

Целые	
Установка $n^{\text{го}}$ бита	$x   (1 \ll n)$
Выключение $n^{\text{го}}$ бита	$x \& \sim(1 \ll n)$
Инверсия $n^{\text{го}}$ бита	$x \wedge (1 \ll n)$
Округление вниз	$n \gg 0$ $5.7812 \gg 0$ // вернёт 5
Умножение на 2	$1n \ll 1;$
Деление на 2	$n \gg 1;$
Умножение на $m^{\text{ую}}$ степень двойки	$n \ll m;$
Деление на $m^{\text{ую}}$ степень двойки	$n \gg m;$
Проверка на чётность (кратность 2)	$(n \& 1) == 1;$
Максимум из двух	$b \& ((a-b) \gg 31)   a \& (\sim(a-b) \gg 31);$
Минимум из двух	$a \& ((a-b) \gg 31)   b \& (\sim(a-b) \gg 31);$
Проверка на одинаковый знак	$(x \wedge y) \gg 0;$
Смена знака	$i = \sim i + 1; // \text{or}$ $i = (i \wedge -1) + 1;$ $// i = -i$

Целые (cont)	
Вернёт $2^n$	$1 \ll n;$
Является ли число степенью 2	$n > 0 \&\& !(n \& (n - 1));$
Остаток от деления на $2^n$ на $m$	$m \& ((1 \ll n) - 1);$
Среднее арифметическое	$(x + y) \gg 1;$ $((x \wedge y) \gg 1) + (x \& y);$
Получить $m^{\text{ый}}$ бит из $n$ (от младшего к старшему)	$(n \gg (m - 1)) \& 1;$
Получить $m^{\text{ый}}$ бит из $n$ (от старшего к младшему)	$n \& \sim(1 \ll (m - 1));$
Проверить включен ли $n^{\text{ый}}$ бит	<code>if (x &amp; (1&lt;&lt;n)) {     n-th bit is set } else {     n-th bit is not set }</code>
Выделение самого правого включенного бита	$x \& (-x)$
Выделение самого правого выключенного бита	$\sim x \& (x+1)$

Целые (cont)	
Выделение правого включенного бита	$x   (x+1)$
Выделение правого выключенного бита	$x \& (x-1)$
$n + 1$	$\sim \sim n$
$n - 1$	$\sim \sim n$
Получение отрицательного значения	$\sim n + 1;$ $(n \wedge -1) + 1;$
if (x == a) x = b;	$x = a \wedge b \wedge x;$
if (x == b) x = a;	
Поменять смежные биты	$((n \& 10101010) \gg 1)   ((n \& 01010101) \ll 1)$
Different rightmost bit of numbers $m$ & $n$	$(n \wedge m) \& \sim(n \wedge m)$ // returns $2^x$ where $x$ is the position of the different bit (0 based)
Common rightmost bit of numbers $m$ & $n$	$\sim(n \wedge m) \& (n \wedge m) + 1$ // returns $2^x$ where $x$ is the position of the common bit (0 based)

Целые	
Округление до ближайшей степени двойки	



By **Arest** (arest)  
[cheatography.com/arest/](https://cheatography.com/arest/)

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

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>

### Целые (cont)

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) - 1;
```

### 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);
```

### Строки

#### Конвертировать в нижний регистр`

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

#### Конвертировать в верхний регистр`

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

#### Инвертировать регистр`

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

#### Позиция буквы в алфавите (англ)

```
(x & "x1F")
Пример: ('a' & "x1F") => 1 ;
('B' & "x1F") => 2
```

#### Позиция большой буквы в алфавите (англ)

```
(x & '?') или (x ^ '@')
Пример: ('C' & '?') => 3 ; ('Z' ^ '@') => 26
```

#### 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

### Десятичные дроби (cont)

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

#### Быстрый обратный квадратный корень

```
return i2f(0x5f3759df - f2i(x) / 2);
```

#### Быстрый n<sup>ый</sup> корень из целого числа

```
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
```

#### 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)



By **Arest** (arest)  
[cheatography.com/arest/](https://cheatography.com/arest/)

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

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>

### 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

++	--	+	-	!	~	(type)	*
&	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/](https://cheatography.com/arest/)

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

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>