Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 3 з дисципліни

«Основи програмування

»

Варіант 13

Виконав студент Калашніков Андрій Євгенович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

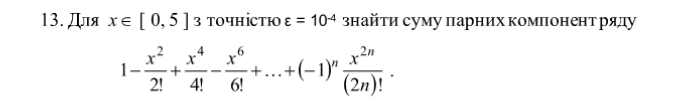
Київ 2021

## Лабораторна робота 3

**Мета:**

Вивчити особливості організації ітераційних циклів.

**Постановка задачі**:



**Математична модель**:

Складемо таблицю імен змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Зміна | Тип | Ім’я | Призначення |
| x | double | Аргумент | Проміжне значення |
| fact | double | Факторіал | Проміжне значення |
| n | double | Порядковий номер члена | Проміжне значення |
| eps | double | Точність | Проміжне значення |
| sum | int | Поточна сума/Член | Проміжне значення |
| sum\_result | double | Результуюча сума | Результат |

Утворимо цикл, який буде працювати доти, допоки модуль sum не буде меншим за 10^(-4). Скористаємось циклом do while.

|  |
| --- |
|  |

Факторіал порахуємо за формулою: fact \*= (2 \* n - 1) \* 2 \* n

Поточну суму порахуємо за формулою: sum = (pow(-1, n) \* pow(x, (2 \* n)) / fact)

, де **pow() піднесення до степеня.** Перевіримо чи є член парним за формулою:

n % 2 == 0. Результуючу суму: sum\_result += sum. Кожну ітерацію збільшуємо n на 1.

**Код на С++ :**

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian"); //Підключаємо для підтримки кирилиці

double x;

double fact = 1; //факторіал

double sum;

double sum\_result = 1;

int n = 1; //порядковий номер члена послідовності

double eps = 0.0001; //точність

cout << "Enter x[0;5] x=";

cin >> x;

if (x != 0)

{

do {

fact \*= (2 \* n - 1) \* 2 \* n; //Знаходження факторіала

sum = (pow(-1, n) \* pow(x, (2 \* n)) / fact); //Знаходження sum

if (n % 2 == 0) //Перевірка на парність

{

sum\_result += sum;

cout << "Sum is=" << sum\_result << endl;

}

n += 1;

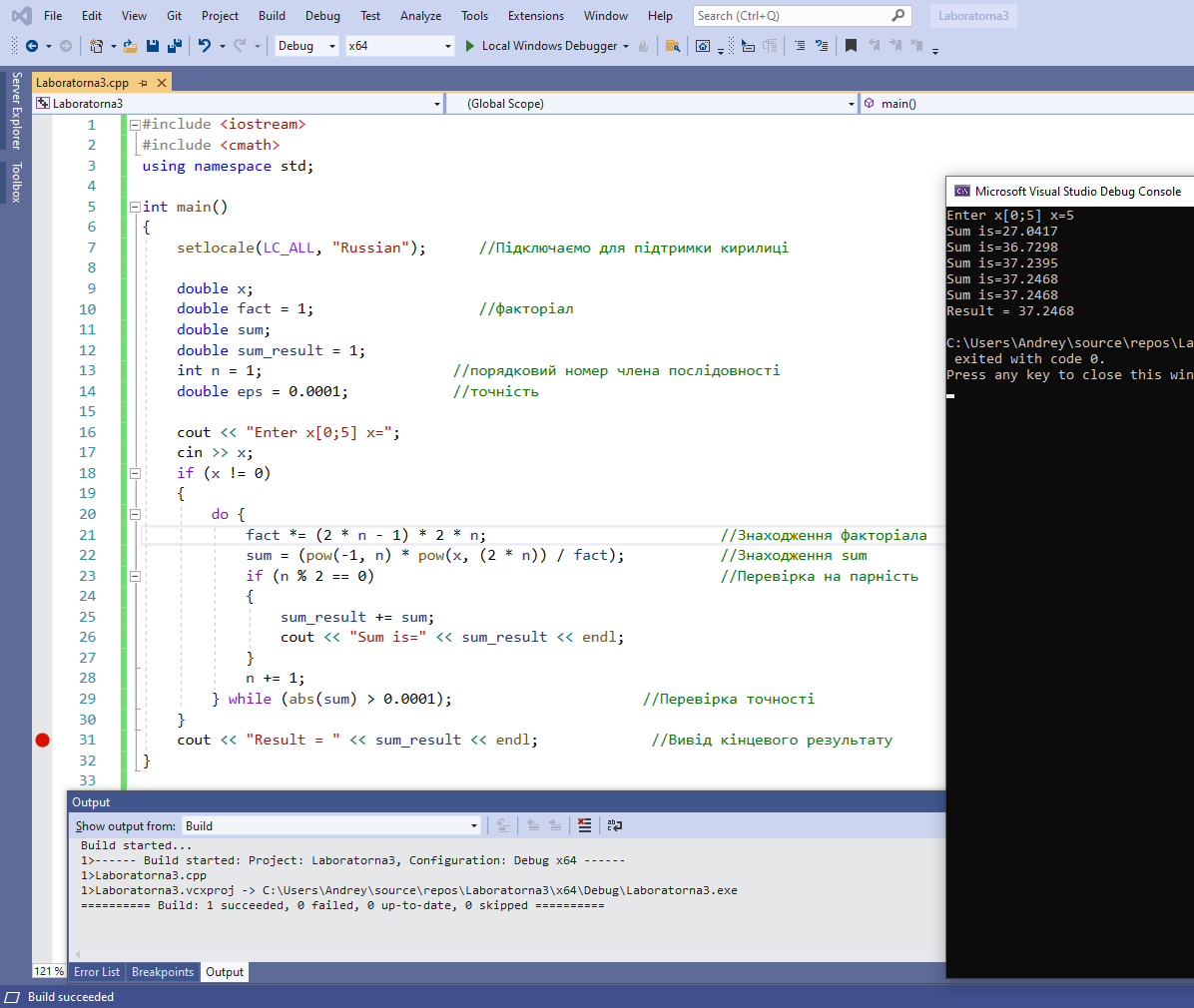
} while (abs(sum) > 0.0001); //Перевірка точності

}

cout << "Result = " << sum\_result << endl; //Вивід кінцевого результату

}

**Скріншот результату роботи:**

****

**Висновок:**

Під час лабораторної роботи ми дослідили організацію циклічних процесів та ітераційні цикли на прикладі C++. Математична модель, код, результати наведені. Програма правильно обчислює суму парних компонентів ряду та видає результат.