Omnicom Balfour Beatty
Java Programming Test

You will have received two CSV files as part of this test.

Get as far through the task as you can in the time you have available and submit what you have managed to complete. Please indicate how much time you spent on the task.

Part 1 – Load the files

Write a Java 8 application using maven as the build system

The application should load the two csv files into SQLite database tables named IMAGES and POSITIONS.

Assume the CSV files may be too large to fit into memory.

Column names should be based on the headers in the CSV files.

Take care not to lose any precision.

You may use open source CSV processing libraries in your pom.

Part 2 – Times

Add a time column to the IMAGES table and populate it (from Java) using the Hour, Minute, Second and Frame values. There are 25 frames per second.

Add a datetime column to the POSITIONS table and populate it (from Java) from the Time values. The Time column values are in dotnet ticks.

Please indicate how you tested the calculation of times.

Part 3- Position the Images

The positions and images are recorded together but at different rates. As such there is not a one-to-one correspondence between position records and image records. We need to find the location at which each image was captured.

Derive a position for each image and store it in the IMAGES table (Latitude and Longitude).

A suggested algorithm for this is to

1. See if there is a position record that matches the time exactly and use that position directly.
2. If not, find the two positions either side of the image time and interpolate between the points to estimate the image latitude and longitude. Feel free to assume that the earth is flat.
3. If the image is out of the time range of the position data then set the latitude and longitude to null.

Feel free to devise another algorithm if you feel it will be more efficient.

Please indicate how you tested the performance and validity of your algorithm and implementation.

Submission

Please send your source code including pom and your populated SQLite database.

We will be looking at comments, Javadoc, variable names and code layout as well as technical aspects.