

2. This question involves the design of a class that will be used to produce practice problems. The following `StudyPractice` interface represents practice problems that can be used to study some subject.

```
public interface StudyPractice
{
    /** Returns the current practice problem. */
    String getProblem();

    /** Changes to the next practice problem. */
    void nextProblem();
}
```

The `MultiPractice` class is a `StudyPractice` that produces multiplication practice problems. A `MultiPractice` object is constructed with two integer values: *first integer* and *initial second integer*. The first integer is a value that remains constant and is used as the first integer in every practice problem. The initial second integer is used as the starting value for the second integer in the practice problems. This second value is incremented for each additional practice problem that is produced by the class.

For example, a `MultiPractice` object created with the call `new MultiPractice(7, 3)` would be used to create the practice problems "7 TIMES 3", "7 TIMES 4", "7 TIMES 5", and so on.

In the `MultiPractice` class, the `getProblem` method returns a string in the format of "*first integer* TIMES *second integer*". The `nextProblem` method updates the state of the `MultiPractice` object to represent the next practice problem.

The following examples illustrate the behavior of the `MultPractice` class. Each table shows a code segment and the output that would be produced as the code is executed.

Example 1

Code segment	Output produced
<code>StudyPractice p1 = new MultPractice(7, 3);</code> <code>System.out.println(p1.getProblem());</code>	7 TIMES 3
<code>p1.nextProblem();</code> <code>System.out.println(p1.getProblem());</code>	7 TIMES 4
<code>p1.nextProblem();</code> <code>System.out.println(p1.getProblem());</code>	7 TIMES 5
<code>p1.nextProblem();</code> <code>System.out.println(p1.getProblem());</code>	7 TIMES 6

Example 2

Code segment	Output produced
<code>StudyPractice p2 = new MultPractice(4, 12);</code> <code>p2.nextProblem();</code> <code>System.out.println(p2.getProblem());</code> <code>System.out.println(p2.getProblem());</code>	4 TIMES 13 4 TIMES 13
<code>p2.nextProblem();</code> <code>p2.nextProblem();</code> <code>System.out.println(p2.getProblem());</code>	4 TIMES 15
<code>p2.nextProblem();</code> <code>System.out.println(p2.getProblem());</code>	4 TIMES 16

Question 2 continues on page 10.

Interface information for this question

```
public interface StudyPractice
```

```
String getProblem()
```

```
void nextProblem()
```

Write the complete `MultiPractice` class. Your implementation must be consistent with the specifications and the given examples.