

```
import java.util.Scanner;
```

You, seconds ago | 1 author (You)

```
public class GcdLcmNaive {
```

Run | Debug

```
public static void main(String[] args) {
```

```
    Scanner sc = new Scanner(System.in);
```

```
    int num1 = sc.nextInt();
```

```
    int num2 = sc.nextInt();
```

```
    int gcd = 0, lcm, count = 1, small;
```

```
    small = (num1 < num2) ? num1 : num2;
```

```
    while (count ≤ small) {
```

```
        if (num1 % count == 0 && num2 % count == 0) {
```

```
            gcd = count;
```

```
        }
```

```
        count++;
```

You, seconds ago • changes

```
    }
```

```
    lcm = (num1 * num2) / gcd;
```

```
    System.out.println(gcd);
```

```
    System.out.println(lcm);
```

```
}
```

```
}
```

NAIVE
Solⁿ.

GCD (n1, n2)

eg:

n1 = 4

n2 = 8

← smallest

loop from 1 to 4

count = 1

4

if the number ÷ is both nos

GCD = 4

★

easily get LCM:

$$\text{LCM} = (n1 \times n2) / \text{GCD};$$