COMP 350 Numerical Computing

Assignment #1: Floating point in C, overflow and underflow, numerical cancellation

Date Given: Friday, September 22. Date Due: 5pm, Monday, October 2, 2017

(To be marked by Mr. Aditya Kashi (aditya.kashi@mail.mcgill.ca))

Submit your assignment including your code through myCourses. Print out your program and computed results so that the TA can easily read. Note that the TA may or may not run your code.

1. (10 points) Write a program to read a sequence of positive numbers and compute the product. Assume that the input numbers do not overflow or underflow the IEEE single precision. The program should have the following properties:

   - The variables in the program should have type either float of int. Double or extended precision variables are not permitted.
   - The program should print the product of the numbers in the following nonstandard format: a floating point number F (in standard decimal exponential format), followed by the string

     `times 10 to the power`,

     followed by an integer K. Here we assume $|K|$ is not bigger than the biggest integer that can be stored.
   - The result should not overflow, i.e., the result should not be ∞, even if the final value, or an intermediate value generated along the way, is bigger than $N_{\max}$, the biggest IEEE single precision floating point number.
   - The intermediate and final results should not underflow, i.e., the intermediate and final values should not be subnormal numbers, even if they are smaller than $N_{\min}$, the smallest positive normalized IEEE single precision floating point number.
   - The program should be reasonably efficient, doing no unnecessary computation (except for comparisons) when none of the intermediate or final values are greater than $N_{\max}$ or smaller than $N_{\min}$. In this case, the integer K displayed should be zero.

   An important part of the assignment is to choose a good test set to properly check the program. Print out your program, and its input and output. Write some comments about your test results.

   Note: If your compiler does not support the macro INFINITY, then compute ∞ from 1.0/0.0 at the beginning, assuming the standard response to division by zero is in effect.

2. For any $x_0 > -1$, the sequence defined recursively by

$$x_{n+1} = 2^{n+1}(\sqrt{1 + 2^{-n}x_n} - 1), \qquad (n \geq 0)$$

   converges to $\ln(x_0 + 1)$.

   (a) (4 points) Let $x_0 = 1$. Use the formula to compute $x_n - \ln(x_0 + 1)$ for $n = 1, 2, \ldots, 60$ in double precision. Explain your results.

   (b) (6 points) Improve the formula to avoid the difficulty you encountered in 2(a). Again compute $x_n - \ln(x_0 + 1)$ for $n = 1, 2, \ldots, 60$ in double precision.

   Note: You should make your code efficient, i.e., avoid unnecessary operations.