First Homework Assignment (5% of grade)

Due Date: **4/14/2017** at **5pm**

Submission Guidelines:

Please follow carefully the instructions posted on the course's github page when submitting your solutions (https://github.com/MSIA/bigdatacourse/blob/master/README.md). Failure to follow the instructions will result in lost points.

Deliverables:

1. Your java source code file: name the file "exercisex.java" where x is either 1 or 2. **NOTE: the class name has to be "exercisex.java" as well.**
2. One output file from a reducer: name the file "exercisex.txt" where x is either 1 or 2.

Notes/tips:

1. All input files are available in /home/public/course. Please copy them directly from /home/public/course to HDFS and not to your home directory on wolf.
2. The directory /home/public/setProject has directions on how to set up a maven project. You can use the same directions and pom file on your personal laptop if you wish to compile there.

Problem 1

Data: /home/public/course/temperature/

Description: Maximum temperature per year: You have two files with temperature readings from various weather stations (if you put them into the same folder in HDFS and run your MapReduce job with that folder as input, it’s going to automatically read both files and pass all rows to the mappers).

You need to write a MapReduce job in java that will calculate the maximum temperature for each year. The output should be pairs (year, temperature).

Example: This is a single record/line in the data (the data is from the NOAA web site):

```
0029029070999991901010106004+64333+023450FM-
12+000599999V0202701N015919999999N0000001N9-
00781+99999102001ADDGF108991999999999999999999
```

- **Year** = positions 15-19
  - In this example: 1901
- **Temperature** = positions 87-92
  - In this example: -0078
  - If Temperature = 9999, it should be interpreted as missing value – i.e. it can be ignored.
- **Temperature quality** = positions 92-93.
  - If Temperature quality is not equal to any one of 0, 1, 4, 5, or 9, then the temperature should be interpreted as missing value – i.e. it can be ignored.

Problem 2

Data: /home/public/course/IBM.csv

Description: The second data set comes from IBM and it is a machine learning classification problem. It’s in csv format. All columns are numeric, except from the last one which is either ‘true’ or ‘false’. Your task is to find the average value of column 4 for all different combinations of columns 30,31,32,33, where the last column is equal to ‘false’.

This exercise is equivalent to the following SQL query (ibm[i] is the i-th column):

```
SELECT AVG(ibm[4]), ibm[30], ibm[31], ibm[32], ibm[33] FROM ibm

GROUP BY ibm[30], ibm[31], ibm[32], ibm[33]

WHERE ibm[last column] = ‘false’
```

Example output: The final result be in the following format (columns 30-33 only take binary values):

```
0,0,0,0, average value of column 4
1,0,0,0, average value of column 4
…
1,1,1,1, average value of column 4
```