

```

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

// Завдання 1: Функції для перевірки точок

// Функція перевіряє, чи належить точка (x, y) заданому трикутнику
bool point_in_triangle(double x, double y, double r) {
    // Умови: точка повинна бути в другій чверті, всередині трикутника і в колі радіуса r
    return x <= 0 && y >= 0 && y <= -x && x * x + y * y <= r * r;
}

// Функція перевіряє, чи належить точка (x, y) сектору кола
bool point_in_sector(double x, double y, double r) {
    // Умови: точка в першій чверті, всередині або на межі кола, вище прямої x + y = r
    return x >= 0 && y >= 0 && x * x + y * y <= r * r && x + y >= r;
}

// Функція для виконання Завдання 1
void task1() {
    double r; // Радіус кола та трикутника
    int n;    // Кількість точок

    // Введення радіуса
    cout << "Enter the radius r: ";
    cin >> r;
    if (!cin || r <= 0) { // Перевірка валідності введеного радіуса
        cout << "Radius must be numeric and positive!" << endl;
        return;
    }

    // Введення кількості точок
    cout << "Enter the number of points n: ";
    cin >> n;
    if (!cin || n <= 0) { // Перевірка валідності кількості точок
        cout << "Number of points must be a positive integer!" << endl;
        return;
    }

    int count_in_triangle = 0; // Лічильник точок у трикутнику
    int count_in_sector = 0;  // Лічильник точок у секторі кола

    // Цикл для обробки координат точок
    for (int i = 1; i <= n; i++) {
        double x, y;
        cout << "Enter the coordinates of point " << i << " (x, y): ";
        cin >> x >> y;
        if (!cin) { // Перевірка валідності введення координат
            cout << "Coordinates must be numeric!" << endl;
            return;
        }
    }
}

```

```

// Перевірка належності точки до трикутника
if (point_in_triangle(x, y, r)) count_in_triangle++;

// Перевірка належності точки до сектора кола
if (point_in_sector(x, y, r)) count_in_sector++;
}

// Виведення результатів перевірки
cout << "Points in triangle: " << count_in_triangle << endl;
cout << "Points in sector: " << count_in_sector << endl;
}

// Завдання 2: Обчислення суми ряду

// Функція для виконання Завдання 2
void task2() {
    int n; // Кількість членів ряду
    double x; // Змінна x
    double sum = 0; // Змінна для накопичення суми

    // Введення значень x та n
    cout << "Enter x and n: ";
    cin >> x >> n;
    if (!cin || n <= 0) { // Перевірка валідності введення
        cout << "Invalid input! n must be a positive integer." << endl;
        return;
    }

    // Цикл для обчислення суми ряду
    for (int k = 1; k <= n; k++) {
        // Обчислення члена ряду
        double term = (pow(x, k) + 1) / tgamma(k);
        sum += term; // Додавання до суми

        // Виведення кожного четвертого члена ряду
        if (k % 4 == 0) {
            cout << "Term " << k << " = " << term << endl;
        }
    }

    // Виведення загальної суми
    cout << "Total sum = " << sum << endl;
}

// Завдання 3: Обчислення суми ряду з точністю

// Функція для виконання Завдання 3
void task3() {
    double x; // Змінна x
    double e = 1e-10; // Точність
    double g = 1e5; // Межа розбіжності
    double u; // Поточний член ряду

```

```

double sum = 0; // Сума ряду
int n = 1;      // Початковий індекс

// Введення значення x
cout << "Enter x (0 < x < 1): ";
cin >> x;

if (x <= 0 || x >= 1) { // Перевірка валідності x
    cout << "Invalid input! x must be between 0 and 1." << endl;
    return;
}

// Ініціалізація першого члена ряду
u = (pow(-1, n) * pow(x, n - 1) * sqrt(3 * n + 1)) / tgamma(n + 1);

// Цикл обчислення ряду до досягнення точності
while (fabs(u) >= e && fabs(u) <= g) {
    sum += u; // Додавання до суми

    // Виведення поточного члена ряду
    cout << "u" << n << " = " << u << endl;

    n++; // Збільшення індексу
    u = (pow(-1, n) * pow(x, n - 1) * sqrt(3 * n + 1)) / tgamma(n + 1); // Наступний член
}

// Виведення результатів
cout << "Last element u" << n << " = " << u << endl;
cout << "Total sum = " << sum << endl;
}

// Головна функція з меню
int main() {
    int choice; // Змінна для вибору користувача

    do {
        // Виведення меню
        cout << "\nMenu:\n";
        cout << "1. Task 1: Point verification\n";
        cout << "2. Task 2: Series sum\n";
        cout << "3. Task 3: Series with precision\n";
        cout << "0. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        // Виконання вибраного завдання
        switch (choice) {
            case 1:
                task1();
                break;
            case 2:
                task2();

```

```
        break;
    case 3:
        task3();
        break;
    case 0:
        cout << "Exiting the program. Goodbye!\n";
        break;
    default:
        cout << "Invalid choice! Please try again.\n";
    }
} while (choice != 0); // Цикл працює, поки користувач не вибере вихід

return 0;
}
```