Purpose

Determining what the program mystery.c needed to do was not all challenging as I thought The dothething function was recursive implementing a variable say n in the way of (n-1) and (n-2) and then calling the add function. Hence, this is a fibonacci sequence.

Then I started tracing the programs using different arguments. I began trying to replicate the assembly code by writing C code and compiling to see what the C code generated. I repeated this process until the assembly code my C code generated was extremely close to the original assembly code in mystery.s.

Optimized assembly

First, the optimized assembly used registers to store local variables often instead of storing them on the stack. This is probably useful because operating with values inside registers is faster than having to access values on the memory of the stack which is much slower than the registers. The second thing I noticed was that conditional statements I made were condensed. This leads to less lines of code overall, and will result in a faster execution.