

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:	Модин-Глазков Б.А.
Группа:	М8О - 112Б - 22
Преподаватель:	Никулин С.П.
Подпись:	
Оценка:	

Москва 2022

Оглавление

1. Задание
2. Описание алгоритма
3. Код программы
4. Протокол выполнения
5. Вывод

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 $[a, b]$ 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 $\text{epsilon} * k$, где epsilon — 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.

6	$x + \frac{x^3}{3!} + \dots + \frac{x^{2n-1}}{(2n-1)!}$	0.0	1.0	sh x
---	---	-----	-----	------

Описание алгоритма

- Импортируем библиотеку `<math.h>`
- Заводим переменные `double a`(начало отрезка) = 0.0, `double b`(конец отрезка) = 1.0, `double epsilon`(машинное эпсилон = 1), `double x`(переменная для работы с точками), `double S` = 0 (сумма членов ряда Тейлора), `double chislo`(член ряда Тейлора), `int n`(кол-во частей при разбиении отрезка), `int k` = 0(количество слагаемых ряда Тейлора)

- Вычисляем машинное эпсилон
- Вводим значения для n , k .
- Начинаем проходить циклом `for` все точки, которые получились при разделении нашего отрезка $[a, b]$ на n равных частей.
- $S = 0$
- Начинаем цикл подсчета суммы членов ряда Тейлора из 100 элементов
- Считаем `chislo`(конкретный член ряда Тейлора) и прибавляем 1 к переменной K
- Если модуль нашего члена ряда Тейлора будет $> \text{epsilon} * k$, то увеличиваем S
- Иначе выходим из цикла
- Печатаем строчку таблицы
- Обнуляем k

Код программы

```
#include <stdio.h>
#include <math.h>
long long factorial(long long i)
{
    if (i==0) return 1;
    else return i*factorial(i-1);
}
int main() {
    double a = 0.0, b = 1.0, epsilon = 1.0, x, S = 0, chislo;
    int n, number = 0, k = 0;
    while (epsilon/2.0 + 1.0 > 1.0) {
        epsilon/=2.0;
    }
    printf("Введите количество частей n, на которые разбивается интервал, а также значение k, нужное для определения точности: ");
    scanf ("%d %d", &n, &k);
    printf("Машинное эпсилон для типа double: epsilon = %.16lf.\n", epsilon);
    printf("-----\n");
    printf("|   x   | Сумма ряда Тейлора | Значение функции | Число итераций |\n");
    printf("|-----|-----|-----|-----|\n");
    for (x = a; x <= b; x+=(b - a)/n) {
        S=0;
        while (number < 100) {
            number += 1;
            chislo = powl(x, 2*number-1)/factorial(2*number-1);
            if (fabs(chislo) > epsilon*k) {
                S += chislo;
            }
            else {
                break;
            }
        }
        printf("| %.3lf | %.16lf | %.16lf | %10d | \n", x, S, (exp(x)-(1.0/exp(x)))/2, number);
        number = 0;
    }
    printf("-----\n");
    return 0;
}
```

```
}
```

Протокол выполнения программы.

```
bogdanmodin@mac ~ % cat >r.c
#include <stdio.h>
#include <math.h>
long long factorial(long long i)
{
    if (i==0) return 1;
    else return i*factorial(i-1);
}
int main() {
    double a = 0.0, b = 1.0, epsilon = 1.0, x, S = 0, chislo;
    int n, number = 0, k = 0;
    while (epsilon/2.0 + 1.0 > 1.0) epsilon/=2.0;

    printf("Введите количество частей n, на которые разбивается интервал, а также
значение k, нужное для определения точности: ");
    scanf ("%d %d", &n, &k);
    printf("Машинное эpsilon для типа double: epsilon = %.16lf.\n", epsilon);
    printf("-----\n");
    printf("|   x   | Сумма ряда Тейлора | Значение функции | Число итераций | \n");
    printf("-----|-----|-----|-----| \n");
    for (x = a; x <= b; x+=(b - a)/(double)n) {
        S=0;
        while(number<1number += 1;
            chislo = powl(x,2*number-1)/factorial(2*number-1);
            if (fabs(chislo) > epsilon*k) {
                S += chislo;
            }
            else {
                break;
            }
        }
        printf("| %.3lf | %.16lf | %.16lf |%10d | \n", x, S, (exp(x)-
(1.0/exp(x)))/2, number);

        number = 0;
    }
    printf("-----\n");
    return 0;
}
```

^C

```
bogdanmodin@mac ~ % gcc r.c
bogdanmodin@mac ~ % ./a.out
Введите количество частей n, на которые разбивается интервал, а также значение k, нужное
для определения точности: 10 5
Машинное эpsilon для типа double: epsilon = 0.0000000000000002.
```

x	Сумма ряда Тейлора	Значение функции	Число итераций
0.000	0.0000000000000000	0.0000000000000000	1
0.100	0.1001667500198440	0.1001667500198441	6
0.200	0.2013360025410935	0.2013360025410940	6
0.300	0.3045202934471426	0.3045202934471427	7
0.400	0.4107523258028145	0.4107523258028155	7
0.500	0.5210953054937474	0.5210953054937474	8
0.600	0.6366535821482409	0.6366535821482412	8
0.700	0.7585837018395336	0.7585837018395336	9
0.800	0.8881059821876229	0.8881059821876229	9
0.900	1.0265167257081749	1.0265167257081751	9
1.000	1.1752011936438014	1.1752011936438014	10

```
bogdanmodin@mac ~ % ./a.out
Введите количество частей n, на которые разбивается интервал, а также значение k, нужное
для определения точности: 20 10
Машинное эpsilon для типа double: epsilon = 0.0000000000000002.
```

x	Сумма ряда Тейлора	Значение функции	Число итераций
0.000	0.0000000000000000	0.0000000000000000	1
0.050	0.0500208359376550	0.0500208359376551	5
0.100	0.1001667500198440	0.1001667500198441	6
0.150	0.1505631331516127	0.1505631331516126	6
0.200	0.2013360025410935	0.2013360025410940	6
0.250	0.2526123168081683	0.2526123168081683	7
0.300	0.3045202934471425	0.3045202934471427	7
0.350	0.3571897294372717	0.3571897294372718	7
0.400	0.4107523258028144	0.4107523258028155	7
0.450	0.4653420169341978	0.4653420169341977	8
0.500	0.5210953054937473	0.5210953054937473	8
0.550	0.5781516037434542	0.5781516037434542	8
0.600	0.6366535821482409	0.6366535821482412	8
0.650	0.6967475261264388	0.6967475261264401	8
0.700	0.7585837018395337	0.7585837018395336	9
0.750	0.8223167319358300	0.8223167319358300	9
0.800	0.8881059821876232	0.8881059821876232	9
0.850	0.9561159599886322	0.9561159599886323	9
0.900	1.0265167257081753	1.0265167257081758	9
0.950	1.0994843179306719	1.0994843179306728	9
1.000	1.1752011936438014	1.1752011936438014	10

bogdanmodin@mac ~ % ./a.out

Введите количество частей n, на которые разбивается интервал, а также значение k, нужное для определения точности: 20 100

Машинное эпсилон для типа double: epsilon = 0.0000000000000002.

x	Сумма ряда Тейлора	Значение функции	Число итераций
0.000	0.0000000000000000	0.0000000000000000	1
0.050	0.0500208359376550	0.0500208359376551	5
0.100	0.1001667500198413	0.1001667500198441	5
0.150	0.1505631331516127	0.1505631331516126	6
0.200	0.2013360025410935	0.2013360025410940	6
0.250	0.2526123168081623	0.2526123168081683	6
0.300	0.3045202934471425	0.3045202934471427	7
0.350	0.3571897294372717	0.3571897294372718	7
0.400	0.4107523258028144	0.4107523258028155	7
0.450	0.4653420169341928	0.4653420169341977	7
0.500	0.5210953054937276	0.5210953054937473	7
0.550	0.5781516037434542	0.5781516037434542	8
0.600	0.6366535821482409	0.6366535821482412	8
0.650	0.6967475261264388	0.6967475261264401	8
0.700	0.7585837018395301	0.7585837018395336	8
0.750	0.8223167319358198	0.8223167319358300	8
0.800	0.8881059821876232	0.8881059821876232	9
0.850	0.9561159599886322	0.9561159599886323	9
0.900	1.0265167257081753	1.0265167257081758	9
0.950	1.0994843179306719	1.0994843179306728	9
1.000	1.1752011936438014	1.1752011936438014	10

bogdanmodin@mac ~ %

Вывод

В ходе выполнения данного задания курсового проекта я научился реализовывать программную версию вычисления значений функции пользуясь рядом Тейлора для этой функции.