Purpose

The purpose of this assignment is to have you work on a mathematical problem/analysis with single dimensional arrays, functions and the program split into multiple files (the main function file, a file containing all other functions, and a header file).

Scenario

When the relationship between two related quantities appears linear, a linear regression equation will give the best fit to that data. The context of this assignment is that you will compute a linear regression equation to fit a straight line to load deflection data for a mechanical coil spring. The computed equation will satisfy the least squares criterion. That is, it will minimize the sum of the squares of the deviations of the observed deflections from those predicted by the equation.

We start with a set of experimental data which plots the relationship between two quantities, the weight of a load on a spring and the corresponding compression of the spring, Fitting a curve to known data enables estimation of spring deflections for other loads not in the data. This curve can be the basis for calibrating a spring scale. The data consists of a collection of experimentally obtained pairs (x_i, y_i) with $i = 0, 1, 2, \ldots, n-1$. The x_i's (independent variables) are the weights; the y_i's (dependent variables) are the corresponding deflections. We want to find a (linear) function f such that $f(x_i) \approx f(y_i)$.

A) Least Squares Method

The method of least squares, a technique used for determining the equation of a straight line for a set of data pairs, will calculate coefficients m and b in the following linear regression equation:

$f(x) = mx + b$

When the original x_i values are substitued into the equation, estimate $f(x_i)$ values are obtained. For each x_i, the *residual* is defined to be the difference beween the observed y_i and the estimated $f(x_i)$. The the least squares method is where the **sum of the square of the residuals (r)** given by:

$$\sum_{i=0}^{n-1} (y_i - f(x_i))^2$$

will be minimized.

When the sum of squared residuals is computed, we can proceed to compute the correlation between the two related quantities by what is known as the Coefficient of Determination. This is a number typically between 0 and 1 where 1 indicates perfect correlation and 0 is least and numbers in between indicates the measure of correlation. To compute this we also need another number which is called the **total sum of squares (t)** given by:

$$\sum_{i=0}^{n-1} (y_i - \bar{y})^2,$$

The **Coefficient of Determination** is given by: **1 – (r/t)**

Algorithm

For the *n* data pairs (x_i,y_i) with $i = 0,, 1, ..., n-1$ the slope *m* and the intercept *b*, for the least squares linear approximation $f(x) = mx + b$ are calculated in the following way:

$$\begin{aligned}
s_x &= \sum_{i=0}^{n-1} x_i \\
s_y &= \sum_{i=0}^{n-1} y_i \\
s_{xy} &= \sum_{i=0}^{n-1} x_i y_i \\
s_{xx} &= \sum_{i=0}^{n-1} x_i^2 \\
m &= (s_x s_y - s_{xy} n)/(s_x^2 - s_{xx} n)) \\
b &= (s_y - s_x m)/n
\end{aligned}$$

After printing the information about the reading of the data, and computing the *m* and *b* values, your program will print the regression equation followed by the original x_i, y_i, estimated $f(x_i)$ values, residuals and correlation.

Next we want to compute and print the sum of squared residuals, the total sum of squares and the coefficient of determination as determined by the following formula:

Sum of squared residuals (r):

$$\sum_{i=0}^{n-1} (y_i - f(x_i))^2$$

Total sum of squares (t):

$$\sum_{i=0}^{n-1} (y_i - \bar{y})^2,$$

Coefficient of determination: 1-(r/t)

B) Pearson Method

The Pearson Coefficient will be a number between -1 and 1, where a value close to 1 indicates good correlation, values close to 0 will indicate no correlation and values close to -1 indicates negative

correlation. The Pearson method is not used for line fitting but just to indicate what kind of correlation exists between the pairs of (x,y) values.

Algorithm

For the *n* data pairs (x_i,y_i) with $i = 0,, 1, ..., n-1$ the Pearson Correlation Coefficient r is found by using the following formula:

The Pearson Coefficient will be a number between -1 and 1, where a value close to 1 indicates good correlation, values close to 0 will indicate no correlation and values close to -1 indicates negative correlation.

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

In the above formula $\overline{x}$ is the mean of x values and $\overline{y}$ is the mean of y values.

For the Pearson method all we need to is to print the above "r" value which is the Pearson correlation coefficient which will be a number between -1 and 1.

Input

You can use input redirection to read the data for this program. In other words you do not programmatically do file input/output like you did for Lab 12L but do it the same way as you did for 6P. a.out < in1 will run your compiled program with redirected input coming from in1.

The input data file consists of a number of pairs of values, where each pair contains two floating point values which represent a load on the spring and the resulting compression distance, respectively. You should read the data into two separate arrays (say **x** for load data and **y** for compression data). Your arrays should be capable of holding a maximum of 1000 data points. If the input file contains more than 1000 data points, the rest of the data should be ignored without your program erroring out. Each line of your input data file consists one pair of load/compression values.

Computations

The computed results for the linear regression function f(x) and the residuals must be stored in two new arrays (say **fx** and **r**) and have the same maximum size as the input arrays. All arrays must be declared in the main function and appropriately passed to each function based on what the function does.

Output

The output must first print out information on the reading of the data as described above. Next, print the linear regression function using your slope and intercept values. A table is displayed next for each original data point and the load value is substituted in the computed regression equation to determine the predicted (estimated) compression. For each data point, print the weight (load), the observed compression, and the residual. Finally, all the correlation values must be printed: sum of squares of the residuals, the total sum of squares and the coefficient of determination. Your output must be in table format where appropriate. All values should be labeled. Your output does not have to look exactly like mine but it should be as complete and as easy to read.

Requirements

- You must use at least the following functions (all of them must be commented):
    - A function that reads in the data to the arrays
    - A function that prints the contents of the input arrays
    - A function that computes both the y-intercept and slope (this function must have output arguments)
    - A function that computes the linear regression function f(x) and residual for each data pair
    - A function that computes the various correlation values: sum of squared residuals, total sum of squares and the coefficient of determination (all of these should be returned as output arguments)
    - A function that displays the complete table showing the residuals and all the correlation values
    - A function that computes the Pearson correlation coefficient
- The program must be split into multiple files:
    - The main must be the only function in one code file (.c)
    - All the other functions should be in a second code file (.c)
    - The other functions should be forward declared (prototype declarations) in a header file (.h) and included in both code files
    - **Be sure to submit all three files!**
- All function definitions must have a block comment describing what they do.
- Work on the program using the modularity of the functions, for example write a program that reads the data and prints it out. Once you have these working you can move on to the other functions
- As always test your program carefully and follow good programming style.

Other Details

- You must follow all the coding style rules as specified in our "coding guidelines". In particular:
    - You must put your name enclosed in a comment box at the top, and keep any other comments that are already there.
    - Keep lines to a maximum length to the point of making your code easily readable. It is a good idea to make your comment box lines 66 characters long and use this as a guide.
    - You must use good names for any variables you create (a full word that describes what it is there for).
- Details that you do not follow are penalized after other scored items are added up, so even if you got a 100 for the functionality of your program, you can still get a lower score because you did not follow all the other requirements for the assignment.

Multiple Data Files

- You will notice that additional input files provided may not necessarily be for load compression data but since we are using the same technique for simple regression analysis of any type of data, your program will (should) produce accurate results.

Submission

Submit this assignment with the code 7P followed by the names of all 3 files:
For example:

```
submit 7P 7P.c functions.c functions.h
```