ЗВІТ

про виконання лабораторної роботи № 5.4

«Функції, що містять арифметичний вираз»

з дисципліни

«Алгоритмізація та програмування»

студентки групи ІК-11

Турковської Марти Михайлівни

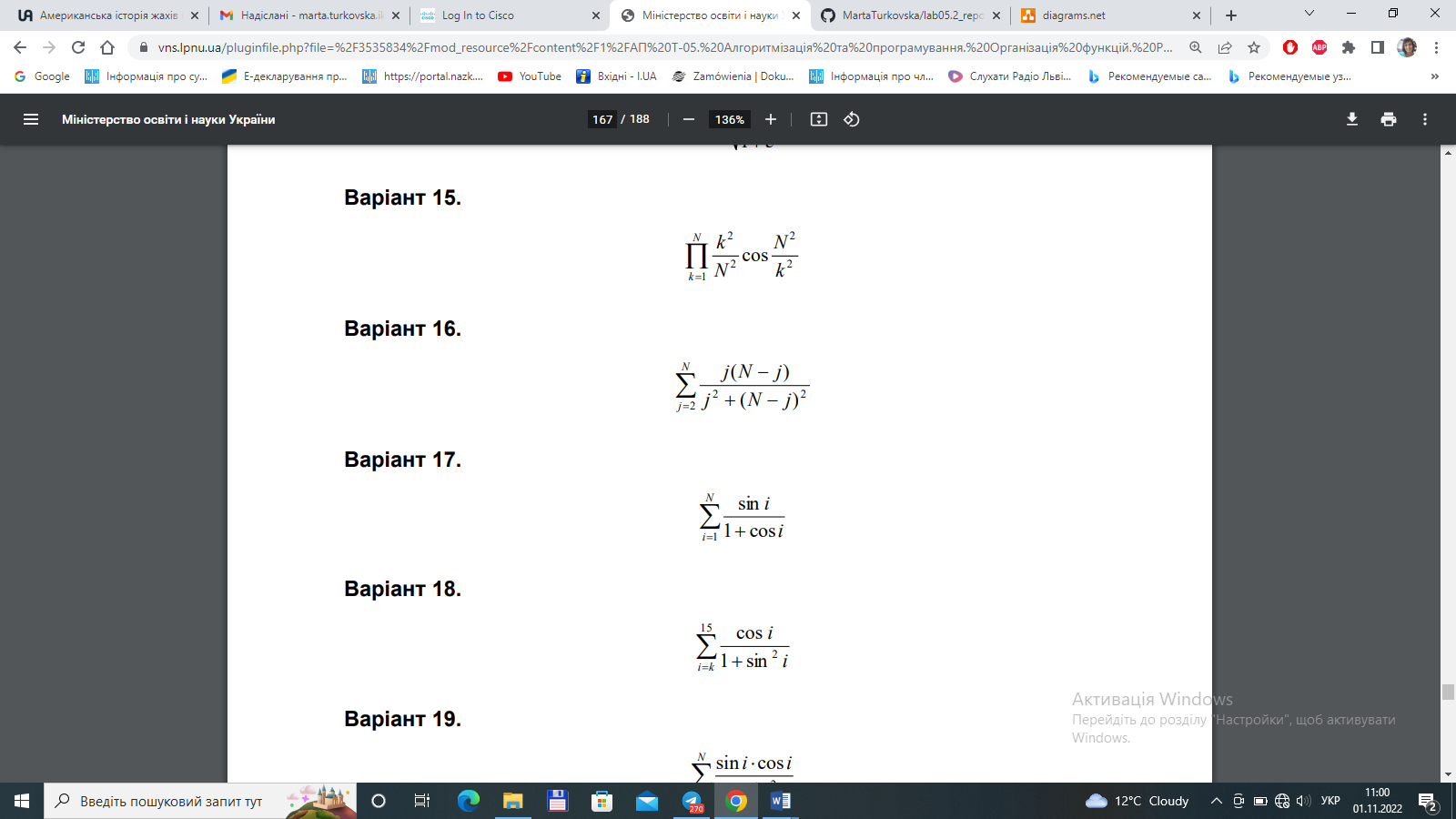
**Умова завдання:**

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;
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-схеми алгоритму програми та кожної функції:**

**UML-діаграма дії програми та кожної функції:**

**Структурна схема програми:**

**Текст програми:**

// Лабораторна робота № 5.4

#include <iostream>

#include <cmath>

using namespace std;

double S0(const int N)

{

double s = 0;

for (int i = 1; i <= N; i++)

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

return s;

}

double S1(const int N, const int i)

{

if (i > N)

return 0;

else

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

}

double S2(const int N, const int i)

{

if (i < 1)

return 0;

else

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

}

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

{

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

if (i >= N)

return t;

else

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

}

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

{

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

if (i <= 1)

return t;

else

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

}

int main()

{

int N;

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

cout << "(iter) S0 = " << S0(N) << endl;

cout << "(rec up ++) S1 = " << S1(N, 1) << endl;

cout << "(rec up --) S2 = " << S2(N, N) << endl;

cout << "(rec down ++) S3 = " << S3(N, 1, 0) << endl;

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

return 0;

}

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

**Результат unit-тесту:**

**Висновок:** навчилися використовувати рекурсивні функції.