CSCI 330 Database Systems
Assignment 1 (Java Exercise: Data Mining)
Total Point: 20 (5% of course grade)
Due: Monday, Jan 29, 2017 (9:00 pm)

Goals of the Assignment:

The purpose of this assignment is to:

1. understand how to process data in a regular text file using Java (this is a simple example of data mining) and
2. make sure that you did not forget how to use Java ☺

Description of Input File and Problem Statement

The name of the input file is: **Stockmarket-1990-2015.txt** which is available in the Assignment section of your Canvas. This input file contains stock information for the last 25 years (1990-2015) for the following seven big technology companies:

1. Apple (AAPL)
2. Facebook (FB)
3. Google (GOOG)
4. International Business Machine (IBM)
5. Intel Corporation (INTC)
6. Microsoft (MSFT)
7. Yahoo (YHOO)

This data is publicly available. The data is sorted by ticker symbol and date.

Each line of **stockmarket-1990-2015.txt** file contains the data for a particular stock (ticker symbol) for one day. The values are separated by tab('\t').

Here is an example from the data for Microsoft (ticker symbol MSFT):

```
MSFT  8/18/2004   26.93  27.5   26.89  27.46  58844000    19.46
MSFT  8/17/2004   27.22  27.38  26.98  27.05  56879700    19.17
MSFT  8/16/2004   27.03  27.2   26.96  27.09  54347200    19.19
MSFT  8/13/2004   27.01  27.25  26.98  27.02  43333200    19.14
MSFT  8/12/2004   27.23  27.31  26.86  26.88  50279700    19.05
MSFT  8/11/2004   27.39  27.51  27.2   27.41  53097300    19.42
```

There are eight fields in each line. A description of each value of the first line (yellow marked) is given below. All other lines of this example as well as the entire **Stockmarket-1990-2015.txt** will have the similar meanings.

1. ticker symbol (MSFT),
2. date (8/18/2004),
3. opening price (26.93),
4. high price (27.50),
5. low price (26.89),
6. closing price (27.46),
7. volume or number of shares traded on that day (58844000 shares), and
8. adjusted closing price (19.46).

The opening and closing prices are the price of the first and last trade, respectively, on the given date. The high and low prices are the highest and lowest prices of any trade during the day. The adjusted price includes corrections to the price to make comparison across days easier.

What to do

1. For each company, identify the **crazy days** in terms of stock price fluctuation. Here is how to calculate the crazy days.
Suppose at Date D, the high price of the share of company X is Hx and Low price is Lx. Then D will be called crazy day if

$(Hx-Lx)/Hx >= 15\%$

Let's consider the example below (yellow marked) which is taken from **Stockmarket-1990-2015.txt.**

YHOO	4/5/2000	162	169.88	158.5	165.56	27371800	82.78
YHOO	4/4/2000	165	171	132.75	167.38	42528400	83.69
YHOO	4/3/2000	168.75	173	159.38	160.13	19322800	80.06

Notice that, for 4/4/200, the high price of share for Yahoo was $171 and low price was $132.75. The fluctuation of stock price is (171-132.75)/171=22.37% which is more than 15%. Hence, 4/4/2000 is considered as a crazy day for Yahoo.

For this assignment, identity all crazy days for each company. Also, identify the **craziest day** for that particular company i.e. in which day of the entire history of stock price, the stock price fluctuated most.

2. Stock split:
Occasionally, a stock may split by some ratio, say 2-for-1. This means that every old share is replaced by some number of new shares, in these case two. Each new share is worth some fraction of the value of the old share. In this case, each is worth half the old value. Thus, the value of each person's holding will remain unchanged.

We recognize such a stock split in the data when the closing value of day "x" is twice the opening value on day "x + 1", the next trading day. There is a complication at this point because the price of the first trade of the new day does not have to be the same as the price of the last trade of the prior day. The new opening price does not have to be exactly half of the previous day's closing price.

Let's consider the example below (yellow marked) which is taken from **Stockmarket-1990-2015.txt.**

AAPL	3/1/2005	44.99	45.11	44.16	44.5	117047000	5.94
AAPL	2/28/2005	44.68	45.14	43.96	44.86	162902600	5.99
AAPL	2/25/2005	89.62	89.91	88.19	88.99	228877600	5.94
AAPL	2/24/2005	88.48	89.31	87.73	88.93	379757000	5.94

Notice that the closing price on 2/25/2005 was \$88.99, and the opening price on 2/28/2005 was \$44.68, giving a ratio of 88.99/44.68 = 1.991, which is slightly lower than 2. Nevertheless AAPL stock experienced a 2:1 stock split on 1997.04.02, and the small variation is due to the initial trading activity on 2/28/2005as described above.

To accommodate such variations, we use the following definition, where C_x is the closing price on day x and O_{x+1} is the opening price on the next trading day x + 1.

Definitions:

- A 2:1 stock split occurs if $| C_x / O_{x+1} - 2.0| < 0.05$
- A 3:1 stock split occurs if $| C_x / O_{x+1} - 3.0| < 0.05$
- A 3:2 stock split occurs if $| C_x / O_{x+1} - 1.5| < 0.05$

Reminder: $|x|$ represents the absolute value of the number x.

For this assignment, you have to detect all splits of ratios 2:1, 3:1, and 3:2 using the above formulas.

Note that "next day" when computing the closing/opening ratio is the next trading day, which may not be the next day on the calendar due to weekends, holidays, etc. For this assignment, adjacent lines in **Stockmarket-1990-2015.txt** represent adjacent trading days.

Output

Your java program should read **Stockmarket-1990-2015.txt** and output all splits, as defined above, to System.out. Your output should begin like this:

```
Processing AAPL
====================

Crazy day: 8/24/2015      15.44%
Crazy day: ......................................
Crazy day: ......................................
……………………………
Total crazy day = N
The craziest day: 1/6/1998 26.25%


2:1 split on 2/25/2005       88.99 -->   44.68
2:1 split on 6/20/2000      101.25 -->   50.50
Total number of splits: 2

Processing FB
===========================

Crazy day: ………………………………
Crazy day: ......................................
Crazy day: ......................................
……………………………
Total crazy day = …….
The craziest day: ………………………..


2:1 split on ………….…………………………..
3:1 split on ……………………………...
3:2 split on ……………………………...
Total number of splits: …………………

etc.
```

You are allowed to vary the spacing. Otherwise, your output should look exactly like the example.

Note that the split is attributed to the day with the higher closing price. The two prices are the closing price for the day of the split and the opening price for the next trading day. Stockmarket-1990-2015.txt is ordered in **descending** chronological order which means that the data for tomorrow comes before the data for today.

Good Design and Style

Although bad design and style won't hurt the performance of your program, it is utmost important, especially when you will work in the real world with other professionals, to follow a good design and style for coding. It will be helpful not only for your colleagues to understand your code better but also for you as well when you will revisit your code later.

Here are some guidelines for good design and style:

- The program is logically organized so that a reader can quickly and easily understand the program structure.
- Functions and variables have meaningful names.
- The program has proper indentation.
- The program follows object oriented approach.

Submission Instructions

- You only need to submit the source code.
- The file name which contains main method should be **Assignment1.java**.
- Do not submit the input file. It will be assumed that the input file will be in the same directory of the program file. Write the file path in your java program in that way.
- If you have more than one java files:
  - Please keep all java files in a single folder.
  - Folder name should be your last name.
  - Zip the folder and name it Assignment1.zip.
- Upload the Assignment1.java/ Assignment1.zip on canvas.

Grading (Total Point: 20)

Your program will be graded in the following way:

- 16 – The program gives the correct output.
- 2 – The program has good design and style.
- 2 – The output is correctly formatted.

Late Policy:

- No late work will be accepted.

Academic Dishonesty

- Copying files from someone else and claiming they are your own is plagiarism.
- Providing files that you created to another student or being party to such actions also amounts to academic dishonesty.

Frequently Asked Questions

1. Should we provide the input file in canvas?
   Answer: No. Assume that the input file already exists in the same folder.

2. Can input file have more than seven companies?
   Answer: Your program should work for any number of companies.

3. I have two java files that define objects and one java file that runs the main method. Is it OK?
   Answer: Yes.

4. Data in Input file follows the descending order. Should we follow the same order in the output file?
   Answer: Yes, for both crazy day and stock split.

5. What does dot (…) mean in the output?
   Answer: It is like a placeholder. Therefore, if there is no data then DO NOT print anything.

6. My output is not printing each company's data in alphabetical order. For example, will I lose points for YHOO records being printed before FB, etc.?
   Answer: Your output must print each company's data in alphabetical order. Points will be deducted if YHOO records being printed before FB.

7. Do you want us to print out the splits as they occur chronologically, or would you prefer that we print out all the 2:1 splits first, then the 3:1 splits, and then the 3:2 splits?
   Answer: Chronologically (recent to old days as they are in the input file)

8. If there are no splits do I print 0 splits?
   Answer: Print like this
   Total number of splits: 0

9. If there are no crazy days do I print 0 crazy days?
   Answer: Print like this
   Total crazy days = 0

10. Is there still a craziest day if there are no crazy days?
    Answer: No.