# Overview

For this exercise, you will write an assembly code subroutine to approximate the square root of an argument using the bisection method (see Wikipedia for details). All math is done with integers, so the resulting square root will also be an integer.

# Requirements

Your code must approximate the square root of an integer between 0 and 231-1. Using integer math is enough (no floating point or fractions are required); your code will return the truncated (integer portion) of the square root.

Your code must be in an **assembly language subroutine** which is called by a C function for testing. Be sure to use registers according to the ARM calling convention.

# Software Design

Base your software on the following pseudocode:

Approximate square root with bisection method

INPUT: Argument x, endpoint values *a*, *b, such that a < b*

OUTPUT: value which differs from sqrt(*x*) by less than *1*

done = 0

a = 0

b = square root of largest possible argument (e.g. ~216).

c = -1

do {

c\_old <- c

c <- (a+b)/2

if (c\*c == x) {

done = 1

} else if (c\*c < x) {

a <- c

} else {

b <- c

}

} while (!done) && (c != c\_old)

return c

# Testing

In the main function, write code to test that your subroutine works correctly.