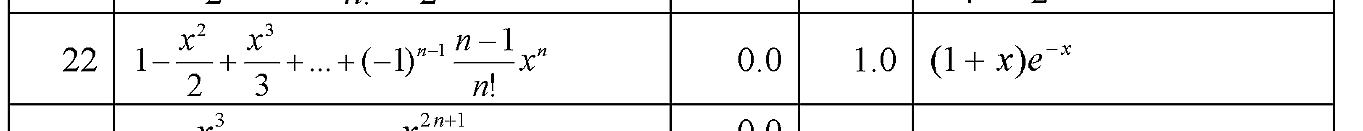
**Задача**

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 *ε\*k*, где *ε* – машинное эпсилон аппаратно реализованного 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 кореектные размеры генерируемой таблицы.

**Вариант**

**Общий метод решения**

Вычисление значений функции на отрезке от 0.0 до 1.0 через ряд Тейлора и с помощью программных средств.

Для нахождения следующего члена суммы (не учитывается знак “-”) найдем его коэффициент(без знака) и воспользуемся определнием факториала.

При *k = n* коэффициент под номером *k* имеет вид:

При *k* = *n+1* имеем:

Для перехода от члена под номером *n* к члену под номером *n+1,* раскроем факториал в 1 и 2 случаях по определению:

1)

2)

Таким образом, для перехода от *n* к *n+1* коэффициенту нужно умножить предыдущий на n и поделить на *(n-1)(n+1),* что равно *(n\*n-1)*.

Далее для смены знака приписываем к предыдущему значению “-” и для увеличения степени умножаем его на “x”.

**Общие сведения о программе**

Аппаратное обеспечение: ноутбук

Операционная система: Ubuntu 16.04 LTS

Язык и система программирования: GNU C

Число строк программы: 36

Местонахождение файлов: /home/zero/Документы/МАИ/ФИ/КП

Компиляция программы в терминале: *gcc 3kp22.c -lm -o 3kp22.helloworld*

Вызов программы: *./3kp22.helloworld*

**Функциональное назначение**

Программа преднозначена для проведения вычислений значения функции в определенной точке двумя способами. Объем данных не ограничен, а значения ограничены размерами переменной типа *long double* для аргумента функции и типом *long double* для значения функции.

**Описание логической структуры**

Программа определяет значение функции в определенной точке при помощи ряда Тейлора, а также с помощью программных средств языка программирования. Ряд Тейлора преобразуется в функцию, которая вычисляет слагаемые ряда. Далее происходит сложение слагаемых до тех пор, пока одно из них не станет ничтожно малым (по модулю меньше *ε*). В результате выводится таблица с текущим значением аргумента, значением функции, вычесленной при помощи ряда Тейлора и с помощью подключаемой библиотечной функции и количество слагаемых.

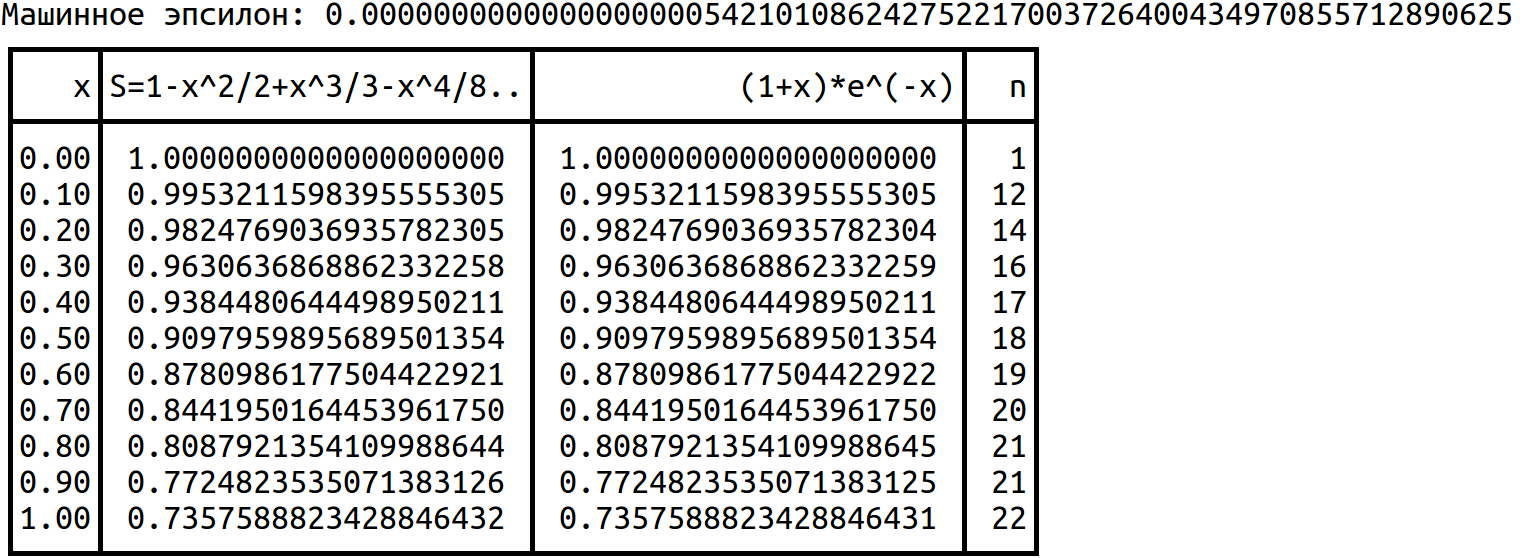
**Описание переменных и констант**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| n | int | Счетчик количества слагаемых |
| step | long double | Шаг, который используется при разбиении отрезка [0.0;1.0] на части |
| a | Левая граница отрезка |
| b | Правая граница отрезка |
| Summ | Сумма слагаемых ряда Тейлора в точке *х* |
| eps | Машинное эпсилон |
| currentTerm | Текущее слагаемое ряда Тейлора |
| x | Значение, в котором вычисляется значение функции |

**Входные данные**

Число разбиений отрезка: 10

**Выходные данные**



**Тестовые примеры**

Не предусмотрены.

**Дневник отладки**

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Место | Событие | Действия по исправлению |
| 17.12.2018 | дом | Из-за погрешности функции *powl* результат отличался на 4 знака. | Использована функция *expl* |

**Выводы по задаче**

Составлена программа на языке Си, вычисляющая значение функции двумя способами. Значения совпадают на первых 18-19 знаках после запятой.

**Протокол**

daniel@daniel:~/$ cat head.txt

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

Курсовая работа №3

Вещественный тип. Приближенные вычисления. Табулирование функций.

Выполнил: Пивницкий Даниэль Сергеевич

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

daniel@daniel:~/$ cat 3kp22.c

#include <stdio.h>

#include <math.h>

int main()

{

long double step, a = 0, b = 1, Summ, eps = 1, currentTerm, x;

int n; //число слагаемых

printf("Введите число разбиений отрезка: ");

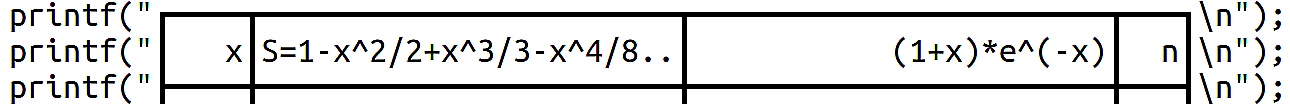
scanf("%Lf",&step);

step = (b-a)/step; //шаг после разбиения

while(1 + eps > 1) //нахождение машинного эпсилон

eps /= 2;

printf("Машинное эпсилон: %.64Lf\n", eps);



for(x = a; x < b + step; x += step)

{

Summ = 1.;

n = 1;

currentTerm = x;

while((currentTerm > eps || currentTerm < -eps) && n <= 100)

{

if(n == 1)

currentTerm = -currentTerm\*x/2;

else

currentTerm = -currentTerm\*x\*(n)/(n\*n-1);

Summ += currentTerm;

n++;

}

printf("┃%.2Lf┃ %.19Lf┃ %.19Lf┃ %2d┃\n",x,Summ,(1+x)\*expl(-x),n);

}

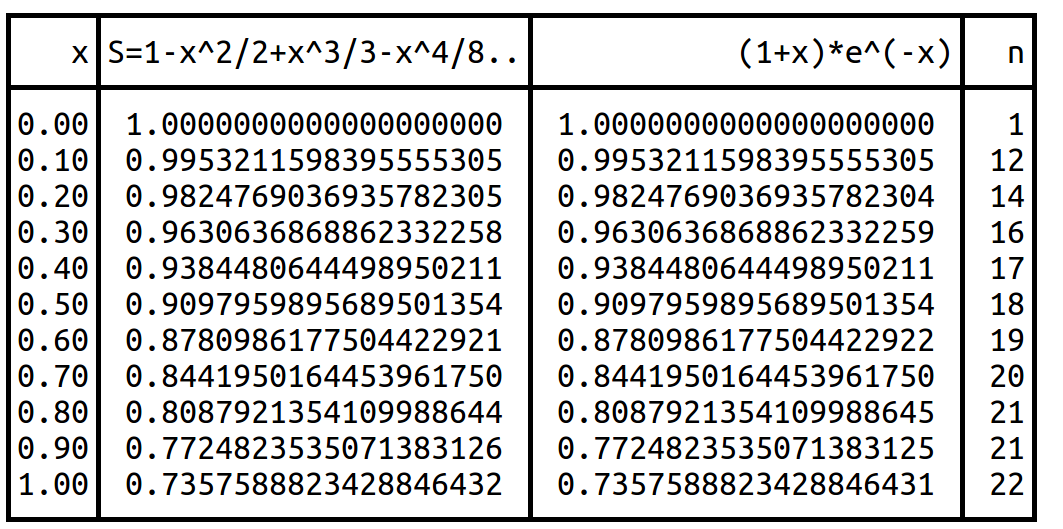
printf("┗━━━━┻━━━━━━━━━━━━━━━━━━━━━━━┻━━━━━━━━━━━━━━━━━━━━┻━━━┛\n");

}

daniel@daniel:~/$ gcc 3kp22.c -lm -o 3kp22.helloworld && ./3kp22.helloworld

Введите число разбиений отрезка: 10

Машинное эпсилон: 0.0000000000000000000542101086242752217003726400434970855712890625



**Заключение**

Мы научились составлять программу на Си, которая печатает таблицу значений элементарной функции, вычисленной двумя способами: по формуле Тейлора и с помощью встроенных функций языка программирования.

Составлена программа на языке Си, вычисляющая значение функции двумя способами. Значения совпадают на первых 18-19 знаках после запятой.