

Assignment MIPS 3

Due Date: April 28

Purpose

In this assignment, you will try out MIPS subroutines, in the form of recursion. To do this properly, you must store local variables in the stack. This forces you to take care of the \$sp and \$fp pointers. *Follow the in-class and text examples carefully!*

Problem

The Fibonacci sequence is a sequence of integers, f_1, f_2, \dots, f_n defined by:

$$f_i = f_{i-1} + f_{i-2}$$

where $f_1 = 1$ and $f_2 = 1$.

Thus, if $n = 5$, the first five values of the Fibonacci sequence are 1, 1, 2, 3, 5. You are to write a **recursive** function that displays n values in the Fibonacci sequence. Furthermore, the function should return the sum (an integer) of the n values.

Input

The program should prompt for n , an integer. Assume that $0 \leq n \leq 44$.¹ No error checking of the input is necessary.

Output

The output should be the Fibonacci sequence, all values on one line, comma separated, and displayed in **reverse** order. That is, if $n = 5$, then the output would be:

The Fibonacci Sequence is:

5, 3, 2, 1, 1

The sum is 12

Specifics

- You **must** do this recursively; no loops are allowed. In order for this to work properly, you must store local values in the stack before each new recursive call. Follow the sample program done in class. Note that it doesn't hurt to allocate a little more storage than you actually need.
- **Do not treat any registers as global variables!** We are attempting to simulate how a good recursive function written in a high-level language gets translated to assembly.
- Your recursive solution does **not** have to follow the above definition. You may find a much simpler and more efficient solution. It still must be implemented using recursion, however.
- An input of 0 should display just the sum, which would also be 0.
- The sum should be **returned** to the main procedure in \$v0.
- For your own sanity, use good commenting style.

¹What's so special about 44?

Notes

- Your best bet is to write a short C++ or Python version first, then translate this to MIPS. I have written the MIPS version, and the final code is only about 140 lines long, including comments and spacing.
- As usual, turn in your source code to me via Canvas, as usual, using the same naming convention as in: `mgousieMIPS3.s` or `mgousieMIPS3.asm`. Turn in a printed copy in class on April 29th.
- Note that the Crazy Model Expo is that same week. Schedule your time accordingly!

Recursion is when a function calls itself until it doesn't.
– Mattias Petter Johansson