Assignment 2

(2) Mystery

Seong Kon Kim

I've found out the mystery by plugging in the value from 0 and up. I realized that as it goes above 2, it continuously adds the 2 previous values to get the next value. The first we found the value for ebx register where it is n-2 index, where it has value of the sum of (n-3) index and (n-2) index from a num array and calls dothething. Next, it will call the dothething again and get the n-1 index and call add function and store the sum value in -8(%ebp). Next get the value of n in edx and store that sum value in the n index and goes on until the input index value comes up. This sequence of code is called Fibonacci sequence.

By comparing the optimized c code and un-optimized, the optimized code has less jumps and less uses of registers. The most significant difference that I saw from two code was that optimized code has efficient way of coding than the other. For example, the unoptimized code will have each line of codes for allocation, arithmetic operations and storing while optimized code will do the same exact procedure in one to two lines of code. The running time of optimized code is faster as expected and by seeing this difference, it does show the reasons why the optimized code is efficient.