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

/**
 * *
 *
 * @author David Park
 *
 */
public class GlossaryTest {
    private static final String INPUT_FILE = "terms.txt";
    private static final String OUTPUT_FOLDER = "data";

    @Before
    public void setUp() {
        File outputFolder = new File(OUTPUT_FOLDER);
        if (!outputFolder.exists()) {
            outputFolder.mkdir();
        }
    }

    @Test
    public void testGenerateHTMLPages() {
        Glossary.generateHTMLPages(INPUT_FILE, OUTPUT_FOLDER);

        File indexFile = new File(OUTPUT_FOLDER + "/index.html");
        assertTrue(indexFile.exists());

        File termPage1 = new File(OUTPUT_FOLDER + "/term1.html");
        assertTrue(termPage1.exists());
    }

    @Test
    public void testReadTerms() {
        SimpleReader inFile = new SimpleReader1L(INPUT_FILE);
        Map<String, String> dictionary = new Map1L<>();

        Glossary.readTerms(inFile, dictionary);

        assertEquals("Definition of Term 1", dictionary.value("Term 1"));
    }

    @Test
    public void testSortTerms() {
        Map<String, String> dictionary = new Map1L<>();
        dictionary.add("Term 2", "Definition of Term 2");
        dictionary.add("Term 1", "Definition of Term 1");

        Queue<String> sortedTerms = Glossary.sortTerms(dictionary);

        assertEquals("Term 1", sortedTerms.dequeue());
        assertEquals("Term 2", sortedTerms.dequeue());
    }

    @Test
    public void testWriteIndexHtml() {
        Queue<String> terms = new Queue1L<>();
        terms.enqueue("Term 1");
```

```
71          terms.enqueue("Term 2");
72
73          Glossary.writeIndexHtml(terms, OUTPUT_FOLDER);
74
75          String expectedIndexContent = "<html><head><title>Glossary Index</title></head><body>
    \n"
76                  + "<h1>Glossary Index</h1><ul>\n"
77                  + "<li><a href=\"Term 1.html\">Term 1</a></li>\n"
78                  + "<li><a href=\"Term 2.html\">Term 2</a></li>\n"
79                  + "</ul></body></html>\n";
80          assertEquals(expectedIndexContent,
81                  this.fileContentsAsString(OUTPUT_FOLDER + "/index.html"));
82      }
83
84      @Test
85      public void testWriteTermPages() {
86          Map<String, String> dictionary = new Map1L<>();
87          dictionary.add("Term 1", "Definition of Term 1");
88          dictionary.add("Term 2", "Definition of Term 2");
89
90          Glossary.writeTermPages(dictionary, OUTPUT_FOLDER);
91
92          String expectedTermPage1Content = "<html><head><title>Term 1</title></head><body>\n"
93                  + "<h1 style='color:red; font-weight:bold; font-style:italic;'>Term 1</h1>\n"
94                  + "<p>Definition of Term 1</p>\n"
95                  + "<p>Return to <a href=\"index.html\">Index</a>.</p>\n"
96                  + "</body></html>\n";
97          assertEquals(expectedTermPage1Content,
98                  this.fileContentsAsString(OUTPUT_FOLDER + "/Term 1.html"));
99      }
100
101     @Test
102     public void testFormatDefinition() {
103         Map<String, String> dictionary = new Map1L<>();
104         dictionary.add("Term 1", "Definition of Term 1");
105         dictionary.add("Term 2", "Definition of Term 2");
106
107         String definition = "This is a definition that contains Term 1 and Term 2.";
108         String formattedDefinition = Glossary.formatDefinition(definition,
109                 dictionary);
110
111         String expectedFormattedDefinition = "This is a definition that contains <a href=
    \"Term 1.html\">Term 1</a> and <a href=\"Term 2.html\">Term 2</a>.";
112         assertEquals(expectedFormattedDefinition, formattedDefinition);
113     }
114
115     private String fileContentsAsString(String filename) {
116         SimpleReader reader = new SimpleReader1L(filename);
117         StringBuilder contents = new StringBuilder();
118
119         while (!reader.atEOS()) {
120             contents.append(reader.nextLine()).append("\n");
121         }
122         reader.close();
123         return contents.toString();
124     }
125 }
```