ЗВІТ

про виконання лабораторної роботи № 5.02

« Обчислення суми ряду Тейлора за допомогою функцій » з дисципліни

«Алгоритмізація та програмування» студента групи РІ-12

Мілянця Мар`яна

# Мета роботи:

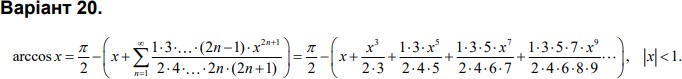
Навчитися використовувати функції, формати виводу, рекурентні співвідношення.

# 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 . 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.



# Текст програми:

#include <iostream> #include <iomanip> #include <cmath> using namespace std;

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

int main()

{

double xp, xk, x, dx, eps, s = 0; 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(5) << "x" << " |"

<< setw(10) << "acos(x)" << " |"

<< setw(7) << "S" << " |"

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

<< endl;

cout << " " << endl; x = xp;

while (x <= xk)

{

S(x, eps, n, s);

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

<< setw(10) << setprecision(5) << acos(x) << " |"

<< setw(10) << setprecision(5) << s << " |"

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

<< endl; x += dx;

}

cout << " " << endl; return 0;

}

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

{

n = 1;

double a = (x \* x \* x) / 6.; double Pi;

Pi = 4 \* atan(1.0); s = Pi / 2. - x - a; do {

n++;

A(x, n, a); s -= a;

} while (abs(a) >= eps);

}

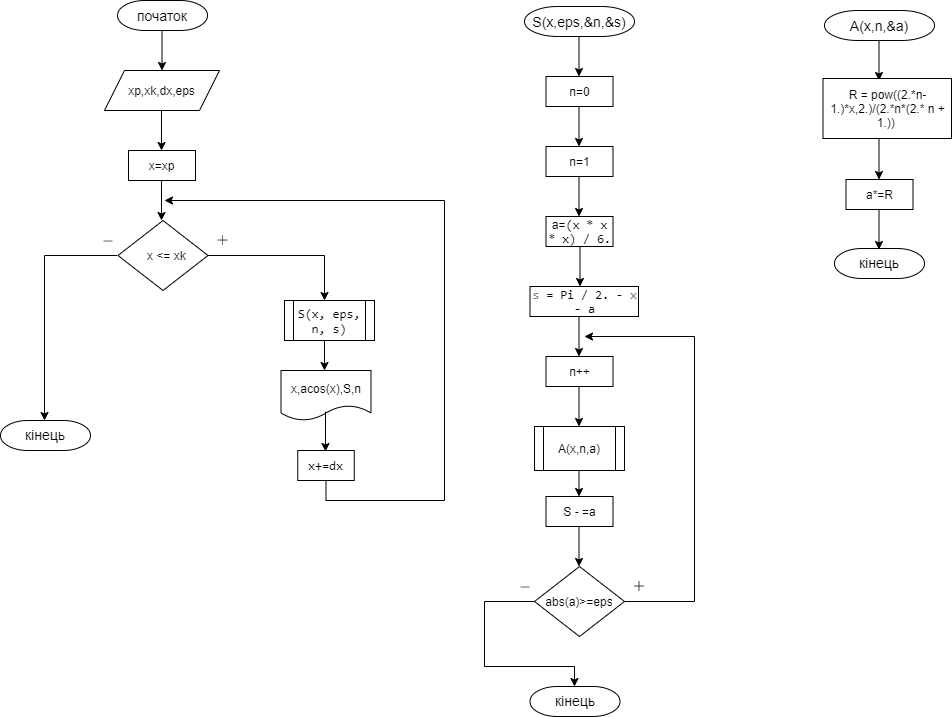
void A(const double x, const int n, double& a)

{

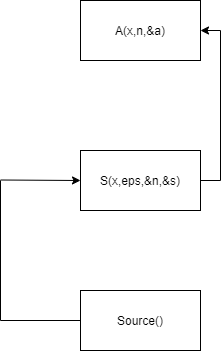
double R = pow((2. \* n - 1.) \* x, 2.) / (2. \* n \* (2. \* n + 1.)); a \*= R;

}

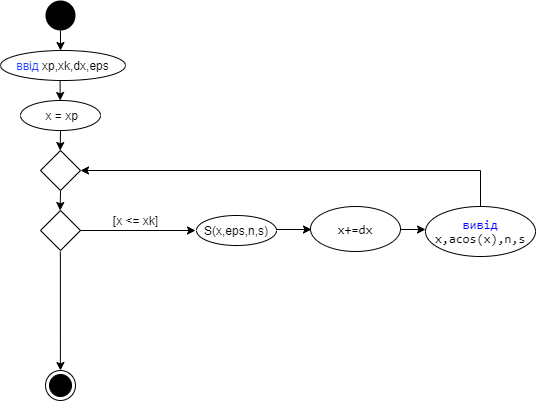
# Блок-схеми алгоритму програми та кожної функції:

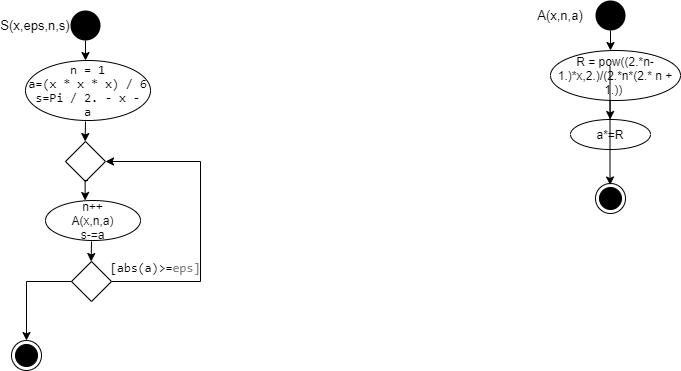


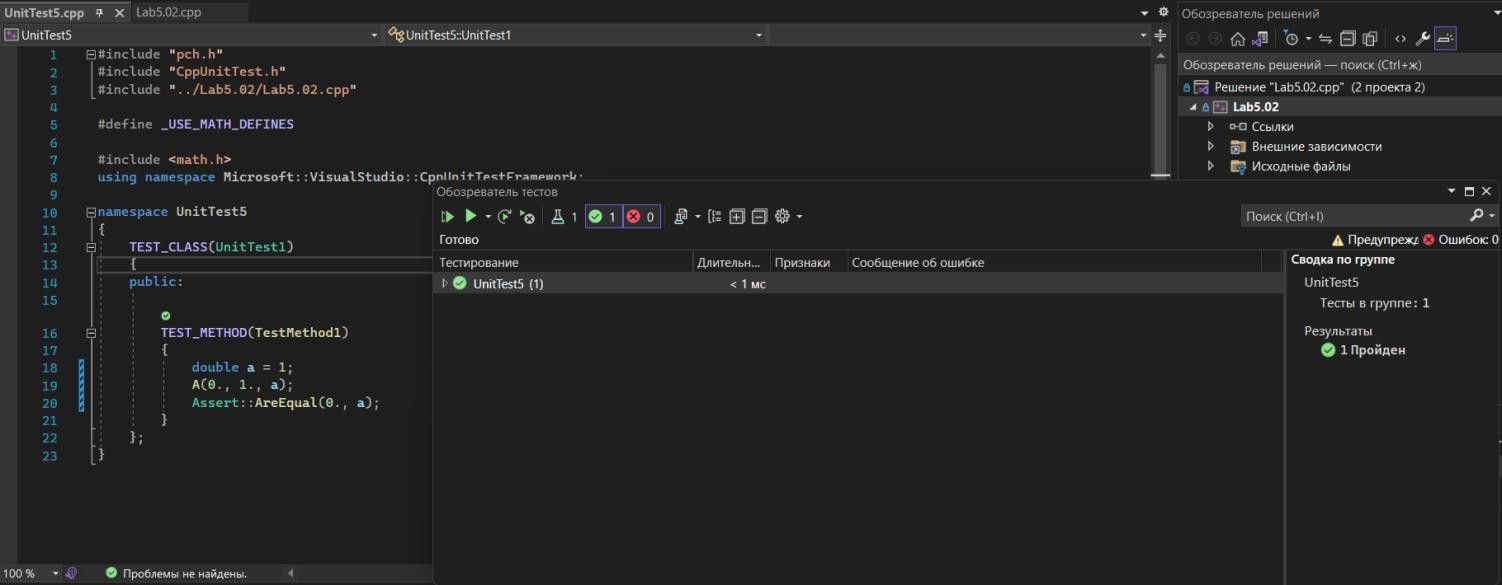
**Структурна схема програми:**



# UML-діаграма дії програми та кожної функції:





**Посилання**: [https://github.com/milianets20/Lab5.02](https://github.com/fxxwol/lab-5.git) **Результати unit-тесту:**

# Текст unit-тесту:

#include "pch.h" #include "CppUnitTest.h"

#include "../Lab5.02/Lab5.02.cpp"

#define \_USE\_MATH\_DEFINES

#include <math.h>

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest5

{

TEST\_CLASS(UnitTest1)

{

public:

TEST\_METHOD(TestMethod1)

{

double a = 1; A(0., 1., a);

Assert::AreEqual(0., a);

}

};

}

**Висновок**: виконуючи лабораторну роботу я навчився використовувати функції, формати виводу, рекурентні співвідношення.