

```
1 import static org.junit.Assert.assertEquals;
2
3 /**
4  * @author Micah Casey-Fusco
5  */
6
7 public class CryptoUtilitiesTest {
8
9     /**
10      * Tests of reduceToGCD
11      */
12
13     @Test
14     public void testReduceToGCD_0_0() {
15         NaturalNumber n = new NaturalNumber2(0);
16         NaturalNumber nExpected = new NaturalNumber2(0);
17         NaturalNumber m = new NaturalNumber2(0);
18         NaturalNumber mExpected = new NaturalNumber2(0);
19         CryptoUtilities.reduceToGCD(n, m);
20         assertEquals(nExpected, n);
21         assertEquals(mExpected, m);
22     }
23
24     @Test
25     public void testReduceToGCD_30_21() {
26         NaturalNumber n = new NaturalNumber2(30);
27         NaturalNumber nExpected = new NaturalNumber2(3);
28         NaturalNumber m = new NaturalNumber2(21);
29         NaturalNumber mExpected = new NaturalNumber2(0);
30         CryptoUtilities.reduceToGCD(n, m);
31         assertEquals(nExpected, n);
32         assertEquals(mExpected, m);
33     }
34
35     /**
36      * Tests of isEven
37      */
38
39     @Test
40     public void testIsEven_0() {
41         NaturalNumber n = new NaturalNumber2(0);
42         NaturalNumber nExpected = new NaturalNumber2(0);
43         boolean result = CryptoUtilities.isEven(n);
44         assertEquals(nExpected, n);
45         assertEquals(true, result);
46     }
47
48     @Test
49     public void testIsEven_1() {
50         NaturalNumber n = new NaturalNumber2(1);
51         NaturalNumber nExpected = new NaturalNumber2(1);
52         boolean result = CryptoUtilities.isEven(n);
53         assertEquals(nExpected, n);
54         assertEquals(false, result);
55     }
56
57     /**
58      */
59 }
```

```
63     * Tests of powerMod
64     */
65
66     @Test
67     public void testPowerMod_0_0_2() {
68         NaturalNumber n = new NaturalNumber2(0);
69         NaturalNumber nExpected = new NaturalNumber2(1);
70         NaturalNumber p = new NaturalNumber2(0);
71         NaturalNumber pExpected = new NaturalNumber2(0);
72         NaturalNumber m = new NaturalNumber2(2);
73         NaturalNumber mExpected = new NaturalNumber2(2);
74         CryptoUtilities.powerMod(n, p, m);
75         assertEquals(nExpected, n);
76         assertEquals(pExpected, p);
77         assertEquals(mExpected, m);
78     }
79
80     @Test
81     public void testPowerMod_17_18_19() {
82         NaturalNumber n = new NaturalNumber2(17);
83         NaturalNumber nExpected = new NaturalNumber2(1);
84         NaturalNumber p = new NaturalNumber2(18);
85         NaturalNumber pExpected = new NaturalNumber2(18);
86         NaturalNumber m = new NaturalNumber2(19);
87         NaturalNumber mExpected = new NaturalNumber2(19);
88         CryptoUtilities.powerMod(n, p, m);
89         assertEquals(nExpected, n);
90         assertEquals(pExpected, p);
91         assertEquals(mExpected, m);
92     }
93
94     /*
95     * Tests for isPrime2
96     */
97
98     @Test
99     public void isPrime2_5() {
100         NaturalNumber n = new NaturalNumber2(5);
101         NaturalNumber nExpected = new NaturalNumber2(5);
102         boolean result = CryptoUtilities.isPrime2(n);
103         assertEquals(nExpected, n);
104         assertEquals(true, result);
105     }
106
107     @Test
108     public void isPrime2_6() {
109         NaturalNumber n = new NaturalNumber2(6);
110         NaturalNumber nExpected = new NaturalNumber2(6);
111         boolean result = CryptoUtilities.isPrime2(n);
112         assertEquals(nExpected, n);
113         assertEquals(false, result);
114     }
115
116     /*
117     * Tests for nextLikelyPrime
118     */
```

```
120
121  @Test
122  public void nextPrime_24() {
123      NaturalNumber n = new NaturalNumber2(24);
124      NaturalNumber nExpected = new NaturalNumber2(29);
125      CryptoUtilities.generateNextLikelyPrime(n);
126      assertEquals(nExpected, n);
127
128  }
129
130  @Test
131  public void nextPrime_8() {
132      NaturalNumber n = new NaturalNumber2(8);
133      NaturalNumber nExpected = new NaturalNumber2(11);
134      CryptoUtilities.generateNextLikelyPrime(n);
135      assertEquals(nExpected, n);
136  }
137
138 }
```