Основи програмування

*Додаток 1*

Міністерство освіти і науки України

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

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації

і управління

Звіт

з лабораторної роботи № 3 з дисципліни

«Основи програмування-1

Базові конструкції»

« Організація циклічних процесів. Ітераційні цикли»

Варіант     19

Виконав студент       ІП-02 Геращенко Дмитро Ігорович

(шифр, прізвище, ім'я, по батькові)

Перевірив              Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

Київ 2020

# Лабораторна робота 3 Організація циклічних процесів. Ітераційні цикли Варіант 19

**Мета** – вивчити особливості організації ітераційниї циклів.

1. Постановка задачі

Обчислити

,

для 0 ≤ x ≤ 2 з точністю до члена ряду, що менше 10-5

1. Текст програми

Python:

**def** **main**():

x = float(input("Введіть змінну x: "))

epsilon = float(input("Введіть точність Ɛ: "))

n = **0** # лічильник поточного елементу

fact = **1** # факторіал з 2ⁿ

prev = float("inf") # попередній елемент aₙ₋₁

curr = (x\*\***2** - **1**) / **2** # обчислюємо перший елемент (при n=0)

s = curr # обчислюємо суму (при n=0)

print("n", "an".center(**18**), "Δa".center(**18**), "∑a".center(**18**))

print("=" \* **58**)

print(f"{n} {curr:18.15f} {'':18} {s:18.15f}")

**while** abs(curr - prev) >= epsilon: # поки не достигнута точність Ɛ

n += **1**

**for** i **in** range(**2**\*\*(n-**1**) + **1**, **2**\*\*n + **1**): # обчислюємо факторіал 2ⁿ

fact \*= i

prev = curr # зберігаємо попередній елемент

curr = ((n+**1**) \* x\*\***2** - **1**) / (**1** + fact) # обчислюємо поточний елемент aₙ

s += curr # обчислюємо суму ряду

print(f"{n} {curr:18.15f} {curr-prev:18.15f} {s:18.15f}")

print(f"**\n**Сума безкінечного збіжного ряду дорівнює{s:18.15f}")

**if** \_\_name\_\_ == "\_\_main\_\_":

main()

C++:

#include <stdio.h>

#include <cmath>

#include <limits>

**using** **namespace** std;

**int** **main**() {

**float** x, epsilon;

printf("Введіть змінну x: "); scanf("%f", &x);

printf("Введіть точність Ɛ: "); scanf("%f", &epsilon);

**int** n = **0**; // лічильник поточного елементу

**double** fact = **1.0**, // факторіал з 2ⁿ

prev = numeric\_limits<**double**>::infinity(), // попередній елемент aₙ₋₁

curr = (pow(x, **2**) - **1**) / **2**, // обчислюємо перший елемент (при n=0)

s = curr; // обчислюємо суму (при n=0)

printf("n an Δa ∑a **\n**");

printf("==========================================================**\n**");

printf("%d %18.15lf %18s %18.15lf**\n**", n, curr, "", s);

**while** (fabs(curr - prev) >= epsilon) { // поки не достигнута точність Ɛ

n += **1**;

**for** (**int** i = pow(**2**, n-**1**) + **1**; i <= pow(**2**, n); i++) // обчислюємо факторіал 2ⁿ

fact \*= i;

prev = curr; // зберігаємо попередній елемент

curr = ((n+**1**) \* pow(x, **2**) - **1**) / (**1** + fact); // обчислюємо поточний елемент aₙ

s += curr; // обчислюємо суму ряду

printf("%d %18.15lf %18.15lf %18.15lf**\n**", n, curr, curr-prev, s);

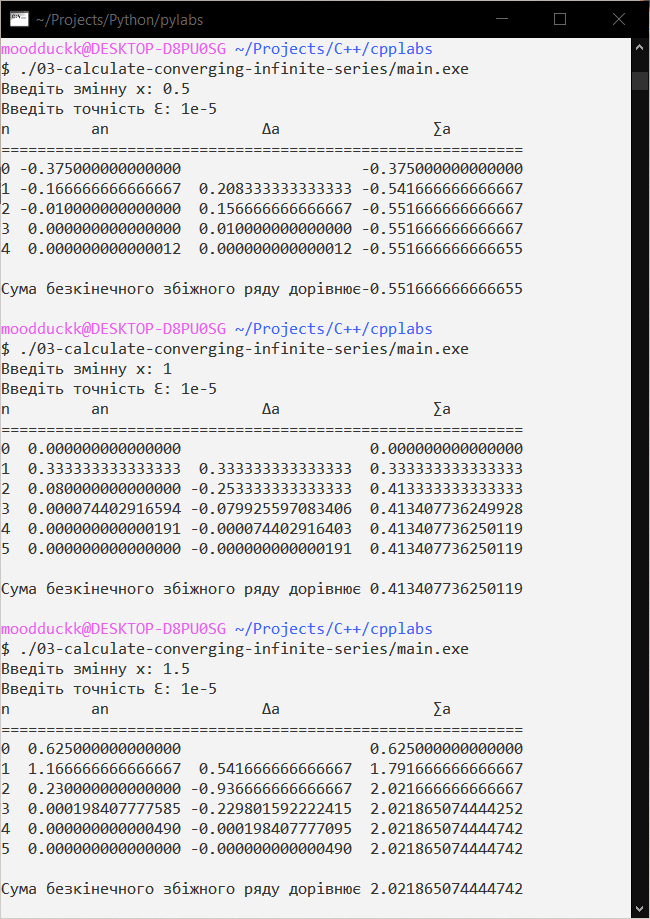
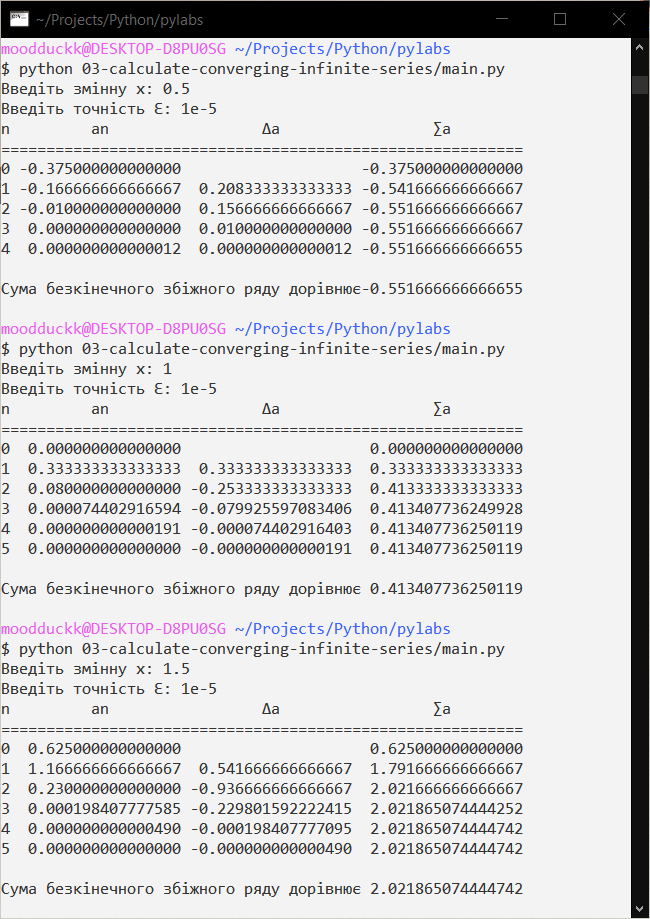
}

printf("**\n**Сума безкінечного збіжного ряду дорівнює%18.15f**\n**", s);

**return** **0**;

}

1. Копії екранних форм результатів роботи



1. Висновок

На цій лабораторній роботі ми використали знання зі застосування ітераційних циклів в обчислювальних процесах та алгоритму обчислення з заданою точністю. В результаті ми написали програму, що обчислює з заданою точністю нескінченний збіжний ряд. Ця програма правильно працює на всіх дійсних x, що входять в діапазон [0; 2], і значення, що обчислює програма, збігаються з графіком залежності даного ряду від x.