# 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 23

Виконав студент Панченко Сергій Віталійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 202

## Лабораторна робота 3

**Постановка задачі**: З точністю 10^(-5) обчислити значення суми

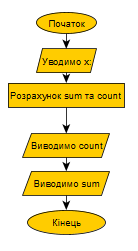
**Математична модель**:

**Складемо таблицю імен змінних**

|  |  |  |  |
| --- | --- | --- | --- |
| Зміна | Тип | Ім’я | Призначення |
| x | double | Аргумент | Проміжне значення |
| sum\_curr | double | Сума поточна | Результат |
| sum\_prev | double | Сума попередня | Проміжне значення |
| powerX | double | Степінь аргументу | Проміжне значення |
| powerOf2 | double | Степінь двійки | Проміжне значення |
| factorialK | double | Факторіал індексу | Проміжне значення |
| member | double | Член послідовності | Проміжне значення |
| count | double | Число членів | Результат |

Утворимо цикл, який буде працювати доти, допоки різниця сум не буде меншою за 10^(-5). Для кожного кроку циклу обчислюємо член послідовності member = (powerX) / (powerOf2 \* factorialK), де відповідні змінні **cтепінь аргументу( pow(x, 2\*k)),** **степінь двійки**(**pow(2, k))** , **факторіал індексу,** який будемо обчислювати за допомогою циклу, де при виході з нього надаємо факторіалу значення «1», щоб при наступній ітерації одні й ті самі числа не помножувалися два рази. **Pow – функція піднесення степеню.**

**Блок-схема:**



**Псевдокод:**

Крок 1: Визначимо основні дії;

Крок 2: задаємо початкові умови для входження в цикл

Крок 3: Деталізуємо основні дії циклу розрахунку суми послідовності;

Крок 4: Деталізуємо знаходження факторіалу;

**Крок 1**

**Початок**

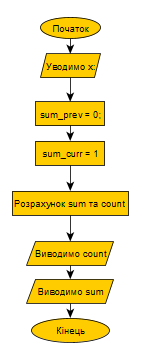
Уводимо X

Розраховуємо в циклі суму та кількість чисел;

Виводимо ці значення;

**Кінець**

**Крок 2**



**Початок**

Уводимо Х

sum\_prev = 0;

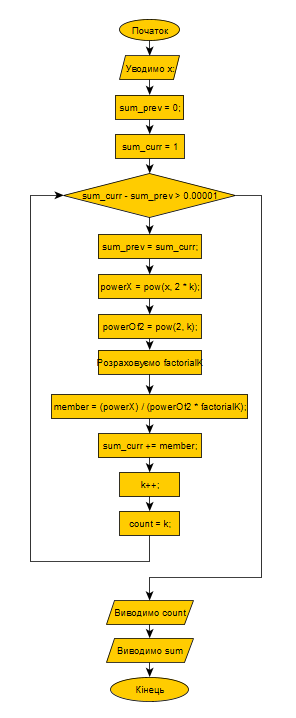
sum\_curr = 1

Розраховуємо в циклі суму та кількість чисел;

Виводимо ці значення;

**Кінець**

**Крок 3:**



**Початок**

Уводимо Х

sum\_prev = 0;

sum\_curr = 1;

Поки (sum\_curr - sum\_prev > 0.00001):

sum\_prev = sum\_curr;

powerX = pow(x, 2 \* k);

powerOf2 = pow(2, k);

Розраховуємо факторіал K

member = (powerX) / (powerOf2 \* factorialK);

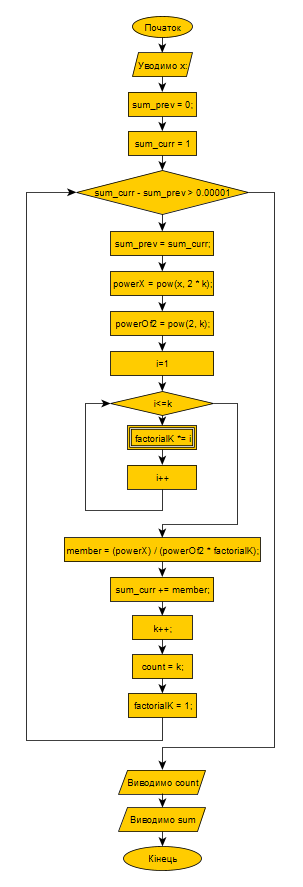
sum\_curr += member;

k++;

count = k;

Виводимо ці значення;

**Кінець**



**Крок 4:**

**Початок**

Уводимо Х

sum\_prev = 0;

sum\_curr = 1;

**Поки** (sum\_curr - sum\_prev > 0.00001):

sum\_prev = sum\_curr;

powerX = pow(x, 2 \* k);

powerOf2 = pow(2, k);

i=1

**Поки** i<=k:

factorialK \*= i

i++

member = (powerX) / (powerOf2 \* factorialK);

sum\_curr += member;

k++;

count = k;

Виводимо ці значення;

**Кінець**

Створимо таблицю для перевірки:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| k | powerOf2 | powerOfX(x=3) | factorialK | member |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 2 | 9 | 1 | 4.5 |
| 2 | 4 | 81 | 2 | 10.125 |
| 3 | 8 | 729 | 6 | 15.1875 |

Отже, вручну розрахувавши члени, бачимо, що алгоритм працює. Число членів виводяться. При x = 3, їх count = 19.

**Висновок:**

Під час лабораторної роботи ми дослідили організацію циклічних процесів та ітераційні цикли. Математична модель, блок-схема, результати наведені. Оскільки формула перевірені вручну результати розрахунку членів послідовності відповідають дійсності, то алгоритм правильно подає результат.