12.010 Computational Methods of Scientific Programming
Fall 2008

**12.010 Homework #3** **Due Thursday October 30, 2008**

**Question (1):** (25-points) (a) Write, compile and run a C or C++ program that generates two tables of the Gamma function. The Gamma function satisfies the following equation

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$$

where z is the argument of the gamma function and can be complex. (See (See http://mathworld.wolfram.com/GammaFunction.html for information on the Gamma function). Gamma functions are rarely computed by directly integrating the equation above. Generally they are evaluated by series expansions.
For one table, table will be generated for z between 1 and 10 in increment of 1 when z is an integer. This table should give values of Γ(z), Γ(z+1/3), Γ(z+1/2) and Γ(z+2/3).
For the second table, Γ(z) should be generated for z between -1 and +1 in increments of 0.1. Results should be tabulated with at least 6-significant digits.

The submission should include

(a) A discussion of the algorithms used in the program with rationales for the choices
(b) The tables generated by the program and
(c) The C or C++ source code.

**Question (2):** (25-points).
Write a program that reads a file containing text, determines

(1) The average and root-mean-square (RMS) scatter about the mean of the number of characters per word and
(2) The average and root-mean-square (RMS) scatter about the mean of the number of words per sentence. A sentence can end with a period or question mark.

The text below is contained in the file Q2_text.txt.

```
The basic analysis of spacecraft tracking data requires relating the
position and velocity of the spacecraft to the position and velocity of
the tracking system.  The coordinate system used in spacecraft
navigation is shown in Figure 1.   The basic measurements are of r and
its time derivative and sequences of these measurements, combined with
knowledge of the tracking station location and equations of motions of
the spacecraft, allow the position of the spacecraft denoted here by
distance r, right ascension, a, and declination, d, and its velocity to
be determined as a function of time.  In addition to knowing the
coordinates of the spacecraft in an inertial coordinate system, the
coordinates of solar system bodies are also needed in this frame.
Tracking data collected on spacecraft near planets can also be used to
improve the ephemerides the planets through the gravitational
perturbations of the spacecraft motions.   Large combined analysis of
tracking data and direct measurements of planets (radar and optical
positions) are used to generate planetary ephemeredes.
```

Hints:
Remember if reads are coded as scanf or getchar then the file Q2_text.txt can be re-directed into the program using:
Q2C < Q2text.txt where Q2C is the name of the program (you can call the program any name you like).

**Question (3):** (50-points) Write a C or C++ program that will compute the motions of a set of particles undergoing mutual gravitational attraction. The program should generate the trajectories of each of the particles with an error tolerance that is proportional to the separation of the particles. The program should be able to handle large numbers of particles and thought should be given as to how to input the initial positions and velocities of particles when there are a large number of particles.

As a test of your program: Evaluate the trajectories of the 6 particles below with an error tolerance of 1.e-6. (This case is similar to the collision of two Sun-Earth-Moon systems) The integration should be run for 515-days and the positions and velocities at the end of 515 days should be included in the output.

The values below give the mass (kg), X and Y position (km) and X and Y velocities (km/s) of the 6 particles to be evaluated.
2.0e+30 kg XY 0.0e+00 0.0e+00 (km) VXY 0.000e+00 0.000e+00 (km/sec)
8.8e+28 kg XY 1.5e+08 0.0e+00 (km) VXY 0.000e+00 2.857e+01 (km/sec)
7.3e+22 kg XY 1.5e+08 1.0e+07 (km) VXY -2.424e+01 2.857e+01 (km/sec)
2.0e+30 kg XY -1.0e+09 0.0e+00 (km) VXY 1.000e+01 0.000e+00 (km/sec)
8.8e+28 kg XY -8.5e+08 0.0e+00 (km) VXY 1.000e+01 2.857e+01 (km/sec)
7.3e+22 kg XY -8.5e+08 1.0e+07 (km) VXY -1.424e+01 2.857e+01 (km/sec)

Your answer to this question should include:

(a) The algorithms used and the design of your program
(b) The C or C++ program source code (I will compile and run your programs). If you program does not run or takes more than few minutes to run, let me know so that I will treat it with caution.
(c) The results from the test case above with positions and velocities at the end of 515 days. You could also explore the effects of changing the accuracy tolerance on the results.