

(rdi, rsi, rdx, rcx, r8, r9)
(rax, rdx)

```
.intel_syntax noprefix
.text
.global foo
.type foo, @function
foo:
```

```
    push rbp
    mov rbp, rsp
    mov DWORD PTR -36[rbp], edi
    mov DWORD PTR -32[rbp], 0
    mov eax, DWORD PTR -36[rbp]
    mov QWORD PTR -24[rbp], rax
    cmp DWORD PTR -36[rbp], 1
    jg label_1
    mov eax, DWORD PTR -36[rbp]
    mov QWORD PTR -24[rbp], rax
    jmp label_2
```

label_1:

```
    mov eax, DWORD PTR -36[rbp]
    sub eax, 1
    mov edi, eax
    call foo
    mov rcx, rdx
    mov rdx, rcx
    mov rbx, rdx
    mov eax, DWORD PTR -36[rbp]
    sub eax, 2
    mov edi, eax
    call foo
    mov rcx, rdx
    mov rdx, rcx
    mov rax, rdx
    add rax, rbx
    mov QWORD PTR -24[rbp], rax
```

label_2:

```
    mov rax, QWORD PTR -32[rbp]
    mov rdx, QWORD PTR -24[rbp]
    mov rcx, rax
    mov rbx, rdx
    mov eax, ecx
    pop rbp
    ret
```

callee
saved

rsp mod 16 = 0.

rcx := rcx := rdx

return value

add rsp, 40

pop rbp

struct A foo(unsigned int n) {

struct A r;

r.a1 = 0;

r.a2 = n;

if (n > 1) {

struct A r1 = foo(n-1)

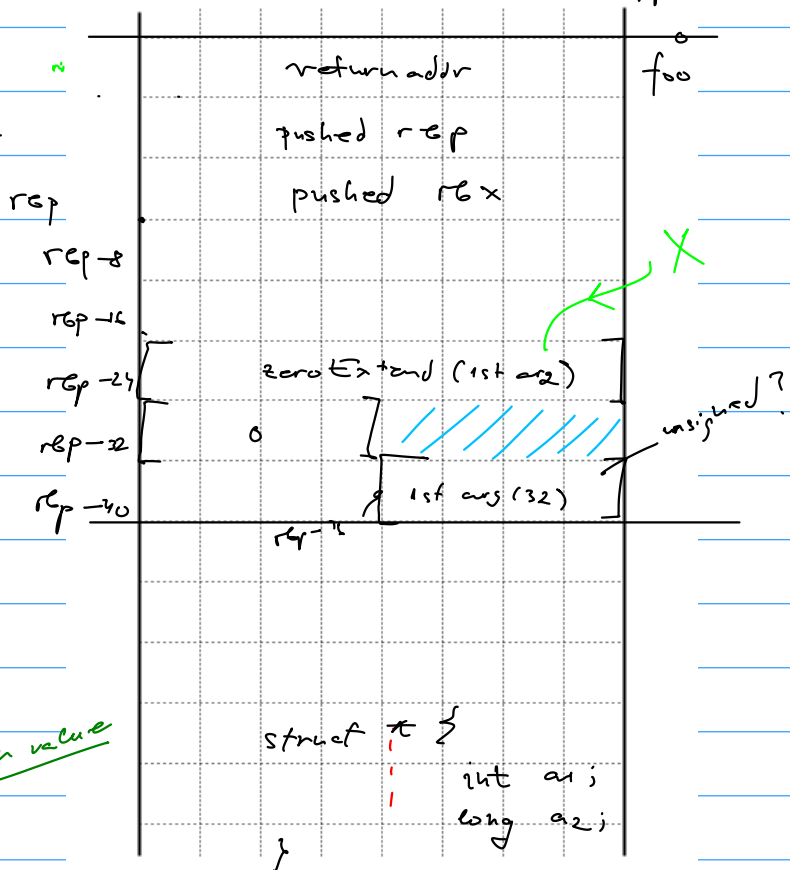
struct A r2 = foo(n-2)

r.a2 = r1.a2 + r2.a2

}

return r;

}



struct A foo(unsigned int n) {

if (n > 1) {

struct A r1 = foo(n-1);

struct A r2 = foo(n-2);

long x = r1.a2 + r2.a2;

struct A r;

r.a1 = 0

r.a2 = x;

return r;

} else {

struct A r;

r.a1 = 0;

r.a2 = n;

return r;

}

leave
mov rsp, rbp
pop rbp

```

.intel_syntax noprefix
.text
.globl foo
.type foo, @function
foo:

```

```

    push rbp
    mov rbp, rsp
    push rbx
    sub rsp, 88
    mov QWORD PTR -88[rbp], rdi
    mov DWORD PTR -92[rbp], esi
    mov DWORD PTR -68[rbp], 0
    jmp label_2
label_1:
    mov eax, DWORD PTR -68[rbp]
    cdqe
    mov edx, DWORD PTR -92[rbp]
    mov DWORD PTR -64[rbp+rax*4], edx
    add DWORD PTR -68[rbp], 1
label_2:
    cmp QWORD PTR -68[rbp], 9
    jle label_1
    mov rax, QWORD PTR -88[rbp]
    mov rcx, QWORD PTR -64[rbp]
    mov rbx, QWORD PTR -56[rbp]
    mov QWORD PTR [rax], rcx
    mov QWORD PTR 8[rax], rbx
    mov rcx, QWORD PTR -48[rbp]
    mov rbx, QWORD PTR -40[rbp]
    mov QWORD PTR 16[rax], rcx
    mov QWORD PTR 24[rax], rbx
    mov rdx, QWORD PTR -32[rbp]
    mov QWORD PTR 32[rax], rdx
    mov rax, QWORD PTR -88[rbp]
    add rsp, 88
    pop rbx
    pop rbp
    ret

```

(rdi, rsi, rdx, rcx, r8, r9) (rax, rdi)

