12.010 Homework #3 **Due Thursday, November 03, 2011**

Question (1): *(25-points) (a) Write, compile and run a C/C++ program which generates a table of error function (erf) and its derivatives for real arguments (z) between -3 and 3 in steps of 0.25. The error function is defined by the equation below (but is rarely evaluated by performing the integration).*

$$\operatorname{erf}(z) \equiv \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2}\, dt.$$

(see http://mathworld.wolfram.com/Erf.html for information the error function)
The values in the table should be given with 5 decimal places. The table should have headers explaining what the columns are. Explain how you designed the program and give an example of the output.
(b) How would you change this program if 10 significant digits were required?
C/C++ source code should also be supplied

Solution:

The algorithm used here is the same as that used in the Fortran solution. However we have added here the ability to pass the desired accuracy of the calculation through the runstring. The default accuracy is 1e-6. Since the erf function is part of the math library (-lm option need when compiled and linked), this version of the program outputs the difference between the series expansion and the math library erf function.

The source code is http://geoweb.mit.edu/~tah/12.010/HW03_01_11.c

Example output:

```
TAHComputer[167] gcc HW03_01_11.c -o HW03_01_11
TAHComputer[168] HW03_01_11
------------------------------------------------------------
TABLE OF ERROR FUNCTION (ERF) within 1.00e-06 AND FIRST DERIVATIVE
------------------------------------------------------------
| Argument |    ERF    | d(ERF)/dx |    ERFC    |   Error    |
|__________|___________|___________|____________|____________|
| -3.000   | -0.99998  |   0.00014 |  -0.99998  | -1.43e-07  |
| -2.750   | -0.99990  |   0.00059 |  -0.99990  | -1.45e-07  |
| -2.500   | -0.99959  |   0.00218 |  -0.99959  |  5.31e-08  |
| -2.250   | -0.99854  |   0.00714 |  -0.99854  |  1.30e-07  |
| -2.000   | -0.99532  |   0.02067 |  -0.99532  | -1.10e-07  |
| -1.750   | -0.98667  |   0.05277 |  -0.98667  |  1.46e-07  |
| -1.500   | -0.96611  |   0.11893 |  -0.96611  |  5.00e-08  |
| -1.250   | -0.92290  |   0.23652 |  -0.92290  |  2.24e-08  |
| -1.000   | -0.84270  |   0.41511 |  -0.84270  |  1.37e-08  |
| -0.750   | -0.71116  |   0.64293 |  -0.71116  |  1.17e-08  |
| -0.500   | -0.52050  |   0.87878 |  -0.52050  |  1.43e-08  |
| -0.250   | -0.27633  |   1.06001 |  -0.27633  | -2.02e-10  |
|  0.000   |  0.00000  |   1.12838 |   0.00000  |  0.00e+00  |
|  0.250   |  0.27633  |   1.06001 |   0.27633  |  2.02e-10  |
|  0.500   |  0.52050  |   0.87878 |   0.52050  | -1.43e-08  |
|  0.750   |  0.71116  |   0.64293 |   0.71116  | -1.17e-08  |
```

1.000	0.84270	0.41511	0.84270	-1.37e-08
1.250	0.92290	0.23652	0.92290	-2.24e-08
1.500	0.96611	0.11893	0.96611	-5.00e-08
1.750	0.98667	0.05277	0.98667	-1.46e-07
2.000	0.99532	0.02067	0.99532	1.10e-07
2.250	0.99854	0.00714	0.99854	-1.30e-07
2.500	0.99959	0.00218	0.99959	-5.31e-08
2.750	0.99990	0.00059	0.99990	1.45e-07
3.000	0.99998	0.00014	0.99998	1.43e-07

As in the Fortran code, this code takes an accuracy value so the number of digits to be output would need to be changed.

Question (2): *(25-points).*
Write a program that reads your name in the form <first name> <middle name> <last name> and outputs the last name first and adds a comma after the name, the first name, and initial of your middle name with a period after the middle initial. If the names start with lower case letters, then these should be capitalized. The program should not be specific to the lengths of your name (ie., the program should work with anyone's name.
As an example. An input of
thomas abram herring
would generate:
Herring, Thomas A.
Hints:
You might look for toupper and tolower as part of the ctype.h header file.

Solution:
This solution is similar to the Fortran code solution. Care should be taken in using scanf for input. In this code fscanf is used so that the maxium number of bytes to be read can be specified. The code also checks the number of characters read but by using fscanf the number should never exceed the maximum allowed.

The solution code is http://geoweb.mit.edu/~tah/12.010/HW03_02_11.c

Example run.
```
TAHComputer[170] gcc HW03_02_11.c -o  HW03_02_11
TAHComputer[171]  HW03_02_11

Enter your names (First, middle, last) tHoMaS aBrAm hErrIng
**********************
* Herring, Thomas A. *
**********************
```

Question (3): *(50-points) Write a C/C++ program that will compute the motion of a bicyclist and the energy used cycling along an oscillating, sloped straight-line path. The path followed will be expressed as*

$$H(x) = Sx + A\sin(2\pi x/\lambda) + B\cos(2\pi x/\lambda)$$

where H(x) is the height of the path above the starting height, S is a slope in m/m, A and B are amplitudes of sinusoidal oscillations in the path. The wavelength of the oscillations is λ. The forces acting on the bicycle are:

$$\text{Wind Drag} \quad F_d = 1/2 A_r C_d \rho V^2$$

$$\text{Rolling Drag} \quad F_r = M_r g C_r$$

where A_r is the cross-sectional area of the rider, C_d is the drag coefficient, r is the density of air and V is the velocity of the bike. For the rolling drag, M_r is the mass of the rider and bike, g is gravitation acceleration and C_r is rolling drag coefficient.

The bicyclist puts power into the bike by pedaling. The force generated by this power is given by

$$\text{Rider force} \quad F_r = P_r / V$$

where F_r is the force produced by the rider, P_r is power used by the rider and V is velocity that the bike is traveling (the force is assumed to act along the velocity vector of the bike). Your program can assume that the power can be used at different rates along the path. The energy used will be the integrated power supplied by the rider. Assume that there is maximum value to the rider force.

Your code should allow for input of the constants above (path and force coefficients). The program can assume a constant power scenario and constant force at low velocities.

As a test of your program use the following constants to compute:

(a) Time to travel and energy used to travel 10 km along a path specified by S=0.001, A=5.0 m, B=0.0 m and λ= 2km, with constant power use of P_r =100Watts and a maximum force available of 20N.

(b) The position and velocity of the bike tabulated at a 100-second interval.

Assume the following values

C_d = 0.9

C_r = 0.007

A_r = 0.67 m^2

ρ = 1.226 kg/m^3

g = 9.8 m/s^2

M_r = 80 kg

Your answer to this question should include:

(a) The algorithms used and the design of your program

(b) The C/C++ program source code (I will compile and run your programs).
(c) The results from the test case above.

Solution:

This solution is very close to the Fortran HW03_03_11-1D solution (i.e., the bike is treated like a roller coaster and the problem is solved in 1-D while accounting for the difference between the sloped and horizontal distances.

As in the Fortran code a header file is used (CBike.h) and this contains definitions and global variables that are accessible in all functions. Prototype function lines are included at the top of the source code.

Notice some of the changes that needed to be made to the code. In particular, we needed to change the function call to nint in the Fortran code to ceil in the C code, and we needed to change the abs call to fabsl. Specifically here if we left the abs call as coded, the wrong answers were generated because abs is the integer values only. We also changed the input so that the user can simply say no and keep the default values. We also coded the input so that the same use of / as in Fortran code was used.

The solution is http://geoweb.mit.edu/~tah/12.010/HW03_03_1D_11.c and http://geoweb.mit.edu/~tah/12.010/CBike.h

Example with defaults.

```
TAHComputer[204] gcc HW03_03_1D_11.c -o  HW03_03_1D_11
TAHComputer[205] HW03_03_1D_11

12.010 Program to solve Bike Motion problem
Given track characteristics and rider/bike properties
the time, energy and path are computed.

Do you want to change defaults (y/n) n

PROGRAM PARAMETERS
++++++++++++++++++
Length of track  10.000 km and error   10.0 mm
Track Slope 0.001 Sin and Cos amplitudes  5.00  0.00 (m) and wavelenghth
2.00 (km)
Rider/Bike Mass  80.00 (kg), Area  0.670 (m**2), Drag and rolling
Coefficient   0.90 0.0070
Rider Power 100.00 (Watts) and max force  20.00 (N)
Output Interval 100.00 (s)

++++++++++++++++++
Step 1.000000 Times 2255.063413 2255.063412 Delta (ms) 0.001372
O*    Time          X_pos        S_pos               S_vel     Energy
O*    (sec)          (m)          (m)                (m/s)     (Joules)
O      0.000         0.0000         0.0000         0.0000         0.00
O    100.000        80.0297        80.0407         1.4896      1600.80
```

O	200.000	306.9129	306.9453	3.3689	6138.93
O	300.000	827.1168	827.1622	6.7990	15413.85
O	400.000	1514.1006	1514.1889	5.8755	25413.85
O	500.000	1951.8144	1951.9321	2.9589	33917.06
O	600.000	2202.5693	2202.7186	2.6557	38932.85
O	700.000	2608.8907	2609.0514	5.8317	46847.61
O	800.000	3293.3369	3293.5471	6.8874	56847.61
O	900.000	3835.0718	3835.2993	3.8807	66461.32
O	1000.000	4117.5262	4117.7907	2.3544	72111.21
O	1100.000	4428.4800	4428.7645	4.5370	78330.75
O	1200.000	5047.1325	5047.4485	7.2508	88281.45
O	1300.000	5682.1018	5682.4419	4.9717	98281.13
O	1400.000	6035.4481	6035.8262	2.4806	105348.84
O	1500.000	6294.4797	6294.8846	3.3237	110530.07
O	1600.000	6804.3476	6804.7643	6.7171	119715.19
O	1700.000	7494.1230	7494.5844	5.9834	129715.19
O	1800.000	7941.7253	7942.2147	3.0317	138353.52
O	1900.000	8193.7065	8194.2280	2.6066	143393.84
O	2000.000	8589.4172	8589.9508	5.7284	151148.99
O	2100.000	9270.0071	9270.5896	6.9615	161148.99
O	2200.000	9821.8172	9822.4164	3.9886	170834.47
O	2255.063	10000.0004	10000.6222	2.6506	174490.72

Time to travel 10.000 km, 2255.06 seconds, 0.63 hrs
Rider Energy 174490.72 Joules, 41676.393 Calories
Kinetic 281.02 Joules
Final Velocity 2.651 m/sec

12.010 Computational Methods of Scientific Programming
Fall 2011