

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет ім. М. Є. Жуковського  
«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.301.272.311.2 ЛР

Виконав студент гр. \_\_\_\_\_311\_\_\_\_\_

16.12.24 Даніл ТАГАЄВ  
(підпис, дата) (П.І.Б.)

Перевірив  
\_\_\_\_\_к.т.н., доц. Олена ГАВРИЛЕНКО  
(підпис, дата) (П.І.Б.)

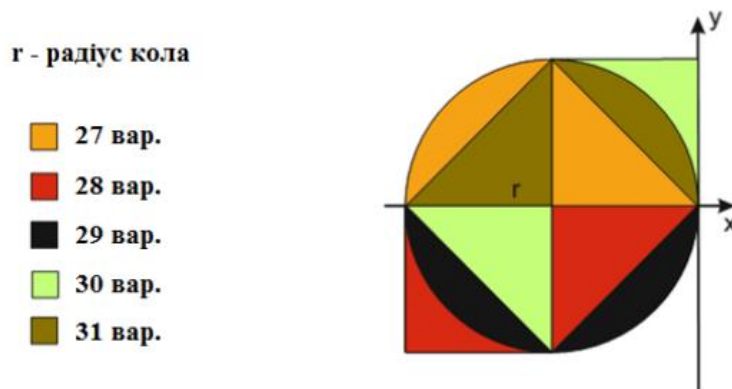
## МЕТА РОБОТИ

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. 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).

Geom28.табл1.варіант2: 28 варіант, червоний колір



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

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

9.табл.2,варіант,2:

$$\sum_{k=1}^n \frac{\sqrt[k]{x} \cdot k \cdot (-1)^k}{x^{k-1}}, 1 \leq x \leq 5$$

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 ( $g = 10^2 \dots 10^5$ ).

9.прод.табл.3,варіант,2:

$$\sum_{n=1}^{\infty} \frac{x^n * n!}{n^{n/2}}$$

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).

## ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі №28

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

Geom 28: PI – константа, дійсний тип із подвійною точністю,  $P = 3,14$

$x$  і  $y$  – координати точки, дійсний тип із подвійною точністю

$r$  – радіус кола, дійсний тип із подвійною точністю

Вихідні дані (ім'я, опис, тип):

Якщо точка знаходиться в указаній площі ( $x > 0$ ,  $y < 0$  і  $x^2 + y^2 \leq r^2$ ),

Виведення повідомлення о знаходженні.

Якщо ні, виведення повідомлення, що точка не знаходиться у фігурі.

Лістинг коду вирішення задачі If 10 наведено в дод. А (стор. 11-14).

Алгоритм вирішення:

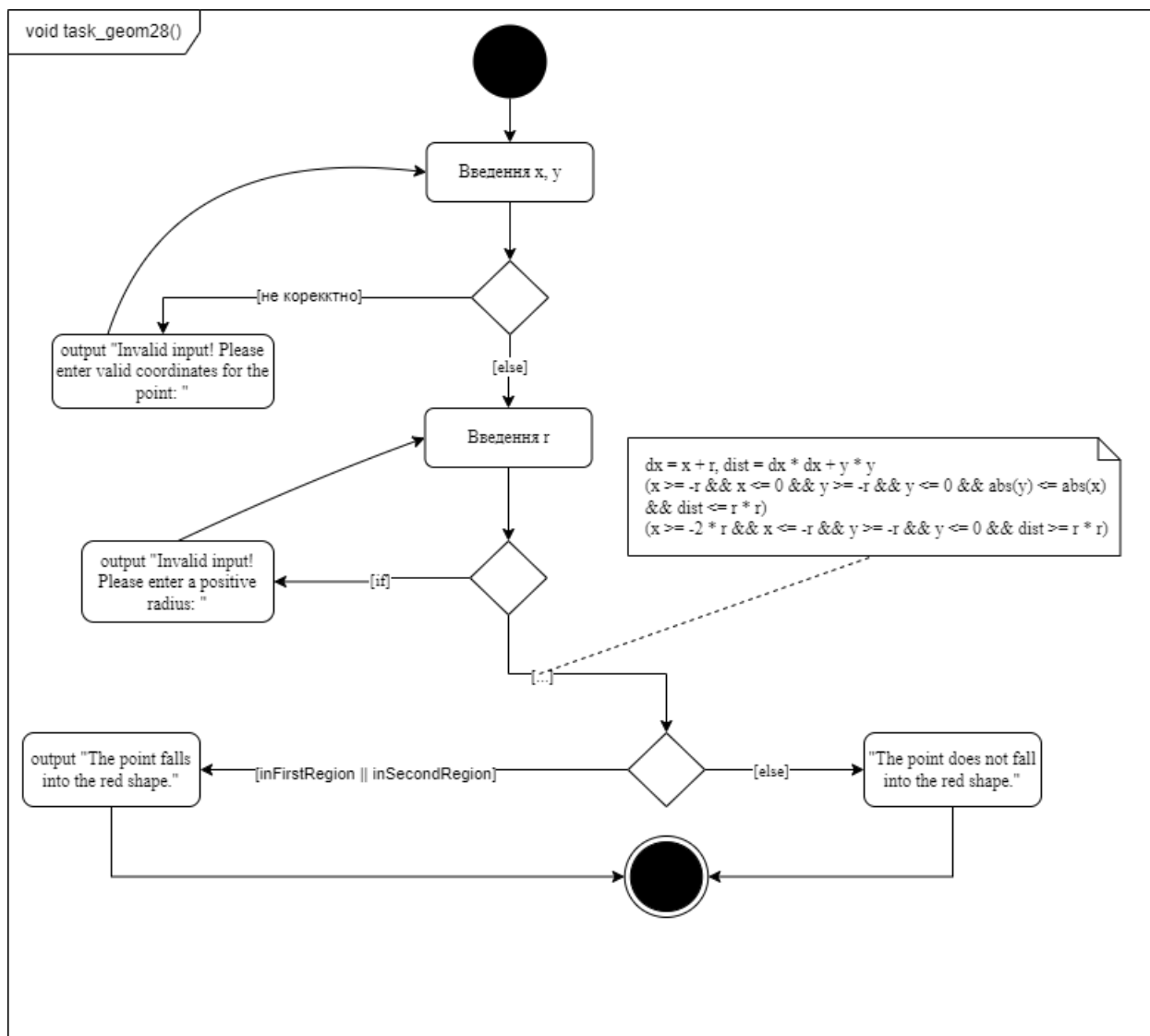


Рисунок 1 – Діаграма активності для алгоритму вирішення задачі Geom28 з циклом

Екран роботи програми показаний на рис. 2, 3 та 4

```

Enter the coordinates of the point (x, y): 5 -5
Enter the radius of the circle r: 50
The point falls into the red shape.
Для продовження натисніть будь-яку клавішу . . .

C:\Users\user\Desktop\lab3\Debug\lab3.exe (процес 8928) завершив роботу з кодом 0 (0x0).
Натисніть будь-яку клавішу, чтобы закрыть это окно:|
  
```

Рисунок 2 - Екран роботи програми завдання Geom28, точка знаходиться у фігурі

```

Enter the coordinates of the point (x, y): 10 10
Enter the radius of the circle r: 100
The point does not fall into the red shape.
Для продолжения нажмите любую клавишу . . .

C:\Users\user\Desktop\lab3\64\Debug\lab3.exe (процесс 17176) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:|

```

Рисунок 3 - Экран роботи програми завдання Geom28, точка не знаходиться у фігуру

```

Menu:
1. Check if the point is inside the red shape
2. Calculate expression
3. Calculate series convergence
4. Exit
Enter your choice: 1
Enter the coordinates of the point (x, y): test
Invalid input! Please enter valid coordinates for the point: |

```

Рисунок 4 - Экран роботи програми завдання Geom28, коли ведено некоректні дані

Завдання 2.

Вирішення задачі №9

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

9:  $x$  – дійсне число, дійсний тип із подвійною точністю

$n$  – натуральне число, цілий тип

Вихідні дані (ім'я, опис, тип):

Кожен четвертий елемент  $k$  (якщо умова виконується)

sum – дійсне число, дійсний тип із подвійною точністю

Лістинг коду вирішення задачі 9 наведено в дод. А (стор. 11-14).

Алгоритм вирішення:

- 1) Виведення запрошення до вводу
- 2) Ведення змінної  $x$ ,  $n$
- 3) Перевірка коректності введення
- 4) Обчислення суми ряду  $k$  від 1 до  $n$ , якщо  $k \% 4 == 0$ , виводиться на екран
- 5) Виведення загальної суми

Екран роботи програми показаний на рис. 5, 6 та 7

```

Menu:
1. Check if the point is inside the red shape
2. Calculate expression
3. Calculate series convergence
4. Exit
Enter your choice: 2
Enter a natural number n: 2
Enter a real number x (1 <= x <= 5): 3
Calculating the sum and partial elements:
Total sum: -1.8453

```

Рисунок 5 - Экран роботи програми завдання 9

```

Menu:
1. Check if the point is inside the red shape
2. Calculate expression
3. Calculate series convergence
4. Exit
Enter your choice: 2
Enter a natural number n: 8
Enter a real number x (1 <= x <= 5): 2
Calculating the sum and partial elements:
Element at k = 4: 0.594604
Element at k = 8: 0.0681567
Total sum: -1.13723

```

Рисунок 6 - Екран роботи програми завдання 9, коли виводиться кожен 4 елемент

```

Menu:
1. Check if the point is inside the red shape
2. Calculate expression
3. Calculate series convergence
4. Exit
Enter your choice: 2
Enter a natural number n: test
Invalid value for n. It must be a natural number. Try again: 2
Enter a real number x (1 <= x <= 5): test
Invalid value for x. It must satisfy 1 <= x <= 5. Try again: |

```

Рисунок 7 - Екран роботи програми завдання 9, коли ведено некоректно дані

Завдання 3.

Вирішення задачі №30

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

Geom 28: x – дійсне число, дійсний тип з подвійною точністю

e = 1E-5, дійсне число, дійсний тип з подвійною точністю

g = 100, дійсне число, дійсний тип з подвійною точністю

n = 1 – ціле число, цілий тип.

Вихідні дані (ім'я, опис, тип):

sum – дійсне число, дійсний тип з подвійною точністю

Лістинг коду вирішення задачі Calculate28 наведено в дод. А (стор. 11-14).

Алгоритм вирішення:

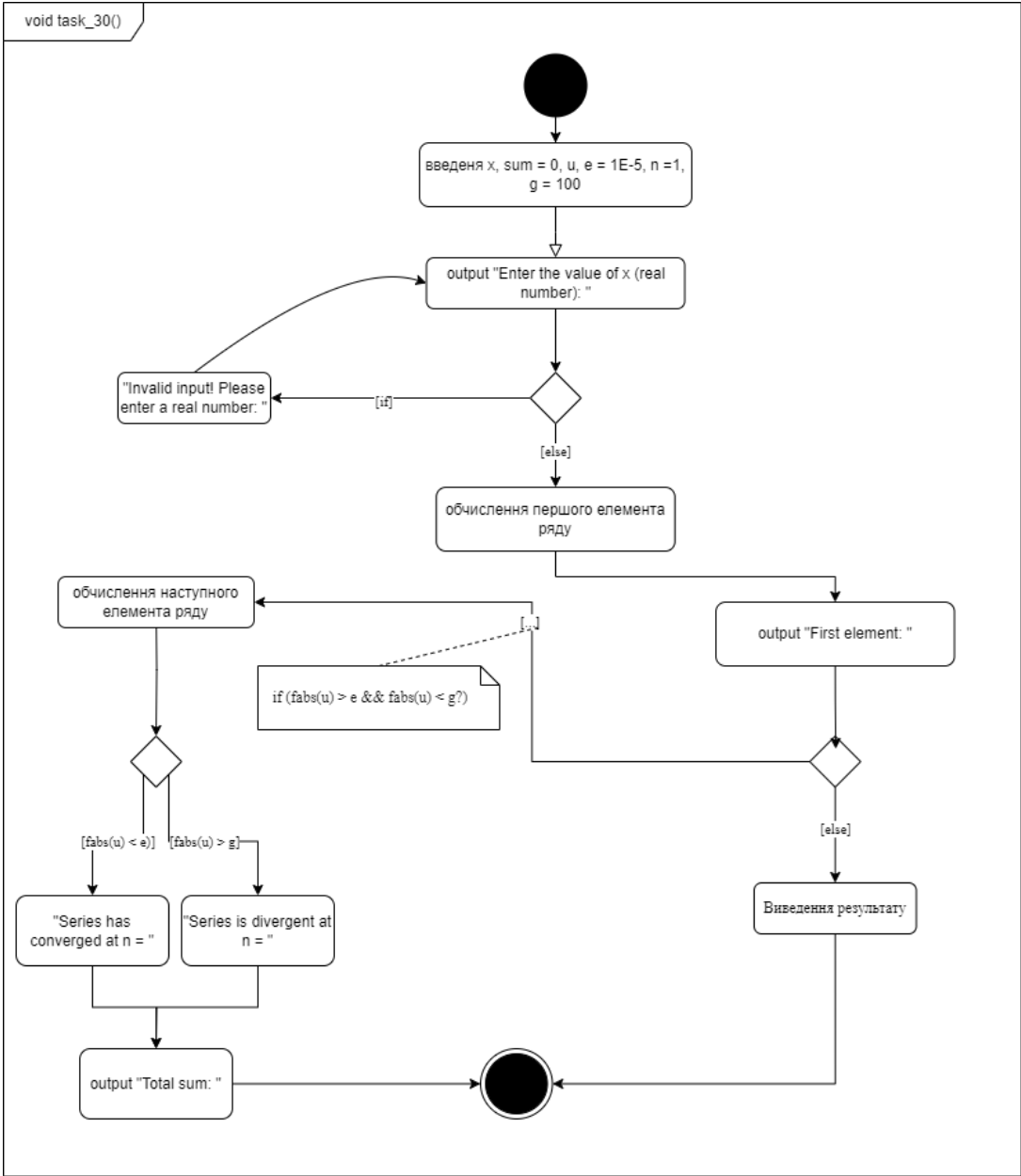


Рисунок 8 – Діаграма активності для алгоритму вирішення задачі 30 з циклом

Екран роботи програми показаний на рис. 9, 10 та 11

```

Menu:
1. Check if the point is inside the red shape
2. Calculate expression
3. Calculate series convergence
4. Exit
Enter your choice: 3
Enter the value of x (real number): 1
First element: 1
Element at n = 2: 1
Element at n = 3: 1.1547
Element at n = 4: 1.5
Element at n = 5: 2.14663
Element at n = 6: 3.33333
Element at n = 7: 5.55376
Element at n = 8: 9.84375
Element at n = 9: 18.4362
Element at n = 10: 36.288
Element at n = 11: 74.7302
Element at n = 12: 160.417
Series is divergent at n = 12. Stopping calculation.
Total sum: 315.403

```

Рисунок 9 – Екран роботи програми завдання 30, зупинка по розбіжності

```

Enter your choice: 3
Enter the value of x (real number): 0.3
First element: 0.3
Element at n = 2: 0.09
Element at n = 3: 0.0311769
Element at n = 4: 0.01215
Element at n = 5: 0.0052163
Element at n = 6: 0.00243
Element at n = 7: 0.00121461
Element at n = 8: 0.000645848
Element at n = 9: 0.00036288
Element at n = 10: 0.000214277
Element at n = 11: 0.000132382
Element at n = 12: 8.5252e-05
Element at n = 13: 5.7046e-05
Element at n = 14: 3.95558e-05
Element at n = 15: 2.83553e-05
Element at n = 16: 2.09701e-05
Element at n = 17: 1.59703e-05
Element at n = 18: 1.25046e-05
Element at n = 19: 1.00517e-05
Element at n = 20: 8.28418e-06
Series has converged at n = 20. Stopping calculation.
Total sum: 0.443821

```

Рисунок 10 – Екран роботи програми завдання 30, зупинка по збіжності

```

Menu:
1. Check if the point is inside the red shape
2. Calculate expression
3. Calculate series convergence
4. Exit
Enter your choice: 3
Enter the value of x (real number): test
Invalid input! Please enter a real number: |

```

Рисунок 11 – Екран роботи програми завдання 30, коли некоректно дані



Завдання 4.

Вирішення задачі Geom28, 9, 30

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

Geom28, 9, 30

Вихідні дані (ім'я, опис, тип): Виведення завдання

Лістинг коду вирішення задачі наведено в дод. А (стор. 11-14).

Алгоритм вирішення:

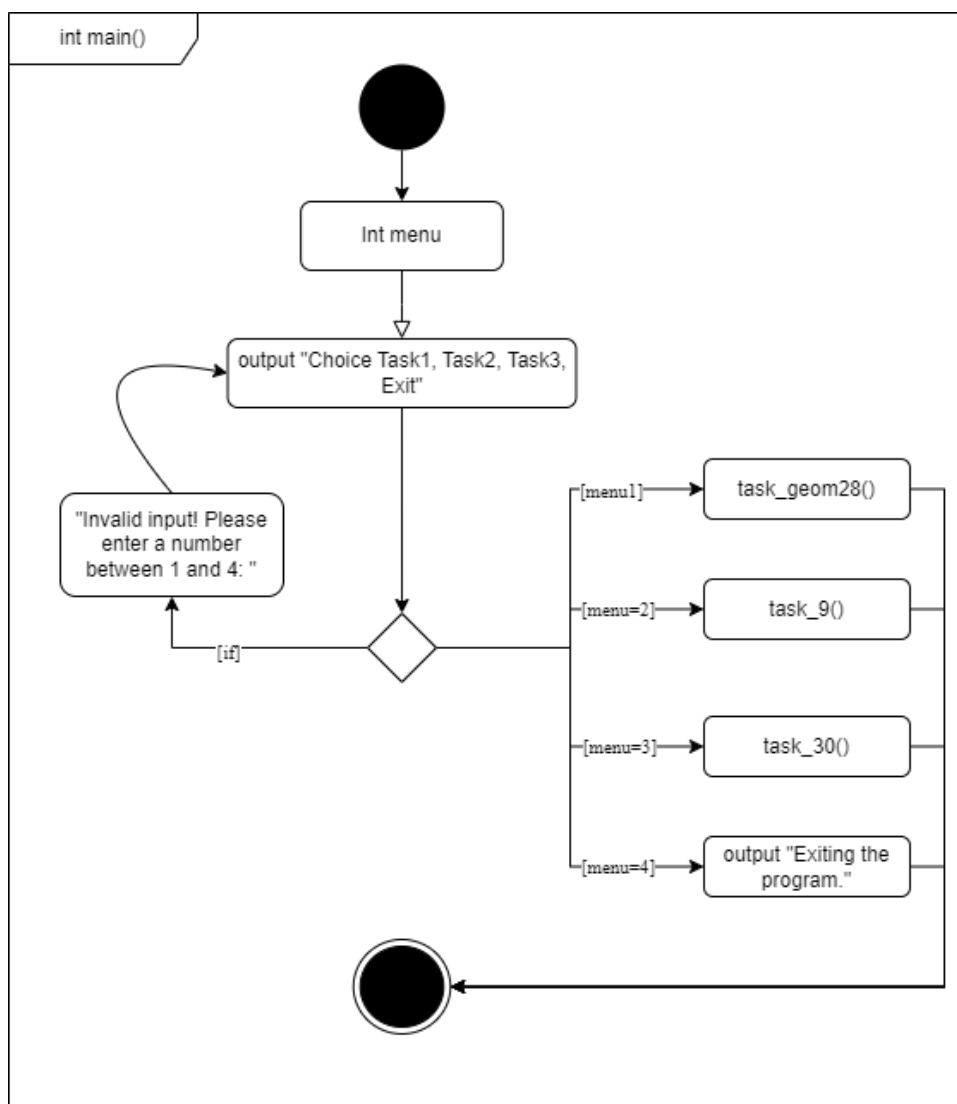


Рисунок 12 – Діаграма активності для всієї програми

Екран роботи програми показаний на рис. 13 та 14

```
Menu:  
1. Check if the point is inside the red shape  
2. Calculate expression  
3. Calculate series convergence  
4. Exit  
Enter your choice: 1  
Enter the coordinates of the point (x, y): |
```

Рисунок 13 – Діаграма активності для всієї програми

```
Menu:  
1. Check if the point is inside the red shape  
2. Calculate expression  
3. Calculate series convergence  
4. Exit  
Enter your choice: test  
Invalid input! Please enter a number between 1 and 4: |
```

Рисунок 14 – Екран роботи всієї програми, коли ведено некоректно дані

## ВИСНОВКИ

Було освоєно алгоритми з використання інструкцій циклу з передумовою, циклу післяумовою і параметризованого циклу мовою C++ в середовищі Visual Studio. Виведення значення кожного четвертого елемента, як дослідити ряд на збіжність, ведення величини для переривання циклу обчислення суми збіжного або розбіжного ряду.

## ДОДАТОК А

### Лістинг коду програми до задач Geom28, 9 та 30

```

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

// Оголошення функцій
void task_geom28(); // Завдання 2: Перевірка, чи точка потрапляє в червону фігуру
bool isValidDouble(double x); // Для перевірки коректності введення числа

void task_9(); // Завдання 9: Обчислення виразу для заданих x і n

void task_30(); // Завдання 30: Дослідження ряду на збіжність
double factorial(int n); // Функція для обчислення факторіалу

// Основна функція меню
int main() {
    int menu;
    while (true) {
        cout << "\nMenu:\n";
        cout << "1. Check if the point is inside the red shape\n";
        cout << "2. Calculate expression\n";
        cout << "3. Calculate series convergence\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";

        // Введення вибору користувача з перевіркою
        while (true) {
            cin >> menu;
            if (cin.fail() || menu < 1 || menu > 4) {
                cin.clear();
                cin.ignore(32767, '\n');
                cout << "Invalid input! Please enter a number between 1 and 4: ";
            }
            else {
                break;
            }
        }

        // Вибір дії
        switch (menu) {
            case 1:
                task_geom28();
                break;
            case 2:
                task_9();
                break;
            case 3:
                task_30();
                break;
            case 4:
                cout << "Exiting the program.\n";
                return 0;
        }
    }
}

// Завдання 28: Перевірка, чи точка потрапляє в червону фігуру

```

```

void task_geom28() {
    double x, y, r;

    // Введення координат точки
    cout << "Enter the coordinates of the point (x, y): ";
    while (true) {
        cin >> x >> y;
        if (cin.fail()) {
            cin.clear();
            cin.ignore(32767, '\n');
            cout << "Invalid input! Please enter valid coordinates for the
point: ";
        }
        else {
            break;
        }
    }

    // Введення радіусу кола
    cout << "Enter the radius of the circle r: ";
    while (true) {
        cin >> r;
        if (cin.fail() || r <= 0) {
            cin.clear();
            cin.ignore(32767, '\n');
            cout << "Invalid input! Please enter a positive radius: ";
        }
        else {
            break;
        }
    }

    // Перевірка, чи точка потрапляє в одну з двох областей
    double dx = x + r, dist = dx * dx + y * y;
    bool inFirstRegion = (x >= -r && x <= 0 && y >= -r && y <= 0 && abs(y) <=
abs(x) && dist <= r * r);
    bool inSecondRegion = (x >= -2 * r && x <= -r && y >= -r && y <= 0 && dist
>= r * r);

    // Виведення результату
    if (inFirstRegion || inSecondRegion) {
        cout << "The point falls into the red shape." << endl;
    }
    else {
        cout << "The point does not fall into the red shape." << endl;
    }
}

// Завдання 9: Обчислення виразу для заданих x і n
void task_9() {
    double x, sum = 0; // Ініціалізація змінних
    int n;

    // Введення натурального числа n
    cout << "Enter a natural number n: ";
    while (true) {
        cin >> n;
        if (cin.fail() || n <= 0) { // n має дійсним типом
            cin.clear();
            cin.ignore(32767, '\n');
            cout << "Invalid value for n. It must be a natural number. Try
again: ";
        }
    }
}

```

```

    } else {
        break; // Якщо n коректне, виходимо з циклу
    }
}

// Введення реального числа x
cout << "Enter a real number x (1 <= x <= 5): ";
while (true) {
    cin >> x;
    if (cin.fail() || x < 0 || x > 5) { // x має бути в межах [1, 5]
        cin.clear();
        cin.ignore(32767, '\n');
        cout << "Invalid value for x. It must satisfy 1 <= x <= 5. Try
again: ";
    }
    else {
        break; // Якщо x коректне, виходимо з циклу
    }
}

// Обчислення суми
cout << "Calculating the sum and partial elements:" << endl;
for (int k = 1; k <= n; k++) {
    double term = (pow(x, 1.0 / k) * k * pow(-1, k)) / pow(x, k - 1);
    sum += term;

    // Виведення кожного четвертого елемента
    if (k % 4 == 0) {
        cout << "Element at k = " << k << ": " << term << endl;
    }
}

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

// Завдання 30: Дослідження ряду на збіжність

// Функція для обчислення факторіалу
double factorial(int n) {
    if (n <= 1) return 1; // Якщо n <= 1, факторіал дорівнює 1
    double res = 1;
    for (int i = 1; i <= n; i++) {
        res *= i; // Множимо всі числа від 1 до n
    }
    return res;
}

// Функція для обчислення ряду та перевірки на збіжність
void task_30() {
    double x, sum = 0, u, e = 1E-5, g = 100;
    // e - мала величина для переривання циклу обчислення суми
    // g - величина для переривання циклу обчислення суми розбіжного ряду
    int n = 1;

    // Введення значення x
    cout << "Enter the value of x (real number): ";
    while (true) {
        cin >> x;

        if (cin.fail()) {
            cin.clear();

```

```

        cin.ignore(32767, '\n');
        cout << "Invalid input! Please enter a real number: ";
    }
    else {
        break; // вихід з циклу, якщо ввід коректний
    }
}

// Обчислення першого елемента ряду
u = (pow(x, n) * factorial(n)) / pow(n, n / 2.0);
sum += u;

// Виведення першого елемента
cout << "First element: " << u << endl;

// Обчислення суми ряду
while (fabs(u) > e && fabs(u) < g) {
    n++; // Збільшуємо n для обчислення наступного елемента
    u = (pow(x, n) * factorial(n)) / pow(n, n / 2.0); // Обчислення кожного
елемента ряду
    sum += u;

    // Виведення поточного елемента
    cout << "Element at n = " << n << ": " << u << endl;

    // Умова зупинки по розбіжності
    if (fabs(u) > g) {
        cout << "Series is divergent at n = " << n << ". Stopping
calculation.\n";
        break;
    }

    // Умова зупинки по збіжності
    if (fabs(u) < e) {
        cout << "Series has converged at n = " << n << ". Stopping
calculation.\n";
        break;
    }
}
// Виведення загальної суми
cout << "Total sum: " << sum << endl;
}

```