Programming Tasks

You are given two programming tasks and are expected to finish them in one hour. By the end of the hour, I will check in with you on your progress. At this point, you should email me the code and libraries you used (I need to be able to compile your program). If you haven't finished the tasks, you have the option of getting an agreed-up extension (up to 30 minutes) or you can simply opt out and submit whatever you have done at that point.

The programming language must be Java and you can use whatever resources and/or open-source libraries you desire as long as I have access to those to validate your programs.

Your performance will be judged according to:

- correctness of your program according to the specifications
- readability of your program
- how much you have finished the tasks
- creativity of your approach.

Task #1:

Suppose we have an algorithm that extracts company names out of text documents. It is not perfect in the sense that it will extract the a company name as well as some descriptive terms in the sentence; however, we are guaranteed that the full company name will be extracted and it will be located in the end of the extracted sentence. For instance, it will output something like

```
Financially troubled property developer Lai Sun Development Co.
Consumer goods exporter and trading firm Li & Fung Ltd.
CEO Lee said Friday that its parent company, Lippo Ltd.
Annual passenger volume for Hong Kong railway operator MTR Corp.'s
A special statement from the CEO of Bank of America.
```

The underlined phrases are company names (please refer to the attached input file for more examples). Your task is to write a program that strips all the descriptive terms and outputs the company names from the output of this algorithm.

Input: a file with a sentence in each line.
Output: a file in which each line consists of two fields separated by '\t'. The first field is the original sentence; the second field is the company name in the original sentence

Task #2:

You are given a customer rating data file. A customer can choose to rate a particular item (say a string) with a set of possible ratings (say, 1-5). The fields in the log file include:

timestamp (to the milli-second, in UTC), ***item***, ***rating***

An exemplary log file would look like this;

```
2013-02-21 00:53:45.039 MjAxMzAyMjIwMTUxNjM= 1
2013-02-21 02:13:41.132 MjAxMzAyMjIwMTUxNjM= 3
2013-02-21 10:54:23.210 MjAxMzAyMjIwMTUxNjQ= 4
2013-02-22 21:54:48.796 MjAxMzAyMjIwMTUxNjU= 3
2013-02-23 00:56:03.679 MjAxMzAyMjIwMTUxNjY= 1
2013-02-23 04:56:38.813 MjAxMzAyMjIwMTUxNjY= 4
2013-02-23 05:23:12.123 MjAxMzAyMjIwMTUxNjY= 1
2013-02-23 08:57:46.586 MjAxMzAyMjIwMTUxNjY= 5
…
```

please refer to the attached input file for more examples.
Your task is to write a program that generate a daily and weekly report on the ratings. Specifically, you will generate something like this:

```
Date       Item                 # of 1 # of 2 # of 3  # of 4  # of 5
2013-02-21 MjAxMzAyMjIwMTUxNjM=   1      0       1       0       0
2013-02-21 MjAxMzAyMjIwMTUxNjQ=   0      0       0       1       0
2013-02-22 MjAxMzAyMjIwMTUxNjU=   0      0       1       0       0
2013-02-23 MjAxMzAyMjIwMTUxNjY=   2      0       0       1       1
```

Specifications:

- two output files: one for daily report and another is for weekly report
- A day is defined by 00:00:00.000 EST to 23:59:59.999 EST. (hence you need to convert the time from UTC to EST)
- In the weekly report, the Date field should be the last day of the week (in trading, it's Friday).

Input: a file with three fields, separated by '\t', in each line.
Output: two files in which each line consists of 7 fields separated by '\t' (as the example above).