CS 330: Database Systems
Winter, 2014
Program 1: Java Exercise (Data Mining)
Due January 31, 23:59

Each line of stockData.txt contains eight string fields, separated by semi-colons. The string values represent price and volume data for a particular stock (ticker) on a particular day. From left to right, the fields are:

1. ticker
2. date
3. opening price
4. high price
5. low price
6. closing price
7. volume
8. adjusted closing price

For example, the first line is as follows.

```
IBM;2014.01.02;       187.21;       187.40;       185.20;       185.53;      4546500;       185.53
```

It means that on January 2, 2014, the stock for International Business Machines (IBM) opened at $187.21 and closed at $185.53. During the trading session, it touched a high value of $187.40 and a low point of $185.20 A total of 4,546,500 shares were exchanged during the day. The last entry is the adjusted closing price, which includes corrections so that the closing price of a given day may be more easily compared with the closing price of the most recent day. In this example, the data are the most recent day available for IBM, and so there is no correction. In this assignment, we will ignore the last two entries of each line.

All data for a particular ticker are grouped in consecutive lines in *reverse* chronological order. For example, the first few lines for IBM are as follows.

```
IBM;2014.01.02;       187.21;       187.40;       185.20;       185.53;      4546500;       185.53
IBM;2013.12.31;       186.49;       187.79;       186.30;       187.57;      3619700;       187.57
IBM;2013.12.30;       185.32;       186.70;       184.67;       186.41;      3018600;       186.41
IBM;2013.12.27;       185.84;       186.50;       184.56;       185.08;      3381600;       185.08
IBM;2013.12.26;       183.56;       185.56;       183.51;       185.35;      3325700;       185.35
IBM;2013.12.24;       181.96;       183.50;       181.91;       183.22;      1613600;       183.22
IBM;2013.12.23;       181.05;       182.93;       180.61;       182.23;      4079900;       182.23
```

Occasionally, a stock may split by some ratio, say 2:1. This means that every outstanding share is replaced with two new shares, each worth half the value of the old share. Consequently, the value of each person's holdings remains unchanged. We recognize such a stock split in the data when the closing value of day x is twice the opening value on day $x + 1$. There is a complication at this point because some initial trading occurs at or near the new opening price before the market actually opens. Specifically some buy/sell orders that are queued for trading can be matched such that each trade occurs a price at least as good as that specified by either party. Such trading moves the new opening price slightly so that it may not be exactly half of the previous day's closing price. For example, consider this short segment of the IBM data.

```
IBM;1999.05.28;       116.00;       116.75;       114.19;       116.00;      6379500;        97.42
IBM;1999.05.27;       116.69;       116.87;       112.62;       116.00;     10552500;        97.42
IBM;1999.05.26;       223.00;       236.63;       221.44;       236.25;     16628000;        99.21
IBM;1999.05.25;       222.50;       226.00;       221.00;       221.19;      9405000;        92.88
```

Notice that the closing price on 1999.05.26 was \$236.25, and the opening price on 1999.05.27 was \$116.69, giving a ratio of $236.25/116.69 = 2.0246$, which is slightly greater than 2. Nevertheless IBM stock experienced a 2:1 stock split on 1999.05.26, and the small variation is due to the initial trading activity on 1999.05.27 as 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 day $x + 1$.

Definitions:

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

Not all splits are 2:1. For this assignment, you are to detect all splits of ratios 2:1, 3:1, and 3:2. The latter two types are detected by replacing the 2.0 in the above criterion by 3.0 and 1.5 respectively, as noted in the definitions above.

For this assignment, you are to write a Java program that reads the data file stockData.txt and reports all splits directly to the console. The output should commence

```
Processing IBM...
2:1 split on 1999.05.26      236.25 -->  116.69
2:1 split on 1997.05.27      179.25 -->   90.50
splits: 2

Processing INTC...
```

and continue in this format for the remaining tickers in the file. **Note that the split is attributed to the day having the high closing price**. The two reported prices are the closing price for the day of the split and the opening price for the next *chronological* trading day. That is, the first entry above means that IBM closed at \$236.25 on 1999.05.26 and opened at \$116.69 on the next *trading day*, which in this case was 1999.05.27, but may actually be more removed from the closing day if a holiday or weekend has intervened. Also, remember that the file is ordered in *descending* chronological order, which means that tomorrow is read before today.

Export your project as a .zip file, following the instructions in the Computer Issues download. Give the file whatever name you wish, such as Assignment1, or your last name, or SplitSeeker. Email the .zip file to the instructor before the deadline.