

Lab 2

[Start Assignment](#)

Due Wednesday by 5pm **Points** 75 **Submitting** a file upload
File Types c and s **Available** until Mar 1 at 10pm

In this lab you will write x86 assembly language routines that will work with the C programming language. For each assembly language routine you must write a test routine in C that takes parameters from the command line and provides appropriate input to the assembly language routine. You will write all routines in 32 bit assembly using the GAS format.

1) Fibonacci -Iterative

This routine will take an integer n as a parameter and will report back the nth Fibonacci number. Your assembler routine must be iterative. If the number cannot be computed in a 32 bit signed integer then your routine must return -1 as an error code. Your test routine will take a number n from the command line and will print out the nth Fibonacci number, or -1, on a line by itself.


submit : testfiboi.c fiboi.s

2) Fibonacci -Recursive

This routine will take an integer n as a parameter and will report back the nth Fibonacci number. Your assembler routine must be recursive. If the number cannot be computed in a 32 bit signed integer then your routine must return -1 as an error code. Your test routine will take a number n from the command line and will print out the nth Fibonacci number, or -1, on a line by itself.

submit : testfibor.c fibor.s

3) 32 Bit unsigned multiply using shift and add

Using the [shift and add algorithm](https://en.wikipedia.org/wiki/Multiplication_algorithm)  (https://en.wikipedia.org/wiki/Multiplication_algorithm) your routine will take two 32 bit integer values and return a 32 bit integer product. If the result overflows,

your routine will return a -1. Note that you are using normal signed integers as input but your routine should only work with non-negative values. You may not use any x86 multiply instructions to solve this problem. Your test routine will take two numbers n_1 , n_2 from the command line and will print out $n_1 * n_2$, or -1, on a line by itself.

submit : testmul.c mul.s

Fibonacci Notes

For the Fibonacci routines $F(0) = 0$ and $F(1) = 1$.

Error Checking

1. You cannot assume that you will receive only valid inputs.
2. You can assume that all inputs will be integers.
3. You may not do any error checking in the C code. This is a class in assembly language, check for all out of bounds conditions in assembler. The C code should simply parse the command line into integers and pass those on to the assembly routines, and then print the returned value. No more, no less.

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs