ЗВІТ

про виконання лабораторної роботи № 5.04

«Обчислення сум та добутків за допомогою рекурсії» з дисципліни

«Алгоритмізація та програмування» студента групи РІ-12

Мілянця Мар`яна

# Мета роботи:

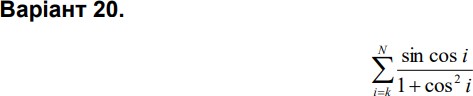
Навчитися використовувати рекурсивні функції.

# Умова завдання:

За допомогою рекурсивних функцій обчислити значення суми чи добутку. В одній програмі слід вивести результати звертання до п’яти функцій, які реалізують такі способи: 1) обчислення здійснюються на рекурсивному

спуску, значення параметра в рекурсивних викликах – спадає; 2) обчислення здійснюються на рекурсивному спуску, значення параметра в рекурсивних викликах – зростає; 3) обчислення здійснюються на рекурсивному підйомі, значення параметра в рекурсивних викликах –

спадає; 4) обчислення здійснюються на рекурсивному підйомі, значення параметра в рекурсивних викликах – зростає; 164 5) обчислення здійснюється ітераційним способом (для контролю правильності рекурсивних способів). Всі п’ять результатів мають збігатися. Всю необхідну функціям інформацію слід передавати лише за допомогою параметрів.



# Текст програми:

#include <iostream> #include <cmath>

using namespace std;

double S0(const int K, const int N)

{

double s = 0;

for (int i = K; i <= N; i++)

s += (sin(cos(i))) / (1 + pow(cos(i), 2)); return s;

}

double S1(const int K, const int N, const int i)

{

if (i > N)

return 0;

else

}

return (sin(cos(i))) / (1 + pow(cos(i), 2)) + S1(K, N, i + 1);

double S2(const int K, const int N, const int i)

{

if (i < K)

return 0;

else

}

return (sin(cos(i))) / (1 + pow(cos(i), 2)) + S2(K, N, i - 1);

double S3(const int K, const int N, const int i, double t)

{

t = t + (sin(cos(i))) / (1 + pow(cos(i), 2)); if (i >= N)

return t;

else

}

return S3(K, N, i + 1, t);

double S4(const int K, const int N, const int i, double t)

{

t = t + (sin(cos(i))) / (1 + pow(cos(i), 2)); if (i <= K)

return t;

else

}

return S4(K, N, i - 1, t);

int main()

{

int K, N;

cout << "K= "; cin >> K; cout << "N= "; cin >> N;

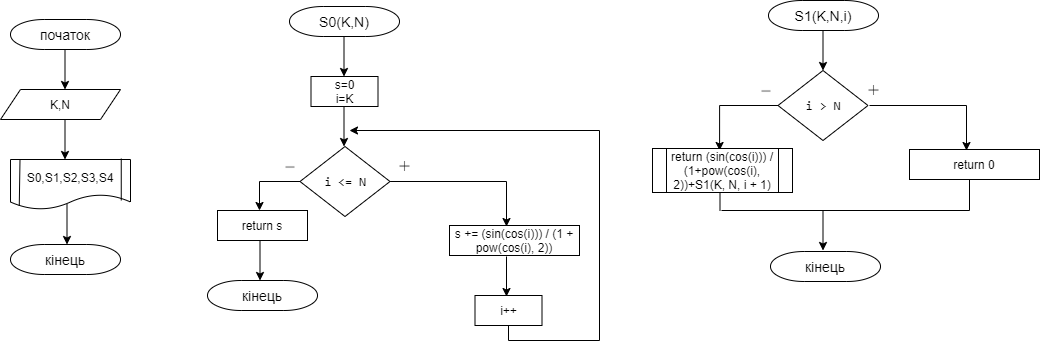
cout << "(iter) S0 = " << S0(K, N) << endl; cout << "(rec up ++) S1 = " << S1(K, N, K) << endl; cout << "(rec up --) S2 = " << S2(K, N, N) << endl;

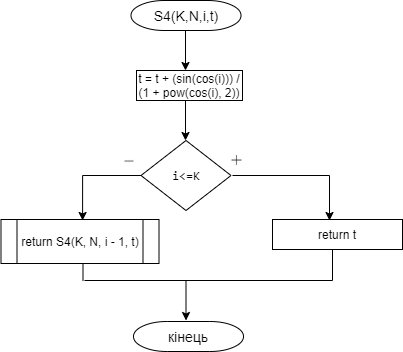
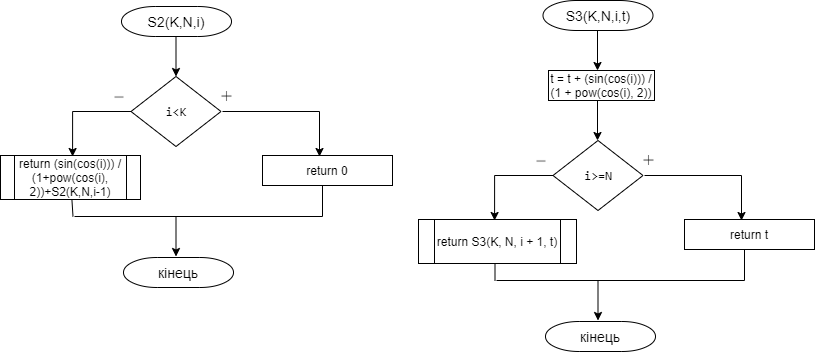
cout << "(rec down ++) S3 = " << S3(K, N, K, 0) << endl; cout << "(rec down --) S4 = " << S4(K, N, N, 0) << endl;

return 0;

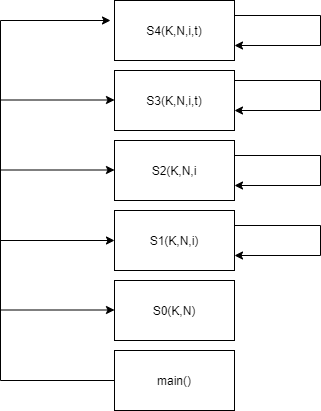
}

# Блок-схеми алгоритму програми та кожної функції:

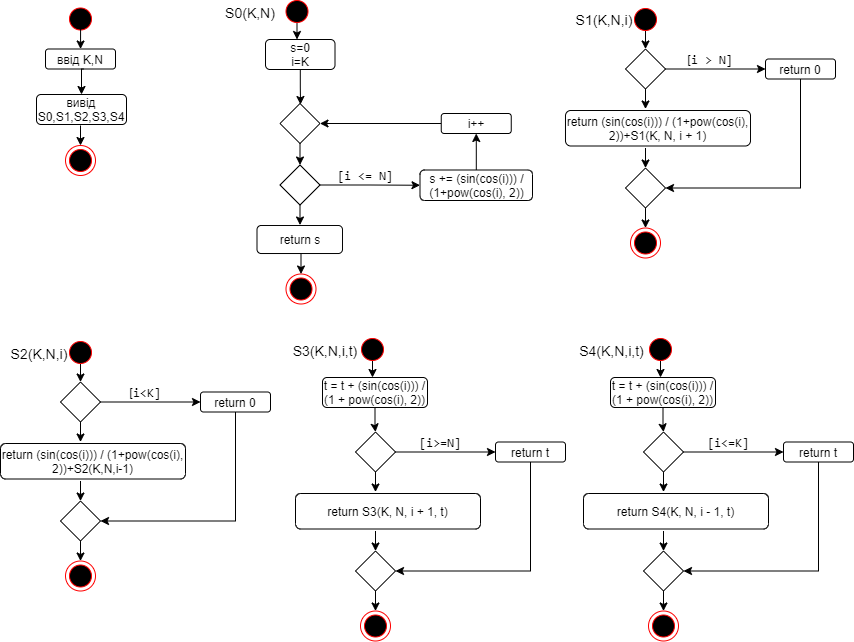




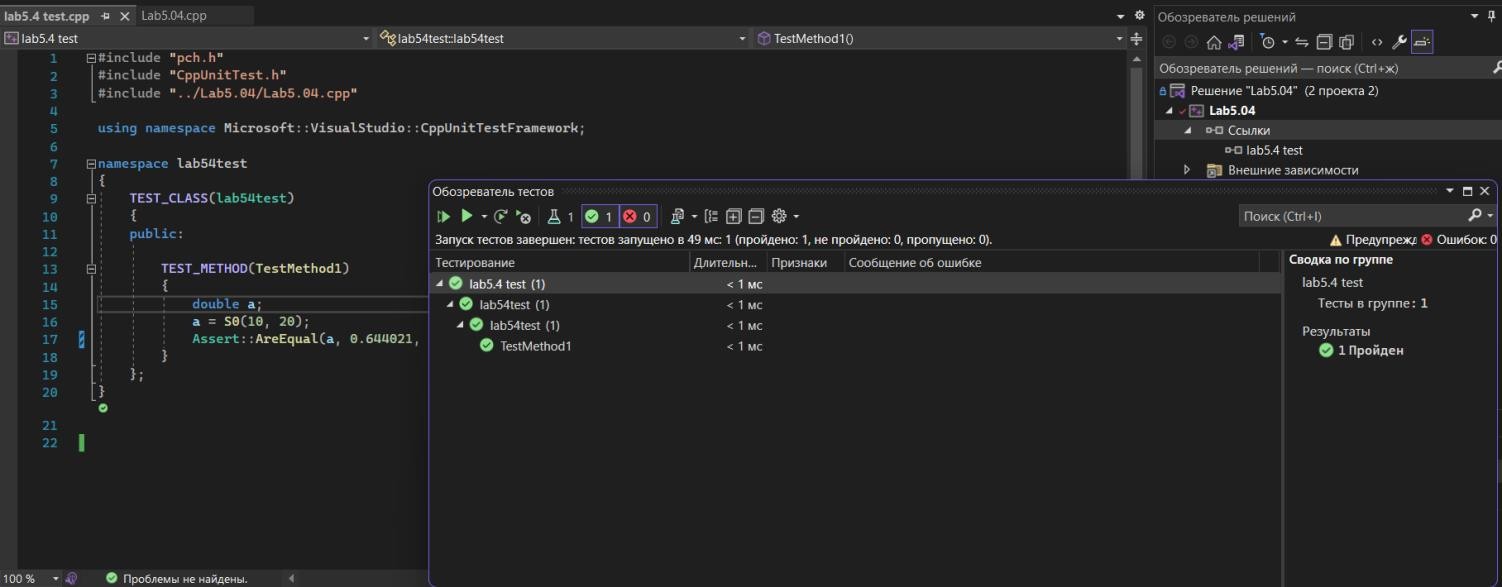
**Структурна схема програми:**



# UML-діаграма дії програми та кожної функції:



**Результати unit-тесту:**



# Текст unit-тесту:

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

#include "../Lab5.04/Lab5.04.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace lab54test

{

TEST\_CLASS(lab54test)

{

public:

TEST\_METHOD(TestMethod1)

{

double a;

a = S0(10, 20);

Assert::AreEqual(a, 0.644021, 0.0001);

}

};

}

**Посилання**: <https://github.com/milianets20/Lab5.4>

**Висновок**: : виконуючи лабораторну роботу я навчився використовувати рекурсивні функції.