CS 101
Fall 2013
Program 5
Algorithm due Sunday, October 20
Program due Sunday, October 27

This program will involve file manipulation. Specifically, we're going to be processing some slightly-messy real-world data and look for patterns.

This data is taken from the Solar Influence Data Center (http://sidc.oma.be/sunspot-data/dailyssn.php), which is maintained by the Royal Observatory of Belgium. You're given a text file containing sunspot observation data going back to 1818. The format is:

Date[YYYYMMDD]   Date[YYYY.Fraction]   SunspotsObserved

Like many real-world data sets, there are some cases of missing data. Although there is an entry for most days, some days have a number for SunspotsObserved of 999. These cases should be excluded from analysis.

This data is also noisy; that is, there is an underlying pattern (sunspots vary in an 11-year cycle), with a large amount of random 'noise' (day to day variation) on top of it. In order to filter out the noise, we adjust or 'smooth' the data by averaging several values together; if the noise is normally-distributed, this will tend to cancel out the noise. It's not exact—the average error of several points may not be exactly 0—but the average will vary less than individual measurements.

For this program you'll carry out data processing in several steps:
1. From the data file you have, produce a new data file called MONTHLY.TXT that has the monthly average for each month, in the format:
Year[YYYY]   Month[MM]   Average(floating point value)

Be sure to omit any missing data before computing the monthly average.

2. From the monthly.txt file, produce smoothed data as follows:
a. For each month, take that month's average, the six months before it, and the six months afterward. The smoothed value is (0.5*first value in the list + 0.5*last value in the list + sum of all other values in the list) / 12. For this assignment, omit the cases near the very beginning and end of the file that don't have data for a full 6 months on either side of them.

b. Save the data in the same format as for the monthly-data file, in a new file called SMOOTH.TXT. This time you'll used the smoothed data rather than the monthly averages for the last column.

3. Produce a graph of the smoothed data (using pylab, or a spreadsheet such as Excel or LibreOffice). Based on the graph, estimate (by eye, not by your program) when the next peak in sunspot activity will occur; include it in a comment at the beginning of your program file.

Getting started:

- Do all the usual things—set up a file, etc.
- Break the problem down into steps. Given a date in YYYYMMDD form, how can you separate out each month's data? (Hint: All data coming in from the file is a string.)

- Remember that you can iterate through a file a line at a time, use readline() to read a line at a time, or readlines() to get a list of lines.
- The string.split() method can be used to break a large string up into a list of smaller strings; by default, it breaks on whitespace.
- The trickiest logic is probably going to be detecting when you've gone from one month's data to the next. It may be helpful to keep track of what the most recently processed line had for the month, to see if it matches the line you're dealing with. Or, since you know that months go from '01' to '12', you can use that.
- Look for places where you can break subtasks out into functions. You should use functional decomposition as appropriate throughout your program.
- Be sure that ALL the data, including the first and last month's, get written to the output files.

Sample output: Here are the first 5 lines of my MONTHLY.TXT and SMOOTH.TXT:

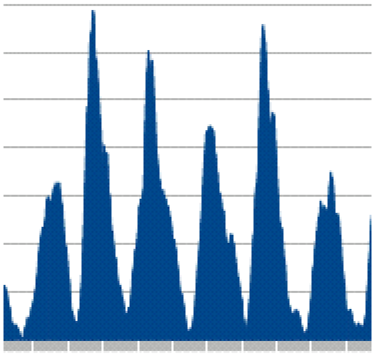
Monthly.txt:

```
1818 01 34.875
1818 02 22.428571428571427
1818 03 25.428571428571427
1818 04 34.523809523809526
1818 05 53.08
```

Smooth.txt:

```
1818 07 27.53084293021793
1818 08 27.588172753172756
1818 09 27.824680689680687
1818 10 26.344961918114084
1818 11 23.692991593752463
```

And here's what part of the chart of smoothed data should look like:



Note: You will NOT be graded on your artistic or layout choices regarding the chart. Just use the chart to estimate when the next peak will occur, and put your prediction in a comment in your source file.