**ЗВІТ**

**про виконання лабораторної роботи №5.4**

**“Організація функцій. Рекурсія. Автоматизоване тестування ”**

**з дисципліни**

**«Алгоритмізація та програмування»**

**студента ІК-11**

**Рехтмана Михайла Володимировича**

м.Львів 2020

**Питання, які необхідно вивчити та описати в звіті:**

1) Загальний синтаксис опису функцій та процедур.

2) Виклики функцій та процедур

3) Передавання інформації у функцію за допомогою значень та за допомогою адрес.

4) Параметри-значення та параметри-змінні.

5) Рекурсивні підпрограми, їх переваги та недоліки.

6) Основні елементи рекурсивних підпрограм.

7) Дії на рекурсивному спуску та дії на рекурсивному підйомі. 8) Обчислення суми та добутку за допомогою рекурсій.

**Умова завдання:**

За допомогою рекурсивних функцій обчислити значення суми чи добутку.

В одній програмі слід вивести результати звертання до п’яти функцій, які реалізують наступні 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;

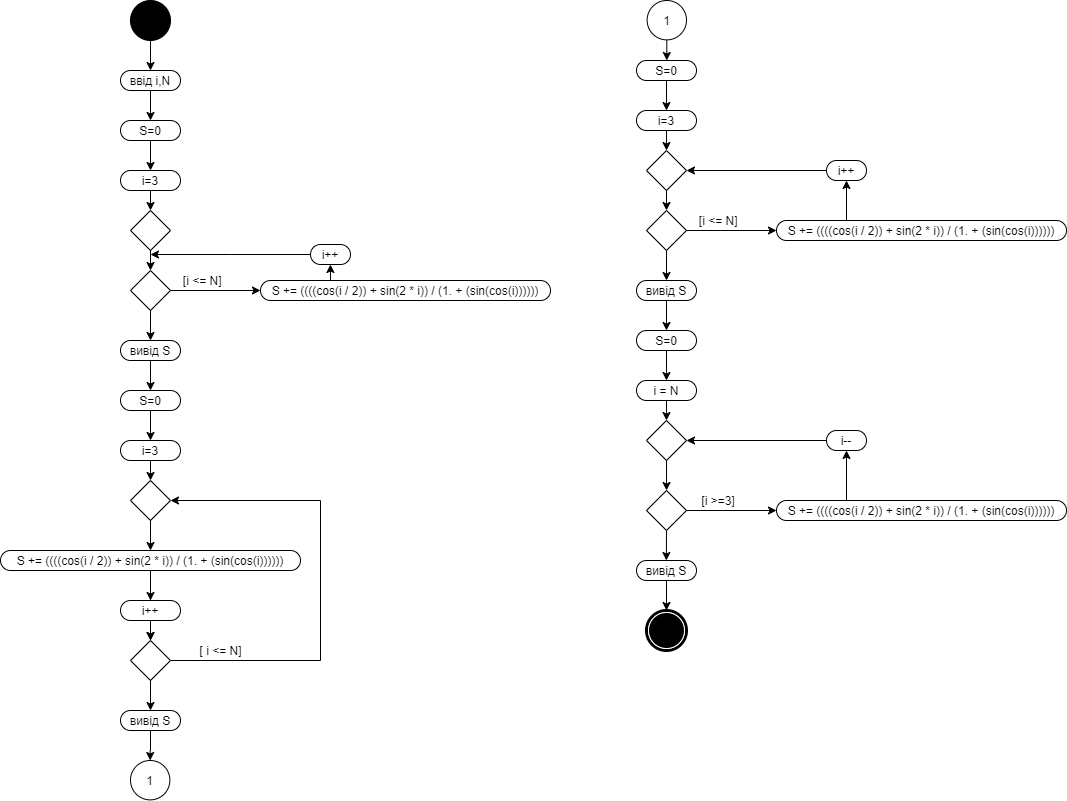
4) обчислення здійснюються на рекурсивному підйомі, значення параметра в рекурсивних викликах – зростає; 135

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.



**Блок-схема алгоритму: UML-діаграма дії:**

****

**Текст програми:**

#include <iostream>

#include <cmath>

using namespace std;

double S0(const int K, const int N)

{

double s = 0;

for (int i = 3; i <= N; i++)

s += (cos(i/2)+sin(2.\*i))/(1.+sin(cos(i)));

return s;

}

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

{

if (i > N)

return 0;

else

return (cos(i / 2) + sin(2. \* i)) / (1. + sin(cos(i))) + S1(K, N, i + 1);

}

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

{

if (i < K)

return 0;

else

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

}

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

{

t = t + (cos(i / 2) + sin(2. \* i)) / (1. + sin(cos(i)));

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 + (cos(i / 2) + sin(2. \* i)) / (1. + sin(cos(i)));

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;

}

**Посилання на git-репозиторій:**

<https://github.com/HockageDano/lab54.git>

**Висоновок:**

Навчився використовувати рекурсивні функції.