

## Terminology: -

**GCD:** GCD or Greatest Common Divisor of two or more positive integers is defined as the largest integer that divides all of them. For Example:  $\text{GCD}(4,8) = 4$ ,  $\text{GCD}(2,3) = 1$ . Two or more numbers are said to be pairwise coprime if the GCD of every pair of numbers is 1.

If  $\text{GCD}(a,b) = d$  then  $\text{GCD}(a/d, b/d) = 1$ .

**LCM:** LCM or Least Common Multiple of two or more positive integers is defined as the smallest positive integer that is a multiple of all of them. For Example:  $\text{LCM}(1,2) = 2$ ,  $\text{LCM}(3,9) = 9$

**Note:** The relation between GCD and LCM of two positive integers say 'a' and 'b', is given as  $\text{GCD}(a,b) * \text{LCM}(a,b) = a * b$ .

## Brute force Algorithm:

A naive way to calculate the GCD will be to iterate from all numbers from 1 to  $\min(a, b)$  and update the answer when we encounter a number that divides both a and b.

**Pseudocode:**

```
/* Input a and b are integers, returns the gcd(a,b) */
function GCD(a,b)

    // we handle the base case
    if min(a, b) == 0
        return max(a, b)
    ans = 1
    for i in range 1 to min(a, b)
        if i divides a and i divides b
            // update the answer when a common divisor is found
            ans = max(ans , i)

    return ans
```

**Time complexity:**  $O(\min(a,b))$ , where a and b are the given integers. Since we are looping i from 1 to  $\min(a, b)$ . The space complexity is  $O(1)$  since constant space is used.