

```
1 import static org.junit.Assert.assertEquals;
14
15 public class StringReassemblyTest {
16
17     @Test
18     public void testCombination1() {
19         String str1 = "yes ";
20         String str2 = " no";
21         int overlap = 2;
22         String ans = StringReassembly.combination(str1, str2, overlap);
23         String expect = "yes no";
24         assertEquals(expect, ans);
25     }
26
27     @Test
28     public void testCombination2() {
29         String str1 = "the other day i caught a ";
30         String str2 = " big ole floppy salmon";
31         int overlap = 1;
32         String ans = StringReassembly.combination(str1, str2, overlap);
33         String expect = "the other day i caught a big ole floppy salmon";
34         assertEquals(expect, ans);
35     }
36
37     @Test
38     public void testCombination3() {
39         String str1 = "something something something ";
40         String str2 = "testy testy testy";
41         int overlap = 0;
42         String ans = StringReassembly.combination(str1, str2, overlap);
43         String expect = "something something something testy testy testy";
44         assertEquals(expect, ans);
45     }
46
47     @Test
48     public void testAddToSetAvoidingSubstrings1() {
49         Set<String> setStr = new Set1L<>();
50         Set<String> setStrAns = new Set1L<>();
51         setStr.add("test!");
52         setStrAns.add("test!");
53         String str1 = "test!";
54         String str2 = "test!";
55         StringReassembly.addToSetAvoidingSubstrings(setStr, str1);
56         assertEquals(setStrAns, setStr);
57         assertEquals(str2, str1);
58     }
59
60     @Test
61     public void testAddToSetAvoidingSubstrings2() {
62         Set<String> setStr = new Set1L<>();
63         Set<String> setStrAns = new Set1L<>();
64         setStr.add("test!");
65         setStrAns.add("test!");
66         setStrAns.add("wowza");
67         String str1 = "wowza";
68         String str2 = "wowza";
69         StringReassembly.addToSetAvoidingSubstrings(setStr, str1);
70         assertEquals(setStrAns, setStr);
71         assertEquals(str2, str1);
```

```
72     }
73
74     @Test
75     public void testAddToSetAvoidingSubstrings3() {
76         Set<String> setStr = new Set1L<>();
77         Set<String> setStrAns = new Set1L<>();
78         setStr.add("test!");
79         setStr.add("something");
80         setStr.add("nothing69");
81         setStr.add(" yes)");
82         setStr.add("ASBFJDKUYVBWE");
83         setStrAns.add("test!");
84         setStrAns.add("something");
85         setStrAns.add("nothing69");
86         setStrAns.add(" yes)");
87         setStrAns.add("ASBFJDKUYVBWE");
88         setStrAns.add("the real answer nice");
89         String str1 = "the real answer nice";
90         String str2 = "the real answer nice";
91         StringReassembly.addToSetAvoidingSubstrings(setStr, str1);
92         assertEquals(setStrAns, setStr);
93         assertEquals(str2, str1);
94     }
95
96     @Test
97     public void testAddToSetAvoidingSubstrings4() {
98         Set<String> setStr = new Set1L<>();
99         Set<String> setStrAns = new Set1L<>();
100        setStr.add("test!");
101        setStr.add("something");
102        setStr.add("nothing69");
103        setStr.add(" yes)");
104        setStr.add("ASBFJDKUYVBWE");
105        setStrAns.add("test!");
106        setStrAns.add("something");
107        setStrAns.add("nothing69");
108        setStrAns.add(" yes)");
109        setStrAns.add("ASBFJDKUYVBWE");
110        String str1 = " yes)";
111        String str2 = " yes)";
112        StringReassembly.addToSetAvoidingSubstrings(setStr, str1);
113        assertEquals(setStrAns, setStr);
114        assertEquals(str2, str1);
115    }
116
117    @Test
118    public void testLinesFromInput1() {
119        SimpleWriter out = new SimpleWriter1L();
120        SimpleReader in = new SimpleReader1L();
121        Set<String> set = new Set1L<>();
122        SimpleReader inFile = new SimpleReader1L("data/cheerTest.txt");
123        Set<String> setAns = new Set1L<>();
124
125        set = StringReassembly.linesFromInput(inFile);
126        setAns.add("Bucks -- Beat");
127        setAns.add("Go Bucks");
128        setAns.add("o Bucks -- B");
129        setAns.add("Beat Mich");
130        setAns.add("Michigan~");
```

```
131
132     assertEquals(setAns, set);
133
134     out.close();
135     in.close();
136 }
137
138 @Test
139 public void testPrintWithLineSeparators1() throws IOException {
140     FileReader file = new FileReader("data/cheerTest.txt");
141     SimpleWriter out = new SimpleWriter1L();
142     String fileString = file.toString();
143
144     out.println("printWithLineSeparators Test Case 1:");
145     out.println("-----");
146     out.println("Result: ");
147     StringReassembly.printWithLineSeparators(fileString, out);
148     out.println();
149     out.println("Expected: ");
150     out.println("Go Bucks -- Beat Michigan");
151
152     file.close();
153 }
154 }
155
```