Exercise #5

- Average
  - Input: a collection of (structured) textual csv files containing the daily value of PM10 for a set of sensors
    - Each line of the files has the following format:
      `sensorid, date, PM10 value (µg/m³)
  - Output: report for each sensor the average value of PM10

Exercise #6

- Max and Min
  - Input: a collection of (structured) textual csv files containing the daily value of PM10 for a set of sensors
    - Each line of the files has the following format:
      `sensorid, date, PM10 value (µg/m³)
  - Output: report for each sensor the maximum and the minimum value of PM10
Exercise #7

- Inverted index
  - Input: a textual file containing a set of sentences
    - Each line of the file has the following format
    
    sentence1|sentence2|sentence3
    
    - Output: report for each word w the list of sentences containing w
      - Do not consider the words "and", "or", "not"

Exercise #7 - Example

- Input file
  - Sentence#1: Hadoop Spark
  - Sentence#2: Hadoop Spark and Java
  - Sentence#3: Hadoop and Big Data

- Output pairs
  - (hadoop, [Sentence#1, Sentence#2, Sentence#3])
  - (spark, [Sentence#1, Sentence#3])
  - (java, [Sentence#2])
  - (big, [Sentence#3])
  - (data, [Sentence#3])

Exercise #8

- Total income for each month of the year and Average monthly income per year
  - Input: a (structured) textual csv files containing the daily income of a company
    - Each line of the files has the following format
    
    date|daily income
    
  - Output:
    - Total income for each month of the year
    - Average monthly income for each year

Exercise #8 - Example

- Input file
  - 2015-11-01  1000
  - 2015-11-02  1305
  - 2015-12-01  500
  - 2015-12-02  750
  - 2016-01-01  345
  - 2016-02-02  1445
  - 2016-02-03  200
  - 2016-02-04  500

- Output
  - (2015-12-1250)  (2015-12-1250)
  - (2016-01-1450)  (2016-01-1450)
  - (2016-02-700)   (2016-02-700)