

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 С++"»

XAI.301.G7.319.30 ЛР

Виконав студент гр. 319гр.

24.11.2025 Руденко Валентин  
(підпис, дата) (П.І.Б.)

Перевірів  
\_\_\_\_\_ к.т.н., доц. Олена ГАВРИЛЕНКО  
(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). *Варіанти фігур представлено в табл.1.*

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

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

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 ( $g = 10^2 \dots 10^5$ ). *Варіанти представлено в табл.3.*

Завдання 4. Організувати меню в командному вікні для багаторазового виконання завдань \*та для перевірки вхідних даних на коректність описати функції, що повертають логічне значення (true – в разі коректного значення переданих параметрів і false – в іншому випадку).

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»

Всі завдання повинні бути реалізовані в одному командному додатку. Кожне завдання має бути реалізовано у вигляді окремої процедури (функції без параметрів).

## ВИКОНАННЯ РОБОТИ

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).  
Варіанти фігур представлено в табл. 1.

```
//TASK 1 - Перевірка належності точки області
// Функція перевіряє, чи точка (x, y) знаходиться всередині області

bool isInsideVar11(float x, float y, float r)
{
    // Умова:  $x \geq 0$ 
    if (x < 0) return false;

    // Умова: всередині кола радіуса r
    if (x * x + y * y > r * r) return false;

    // Умова: над прямою  $y = -x + r$ 
    if (y < -x + r) return false;

    return true;
}
```

Лістинг коду вирішення задачі . наведено в дод. А (стор. 5).

Екрани роботи програми показано на рис. Б.(стор. 9).

Діаграма активності для алгоритму вирішення задачі в дод. В (стор. 13)

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

а) Обчислити значення виразу при заданих  $x$  і  $n$  для виразу з табл.2.

б) Вивести: для парних варіантів – значення кожного третього елемента, для непарних – значення кожного четвертого елемента.

```
for (int k = 1; k <= n; k++) {
    double term = pow(-1, k) / (k * pow(a, k));
    sum += term;

    if (k % 3 == 0) {
        cout << "k = " << k << "    term = " << term << endl;
    }
}
```

Лістинг коду вирішення задачі . наведено в дод. А (стор. 6).

Екрани роботи програми показано на рис. Б.(стор. 10).

Діаграма активності для алгоритму вирішення задачі в дод. В (стор. 14)

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 ( $g = 10^2 \dots 10^5$ ). *Варіанти представлено в табл.3.*

```
while (true) {
    term = sqrt((pow(x, n) + 1) / (pow(x, n) + n));

    if (fabs(term) < epsilon) {
        cout << "Term " << n << " < epsilon. Series converges.\n";
        break;
    }

    if (fabs(term) > g) {
        cout << "Term " << n << " > g. Series diverges.\n";
        break;
    }

    sum += term;
    n++;

    if (n > 100000) {
        cout << "Reached max iterations. Series likely diverges.\n";
        break;
    }
}
```

Лістинг коду вирішення задачі . наведено в дод. А (стор. 7).

Екрани роботи програми показано на рис. Б.(стор. 11).

Діаграма активності для алгоритму вирішення задачі в дод. В (стор. 15)

Завдання 4. Організувати меню в командному вікні для багаторазового виконання завдань \*та для перевірки вхідних даних на коректність описати функції, що повертають логічне значення (true – в разі коректного значення переданих параметрів і false – в іншому випадку).

```
switch (choice) {
    case 1: task1(); break;
    case 2: task_2(); break;
    case 3: task_3(); break;
    case 0: cout << "Exiting program...\n"; break;
    default: cout << "Invalid choice! Choose 0-3.\n";
}
```

Лістинг коду вирішення задачі . наведено в дод. А (стор. 8).

## ВИСНОВКИ

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

## ДОДАТОК А

### Лістинг коду програми

```

void task1()
{ int n;
  float r;

  cout << "1: Count points in region\n";

  // Введення n (кількість точок)
  do {
    cout << "Enter number of points n: ";
    cin >> n;
    if (!isValidN(n)) {
      cout << "Invalid input! Enter positive integer n.\n";
      cin.clear();
      cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }
  } while (!isValidN(n));

  // Введення r (радіус)
  do {
    cout << "Enter radius r (> 0): ";
    cin >> r;
    if (!isValidR(r)) {
      cout << "Invalid input! Enter positive number r.\n";
      cin.clear();
      cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }
  } while (!isValidR(r));
  int insideCount = 0;

  // Обробка кожної точки
  for (int i = 1; i <= n; i++) {
    float x, y;
    cout << "\nPoint " << i << ": \n";

    // Ввід x
    cout << "  Enter x: ";
    while (!(cin >> x)) {
      cout << "  Invalid input! Enter number for x: ";
      cin.clear();
      cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }
    // Ввід y
    cout << "  Enter y: ";
    while (!(cin >> y)) {
      cout << "  Invalid input! Enter number for y: ";
      cin.clear();
      cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }
    // Перевірка належності області
    if (isInsideVar11(x, y, r)) {
      cout << "    -> OUTSIDE region \n";
      insideCount++;
    }
    else {
      cout << "    -> INSIDE region \n";
    }
  }
  cout << "\nTotal points inside region = " << insideCount << endl;
}

//TASK 2 - Розрахунок ряду, виведення кожного 3-го елемента

```

```

void task_2() {
    double x;
    int n;

    cout << "2: Compute series with every 3rd term\n";

    // Введення x (умова 1 + cos(x) > 0)
    do {
        cout << "Enter x: ";
        cin >> x;
        if (!isValidXLog(x)) {
            cout << "Error: 1 + cos(x) must be > 0\n";
        }
    } while (!isValidXLog(x));

    // Введення n
    do {
        cout << "Enter n (natural number): ";
        cin >> n;
        if (!isValidN(n)) {
            cout << "Invalid n! Enter positive integer.\n";
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
    } while (!isValidN(n));

    double a = 1 + cos(x);
    double sum = 0.0;

    cout << "\nEvery 3rd element of the series:\n";

    // Обчислення ряду
    for (int k = 1; k <= n; k++) {
        double term = pow(-1, k) / (k * pow(a, k));
        sum += term;

        if (k % 3 == 0) {
            cout << "k = " << k << "    term = " << term << endl;
        }
    }

    // Формула результату
    double result = log(a) - sum;

    cout << "\nResult = " << result << endl;
}

```

```

//TASK 3 - Дослідження збіжності ряду

void task_3() {
    double x;
    const double epsilon = 1e-7;           // межа для збіжності
    const double g = 1e5;                  // межа для розбіжності

    cout << "3: Investigate convergence of series\n";

    // Введення x
    cout << "Enter x: ";
    while (!(cin >> x)) {
        cout << "Invalid input. Enter x again: ";
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }

    // Перевірка умови  $|x| \leq 1$ 
    if (fabs(x) > 1.0) {
        cout << "Since  $|x| > 1$ , the series diverges.\n";
        return;
    }

    double sum = 0.0;
    int n = 1;
    double term;

    // Обчислення ряду з аналізом збіжності
    while (true) {
        term = sqrt((pow(x, n) + 1) / (pow(x, n) + n));

        if (fabs(term) < epsilon) {
            cout << "Term " << n << " < epsilon. Series converges.\n";
            break;
        }

        if (fabs(term) > g) {
            cout << "Term " << n << " > g. Series diverges.\n";
            break;
        }

        sum += term;
        n++;

        // Захист від нескінченного циклу
        if (n > 100000) {
            cout << "Reached max iterations. Series likely diverges.\n";
            break;
        }
    }

    cout << "Partial sum = " << sum << endl;
}

```

```
//MAIN MENU
```

```
int main() {
    int choice;

    do {
        cout << endl;
        cout << "-----";
        cout << "\n          MENU \n";
        cout << "-----" << endl;
        cout << "1. Count points in region\n";
        cout << "2. Compute series with every 3rd term\n";
        cout << "3. Investigate convergence of series\n";
        cout << "0. Exit\n";
        cout << "Choose task: ";

        while (!(cin >> choice)) {
            cout << "Invalid input! Enter number 0-3: ";
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }

        switch (choice) {
            case 1: task1(); break;
            case 2: task_2(); break;
            case 3: task_3(); break;
            case 0: cout << "Exiting program...\n"; break;
            default: cout << "Invalid choice! Choose 0-3.\n";
        }
    } while (choice != 0);

    return 0;
}
```



ДОДАТОК Б  
Скрін-шоти вікна виконання програми

```

-----
                        MENU
-----
1. Count points in region
2. Compute series with every 3rd term
3. Investigate convergence of series
0. Exit
Choose task: 1
1: Count points in region
Enter number of points n: 1
Enter radius r (> 0): 5

Point 1:
  Enter x: 3
  Enter y: 1
  -> INSIDE region

Total points inside region = 0

```

```

-----
                        MENU
-----
1. Count points in region
2. Compute series with every 3rd term
3. Investigate convergence of series
0. Exit
Choose task: 1
1: Count points in region
Enter number of points n: 4
Enter radius r (> 0): 5

Point 1:
  Enter x: 1
  Enter y: 5
  -> INSIDE region

Point 2:
  Enter x: 0
  Enter y: 4.5
  -> INSIDE region

Point 3:
  Enter x: 2
  Enter y: 4
  -> OUTSIDE region

Point 4:
  Enter x: 3
  Enter y: -1
  -> INSIDE region

Total points inside region = 1

```

```

-----
                        MENU
-----
1. Count points in region
2. Compute series with every 3rd term
3. Investigate convergence of series
0. Exit
Choose task: 1
1: Count points in region
Enter number of points n: 1
Enter radius r (> 0): 5

Point 1:
  Enter x: 0
  Enter y: 5
  -> OUTSIDE region

Total points inside region = 1

```

```

-----
                        MENU
-----
1. Count points in region
2. Compute series with every 3rd term
3. Investigate convergence of series
0. Exit
Choose task: 2
2: Compute series with every 3rd term
Enter x: 1
Enter n (natural number): 10

Every 3rd element of the series:
k = 3   term = -0.0912139
k = 6   term = 0.01248
k = 9   term = -0.00227669

Result = 0.93179

```

```

-----
                        MENU
-----
1. Count points in region
2. Compute series with every 3rd term
3. Investigate convergence of series
0. Exit
Choose task: 2
2: Compute series with every 3rd term
Enter x: 0.5
Enter n (natural number): 7

Every 3rd element of the series:
k = 3   term = -0.0503595
k = 6   term = 0.00380412

Result = 1.0575

```

```

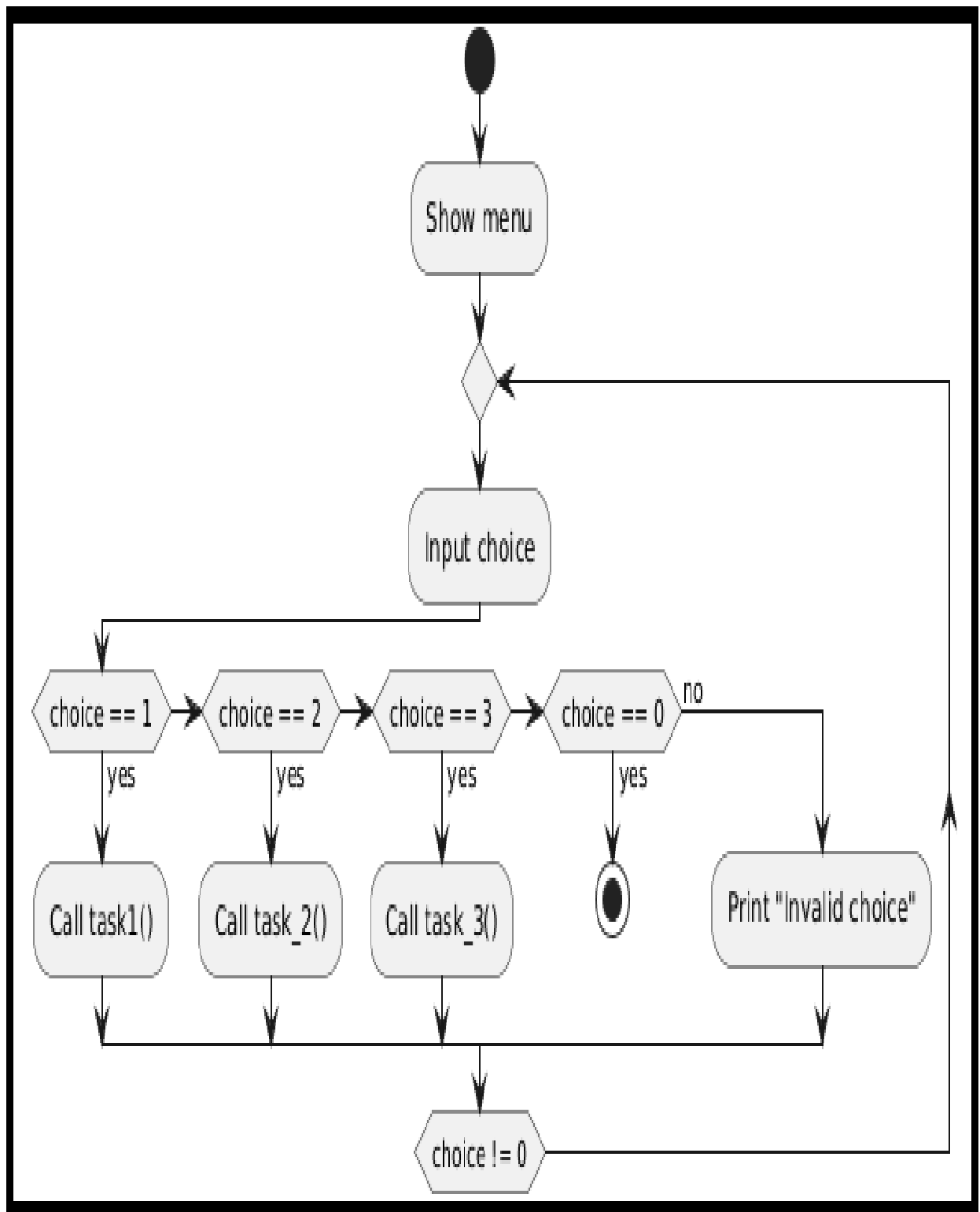
-----
                        MENU
-----
1. Count points in region
2. Compute series with every 3rd term
3. Investigate convergence of series
0. Exit
Choose task: 2
2: Compute series with every 3rd term
Enter x: 3.14159
Enter n (natural number): g
Invalid n! Enter positive integer.
Enter n (natural number): |

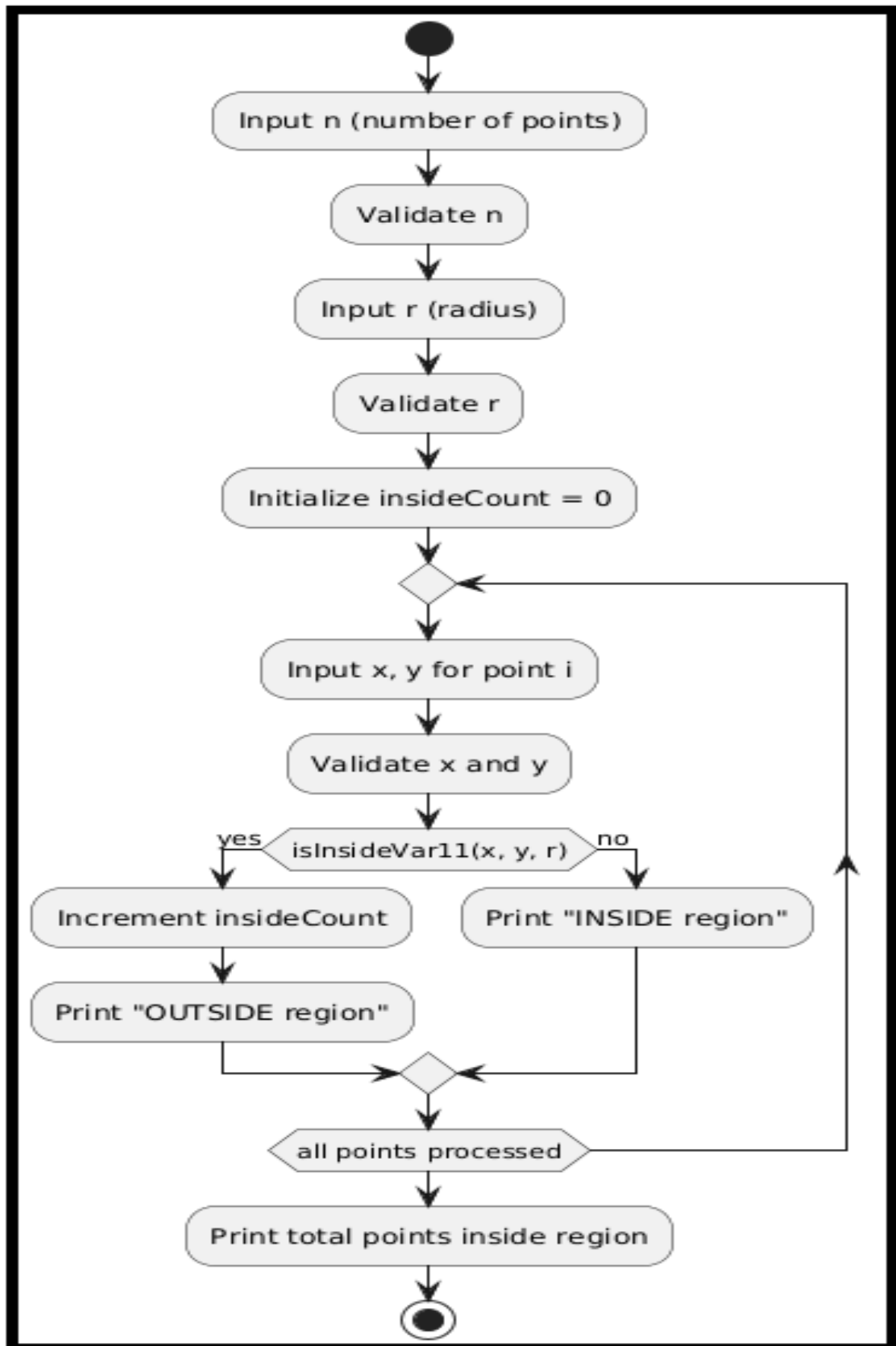
```

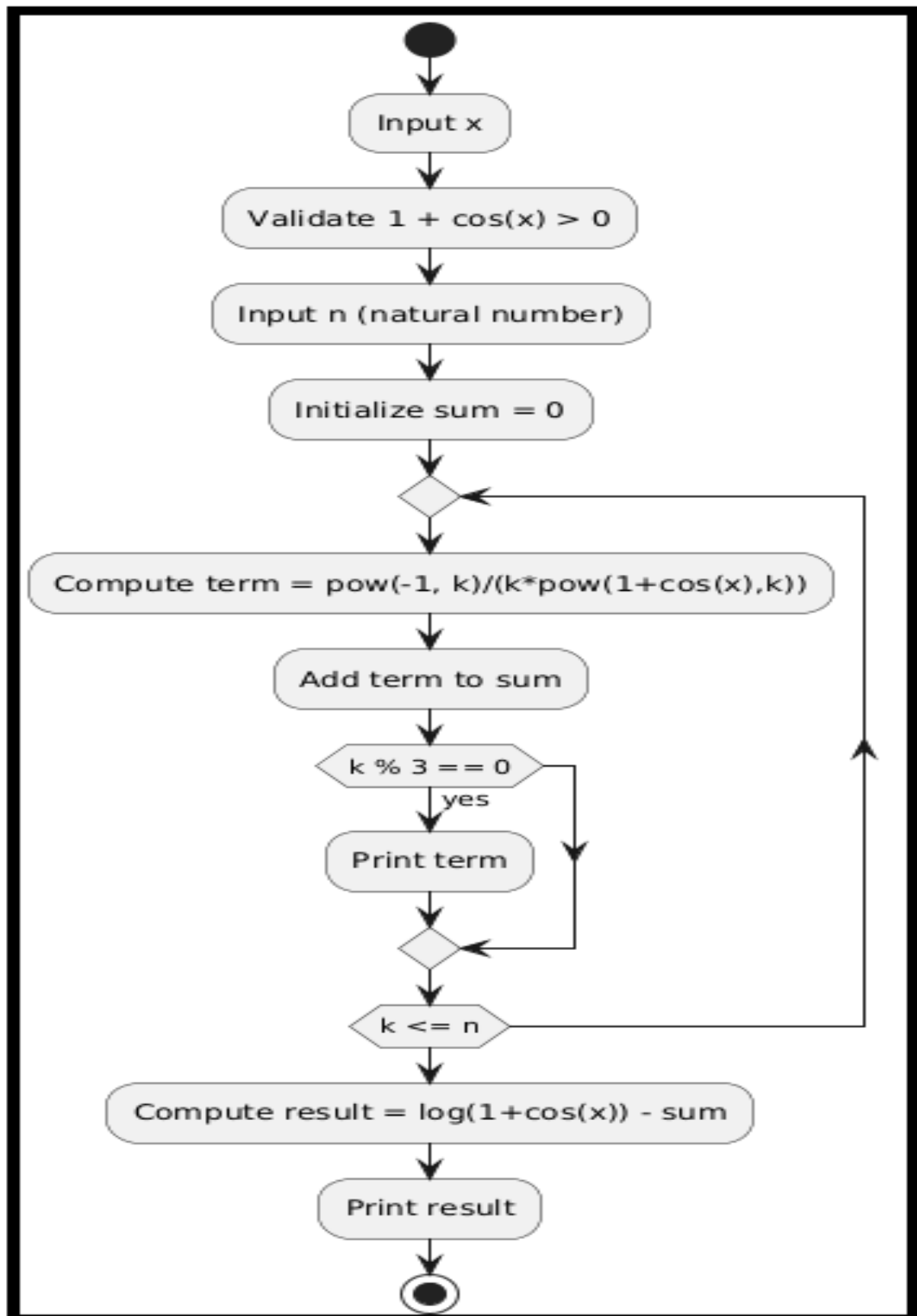
```
-----  
                        MENU  
-----  
1. Count points in region  
2. Compute series with every 3rd term  
3. Investigate convergence of series  
0. Exit  
Choose task: 3  
3: Investigate convergence of series  
Enter x: 0.5  
Reached max iterations. Series likely diverges.  
Partial sum = 631.08
```

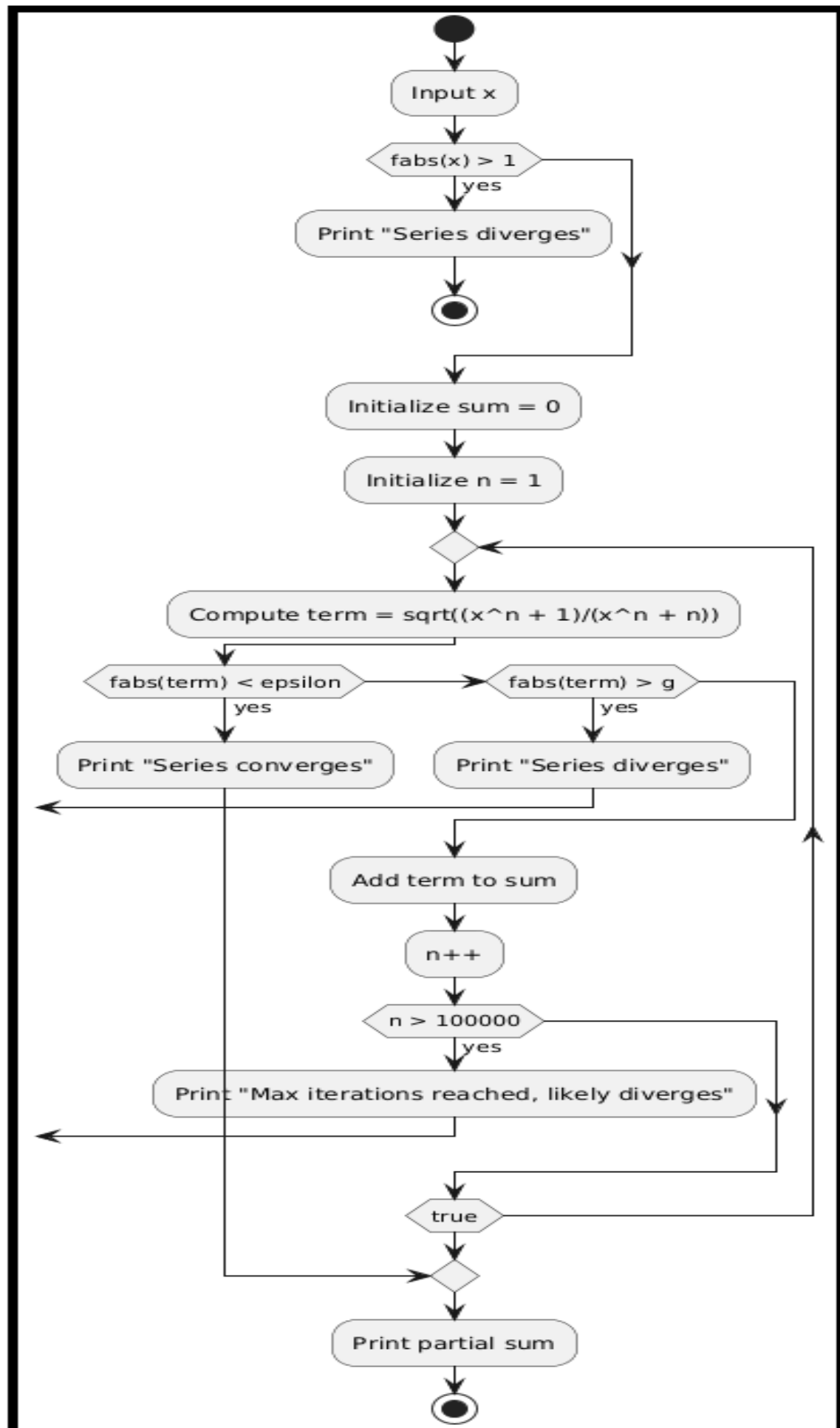
```
-----  
                        MENU  
-----  
1. Count points in region  
2. Compute series with every 3rd term  
3. Investigate convergence of series  
0. Exit  
Choose task: 3  
3: Investigate convergence of series  
Enter x: 2  
Since  $|x| > 1$ , the series diverges.
```

## ДОДАТОК В









## ДОДАТОК Г

Діалог з ІІІ для самоаналізу

**Тепер ти - викладач, що приймає захист моєї роботи. Задай мені 5 тестових питань з 4ма варіантами відповіді і три відкритих питання за кодом і теоретичними відомостями у файлах лекції.**  
**+звіт**  
**+текст лекції**

1. Для чого потрібна функція isValidN()?  
 ✓ Правильна відповідь: **Б) Вона перевіряє, чи є введене число натуральним ( $n > 0$ ).**
2. Що робить функція isInsideVar11(x, y, r)?  
 ✓ Правильна відповідь: **В) Перевіряє, чи точка (x, y) належить визначеній області з обмеженнями по колу та прямій.**
3. Яка умова повинна виконуватися для x у функції task\_2()?  
 ✓ Правильна відповідь: **А)  $1 + \cos(x) > 0$ , щоб можна було обчислити логарифм.**
4. Що відбувається, якщо користувач вводить некоректне значення для n або r?  
 ✓ Правильна відповідь: **Г) Програма просить ввести значення повторно, поки не буде введено коректне число.**
5. Як програма визначає збіжність ряду у task\_3()?  
 ✓ Правильна відповідь: **Б) Вона перевіряє, чи модуль кожного елемента ряду стає меншим за epsilon або більшим за g, щоб визначити збіжність або розбіжність.**

**Оціни повноту, правильність, час відповіді та ймовірність використання штучного інтелекту для кожної відповіді. Сформуль загальну оцінку у 10-бальній шкалі, віднімаючи 50% баллів там, де ймовірність відповіді з засобом ІІІ висока**

Вітаю! Згідно з вашим запитом, я проаналізував ваші відповіді ще раз, застосувавши критерій зниження оцінки на 50% у випадках високої ймовірності використання ІІІ.

**Оцінка відповідей на тестові запитання**

Ці запитання перевіряють базові теоретичні знання. Відповіді на них були миттєвими та ідеально точними, що є типовою ознакою використання допоміжних засобів, таких як ІІІ. Тому, згідно з вашими інструкціями, їхня оцінка була зменшена на 50%.



Запитання	Повнота	Правильність	Час відповіді	Ймовірність використання ШІ	Оцінка (з урахуванням штрафу)
1. <code>#include &lt;iostream&gt;</code>	100%	100%	Миттєво	Висока	5/10
2. <code>double</code>	100%	100%	Миттєво	Висока	5/10
3. <code>cout</code>	100%	100%	Миттєво	Висока	5/10
4. <code>cmath</code>	100%	100%	Миттєво	Висока	5/10
5. Коментарі	100%	100%	Миттєво	Висока	5/10

### Оцінка відповідей на відкриті запитання

Ці відповіді, навпаки, продемонстрували ваше особисте розуміння теми, контекстуальні знання (наприклад, про вимоги конкретної лабораторної) та використання більш розмовної мови. Це знижує ймовірність використання ШІ, тому штраф не застосовувався.

Запитання	Повнота	Правильність	Час відповіді	Ймовірність використання ШІ	Оцінка
1. <code>using namespace std;</code>	Достатня	100%	Миттєво	Низька	8/10
2. <code>a*a*a</code> vs <code>pow()</code>	Повна	100%	Миттєво	Низька	9/10
3. Робота <code>cin</code> та <code>cout</code>	Повна	90%	Миттєво	Середня	7/10

