

## Раздел «Язык Си» . Coffeelf :

- Условные операторы
  - Условие
    - Почему равно ==, а присвоить =
  - if - вычислим модуль числа
  - if else - чет/нечет
  - else if - положительное, отрицательное, ноль
  - Когда что использовать?
  - Краткое содержание

Предупреждение:

## Условные операторы

**Задача:** Через реку можно переплыть на лодке, которая вмещает  $K$  пассажиров. На берегу стоит  $N$  человек и хочет переправиться на другой берег. Сколько нужно сделать рейсов, чтобы переправить на другой берег всех людей? 1 рейс – это когда лодка плавает туда и обратно.

Чтобы решить задачу, можно придумать математическую формулу.

Придумайте формулу. Проверьте для разных значений  $N$  и  $K$ .

Проще писать программу так, как думает обычный человек (не математик) :

- вычислим количество рейсов лодки, когда она полная
- если на берегу кто-то остался
- то добавим +1 рейс

```
#include <stdio.h>

int main()
{
    int n,           // пассажиров на берегу
        k,           // пассажиров за один рейс может перевезти лодка
        reisov,      // количество рейсов
        ostalos;     // в неполной лодке

    scanf("%d%d", &n, &k); // читаем n и k

    reisov = n/k;      // количество рейсов, когда лодка полная
    ostalos = n%k;     // сколько человек осталось после этого на берегу

    if (ostalos > 0) { // если на берегу остались люди
        reisov += 1;  // добавить еще 1 рейс с неполной лодкой
    }

    printf ("%d\n", reisov); // напечатать количество рейсов
    return 0;
}
```

Как писать условный оператор if:

```
if (условие)
    команды_ДА
```

- 💡 Запомните:
  - if – ключевое слово
  - условие пишем в круглых скобках ( )
  - команды пишем с **отступом**; с одинаковым отступом; используйте для отступа табуляцию.

Поиск

Поиск

Раздел «Язык Си»

Главная  
Зачем учить C?  
Определения

Инструменты:

Поиск  
Изменения  
Index  
Статистика

Разделы

Информация  
Алгоритмы  
Язык Си  
Язык Ruby  
Язык  
Ассемблера  
E! Judge  
Парадигмы  
Образование  
Сети  
Objective C

Login&gt;&gt;

- Если команд несколько, то нужно их объединить в **блочный оператор**, то есть написать внутри { }
- **Одну** команду можно писать БЕЗ { }
- Лучше сначала ВСЕГДА ставить { } (пока не перестанете их забывать)

## Условие

Условие может быть либо истина (ДА), либо ложь (НЕТ).

💡 0 – это ложь, все остальное – истина.

В условии используются операторы сравнения.

Python	Математика	Русский
<code>a == b</code>	$a = b$	а равно b
<code>a != b</code>	$a \neq b$	а не равно b
<code>a &lt; b</code>	$a < b$	а меньше b
<code>a &lt;= b</code>	$a \leq b$	а меньше или равно b (а не больше b)
<code>a &gt; b</code>	$a > b$	а больше b
<code>a &gt;= b</code>	$a \geq b$	а больше или равно b (а не меньше b)

⚠️ Нельзя внутри операторов сравнения ставить пробел

⚠️ Правильно `<=` (меньше или равно), а не `=<` Как говорим, так и пишем.

## Почему равно `==`, а присвоить `=`

⚠️ `a=2` – присвоить, `a==2` – сравнить

`a = 2` или `a == 2`

`a = 2` – это "а присвоить 2".

Операторы для "сравнить" и "присвоить" должны быть разными.

В программе чаще используют "присвоить", поэтому "присвоить" – это один знак = "равно?" – это два знака `==` подряд.

## if – вычислим модуль числа

Надо:  $x = |x|$ .

Решение: Если число отрицательное, возьмем  $-x$ .

(Никаких "иначе", "если число положительное" и т.п.)

```
if (x < 0)
    x = -x;
```

## if else – чет/нечет

Надо: Написать четное число (even) или нечетное (odd)

Решение: Проверим, равен ли 0 остаток от деления числа на 2.

Если равен,

то напечатаем EVEN

иначе

напечатаем ODD

```
if (x % 2 == 0)        // если остаток от деления на 2 равен 0
    printf ("EVEN");
else                  // иначе
    printf ("ODD");
```

## else if – положительное, отрицательное, ноль

Надо: Написать положительное число, отрицательное или ноль.

```
Решение: Если число > 0
    напечатать 'положительное'
иначе если число < 0
    напечатать 'отрицательное'
иначе (остался ноль)
    напечатать 'ноль'
```

Можно написать, отступая внутри каждого if и else еще на одну табуляцию:

```
if (x > 0)                // если больше 0
    printf ("положительное");
else                     // иначе меньше или равно 0
    if (x < 0)           // меньше 0
        printf ("отрицательное");
    else                // равно 0
        printf ("ноль");
```

Можно расставить отступы по-другому, чтобы подчеркнуть, что это три разных варианта одной и той же характеристики. Для компилятора это будет точно такой же код:

```
if (x > 0)                // больше 0
    printf ("положительное");
else if (x < 0)           // меньше 0
    printf ("отрицательное");
else                     // равно 0
    printf ("ноль");
```

## Когда что использовать?

- **if – один случай** (иначе ничего не надо делать):
  - если есть вода, я пью
  - если начнется дождь, надо открыть зонт
  - если на улице холодно, надо одеть шапку
- **if / else – или одно или другое** – 2 взаимоисключающих случая (одновременно быть не могут)
  - если на улице дождь, надо надеть резиновые сапоги, иначе надеть сандалии (нельзя сразу надеть и сапоги, и сандалии)
  - если есть кофе, я выпью кофе, иначе я выпью чай (я не буду пить чай и кофе вместе)
- **if / else if (несколько раз) / else – несколько взаимоисключающих случаев**
  - если есть кофе, я выпью кофе, иначе если есть чай, я выпью чай, иначе если есть сок, я выпью сок, иначе я выпью воду
    - **else части может не быть**
- **if / if / if (много раз)** – много раз делаем выбор, **один выбор не исключает другого.**
  - если есть мясо, я ем мясо; если есть картошка, я ем картошку; если есть чай, я пью чай

Например, напечатаем все характеристики числа:

- большое (если оно больше 10000000)
- четное или нечетное
- положительное, отрицательное или ноль

```
// характеристика: большой
if (x > 10000000) {
    printf ("большое ");
}

// характеристика: четное / нечетное
if (x % 2 == 0) {
    printf ("четное ");
}
else {
    printf ("нечетное ");
}

// характеристика: положительное / отрицательное / ноль
```

```
if (x > 0) {  
    printf ("положительное ");  
}  
else if (x < 0) {  
    printf ("отрицательное ");  
}  
else {  
    printf ("ноль ");  
}
```

- Для числа 16 напечатает: четное положительное
- Для числа -16 напечатает: четное отрицательное
- Для числа 1000016 напечатает: большое четное, положительное
- Для числа 163 напечатает: нечетное, положительное
- Для числа 0 напечатает: четное ноль

## Краткое содержание

Блоки else и else if не обязательны

```
if (условие) {  
    командыДА;  
}
```

```
if (условие) {  
    командыДА;  
}  
else {  
    командыНЕТ;  
}
```

```
if (условие_1) {  
    командыДА_1;  
}  
else if (условие_2) {  
    командыДа_2;  
}  
...  
else if (условие_N) {  
    командыДа_N;  
}  
else {  
    команды_НЕТ;  
}
```

-- TatyanaDerbysheva - 01 Apr 2016

(с) Материалы раздела "Язык Си" публикуются под лицензией GNU Free Documentation License.