

Целые	Целые (cont)	Целые (cont)
Установка $n^{\text{го}}$ бита	<code>x (1 < <n)</code>	Different rightmost bit of numbers m & n
Выключение $n^{\text{го}}$ бита	<code>x & ~(1 <<n)</code>	<code>(n^m) & - (n^m) // returns 2^x where x is the position of the different bit (0 based)</code>
Инверсия $n^{\text{го}}$ бита	<code>x ^ (1 < <n)</code>	Common rightmost bit of numbers m & n
Округление вниз	<code>n >> 0</code> <code>5.7812 >> 0 // вернёт 5</code>	<code>~(n^m) & (n^m)+1 // returns 2^x where x is the position of the common bit (0 based)</code>
Умножение на 2	<code>1n << 1;</code>	
Деление на 2	<code>n >> 1;</code>	
Умножение на $m^{\text{ю}}$ степень двойки	<code>n << m;</code>	
Деление на $m^{\text{ю}}$ степень двойки	<code>n >> m;</code>	
Проверка на чётность (кратность 2)	<code>(n & 1) == 1;</code>	
Максимум из двух	<code>b & ((a-b) >> 31) a & (~(a-b) >> 31);</code>	
Минимум из двух	<code>a & ((a-b) >> 31) b & (~(a-b) >> 31);</code>	
Проверка на одинаковый знак	<code>(x ^ y) >= 0;</code>	
Смена знака	<code>i = ~i + 1; // or</code> <code>i = (i ^ -1) + 1; // i = -i</code>	
Вернёт 2^n	<code>1 << n;</code>	
Является ли число степенью 2	<code>n > 0 && !(n & (n - 1))</code>	
Остаток от деления на 2^n на m	<code>m & ((1 << n) - 1);</code>	
Среднее арифметическое	<code>(x + y) >> 1;</code> <code>((x ^ y) >> 1) + (x & y);</code>	
	Получить $m^{\text{ый}}$ бит из n (от младшего к старшему)	
	<code>(n >> (m-1)) & 1;</code>	
	Получить $m^{\text{ый}}$ бит из n (от старшего к младшему)	
	<code>n & ~(1 << (m-1));</code>	
	Проверить включен ли $n^{\text{ый}}$ бит	
	<code>`if (x & (1 < <n)) {</code> <code> n-th bit is set</code> <code>} else {</code> <code> n-th bit is not set</code> <code>}</code>	
	Выделение самого правого включенного бита	
	<code>x & (-x)</code>	
	Выделение самого правого выключенного бита	
	<code>~x & (x+1)</code>	
	Выделение правого включенного бита	
	<code>x (x+1)</code>	
	Выделение правого выключенного бита	
	<code>x & (x-1)</code>	
	Получение отрицательного значения	
	<code>~n + 1;</code> <code>(n ^ -1) + 1;</code>	
	if (x == a) x = b;	
	<code>x = a ^ b ^ x;</code>	
	if (x == b) x = a;	
	Поменять смежные биты	
	<code>((n & 10101010) >> 1) ((n & 01010101) << 1)</code>	

Округление до ближайшей степени двойки

unsigned int v; // работает только с 32 битными числами

```
v--;  
v |= v >> 1;  
v |= v >> 2;  
v |= v >> 4;  
v |= v >> 8;  
v |= v >> 16;  
v++;
```

Получение максимального целого

```
int maxInt = ~(1 << 31);  
int maxInt = (1 << 31) - 1;  
int maxInt = (1 << -1) - 1;  
int maxInt = -1u >> 1;
```

Получение минимального целого

```
int minInt = 1 << 31;  
int minInt = 1 << -1;
```

Получение максимального long

```
long maxLong = ((long)1 << 127) -
```

Остаток от деления

```
n & 0b1; // на 2  
n & 0b11; // на 4  
n & 0b111; // на 8
```

И так далее

Проверка равенства

```
(a^b) == 0; // a == b  
!(a^b) // использовать внутри if()
```

Обмен значениями

```
//version 1  
a ^= b;  
b ^= a;  
a ^= b;
```

```
//version 2  
a = a ^ b ^ (b = a)
```

Получение абсолютного значения

```
//version 1  
x < 0 ? -x : x;
```

```
//version 2  
(x ^ (x >> 31)) - (x >> 31);
```



By **Arest** (arest)
cheatography.com/arest/

Published 24th November, 2021.
Last updated 24th November, 2021.
Page 1 of 3.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Строки

Конвертировать в нижний регистр`

(x | '')

Пример: ('a' | ' ') => 'a' ; ('A' | ' ') => 'a'

Конвертировать в верхний регистр

(x & '_')

Пример: ('a' & '_') => 'A' ; ('A' & '_') => 'A'

Инvertировать регистр

(x ^ '')

Пример: ('a' ^ ' ') => 'A' ; ('A' ^ ' ') => 'a'

Позиция буквы в алфавите (англ)

(x & "\x1F")

Пример: ('a' & " \x1 F") => 1 ; ('B' & " \x1 F") => 2

Позиция большой буквы в алфавите (англ)

(x & '?') или (x ^ '@')

Пример: ('C' & '?') => 3 ; ('Z' ^ '@') => 6

Позиция строчной буквы в алфавите (англ)

(x ^ '')

Пример: ('d' ^ ' ') => 4 ; ('x' ^ ' ') => 24

Десятичные дроби

Примечание: хаки с float могут не работать на Ардуино! Разбить float в массив бит (unsigned uint32_t)

```
#include <stdint.h>
typedef union {float flt; uint32_t bits} lens_t;
uint32_t f2i(float x) {
    return ((lens_t) { .flt = x }).bits;
}
```

Вернуть массив бит обратно в float

```
float i2f(uint32_t x) {
    return ((lens_t) { .bits = x }).flt;
}
```

Быстрый обратный квадратный корень

```
return i2f(0x 5f3 759df - f2i(x) / 2
);
```

Быстрый n^{ый} корень из целого числа

Десятичные дроби (cont)

```
float root(float x, int n) {
#define MAN_MASK 0x7ffff
#define EXP_MASK 0x7f800000
#define EXP_BIAS 0x3f800000
    uint32_t bits = f2i(x);
    uint32_t man = bits & MAN_MASK;
    uint32_t exp = (bits & EXP_MASK) - EXP_BIAS;
    return i2f((man + man / n) | ((EXP_BIAS + exp / n) & EXP_MASK));
}
```

Быстрая степень

```
return i2f((1 - exp) (0x3f800000 - 0x5c416) + f2i(x) exp)
```

Быстрый натуральный логарифм

```
#define EPSILON 1.1920928955078-125e-07
#define LOG2 0.6931471805599453
return (f2i(x) - (0x3f800000 - 0x66774)) EPSILON LOG2 => 2
```

Быстрая экспонента

```
return i2f(0x3f800000 + (uint32_t)(x * (0x800000 + 0x38aa22)))
```

Примечание: хаки с float могут не работать на Ардуино! Разбить float в массив бит (unsigned uint32_t)

Другое

Быстрая конвертация цвета R5G5B5 в R8G8B8

```
R8 = (R5 << 3) | (R5 >> 2)
```

```
G8 = (R5 << 3) | (R5 >> 2)
```

```
B8 = (R5 << 3) | (R5 >> 2)
```

Приоритет операций

Чтобы не плодить скобки, нужно знать приоритет операций

:: ++ - () [] . - >

Приоритет операций (cont)

++	--	+	-	!	~	(t
&	sizeof	new,	delete,	.*	-	*
		new[]	delete[]		>*	
%	+	-	<<	>>	<	<
>=	==	!=	&	^		&
?:	=	+=	-=	*=	/=	%
>>=	&=	^=	=			

Чтобы не плодить скобки, нужно знать приоритет операций

Приоритет операций

Чтобы не плодить скобки, нужно знать приоритет операций. В C++ он такой:

Приоритет операций

```
:: ++ -- () [] . -> ++ -- + - ! ~ (type) * & sizeof
new, new[] delete, delete[] .* -> ** / % + - <<
>> < <= > >= == != & ^ | && || ?: = += -= *=
/= %= <<= >>= &= ^= |= :: ++ -- () [] . -> ++ --
+ - ! ~ (type) * & sizeof new, new[] delete,
delete[] .* -> ** / % + - << >> < <= > >= ==
!= & ^ | && || ?: = += -= *= /= %= <<= >>=
&= ^= |=
```



By **Arest** (arest)
cheatography.com/arest/

Published 24th November, 2021.
Last updated 24th November, 2021.
Page 2 of 3.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>