

```
1: # lab 3 exercise 4-1
2:     .text
3:     add $t0, $t0, $t0
4:     jal f
5:     # jump back here
6:
7: f:   add $t0, $t0, $t0
8:     add $t0, $t0, $t0
9:     jr $ra
10:
11: # (a) jal f stores the return address in $ra and jumps to f
12: # (b) jr $ra sets the PC to the value stored in $ra
13: # (c) Overall, the program uses function calls with jal and returning with jr.
14:     # jal f saves the return address in $ra and jumps to f.
15:     # jr $ra brings execution to line 5 (after jal).
16:     # This program runs the way it does because of how functions are set up in assembly.
```