Министерство науки и высшего образования РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа киберфизических систем и управления

УДК 004.421

УТВЕРЖДАЮ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_г.

Отчет

по дисциплине «Теория и технология программирования»

Лабораторная работа №4 «Вычисления значений рядных функций», вариант 14

Выполнил:

Студент

гр.3530902/90001 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Непушкин С.А. подпись, дата

Доцент ВШКФСиУ,

Кандидат технических наук

С. В. Хлопин \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Санкт-Петербург

2019

**1. Цель работы.**

Цель лабораторной работы состоит в формировании умений:

- вычисления значений рядных функций.

**2. Задание:**

Написать программу для вычисления значения функции, заданной с помощью ряда.

Вычисление ряда проводить до условия минимизации значения разности двух

соседних членов ряда меньше заданного эпсилон. Вычислить значения невязки

значений рядной и стандартной Windows ("math.h")

функций , проанализировать динамику

изменения значения невязки в зависимости от количества слагаемых в ряде. Для

остановки счета рядов необходимо использовать следующее условие:.

**3. Текст программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

#include <locale.h>

double sh(double x)

{

return (exp(x) - exp(-x)) / 2;

}

double shEps(double x, double eps)

{

double res = 0;

long double rn = x;

for (int i = 1; rn > eps || rn < -eps; i++)

{

res = res + rn;

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

}

return res;

}

int isDigit(char c)

{

return ((c >= '0') && (c <= '9'));

}

int isDash(char c)

{

return (c == '-');

}

int isDot(char c)

{

return ((c == '.') || (c == ','));

}

int charToDigit(char c)

{

if (isDigit(c))

{

return c - '0';

}

return -1;

}

//0 - Ошибка; 1 - Ввелось корректно

int inputDouble(double \*var)

{

double value = 0;

double negpower = 1;

int dot = 0;

char c = 0;

int digit = 0;

int sign = 1;

int pos = 0;

while ((c = getchar()) != '\n')

{

pos++;

if (isDash(c))

{

if (pos == 1)

{

sign = -1;

continue;

}

else

{

return 0;

}

}

if (isDot(c))

{

if (dot == 0)

{

dot = 1;

continue;

}

else

{

return 0;

}

}

digit = charToDigit(c);

if (digit == -1)

{

return 0;

}

if (dot)

{

negpower /= 10;

value += digit \* negpower;

}

else

{

value = value \* 10 + digit;

}

}

if (pos == 0)

{

return 0;

}

\*var = sign \* value;

return 1;

}

int main()

{

setlocale(LC\_ALL, "Russian");

double xstart;

double xend;

double xstep;

double xideal;

double eps;

printf("Введите x\_start = ");

if (!inputDouble(&xstart))

{

printf("Некорректный ввод числа\n");

return 0;

}

printf("Введите x\_end = ");

if (!inputDouble(&xend))

{

printf("Некорректный ввод числа\n");

return 0;

}

printf("Введите x\_step = ");

if (!inputDouble(&xstep))

{

printf("Некорректный ввод числа\n");

return 0;

}

printf("Введите epsilon = ");

if (!inputDouble(&eps))

{

printf("Некорректный ввод числа\n");

return 0;

}

printf("\n");

printf(" x | mySh(x) | sh(x) | delta \n");

double x;

for (x = xstart; x < xend - xstep/10; x = x + xstep)

{

double func = sh(x);

double myfunc = shEps(x, eps);

double delta = (func - myfunc)\*(func + myfunc);

if (delta < 0) delta \*= -1;

delta = sqrt(delta);

printf("%9.2lf | %28.10lf | %28.10lf | %10.20lf\n", x, myfunc, func, delta);

}

double func = sh(xend);

double myfunc = shEps(xend, eps);

double delta = (func - myfunc)\*(func + myfunc);

if (delta < 0) delta \*= -1;

delta = sqrt(delta);

printf("%9.2lf | %28.10lf | %28.10lf | %10.20lf\n", xend, myfunc, func, delta);

printf("Введите x\_ideal = ");

if (!inputDouble(&xideal))

{

printf("Некорректный ввод числа\n");

return 0;

}

eps = 1;

printf("\n");

printf(" epsilon | mySh(x) | sh(x) | delta \n");

for (int i = 0; i < 7; i++)

{

eps /= 10;

double func = sh(xideal);

double myfunc = shEps(xideal, eps);

double delta = (func - myfunc)\*(func + myfunc);

if (delta < 0) delta \*= -1;

delta = sqrt(delta);

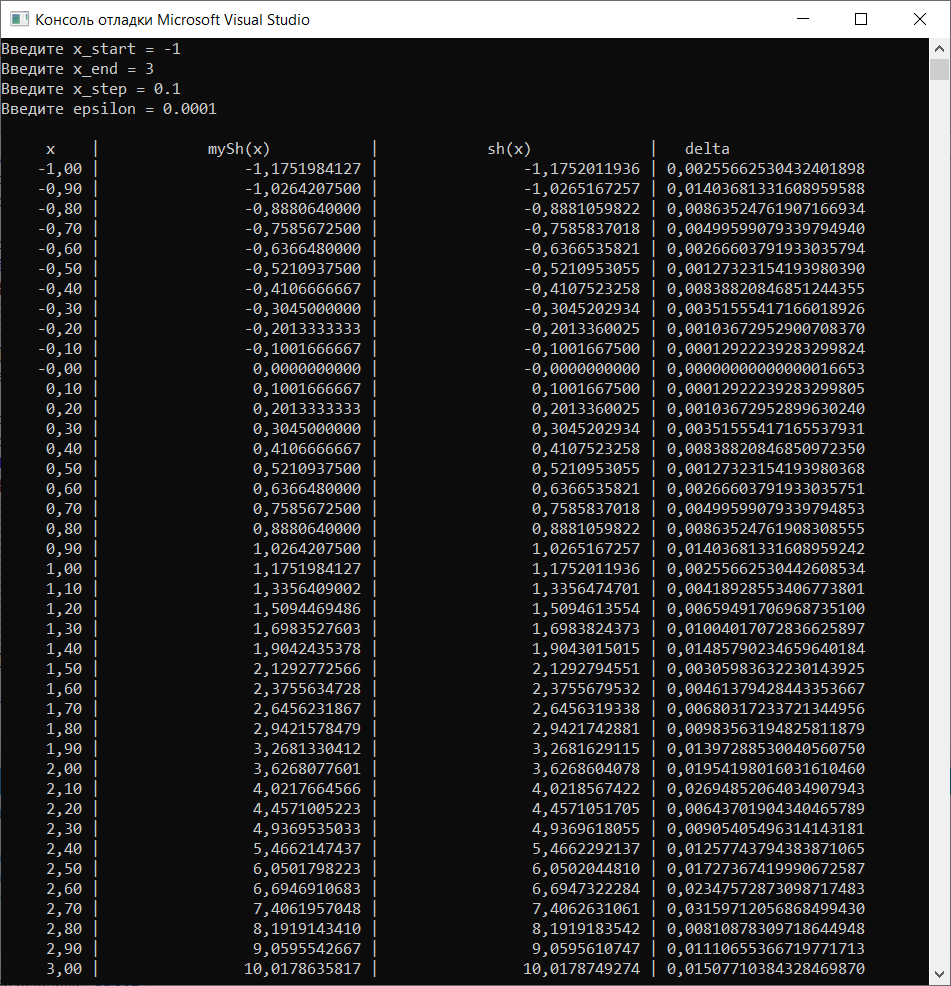
printf("%9.7lf | %28.10lf | %28.10lf | %10.20lf\n", eps, myfunc, func, delta);

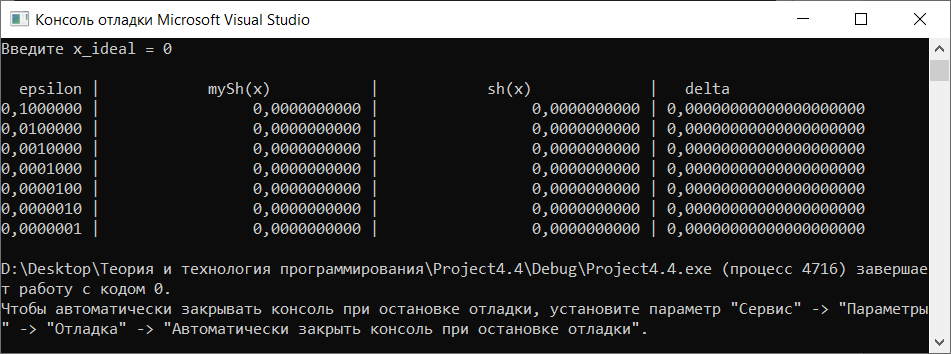
}

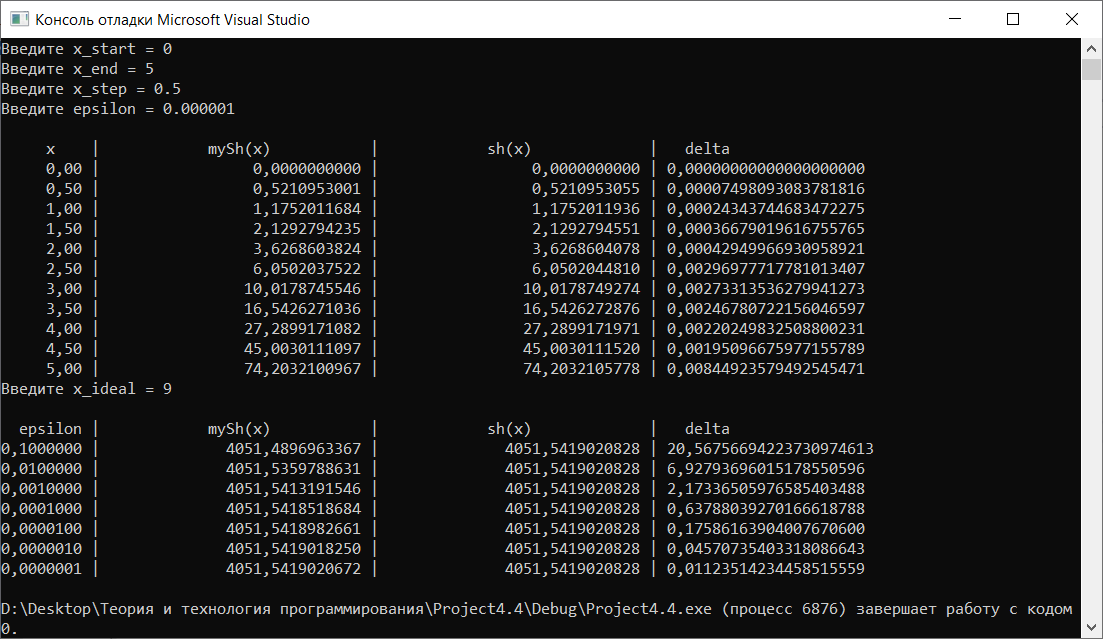
return 0;

}

**4. Пример работы программы**







**Вывод**

В ходе лабораторной работы я:

-узнал о том, что при вводе переменной типа double сохраняется не вводимое число, а приближенное к нему

-узнал о том, как считать функцию sh

-узнал, что к переменным типа double нельзя применять функцию abs

-столкнулся с проблемой обнуления переменной при решении квадратного уравнения, но успешно решил ее