1. **[100 points]** You will receive the source code (hw3.1.c) for a C program that will read an ascii file containing a series of \((x, y, \sigma_y)\) measurements into the arrays \(x[]\), and \(y[]\), \(vy[]\) and then call the following routines:

1. **void LinearFit(float x[], float y[], float vy[], int n, float *m, float *b, float *vm, float *vb, float *vmb)**
   
   This subroutine performs a linear least-squares fit on the \(n\) data pairs \((x, y)\) contained in the arrays \(x[1, \ldots, n]\), and \(y[1, \ldots, n]\) with a variance in \(y\) given in \(vy[1, \ldots, n]\). This subroutine returns the slope in \(m\), intercept in \(b\), slope variance in \(vm\), intercept variance in \(vb\), and covariance of slope and intercept in \(vmb\).

2. **float LinearCorrCoeff(float x[], float y[], int n)**
   
   This subroutine returns the linear correlation coefficient \(r^2\) for the \(n\) data pairs \((x, y)\) contained in the arrays \(x[1, \ldots, n]\), and \(y[1, \ldots, n]\).

3. **float LinearFitChiSquared(float x[], float y[], float vy[], int n, float m, float b)**
   
   This subroutine returns the value of chi-squared given the values \(m\) and \(b\) for the \(n\) data pairs \((x, y)\) contained in the arrays \(x[1, \ldots, n]\), and \(y[1, \ldots, n]\) with a variance in \(y\) given in \(vy[1, \ldots, n]\).

4. **void ExtractProjection(int nbin_m, int nbin_b, float h2d[][], float h_m[], float h_b[])**
   
   This subroutine extracts the one-dimensional probability distributions in slope, \(m\), and intercept, \(b\), from the two-dimensional probability distribution for slope versus intercept contained in the two-dimensional array \(h2d[1, \ldots, nbin_m][1, \ldots, nbin_b]\) into the arrays \(h_m[1, \ldots, nbin_m]\) and \(h_b[1, \ldots, nbin_b]\), respectively.

Write these four subroutines **[40 points]** and then, using this program,

(a) **[5 points]** report the best fit values for \(m\) and \(b\).

(b) **[5 points]** report the variances in \(m\) and \(b\), the covariance in \(m\) and \(b\), and the correlation coefficient of errors in \(m\) and \(b\).

(c) **[10 points]** plot the best fit line and the data.

(d) **[5 points]** report \(\chi^2\) and the probability another model would have given a worst \(\chi^2\).

(e) **[5 points]** report the linear correlation coefficient \(r^2_{xy}\).

(f) **[10 points]** make a contour plot of the two-dimensional probability distribution obtained using the \(\chi^2\) function. Also, plot the one-dimensional probability distributions for the slope and intercept obtained from this two-dimensional probability distribution.

(g) **[10 points]** make a contour plot of the two-dimensional probability distribution obtained using the bootstrap method. Also, plot the one-dimensional probability distributions for the slope and intercept obtained from this two-dimensional probability distribution.

(h) **[10 points]** explain the difference, in your own words, between the approaches used in (f) and (g) to obtain the one- and two-dimensional probability distributions.

2. **[50 points]** Using the program **gnufit**, fit the \((x, y)\) pair data set to the function \(y = mx + b\) and the function \(y = ax^2 + mx + b\), and then for each function:

(a) **[10 points]** report the best fit values.

(b) **[10 points]** plot the best fit curve and the data.

(c) **[5 points]** report the 68\% confidence intervals for the best fit values.

(d) **[5 points]** report \(\chi^2\) and the probability that a worst \(\chi^2\) could be obtained.

(e) **[20 points]** Use \(\chi^2\) to determine which model function best describes the data, and explain your reasoning.