

## 12.010 Homework #3

Due Thursday November 4, 2010

**Submissions:** For these and subsequent homework, your submissions should include

- (1) A description of how you solved the problem;
- (2) The specific output requested for each problem; and
- (3) The C or C++ code that you wrote (either language can be used). (I will compile and run your codes.) If your code has “strange” behavior such as crashing, infinite loops, or excessively long run times (more than a few seconds), let me know in your submissions.

I strongly recommend that you start early on these problems and send questions or come to see me as soon as you encounter problems. You can email [tah@mit.edu](mailto:tah@mit.edu) to set a time for an appointment. Office hours are Tuesday 1:00-2:00 pm.

**Question (1): (25-points)** Write a program that reads a file containing text, counts the number of characters (letters a-z) and words in the text, and output the text with capital letters at the beginning of each sentence. The text below is contained in the file [Q1\\_text.txt](#).

we take as a self evident foundational principle that the set of effects to be considered as contributing to local station displacements and the conventional models to be applied for their compensation should be guided by rational and well considered bases, and should not be developed haphazardly or randomly. for historical reasons and general consistency, it might be prudent to retain some past practices even if they are not fully consistent with the adopted principles; but future expansions should be determined by specified rules. this position paper proposes such a set of guidelines and rationales for IERS conventions updates.

Hints:

- (1) Look at the ASCII table and check the relationship between upper and lower case letters. In C/C++ strings are just integer arrays (declared as char) and the entries in these arrays can be changed with integer arithmetic.
- (2) Remember if reads are coded with scanf or getchar (both of which read from stdin) then the file Q1\_text.txt can be re-directed into the program using: Q1C < Q1text.txt where Q1C is the name of the program (you can call the program any name you like).

**Question (2): (25-points)** Write a program to accurately sum a large number of floating point numbers. The floating point numbers to be summed should be constructed as float random numbers of the form

$$\text{Value} = \text{mantissa} * 10^{\text{exponent}}$$

Where mantissa and exponent are generated as random numbers between 0-1 and exponent as random number between 0 and max\_exponent. (Use “man 3 rand” to see how the rand function works in C)

max\_exponent should be a user specified value (typical value would be 10) and the number of values to be summed should also be user specified (typical value 10000). The sum should be made using a float variable and your results can be compared to a double

summation to see how accurate your result is. The program should try to achieve the maximum summation accuracy and output the accuracy of the float sum. Your submission should explain how your algorithm achieves its accuracy using only float variables and the different methods you considered. Can a formula be found that allows a prediction of the accuracy of the algorithm?

**Question 3: (50-points)** Solution to the Lorenz Strange Attractor problem. Implement a solution to the following differential equations that define the Lorenz Strange Attractor problem:

$$x' = \sigma(y - x)$$

$$y' = -xz + Rx - y$$

$$z' = xy - Bz$$

where  $x'$ ,  $y'$  and  $z'$  are  $dx/dt$ ,  $dy/dt$  and  $dz/dt$  where  $t$  is time;  $\sigma$  is Prandtl numbers,  $R$  is the ratio of Rayleigh numbers and  $B$  is geometric factor.

Your program should take as input values for  $\sigma$ ,  $R$  and  $B$ ; the initial values of  $x$ ,  $y$ , and  $z$ ; the length of time for the solution and the accuracy of the solution. The accuracy can be specified as relative or absolute error. (Absolute error is the absolute difference between your solution and the real solution; relative error is the absolute error divided by the typical size of the variable. The relative error is not dependent on the units of  $x, y, z$ ).

Your solution should explain how you solved the differential equations and the method to used to achieve the desired accuracy of the solution.

To test your program: Output at 1 second intervals, the solution using:

$$\sigma = 10 ; R = 40; \text{ and } B = 8/3;$$

initial values of  $x=0$ ,  $y=-10$ , and  $z=0$ ;

Integration should be for 20-seconds. (Note, the integration step size would normally be less than the output interval). The accuracy should be errors of order  $10^{-4}$ . (The program should be capable of higher accuracy than this).