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

import org.junit.Test;

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

/**
 * @author Feras Akileh
 *
 */
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_30_21() {
        NaturalNumber n = new NaturalNumber2(30);
        NaturalNumber nExpected = new NaturalNumber2(3);
        NaturalNumber m = new NaturalNumber2(21);
        NaturalNumber mExpected = new NaturalNumber2(0);
        CryptoUtilities.reduceToGCD(n, m);
        assertEquals(nExpected, n);
        assertEquals(mExpected, m);
    }

    @Test
    public void testReduceToGCD_12_15() {
        NaturalNumber n = new NaturalNumber2(12);
        NaturalNumber nExpected = new NaturalNumber2(3);
        NaturalNumber m = new NaturalNumber2(15);
        NaturalNumber mExpected = new NaturalNumber2(0);
        CryptoUtilities.reduceToGCD(n, m);
        assertEquals(nExpected, n);
        assertEquals(mExpected, m);

    }

    @Test
    public void testReduceToGCD_214748365_631830834() {
        NaturalNumber n = new NaturalNumber2(214748365);
        NaturalNumber nExpected = new NaturalNumber2(1);
        NaturalNumber m = new NaturalNumber2(631830834);
        NaturalNumber mExpected = new NaturalNumber2(0);
        CryptoUtilities.reduceToGCD(n, m);
        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_23() {
87          NaturalNumber n = new NaturalNumber2(23);
88          NaturalNumber nExpected = new NaturalNumber2(23);
89          boolean result = CryptoUtilities.isEven(n);
90          assertEquals(nExpected, n);
91          assertEquals(false, result);
92      }
93
94      @Test
95      public void testIsEven_564389429() {
96          NaturalNumber n = new NaturalNumber2(564389429);
97          NaturalNumber nExpected = new NaturalNumber2(564389429);
98          boolean result = CryptoUtilities.isEven(n);
99          assertEquals(nExpected, n);
100         assertEquals(false, result);
101     }
102
103     @Test
104     public void testIsEven_624() {
105         NaturalNumber n = new NaturalNumber2(624);
106         NaturalNumber nExpected = new NaturalNumber2(624);
107         boolean result = CryptoUtilities.isEven(n);
108         assertEquals(nExpected, n);
109         assertEquals(true, result);
110     }
111
112     /*
113      * Tests of powerMod
114      */
115
116     @Test
117     public void testPowerMod_17_5_15() {
118         NaturalNumber n = new NaturalNumber2(17);
```

```
119         NaturalNumber p = new NaturalNumber2(5);
120         NaturalNumber m = new NaturalNumber2(15);
121         //out.println("Running this");
122         CryptoUtilities.powerMod(n, p, m);
123         //out.println("Exit with n=" + n.toString());
124         assertEquals("2", n.toString());
125         assertEquals("5", p.toString());
126         assertEquals("15", m.toString());
127     }
128 
129     @Test
130     public void testPowerMod_0_0_2() {
131         NaturalNumber n = new NaturalNumber2(0);
132         NaturalNumber nExpected = new NaturalNumber2(1);
133         NaturalNumber p = new NaturalNumber2(0);
134         NaturalNumber pExpected = new NaturalNumber2(0);
135         NaturalNumber m = new NaturalNumber2(2);
136         NaturalNumber mExpected = new NaturalNumber2(2);
137         CryptoUtilities.powerMod(n, p, m);
138         assertEquals(nExpected, n);
139         assertEquals(pExpected, p);
140         assertEquals(mExpected, m);
141     }
142 
143     @Test
144     public void testPowerMod_17_18_19() {
145         NaturalNumber n = new NaturalNumber2(17);
146         NaturalNumber nExpected = new NaturalNumber2(1);
147         NaturalNumber p = new NaturalNumber2(18);
148         NaturalNumber pExpected = new NaturalNumber2(18);
149         NaturalNumber m = new NaturalNumber2(19);
150         NaturalNumber mExpected = new NaturalNumber2(19);
151         CryptoUtilities.powerMod(n, p, m);
152         assertEquals(nExpected, n);
153         assertEquals(pExpected, p);
154         assertEquals(mExpected, m);
155     }
156 
157     @Test
158     public void testPowerMod_8_8_8() {
159         NaturalNumber n = new NaturalNumber2(8);
160         NaturalNumber p = new NaturalNumber2(8);
161         NaturalNumber m = new NaturalNumber2(8);
162         CryptoUtilities.powerMod(n, p, m);
163         assertEquals("0", n.toString());
164         assertEquals("8", p.toString());
165         assertEquals("8", m.toString());
166     }
167 
168     /*
169      * Tests of isWitnessToCompositeness
170      */
171 
172     @Test
173     public void isWitnessToCompositeness_2_33() {
174         NaturalNumber w = new NaturalNumber2(2);
175         NaturalNumber wExpected = new NaturalNumber2(2);
176         NaturalNumber n = new NaturalNumber2(33);
177         NaturalNumber nExpected = new NaturalNumber2(33);
```

```
178         boolean result = CryptoUtilities.isWitnessToCompositeness(w, n);
179         assertEquals(wExpected, w);
180         assertEquals(nExpected, n);
181         assertEquals(true, result);
182     }
183 
184     @Test
185     public void isWitnessToCompositeness_IntegerMaxVal() {
186         NaturalNumber w = new NaturalNumber2(17);
187         NaturalNumber wExpected = new NaturalNumber2(17);
188         NaturalNumber n = new NaturalNumber2(Integer.MAX_VALUE);
189         NaturalNumber nExpected = new NaturalNumber2(Integer.MAX_VALUE);
190         boolean result = CryptoUtilities.isWitnessToCompositeness(w, n);
191         assertEquals(wExpected, w);
192         assertEquals(nExpected, n);
193         assertEquals(true, result);
194     }
195 
196     @Test
197     public void isWitnessToCompositeness_17_214748364() {
198         NaturalNumber w = new NaturalNumber2(17);
199         NaturalNumber wExpected = new NaturalNumber2(17);
200         NaturalNumber n = new NaturalNumber2(214748364);
201         NaturalNumber nExpected = new NaturalNumber2(214748364);
202         boolean result = CryptoUtilities.isWitnessToCompositeness(w, n);
203         assertEquals(wExpected, w);
204         assertEquals(nExpected, n);
205         assertEquals(true, result);
206     }
207 
208     /*
209      * Tests of isPrime2
210      */
211 
212     @Test
213     public void testIsPrime2_2439549() {
214         NaturalNumber n = new NaturalNumber2(2439549);
215         NaturalNumber nExpected = new NaturalNumber2(2439549);
216         boolean result = CryptoUtilities.isPrime2(n);
217         assertEquals(nExpected, n);
218         assertEquals(false, result);
219     }
220 
221     @Test
222     public void testIsPrime2_24() {
223         NaturalNumber n = new NaturalNumber2(24);
224         NaturalNumber nExpected = new NaturalNumber2(24);
225         boolean result = CryptoUtilities.isPrime2(n);
226         assertEquals(nExpected, n);
227         assertEquals(false, result);
228     }
229 
230     @Test
231     public void testIsPrime2_4() {
232         NaturalNumber n = new NaturalNumber2(4);
233         NaturalNumber nExpected = new NaturalNumber2(4);
234         boolean result = CryptoUtilities.isPrime2(n);
235         assertEquals(nExpected, n);
236         assertEquals(false, result);
```

```
237     }
238
239     /*
240      * Tests of generateNextLikelyPrime
241      */
242
243     @Test
244     public void generateNextLikelyPrime_16() {
245         NaturalNumber n = new NaturalNumber2(16);
246         NaturalNumber nExpected = new NaturalNumber2(17);
247         CryptoUtilities.generateNextLikelyPrime(n);
248         assertEquals(nExpected, n);
249     }
250
251     @Test
252     public void generateNextLikelyPrime_96() {
253         NaturalNumber n = new NaturalNumber2(96);
254         NaturalNumber nExpected = new NaturalNumber2(97);
255         CryptoUtilities.generateNextLikelyPrime(n);
256         assertEquals(nExpected, n);
257     }
258
259     @Test
260     public void generateNextLikelyPrime_56236() {
261         NaturalNumber n = new NaturalNumber2(56236);
262         NaturalNumber nExpected = new NaturalNumber2(56237);
263         CryptoUtilities.generateNextLikelyPrime(n);
264         assertEquals(nExpected, n);
265     }
266
267 }
```