Non-leaf functions

- Steps
  - Before doing any work with registers
    - Save $ra on the stack
    - If you will use any $sX registers, save their old values on the stack, since the caller will need them after returning
    - If you will need the values of any $aX registers after a function call, store them on the stack
  - Do your work
  - Before returning
    - Restore $ra and $sX registers from the stack

Recursion

- When a function calls itself

- Factorial

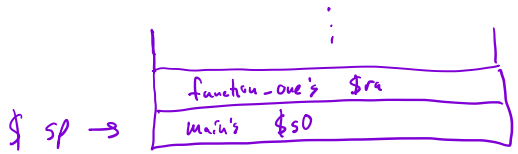
$$n! = \underbrace{1 * 2 * 3 * \dots * (n-1)}_{(n-1)!} * n$$

$$n! = n * (n-1)!$$

$$0! = 1$$

```
int factorial(int n) {
    if (n == 0)
        return 1;

    return n * factorial(n-1);
}
```

```c
#include <stdio.h>

int function_one(int x, int y);
int function_two(int x, int y, int z);

int main() {
  printf("Enter x: ");
  int x;
  scanf("%i", &x);

  printf("Enter y: ");
  int y;
  scanf("%i", &y);

  int z = function_one(x, y);

  printf("z: %i\n", z);

  return 0;
}
```




```c
                    $a0     $a1     $a2
int function_two(int x, int y, int z) {
  return (x + y) * z;
}          $t0
          $L0 → $v0

                    $a0     $a1
int function_one(int x, int y) {
  int z = x * y;
      $s0
                          $a0 $a1 $a2
  int a = function_two(x, y, z);
      $v0
          $v0   $s0
  return a + z;
}        $v0
```

Must save $ra and $s0 on the stack