```
 1 import static org.junit.Assert.assertEquals;
 2
 3 import org.junit.Test;
 4
 5 import components.set.Set;
 6 import components.set.Set1L;
 7 import components.simplereader.SimpleReader;
 8 import components.simplereader.SimpleReader1L;
 9 import components.simplewriter.SimpleWriter;
10 import components.simplewriter.SimpleWriter1L;
11
12 /**
13  *
14  * @author ansh pachauri
15  *
16  */
17 public class StringReassemblyTest {
18
19     /**
20      * test of combination.
21      */
22
23     @Test
24     public void testCombination0() {
25         String str1 = "abcd";
26         String str2 = "cdefghi";
27         int overlap = 2;
28         String result = StringReassembly.combination(str1,
   str2, overlap);
29         assertEquals("abcdefghi", result);
30     }
31
32     /**
33      * test of combination.
34      */
35     @Test
36     public void testCombination1() {
37         String str1 = "abcdefghi";
38         String str2 = "cdefghijklmnopqrst";
```

```
39          int overlap = 7;
40          String result = StringReassembly.combination(str1,
  str2, overlap);
41          assertEquals("abcdefghijklmnopqrst", result);
42      }
43
44      /**
45       * test of combination.
46       */
47      @Test
48      public void testCombination2() {
49          String str1 = "abcd";
50          String str2 = "abcd";
51          int overlap = 4;
52          String result = StringReassembly.combination(str1,
  str2, overlap);
53          assertEquals("abcd", result);
54      }
55
56      /**
57       * test of addToSetAvoidingSubstrings.
58       */
59
60      @Test
61      public void testAddToSetAvoidingSubstrings0() {
62          Set<String> strSet = new Set1L<String>();
63          strSet.add("Hey");
64          strSet.add("How are you?");
65          strSet.add("I am good");
66          Set<String> check = new Set1L<String>();
67          check.add("Hey");
68          check.add("How are you?");
69          check.add("I am good");
70          String str = "good";
71          StringReassembly.addToSetAvoidingSubstrings(strSet,
  str);
72          assertEquals(check, strSet);
73      }
74
```

```
 75     /**
 76      * test of addToSetAvoidingSubstrings.
 77      */
 78     @Test
 79     public void testAddToSetAvoidingSubstrings1() {
 80         Set<String> strSet = new Set1L<String>();
 81         strSet.add("Hey");
 82         strSet.add("How are you?");
 83         strSet.add("I am good");
 84         Set<String> check = new Set1L<String>();
 85         check.add("Hey");
 86         check.add("How are you?");
 87         check.add("I am good");
 88         check.add("Nathan");
 89         String str = "Nathan";
 90         StringReassembly.addToSetAvoidingSubstrings(strSet,
    str);
 91         assertEquals(check, strSet);
 92     }
 93 
 94     /**
 95      * test of addToSetAvoidingSubstrings.
 96      */
 97     @Test
 98     public void testAddToSetAvoidingSubstrings2() {
 99         Set<String> strSet = new Set1L<String>();
100         strSet.add("Hey");
101         strSet.add("How are you?");
102         strSet.add("I am good, Nathaniel");
103         Set<String> check = new Set1L<String>();
104         check.add("Hey");
105         check.add("How are you?");
106         check.add("I am good, Nathaniel");
107         String str = "Nathan";
108         StringReassembly.addToSetAvoidingSubstrings(strSet,
    str);
109         assertEquals(check, strSet);
110     }
111 
```

```
112     /**
113      * test of addToSetAvoidingSubstrings.
114      */
115     @Test
116     public void testAddToSetAvoidingSubstrings3() {
117         Set<String> strSet = new Set1L<String>();
118         strSet.add("Hey");
119         strSet.add("How are you?");
120         strSet.add("I am good");
121         Set<String> check = new Set1L<String>();
122         check.add("Hey");
123         check.add("How are you?");
124         check.add("I am good Nathan");
125         String str = "I am good Nathan";
126         StringReassembly.addToSetAvoidingSubstrings(strSet,
    str);
127         assertEquals(check, strSet);
128     }
129 
130     /**
131      * test of linesFromInput.
132      */
133     @Test
134     public void testLinesFromInput0() {
135         String fileName = "testFile";
136         SimpleWriter fileOut = new SimpleWriter1L(fileName);
137         SimpleReader input = new SimpleReader1L(fileName);
138         String str = "abcdefgh";
139         fileOut.print(str);
140         Set<String> check = new Set1L<>();
141         check.add("abcdefgh");
142 
143         Set<String> result =
    StringReassembly.linesFromInput(input);
144 
145         assertEquals(check, result);
146     }
147 
148     /**
```

```
149     * test of linesFromInput.
150     */
151    @Test
152    public void testLinesFromInput1() {
153        String fileName = "testFile";
154        SimpleWriter fileOut = new SimpleWriter1L(fileName);
155        SimpleReader input = new SimpleReader1L(fileName);
156        String str = "a\nb\nc\ndefgh";
157        fileOut.print(str);
158        Set<String> check = new Set1L<>();
159        check.add("a");
160        check.add("b");
161        check.add("c");
162        check.add("defgh");
163
164        Set<String> result =
   StringReassembly.linesFromInput(input);
165
166        assertEquals(check, result);
167    }
168
169    /**
170     * test of linesFromInput.
171     */
172    @Test
173    public void testLinesFromInput2() {
174        String fileName = "testFile";
175        SimpleWriter fileOut = new SimpleWriter1L(fileName);
176        SimpleReader input = new SimpleReader1L(fileName);
177        String str = "a\nb\nc\nd\ne\nf\ng";
178        fileOut.print(str);
179        Set<String> check = new Set1L<>();
180        check.add("a");
181        check.add("b");
182        check.add("c");
183        check.add("d");
184        check.add("e");
185        check.add("f");
186        check.add("g");
```

```
187
188         Set<String> result =
    StringReassembly.linesFromInput(input);
189
190         assertEquals(check, result);
191     }
192
193     /**
194      * test of printWithLineSeparators.
195      */
196     @Test
197     public void printWithLineSeparators0() {
198
199         String fileName = "testFile";
200         SimpleWriter fileOut = new SimpleWriter1L(fileName);
201         SimpleReader input = new SimpleReader1L(fileName);
202         String str = "abcdefg";
203         String check = "abcdefg";
204         StringReassembly.printWithLineSeparators(str, fileOut);
205         String str1 = "";
206         while (!input.atEOS()) {
207             str1 += input.nextLine();
208             if (!input.atEOS()) {
209                 str1 += "\n";
210             }
211         }
212         assertEquals(check, str1);
213     }
214
215     /**
216      * test of printWithLineSeparators.
217      */
218     @Test
219     public void printWithLineSeparators1() {
220
221         String fileName = "testFile";
222         SimpleWriter fileOut = new SimpleWriter1L(fileName);
223         SimpleReader input = new SimpleReader1L(fileName);
224         String str = "a~b~c~d~e~f~g";
```

```
225         String check = "a\nb\nc\nd\ne\nf\ng";
226         StringReassembly.printWithLineSeparators(str, fileOut);
227         String str1 = "";
228         while (!input.atEOS()) {
229             str1 += input.nextLine();
230             if (!input.atEOS()) {
231                 str1 += "\n";
232             }
233         }
234         assertEquals(check, str1);
235     }
236 
237     /**
238      * test of printWithLineSeparators.
239      */
240     @Test
241     public void printWithLineSeparators2() {
242 
243         String fileName = "testFile";
244         SimpleWriter fileOut = new SimpleWriter1L(fileName);
245         SimpleReader input = new SimpleReader1L(fileName);
246         String str = "a~b~c~~~defg";
247         String check = "a\nb\nc\n\n\ndefg";
248         StringReassembly.printWithLineSeparators(str, fileOut);
249         String str1 = "";
250         while (!input.atEOS()) {
251             str1 += input.nextLine();
252             if (!input.atEOS()) {
253                 str1 += "\n";
254             }
255         }
256         assertEquals(check, str1);
257     }
258 
259 }
260 
```