**Coding Exercise:**

Consider a monitoring system, which monitors 1000 servers. Each server has 2 CPUs. Each server generates a log for CPU usage every minute.

The format is like this:

timestamp IP cpu\_id usage

1414689783 192.168.1.10 0 87

1414689783 192.168.1.10 1 90

1414689783 192.168.1.11 1 93

1.      Please write a simulator to generate the logs for one day, say 2014-10-31, just use random numbers between 0% to 100% as CPU usage. The generator should write data to files in a directory.  The timestamp is Unix time.

2.      Please create a command line tool which takes a directory of data files as a parameter and lets you query CPU usage for a specific CPU in a given time period. It is an interactive command line tool which reads a user’s commands from stdin.

3.      The tool may take several minutes to initialize, but the query result should be returned within 1 second.

4.      The tool should support two commands:

·  One command will print results to stdout. Its syntax is QUERY IP cpu\_id time\_start time\_end. Time\_start and time\_end should be specified in the format YYYY-MM-DD HH:MM where YYYY is a four digit year, MM is a two digit month (i.e., 01 to 12), DD is the day of the month (i.e., 01 to 31), HH is the hour of the day, and MM is the minute of an hour.

·  The second command to support is EXIT.  It will exit the tool.

E.g.

To run the generator:

./generate.sh DATA\_PATH

To run the interactive query tool:

./query.sh DATA\_PATH

>QUERY 192.168.1.10 1 2014-10-31 00:00 2014-10-31 00:05

CPU1 usage on [192.168.1.10](http://192.168.1.10/):

(2014-10-31 00:00, 90%), (2014-10-31 00:01, 89%), (2014-10-31 00:02, 87%), (2014-10-31 00:03,  94%) (2014-10-31 00:04, 88%)

>QUERY 192.168.1.12 0 2014-10-31 00:00 2014-10-31 00:05

CPU0 usage on [192.168.1.12](http://192.168.1.12/):

(2014-10-31 00:00, 90%), (2014-10-31 00:01, 89%), (2014-10-31 00:02, 87%), (2014-10-31 00:03,  94%), (2014-10-31 00:04, 88%)

>EXIT

Please note:

1.      Go through the requirements thoroughly, and follow the requirements.

2.      You may use C, C++, Java or a scripting language like Perl, Python or PHP.

3.      Try to use C++ or Java to implement. This will demonstrate your CS knowledge in the code, such as data structures, algorithms and performance tuning.

4.      Make the program look clear and professional rather than a college assignment, e.g. Add comments (JavaDoc or Doxgen format preferred), design document and user manual.

5.      Please provide the code, along with documentation and project file to compile it if needed, and an output file demonstrating its operation.

6.      Tell us how much time you spent on the exercise.

7.      It is not necessary to submit the exercise quickly. You need to do enough testing to make sure all the code works well and is bug free.

8.      Please provide the code, along with documentation and project file to compile it if needed.

9.      Do not send us the output file of your code. We can generate the output file by running your code