This exercise gives you some practice defining, implementing, and using abstract data types. You will implement an abstract data type for **nonnegative rational numbers**. Represent a rational number using **two BigInt values**, a numerator and a denominator. For example, the rational number 5/3 has numerator 5 and denominator 3. However, the rational number should always be stored in its **lowest form**, i.e. not 4/8, but 1/2. You'll find BigInt’s gcd operation useful; a.gcd(b) returns the greatest common divisor of a and b.

Implement your data type as a Java class called Rat. It should have the following parts:

1. **Representation**, consisting of private fields
2. **Rep invariant**, implemented by the checkRep() method
3. **Abstraction function**, implemented by the toString() method
4. **Constructor** Rat(BigInt num, BigInt denom)
5. **Operations** provided by the following methods:
   ```java
   public boolean isZero()
   public boolean isOne()
   public Rat plus(Rat that)
   public Rat minus(Rat that)
   public Rat times(Rat that)
   public Rat divide(Rat that)
   ```
6. **Preconditions** and **postconditions** on the constructor and operations, where needed.

Hand in your solution on paper. You don’t have to test your solution, but if you want to, the code for BigInt can be found in the lecture examples repository. For reference, the public operations of BigInt are shown below.

```java
public class BigInt {
    public BigInt(int n);
    public BigInt(String s);
    public boolean isZero();
    public boolean isOne();
    public boolean isEven();
    public BigInt incr();
    public BigInt decr();
    public BigInt half();
    public BigInt twice();
    public BigInt plus(BigInt that);
    public BigInt minus(BigInt that);
    public BigInt times(BigInt that);
    public boolean lessThan(BigInt that);
    public BigInt timesPowerOf10(int k);
    public BigInt divide(BigInt that);
    public BigInt remainder(BigInt that);
    public BigInt gcd(BigInt that);
    public boolean equals(Object obj);
    public String toString();
}
```