```java
import components.map.Map;

/**
 * {@code Queue} represented as a {@code Sequence} of entries, with
 * implementations of primary methods.
 *
 * @param <T>
 *            type of {@code Queue} entries
 * @correspondence this = $this.entries
 */
public class HelloWorld {

    public static void main(String[] args) {

    }

    /**
     * Raises the salary of all the employees in {@code map} whose name starts
     * with the given {@code initial} by the given {@code raisePercent}.
     *
     * @param map
     *            the name to salary map
     * @param initial
     *            the initial of names of employees to be given a raise
     * @param raisePercent
     *            the raise to be given as a percentage of the current salary
     * @updates map
     * @requires [the salaries in map are positive] and raisePercent > 0
     * @ensures <pre>
     * DOMAIN(map) = DOMAIN(#map)  and
     * [the salaries of the employees in map whose names start with the given
     *  initial have been increased by raisePercent percent (and truncated to
     *  the nearest integer); all other employees have the same salary]
     * </pre>
     */
    private static void giveRaise(components.map.Map<String,
```

```
Integer> map,
37              char initial, int raisePercent) {
38          Map<String, Integer> newMap = map.newInstance();
39          int size = map.size();
40          for (int i = 0; i < size; i++) {
41              components.map.Map.Pair<String, Integer> temp =
map.removeAny();
42              if (temp.key().charAt(0) == initial) {
43                  newMap.add(temp.key(), temp.value() *
(raisePercent));
44              } else {
45                  newMap.add(temp.key(), temp.value());
46              }
47          }
48          map.transferFrom(newMap);
49      }
50
51      /**
52       * Raises the salary of all the employees in {@code map} whose
name starts
53       * with the given {@code initial} by the given {@code
raisePercent}.
54       *
55       * @param map
56       *            the name to salary map
57       * @param initial
58       *            the initial of names of employees to be given a
raise
59       * @param raisePercent
60       *            the raise to be given as a percentage of the
current salary
61       * @updates map
62       * @requires [the salaries in map are positive] and
raisePercent > 0
63       * @ensures
64       *
65       *          <pre>
66       * DOMAIN(map) = DOMAIN(#map)  and
67       * [the salaries of the employees in map whose names start with
the given
68       *  initial have been increased by raisePercent percent (and
truncated to
69       *  the nearest integer); all other employees have the same
salary]
```

```
70      *          </pre>
71      */
72     public static void giveRaise(java.util.Map<String, Integer>
   map,
73             char initial, int raisePercent) {
74
75         for (java.util.Map.Entry<String, Integer> x :
   map.entrySet()) {
76             if (x.getKey().charAt(0) == initial) {
77                 x.setValue(x.getValue() * (initial));
78             }
79         }
80     }
81 }
```