Московский авиационный институт

(национальный исследовательский университет)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Курсовой проект

по курсу «Вычислительные системы»

1 семестр

Задание 3.

Тема: «Вещественный тип. Приближённые вычисления. Табулирование функций.»

|  |  |
| --- | --- |
| **Студент:** | Чапалда М.О |
| **Группа:** | М8О - 101Б - 22 |
| **Преподаватель:** | Крылов С.С |
| **Подпись:** |  |
| **Оценка:** |  |

Москва 2022

Оглавление

[Задание. 2](#__RefHeading___Toc518_3306786435)

[Формулировка. 2](#__RefHeading___Toc520_3306786435)

[Вариант задания. 3](#__RefHeading___Toc522_3306786435)

[Работа. 3](#__RefHeading___Toc524_3306786435)

[Код программы. 3](#__RefHeading___Toc526_3306786435)

[Описание алгоритма. 4](#__RefHeading___Toc528_3306786435)

[Протокол выполнения программы. 5](#__RefHeading___Toc535_3306786435)

[Вывод. 6](#__RefHeading___Toc542_3306786435)

# Задание.

## Формулировка.

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 [a, b] на n равных частей (n + 1 точка включая концы отрезка), находящихся в рекомендованной области хорошей точности формулы Тейлора. Вычисления по формуле Тейлора проводить по экономной в сложностном смысле схеме с точностью EPS\*k, где EPS — машинное эпсилон аппаратно реализованного вещественного типа для данной ЭВМ, а k — экспериментально подбираемый коэффициент, обеспечивающий приемлемую сходимость. Число итераций должно ограничиваться сверху числом порядка 100. Программа должна сама определять машинное EPS и обеспечивать корректные размеры генерируемой таблицы.

## Вариант задания.



# Работа.

## Код программы.

#include <stdio.h>

#include <math.h>

int fact(int n)

{

    if (n == 0 || n == 1)

        return 1;

    else

    {

        return n \* fact(n - 1);

    }

}

int main()

{

    double machine\_epsilon;

while (1+machine\_epsilon>1){

    machine\_epsilon/=2;

}

    printf("Vvedite kolichestvo ravnykh razbiyeniy otrezka [a, b]: n = ");

    int n;

    scanf("%d", &n);

    if (n < 1) {

        printf("Vvedennoye znacheniye dolzhno byt' bol'she nulya!\n");

        return 0;

    }

    double e0 = machine\_epsilon;

    double d = 1 / (double)n;

    printf("Mashinnoye epsilon dlya tipa long double: e = %21.19f\n", e0);

    printf("+-----------------------------------------------------------------------------------------------------+\n");

    printf("|            Tablitsa znacheniy ryada Teylora i standartnoy funktsii dlya f(x) = exp(pow(x, 2))       |\n");

    printf("+-------+-----------------------------------------+---------------------------------+-----------------+\n");

    printf("|   x   |chastichnaya summa ryada dlya exp(pow(x, 2))|  znacheniye funktsii f(x)    |chislo iteratsiy|\n");

    printf("+-------+-----------------------------------------+---------------------------------+-----------------+\n");

    for (double x = 0; x <= 1.0; x += d) {

        double math\_func = exp(pow(x, 2));

        double res = 1;

        int n1 = 1;

        double numerator = 1;

        double denominator = 1;

        double cur = 1;

        while(cur > e0 || -cur > e0 || n1 <= 100){

            numerator \*= x \* x;

            denominator \*= n1;

            cur = numerator / denominator;

            res += cur;

            n1++;

        }

        printf("| %.3f\t|          %.19f          |      %.19f      |        %d        |\n", x, res, math\_func, n1);

    }

    printf("+-------+-----------------------------------------+---------------------------------+-----------------+\n");

    return 0;

}

## Описание алгоритма.

Подключаем заголовочный файл **math.h** для вызова функций **pow()**, **exp()** и **fabs()**, напишем функцию **fact()**, возвращающую факториал. Так как функция **fact()** определена только для целых чисел и возвращает целое число, то ее вызов никак не повлияет на погрешность результата.

Задаем машинный эпсилон для типа long double при помощи функции **machine\_epsilon()**, которая делит переменную **e** на 2 пока (1 + **e**/2) > 1. Если (1 + **e**/2) = 1, то **e** будет равно машинному нулю; функция возвращает результат предыдущей итерации. Присваиваем его переменной **e0**.

Задаем ввод количества равных разбиений отрезка [a,b], **n**, если **n** > 1, то объявляем число **d**, равное **b**/**n** - единичный отрезок.

Объявляем **x** типа double - аргумент f(x).

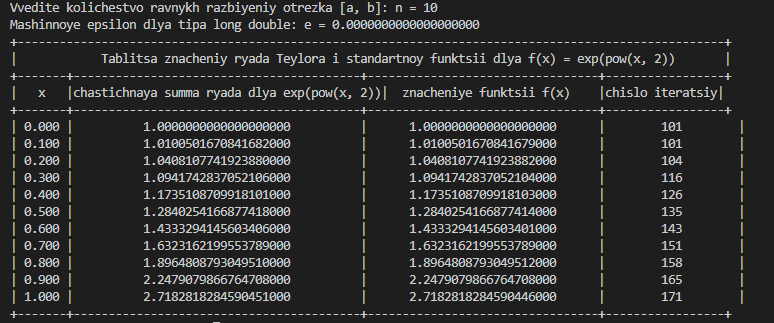
В цикле for проходим по всем точкам разбиений {0, n}, присваиваем аргументу значение **d** \* i (произведение единичного отрезка на номер итерации), вычисляем значение данной функции **math\_func** с помощью функций из **math.h**.

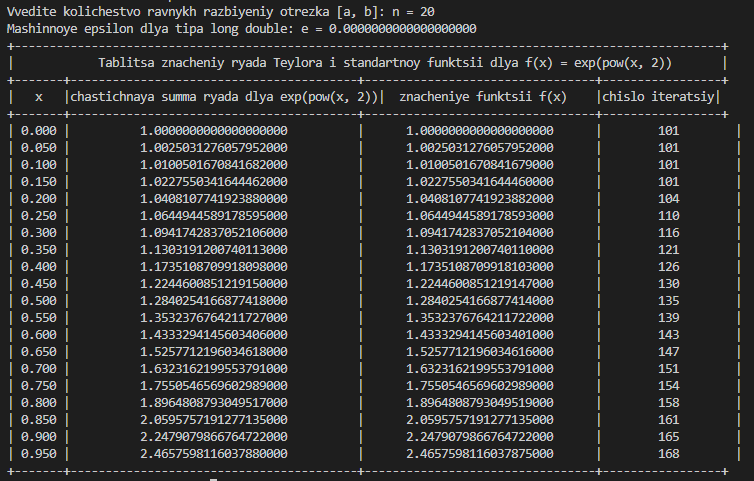
Суммируем данный ряд Тейлора, объявляем переменные **res** - результат, **n1** - количество итераций, определяющее точность значения f(x), **numenator** и **denomenator** - числитель и знаменатель члена ряда.

Первый член ряда Тейлора уже задан объявим **cur** - значение дроби, начиная со второго члена циклически умножаем числитель и знаменатель согласно условию, прибавляем к переменной **res** значение дроби. Условие на выход из цикла: если модуль значения члена ряда меньше **e0**.

В виде таблицы выводим переменные **x**, **res**, **math\_func**, **n1**.

## Протокол выполнения программы.





# Вывод

В ходе выполнения данного задания курсового проекта я научилась реализовывать программную версию вычисления значений функции пользуясь рядом Тейлора для этой функции. Научилась выводить данные в виде таблицы, оптимизировать вывод значений переменных с плавающей точкой. Научилась с помощью алгоритма определять машинное эпсилон. Закрепила свои навыки работы в языке программирования Си.