**CMPE 310 Assembly Language Project III Mar. 5, 2004**

## Assembly Project for CMPE 310

Assigned: Friday, Mar. 5
Due: Friday, March 19th (during your lab)

**Project Description:**
Write an 80x86 assembly program using nasm that performs the following functions:

- Reads a set of floating point numbers as ASCII characters from a file, converts them to single precision IEEE floating point representation (see the slides or text if you forgot the definition of the standard), saves the floating point values into an array, prints the floating point number (one per line) and the sum to both the screen and an output file called project3.out. The numbers will be given in standard floating point representation as, e.g. (-)9.999999E+/-99, i.e. (-) is optional, the number of digits to the right of the decimal point is variable, E is always followed by a + or a - and two digits where 09 is used for single digit exponents such as 9.
- **NOTE:** You can **NOT** use fscanf or any other C library function to do the conversion to floating point. You may want to use the floating point assembly instructions to help with the conversion. We'll cover the floating point assembly notes this Friday.
- You can still use printf or fprintf for printing from your code.
- As before, the data file name is to be read from the command line. You are welcome to use my code examples and macros to do this project.
- You may assume the number of values in the data file never exceeds 100,000 elements. Therefore, you may statically declare hundred thousand 32-bit double words in your data segment.
- Format of the data file: Assume the file gives the number of data points on the first line. Every line following the first line contains exactly one floating point value.

You must use the submit program to submit your code. The class name is cmpe310 and the project name is proj3. You are also required to turn in a hardcopy. Follow all the instructions given in project 1 section **Turning in your project**. The breakdown of the points will also be similar as project 1. Submit the project (project3.asm) file, any code that you use from our examples should be in (common_code.asm). Properly format your code using the enscript command before printing out the hardcopy.

*The gdb typescript file (project3.txt) submission is optional for this project. If you* ***project works as described above*** *you* ***don't have to submit this file****. If your project* ***works partially it is required*** *that you submit the gdb file showing what part of your project worked at the time of submission.*

You can construct your own data files for this in the format described above. We will test your code on our own examples. The submitted program is due before or during your lab on Friday. You must turn in the hardcopy during the lab and it must be identical to the code that you submitted.

**THE LABS ARE INDIVIDUAL EFFORTS: INSTANCES OF CHEATING WILL RESULT IN YOU FAILING THE COURSE.**