

# Mystery

## What the Program Does

The mystery program computes the Fibonacci sequence. It has an add function that is called from within the fib function. It is passed two numbers a call to fib(n-1) and fib(n-2). It just adds them up and returns the sum of the two.

## Figuring it out

I figured out the basic functionality of the program(Fibonacci sequence) by that I saw that it was calling itself over and over again. Furthermore, since it was stated that it was a well know computation, I was able to figure out that it was the Fibonacci sequence, and a quick check of the function confirmed that. I was unable to figure out the optimization that was made. Although, there is a recursive call, the optimizations that I made to my code using recursive calls all resulted to run slightly slower than the given code. So, I resorted to using a loop, which made the code run faster.

## Analysis – Code Optimization

Comparing my optimized code with unoptimized code, the changes the compiler made were that instead of using `leal (%edx, %eax), %eax` (where it takes what's in both `%edx` and `%eax`, adds them up and puts it in `%eax` ) it used `addl 8(%ebp), %eax` where it first took the what was at `8(%ebp)` and added with `%eax` and put it in `%eax`. Some other optimizations it did was use `subl` instead of pushing values. The compiler made these changes because it minimized the number of push and pop instructions. It tries to not use the stack as much as possible. If calls to the stack are minimized then the program will run faster.

## Error Checking

While the mystery.c file given to us does very little error checking. The error checking my mystery.c does is if an invalid number/letter/character is inputted it will return 0. If there is an overflow, it will simply return 0.