**6 kyu**

**Simple Fun #228: Greatest Common Prime Divisor**

3094% of 1635[myjinxin2015](https://www.codewars.com/users/myjinxin2015)

JavaScript

* [TRAIN AGAIN](https://www.codewars.com/kata/simple-fun-number-228-greatest-common-prime-divisor/train/javascript)
* [NEXT KATA](https://www.codewars.com/trainer/javascript)

Details

[Solutions](https://www.codewars.com/kata/simple-fun-number-228-greatest-common-prime-divisor/solutions/javascript)

[Discourse](https://www.codewars.com/kata/simple-fun-number-228-greatest-common-prime-divisor/discuss/javascript)

* Add to Collection
* |
* Share this kata:

**Task**

Define GCPD (Greatest Common Prime Divisor) as the largest prime number that divides both given positive integers. Your task is to find GCPD of the given integers a and b or -1 if it doesn't exist.

**Input/Output**

[input] integer a

2 ≤ a ≤ 1000

[input] integer b

2 ≤ b ≤ 1000

[output] an integer

GCPD of a and b or -1 if it doesn't exist.

**Example**

For a = 12 and b = 18, the output should be 3

For a = 12 and b = 13, the output should be -1

<https://www.codewars.com/kata/5906dcc359a2f71fe700002e/solutions/javascript>

* [Voile](https://www.codewars.com/users/Voile)
* **const primes = [];**
* **for(let n=2;n<1000;n++) if(primes.every(p=>n%p)) primes.push(n);**
* **primes.reverse();**
* **function greatestCommonPrimeDivisor(a, b) {**
* **return primes.find(p=>a%p===0&&b%p===0)||-1;**

**}**

* + - Best Practices1
    - Clever1
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5909308c9d03b9f9ba00055f&review_id=5906df3ca976efdac800002d)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5906df3ca976efdac800002d/groups/5909308c9d03b9f9ba00055f)
* [nachoMonllor](https://www.codewars.com/users/nachoMonllor)
* **function SieveOfEratosthenes(n)**
* **{**
* **// Create a boolean array "prime[0..n]" and initialize**
* **// all entries it as true. A value in prime[i] will**
* **// finally be false if i is Not a prime, else true.**
* **var prime = [];**
* **for (let i = 0; i < n; i++)**
* **prime.push(true);**
* **for (let p = 2; p \* p <= n; p++)**
* **{**
* **if (prime[p] == true)**
* **{**
* **// Update all multiples of p**
* **for (let i = p \* p; i <= n; i += p)**
* **prime[i] = false;**
* **}**
* **}**
* **var primos = [];**
* **// Print all prime numbers**
* **for (let i = 2; i <= n; i++)**
* **{**
* **if (prime[i] == true)**
* **{**
* **primos.push(i);**
* **}**
* **}**
* **return primos;**
* **}**
* **function greatestCommonPrimeDivisor(a, b) {**
* **//coding and coding..**
* **var lim = Math.min(a / 2, b / 2);**
* **var factors = SieveOfEratosthenes(lim);**
* **for(let i = factors.length - 1; i >= 0; i--)**
* **{**
* **if(a % factors[i] == 0 && b % factors[i] == 0)**
* **{**
* **return factors[i];**
* **}**
* **}**
* **return -1;**
* **}**
  + - Best Practices0
    - Clever0
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5c11eaa410e90fd97d000bb7&review_id=5906df3ca976efdac800002d)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5906df3ca976efdac800002d/groups/5c11eaa410e90fd97d000bb7)
* [monadius](https://www.codewars.com/users/monadius)
* **function greatestCommonPrimeDivisor(a, b) {**
* **const gcd = (a, b) => b ? gcd(b, a % b) : a;**
* **let d = gcd(a, b);**
* **let max = -1;**
* **for (let p = 2; p \* p <= d; p++) {**
* **while (d % p === 0) {**
* **d /= p;**
* **max = p;**
* **}**
* **}**
* **return d > 1 ? d : max;**

**}**

* + - Best Practices0
    - Clever0
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5bad6111b684d3ad04006385&review_id=5906df3ca976efdac800002d)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5906df3ca976efdac800002d/groups/5bad6111b684d3ad04006385)
* [Len512](https://www.codewars.com/users/Len512)
* **function greatestCommonPrimeDivisor(a, b) {**
* **var primesA = primeFactors(a), primesB = primeFactors(b)**
* **while (primesA.length) {**
* **var max = primesA.pop()**
* **if (primesB.includes(max)) return max**
* **}**
* **return -1**
* **}**
* **function primeFactors(n){**
* **let f = []**
* **while (n % 2 === 0) {**
* **f.push(2)**
* **n /= 2**
* **}**
* **for (let i = 3; i <= Math.sqrt(n); i++){**
* **while (n % i === 0){**
* **f.push(i)**
* **n /= i**
* **}**
* **}**
* **if (n > 1) {**
* **f.push(n)**
* **}**
* **return f**

**}**

* + - Best Practices0
    - Clever0
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5acf1a27e68d606752002b2f&review_id=5906df3ca976efdac800002d)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5906df3ca976efdac800002d/groups/5acf1a27e68d606752002b2f)
* [otheabould](https://www.codewars.com/users/otheabould)
* **function greatestCommonPrimeDivisor(a, b) {**
* **const min = Math.min(a, b);**
* **const getRange = num => [ ...Array(num).keys() ].slice(2);**
* **const isPrime = num => getRange(num)**
* **.every(n => num % n)**
* **return getRange(min)**
* **.filter(n => isPrime(n))**
* **.reverse()**
* **.find(n => !(a % n) && !(b % n)) || -1;**

**}**