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

/**
 * @author Shyam Sai Bethina
 *
 */
public class CryptoUtilitiesTest {

    /*
     * Tests of reduceToGCD
     */

    //boundary
    @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);
    }

    //routine
    @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);
    }

    //routine
    @Test
    public void testReduceToGCD_9_3() {
        NaturalNumber n = new NaturalNumber2(9);
        NaturalNumber nExpected = new NaturalNumber2(3);
        NaturalNumber m = new NaturalNumber2(3);
        NaturalNumber mExpected = new NaturalNumber2(0);
        CryptoUtilities.reduceToGCD(n, m);
```

```
50          assertEquals(nExpected, n);
51          assertEquals(mExpected, m);
52      }
53
54      //challenging/routine
55      @Test
56      public void testReduceToGCD_100_100() {
57          NaturalNumber n = new NaturalNumber2(100);
58          NaturalNumber nExpected = new NaturalNumber2(100);
59          NaturalNumber m = new NaturalNumber2(100);
60          NaturalNumber mExpected = new NaturalNumber2(0);
61          CryptoUtilities.reduceToGCD(n, m);
62          assertEquals(nExpected, n);
63          assertEquals(mExpected, m);
64      }
65
66      /*
67       * Tests of isEven
68       */
69
70      //bounday
71      @Test
72      public void testIsEven_0() {
73          NaturalNumber n = new NaturalNumber2(0);
74          NaturalNumber nExpected = new NaturalNumber2(0);
75          boolean result = CryptoUtilities.isEven(n);
76          assertEquals(nExpected, n);
77          assertEquals(true, result);
78      }
79
80      //challenging
81      @Test
82      public void testIsEven_1() {
83          NaturalNumber n = new NaturalNumber2(1);
84          NaturalNumber nExpected = new NaturalNumber2(1);
85          boolean result = CryptoUtilities.isEven(n);
86          assertEquals(nExpected, n);
87          assertEquals(false, result);
88      }
89
90      //routine
91      @Test
92      public void testIsEven_10() {
93          NaturalNumber n = new NaturalNumber2(10);
```

```
 94         NaturalNumber nExpected = new NaturalNumber2(10);
 95         boolean result = CryptoUtilities.isEven(n);
 96         assertEquals(nExpected, n);
 97         assertEquals(true, result);
 98     }
 99 
100     //routine
101     @Test
102     public void testIsEven_123456789() {
103         NaturalNumber n = new NaturalNumber2(123456789);
104         NaturalNumber nExpected = new NaturalNumber2(123456789);
105         boolean result = CryptoUtilities.isEven(n);
106         assertEquals(nExpected, n);
107         assertEquals(false, result);
108     }
109 
110     /*
111      * Tests of powerMod
112      */
113 
114     //boundary
115     @Test
116     public void testPowerMod_0_0_2() {
117         NaturalNumber n = new NaturalNumber2(0);
118         NaturalNumber nExpected = new NaturalNumber2(1);
119         NaturalNumber p = new NaturalNumber2(0);
120         NaturalNumber pExpected = new NaturalNumber2(0);
121         NaturalNumber m = new NaturalNumber2(2);
122         NaturalNumber mExpected = new NaturalNumber2(2);
123         CryptoUtilities.powerMod(n, p, m);
124         assertEquals(nExpected, n);
125         assertEquals(pExpected, p);
126         assertEquals(mExpected, m);
127     }
128 
129     //routine
130     @Test
131     public void testPowerMod_17_18_19() {
132         NaturalNumber n = new NaturalNumber2(17);
133         NaturalNumber nExpected = new NaturalNumber2(1);
134         NaturalNumber p = new NaturalNumber2(18);
135         NaturalNumber pExpected = new NaturalNumber2(18);
136         NaturalNumber m = new NaturalNumber2(19);
137         NaturalNumber mExpected = new NaturalNumber2(19);
```

```
138         CryptoUtilities.powerMod(n, p, m);
139         assertEquals(nExpected, n);
140         assertEquals(pExpected, p);
141         assertEquals(mExpected, m);
142     }
143 
144     //routine
145     @Test
146     public void testPowerMod_3_6_7() {
147         NaturalNumber n = new NaturalNumber2(3);
148         NaturalNumber nExpected = new NaturalNumber2(1);
149         NaturalNumber p = new NaturalNumber2(6);
150         NaturalNumber pExpected = new NaturalNumber2(6);
151         NaturalNumber m = new NaturalNumber2(7);
152         NaturalNumber mExpected = new NaturalNumber2(7);
153         CryptoUtilities.powerMod(n, p, m);
154         assertEquals(nExpected, n);
155         assertEquals(pExpected, p);
156         assertEquals(mExpected, m);
157     }
158 
159     //boundary, very big number as modulus
160     @Test
161     public void testPowerMod_11_23_187() {
162         NaturalNumber n = new NaturalNumber2(11);
163         NaturalNumber nExpected = new NaturalNumber2(88);
164         NaturalNumber p = new NaturalNumber2(23);
165         NaturalNumber pExpected = new NaturalNumber2(23);
166         NaturalNumber m = new NaturalNumber2(187);
167         NaturalNumber mExpected = new NaturalNumber2(187);
168         CryptoUtilities.powerMod(n, p, m);
169         assertEquals(nExpected, n);
170         assertEquals(pExpected, p);
171         assertEquals(mExpected, m);
172     }
173 
174     /*
175      * Tests of isWitnessToCompositeness
176      */
177 
178     //routine
179     @Test
180     public void testIsWitnessToCompositeness_10_100() {
181         NaturalNumber n = new NaturalNumber2(10);
```

```
182         NaturalNumber nExpected = new NaturalNumber2(10);
183         NaturalNumber p = new NaturalNumber2(100);
184         NaturalNumber pExpected = new NaturalNumber2(100);
185         boolean result =
    CryptoUtilities.isWitnessToCompositeness(n, p);
186         assertEquals(nExpected, n);
187         assertEquals(pExpected, p);
188         assertEquals(true, result);
189     }
190 
191     //boundary
192     @Test
193     public void testIsWitnessToCompositeness_2_5() {
194         NaturalNumber n = new NaturalNumber2(2);
195         NaturalNumber nExpected = new NaturalNumber2(2);
196         NaturalNumber p = new NaturalNumber2(5);
197         NaturalNumber pExpected = new NaturalNumber2(5);
198         boolean result =
    CryptoUtilities.isWitnessToCompositeness(n, p);
199         assertEquals(nExpected, n);
200         assertEquals(pExpected, p);
201         assertEquals(false, result);
202     }
203 
204     //routine
205     @Test
206     public void testIsWitnessToCompositeness_120_122() {
207         NaturalNumber n = new NaturalNumber2(120);
208         NaturalNumber nExpected = new NaturalNumber2(120);
209         NaturalNumber p = new NaturalNumber2(122);
210         NaturalNumber pExpected = new NaturalNumber2(122);
211         boolean result =
    CryptoUtilities.isWitnessToCompositeness(n, p);
212         assertEquals(nExpected, n);
213         assertEquals(pExpected, p);
214         assertEquals(true, result);
215     }
216 
217     //routine
218     @Test
219     public void testIsWitnessToCompositeness_100_103() {
220         NaturalNumber n = new NaturalNumber2(100);
221         NaturalNumber nExpected = new NaturalNumber2(100);
222         NaturalNumber p = new NaturalNumber2(103);
```

```
223         NaturalNumber pExpected = new NaturalNumber2(103);
224         boolean result =
    CryptoUtilities.isWitnessToCompositeness(n, p);
225         assertEquals(nExpected, n);
226         assertEquals(pExpected, p);
227         assertEquals(false, result);
228     }
229
230     /*
231      * Tests of isPrime2
232      */
233     //routine
234     @Test
235     public void testIsPrime2_7() {
236         NaturalNumber n = new NaturalNumber2(7);
237         NaturalNumber nExpected = new NaturalNumber2(7);
238         boolean result = CryptoUtilities.isPrime2(n);
239         assertEquals(nExpected, n);
240         assertEquals(true, result);
241     }
242
243     //challenging
244     @Test
245     public void testIsPrime2_997() {
246         NaturalNumber n = new NaturalNumber2(997);
247         NaturalNumber nExpected = new NaturalNumber2(997);
248         boolean result = CryptoUtilities.isPrime2(n);
249         assertEquals(nExpected, n);
250         assertEquals(true, result);
251     }
252
253     //routine
254     @Test
255     public void testIsPrime2_10() {
256         NaturalNumber n = new NaturalNumber2(10);
257         NaturalNumber nExpected = new NaturalNumber2(10);
258         boolean result = CryptoUtilities.isPrime2(n);
259         assertEquals(nExpected, n);
260         assertEquals(false, result);
261     }
262
263     //challenging
264     @Test
265     public void testIsPrime2_998() {
```

```
266         NaturalNumber n = new NaturalNumber2(998);
267         NaturalNumber nExpected = new NaturalNumber2(998);
268         boolean result = CryptoUtilities.isPrime2(n);
269         assertEquals(nExpected, n);
270         assertEquals(false, result);
271     }
272 
273     /*
274      * Tests of generateNextLikelyPrime
275      */
276     //boundary
277     @Test
278     public void testGenerateNextLikelyPrime_996() {
279         NaturalNumber n = new NaturalNumber2(996);
280         NaturalNumber nExpected = new NaturalNumber2(997);
281         CryptoUtilities.generateNextLikelyPrime(n);
282         assertEquals(nExpected, n);
283     }
284 
285     //routine
286     @Test
287     public void testGenerateNextLikelyPrime_17() {
288         NaturalNumber n = new NaturalNumber2(17);
289         NaturalNumber nExpected = new NaturalNumber2(17);
290         CryptoUtilities.generateNextLikelyPrime(n);
291         assertEquals(nExpected, n);
292     }
293 
294     //boundary
295     @Test
296     public void testGenerateNextLikelyPrime_2() {
297         NaturalNumber n = new NaturalNumber2(2);
298         NaturalNumber nExpected = new NaturalNumber2(3);
299         CryptoUtilities.generateNextLikelyPrime(n);
300         assertEquals(nExpected, n);
301     }
302 
303     //challenging
304     @Test
305     public void testGenerateNextLikelyPrime_550() {
306         NaturalNumber n = new NaturalNumber2(550);
307         NaturalNumber nExpected = new NaturalNumber2(557);
308         CryptoUtilities.generateNextLikelyPrime(n);
309         assertEquals(nExpected, n);
```

```
310     }
311
312 }
313
```