



- [Главная](#)

- [История](#) ▾

- [Домеханический этап](#) ▸

- [Зарождение счета](#)
 - [Бирки](#)
 - [Китайские счетные палочки](#)
 - [Узелковая письменность](#)
 - [Абак](#)
 - [Китайская счетная доска](#)
 - [Суаньпань](#)
 - [Соробан](#)
 - [Счет на линиях](#)
 - [Счет костями](#)
 - [Дощаный счет](#)
 - [Счеты](#)
 - [Палочки Непера](#)
 - [Логарифмы](#)
 - [Логарифмическая линейка](#)

- [Механический этап](#) ▸

- [Счетная машина Леонардо](#)
 - [Вычисляющие часы](#)
 - [Паскалин](#)
 - [Счетная машина Морленда](#)
 - [Калькулятор Лейбница](#)
 - [Рабдологический абак](#)
 - [Арифмометр Полени](#)
 - [Машина Перейры](#)
 - [Счетная машина Якобсона](#)

- [Первое поколение ЭВМ](#)

- [Второе поколение ЭВМ](#)

- [Третье поколение ЭВМ](#)

- [Персональные ЭВМ](#)

- [Устройство ПК](#) ▾

- [Материнская плата](#)

- [Процессор](#) ▸

- [Устройство процессора](#)
 - [Хронология ЦП Intel](#)

- [Оперативная память](#) ▸

- [Типы ОЗУ](#)
 - [Динамическая оперативная память](#)
 - [Статическая оперативная память](#)
 - [Магниторезистивная оперативная память](#)

- [Постоянная память](#)

- [Периферия](#)

- [ВС и Сети](#) ▾

- [Надежность ВС](#) ▸

- [Надежность](#)
 - [Повышение надежности](#)
 - [Резервирование оборудования](#)
 - [Контроль исправности](#)
 - [Высоконадежные ВС](#)
 - [Программное резервирование](#)
 - [Избыточное кодирование](#)
 - [Контрольная сумма CRC](#)
 - [Коды Хемминга](#)

- [Бортовые ВС](#)

- [Сети](#)

- [Разработка ПО](#) ▾

- [Системы контроля версий](#) ▸

- [Введение в СКВ](#)
 - [Обзор СКВ](#)
 - [Начинаем работать с СКВ GIT](#)

- [Виртуальные ЭВМ](#) ▸

- [Виртуальные машины](#)

- [Поиск](#)

- [Содержание](#)

- [Технология виртуализации](#)
- [Начинаем работать с VMware](#)
- [Начинаем работать с VirtualBox](#)
- [Язык Си](#)
- [Стиль Си](#)
- [Полезное ▾](#)
 - [Бесплатное ПО первой необходимости](#)
 - [Заставки на рабочий стол ▸](#)
 - [Абстракция](#)
 - [Архитектура](#)
 - [Природа](#)

[Главная](#) ▸ [Разработка ПО](#) ▸ [функции языка Си](#)

Описание функций языка Си

" F " [A](#)[B](#)[C](#)[D](#)[E](#)[F](#)[G](#)[H](#)[I](#)[J](#)[K](#)[L](#)[M](#)[N](#)[O](#)[P](#)[Q](#)[R](#)[S](#)[T](#)[U](#)[V](#)[W](#)[X](#)[Y](#)[Z](#)

[fabs,fabsf,fabsl](#) ◀ **fmod, fmodf, fmodl – вычисление остатка от деления.**

[fclose](#) ◀

Синтаксис:

[fcloseall](#) ◀ `#include < math.h >`

[fdim,fdimf,fdiml](#) ◀ `double fmod (double x, double y);`

`float fmodf (float x, float y);`

[fdopen](#) ◀ `long double fmodl (long double x, long double y);`

[fegetround](#) ◀ **Аргументы:**

[feof](#) ◀ `x – делимое.`

`y – делитель.`

[ferror](#) ◀

Возвращаемое значение:

[fesetround](#) ◀ `Остаток от деления x на y.`

[fgetc](#) ◀ `Если y= 0, то возвращаемому значению присваивается nan, а переменной errno будет присвоен код ошибки EDOM.`

[fgets](#) ◀

Описание:

[fileno](#) ◀ `Функция рассчитывает остаток от деления x на y по формуле x-n*y, где n – это результат деления x на y округленный в сторону нуля.`

[finite,finitef,finitel](#) ◀

[fflush](#) ◀ `При этом аргументы и возвращаемое значение функции fmod задаются числами с плавающей точкой двойной точности (тип double, точность не менее десяти значащих десятичных цифр, разрядность – 64).`

[floor,floorf,floorl](#) ◀

[fma,fmaf,fmal](#) ◀ `Аргументы и возвращаемое значение функции fmodf задаются числами с плавающей точкой (тип float, точность не менее шести значащих десятичных цифр, разрядность – 32).`

[fmax,fmaxf,fmaxl](#) ◀

[fmin,fminf,fminl](#) ◀ `Аргументы и возвращаемое значение функции fmodl задаются числами с плавающей точкой повышенной точности (тип long double, точность не менее десяти значащих десятичных цифр, разрядность – 80).`

[fmod,fmodf,fmodl](#) ◀

[fopen](#) ◀

Пример:

[fpclassify](#) ◀ `В примере рассчитывается остаток от деления 456.876 на 32.957 с помощью функций fmod, fmodf и fmodl, и результат выводится на консоль. Обратите внимание на точность полученных результатов. При расчете остатка от деления с помощью функции fmodf, будет самая маленькая точность, а с помощью функции fmodl – самая большая.`

[fputc](#) ◀

[fputs](#) ◀

[freopen](#) ◀

[frexp,frexpf,frexpl](#) ◀

[fseek](#) ◀

[fseeko](#) ◀

[ftell](#) ◀

[ftello](#) ◀

```
#include < stdio.h > //Для printf
#include < math.h > //Для fmod, fmodf, fmodl

int main (void)
{
    //Вывод значения аргумента
    printf ("Делимое: 456.876, делитель: 32.957\n");
    //Расчет остатка от деления с помощью функцией fmodf и вывод результата
    printf ("fmodf : %.20f\n", fmodf (456.876, 32.957) );
    //Расчет остатка от деления с помощью функцией fmod и вывод результата
    printf ("fmod : %.20f\n", fmod (456.876, 32.957) );
    //Расчет остатка от деления с помощью функцией fmodl и вывод результата
    printf ("fmodl : %.20Lf\n", fmodl (456.876, 32.957) );

    return 0;
}
```

Результат:

Делимое: 456.876, делитель: 32.957
fmodf : 28.43499755859375000000
fmod : 28.434999999999996674660
fmodl : 28.434999999999996674660

Смотри так же:

[fmod, fmodf, fmodl](#)
[remainder, remainderf, remainderl](#)
[drem, dremf, drem](#)
[remquo, remquoof, remquo](#)

