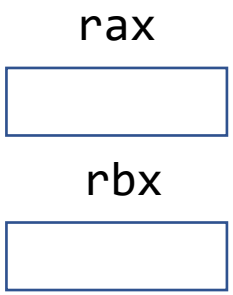
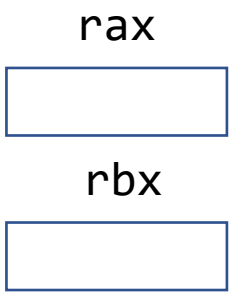
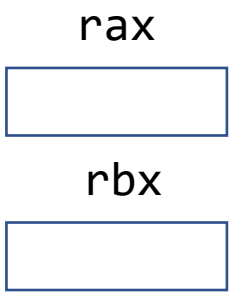
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```



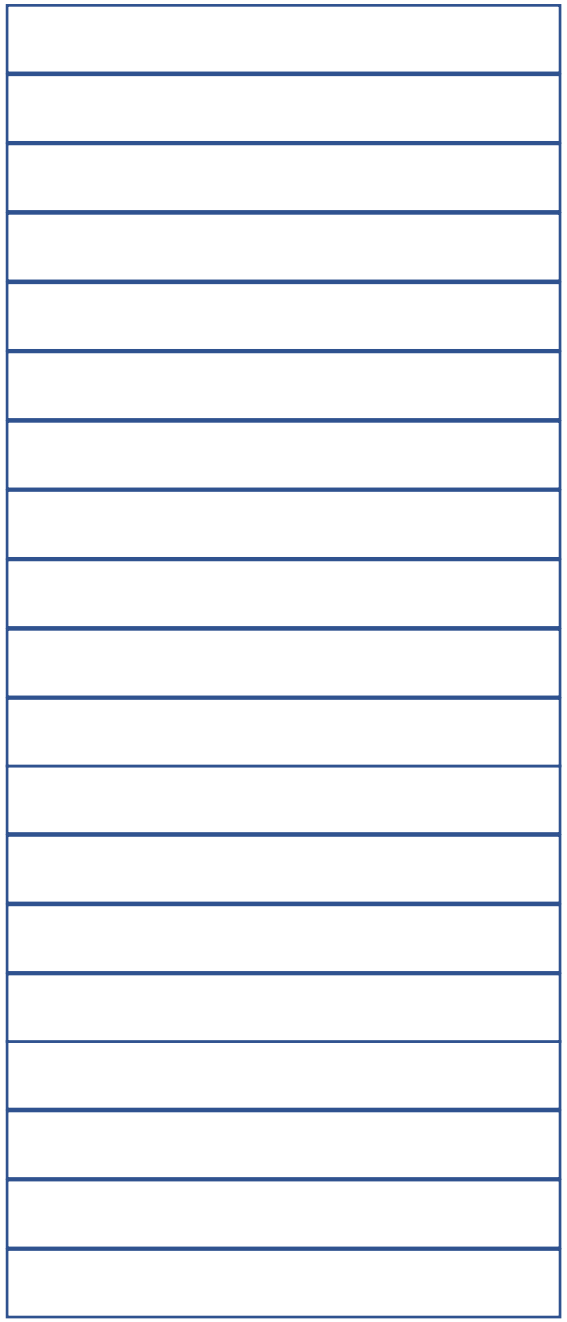
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

3 (t: 110)

rbx

[rsp]



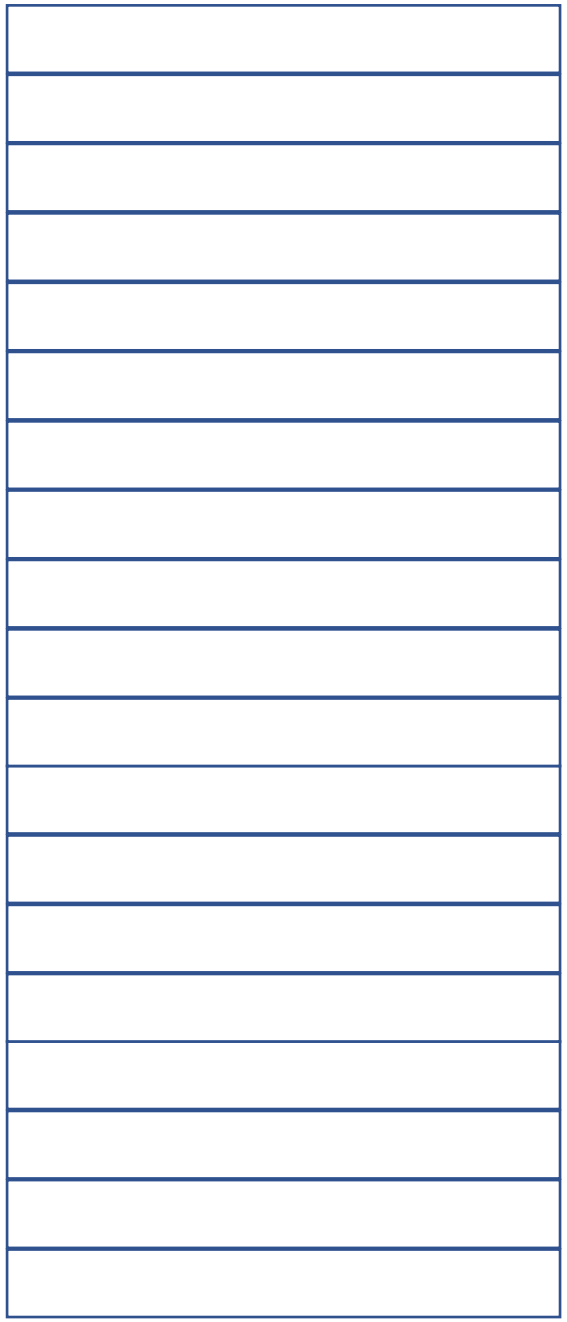
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

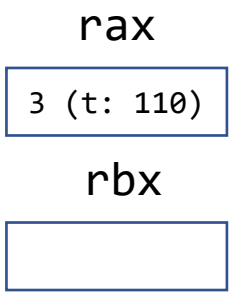
3 (t: 110)

rbx

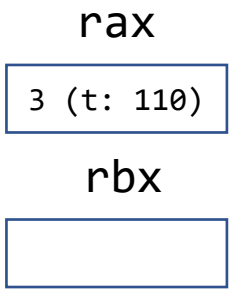
[rsp]



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

3 (t: 110)

rbx

[rsp]

<Ret. Ptr. >

3 (t: 110)

<RDI>

When we run a call instruction, [rsp] is moved one automatically and a return pointer to the *next instruction* to be executed put at that place.

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

3 (t: 110)

rbx

[rsp]

<Ret. Ptr. >

3 (t: 110)

<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

3 (t: 110)

rbx

[rsp]

<Ret. Ptr. >

3 (t: 110)

<RDI>

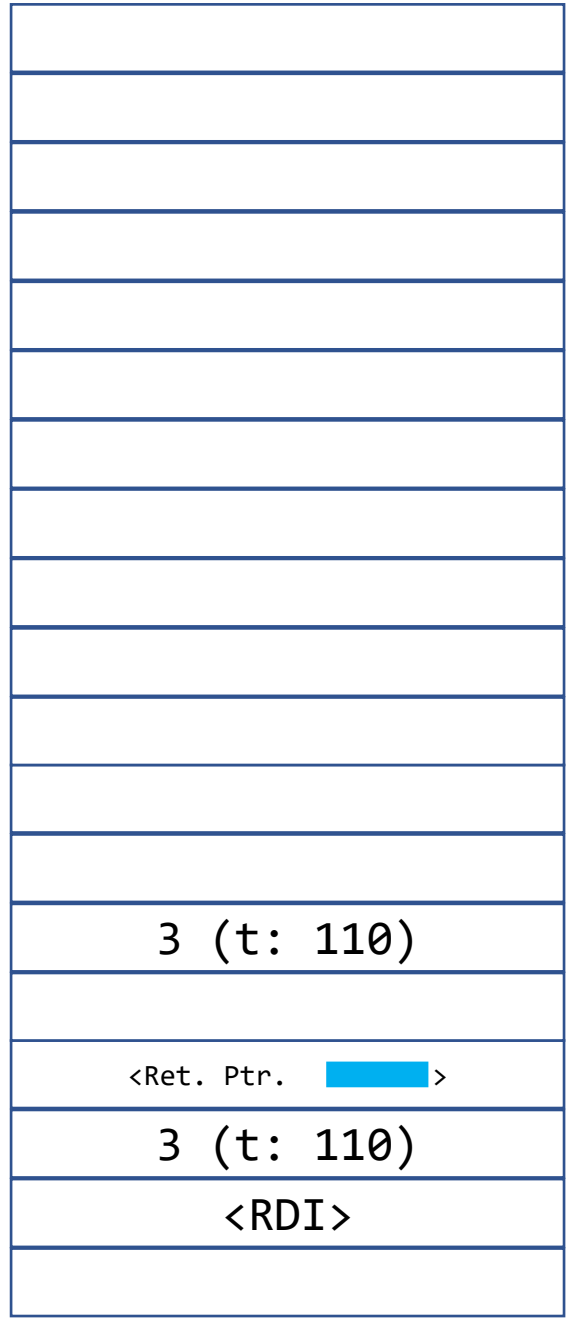
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

3 (t: 110)

rbx

[rsp]



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

(omit)

rbx

[rsp]

3 (t: 110)
<Ret. Ptr. >
3 (t: 110)
<RDI>

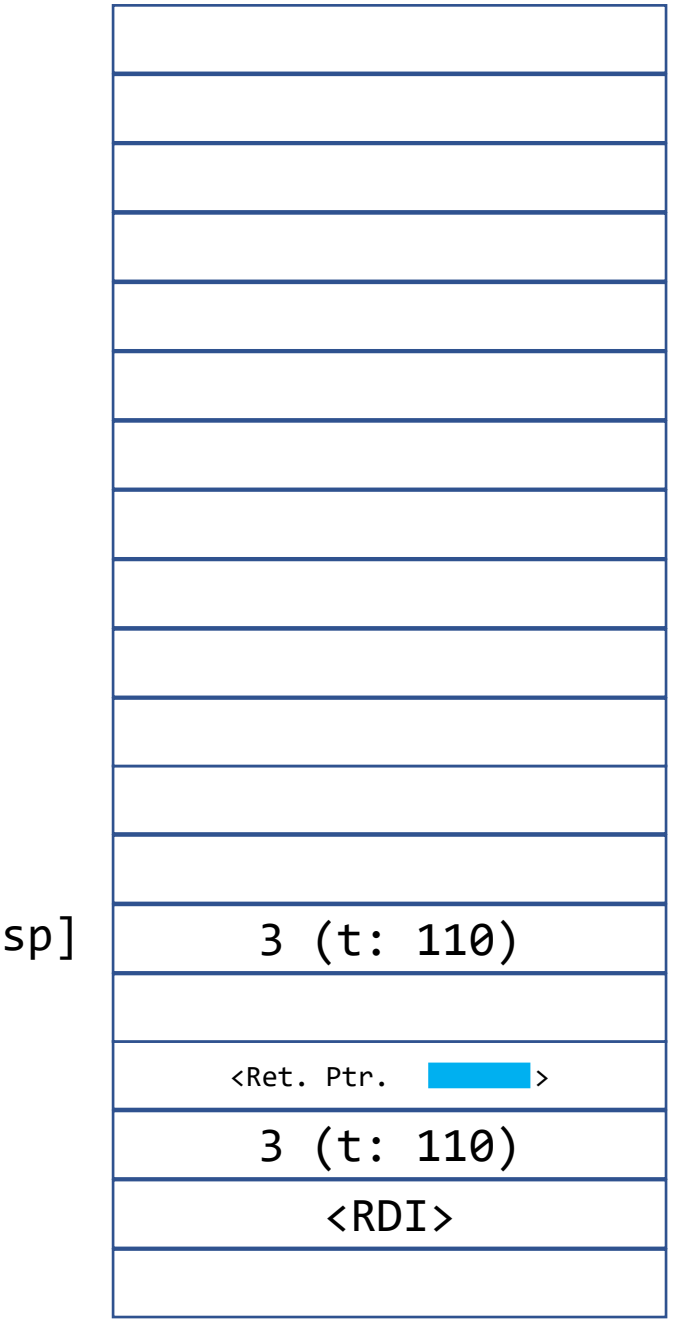
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

3 (t: 110)

rbx

[rsp]



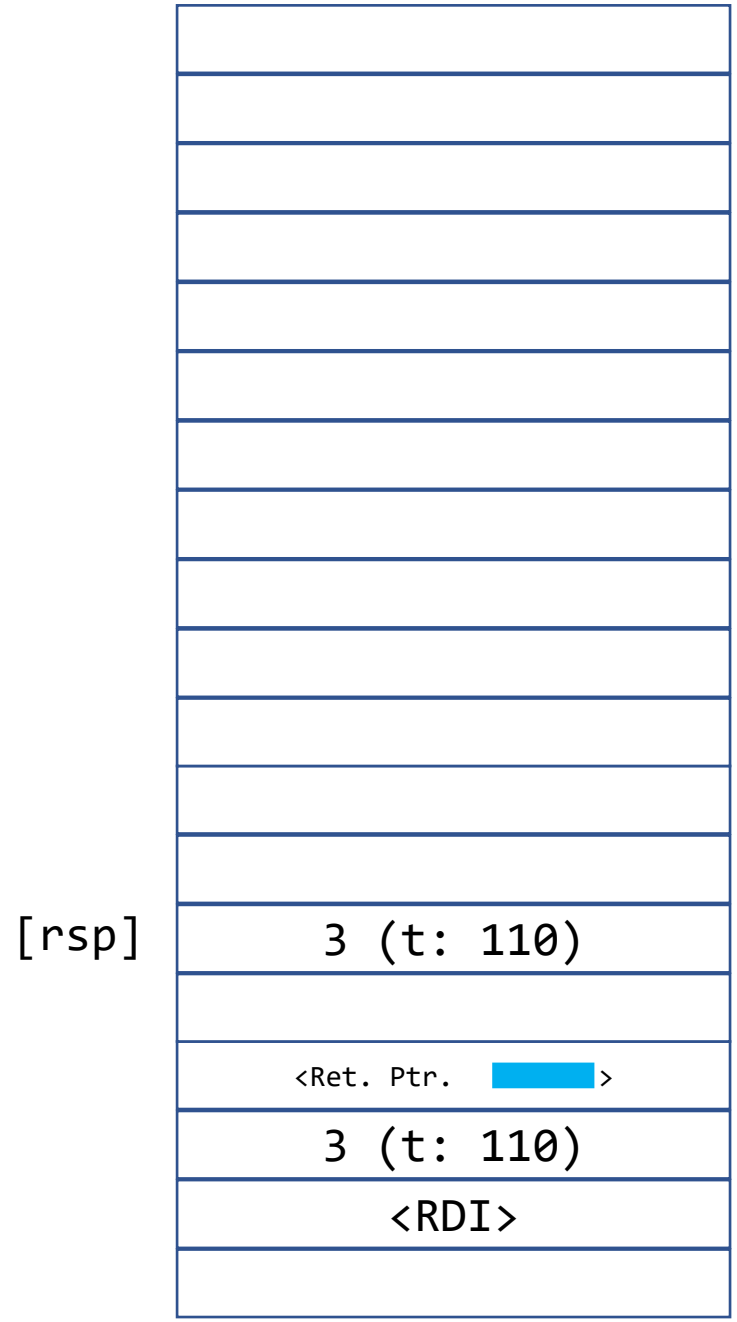
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

3 (t: 110)

rbx

[rsp]



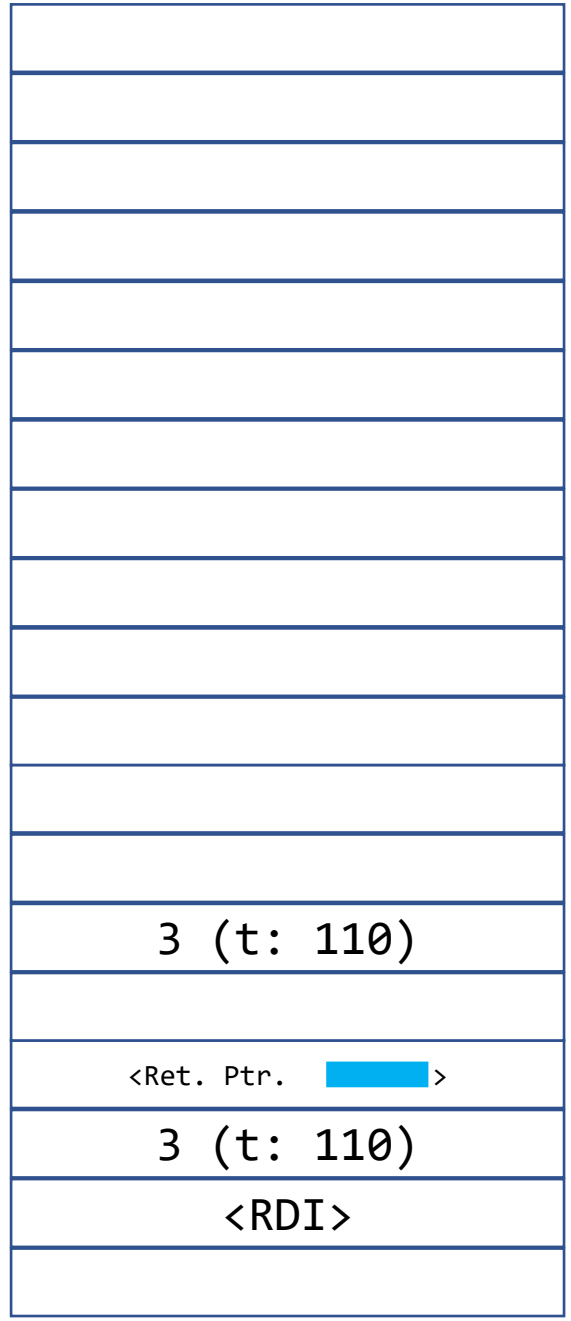
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

3 (t: 110)

rbx

[rsp]



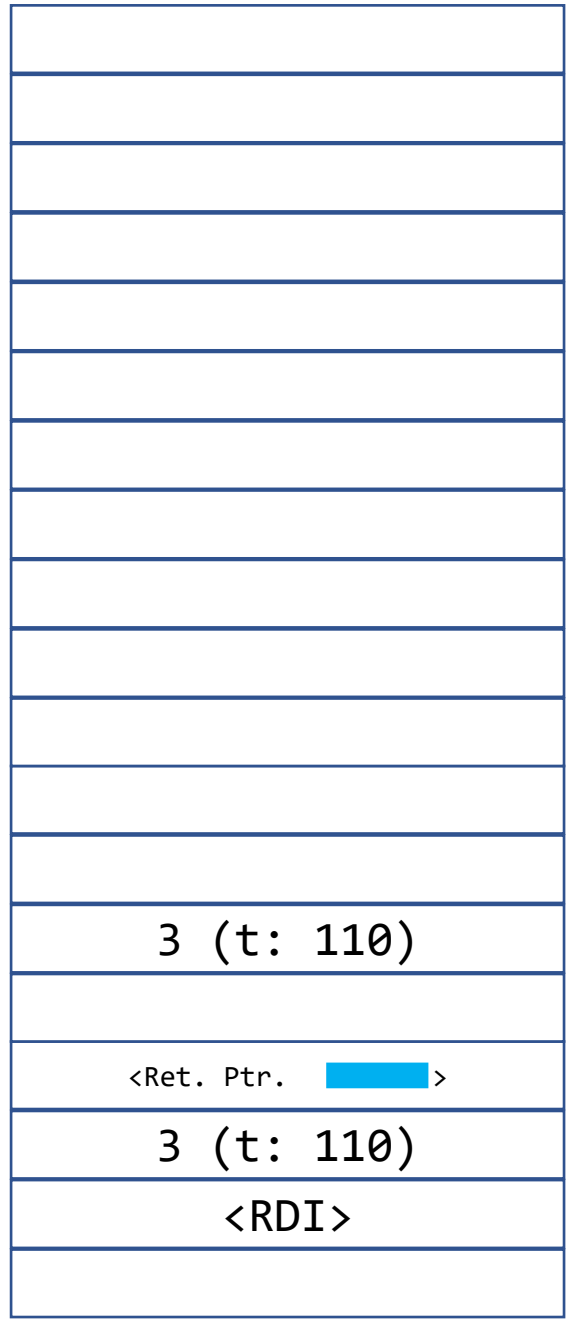
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

2 (t: 100)

rbx

[rsp]



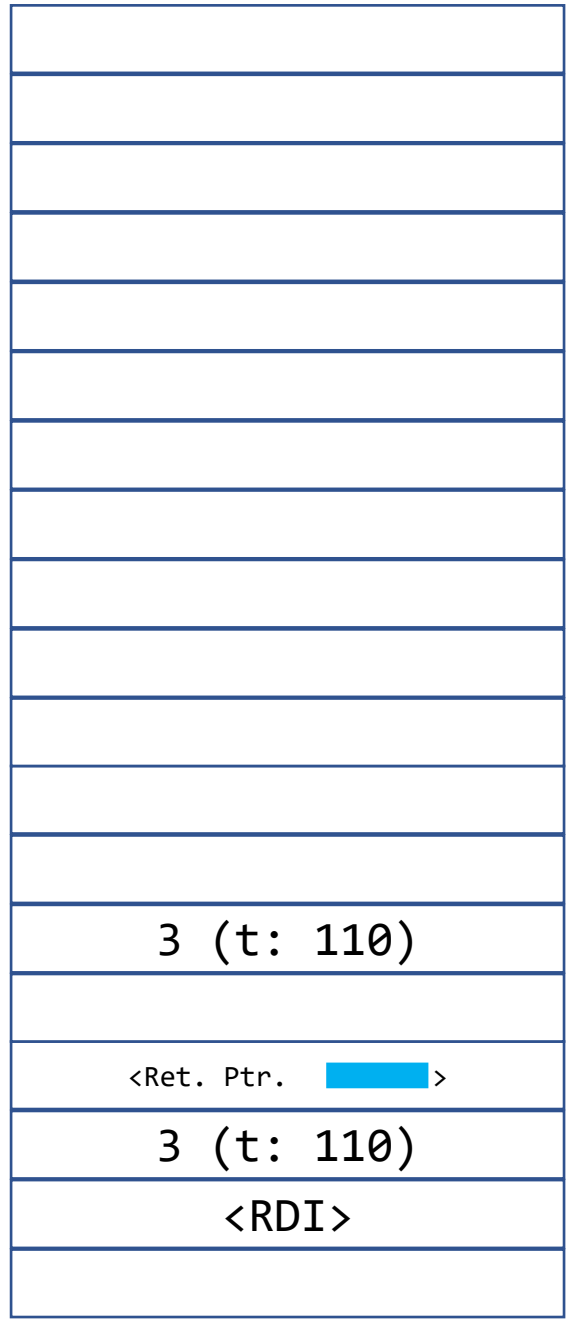
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

2 (t: 100)

rbx

[rsp]

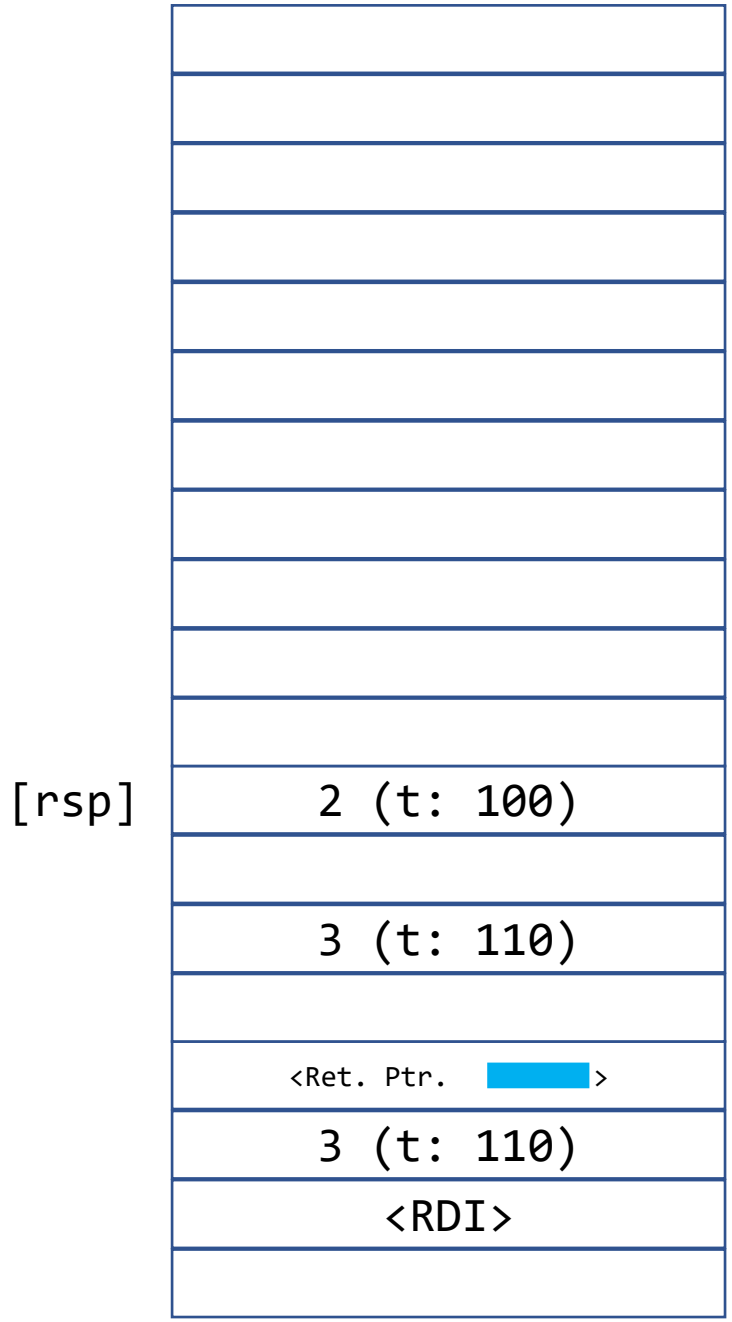


```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

2 (t: 100)

rbx

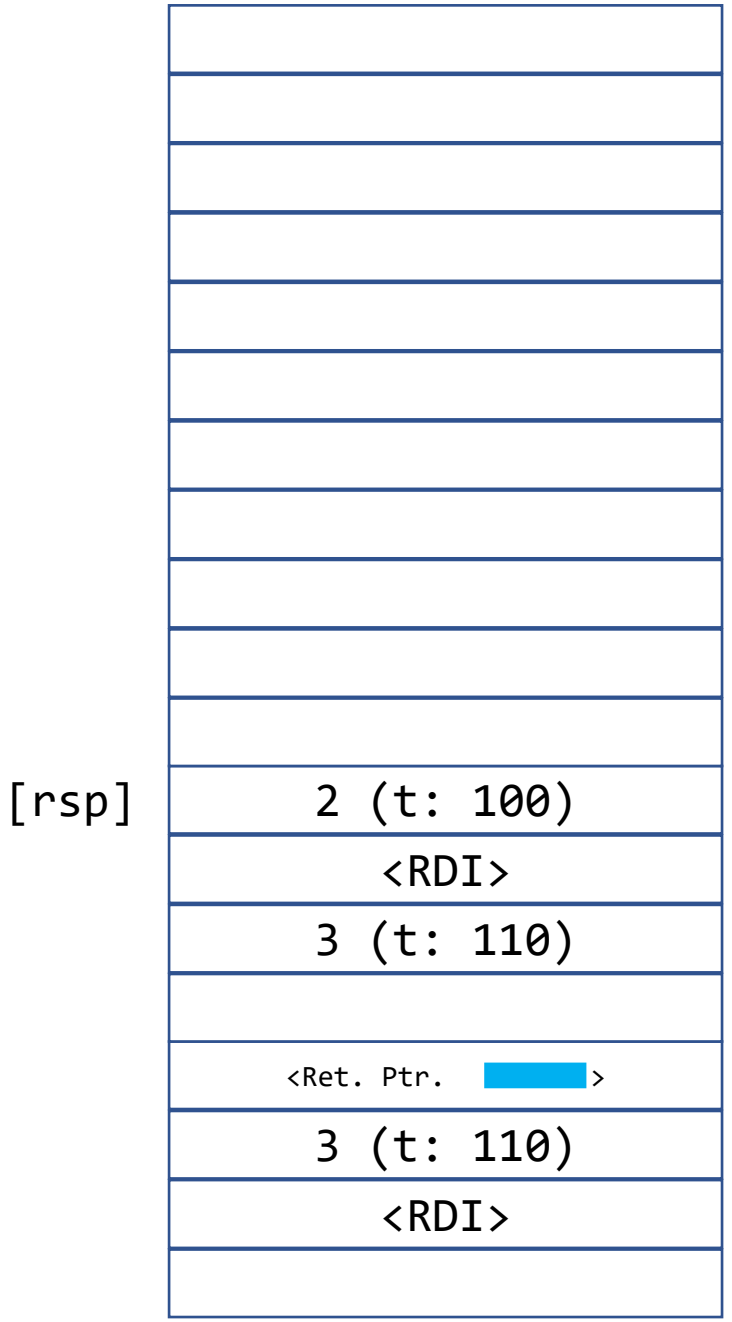


```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

2 (t: 100)

rbx



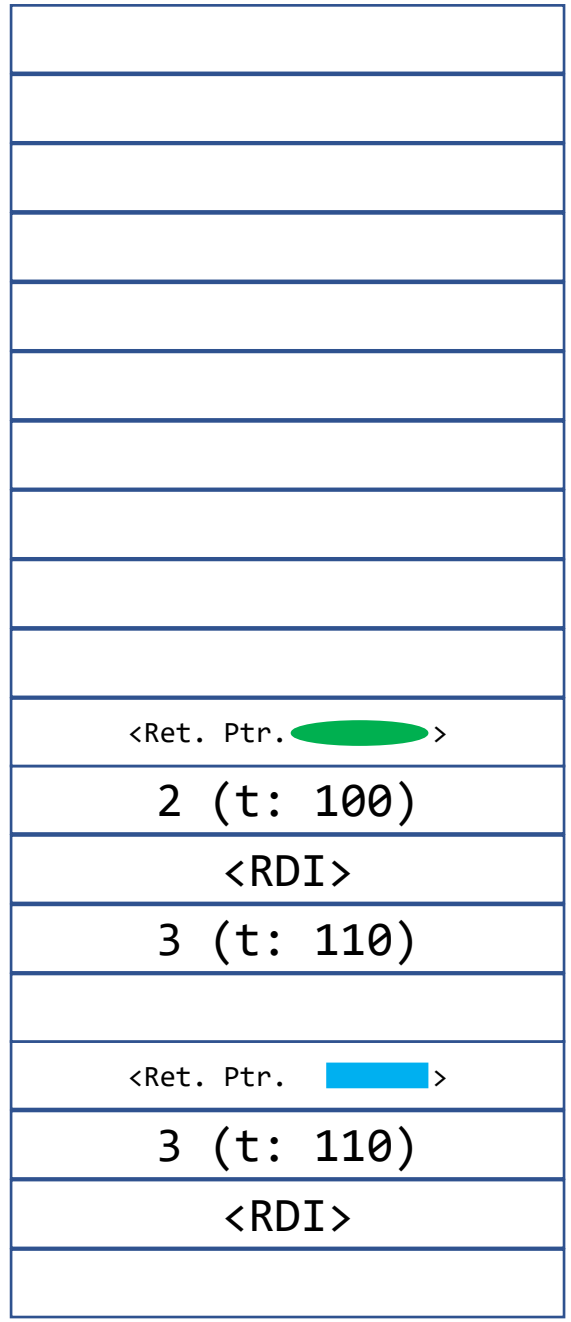
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

2 (t: 100)

rbx

[rsp]



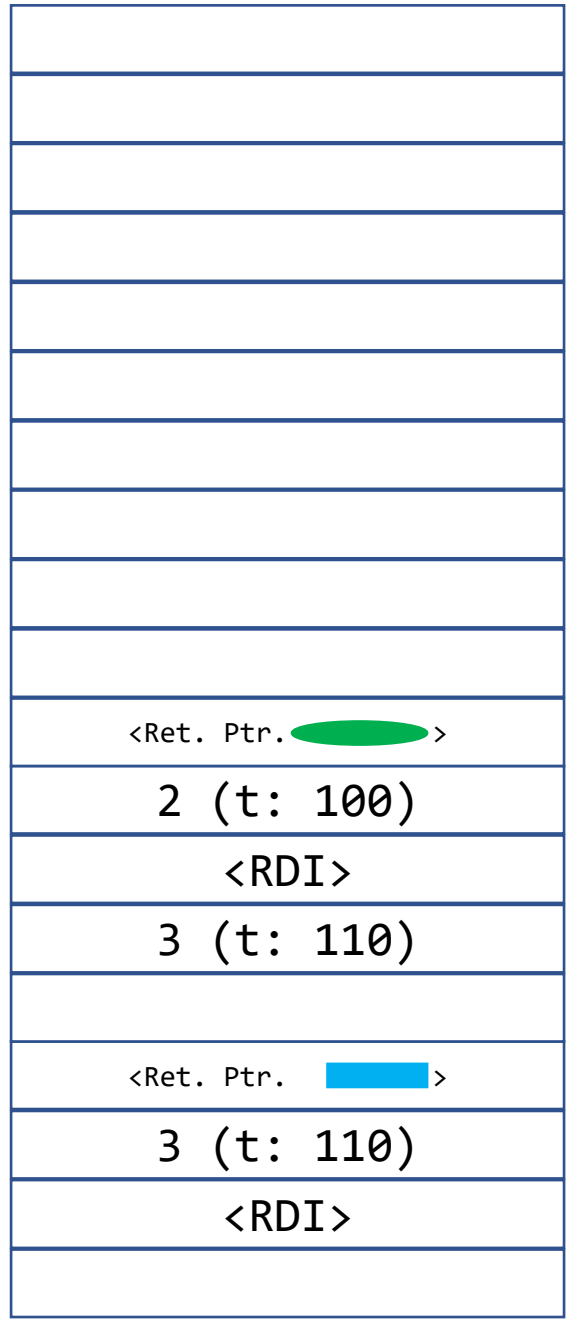
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

2 (t: 100)

rbx

[rsp]



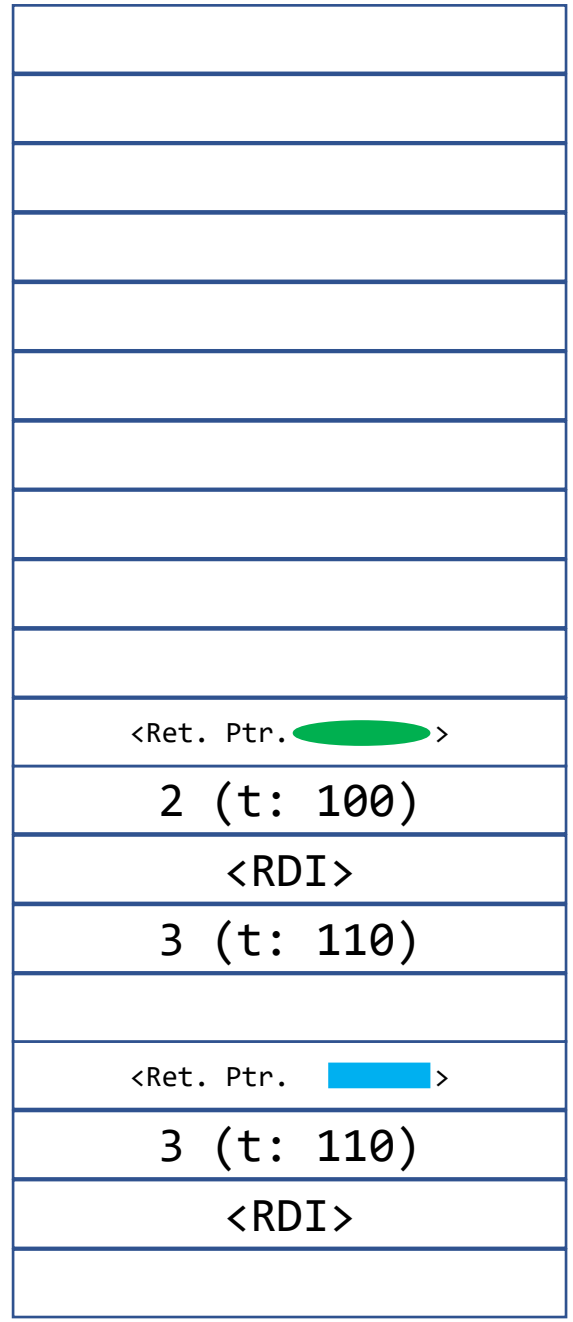
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

2 (t: 100)

rbx

[rsp]



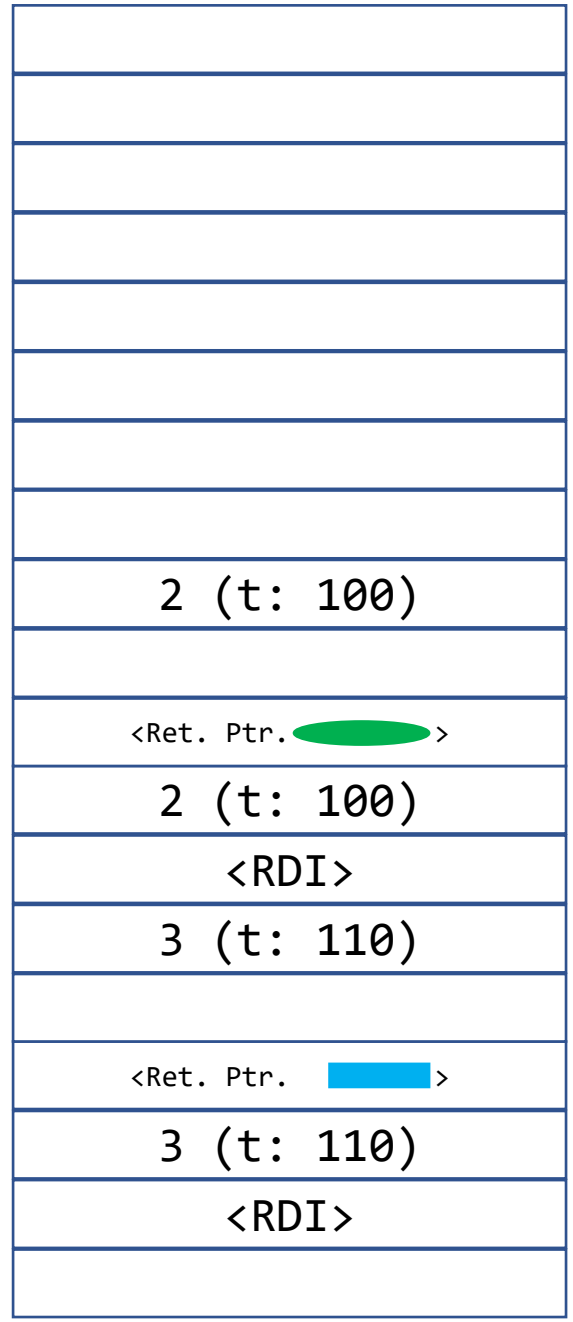
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

2 (t: 100)

rbx

[rsp]



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

(omit)

rbx

[rsp]

2 (t: 100)

<Ret. Ptr. >

2 (t: 100)

<RDI>

3 (t: 110)

<Ret. Ptr. >

3 (t: 110)

<RDI>

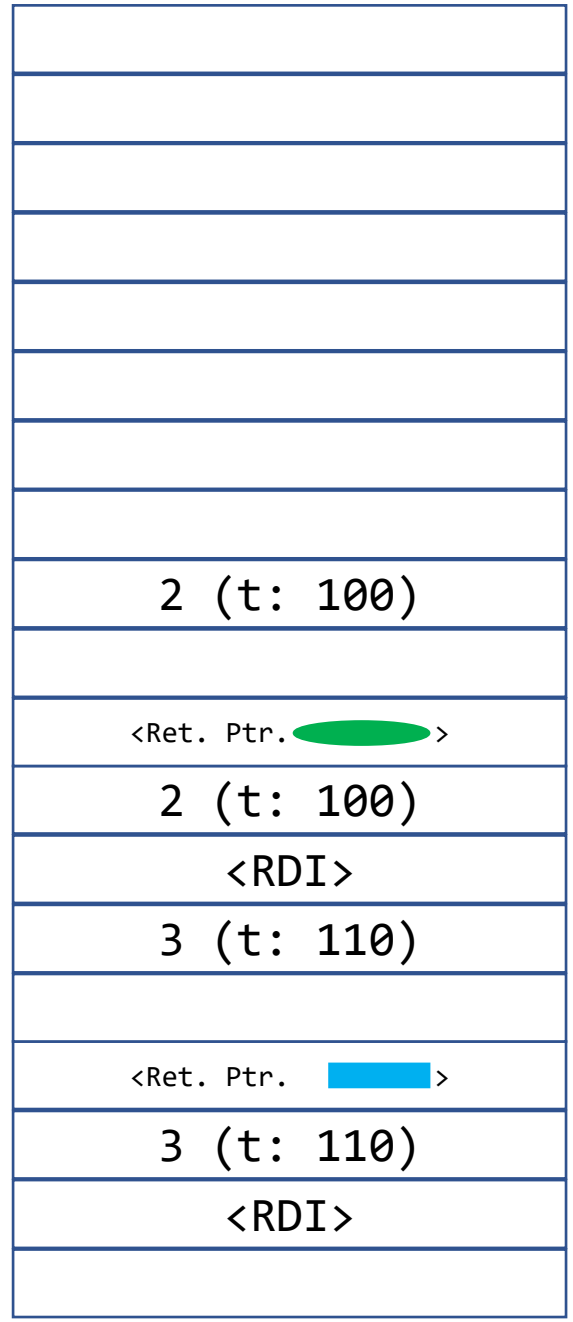
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

2 (t: 100)

rbx

[rsp]

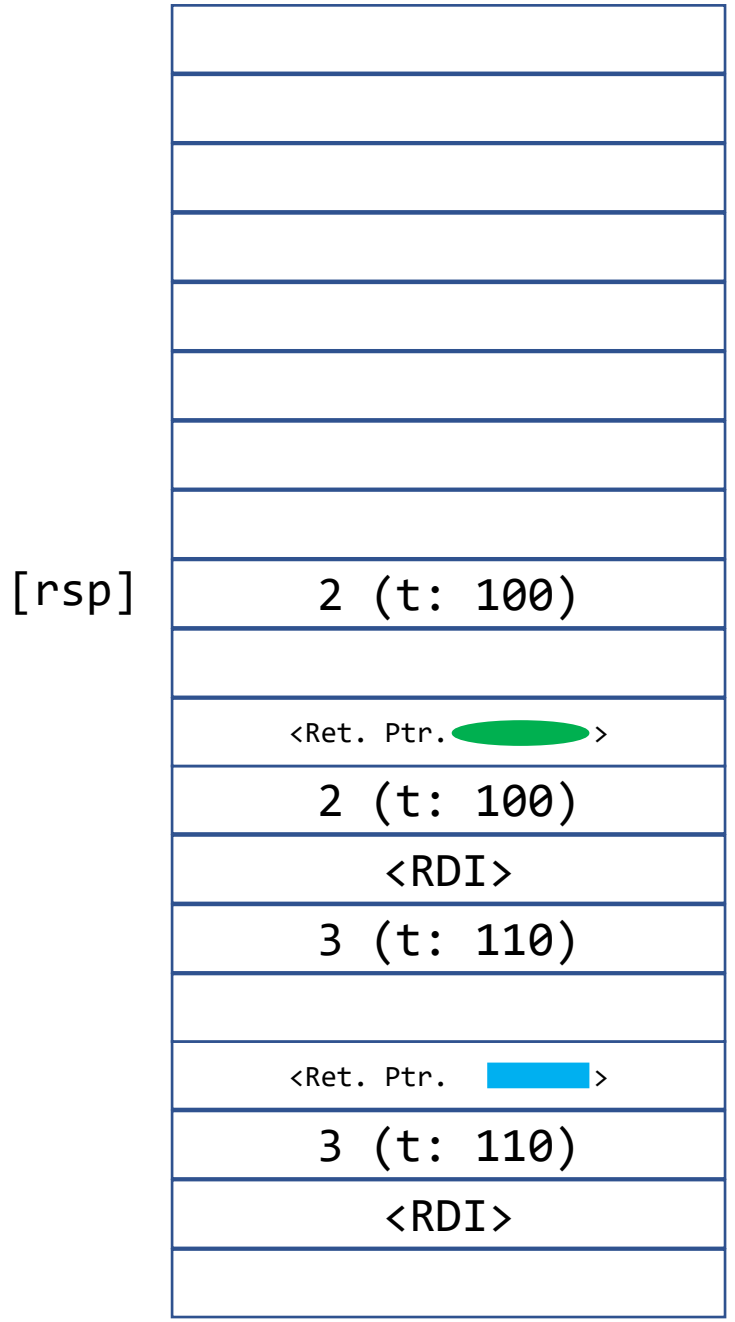


```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

2 (t: 100)

rbx

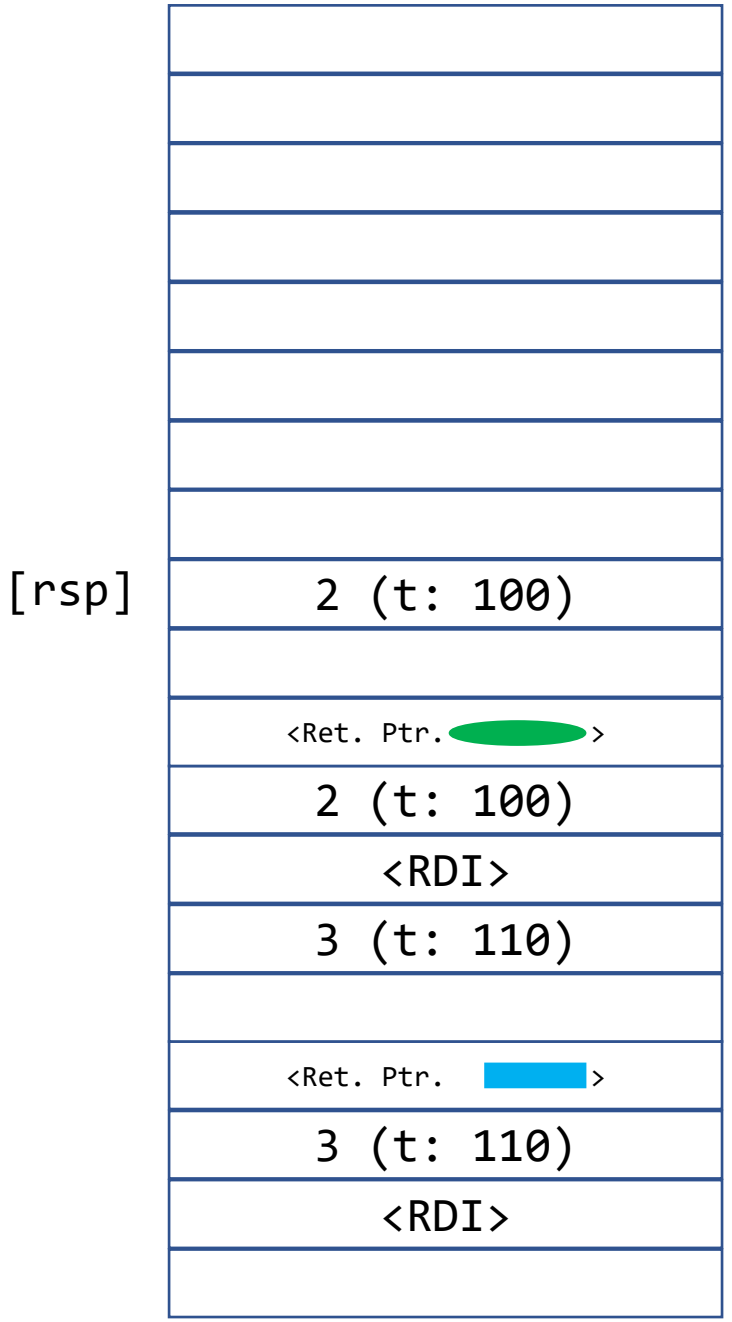


```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

2 (t: 100)

rbx

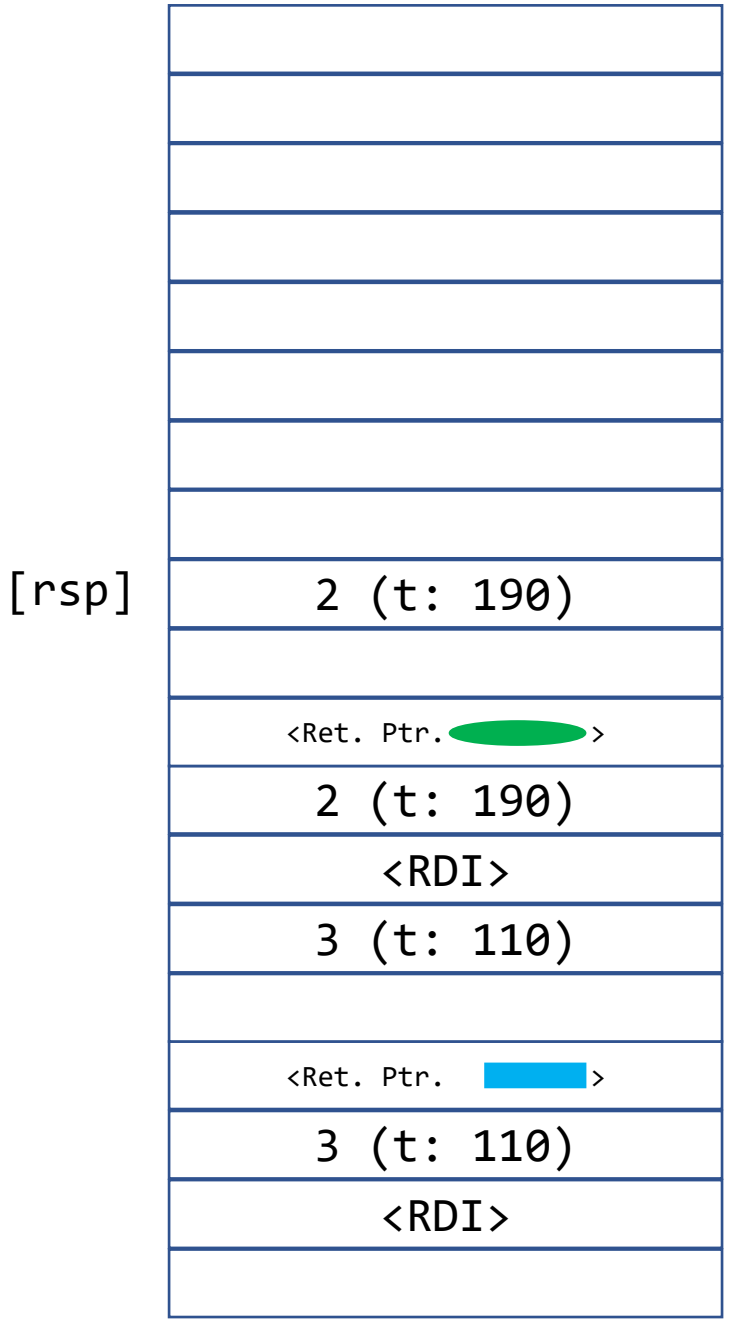


```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

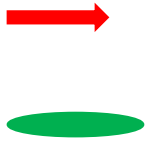
rax

1 (t: 10)

rbx



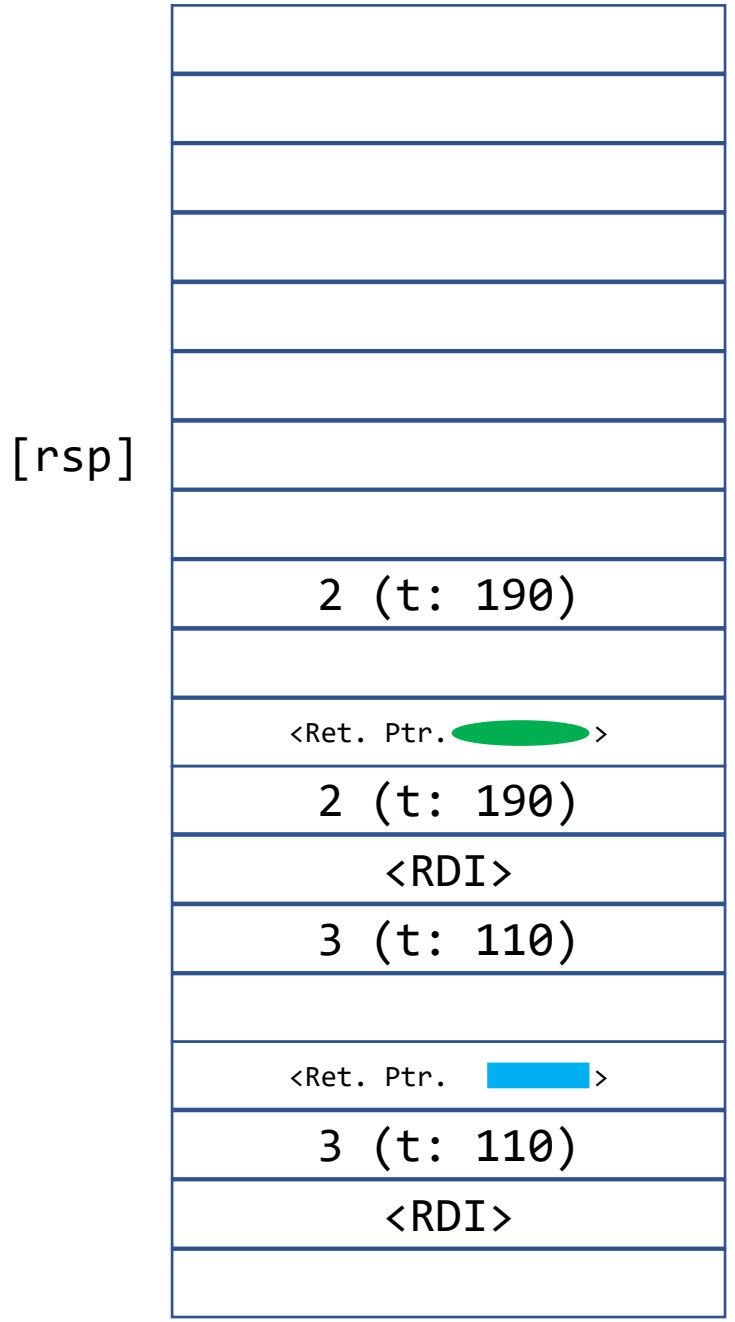
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```



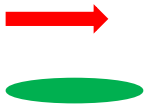
rax

1 (t: 10)

rbx



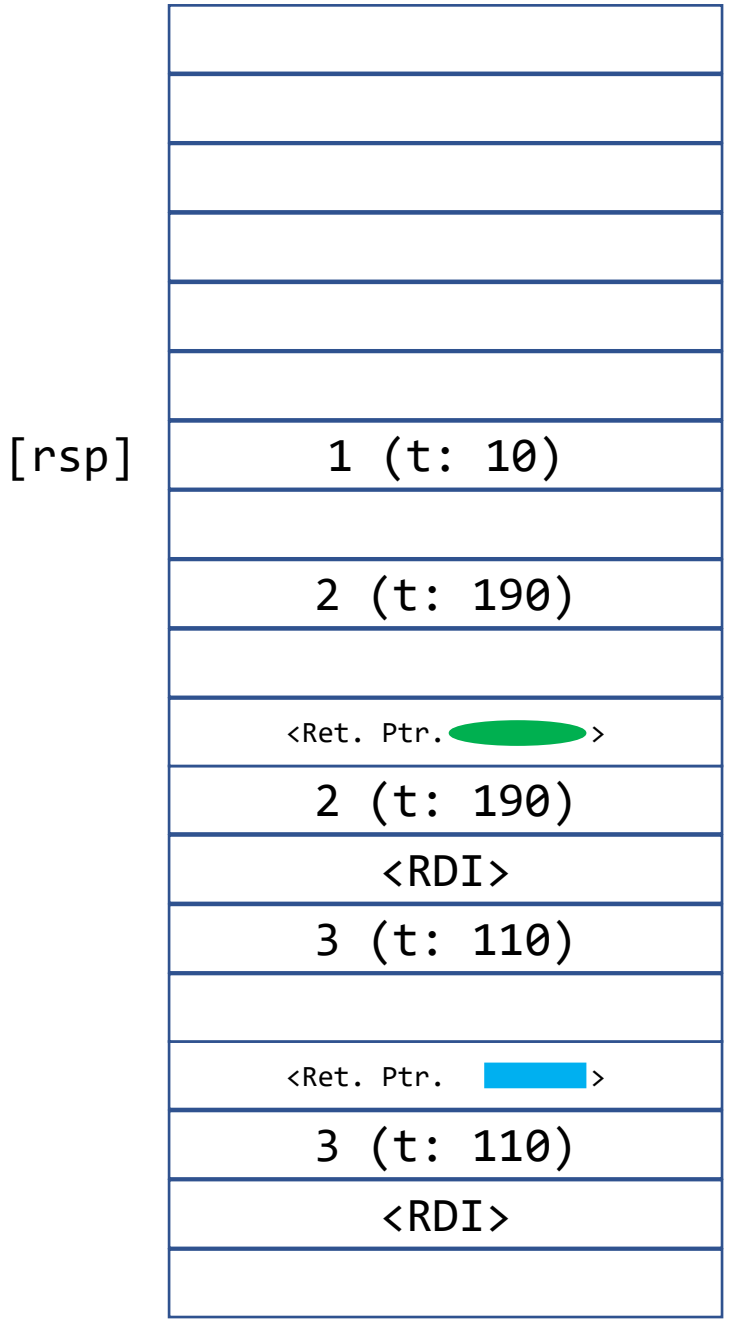
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```



rax

1 (t: 10)

rbx



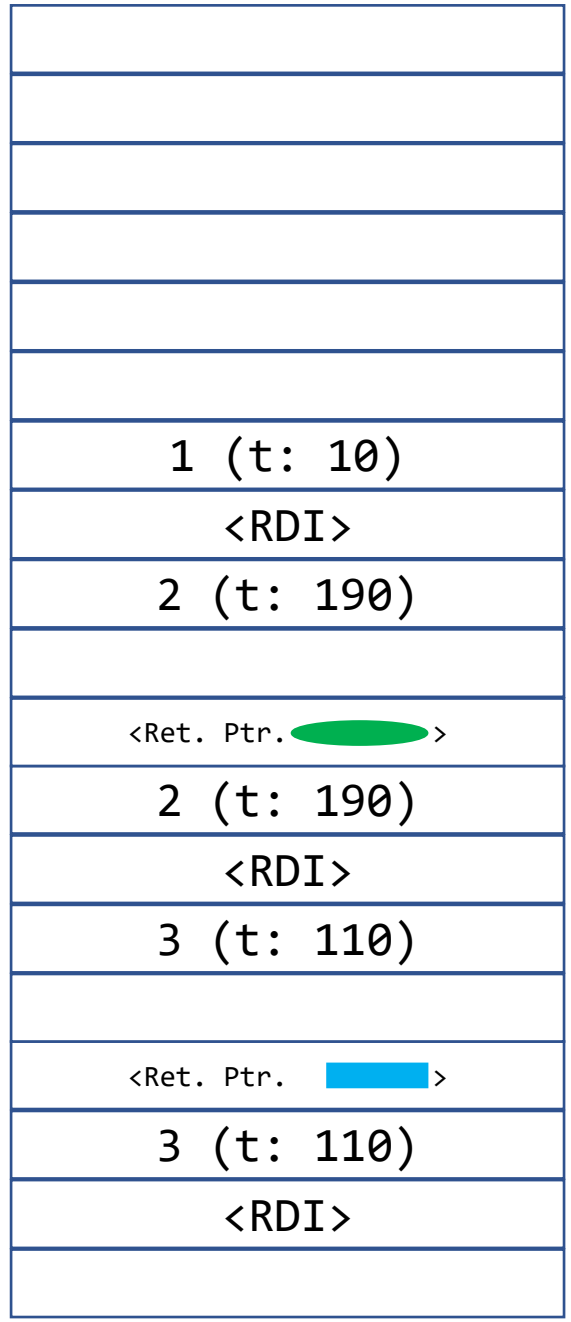
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

1 (t: 10)

rbx

[rsp]



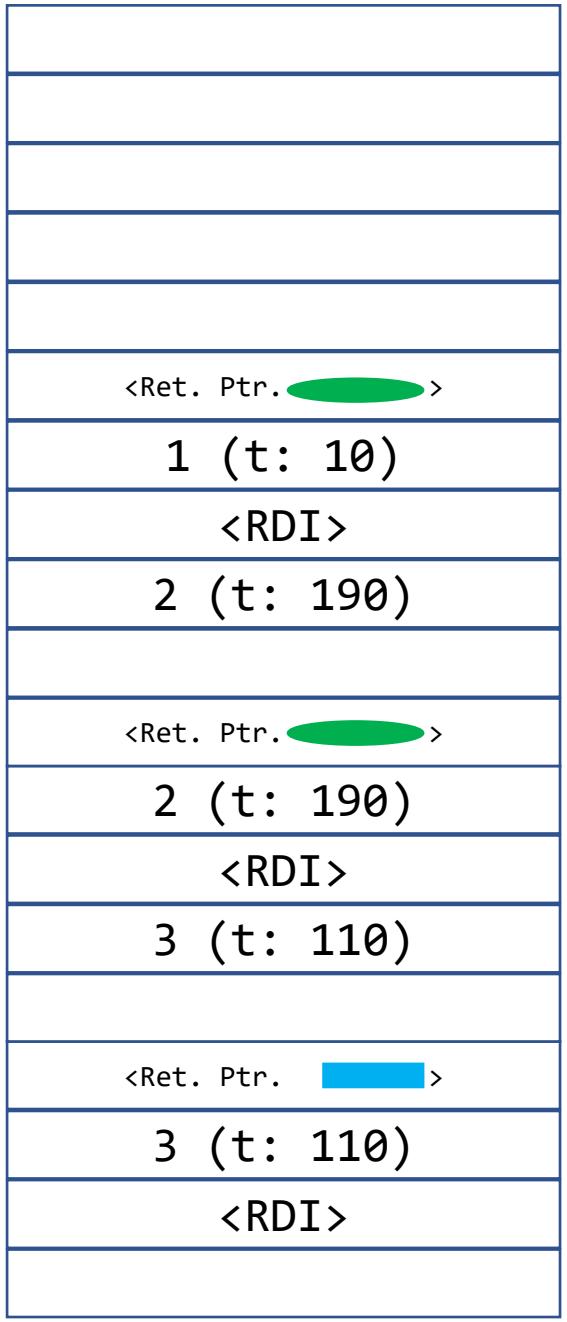
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

1 (t: 10)

rbx

[rsp]



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

1 (t: 10)

rbx

[rsp]

<Ret. Ptr. >

1 (t: 10)

<RDI>

2 (t: 190)

<Ret. Ptr. >

2 (t: 190)

<RDI>

3 (t: 110)

<Ret. Ptr. >

3 (t: 110)

<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

1 (t: 10)

rbx

[rsp]

<Ret. Ptr. >

1 (t: 10)

<RDI>

2 (t: 190)

<Ret. Ptr. >

2 (t: 190)

<RDI>

3 (t: 110)

<Ret. Ptr. >

3 (t: 110)

<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

1 (t: 10)

rbx

[rsp]

<Ret. Ptr. >

1 (t: 10)

<RDI>

2 (t: 190)

<Ret. Ptr. >

2 (t: 190)

<RDI>

3 (t: 110)

<Ret. Ptr. >

3 (t: 110)

<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

(omit)

rbx

[rsp]

<Ret. Ptr. >

1 (t: 10)

<RDI>

2 (t: 190)

<Ret. Ptr. >

2 (t: 190)

<RDI>

3 (t: 110)

<Ret. Ptr. >

3 (t: 110)

<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

1 (t: 10)

rbx

[rsp]

<Ret. Ptr. >

1 (t: 10)

<RDI>

2 (t: 190)

<Ret. Ptr. >

2 (t: 190)

<RDI>

3 (t: 110)

<Ret. Ptr. >

3 (t: 110)

<RDI>

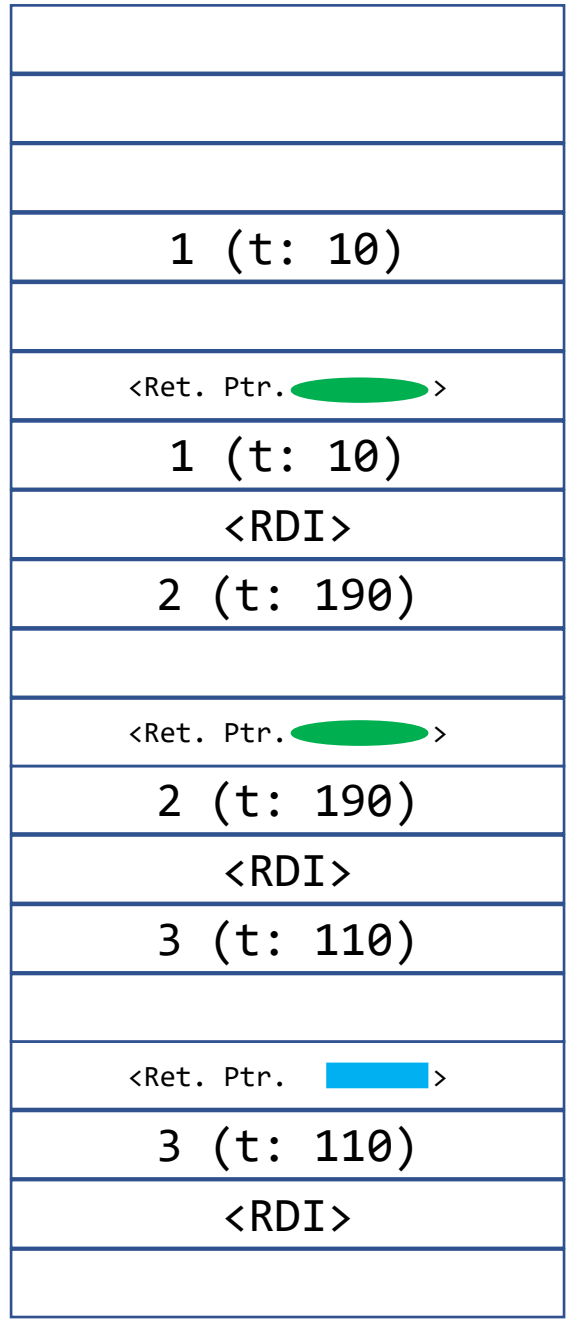
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

1 (t: 10)

rbx

[rsp]



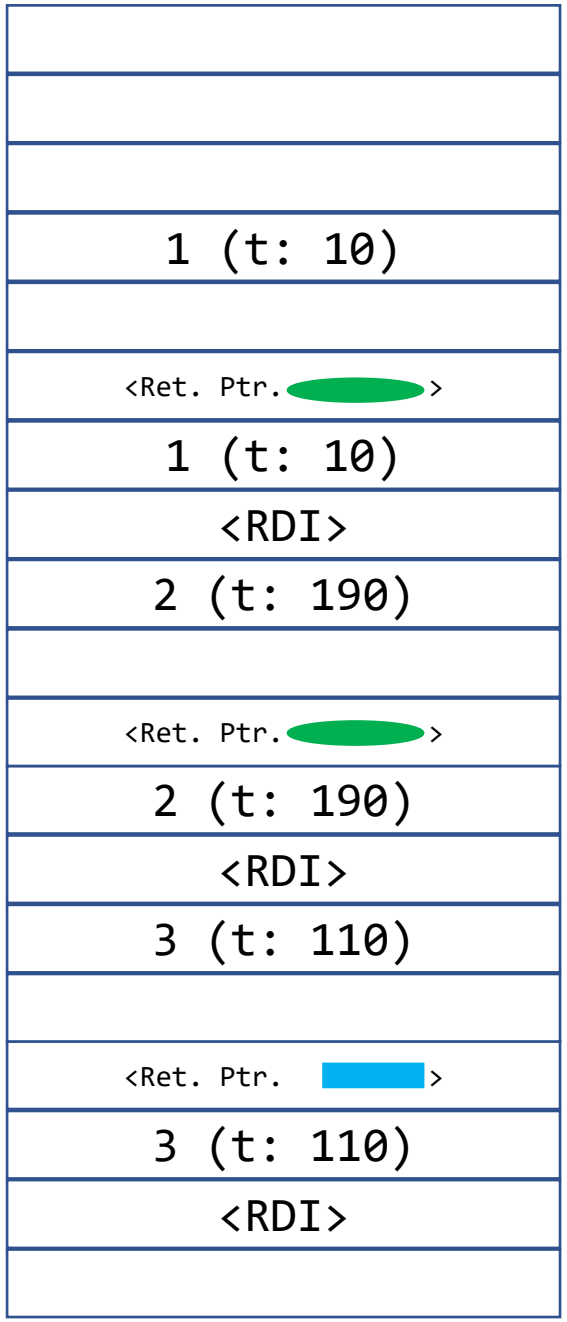
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

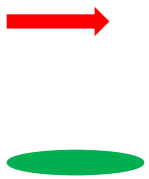
1 (t: 10)

rbx

[rsp]



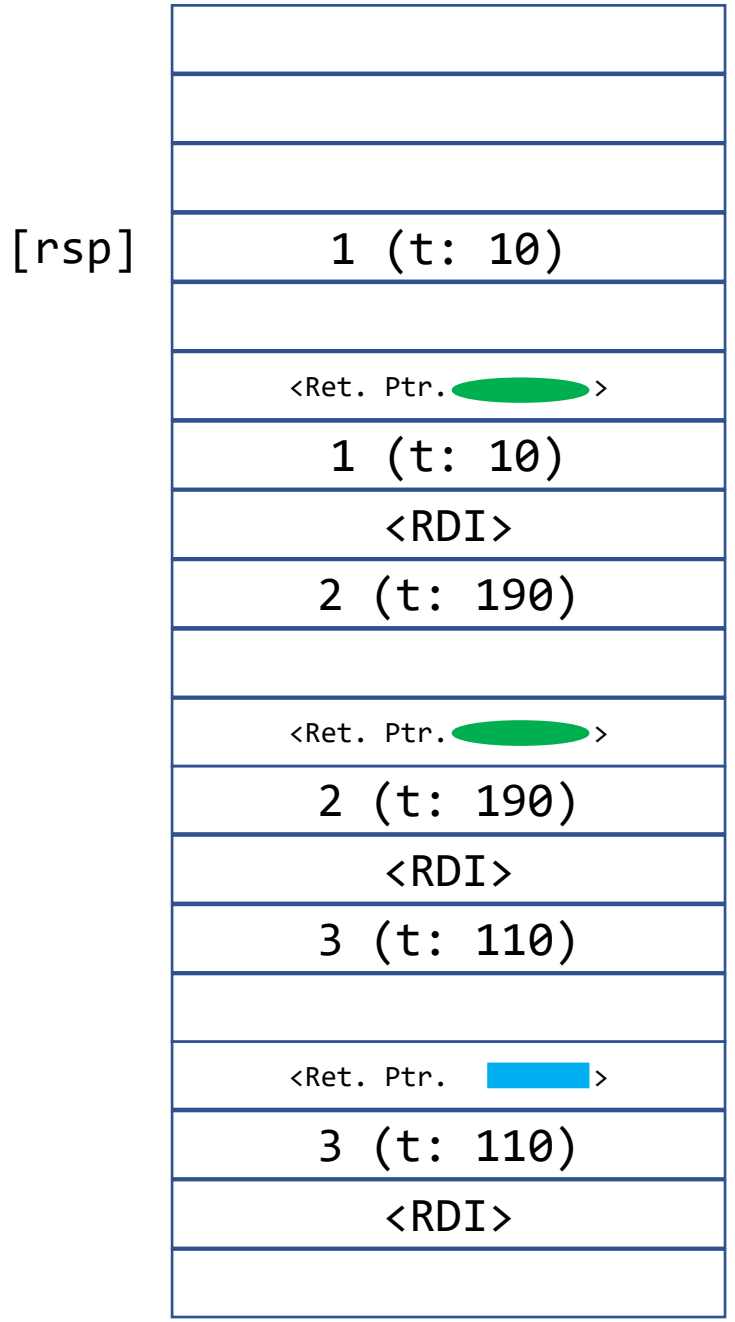
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```



rax

0 (t: 0)

rbx



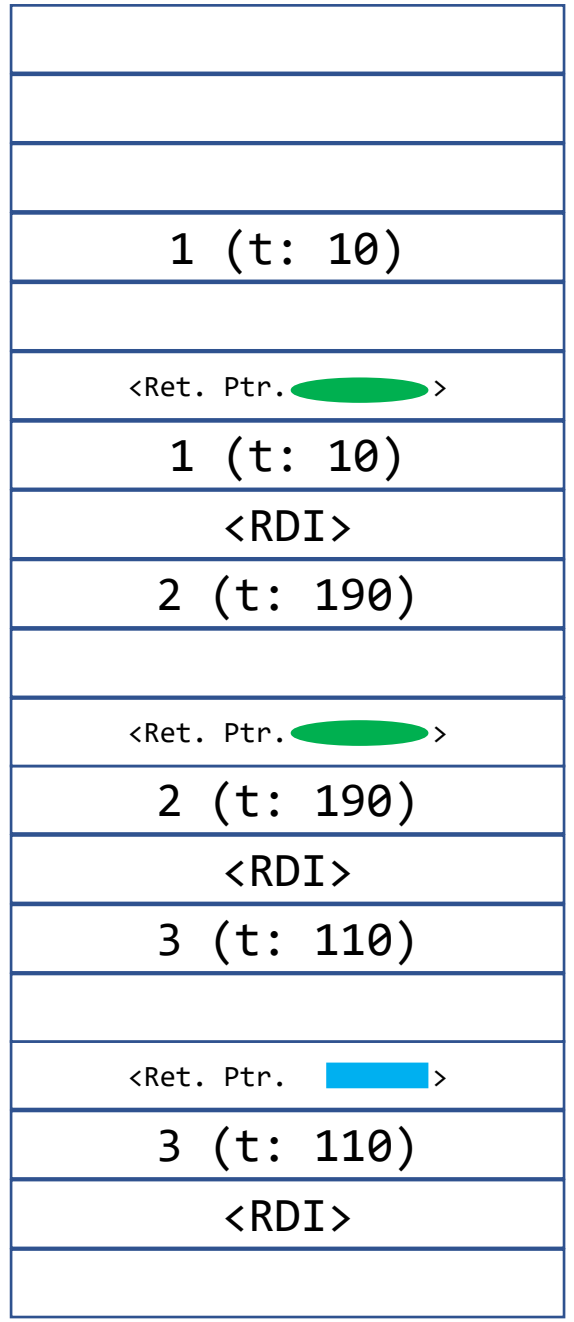
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

0 (t: 0)

rbx

[rsp]



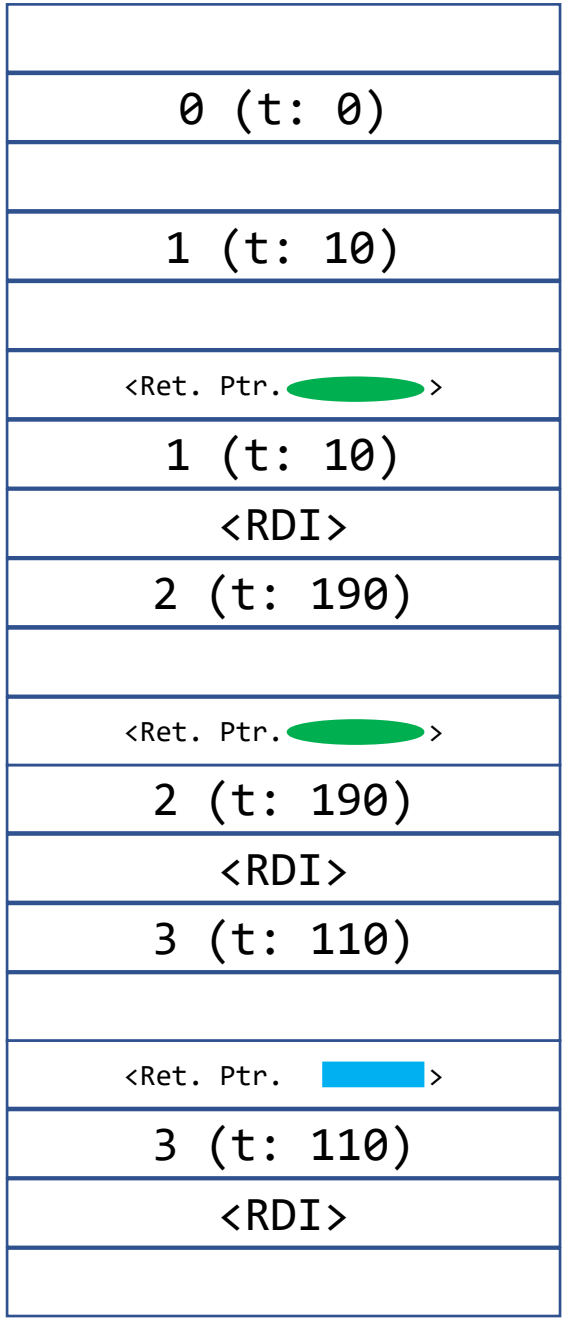
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

0 (t: 0)

rbx

[rsp]



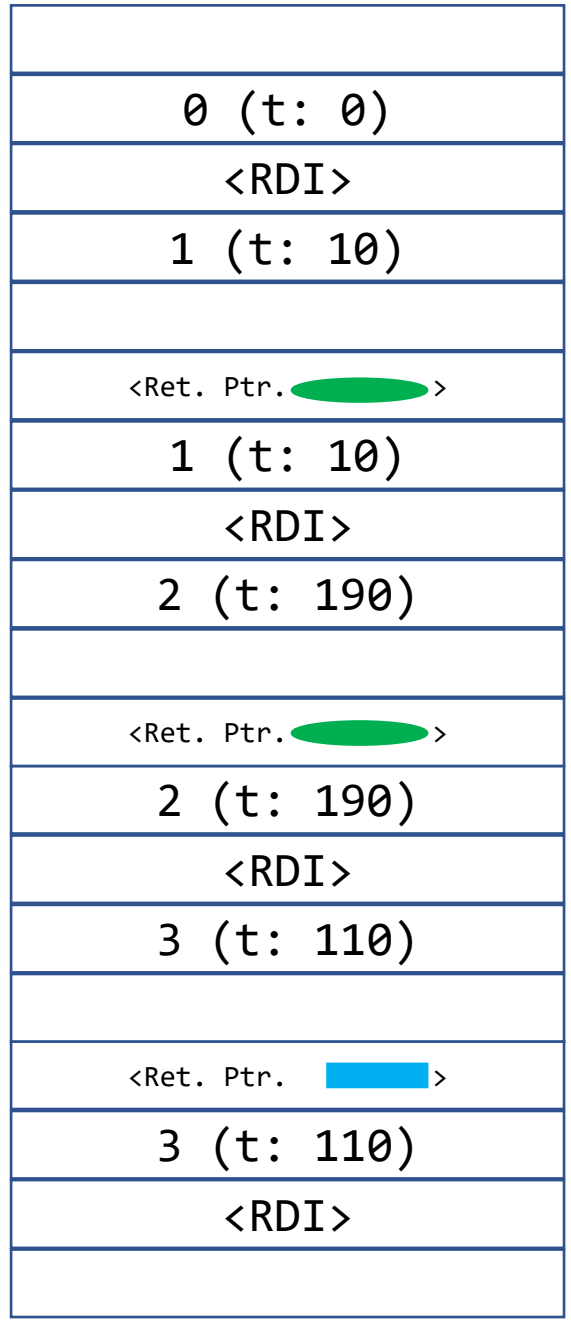
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

0 (t: 0)

rbx

[rsp]



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

0 (t: 0)

rbx

[rsp]

<Ret. Ptr. >
0 (t: 0)
<RDI>
1 (t: 10)
<Ret. Ptr. >
1 (t: 10)
<RDI>
2 (t: 190)
<Ret. Ptr. >
2 (t: 190)
<RDI>
3 (t: 110)
<Ret. Ptr. >
3 (t: 110)
<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

0 (t: 0)

rbx

[rsp]

<Ret. Ptr. >

0 (t: 0)

<RDI>

1 (t: 10)

<Ret. Ptr. >

1 (t: 10)

<RDI>

2 (t: 190)

<Ret. Ptr. >

2 (t: 190)

<RDI>

3 (t: 110)

<Ret. Ptr. >

3 (t: 110)

<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

[rsp]

(Continued from top of right stack frame)

rax

0 (t: 0)

rbx

<Ret. Ptr. >

0 (t: 0)

<RDI>

1 (t: 10)

<Ret. Ptr. >

1 (t: 10)

<RDI>

2 (t: 190)

<Ret. Ptr. >

2 (t: 190)

<RDI>

3 (t: 110)

<Ret. Ptr. >

3 (t: 110)

<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

[rsp]

(Continued from top of right stack frame)

rax

0 (t: 0)

rbx

<Ret. Ptr. >

0 (t: 0)

<RDI>

1 (t: 10)

<Ret. Ptr. >

1 (t: 10)

<RDI>

2 (t: 190)

<Ret. Ptr. >

2 (t: 190)

<RDI>

3 (t: 110)

<Ret. Ptr. >

3 (t: 110)

<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

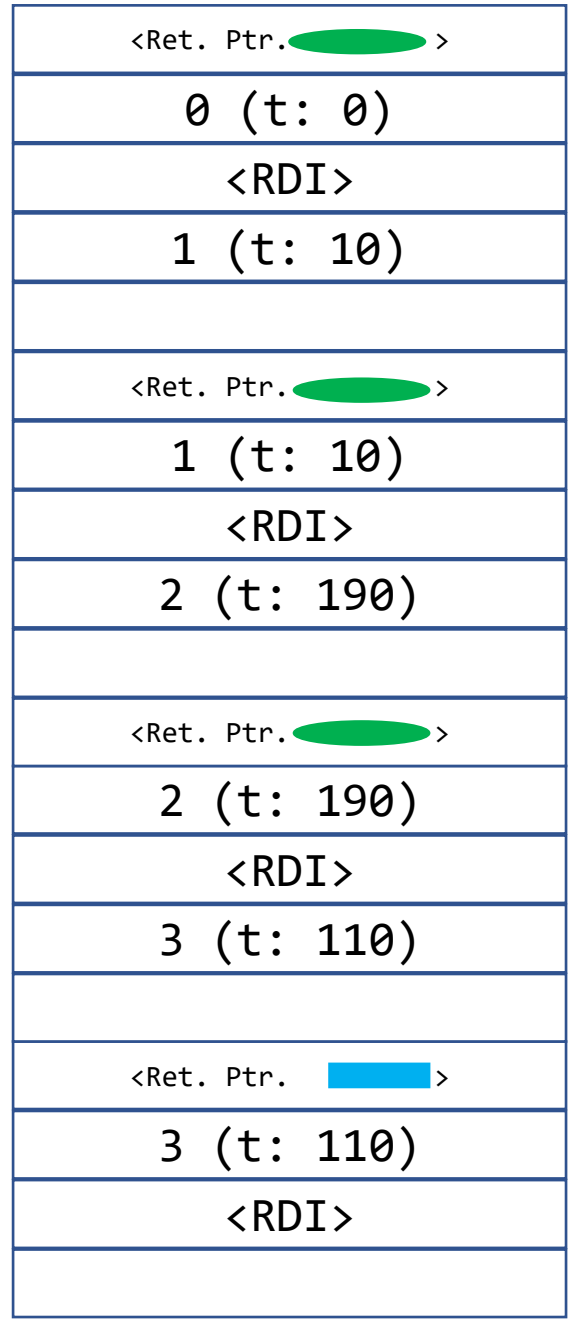
[rsp]

(Continued from top of right stack frame)

rax

0 (t: 0)

rbx



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

[rsp]

(Continued from top of right stack frame)

rax

(omit)

rbx

<Ret. Ptr. >
0 (t: 0)
<RDI>
1 (t: 10)

<Ret. Ptr. >
1 (t: 10)
<RDI>
2 (t: 190)

<Ret. Ptr. >
2 (t: 190)
<RDI>
3 (t: 110)

<Ret. Ptr. >
3 (t: 110)
<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

[rsp]

(Continued from top of right stack frame)

rax

(omit)

rbx

<Ret. Ptr. >

0 (t: 0)

<RDI>

1 (t: 10)

<Ret. Ptr. >

1 (t: 10)

<RDI>

2 (t: 190)

<Ret. Ptr. >

2 (t: 190)

<RDI>

3 (t: 110)

<Ret. Ptr. >

3 (t: 110)

<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

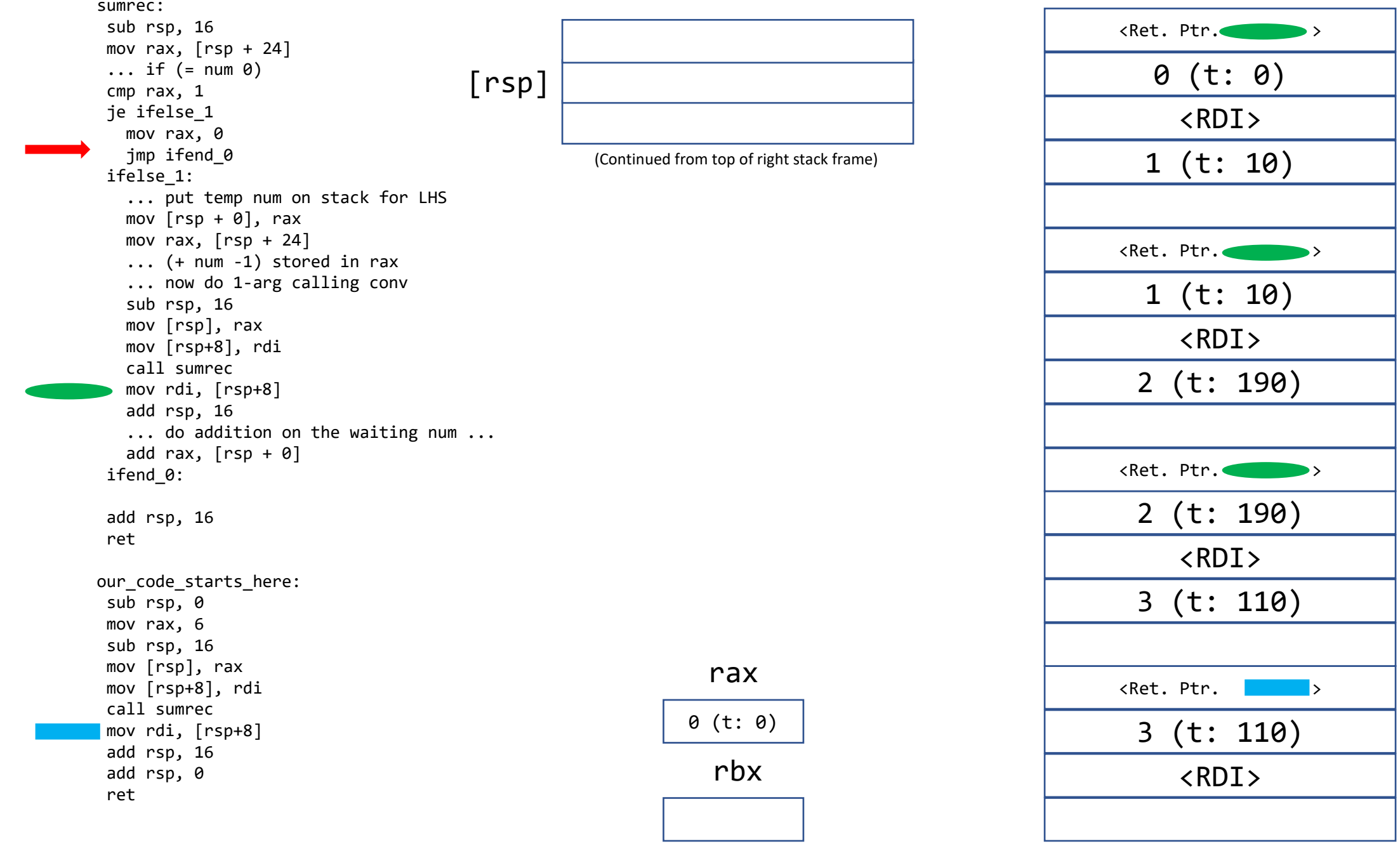
[rsp]

(Continued from top of right stack frame)

rax

0 (t: 0)

rbx



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

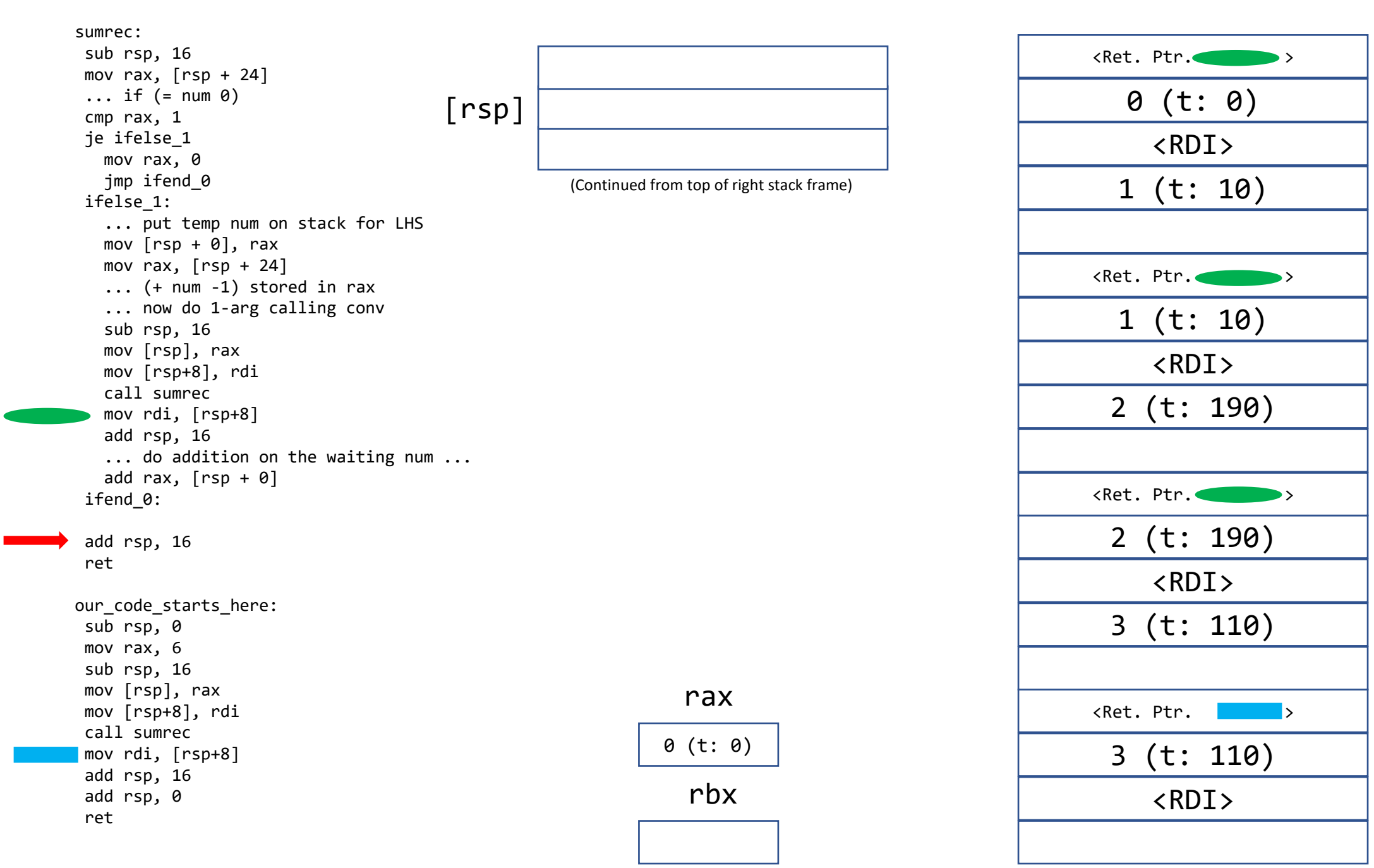
[rsp]

(Continued from top of right stack frame)

rax

0 (t: 0)

rbx



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

0 (t: 0)

rbx

[rsp]

<Ret. Ptr. >
0 (t: 0)
<RDI>
1 (t: 10)
<Ret. Ptr. >
1 (t: 10)
<RDI>
2 (t: 190)
<Ret. Ptr. >
2 (t: 190)
<RDI>
3 (t: 110)
<Ret. Ptr. >
3 (t: 110)
<RDI>

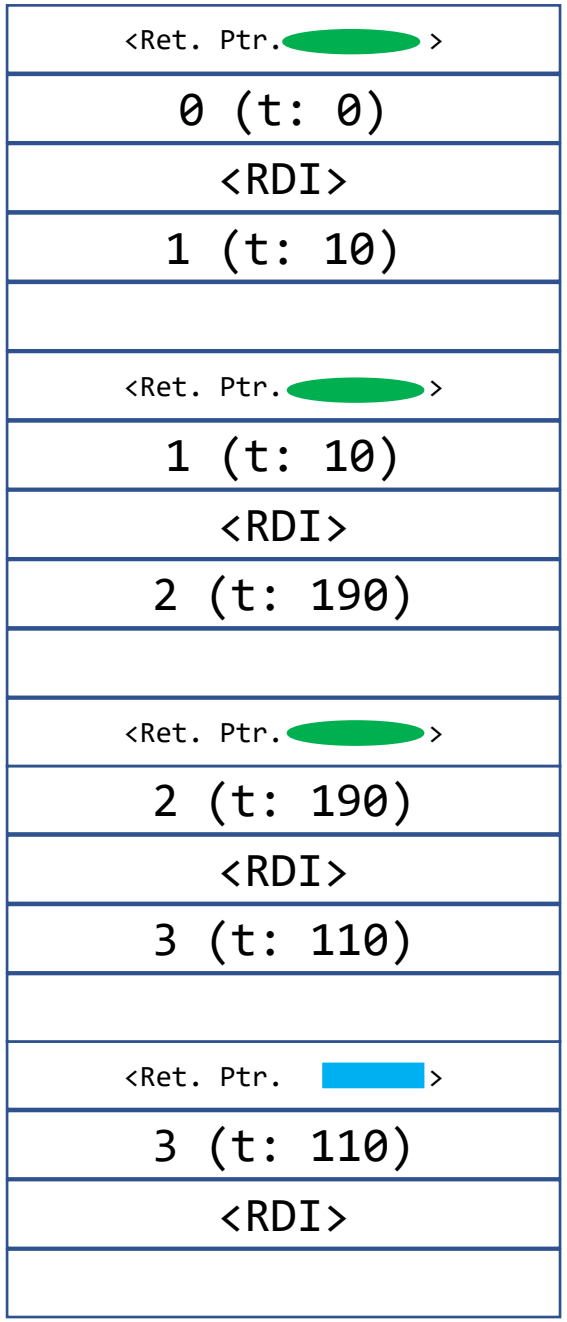
```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

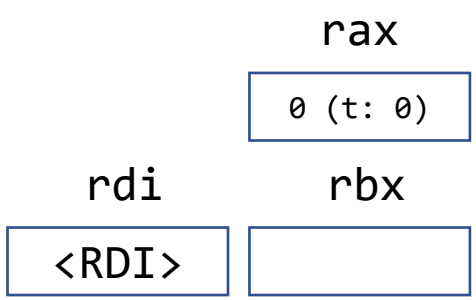
0 (t: 0)

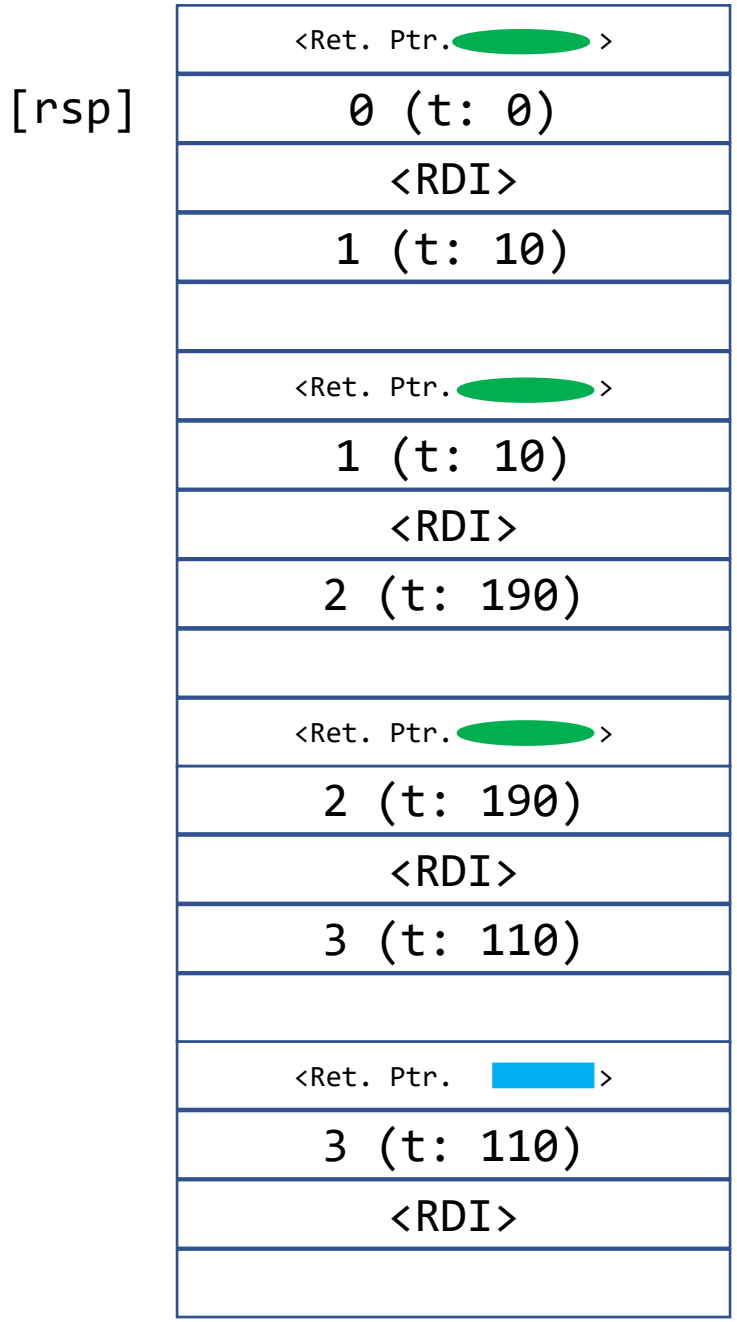
rbx

[rsp]



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```





```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

0 (t: 0)

rdi	rbx
<RDI>	

[rsp]

<Ret. Ptr.  >
0 (t: 0)
<RDI>
1 (t: 10)
<Ret. Ptr.  >
1 (t: 10)
<RDI>
2 (t: 190)
<Ret. Ptr.  >
2 (t: 190)
<RDI>
3 (t: 110)
<Ret. Ptr.  >
3 (t: 110)
<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

1 (t: 10)

rdi	rbx
<RDI>	

[rsp]

<Ret. Ptr.  >
0 (t: 0)
<RDI>
1 (t: 10)
<Ret. Ptr.  >
1 (t: 10)
<RDI>
2 (t: 190)
<Ret. Ptr.  >
2 (t: 190)
<RDI>
3 (t: 110)
<Ret. Ptr.  >
3 (t: 110)
<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

1 (t: 10)

rdi | rbx

<RDI>	

[rsp]

<Ret. Ptr. >
0 (t: 0)
<RDI>
1 (t: 10)
<Ret. Ptr. >
1 (t: 10)
<RDI>
2 (t: 190)
<Ret. Ptr. >
2 (t: 190)
<RDI>
3 (t: 110)
<Ret. Ptr. >
3 (t: 110)
<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

1 (t: 10)

rdi	rbx
<RDI>	

[rsp]

<Ret. Ptr.  >
0 (t: 0)
<RDI>
1 (t: 10)
<Ret. Ptr.  >
1 (t: 10)
<RDI>
2 (t: 190)
<Ret. Ptr.  >
2 (t: 190)
<RDI>
3 (t: 110)
<Ret. Ptr.  >
3 (t: 110)
<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

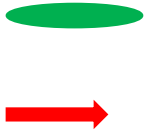
1 (t: 10)

rdi	rbx
<RDI>	

[rsp]

<Ret. Ptr. >
0 (t: 0)
<RDI>
1 (t: 10)
<Ret. Ptr. >
1 (t: 10)
<RDI>
2 (t: 190)
<Ret. Ptr. >
2 (t: 190)
<RDI>
3 (t: 110)
<Ret. Ptr. >
3 (t: 110)
<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```



rax

1 (t: 10)

rdi | rbx

<RDI> |

[rsp]

<Ret. Ptr. >
0 (t: 0)
<RDI>
1 (t: 10)

<Ret. Ptr. >
1 (t: 10)
<RDI>
2 (t: 190)

<Ret. Ptr. >
2 (t: 190)
<RDI>
3 (t: 110)

<Ret. Ptr. >
3 (t: 110)
<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

3 (t: 110)

rdi	rbx
<RDI>	

[rsp]

<Ret. Ptr.  >
0 (t: 0)
<RDI>
1 (t: 10)
<Ret. Ptr.  >
1 (t: 10)
<RDI>
2 (t: 190)
<Ret. Ptr.  >
2 (t: 190)
<RDI>
3 (t: 110)
<Ret. Ptr.  >
3 (t: 110)
<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

3 (t: 110)

rdi | rbx

<RDI> |

[rsp]

<Ret. Ptr. >
0 (t: 0)
<RDI>
1 (t: 10)

<Ret. Ptr. >
1 (t: 10)
<RDI>
2 (t: 190)

<Ret. Ptr. >
2 (t: 190)
<RDI>
3 (t: 110)

<Ret. Ptr. >
3 (t: 110)
<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

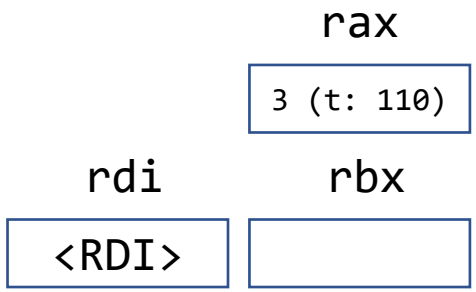
3 (t: 110)

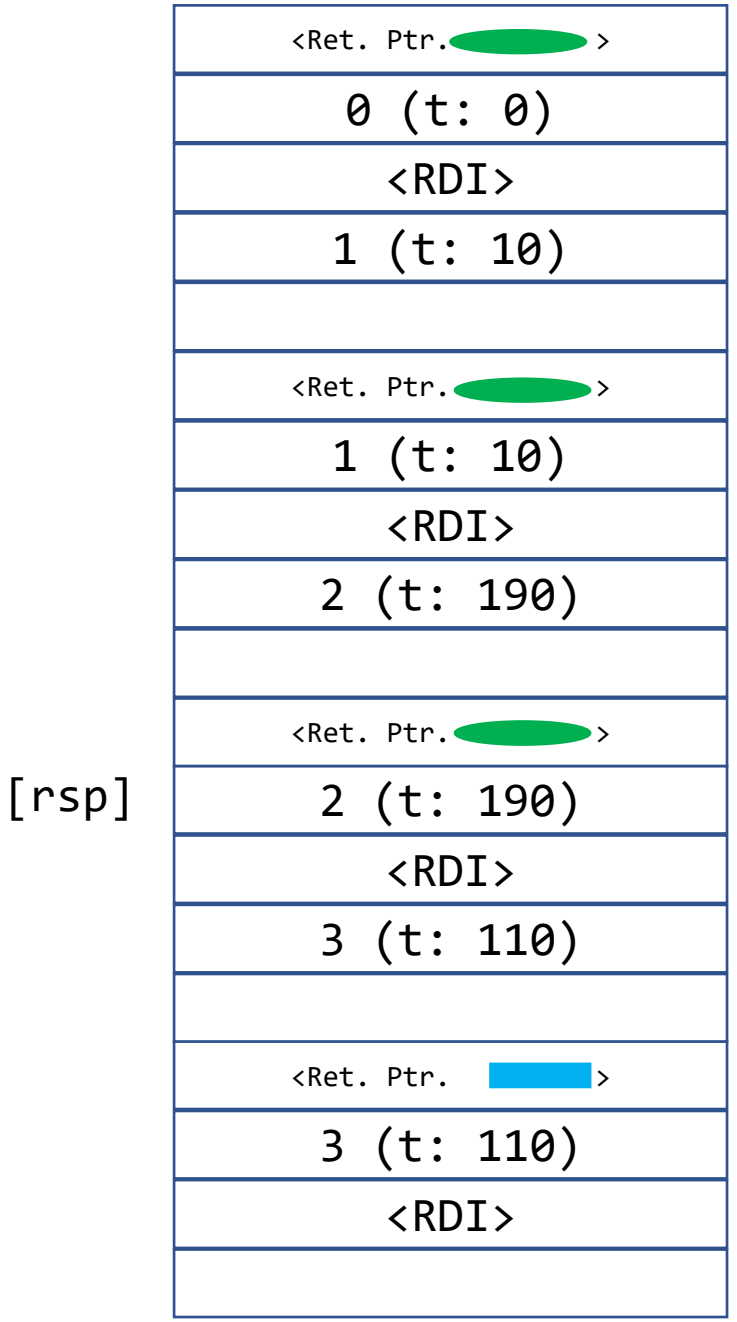
rdi	rbx
<RDI>	

[rsp]

<Ret. Ptr. >
0 (t: 0)
<RDI>
1 (t: 10)
<Ret. Ptr. >
1 (t: 10)
<RDI>
2 (t: 190)
<Ret. Ptr. >
2 (t: 190)
<RDI>
3 (t: 110)
<Ret. Ptr. >
3 (t: 110)
<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```





```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

3 (t: 110)

rdi	rbx
<RDI>	

[rsp]

<Ret. Ptr.  >
0 (t: 0)
<RDI>
1 (t: 10)
<Ret. Ptr.  >
1 (t: 10)
<RDI>
2 (t: 190)
<Ret. Ptr.  >
2 (t: 190)
<RDI>
3 (t: 110)
<Ret. Ptr.  >
3 (t: 110)
<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

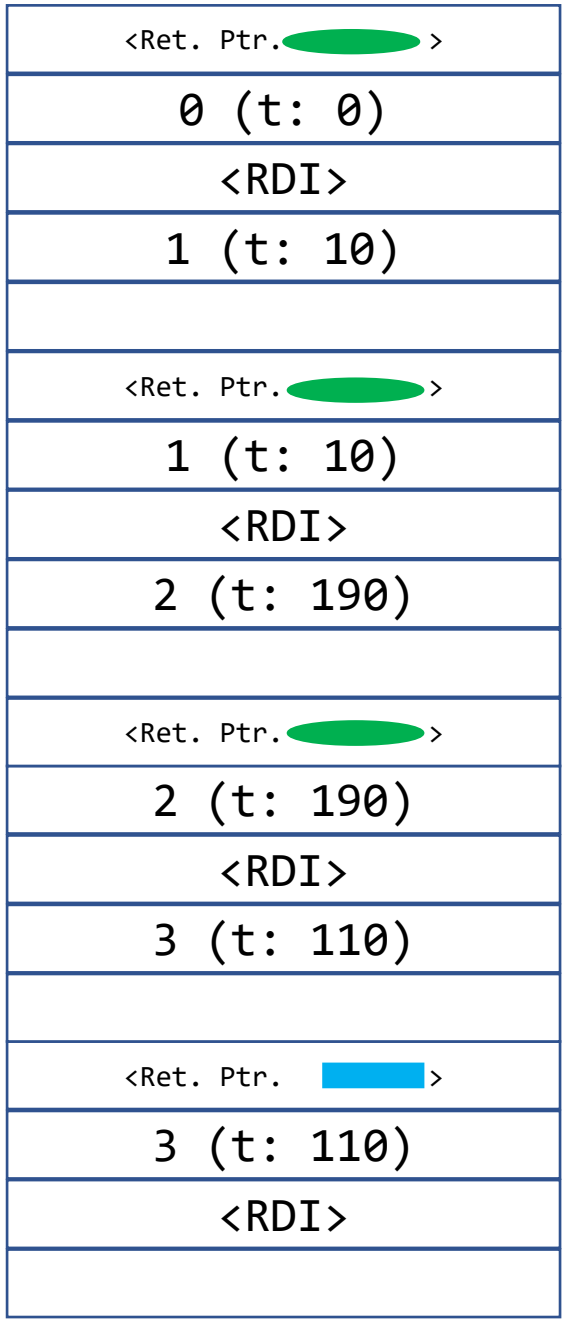
rax

6 (t: 1100)

rdi | rbx

<RDI> |

[rsp]



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

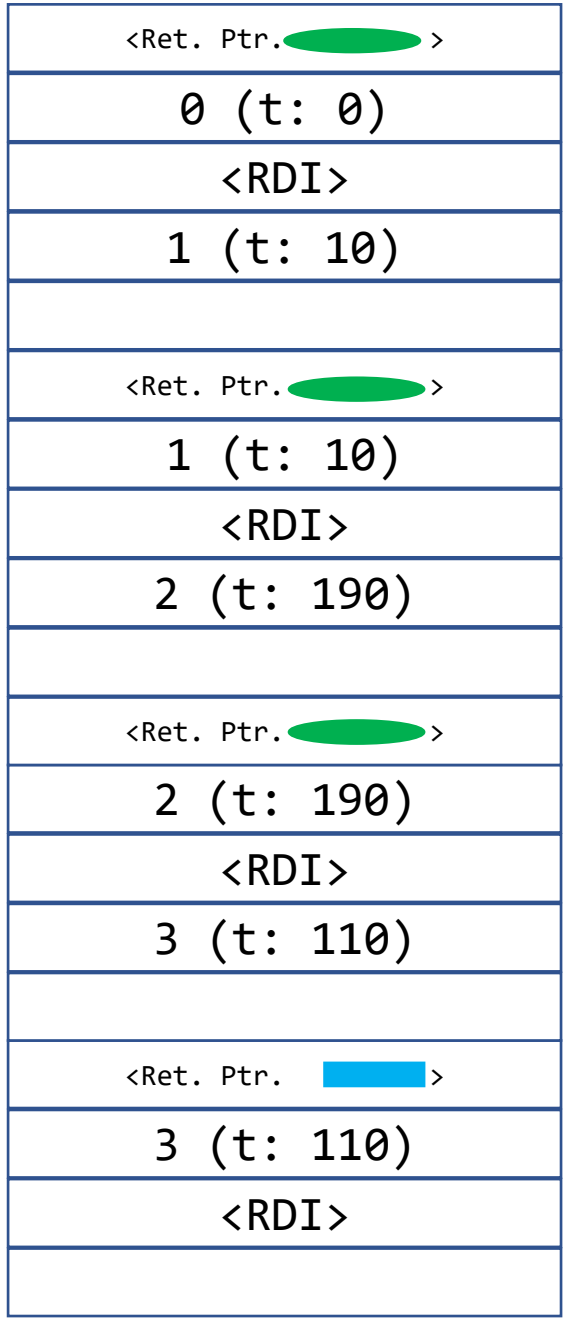
rax

6 (t: 1100)

rdi | rbx

<RDI> |

[rsp]



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

6 (t: 1100)

rdi | rbx

<RDI> |

[rsp]

<Ret. Ptr.  >

0 (t: 0)

<RDI>

1 (t: 10)

<Ret. Ptr.  >

1 (t: 10)

<RDI>

2 (t: 190)

<Ret. Ptr.  >

2 (t: 190)

<RDI>

3 (t: 110)

<Ret. Ptr.  >

3 (t: 110)

<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

6 (t: 1100)

rdi | rbx

<RDI> |

[rsp]

<Ret. Ptr.  >

0 (t: 0)

<RDI>

1 (t: 10)

<Ret. Ptr.  >

1 (t: 10)

<RDI>

2 (t: 190)

<Ret. Ptr.  >

2 (t: 190)

<RDI>

3 (t: 110)

<Ret. Ptr.  >

3 (t: 110)

<RDI>

```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

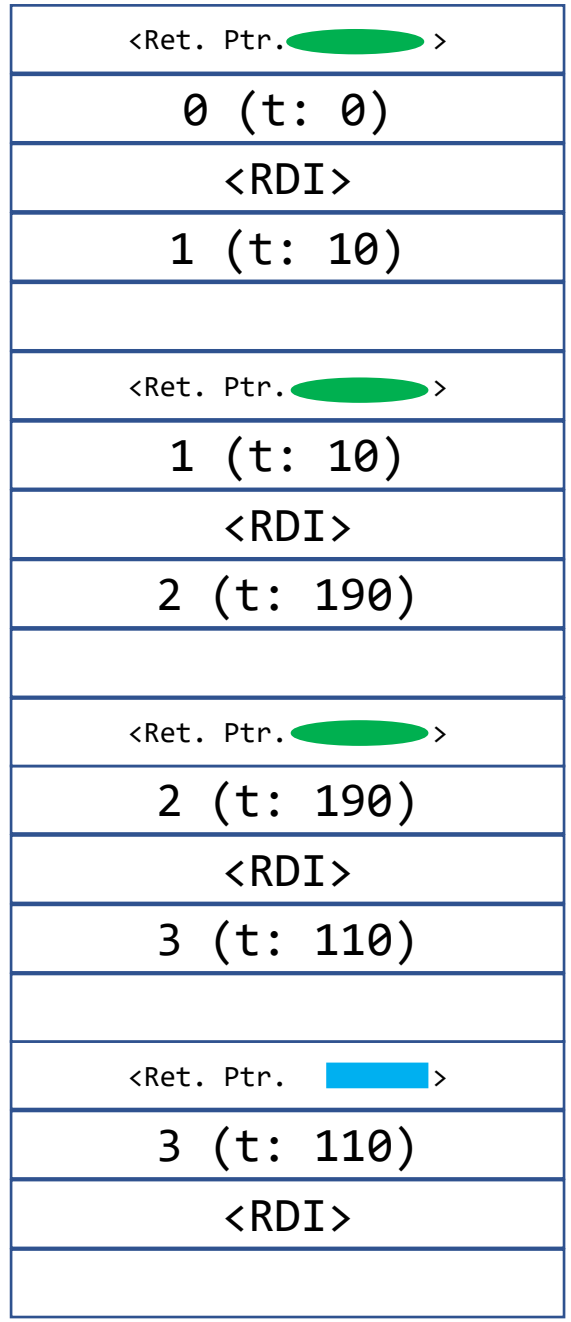
rax

6 (t: 1100)

rdi rbx

<RDI>

[rsp]



```
sumrec:
 sub rsp, 16
 mov rax, [rsp + 24]
 ... if (= num 0)
 cmp rax, 1
 je ifelse_1
   mov rax, 0
   jmp ifend_0
 ifelse_1:
   ... put temp num on stack for LHS
   mov [rsp + 0], rax
   mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
   sub rsp, 16
   mov [rsp], rax
   mov [rsp+8], rdi
   call sumrec
   mov rdi, [rsp+8]
   add rsp, 16
   ... do addition on the waiting num ...
   add rax, [rsp + 0]
 ifend_0:

 add rsp, 16
 ret

our_code_starts_here:
 sub rsp, 0
 mov rax, 6
 sub rsp, 16
 mov [rsp], rax
 mov [rsp+8], rdi
 call sumrec
 mov rdi, [rsp+8]
 add rsp, 16
 add rsp, 0
 ret
```

rax

6 (t: 1100)

rdi | rbx

<RDI> |