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

/**
 * @author Yiming Cheng
 *
 */
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_11_121() {
        NaturalNumber n = new NaturalNumber2(11);
        NaturalNumber nExpected = new NaturalNumber2(11);
        NaturalNumber m = new NaturalNumber2(121);
        NaturalNumber mExpected = new NaturalNumber2(0);
        CryptoUtilities.reduceToGCD(n, m);
        assertEquals(nExpected, n);
        assertEquals(mExpected, m);
    }

    /*
     * Tests of isEven
     */

    @Test
    public void testIsEven_0() {
        NaturalNumber n = new NaturalNumber2(0);
        NaturalNumber nExpected = new NaturalNumber2(0);
        boolean result = CryptoUtilities.isEven(n);
        assertEquals(nExpected, n);
        assertEquals(true, result);
    }

    @Test
    public void testIsEven_1() {
        NaturalNumber n = new NaturalNumber2(1);
        NaturalNumber nExpected = new NaturalNumber2(1);
```

```
68          boolean result = CryptoUtilities.isEven(n);
69          assertEquals(nExpected, n);
70          assertEquals(false, result);
71      }
72
73      /*
74       * Tests of powerMod
75       */
76
77      @Test
78      public void testPowerMod_0_0_2() {
79          NaturalNumber n = new NaturalNumber2(0);
80          NaturalNumber nExpected = new NaturalNumber2(1);
81          NaturalNumber p = new NaturalNumber2(0);
82          NaturalNumber pExpected = new NaturalNumber2(0);
83          NaturalNumber m = new NaturalNumber2(2);
84          NaturalNumber mExpected = new NaturalNumber2(2);
85          CryptoUtilities.powerMod(n, p, m);
86          assertEquals(nExpected, n);
87          assertEquals(pExpected, p);
88          assertEquals(mExpected, m);
89      }
90
91      @Test
92      public void testPowerMod_17_18_19() {
93          NaturalNumber n = new NaturalNumber2(17);
94          NaturalNumber nExpected = new NaturalNumber2(1);
95          NaturalNumber p = new NaturalNumber2(18);
96          NaturalNumber pExpected = new NaturalNumber2(18);
97          NaturalNumber m = new NaturalNumber2(19);
98          NaturalNumber mExpected = new NaturalNumber2(19);
99          CryptoUtilities.powerMod(n, p, m);
100         assertEquals(nExpected, n);
101         assertEquals(pExpected, p);
102         assertEquals(mExpected, m);
103     }
104
105     /*
106      * Tests of isWitnessToCompositeness
107      */
108     @Test
109     public void isWitnessToCompositeness_11_29() {
110         NaturalNumber w = new NaturalNumber2(11);
111         NaturalNumber n = new NaturalNumber2(29);
112         boolean result = CryptoUtilities.isWitnessToCompositeness(w, n);
113         assertEquals(false, result);
114     }
115
116     @Test
117     public void isWitnessToCompositeness_7_35() {
118         NaturalNumber w = new NaturalNumber2(7);
119         NaturalNumber n = new NaturalNumber2(35);
120         boolean result = CryptoUtilities.isWitnessToCompositeness(w, n);
121         assertEquals(true, result);
122     }
123
124     /*
125      * Tests of isPrime1
126      */
127     @Test
128     public void testisPrime1_5() {
129         NaturalNumber n = new NaturalNumber2(5);
```

```
130         NaturalNumber nExpected = new NaturalNumber2(5);
131         boolean result = CryptoUtilities.isPrime1(n);
132         assertEquals(n, nExpected);
133         assertEquals(result, true);
134     }
135 
136     @Test
137     public void testisPrime1_8() {
138         NaturalNumber n = new NaturalNumber2(8);
139         boolean result = CryptoUtilities.isPrime1(n);
140         assertEquals(false, result);
141     }
142 
143     /*
144      * Tests of isPrime2
145      */
146     @Test
147     public void testisPrime2_24() {
148         NaturalNumber n = new NaturalNumber2(24);
149         boolean result = CryptoUtilities.isPrime2(n);
150         assertEquals(false, result);
151     }
152 
153     @Test
154     public void testisPrime2_29() {
155         NaturalNumber n = new NaturalNumber2(29);
156         boolean result = CryptoUtilities.isPrime2(n);
157         assertEquals(true, result);
158     }
159 
160     /*
161      * Tests of generateNextLikelyPrime
162      */
163     @Test
164     public void testgenerateNextLikelyPrime2() {
165         NaturalNumber n = new NaturalNumber2(2);
166         NaturalNumber nExpected = new NaturalNumber2(2);
167         CryptoUtilities.generateNextLikelyPrime(n);
168         assertEquals(nExpected, n);
169     }
170 
171     @Test
172     public void testgenerateNextLikelyPrime11() {
173         NaturalNumber n = new NaturalNumber2(11);
174         NaturalNumber nExpected = new NaturalNumber2(11);
175         CryptoUtilities.generateNextLikelyPrime(n);
176         assertEquals(nExpected, n);
177     }
178 
179 }
```