

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальними апаратами
Кафедра систем управління літальними апаратами

Лабораторна робота № 5
з дисципліни «Алгоритмізація та програмування»
Тема: «Реалізація циклічних алгоритмів мовою C++»

XAI.301.312.9LP

Виконав студент гр. 312

12.12.2023
(підпис, дата)

Зубов Євген Павлович
(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.

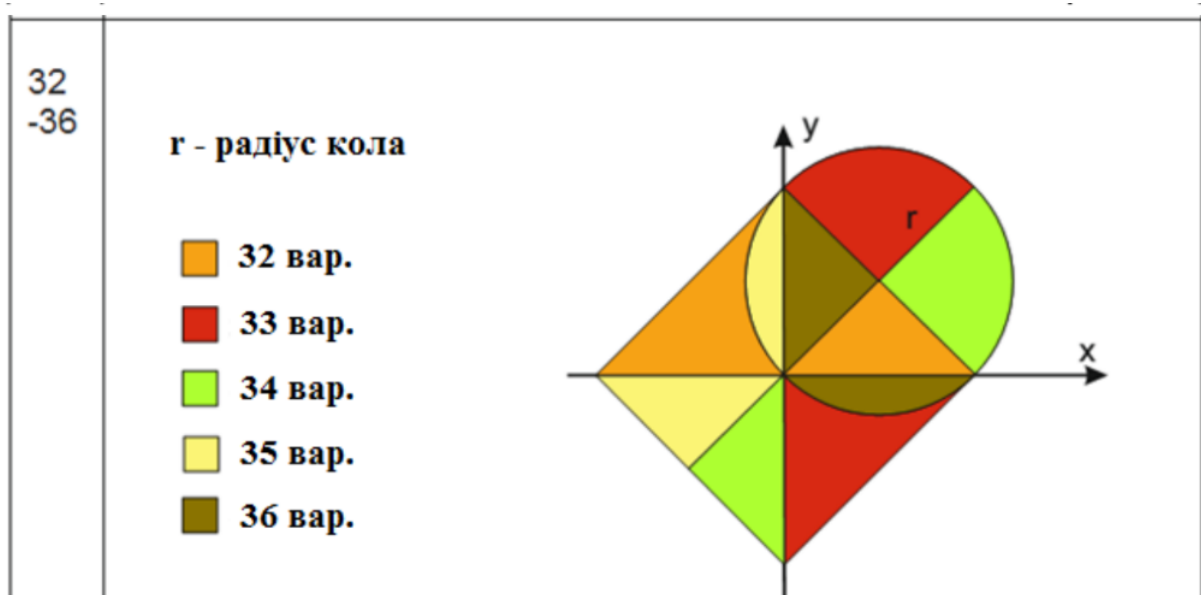
ПОСТАНОВКА ЗАДАЧІ

Створити порожній проект. Додати вихідний файл main.cpp.

Додати в файл програмний код для вирішення двох задач відповідно до варіанту.

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).

Geo32.



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.

SumOfSeries24

24	$\left(-\sum_{k=0}^n \frac{(-1)^k \left(-\frac{\pi}{2} + \sqrt{x} \right)^{1+2k}}{(1+2k)!} \right)^{x/3}$
----	---

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

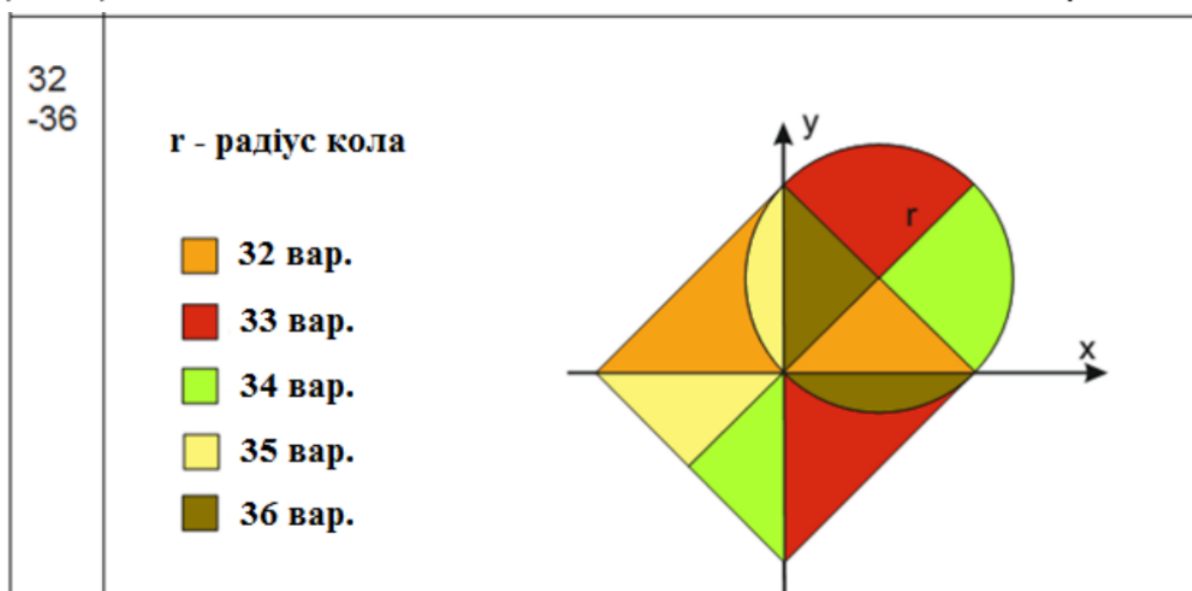
SumOfSeries34

34	$\sum_{n=0}^{\infty} \frac{x^{3n}}{(2n+1)!}$
----	--

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі. Geo32



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

г - число, int

Вихідні дані:

Ar1 - рівняння першої фігури(верхньої),bool

Ar2 - рівняння другої фігури(нижньої),bool

Алгоритм вирішення показано на рис. 1

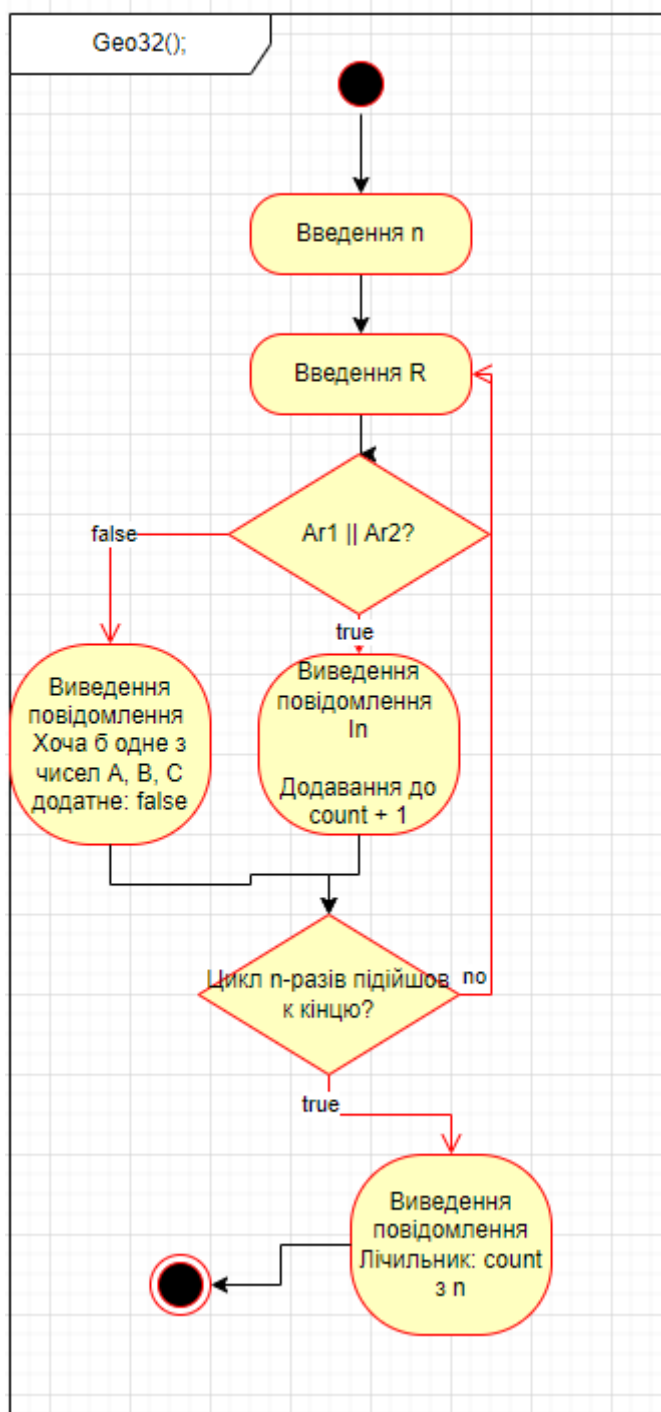


Рисунок 1 – Geo32

Лістинг коду вирішення задач If23 наведено в дод. А (стор.7).

Екран роботи програми показаний на рис. Б.

№2.SumOfSeries24

24	$\left(-\sum_{k=0}^n \frac{(-1)^k \left(-\frac{\pi}{2} + \sqrt{x} \right)^{1+2k}}{(1+2k)!} \right)^{x/3}$
25	

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

n, x- числа,int

Вихідні дані:

Виводиться відповідь - сума ряду, не є типом даних

Алгоритм вирішення показано на рис. 2

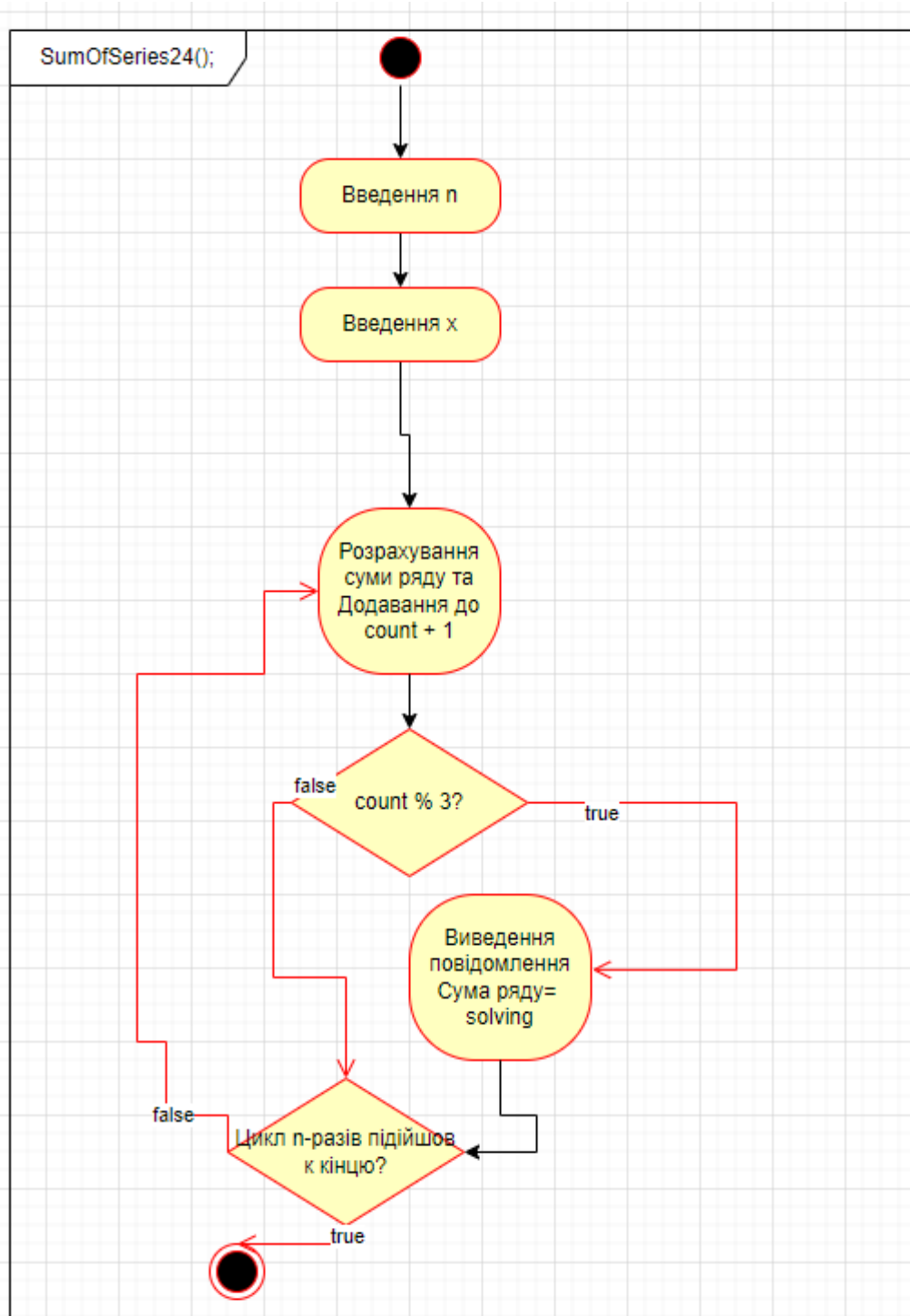


Рисунок 2 - SumOfSeries24

Лістинг коду вирішення задачі SumOfSeries24 наведено в дод. А (стор. 8). Екран роботи програми показаний на рис. Б

№3 SumOfSeries34

34	$\sum_{n=0}^{\infty} \frac{x^{3n}}{(2n+1)!}$
----	--

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

x - число, double

g - 1+E7, стала, double

Вихідні дані:

sum - число, double , сума ряду

Лістинг коду вирішення задачі SumOfSeries34 наведено в дод. А (стор. 8). Екран роботи програми показаний на рис. Б

ВИСНОВКИ

Закріплено на практиці введення та виведення програмних даних в C++. Отримано навички з оформлення звіту для лабораторної роботи

ДОДАТОК А

Лістинг коду програми до задач Begin10, Begin23

```
#include <iostream>
#include <locale.h>
#include <cmath>

using namespace std;

void geo32(); //Декларування функції типу void(задача geo32)

void SumOfSeries24(); //Декларування функції типу void(задача Solve24)
double factorial(int temp);

void SumOfSeries34(); //Декларування функції типу void(задача Solve32)

int main()
{
    int choice = 0; //Декларування змінної задля choice

    while (choice != 4) {
        setlocale(LC_ALL, "ukrainian");
        cout.clear();
        cout << "Виберіть яку з цих задач будете вирішувати: "
              "\n1. Geometry#32"
              "\n2. Solve26.1"
              "\n3. Solve34.2"
              "\n4. Вийти з програми" << endl;

        std::cin >> choice;

        switch (choice)
        {
            case 1:
            {
                // Geo32
                geo32();
                break;
            }
            case 2:
```

```

    {
        // Solve24
        SumOfSeries24();
        break;
    }
    case 3:
    {
        // Solve32
        SumOfSeries34();
        break;
    }
    case 4:
    {
        cout << "Програма завершена";
        break;
    }
    default:
    {
        cout << "Такого варіанту немає\n";
        break;
    }
}
}

void geo32() { // Функція типу void, що не повертає значень
    float x, y, r; // Декларування змінних типу float
    bool ar1, ar2; // Декларування змінних типу bool
    int n, count = 0;

    cout << "Кількість разів: ";
    cin >> n;

    cout << "Уведіть радіус кола: ";
    std::cin >> r;

    for (int i = 1; i <= n; i++)
    {

        if (r > 0)
        {
            cout << "Уведіть точку за координатою x : ";
            cin >> x;
            cout << "Уведіть точку за координатою y : ";

```

```

        cin >> y;

        ar1 = pow((x - r * 3 / 4), 2) + pow((y - r * 2 / 3), 2) < pow(r,
2) && y < -x + 2 * r * 3 / 4 && y < x && y > 0; //Рівняння першої
фігури(верхньої)
        ar2 = pow((x - r * (3.0 / 4)), 2) + pow((y - r * (2.0 / 3)), 2) >
pow(r, 2) && y > 0 && x < 0 && y < 2 * r && y < x + 2 * r * 2 / 3;
//Рівняння другої фігури(нижньої)

        if (ar1 || ar2) {
            count++;
            cout << "In" << endl;
        }
        else {

        }

    }
    else {
        cout << "Радіус не може бути від'ємним" << endl;
    }
}
cout << "Лічильник : " << count << " з " << n << endl;
}

void SumOfSeries24()
{
    int k = 0, n;
    const double pi = 3.14;
    double x;

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

    cout << "Обмеження n: ";
    cin >> n;

    double sum = 0;

    if (!cin || n < 0)
    {

```

```

        cout << "Неправильне введення даних" << endl;
    }
    else
    {
        for (k; k <= n; k++)
        {
            int temp = 1 + (2 * k);
            int count = 0;

            double part1 = pow(-1, k) * pow((-1 * pi / 2.0 + sqrt(x)),
temp);

            double part2 = factorial(temp);

            double solving = pow((-1.0 * (part1 / part2)), x);

            sum += solving;
            solving = 0;

            count = k % 3;

            if (count == 2) {
                cout << "Сума ряду: " << sum << endl;
            }
        }
    }
}

double factorial(int temp)
{
    int equals = 1;
    for (int i = 1; i <= temp; i++)
    {
        equals *= i;
    }
    return equals;
}

void SumOfSeries34()
{
    int k = 0;
    const double pi = 3.14;
    double x, g = 1E+7;

```

```
cout << "Введіть x: ";
cin >> x;

double sum = 0;

do
{
    k++;
    int count = 0;

    double part1 = pow(x, 3 * k);
    double part2 = factorial(2 * k + 1);

    double solving = part1 / part2;

    sum += solving;
    solving = 0;

    count = k % 3;

    if (count == 2) {
        cout << "Сума ряду: " << sum << endl;
    }
} while (sum < g);
}
```

ДОДАТОК Б

Скріншоти вікна виконання програми

Рисунок Б

```

E:\C++ Workshop\GitHub\lab.khai\x64\Debug\lab.khai.exe
Виберіть яку з цих задач будете вирішувати:
1. Geometry#32
2. Solve26.1
3. Solve34.2
4. Вийти з програми
1
Кількість разів: 5
Уведіть радіус кола: 5
Уведіть точку за координатою x : -2
Уведіть точку за координатою y : 1
In
Уведіть точку за координатою x : 2
Уведіть точку за координатою y : 1
In
Уведіть точку за координатою x : 2
Уведіть точку за координатою y : 0.5
In
Уведіть точку за координатою x : -4
Уведіть точку за координатою y : 0.3
In
Уведіть точку за координатою x : -5
Уведіть точку за координатою y : 1
In
Лічильник : 5 з 5
Виберіть яку з цих задач будете вирішувати:
1. Geometry#32
2. Solve26.1
3. Solve34.2
4. Вийти з програми
2
Введіть x: 4
Обмеження n: 5
Сума ряду: 0.034188
Сума ряду: 0.034188
Виберіть яку з цих задач будете вирішувати:
1. Geometry#32
2. Solve26.1
3. Solve34.2
4. Вийти з програми
3
Введіть x: 4
Сума ряду: 44.8
Сума ряду: 169.946
Сума ряду: -973175
Виберіть яку з цих задач будете вирішувати:
1. Geometry#32
2. Solve26.1
3. Solve34.2
4. Вийти з програми

```