**Deliverables**

glm98Project1.java - Main project source code. It contains 4 classes, all in the same file for compilation convenience. Classes are as follows:  
Main (Public) Class: glm98Project1

Contains the bulk of the project’s source. Has all the I/O, as well as the source for the solutions to problems 1 and 2.

Helper Classes:

Event:

Represents an Event, namely a server going down or being repaired.

EventComparator:

Simple comparator class to allow for sorting of Events.

Process:

Represents as process, with a pid, arrival time, and requested service time

**Results**

For problem 1 all 1000 tuples are available for viewing. Once every tuple has been printed, the actual arrival rate and service time are displayed. **These are always very close to their expected values of 2 and 1 respectively, only rarely drifting by more than .001 or so.** e.g., for one run of the program this was the result:

Actual average arrival rate: 1.96149207787269

Actual average service time: 1.0457644529293646

For problem 2, there are enough instances of the servers going down and being repaired that the first chunk of the results are not available in the terminal’s history when scrolling through it. What is visible shows the timestamp of the exact hour the system went down or was repaired. The program will also display when the first occurrence of a total failure happens, though this is usually cut off by the terminal because it happens sufficiently early. Once the simulation has concluded, the average time to full system failure across all runs is displayed, and the user is prompted to restart, if so desired.

There seemed to be a lot more variance displayed in the time between failures on this problem. Sometimes it seemed near instant, other times it would be well into the thousands of hours before a failure. I believe a Uniform distribution is not the best choice to model this. A Normal distribution would probably be more appropriate.