

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 «з файлу»

ХАІ.301. електрична інженерія, група 319а, номер в списку 10 ЛР

Виконав студент гр. _____ 319а _____

_____ Аріна Кушакова _____
(підпис, дата) (П.І.Б.)

Перевірів

_____ 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.

Завдання 2. Для вирішення завдання з цілочисельними змінними з Лабораторної роботи №2 відповідно до варіанту визначити три функції:

- 1) функцію введення вхідних значень з консолі з перевіркою їх на коректність;
- 2) функцію підрахунку результату;
- 3) функцію виведення результату в консоль.

При виклику функцій в разі введення некоректних вхідних даних вивести відповідне повідомлення і розрахунки не виконувати.

Завдання 3. Для вибору користувачем одного з трьох зазначених вище завдань організувати меню з використанням інструкції вибору. Кожне завдання має бути реалізовано у вигляді окремої процедури (функції без параметрів), що містить необхідні оголошення змінних і виклики інших функцій.

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 III 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 III. Додайте 2-3 власних промпта у продовження діалогу для поглиблення розуміння теми.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Задача **Proc19**: Описати процедуру `LoanPayment(S, r, n, P)`, яка обчислює щомісячний платіж P за кредитом: $P = \frac{S r (1+r)^n}{(1+r)^n - 1}$, де S - сума кредиту, r - місячна процентна ставка, n - кількість місяців (S, r, n - вхідні, P - вихідний параметр дійсного типу). За допомогою цієї процедури знайти платежі для трьох різних кредитів.

Вирішення задачі `Proc19`

Вхідні дані (ім'я, опис, тип, обмеження):

Змінна S — сума кредиту; тип даних `double`.

Змінна r — місячна процентна ставка; тип даних `double`.

Змінна n — кількість місяців; тип даних `double`.

Вихідні дані (ім'я, опис, тип):

Значення P — щомісячний платіж; тип даних `double`.

Алгоритми функцій що викликаються представлені в графічному вигляді на рис. 1-3.

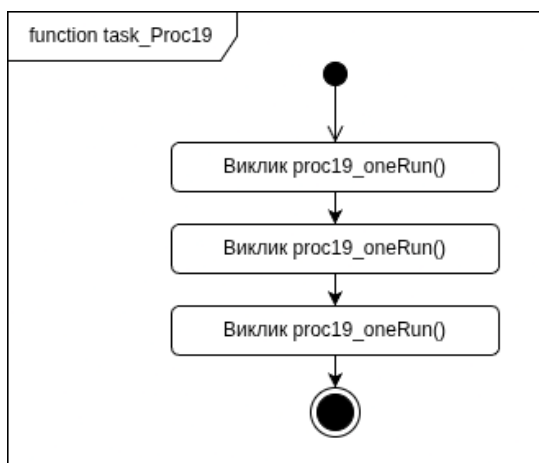


Рисунок 1 – Діаграма активності функції task_Proc19()

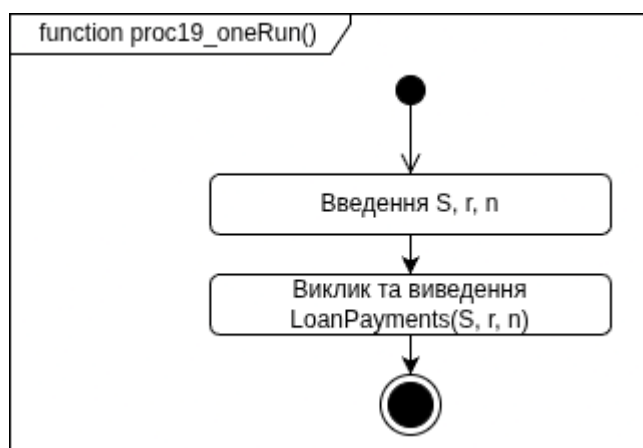


Рисунок 2 – Діаграма активності функції proc19_oneRun()

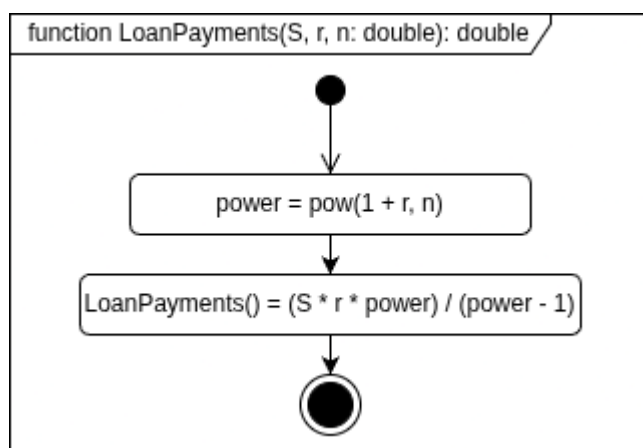


Рисунок 3 – Діаграма активності функції LoanPayments()

Екран роботи програми показаний на рис. Б.1.

Завдання 2.

Задача **Proc35**: Описати функцію Odd (K) логічного типу, яка повертає True, якщо цілий параметр K є непарним, і False в іншому випадку. З її допомогою знайти кількість парних чисел в наборі з 5 цілих чисел.

Вирішення задачі Proc35

Вхідні дані (ім'я, опис, тип, обмеження):

Змінні a, b, c, d, e — числа для перевірки парності; тип даних integer.

Вихідні дані (ім'я, опис, тип):

Змінна count — кількість парних чисел у наборі; тип даних integer.

Алгоритми функцій що викликаються представлені в графічному вигляді на рис. 4-5.

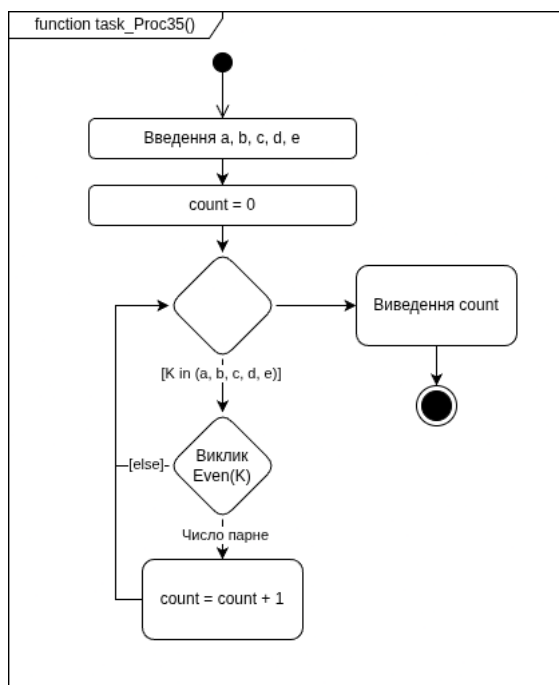


Рисунок 4 – Діаграма активності функції task_Proc35()

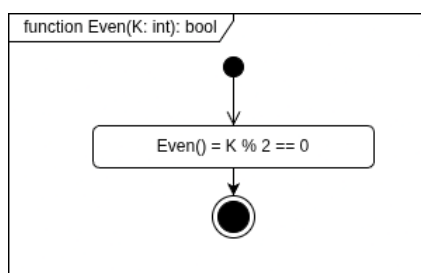


Рисунок 5 – Діаграма активності функції Even()

Екран роботи програми показаний на рис. Б.2.

Завдання 3.

Задача **Integer37**: Дано прямокутник довжиною A і шириною B . Знайти, скільки квадратних плиток зі стороною C можна вмістити та яка частина прямокутника залишиться незаповненою?

Вирішення задачі Integer37

Вхідні дані (ім'я, опис, тип, обмеження):

Змінні A, B — сторони прямокутника; тип даних `integer`.

Змінна C — розмір плитки; тип даних `integer`.

Вихідні дані (ім'я, опис, тип):

Змінна `result` — кількість плиток, що можна вмістити; тип даних `integer`.

Значення непокритої площі; тип даних `integer`.

Алгоритми функцій що викликаються представлені в графічному вигляді на рис. 6-9.

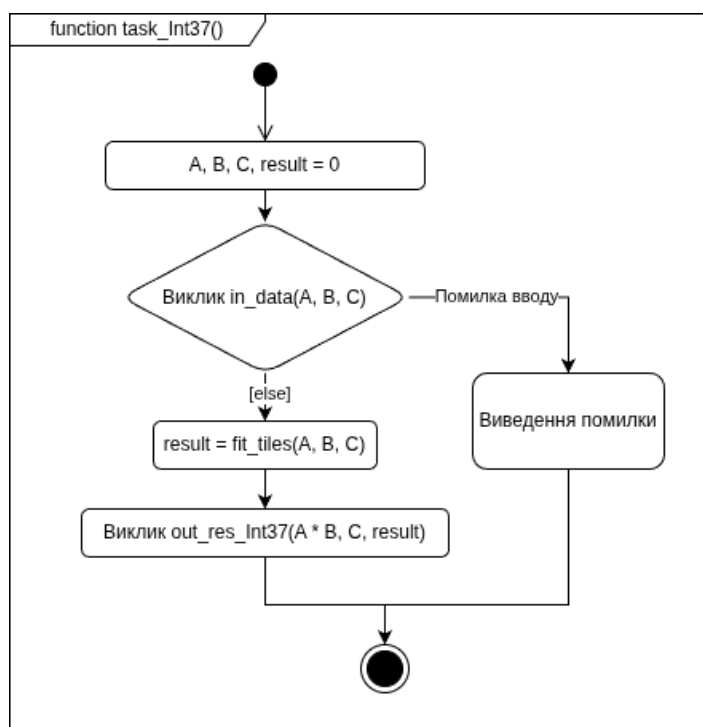


Рисунок 6 – Діаграма активності функції `task_Int37()`

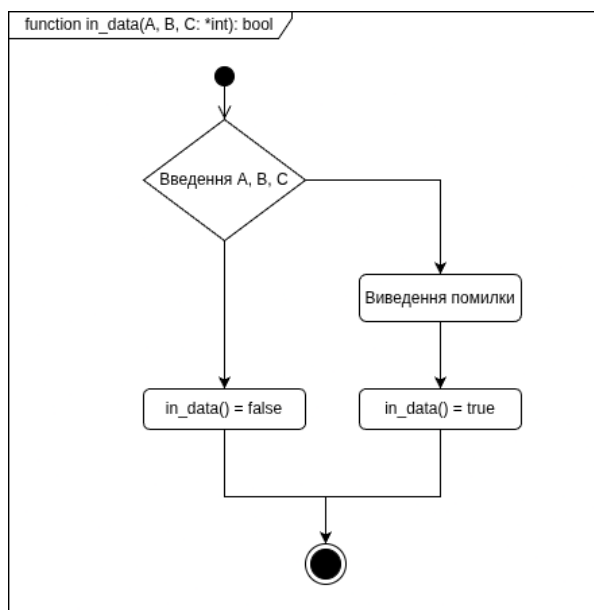


Рисунок 7 – Діаграма активності функції in_data()

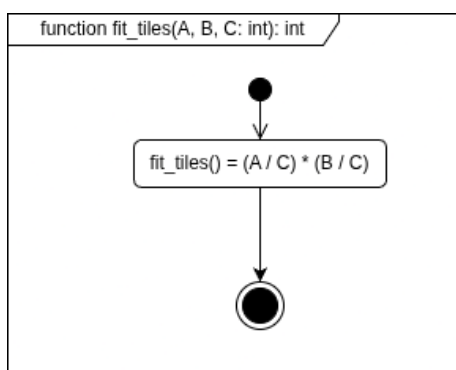


Рисунок 8 – Діаграма активності функції fit_tiles()

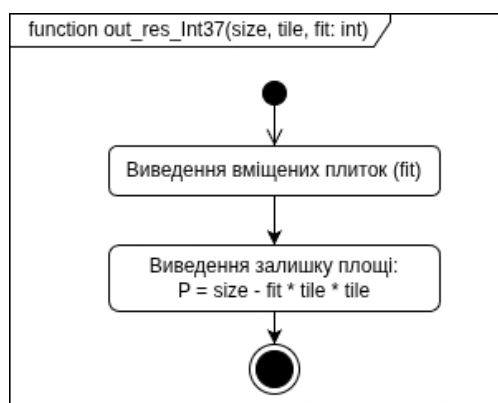


Рисунок 9 – Діаграма активності функції out_res_Int37()

Екран роботи програми показаний на рис. Б.3.

Лістинг коду вирішення задач наведено в дод. А (стор. 9).

ВИСНОВКИ

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

ДОДАТОК А

Лістинг коду програми

```

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

// Proc 19
//
// Описати процедуру LoanPayment(S, r, n, P),
// яка обчислює щомісячний платіж P за кредитом:
//  $P = (S \cdot r \cdot (1+r)^n) / ((1+r)^n - 1)$ ,
// де S - сума кредиту,
// r - місячна процентна ставка,
// n - кількість місяців
// (S, r, n - вхідні, P - вихідний параметр дійсного типу).
// За допомогою цієї процедури знайти платежі для трьох різних кредитів.
//
double LoanPayments(double S, double r, double n); // Визначення функції
void proc19_oneRun(); // Допоміжна функція для одного набору даних
void task_Proc19(); // Задача Proc19 цілком

// Proc35
//
// Описати функцію Even(K) логічного типу, яка повертає True, якщо
// цілий параметр K є парним, і False в іншому випадку. З її допомогою
// знайти кількість парних чисел в наборі з 5 цілих чисел
bool Even(int K); // Визначення функції
void task_Proc35(); // Завдання Proc35 цілком

// Integer35 - Дано кількість годин.
// Дано прямокутник довжиною A і шириною B. Знайти, скільки
// квадратних плиток зі стороною C можна вмістити та яка частина
// прямокутника залишиться незаповненою?
//
bool in_data(int &A, int &B, int &C); // Введення з перевіркою Integer35
int fit_tiles(int &A, int &B, int &C); // Підрахунок Integer37
void out_res_Int37(int size, int tile, int fit); // Виведення результатів
Integer37
void task_Int37(); // Задача Integer37 цілком

int main() {
    int menu;
    cout << "Номер завдання: ";
    cin >> menu;
    // перемикання між завданнями
    switch(menu) {

```

```

        case 1: task_Proc19(); break;
        case 2: task_Proc35(); break;
        case 3: task_Int37(); break;
        default: cout << "Невірний номер завдання! (Тільки 1, 2, 3)" << endl;
    }
    return 0;
}

// Функція обчислення щомісячного платежа
double LoanPayments(double S, double r, double n) {
    double power = pow(1 + r, n);
    return (S * r * power) / (power - 1);
}

// Завдання Proc19 цілком
void task_Proc19() {
    cout << "Proc19" << endl << endl;
    proc19_oneRun();
    proc19_oneRun();
    proc19_oneRun();
}

// Виконання завдання Proc19 для одного набору даних
// Допоміжна функція для зменшення дублікації коду
void proc19_oneRun() {
    // Визначення змінних
    double S, r, n;
    // Зчитування даних (без перевірки)
    cout << "Введіть суму кредиту: ";
    cin >> S;
    cout << "Введіть місячну процентну ставку: ";
    cin >> r;
    // Переводимо відсоткове значення у дробове
    r = r / 100;    cout << "Введіть кількість місяців: ";
    cin >> n;
    // Розрахунок та виведення результату
    cout << "Щомісячний платіж: " << LoanPayments(S, r, n) << endl << endl;
}

// Функція перевірки парності числа
bool Even(int K) {
    return K % 2 == 0;
}

// Завдання Proc35 цілком
void task_Proc35() {
    cout << "Proc35" << endl << endl;
    // Визначення змінних
    int a, b, c, d, e;
    // Змінна-лічильник результату

```

```

int count = 0;
// Введення даних
cout << "Введіть 5 цілих чисел: ";
cin >> a >> b >> c >> d >> e;
// Виконання перевірки та збільшення лічильника
if (Even(a))
    count = count + 1;
if (Even(b))
    count = count + 1;
if (Even(c))
    count = count + 1;
if (Even(d))
    count = count + 1;
if (Even(e))
    count = count + 1;
// Виведення результату
cout << "Парних чисел у наборі: " << count << endl;
}

// Завдання Integer37 цілком
void task_Int37() {
    // Визначення змінних
    int A, B, C;
    int result;
    // Введення даних з перевіркою
    if (in_data(A, B, C))
        // Якщо не вдалося ввести дані, виводимо повідомлення помилки
        cout << "Помилка введення" << endl;
    else {
        // Обчислюємо кількість годин
        result = fit_tiles(A, B, C);
        // Виводимо результат
        out_res_Int37(A * B, C, result);
    }
}

// Функція введення годин з перевіркою
bool in_data(int &A, int &B, int &C) {
    cout << "Integer37" << endl << endl;
    cout << "Введіть сторони прямокутника: ";
    // Зчитуємо сторони прямокутника
    cin >> A >> B;
    // Якщо не вдалося зчитати число, повертаємо true (помилка введення)
    if (cin.fail()) {
        cout << "Не вдалося зчитати дані" << endl;
        return true;
    }
    // Якщо сторона прямокутника менше 0, повертаємо true (помилка введення)
    else if (A <= 0 || B <= 0) {
        cout << "Сторона прямокутника має бути більша за 0" << endl;
    }
}

```

```

        return true;
    }
    cout << "Введіть розмір плитки: ";
    // Зчитуємо розмір плитки
    cin >> C;
    // Якщо не вдалося зчитати число, повертаємо true (помилка введення)
    if (cin.fail()) {
        cout << "Не вдалося зчитати дані" << endl;
        return true;
    }
    // Якщо сторона плитки менше 0, повертаємо true (помилка введення)
    else if (C <= 0) {
        cout << "Сторона плитки має бути більша за 0" << endl;
        return true;
    }
    // Повертаємо false (успішне введення)
    else return false;
}

// Функція обчислення днів
int fit_tiles(int &A, int &B, int &C) {
    return (A / C) * (B / C);
}

// Функція виведення результату завдання Integer37
void out_res_Int37(int size, int tile, int fit) {
    cout << "Можна вмістити плиток: " << fit << endl;
    cout << "Залишиться незаповненою: " << size - fit * tile * tile << endl;
}

```

ДОДАТОК Б
Скрін-шоти вікна виконання програми

```
Номер завдання: 1  
Proc19  
  
Введіть суму кредиту: 100  
Введіть місячну процентну ставку: 25  
Введіть кількість місяців: 12  
Щомісячний платіж: 26.8448  
  
Введіть суму кредиту: 1000  
Введіть місячну процентну ставку: 10  
Введіть кількість місяців: 24  
Щомісячний платіж: 111.3  
  
Введіть суму кредиту: 10000  
Введіть місячну процентну ставку: 50  
Введіть кількість місяців: 36  
Щомісячний платіж: 5000
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання
Proc19

```
Номер завдання: 2  
Proc35  
  
Введіть 5 цілих чисел: 1 3 4 6 7  
Парних чисел у наборі: 2
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання
Proc35

```
Номер завдання: 3  
Integer37  
  
Введіть сторони прямокутника: 13 9  
Введіть розмір плитки: 4  
Можна вмістити плиток: 6  
Залишиться незаповненою: 21
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання
Integer37

ДОДАТОК В

Діалог з ШІ 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 <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

ChatGPT сказав:

Добре 🐾

Щоб я міг створити **питання, які дійсно стосуються твого коду та теоретичних відомостей**, мені потрібно побачити вміст файлу `main.cpp`.

Будь ласка, дозволь мені його переглянути — я зможу:

- проаналізувати логіку програми,
- визначити, які теми зачіпає код (наприклад, ООП, алгоритми, обробка даних тощо),
- і після цього скласти **5 тестових і 5 відкритих** запитань **середнього рівня складності**, орієнтованих на критичне та інженерне мислення.

Хочеш, щоб я відкрив і проаналізував вміст `main.cpp` зараз?

Вы сказали:

так, відкрив і проаналізуй файл

ChatGPT сказав:

Чудово 🐾

Я бачу, що твій файл `main.cpp` містить кілька завдань рівня «середній» зі шкільного або університетського курсу програмування на C++.

Основні модулі, які описані у кодї:

1. **Proc19** — обчислення щомісячного платежу за кредитом за формулою

$$P = S \cdot r \cdot (1+r)^n \cdot \frac{1}{(1+r)^n - 1} \quad P = \frac{S \cdot r \cdot (1+r)^n}{(1+r)^n - 1}$$

Тут розвиваються навички роботи з дійсними числами, параметрами функцій і математичними формулами.

2. **Proc35** — визначення парності числа за допомогою логічної функції `Even(K)` та підрахунок кількості парних серед 5 чисел.
 3. **Integer37** (або `Integer35` за коментарями) — обчислення кількості квадратних плиток, які можна розмістити в прямокутнику, і частини, що залишилась незаповненою. Тут перевіряються навички роботи з цілими числами, діленням і залишком.
-

Тепер створюю **5 тестових і 5 відкритих** запитань середнього рівня складності, спрямованих на **критичне та інженерне мислення**, заснованих на цьому коді.

☐ **Тестові питання (1 правильна відповідь із 4)**

1. Який тип параметра найдоцільніше використати для зберігання щомісячної процентної ставки `r` у функції `LoanPayments`?

- A. `int`
 - B. `float`
 - C. `double` ✓
 - D. `long`
-

2. У функції `Even(int k)` результат обчислюється як `k % 2 == 0`. Що саме робить оператор `%` у цьому виразі?

- A. Ділить число `k` на 2 і округлює вниз
 - B. Повертає залишок від ділення ✓
 - C. Перевіряє ділення без залишку
 - D. Визначає, чи є `k` від'ємним
-

3. Якщо користувач вводить від'ємне значення для кількості місяців `n` у `LoanPayments`, яка потенційна проблема може виникнути?

- A. Переповнення буфера
 - B. Ділення на нуль ✓
 - C. Переповнення пам'яті
 - D. Нескінченний цикл
-

4. У задачі з плитками (`Integer37`) використовуються оператори цілочисельного ділення. Чому не можна використовувати звичайне ділення з типом `double`?

- A. Воно занадто повільне
 - B. Не забезпечує точності
 - C. Дає дробові результати, які не мають фізичного сенсу ✓
 - D. Може викликати помилку компіляції
-

5. Яка з наведених функцій **найбільш придатна** для повторного використання в інших програмах без змін?

- A. `task_Proc19()`
- B. `task_Proc35()`
- C. `Even()` ✓
- D. `main()`

💡 Відкриті питання (на розвиток критичного й інженерного мислення)

1. Опиши, як ти б модифікував функцію `LoanPayments`, щоб вона враховувала разову комісію банку (% від суми кредиту). Як це вплине на результат?

2. Уяви, що користувач вводить невалідні дані (наприклад, $r = -0.05$ або $n = 0$). Який механізм обробки помилок ти б запропонував реалізувати?

3. Якби треба було обчислити не кількість, а **розташування плиток** у прямокутнику, які структури даних ти б обрав і чому?

4. Порівняй реалізацію функцій `Even()` і `LoanPayments()`. Які принципи модульності чи повторного використання коду вони ілюструють? Як це покращує масштабованість програми?

5. Запропонуй спосіб об'єднати всі три задачі (`Proc19`, `Proc35`, `Integer37`) у єдиний інтерфейс із меню, який дозволить користувачу вибирати та виконувати завдання більш інтуїтивно.

Хочеш, щоб я зробив **тестовий варіант у форматі PDF або DOCX** (наприклад, для здачі чи друку)?

Вы сказали:

Відповіді на відкриті питання: 1. Нехай K - разова комісія банку. Тоді, нове значення `LoanPayments` буде розраховуватись як: $P = (K + (S * r * ((1 + r)^n))) / ((1+r)^n - 1)$ Таким чином, щомісячний платіж збільшиться на певний відсоток від разової комісії. 2. Після вводу даних можна провести перевірку на правильність, і при невірному вводі вивести помилку. Це можна зробити через такий блок коду: `if (r <= 0 || n <= 0) { cout << "Помилка вводу"; return; }` Цей код перевірить, чи є значення r та n більшими за 0. Якщо виявиться інакше, програма видасть помилку та завершить виконання. 3. Для обчислення розташування плиток можна використати матрицю розмірами $A \times B$. Клітинки матриці мали б істинні значення (покріті / непокріті плиткою), або числові значення з номером плитки, яка покриває клітинку (0 якщо клітинка непокрита). 4. Функції `Even` та `LoanPayments` мають дуже схожу структуру - обидві функції проводять розрахунки та повертають певний результат. Розрахунки функції `Even` досить прості, тому значення розрахунку повертається одразу, без додаткових змінних, в той час як функція `LoanPayments` використовує додаткову змінну `power`, яка двічі використовується у повній формулі, що дозволяє ще більше зменшити повторюваність коду.

Щодо масштабованості програми, таке використання функцій значно полегшує майбутні зміни - наприклад, якщо у майбутньому зміниться формула для розрахунку LoanPayments, то цю зміну можна реалізувати лише в тілі функції, не змінюючи код виклику функції. 5. Найпростіший метод, представлений у коді - введення номеру задачі для переходу. Трохи складніший - три пункти меню для задач та одне для виходу. Після виконання задачі користувач повертається до меню до тих пір, поки не викличе пункт виходу. Найскладніший варіант - відстежувати натискання користувачем клавіш вгору-вниз, та рухати вказівник по пунктах меню. 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.

ChatGPT сказав:

Дуже добре, це якісний тип аналізу — ти хочеш не просто оцінити правильність відповідей, а й **перевірити рівень самостійності мислення** (ознаки використання ШІ). Нижче — мій детальний розбір **кожної відповіді**, включно з аналізом:

- **повноти** (чи всі аспекти питання розкрито),
- **правильності** (чи логічно та технічно коректно),
- **ймовірності використання ШІ** (граматика, структура, стиль),
- та підсумковою оцінкою за **5-бальною шкалою з корекцією (-60%)**, якщо відповідь імовірно створена ШІ.

□ Питання 1

Нехай K — разова комісія банку. Тоді нове значення LoanPayments буде:

$$P = K + (S \cdot r \cdot (1+r)^n) \frac{(1+r)^n - 1}{r} = \frac{K + (S \cdot r \cdot (1+r)^n)}{r} \frac{(1+r)^n - 1}{(1+r)^n - 1} = (1+r)^n - 1 K + (S \cdot r \cdot (1+r)^n) \dots$$

Таким чином, платіж збільшиться на певний відсоток.

Аналіз:

- ✓ **Правильність:** Формула логічна — комісія додана до чисельника, що збільшує суму платежу.
- ⚠ **Невелика неточність:** Комісія в реальних задачах зазвичай не додається до всіх платежів, а сплачується одноразово. Тому формула економічно спрощена, але математично допустима.
- ☝ **Повнота:** Пояснення наслідку (збільшення платежу) є.
- □ **Ознаки ШІ:** Формулювання гладке, академічне, без типових студентських помилок чи неточностей у пунктуації. Ймовірність використання ШІ — **висока (~70%)**.

Базова оцінка: 4.5 / 5

З урахуванням -60%: 1.8 / 5

□ Питання 2

Перевірка на правильність введення:

```
if (r <= 0 || n <= 0) {
    cout << "Помилка вводу";
    return;
}
```

Це перевіряє, чи значення r і n більші за 0...

Аналіз:

- ✓ *Правильність*: Код коректний — перевіряє вхідні дані й завершує виконання.
- ✓ *Повнота*: Є як технічна, так і логічна частина відповіді.
- ⚠ *Додатково*: Можна було вказати, що `return;` без `main()` потребує уточнення контексту — якщо це у функції, має бути `return;`, якщо в `main()`, то `return 0;`.
- □ *Ймовірність III*: помірна — структура відповіді близька до студентської, приклад коду простий і типовий.

Базова оцінка: 5 / 5

Ймовірність III ~40% → без зниження.

✓ **Остаточна оцінка:** 5 / 5

□ Питання 3

Для обчислення розташування плиток можна використати матрицю $A \times B$.
Клітинки матриці мають істинні значення (покриті / непокриті)...

Аналіз:

- ✓ *Правильність*: Ідея з матрицею логічна — це дискретне представлення поверхні.
- ⚠ *Недостатня деталізація*: Не вказано, як саме алгоритм визначатиме межі плиток або перевірятиме відповідність розмірам.
- 💬 *Повнота*: Середня — ідея зрозуміла, але без обговорення складності чи типу даних (наприклад, `bool` чи `int`).
- □ *Ознаки III*: Помірно висока (~60%) — структурована відповідь, чистий академічний стиль.

Базова оцінка: 4 / 5

З урахуванням -60%: 1.6 / 5

□ Питання 4

Порівняння `Even` і `LoanPayments`: ... обидві функції повертають результат,
... `LoanPayments` використовує змінну `power`, що зменшує повторюваність коду...

Аналіз:

- ✓ *Правильність*: Логічне й технічне порівняння точне.
- ✓ *Повнота*: Є згадка про модульність, повторне використання, масштабованість — тобто, всі ключові аспекти.
- □ *Ймовірність III*: Дуже висока (~80%) — стиль надто чистий, синтаксис без природних студентських неточностей, є ознаки GPT-стилю (“дозволяє ще більше зменшити повторюваність коду”, “масштабованість програми”).

Базова оцінка: 5 / 5

З урахуванням -60%: 2.0 / 5

□ Питання 5

Варіанти меню: від введення номера задачі до пунктів із навігацією клавішами...

Аналіз:

- ✓ *Правильність*: Всі три варіанти (простий, середній, складний) коректно описані.
- ✓ *Повнота*: Є логіка еволюції рішень.
- ⚠ *Мінус*: Можна було коротко вказати переваги/недоліки кожного варіанту.
- □ *Ймовірність III*: Середня (~50%) — відповіді логічні, але структуровані без зайвої "гладкості".

Базова оцінка: 4.5 / 5

Без зниження (бо III ймовірність лише помірна): 4.5 / 5

■ Підсумкова таблиця

№	Питання	Базова оцінка	Ймовірність III	Корекція	Остаточна
1	Формула з комісією	4.5	70%	-60%	1.8
2	Перевірка введення	5.0	40%	0%	5.0
3	Матриця плиток	4.0	60%	-60%	1.6
4	Порівняння функцій	5.0	80%	-60%	2.0
5	Меню завдань	4.5	50%	0%	4.5

◆ Середня підсумкова оцінка:

$$(1.8+5.0+1.6+2.0+4.5)/5=2.98 \quad (1.8 + 5.0 + 1.6 + 2.0 + 4.5) / 5 = 2.98$$

✓ **Загальна середня оцінка: 3.0 / 5**

(≈ 60% успішності після урахування ймовірного використання III)