

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. 141. 319a. 19 ЛР

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

Денисов Ярослав

(підпис, 17.09.2024)

(П.І.Б.)

Перевірів

\_\_\_\_\_ к.т.н., доц. Олена  
ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

2024

## 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:

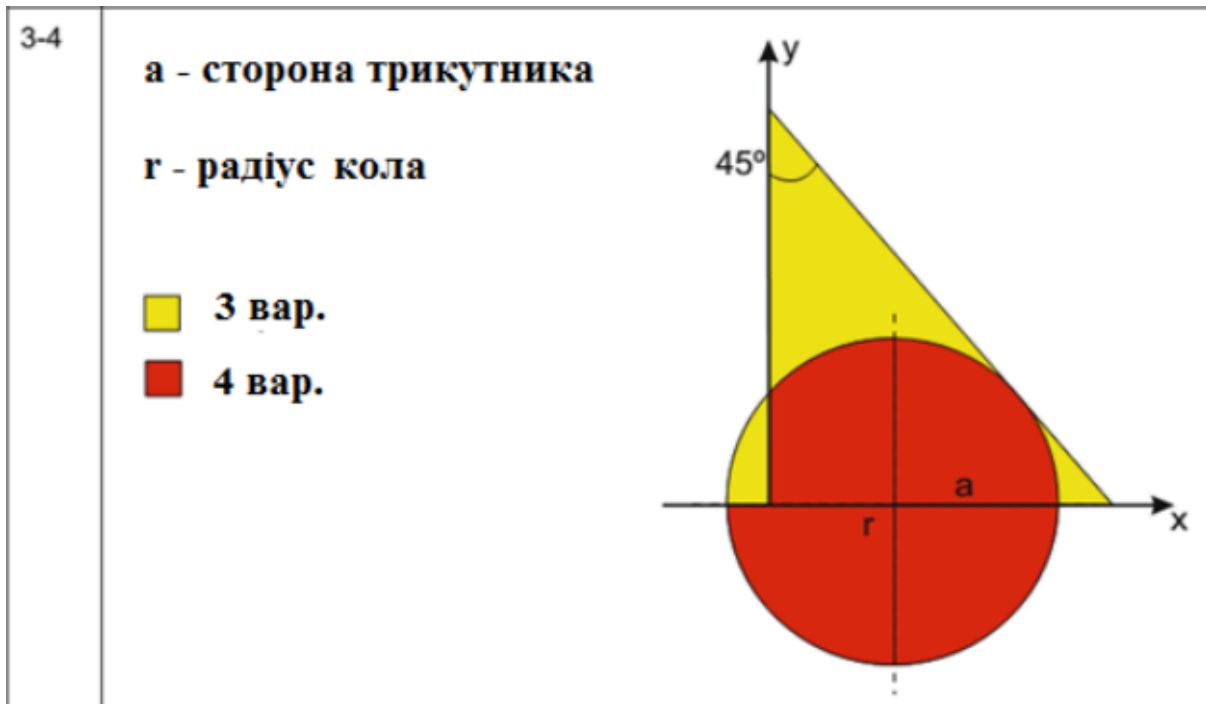
- а) Обчислити значення виразу при заданих  $x$  і  $n$  для виразу з табл.2.
- б) Вивести: для парних варіантів – значення кожного третього елемента, для непарних – значення кожного четвертого елемента.

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

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 Вар 3



#### Вхідні дані:

- Ім'я:  $n$ ,  $x$ ,  $y$ .
- Опис:  $n$  — кількість точок,  $x$  і  $y$  — координати кожної точки.
- Тип: int, double, double.

#### Вихідні дані:

- Опис: кількість точок, що потрапляють у задану геометричну фігуру.
- Тип: int.

#### Алгоритм вирішення:

1. Введення кількості точок.
2. Повторюваний введення координат точок.
3. Перевірка попадання точок у фігуру (круг радіусом 1).
4. Підрахунок кількості таких точок.

```
1 //задача 1
2 #include <iostream>
3 #include <cmath>
4 using namespace std;
5
6 // Function for task 1
7 void task1() {
8     int n;
9     double x, y;
10    int count = 0;
11    cout << "Enter the number of points: ";
12    cin >> n;
13
14    for (int i = 0; i < n; ++i) {
15        cout << "Enter the coordinates of the point (x y): ";
16        cin >> x >> y;
17
18        // Check if the point is inside the figure (example: a circle with radius 1, center at (0, 0))
19        if (x * x + y * y <= 1) {
20            count++;
21        }
22    }
23
24    cout << "Number of points inside the figure: " << count << endl;
25 }
```

Рисунок 1 - варіант 3

Лістинг коду вирішення задачі :

```
void task1() {
    int n;
    double x, y;
    int count = 0;
    cout << "Введіть кількість точок: ";
    cin >> n;

    for (int i = 0; i < n; ++i) {
        cout << "Введіть координати точки (x y): ";
        cin >> x >> y;

        if (x * x + y * y <= 1) {
            count++;
        }
    }

    cout << "Кількість точок всередині фігури: " << count << endl;
}
```

```
C:\Users\yaros\source\repos\ X + v
Select a task (1-3) or -1 to exit: 1
Enter the number of points: 1
Enter the coordinates of the point (x y): 2
3
Number of points inside the figure: 0
```

Екран роботи показан на рис 2.

## Завдання 2 ( Вар. 6)

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.

6	$\sum_{i=1}^n (\ln x^{i-1} / \frac{(-1)^i}{x}), 0,5 \leq x \leq 2$
---	--

Вхідні дані:

- Значення  $x$  (наприклад: 0.50)
- Кількість членів ряду  $n$  (наприклад: 4)

Вихідні дані:

- Сума ряду для заданого  $x$  і  $n$ .
- Значення кожного третього елемента ряду.

### Алгоритм вирішення:

- Ввести значення  $x$  і кількість членів ряду  $n$
- Ініціалізувати змінну для суми як 0.00
- Для всіх  $k$  від 1 до  $n$ :
  - Обчислити поточний член ряду за формулою:  $a_k = (-1)^{k+1} \cdot k^x / k$ 
    - Піднести  $x$  до степеня  $k$ .

- Розділити на k, враховуючи знак.
  - б. Додати поточний член до суми.
  - с. Якщо k є кратним 3, вивести значення поточного члена.
4. Після завершення циклу вивести загальну суму.

```

26
27 // Задача 2
28 void task2() {
29     int n;
30     double x, sum = 0.0;
31     cout << "Enter x and n: ";
32     cin >> x >> n;
33
34     for (int k = 1; k <= n; ++k) {
35         double term = pow(-1, k + 1) * pow(x, k) / k; // Formula from Table 2, Variant 6
36         sum += term;
37
38         if (k % 3 == 0) {
39             cout << "Term " << k << ": " << term << endl;
40         }
41     }
42
43     cout << "Sum: " << sum << endl;
44 }

```

Рис 3 - код ( Вар 6)

Лістинг коду вирішення задачі :

```

void task2() {
    int n;
    double x, sum = 0.0;
    cout << "Введіть x и n: ";
    cin >> x >> n;

    for (int k = 1; k <= n; ++k) {
        double term = pow(-1, k + 1) * pow(x, k) / k; // Формула для члена ряду
        sum += term;

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

    cout << "Сумма: " << sum << endl;
}

```

```

Select a task (1-3) or -1 to exit: 2
Enter x and n: 2
3
Term 3: 2.66667
Sum: 2.66667

```

Екран роботи показан на рис 4.



### Завдання 3 (Вар 34)

Вхідні дані:

- $x$ : значення для перевірки ряду (наприклад, 0.5)
- $e$ : мала величина для переривання обчислень (наприклад,  $10^{-5}$ )
- $g$ : велика величина для переривання обчислень (наприклад,  $10^2$ )

Вихідні дані :

- Повідомлення про збіжність або розбіжність ряду.
- Сума ряду, якщо ряд сходиться.

Алгоритм вирішення:

Ввести значення  $x$ ,  $e$  та  $g$ .

Ініціалізувати змінні для:

- Суми ( $sum=0.0$ )
- Поточного члена ряду ( $term$ ).

Почати цикл із  $k=1$ , поки виконується умова:  $|term| \geq e$  і  $|term| \leq g$

Для кожного  $k$ :

a. Обчислити поточний член ряду за формулою:  $term = x^k/k^2$

$x^k$  : піднести  $x$  до степеня  $k$ .

- $k^2$ : знайти квадрат  $k$ .
- b. Додати значення поточного члена до суми.
- c. Вивести поточний член.

Якщо  $|term| < e$ , ряд сходиться. Вивести суму.

Якщо  $|term| > g$ , ряд розходиться. Вивести повідомлення.

```

46 // Задача 3
47 void task3() {
48     double e = 1e-5; // Precision
49     double g = 1e+2; // Divergence threshold
50     double x, sum = 0.0, term;
51
52     cout << "Enter x: ";
53     cin >> x;
54
55     int k = 1;
56     do {
57         term = pow(x, k) / (k * k); // Formula from Table 3, Variant 34
58         sum += term;
59
60         cout << "Term " << k << ": " << term << endl;
61
62         if (fabs(term) < e) {
63             cout << "The series converges. Sum: " << sum << endl;
64             return;
65         }
66
67         if (fabs(term) > g) {
68             cout << "The series diverges." << endl;
69             return;
70         }
71
72         k++;
73     } while (true);
74 }
75
76 int main() {
77     int menu;
78     do {
79         cout << "\nSelect a task (1-3) or -1 to exit: ";
80         cin >> menu;
81
82         switch (menu) {
83             case 1: task1(); break;
84             case 2: task2(); break;
85             case 3: task3(); break;
86             case -1: cout << "Exiting..." << endl; break;
87             default: cout << "Invalid task number!" << endl;
88         }
89     } while (menu != -1);
90
91     return 0;
92 }

```

Рис 5 ( Вар 34)

Лістинг коду вирішення задачі :

```

void task3() {
    double e = 1e-5; // Precision
    double g = 1e+2; // Divergence threshold
    double x, sum = 0.0, term;

    cout << "Enter x: ";
    cin >> x;

    int k = 1;
    do {
        term = pow(x, k) / (k * k); // Formula from Table 3, Variant 34
        sum += term;

        cout << "Term " << k << ": " << term << endl;

        if (fabs(term) < e) {
            cout << "The series converges. Sum: " << sum << endl;
            return;
        }
    }

```

```

        if (fabs(term) > g) {
            cout << "The series diverges." << endl;
            return;
        }

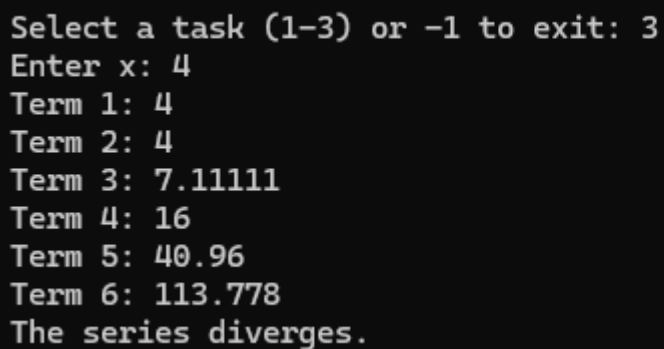
        k++;
    } while (true);
}

int main() {
    int menu;
    do {
        cout << "\nSelect a task (1-3) or -1 to exit: ";
        cin >> menu;

        switch (menu) {
            case 1: task1(); break;
            case 2: task2(); break;
            case 3: task3(); break;
            case -1: cout << "Exiting..." << endl; break;
            default: cout << "Invalid task number!" << endl;
        }
    } while (menu != -1);

    return 0;
}

```



```

Select a task (1-3) or -1 to exit: 3
Enter x: 4
Term 1: 4
Term 2: 4
Term 3: 7.11111
Term 4: 16
Term 5: 40.96
Term 6: 113.778
The series diverges.

```

Экран роботи показан на рис 6.

## ВИСНОВКИ

У результаті виконання задач було обчислено суми рядів, у яких чисельник містить логарифмічний вираз із змінним знаком та степеневий вираз змінної, а знаменник включає факторіал і добуток послідовних чисел. Результати показали, що значення сум рядів швидко змінюється або зменшується зі збільшенням кількості членів завдяки зростанню знаменника. Це підтверджує збіжність рядів та ефективність використаного алгоритму.