

CPS 5401 Intro to Computational Science    Name \_\_\_\_\_  
Fall 2012 Shirley Moore, Instructor  
Fortran Exercises  
Due September 18, 2012

Turn in a gzipped tar file named <your name>-fortran-lab.tar.gz that contains your program source code, input files, and output files for your solutions to the problems below by emailing the file to the instructor. You may turn in your answers to the questions below either electronically (included in your tar file) or on paper. Your tar file should unpack into a directory named <your name-fortran-lab> with subdirectories for each of the two problems. Do not include object or executable files in your tar file. The instructor should be able to build your program for problem 1 by using the gfortran command and build your programs for problem 2 by using your makefile.

1. Write a Fortran 90 program containing an internal function that returns the standard deviation from the mean of an array of real values input from the user. Note that if the mean of a sequence of values  $(x_i, i = 1, n)$  is denoted by  $m$  then the standard deviation,  $s$ , is defined as:

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - m)^2}{n}}$$

[Hint: In Fortran 90 `SUM(X)` is the sum of the elements of `X`.]

To demonstrate correctness compute the standard deviation of the following numbers (10 of them):

5.0 3.0 17.0 -7.56 78.1 99.99 0.8 11.7 33.8 29.6

and also of the following 14,

1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0  
14.0

and redirect your output for both of the above to a file.

2. Download the file `fortran-lab2.tar.gz` from the course website Assignments page and unpack the file.

a. Which file contains the main program? (Hint: try typing  
`grep program *.f90`)

b. Which file contains a module?

c. Give one command or a sequence of commands that will successfully build the executable and run the executable.

d. Construct a makefile that will build the executable automatically, updating only those files that need to be rebuilt. Use optimization level `-O3` and use variables for the compiler name and the compiler flags. Build and run the executable and redirect the output for your build to a file named `make.out` and your run to a file named `<prog>.out`, where `<prog>` is the executable name.

e. Create a second main program by modifying the original main program to read the matrix from a file. Modify your makefile so that you can build both executables at once by typing `'make all'` or build one or the other by typing `make '<prog>'` where `<prog>` is the program name.