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 № 2 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Організація циклічних процесів. Ітераційні цикли»

Варіант\_**20**\_\_

Виконав студент: **ІП-11 Лошак Віктор Іванович**

(шифр, прізвище, ім'я, по батькові)

Перевірив: **Мартинова О.П.**

Київ 2021

**Лабораторна робота 3**

**Організація циклічних процесів. Ітераційні цикли**

**Мета:**

Вивчити особливості організації ітераційних циклів.

**Варіант №20**

**Задача:** *Для заданого х і , знайти з точністю :*

**Розв’язок:**

1)Постановка задачі

Результатом розв’язку є число що є сумою заданих доданків. Значення суми на пряму залежить від даних отриманих з користувацького вводу(параметра n , x). 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.

2)Побудова математичної моделі

Складемо таблицю імен змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Користувацький ввід | дійсний | х | Початкове дане |
| Користувацький ввід | дійсний | n | Початкове дане |
| Лічильник ітерацій циклу | дійсний | k | Проміжний результат |
| Значення факторіалу k | дійсний | counterFact | Проміжний результат |
| Значення факторіалу (k+n) | дійсний | sumCounterAndInputFact | Проміжний результат |
| Умова зупинки основного циклу програми | логічний | finish | Проміжний результат |
| Змінна що позначає стан завершення циклу | логічний | success | Проміжний результат |
| Результуюча сума виразу | З плаваючою точкою | sum | Кінцевий результат |
| Доданок всередині суми | З плаваючою точкою | add | Проміжний результат |

3)Програмні специфікації напишемо у псевдокоді та графічній формі у вигляді блок-схеми.

*Крок 1:* Визначимо основні дії.

*Крок 2:* Деталізуємо дію вводу і оголошення основних змінних програми.

*Крок 3:* Деталізуємо головний цикл програми(обчислення факторіалів, вирахування доданку суми, перевірку завершення циклу.

Псевдокод

*Крок1*

**Початок**

-ввід x,n

-оголошення k,

counterFact ,

sumCounterAndInputFact,

finish ,

success,

sum, add

-основний цикл програми

**Кінець**

*Крок2*

**Початок**

-ввід x,n

- k: = 0;

counterFact := 1;

sumCounterAndInputFact: = 1;

finish := false;

success:=false;

sum: = 0;

add := 0;

-основний цикл програми

**Кінець**

*Крок3*

**Початок**

-ввід x,n

- k: = 0;

counterFact := 1;

sumCounterAndInputFact: = 1;

finish := false;

success:=false;

sum: = 0;

add := 0;

- while (!finish)

Вивід значення поточної ітераці

**if** (k != 0)

counterFact \*= k

**if** ((n + k) != 0)

**if** (k == 0)

copy=n

**while** (copy > 0)

sumCounterAndInputFact \*= (copy)

copу—

**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)

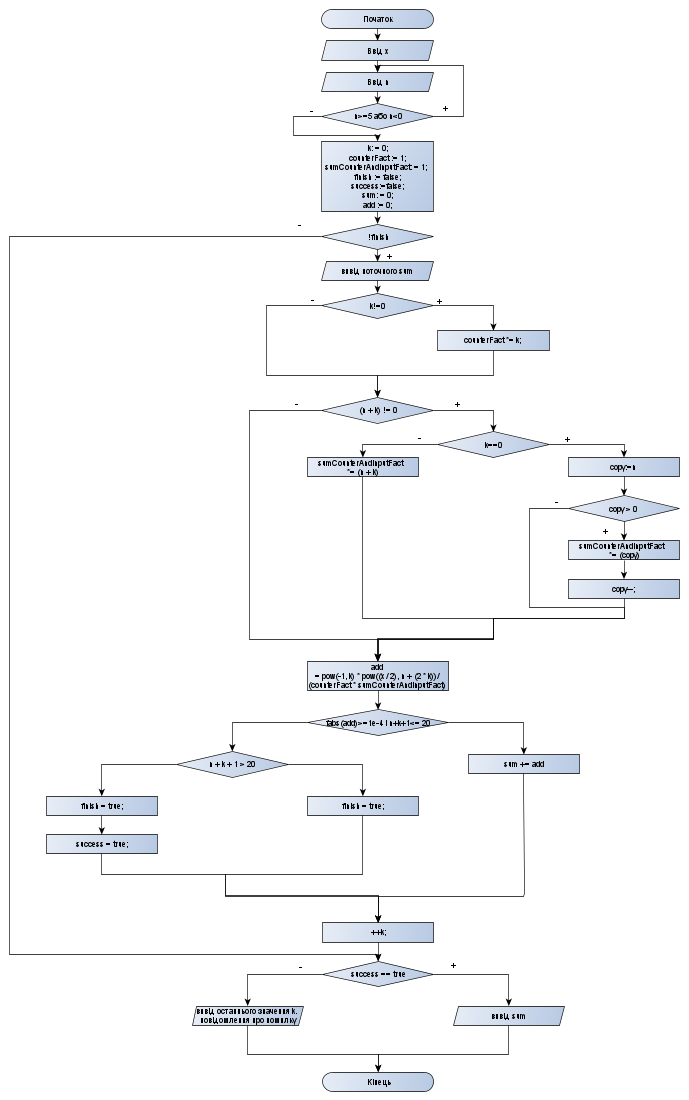
вивід суми

else

остання ітерація, повідомлення про помилку }

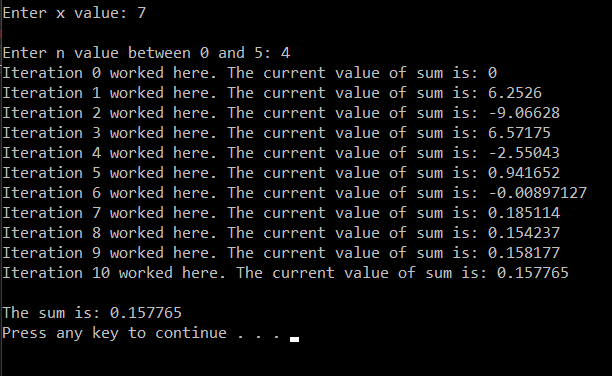
**Кінець**

Блок-схема:

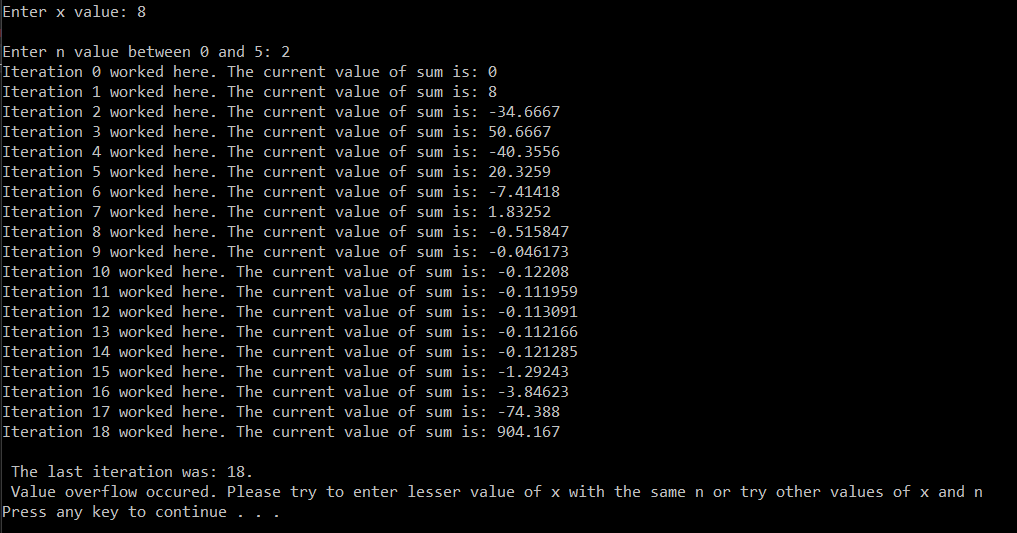


Виконання коду на С++:

-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.