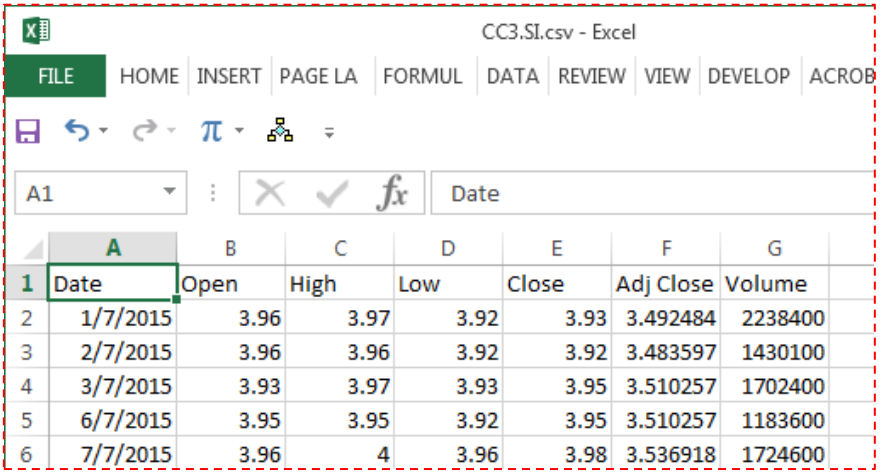
In the Python programs, we assume students will import Matplotlib, Pandas, Numpy and Scipy as the following:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

Write a Python program according to the following instructions.

(1) **Use one command with `pandas.read_csv`** to load data from the CSV file **`CC3.SI.csv`** (see the figure) to a DataFrame, using the **first column as the row labels**, the **first row as column names** and **parse the row index as dates**. Name this DataFrame as **`data`**.



	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Adj Close	Volume
2	1/7/2015	3.96	3.97	3.92	3.93	3.492484	2238400
3	2/7/2015	3.96	3.96	3.92	3.92	3.483597	1430100
4	3/7/2015	3.93	3.97	3.93	3.95	3.510257	1702400
5	6/7/2015	3.95	3.95	3.92	3.95	3.510257	1183600
6	7/7/2015	3.96	4	3.96	3.98	3.536918	1724600

```
data = pd.read_csv('CC3.SI.csv', index_col=0, parse_dates=True)
```

(2) **Use one command with `pandas.DataFrame.drop`** to drop the row in **`data`** whose "Volume" is zero.

```
data.drop(data.index[data['Volume']==0],inplace=True)
```

(3) **Use one command** to add a new column with the name "**`15d`**" to **`data`** which is the rolling mean of 15 (rounded to 3 decimal places) of the data in the column "**`Adj Close`**" computed by the library function **`pandas.Series.rolling`** (not **`pandas.DataFrame.rolling`**).

```
data['15d']=np.round(data['Adj Close'].rolling(15).mean(), 3)
```

(4) **Use one command** to add a new column with the name "**`50d`**" to **`data`** which is the rolling mean of 50 (rounded to 3 decimal places) of the data in the column "**`Adj Close`**" computed by the library function **`pandas.Series.rolling`** (not **`pandas.DataFrame.rolling`**).

```
data['50d']=np.round(data['Adj Close'].rolling(50).mean(), 3)
```

(5) **Use one command with the operator `-`** to compute the difference (stored as a Series with the same labels as those of **`data`**) between the data in the column "**`15d`**" and the data in the column "**`50d`**" (i.e. "**`15d`**"−"**`50d`**") within the same row. Name this the difference as **`x`**.

```
x=data['15d']-data['50d']
```

(6) **Use one command** to update the Series **x** by changing all the positive numbers to **1**.

```
x[x>0]=1
```

(7) **Use one command** to update the Series **x** by changing all the negative numbers and zero to **0**.

```
x[x<=0]=0
```

(8) **Use one command** to compute and store the first discrete difference of the elements in the Series **x** as a Series with name **y**.

```
y=x.diff()
```

(9) **Use one command** to find and store the labels of those elements in the Series **y** whose values are negative as **idxSell**.

```
idxSell=y.index[y<0]
```

(10)**Use one command** to find and store the labels of those elements in the Series **y** whose values are positive as **idxBuy**.

```
idxBuy=y.index[y>0]
```

(11)**Use one command** to add a new column with the name "**crossSell**" to **data** and make every value equal to the special number **NaN**.

```
data['crossSell']=np.nan
```

(12)**Use one command** to update the column "**crossSell**" in **data** in the rows where row labels are equal to **idxSell** using the values in the column "**Adj Close**" within the same row.

```
data.loc[idxSell,'crossSell']=data['Adj Close'][idxSell]
```

(13)**Use one command** to add a new column with the name "**crossBuy**" to **data** and make every value equal to the special number **NaN**.

```
data['crossBuy']=np.nan
```

(14)**Use one command** to update the column "**crossBuy**" in **data** in the rows where row labels are equal to **idxBuy** using the values in the column "**Adj Close**" within the same row.

```
data.loc[idxBuy,'crossBuy']=data['Adj Close'][idxBuy]
```

(15)**Use one command with pandas.DataFrame.plot** to plot the 5 columns of **data** ("**Adj Close**", "**15d**", "**50d**", "**crossSell**" and "**crossBuy**") using 5 different line styles (black solid line, blue solid line, cyan solid line, red circle and yellow circle) and linewidth **1**.

```
data[['Adj Close', '15d', '50d','crossSell','crossBuy']].plot(
ax=ax, style=['k-','b-','c-','ro','yo'], linewidth=1)
```

(16)**Use one command** to display the figure.

```
fig, ax=plt.subplots()
```