

# CS2610

## Lab 3 : Stack and Recursion

Write recursive programs in RISC-V assembly to understand argument passing, stack maintenance, and local variable allocation.

### Problem 1: Fibonacci Sequence (40 points)

Write a recursive RISC V assembly program that computes the sum of the first N elements of the Fibonacci sequence.

1. The value of N should be defined in memory.
2. Recursion is mandatory.
3. Use the formula  $f(N) = f(N - 1) + f(N - 2)$  for  $N > 1$ ,  $f(0) = 0$  and  $f(1) = 1$  directly in your implementation.
4. Maintain global variables to keep track of lowest and highest sp values. At the end of the main, before returning, you must add a label check. At this point, a0 = sum of (fib(N)) and a1 = stack usage must hold.

### Problem 2: Tower of Hanoi (60 points)

Write an assembly program to print the moves required to solve the tower of hanoi problem recursively.

1. See [here](#) for the recursive formulation.
2. In newlines, print each move as "Move from rod X to rod Y" where X and Y are towers.
3. This program will be considerably longer than your previous programs. Saving and restoring local variables appropriately will be necessary.

To print a string, you can make the *write* syscall in RISC V as shown below:

```
str : .asciz "Move from rod X to rod Y\n"  #define the string in data section
of your code.
x : .word 14      # idx of x in str
y : .word 23      # idx of y in str
#use the following snippet in your code to print the moves
#before printing the string, modify the values of x and y using their indices.
li a0, 1          #pass the file descriptor: 1 is stdout
la a1, str        #pass the address of the string
li a2, 25         #pass the length of the string
li a7, 64         #pass the syscall number for write
ecall            #make syscall
```

**Hint:** Here is the C code for the problem to help you get started.

```
void towerOfHanoi(int n, char from_rod, char to_rod, char aux_rod)
{
    if (n == 1)
    {
        printf("\n Move from rod %c to rod %c", from_rod, to_rod);
        return;
    }
    towerOfHanoi(n-1, from_rod, aux_rod, to_rod);
    printf("\n Move from rod %c to rod %c", n, from_rod, to_rod);
    towerOfHanoi(n-1, aux_rod, to_rod, from_rod);
}

int main()
{
    int n = 10; // Number of disks
    towerOfHanoi(n, '1', '3', '2'); // 1, 2 and 3 are names of rods
    return 0;
}
```

You can use the above C code to test the output of your solution.

**Submit screenshots for n = 1, 2, 5, and 10.**

## What you need to submit:

1. Code files
2. Screenshots of the output

### Note:

1. All the files should be submitted in a zipped folder through Moodle.
2. The zipped folder should be named  $\langle Roll\_No \rangle\_Lab2.zip$ .

## Resources

[RISCV-unprivileged-spec](#)