CSE 231, Fall 2007

Programming Project 06

Assignment Overview
This assignment is worth 50 points (5.0% of the course grade) and must be completed and turned in before 11:59pm on Monday, October 15, 2007. The purpose of this project is to work more with lists, looping, and file I/O. For this assignment, you will work with real data from the Web to answer a question.

Question
Solar storms can damage satellites. When will the next damaging solar storms occur?

Background
Major solar storms are usually associated with peaks in sunspot activity, and sunspot activity is periodic. The next expected peak in sunspot activity is the next likely time for damaging solar storms. Sunspot data has been collected regularly for over 300 years, and the raw data is available on the Web. Early data collection was monthly, but after World War II daily data collection began, and that is the data we will use. Unfortunately, raw data is noisy so to do a reasonable prediction the data needs to be "smoothed" (explained below). The standard sunspot smoothing is based on monthly averages.

Program Specifications
You are to get the raw sunspot data from the Web, calculate the monthly averages, and then smooth the data. Looking at that data, predict the next year for solar storms.

Your program is to take no input (except for reading the file) and produces two files.

Sunspot Data Collection and Manipulation

We will use data from

```
ftp://ftp.ngdc.noaa.gov/STP/SOLAR_DATA/SUNSPOT_NUMBERS/AMERICAN_NUMBERS
```

From that site you are to copy this file to the folder/directory where your Python program is:

```
RADAILY.PLT
```

In the file, each line has data separated by spaces in this format:

```
year month day sunspotCount
```

Watch out that the final data of 2007 is missing, that is, there are lines with year, month, and day, but no sunspot data. Ignore those months.

(At that same Web location there are some useful files for checking your results. The location MOTHLY is a page with the monthly averages, and MONTLY.PLT is the corresponding data file. The location SMOOTHED is a page with smoothed monthly data, and SMOOTHED.PLT is the corresponding data file. That data should differ from yours only in the last digit—due to rounding errors. Unfortunately, there are 44 differences in the monthly average data, but the first difference doesn't appear until 1947 so data before that point is fine for comparison. As a challenge for the curious: can you find the 44 differences? I used a difference of greater than 0.2 to find the 44 differences.)

Your high level algorithm will be:

1. Read the raw sunspot data from RADAILY.PLT.
2. For each month find the average number of sunspots. Write the averaged data to a file named MONTHLY using a format similar to the original data. Each line will have:
   ```
   year month sunspotAverage
   ```
3. Smooth the monthly averages (using the algorithm provided below). Write the smoothed data to a file named SMOOTHED using the same format.
4. By eye (not with a computer) look at your data and predict the next year for damaging solar storms (peak sunspots). Write your prediction in a comment at the head of your program.

Smoothing Algorithm
The smoothing algorithm can be found at:

    http://www.ngdc.noaa.gov/stp/IONO/sunspot.html

The smoothing algorithm roughly works by taking six months of data on either side of a month and averaging. The algorithm has a strange twist in that the first and last months in the smoothing average only supply *half* their month's value. The detailed formula is provided on the Web page listed above.

Deliverables
You must use handin to turn in three files: **MONTHLY**, **SMOOTHED**, and **proj06.py** – this is your source code solution; be sure to include your section, the date, the project number and comments describing your code. Please be sure to use the specified file names, and save a copy of your proj06.py file to your H drive as a backup.

Assignment Notes

a. The hardest part of this project is calculating the monthly averages. Finding averages is not hard, but getting the details right is hard. I had the most problems at the transitions from month-to-month and year-to-year.
b. To keep track of months and years I had two variables that I initialized to '1945' and '01' before my loop that read in data. That is, my program had those two values hard coded into the program—you may do the same or similar, if you wish.
c. When smoothing the data you need to be careful to **not** include already-smoothed data in your calculation. (Depending on how you do your smoothing this may or may not be an issue.)
d. If you have a list of lists, be careful if you make a copy. (Depending on your algorithm you may or may not be making a copy.) A "shallow copy", e.g. L2 = L1, may not be what you want because making changes in the new copy will also

change the original (see List chapter). For example, for a list of lists, even L2 = L1[:] doesn't make a deep enough copy: you end up with a list of references to the original lists. You will need to do L2 = [x[:] for x in L1]. That is, do a deeper copy of each list.

e. For writing a file refer to the lecture slides. Remember that when writing to files you output strings (so you may need to use `str()`).