**СТРУКТУРА**

**звіту з лабораторної роботи**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**імені ІГОРЯ СІКОРСЬКОГО”**

**ЗВІТ**

**з лабораторної роботи №2**

**з навчальної дисципліни “Програмування складних алгоритмів”**

**Тема: Рекурсивні алгоритми**

**Варіант №1**

**Виконав студент групи ТР–15**

Руденко Владислав Ігорович

з оцінкою \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Перевірив доцент кафедри**

Андрій ОНИСЬКО\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ року

**Київ 2022**

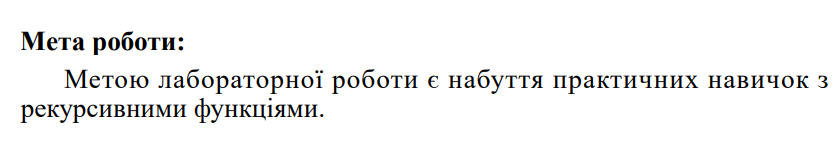
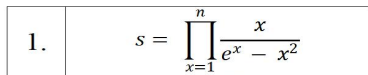
**Теоретична частина.**

*Рекурсія* - ситуація, коли деякий об’єкт містить у своєму описі самого себе, тобто є власною частиною. Якщо коротко, рекурсія дозволяє відійти від найпростіших імперативно написаних ітерацій до більш декларативного стилю написання коду.

У програмуванні рекурсія - виклик функції (процедури) з неї самої, безпосередньо (проста рекурсія) або через інші функції (складна рекурсія), наприклад, функція A викликає функцію B, а функція B - функцію A. Кількість вкладених викликів функції або процедури називається глибиною рекурсії.

Могутність рекурсивного визначення об'єкта в тому, що таке кінцеве визначення здатне описувати нескінченно велику кількість об'єктів. За допомогою рекурсивної програми можна описати нескінченне обчислення, причому без явних повторень частин програми.  
  
Порівнюючи швидкість обчислення чисел Фібоначчі за допомогою ітеративної та рекурсивної функції можна помітити, що ітеративна функція виконується майже миттєво, незалежно від значення n. При використанні ж рекурсивної функції вже за n=40 помітна затримка при обчисленні, а за великих n результат утворюється дуже скоро.

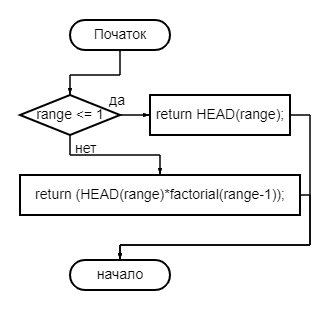
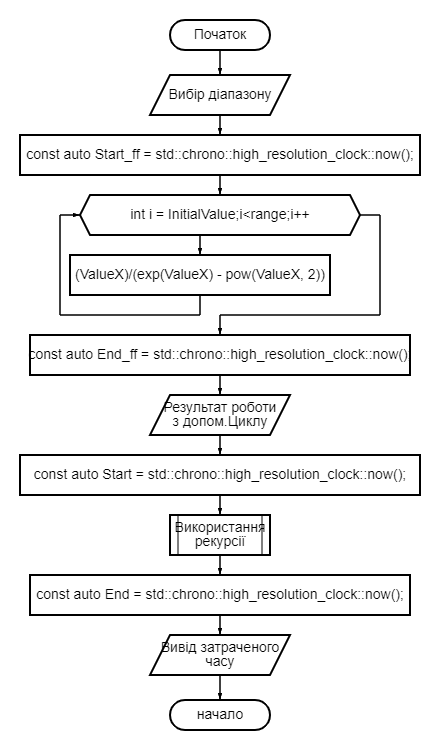
**І. Завдання + Мета:**

*Варіант роботи*

**ІІ. Результати виконання лабораторної роботи.**

**Блок Схеми:**

****

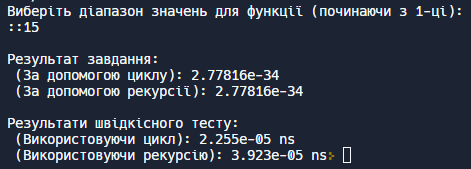
Результати роботи - 1 (основна схема) Результати роботи - 1 (схема рекурсії)

**Опис програми (написана на мові Сі++):**Програма виконує розв’язок підібраної варіантом функції двома способами. Перший використовує цикли для отриманя результату, інший же спосіб використовує рекурсивний алгоритм, що посилає сам на себе, поки не буде отримано необіхдного результату. Також програма вираховує час виконання кожним з варіантів.

**Результати роботи:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поріврнянн часу виконання двох варіантів розв’язку | | | | |
| Діапазон значень | Результат обчислень (З викр. Циклів) | Результат обчислень (З викр. Рекурсії) | Час виконання *(*З викр. Циклів*) ns* | Час виконання *(*З викр. Рекурсії*) ns* |
| **1x3** | **0.343445** | **0.343445** | **1.875e-05** | **1.144e-05** |
| **1x5** | **0.00963196** | **0.00963196** | **2.0211e-05** | **1.324e-05** |
| **1x10** | **1.3101e-13** | **1.3101e-13** | **3.946e-0** | **1.637e-05** |
| **1x20** | **4.7148e-65** | **4.7148e-65** | **2.095e-05** | **1.291e-05** |

**Результати роботи у вигляді скріншотів:**



Результати роботи - 2

**Посилання на repl.it:**

<https://replit.com/join/ajgfsibwly-hetik>

**Ш. Висновки.**

Під час виконання лабораторної роботи №2 було здобуто навички роботи з рекурсивними функціями, за для їх вирішення було використано 2 методи опрацювання формул. Під час виконання було напсано програму яка вираховує результати формули двома способами. Один з них це використання рекурсивного алгоритму, він сприяв збільшеню шкидкості обчислення, але при використанні малих діапзонів обчислення. При збільшені діапазону рекурсивний метод значимо сповільнює обчислення. Також було виявлено та виправлено декілька помилок.

**Програмний код:**

*#include <iostream>*

*#include <math.h>*

*#include <chrono>*

*#define InitialValue 1*

*double HEAD(int ValueX)*

*{*

*return (ValueX)/(exp(ValueX) - pow(ValueX, 2));*

*}*

*double factorial(int range){*

*if(range <= 1)*

*return HEAD(range);*

*else*

*return (HEAD(range)\*factorial(range-1));*

*}*

*int main(void) {*

*int range;*

*double result\_ff=1;*

*std::cout << "Лабораторна робота №1. Рекурсивні алгоритми by Руденко Владислав.\n\nВиберіть діапазон значень для функції (починаючи з 1-ці):\n::";*

*scanf("%i", &range);*

*const auto Start\_ff = std::chrono::high\_resolution\_clock::now();*

*for(int i = InitialValue;i<range;i++)*

*result\_ff \*= HEAD(i);*

*const auto End\_ff = std::chrono::high\_resolution\_clock::now();*

*std::cout << "\nРезультат завдання: \n (За допомогою циклу): " << result\_ff;*

*const auto Start = std::chrono::high\_resolution\_clock::now();*

*std::cout << "\n (За допомогою рекурсії): " << factorial(range-1);*

*const auto End = std::chrono::high\_resolution\_clock::now();*

*std::cout << "\n\nРезультати швідкісного тесту:\n (Використовуючи цикл): " << std::chrono::duration<float>(End\_ff - Start\_ff).count() << " ns";*

*std::cout << "\n (Використовуючи рекурсію): " << std::chrono::duration<float>(End - Start).count() << " ns";*

*}*