CSCI-21 Assignment #7, due 11/10/22

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

Using the simple linkage convention, 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, $\left|\frac{x^2}{n}\right|$, n != 0.0, is very close to one – 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 the prompting. Call this subroutine four times, twice for each point's coordinates. Pass the subroutine the address of the prompt string (in $a0) and have it return the single-precision float (in $f0). Remember the rules for register use with the simple linkage convention.

Calculate the distance by using the Pythagorean Theorem:

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

The Newton subroutine will just calculate the square root. The main program will calculate the sum of Δx and Δy itself.

Use all single-precision (32-bit float) arithmetic for this assignment. Use the simple register-based linkage convention (integer-type argument in $a0, floating-point argument in $f12, floating-point return value in $f0, nothing on the stack) for this program.