

Битовые

Логическая операция И (AND)

Обозначение AND: &

| a | b | a & b |
|----------|----------|------------------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Логическая операция ИЛИ (OR)

Обозначение OR: |

| a | b | a b |
|---|---|-------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Логическая операция исключающее ИЛИ (XOR).

Обозначение XOR: ^

| a | b | $A \wedge b$ |
|---|---|--------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Логическая операция НЕ

Обозначение NOT: \sim

| a | $\sim a$ |
|----------|----------------------------|
| 0 | 1 |
| 1 | 0 |

Таблица истинности для C++

| <u>a</u> | <u>b</u> | <u>a && b</u> | <u>a b</u> | <u>!a</u> | <u>!b</u> |
|--------------|--------------|-----------------------|---------------|--------------|--------------|
| <u>false</u> | <u>false</u> | <u>false</u> | <u>false</u> | <u>true</u> | <u>true</u> |
| <u>false</u> | <u>true</u> | <u>false</u> | <u>true</u> | <u>true</u> | <u>false</u> |
| <u>true</u> | <u>false</u> | <u>false</u> | <u>true</u> | <u>false</u> | <u>true</u> |
| <u>true</u> | <u>true</u> | <u>true</u> | <u>true</u> | <u>false</u> | <u>false</u> |

Логическая операция в сочетании с логическим выражением

| | | |
|------------------------|------------------------|--------------------|
| bool res; | // операция (OR) | // операция ! |
| int a, b; | a = 0; | (логическое |
| | b = 0; | "НЕТ") |
| // операция && (AND) | res = a b; // False | a = 0; |
| a = 8; b = 5; | b = 7; | res = !a; // True |
| res = a && b; // True | res = a b; // True | a = 15; |
| a = 0; | | res = !a; // False |
| res = a && b; // False | | |

XOR (исключающее «ИЛИ»): $\text{res} = (x \parallel y) \&\& !(x \&\& y)$

Логические операции в условных выражениях

```
cin >> value ;
```

```
if (value > 10 && value < 20)
```

```
    cout << "Your value is between 10 and 20" << endl;
```

```
else
```

```
    cout << "Your value is not between 10 and 20" << endl;
```

Как и с логическим ИЛИ, мы можем комбинировать сразу несколько условий И:

```
if (value > 10 && value < 20 && value != 16)
```

Логические выражения можно преобразовывать в соответствии с законами алгебры логики

| | | | |
|------------|-----|---|----------|
| Конъюнкция | and | & | \wedge |
| Дизъюнкция | or | | \vee |
| Отрицание | not | ! | \neg |

Законы рефлексивности

$$a \wedge a = a$$

$$a \vee a = a$$

Законы коммутативности

$$a \wedge b = b \wedge a$$

$$a \vee b = b \vee a$$

Законы ассоциативности

$$(a \wedge b) \wedge c = a \wedge (b \wedge c)$$

$$(a \vee b) \vee c = a \vee (b \vee c)$$

Законы дистрибутивности

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

Закон отрицания отрицания

$$\neg(\neg a) = a$$

Законы де Моргана

$$\neg(a \wedge b) = \neg a \vee \neg b$$

$$\neg(a \vee b) = \neg a \wedge \neg b$$

Законы поглощения

$$a \wedge (a \vee b) = a$$

$$a \vee (a \wedge b) = a$$

```

#include <iostream>

using namespace std;

int main(){

    int a,b;

    cout << "input a: "; cin >> a;

    cout << "input b: "; cin >> b;

    //Применение логического ИЛИ

    if (a == 100 || b == 100){

        cout << "a == 100 || b == 100"<< '\n' }

    else {        cout << "a != 100 && b != 100"<< '\n';        }

    //Применение логического И

    if (a == 100 && b == 100){

        cout << "a == 100 && b == 100"<< '\n';        }

    else {        cout << "(a, b) != 100"<< '\n';        }

    return 0;}

```

```

input a: 12
input b: 100
a == 100 || b == 100
(a, b) != 100

```

```

input a: 231
input b: 111
a != 100 && b != 100
(a, b) != 100

```