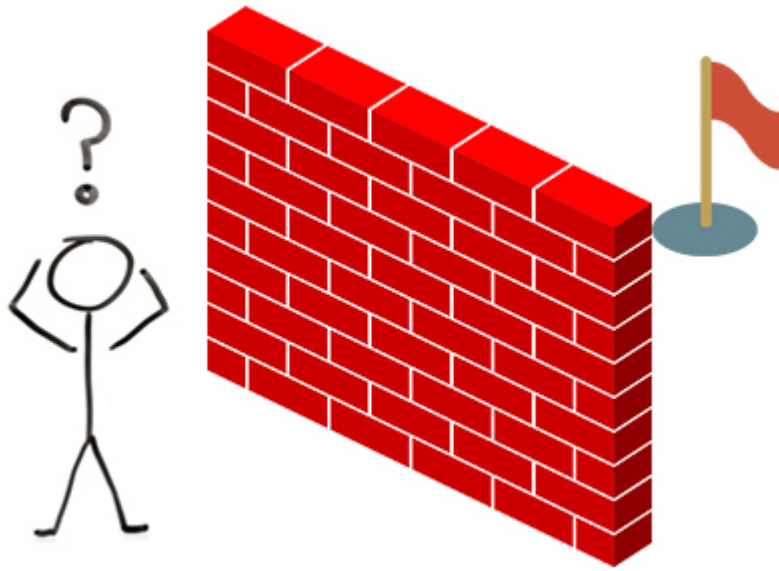


How to solve Coprimes in array



You seems to have hit the wall, but don't worry, you will be facing how to solve this problem along this document.

Decompose

First of all you must remember the golden rule "**Don't solve one big problem, solve many tiny problems**".

Definition of the problem: Given a set of N positive integer numbers, determine when this are coprimes or not.

As a statement, we know what is coprime, a coprime are two numbers which is Highest Common Factor (HCF) is 1.

It seems like an easy problem, we only need to check if a set HCF is 1, but maybe you are wondering, How I implement this?

You have been given the details to implement with two numbers, but the problem states **given a set of N positive integer numbers**, so how can you validate this? First of all, don't care about the specific details, first of all you need to implement **validate if pair of numbers are coprimes**.

You can try solving this by yourself or keep reading.

A pair of numbers are coprimes

For this task, also we need to remember the golden rule, decompose the problem.

Problem: Validate if a pair of numbers are coprimes.


Definitions:

- **Coprime:** Two numbers are coprime if is Highest Common Factor (HCF) is 1.
- **Factor:** Is a number that divides another number evenly i.e. with no remainder.


Pieces of the problem:

1. Get the Factors of each number. (Factors A, Factors B)
2. Check if there is a number in Factors B that is repeated in Factors A.

4: 1, 2, 2
7: 1, 7



4: 1, 2, 2
8: 1, 2, 2, 2



Step 2 it's an easy task, the problem you are facing now is to **get the factors of a given number**

Again. You can try solving this by yourself or keep reading.

Getting the factors of a number

We have reached a point where we can't decompose the problem, in other words, we are now in the roots of the problem. Once we solve this problem, everything is going to be like the smooth.

We already know this:

Factor: Is a number that divides another number evenly i.e. with no remainder.

So, our given problem is to get the factor of a number. But how are we gonna get all the factors of a number? Well, the answer is really easy, remember the definition of **Factor**.

In many programming languages, for not saying all of them, we have an operator called *modulo*, the expression is the following:

```
int remainder = 5 % 3; // 2
```

Tip: Modulo is going to return the remainder of a division

Maybe you're asking why this instruction is important. It's because we are going to need this instruction to get the factors of a number. There are many ways to solve this problem, the brute force way or the efficient way. Both of them, use the following rule with the modulo operator:

```
// If the number has no remainder when is divided by iterator
if (number % iterator == 0) {
    // Iterator is a factor of the number
} else {
    // Iterator is not a factor of the number
}
```

But the previous fragment of code, will only validate one number, we need to get all the factors of a number. For example:

Given 8 as the number, we want to get his factors: $1 \times 2 \times 2 \times 2$

The algorithm to get the factors is the following:

1. First of all we can exclude the 1, we already know that any number can be divided by 1, so it's something that we are omitting.
2. If 1 is excluded we start with an iterator of 2.
3. Check if $\text{number} \geq \text{iterator} * \text{iterator}$: (?)
 1. Check if the number have no remainder when it's divided by iterator
 1. Yes
 1. Save the iterator as a factor
 2. Number is the division of number between iterator.
 3. Repeat step 3.
 2. Not
 1. Increment our iterator by 1.
 2. Repeat step 3.
 4. Save the iterator as a factor.

Notes:

1. Why $\text{iterator} * \text{iterator}$, it's only a validation that will determine if there is any factors to get, this is an efficient way.

Once you have coded the algorithm, you can solve how to determine the coprimes.

Wrapping up

You have reached the end of this guide, with the previous information you have:

1. How to get the factors of a number
2. How to determine if 2 numbers are coprime

Now, you can deal with the current problem: *Determine if a set of numbers are coprime.*

As a partner in this track, I will mentor you to consider the following to solve this problem.

1. You have received all the info that you need to make an efficient solution.
2. You don't need to store all the factors in memory.
3. Think it in other way, we can check if the iterator is factor of the set of numbers?.
 1. Remember that 8 can be divided 3 times by 2, but 16 can be divided 4 times by 2, how can you validate this before incrementing the iterator to 3.
 2. Once you have encountered an iterator that is factor of the set, you can finish the search.

Well, now, I can only say Good Luck!!!