Lab Aa

Goals-

write, compile, and test a program that uses pointers; upload it to flip for testing; submit it to TEACH
create and use simple makefiles and simple archiving of all files in a zip file
compile a program written in multiple source code files

1) In a file average.h, declare the function double avg( ). In a file summation.h, declare the function double sum(). In average.cpp define the function double avg () and in summation.cpp define the function double sum (). Each function will have parameters necessary to receive an array of doubles. They will calculate the average and summation of the input array. The arrays should be passed read only. Ensure they are passed so the functions cannot change them.

   Your main function should be in a separate source file. Be certain to include (using the appropriate #include statements) all the files you need to access the functions. You should call avg() and sum() in your main function. You can call them several times with hard-coded data, or data entered by the user. Your program should demonstrate the functions work correctly. You can also demonstrate error modes. Compile, link, and execute your program. If you are not doing your work on flip transfer your files to flip and test your program.

   Test your program thoroughly. Develop a test suite; i.e. the collection of tests you use, specifying any data, and ensuring you test boundary conditions where appropriate. Save your description of the test procedures.

   Create a makefile to build your program. Simple makefiles are little more than listing what you would be typing into the command line.

2) The makefile should include targets for lab Ab and clean. ALL of your files should be included in a zip file archive. Submit the zip file to TEACH.

IMPORTANT RECOMMENDATION!- Operations in a makefile are executed as they would if typed into the UNIX command line. As you may have discovered, the UNIX command line is unforgiving. ☺ So I strongly recommend you do this until you are comfortable with creating makefiles. Complete your work, other than the makefile. Create a subdirectory in your working directory. Copy ALL project-related files to the new subdirectory. Test and revise your makefile in that directory. If you inadvertently delete a file you will still have the original. Once it is working as you desire, you can copy the makefile to your original directory and submit to TEACH.

Grading

Programming style- 1 point

Both functions pass arrays correctly- 2 points

You pass the arrays such that the functions cannot modify them- 1 point

Execute a program requiring multiple source files (#2)- 3 points

Create a makefile with targets for your program, and clean. Submit all files in zip archive- 2 points

You provide a copy of your test procedures, including data files if needed. – 1 point

NOTICE-

This is the first lab where you will exchange code with a classmate. You will give your function header and source files to another student. You will use them in a new (small) project. Ensure your header files contain complete descriptions of how your functions are to be called. You will get code from another student in the class. You will use your test procedures to first test that code. One of the elements of the next lab is how well your test procedures worked. In other words, if you find a problem not included in your tests, then your testing was incomplete. Please pay attention to both the documentation and the testing.