ЗВІТ

про виконання лабораторної роботи No < 5.2 >

«Обчислення суми ряду Тейлора за допомогою функцій»

з дисципліни

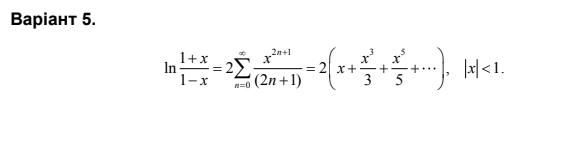
«Алгоритмізація та програмування»

Студента групи КН-106Б

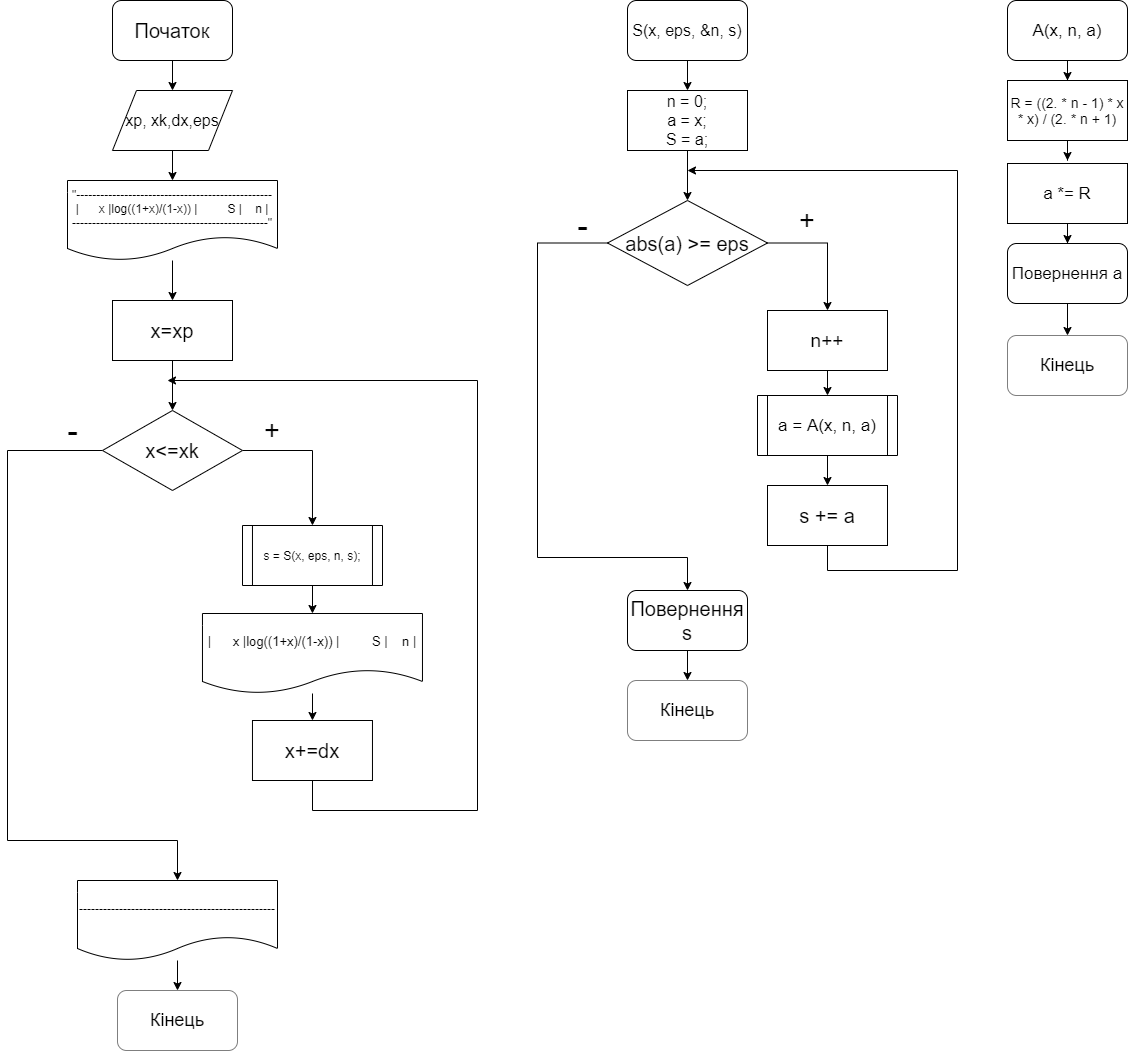
Онишківа Остапа Володимировича

**Мета:** Навчитися використовувати функції, формати виводу, рекурентні співвідношення.

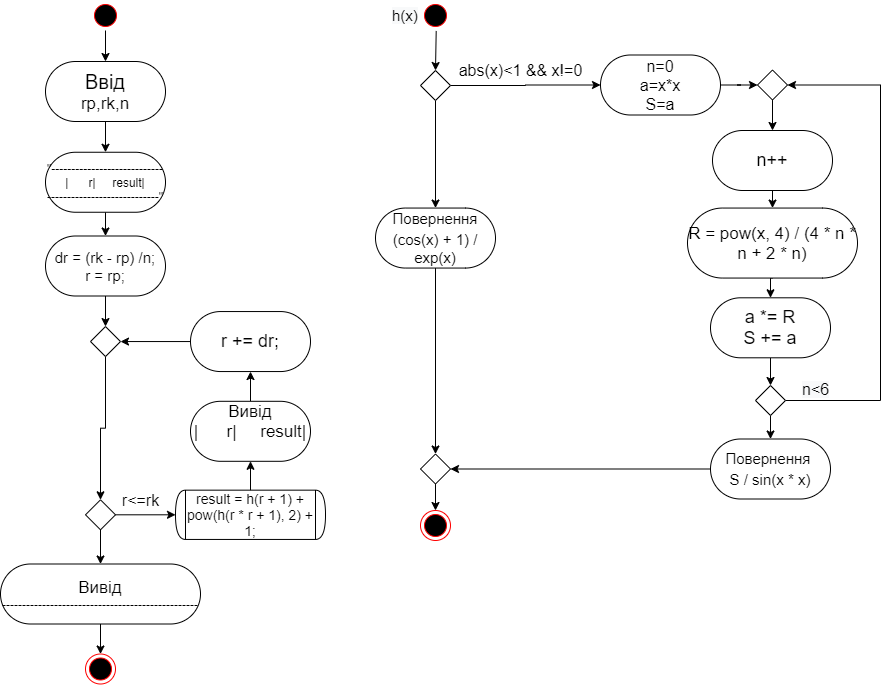
**Умова завдання:** Обчислити і вивести на екран у вигляді таблиці значення функції, заданої за допомогою ряду Тейлора, на інтервалі від хпоч до хкін з кроком dx та точністю eps. Параметри xпоч, xкін, dx, eps вводяться з клавіатури. Таблиця має містити заголовок та шапку. Кожний рядок таблиці має містити значення аргументу, значення функції, значення суми ряду та кількість порахованих доданків. Суму ряду Тейлора та значення доданку обчислювати за допомогою допоміжних алгоритмів, реалізованих за допомогою окремих функцій. При обчисленні значення доданків використовувати рекурентні співвідношення. Всю необхідну функціям інформацію слід передавати лише за допомогою параметрів.

****

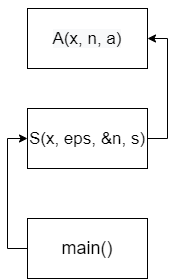
**Блок-схема алгоритму:**

****

**UML-діаграма алгоритму:**

****

**Структурна схема:**

****

**Текст програми:**

// Lab\_5.2.cpp

// < Онишківа Остапа >

// Лабораторна робота No 5.2

// Обчислення суми ряду Тейлора за допомогою функцій.

// Варіант 5

#include<iostream>

#include<cmath>

#include<iomanip>

using namespace std;

double A(const double x, const int n, double a);

double S(const double x, const double eps, int& n, double s);

int main()

{

double x, xp, xk, dx, s=0, eps;

int n=0;

cout << "xp = "; cin >> xp;

cout << "xk = "; cin >> xk;

cout << "dx = "; cin >> dx;

cout << "eps = "; cin >> eps;

cout << fixed;

cout << "-------------------------------------------------" << endl;

cout << "|" << setw(7) << "x" << " |"

<< setw(16) << "log((1+x)/(1-x))" << " |"

<< setw(10) << "s" << " |"

<< setw(5) << "n" << " |"

<< endl;

cout << "-------------------------------------------------" << endl;

x = xp;

while (x <= xk)

{

s = S(x, eps, n, s);

cout << "|" << setw(7) << setprecision(2) << x << " |"

<< setw(16) << setprecision(5) << log((1 + x) / (1 - x)) << " |"

<< setw(10) << setprecision(5) << 2. \* s << " |"

<< setw(5) << n << " |"

<< endl;

x += dx;

}

cout << "-------------------------------------------------" << endl;

return 0;

}

double S(const double x, const double eps, int& n, double s)

{

n = 0;

double a = x;

s = a;

while (abs(a) >= eps)

{

n++;

a = A(x, n, a);

s += a;

}

return s;

}

double A(const double x, const int n, double a)

{

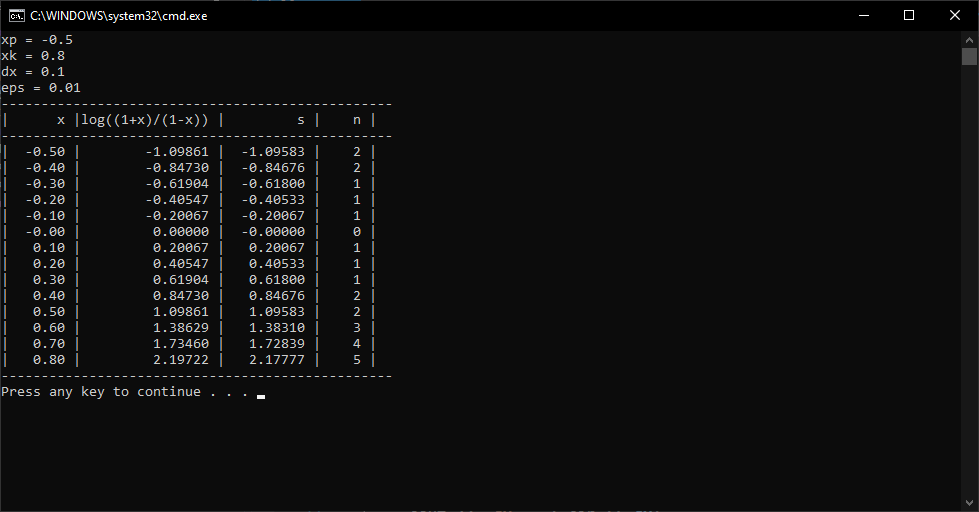
double R = ((2. \* n - 1) \* x \* x) / (2. \* n + 1);

a \*= R;

return a;

}

**Результат виконання:**



**Посилання на git-репозиторій з проектом:**

**Текст програми unit-тесту:**

#include "pch.h"

#include "CppUnitTest.h"

#include "../Lab\_5.2/Lab\_5.2.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest1

{

TEST\_CLASS(UnitTest1)

{

public:

TEST\_METHOD(TestMethod1)

{

double t = A(0.5, 2, 1);

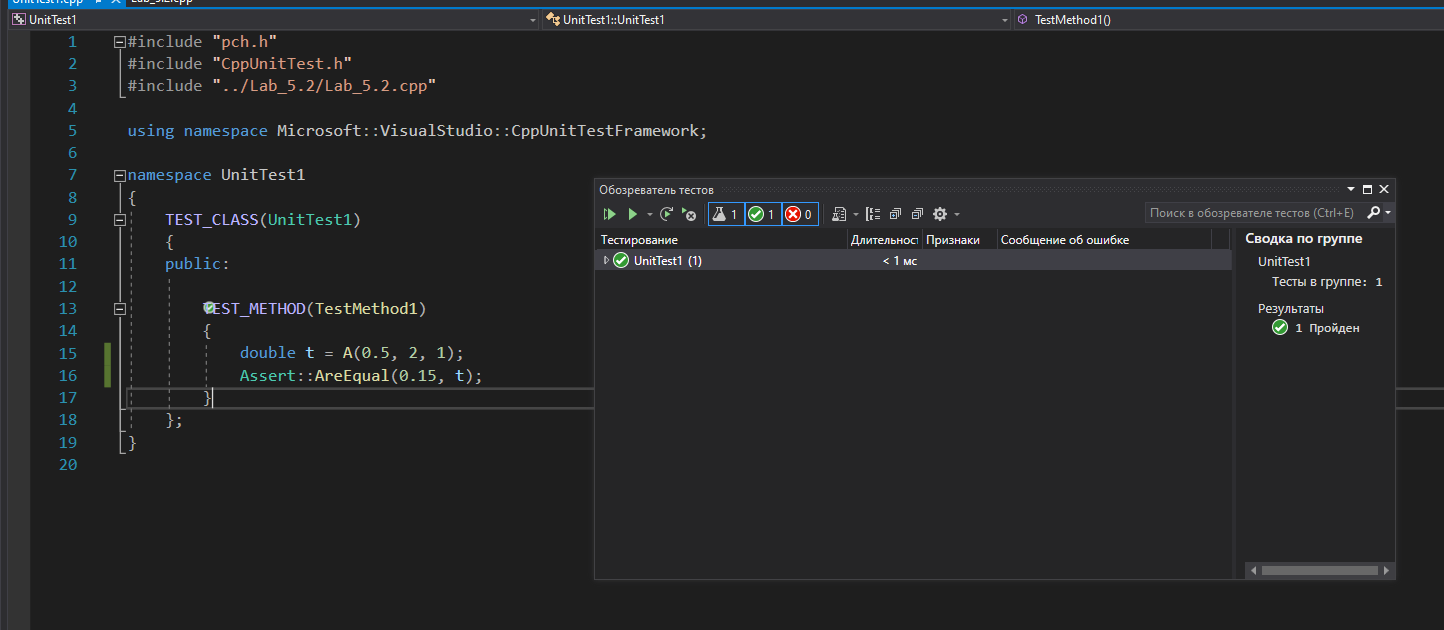
Assert::AreEqual(0.15, t);

}

};

}

**Результат unit-тесту:**

****

**Висновки:** на цій лабораторній роботі я навчився користуватися функціями, форматами виводу, рекурентними співвідношеннями, малювати структурні схеми програми та робити unit-тести.