**Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

**Лабораторна робота №1**

з дисципліни

«Алгоритми і структури даних»

Виконав: Перевірила:

Студент групи ІМ-42 Сергієнко А. М.

Лобань Михайло Юрійович

номер у списку групи: 20

Київ 2025

**Загальна постановка завдання**

Дане натуральне число n. Знайти суму перших n членiв ряду чисел, заданого рекурентною формулою. Розв’язати задачу трьома способами:

1) у програмi використати рекурсивну функцiю, яка виконує обчислення i членiв ряду, i суми на рекурсивному спуску;

2) у програмi використати рекурсивну функцiю, яка виконує обчислення i членiв ряду, i суми на рекурсивному поверненнi;

3) у програмi використати рекурсивну функцiю, яка виконує обчислення членiв ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненнi.

**Завдання за варіантом**

Зображення, що містить текст, Шрифт, білий, алгебра

Вміст, створений ШІ, може бути неправильним.

**Текст програми**

#include <stdio.h>

#include <stdlib.h>

typedef struct {

double sum;

double F;

} Result;

double sum1(int n, double x, int i, double F, double sum) {

if (x < 1 && x > -1) {

sum += F;

printf("F%d = %lf, sum = %lf\n", i, F, sum);

if (i == n) {

return sum;

}

return sum1(n, x, i + 1, F \* (x \* x \* (2 \* i - 1)) / (2 \* i + 1), sum);

}

return sum;

}

Result sum2(int n, double x, int i) {

Result result;

if (x < 1 && x > -1) {

if (i == 1) {

result.F = x;

result.sum = x;

printf("F%d = %lf, sum = %lf\n", i, result.F, result.sum);

return result;

}

Result prev = sum2(n, x, i - 1);

result.F = prev.F \* (x \* x \* (2 \* i - 3)) / (2 \* i - 1);

result.sum = prev.sum + result.F;

printf("F%d = %lf, sum = %lf\n", i, result.F, result.sum);

return result;

}

return result;

}

double sum3(int n, double x, int i, double F) {

if (x < 1 && x > -1) {

if (i == n) {

printf("F%d = %lf, sum = %lf\n", i, F, F);

return F;

}

double sum = sum3(n, x, i + 1, F \* (x \* x \* (2 \* i - 1)) / (2 \* i + 1));

sum += F;

printf("F%d = %lf, sum = %lf\n", i, F, sum);

return sum;

}

return 0;

}

double loops(int n, double x, int i, double F, double sum) {

if (x < 1 && x > -1) {

for(int i = 1; i <= n; i++) {

sum += F;

printf("F%d = %lf, sum = %lf\n", i, F, sum);

F \*= (x \* x \* (2 \* i - 1)) / (2 \* i + 1);

}

return sum;

}

return sum;

}

int main() {

int n = 5;

int i = 1;

double x = 0;

double F = x;

printf("=== First method ===\n");

double result1 = sum1(n, x, i, F, 0);

printf("Result: %lf\n\n", result1);

printf("=== Second method ===\n");

double result2 = sum2(n, x, n).sum;

printf("Result: %lf\n\n", result2);

printf("=== Third method ===\n");

double result3 = sum3(n, x, i, F);

printf("Result: %lf\n\n", result3);

printf("=== Loops ===\n");

double result\_loops = loops(n, x, i, F, 0);

printf("Result: %lf\n\n", result\_loops);

return 0;

}

**Результати тестування програми**

* **X = 0.5**

Зображення, що містить текст, знімок екрана, Шрифт, меню

Вміст, створений ШІ, може бути неправильним.

* **X=0.4**

Зображення, що містить текст, знімок екрана, Шрифт, чорно-білий

Вміст, створений ШІ, може бути неправильним.

* **X = 0.2**

Зображення, що містить текст, знімок екрана, Шрифт, чорно-білий

Вміст, створений ШІ, може бути неправильним.

* **Перевірка за допомогою калькулятора:**

Зображення, що містить текст, знімок екрана, Шрифт

Вміст, створений ШІ, може бути неправильним.

**Графiк залежностi похибки обчислення заданої функцiї вiд значення x при фiксованому значеннi n=5.**

Зображення, що містить текст, ряд, Графік, схема

Вміст, створений ШІ, може бути неправильним.

**Висновки**

**x**

Після виконання лабораторної роботи, засвоїв теоретичний матерiал та набв практичного досвiду створення рекурсивних алгоритмiв та написання вiдповiдних їм програм.