```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 David Park
 *
 */
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);
    }

    /*
     * 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);
```

```
58          boolean result = CryptoUtilities.isEven(n);
59          assertEquals(nExpected, n);
60          assertEquals(false, result);
61      }
62
63      /*
64       * Tests of powerMod
65       */
66
67      @Test
68      public void testPowerMod_0_0_2() {
69          NaturalNumber n = new NaturalNumber2(0);
70          NaturalNumber nExpected = new NaturalNumber2(1);
71          NaturalNumber p = new NaturalNumber2(0);
72          NaturalNumber pExpected = new NaturalNumber2(0);
73          NaturalNumber m = new NaturalNumber2(2);
74          NaturalNumber mExpected = new NaturalNumber2(2);
75          CryptoUtilities.powerMod(n, p, m);
76          assertEquals(nExpected, n);
77          assertEquals(pExpected, p);
78          assertEquals(mExpected, m);
79      }
80
81      @Test
82      public void testPowerMod_17_18_19() {
83          NaturalNumber n = new NaturalNumber2(17);
84          NaturalNumber nExpected = new NaturalNumber2(1);
85          NaturalNumber p = new NaturalNumber2(18);
86          NaturalNumber pExpected = new NaturalNumber2(18);
87          NaturalNumber m = new NaturalNumber2(19);
88          NaturalNumber mExpected = new NaturalNumber2(19);
89          CryptoUtilities.powerMod(n, p, m);
90          assertEquals(nExpected, n);
91          assertEquals(pExpected, p);
92          assertEquals(mExpected, m);
93      }
94
95      @Test
96      public void isPrime2_50() {
97          NaturalNumber n = new NaturalNumber2(50);
98          assertTrue(CryptoUtilities.isPrime2(n));
99      }
100
101     @Test
102     public void isWitness2_30() {
103         NaturalNumber two = new NaturalNumber2(2);
104         NaturalNumber thirty = new NaturalNumber2(30);
105         assertTrue(CryptoUtilities.isWitnessToCompositeness(two, thirty));
106     }
107
108 }
```