```java
import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertTrue;

import org.junit.Test;

import components.naturalnumber.NaturalNumber;
import components.naturalnumber.NaturalNumber2;

/**
 * @author Put your name here
 *
 */
public class CryptoUtilitiesTest {

    /*
     * Tests of reduceToGCD
     */

    @Test
    public void testReduceToGCD_0_0() {
        NaturalNumber n = new NaturalNumber2(0);
        NaturalNumber nExpected = new NaturalNumber2(0);
        NaturalNumber m = new NaturalNumber2(0);
        NaturalNumber mExpected = new NaturalNumber2(0);
        CryptoUtilities.reduceToGCD(n, m);
        assertEquals(nExpected, n);
        assertEquals(mExpected, m);
    }

    @Test
    public void testReduceToGCD_8957646889654_26() {
        NaturalNumber n = new NaturalNumber2("8957646889654");
        NaturalNumber nExpected = new NaturalNumber2(2);
        NaturalNumber m = new NaturalNumber2(26);
        NaturalNumber mExpected = new NaturalNumber2();
        CryptoUtilities.reduceToGCD(n, m);
        assertEquals(nExpected, n);
        assertEquals(mExpected, m);
    }

    @Test
    public void testReduceToGCD_30_21() {
        NaturalNumber n = new NaturalNumber2(30);
        NaturalNumber nExpected = new NaturalNumber2(3);
```

```
45          NaturalNumber m = new NaturalNumber2(21);
46          NaturalNumber mExpected = new NaturalNumber2(0);
47          CryptoUtilities.reduceToGCD(n, m);
48          assertEquals(nExpected, n);
49          assertEquals(mExpected, m);
50      }
51
52      @Test
53      public void testReduceToGCD_12_16() {
54          NaturalNumber n = new NaturalNumber2(12);
55          NaturalNumber nExpected = new NaturalNumber2(4);
56          NaturalNumber m = new NaturalNumber2(16);
57          NaturalNumber mExpected = new NaturalNumber2(0);
58          CryptoUtilities.reduceToGCD(n, m);
59          assertEquals(nExpected, n);
60          assertEquals(mExpected, m);
61      }
62
63      /*
64       * Tests of isEven
65       */
66
67      @Test
68      public void testIsEven_0() {
69          NaturalNumber n = new NaturalNumber2(0);
70          NaturalNumber nExpected = new NaturalNumber2(0);
71          boolean result = CryptoUtilities.isEven(n);
72          assertEquals(nExpected, n);
73          assertEquals(true, result);
74      }
75
76      @Test
77      public void testIsEven_1() {
78          NaturalNumber n = new NaturalNumber2(1);
79          NaturalNumber nExpected = new NaturalNumber2(1);
80          boolean result = CryptoUtilities.isEven(n);
81          assertEquals(nExpected, n);
82          assertEquals(false, result);
83      }
84
85      @Test
86      public void testIsEven_72918209334() {
87          NaturalNumber n = new NaturalNumber2("72918209334");
88          NaturalNumber nExpected = new
```

```
   NaturalNumber2("72918209334");
 89         boolean result = CryptoUtilities.isEven(n);
 90         assertEquals(nExpected, n);
 91         assertEquals(true, result);
 92     }
 93
 94     @Test
 95     public void testIsEven_72918209331() {
 96         NaturalNumber n = new NaturalNumber2("72918209331");
 97         NaturalNumber nExpected = new
   NaturalNumber2("72918209331");
 98         boolean result = CryptoUtilities.isEven(n);
 99         assertEquals(nExpected, n);
100         assertEquals(false, result);
101     }
102
103     /*
104      * Tests of powerMod
105      */
106
107     @Test
108     public void testPowerMod_0_0_2() {
109         NaturalNumber n = new NaturalNumber2(0);
110         NaturalNumber nExpected = new NaturalNumber2(1);
111         NaturalNumber p = new NaturalNumber2(0);
112         NaturalNumber pExpected = new NaturalNumber2(0);
113         NaturalNumber m = new NaturalNumber2(2);
114         NaturalNumber mExpected = new NaturalNumber2(2);
115         CryptoUtilities.powerMod(n, p, m);
116         assertEquals(nExpected, n);
117         assertEquals(pExpected, p);
118         assertEquals(mExpected, m);
119     }
120
121     @Test
122     public void testPowerMod_17_18_19() {
123         NaturalNumber n = new NaturalNumber2(17);
124         NaturalNumber nExpected = new NaturalNumber2(1);
125         NaturalNumber p = new NaturalNumber2(18);
126         NaturalNumber pExpected = new NaturalNumber2(18);
127         NaturalNumber m = new NaturalNumber2(19);
128         NaturalNumber mExpected = new NaturalNumber2(19);
129         CryptoUtilities.powerMod(n, p, m);
130         assertEquals(nExpected, n);
```

```
131         assertEquals(pExpected, p);
132         assertEquals(mExpected, m);
133     }
134
135     @Test
136     public void testPowerMod_0_41_42() {
137         NaturalNumber n = new NaturalNumber2();
138         NaturalNumber nExpected = new NaturalNumber2();
139         NaturalNumber p = new NaturalNumber2(20);
140         NaturalNumber pExpected = new NaturalNumber2(20);
141         NaturalNumber m = new NaturalNumber2(42);
142         NaturalNumber mExpected = new NaturalNumber2(42);
143         CryptoUtilities.powerMod(n, p, m);
144         assertEquals(nExpected, n);
145         assertEquals(pExpected, p);
146         assertEquals(mExpected, m);
147     }
148
149     @Test
150     public void testPowerMod_8_32_33() {
151         NaturalNumber n = new NaturalNumber2(8);
152         NaturalNumber nExpected = new NaturalNumber2(31);
153         NaturalNumber p = new NaturalNumber2(32);
154         NaturalNumber pExpected = new NaturalNumber2(32);
155         NaturalNumber m = new NaturalNumber2(33);
156         NaturalNumber mExpected = new NaturalNumber2(33);
157         CryptoUtilities.powerMod(n, p, m);
158         assertEquals(nExpected, n);
159         assertEquals(pExpected, p);
160         assertEquals(mExpected, m);
161     }
162
163     @Test
164     public void testPowerMod_21_2_33() {
165         NaturalNumber n = new NaturalNumber2(21);
166         NaturalNumber nExpected = new NaturalNumber2(12);
167         NaturalNumber p = new NaturalNumber2(2);
168         NaturalNumber pExpected = new NaturalNumber2(2);
169         NaturalNumber m = new NaturalNumber2(33);
170         NaturalNumber mExpected = new NaturalNumber2(33);
171         CryptoUtilities.powerMod(n, p, m);
172         assertEquals(nExpected, n);
173         assertEquals(pExpected, p);
174         assertEquals(mExpected, m);
```

```
175     }
176
177     /*
178      * Tests of isWitnessToCompositeness
179      */
180
181     @Test
182     public void testIsWitnessToCompositeness_2_4() {
183         NaturalNumber w = new NaturalNumber2(2);
184         NaturalNumber wExpected = new NaturalNumber2(2);
185         NaturalNumber n = new NaturalNumber2(4);
186         NaturalNumber nExpected = new NaturalNumber2(4);
187         assertTrue(CryptoUtilities.isWitnessToCompositeness(w,
    n));
188         assertEquals(nExpected, n);
189         assertEquals(wExpected, w);
190     }
191
192     @Test
193     public void testIsWitnessToCompositeness_5_19() {
194         NaturalNumber w = new NaturalNumber2(5);
195         NaturalNumber wExpected = new NaturalNumber2(5);
196         NaturalNumber n = new NaturalNumber2(19);
197         NaturalNumber nExpected = new NaturalNumber2(19);
198         assertTrue(!CryptoUtilities.isWitnessToCompositeness(w,
    n));
199         assertEquals(nExpected, n);
200         assertEquals(wExpected, w);
201     }
202
203     @Test
204     public void testIsWitnessToCompositeness_485_991() {
205         NaturalNumber w = new NaturalNumber2(485);
206         NaturalNumber wExpected = new NaturalNumber2(485);
207         NaturalNumber n = new NaturalNumber2(991);
208         NaturalNumber nExpected = new NaturalNumber2(991);
209         assertTrue(!CryptoUtilities.isWitnessToCompositeness(w,
    n));
210         assertEquals(nExpected, n);
211         assertEquals(wExpected, w);
212     }
213
214     @Test
215     public void testIsWitnessToCompositeness_917_1011() {
```

```
216         NaturalNumber w = new NaturalNumber2(917);
217         NaturalNumber wExpected = new NaturalNumber2(917);
218         NaturalNumber n = new NaturalNumber2(1011);
219         NaturalNumber nExpected = new NaturalNumber2(1011);
220         assertTrue(CryptoUtilities.isWitnessToCompositeness(w,
    n));
221         assertEquals(nExpected, n);
222         assertEquals(wExpected, w);
223     }
224
225     @Test
226     public void testIsWitnessToCompositeness_2_9000() {
227         NaturalNumber w = new NaturalNumber2(2);
228         NaturalNumber wExpected = new NaturalNumber2(2);
229         NaturalNumber n = new NaturalNumber2(9000);
230         NaturalNumber nExpected = new NaturalNumber2(9000);
231         assertTrue(CryptoUtilities.isWitnessToCompositeness(w,
    n));
232         assertEquals(nExpected, n);
233         assertEquals(wExpected, w);
234     }
235
236     /*
237      * Tests of isPrime2
238      */
239     @Test
240     public void testIsPrime2_2() {
241         NaturalNumber n = new NaturalNumber2(2);
242         NaturalNumber nExpected = new NaturalNumber2(2);
243         assertTrue(CryptoUtilities.isPrime2(n));
244         assertEquals(nExpected, n);
245     }
246
247     @Test
248     public void testIsPrime2_3() {
249         NaturalNumber n = new NaturalNumber2(3);
250         NaturalNumber nExpected = new NaturalNumber2(3);
251         assertTrue(CryptoUtilities.isPrime2(n));
252         assertEquals(nExpected, n);
253     }
254
255     @Test
256     public void testIsPrime2_15() {
257         NaturalNumber n = new NaturalNumber2(15);
```

```
258         NaturalNumber nExpected = new NaturalNumber2(15);
259         assertTrue(!CryptoUtilities.isPrime2(n));
260         assertEquals(nExpected, n);
261     }
262
263     @Test
264     public void testIsPrime2_1009() {
265         NaturalNumber n = new NaturalNumber2(1009);
266         NaturalNumber nExpected = new NaturalNumber2(1009);
267         assertTrue(CryptoUtilities.isPrime2(n));
268         assertEquals(nExpected, n);
269     }
270
271     @Test
272     public void testIsPrime2_1013() {
273         NaturalNumber n = new NaturalNumber2(1013);
274         NaturalNumber nExpected = new NaturalNumber2(1013);
275         assertTrue(CryptoUtilities.isPrime2(n));
276         assertEquals(nExpected, n);
277     }
278
279     /*
280      * Tests of generateNextLikelyPrime
281      */
282     @Test
283     public void testGenerateNextLikelyPrime_1013() {
284         NaturalNumber n = new NaturalNumber2(1013);
285         NaturalNumber nExpected = new NaturalNumber2(1013);
286         CryptoUtilities.generateNextLikelyPrime(n);
287         assertEquals(nExpected, n);
288     }
289
290     @Test
291     public void testGenerateNextLikelyPrime_5004() {
292         NaturalNumber n = new NaturalNumber2(5004);
293         NaturalNumber nExpected = new NaturalNumber2(5009);
294         CryptoUtilities.generateNextLikelyPrime(n);
295         assertEquals(nExpected, n);
296     }
297
298     @Test
299     public void testGenerateNextLikelyPrime_6() {
300         NaturalNumber n = new NaturalNumber2(6);
301         NaturalNumber nExpected = new NaturalNumber2(7);
```

```
302         CryptoUtilities.generateNextLikelyPrime(n);
303         assertEquals(nExpected, n);
304     }
305
306     @Test
307     public void testGenerateNextLikelyPrime_19() {
308         NaturalNumber n = new NaturalNumber2(19);
309         NaturalNumber nExpected = new NaturalNumber2(19);
310         CryptoUtilities.generateNextLikelyPrime(n);
311         assertEquals(nExpected, n);
312     }
313
314     @Test
315     public void testGenerateNextLikelyPrime_100() {
316         NaturalNumber n = new NaturalNumber2(100);
317         NaturalNumber nExpected = new NaturalNumber2(101);
318         CryptoUtilities.generateNextLikelyPrime(n);
319         assertEquals(nExpected, n);
320     }
321 }
322
```