Consider the following unoptimized code:

```plaintext
func_F:    // A function func_F
    // has a single parameter passed in register rdi

    enter $(8*3), $0 // function prologue
    // sets stack for local variables

    movq $0, -8(%rbp)
    movq $0, -16(%rbp)
    movq $0, -24(%rbp)

    movq %rdi, -8(%rbp)
    movq $1, %r10
    movq %r10, -16(%rbp)
    movq -8(%rbp), %r10
    movq %r10, -24(%rbp)

    movq -16(%rbp), %r10
    addq %r10, -24(%rbp)
    movq -24(%rbp), %rax

    leave   // function epilogue
    ret     // returns to the caller function
```

Write down the Decaf source code that corresponds to `func_F`:

What would an optimized version of this Decaf source code look like?

Did you find the Project 2 feedback at all helpful/useful? Besides sending them out earlier (I promise I’ll be on campus for future ones), how could they be more helpful for your team?