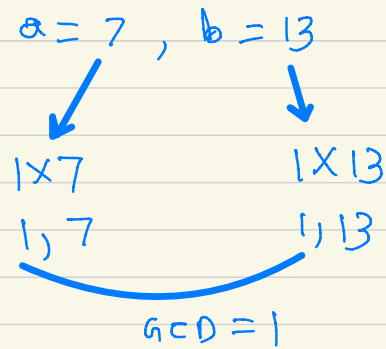
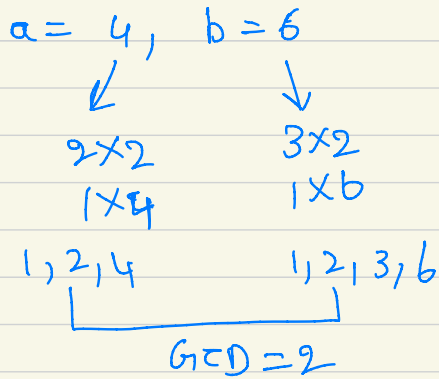


## Greatest common Divisor

GCD



## Naive solution

\*) **point to know:** we know that GCD of two numbers cannot be greater than minimum of 2 numbers given

Eg:  $a = 100$      $b = 200$

GCD = 100

As said previously the GCD of two numbers cannot exceed minimum of 2 numbers

$a = 10, b = 15$

min = 10  
min = 9  
min = 8  
min = 7  
min = 6  
min = 5

```
int min = Math.min(a, b); a=10 b=15
```

```
while (min > 0)
{
    if (a % min == 0 && b % min == 0)
    {
        break;
    }
    min--;
}
return min;
```

**Time complexity**

$O(\min(a, b))$

# Euclidean Algorithm

Basic Idea:

Let 'b' be smaller than 'a' then gcd can be written as

$$\text{gcd}(a, b) = \text{gcd}(a-b, b)$$

This approach uses repetitive subtraction

```
int gcd (int a, int b)
{
    while (a != b)
    {
        if (a > b)
            a = a - b;
        else
            b = b - a;
    }
    return a;
}
```

a = 12, b = 15

a = 12, b = 3

a = 9, b = 3

a = 6, b = 3

a = 3, b = 3

optimized Implementation

```
int gcd (int a, int b)
{
    if (b == 0)
        return a;
    else
        return (b, a % b);
}
```

gcd (12, 15)

↳ gcd (15, 12)

↳ gcd (12, 3)

↳ gcd (3, 0)