Functions

Convention 1

- **a0-a3** are the **argument registers**
- **v0-v1** are the **return value registers**
  - This is just a **convention,** there's nothing special about them
    - Does th[illegible]an I can pass values in (e.g.) s-registers?
      - Yessss[illegible]..????¬.¬
    - Will I lose any points in midterms/labs/projects if I do?
      - Yesssssssss!!!!¬.¬

- By convention! We **never** do that!
  - **ALWAYS** pass arguments in ***a-registers***
  - **ALWAYS** return arguments in ***v-registers***

Put the arguments in **a** registers before calling another function using **jal**.

Make sure to load the return value in v0 to another register before you use v0 to make a syscall.

https://luisfnqoliveira.gitlab.io/cs447_s2021/slides/08-Functions-and-the-Stack.pdf

Convention 2

functions are required to put these registers **back the way they were before they were called.**

Saved	Unsaved
s0-s7	v0-v1
sp	a0-a3
ra*	t0-t9

anyone can change these. after you call a function, **they might have totally different values from before you called it.**

https://luisfnqoliveira.gitlab.io/cs447_s2021/slides/08-Functions-and-the-Stack.pdf

Exercise 1

Code:

```
func1: push ra
       li s0, 10
       li s1, 20
       jal add
       pop ra

add:   add a0, s0, s1
       jr ra
```

Do the functions follow all the conventions?

Exercise 2

Code:

```
func1: push ra
       li a0, 10
       li a1, 20
       jal add
       pop ra

add:   push s0
       add s0, a0, a1
       move v0, s0
       pop s0
       jr ra
```

Do the functions follow all the conventions?

Exercise 3

```
main:   li a0, 1
        jal func2
        ..
        li v0, 10 #exit code
        syscall
func2: push s0
        push ra
        add s0, a0, 2
        move a0, s0
        jal func3
        pop ra
        pop s0
        jr ra

func3:  move v0, a0
        jr ra
```

- What conventions are broken?
- Will this function execute correctly?

Exercise 4

```
main:   li a0, 1
        li a1, 2
        jal func1
        add s0, v0, s0
        jal func2
        sub s1, v1, s1
        li v0, 10
        syscall

func1: move v0, a0
        move s0, v0
        jr ra

func2: move v1, a1
        move s1, v1
        jr ra
```

- What is the final value of s0 and s1?
  - s0 = 2, s1 = 0
- What conventions are broken?

Exercise 5

- How big is the activation frame of function add?

```
add:
    push ra
    push s0
    push s1
    push s2

    # Code Omitted

    pop s2
    pop s1
    pop s0
    pop ra
```

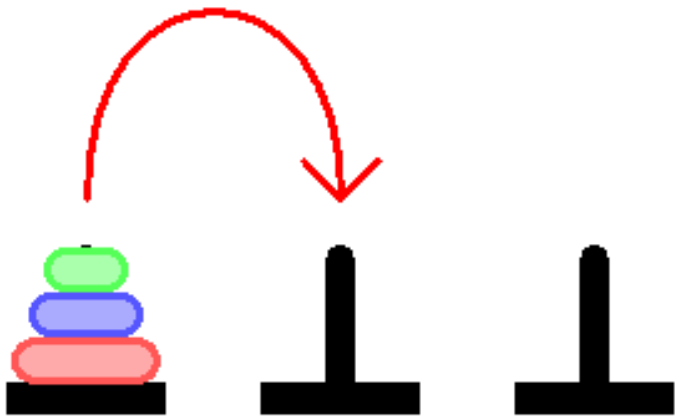
- 8 Bytes
- 12 Bytes
- 16 Bytes
- 4 Bytes

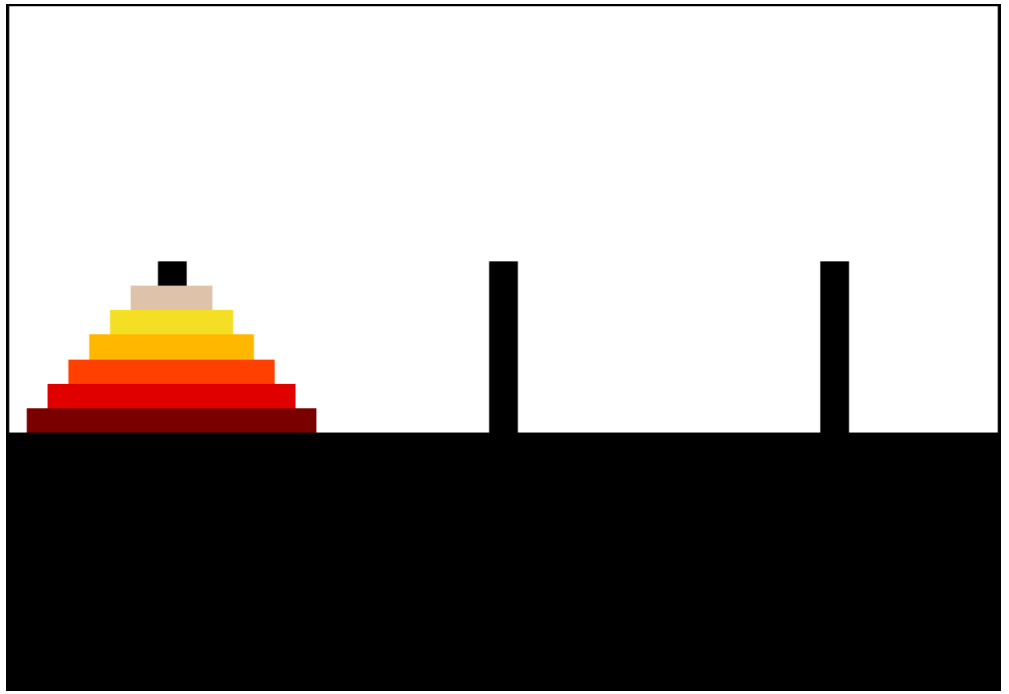
Tower of Hanoi

- There are three towers
- *N* round disks, with decreasing sizes, placed on the first tower
- You need to move all of the disks from the first tower to the last tower
- Larger disks can not be placed on top of smaller disks
- The third tower can be used to temporarily hold disks

https://www.cs.drexel.edu/~jjohnson/2004-05/fall/cs270/lectures/lec1.ppt

Tower of Hanoi





https://en.wikipedia.org/wiki/Tower_of_Hanoi
https://askgif.com/blog/85/discuss-towers-of-hanoi-puzzle/

What you need to do ?

- Print the correct sequence using recursion

Example sequence

```
Moving disk 1 from src to aux
Moving disk 2 from src to dest
...
...
...
```