**Московский авиационный институт**

**(Национальный исследовательский университет)**

**Курсовой проект**

**по курсу**

**«Фундаментальная информатика»**

**I семестр**

**Задание 3**

| Студент:  Группа:  Руководитель:  Оценка:  Дата: | Астахова А. С.  М8О-113Б-22  Никулин С. П. |
| --- | --- |

**Москва**

**2022г.**

**Задание**

Составить программу на Си, которая печатает таблицу значений элементарной функции, вычисленной двумя способами: по формуле Тейлора и с помощью встроенных функций языка программирования. В качестве аргументов таблицы взять точки разбиения отрезка на равных частей, находящихся в рекомендованной области хорошей точности формулы Тейлора. Вычисления по формуле Тейлора проводить по экономной в сложностном смысле схеме с точностью , где - машинное эпсилон, аппаратно реализованного вещественного типа для данной ЭВМ, а – экспериментально подбираемый коэффициент, обеспечивающий приемлемую сходимость. Число итераций должно ограничиваться сверху числом порядка 100. Программа должна сама определять машинное и обеспечивать корректные размеры генерируемой таблицы.

Вариант 2

Функция:

Отрезок:

Ряд:

**Алгоритм решения**

1. Найти число эпсилон
2. Распечатать заголовок таблицы.
3. В цикле for обработать все значения x.
4. Пока числа итераций не достигло 100 и модуль вычисленного члена больше машинного эпсилона, выводить следующий значения: значение ‘x’, сумму ряда Тейлора, значение функции и число итераций.
5. Вывод результатов.
6. Если x при вычислениях попадет не в b, а в окрестность, больше b, то вычислим оба значение для x=b.

**Листинг программного кода**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

#include <locale.h>

#include <stdbool.h>

int main()

{

setlocale(LC\_ALL, "Rus");

double eps = 1.0;

while (1.0 + eps / 2 > 1) {

eps /= 2;

}

double A = 0.0;

double B = 0.5;

int n;

double func;

int steps;

double step = 0.05;

printf("Введите количество разбиений отрезка: ");

scanf("%d", &steps);

printf("\n");

step = (B - A) / steps;

printf("--------------------------------------------------------------------------------------------------");

printf("\n|\tx\t|\tряд Тейлора \t|\tзначение функции \t| число итераций |\n");

printf("--------------------------------------------------------------------------------------------------");

bool flag = false;

for (double x = A; x <= B; x+=step) {

double elem, sum\_elem = 0;

n = 1;

do {

elem = pow(x, n) / n;

sum\_elem += 2 \* elem;

n+=2;

} while (fabs(elem) > eps && n <= 100);

func = log((1 + x) / (1 - x));

printf("\n|\t%.2f\t|\t%.18f\t|\t%.18f\t|\t%d\t|", x, sum\_elem, func, n);

if (x + step > B && x != B) {

flag = true;

}

}

if (flag) {

double elem, sum\_elem = 0;

n = 1;

do {

elem = pow(B, n) / n;

sum\_elem += 2 \* elem;

n += 2;

} while (fabs(elem) > eps && n <= 100);

func = log((1 + B) / (1 - B));

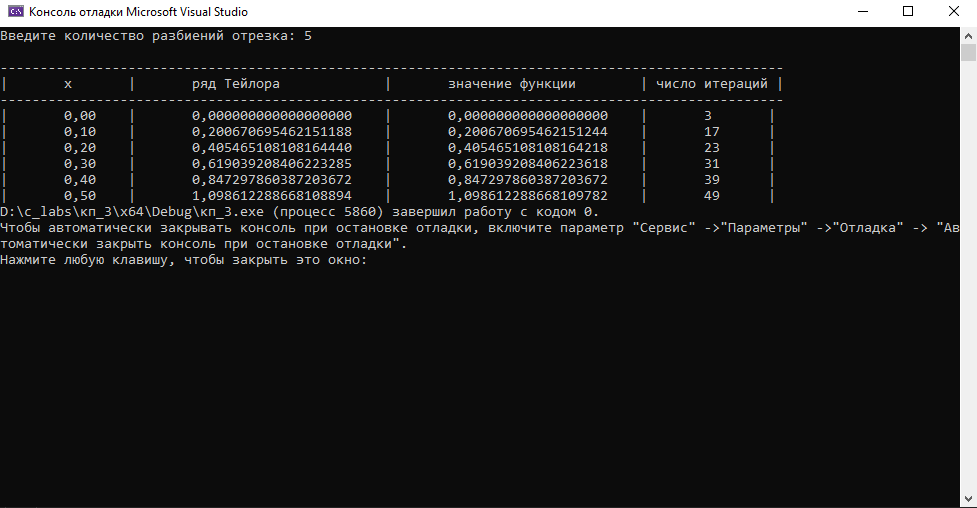
printf("\n|\t%.2f\t|\t%.18f\t|\t%.18f\t|\t%d\t|", B, sum\_elem, func, n);

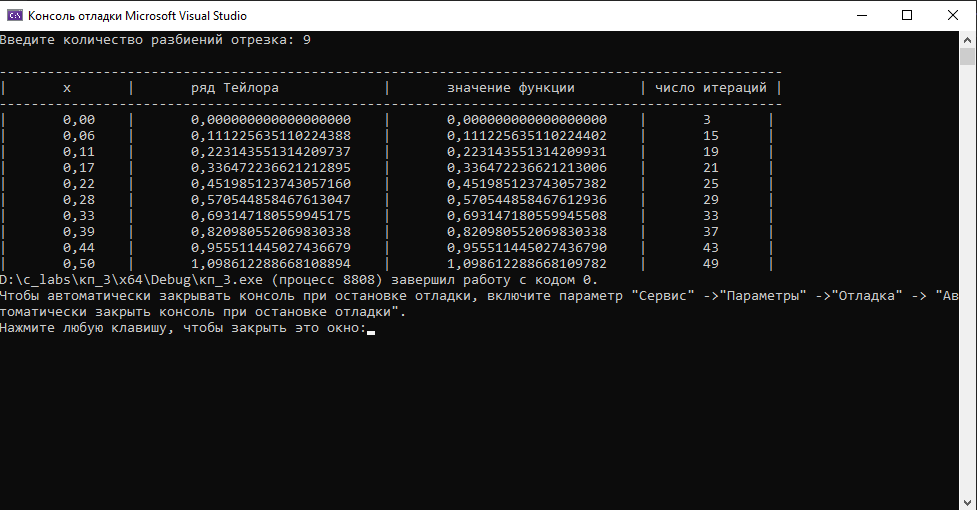
}

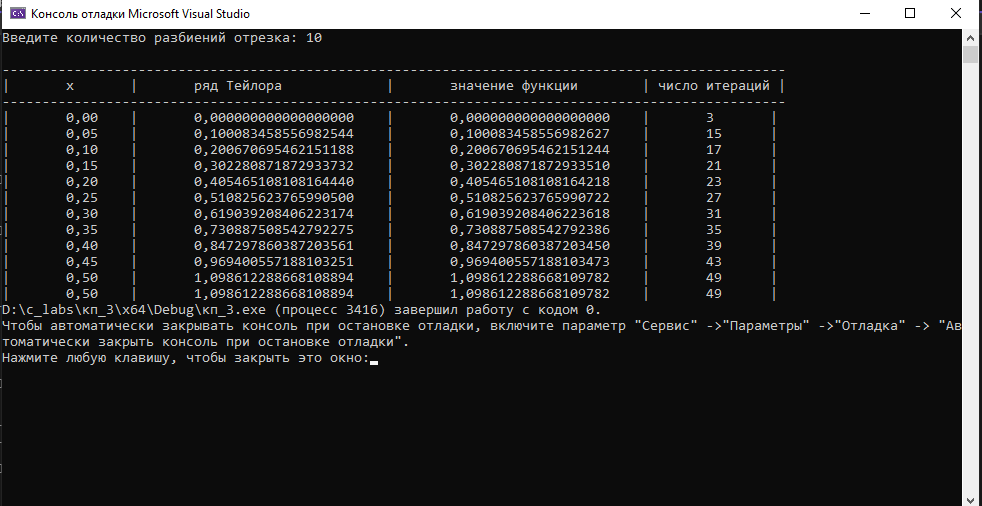
return 0;

}

**Результат работы программы**







**Заключение**

Таблица показывает, что значения ряда Тейлора имеют отличия от встроенной функции примерно после 15 знака после запятой. Это означает, что, несмотря на точность данного метода задания функций, он не является совершенным.