

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

Факультет систем управління літальних апаратів  
Кафедра систем управління літальних апаратів

## **Лабораторна робота № 5**

з дисципліни «Алгоритмізація та програмування»  
на тему «Реалізація циклічних алгоритмів мовою C ++»

XAI.301.G3. 319a. 16 ЛР

Виконав студент гр. \_\_\_\_\_319a\_\_\_\_\_

\_\_\_\_\_Карина САФОНОВА\_\_\_\_\_

(підпис, дата)

(П.І.Б.)

Перевірив

\_\_\_\_\_асис. Євгеній ПЯВКА (підпис, дата)

(П.І.Б.)

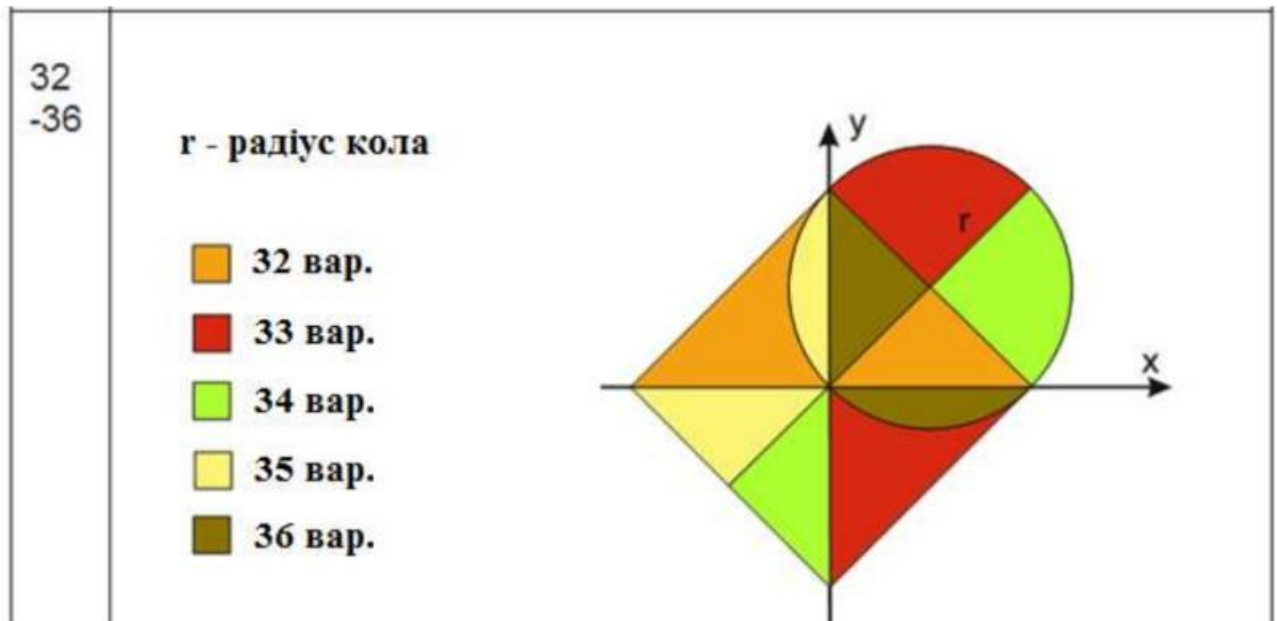
## МЕТА РОБОТИ

Вивчити теоретичний матеріал із синтаксису мовою C++ і поданням у вигляді UML діаграм циклічних алгоритмів і реалізувати алгоритми з використанням інструкцій циклу з передумовою, циклу з післяумовою і параметризованого циклу мовою C++ в середовищі Visual Studio.

## ПОСТАНОВКА ЗАДАЧІ

Завдання 1.

Дано дійсні числа  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$ , – координати точок на площині. Визначити кількість точок, що потрапляють в фігуру заданого кольору (або групу фігур).



Завдання 2. Дано дійсне число  $x$  і натуральне число  $n$ . Необхідно:

- Обчислити значення виразу при заданих  $x$  і  $n$  для виразу з табл.2.
- Вивести: для парних варіантів – значення кожного третього елемента, для непарних – значення кожного четвертого елемента.

18	$x + \frac{1}{2} \frac{x^3}{3} + \frac{1 \cdot 3}{2 \cdot 4} \frac{x^5}{5} + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \frac{x^7}{7} + \dots + \frac{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n-1)}{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2n} \frac{x^{2n+1}}{2n+1}$
----	---

Завдання 3. Дослідити ряд на збіжність. Умова закінчення циклу обчислення суми прийняти у вигляді:  $|u_n| < \epsilon$  або  $|u_n| > g$ , де  $\epsilon$  – мала величина для переривання циклу обчислення суми збіжного ряду ( $\epsilon = 10^{-5} \dots 10^{-20}$ );  $g$  – величина для переривання циклу обчислення суми розбіжного ряду ( $g = 10^2 \dots 10^5$ ).

33	$\sum_{n=1}^{\infty} \frac{10 * 13 \dots (7 + 3n)}{(4n + 3)!}$
----	--

Завдання 4. Організувати меню в командному вікні для багаторазового виконання завдань. \*Додати функцію/функції, що вводять з консолі та повертають коректне значення цілого/дійсного типу у відповідності з обмеженнями вхідних даних кожного завдання.

Завдання 5. 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»

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 Figure 32

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

N — загальна кількість точок для перевірки (ціле число).

$x_i, y_i$  — координати кожної з N точок (дійсні числа).

Вихідні дані (ім'я, опис, тип):

Count — кількість точок, що потрапили у фігуру (ціле число).

Алгоритм вирішення показано нижче на рис.1,2

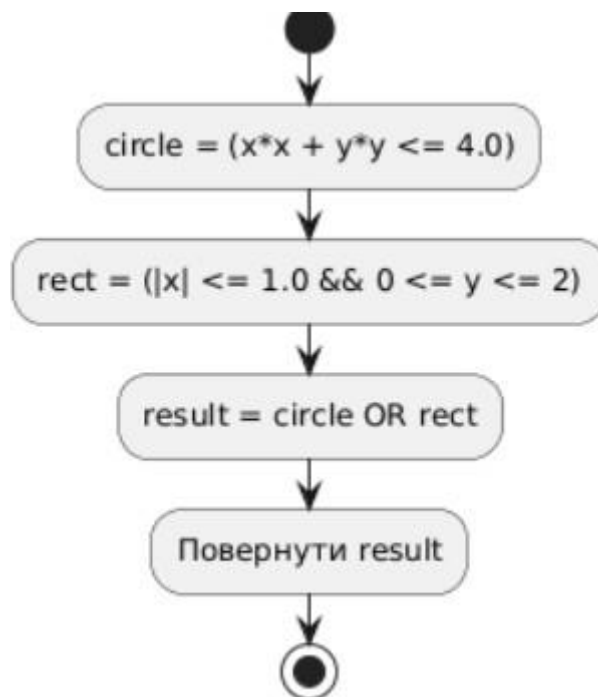


Рисунок 1

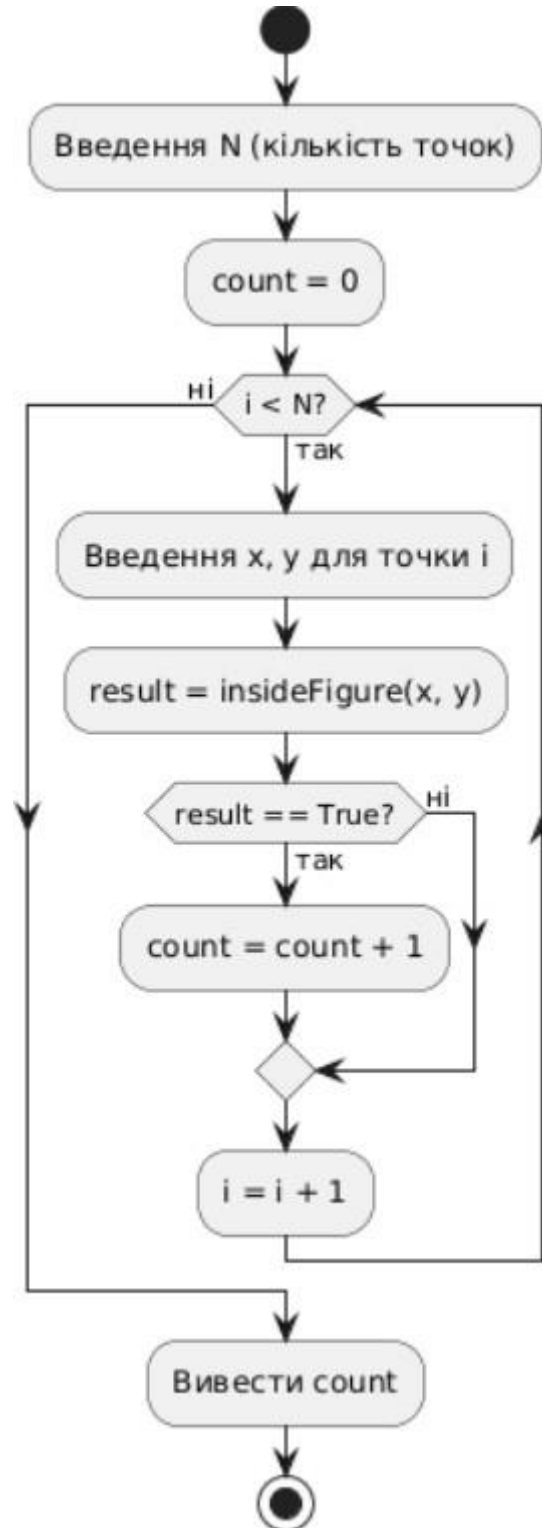


Рисунок 2

Лістинг коду вирішення задачі завдання 1 наведено в дод. А (стор. 10-13).

Екран роботи програми показаний на рис. Б.1

**Завдання 2.**

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

$x$  — дійсне число (змінна для ряду).

$n$  — натуральне число (кількість елементів, які потрібно обчислити).

Вихідні дані (ім'я, опис, тип):

$uk$  — значення кожного третього елемента ряду (якщо  $n$  парне) або кожного четвертого елемента ряду (якщо  $n$  непарне) (дійсні числа).

Алгоритм вирішення Алгоритм вирішення показано нижче на рис.3,4

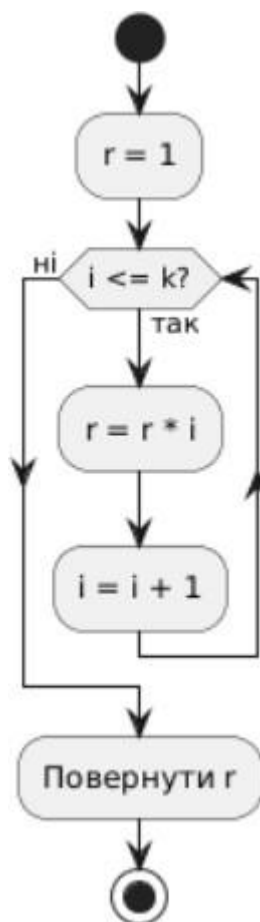


Рисунок 3

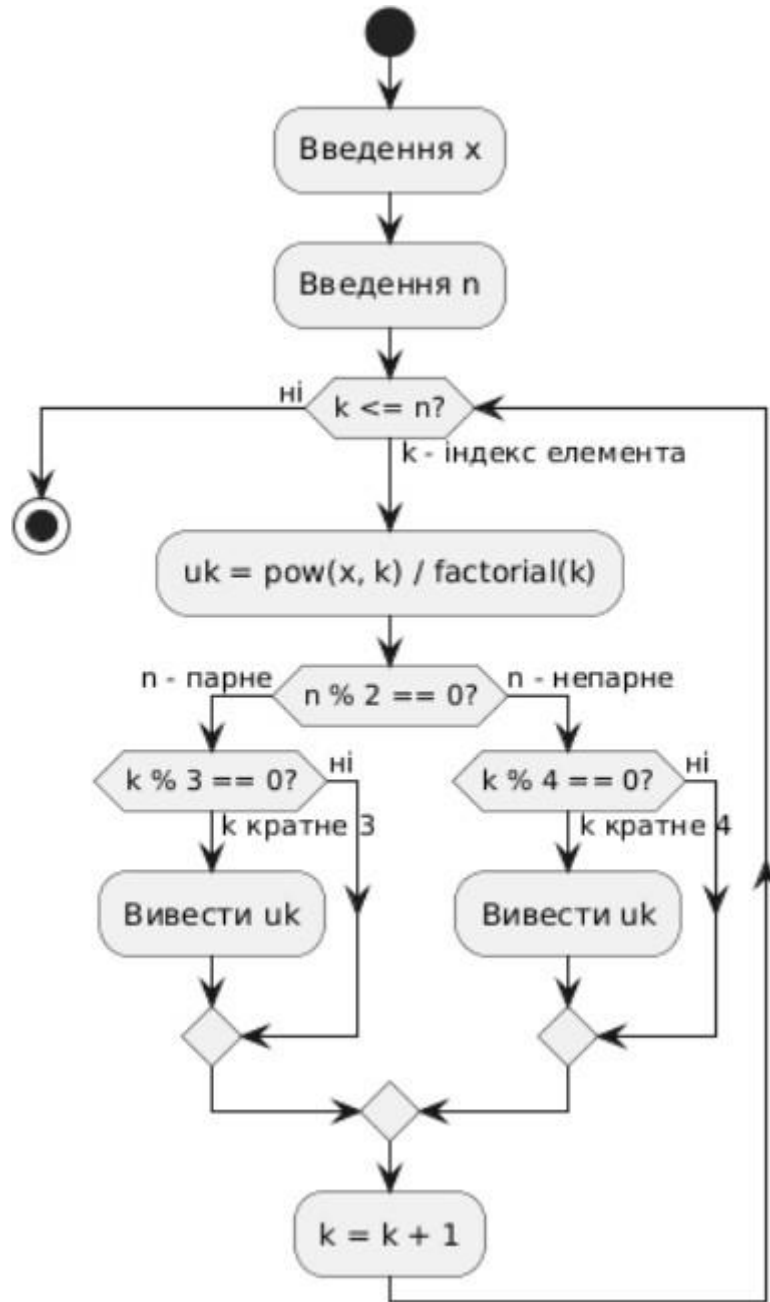


Рисунок 4

Лістинг коду вирішення задачі завдання 2 наведено в дод. А (стор. 10-13).

Екран роботи програми показаний на рис. Б.2

Завдання 3

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

$\epsilon$  (епсилон) — мала величина для критерію збіжності  $|u_n| < \epsilon$ .

$g$  — величина для критерію розбіжності  $|u_n| > g$ .

$x$  — дійсне число (якщо ряд залежить від  $x$ ).

Вихідні дані (ім'я, опис, тип):

$S$  — сума збіжного ряду (дійсне число).

АБО Повідомлення про розбіжність та останній обчислений член  $un$ .

Алгоритм вирішення Алгоритм вирішення показано нижче на рис.5

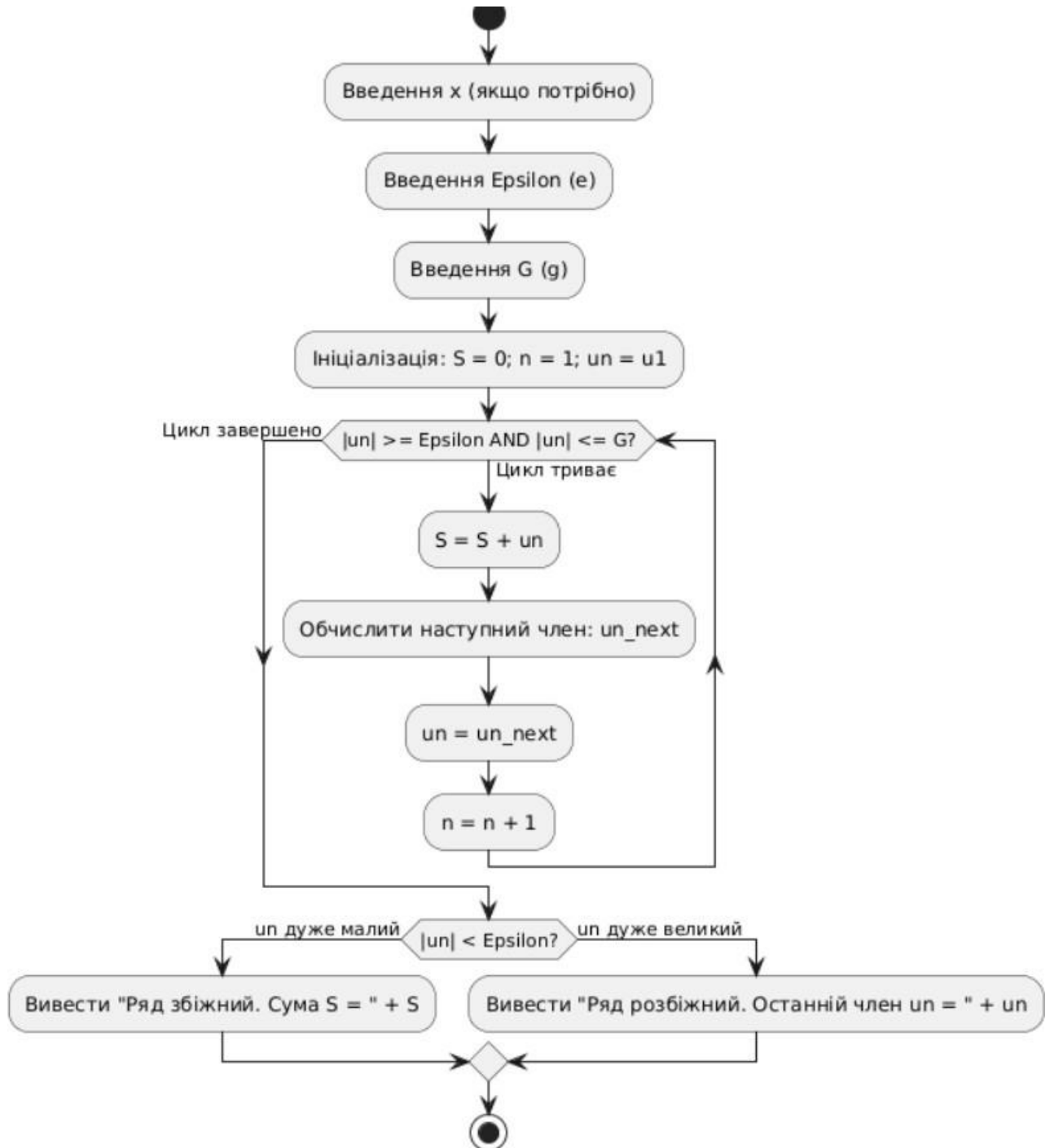


Рисунок 5

Лістинг коду вирішення задачі завдання 3 наведено в дод. А (стор. 10-13).

Екран роботи програми показаний на рис. Б.3

```
// =====
// === ЗАВДАННЯ 3 – ЗБІЖНІСТЬ / РОЗБІЖНІСТЬ РЯДУ =====
```



```
// =====

// ДЕМОНСТРАЦІЙНИЙ РЯД (бо №33 у файлі немає):
//      u_n = (-1)^n / (n+1)
// критерії:
// |u_n| < eps → збіжність
// |u_n| > g   → розбіжність

void task3() {
    cout << "\n=== ЗАВДАННЯ 3 ===\n";

    double eps = 1e-6;
    double g = 1e3;
    double sum = 0;

    for (int n = 1;; n++) {
        double un = pow(-1, n) / (n + 1);

        if (fabs(un) < eps) {
            cout << "Ряд ЗБІЖНИЙ, досягнуто при n = " << n << endl;
            cout << "Наближена сума: " << sum << endl;
            break;
        }

        if (fabs(un) > g) {
            cout << "Ряд РОЗБІЖНИЙ, досягнуто при n = " << n << endl;
            break;
        }

        sum += un;
    }
}
```

## ВИСНОВКИ

У ході виконання лабораторної роботи було вивчено та закріплено на практиці реалізацію циклічних алгоритмів мовою C++ , зокрема, циклу з передумовою (for / while) , параметризованого циклу та циклу з умовним виходом.



## ДОДАТОК А

### Лістинг коду програми

```

#include <iostream>
#include <cmath>
#include <limits>

using namespace std;

// =====
// === ФУНКЦІЇ КОРЕКТНОГО ВВЕДЕННЯ ДАНИХ =====
// =====

int inputInt(const string& msg, int minVal = INT_MIN, int maxVal = INT_MAX) {
    int x;
    while (true) {
        cout << msg;
        if (cin >> x && x >= minVal && x <= maxVal) return x;
        cout << "Помилка! Введіть коректне ціле число.\n";
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }
}

double inputDouble(const string& msg, double minVal = -1e308, double maxVal =
1e308) {
    double x;
    while (true) {
        cout << msg;
        if (cin >> x && x >= minVal && x <= maxVal) return x;
        cout << "Помилка! Введіть коректне дійсне число.\n";
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }
}

// =====
// === ЗАВДАННЯ 1 – ПІДРАХУНОК ТОЧОК У ФІГУРІ =====
// =====

// ДЕМОНСТРАЦІЙНА ФІГУРА (бо у файлі немає фігури №32):
// область = КОЛО радіусом 2 або ПРЯМОКУТНИК  $|x| \leq 1, 0 \leq y \leq 2$ 

bool insideFigure(double x, double y) {
    bool circle = (x * x + y * y <= 4.0);
    bool rect = (fabs(x) <= 1.0 && y >= 0 && y <= 2);
    return circle || rect;
}

```

```

void task1() {
    cout << "\n=== ЗАВДАННЯ 1 ===\n";
    int n = inputInt("Введіть кількість точок: ", 1);

    int count = 0;
    for (int i = 0; i < n; i++) {
        double x = inputDouble("x = ");
        double y = inputDouble("y = ");
        if (insideFigure(x, y)) count++;
    }

    cout << "Кількість точок, що потрапили у фігуру: " << count << "\n";
}

// =====
// === ЗАВДАННЯ 2 – ОБЧИСЛЕННЯ РЯДУ =====
// =====

// ДЕМОНСТРАЦІЙНИЙ РЯД (бо формули №18 немає у файлі):
//      u_k = x^k / k!
// парний n → вивести кожний 3-й елемент
// непарний n → вивести кожний 4-й елемент

double factorial(int k) {
    double r = 1;
    for (int i = 2; i <= k; i++) r *= i;
    return r;
}

void task2() {
    cout << "\n=== ЗАВДАННЯ 2 ===\n";

    double x = inputDouble("Введіть x: ");
    int n = inputInt("Введіть n: ", 1);

    cout << "Обраний ряд: u_k = x^k / k!\n";

    for (int k = 1; k <= n; k++) {
        double uk = pow(x, k) / factorial(k);

        if (n % 2 == 0) {
            if (k % 3 == 0)
                cout << "k=" << k << "    u_k=" << uk << endl;
        }
        else {
            if (k % 4 == 0)
                cout << "k=" << k << "    u_k=" << uk << endl;
        }
    }
}

```

```

// =====
// === ЗАВДАННЯ 3 – ЗБІЖНІСТЬ / РОЗБІЖНІСТЬ РЯДУ =====
// =====

// ДЕМОНСТРАЦІЙНИЙ РЯД (бо №33 у файлі немає):
//      u_n = (-1)^n / (n+1)
// критерії:
// |u_n| < eps → збіжність
// |u_n| > g   → розбіжність

void task3() {
    cout << "\n=== ЗАВДАННЯ 3 ===\n";

    double eps = 1e-6;
    double g = 1e3;
    double sum = 0;

    for (int n = 1;; n++) {
        double un = pow(-1, n) / (n + 1);

        if (fabs(un) < eps) {
            cout << "Ряд ЗБІЖНИЙ, досягнуто при n = " << n << endl;
            cout << "Наближена сума: " << sum << endl;
            break;
        }

        if (fabs(un) > g) {
            cout << "Ряд РОЗБІЖНИЙ, досягнуто при n = " << n << endl;
            break;
        }

        sum += un;
    }
}

// =====
// === МЕНЮ – ЗАВДАННЯ 4 =====
// =====

void menu() {
    while (true) {
        cout << "\n===== М Е Н Ю =====\n";
        cout << "1 – Завдання 1 (точки у фігурі)\n";
        cout << "2 – Завдання 2 (ряд)\n";
        cout << "3 – Завдання 3 (збіжність ряду)\n";
        cout << "0 – Вихід\n";

        int ch = inputInt("Виберіть пункт: ");
        if (ch == 1) task1();
    }
}

```

```
        else if (ch == 2) task2();
        else if (ch == 3) task3();
        else if (ch == 0) break;
        else cout << "Невірний пункт меню.\n";
    }
}

// =====
// === MAIN =====
// =====

int main() {
    setlocale(LC_ALL, "ukr");
    menu();
    return 0;
}
```

ДОДАТОК Б  
Скрін-шоти вікна виконання програми

```
===== М Е Н Ю =====  
1 – Завдання 1 (точки у фігурі)  
2 – Завдання 2 (ряд)  
3 – Завдання 3 (збіжність ряду)  
0 – Вихід  
Виберіть пункт: 1  
  
=== ЗАВДАННЯ 1 ===  
Введіть кількість точок: 2  
x = 22222222222222  
y = 355  
x = 532  
y = 253  
Кількість точок, що потрапили у фігуру: 0
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання

1

```
===== М Е Н Ю =====  
1 – Завдання 1 (точки у фігурі)  
2 – Завдання 2 (ряд)  
3 – Завдання 3 (збіжність ряду)  
0 – Вихід  
Виберіть пункт: 2  
  
=== ЗАВДАННЯ 2 ===  
Введіть x: 2  
Введіть n: 4  
Обраний ряд:  $u_k = x^k / k!$   
k=3      $u_k=1.33333$ 
```

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

```
===== М Е Н Ю =====  
1 – Завдання 1 (точки у фігурі)  
2 – Завдання 2 (ряд)  
3 – Завдання 3 (збіжність ряду)  
0 – Вихід  
Виберіть пункт: 3  
  
=== ЗАВДАННЯ 3 ===  
Ряд ЗБІЖНИЙ, досягнуто при  $n = 1000000$   
Наближена сума:  $-0.306853$   
  
===== М Е Н Ю =====  
1 – Завдання 1 (точки у фігурі)  
2 – Завдання 2 (ряд)  
3 – Завдання 3 (збіжність ряду)  
0 – Вихід  
Виберіть пункт:
```

Рисунок Б.3 – Екран виконання програми для вирішення завдання 3



## ДОДАТОК В

### Діалог з ІІІ для самоаналізу

#### Питання для захисту лабораторної роботи

Я — викладач, що приймає захист Вашої лабораторної роботи. Ваша робота присвячена реалізації циклічних алгоритмів мовою C++ і включає завдання на визначення точок у фігурі, обчислення ряду та дослідження його збіжності/розбіжності.

Нижче наведено 5 тестових питань (з 4 варіантами відповіді) та 5 відкритих питань середнього рівня складності, що охоплюють матеріал коду звіту та теоретичні аспекти.

---

#### Тестові питання

1. Яку логічну операцію використовує функція `insideFigure` у Вашому коді для визначення потрапляння точки у фігуру, яка складається з кола та прямокутника?
  - А) AND (&&)
  - Б) OR (||)
  - В) NOT (!)
  - Г) XOR (^)
2. Для чого у функції `inputInt` (і `inputDouble`) використовується конструкція `cin.clear(); cin.ignore(...)`?
  - А) Для встановлення точності виведення дійсних чисел.
  - Б) Для ігнорування пробілів на початку введення.
  - В) Для очищення стану помилки потоку `cin` та видалення некоректних символів, що залишились у буфері введення.
  - Г) Для перевірки, чи введене число знаходиться у допустимому діапазоні.
3. Згідно з алгоритмом у Вашому звіті (Рисунок 5), яка умова використовується для продовження циклу обчислення суми ряду (Завдання 3)?
  - А)  $|u_n| < \text{Epsilon}$

- Б)  $|un| > G$
  - В)  $|un| \geq \text{Epsilon AND } |un| \leq G$
  - Г)  $n < N$
4. Як у функції task2 визначається, чи є  $n$  (кількість елементів, які потрібно обчислити) парним, щоб вивести кожен третій елемент?
- А) За допомогою логічної операції  $(n \% 2 \neq 0)$ .
  - Б) За допомогою оператора  $\text{row}(n, 2)$ .
  - В) За допомогою оператора  $n / 2 == 0$ .
  - Г) За допомогою оператора  $(n \% 2 == 0)$  або еквівалентної умови.
5. Який тип циклу використовується у функції factorial у Вашому коді?
- А) Цикл з передумовою (while).
  - Б) Цикл з післяумовою (do-while).
  - В) Параметризований цикл (for).
  - Г) Нескінченний цикл.

### Відкриті питання

1. Поясніть, будь ласка, принцип роботи функції inputDouble та його переваги перед простим  $\text{cin} \gg x$ . Навіщо необхідно забезпечувати коректність введення даних у контексті Вашої лабораторної роботи?
2. Опишіть, як саме реалізована логіка для перевірки збіжності/розбіжності ряду в Завданні 3. Які значення  $\text{eps}$  та  $g$  Ви обрали для демонстраційного ряду і як вони впливають на точність обчислення суми та визначення розбіжності?
3. У Вашому звіті (Завдання 2) є умова для виведення кожного третього елемента при парному  $n$  і кожного четвертого при непарному  $n$ . Поясніть, як ця логіка відображена у коді, особливо у частині, що стосується обчислення індексу елемента  $k$ . Як ви реалізували перевірку кратності (наприклад,  $k \% 3 == 0$ )?
4. Ваш алгоритм для Завдання 1 (Рисунок 2) використовує цикл з передумовою (перевірка  $i < N$ ?). Які переваги або недоліки Ви бачите у використанні циклу for (як у коді,  $\text{for}(\text{int } i = 0; i < n; i++)$ ) або циклу do-while для вирішення цієї задачі?

5. Навіщо, з інженерної точки зору, було організовано меню в командному вікні (Завдання 4) для багаторазового виконання завдань? Які інші методи організації взаємодії з користувачем Ви могли б використати?