Individual Project 1: due Wednesday, January 18

Write a program which will perform analysis on data series received from deep space by a large radio telescope. The data is contained in an ASCII file, with two real numbers on each line: time and signal strength. There may also be comments, extending from a ‘#’ character to the end of the line. The time values are guaranteed to be in increasing order, and it is hypothesized that the signal strength is generally increasing linearly over time, but random noise interferes with this. In particular, certain random cosmic “storms” can introduce enough interference to make the data temporarily unusable. Your program will validate the data, and if it is found reliable, calculate coefficients of a regression line designed to most nearly fit the data.

Needless to say, it will do this using the C++ Standard Template Library as elegantly and efficiently as you can manage. The project must be completely your own work, except that you may have access to all of the code written by your team for the labs in this course, and of course you may confer with the instructor.

data validation

Begin by reading the data into a list. You cannot do this with copy, because it can only copy to a list with enough existing nodes to hold all the data, and your program does not know in advance how much data there will be. Instead, use the iterator form of the list constructor to initialize it from an input iterator bound to the data file.

The data is considered invalid if some signal strength differs by more than ±10% from the previous one, unless at least 600 seconds have passed since the previous one (due to a temporary equipment outage, for instance). Use the adjacent_find algorithm to test for this, or some other equally elegant STL mechanism.

finding the regression line

If the data does not validate, print messages pointing out the offending values and quit the program (using exit (EXIT_FAILURE) from <cstdlib>). But if the data is valid, perform an analysis to find the best-fitting regression line. Calculate the following statistics from the given data:

- n: The number of data points
- SumT: The sum of all the time values
- SumS: The sum of all the signal strength values
- SumST: The sum of the products of time times signal strength
- SumT2: The sum of the squares of the time values

Once these are computed, the equation of the best-fitting line is

\[ S = mT + b \]

where

\[
m = \frac{(n \text{ SumST} - \text{SumS SumT})}{(n \text{ SumT2} - \text{SumT}^2)} \]

\[
b = \frac{(\text{SumT2 SumS} - \text{SumT SumST})}{(n \text{ SumT2} - \text{SumT}^2)} \]

Use a single pass over the data using the for_each algorithm, or some other STL mechanism, to accumulate the statistics and then compute the equation of the line.

printing the results

Output the equation of the regression line, and then a list showing the input data, the signal strength predicted by the regression model, and the “noise”: the difference between the measured signal strength and the predicted. Again, use the STL for this part. End main with return EXIT_SUCCESS.

testing and submission

Test your program against data files you make up yourself, and against proj1a.dat, proj1b.dat, and proj1c.dat available by anonymous ftp from ftp.cs.csustan.edu/pub/rayz/cs3850/. Abbreviated output (the first ten lines) from the first two are also available in proj1a.out and proj1b.out, to help you check the accuracy of your work. Turn in a complete, commented program listing and complete printouts of sample runs using these data files.