

Problem 4: Linked List Recursion Tracing

Returns linked list of 1, 5, 2, 6, 3

||re Question a: in1 = 1, 2, 3, 4 in2 = 5, 6

✓ ||rec((1,2,3,4), (5,6))

X if

X elseif

else

↳ in1->next = ||rec(in2, in1->next) = 5

[list: 15263]

||rec((5,6), (2,3,4))

X if

X elseif

else

↳ in1->next = ||rec(in2, in1->next) = 2.

[list: 5263]

Ques

Be

func

||rec(2,3,4, 6)

X if

X elseif

else

↳ in1->next = ||rec(in2, in1->next) = 6

[list: 263]

||rec(6, (3,4))

X else

↳ in1->next = ||rec(in2, in1->next) = 3.

[list: 63]

||rec((3,4), null)

else if (in2 = null)

return in1

Problem 4.

Question b: $\text{in1} = \text{nullptr}, \text{in2} = 2, 6$

```
if (rec(nullptr, (5, 6))  
    ✓ if (in1 == nullptr)  
        - return in2  
    [2 is returned]
```

Question a notes: Boxed numbers/words
indicates it was written on the
way out of the recursive function.

Question b notes: Nothing, very simple answer.

Because in1 is null from the start, the
function doesn't get to the recursive call.