**Problem details:**

There are multiple log files that need to be processed by an EventCounter class. EventCounter class must be implemented as per IEventCounter interface. Each file contains date time and stage value per line. Consecutive log entries with same stage value are considered duplicates. Stage values can range from 0 to 3. Both methods in EventCounter (ParseEvents() and GetEventCount()) should be thread safe.

There are four statements mentioned in the assignment that determines if a sequence is faulty or not:

1. Stage 3 for five minutes or more

2. Stage 2,

3. Any number of cycles between stage 2 and 3 for any duration

4. Stage 0

It means the following sequence should be deemed faulty.

Case 1:

2001 - 01 - 01 22:24:00 3

2001 - 01 - 01 22:29:00 2

2001 - 01 - 01 22:37:00 0

Case 2:

2001 - 01 - 01 22:24:00 3

2001 - 01 - 01 22:29:00 2

2001 - 01 - 01 22:37:00 3

2001 - 01 - 01 22:38:00 0

Case 3:

2001 - 01 - 01 22:24:00 3

2001 - 01 - 01 22:29:00 2

2001 - 01 - 01 22:37:00 3

2001 - 01 - 01 22:38:00 2

2001 - 01 - 01 22:38:00 0

Case 4:

2001 - 01 - 01 22:24:00 3

2001 - 01 - 01 22:29:00 2

2001 - 01 - 01 22:37:00 3

2001 - 01 - 01 22:38:00 2

2001 - 01 - 01 22:39:00 3

2001 - 01 - 01 22:40:00 0

**Problem objective:**

Find and count faulty sequences in the log entries.

**Assumptions:**

* Each file represents data for one device.
* An exact log entry appearing more than once is considered duplicate and should be ignored.
* Empty log lines are ignored.
* Incorrect stage values (not between 0-3) are ignored.
* Invalid data in log files is ignores. Such as incorrect date format or stage number.
* For condition: “Any number of cycles between stage 2 and 3 for any duration”. Based on this statement, if n = "any number of cycles between 2 and 3" then do you mean n >= 0. It means that we may have zero cycles, or we may have up to n cycles between 2 and 3.

**Instructions for reviewing:**

* For organic testing, please run the RCAssignment.exe file, located in the output folder, and follow instructions on the screen. The application has a component that generates sample input files for you. It allows you to manually verify the results computed against randomly generated data.
* For unit testing, please review the EventCounterTests.cs file.
* To run your own set of unit tests, obtain an instance of EventCounter class like this: IEventCounter eventCtr = EventCounter.Instance;