Q. Given two numbers a and b, find the greatest common divisor of a & b (greatest common divisor → gcd → hcf → highest common factor).

→ Ex → a = 28
b = 24

ans → **4**

$a = 28 \rightarrow 2 \times 2 \times 7$

$b = 24 \rightarrow 2 \times 2 \times 2 \times 3$

4 ans

What to do ??

gcd

we want to detect a no. that can divide both a and b.

There can be multiple numbers that can divide both a and b.

Among those multiple no. we need to get the biggest no.

a b

28 , 24

2   2

at any no. i $\rightarrow$ if (a % i == 0 &&
b % i == 0)

potential answer = ~~2~~ <u>4</u>

1) can 2 divide both a & b

28 % 2 == 0 && 24 % 2 == 0

2) Can 3 divide both a & b

28 % 3 == 0 && 24 % 3 == 0 $\rightarrow$ false

3) Can 4 divide both a & b

28 % 4 == 0 && 24 % 4 == 0 $\rightarrow$ <u>true</u>

Prime no.

7 , 11 → 1

Note → 1 will be our min ans.

inbuilt func^n

```
function gcd (a, b) {
    let ans = 1;
    for (let i = 2 ; i <= Math.min(a, b) ; i++) {
        if( a % i == 0   &&   b % i == 0) {
            ans = i;
        }
    }
    return ans;
}
```

Euclid's Algo

gcd → a, b

$x = 14$

↳ 2 & 7

are divisors

Say, we divide 33 with 5

33 → a, 5 → b

$$\begin{array}{r|l} 5 & 33 \;(6 \to q \\ & \underline{30} \\ & \underline{3} \to r \end{array}$$

to generalize,

$33 = 5 \times 6 + 3$

$a = b \times q + r$

say, we divide a with b, such that

q is the quotient & r is the remainder.

a/b $\qquad\qquad$ $a>b$

$\rightarrow$ $\boxed{a = b * q + r}$

$\boxed{a - bq = r}$ —— (1)

Now, let's say gcd of a and b is some no.

G.

if G is the gcd of a and b

then $a \% G == 0$ and $b \% G == 0$

Q. what is bq ??

bq is a multiple of b.

if G divides b ($b \% G == 0$) $\Rightarrow$

G divides bq $\rightarrow$ ($bq \% G == 0$)

multiple of b

$\boxed{a - bq} = \boxed{r}$

LHS

RHS

r is also going to be divisible by c.

a is divisible by c

bq is also divisible by c.

$a = 78$ $\qquad$ $b = 36$

$$36 \,)\, 78 \,(\, 2$$
$$72$$
$$\overline{6}$$

$r = 6$

$$78 = 36 \times 2 + 6$$

$$78 - 36 \times 2 = 6$$

$a - bq = r$

$G \rightarrow$ gcd of a & b

G is the gcd of a, b, r

$a > b > r$

→ 78 and 24

$r \Rightarrow 78 \% 24 \rightarrow 6$

whatever is the gcd of 24 and 6 will be the gcd of 78 & 24.

⟶ gcd 24 and 6

$r \Rightarrow 24 \% 6 \Rightarrow 0$

if x and y $\longrightarrow$ y

gcd x and y

$(x \% y == 0)$

Ex $\rightarrow$ 24 and 12 $\longrightarrow$ 12

$24 \% 12 == 0$

$$\gcd(a \quad , \quad b)$$

$$r = a \% b$$

$$\downarrow$$

$$a \% b$$

$$\gcd(b, r)$$

$$r_2 = b \% r$$

$$\gcd(r, r_2)$$

$$\vdots$$

$a = 105$ $\qquad$ $b = 36$

$r = a \% b \Rightarrow 33$

$a = 36$ $\qquad$ $b = 33$

$r_2 = a \% b \rightarrow \underline{\underline{3}}$

$a = 33$ $\qquad$ $b = 3$ $\qquad$ $\gcd(33, 3) = 3$

$r_3 = 33 \% 3 == 0$

Q Given a number x, Calculate the sum of digits of the no. x.

$x > 0$

Ex → 4136

ans → **14** ⟶ $(4+1+3+6)$

4136

if we need the sum of digits of the no, we first of all need to extract the digits out of the no.

4136 % 10 → 6

if we do $x \% 10$ → we get last digit of the no.

12683 % 10 → $\underline{\underline{3}}$

x ⤵

4136 % 10 → 6 Sum = 0 + 6

Math.floor(4136 / 10)

↳ Math.floor(413.6)

↳ $\underline{413}$

by doing Math.floor(x/10) we can eliminate the last <u>digit</u>.

$413 \% 10 \rightarrow 3$

$$\text{Sum} = 0 + 3 + 1 + 4$$

Math.floor(413/10) → 41

$41 \% 10 \rightarrow 1$

Math.floor(41/10) → 4

$4 \% 10 \rightarrow 4$

Math.floor(4/10) → 0

terminate

```
let Sum = 0;
while (x > 0) {

    let lastDigit = x % 10;
    Sum += lastDigit;
    x = Math.floor(x/10)

}
```

→ return Sum;

$x = 0$

Sum = 14

lastDigit = 4

Q:

0th 1st 2nd 4th 5th 7th 8th .....

0, 1, 1, 2, 3, 5, 8, 13, 21 .....

3rd 6th

The above series of numbers are called fibonacci <u>numbers</u>. Given a value n, write a code to print the first n fibonacci <u>numbers</u>.

<u>Ex</u> :- $n = 6$ →
0
1
1
2
3
5
8

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55

skip

starter values

apart from 0th & 1st fib, if we consider any fib number, then the no. is actually sum of the prev 2 fibonacci no.

$$f_i = f_{i-1} + f_{i-2}$$

$i \geq 2$

ith fibonacci

0
1
1
2
3
second last → 5
last → 8
13

element of seq → every time add the prev 2 fib no & print the result.

We can maintain 2 variables & keep a track of last & 2nd last fib.

We need to repeat n times.

```
function printfibonocci (n) {
    if (n==0) {
        console.log (0);
        return;
    }
    if (n >=1) {
        console.log (0);
        console.log (1);
    }
    let last =1;
    let secondlast = 0;
    for (let i=2; i<= n; i++)
            let ans = last + secondlast;
            console.log (ans);
            secondlast = last;
            last = ans;
    }
}
```