

AP Tutorials

(Computation Sciences Division)

February 3, 2026

Aim

This document outlines the **algorithm**, **flowchart**, and **Java program** for computing:

1. HCF (Highest Common Factor) / GCD (Greatest Common Divisor)
2. LCM (Least Common Multiple)

The HCF and LCM are presented as **separate programs**.

Definitions

For two non-zero integers a and b :

- $\text{gcd}(a, b)$ (HCF) is the largest positive integer that divides both a and b .
- $\text{lcm}(a, b)$ is the smallest positive integer that is a multiple of both a and b .

A useful relation is

$$\text{lcm}(a, b) = \frac{|a b|}{\text{gcd}(a, b)}.$$

1 Program 1: HCF (GCD) using Euclid's Algorithm

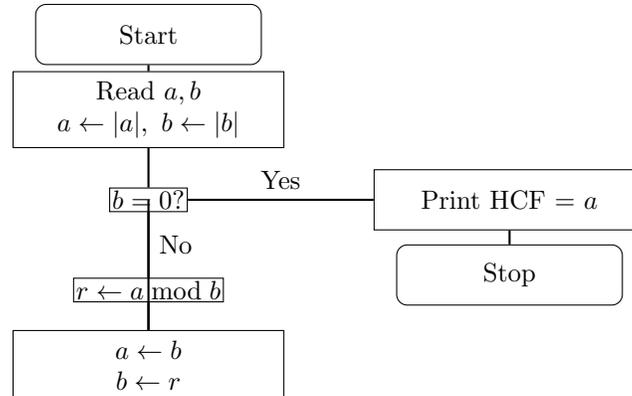
Algorithm

Input: Two integers a and b (not both zero).

1. Read a, b .
2. Set $a \leftarrow |a|, b \leftarrow |b|$.
3. While $b \neq 0$:
 - (a) $r \leftarrow a \bmod b$
 - (b) $a \leftarrow b$
 - (c) $b \leftarrow r$
4. Output a as $\text{gcd}(a, b)$.

Output: HCF/GCD.

Flowchart



Java Code (HCF)

```
1 import java.util.Scanner;
2
3 public class HCF_Euclid {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Enter first integer a: ");
8         long a = sc.nextLong();
9         System.out.print("Enter second integer b: ");
10        long b = sc.nextLong();
```

```
11
12     a = Math.abs(a);
13     b = Math.abs(b);
14
15     // Handle the special case where both are 0
16     if (a == 0 && b == 0) {
17         System.out.println("HCF is undefined when both numbers are 0."
18             );
19         sc.close();
20         return;
21     }
22
23     while (b != 0) {
24         long r = a % b;
25         a = b;
26         b = r;
27     }
28
29     System.out.println("HCF (GCD) = " + a);
30     sc.close();
31 }
```

2 Program 2: LCM using the GCD Relation

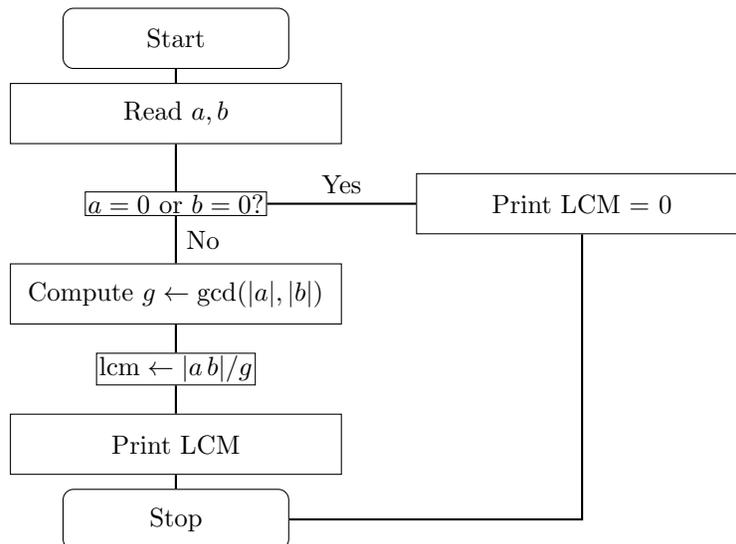
Algorithm

Input: Two integers a and b .

1. Read a, b .
2. If $a = 0$ or $b = 0$, output LCM = 0 and stop.
3. Compute $g \leftarrow \gcd(|a|, |b|)$ using Euclid's algorithm.
4. Compute $\text{lcm} \leftarrow \frac{|a b|}{g}$.
5. Output LCM.

Output: LCM.

Flowchart



Java Code (LCM)

```
1 import java.util.Scanner;
2
3 public class LCM_UsingGCD {
4
5     private static long gcd(long a, long b) {
6         a = Math.abs(a);
7         b = Math.abs(b);
```

```
8     while (b != 0) {
9         long r = a % b;
10        a = b;
11        b = r;
12    }
13    return a;
14 }
15
16 public static void main(String[] args) {
17     Scanner sc = new Scanner(System.in);
18
19     System.out.print("Enter first integer a: ");
20     long a = sc.nextLong();
21     System.out.print("Enter second integer b: ");
22     long b = sc.nextLong();
23
24     if (a == 0 || b == 0) {
25         System.out.println("LCM = 0");
26         sc.close();
27         return;
28     }
29
30     long g = gcd(a, b);
31
32     // Compute |a*b|/g in a way that reduces overflow risk
33     long lcm = Math.abs(a / g * b);
34
35     System.out.println("LCM = " + lcm);
36     sc.close();
37 }
38 }
```