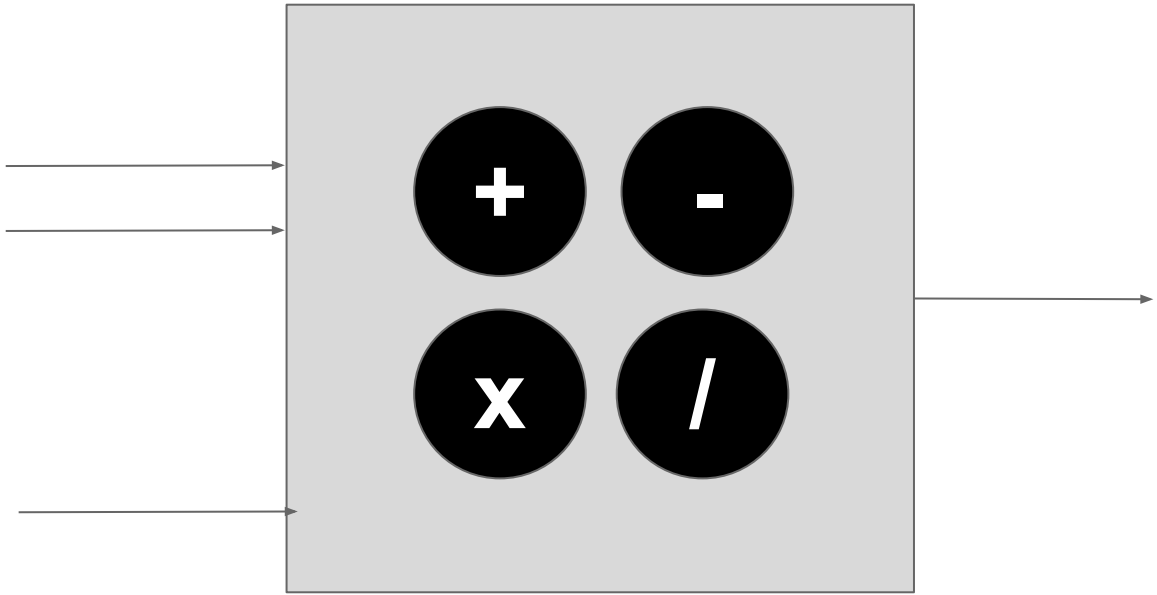
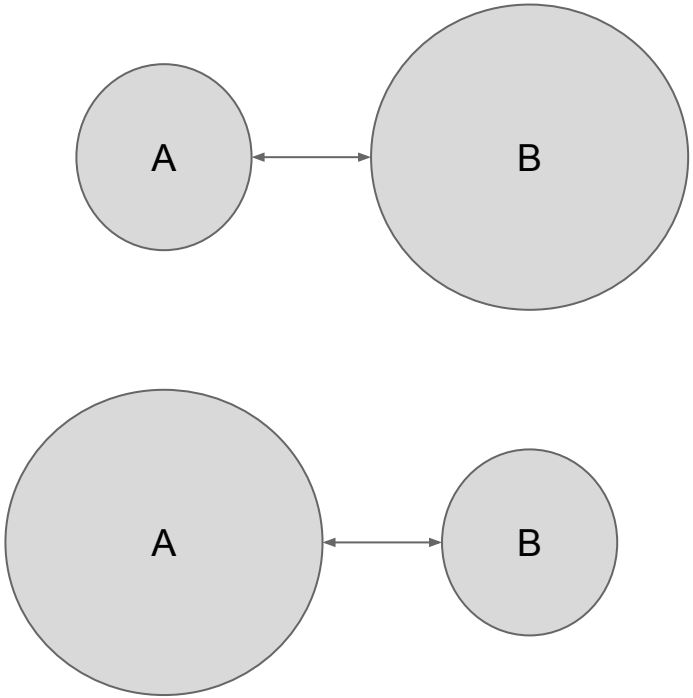
Conditional functions

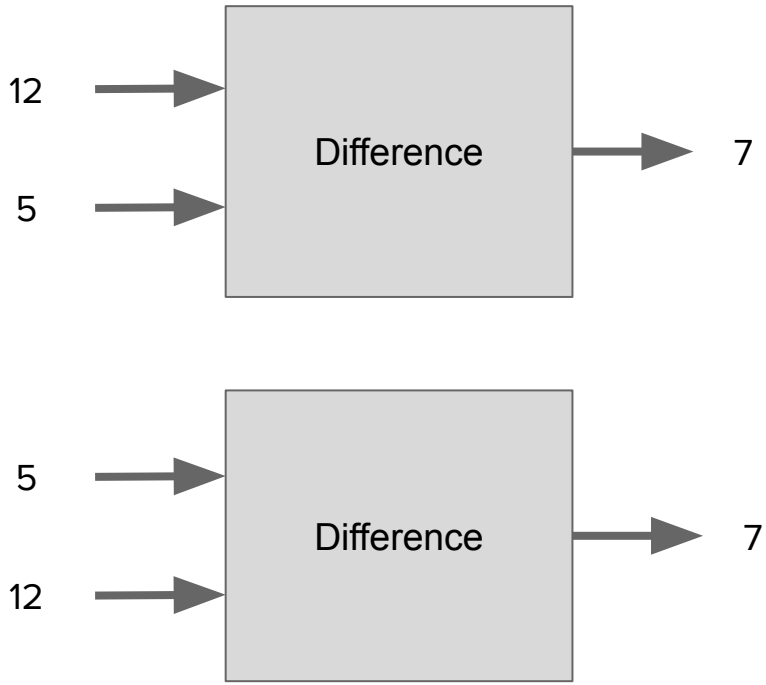
Functions so far...



Difference between two numbers



So difference:



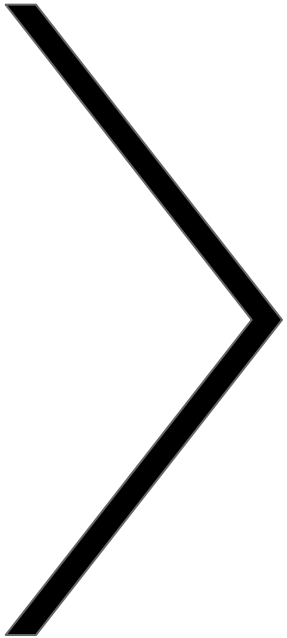
How do we solve this:

You will say it’s very easy:

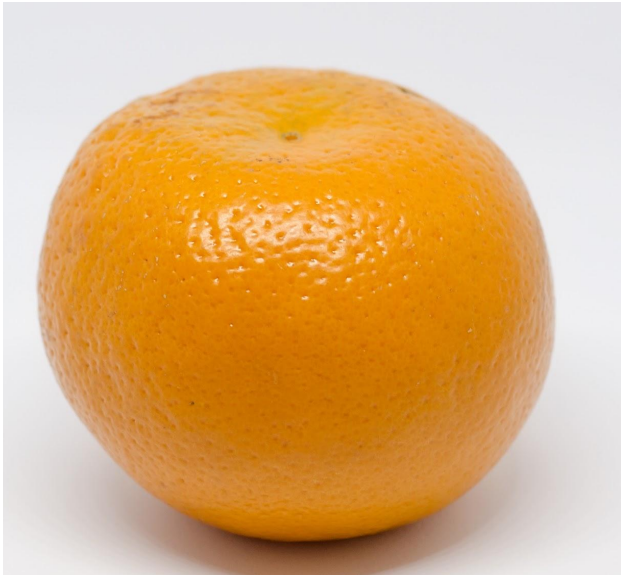
if firstInput > secondInput then output (firstInput - secondInput)

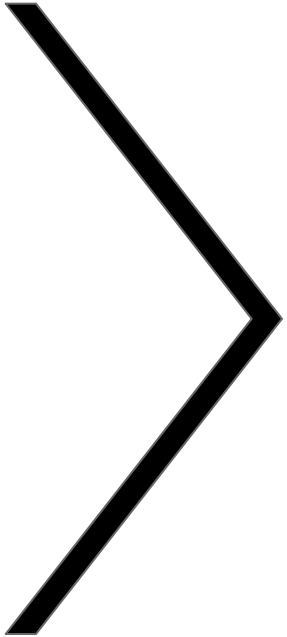
Otherwise output (secondInput - firstInput)

You made an assumption:



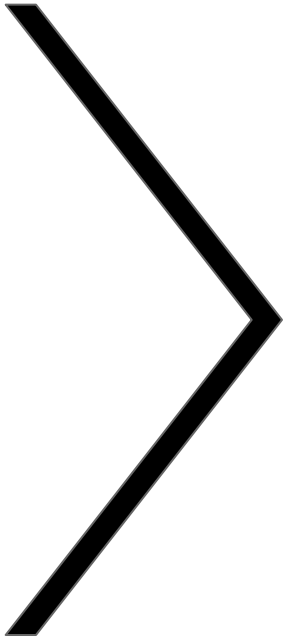
Can you do this comparison?





How about now?

Position (100, 20) on the screen



Position (20, 40) on the screen

So what makes something GREATER than something else

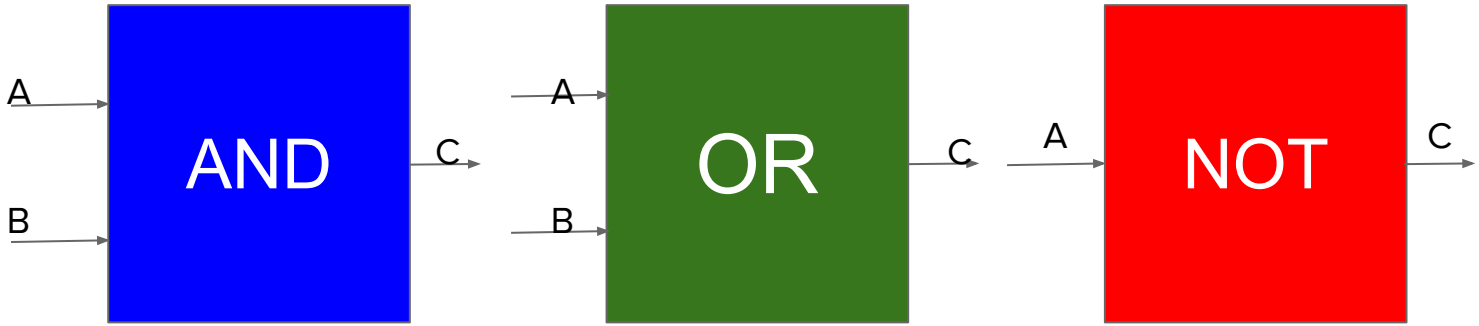
Numbers - Integers and Real - are obviously comparable.

2 > 1 ===> YES
11000 > 122 == > DEFINITELY
3.141592653589 > 3.1 ==> YEAH!!

But how is the computer comparing them?

123 > 50
---
1 > 0

Review : Gates



A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

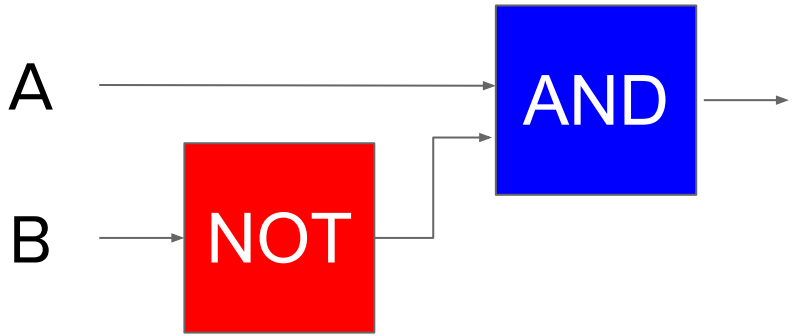
A	C
0	1
1	0

Comparing 1s and 0s

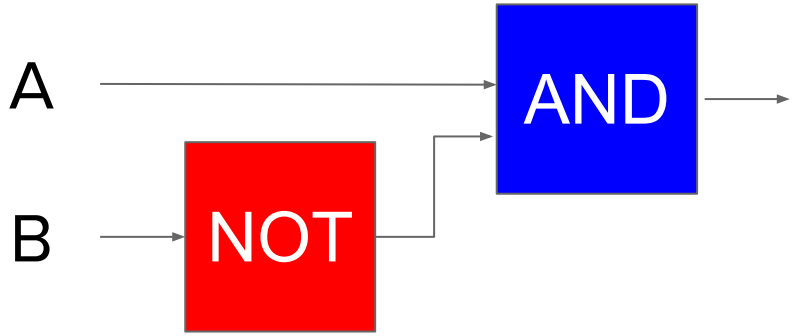
A	B	A > B
0	0	0
0	1	0
1	0	1
1	1	0

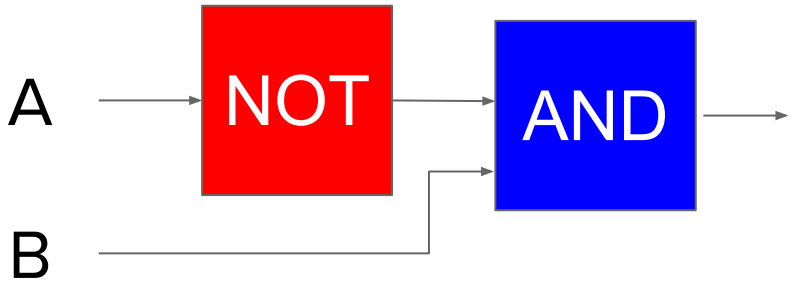
Comparing 1s and 0s

A	B	A > B
0	0	0
0	1	0
1	0	1
1	1	0



Comparing the other way around:





Representing numbers

$$123 = 3 \times 1 \quad + \quad 2 \times 10 \quad + \quad 1 \times 100$$

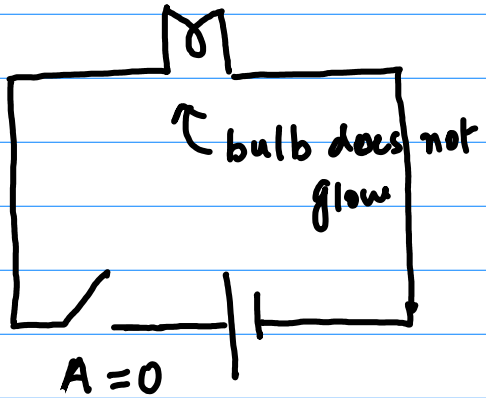
Representing numbers

$$123 = 1x2^0 + 1x2^1 + 0x2^2 + 1x2^3 + 1x2^4 + 1x2^5 + 1x2^6$$

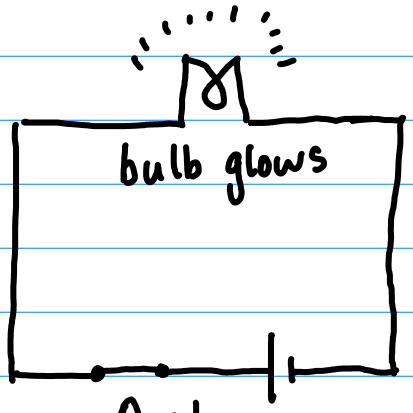
Reverse order

LOGIC GATES

① Consider a simple circuit :-

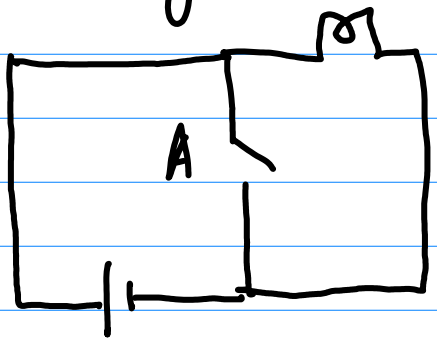


A = 0
(this is zero)



A = 1
(this is one)

② We can arrange the switch A differently



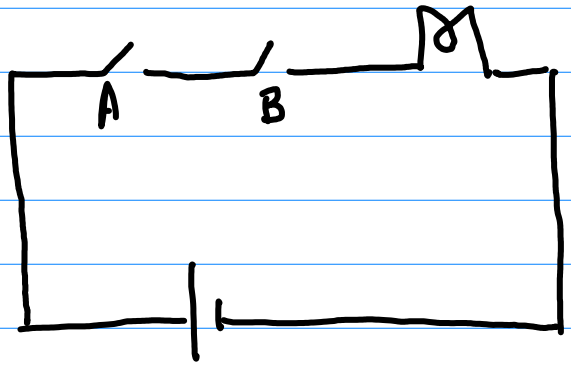
NOT

In this arrangement,
when A = 0, the bulb glows
when A = 1 the bulb does not glow.

The bulb is then NOT(A)

A	Bulb = not A
0	1
1	0

③ Let us consider next a circuit with two switches

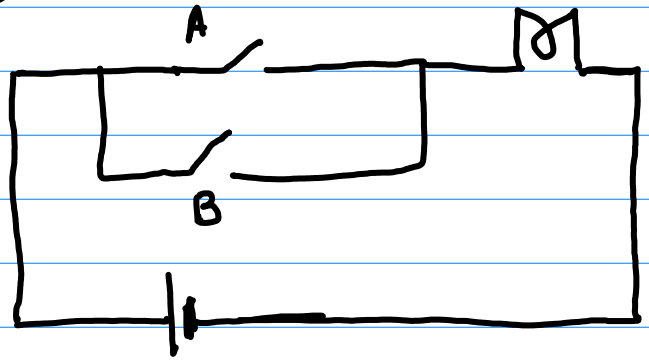


AND

Only if both A and B are 1 the bulb will glow...

A	B	Bulb = A and B
0	0	0
0	1	0
1	0	0
1	1	1

④ Finally let's consider this one →



OR

A	B	Bulb = A or B
0	0	0
0	1	1
1	0	1
1	1	1

If either switch A or B is 1, the bulb glows.

(5) Using AND, OR, NOT to do comparisons

A	B	A > B	A < B	A == B
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1