

**Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский Авиационный Институт»
(Национальный Исследовательский Университет)**

**Факультет №8 «Информационные технологии и прикладная
математика»**

Курсовой проект
по курсу «Вычислительные системы»
1 семестр

Задание 3

Автор работы:

студент 1 курса, группа М8О-102Б-21

Кажекин Денис Андреевич

Руководитель проекта:

Доцент кафедры 806 Никулин С.П.

Дата сдачи:

Москва 2021

Цель работы

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 $\varepsilon \times k$, где ε — машинное эpsilon 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 ε и обеспечивать корректные размеры генерируемой таблицы.

Вариант №16

№	ряд	a	b	функция
16	$1 + 3x^2 + \dots + \frac{2n+1}{n!} x^{2n}$	0.0	1.0	$(1 + 2x^2)e^{x^2}$

Алгоритм

- 1) Деля единицу пополам до тех пор, пока ее значение не превратится в машинный ноль, вычислить машинное эpsilon.
- 2) Распечатать результат
- 3) Запросить у пользователя необходимое число разбиений отрезка.
- 4) Разделить отрезок значений x на соответствующее число частей и вычислить шаг изменения переменной x .
- 5) Распечатать заголовок таблицы.
- 6) В цикле «for» обработать все значения x . Пока значение вычисленного члена по модулю не будет меньше машинного эpsilon, для каждого значения x вычислять сумму членов ряда Тейлора.

Описание программы

Использованные переменные:

тип `double`:

- 1) `x` - начало отрезка;
- 2) `chast` - шаг разбиения;
- 3) `function` - значение функции на данном `X`;
- 4) `polynom_sum` - сумма ряда Тейлора;
- 5) `chlen_ryada` - значение члена ряда Тейлора;
- 6) `e` - машинное эпсилон для данной ЭВМ;

тип `int`:

- 1) `iter` – номер члена полинома;
- 2) `n` - число разбиений отрезка;
- 3) `k` - экспериментально подобранный коэффициент, обеспечивающий приемлемую сходимость.

Входные данные: целое число разбиений отрезка

Выходные данные: значение машинного эпсилон, таблица с вычисленными двумя способами - по формуле Тейлора и при помощи встроенных функций библиотеки `math.h` - значениями функции в точках разбиения заданного отрезка и на его концах.

Программа

```
#include "stdio.h"
#include "math.h"

long int factorial(long int n)
{
    if (n == 0 || n == 1) return 1;
    return n * factorial(n - 1);
}

int main(){
    int n, k = 1, iter;
    double e = 1, chast, function, x = 0.0, chlen_ryada, polynom_sum=0;
    while(1 + e/2 > 1){
        e /= 2;
    }
    printf("Введите число разбиения отрезка [0.0, 1.0]: ");
    scanf("%i", &n);
    printf("Машинное эpsilon: %.20f\n", e);
    chast = 1.0 / n;
    e *= k;
    printf("=====\n");
    printf(" X\t\t Сумма ряда\t\t Значение функции\t i\n");
    for(int i = 0; i <= n; i++){
        function = (1 + 2 * pow(x,2)) * exp(pow(x,2));
        iter = 0;
        while(iter <= 100){
            chlen_ryada = (((2 * iter) + 1) * pow(x,(2*iter))) / factorial(iter));
            if((chlen_ryada > e) || (chlen_ryada < -e)){
                polynom_sum += chlen_ryada;
            }
            else{
                break;
            }
            iter += 1;
        }
    }
    printf("%.1f\t %.20f\t %.20f\t %i\n",x, polynom_sum, function, iter);
}
```

```

    polynom_sum = 0;
    x += chast;
}
printf("=====\\n");
return 0;
}

```

Результат:

```

deniskazhekin@MacBook-Air-Denis untitled4 % gcc main.c
deniskazhekin@MacBook-Air-Denis untitled4 % ./a.out
Введите число разбиения отрезка [0.0, 1.0]: 5
Машинное эpsilon: 0.000000000000000022204

```

```

=====
X                Сумма ряда                Значение функции                i
0.0              1.0000000000000000000000  1.0000000000000000000000        1
0.2              1.12407563612777927631    1.12407563612777927631        9
0.4              1.54903434970918962854    1.54903434970918962854       12
0.6              2.46532659304378576337    2.46532659304378576337       14
0.8              4.32397640481529066392    4.32397640481529066392       17
1.0              8.15484548537713216376    8.15484548537713571648       20
=====

```

```

deniskazhekin@MacBook-Air-Denis untitled4 % gcc main.c
deniskazhekin@MacBook-Air-Denis untitled4 % ./a.out
Введите число разбиения отрезка [0.0, 1.0]: 10
Машинное эpsilon: 0.000000000000000022204

```

```

=====
X                Сумма ряда                Значение функции                i
0.0              1.0000000000000000000000  1.0000000000000000000000        1
0.1              1.03025117042585145910    1.03025117042585123706        7
0.2              1.12407563612777927631    1.12407563612777927631        9
0.3              1.29112565477214813470    1.29112565477214835674       10
0.4              1.54903434970918962854    1.54903434970918962854       12
0.5              1.92603812503161231362    1.92603812503161209158       13
0.6              2.46532659304378487519    2.46532659304378487519       14
0.7              3.23198611551164960431    3.23198611551165004840       16
0.8              4.32397640481528888756    4.32397640481528888756       17
0.9              5.88951892509235364059    5.88951892509235275242       18
1.0              8.15484548537712861105    8.15484548537713216376       20
=====

```

Заключение:

Таблица показывает, что значения ряда Тейлора имеют отличия от встроенной функции примерно после 13-15 знака после запятой. Это означает, что, несмотря на точность данного метода задания функций, он не является совершенным.

Такого рода программы не имеют прикладного применения, т.к. они затрачивают много времени при компиляции и являются бессмысленными при наличии встроенных функций, однако дают представление о том, как задаются математические функции на языке Си и прочих языках программирования.