Functions in MIPS

- Can call functions using labels and "jump and link":
  - Label defines address of instruction in memory
  - jal LABEL will store the current PC contents in the special $ra ("return address") register, and set the PC to the address specified by LABEL
  - jr $ra will jump back to the address stored in the $ra register
- Can use stack area of RAM (with the stack pointer register, $sp) to store data so that it can be restored once the function call has finished

Functions in MIPS: example

Simple function to add 1 to $s0, save it, and then restore the value

Call the function inc_and_save

Save $s0 to stack, and restore afterwards

```
.text
main:
  addi $s0, $zero, 1
  addi $s1, $zero, 0
  jal inc_and_save # call the function

  # exit program via syscall
  lw $v0, 10
  syscall

inc_and_save:
  # store current value of $s0 on the stack
  addi $sp, $sp, -4 # an integer is 4 bytes
  sw $s0, 0($sp)
  # increment $s0 and save it to ram
  # add the address in $s1
  addi $s0, $s0, 1
  sw $s0, 0($s1)
  # restore original value of $s0
  lw $s0, 0($sp)
  addi $sp, $sp, 4 # restore stack pointer
  # return to function call location
  jr $ra
```