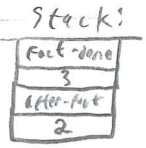
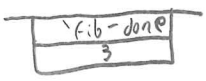
5.5! Let's simulate for n=3 as the second-smallest non-trivial input possible.

(a) Factorial simulation

```
c ← 'fact-done
n ← 3
save c
save n
n ← n-dec (2)
c ← 'after-fact
save c
save n
n ← n-dec (1)
c ← 'after-fact
branch base-case
val ← 1
goto c ('after-fact)
restore n (n ← 2)
restore c (c ← 'after-fact)
val ← val * n (1·2=2)
goto c ('after-fact)
restore n (n ← 3)
restore c (c ← 'fact-done)
val ← val * n (2·3=6)
goto c ('fact-done)
done.
```

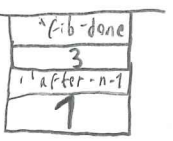


@8:

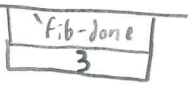


@14:

@24:



@32:



@39:

@46:

```
43 n ← (reg val) (1)
44 restore val (1)
45 restore c ('fib-done)
46 val ← val + n (1+1)
47 goto c ('fib-done)
48 done.
```

(b) Fib-simulation

```
1  c ← 'fib-done
2  n ← 3
3  nobranch (3<2?)
4  save c
5  c ← 'afterfib-n-1
6  save n
7  n ← n-dec (2)
8  goto 'fib-loop
9  nobranch (2<2?)
10 save c
11 c ← 'afterfib-n-1
12 save n
13 n ← n-dec (1)
14 goto 'fib-loop
15 branch 'immediate-answer (1<2?)
16 val ← (reg n) (1)
17 goto c
18 restore n (n ← 2)
19 restore c (c ← 'afterfib-n-1)
20 n ← n-2 (0)
21 save c
22 c ← 'afterfib-n-2
23 save val
24 goto 'fib-loop
25 branch 'immediate-answer (0<2?)
26 val ← (reg n) (0)
27 goto c ('afterfib-n-2)
28 n ← (reg val) (0)
29 restore val (1)
30 restore c ('after-n-1)
31 val ← val + n (1+0)
32 goto c ('after-n-1)
33 restore n (3)
34 restore c
35 n ← n-2 (1)
36 save c
37 c ← 'afterfib-n-2
38 save val
39 goto 'fib-loop
40 branch 'immediate-answer (1<2?)
41 val ← (reg n) (1)
42 goto c ('afterfib-n-2)
```