Mystery Readme

Design

I figured out what mystery.s did by going through the assembly and writing out the registers and ebp memory locations. There were 3 functions: add, dothething, and main. Add was just simple addition of two variables. Line 10 gave away what the dothething function actually was. I saw there was double recursion and found out it was dothething(a-1) +dothething(a-2). This directly correlates to finding the nth Fibonacci number. Later I found out num was actually an array and put the pieces together. I think I may be missing some pieces, but my program works. Because the input is a 32-bit integer overflow occurs when the user enters a number greater than 46.

Compiler Optimization

It seems the compiler reused registers more often in the optimized version. Therefore its memory use was more efficient allowing for the program to run faster.