Variables can have longer lifetimes - registers $s0 to $s7

Function in C

```
int bar(int x) {
    return f(x) + g(x) + h(x);
}
```

Code

The function is non-leaf, so we have to save the return address.
Register allocation: the value of x is in $s0,
f(x) will be in $s1, f(x) + g(x) will be in $s2.
The convention says: if we use a register in the set $s0 to $s7, we have to save and restore the previous contents. So:

```
bar:    addi    $sp,$sp,-16
        sw      $ra,12($sp)
        sw      $s0,8($sp)
        sw      $s1,4($sp)
        sw      $s2,0($sp)   # end of prologue

        move    $s0,$a0      # x is now in $s0

        jal     f
        move    $s1,$v0      # f(x) is in $s1

        move    $a0,$s0      # x: argument for g()
        jal     g
        add     $s2,$s1,$v0  # f(x) + g(x) in $s2

        move    $a0,$s0      # x: argument for h()
        jal     h
        add     $v0,$s2,$v0  # f(x) + g(x) + h(x) in $v0

        lw      $s2,0($sp)   # epilogue
        lw      $s1,4($sp)
        lw      $s0,8($sp)
        lw      $ra,12($sp)
        addi    $sp,$sp,16

        jr      $ra
```