New York University
School of Continuing and Professional Studies
Division of Programs in Information Technology

Introduction to Python
Homework, Session 6

6.1 Sorting: given the lines showing Open and Close stock prices in stock_prices.csv, attempt to discover the "biggest one-day gainers" for the year by sorting the file by the difference between Close and Open prices (i.e., Close minus Open, indicating how much higher the Closing price was over the Opening price) on a given line.

Starting with this code:

```
stock_price_file = '../python_data/stock_prices.csv'
lines = open(stock_price_file).readlines()[1:]
for line in sorted(lines, key=by_closelessopen, reverse=True)[0:5]:
    line = line.rstrip()
    ticker, date, open, high, low, close, volume = line.split(',')
    print('{} ({}):  {} ({}->{})'.format(ticker,
                                         date,
                                         (round(float(close)-float(open), 2)),
                                         open,
                                         close))
```

Your code will only implement the by_openclose function. Please see the homework discussion for details.

Special note: a common misconception is that the sort function should accept the list of elements as an argument and return a sorted list of elements. Sort funtions don't work with a list. Instead, they take one value (an item to be sorted) and return one value (the value by which it should be sorted), not a list of values. The returned value is the "value by which this element should be sorted" in order to inform the sorting that sort() will do.

Expected output (the display below is accounted for in the starter code; you need only supply the by_openclose function:

```
GOOG (17-Jul-15):  23.93 (649.00->672.93)
GOOG (30-Jan-15):  18.66 (515.86->534.52)
GOOG (26-Aug-15):  18.27 (610.35->628.62)
GOOG (24-Aug-15):  16.61 (573.00->589.61)
GOOG (16-Jul-15):  14.73 (565.12->579.85)
```

6.2 Read a config from a multidimensional structure and print out in a readable format. For this assignment we'll use the config.py file from the data directory on the website (note: this file appears as config.py_save_as_py; please save it as config.py, i.e. removing the _save_as_py extension). Place this file in the same directory as your script.

Starting with this code:

```
from config import conf
```

conf is a list of dicts containing the config data. Loop through each dict and print out the info as indicated. Use spaces and empty print statements to organize elements as shown.

You can begin by printing conf itself. You'll see it printed out as a glob of text that's hard to read, but make sure to view the file in the browser when you first click on it to show you what you're working with here.

Expected Output:

```
domain:  www.example1.com
db_host:  localhost1
db_port:  27017
plugins:
  plugin1
  eslint-plugin-plugin1
  plugin2
  plugin3


domain:  www.example2.com
db_host:  localhost2
db_port:  27017
plugins:
  plugin2
  eslint-plugin-plugin2


domain:  www.example3.com
db_host:  localhost3
db_port:  27017
plugins:
  plugin3
  eslint-plugin-plugin3
```