# 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-1.

Базові конструкції»

«Організація циклічних процесів. Ітераційні цикли»

Варіант 20

Виконав студент Лошак Віктор Іванович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 3**

**Мета:**

Вивчити особливості організації ітераційних циклів.

**Тема:**

Для заданого х і , знайти з точністю :

**Математична модель:**

Для обчислення виразу при заданому x та n використаємо арифметичний цикл. Значенням х є число що вводиться користувачем за запитом на початку програми. В ході аналізу заданої формули приходимо до висновку що для знаходження суми з заданою точністю необхідно визначити тип заданої послідовності і якщо вона спадна, виконати цикл що буде проводити ітерації до моменту коли наступний доданок стане меншим за вказану точність. Виконання таких дій можливе лише якщо послідовність є спадною або має точку максимуму. Вказана послідовність має один екстремум і точку максимуму що залежить від заданих користувачем значень. Для обчислення певних k-тих доданків суми нам необхідно знайти значення для кожного з цих доданків. Апаратні можливості комп’ютера накладають певні обмеження на дане значення , а саме: при знаходжені факторіала числа більшого ніж 20 відбувається переповнення змінної. Зважаючи на це ми обмежимо кількість ітерацій циклу( таким чином щоб умова виконувалась при будь якому значенні х. Значенням n є число що вводиться користувачем за запитом на початку програми. Для правильного виконання програми робимо перевірку числа на приналежність вказаному в умові проміжку , і у випадку якщо число не задовольняє умову просимо користувача ввести нове значення. Для обчислення факторіалів і скористаємося внутрішніми умовними операторами. Ззовні виконання основного циклу програми ініціалізуємо змінні що міститимуть значення факторіалів. У кожній ітерації будемо множити задані змінні на поточні значення та . Варто зазначити що інкремент виконується в кожній ітерації. Також на початку кожної ітерації повідомляємо користувача про поточне значення суми. Для обчислення значення кожного доданку використовуємо фунціонал сторонніх бібліотек. Для використання математичних функцій степеня використаємо бібліотеки <cmath> для С++ . Для обчислення суми використовуємо логічні оператори. У разі якщо доданок більше ніж задана точність- додаємо значення до суми, інакше перериваємо головний цикл.

Якщо за задану кількість ітерацій не вдалося встановити точну суму повідомляємо користувача про помилку і просимо ввести інші значення.

**1)**Приклад коду на С++:

# include <iostream>

#include <cmath>

#include <iomanip>

#include <algorithm>

using namespace std;

//максимально велике числот що поміщається в змінну лонг лонг і при цьому не перевантажує її це факторіал 20,

//отже коли користувач вводить двні необхідно прослідкувати щоб сума к та n не перевищувала 20,

//тож встановимо обмеження на х, враховуючи конкретні значення n

int main( )

{

long double x, n;

cout << "Enter x value: ";

cin >> x;

do

{

cout << "\nEnter n value between 0 and 5: ";

cin >> n;

}

while (n >= 5 || n < 0);

int k = 0;

long long counterFact = 1;

long long sumCounterAndInputFact = 1;

bool finish = false, success=false;

long double sum = 0, add = 0;

while (!finish)

{

std::cout << "Iteration " << k<< " worked here. The current value of sum is: "<<sum << std::endl;

if (k != 0)

{

counterFact \*= k;

}

if ((n + k) != 0)

{

if (k == 0)

{

long long copy=n;

while (copy > 0)

{

sumCounterAndInputFact \*= (copy);

copy--;

}

}

else

{

sumCounterAndInputFact \*= (n + k);

}

}

add = pow(-1, k) \* pow((x / 2), n + (2 \* k)) / (counterFact \* sumCounterAndInputFact);

if (fabs(add) >= 1e-4 && n + k+1 <= 20)

{

sum += add;

}

else if (n + k + 1 > 20)

{

finish = true;

}

else

{

finish = true;

success = true;

}

++k;

}

if (success == true)

{

cout << "\nThe sum is: " << sum<< endl;

}

else

{

cout << "\n The last iteration was: " << k-1

<<".\n Value overflow occured. Please try to enter lesser value of x with the same n or try other values of x and n" << endl;

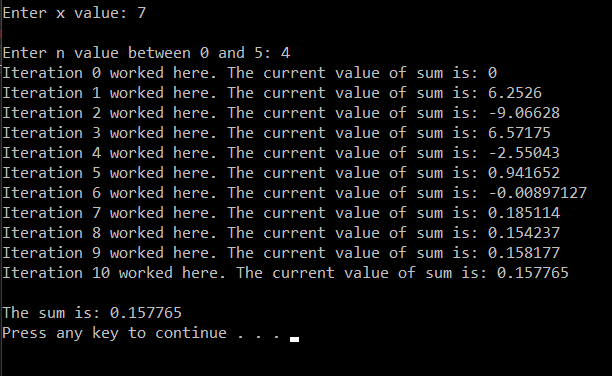
}

return 0;

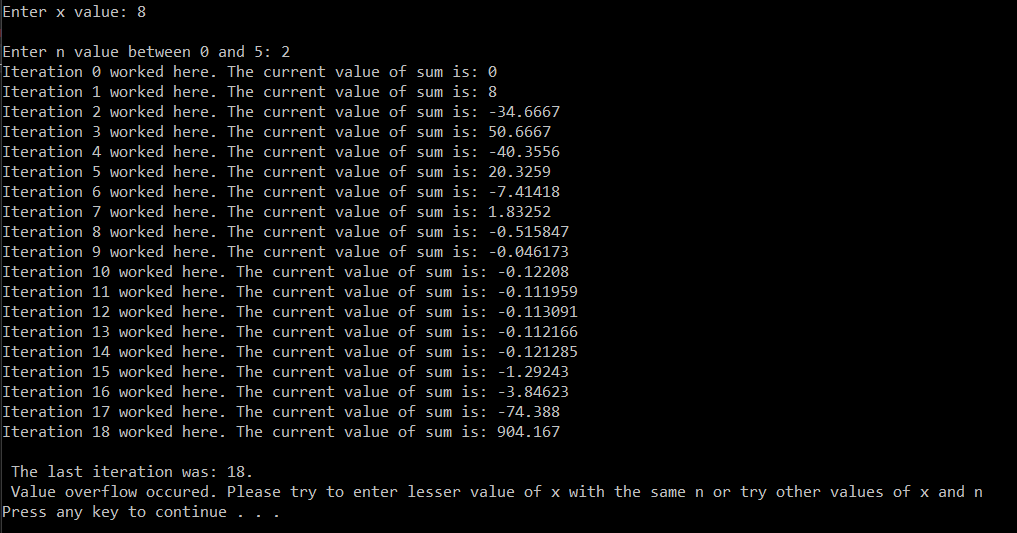
}

Виконання коду на С++:

-У випадку підходящих значень вводу х та n:



-У випадку незадовільних значень x та n:



**Висновок:**

Отже за допомогою даного алгоритму ми успішно обрахували і відобразили на екрані значення заданої суми при коректних та некоректних ввідних даних від користувача, за допомогою використання ітераційних циклів, де параметр n та х вводяться користувачем, на компільованій мові С++, а також ознайомились з нюансами організації комп’ютерної пам’яті та проблемах що виникають при переповненні змінних. Ми також створили механізми валідації користувацького вводу та аналізували задане рівняння суми для знаходження оптимальних шляхів виконання завдання.