

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»

XAI.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.)

2025

МЕТА РОБОТИ

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

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити дві задачі з реалізації функцій. Дляожної 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>
#include <cmath>
using namespace std;

// ===== ОГОЛОШЕННЯ ФУНКЦІЙ (до main) =====

// Proc43: логічна функція – чи можна утворити трикутник з a,b,c
bool IsTriangle(double a, double b, double c);

// Proc14: процедура – правий циклічний зсув A->B, B->C, C->A
void ShiftRight3(double &A, double &B, double &C);

// Процедури-завдання (кожне – окремий сценарій з введенням/виводом)
void task_Proc14(); // правий циклічний зсув для двох трійок
void task_Proc43(); // перевірка трикутника для трьох наборів

// Меню
void show_menu();
void run_menu();

// ===== main =====
int main() {
    cout << "ЛР №4 – Функції в C++ (варіанти: Proc14, Proc43)\n";
    run_menu();
    return 0;
}

// ===== ВИЗНАЧЕННЯ ФУНКЦІЙ (після main) =====

// Proc43: умова існування трикутника – сума будь-яких двох сторін > третьої
bool IsTriangle(double a, double b, double c) {
    return (a > 0 && b > 0 && c > 0) &&
           (a + b > c) && (a + c > b) && (b + c > a);
}

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

```

```

}

// Завдання Proc14
void task_Proc14() {
    cout << "\n--- Proc14: Правий циклічний зсув для двох наборів ---\n";
    double A1, B1, C1, A2, B2, C2;

    cout << "Введіть перший набір (A1 B1 C1) у діапазоні [-100,100]: ";
    if (!(cin >> A1 >> B1 >> C1) ||
        fabs(A1) > 100 || fabs(B1) > 100 || fabs(C1) > 100)
    {
        cout << "Некоректні дані. Обчислення перервано.\n";
        cin.clear(); cin.ignore(numeric_limits<streamsize>::max(), '\n');
        return;
    }

    cout << "Введіть другий набір (A2 B2 C2) у діапазоні [-100,100]: ";
    if (!(cin >> A2 >> B2 >> C2) ||
        fabs(A2) > 100 || fabs(B2) > 100 || fabs(C2) > 100)
    {
        cout << "Некоректні дані. Обчислення перервано.\n";
        cin.clear(); cin.ignore(numeric_limits<streamsize>::max(), '\n');
        return;
    }

    // без циклів, як вимагається для завдання 1
    ShiftRight3(A1, B1, C1);
    ShiftRight3(A2, B2, C2);

    cout << "Після зсуву:\n";
    cout << " Набір 1: " << A1 << " " << B1 << " " << C1 << "\n";
    cout << " Набір 2: " << A2 << " " << B2 << " " << C2 << "\n";
}

// Завдання Proc43
void task_Proc43() {
    cout << "\n--- Proc43: Перевірка можливості утворити трикутник ---\n";

    auto read_set = [] (const char* label, double &a, double &b, double &c) -> bool{
        cout << "Введіть " << label << " (a b c), додатні, у межах (0,100]: ";
        if (!(cin >> a >> b >> c)) return false;
        return (a > 0 && b > 0 && c > 0 && a <= 100 && b <= 100 && c <= 100);
    };

    double a1,b1,c1,a2,b2,c2,a3,b3,c3;
    if (!read_set("набір 1",a1,b1,c1) ||
        !read_set("набір 2",a2,b2,c2) ||
        !read_set("набір 3",a3,b3,c3))
}

```

```

{
    cout << "Некоректні дані. Обчислення перервано.\n";
    cin.clear(); cin.ignore(numeric_limits<streamsize>::max(), '\n');
    return;
}

cout << "Набір 1: " << (IsTriangle(a1,b1,c1) ? "Можна" : "Не можна") << "
утворити трикутник\n";
cout << "Набір 2: " << (IsTriangle(a2,b2,c2) ? "Можна" : "Не можна") << "
утворити трикутник\n";
cout << "Набір 3: " << (IsTriangle(a3,b3,c3) ? "Можна" : "Не можна") << "
утворити трикутник\n";
}

// Меню
void show_menu() {
    cout << "\n===== МЕНЮ =====\n";
    cout << "1 - Proc14: Правий циклічний зсув (2 набори)\n";
    cout << "2 - Proc43: Чи можна утворити трикутник (3 набори)\n";
    cout << "0 - Вихід\n";
    cout << "Ваш вибір: ";
}

void run_menu() {
    while (true) {
        show_menu();
        int cmd;
        if (!(cin >> cmd)) { // захист від сміттєвого вводу
            cin.clear(); cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Введіть номер пункту меню.\n";
            continue;
        }
        switch (cmd) {
            case 1: task_Proc14(); break;
            case 2: task_Proc43(); break;
            case 0: cout << "Готово. Успіхів!\n"; return;
            default: cout << "Невідомий пункт.\n";
        }
    }
}
}

```

ДОДАТОК Б
Скрін-шоти вікна виконання програми
(14)

```

===== МЕНЮ =====
1 - Proc14: Правий циклічний зсув (2 набори)
2 - Proc43: Чи можна утворити трикутник (3 набори)
0 - Вихід
Ваш вибір: 1

--- Proc14: Правий циклічний зсув для двох наборів ---
Введіть перший набір (A1 B1 C1) у діапазоні [-100,100]: -100
100
54

```

```

116     show_menu();
117
118     int cmd;
119     if (!(cin >> cmd)) { // захист від сміттєвого вводу
120         cin.clear(); cin.ignore(numeric_limits<streamsize>::max(), '\n');
121         cout << "Введіть номер пункту меню.\n";
122         continue;
123     }
124     switch (cmd) {
125         case 1: task_Proc14(); break;
126         case 2: task_Proc43(); break;
127         case 0: cout << "Готово. Успіхів!\n"; return;
128     }
129 }
130 }
```

```

--- Proc14: Правий циклічний зсув для двох наборів ---
Введіть перший набір (A1 B1 C1) у діапазоні [-100,100]: -100
100
54
Введіть другий набір (A2 B2 C2) у діапазоні [-100,100]: -53
100
32
Після зсуву:
    Набір 1: 54 -100 100
    Набір 2: 32 -53 100

```

The screenshot shows a Windows desktop environment. In the center is a terminal window titled "input". The code in the terminal is:

```
117
118     int cmd;
119     if (!(cin >> cmd)) { // захист від сміттєвого вводу
120         cin.clear(); cin.ignore(numeric_limits<streamsize>::max(), '\n');
121         cout << "Введіть номер пункту меню.\n";
122         continue;
123     }
124     switch (cmd) {
125     case 1: task_Proc14(); break;
126     case 2: task_Proc43(); break;
127     case 0: cout << "Готово. Успіхів!\n"; return;
128     default: cout << "Невідомий пункт.\n";
129 }
130 }
```

Below the code, the terminal displays the following output:

```
100
32
Після зсуву:
    Набір 1: 54 -100 100
    Набір 2: 32 -53 100

===== МЕНЮ =====
1 - Proc14: Правий циклічний зсув (2 набори)
2 - Proc43: Чи можна утворити трикутник (3 набори)
0 - Вихід
Ваш вибір:
```

The taskbar at the bottom of the screen shows various application icons, including Microsoft Word, Microsoft Excel, Microsoft Powerpoint, Microsoft Edge, File Explorer, Google Chrome, Spotify, and others.

Скрін-шоти вікна виконання програми
(43)

```

106 void show_menu() {
107     cout << "\n===== МЕНЮ =====\n";
108     cout << "1 – Proc14: Правий циклічний зсув (2 набори)\n";
109     cout << "2 – Proc43: Чи можна утворити трикутник (3 набори)\n";
110     cout << "0 – Вихід\n";
111     cout << "Ваш вибір: ";
112 }
113
114 void run_menu() {
115     while (true) {
116         show_menu();
117         int cmd;
118         if (!(cin >> cmd)) { // захист від смішевого вводу
119             cin.clear(); cin.ignore(numeric_limits<streamsize>::max(), '\n');
120             cout << "Введіть номер пункту меню.\n";
121             continue;
122         }
123         switch (cmd) {
124             case 1: task_Proc14(); break;
125             case 2: task_Proc43(); break;
126             case 0: cout << "Готово. Успіхів!\n"; return;
127             default: cout << "Невідомий пункт.\n";
128         }
129     }
130 }
```

ЛР №4 – Функції в C++ (варіанти: Proc14, Proc43)

```

===== МЕНЮ =====
1 – Proc14: Правий циклічний зсув (2 набори)
2 – Proc43: Чи можна утворити трикутник (3 набори)
0 – Вихід
Ваш вибір: 2

--- Proc43: Перевірка можливості утворити трикутник ---
Введіть набір 1 (a b c), додатні, у межах (0,100]: 10
10

```

```

111     cout << "Ваш вибір: ";
112 }
113
114 void run_menu() {
115     while (true) {
116         show_menu();
117         int cmd;
118         if (!(cin >> cmd)) { // захист від смішного вводу
119             cin.clear(); cin.ignore(numeric_limits<streamsize>::max());
120             cout << "Введіть номер пункту меню.\n";
121             continue;
122         }
123         switch (cmd) {
124             case 1: task_Proc14(); break;
125             case 2: task_Proc43(); break;
126             case 0: cout << "Готово. Успіхів!\n"; return;
127             default: cout << "Невідомий пункт.\n";
128         }
129     }
130 }
```

Ваш вибір: 2

--- Proc43: Перевірка можливості утворити трикутник ---

Введіть набір 1 (a b c), додатні, у межах (0,100]: 10

10

10

Введіть набір 2 (a b c), додатні, у межах (0,100]: 10

5

3

Введіть набір 3 (a b c), додатні, у межах (0,100]: 15

52

126 case 0: cout << "Готово. Успіхів!\n"; return;
127 default: cout << "Невідомий пункт.\n";
128 }
129 }
130 }

52

32

Набір 1: Можна утворити трикутник

Набір 2: Не можна утворити трикутник

Набір 3: Не можна утворити трикутник

===== МЕНЮ =====

1 - Proc14: Правий циклічний зсув (2 набори)

2 - Proc43: Чи можна утворити трикутник (3 набори)

0 - Вихід

Ваш вибір:

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 зайдеться.

Без тимчасової змінної?
Значення перезапищуються; 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 засіданні школи, відповідаючи 60% відповідам, дає
ймовірність
відповіді з засобом ШІ висока. Обчисли загальну середню оцінку»

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

- повноту та правильність
- ймовірність, що відповідь створено ШІ
- застосувала 5-балльну шкалу, мінус 60% балів, якщо відповідь виглядає як AI-генерована
- обчислило середню

Твої початкові відповіді:

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 зайдеться.

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

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

Оцінювання відповідей

| Питання | Аналіз змісту | Імовірність ШІ-відповіді | Базовий бал (0-5) | З урахуванням штрафу |
|---------------------------|---------------------------------|---------------------------------|-------------------|----------------------|
| Процедура vs Функція | Правильно, конкретно, достатньо | Нічого — коротка людська манера | 5 | 5 |
| Чому посилання? | Дуже по суті, точно | Нічого | 5 | 5 |
| Перевірка трикутника | Правильно, логічне пояснення | Середня — звучить «стандартно» | 5 | $5 \times 0.4 = 2$ |
| Без тимчасової змінної? | Правильно, добре пояснено | Нічого | 5 | 5 |
| Навіщо перевірка значень? | Правильно, логічно | Середня — звучить «академічно» | 5 | $5 \times 0.4 = 2$ |

■ Середній бал

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

■ Середня оцінка: 3.8 / 5

⌚ Висновок

- Відповіді правильні і змістовні
- Частини виглядає дуже лаконічно і «по-людськи», що знижує підозру на ШІ
- Два місця звичайно 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++; }
```

