CSCI-21 Assignment #6, due 3/30/16

Isaac Newton's method to approximate square roots works like this: given an approximation x of $\sqrt{n}$, a better approximation is $\dfrac{x + \dfrac{n}{x}}{2}$. Repeat until you get the desired precision.

Write a subroutine sqrt implementing Newton's algorithm for approximating a square root. Start with an arbitrary guess for the approximation (1.0 is a good starting point), and loop until your approximation squared is within 10^{-5} of n, i.e. $\left| x^2 - n \right| < 0.00001$. (A better way is to test if the absolute value of the ratio, $\left| \dfrac{x^2}{n} \right|$, is close to zero – why?) You may adapt the newton.asm program from my Web site for this.

Write a MIPS program that prompts the user for the (x, y) coordinates of two points in the real (Cartesian) plane, and then calculates and displays the distance between the points with reasonable descriptive text. Prompt for the coordinates using a little subroutine using simple linkage — do not duplicate the code for this.

Calculate the distance by using the Pythagorean Theorem:

$$d = \sqrt{\Delta x^2 + \Delta y^2}$$

Use all single-precision (32-bit float) arithmetic for this assignment. Use the simple register-based linkage convention (arguments in $a registers, return value in $v0, nothing on the stack) for this program.