

```
1: # lab 3 exercise 4-2
2:      .text
3:      jal f          # Jump to f, store return address in $ra
4:
5:      add $t0, $t0, $t0
6:      j exit          # Jumps to exit
7:
8: f:
9:      add $t0, $t0, $t0
10:     add $t0, $t0, $t0
11:     jr $ra
12:
13: exit:
14:     # (2a) jal f jumps to f, does lines 9 and 10, and then jr $ra, which returns to the
    instruction after jal (line 5).
15:     # line 5 adds and then j exit jumps to line 13, all this occurs due to how functions
    are set up in assembly.
16:
17:     # (2b) The jal instruction is called jump-and-link because it jumps to the function f
18:     # while also linking the return address (stores PC + 4 in $ra).
19:
20:
21:
```