

COS 126

General Computer Science

Spring 2010

Programming Exam 2

This test has 1 question. You have 50 minutes. The exam is open book, open note, and open web. You may use code from your programming assignments or the Introduction to Programming in Java booksite. No communication with any non-staff member is permitted. Submit your solution via Dropbox. **Write out and sign the Honor Code pledge before turning in this exam.**

"I pledge my honor that I have not violated the Honor Code during this examination."

Name:

Signature

NetID:

Total

| | | |
|------|----------|---------|
| P01 | TTh 1:30 | Jen |
| P01A | TTh 1:30 | Tim |
| P01B | TTh 1:30 | Donna |
| P01C | TTh 1:30 | Doug |
| P01D | TTh 1:30 | Casey |
| P02 | TTh 2:30 | Doug |
| P03 | TTh 3:30 | Jen |
| P04 | TTh 7:30 | Aleksey |
| P05 | WF 10 | Kevin |
| P06 | WF 1:30 | Donna |
| P06A | WF 1:30 | Chris |
| P06B | WF 1:30 | Kevin |

Do not remove this exam from the exam room.

Problem. Write a data type `BinaryInteger` that represents an immutable nonnegative binary integer of arbitrary length (possibly more than 64 bits) by implementing the following public API:

| | | |
|---|---|---|
| <code>public class BinaryInteger</code> | | |
| | <code>BinaryInteger(String s)</code> | <i>construct binary integer from string</i> |
| | <code>String toString()</code> | <i>string representation</i> |
| | <code>int length()</code> | <i>number of bits</i> |
| | <code>BinaryInteger not()</code> | <i>bitwise not of this binary integer</i> |
| | <code>BinaryInteger xor(BinaryInteger b)</code> | <i>bitwise xor of two binary integers</i> |
| | <code>int leadingZeros()</code> | <i>number of leading zeros</i> |
| | <code>boolean isGreaterThan(BinaryInteger b)</code> | <i>is this binary integer > b?</i> |

- *Constructor.* The constructor creates a binary integer based on the string argument. Throw a `RuntimeException` if the string argument contains a character other than '0' or '1'. Leading zeros are permitted.
- *String representation.* The `toString()` method returns a string representation of the binary integer.
- *Length.* The `length()` method returns the number of bits (including any leading zeros).
- *Bitwise not.* The `not()` method returns a binary integer (of the same length) in which all of the bits are flipped.
- *Bitwise xor.* The `xor()` method returns a binary integer (of the same length) which is the bitwise exclusive or of the two integers. Throw a `RuntimeException` if the two binary integers have different lengths.
- *Number of leading zeros.* The `leadingZeros()` method returns the number of consecutive leading (leftmost) zeros in the binary integer. For example, the number of leading zeros in the binary integer 00011110 is 3.
- *Comparison.* The `isGreaterThan()` method returns `true` if the invoking binary integer is greater than the argument binary integer, and `false` otherwise. The two binary integers can be of different lengths. Don't worry about negative integers.

Test client. The following test client demonstrates some of the desired behavior of the data type.

```
public class TestBinaryInteger {  
    public static void main(String[] args) {  
        BinaryInteger a = new BinaryInteger("00011110");  
        BinaryInteger b = new BinaryInteger("01010000");  
        System.out.println("a           = " + a);  
        System.out.println("b           = " + b);  
        System.out.println("a.length()     = " + a.length());  
        System.out.println("a.not()       = " + a.not());  
        System.out.println("a.xor(b)      = " + a.xor(b));  
        System.out.println("a.leadingZeros() = " + a.leadingZeros());  
        System.out.println("a.isGreaterThan(b) = " + a.isGreaterThan(b));  
    }  
}
```

Here is the corresponding output.

```
% java TestBinaryInteger  
a           = 00011110  
b           = 01010000  
a.length()     = 8  
a.not()       = 11100001  
a.xor(b)      = 01001110  
a.leadingZeros() = 3  
a.isGreaterThan(b) = false
```

Submission. Submit the single file `BinaryInteger.java` via Dropbox at

https://dropbox.cs.princeton.edu/COS126_S2010/Exam2

Be sure to click the *Check All Submitted Files* button to verify that your submission compiles cleanly.

Grading. Your program will be graded on correctness and clarity (including comments). You will receive partial credit for a program that correctly implements some of the methods in the API.

You will receive a substantial penalty if your program does not compile or if you do not follow the prescribed API.