```java
package edu.ics211.h09;

import static org.junit.Assert.assertEquals;

/**Tests HexadecialSudoku class.
 * @author khj (I didn't see Cam's until I had already written this.)
 *
 */
public class HexadecimalSudokuTester {

  /**tests the legal values method.
   *
   */
  @Test
  public void testLegalValues() {
    //Given: An unsolved sudoku puzzle
    //Puzzle is from C.Moore's code.
    int[][] example1 = { { 11, 2, 5, -1, 4, -1, 9, -1, 6, 14, -1, 1, -1, 3, -1, -1 },
        { 14, -1, 0, 9, -1, -1, 2, 12, 13, -1, 3, -1, 15, -1, -1, -1 },
        { 1, -1, -1, -1, -1, -1, 7, -1, -1, 9, -1, 2, 11, 5, 14, 0 },
        { 13, 8, -1, -1, 5, -1, -1, 0, -1, -1, 15, -1, -1, 9, -1, 2 },
        { 0, 7, 14, 2, -1, -1, -1, 9, -1, -1, -1, 5, -1, -1, 3, 15 },
        { 3, -1, -1, -1, 10, -1, -1, -1, 2, 4, 13, 15, -1, -1, 6, 11 },
        { 12, -1, 10, 13, -1, -1, -1, -1, 8, -1, -1, -1, 7, -1, 5, 9 },
        { 6, 11, -1, -1, -1, 15, -1, -1, -1, 12, 9, 3, -1, -1, 10, -1 },
        { 2, -1, -1, -1, 3, 7, 11, 4, 5, -1, -1, -1, 0, 13, -1, 8 },
        { 7, 6, 12, 8, -1, -1, -1, -1, 0, 13, -1, 11, 4, -1, -1, -1 },
        { 4, 9, 3, -1, -1, -1, -1, -1, 15, -1, 12, 7, 6, -1, 1, -1 },
        { 10, -1, 11, -1, 15, -1, 12, 1, 3, -1, -1, 14, 9, 7, -1, -1 },
        { 9, -1, 2, -1, 7, 4, 0, -1, -1, -1, 5, -1, -1, 8, 13, -1 },
        { 8, 3, 7, -1, -1, 9, 6, -1, 12, -1, -1, -1, -1, -1, -1, 14 },
        { 15, -1, 4, -1, 12, -1, 8, 10, -1, -1, -1, -1, 1, 6, 9, 7 },
        { 5, 12, -1, 6, -1, 3, 15, -1, 9, 0, -1, -1, 2, -1, -1, -1 } };


    //When: I test the legal values for the first empty cell
    ArrayList<Integer> legal = HexadecimalSudoku.legalValues(example1, 0, 3);

    //Then: It should return a list containing 7, 10, 12, 15
    assertEquals("incorrect amount of legal values", legal.size(), 4);
    assertEquals("should be 7", legal.get(0), Integer.valueOf(7));
    assertEquals("should be 10", legal.get(1), Integer.valueOf(10));
    assertEquals("should be 12", legal.get(2), Integer.valueOf(12));
    assertEquals("should be 15", legal.get(3), Integer.valueOf(15));
  }

  /**Tests solveSudoku method on 2 puzzles.
   * puzzles & sudoku reader provided by troy.
   *
   */
  @Test
  public void testSudoku() {
    //Given: A sudoku puzzle
    int[][] sudoku1 = SudokuReader.readSudoku("puzzle.txt", 16);
    int[][] sudoku2 = SudokuReader.readSudoku("puzzle2.txt", 16);

    //When: we run the solveSudoku method & print the solved puzzle
    HexadecimalSudoku.solveSudoku(sudoku1);
    System.out.println("sudoku1 solution\n" + HexadecimalSudoku.toString(sudoku1, true));
```

```
64      HexadecimalSudoku.solveSudoku(sudoku2);
65      System.out.println("sudoku2 solution\n" + HexadecimalSudoku.toString(sudoku2, true));
66
67      //Then: if it is solvable, it should be filled and the checkSudoku method should pass
68      assertTrue(HexadecimalSudoku.isFilled(sudoku1));
69      assertTrue(HexadecimalSudoku.isFilled(sudoku2));
70      assertTrue(HexadecimalSudoku.checkSudoku(sudoku1, true));
71      assertTrue(HexadecimalSudoku.checkSudoku(sudoku2, true));
72
73  }
74
75}
76
```