

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 на мові C ++»

XAI.301.173.310.02 ЛР

Виконав студент гр. \_\_\_\_\_ 310 \_\_\_\_\_

14.12.2023 \_\_\_\_\_ Софія ПОЛЯКОВА \_\_\_\_\_  
(підпис, дата) (П.І.Б.)

Перевірів

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

2023

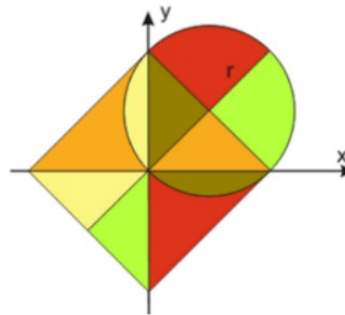
## МЕТА РОБОТИ

Вивчити теоретичний матеріал із синтаксису мовою C++ і поданням у вигляді блок-схем циклічних алгоритмів і реалізувати алгоритми з використанням інструкцій циклу з перед-умовою, циклу з після-умовою і параметричного 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).

г - радіус кола



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

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

6

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

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 ( $g = 102 \dots 105$ ).

14

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

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 geom35.

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

- 1)  $r$  — радіус, ціле число,  $r > 0$ ;
- 2)  $x, y$  — координати точки, дійсний тип;
- 3)  $n$  — кількість спроб, ціле число.

Вихідні дані (ім'я, опис, тип):

- 1) count — кількість вдалих спроб, ціле число;
- 2)  $n$  — кількість спроб, ціле число.

Алгоритм вирішення показано на рис. 1.

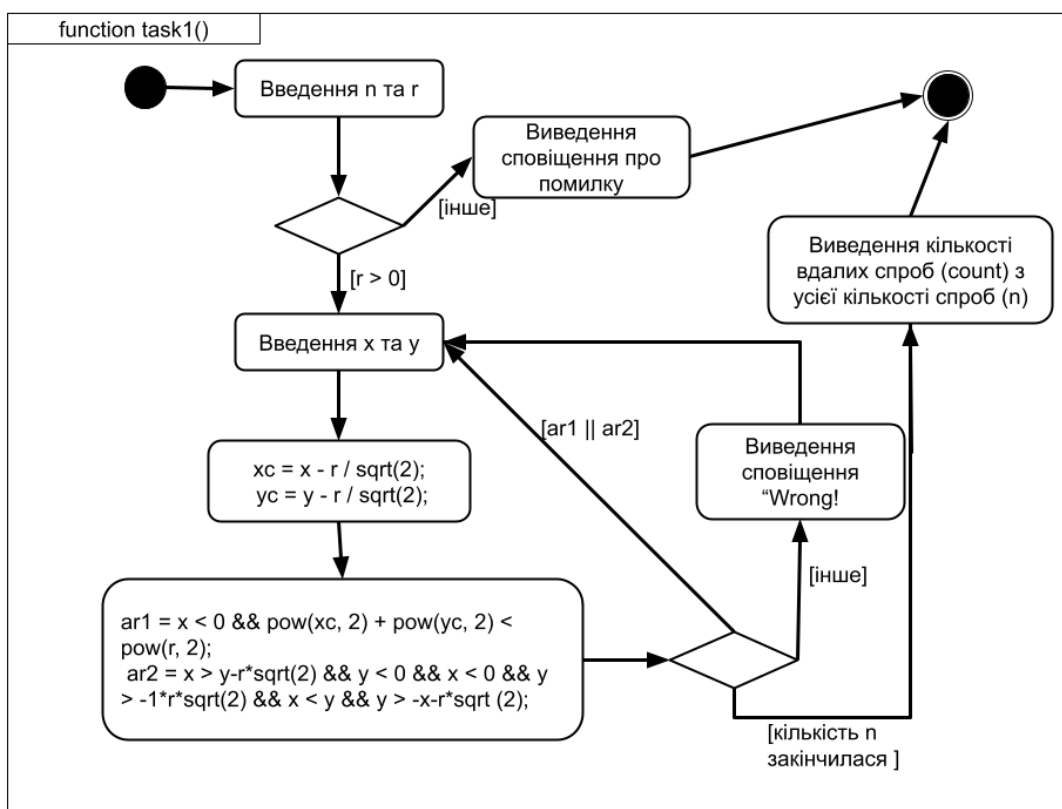


Рисунок 1 – Алгоритм вирішення geom35

Лістинг коду вирішення задачі geom35 наведено в дод. А (стор. 9).

Екран роботи програми показаний на рис. Б.1.

## Завдання 2.

Вирішення задачі rangeб.

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

1)  $x$  — невідоме число, дійсний тип,  $0.5 \leq x \leq 2$ ;2)  $n$  — межа, ціле число,  $n > 0$ .

Вихідні дані (ім'я, опис, тип):

 $sum$  — сума рядів, дійсний тип.

Алгоритм вирішення показано на рис. 2.

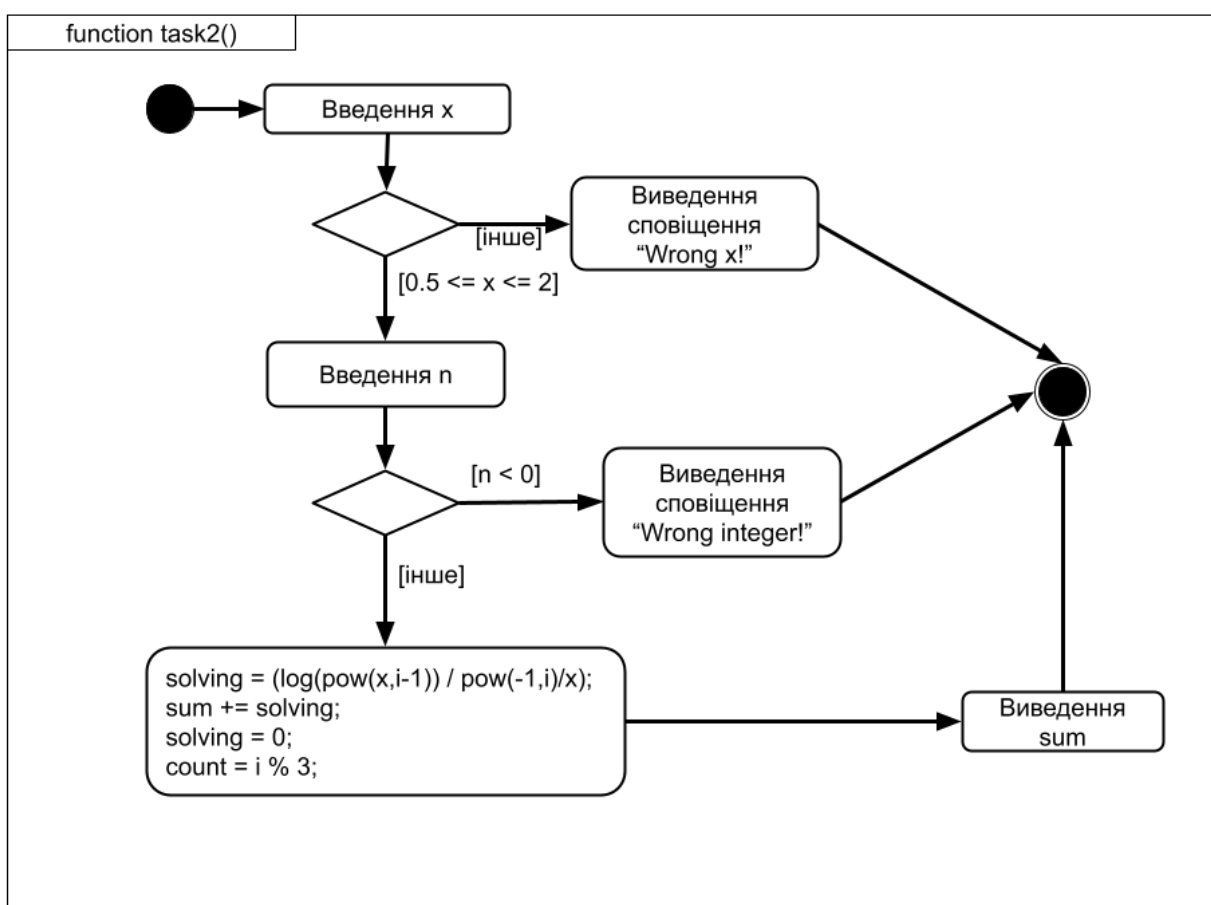


Рисунок 2 – Алгоритм вирішення rangeб

Лістинг коду вирішення задачі rangeб наведено в дод. А (стор. 10).

Екран роботи програми показаний на рис. Б.2.

Завдання 3.

Вирішення задачі range14.

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

x — невідоме число, дійсний тип.

Вихідні дані (ім'я, опис, тип):

sum — сума діапазону, дійсний тип.

Алгоритм вирішення показано на рис. 3.

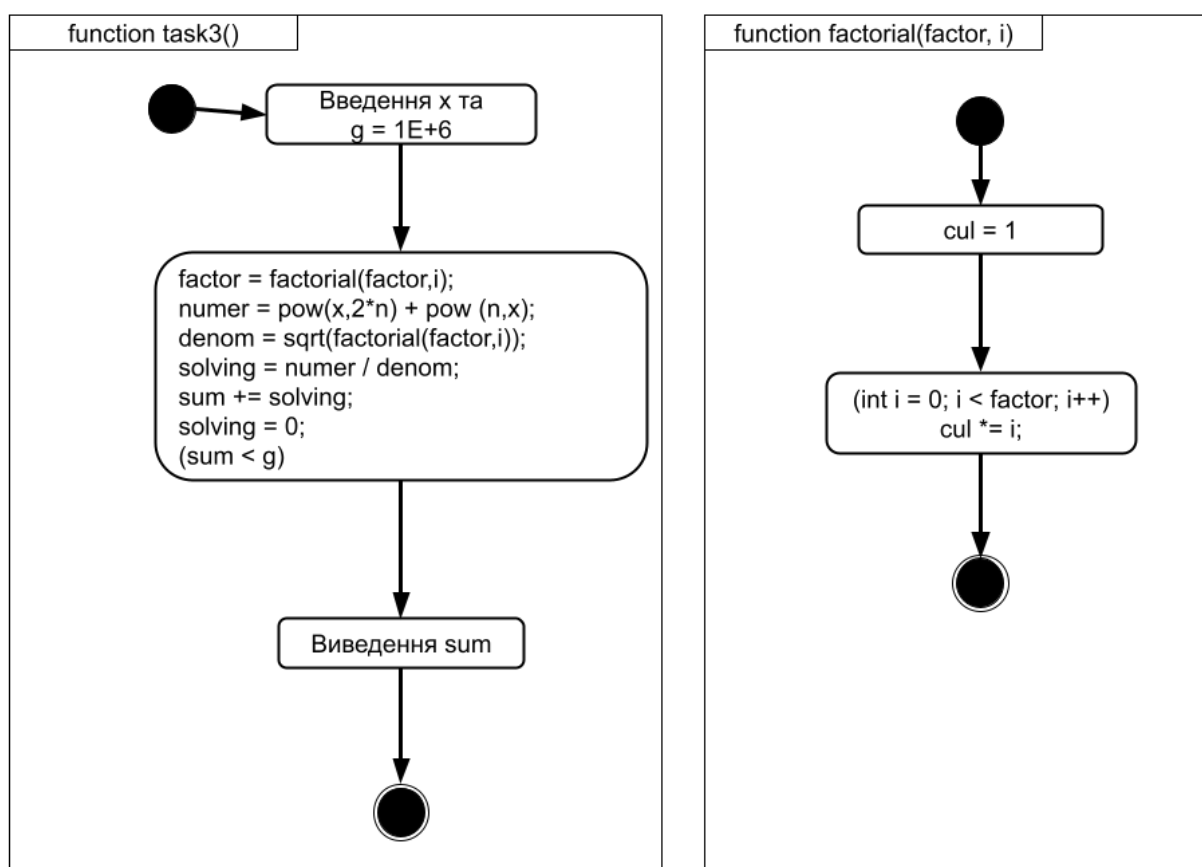


Рисунок 3 – Алгоритм вирішення range14

Лістинг коду вирішення задачі range14 наведено в дод. А (стор. 11).

Екран роботи програми показаний на рис. Б.3.

Завдання 4.

Вирішення задачі Menu.

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

1) task\_num — номер завдання, цілі числа.

Вихідні дані (ім'я, опис, тип):

в залежності від введеного числа, перехід до виконання задач 1, 2, 3, якщо вводиться 0 — вихід з програми.

Алгоритм вирішення показано на рис. 4.

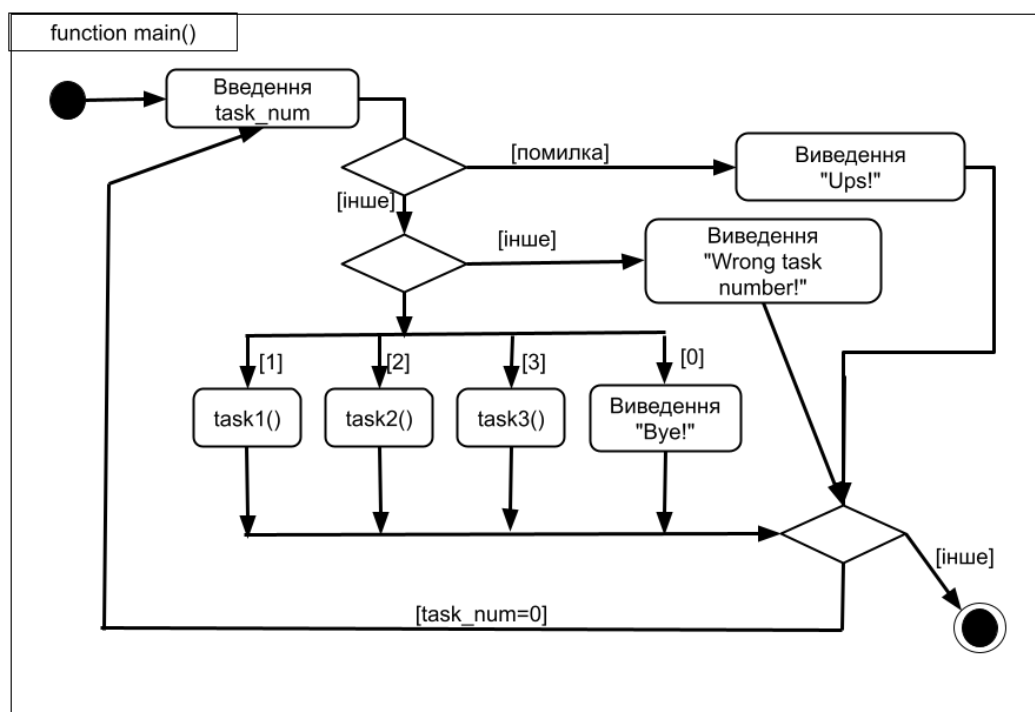


Рисунок 4 – Алгоритм вирішення Menu

Лістинг коду вирішення задачі Menu наведено в дод. А (стор. 9).

Екран роботи програми показаний на рис. Б.4.

## ВИСНОВКИ

Було вивчено теоретичний матеріал із синтаксису мовою C ++ і поданням у вигляді блок-схем циклічних алгоритмів і реалізовано алгоритми з використанням інструкцій циклу з перед-умовою, циклу з після-умовою і параметричного циклу мовою C ++ в середовищі Visual Studio. Відпрацьовано на коді вирішення математичних виразів. Отримано навички з структурування коду для полегшого читання та відпрацювання та створювання блок-схем.



## ДОДАТОК А

### Лістинг коду програми

```

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

void task1(); // task1
void task2(); // task2
void task3(); // task3
int factorial(int factor, int i); // factorial calculation
// start
int main() // menu
{
    int task_num; // diclaration integer
    do {
        cout << " Enter task number (0 - exit): ";
        cin >> task_num; // input integer
        if (!cin) {
            cout << " Ups!" << endl; continue; // error notification
        }
        switch (task_num)
        {
            case 1 : task1(); break; // task1
            case 2 : task2(); break; // task2
            case 3 : task3(); break; // task3
            case 0 : cout << " Bye!" << endl; break; // exit from the program
            default : cout << " Wrong task number!" << endl;
                // output for incorrect numbers
        };
    } while (task_num != 0); // end the program
    return 0;
}
// begin
void task1 () // geom35
{
    cout << " geom35 " << endl; // task notification
    float x, y, r, xc, yc; // declaration integers
    bool ar1, ar2;
    int n, count = 0;
    cout << " Number of n: ";
    cin >> n; // input number of attempts

    cout << " Enter the radius: ";
    cin >> r; // input number of radius
    for (int i = 1; i <= n; i++)
    {

        if (r > 0) // data is correct
    
```

```

{
    cout << " Enter x: ";
    cin >> x; // input x
    cout << " Enter y: ";
    cin >> y; // input y

    xc = x - r / sqrt(2); // intermediate data
    yc = y - r / sqrt(2);
    ar1 = x < 0 && pow(xc, 2) + pow(yc, 2) < pow(r, 2); // calculation for
first figure
    ar2 = x > y-r*sqrt(2) && y < 0 && x < 0 && y > -1*r*sqrt(2) && x < y
&& y > -x-r*sqrt (2);
    // calculation for second figure
    if (ar1 || ar2) { // the point falls into area
        count++; // increasing the number of correct attempts
    }
    else {
        cout << " Wrong!" << endl; // the point does not fall into the area
    }
    cout << " The number of successful attempts " << count << " from " << n
<< endl;
    } // output
    else {
        cout << " Wrong radius! " << endl; //error notification
    }
}
} //end

void task2 () // range6
{ // begin
    cout << " range6 " << endl;

    int i = 1, n; // diclaration integers
    double x, solving;
    cout << " Enter x: ";
    cin >> x; // input x
    if ( 0.5 > x || x > 2){
        cout << " Wrong x! " << endl;
    } // error notification
    else // data is correct
    {
        cout << " Limit n: ";
        cin >> n; // input n
        long double sum = 0;          cout.precision(10); // no more than 10
digits after the decimal point
        if (!cin || n < 0)
        {
            cout << " Wrong integer!" << endl;
        } // error notification
        else

```

```

{
    for (i; i <= n; i++)
    { // calculation
        int count = 0;
        long double solving = (log(pow(x,i-1)) / pow(-1,i)/x);
        sum += solving;
        solving = 0;

        count = i % 3;

        if (count == 2) {
            cout << " Sum of range: " << sum << endl;
        } // output results
    }
}

}

} // end

void task3 () // range14
{ // begin
    cout << " range14 " << endl;
    int n = 1, i; // diclaration
    double numer, denom, x, g = 1E+6;
    cout << " Enter x: ";
    cin >> x; // input x

    cout << " Range of the row sum: ";
    cin >> g; // input g

    long double sum = 0;

    do
    {
        n++;
        int factor = factorial(factor,i); // factorial
        int count = 0;
        long double numer = pow(x,2*n) + pow (n,x); // calculation numerator
        long double denom = sqrt(factorial(factor,i)); // calculation
denominator
        // calculation
        long double solving = numer / denom;
        sum += solving;
        solving = 0;
        count = n % 3;
        if (count == 2) {
            cout << " Sum of range: " << sum << endl;
            // output results
        }
    } while (sum < g);
} //end

```

```
int factorial(int factor, int i) // factorial calculation
{ // begin
    int cul = 1;
    for(int i = 0; i < factor; i++)
    {
        cul *= i;
    }
    return cul;
} // end
// finish
```

ДОДАТОК Б  
Скрін-шоти вікна виконання програми

```
geom35
Number of n: 3
Enter the radius: 2
Enter x: -3
Enter y: -2
Wrong!
The number of successful attempts 0 from 3
Enter x: -1
Enter y: -0.5
The number of successful attempts 1 from 3
Enter x: -0.2
Enter y: -0.1
The number of successful attempts 2 from 3
Enter task number (0 - exit): 1
geom35
Number of n: 3
Enter the radius: -1
Wrong radius!
Wrong radius!
Wrong radius!
Enter task number (0 - exit): 0
Bye!
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання  
geom35

```
range6
Enter x: 2
Limit n: 54
Sum of range: 0.3465735903
Sum of range: -0.6931471806
Sum of range: 1.386294361
Sum of range: -1.732867951
Sum of range: 2.426015132
Sum of range: -2.772588722
Sum of range: 3.465735903
Sum of range: -3.812309493
Sum of range: 4.505456674
Sum of range: -4.852030264
Sum of range: 5.545177444
Sum of range: -5.891751035
Sum of range: 6.584898215
Sum of range: -6.931471806
Sum of range: 7.624618986
Sum of range: -7.971192576
Sum of range: 8.664339757
Sum of range: -9.010913347
Enter task number (0 - exit): 2
range6
Enter x: -1
Wrong x!
Enter task number (0 - exit): 2
range6
Enter x: 2
Limit n: -3
Wrong integer!
Enter task number (0 - exit): 0
Bye!
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання  
range6

```
Enter task number (0 - exit): 3
range14
Enter x: 9
Sum of range: 7073
Enter task number (0 - exit): 3
range14
Enter x: 573
Sum of range: 3.09163e+172
Enter task number (0 - exit): 0
Bye!
```

Рисунок Б.3 – Екран виконання програми для вирішення завдання range14

```
Enter task number (0 - exit): 1
geom35
Number of n: 2
Enter the radius: 4
Enter x: -1
Enter y: -2
Wrong!
The number of successful attempts 0 from 2
Enter x: -2
Enter y: -1
The number of successful attempts 1 from 2
Enter task number (0 - exit): 3
range14
Enter x: 456
Sum of range: 1.86071e+137
Enter task number (0 - exit): -3
Wrong task number!
Enter task number (0 - exit): 0
Bye!
```

Рисунок Б.4 – Екран виконання програми для вирішення завдання Menu