CS/COE 0447 Fall 2009
Lab 5: More Functions
Solution

```
#Problem 1:

addition:
     # Function Prologue
     # 1. Adjust stack pointer to create activation record
     # We need space for the return address and three saved registers
     # (x, y, z)
     addi $sp, $sp, -16
     # 2. Save $sx registers (if you use them)
     sw $s0, 0($sp)      # $s0 will hold x
     sw $s1, 4($sp)      # $s1 will hold y
     sw $s2, 8($sp)      # $s2 will hold z
     # 3. Save return address (if leaf function)
     sw $ra, 12($sp)


     # function body
     # x = $a0
     move $s0, $a0
     #  = $a1
     move $s1, $a1
     #z = x + y
     add  $s2, $s0, $s1
     #Set return value
     move $v0, $s2


     # Function Epilogue
     # 1. Restore $sx registers (if any)
     lw $s0, 0($sp)
     lw $s1, 4($sp)
     lw $s2, 8($sp)
     # 2. Restore return address (if leaf function)
     lw $ra, 12($sp)
     # 3. Adjust stack pointer to pop activation record
     addi $sp, $sp, 16
     # 4. Return to caller
     jr $ra
```

```
#Problem 2:

fib:
     # Function Prologue
     # 1. Adjust stack pointer to create activation record
     # We need space for the return address, the parameter n and a
     # register to store the return of the first call to fib
     addi $sp, $sp, -12
     # 2. Save $sx registers (if you use them)
     sw $s0, 0($sp)      # $s0 will hold n
     sw $s1, 4($sp)      # $s1 will hold return value of first call
                         # to fib
     # 3. Save return address (if leaf function)
     sw $ra, 8($sp)

     # function body
     move $s0, $a0  # n = $a0
     slti $t0, $s0, 2         #if(n <= 1)
     beq $t0, $zero, else
     move $v0, $s0       #Set return value
     j end               #Skip else body
else:
     addi $a0, $s0, -1   #Set parameter to first call (n-1)
     jal fib             #Make call
     move $s1, $v0       #Save return value of first call
     addi $a0, $s0, -2   #Set parameter to first call (n-1)
     jal fib             #Make call
     add $v0, $s1, $v0   #Set return value
end:
     # Function Epilogue
     # 1. Restore $sx registers (if any)
     lw $s0, 0($sp)
     lw $s1, 4($sp)
     # 2. Restore return address (if leaf function)
     lw $ra, 8($sp)
     # 3. Adjust stack pointer to pop activation record
     addi $sp, $sp, 12
     # 4. Return to caller
     jr $ra
```