



Programming Lab 3A

[Click to download Lab3A-Main.c](#)

Functions & Parameters

Topics: Passing parameters, function return values, nested functions, preserving and restoring registers across function calls, calling C functions from assembly.

Prerequisite Reading: Chapters 1-3

Revised: January 1, 2022

Assignment: The main program contains each of the four functions shown below. The program may be compiled and executed without writing any assembly. However, your task is to create assembly language replacements for each of these C functions. The original C functions are defined as “weak”, so that the linker will automatically replace them in the executable image by those you create in assembly; you do not need to remove the C versions. Functions Square and SquareRoot are provided in the main program; do not recreate them in assembly – just call them from your assembly language code for functions Square2x and Last.

<pre>int32_t Add(int32_t x, int32_t y) { return x + y ; }</pre>	<pre>int32_t Less1(int32_t x) { return x - 1 ; }</pre>
<pre>int32_t Square2x(int32_t x) { return Square(x + x) ; }</pre>	<pre>int32_t Last(int32_t x) { return x + SquareRoot(x) ; }</pre>

Code and test your functions one at a time using the main program downloaded using the link above. If your code works correctly, the display should look like the image shown. Press the blue pushbutton to cycle through all the test cases to verify that everything is correct. Color is used to indicate the status of a function:

Gray	Function is never called.
Yellow	Function is provided by the main program.
Orange	Function never returns from being called.
Green	Function returns the correct value.
Red	Function returns an incorrect value.
Magenta	Function doesn't keep SP double-word aligned
Blue	Function doesn't preserve R4-R11

IMPORTANT – The .thumb_func directive:

The ".thumb_func" assembler directive specifies that the next label is the entry point of a function that contains instructions from the Thumb subset of the ARM processor and causes the binary representation of instructions that branch to that label to be generated somewhat differently. Thus in a source code file that contains more than one function, it is imperative that you place a .thumb_func directive immediately before the entry point label of every function.

Note: The ARM Procedure Call Standard requires the stack pointer be double-word aligned when functions are called. Failure to observe this requirement can cause a program to behave incorrectly or crash.

