```
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
                                                              rax
mov [rsp+8], rdi
call sumrec
mov rdi, [rsp+8]
add rsp, 16
                                                              rbx
add rsp, 0
 ret
                                                                                     [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
                                                              rax
mov [rsp+8], rdi
call sumrec
mov rdi, [rsp+8]
add rsp, 16
                                                              rbx
add rsp, 0
ret
                                                                                     [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
                                                              rax
mov [rsp+8], rdi
call sumrec
mov rdi, [rsp+8]
add rsp, 16
                                                              rbx
add rsp, 0
ret
                                                                                     [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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                          3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                              rbx
add rsp, 0
ret
                                                                                     [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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                          3 (t: 110)
                                                                                     [rsp]
mov rdi, [rsp+8]
add rsp, 16
                                                              rbx
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
                                                             rax
mov [rsp+8], rdi
call sumrec
                                                          3 (t: 110)
                                                                                                        3 (t: 110)
                                                                                    [rsp]
mov rdi, [rsp+8]
add rsp, 16
                                                             rbx
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
                                                             rax
mov [rsp+8], rdi
call sumrec
                                                         3 (t: 110)
                                                                                                       3 (t: 110)
mov rdi, [rsp+8]
                                                                                    [rsp]
add rsp, 16
                                                             rbx
                                                                                                            <RDI>
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
                                                             rax
                                                                                    [rsp]
                                                                                                      <Ret. Ptr.
mov [rsp+8], rdi
call sumrec
                                                         3 (t: 110)
                                                                                                       3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                             rbx
                                                                                                            <RDI>
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:
                                                                                    [rsp]
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                             rax
mov [rsp+8], rdi
                                                                                                      <Ret. Ptr.
call sumrec
                                                          3 (t: 110)
                                                                                                       3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                             rbx
                                                                                                            <RDI>
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:
                                                                                   [rsp]
                                                                                                      3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                            rax
mov [rsp+8], rdi
                                                                                                     <Ret. Ptr.
call sumrec
                                                         3 (t: 110)
                                                                                                      3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                             rbx
                                                                                                           <RDI>
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:
                                                                                   [rsp]
                                                                                                      3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                            rax
mov [rsp+8], rdi
                                                                                                     <Ret. Ptr.
call sumrec
                                                           (omit)
                                                                                                      3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                            rbx
                                                                                                           <RDI>
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:
                                                                                   [rsp]
                                                                                                      3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                            rax
mov [rsp+8], rdi
                                                                                                     <Ret. Ptr.
call sumrec
                                                         3 (t: 110)
                                                                                                      3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                             rbx
                                                                                                          <RDI>
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:
                                                                                   [rsp]
                                                                                                      3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                            rax
mov [rsp+8], rdi
                                                                                                     <Ret. Ptr.
call sumrec
                                                         3 (t: 110)
                                                                                                      3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                             rbx
                                                                                                          <RDI>
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:
                                                                                   [rsp]
                                                                                                      3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                            rax
mov [rsp+8], rdi
                                                                                                     <Ret. Ptr.
call sumrec
                                                         3 (t: 110)
                                                                                                      3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                             rbx
                                                                                                           <RDI>
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:
                                                                                   [rsp]
                                                                                                      3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                            rax
mov [rsp+8], rdi
                                                                                                     <Ret. Ptr.
call sumrec
                                                         2 (t: 100)
                                                                                                      3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                             rbx
                                                                                                          <RDI>
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:
                                                                                   [rsp]
add rsp, 16
ret
our_code_starts_here:
                                                                                                      3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                            rax
mov [rsp+8], rdi
                                                                                                     <Ret. Ptr.
call sumrec
                                                         2 (t: 100)
                                                                                                      3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                             rbx
                                                                                                           <RDI>
add rsp, 0
ret
```

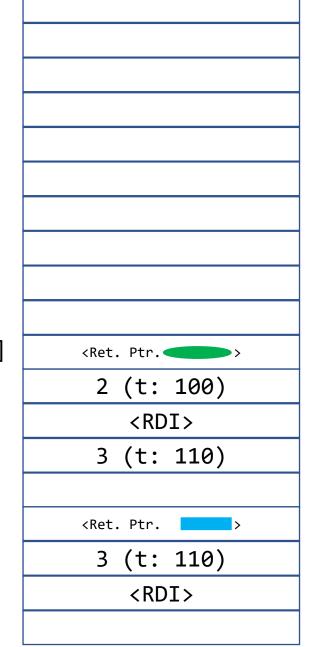
```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
... if (= num 0)
cmp rax, 1
je ifelse_1
  mov rax, 0
  jmp ifend_0
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
  sub rsp, 16
  mov [rsp], rax
  mov [rsp+8], rdi
  call sumrec
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
ifend_0:
                                                                                                     2 (t: 100)
                                                                                   [rsp]
add rsp, 16
ret
our_code_starts_here:
                                                                                                      3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                            rax
mov [rsp+8], rdi
                                                                                                    <Ret. Ptr.
call sumrec
                                                        2 (t: 100)
                                                                                                      3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                            rbx
add rsp, 0
                                                                                                          <RDI>
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:
                                                                                                     2 (t: 100)
                                                                                  [rsp]
add rsp, 16
ret
                                                                                                         <RDI>
our_code_starts_here:
                                                                                                     3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                           rax
mov [rsp+8], rdi
                                                                                                   <Ret. Ptr.
call sumrec
                                                        2 (t: 100)
                                                                                                     3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                            rbx
add rsp, 0
                                                                                                         <RDI>
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]
                                                                                      [rsp]
ifend_0:
add rsp, 16
 ret
our_code_starts_here:
sub rsp, 0
mov rax, 6
 sub rsp, 16
 mov [rsp], rax
                                                              rax
mov [rsp+8], rdi
 call sumrec
                                                           2 (t: 100)
 mov rdi, [rsp+8]
add rsp, 16
                                                               rbx
 add rsp, 0
 ret
```

<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]
                                                                                      [rsp]
ifend_0:
add rsp, 16
ret
our_code_starts_here:
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           2 (t: 100)
mov rdi, [rsp+8]
add rsp, 16
                                                               rbx
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
                                                                                      [rsp]
  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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           2 (t: 100)
mov rdi, [rsp+8]
add rsp, 16
                                                               rbx
add rsp, 0
ret
```

<Ret. Ptr. >>

2 (t: 100)

<RDI>

3 (t: 110)

3 (t: 110)

<RDI>

<Ret. Ptr.

```
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
                                                                                      [rsp]
  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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           2 (t: 100)
mov rdi, [rsp+8]
add rsp, 16
                                                               rbx
add rsp, 0
ret
```

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
                                                                                 [rsp]
                                                                                                   2 (t: 100)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                                  <Ret. Ptr. >>
ifend_0:
                                                                                                   2 (t: 100)
add rsp, 16
ret
                                                                                                       <RDI>
our_code_starts_here:
                                                                                                   3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                          rax
mov [rsp+8], rdi
                                                                                                  <Ret. Ptr.
call sumrec
                                                          (omit)
                                                                                                   3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                           rbx
add rsp, 0
                                                                                                       <RDI>
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
                                                                                 [rsp]
                                                                                                   2 (t: 100)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                                  <Ret. Ptr. >>
ifend_0:
                                                                                                   2 (t: 100)
add rsp, 16
ret
                                                                                                       <RDI>
our_code_starts_here:
                                                                                                   3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                          rax
mov [rsp+8], rdi
                                                                                                  <Ret. Ptr.
call sumrec
                                                       2 (t: 100)
                                                                                                   3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                           rbx
add rsp, 0
                                                                                                       <RDI>
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
                                                                                 [rsp]
                                                                                                   2 (t: 100)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                                  <Ret. Ptr. >>
ifend_0:
                                                                                                   2 (t: 100)
add rsp, 16
ret
                                                                                                       <RDI>
our_code_starts_here:
                                                                                                   3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                          rax
mov [rsp+8], rdi
                                                                                                  <Ret. Ptr.
call sumrec
                                                       2 (t: 100)
                                                                                                   3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                           rbx
add rsp, 0
                                                                                                       <RDI>
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
                                                                                 [rsp]
                                                                                                   2 (t: 100)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                                  <Ret. Ptr. >>
ifend_0:
                                                                                                   2 (t: 100)
add rsp, 16
ret
                                                                                                       <RDI>
our_code_starts_here:
                                                                                                   3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                          rax
mov [rsp+8], rdi
                                                                                                  <Ret. Ptr.
call sumrec
                                                       2 (t: 100)
                                                                                                   3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                           rbx
add rsp, 0
                                                                                                       <RDI>
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
                                                                                 [rsp]
                                                                                                   2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                                  <Ret. Ptr. >>
ifend_0:
                                                                                                   2 (t: 190)
add rsp, 16
ret
                                                                                                       <RDI>
our_code_starts_here:
                                                                                                   3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                          rax
mov [rsp+8], rdi
                                                                                                  <Ret. Ptr.
call sumrec
                                                        1 (t: 10)
                                                                                                   3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                           rbx
add rsp, 0
                                                                                                       <RDI>
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
                                                                                 [rsp]
  sub rsp, 16
  mov [rsp], rax
  mov [rsp+8], rdi
  call sumrec
                                                                                                   2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                                  <Ret. Ptr. >>
ifend_0:
                                                                                                   2 (t: 190)
add rsp, 16
ret
                                                                                                       <RDI>
our_code_starts_here:
                                                                                                   3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                          rax
mov [rsp+8], rdi
                                                                                                  <Ret. Ptr.
call sumrec
                                                        1 (t: 10)
                                                                                                   3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                           rbx
add rsp, 0
                                                                                                       <RDI>
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
                                                                                [rsp]
                                                                                                   1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
  mov [rsp+8], rdi
  call sumrec
                                                                                                   2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                                 <Ret. Ptr. >>
ifend_0:
                                                                                                   2 (t: 190)
add rsp, 16
ret
                                                                                                       <RDI>
our_code_starts_here:
                                                                                                   3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                          rax
mov [rsp+8], rdi
                                                                                                 <Ret. Ptr.
call sumrec
                                                       1 (t: 10)
                                                                                                   3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                           rbx
add rsp, 0
                                                                                                       <RDI>
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
                                                                                [rsp]
                                                                                                   1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                      <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                                  2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                                 <Ret. Ptr. >>
ifend_0:
                                                                                                  2 (t: 190)
add rsp, 16
ret
                                                                                                      <RDI>
our_code_starts_here:
                                                                                                  3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                          rax
mov [rsp+8], rdi
                                                                                                 <Ret. Ptr.
call sumrec
                                                       1 (t: 10)
                                                                                                  3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                          rbx
add rsp, 0
                                                                                                      <RDI>
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]
                                                                                    [rsp]
                                                                                                     <Ret. Ptr.
   ... (+ 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]
                                                                                                     <Ret. Ptr. >>
ifend 0:
add rsp, 16
 ret
our_code_starts_here:
sub rsp, 0
mov rax, 6
 sub rsp, 16
 mov [rsp], rax
                                                            rax
mov [rsp+8], rdi
                                                                                                     <Ret. Ptr.
call sumrec
                                                          1 (t: 10)
 mov rdi, [rsp+8]
 add rsp, 16
                                                             rbx
 add rsp, 0
 ret
```

1 (t: 10)

<RDI>

2 (t: 190)

2 (t: 190)

<RDI>

3 (t: 110)

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]
                                                                                      [rsp]
   ... (+ 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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           1 (t: 10)
mov rdi, [rsp+8]
add rsp, 16
                                                               rbx
add rsp, 0
ret
```

<Ret. Ptr.

1 (t: 10)

<RDI>

2 (t: 190)

<Ret. Ptr. >>

2 (t: 190)

<RDI>

3 (t: 110)

3 (t: 110)

<RDI>

<Ret. Ptr.

```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
 ... if (= num 0)
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend_0
                                                                               [rsp]
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                               <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                                  1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                     <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                                 2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                               <Ret. Ptr. >>
ifend 0:
                                                                                                 2 (t: 190)
add rsp, 16
ret
                                                                                                     <RDI>
our_code_starts_here:
                                                                                                 3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                         rax
mov [rsp+8], rdi
                                                                                               <Ret. Ptr.
call sumrec
                                                      1 (t: 10)
                                                                                                 3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                          rbx
add rsp, 0
                                                                                                     <RDI>
ret
```

```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
... if (= num 0)
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend_0
                                                                               [rsp]
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                               <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                                  1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                     <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                                 2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                               <Ret. Ptr. >>
ifend 0:
                                                                                                 2 (t: 190)
add rsp, 16
ret
                                                                                                     <RDI>
our_code_starts_here:
                                                                                                 3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                         rax
mov [rsp+8], rdi
                                                                                               <Ret. Ptr.
call sumrec
                                                      1 (t: 10)
                                                                                                 3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                          rbx
add rsp, 0
                                                                                                     <RDI>
ret
```

```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
 ... if (= num 0)
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend_0
                                                                               [rsp]
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                                <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                                  1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                     <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                                 2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                                <Ret. Ptr. >>
ifend 0:
                                                                                                 2 (t: 190)
add rsp, 16
ret
                                                                                                     <RDI>
our_code_starts_here:
                                                                                                 3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                         rax
mov [rsp+8], rdi
                                                                                                <Ret. Ptr.
call sumrec
                                                        (omit)
                                                                                                 3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                          rbx
add rsp, 0
                                                                                                     <RDI>
ret
```

```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
... if (= num 0)
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend_0
                                                                               [rsp]
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                               <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                                  1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                     <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                                 2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                               <Ret. Ptr. >>
ifend 0:
                                                                                                 2 (t: 190)
add rsp, 16
ret
                                                                                                     <RDI>
our_code_starts_here:
                                                                                                 3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                         rax
mov [rsp+8], rdi
                                                                                               <Ret. Ptr.
call sumrec
                                                      1 (t: 10)
                                                                                                 3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                          rbx
add rsp, 0
                                                                                                     <RDI>
ret
```

```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
... if (= num 0)
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend_0
                                                                              [rsp]
                                                                                                 1 (t: 10)
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                               <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                                 1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                    <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                                2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                               <Ret. Ptr. >>
ifend_0:
                                                                                                2 (t: 190)
add rsp, 16
ret
                                                                                                    <RDI>
our_code_starts_here:
                                                                                                3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                         rax
mov [rsp+8], rdi
                                                                                               <Ret. Ptr.
call sumrec
                                                      1 (t: 10)
                                                                                                3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                         rbx
add rsp, 0
                                                                                                    <RDI>
ret
```

```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
... if (= num 0)
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend_0
                                                                              [rsp]
                                                                                                 1 (t: 10)
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                               <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                                 1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                    <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                                2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                               <Ret. Ptr. >>
ifend_0:
                                                                                                2 (t: 190)
add rsp, 16
ret
                                                                                                    <RDI>
our_code_starts_here:
                                                                                                3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                        rax
mov [rsp+8], rdi
                                                                                               <Ret. Ptr.
call sumrec
                                                      1 (t: 10)
                                                                                                3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                         rbx
add rsp, 0
                                                                                                    <RDI>
ret
```

```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
... if (= num 0)
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend_0
                                                                              [rsp]
                                                                                                 1 (t: 10)
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                               <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                                 1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                    <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                                2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                               <Ret. Ptr. >>
ifend_0:
                                                                                                2 (t: 190)
add rsp, 16
ret
                                                                                                    <RDI>
our_code_starts_here:
                                                                                                3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                        rax
mov [rsp+8], rdi
                                                                                               <Ret. Ptr.
call sumrec
                                                      0 (t: 0)
                                                                                                3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                         rbx
add rsp, 0
                                                                                                    <RDI>
ret
```

```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
... if (= num 0)
                                                                              [rsp]
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend_0
                                                                                                 1 (t: 10)
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                               <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                                 1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                    <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                                2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                               <Ret. Ptr. >>
ifend_0:
                                                                                                2 (t: 190)
add rsp, 16
ret
                                                                                                    <RDI>
our_code_starts_here:
                                                                                                3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                        rax
mov [rsp+8], rdi
                                                                                               <Ret. Ptr.
call sumrec
                                                      0 (t: 0)
                                                                                                3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                         rbx
add rsp, 0
                                                                                                    <RDI>
ret
```

```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
... if (= num 0)
                                                                              [rsp]
                                                                                                 0 (t: 0)
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend_0
                                                                                                1 (t: 10)
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                              <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                                1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                    <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                                2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                              <Ret. Ptr. >>
ifend_0:
                                                                                                2 (t: 190)
add rsp, 16
ret
                                                                                                    <RDI>
our_code_starts_here:
                                                                                                3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                        rax
mov [rsp+8], rdi
                                                                                              <Ret. Ptr.
call sumrec
                                                      0 (t: 0)
                                                                                                3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                         rbx
add rsp, 0
                                                                                                    <RDI>
ret
```

```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
                                                                                                 0 (t: 0)
... if (= num 0)
                                                                              [rsp]
cmp rax, 1
je ifelse 1
                                                                                                   <RDI>
  mov rax, 0
  jmp ifend_0
                                                                                                1 (t: 10)
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                              <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                                1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                   <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                               2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                              <Ret. Ptr. >>
ifend_0:
                                                                                               2 (t: 190)
add rsp, 16
ret
                                                                                                   <RDI>
our_code_starts_here:
                                                                                               3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                        rax
mov [rsp+8], rdi
                                                                                              <Ret. Ptr.
call sumrec
                                                      0 (t: 0)
                                                                                               3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                         rbx
add rsp, 0
                                                                                                   <RDI>
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
                                                               rax
mov [rsp+8], rdi
 call sumrec
                                                            0 (t: 0)
 mov rdi, [rsp+8]
 add rsp, 16
                                                               rbx
 add rsp, 0
 ret
```

```
[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
                                                               rax
mov [rsp+8], rdi
call sumrec
                                                            0 (t: 0)
mov rdi, [rsp+8]
add rsp, 16
                                                               rbx
add rsp, 0
ret
```

```
<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)
                                      [rsp]
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend 0
                                                   (Continued from top of right stack frame)
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
                                                               rax
mov [rsp+8], rdi
call sumrec
                                                             0 (t: 0)
mov rdi, [rsp+8]
add rsp, 16
                                                                rbx
add rsp, 0
 ret
```

```
<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)
                                      [rsp]
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend 0
                                                   (Continued from top of right stack frame)
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
                                                               rax
mov [rsp+8], rdi
call sumrec
                                                             0 (t: 0)
mov rdi, [rsp+8]
add rsp, 16
                                                                rbx
add rsp, 0
 ret
```

```
<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)
                                      [rsp]
 cmp rax, 1
 je ifelse 1
  mov rax, 0
  jmp ifend 0
                                                   (Continued from top of right stack frame)
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
                                                               rax
mov [rsp+8], rdi
call sumrec
                                                             0 (t: 0)
mov rdi, [rsp+8]
add rsp, 16
                                                                rbx
add rsp, 0
 ret
```

```
<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)
                                      [rsp]
cmp rax, 1
 je ifelse 1
  mov rax, 0
  jmp ifend 0
                                                    (Continued from top of right stack frame)
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
                                                                rax
mov [rsp+8], rdi
call sumrec
                                                               (omit)
mov rdi, [rsp+8]
add rsp, 16
                                                                rbx
add rsp, 0
 ret
```

```
<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)
                                      [rsp]
cmp rax, 1
 je ifelse 1
  mov rax, 0
   jmp ifend 0
                                                    (Continued from top of right stack frame)
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
                                                                rax
mov [rsp+8], rdi
call sumrec
                                                               (omit)
mov rdi, [rsp+8]
add rsp, 16
                                                                rbx
add rsp, 0
 ret
```

```
<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)
                                      [rsp]
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend 0
                                                   (Continued from top of right stack frame)
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
                                                               rax
mov [rsp+8], rdi
call sumrec
                                                             0 (t: 0)
mov rdi, [rsp+8]
add rsp, 16
                                                                rbx
add rsp, 0
 ret
```

```
<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)
                                      [rsp]
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend 0
                                                    (Continued from top of right stack frame)
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
                                                               rax
mov [rsp+8], rdi
call sumrec
                                                             0 (t: 0)
mov rdi, [rsp+8]
add rsp, 16
                                                                rbx
add rsp, 0
 ret
```

```
sumrec:
                                                                             [rsp]
sub rsp, 16
                                                                                             <Ret. Ptr. >
mov rax, [rsp + 24]
                                                                                                0 (t: 0)
... if (= num 0)
cmp rax, 1
je ifelse 1
                                                                                                   <RDI>
  mov rax, 0
  jmp ifend_0
                                                                                                1 (t: 10)
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                             <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                                1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                   <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                               2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                             <Ret. Ptr. >>
ifend_0:
                                                                                               2 (t: 190)
add rsp, 16
ret
                                                                                                   <RDI>
our_code_starts_here:
                                                                                               3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                        rax
mov [rsp+8], rdi
                                                                                             <Ret. Ptr.
call sumrec
                                                      0 (t: 0)
                                                                                               3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                                        rbx
add rsp, 0
                                                                                                   <RDI>
ret
```

```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
 ... if (= num 0)
                                                                                      [rsp]
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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                            0 (t: 0)
mov rdi, [rsp+8]
add rsp, 16
                                                               rbx
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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                            0 (t: 0)
mov rdi, [rsp+8]
add rsp, 16
                                               rdi
                                                               rbx
add rsp, 0
ret
                                             <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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                            0 (t: 0)
mov rdi, [rsp+8]
add rsp, 16
                                               rdi
                                                               rbx
add rsp, 0
ret
                                             <RDI>
```

```
<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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           1 (t: 10)
mov rdi, [rsp+8]
add rsp, 16
                                               rdi
                                                               rbx
add rsp, 0
ret
                                             <RDI>
```

```
<Ret. Ptr. >
             0 (t: 0)
               <RDI>
[rsp]
             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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           1 (t: 10)
mov rdi, [rsp+8]
add rsp, 16
                                              rdi
                                                              rbx
add rsp, 0
ret
                                            <RDI>
```

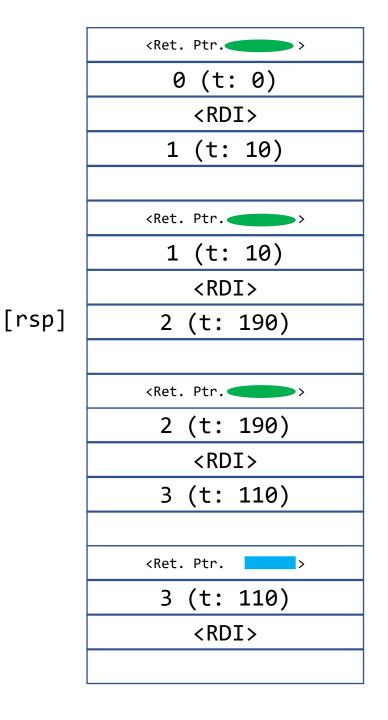
```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
 ... if (= num 0)
cmp rax, 1
je ifelse_1
  mov rax, 0
  jmp ifend_0
ifelse_1:
   ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
  sub rsp, 16
  mov [rsp], rax
  mov [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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           1 (t: 10)
mov rdi, [rsp+8]
add rsp, 16
                                              rdi
                                                              rbx
add rsp, 0
ret
                                            <RDI>
```

<ret. ptr.=""></ret.>
0 (t: 0)
<rdi></rdi>
1 (t: 10)
<ret. ptr.="">&gt;</ret.>
1 (t: 10)
<rdi></rdi>
2 (t: 190)
<ret. ptr.<="" td=""></ret.>
2 (t: 190)
<rdi></rdi>
3 (t: 110)
<ret. ptr.=""></ret.>
3 (t: 110)
<rdi></rdi>

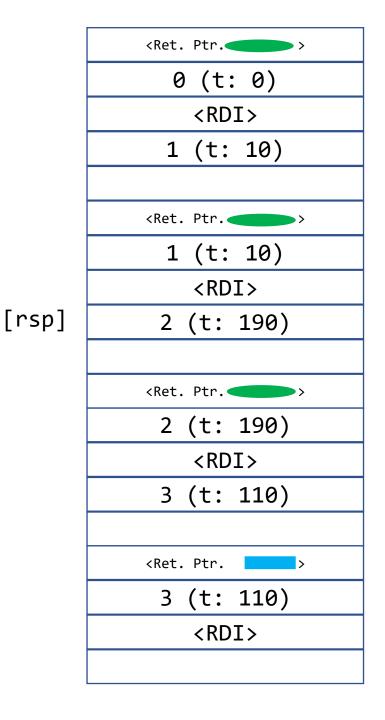
```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
 ... if (= num 0)
cmp rax, 1
je ifelse_1
  mov rax, 0
  jmp ifend_0
ifelse_1:
   ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
  sub rsp, 16
  mov [rsp], rax
  mov [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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           1 (t: 10)
mov rdi, [rsp+8]
add rsp, 16
                                              rdi
                                                              rbx
add rsp, 0
ret
                                            <RDI>
```

<ret. ptr.<="" th=""></ret.>
0 (t: 0)
<rdi></rdi>
1 (t: 10)
<ret. ptr.="">&gt;</ret.>
1 (t: 10)
<rdi></rdi>
2 (t: 190)
<ret. ptr.="" td="" ◆<=""></ret.>
2 (t: 190)
<rdi></rdi>
3 (t: 110)
<ret. ptr.=""></ret.>
3 (t: 110)
<rdi></rdi>

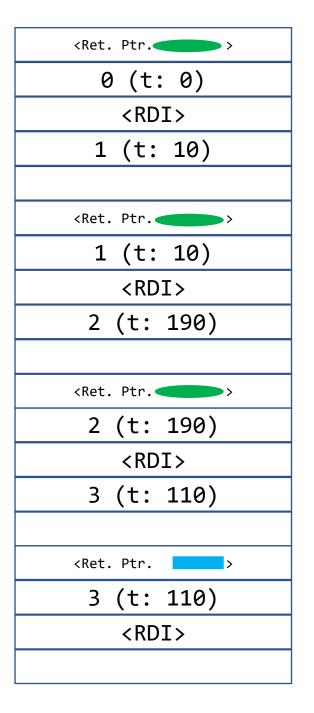
```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
 ... if (= num 0)
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend 0
ifelse_1:
   ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
  sub rsp, 16
  mov [rsp], rax
  mov [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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           1 (t: 10)
mov rdi, [rsp+8]
add rsp, 16
                                               rdi
                                                               rbx
add rsp, 0
ret
                                             <RDI>
```



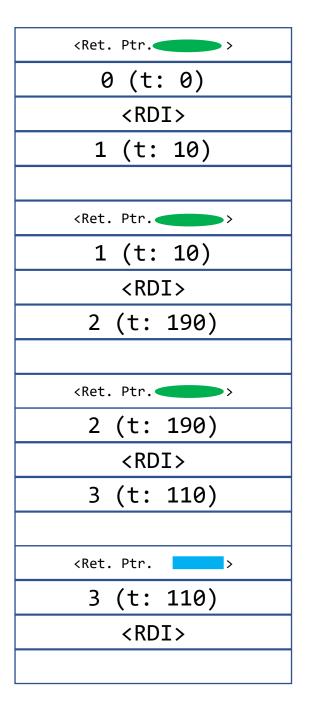
```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
 ... if (= num 0)
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend 0
ifelse_1:
   ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
  sub rsp, 16
  mov [rsp], rax
  mov [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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                               rdi
                                                               rbx
add rsp, 0
ret
                                             <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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                               rdi
                                                               rbx
add rsp, 0
ret
                                             <RDI>
```



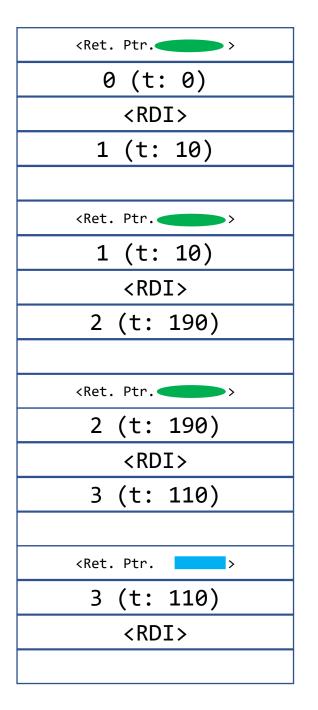
```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
 ... if (= num 0)
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend 0
ifelse_1:
   ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
  sub rsp, 16
  mov [rsp], rax
  mov [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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                          3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                               rdi
                                                              rbx
add rsp, 0
ret
                                             <RDI>
```



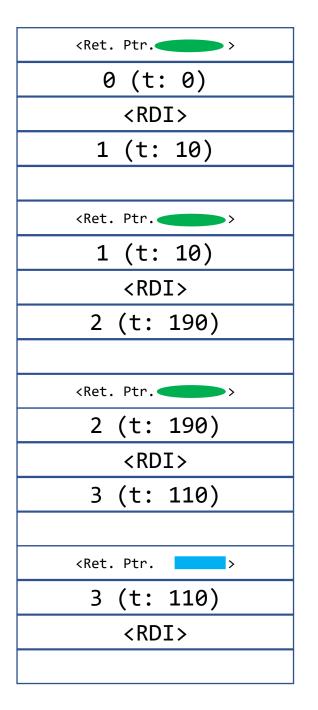
```
sumrec:
sub rsp, 16
mov rax, [rsp + 24]
 ... if (= num 0)
cmp rax, 1
je ifelse 1
  mov rax, 0
  jmp ifend 0
ifelse_1:
   ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
   ... (+ num -1) stored in rax
   ... now do 1-arg calling conv
  sub rsp, 16
  mov [rsp], rax
  mov [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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                               rdi
                                                               rbx
add rsp, 0
ret
                                             <RDI>
```

```
<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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                               rdi
                                                               rbx
add rsp, 0
ret
                                             <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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           6 (t: 1100)
mov rdi, [rsp+8]
add rsp, 16
                                               rdi
                                                               rbx
add rsp, 0
ret
                                             <RDI>
```

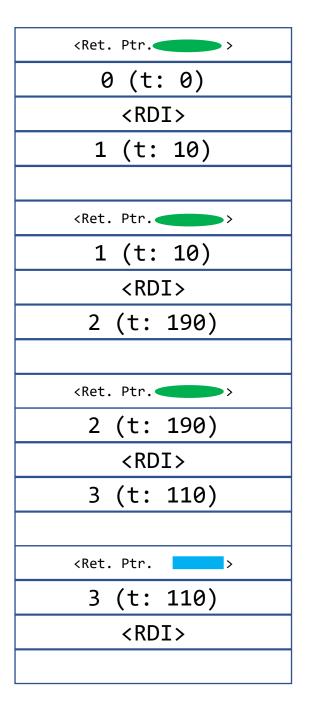


```
sumrec:
sub rsp, 16
                                                                                             <Ret. Ptr. >
mov rax, [rsp + 24]
                                                                                                0 (t: 0)
... if (= num 0)
cmp rax, 1
je ifelse 1
                                                                                                  <RDI>
  mov rax, 0
  jmp ifend 0
                                                                                               1 (t: 10)
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                             <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                               1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                  <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                               2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                             <Ret. Ptr. >>
ifend_0:
                                                                                              2 (t: 190)
add rsp, 16
ret
                                                                                                  <RDI>
our_code_starts_here:
                                                                                               3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                       rax
                                                                            [rsp]
mov [rsp+8], rdi
                                                                                             <Ret. Ptr.
call sumrec
                                                     6 (t: 1100)
                                                                                               3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                          rdi
                                                        rbx
add rsp, 0
                                                                                                  <RDI>
ret
                                        <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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                          6 (t: 1100)
                                                                                     [rsp]
mov rdi, [rsp+8]
add rsp, 16
                                               rdi
                                                              rbx
add rsp, 0
ret
                                            <RDI>
```

```
sumrec:
sub rsp, 16
                                                                                             <Ret. Ptr. >
mov rax, [rsp + 24]
                                                                                                0 (t: 0)
... if (= num 0)
cmp rax, 1
je ifelse 1
                                                                                                  <RDI>
  mov rax, 0
  jmp ifend 0
                                                                                               1 (t: 10)
ifelse_1:
  ... put temp num on stack for LHS
  mov [rsp + 0], rax
  mov rax, [rsp + 24]
                                                                                             <Ret. Ptr.
  ... (+ num -1) stored in rax
  ... now do 1-arg calling conv
                                                                                               1 (t: 10)
  sub rsp, 16
  mov [rsp], rax
                                                                                                  <RDI>
  mov [rsp+8], rdi
  call sumrec
                                                                                              2 (t: 190)
  mov rdi, [rsp+8]
  add rsp, 16
  ... do addition on the waiting num ...
  add rax, [rsp + 0]
                                                                                             <Ret. Ptr. >>
ifend_0:
                                                                                              2 (t: 190)
add rsp, 16
ret
                                                                                                  <RDI>
our_code_starts_here:
                                                                                              3 (t: 110)
sub rsp, 0
mov rax, 6
sub rsp, 16
mov [rsp], rax
                                                       rax
mov [rsp+8], rdi
                                                                                             <Ret. Ptr.
call sumrec
                                                    6 (t: 1100)
                                                                            [rsp]
                                                                                              3 (t: 110)
mov rdi, [rsp+8]
add rsp, 16
                                          rdi
                                                        rbx
add rsp, 0
                                                                                                  <RDI>
ret
                                        <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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           6 (t: 1100)
mov rdi, [rsp+8]
 add rsp, 16
                                               rdi
                                                               rbx
add rsp, 0
 ret
                                             <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
                                                              rax
mov [rsp+8], rdi
call sumrec
                                                           6 (t: 1100)
mov rdi, [rsp+8]
add rsp, 16
                                               rdi
                                                               rbx
 add rsp, 0
ret
                                             <RDI>
```

