**6 kyu**

**Simple Fun #303: Prime Product**

2094% of 1735[myjinxin2015](https://www.codewars.com/users/myjinxin2015)

JavaScript

* [TRAIN AGAIN](https://www.codewars.com/kata/simple-fun-number-303-prime-product/train/javascript)
* [NEXT KATA](https://www.codewars.com/trainer/javascript)

Details

[Solutions](https://www.codewars.com/kata/simple-fun-number-303-prime-product/solutions/javascript)

[Discourse](https://www.codewars.com/kata/simple-fun-number-303-prime-product/discuss/javascript)

* Add to Collection
* |
* Share this kata:

**Task**

We know that some numbers can be split into two primes. ie. 5 = 2 + 3, 10 = 3 + 7. But some numbers are not. ie. 17, 27, 35, etc..

Given a positive integer n. Determine whether it can be split into two primes. If yes, return the maximum product of two primes. If not, return 0 instead.

**Input/Output**

[input] integer n

A positive integer.

0 ≤ n ≤ 100000

[output] an integer

The possible maximum product of two primes. or return 0 if it's impossible split into two primes.

**Example**

For n = 1, the output should be 0.

1 can not split into two primes

For n = 4, the output should be 4.

4 can split into two primes 2 and 2. 2 x 2 = 4

For n = 20, the output should be 91.

20 can split into two primes 7 and 13 or 3 and 17. The maximum product is 7 x 13 = 91

<https://www.codewars.com/kata/simple-fun-number-303-prime-product/javascript>

<script>

**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] is not changed,*

*// then it is a prime*

**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** primeProduct(  n)

        {

**var** fp = SieveOfEratosthenes(n);

**var** max\_prod = 1;

**for**(let i = fp.length-1; i>=0; i--)

            {

**var** resto = n - fp[i];

**if**(fp.includes(resto))

                {

                    max\_prod = Math.max(max\_prod, fp[i] \* resto);

                }

            }

**return** max\_prod;

        }

 document.write(primeProduct(100));

</script>