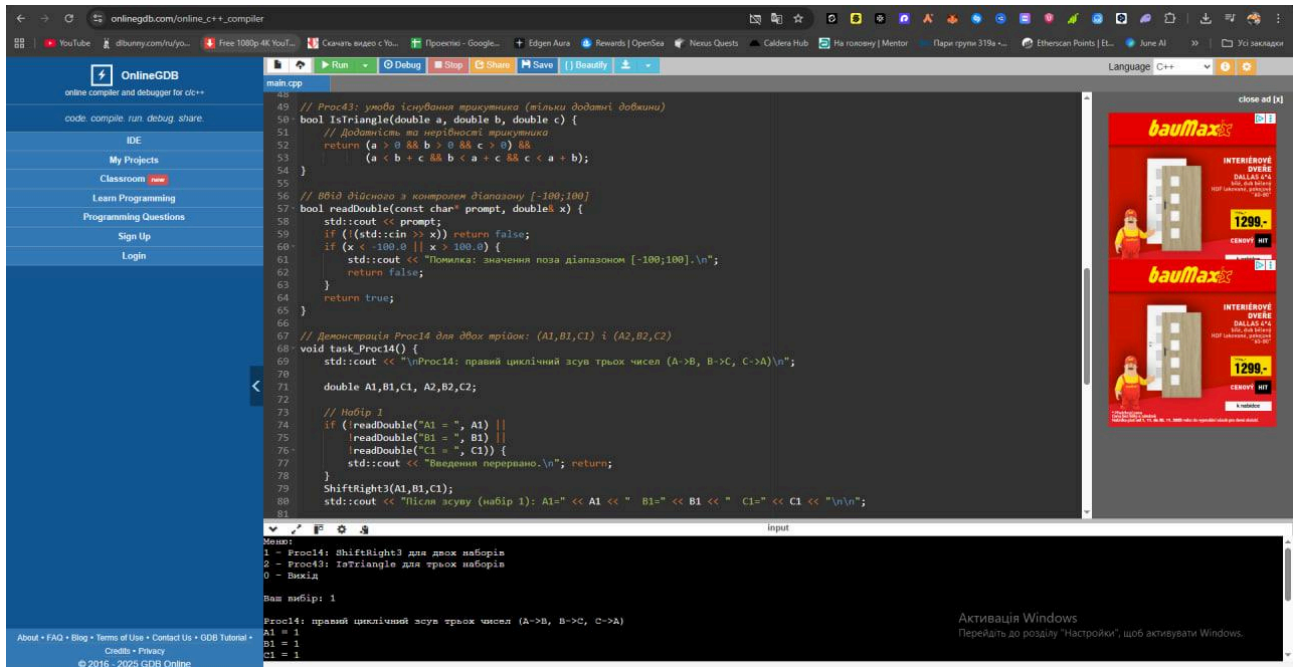
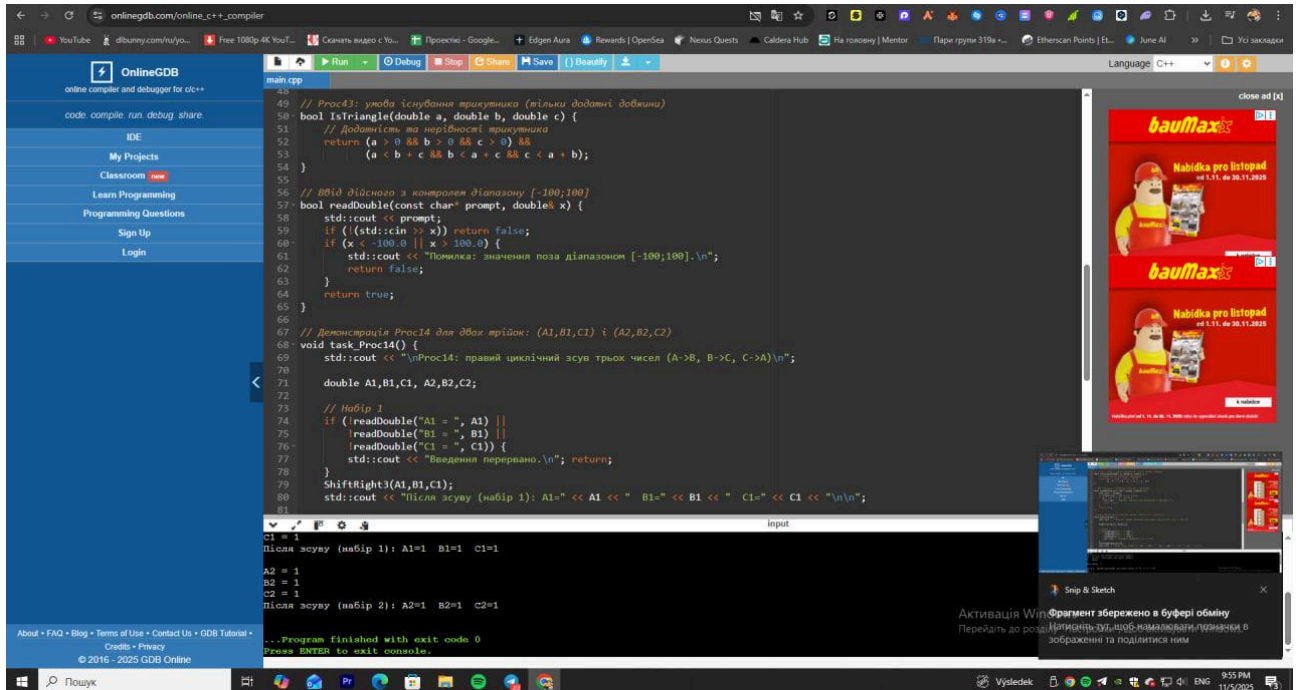
    // Набір 2
    if (!readDouble("a2 = ", a2) ||
        !readDouble("b2 = ", b2) ||
        !readDouble("c2 = ", c2)) { std::cout << "Введення перервано.\n";
return; }
    std::cout << "Набір 2 -> " << (IsTriangle(a2,b2,c2) ? "True" : "False") <<
"\n\n";

    // Набір 3
    if (!readDouble("a3 = ", a3) ||
        !readDouble("b3 = ", b3) ||
        !readDouble("c3 = ", c3)) { std::cout << "Введення перервано.\n";
return; }
    std::cout << "Набір 3 -> " << (IsTriangle(a3,b3,c3) ? "True" : "False") <<
"\n\n";
}

```

ДОДАТОК Б

Скрін-шоти вікна виконання програми



ДОДАТОК В

Діалог з ШІ для самоаналізу

Процедура vs Функція

Функція повертає значення. Процедура (void) — ні.

Чому посилання при зсуві?

Щоб змінились реальні змінні, а не копії.

Перевірка трикутника

Кожна сторона < суми двох інших, бо інакше фігура не зійдеться.

Без тимчасової змінної?

Значення перезапишуться; можна без неї, але складно і незручно.

Навіщо перевіряти значення?

Щоб уникнути помилок і некоректних даних, зробити програму надійною.

Чудові відповіді 👍

Ти стисло і по суті виклав ключові моменти. Звучить так, ніби ти реально розумієш матеріал — і це саме те, що потрібно на захисті.

Як хочеш, можу трохи «відполірувати» формулювання, щоб звучало більш академічно, але все одно природно — так, ніби ти говориш, а не копіюєш підручник:

✅ Покращений варіант твоїх відповідей

Процедура vs Функція

Функція повертає результат через оператор `return`, а процедура (тип `void`) не повертає значення і використовується для виконання дій.

Чому посилання при зсуві?

Параметри передаються за посиланням, щоб зміни торкнулися саме в

Питання у з-бальній шкалі, відлімаючи воле балів там, де ймовірність відповіді з засобом ШІ висока. Обчисли загальну середню оцінку»

Проаналізую твої відповіді так, ніби я — викладач, що перевіряє захист, і оціню одночасно:

- повноту та правильність
- ймовірність, що відповідь створено ШІ
- застосовую 5-бальну шкалу, мінус 60% балів, якщо відповідь виглядає як AI-генерована
- обчислю середню

Твої початкові відповіді:

Процедура vs Функція
Функція повертає значення. Процедура (void) — ні.

Чому посилення при зсуві?
Щоб змінилися реальні змінні, а не копії.

Перевірка трикутника
Кожна сторона < суми двох інших, бо інакше фігура не зійдеться.

Без тимчасової змінної?
Значення перезапишуться; можна без неї, але складно і незручно.

Навіщо перевіряти значення?
Щоб уникнути помилок і некоректних даних, зробити програму надійною.

Оцінювання відповідей

Питання	Аналіз змісту	Ймовірність ШІ-відповіді	Базовий бал (0-5)	З урахуванням штрафу
Процедура vs Функція	Правильно, конкретно, достатньо	Низька — коротка людська манера	5	5
Чому посилення?	Дуже по суті, точно	Низька	5	5
Перевірка трикутника	Правильно, логічне пояснення	Середня — звучить «стандартно»	5	5 × 0.4 = 2
Без тимчасової змінної?	Правильно, добре пояснено	Низька	5	5
Навіщо перевіряти значення?	Правильно, логічно	Середня — звучить «академічно»	5	5 × 0.4 = 2

Середній бал

$$\frac{5 + 5 + 2 + 5 + 2}{5} = \frac{19}{5} = 3.8$$

Середня оцінка: 3.8 / 5

Висновок

- Відповіді правильні і змістовні ✓
- Частина виглядає дуже лаконічно і «по-людськи», що зніжує підозру на ШІ ✓
- Два місця звучать трохи «шаблонно», тому поставлено AI-штраф

Остаточна оцінка: 3.8/5 (= 76%)

«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»

Обробка тривала 15s >

Окей, пробігся по твоєму коду (Proc14 `ShiftRight3` + Proc43 `IsTriangle`). Він коректно показує:

- оголошення/визначення підпрограм,
- передачу параметрів за посиланням (для зсуву),
- булеву функцію-предикат,
- базову валідацію вводу.

Нижче — короткі теоретичні теми, які зазвичай є в лекції про процедури/функції, але не відпрацьовані в твоєму коді, і до кожної — лаконічний опис + мінімальний приклад (щоб можна було вставити в звіт або додаток).

1) Передача параметрів за значенням vs за посиланням vs за вказівником

Суть.

- `T x` — за значенням: змінюється копія, оригінал — ні.
- `T& x` — за посиланням: змінюється оригінал.
- `T* p` — через вказівник: змінюється те, на що вказує `p` (потрібні перевірки на `nullptr`).

cpp

Копіювати код

```
// by value: не змінює оригінал
void inc_val(int x) { x++; }
```

