```
#####################################################################
# Leaf example (PH p. 134):
#   a MIPS function that doesn't call any other functions
#### C code for the function: ######################################
#   int f;
#
#   int leaf_example (int g, int h, int i, int j)
#   {
#       f = (g + h) - (i + j);
#   }
#   void main(void)
#   {
#       leaf_example(5, -20, 13, 3);
#       printf("\nThe value of f is: %i\n", f);
#   }
#### Output of a run of the C program ##############################
#   four06% cc -o leaf leaf.c
#   four06% leaf
#
#   The value of f is: -31
#   four06%
#### Assumptions about the MIPS variables: #########################
#  f is in $s0 and the parameters follow MIPS conventions:
#    g, h, i and j are in registers $a0 through $a3
###################################
                                  #
        .globl main               #
main:                             # make main a global label
        addu    $s7, $0, $ra      # save the return address
                                  #
        addi    $s0, $0, -1       # initialize $s0 to -1
        addi    $a0, $0, 5        # g = 5
        addi    $a1, $0, -20      # h = -20
        addi    $a2, $0, 13       # i = 13
        addi    $a3, $0, 3        # j = 3
        jal     leaf_example      # call the function leaf_example
        add     $s0, $0, $v0      # set f to the computed value
                                  #
#### Now print out a message and value of f ########################
        li      $v0, 4            # print_str (system call 4)
        la      $a0, Mess         # takes the addr of string as an arg
        syscall                   #
                                  #
        li      $v0, 1            # print_int (system call 1)
        add     $a0, $0, $s0      # put value to print in $a0
        syscall                   #
                                  #
                                  # usual stuff at the end of the main
        addu    $ra, $0, $s7      # restore the return address
        jr      $ra               # return from the main program
                                  #
                                  #
```

```
#### Start of function Leaf_example ###################################
leaf_example:
        addi    $sp, $sp, -12    # make space on the stack for three items
        sw      $t1, 8($sp)     # save register $t1 for use afterwards
        sw      $t0, 4($sp)     # save register $t2 for use afterwards
        sw      $s0, 0($sp)     # save register $s0 for use afterwards
        add     $t0, $a0, $a1   # register $t0 contains g + h
        add     $t1, $a2, $a3   # register $t1 contains i + j
        sub     $s0, $t0, $t1   # f = (g + h) - (i + j) = $t0 - $t1
        add     $v0, $s0, $0    # returns f
        lw      $s0, 0($sp)     # restore register $s0 for caller
        lw      $t0, 4($sp)     # restore register $t0 for caller
        lw      $t1, 8($sp)     # restore register $t1 for caller
        addi    $sp, $sp, 12    # adjust the stack for return (delete 3 items)
        jr      $ra             # jump back to calling program
#### End of function Leaf_example #####################################

        .data
Mess:   .asciiz "\nThe value of f is: " # string to print
##### Output of a run of this MIPS program ############################
#   four06% spim -file leaf.s
#   SPIM Version 6.0 of July 21, 1997
#   Copyright 1990-1997 by James R. Larus (larus@cs.wisc.edu).
#   All Rights Reserved.
#   See the file README for a full copyright notice.
#   Loaded: /usr/local/lib/trap.handler
#
#   The value of f is: -31four06%
######################################################################
```