AND	&&
OR	||
NOT	!

AND

a	b	a && b
false	false	false
false	true	false
true	false	false
true	true	true

OR

a	b	a || b
false	false	false
false	true	true
true	false	true
true	true	true

NOT

a	!a
true	false
false	true

```cpp
bool a {true};
bool b {false};
bool c {true};

std::cout << std::boolalpha;
std::cout << "a : " << a << std::endl;
std::cout << "b : " << b << std::endl;
std::cout << "c : " << c << std::endl;



//AND : Evaluates to true when all operands are true.
//      A single false operand will drag
//      the entire expression to evaluating false.

std::cout << std::endl;
std::cout << "Basic AND operations" << std::endl;

std::cout << " a && b : " << (a && b) << std::endl;
std::cout << " a && c : " << (a && c ) << std::endl;
std::cout << " a && b && c :" << (a && b && c) << std::endl;
```

```
//OR : Evaluates to true when at least one operand true.
//      A single true operand will push
//      the entire expression to evaluating true.
std::cout << std::endl;
std::cout << "Basic OR operations" << std::endl;
std::cout << " a || b : " << (a || b) << std::endl;
std::cout << " a || c : " << (a || c ) << std::endl;
std::cout << " a ||b || c :" << (a || b || c) << std::endl;
```

```
//OR : Evaluates to true when at least one operand true.
//      A single true operand will push
//      the entire expression to evaluating true.
std::cout << std::endl;
std::cout << "Basic OR operations" << std::endl;
std::cout << " a || b : " << (a || b) << std::endl;
std::cout << " a || c : " << (a || c ) << std::endl;
std::cout << " a ||b || c :" << (a || b || c) << std::endl;
```

Oh! Whenever I say a AND b or a AND b AND c, I really mean the OR operator. Sorry for that.

```cpp
//NOT : Negates whateve operand you put it with
std::cout << std::endl;
std::cout << "Basic NOT operations" << std::endl;

std::cout << "!a : " << !a << std::endl;
std::cout << "!b : " << !b << std::endl;
std::cout << "!c : " << !c << std::endl;
```

Combine logical operators in expression

```
std::cout << "!(a &&b) || c : " << (!(a &&b) || c) << std::endl;
```

Logical and Relational combined in expressions

```
int d{45};
int e{20};
int f{11};

std::cout << std::endl;
std::cout << "Relational and logic operations on integers" << std::endl;
std::cout << "d : " << d << std::endl;
std::cout << "e : " << e << std::endl;
std::cout << "f : " << f << std::endl;

std::cout << std::endl;

std::cout << "(d > e) && (d > f) : " << ((d > e) && (d > f)) << std::endl;
std::cout << "(d==e) || (e <= f ) : " << ((d==e) || (e <= f ) ) << std::endl;
std::cout << "(d < e) || (d > f) : " << ((d < e) || (d > f)) << std::endl;
std::cout << "(f > e) || (d < f) : " << ((f > e) || (d < f)) << std::endl;
std::cout << "(d > f) && (f <= d) : " << ((d > f) && (f <= d)) << std::endl;
std::cout << "(d > e) && (d <= f) : " << ((d > e) && (d <= f)) << std::endl;
std::cout << "(! a) && (d == e) : " << ((! a) && (d == e)) << std::endl;
std::cout << "(! a) && (d == e) : " << ((! a) && (d == e)) << std::endl;
```