**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 № 2.1**

з дисципліни  
«Алгоритми і структури даних»

Виконав Перевірила:

Студент групи ІП-03 Сергієнко А. А.  
Пашковський Євгеній Сергійович  
номер у списку групи: 17

Київ 2021

**Завдання**

Дане натуральне число n. Знайти суму перших n членів ряду чисел, заданого рекурентною формулою. Розв’язати задачу трьома способами (написати три програми):

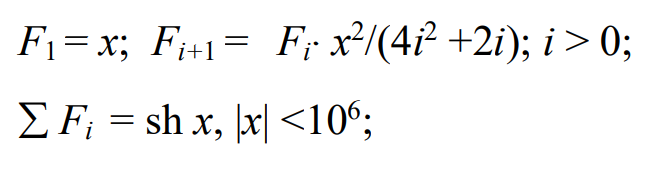
1) в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному спуску;

2) в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному поверненні;

3) в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні.

Програми повинні працювати коректно для довільного натурального n включно з n = 1.

Варіант 17:



**Текст програм**

1. в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному спуску:

#include <stdio.h>

#include <math.h>

#include <time.h>

int main()

{

system("cls");

double x;

unsigned int n = 0;

printf("x: ");

scanf("%lf", &x);

while (n < 1) {

printf("n: ");

scanf("%u", &n);

}

clock\_t begin = clock();

double recursiveFunc(double x, double F, double sumF, unsigned int i)

{

F = i == 0 ? x : F \* ((x \* x) / (4 \* i \* i + 2 \* i));

sumF += F;

i++;

return i < n ? recursiveFunc(x, F, sumF, i) : sumF;

}

double sumF = recursiveFunc(x, 0, 0, 0);

printf("sumF = %.15lf\n", sumF);

printf("Delta: %.15lf\n", fabs((double)(sinh(x) - sumF)));

clock\_t end = clock();

printf("Elapsed: %lf seconds\n", (double)(end - begin) / CLOCKS\_PER\_SEC);

system("pause");

return 0;

}

1. в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному поверненні:

#include <stdio.h>

#include <math.h>

#include <time.h>

int main()

{

system("cls");

double x;

unsigned int n = 0;

printf("x: ");

scanf("%lf", &x);

while (n < 1) {

printf("n: ");

scanf("%u", &n);

}

clock\_t begin = clock();

double sumF = 0;

double recursiveFunc(double x, unsigned int i)

{

double F;

if (i == 0)

{

F = x;

} else {

double m = (x \* x) / (4 \* i \* i + 2 \* i);

F = m \* recursiveFunc(x, i - 1);

}

sumF += F;

return F;

}

double F = recursiveFunc(x, n);

printf("sumF = %.15lf\n", sumF);

printf("Delta: %.15lf\n", fabs((double)(sinh(x) - sumF)));

clock\_t end = clock();

printf("Elapsed: %lf seconds\n", (double)(end - begin) / CLOCKS\_PER\_SEC);

system("pause");

return 0;

}

1. в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні

#include <stdio.h>

#include <math.h>

#include <time.h>

int main()

{

system("cls");

double x;

unsigned int n = 0;

printf("x: ");

scanf("%lf", &x);

while (n < 1) {

printf("n: ");

scanf("%u", &n);

}

clock\_t begin = clock();

double recursiveFunc(double x, double F, unsigned int i)

{

F = i == 0 ? x : F \* ((x \* x) / (4 \* i \* i + 2 \* i));

i++;

return i < n ? F + recursiveFunc(x, F, i) : F;

}

double sumF = recursiveFunc(x, 0, 0);

printf("sumF = %.15lf\n", sumF);

printf("Delta: %.15lf\n", fabs((double)(sinh(x) - sumF)));

clock\_t end = clock();

printf("Elapsed: %lf seconds\n", (double)(end - begin) / CLOCKS\_PER\_SEC);

system("pause");

return 0;

}

**Тестування програми**

*Циклічна програма:*

#include <stdio.h>

#include <math.h>

#include <time.h>

int main()

{

system("cls");

double x;

unsigned int n = 0;

printf("x: ");

scanf("%lf", &x);

while (n < 1) {

printf("n: ");

scanf("%u", &n);

}

clock\_t begin = clock();

double F = x;

double sumF = F;

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

{

F = F \* ((x \* x) / (4 \* i \* i + 2 \* i));

sumF += F;

}

// printf("\nF%u = %.15lf\n", n, recursiveFunc(x));

printf("sumF = %.15lf\n", sumF);

printf("Delta: %.15lf\n", fabs((double)(sinh(x) - sumF)));

clock\_t end = clock();

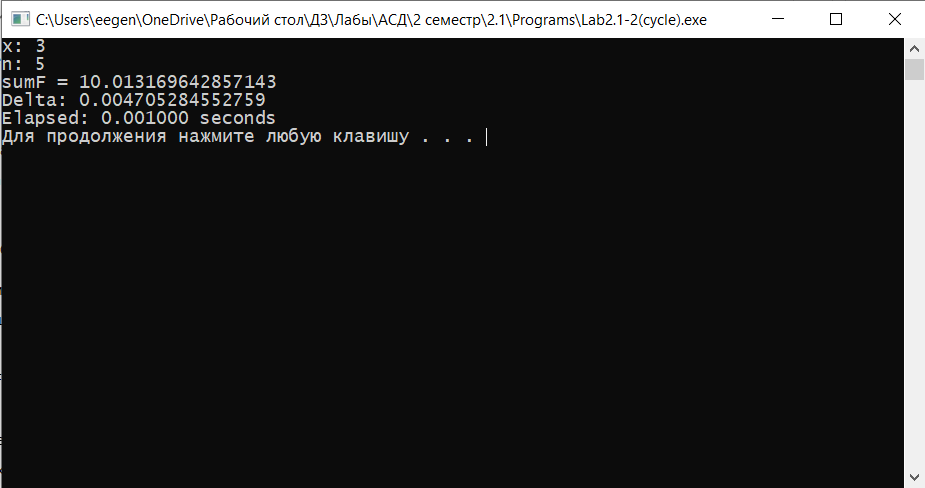
printf("Elapsed: %lf seconds\n", (double)(end - begin) / CLOCKS\_PER\_SEC);

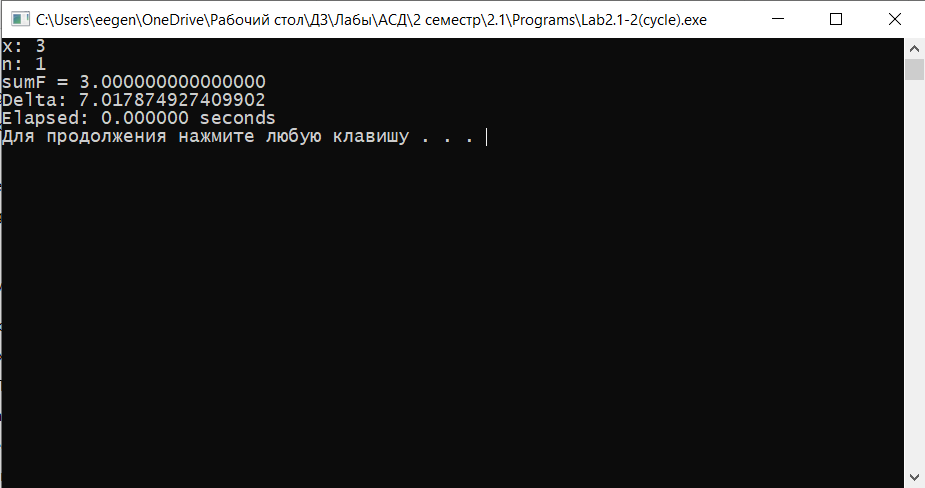
system("pause");

return 0;

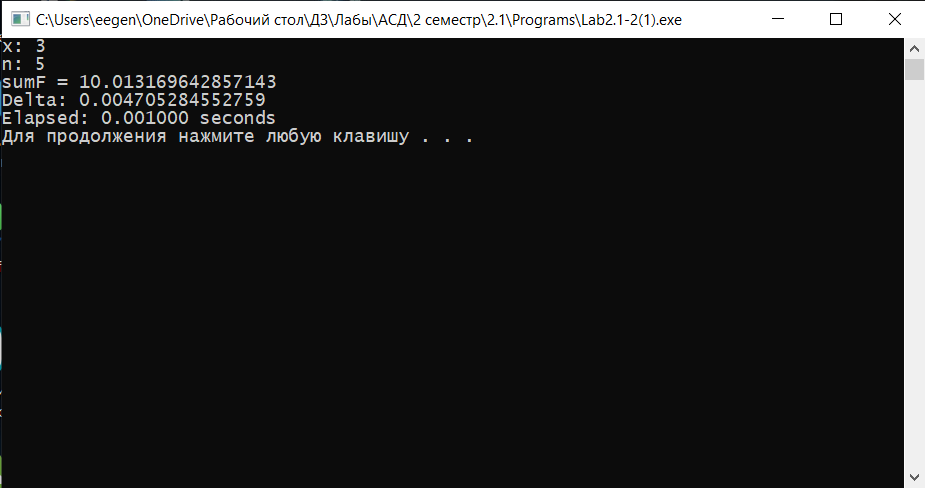
}

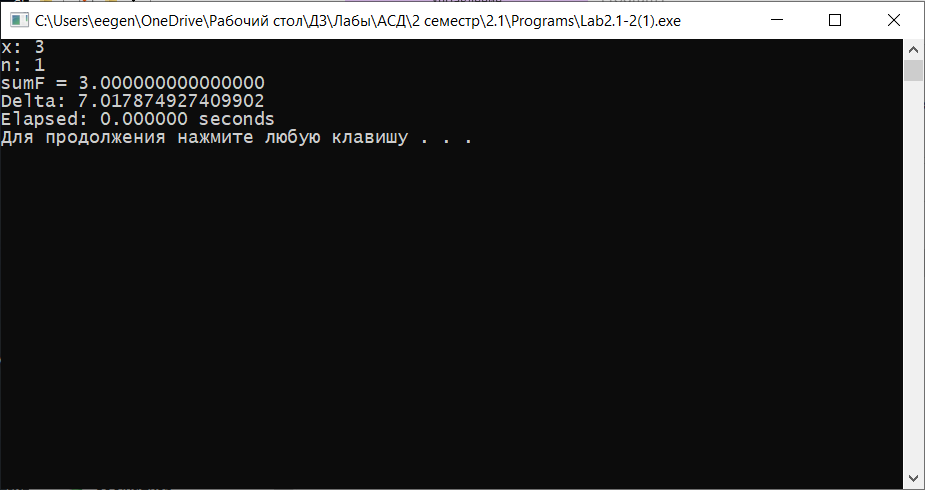
*Результат циклічної програми:*

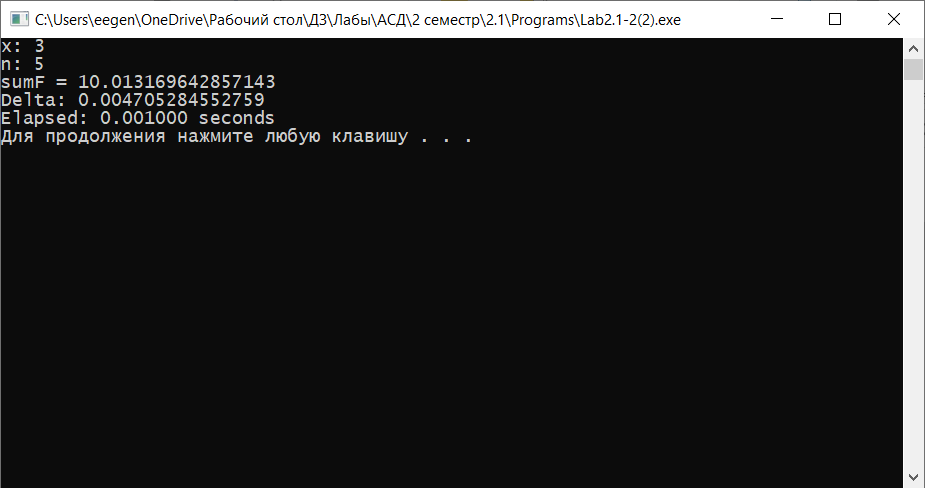


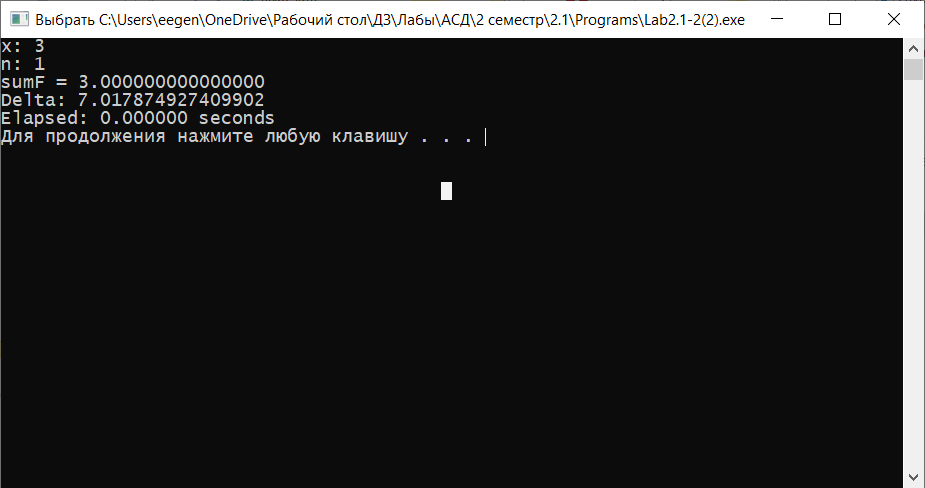


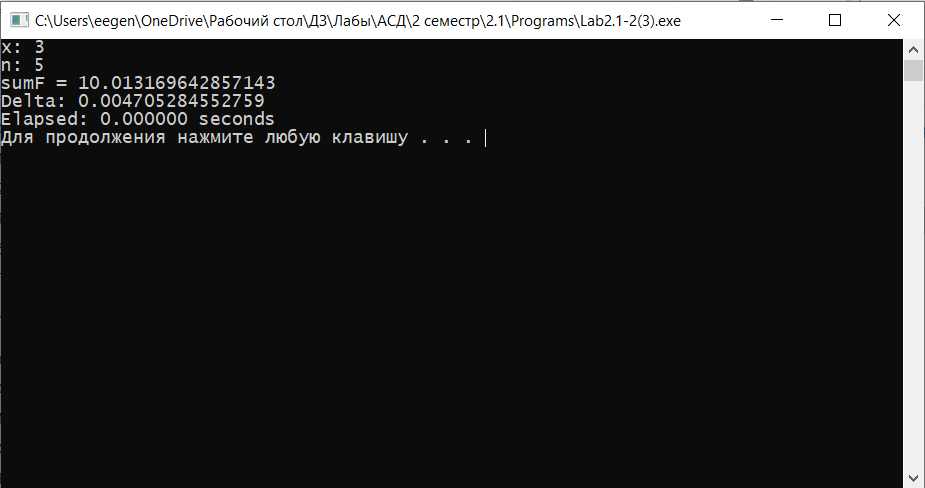
*Результат трьох програм:*

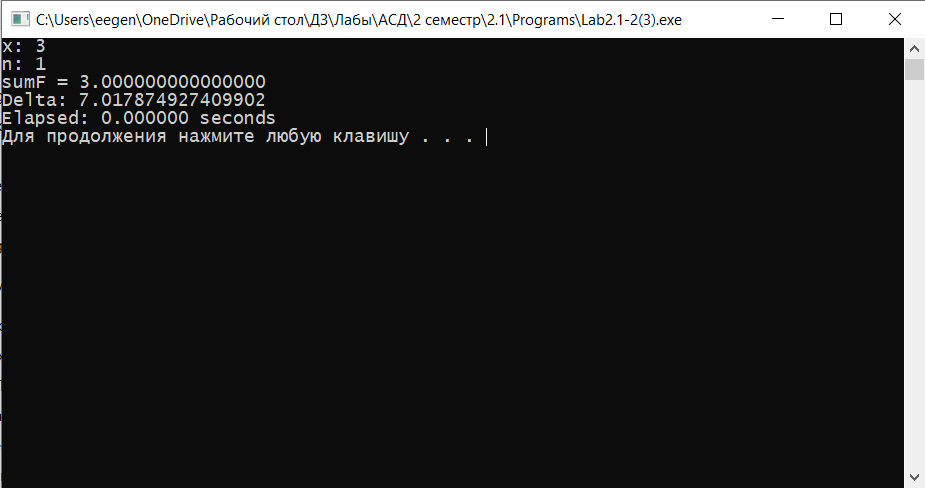












*Обчислення за допомогою калькулятора:*



*Графік залежності похибки від x при n = 5:*