

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. 175. 318. 08 ЛР

Виконав студент гр. _____ 318

_____ Каріна ГЛЄБОВА
(підпис, дата) (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

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).

Geom 4. Дано дійсні числа (x_i, y_i) , $i = 1, 2, \dots, n$, – координати точок на площині. Визначити кількість точок, що потрапляють в фігуру червоного кольору.

На рис. 1 показана фігура із завдання 1.

a - сторона трикутника

r - радіус кола

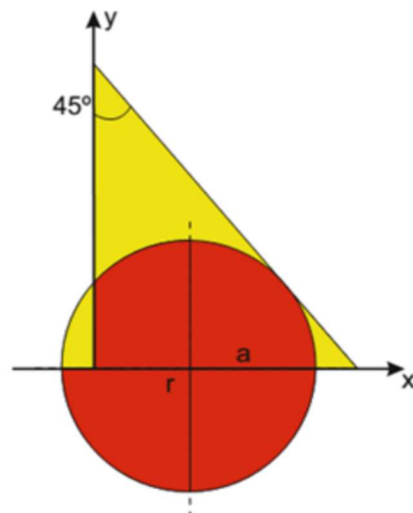


Рисунок 1 – Фігура із завдання 1

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

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

Таб. 2, №8.

На рис. 2 показаний вираз для завдання 2.

$$\frac{(1-x)(1-2x)\dots(1-nx)}{1+2x)(1+4x)\dots(1+2nx)}, 0 \leq x \leq 2$$

Рисунок 2 – Вираз для завдання 2

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

Таб. 3, №6.

На рис. 3 показаний вираз для завдання 3.

$$\sum_{n=1}^{\infty} \frac{3n * n!}{n^n}$$

Рисунок 3 – Вираз для завдання 3

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 Geom 4.

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

double r – радіус кола, дійсний із подвійною точністю тип.

int n – кількість точок у колі, дійсний тип.

double x – координата точки, дійсний із подвійною точністю тип.

double y – координата точки, дійсний із подвійною точністю тип.

Вихідні дані (ім'я, опис, тип):

float cath_small_yell_tris – дійсне число, четверта частина радіуса, відрізок жовтої частини на колі.

float dist_2_chord – дійсне число, відрізок радіуса без урахування жовтої частини.

double zmensh – дійсне число із подвійною точністю, зменшуване у формулі знаходження повної площі хорди.

double vdmk – дійсне число із подвійною точністю, зменшуване у формулі знаходження повної площі хорди.

double S_chord – дійсне число із подвійною точністю, площа повної хорди.

double area_yellow – дійсне число із подвійною точністю, площа жовтої частини на колі.

На рис. 3 показаний приклад діаграми для завдання Geom 4.

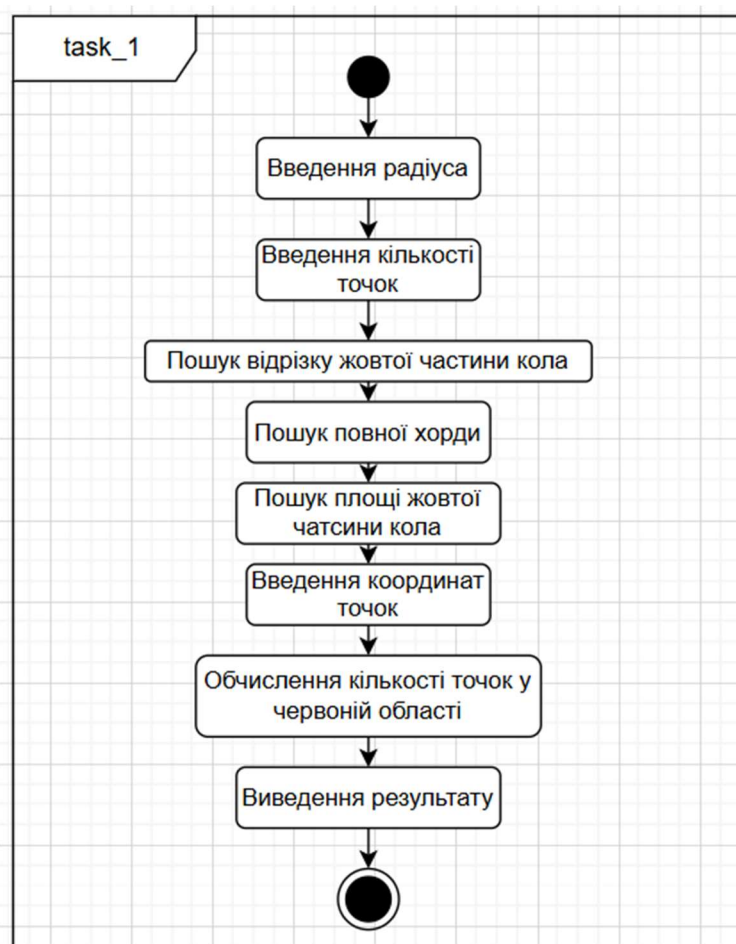


Рисунок 3 – Діаграма для завдання Geom 4

Лістинг коду вирішення завдання 1 наведено в дод. А (стор. 9-11)

Екран роботи програми показаний на рис. Б.1 (додат. Б, стор. 12)

Завдання 2.

Вирішення задачі 8 із таблиці 2.

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

int x – ціле число, дійсний тип.

int n – ціле натуральне число, дійсний тип.

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

double num – чисельник, дійсний із подвійною точністю тип.

double denom – знаменник, дійсний із подвійною точністю тип.

double result – значення виразу, дійсний із подвійною точністю тип, частка від ділення чисельника на знаменник.

int i – елемент послідовності, дійсний тип.

На рис. 4 показаний приклад діаграми для завдання 2.

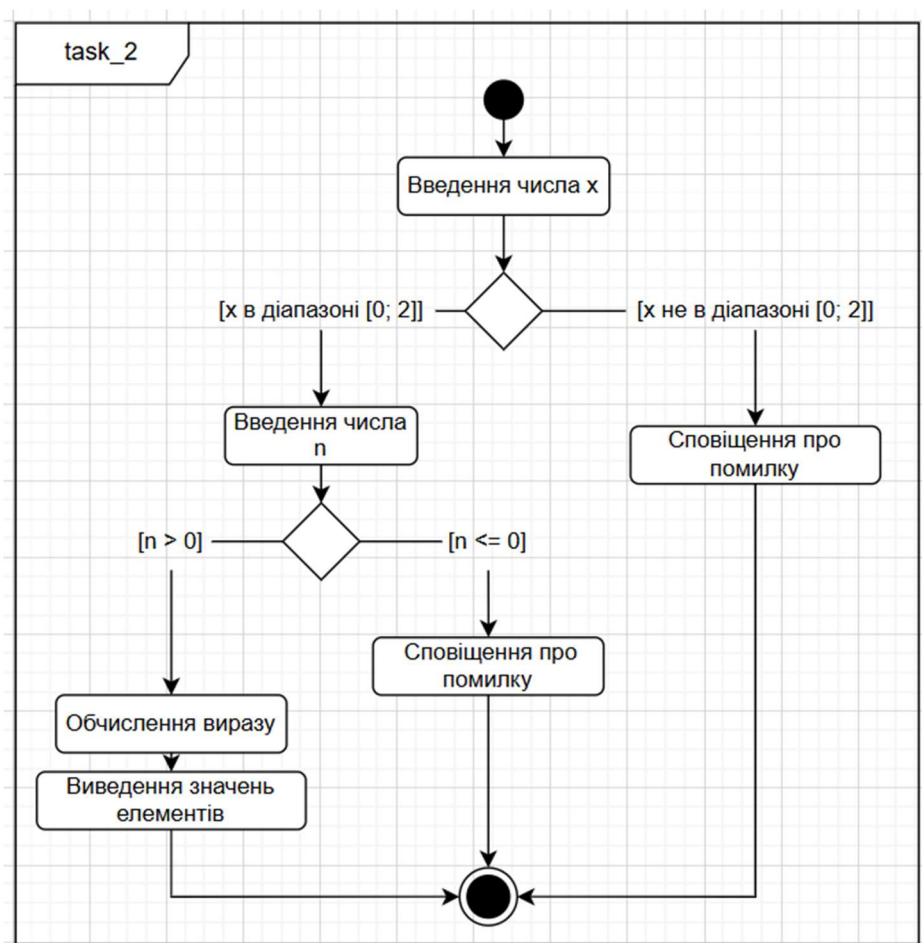


Рисунок 4 – Діаграма для завдання 2

Лістинг коду вирішення завдання 2 наведено в дод. А (стор. 9-11).

Екран роботи програми показаний на рис. Б.2 (додат. Б, стор. 12)

Завдання 3.

Вирішення завдання 6 із таблиці 3.

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

`const double e = 1e-10` – константа, мала величина для переривання циклу обчислення суми збіжного ряду, дійсний із подвійною точністю тип.

`const double g = 1e5` – константа, величина для переривання циклу обчислення суми розбіжного ряду, дійсний із подвійною точністю тип.

`const int M = 1000` – константа, крок виводу результатів, дійсний тип.

`const int maxIterations = 1000000` – константа, максимальна кількість ітерацій, дійсний тип.

`int n` – початковий індекс, дійсний тип.

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

`double sum` – сума ряду, дійсний із подвійною точністю тип.

На рисунку 5 показаний приклад діаграми для завдання 3

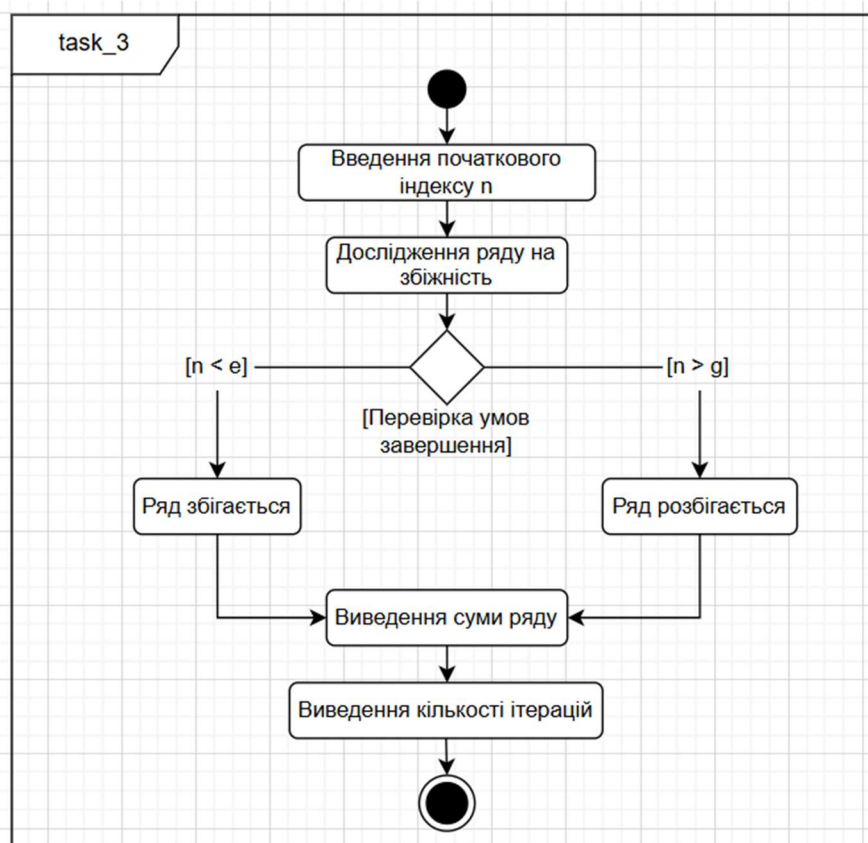


Рисунок 5 – Діаграма для завдання 3

Лістинг коду вирішення задачі 6 із таблиці 3 наведено в дод. А (стор. 9-11).

Екран роботи програми показаний на рис. Б.3 (додат. Б, стор. 12)

Завдання 4.

Організація меню.

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

“Номер завдання:” – введення номеру завдання.

Вихідні дані (ім'я, опис, тип):

task_1 – якщо ввели число «1», виводяться розрахунки задачі Geom 4.

task_2 – якщо ввели число «2», виводяться розрахунки задачі 8 із таблиці 2.

task_3 – якщо ввели число «3», виводяться розрахунки задачі 6 із таблиці 3.

На рис. 6 показаний приклад діаграми для завдання 4

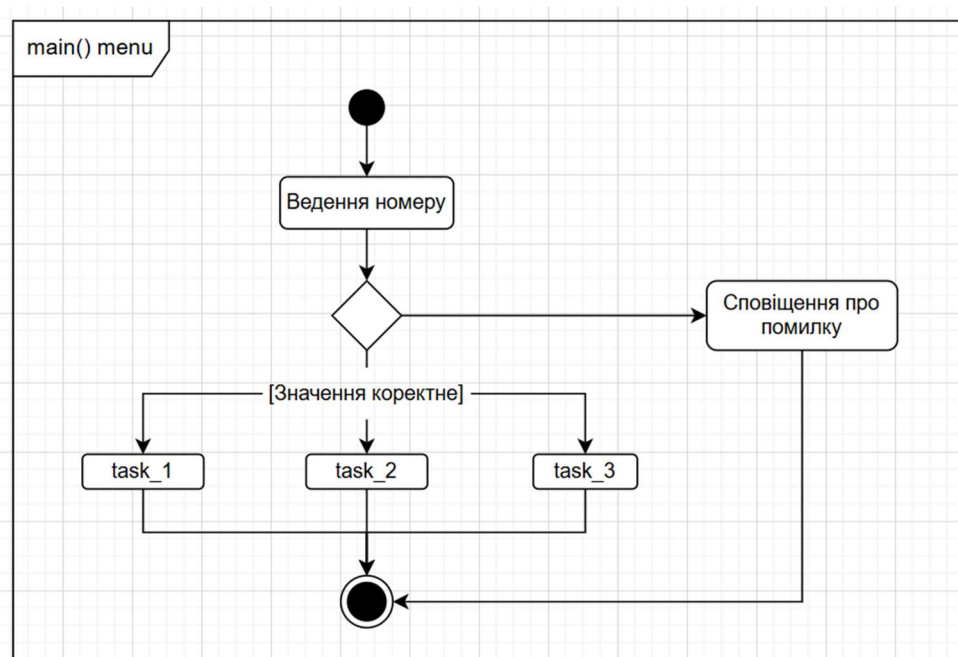


Рисунок 6 – Діаграма для завдання 4

Лістинг коду вирішення завдання 4 наведено в дод. А (стор. 9-11).

ВИСНОВКИ

Було вивчено теоретичний матеріал із синтаксису мови C++ та опрацьовано подання у вигляді UML діаграм алгоритмів. Було реалізовано алгоритми з використанням інструкцій циклу з передумовою, циклу з післяумовою і параметризованого циклу мовою C++ в середовищі Visual Studio.

ДОДАТОК А

Лістинг коду програми

```

#include <iostream>
#include "windows.h"
#include <cmath>
#include <iomanip> // бібліотека для форматування виводу
using namespace std;

//Geom4. Дано дійсні числа  $(X_i, Y_i)$ ,  $i = 1, 2, \dots, n$ , – координати точок на площині.
Визначити кількість
//точок, що потрапляють в фігуру червоного кольору.
void task_1(); //завдання 1, декларація функції

//Таб.2, #8. Дано дійсне число  $x$  і натуральне число  $n$ . Необхідно обчислити значення
виразу
//при заданих  $x$  і  $n$  та вивести для парних варіантів значення кожного третього, для
непарного –
//кожного четвертого елемента.
void task_2(); //завдання 2, декларація функції

// Таб.3, #6. Дослідити ряд на збіжність.
void task_3(); // завдання 3, декларація функції

int main() {
    SetConsoleOutputCP(1251);
    int menu; // зміна для номеру завдання
    do
    { // початок циклу
        cout << "Номер завдання:"; //введення номеру завдання
        cin >> menu; // обирання номеру завдання
        cout << endl; //вільна строка
        switch (menu) {
            case 1: task_1(); break; // 1 – завдання 1
            case 2: task_2(); break; // 2 – завдання 2
            case 3: task_3(); break; // 3 – завдання 3
            case -1: cout << "Вихід..." << endl; break; // -1 – вихід
            default: cout << "Помилка! Лише 1, 2 або 3!" << endl; // інший номер –
повторити
        }
        cout << endl;
    } // кінець циклу
    while (menu != -1); // умова виконання циклу
    return 0;
}

// функція перевірки входження точки в коло
bool isInCircle(int x, int y, int r, float area_yellow) {
    double distance = x * x + y * y;
    return (distance <= r * r) || (x <= 0 && y >= 0 && distance <= area_yellow) || (x
<= 0 && y <= 0 && distance <= r * r) || (x >= 0 && y <= 0 && distance <= r * r);
}

void task_1() {
    double r;
    int n;

    cout << "Введіть радіус кола: "; // введення радіуса
    cin >> r;

    cout << "Введіть кількість точок: "; // введення кількості точок

```

```

cin >> n;

// знаходимо відрізок жовтої частини кола
float cath_small_yell_tris = r / 4;
float dist_2_chord = r - cath_small_yell_tris;

// знаходимо площу повної хорди
double zmesh = r * r * acos(dist_2_chord / r);
double vdmk = dist_2_chord * sqrt(r * r - dist_2_chord * dist_2_chord);

double S_chord = zmesh - vdmk;
double area_yellow = S_chord / 2; // площа жовтої частини на колі

int count = 0; // лічильник точок у червоній фігурі

// цикл для введення точок
for (int i = 0; i < n; ++i) {
    double x, y;
    cout << "Введіть координати точки " << i + 1 << " (x, y): ";
    cin >> x >> y;

    if (isInCircle(x, y, r, area_yellow)) {
        count++;
    }
}

cout << "Кількість точок у червоній області: " << count << endl;
}

void task_2() { // завдання 2, реалізація
    double x;
    int n;

    cout << "Введіть значення x: "; // введення числа x
    cin >> x;
    if (x < 0 || x > 2) { //перевірка на коректність
        cout << "Помилка! x має бути в діапазоні [0; 2]." << endl; // сповіщення про
        помилку
        return;
    }
    cout << "Введіть натуральне число n: "; // введення числа n
    cin >> n;
    if (n <= 0) { //перевірка на коректність
        cout << "Помилка! n має бути натуральним числом." << endl;
        return;
    }

    double num = 1.0; // чисельник
    double denom = 1.0; // знаменник

    for (int i = 1; i <= n; i++) {
        num *= (1 - i * x); // обчислення чисельника
        denom *= (1 + 2 * i * x); // обчислення знаменника
    }
    double result = num / denom; // розрахунок виразу
    cout << "Значення виразу: " << result << endl; // виведення результату

    cout << "Елементи послідовності: " << endl; // виведення значень елементів
    if (n % 2 == 0) {
        for (int i = 3; i <= n; i += 3) {
            cout << "Елемент " << i << ": " << (1 - i * x) << endl; // для парних
            виведення кожного третього елемента
        }
    }
}

```

```

    }
}
else {
    for (int i = 4; i <= n; i += 4) {
        cout << "Елемент " << i << ": " << (1 - i * x) << endl; // для непарних
// виведення кожного четвертого елемента
    }
}
}

double term(int n) { // функція для обчислення n-го члена ряду
    return (3.0 * n * tgamma(n + 1)) / pow(n, n); // tgamma - факторіал
}

void task_3() { // завдання 2, реалізація
    const double e = 1e-10; // мала величина для збіжності
    const double g = 1e5; // верхня межа для розбіжності ряду
    const int M = 1000; // крок виводу результатів
    const int maxIterations = 1000000; // максимальна кількість ітерацій

    double sum = 0.0; //сума ряду

    int n;
    cout << "Введіть початковий індекс n:"; // введення індекса
    cin >> n;

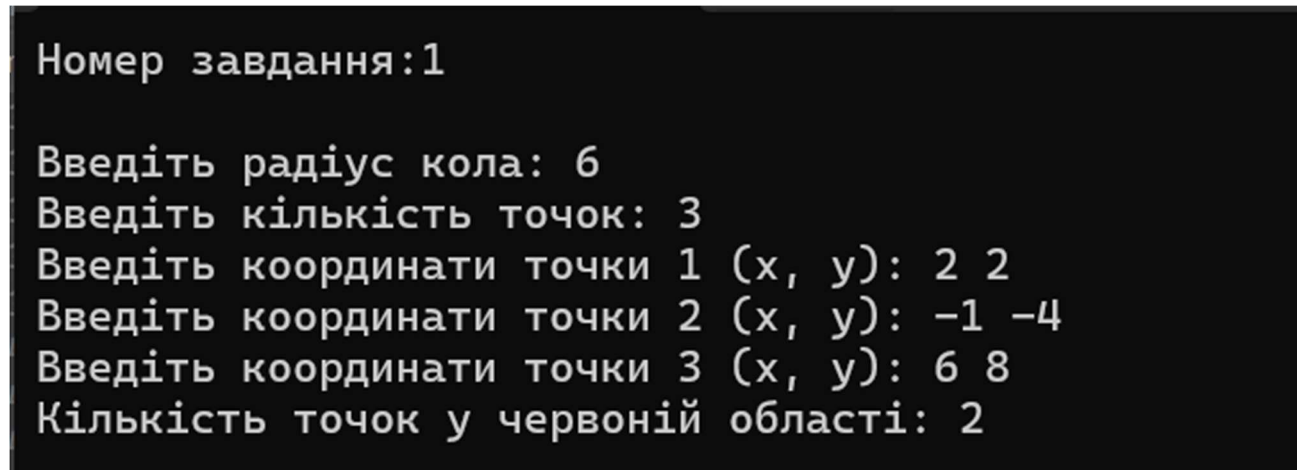
    cout << fixed << setprecision(15); // форматування до 15 знаків після коми
    cout << "Дослідження ряду на збіжність:\n\n";

    while (n <= maxIterations) {
        double currentTerm = term(n); // перевірка умов завершення
        if (fabs(currentTerm) < e) {
            cout << "Ряд збігається на " << n << "-й ітерації.\n";
            break;
        }
        if (fabs(currentTerm) > g) {
            cout << "Ряд розбігається на " << n << "-й ітерації.\n";
            break;
        }
        sum += currentTerm; // додавання поточного члена до суми
        if (n % M == 0) {
            cout << "Ітерація" << n << ", поточна сума: " << sum << "\n"; //
        }
        n++;
    }
    cout << "\nКінцева сума ряду: " << sum << endl; // виведення остаточного
// результату
    cout << "Кількість ітерацій: " << n << endl;
}

```

ДОДАТОК Б
Скрін-шоти вікна виконання програми

На рис. Б.1 показаний приклад виконання завдання 1.

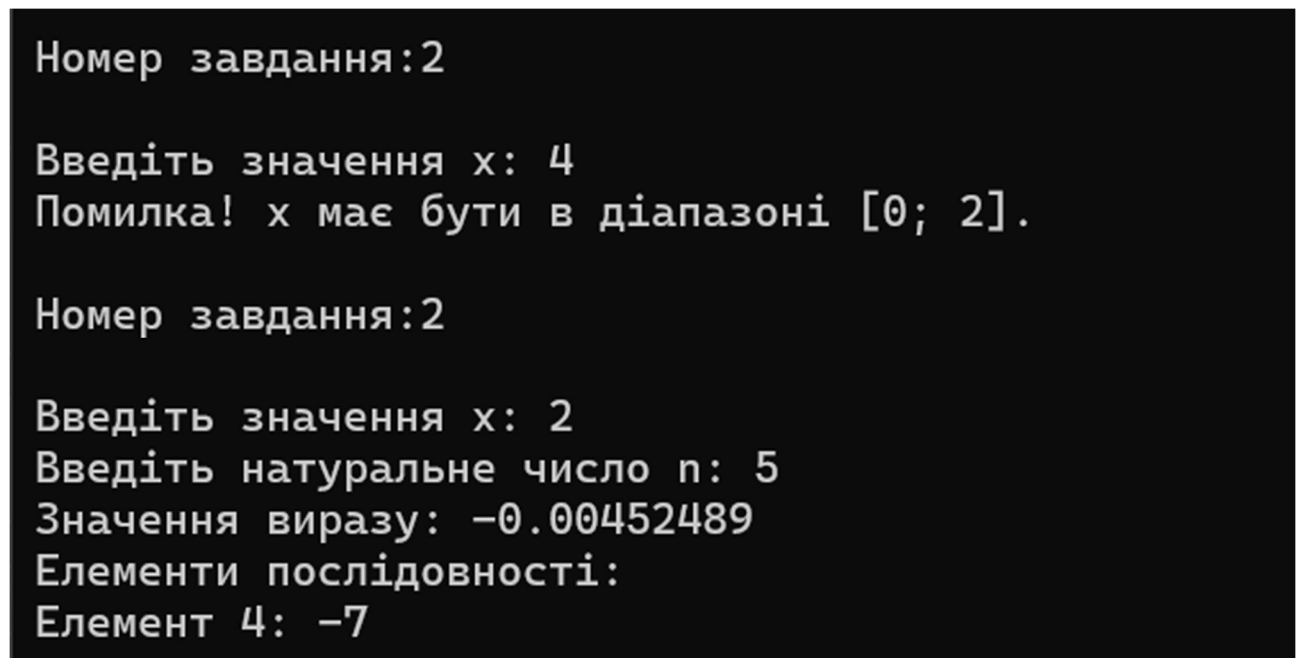


```
Номер завдання:1

Введіть радіус кола: 6
Введіть кількість точок: 3
Введіть координати точки 1 (x, y): 2 2
Введіть координати точки 2 (x, y): -1 -4
Введіть координати точки 3 (x, y): 6 8
Кількість точок у червоній області: 2
```

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

На рис. Б.2 показаний приклад виконання завдання 2.



```
Номер завдання:2

Введіть значення x: 4
Помилка! x має бути в діапазоні [0; 2].

Номер завдання:2

Введіть значення x: 2
Введіть натуральне число n: 5
Значення виразу: -0.00452489
Елементи послідовності:
Елемент 4: -7
```

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

На рис. Б.3 показаний приклад виконання завдання 3.

Номер завдання:3

Введіть початковий індекс n:2

Дослідження ряду на збіжність:

Ряд збігається на 31-й ітерації.

Кінцева сума ряду: 7.209057624792099

Кількість ітерацій: 31

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