```
# lab 3 exercise 4-3
f:
    add $t0, $t0, $t0

    # Save $ra and $t0 on stack before calling another function
    addi $sp, $sp, -8
    sw $ra, 0($sp)
    sw $t0, 4($sp)

    jal g

    # Restore $t0 and $ra after function call
    lw $t0, 4($sp)
    lw $ra, 0($sp)
    addi $sp, $sp, 8

    add $t0, $t0, $t0
    jr $ra

# (3a)
# jal f jumps to f, storing return the address in $ra.
# jal g jumps to g, but overwrites $ra with return address for f.
# g executes and jr $ra returns to the saved address (which was for f).
# However, the $ra value from f was lost, so jr $ra in f returns incorrectly.

#(3c)
# Use a stack framework with push and pop operations.
# Every function should push $ra before calling another function, execute its code, and then pop the saved values before returning.

# (3e) f should push all the registers it needs onto the stack before calling g, and pop them after.
```