NumberSystem (Recursion) FRQ

SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA.

Assume that the classes listed in the Java Quick Reference have been imported where appropriate. Unless otherwise noted in the question, assume that parameters in method calls are not `null` and that methods are called only when their preconditions are satisfied.
In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

This question involves computing the greatest common factor between two positive integers and using greatest common factors to reduce fractions. You will write two methods in the `NumberSystem` class that follows.

```
public class NumberSystem
{
  /** Precondition: a and b are positive integers.
  * Returns the greatest common factor of a and b, as described in
  part (a).
  */
  public static int gcf(int a, int b)
  { /* to be implemented in part (a) */ }

  /** Precondition: numerator and denominator are positive
  integers.
  * Reduces the fraction numerator / denominator
  * and prints the result, as described in part (b).
  */
  public static void reduceFraction(int numerator, int denominator)
  { /* to be implemented in part (b) */ }
}
```

The greatest common factor (GCF) of two integers a and b is the largest integer that divides evenly into both a and b. For example, the GCF of 8 and 12 is 4.

The greatest common factor can be computed using the following rules.

Case I: If a is evenly divisible by b, then the GCF is b.
Case II: If a is not evenly divisible by b, then the GCF of a and b is equal to the GCF of b and the remainder when a is divided by b.

If the rule in case II is repeatedly applied, it is guaranteed to eventually result in case I. Consider the following examples.

Example 1

In determining the GCF of 30 and 3, case I applies because 30 is evenly divisible by 3. Therefore, the GCF of 30 and 3 is 3.

Example 2

In determining the GCF of 3 and 30, case II applies because 3 is not evenly divisible by 30. The GCF of 3 and 30 will be equal to the GCF of 30 and the remainder when 3 is divided by 30, or 3.
In determining the GCF of 30 and 3, case I applies because 30 is evenly divisible by 3. The GCF of 30 and 3 is 3, and therefore the GCF of 3 and 30 is also 3.

Example 3

In determining the GCF of 24 and 9, case II applies because 24 is not evenly divisible by 9. The GCF of 24 and 9 will be equal to the GCF of 9 and the remainder when 24 is divided by 9, or 6.
In determining the GCF of 9 and 6, case II applies because 9 is not evenly divisible by 6. The GCF of 9 and 6 will be equal to the GCF of 6 and the remainder when 9 is divided by 6, or 3.
In determining the GCF of 6 and 3, case I applies because 6 is evenly divisible by 3. The GCF of 6 and 3 is 3, and therefore the GCF of 24 and 9 is also 3.

Example 4

In determining the GCF of 7 and 3, case II applies because 7 is not evenly divisible by 3. The GCF of 7 and 3 will be equal to the GCF of 3 and the remainder when 7 is divided by 3, or 1.
In determining the GCF of 3 and 1, case I applies because 3 is evenly divisible by 1. The GCF of 3 and 1 is 1, and therefore the GCF of 7 and 3 is also 1.

(a) The `gcf()` method returns the greatest common factor of parameters a and b, as determined by case I and case II. Write the gcf method below. You are encouraged to implement this method recursively.

```
/** Precondition: a and b are positive integers.
* Returns the greatest common factor of a and b, as described in part (a).
*/

public static int gcf(int a, int b)
```

Testing Code

In a fraction, the numerator is the number above the fraction bar and the denominator is the number below the fraction bar. A fraction can be reduced according to the following rules.

If the numerator is evenly divisible by the denominator, then the fraction reduces to the result when the numerator is divided by the denominator.
If the numerator is not evenly divisible by the denominator, then the reduced numerator will be equal to the numerator divided by the GCF of the numerator and the denominator. Similarly, the reduced denominator will be equal to the denominator divided by the GCF of the numerator and the denominator.

The `reduceFraction()` method is intended to reduce the fraction numerator / denominator and print the result. Examples of the intended behavior of the method are shown in the table.

Method Call	Message Printed	Explanation
`reduceFraction(30, 3)`	30/3 reduces to 10	30 is evenly divisible by 3. The fraction reduces to the result 10.
`reduceFraction(8, 20)`	8/20 reduces to 2/5	8 is not evenly divisible by 20. The numerator and denominator are each divided by the GCF, which is 4.
`reduceFraction(24, 9)`	24/9 reduces to 8/3	24 is not evenly divisible by 9. The numerator and denominator are each divided by the GCF, which is 3.
`reduceFraction(7, 3)`	7/3 reduces to 7/3	7 is not evenly divisible by 3. The numerator and denominator are each divided by the GCF, which is 1.

(b) Write the `reduceFraction()` method below. Assume that `gcf()` works as specified, regardless of what you wrote in part (a). You must use `gcf()` appropriately to receive full credit.

```
/** Precondition: numerator and denominator are positive integers.
* Reduces the fraction numerator / denominator
* and prints the result, as described in part (b).
*/
public static void reduceFraction(int numerator, int denominator)
```

Testing Code