Homework #8 due 3/12/06

Using MS Visual C++ write the following programs. Please email your homework to Steve at sjbrown8@eng.usf.edu.

Exercises
1) Write a C++ program that has an array in the main of type and of size of your choice initialized with data in no particular order. Now design and write a statistical function that utilizes references and your previous solution to ordering arrays (prior homework!). Your function should report what the median of your array is. The median = the middle most number when data is listed in order. If the data are arranged according to size (like from smaller to larger) and n is an odd number, the median is defined as the \( \frac{n+1}{2} \) number. If n is an even number the median is the average of the two numbers, \( \frac{n}{2} \) and \( \frac{n+2}{2} \). Please insure that your program works correctly!!!

2) Do problem #1 again but add 2 additional functions that basically do the same thing but with different C++ types. (You could use char, float, double, etc.) Make sure each function has a different name and have your main program demonstrate that all the functions work correctly with different arrays appropriate to the type. Use enough output to CLEARLY indicate to the reader what is happening. Protect the referenced variables with the C++ reserved word const when appropriate. [HINT: cut & paste]

3) Demonstrate function overloading with problem #2, by changing all the median functions to the same name. Insure that the argument lists are different so that the compiler can differentiate between the overloaded functions.

4) Demonstrate function templates by changing problem #3 above to have one templated median function. Each function that is templated should have a template command before the function prototype and the function itself. The program will still work on each different type variable.

5) Write a C++ function that takes any two integers as an argument and returns the median of those two integers back to the main program. Demonstrate that your function works by utilizing it in a main program of your choice.

6) Do problem #5 again but template it so that it works similarly to problem #4 with at least three different calls from the main program (each call which is a function should be a different type as well).

7) Now overload the templates in problem #4 and problem #6. Write a C++ main program that permits the finding of any median between any two types or the median of any array all with the same function name call (but with different arguments or return types.)

8) In electrical engineering we often need to calculate with complex numbers. Make your own structure called Complex that contains a real and an imaginary component. Also write seven functions that work with your Complex type. The first function should deal with user input for variables of Complex type. This function should return to the main a ‘Complex’ number input by the user. The second function should clearly output Complex types. The final five functions should add, subtract, conjugate, scalar multiply, and complex multiply all Complex numbers.
passed to them. They should return the Complex answer back to the main in all cases. Write, comment, and ‘cout’ the main to satisfactorily document that all your functions work correctly.